Assignment 11

Post Date: 18 Jan 2016  Due Date: 25 Jan 2016, 1 pm
You are permitted and encouraged to work in groups of two.

Problem 1: Capacity Scaling  5 Points

Apply the Capacity Scaling algorithm to the min-cost flow problem below.

Problem 2: Truck Loading  5 Points

Suppose a ship arrives with \( n \) containers of positive weight \( w_1, \ldots, w_n \). You want to load the containers into trucks, each of which can hold cargo of weight at most \( K \geq \max_i w_i \).

You apply the following scheme: Start with an empty truck and fill it with containers 1, 2, \ldots until the next container does not fit into the truck. Then send this truck off and continue the process by filling the next truck. Iterate this scheme until all containers were loaded into trucks.

Show that the number of trucks used by this scheme is at most double the minimum number of trucks possible.

[Please turn over]
Problem 3: Approximating Knapsack 10 Points

In the course of your recent software project, you realize that you have to solve the Knapsack problem good enough. Your colleague proposes the following simple algorithm that supposedly produces a good approximation to the Knapsack problem:

Algorithm 1: Approximation Algorithm for Knapsack

Input: positive integers $w_1, \ldots, w_n$, $v_w, \ldots, v_n$, $W$ such that $v_1/w_1 \geq \cdots \geq v_n/w_n$

Output: A feasible solution $I \subseteq \{1, \ldots, n\}$

$I \leftarrow \emptyset$;

for $i = 1, \ldots, n$ do

if $\sum_{j \in I \cup \{i\}} w_i \leq W$ then

$I \leftarrow I \cup \{i\}$;

for $i = 1, \ldots, n$ do

if $w_i \leq W$ and $v_i > v(I)$ then

$I \leftarrow \{i\}$;

You want to understand this algorithm. First, you study if it’s possible to optimize this algorithm.

(a) Suppose you omit the first for loop of Algorithm 1. Show that such an algorithm is not a $k$-approximation algorithm for any $k \in \mathbb{N}_+$. 

(b) Suppose you omit the second for loop of Algorithm 1. Show that such an algorithm is not a $k$-approximation algorithm for any $k \in \mathbb{N}_+$. 

Next, you try to analyze the quality of the approximation produced by the algorithm. For this, you consider the following continuous variant of the Knapsack problem:

Instance: weights $(w_1, \ldots, w_n) \in \mathbb{N}^n$, values $(v_1, \ldots, v_n) \in \mathbb{N}^n$, maximum weight $W \in \mathbb{N}$.

feasible solutions: vectors $I = (x_1, \ldots, x_n) \in [0, 1]^n$ with $\sum_{i=1}^n x_i w_i \leq W$.

value of $I$: $v(I) = \sum_{i=1}^n x_i v_i$

optimum solution: a feasible solution with maximum value.

(c) Suppose you are given an instance of the continuous Knapsack problem such that $v_1/w_1 \geq \cdots \geq v_n/w_n$. Show that there exists an integer $k \in \{1, \ldots, n\}$ and an optimum solution $(x_1, \ldots, x_n)$ such that $x_i = 1$ for $1 \leq i < k$, $0 \leq x_k \leq 1$, and $x_i = 0$ for $n \geq i > k$.

(d) Show that the unmodified Algorithm 1 is a 2-approximation algorithm.

Hint: You may use (c), even if you did not prove it.

(e) Is Algorithm 1 a $k$-approximation algorithm for some $k < 2$? Prove your claim.