Universität Bremen FB3: Mathematik/Informatik

Dr. Martin Böhm
Dr. Ruben Hoeksma
Prof. Dr. Nicole Megow

Summer 2019

Approximation Algorithms

Homework sheet 2 (Deadline 30.04.19 12:00 – before the lecture starts.)

All exercises must be done individually. Feel free (and encouraged) to discuss among each otherut each solution must be written independently. Two or more submissions found with exactly the same solution on any of the exercises will be awarded no points for the entire exercise sheet. The same holds in case of directly copying from any other source.

Exercise 1 (*4 points*) Consider the two traveling salespeople problem

Input: A graph $G = (V, E)$ with weights on the edges $w(e)$ for all $e \in E$, a starting node $v^* \in V$, and a tour length W.
Task: Give two tours T_1, T_2 starting in v^* such that $T_1 \cup T_2$ visits all the vertices in V and the total weight of each tour is at most W.

Show that the two traveling salespeople problem is NP-hard.
HINT: use that TSP is NP-hard.

Exercise 2 (*8 points*). Consider the bin packing problem that is defined as follows.

Input: n items with integer sizes $a_1, \ldots, a_n$ an integer bin capacity B.
Task: Divide the items into k subset $S_1, \ldots, S_k$ such that $\sum_{i \in S_j} a_i \leq B$ for all $j \in \{1, \ldots, k\}$.
Objective: Minimize k, the number of bins used.

Consider also the Next Fit algorithm:

Next Fit

---

(1) Open a new bin ($k := 1$, $S_k := \emptyset$). Capacity $:= B$.

While $i < n$:

(2) If $a_i \leq$ Capacity: place i in the bin ($S_k := S_k \cup \{i\}$).
Capacity := Capacity $- a_i$.

(3) Else: open new bin ($k := k + 1$). Place i in the bin ($S_k := \{i\}$).
Capacity $:= B - a_i$.

---

a) Prove that Next Fit is a 2-approximation for the bin packing problem.

b) Prove that there cannot exist a $(3/2 - \varepsilon)$-approximation for the bin packing problem unless P = NP.
HINT: Use that the partition problem is NP-hard.

Partition problem:

Input: n items with integer sizes $a_1, \dots, a_n$.
Task: Partition the items into two subset S_1, S_2 such that $\sum_{i \in S_1} a_i = \sum_{i \in S_2} a_i$.

Exercise 3 (*8 points*) Given an undirected graph $G = (V, E)$, a *cut* in G is a subset $S \subset V$. For $S, S' \subset V$ with $S \cap S' = \emptyset$ we denote by $\delta(S, S') \subset E$ the set of edges with one endpoint in S and one endpoint in S' and we call $|\delta(S, V \setminus S)|$ the *size* of the cut S. MAX CUT is the following problem[1]:

Input: an undirected graph $G = (V, E)$
Task: Find a cut S in G of maximum size $|\delta(S, V \setminus S)|$.

Consider the local search move *flip*: given a set S, find a $v \in S$ such that the cut $S \setminus \{v\}$ is larger than S or find a $v \notin S$ such that the cut $S \cup \{v\}$ is larger than S.

a) Prove that a local optimal solution (one where no flip move results in a improvement) is a 1/2-approximation for MAX CUT.

b) Prove that a local search algorithm that repetitively makes improving flip moves terminates in time polynomial in the input size of MAX CUT.

[1]Note: we saw this problem in homework set 1 as well.