Exercises

Approximation Algorithms

Spring 2010

Sheet 5

Note: This is just one way, a solution could look like. We do not guarantee correctness. It is your task to find and report mistakes.

Exercise 1
Consider MULTI CONSTRAINT KNAPSACK where \( n \) objects with profits \( p_i \in \mathbb{Q}_+ \) and budget requirement \( a_i^j \in [0, 1] \) are given. Let

\[
OPT = \max_{I \subseteq \{1, \ldots, n\}} \left\{ \sum_{i \in I} p_i \left| \sum_{i \in I} a_i^j \leq 1 \forall j = 1, \ldots, k \right. \right\}
\]

(since after scaling we may assume that the available budgets are all 1). We consider the number \( k \) of budgets to be a fixed constant. Show that for any \( \varepsilon > 0 \) one can compute in time polynomial in \( n \) and \( 1/\varepsilon \) a solution \( I \subseteq \{1, \ldots, n\} \) with profit \( \sum_{i \in I} p_i \geq OPT \) which is nearly feasible, i.e. \( \sum_{i \in I} a_i^j \leq 1 + \varepsilon \) for all \( j = 1, \ldots, k \).

Hints: Find a suitable rounding + dynamic programming.

Solution:
Suppose \( \frac{1}{\varepsilon} \in \mathbb{Z} \). Round all \( a_i^j \) down to the nearest multiple of \( \frac{\varepsilon}{n} \) and denote the new numbers with \( \bar{a}_i^j \in \mathbb{Z} \frac{\varepsilon}{n} \).

Let \( OPT' \) be the best value of the new MCK instance. Due to the rounding down \( OPT' \geq OPT \).

For any \( b_1, \ldots, b_k \in \{0, \frac{\varepsilon}{n}, 2\frac{\varepsilon}{n}, \ldots, 1\} \) consider table entries

\[
A(m,b_1,\ldots,b_k) = \text{best profit from a subset of items } \{1,\ldots,m\} \text{ such that budget of type } j \text{ is } \leq b_j \forall j = 1,\ldots,k
\]

\[
= \max_{I \subseteq \{1,\ldots,m\}} \left\{ \sum_{i \in I} p_i \left| \sum_{i \in I} \bar{a}_i^j \leq b_j \forall j = 1,\ldots,k \right. \right\}
\]

Clearly the number of table entries is \( O(n^{k+1}/\varepsilon^k) \). We compute a single entry by

\[
A(m,b_1,\ldots,b_k) = \max \left\{ A(m-1,b_1,\ldots,b_k), A(m-1,b_1-\bar{a}_m,\ldots,b_k-\bar{a}^k_m) + p_m \right\}
\]

We can compute each entry in \( O(1) \). Finally the solution \( I \) leading to

\[
OPT' = A(n,1,\ldots,1)
\]

can be easily reconstructed. It has profit \( \sum_{i \in I} p_i = OPT' \geq OPT \). It might violate some budgets, but

\[
\sum_{i \in I} a_i^j \leq \sum_{i \in I} (\bar{a}_i^j + \frac{\varepsilon}{n}) \leq \sum_{i \in I} \bar{a}_i^j + n \cdot \frac{\varepsilon}{n} \leq 1 + \varepsilon
\]
Exercise 2
For the Bin Covering problem, we are given an instance $I = (a_1, \ldots, a_n)$ with items $a_i \in [0, 1]$ and aim at maximizing the number of bins, which are covered (that means the size of the assigned items is at least 1):

$$OPT = \max \left\{ k \mid \exists I_1 \cup \cdots \cup I_k = \{1, \ldots, n\} : \forall j : \sum_{i \in I_j} a_i \geq 1 \right\}$$

We assume that $a_i \geq \delta, i = 1, \ldots, n$ for a constant $\delta > 0$. Give an asymptotic PTAS for this problem under the above assumption (i.e. a polynomial time algorithm that for any fixed $\varepsilon > 0$, covers at least $(1 - \varepsilon)OPT - O(1)$ bins).

**Hint:** Adapt the APTAS of Fernandez de la Vega & Lueker from the lecture. In the grouping, you should round down the item sizes (instead of rounding them up).

**Solution:**

We apply the following grouping strategy

- **INPUT:** Instance $I = (a_1, \ldots, a_n), k \in \mathbb{N}$

- **OUTPUT:** Instance $I' = (d_1, \ldots, d_n)$ with $d_i \leq a_i$ and $\leq k$ different item sizes

Suppose for simplicity $n/k \in \mathbb{Z}$.

1. Sort $a_1 \leq a_2 \leq \cdots \leq a_n$

2. Partition items into $k$ consecutive groups of $[n/k]$ items (the last group might have less items)

3. Let $d_i$ be the size of the smallest item in $i$’s group

First of all, any solution for $I'$ gives a solution to $I$ with the same number of bins. We claim that the other way around

**Claim:** $OPT(I') \geq OPT(I) - \lceil n/k \rceil$.

**Proof:** To see this consider any solution to $I$. For every item in group $j$, we find an item of $I'$ in group $j + 1$ (which is at least as big). This way we again cover all bins that did not contain any item from the last group. But these are at most $\lfloor n/k \rfloor$ many.

**Claim:** For an instance $I'$ with $a_i \geq \delta$ and $1/\varepsilon^2$ many different item sizes, one can find an optimum Bin Covering solution in polynomial time (considering $\varepsilon, \delta$ as constant).

**Proof:** In the optimum solution w.l.o.g. no pattern contains more than $2/\delta$ items, since then one could split it into 2 bins, each containing $1/\delta$ items of size $\geq \delta$ each. Hence there are $O((1/\varepsilon^2)^{1/\delta})$ many different patterns and $n^{O((1/\varepsilon^2)^{1/\delta})}$ possible solutions that one has to check.

We consider the following algorithm

1. Choose $k := 1/\varepsilon^2$

2. Apply linear grouping to $I \rightarrow I'$

3. Find an optimum solution to $I'$

Assume $\varepsilon \leq \delta$. Clearly $OPT \geq \frac{\delta n}{2} \geq \frac{n}{2}$ (again since $a_i \geq \delta$ and in $OPT$ never more than $2/\delta$ items are assigned to the same bin). Using the grouping we produce a solution of value

$$OPT(I') \geq OPT - \lceil \varepsilon^2 n \rceil = OPT - \left[ 2\varepsilon \cdot \frac{\varepsilon n}{2} \right] \geq (1 - 2\varepsilon)OPT - 1.$$