Prof. Nicole Megow
Dr. Syamantak Das

Summer 2017

Approximation Algorithms

Exercise Sheet 3

Exercise 1 (*10 points*) **Steiner Forest** In the Min Cost Steiner Forest problem, we are given an undirected weighted graph $G(V, E)$ with $c : E \to \mathbb{R}^+ \cup \{0\}$ being the cost function on the edges. We are further given a set of k source-sink pairs $R = \{(s_1, t_1), (s_2, t_2), \cdots (s_k, t_k)\}$, where $s_i, t_i \in V$, for all $i = 1, 2, \cdots k$. The goal is to find a sub-graph of minimum possible cost such that there is a (s_i, t_i) path in this sub-graph, for all $i = 1, 2, \cdots k$.

Formulate a cut-based LP relaxation for this problem and prove that it is indeed a valid relaxation. You need to show that given any integer solution to the LP, you can create a feasible subgraph to the problem with same cost. On the other hand, given a feasible subgraph, you need to demonstrate the existence of a feasible solution with the same cost.

Exercise 2 (*10 points*) **Maximum Weight Independent Set of Intervals**. In this problem, we are given a time horizon $\mathcal{T} = [0, T]$. We are given k sub-intervals of $\mathcal{T}$, denoted by $I_1, I_2, \cdots I_k$, where any sub-interval I_j is specified by the pair $[s_j, t_j] \subseteq \mathcal{T}$ where $s_j \leq t_j$ and s_j and t_j are respectively the start and end times of the interval I_j. Further, each sub-interval I_j has a non-negative weight associated with it, denoted by w_j. The algorithmic task is to pick a *maximum weight* subset of sub-intervals out of $I_1, I_2, \cdots I_k$ such that the picked sub-intervals are pairwise non-overalapping. Formally, if I_j and $I_{j'}$ are any two intervals picked by the solution, then $I_j \cap I_{j'} = \phi$.

Formulate an ILP for the above problem. How many constraints are there in your ILP? Can you bring down the number of constraints to be a polynomial in k (in case your original ILP has bigger number of constraints) ?

Exercise 3 (*10 points*) The following problem arises in telecommunication networks and is known as the *SONET ring-loading* problem. The network consists of n points marked 1 through n on a circle or ring. We are given a set of calls C. Each call, denoted by a pair (i, j) originates at i and needs to be routed to j. There are only two directions in which the call can be routed - clockwise or anticlockwise. The load on a link $(i, (i + 1) \mod n)$, L_i is the total number of calls that is routed through this link. The goal is to route all calls so as to minimize the maximum load on any link, $\max_{i=1}^{n} L_i$.

Write an LP relaxation to the above problem. Use it to give a 2-approximation algorithm for the above problem.