On Efficient Storage Packing

Joint work with

Olga Gerber
Klaus Jansen
Transportation

Minimize the number of bins in your packing!
Minimize the area of your packing!
Minimize the waste in your packing!
Maximize the profit of your packing!
2D Packing

- Strip Packing
  - minimum height
- Bin Packing
  - minimum number of bins
- Storage Minimization
  - minimum area
- Storage Packing
  - maximum profit

All problems are NP-hard!

35 years of research

Over 100 papers, but only a few results on storage packing!
Problem Definition

GIVEN:
• A large rectangle $R$, as $[0,a] \times [0,b]$
• A set $L$ of weighted rectangles
  $R_i \ (i=1, \ldots, n)$ as $[0,a_i] \times [0,b_i]$ and $w_i$
• No rotations

WANTED: A packing of a subset of $L$ into $R$

OBJECTIVE: Maximize the total weight of the packed rectangles
Problem Definition

\[ \text{WEIGHT} = w_1 + w_3 + w_4 + w_5 + w_6 \]
Known Results

- **NP-hard for packing squares into a square**
  

- **An ass. 4/3-app. for packing squares with unit weights into a rectangle.**
  

- **A fast (3+ε)-app. and a very slow (2+ε)-app. for packing weighted rectangles into a rectangle**
  
**Given:** a set \( L \) of \( n \) rectangles with \( a_i, b_i \in (0, 1] \) and weights \( w_i > 0 \).

**Goal:** find a packing of the rectangles of \( L' \) within a unit square \([0,1] \times [0,1] \), s.t. total weight is maximum.

\[
W_\varepsilon(L) \geq (1 - \varepsilon)\text{OPT}
\]
Covering the Maximum Area by Squares

**Given:** a set $Q$ of $n$ squares with side lengths $s_i \in (0, 1]$ and profits = areas.

**Wanted:** find $Q' \subset Q$ and a packing of $Q'$ within $[0,1] \times [0,1]$, s.t. total area of packed squares is maximum.

$$A_\epsilon(Q) \geq (1 - \epsilon)\text{OPT}.$$
Idea:

• Cut a rectangle by horizontal lines into several fractions of equal width.
• Pack fractions independently.
• Sum up fractional weights

Gives a better upper bound than the one by knapsack

Can be used in heuristics as a subroutine!
WEIGHT = w1 + w2 + x1*w3 + x4*w4 + w5 + w6 + w7
Let \( a = 1 \). A configuration is a subset of rectangles \( C \subseteq L \) whose total width is at most 1, i.e., they can occur on the same level.

Let \( \#C \) be the total number of configurations.

For each \( C_j \) \((j = 1, \ldots, \#C)\) we use a variable \( y_j \geq 0 \) whose interpretation is the height of \( C_j \).

So, \( \sum_{j=1}^{\#C} y_j \leq b \) and \( \sum_{j: R_i \in C_j} y_j \geq b_i\).
LP Formulation

maximize \sum_{i=1}^{n} x_i \cdot w_i \\
subject to \sum_{j : R_i \in C_i} y_j \geq x_i \cdot b_i, \text{ for all } i = 1, \ldots, n, \\
\sum_{j = 1}^{\#C} y_j \leq b, \\
y_j \geq 0, \text{ for all } j = 1, \ldots, \#C, \\
x_i \in [0, 1], \text{ for all } i = 1, \ldots, n.

x_i \text{ is a fraction of } R_i

Idea:

Find an approximate solution

Lemma: \text{OPT} \text{ is an upper bound on OPT}
Resource-Sharing Problem

\[
\begin{align*}
\text{maximize} & \quad \lambda \\
\text{subject to} & \quad \sum_{i=1}^{n} x_i \cdot (w_i/w) \geq \lambda, \\
& \quad \sum_{j : R_i \in C_j} [y_j/b_i] - x_i + 1 \geq \lambda, \quad \text{for all } i = 1, \ldots, n, \\
& \quad \sum_{j=1}^{\#C} y_j/b \leq 1, \\
& \quad y_j \geq 0, \quad \text{for all } j = 1, \ldots, \#C, \\
& \quad x_i \in [0, 1], \quad \text{for all } i = 1, \ldots, n.
\end{align*}
\]

\[w \in [w_{\text{max}}, nw_{\text{max}}]\]

**Lemma:** If \(\lambda^* < 1\), then \(w\) is larger than \(\overline{\text{OPT}}\)

**Idea:** search over

\[w \in \{(1 + \varepsilon^2 \cdot \ell)w_{\text{max}} | \ell = 0, 1, \ldots, (n - 1)/\varepsilon^2\}\]

\[w \geq \overline{\text{OPT}} - \varepsilon^2 w_{\text{max}} \geq (1 - \varepsilon^2)\overline{\text{OPT}}\]
Approximate Solutions

\[ \lambda \geq \lambda^*(1 - \bar{\varepsilon}) \]
\[ \sum_{i=1}^{n} x_i \cdot (w_i/w) \geq \lambda, \]
\[ \sum_{j:R_i \in C_j} y_j / b_i - x_i + 1 \geq \lambda, \quad \text{for all } i = 1, \ldots, n, \]
\[ \sum_{j=1}^{\#C} y_j / b \leq 1, \]
\[ y_j \geq 0, \quad \text{for all } j = 1, \ldots, \#C, \]
\[ x_i \in [0, 1], \quad \text{for all } i = 1, \ldots, n. \]

\[ \lambda \geq (1 - \varepsilon) \]
\[ \sum_{i=1}^{n} x_i \cdot w_i \geq (1 - 2\varepsilon)w, \]
\[ \sum_{j:R_i \in C_j} y_j \geq b_i \cdot x_i, \quad \text{for all } i = 1, \ldots, n, \]
\[ \sum_{j=1}^{\#C} y_j \leq b(1 + 2\varepsilon), \]
\[ y_j \geq 0, \quad \text{for all } j = 1, \ldots, \#C, \]
\[ x_i \in [0, 1], \quad \text{for all } i = 1, \ldots, n. \]

Idea: use \( \bar{\varepsilon} \)-approximate solutions in the search
Resource Sharing Problem (RSP)

\[
\begin{align*}
\text{maximize} & \quad \lambda \\
\text{subject to} & \quad f_m(z) \geq \lambda, \text{ for } m = 0, \ldots, M. \\
& \quad z \in B. \\
& \quad f_m(z) \geq (1 - \bar{\epsilon})\lambda^*, \text{ for } m = 0, \ldots, M.
\end{align*}
\]

\[
\sum_{m=0}^{M} p_m = 1 \quad p_m \geq 0 \quad (m = 0, \ldots, M)
\]

\[
\text{maximize} \quad \Lambda(p, z) = \sum_{m=0}^{M} p_m f_m(z) \\
\text{subject to} \quad z \in B.
\]

Theorem [Grigoriadis et al, Jansen]:

If for any \( \bar{t} = \Theta(\bar{\epsilon}) \) and \( p \) with \( p_m = \Omega([\bar{\epsilon}/M]^q) \) can find \((p, \bar{t})\)-app. solution, then also an \( \bar{\epsilon} \)-app. primal solution.

In \( O(M(\ln M + \bar{\epsilon}^{-2} \ln \bar{\epsilon}^{-1})) \) steps and \( O(M \ln \ln (M \bar{\epsilon}^{-1})) \) overhead.
The Block Problem

\[ w \in [w_{\text{max}}, nw_{\text{max}}] \quad \sum_{i=0}^{n} p_i = 1 \quad p_i \geq 0 \ (i = 0, \ldots, n) \]

\[
\max p_0[\sum_{i=1}^{n} x_i (w_i/w)] + \sum_{i=1}^{n} p_i[\sum_{j:R_i \in C_j} y_j/b_i - x_i + 1]
\]

\[ x_i \in [0, 1] \text{ for all } i = 1, \ldots, n \]

\[ \sum_{j=1}^{\#C} y_j/b \leq 1 \text{ and } y_j \geq 0, \text{ for all } j = 1, \ldots, \#C \]

Idea: find a (t,p)-approximate solution.
Approximate Solution

\[ c_i = p_0(w_i/w) - p_i \sum_{j: R_i \in C_j} 1 \]

\[ \text{maximize } \Lambda(p, x) = \sum_{i=1}^{n} c_i \cdot x_i \]

\[ x_i \in [0, 1] \text{ for all } i = 1, \ldots, n. \]

\[ d_j = \sum_{R_i \in C_j} p_i / b_i \]

\[ \sum_{j=1}^{\#C} y_j / b \leq 1, \]

\[ y_j \geq 0, \text{ for all } j = 1, \ldots, \#C. \]

\[ \text{maximize } \Lambda(p, y) = \sum_{j=1}^{\#C} d_j \cdot y_j \]

Lemma 4.1. Let \( x^* \) and \( y^* \) be defined such that

- \( x^*_i = 0 \) if \( c_i \) is non-positive, and \( x^*_i = 1 \) otherwise \((i = 1, \ldots, n)\),

- \( y^*_i = b \), and \( y^*_j = 0 \) for all \( C_j \neq C_k \) \((j = 1, \ldots, \#C)\), where \( C_k \) is a configuration with \( d_k = \max_{j=1}^{\#C} d_j \).

Then, \( x^* \) and \( y^* \) define an optimal solution for the block problem.

Lemma 4.4. Let \( T \) be some positive value. Then, for any price vector \( p \) whose positive coordinates \( p_i = \Omega(1/T) \) \((i = 0, \ldots, n)\) and any accuracy \( \bar{t} > 0 \), there is a block solver algorithm BSA\((p, \bar{t})\) which finds a \((p, \bar{t})\)-approximate solution for the block problem in \( O(n^2 \cdot T) + KS(n, \bar{t}) \) time.
Overall Algorithm

Step 1: Define $\bar{\varepsilon} = \varepsilon^2/n$.

Search $\bar{\varepsilon}$-app. solutions of $n/\varepsilon^2$ instances of RSP

Find packing within $[0, 1] \times [0, (1 + 2\varepsilon)b]$

The weight sums to at least $(1 - 3\varepsilon)OPT$

Step 2: Partition the packing into $1/(2\varepsilon)$ groups.

Drop the one of minimum weight.

Get a packing within $[0, 1] \times [0, b]$

of weight at least $(1-5\varepsilon)OPT$
Conclusions

• A wide range of applications
• A large number of algorithms known

Efficient Packing Algorithms (Liverpool-Kiel)
• Software package
• Website with all test results
• New fast algorithmic solutions