Combinatorial Optimization 2
Summer Term 2019
sixth work sheet

34. The Bin Packing Problem (BPP) is defined as follows. An instance I consists of a list of nonnegative numbers a_1, a_2,..., a_n not larger than 1. The task is to find a pair (k, f) with a number $k \in \mathbb{N}$ and a mapping $f\colon \{1, 2, \dots, n\} \to \{1, 2, \dots, k\}$ such that $\sum_{i: f(i)=j} a_i \leq 1$ for all $j \in \{1, 2, \dots, k\}$ and k is the smallest natural number for which such a mapping f exists. The smallest k as above is called the optimal value of the problem for instance I and is denoted by $OPT(I)$.

    The Next-Fit (NF) Algorithm for BPP works with two quantities k and S initialized by $k := 1$ and $S := 0$. Then, for $i = 1$ to n, it checks whether $S + a_i > 1$ and sets $k := k + 1$, $S := 0$ in the "yes" case. It further sets $f(i) := k$ and $S := S + a_i$.

    For an instance I of BPP consisting of the n numbers $a_i \in [0, 1]$, $i \in \{1, 2, \dots, n\}$, denote by $SUM(I)$ the sum $\sum_{i=1}^{n} a_i$. Further denote by $NF(I)$ the output k of the NF algorithm applied to instance I. Show that the following inequalities hold:

$$NF(I) \leq 2\lceil SUM(I) \rceil - 1 \leq 2OPT(I) - 1\,.$$

35. Let k be fixed. Describe a pseudopolynomial algorithm which - given an instance I of the Bin Packing Problem - finds a solution for I which uses no more than k bins or decides that no such solution exists.

36. Consider the BBP restricted to instances with a_1, a_2, ..., a_n such that $a_i > \frac{1}{3}$, for all $i \in \{1, 2, \dots, n\}$.

    (a) Reduce the problem to the cardinality matching problem.

    (b) Show how to solve the problem in $O(n \log n)$ time.

37. Suppose that the n cities of a Travelling Salesman Problem (TSP) are partitioned into m clusters such that the distance between two cities is 0 if and only if they belong to the same cluster.

    (a) Prove that there exists an optimum TSP tour with at most $m(m - 1)$ edges of positive weight.

    (b) Prove that such a TSP can be solved in polynomial time if m is fixed.

38. Decribe a polynomial time algorithm which optimally solves any TSP instance where the input is the metric closure of a weighted tree.

    (Recall that the metric closure of an undirected graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}_+$ is a graph $\bar{G} = (\bar{V}, \bar{E})$ with edge weights $\bar{c}\colon \bar{E} \to \mathbb{R}$, where $\bar{V} := V$, $\bar{E} := \Big\{\{i, j\}\colon$ there exists an i-j-path in $G\Big\}$ and $\bar{c}(\{i, j\})$ equals the length of a shortest i-j-path in (G, c) for any $\{i, j\} \in \bar{E}$.