CO453: Network Design – Winter 2007

Instructor: Chaitanya Swamy

Solutions to Assignment 5

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: Given fractional values y_i^* for the facilities in $\mathcal{F}$, we can obtain the optimal fractional assignment x_{ij}^* for each client $j \in \mathcal{C}$ to facilities as follows. We consider facilities in $\mathcal{F}$ in increasing order of c_{ij} and assign j to each facility i to the maximum possible extent (i.e., set x_{ij}^* as high as possible) until j is assigned to a total extent of 1. More precisely, we initialize $x_{ij}^* = 0$ for all facilities i, and considering facilities in $\mathcal{F}$ in increasing order of c_{ij}, we set $x_{ij}^* = \min(y_i^*, 1 - \sum_{i'<i} x_{i'j}^*)$, where $i' < i$ means that i' comes before i in the sorted order.

It is easy to argue that x^* is an optimal assignment of clients to facilities given the facility-values y_i^*. If not, consider an optimal assignment $\tilde{x}$. There must be some client j such that $\sum_i c_{ij}\tilde{x}_{ij} < \sum_i c_{ij}x_{ij}^*$. Ordering the facilities in increasing order of c_{ij}, note that for any assignment $(x_{ij})_i$, we can write

$$\sum_i c_{ij}x_{ij} = c_{1j}\sum_{i\geq 1} x_{ij} + (c_{2j} - c_{1j})\sum_{i\geq 2} x_{ij} + \ldots + (c_{mj} - c_{m-1,j})x_{mj}$$

where there are m facilities in all. So if $\sum_i c_{ij}\tilde{x}_{ij} < \sum_i c_{ij}x_{ij}^*$, there must be some index i' such that $\sum_{i\geq i'} \tilde{x}_{ij} < \sum_{i\geq i'} x_{ij}^*$. Note that we must have $x_{i'j}^* > 0$ (since $\sum_{i\geq i'} x_{ij}^* > 0$), so $x_{ij} = y_i^*$ for all $i < i'$. But this gives a contradiction because then $\sum_{i<i'} \tilde{x}_{ij} > \sum_{i<i'} x_{ij}^* = \sum_{i<i'} y_i^*$.

Q2(a) If $R_j^\alpha > C_j^*/(1-\alpha)$, then we obtain the contradiction that

$$\sum_i c_{ij}x_{ij}^* \geq \sum_{i\in F_j^*: i\geq i_j} c_{ij}x_{ij}^* \geq R_j^\alpha \cdot \sum_{i\in F_j^*: i\geq i_j} x_{ij}^* > R_j^\alpha \cdot (1-\alpha) > C_j^*.$$

The second inequality follows since we are considering facilities in sorted order of distance c_{ij}, and the third since i_j is the first facility such that $\sum_{i\in F_j^*: i\leq i_j} \geq \alpha$.

(b) For every client j, we have $\sum_i \tilde{x}_{ij} = \sum_{i\in N_j^\alpha} x_{ij}^*/\alpha \geq 1$ since by definition of the set N_j^α, we have $\sum_{i\in N_j^\alpha} x_{ij}^* \geq \alpha$. Also, clearly $\tilde{x}_{ij} \leq x_{ij}^*/\alpha \leq y_i^*/\alpha = \tilde{y}_i$. So $(\tilde{x}, \tilde{y})$ satisfies all constraints of (UFL-LP).

(c) Consider $k \in \mathcal{C} \setminus \mathcal{C}'$ and let $j = \mathsf{nbr}(k)$. By definition, this means that j comes before k in the list L, so we have $R_j^\alpha \leq R_k^\alpha$. Also, by definition we have that $N_k^\alpha \cap N_j^\alpha \neq \emptyset$, so if $i \in N_k^\alpha \cap N_j^\alpha$, then $c_{jk} \leq c_{ij} + c_{ik} \leq R_j^\alpha + R_k^\alpha \leq 2R_k^\alpha$.

(d) For each client $j \in \mathcal{C}'$, we open the cheapest facility in N_j^α, whose facility-cost is therefore at most the cost of any weighted-average of the costs of the facilities in N_j^α. In particular, taking the weights $\tilde{x}_{ij}$, which add up to 1, the cost of facility opened from N_j^α is at most $\sum_{i\in N_j^\alpha} f_i\tilde{x}_{ij} \leq \sum_{i\in N_j^\alpha} f_i\tilde{y}_i$. Adding this over all clients in $\mathcal{C}'$, and since the sets N_j^α are disjoint for clients in $\mathcal{C}'$, we obtain that the total facility-opening cost is at most $\sum_i f_i \tilde{y}_i$. If $j \in \mathcal{C}'$, then it is assigned to a facility in N_j^α, therefore its assignment cost is at most R_j^α. If $j \notin \mathcal{C}'$, then it is assigned to a facility $i \in N_{\mathsf{nbr}(j)}^\alpha$ and we have $c_{ij} \leq c_{i,\mathsf{nbr}(j)} + c_{j,\mathsf{nbr}(j)} \leq R_{\mathsf{nbr}(j)}^\alpha + 2R_j^\alpha \leq 3R_j^\alpha$, where the last two inequalities follow from part (c).

Thus, the total cost incurred is at most

$$\sum_i f_i \tilde{y}_i + \sum_j 3R_j^\alpha \leq \frac{1}{\alpha} \cdot \sum_i f_i y_i^* + \frac{3}{1-\alpha} \cdot \sum_j C_j^* \leq \max\Big(\frac{1}{\alpha}, \frac{3}{1-\alpha}\Big) \cdot OPT,$$

where $OPT = \sum_i f_i y_i^* + \sum_j C_j^*$ is the cost of the optimum LP solution, which is a lower bound on the optimal cost. Hence, we obtain a $\max\big(\frac{1}{\alpha}, \frac{3}{1-\alpha}\big)$-approximation algorithm.

Q3: Recall the following primal and dual linear programs for metric UFL with penalties.

$$\begin{aligned} \min \quad & \sum_i f_i y_i + \sum_{j,i} c_{ij} x_{ij} + \sum_j p_j z_j && \text{(P)} \\ \text{s.t.} \quad & \sum_i x_{ij} + z_j \geq 1 && \forall j \\ & x_{ij} \leq y_i && \forall i, j \\ & x_{ij}, y_i, z_j \geq 0 && \forall i, j. \end{aligned}$$

$$\begin{aligned} \max \quad & \sum_j \alpha_j && && \text{(D)} \\ \text{s.t.} \quad & \alpha_j \leq c_{ij} + \beta_{ij} && \forall i, j \\ & \alpha_j \leq p_j && \forall j && (1) \\ & \sum_j \beta_{ij} \leq f_i && \forall i \\ & \alpha_j, \beta_{ij} \geq 0 && \forall i, j. \end{aligned}$$

The variable z_j in (P) denotes if we incur the penalty for client j. This adds the constraint (1) in the dual program. Intuitively, there is no incentive for j to pay an amount (to get itself assigned to a facility) that is more than the penalty p_j it incurs for not being assigned to a facility. In order to satisfy (1) and construct a feasible dual solution, we now stop increasing the dual variable α_j in the primal-dual process when α_j becomes equal to p_j. This is the only change in the primal-dual process. The entire process is summarized below.

1. **Initialization** Set $\alpha_j = 0$, $\beta_{ij} = 0$ for all i, j, time $t \leftarrow 0$. Time increases at rate 1 as the algorithm proceeds. Let $F' \leftarrow \emptyset$, $C \leftarrow \mathcal{C}$, $P \leftarrow \emptyset$. F' will denote the set of tentatively open facilities; $\mathcal{C} \setminus C$ is all the clients that we have either tentatively assigned to tentatively-open facilities or tentatively decided to incur a penalty on, so C is all the clients that we still need to process; and P is the set of clients for which we tentatively incur penalty.

2. **Primal-Dual process** While $C \neq \emptyset$, raise the α_j values for all $j \in C$ uniformly at rate 1, until one of the following events happens:

   (a) For some facility i and client $j \in C$, we have $\alpha_j = c_{ij}$: if $i \in F'$, we tentatively assign j to i, which is denoted by setting $i(j) = i$, and set $C \leftarrow C \setminus \{j\}$; otherwise we start increasing β_{ij} (which is currently 0) at the same rate as α_j.

   (b) For some facility $i \notin F'$, we have $\sum_j \beta_{ij} = f_i$: we add i to F'; for every client $j \in C$ such that $\alpha_j \geq c_{ij}$, we set $i(j) = i$ (j is tentatively assigned to i) and remove j from C.

   (c) For some client $j \in C$, we have $\alpha_j = p_j$: we remove j from C (notice that j has not been tentatively assigned to a facility in F'), and add j to P.

As in the Jain-Vazirani algorithm, opening all the tentatively open facilities can result in a very costly solution because a client could be paying for multiple facilities. So will need a clean-up step where we pick a subset of the tentatively open facilities and open these. This is done exactly as in the Jain-Vazirani algorithm.

Clean-up step and opening facilities For each $i \in F'$, let $C_i = \{j : \beta_{ij} > 0\}$. Pick a maximal subset $F'' \subseteq F'$ of tentatively-open facilities with the property that the sets C_i are pairwise-disjoint for facilities $i \in F''$. For every facility $i \in F' \setminus F''$, let $\mathsf{nbr}(i)$ denote a facility $i' \in F''$ such that $C_i \cap C_{i'} \neq \emptyset$ (there must be such a facility F'' is a maximal subset). Open all the facilities in F''.

Notice that as in question 1, once we have decided which facilities to open, the rest of the decisions (which clients to incur penalty on, how to assign remaining clients to facilities) are easy: we simply assign every client j to the nearest open facility i, provided $c_{ij} \leq p_j$, otherwise we do not assign j and incur the penalty p_j. However, *for purposes of analysis*, it will be convenient to specify a particular way of choosing which clients to assign to facilities and assigning them to open facilities. There are two ways of doing this, both of which yield a 3-approximation algorithm. The first scheme will have certain desirable monotonicity properties with regard to the clients for which we incur penalty; the second will satisfy a stronger approximation property.

Taking care of clients: algorithm I We first choose to incur penalty for all clients in P. For every *remaining* client j, if $j \in C_i$ for some $i \in F''$, then assign j to i. Otherwise, let $i = i(j) \in F'$ (note that $i(j)$ is well-defined). If $i \in F''$, we assign j to i, otherwise we assign j to $\mathsf{nbr}(i)$ (which is in F'').

Taking care of clients: algorithm II Here we may also assign some clients in P to facilities. Consider any client $j \in \mathcal{C}$. If $j \in C_i$ for some $i \in F''$, we assign j to i. Observe that $C_i \cap P$ *could be non-empty.* Otherwise, if $j \in P$, we incur the penalty for j. Otherwise j was tentatively assigned to some facility $i = i(j) \in F'$. If $i \in F''$, we assign j to i; else we assign j to $\mathsf{nbr}(i)$.

Analysis: Let $P' \subseteq P$ denote the final set of clients for which we incur penalty. Let $f(j)$ be the facility in F'' to which client $j \notin P'$ is assigned. Let $\mathcal{C}_1 = \bigcup_{i \in F''} C_i$. Let OPT denote the common optimal value of (P) and (D). Most of the analysis is common to both algorithms.

Lemma 1. *The cost of opening facilities in F'' is $\sum_{j \in \mathcal{C}_1} \beta_{f(j)j}$.*

Proof. For every facility $i \in F''$, we have $\sum_{j \in C_i} \beta_{ij} = f_i$ since i was tentatively-opened. Adding this over all facilities in F'', and since the sets C_i are disjoint for facilities in F'', we get that $\sum_{i \in F''} = \sum_{j \in \mathcal{C}_1} \beta_{f(j)j}$. ■

For a facility $i \in F'$, let t_i be the time at which facility i was tentatively opened. Observe the following simple facts. First, note that by the definition of the primal-dual process, if $\beta_{ij} > 0$, then we must have $c_{ij} < \alpha_j$ since once we start raising β_{ij} (step 2(a)), we maintain the equality $\alpha_j = c_{ij} + \beta_{ij}$ at all times. Second, if $\beta_{ij} > 0$ and $i \in F'$, then we must have $\alpha_j \leq t_i$ because at time t_i, when we tentatively open facility i, if $j \in C$ then the algorithm will tentatively assign j to i (step 2(b)). Third, observe that if $i(j) = i$, then $\alpha_j \geq t_i$. This is simply because α_j is precisely the time at which j is tentatively assigned (in step 2(a) or 2(b)), and since j is tentatively assigned to i, this can only happen at a time at or after the time when i is tentatively opened, which is t_i. Finally, clearly for any client $j \in P$ we have $\alpha_j = p_j$ by definition. These facts are stated collectively below.

Fact 2.

(i) if $\beta_{ij} > 0$ *then* $c_{ij} < \alpha_j$*;*

(ii) if $\beta_{ij} > 0$ *and* $i \in F'$*, then* $\alpha_j \le t_i$*;*

(iii) if $i(j) = i$ *then* $\alpha_j \ge t_i$*;*

(iv) if $j \in P$ *then* $\alpha_j = p_j$*.*

Lemma 3. *Consider a client* j *that is assigned to an open facility. If* $j \in \mathcal{C}_1$*, then* $c_{f(j)j} = \alpha_j - \beta_{f(j)j}$*, otherwise* $c_{f(j)j} \le 3\alpha_j$*.*

Proof. Let $i = f(j)$, the facility in F'' to which j is assigned. If $j \in \mathcal{C}_1$, then $j \in C_i$, so $c_{ij} = \alpha_j - \beta_{ij}$. Otherwise if $i = i(j)$, then $c_{ij} \le \alpha_j$ since j can only be tentatively assigned to i (in step 2(a) or 2(b)) if it has already reached i. In fact, since $j \notin C_i$, $\beta_{ij} = 0$, so we have $\alpha_j = c_{ij}$. Finally, if $j \notin C_i$ and $i \ne i(j)$, then $i = \mathsf{nbr}(i')$ where $i' = i(j)$. By the definition of $\mathsf{nbr}(.)$, there must be some client $k \in C_i \cap C_{i'}$, that is, we have $\beta_{ik},\ \beta_{i'k} > 0$. So we can bound $c_{ij} \le c_{ik} + c_{i'k} + c_{i'j}$ using the triangle inequality, and Fact 2 will allow us to bound each of these three terms. By parts (i) and (ii), we get that $c_{ik}, c_{i'k} < \alpha_k$ and $\alpha_k \le t_{i'}$. By part (iii), we have $t_{i'} \le \alpha_j$. Also, as observed earlier, $c_{i'j} \le \alpha_j$. Combining these inequalities, we get that $c_{ij} \le 2\alpha_k + \alpha_j \le 3\alpha_j$. ∎

We now prove the 3-approximation guarantees for both algorithms.

Theorem 4. *Algorithm I returns a solution of total cost at most* $3 \cdot OPT$*.*

Proof. For algorithm I, we have $P' = P$. So the total cost incurred is

$$\begin{aligned}
\sum_{i \in F''} f_i + \sum_{j \notin P} c_{f(j)j} + \sum_{j \in P} p_j \\
&\le \sum_{j \in \mathcal{C}_1} \beta_{f(j)j} + \sum_{j \in \mathcal{C}_1} \left(\alpha_j - \beta_{f(j)j}\right) + \sum_{j \notin \mathcal{C}_1 \cup P} 3\alpha_j + \sum_{j \in P} \alpha_j \qquad \text{(Lemmas 1 and 3, Fact 2 (iii))} \\
&= \sum_{j \in \mathcal{C}_1} \alpha_j + \sum_{j \in P} \alpha_j + \sum_{j \notin \mathcal{C}_1 \cup P} 3\alpha_j \\
&\le \sum_{j \in \mathcal{C}_1 \cup P} 2\alpha_j + \sum_{j \notin \mathcal{C}_1 \cup P} 3\alpha_j \qquad \text{(the factor of 2 comes since } \mathcal{C}_1 \cap P \text{ could be non-empty)} \\
&\le \sum_{j \in \mathcal{C}} 3\alpha_j \ \le\ 3 \cdot OPT.
\end{aligned}$$

∎

Algorithm I has a useful monotonicity property. One can show that if $j \in P$, and we *lower* the penalty from p_j to $p'_j < p_j$ and run the primal-dual process on the new instance, then we will again tentatively incur penalty for j. Since algorithm I incurs the penalty for precisely the set of clients for which we tentatively incur penalty (i.e., the set P), this means that if we incur the penalty for a client j under input p_j, we also incur penalty for j under input p'_j.

We can similarly prove a performance guarantee of 3 for Algorithm II. The difference between the two algorithms is that in Algorithm II, we specifically ensure that the clients we choose to incur penalty on (that is, the clients in P') *do not pay for opening facilities.* Thus, we can prove the following stronger statement.

Theorem 5. *For Algorithm II, the total cost incurred is at most* $3 \cdot OPT$*. In fact, we have* $\sum_{i \in F''} 3f_i + \sum_{j \notin P'} c_{f(j)j} + \sum_{j \in P'} 3p_j \leq 3 \cdot OPT$*.*

Proof.

$$\begin{aligned}
\sum_{i \in F''} 3f_i + \sum_{j \notin P'} c_{f(j)j} + \sum_{j \in P'} 3p_j \\
&\leq \sum_{j \in \mathcal{C}_1} 3\beta_{f(j)j} + \sum_{j \in \mathcal{C}_1} 3\big(\alpha_j - \beta_{f(j)j}\big) + \sum_{j \notin \mathcal{C}_1 \cup P'} 3\alpha_j + \sum_{j \in P'} 3\alpha_j \\
&= \sum_{j \in \mathcal{C}_1} 3\alpha_j + \sum_{j \in P'} 3\alpha_j + \sum_{j \notin \mathcal{C}_1 \cup P'} 3\alpha_j \\
&= \sum_{j \in \mathcal{C}_1 \cup P'} 3\alpha_j + \sum_{j \notin \mathcal{C}_1 \cup P} 3\alpha_j \qquad \text{(since the sets } \mathcal{C}_1 \text{ and } P' \text{ are now } \textit{disjoint} \text{ by construction)} \\
&= \sum_{j \in \mathcal{C}} 3\alpha_j \ \leq \ 3 \cdot OPT.
\end{aligned}$$

■

However, Algorithm II may or may not have the monotonicity property mentioned above, depending on how the maximal subset F'' is chosen. (Although, note that, as mentioned earlier, lowering p_j means that j is still a client for which we *tentatively incur penalty* if this was the case earlier.)