Dr. Ulf-Peter Schroeder

Paderborn, 7. Juni 2017
Abgabe: Keine!

Übungen zur Vorlesung
Methoden des Algorithmenentwurfs
SS 2017
Blatt 7

Aufgabe 17:

a) An *independent set* of an undirected graph $G = (V, E)$ is a subset $U \subseteq V$ such that for every two vertices in U, there is no edge connecting the two. A maximum independent set is a largest independent set for a given graph G. The problem of finding such a set is called the $IndependentSetProblem$. Reformulate this problem as an ILP.

b) A problem instance of $MaxCut$ is an undirected, connected graph $G = (V, E)$ with $|E| \geq 1$. A feasible solution of $MaxCut$ is a partition $S = [V_\ominus, V_\oplus]$ of the node set V, so that $V_\ominus \cup V_\oplus = V$ and $V_\ominus \cap V_\oplus = \emptyset$. S is called a *cut* in G. An edge $\{u, v\} \in E$ is called *cut-edge* with respect to S, if $u \in V_\ominus$ and $v \in V_\oplus$. Let

$$C(S) = |\{\{u, v\} \in E | u \in V_\ominus, v \in V_\oplus\}|$$

define the measurement of a cut S, called *cut-size*. The problem is to find a cut with maximum cut-size. Reformulate the $MaxCut$ problem as an ILP.

Aufgabe 18:
The problem $FacilityLocation$ is defined as follows:
Let F be a set of potential facilities and C a set of clients. Denote by f_i the cost of opening facility $i \in F$ and by c_{ij} the cost of connecting client $j \in C$ to the (open) facility $i \in F$. The c_{ij}'s satisfy the triangle inequality. The problem is to find a subset $I \subseteq F$ of facilities to open and an assignment $\phi : C \longrightarrow I$ of clients to open facilities minimizing the total cost

$$cost(I, \phi) = \sum_{i \in I} f_i + \sum_{j \in C} c_{\phi(j)j}.$$

Reformulate the $FacilityLocation$ problem as an ILP.

Aufgabe 19:
We consider the „Weighted Set-Cover Problem“ which is based on a set U of n elements and a list $S_1, S_2, \ldots, S_m$ of subsets of U. We say that a *set cover* is a collection of these sets whose union is equal to all of U. Moreover, each set S_i has an associated *weight* $w_i \geq 0$. The goal is to find a set cover C so that the total weight $\sum_{S_i \in C} w_i$ is minimized. Someone suggests the following „Greedy-Set-Cover Algorithm“ for the Weighted Set-Cover Problem:

Start with $R = U$ and no sets selected
while $R \neq \emptyset$ do
Select set S_i that minimizes $\frac{w_i}{|(S_i \cap R)|}$
Delete set S_i from R
ENDwhile
Return the selected sets

Show that the „Greedy-Set-Cover Algorithm“ can be thought of as a *pricing algorithm* and prove that the algorithm finds a solution within an approximation factor $O(\log n)$.