

CO453: Network Design – Winter 2007

Instructor: Chaitanya Swamy

Solutions to Assignment 6

You must give a proof of correctness of any algorithm you design, and argue briefly why it runs in polynomial time. You may use any proof or algorithm covered in class directly.

Q1(a) Here we have $A := \{T \subseteq E : T \text{ is a spanning tree of } G\}$. This is the set of all outputs to the problem. The valuation $v_i : A \mapsto \mathbb{R}$ of player i is completely determined by the costs c_e of the edges in his set S_i . Given values $c_e \forall e \in S_i$, we have $v_i(T) = \sum_{e \in T \cap S_i} c_e$, which is the cost incurred by player i if tree T is chosen. Thus, we can represent the domain $V_i \subseteq \mathbb{R}^A$ of player i as

$$V_i := \left\{ v_i : A \mapsto \mathbb{R} : \exists c_e \geq 0 \quad \forall e \in S_i \quad \text{s.t.} \quad v_i(T) = \sum_{e \in T \cap S_i} c_e \right\}.$$

Finally, the target function g is the MST function, that is, $g(v_1, \dots, v_k) = T$ where $\sum_{i=1}^k v_i(T) = \min_{T' \in A} \sum_{i=1}^k v_i(T')$. Observe that if the functions v_i , $i = 1, \dots, k$ are specified by the edge costs $\{c_e\}_{e \in E}$, then by the definitions of the functions v_i , $\sum_i v_i(T)$ is precisely $\sum_{e \in T} c_e$.

(b) The target function g is exactly of the form for which the VCG result applies. Thus, by the VCG theorem, any payments to the players of the form $p_i(v) = -\sum_{j \neq i} v_j(T) + h_i(v_{-i})$, where $T = g(v)$, truthfully implement g . With these payments, the utility of a player i when he declares his true valuation is $\text{utility}_i(\bar{v}_i; (\bar{v}_i, v_{-i})) = h_i(v_{-i}) - (\bar{v}_i(T) + \sum_{j \neq i} v_j(T))$. To ensure that this expression is nonnegative, we may choose any tree T_i in the graph $G_i = (V, E \setminus S_i)$, where the choice depends only on v_{-i} , and set $h_i(v_{-i}) = \sum_{j \neq i} v_j(T_i) = \sum_{e \in T_i} c_e$, since then $\text{utility}_i(\bar{v}_i; (\bar{v}_i, v_{-i})) = c(T_i) - c(T) \geq 0$ because T is the MST with respect to the reported edge costs. In particular, we can set T_i to be the MST in G_i . This has the additional desirable “normalization” property that if $T \cap S_i = \emptyset$, that is, player i does not incur any cost for the tree T that is chosen, then he also gets 0 payment.

Q2(a) Consider replacing the exact algorithm g in the VCG-mechanism by an α -approximation algorithm f for the set cover problem. That is, for every weight-vector w , we have $f(w) = \mathcal{S}'$ where $\sum_{S \in \mathcal{S}'} w_S \leq \alpha \cdot \min_{\mathcal{S}'' \in A} \sum_{S \in \mathcal{S}''} w_S$. Then the payment $p_T(w)$ for a player T is given by $-\sum_{S \neq T} w_S(\mathcal{S}') + h_T(w_{-T})$, where $\mathcal{S}' = f(w)$. Thus, the utility of player T whose true value is \bar{w}_T and who reports w_T , is precisely the *negative of the total weight of \mathcal{S}' under the weight-vector $w' = (\bar{w}_T, w_{-T})$* plus a constant (the term $h_T(w_{-T})$ is a constant for player T since w_{-T} is fixed).

The proof of truthfulness of the VCG-mechanism crucially relied on the fact that since we have an exact algorithm for set cover, the w' -weight is minimized by the set cover \mathcal{S}^* output by the algorithm on the input w' ; hence, the utility of player T is maximized by reporting his true value \bar{w}_T . With an approximation algorithm f however, it is no longer true that its output, say $\hat{\mathcal{S}}$, on the input w' minimizes the w' -weight over all set covers $\mathcal{S}' \in A$. Instead, we only have that the w' -weight of $\hat{\mathcal{S}}$ is within a factor α times the w' -weight of the optimal set cover. Thus, there could be a set cover \mathcal{S}'' , and an input w_T other than \bar{w}_T such that $\mathcal{S}'' = g(w_T, w_{-T})$, with the property that the w' -weight of \mathcal{S}'' is *lower* than the w' -weight of $\hat{\mathcal{S}}$. Under this circumstance, player T stands to gain by “gaming” the approximation algorithm f , and reporting w_T instead of the true value \bar{w}_T .

(b) Fix a player S and the inputs w_{-S} of the others. We need to argue that if $w_S^1 > w_S^2$, then if the algorithm picks set S on input (w_S^1, w_{-S}) , then it also picks set S under the input (w_S^2, w_{-S}) . The algorithm f picks a set S' iff $x_{S'}^* > 0$, where x^* is the LP-optimal solution. So as mentioned in the hint, it suffices to show that if x^1 and x^2 optimal solutions respectively to the inputs (w_S^1, w_{-S}) and (w_S^2, w_{-S}) , then $x_S^2 \geq x_S^1$.

Notice that both x^1 and x^2 are feasible fractional set covers. Thus, since x^1 is the optimal solution under input (w_S^1, w_{-S}) , we have

$$w_S^1 x_S^1 + \sum_{S' \neq S} w_{S'} x_{S'}^1 \leq w_S^1 x_S^2 + \sum_{S' \neq S} w_{S'} x_{S'}^2. \quad (1)$$

Similarly since x^2 is the optimal solution under input (w_S^2, w_{-S}) , we have

$$w_S^2 x_S^2 + \sum_{S' \neq S} w_{S'} x_{S'}^2 \leq w_S^2 x_S^1 + \sum_{S' \neq S} w_{S'} x_{S'}^1. \quad (2)$$

Adding (1) and (2) and simplifying we obtain that

$$(w_S^1 - w_S^2)(x_S^1 - x_S^2) \leq 0.$$

Since $w_S^1 > w_S^2$, this implies that $x_S^1 \leq x_S^2$.