Approximation Algorithms, exercise sheet 5

November 18, 2013

1. The simple knapsack problem

Consider the greedy algorithm for the simple knapsack problem (the version where the profit equals the weight) that was treated in the lecture. Prove that it does not approximate better than with a factor 2.

Suppose we change the algorithm in the following way. We choose a fixed $k \in \mathbb{N}$. First we sort the a_s from heavy to light, i.e, $a_1 \geq a_2 \geq \cdots \geq a_n$. Then we compute a set $T \subset \{a_1, \ldots, a_k\}$ such that $\text{cost}(T) \leq B$ is maximal. Finally, for $i = k+1, \ldots, n$ we add a_i to T if $\text{cost}(T \cup \{a_i\}) \leq B$. Does this change the approximation factor?

2. Approximation of the general knapsack problem

Consider the following algorithm. First sort the elements such that $\frac{c_1}{a_1} \geq \cdots \geq \frac{c_n}{a_n}$. Let $T := \emptyset$ and for $i = 1, \ldots, n$, if $\text{cost}(T) + a_i \leq B$ then $T := T \cup \{i\}$.

What can you say about how well this algorithm approximates the optimal solution?

Suppose we change this algorithm a little. After we have executed it, we search for the largest c_i. We compare the solutions T and $\{i\}$ and output the better one.

Does this improve things?