Lecture 10. The Bin Packing Problem

The one dimensional bin packing problem is defined as follows. Given a set $L = \{1, \ldots, n\}$ of items and theirs weights $w_i \in (0,1), i \in L$. We wish to partition the set $L$ into minimal number $m$ of subsets $B_1, B_2, \ldots, B_m$ in such a way that

$$\sum_{i \in B_j} w_i \leq 1, \ 1 \leq j \leq m.$$ 

The sets $B_j$ we will call bins.

In other words, we wish to pack all items in a minimal number of bins.

It is NP-hard problem in the strong sense.
Mathematical Model

Decision variables:

\[ y_j = \begin{cases} 1 & \text{if bin } j \text{ is used} \\ 0 & \text{otherwise} \end{cases}; \quad x_{ij} = \begin{cases} 1 & \text{if item } i \text{ is in bin } j \\ 0 & \text{otherwise} \end{cases} \]

\[
\min \sum_{j=1}^{n} y_j
\]

s.t.

\[
\sum_{i=1}^{n} w_i x_{ij} \leq y_j, \quad j = 1, \ldots, n;
\]

\[
\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1, \ldots, n;
\]

\[
y_i, x_{ij} \in \{0,1\}, \quad i, j = 1, \ldots, n.
\]

Can we find the optimal solution for the linear programming relaxation in polynomial time?
Bad News

- There is much symmetry in the model.
- The problem is hard to approximate.

**Theorem 1.** The existence of a polynomial time \( \left( \frac{3}{2} - \varepsilon \right) \)-approximation algorithm for any positive \( \varepsilon \) implies \( P = NP \).

**Proof.** Let us consider the following NP-complete problem. Given \( n \) positive numbers \( a_1, \ldots, a_n \). Is it possible to partition this set into two subsets \( A_1, A_2 \) in such a way that \( \sum_{i \in A_1} a_i = \sum_{i \in A_2} a_i \) ?

We put \( C = \frac{1}{2} \sum_{i=1}^{n} a_i \), \( w_i = \frac{a_i}{c} \), \( i = 1, \ldots, n \), and apply our \( \left( \frac{3}{2} - \varepsilon \right) \)-approximation algorithm. If we get 2 bins, then answer is Yes, otherwise No. It is **exact** answer! ■
**Strong Heuristic (FFD)**

Rank the items by the weights:

\[ w_1 \geq w_2 \geq \ldots \geq w_n \]

and apply the First Fit strategy:

- put the first item in the first bin;
- at the step \( k \), we try to put item \( k \) into the used bins and, if it is not possible, we put item \( k \) into a new bin.

**Theorem 2.** \( FFD(L) \leq \frac{11}{9} \OPT(L) + 4 \) for all \( L \) and there exist some instances for the bin packing problem with

\[ FFD(L) \geq \frac{11}{9} \OPT(L). \]
**Hard Example**

\[ L = \{1, \ldots, 30m\} \]

\[
\begin{align*}
    w_i = & \begin{cases} 
        \frac{1}{2} + \epsilon, & 1 \leq i \leq 6m \\
        \frac{1}{4} + 2\epsilon, & 6m < i \leq 12m \\
        \frac{1}{4} + \epsilon, & 12m < i \leq 18m \\
        \frac{1}{4} - 2\epsilon, & 18m < i \leq 30m
    \end{cases}
\end{align*}
\]

\[ \text{OPT}(L) = 9m \]

\[ FFD(L) = 11m \]
Huge Reformulation

**Given**  
$L = \{1, \ldots, n\}$ is the set of items;  
$w_i > 0$ is the weight of item $i$;  
$n_i > 0$, integer, is the number of identical items $i$  
$a_{ij}$ is the number of identical items $i$ in packing pattern $j$.

**Find**  
a partition of all items into a minimal number of bins.

**Variables:**  
$x_j \geq 0$, integer, is the number of bins for the pattern $j$

$$
\begin{align*}
\min & \sum_{j \in J} x_j \\
\text{s.t.} & \sum_{j \in J} a_{ij}x_j \geq n_i, \ i \in L; \\
& x_j \geq 0, \text{ integer, } j \in J.
\end{align*}
$$

$J$ is the set of all possible patterns.
**LP-Based Heuristic**

Solve the linear programming relaxation

\[
\begin{align*}
\min & \quad \sum_{j \in J} x_j \\
\text{s.t.} & \quad \sum_{j \in J} a_{ij}x_j \geq n_i, \quad i \in L; \\
& \quad x_j \geq 0, \quad j \in J.
\end{align*}
\]

Put \( x_j = [x_j^*], \quad j \in J \). It is a feasible solution with deviation from the optimum at most

\[
\varepsilon = \frac{\sum_{j \in J} ([x_j^*] - x_j^*)}{\sum_{j \in J} x_j^*}
\]

where \( x_j^* \) is the optimal solution for the LP model.

*Can we solve LP?*
The Column Generation Method

Let us consider a subset $J' \subset J$ of patterns and assume that the following subproblem

$$\min \sum_{j \in J'} x_j$$

s.t.

$$\sum_{j \in J'} a_{ij} x_j \geq n_i, \quad i \in L;$$

$$x_j \geq 0, \quad j \in J';$$

has at least one feasible solution.

Denote by $x_j^*$ the optimal solution to this subproblem.
The Dual Problem

\[
\begin{align*}
\max & \sum_{i \in L} n_i \lambda_i \\
\sum_{i \in L} a_{ij} \lambda_i & \leq 1, \ j \in J'; \\
\lambda_i & \geq 0, \ i \in L.
\end{align*}
\]

Denote by \( \lambda_i^* \geq 0 \) its optimal solution. If

\[
\sum_{i \in L} a_{ij} \lambda_i^* \leq 1, \ \text{for} \ j \in J \setminus J';
\]  

then

\[
\bar{x}_j = \begin{cases} 
  x_j^*, & j \in J' \\
  0, & j \in J \setminus J'
\end{cases}
\]

is the optimal solution for the LP problem.
How to Check (*)?

Let us consider the following knapsack problem:

\[ \alpha = \max \sum_{i \in L} \lambda_i^* y_i \]

s.t. \[ \sum_{i \in L} w_i y_i \leq 1; \] (capacity of bin)

\[ y_i \geq 0, \text{ integer, } i \in L. \]

If \( \alpha \leq 1 \) then (*) is satisfied.

If \( \alpha > 1 \) then we have got a new pattern and include it in \( J' \).
The Framework of the Method

1. Select an initial subset $J' \subset J$.
2. Solve the subproblem for $J'$ and its dual one, get $x^*_j, \lambda^*_i$.
3. Solve the knapsack problem for $\lambda^*$ and compute $\alpha$.
4. If $\alpha \leq 1$ then STOP.
5. Include new pattern $j_0$: $a_{ij_0} = y^*_i$, $i \in L$, into subset $J'$ and goto 2.

**Surprise:** As a rule, solution $x_j = \lfloor x^*_j \rfloor$, $j \in J$ is optimal for the bin packing problem. If it is not true, we have at most one additional bin only!
Two-Dimensional Packing Problem

Given: \( n \) rectangles with size \( w_i \times l_i, \ i = 1, ..., n \).

Find: a packing of the rectangles into a rectangle area with minimal square.

Rotations are forbidden

\[ L \times M \rightarrow \min \]

It is guillotine solution.
The Strip Packing Problem

Дано: \( n \) rectangles with size \( w_i \times l_i, \ i \in L \), and large strip with width \( W \).

Find: a packing of rectangles into the strip with minimal length.

For \( l_i = 1 \) we have one-dimentional bin packing problem (NP-hard)

Hometask. Design a linear integer programming model for the strip packing problem (with and without 90° rotations).
The Two-Dimensional Knapsack Problem

**Given:** \( n \) rectangles with size \( w_i \times l_i \), profit \( c_i \) for each rectangle, and the size of a vehicle \( W \times L \).

**Find:** a subset of rectangles with maximal total profit which can be packed into the vehicle.

For \( l_i = L \), we have the classical knapsack problem.

**Hometask.** Design a linear integer programming model for the two-dimensional knapsack problem.
The Two-Dimensional Bin Packing Problem

**Given:** $n$ rectangles with size $w_i \times l_i$ and the size of a vehicle $W \times L$.

**Find:** a packing all rectangles into the minimal number of vehicles.

**Hometask.** Design LP-based heuristic for the two-dimensional bin packing problem.