Approximation Algorithms
Winter term 2021/22

Prof. Dr. Martin Hoefer
Tim Koglin, Lisa Wilhelmi

GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institute of Computer Science
Algorithms and Complexity

Assignment 7

Issued: 07.12.2021
Due: 14.12.2021, **10:15h**

This is the last assignment that counts for Part 1 of the course. Its scope is halved again due to organizational reasons. It will be discussed in the exercise session on Friday, Dec 17th.

Exercise 7.1 *LP for* KNAPSACK (2 + 2 + 2 points)

In the (integral) KNAPSACK problem, a set $[n]$ of items is given. Each item $i \in [n]$ has a profit $w_i \geq 0$ and a size $g_i \geq 0$. The size bound of the knapsack is $G \geq 0$ (w.l.o.g. $g_i \leq G$ for every item i). A subset $S \subseteq [n]$ is feasible if its overall size, i.e., the sum over the sizes of its elements, does not exceed G. The goal is to find a feasible set with maximal overall profit.

a) Formulate an LP for the *fractional* KNAPSACK problem, in which items can be packed in arbitrary fractions $x_i \in [0, 1]$ into the knapsack.

For every item $i \in [n]$, define its *profit density* by $w_i/g_i \geq 0$. Based on this quantity, an optimal solution for the fractional KNAPSACK instance can be obtained as follows:

1. Order all items in a non-increasing fashion with respect to their profit densities.
2. According to this order, add all items to the solution S consecutively until the first element i' is found that would exceed the size bound G, i.e., $g_{i'} + \sum_{j \in S} g_j > G$.
3. Include a fraction $x_{i'}$ of item i' such that the overall size of solution S is exactly G. The fraction is defined by $x_{i'} = (G - \sum_{j \in S} g_j)/g_{i'}$.

This suggests a simple Greedy algorithm *MaxDen* for the fractional KNAPSACK problem that performs the first two steps of the above procedure (and thus returns S as solution, without adding i' fractionally). Note that *MaxDen* also computes a solution for the (integral) KNAPSACK problem.

b) Show that the approximation ratio of *MaxDen* cannot be upper bounded by any constant.
Hint: It is sufficient to consider an instance where $n = 2$.

c) A slight modification of *MaxDen* significantly improves the approximation ratio: Compute the solution S of *MaxDen* and determine the associated item i' (compare step 3). Then, return $S^* \in \arg\max \{\sum_{j \in S} w_j, w_{i'}\}$ as final solution.

Show that this modification yields a 2-approximation for the (integral) KNAPSACK problem.

Exercise 7.2 *Deterministic Rounding* (4 points)

In the Weighted Set Cover problem, a set of elements $E := [m]$ and a family of subsets of E is given. For every $j \in [n]$, the subset $S_j \subseteq E$ is assigned to a weight $w_j \geq 0$. The goal is to compute a selection of indices $C \subseteq [n]$ with minimal sum of weights such that, for each element $i \in E$, there exists a $j \in C$ with $i \in S_j$. Assume that any element $i \in E$ is contained in exactly $f \in \mathbb{N}_{>0}$ subsets.

Design an algorithm for Weighted Set Cover using deterministic rounding. Show both correctness and that the approximation ratio of the algorithm is at most f.

The assignments and further information on the course are provided on our website:
https://algo.cs.uni-frankfurt.de/lehre/apx/winter2122/apx2122.shtml

Contacts: {koglin,wilhelmi}@em.uni-frankfurt.de.