Prof. Nicole Megow
Dr. Syamantak Das

Summer 2017

Approximation Algorithms

Exercise Sheet 6

Exercise 1 (*15 points*) Consider the following algorithm for the 0/1-knapsack problem. Order the items as $\{1, 2, \cdots n\}$ such that $v_1/s_1 \geq v_2/s_2 \geq v_3/s_3 \cdots v_n/s_n$. Further, let $i^\star = \text{argmax}_i\, v_i$ - that is the element $i^\star$ has maximum value. Now add items greedily till element k according to the ordering such that adding $k+1$ makes the knapsack overflow. Return either $\{1, 2, \cdots k\}$ or the singleton $\{i^\star\}$, whichever has larger value.

a) (*5 points*) Prove that this gives a 1/2-approximation

b) (*10 points*) Recall that while designing the FPTAS for knapsack, we ensured by scaling that the upper and lower bound on the optimal are within a factor of n of each other. Use a refined approximation scheme using part a) that eliminates a factor n from the running time of the algorithm.

Exercise 2 (*20 points*) Consider the following scheduling problem. There are n jobs to be scheduled on a single machine. Each job j has a processing size of $p_j \geq 0$, weight $w_j \geq 0$ and a deadline d_j by which the job must be completed. Further, once started, the job cannot be interrupted till completion. The task is to find a schedule for all jobs such that the total weight of jobs that finish by their deadlines is maximized.

a) (*5 points*) Prove that there exists an optimal schedule in which all on-time jobs are scheduled before all late-jobs. Further, prove that the on-time jobs are scheduled in earliest deadline first ordering

b) (*10 points*) Use the above structure to design a dynamic programming optimal algorithm for the problem that runs in $\mathcal{O}(nW)$ time where $W = \sum_{j=1}^{n} w_j$.

c) (*5 points*) Use the above to design a PTAS (or even better an FPTAS) for the problem.