高等演算法 HW1

許博翔

March 21, 2024

Problem 1. If $c_1 + (n-1)c_m > B$, then no worker can select the first item, and we can remove all $p_{11}, p_{21}, \ldots, p_{n1}$, so let's suppose that $c_1 + (n-1)c_m \le B$.
Let $b = \frac{p_{11}\epsilon}{2n}$, and $p'_{ij} := \lceil \frac{p_{ij}}{b} \rceil b$, which we'll call it "new productivity".
Let $q_{ij} := \frac{p'_{ij}}{b} = \lceil \frac{p_{ij}}{b} \rceil$.
Let dp_{ij} := the minimum cost that can achieved with new productivity jb by $W_1, W_2, \ldots, W_i$, and r_{ij} the item that W_i should select to achieve such minimum cost. The range: $1 \le i \le n,\ 0 \le j \le Q$, $Q := \sum_{k=1}^{n} q_{k1}$.
The base case $i = 1$:

$$dp_{1j} = \begin{cases} \min\limits_{k: q_{1k} = j} (c_k), \text{ if } \exists k \text{ s.t. } q_{1k} = j \\ B+1, \text{ otherwise} \end{cases}$$

$$r_{1j} = \begin{cases} k, \text{ where } c_k = dp_{1j} \text{ and } q_{1k} = j, \text{ if such } k \text{ exists} \\ -1, \text{ otherwise} \end{cases}$$

One can run from $i = 2$ to n, from $j = 0$ to Q to get the values of dp_{ij} using

$$dp_{ij} = \begin{cases} \min\limits_{k: q_{ik} \le j} (dp_{i-1, j-q_{ik}} + c_k), \text{ if } \exists k \text{ s.t. } q_{ik} \le j \\ B+1, \text{ otherwise} \end{cases}$$

$$r_{ij} = \begin{cases} k, \text{ where } dp_{i-1, j-q_{ik}} + c_k = dp_{ij} \text{ and } q_{ik} \le j, \text{ if such } k \text{ exists} \\ -1, \text{ otherwise} \end{cases}$$

Denote the optimal solution as ALG, and the value of ALG (denote as $val'(ALG)$ is the maximum new productivity that can be achieved with cost at most B, which is $\max\limits_{j:dp_{nj}\le B}(jb)$, and we can recursively find the selected item that can achieve this using r_{ij}.

The above can be done in $O(nQm)$ time complexity.

Let j_i denote the selected item by W_i in ALG, and let $\sum\limits_{i=1}^{n} p_{ij_i}$ be the productivity value of this algorithm (denoted as $val(ALG)$).

Let k_i denote the selected item by W_i in OPT, and let the new productivity value of these selected item be $val'(OPT)$.

By the definition of OPT, $val(ALG) \le val(OPT)$.

Since the above dp algorithm obtains optimal solution of new productivity, $val'(ALG) \ge val'(OPT)$.

$val(ALG) = \sum\limits_{i=1}^{n} p_{ij_i} > \sum\limits_{i=1}^{n}(\lceil \frac{p_{ij_i}}{b} \rceil - 1)b = val'(ALG) - nb \ge val'(OPT) - nb = \sum\limits_{i=1}^{n} \lceil \frac{p_{ik_i}}{b} \rceil b - nb \ge \sum\limits_{i=1}^{n} p_{ik_i} - nb = val(OPT) - nb = val(OPT) - \frac{p_{11}\epsilon}{2}$.

By what we suppose in the first line, $p_{11} \le p_{11} + p_{2m} + p_{3m} + \cdots + p_{nm} \le val(OPT)$ (since W_1 can select 1, while $W_2, W_3, \ldots, W_n$ select m).

$\Rightarrow val(ALG) \ge val(OPT) - \frac{val(OPT)\epsilon}{2} = (1 - \frac{\epsilon}{2})val(OPT)$.

$\Rightarrow val(ALG) \le val(OPT) \le \frac{val(ALG)}{1-\frac{\epsilon}{2}} \le (1+\epsilon)val(ALG)$.

The time complexity of this algorithm $= O(nQm) = O(n\sum\limits_{k=1}^{n} q_{k1}m) = O(n\sum\limits_{k=1}^{n} \lceil \frac{p_{k1}}{b} \rceil m) = O(n\sum\limits_{k=1}^{n} \lceil \frac{2np_{k1}}{p_{11}\epsilon} \rceil m) = O(n\sum\limits_{k=1}^{n} \frac{2n}{\epsilon} m) = O(\frac{n^3 m}{\epsilon})$

Problem 2. Let $p_j := p_{1j} = p_{2j} = \cdots = p_{nj}$.

First, there is a 4-approximation.

Let $k = \min\limits_{i:p_{4i+1}+p_{4i+2}+p_{4i+3}+p_{4i+4}<P}(i)$.

That is, for $i = 1, 2, \ldots, k$, $p_{4i+1} + p_{4i+2} + p_{4i+3} + p_{4i+4} \ge P$.

Since $p_{4k+1} \ge p_{4k+2} \ge \cdots \ge p_m$, for all 4 distinct elements a, b, c, d of the multiset $\{p_{4k+1}, p_{4k+2}, \ldots, p_m\}$, $a + b + c + d \le p_{4i+1} + p_{4i+2} + p_{4i+3} + p_{4i+4} < P$.

Problem 3.

Problem 4.