Algorithmic aspects of some 2D-Packing problems

Nikolay Kuzyurin
Institute for System Programming RAS (Moscow)

1D-packing problems: examples

- m-processor scheduling (Graham, 1966)
- bin packing

Both problems are NP-hard

Approximation ratio

$$R_A = \sup_I \left\{ \frac{A(I)}{OPT(I)} \right\}$$

m-processor scheduling

given a list L of intervals of sizes $0 < d_i, i = 1, \ldots, n$, pack all d_i in m lines minimizing the maximum sum of sizes in each line.

For $m \geq 2$ the problem is NP-hard.

m-processor scheduling

given a list L of intervals of sizes $0 < d_i$, $i = 1, \ldots, n$, pack all d_i in m lines minimizing the maximum sum of sizes in each line.

For $m \geq 2$ the problem is NP-hard.

2-approximation on-line algorithm ($R_A \leq 2$)
Graham, 1966

on-line vs *off-line* algorithms

m-processor on-line scheduling

Improvements:

$R_A \leq 1.986$ (Bartal, STOC 1992),

$R_A \leq 1.945$ (Karger, SODA 1994),

$R_A \leq 1.923$ (Albers, STOC 1997)

m-processor on-line scheduling

Improvements:

$R_A \leq 1.986$ (Bartal, STOC 1992),

$R_A \leq 1.945$ (Karger, SODA 1994),

$R_A \leq 1.923$ (Albers, STOC 1997)

Lower bounds for on-line algorithms:

$R_A \geq 1.837$ (Bartal, IPL 1994),

$R_A \geq 1.852$ (Albers, STOC 1997)

bin packing: average case analysis :

given a list L of items of sizes $0 < d_i \leq 1, i = 1, \ldots, n$, pack all d_i in minimum number of bins of size 1

$d_i \in U[0,1]$, $w_A(L) = A(L) - s(L)$, where
$A(L)$ is the number of bins used by A, $s(L)$ is the sum of sizes of items in L

bin packing: average case analysis :

given a list L of items of sizes $0 < d_i \leq 1, i = 1, \ldots, n$, pack all d_i in minimum number of bins of size 1

$d_i \in U[0,1]$, $w_A(L) = A(L) - s(L)$, where $A(L)$ is the number of bins used by A, $s(L)$ is the sum of sizes of items in L

$Ew_{BF} = O(n^{1/2}(\log n)^{3/4})$, P. Shor, 1986

$Ew_{FF} = O(n^{2/3})$ Shor, 1986

$Ew_A(L) > cn^{1/2}$ for any on-line algorithm A

Discrete vs continous distributions

Bin Packing with Discrete Item Sizes, Part I: Perfect Packing Theorems and the Average Case Behavior of Optimal Packings, SIAM J. Discrete Math. (2000)
E. G. Coffman, Jr., C. Courcoubetis, M. R. Garey, D. S. Johnson, P. W. Shor, R. R. Weber, and M. Yannakakis

Strip packing problem

Input:

- $I = (R_1, \ldots, R_N)$ — list of rectangles
- i-th rectangle:
  - $h(R_i)$ — height,
  - $w(R_i)$ — width

Objective: Find orthogonal packing of I inside a unit width strip without rotations and intersections so that the height of packing is minimal.

Strip packing problem

Input:

- $I = (R_1, \ldots, R_N)$ — list of rectangles
- i-th rectangle:
  - $h(R_i)$ — height,
  - $w(R_i)$ — width

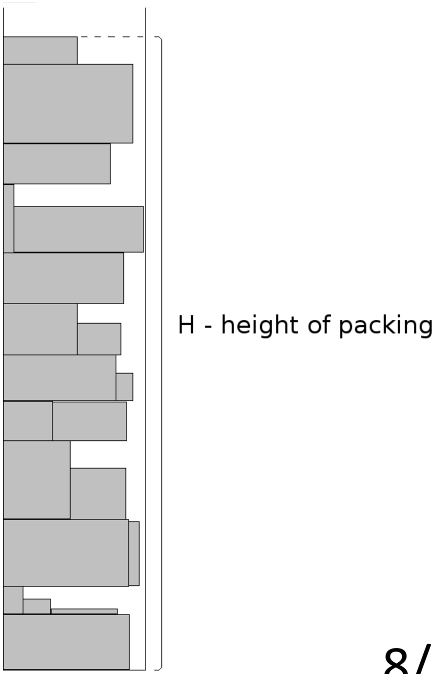
Objective: Find orthogonal packing of I inside a unit width strip without rotations and intersections so that the height of packing is minimal.

Applications

- VLSI design
- Cutting stock problem
- Scheduling of parallel jobs on a cluster

Packing example

N = 20



Strip packing: approximation algorithms

Strip packing is NP-hard (1980)

$\Rightarrow$ Approximation algorithms

Approximation ratio

$$R_A = \sup_I \left\{ \frac{A(I)}{OPT(I)} \right\}$$

Asymptotic approximation ratio

$$R_A^\infty = \lim_{k \to \infty} \sup_I \left\{ \frac{A(I)}{OPT(I)} \mid OPT(I) \geq k \right\}$$

Strip packing: off-line approximation algorithms

FPTAS (Kenyon, Remila, 2000)

$$R_A \leq 1 + \varepsilon + \frac{1}{OPT \cdot \varepsilon^2}$$

M. Sviridenko
Improve the quality of the algorithm

$$R_A \leq 1 + \varepsilon + \frac{\log(1/\varepsilon)}{OPT \cdot \varepsilon}$$

Strip packing: on-line algorithms. Worst case analysis

On-line algorithms with **asymptotic** approximation ratios

1983 Baker, Schwarz, Shelf algorithms, $R_A^\infty \leq 1.7 + \varepsilon$

1997 Csirik, Woeginger $R_A^\infty \leq 1.69103$

2007 Han, Iwama, Ye, Zhang $R_A^\infty \leq 1.58889$

Strip packing: on-line algorithms. Worst case analysis

On-line algorithms with **asymptotic** approximation ratios

1983 Baker, Schwarz, Shelf algorithms, $R_A^\infty \leq 1.7 + \varepsilon$

1997 Csirik, Woeginger $R_A^\infty \leq 1.69103$

2007 Han, Iwama, Ye, Zhang $R_A^\infty \leq 1.58889$

Lower bound

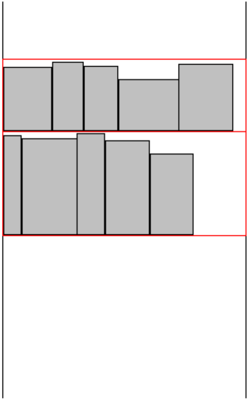
- van Vliet $R_A^\infty \geq 1.54$

The idea of shelf algorithm

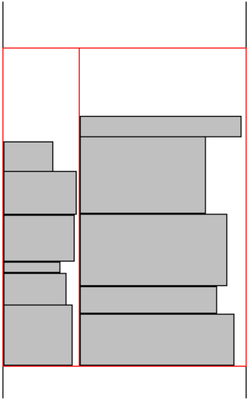
The idea of shelf algorithms is divide the strip on shelfs

The idea of algorithm Iwama et al is to divide the strip on **slips**.

shelfs



slips



Average case analysis of algorithms

Standard probabilistic model: $h(R_i)$, $w(R_i)$ are independent random variables uniformly distributed in $[0, 1]$

Denote uncovered area of a strip as

$$S = H - \sum_i h(R_i) w(R_i)$$

The goal is to minimize $\mathbf{E}\, S$

On-line algorithms: open-end and closed-end

Closed-end on-line algorithm:
we know the number of items in advance

On-line algorithms: open-end and closed-end

Closed-end on-line algorithm:
we know the number of items in advance

Open-end on-line algorithm: we don’t know the number of items in advance

Best known results in terms of average-case analysis

1993 $\mathbf{E}\,S = O(N^{1/2})$ — **Off-line algorithm**, Coffman, Shor.

1993 $\mathbf{E}\,S = O(N^{2/3})$ — **Closed-end on-line algorithm** (the number of rectangles N is known in advance), Coffman, Shor.

Best known results in terms of average-case analysis

1993 $\mathbf{E}\, S = O(N^{1/2})$ — **Off-line algorithm**, Coffman, Shor.

1993 $\mathbf{E}\, S = O(N^{2/3})$ — **Closed-end on-line algorithm** (the number of rectangles N is known in advance), Coffman, Shor.

2010 $\mathbf{E}\, S = O(N^{2/3})$ — **Open-end on-line** (an algorithm does not know the number of rectangles), Kuzyurin, Pospelov.

About shelf algorithms: average case

It is impossible to improve upper bound

$$\mathbf{E}\, S = O(N^{2/3})$$

in the class of shelf on-line algorithms.

New algorithm for closed-end SP

M. Trushnikov[1] proposed new **on-line** algorithm for **closed-end** strip packing.

[1] Master thesis, Lomonosov Moscow State University, 2011

New algorithm for closed-end SP

M. Trushnikov[1] proposed new **on-line** algorithm for **closed-end** strip packing.

Experimentally he showed that

- $\mathbf{E}\, S = CN^{1/2}$

[1] Master thesis, Lomonosov Moscow State University, 2011

Experimental results

N	C
80 000	1.5655
150 000	1.5716
500 000	1.5798
1 000 000	1.5798
4 000 000	1.5878
15 000 000	1.5975
30 000 000	1.5897
100 000 000	1.5934
300 000 000	1.6006
800 000 000	1.5912
1 000 000 000	1.6044
1 500 000 000	1.6027
2 000 000 000	1.5949

The idea of new algorithm (Trushnikov)

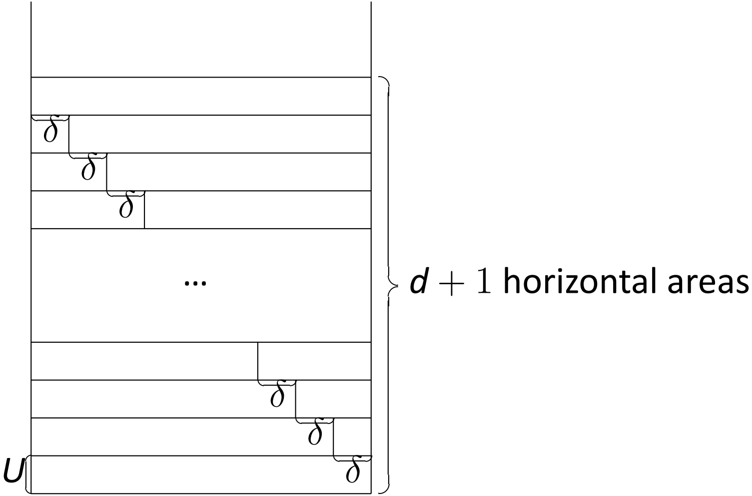
Notations

$$d = \left\lfloor \frac{N/4}{\sqrt{N}} \right\rfloor, \quad \delta = \frac{1}{d}$$

$$U = \frac{N/4}{d} = \sqrt{N} + O(1).$$

At the bottom of the strip we introduce $d+1$ horizontal areas (called containers) each of height U (see the picture below).

Algorithm



Algorithm

Each even rectangle we will pack in the first pyramid and each odd one in the second.

Rectangles which constitute the pyramid we will call **containers**.

Algorithm

Each even rectangle we will pack in the first pyramid and each odd one in the second.

Rectangles which constitute the pyramid we will call **containers**.

Enumerate containers inside the pyramid by numbers from 1 up to d such that the i th one has width $i\delta$.

Rectangles inside containers will be packed one by one: the first at the bottom, next one above the first and so on.

The steps of the Algorithm

Let we obtain as input current rectangle of width w.

- Find i, such that $(i-1)\delta < w \leq i\delta$. We will call this rectangle be *assigned* to the i th container.

The steps of the Algorithm

Let we obtain as input current rectangle of width w.

- Find i, such that $(i-1)\delta < w \leq i\delta$. We will call this rectangle be *assigned* to the i th container.
- Then find minimal j such that $i \leq j \leq d$ and in the j th container it is enough room to pack the rectangle.

The steps of the Algorithm

Let we obtain as input current rectangle of width w.

- Find i, such that $(i-1)\delta < w \leq i\delta$. We will call this rectangle be *assigned* to the i th container.
- Then find minimal j such that $i \leq j \leq d$ and in the j th container it is enough room to pack the rectangle.
- If such j exists we pack the rectangle into the j th container.

The steps of the Algorithm

Let we obtain as input current rectangle of width w.

- Find i, such that $(i-1)\delta < w \leq i\delta$. We will call this rectangle be *assigned* to the i th container.
- Then find minimal j such that $i \leq j \leq d$ and in the j th container it is enough room to pack the rectangle.
- If such j exists we pack the rectangle into the j th container.
- If no, then put the rectangle above current packing.

Such rectangles we will call **unpacked**.

Theorem. The expected wasted area of packing obtained by the Algorithm is

$$\mathbf{E}\, S = \tilde{O}(\sqrt{N}) = O(N^{1/2}(\log N)^{3/2})$$

Outline of the proof

Let Σ is the square of all N rectangles. Obviously $\mathbb{E}\Sigma = N/4$.

The height of the pyramids is

$$(d+1)U = N/4\left(\frac{d+1}{d}\right) = N/4 + \frac{N}{4\lfloor\frac{N/4}{\sqrt{N}}\rfloor} = N/4 + O(N^{1/2}).$$

Outline of the proof

Let Σ is the square of all N rectangles. Obviously $\mathbb{E}\Sigma = N/4$.

The height of the pyramids is

$$(d+1)U = N/4\left(\frac{d+1}{d}\right) = N/4 + \frac{N}{4\lfloor\frac{N/4}{\sqrt{N}}\rfloor} = N/4 + O(N^{1/2}).$$

We will consider only one of the two pyramids and only $\lfloor N/2 \rfloor$ rectangles packed into this pyramid.
Let us enumerate these $\lfloor N/2 \rfloor$ rectangles by numbers from 1 up to $\lfloor N/2 \rfloor$ in the order of arriving rectangles.

Let $\mathbb{M}\{n_1, n_2\}$ be the expectation of the number of **unpacked** rectangles when the Algorithm packs rectangles with numbers from the interval $[n_1, n_2]$

It is sufficient to prove that
$\mathbb{M}\left\{1, \lfloor N/2 \rfloor\right\} = O(N^{1/2}(\log N)^{3/2})$.

Main results

Define two numbers k_0 and k_1:

$$k_0 = \lfloor N/2 \rfloor - \lfloor N^{3/4}\sqrt{\log N} \rfloor, \quad k_1 = \lfloor N/2 \rfloor - \lfloor N^{1/2} \rfloor.$$

Main results

Define two numbers k_0 and k_1:

$$k_0 = \lfloor N/2 \rfloor - \lfloor N^{3/4}\sqrt{\log N} \rfloor, \quad k_1 = \lfloor N/2 \rfloor - \lfloor N^{1/2} \rfloor.$$

Obviously

$$\mathbb{M}\{1, \lfloor N/2 \rfloor\} = \mathbb{M}\{1, k_0\} + \mathbb{M}\{k_0+1, k_1\} +$$
$$\mathbb{M}\{k_1+1, \lfloor N/2 \rfloor\}$$

Main results

Lemma 1. $\mathbb{M}\left\{k_1+1, \lfloor N/2 \rfloor\right\} = O(N^{1/2})$.

Lemma 2. $\mathbb{M}\left\{1, k_0\right\} \to 0, \;\; N \to \infty$,

Lemma 3. $\mathbb{M}\left\{k_0+1, k_1\right\} = O(N^{1/2}(\log N)^{3/2})$

For any rectangle the probability p to be assigned to each container is $1/d$.

For any rectangle the probability p to be assigned to each container is $1/d$.

Let into the pyramid was packed k rectangles. For any fixed container let $X_i,\ 1 \leq i \leq k$ be random variable that is equal to the height of the i th rectangle when it was assigned to this container and 0 otherwise.

For any rectangle the probability p to be assigned to each container is $1/d$.

Let into the pyramid was packed k rectangles. For any fixed container let $X_i,\ 1 \le i \le k$ be random variable that is equal to the height of the i th rectangle when it was assigned to this container and 0 otherwise.

$X_i = \xi_i \eta_i$, where ξ_i — random variable equal 1 with probability p and 0 with probability $1 - p$,
and η_i — iid random variable on $(0, 1]$.
Let $X = X_1 + X_2 + \cdots + X_k$.

Лемма

Let $X = X_1 + X_2 + \cdots + X_k$*, be the random variable such that* $X_i = \xi_i \eta_i$*, (see above) and all variables* $\xi_i, \eta_i, i = 1, \ldots, k$ *are independent.*
Then for any $\alpha \in (0,1)$

$$\mathbb{P}\left\{X > (1+\alpha)\mathbb{E}X\right\} \leq e^{-\frac{5}{9}\alpha^2 \mathbb{E}X}.$$

Лемма

Let the Algorithm already packed $\lfloor N/2 \rfloor - \lfloor N^{1/2+\beta} \rfloor$ rectangles and $0 < \beta < 1/4$. Fix any γ such that

$$1/2 - 2\beta + \frac{\ln(5 \ln N)}{\ln N} \le \gamma < 1/2.$$

Let C be the event: there exists a container from $\lceil N^{\gamma} \rceil$ ***lowest*** *containers of the pyramid with height of packing inside less then $U - 1$. Then for sufficiently large N*

$$\mathrm{P}\{C\} \ge 1 - \frac{1}{N^{1.1}}$$

.

Define partition of $[0, 1/4]$ into $n = \left\lfloor \frac{\ln N}{6 \ln (5 \ln N)} \right\rfloor$ equal parts. Define $\mathbb{M}_i$ as follows

$$\mathbb{M}_i = \mathbb{M}\left\{ \lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}i} \rfloor, \lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}(i-1)} \rfloor \right\}.$$

Define partition of $[0, 1/4]$ into $n = \left\lfloor \frac{\ln N}{6 \ln(5 \ln N)} \right\rfloor$ equal parts. Define $\mathbb{M}_i$ as follows

$$\mathbb{M}_i = \mathbb{M}\left\{ \lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}i} \rfloor, \lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}(i-1)} \rfloor \right\}.$$

$$\mathbb{M}\{k_0 + 1, k_1\} = \sum_{i=1}^{n} \mathbb{M}_i =$$

$$\sum_{i=1}^{n} \mathbb{M}\left\{ \lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}i} \rfloor, \lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}(i-1)} \rfloor \right\}$$

Let us estimate $\mathbb{M}_i$ for $2 \leq i \leq n$.

Apply lemma with parameters

$$\beta = \frac{i-1}{4n},$$

$$\gamma = \frac{1}{2} - \frac{i-1}{3n}.$$

Let us estimate $\mathbb{M}_i$ for $2 \leq i \leq n$.

Apply lemma with parameters

$$\beta = \frac{i-1}{4n},$$

$$\gamma = \frac{1}{2} - \frac{i-1}{3n}.$$

Obviously

$$\gamma \geq \frac{1}{2} - 2\beta + \frac{\ln(5\ln N)}{\ln N}$$

and The Algorithm packed $\lfloor \frac{N}{2} \rfloor - \lfloor N^{1/2+\beta} \rfloor$ rectangles (the conditions of lemma hold).

By the lemma the lowest $\lceil N^{\gamma} \rceil$ containers of the pyramid after packing of $\lfloor \frac{N}{2} \rfloor - \lfloor N^{1/2+\beta} \rfloor$ rectangles cannot be packed all with heights greater than $U - 1$.

By the lemma the lowest $\lceil N^{\gamma} \rceil$ containers of the pyramid after packing of $\lfloor \frac{N}{2} \rfloor - \lfloor N^{1/2+\beta} \rfloor$ rectangles cannot be packed all with heights greater than $U - 1$.

This means that a rectangle can become **unpacked** iff it will be assigned into one of the lowest $\lceil N^{\gamma} \rceil$ containers of the pyramid.

By the lemma the lowest $\lceil N^{\gamma} \rceil$ containers of the pyramid after packing of $\lfloor \frac{N}{2} \rfloor - \lfloor N^{1/2+\beta} \rfloor$ rectangles cannot be packed all with heights greater than $U - 1$.

This means that a rectangle can become **unpacked** iff it will be assigned into one of the lowest $\lceil N^{\gamma} \rceil$ containers of the pyramid.

There are d containers in the pyramid. Thus, the probability that every ractagle from the interval considered can become unpacked is at most $\frac{\lceil N^{\gamma} \rceil}{d}$.

The total number of rectangles in the interval

$$\left(\lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}i} \rfloor, \lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}(i-1)} \rfloor\right)$$

is at most $N^{\frac{1}{2}+\frac{i}{4n}}$.

Thus,

$$\mathbb{M}_i = \mathbb{M}\left\{\lfloor N/2\rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}i}\rfloor, \lfloor N/2\rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}(i-1)}\rfloor\right\} =$$

$$= O\left(\frac{N^{\frac{1}{2}+\frac{i}{4n}} \cdot N^{\frac{1}{2}-\frac{i-1}{3n}}}{N^{1/2}}\right) = O\left(N^{\frac{1}{2}+\frac{-i+4}{12n}}\right).$$

For $i \geq 5$ $\mathbb{M}_i = O(N^{\frac{1}{2}-\frac{1}{12n}})$.

$$\sum_{i=5}^{n} \mathbb{M}_i = O\left(nN^{\frac{1}{2}-\frac{1}{12n}}\right) =$$

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

Moreover

$$\sum_{i=2}^{4} \mathbb{M}_i = O\left(3 \cdot M_2\right) = O\left(N^{\frac{1}{2}+\frac{1}{6n}}\right) =$$

$$O\left(N^{\frac{1}{2}+\frac{\ln 5 \ln N}{\ln N}}\right) = O\left(N^{\frac{1}{2}} \log N\right).$$

For $i = 1$ the number of rectangles in the interval

$$\left(\lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}+\frac{1}{4n}} \rfloor, \lfloor N/2 \rfloor - \lfloor N^{\frac{1}{2}} \rfloor \right)$$

is

$$O\left(N^{\frac{1}{2}+\frac{1}{4n}}\right) = O\left(N^{\frac{1}{2}}(\log N)^{3/2}\right)$$

Finally

$$\mathbb{M}\{k_0+1, k_1\} = \sum_{i=1}^{n} \mathbb{M}_i = O\left(N^{\frac{1}{2}}(\log N)^{3/2}\right).$$

Open questions

Process. The are n enumerated urns, each can contain at most n balls and there are n^2 balls.

At the beginning all urns are empty.

At the current step the current ball goes to any urn with probability n^{-1}.

Process. If the urn is not full (contains less than n balls), the ball will be packed into this urn.

In opposite case it moves to the urn with number less by 1. If it is not full the ball will be packed into this urn, else it moves to the next urn with number less by 1.

Problem

If the ball was moved to the urn with number 1 and the urn is full, the ball is **unpacked**.

Question: *Is it true that the expectation of the unpacked balls is* $O(n)$?

Generalized multiple-strip packing

- MSP: Multiple strip packing problem there are **M** strips of unit width instead of one.
- Generalized MSP (Initially addressed by Zhuk, 2006): There are **M** strips of widths $w_1, \ldots, w_M$,

$$w_1 \geq w_2 \geq \ldots \geq w_M$$

Generalized multiple-strip packing

There are examples of inputs for Generalized MSP such that very natural heuristics give

$$R_A^{\infty} \to \infty$$

Generalized multiple-strip packing

Zhuk proved (2007) for generalized MSP that

- there is an on-line algorithm A

$$R_A^\infty \leq 2e$$

Generalized multiple-strip packing

Zhuk proved (2007) for generalized MSP that

- there is an on-line algorithm A

$$R_A^\infty \leq 2e$$

- For any on-line algorithm A: $R_A^\infty \geq e$

Notations.

Define $A(T)$ as a vector $y = (y_1, \ldots, y_m)$, where y_k is the sum of squares of rectangles from T packed by algorithm A into the k th strip.

$h(T)$ efficiently computable function

$h(T)$ is the lower bound of the height of optimal packing.

An idea of balancing: how to get constant approximation

restriction for small rectangles to go to large strips

Concrete rule: Let a set of rectangles T was packed and

$$A_r(T) = y = (y_1, \ldots, y_m).$$

Next rectangle R will be packed as follows:

An idea of balancing: how to get constant approximation

restriction for small rectangles to go to large strips

Concrete rule: Let a set of rectangles T was packed and

$$A_r(T) = y = (y_1, \ldots, y_m).$$

Next rectangle R will be packed as follows:

1. Compute $h = h(T + \{R\})$.

An idea of balancing: how to get constant approximation

restriction for small rectangles to go to large strips

Concrete rule: Let a set of rectangles T was packed and

$$A_r(T) = y = (y_1, \ldots, y_m).$$

Next rectangle R will be packed as follows:

1. Compute $h = h(T + \{R\})$.
2. Find k, such that

$$k = \max i : w(R) \leq w_i \text{ and } \frac{y_i}{w_i} \leq eh.$$

If such k exists we pack R into the k th strip by shelf algorithm.

Directions for future work

Special cases: all strips have equal widths (MSP)

Directions for future work

Special cases: all strips have equal widths (MSP)

strips have widths of special form (say, powers of 2)

Directions for future work

Special cases: all strips have equal widths (MSP)

strips have widths of special form (say, powers of 2)

strips have constant number of different widths

MSP: on-line vs off-line

Off-line

- AFPTAS, 2009, Bougeret, Dutot, Jansen, Otte, Trystam
- $R_A \leq 2$ 2009, Bougeret, Dutot, Jansen, Otte, Trystam

MSP: on-line vs off-line

Off-line

- AFPTAS, 2009, Bougeret, Dutot, Jansen, Otte, Trystam
- $R_A \leq 2$ 2009, Bougeret, Dutot, Jansen, Otte, Trystam

On-line

- $R_A \leq 3 + \delta_m$, Ye, Han, Zhang, 2009
- $R_A \leq 2.7 + \delta_m$, Ye, Han, Zhang, 2009 randomized on-line algorithm

The case of related machines

We have m one-processor machines, each machine with its own speed and a list of tasks.

Deterministic on-line algorithm with $R_A \leq 8$ and randomized algorithm with $R_A \leq 2e$ (1993)

The case of related machines

We have *m* one-processor machines, each machine with its own speed and a list of tasks.

Deterministic on-line algorithm with $R_A \leq 8$ and randomized algorithm with $R_A \leq 2e$ (1993)

Deterministic on-line algorithm with $R_A \leq 3 + \sqrt{8}$ and randomized algorithm with $R_A \leq 4.31$ (1997)
Berman, Charicar, Karpinsky

The case of related MSP

We have GMSP, each strip with its own 'speed'. This means that if we pack a rectangle into the strip its height decreases proportionally its 'speed'.

Zhuk (2012)
Deterministic on-line algorithm with $R_A^\infty \le 2e$.

Multiple Strip Packing: average case

All strips have equal widths

Our results on average case analysis for MSP

Modified T-algorithm with every new rectangle placed on the emptiest strip and then using Trushnikov's algorithm.

Theorem
$\mathbf{E}\, S_{max} = \tilde{O}(N^{1/2})$ for $M = const$.

Experiments show that $\mathbf{E}\, S_{max} = O(N^{1/2})$ even for $M = N^{1/3}$

$$\mathbf{E}\, S_{max} = CN^{1/2}$$

M	N	C
21	10 000	1.663
34	40 000	1.6415
54	160 000	1.6937
86	640 000	1.7065
136	2 560 000	1.7238
273	20 480 000	1.5822
434	81 920 000	1.6312
547	163 840 000	1.7506
689	327 680 000	1.7396
868	655 360 000	1.6455
1000	1 000 000 000	1.5631

Experiments (average case) for MSP

For $M = N^{1/2}$ average waste grows faster than $N^{1/2}$

M	N	C
200	40 000	3.0043
400	160 000	3.7113
800	640 000	4.8146
1131	1 280 000	5.1267
1600	2 560 000	4.7967
2262	5 120 000	3.9807
3200	10 240 000	5.321
4525	20 480 000	5.4551
6400	40 960 000	7.5701
9050	81 920 000	8.067
12800	163 840 000	9.3379
18101	327 680 000	7.6747
31623	1 000 000 000	16.4354

Resume

- On-line approximation algorithm for MSP
- New closed-end on-line algorithm for strip packing
- It is shown experimentally that $\mathbf{E}\, S = O(N^{1/2})$.
- It is proved that the algorithm provides $\mathbf{E}\, S = \tilde{O}(N^{1/2})$

Future work:

- improve analysis of new algorithm (prove $\mathbf{E}\, S = O(N^{1/2})$
- adapt it to MSP
- adapt it to more general types of distributions
- what about open-end on-line algorithms?