Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Harald Räcke
Chintan Shah

Summersemester 2014
Übungsblatt 7
June 1, 2015

Efficient Algorithms and Datastructures II

Aufgabe 1 (10 Punkte)

Let $G = (V, E)$ be a given graph and $c_e \geq 0$ be the cost of edge e. Let $\{(s_1, t_1), \ldots, (s_k, t_k)\}$ be a set of specified pairs of vertices. In the minimum multicut problem, we wish to find a minimum cost set of edges F such that $\forall i, s_i$ and t_i are in different components of $G' = (V, E \setminus F)$.

(a) Write an Integer Linear Program (ILP) for solving this problem, where you have a variable for each edge and a constraint for each path from s_i to t_i, for all i.

(b) Relax this ILP to a Linear Program, say (P).

(c) Show how to solve (P) efficiently.

Aufgabe 2 (10 Punkte)

Given a directed graph $G = (V, E)$, a special vertex r and a positive cost c_{ij} for each edge $(i, j) \in E$, the minimum-cost arborescence problem is to find a subgraph of minimum cost that contains directed paths from r to all other vertices.

(a) Observe that the following ILP solves the minimum-cost arborescence problem:

$$\begin{array}{llcll} \text{minimize} & \sum_{(i,j)\in E} c_{ij} x_{ij} & & & \\ \text{subject to} & \sum_{i\in S, j\notin S, (i,j)\in E} x_{ij} & \geq & 1 & \forall S \subseteq V, S \ni r \\ & x_{ij} & \in & \{0,1\} & \forall (i,j) \in E \end{array}$$

(b) Show how to efficiently solve the LP obained by relaxing the above ILP.

Aufgabe 3 (10 Punkte)

Let $G = (V, E)$ be a given graph. Consider the following ILP:

$$\begin{array}{llcll} \text{maximize} & \sum_{i} x_i & & & \\ \text{subject to} & x_i + x_j & \leq & 1 & \forall (i,j) \in E \\ & \sum_{i\in C} x_i & \leq & \frac{|C|-1}{2} & \forall \text{ odd cycles } C \\ & x_i & \in & \{0,1\} & \forall i \in V \end{array}$$

(a) Explain in your own words, which problem the above ILP solves.

(b) Relax this ILP to an LP so that $0 \leq x_i \leq 1, \forall i \in V$ and show how to solve this LP efficiently.