Exercises

Approximation Algorithms

Spring 2010

Sheet 3

Note: This is just one way, a solution could look like. We do not guarantee correctness. It is your task to find and report mistakes.

Exercise 1
Consider the $k$-SET COVERING problem: Given a family of sets $S_1, \ldots, S_m \subseteq U$ of cardinality $|S_i| \leq k$ with cost $c(S_i)$, find a subset of these sets that minimize the cost, while each element has to be covered at least once. Recall the linear programming relaxation

$$\min \sum_{i=1}^{m} c(S_i) \cdot x_i \quad (LP)$$

$$\sum_{i : j \in S_i} x_i \geq 1 \quad \forall j \in U$$

$$x_i \geq 0 \quad \forall i$$

where $x_i$ indicates, whether to take set $S_i$.

i) Let $x^*$ be an optimum basic solution for $(LP)$. Prove that there is an $i$ with $x_i^* \geq \frac{1}{k}$.

ii) Consider the following iterative rounding algorithm:

(1) WHILE $U \neq \emptyset$ DO

(2) Compute an optimum basic solution $x^*$

(3) Choose $i$ with $x_i^* \geq \frac{1}{k}$

(4) Buy set $S_i$, delete elements in $S_i$ from the instance

(5) Output bought sets

Prove that this algorithm gives a $k$-approximation.

Hint: How much does the value of the optimum fractional solution decrease in each iteration compared to the bought set?
Solution:

i) Let $x^*$ be a basic solution of $(LP)$. Apart from the non-negativity, the LP has $n = |U|$ constraints (one for each element). From the lecture we know that $|\{i \mid x^*_i > 0\}| \leq n$. On the other hand, since each set covers at most $k$ elements, we must have $\sum_{i=1}^m x^*_i \geq \frac{n}{k}$. Hence the average value $x^*_i$ of all $i$ with $x^*_i > 0$ must be at least $\frac{n/k}{n} = \frac{1}{k}$. Especially the highest value must be $\geq \frac{1}{k}$.

ii) Consider the first step and let $x^*$ be the fractional solution. For buying $S_i$ we pay $c(S_i)$. As new fractional solution we define $\hat{x}^*_j := x^*_j$ for $j \neq i$ and $\hat{x}^*_i := 0$. $\hat{x}^*$ is in fact feasible. After reindexing let $S^1, \ldots, S^\ell$ be the edges, which are chosen by the algorithm. Let $OPT^*_{\ell}$ be the value of the optimum fractional solution at the beginning of the $i$th step (and $OPT^*_{\ell+1} = 0$). Then

\[ APX = \sum_{i=1}^\ell c(S^i) \]

Furthermore

\[ OPT^*_{\ell} = OPT^*_{\ell+1} = \sum_{i=1}^\ell \left( OPT^*_{\ell} - OPT^*_{\ell+1} \right) \geq \frac{1}{k} \sum_{i=1}^\ell c(S^i) = \frac{1}{k} APX \geq \frac{1}{k+1} c(S^i) \]

Hence $APX \leq k \cdot OPT^*_{\ell}$.

Exercise 2

For the STEINER TREE problem, we are given an undirected weighted graph $G = (V, E)$ with a cost function $c : E \rightarrow \mathbb{Q}_+$ and a set of terminals $R \subseteq V$. It is the goal to find a tree $T$ that connects all terminals. A natural linear programming relaxation is

\[ \min \sum_{e \in E} c_e x_e \quad \text{(LP)} \]

\[ \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \subseteq V : 1 \leq |S \cap R| < |R| \]

\[ x_e \geq 0 \quad \forall e \in E \]

Here $\delta(S) = \{ \{u, v\} \in E \mid u \in S, v \notin S \}$ are the edges, crossing $S$. Show that one can compute an optimum fractional solution for $(LP)$ in polynomial time (to be precise: Show that the LP can be solved in time polynomial in $n = |V|$ and the encoding length $\langle c \rangle$ of $c$).

**Hint:** Use the Ellipsoid method from the lecture. Recall that the $s$-$t$ MINCUT problem is polynomial time solvable: Given a graph $G = (V, E)$, nodes $s, t \in V$ and capacities $w : E \rightarrow \mathbb{Q}_+$, compute an $s$-$t$ cut $S \subseteq V$ with $s \in S, t \notin S$ that minimizes $\sum_{e \in \delta(S)} w(e)$.

**Solution:**

We want to apply the Ellipsoid algorithm (see the Theorem from the lecture). First of all the dimension of the $(LP)$ is $m = |E| \leq \binom{n}{2} \leq n^2$. Each left hand side of a constraint is a 0/1-vector with $\leq m$ entries. Hence the encoding length of such a constraint is $O(m) = O(n^2)$. Hence $\varphi = \langle c \rangle + O(n^2)$ is a feasible choice. We next consider the separation problem:
Let $y \in \mathbb{Q}^E$ be a given vector. We check first if there is an $i$ with $y_i < 0$. If yes, return $e_i$ since $y_i = e_i^T y < 0 \leq e_i^T x = x_i$ for every feasible $x$. Next, we want to find the set $S \subseteq V$ that minimizes $\sum_{e \in \delta(S)} y_e$. But

$$
\min \{ \sum_{e \in \delta(S)} x_e : S \subseteq V : 1 \leq |S \cap R| < |R| \}
$$

$$
= \min_{s,t \in R} \left\{ \min_{S \subseteq V, s \in S, t \notin S} \left\{ \sum_{e \in \delta(S)} y(e) \right\} \right\}
$$

$$
= \min_{s,t \in R} \{ \text{value of } s-t \text{ MinCut with capacities } y(e) \}
$$

The set $S$ attaining this minimum can be computed in time $\text{poly}(\langle y \rangle, n) = \text{poly}(\varphi)$. If $\sum_{e \in \delta(S)} y(e) \geq 1$ then $y$ is feasible. Otherwise the characteristic vector of $\sum_{e \in \delta(S)} y(e)$ is a feasible output for the separation oracle since

$$
\sum_{e \in \delta(S)} y(e) < 1
$$

with $1 \leq |R \cap S| < |R|$ yields a violated inequality.