高等演算法 HW2

許博翔

April 12, 2024

Notation 1. $N(v) :=$ the neighborhood of v.

Problem 0.

Problem 1, 2: All by myself.

Problem 3, 4, 5: Discuss with: B10401113 張有朋、B10902101 鄭百里

Problem 1.

Equivalent ILP:

$\min \sum_{v \in V} w_v x_v.$

subject to : $x_u + x_v \geq 1, \ \forall uv \in E.$

$x_v \in \{0, 1\}, \ \forall v \in V.$

LP relaxation:

$\min \sum_{v \in V} w_v x_v.$

subject to : $x_u + x_v \geq 1, \ \forall uv \in E.$

$x_v \geq 0, \ \forall v \in V.$

Dual LP:

$\max \sum_{e \in E} \alpha_e.$

subject to : $\sum_{u \in N(v)} \alpha_{uv} \leq w_v, \ \forall v \in V.$

$\alpha_e \geq 0, \ \forall e \in E.$

The rephrased algorithm:

1. Initially, the residual weight $r_v = w_v$ for every vertex v. The vertex cover S is empty. All variables x_v, α_e are 0.

2. Repeat until all edges are covered by S:
    (a) Pick any edge $e = uv$ that is not covered by S.
    (b) Set α_{uv} to $\min(r_u, r_v)$, and reduce the residual weights r_u and r_v by α_{uv}.
    (c) Add all vertices v with 0 residual weights to S, and set x_v to 1.

We want to prove that:

1. $x_u + x_v \geq 1, \ \forall uv \in E.$
2. $\displaystyle\sum_{u \in N(v)} \alpha_{uv} \leq w_v, \ \forall v \in V.$
3. $x_u + x_v \leq 2$ or $\alpha_{uv} = 0, \ \forall uv \in E.$
4. $\displaystyle\sum_{u \in N(v)} \alpha_{uv} \geq w_v$ or $x_v = 0, \ \forall v \in V.$

Proof of 1.:

If $x_u + x_v < 1$, then $x_u = x_v = 0$.

$\Rightarrow$ neither u nor v is in S.

$\Rightarrow$ uv is not covered by S, contradiction.

$\therefore x_u + x_v \geq 1$.

Proof of 2.:

By 2(b) of the algorithm, the amount of change of r_v is the amount of α_e for e connected to v.

$\therefore \displaystyle\sum_{u \in N(v)} \alpha_{uv} =$ the amount of change of $r_v = w_v - r_v \leq w_v$.

Proof of 3.:

$x_u + x_v \leq 1 + 1 \leq 2$.

Proof of 4.:

If $x_v \neq 0$, then r_v is set to 0 by 2(c).

$\Rightarrow \displaystyle\sum_{u \in N(v)} \alpha_{uv} = w_v - r_v = w_v$.

Since the complementary slackness conditions 3. 4. are satisfied, this is a 2-approximation.

Problem 2. Run phase 1 in class.

Let K denote the given k in the problem description (since k is frequently used in this proof).

Let $S_j := \{i : \alpha_j > c_{ij}\},\ T_j := \{i : \alpha_j = c_{ij}\},\ U_j := \begin{cases} S_j, \text{ if } S_j \neq \emptyset \\ T_j, \text{ otherwise} \end{cases}$.

Since the algorithm in phase 1 guaranteed that all clients are connected.

$\therefore S_j \cup T_j \neq \emptyset,\ \forall j.$

In another word, $U_j \neq \emptyset,\ \forall j$.

Let's modify the phase 2 in class.

Let $I :=$ the set of all temporarily facilities, and serve client j with an arbitrary element $p_j \in U_j$.

Let $B_i := \{j : p_j = i\}$.

$\forall i \in I,\ \sum_{j \in B_i} (\alpha_j - c_{ij}) \leq \sum_{j : i \in U_j} (\alpha_j - c_{ij}) \overset{\alpha_j = c_{ij} \text{ for } i \in T_j}{=} \sum_{j : i \in U_j \cap S_j} (\alpha_j - c_{ij}) \overset{\text{by the definition of } U_j}{=} \sum_{j : i \in S_j} (\alpha_j - c_{ij}) \overset{\text{by phase 1}}{=} f_i.$

$\Rightarrow \sum_{j \in B_i} \alpha_j \leq \sum_{j \in B_i} c_{ij} + f_i.$

$f_i + \sum_{j \in B_i} c_{ij} \leq f_i + \sum_{j : i \in U_j} c_{ij} = \sum_{j : i \in U_j} \alpha_j.$

$\Rightarrow \sum_{i \in I} \left(f_i + \sum_{j \in B_i} c_{ij} \right) \leq \sum_{i \in I} \sum_{j : i \in U_j} \alpha_j = \sum_j |U_j| \alpha_j \leq \sum_j k \alpha_j$ (since if $c_{ij} = \infty$, then $\alpha_j < c_{ij}$).

Also, if $i \notin I$, $x_i = 0$.

$\therefore \sum_i x_i \left(\sum_j \beta_{ij} c_{ij} + f_i \right) = \sum_{i \in I} \left(\sum_{j \in B_i} c_{ij} + f_i \right) \leq \sum_j k \alpha_j.$

$\therefore$ this is a k-approximation.

Problem 3. Let K denote the given k in the problem description (since k is frequently used in this proof).

Let $S_j := \{i : \alpha_j > c_{ij}\},\ T_j := \{i : \alpha_j = c_{ij}\},\ U_j := \begin{cases} S_j, \text{ if } S_j \neq \emptyset \\ T_j, \text{ otherwise} \end{cases}$.

Since the algorithm in phase 1 guaranteed that all clients are connected.

$\therefore S_j \cup T_j \neq \emptyset,\ \forall j.$

In another word, $U_j \neq \emptyset,\ \forall j$.

Algorithm in phase 2:

1. $I := \emptyset, J :=$ the set of all temporarily open facilities, $S := \emptyset$.
2. while $J \neq \emptyset$
   (a) Let $i \in J$ s.t. $q_i := \sum\limits_{j \notin S : i \in U_j} \alpha_j$ is maximized, and let $S^{(i)} := S^c$.
   (b) Let A_i denote all facilities in J that are conflict with i.
   (c) Remove $A_i \cup \{i\}$ from J, add i to I.
   (d) for all $j \notin S$ with $i \in U_j$, serve j with i, and add j to S.
3. for all $j \notin S$, select an arbitrary $i \in U_j$, it must be in some A_k for some k by 2(b), serve j with k.

The maximality of I is guaranteed by the condition of the while loop.
Let p_j denote the facility that serves j in the above algorithm, and $B_i := \{j : p_j = i\}$.
By the definition of temporarily open and that no two facilities in I are confict with each other, $\forall i \in I,\ \{j : i \in S_j\} \subseteq B_i$.
$\Rightarrow \forall i \in I,\ \sum\limits_{j \in S \cap B_i} (\alpha_j - c_{ij}) = f_i$.
$\Rightarrow \forall i \in I,\ f_i + \sum\limits_{j \in S \cap B_i} c_{ij} = \sum\limits_{j \in S \cap B_i} \alpha_j$.
$\forall j \notin S$, by 3., there's $i \in U_j$ s.t. i conflicts with p_j. By the definition of conflict, $\exists k$ s.t. $\alpha_k - c_{ik} > 0$ and $\alpha_k - c_{p_j k} > 0$.
$\Rightarrow c_{p_j j} \leq c_{ij} + c_{ik} + c_{p_j k} < c_{ij} + 2\alpha_k \overset{\because i \in U_j}{\leq} \alpha_j + 2\alpha_k \leq 3\alpha_j$.
The last inequality above is because $\alpha_k =$ the time that k is connected $=$ the time that i is temporarily open $\leq$ the time that j is connected $= \alpha_j$.
$\forall i \in I$:
$\sum\limits_{j \in S \cap B_i} \alpha_j \overset{2(\text{d})}{=} \sum\limits_{j \in S^{(i)}} \alpha_j = q_i$.
$\sum\limits_{j \in B_i \setminus S} \alpha_j \overset{\text{there's } k \in U_j \text{ s.t. } k \in A_i}{\leq} \sum\limits_{k \in A_i} \sum\limits_{j \in B_i \setminus S : k \in U_j} \alpha_j \leq \sum\limits_{k \in A_i} \sum\limits_{j \in S^{(i)}} \alpha_j = \sum\limits_{k \in A_i} q_k \overset{2(\text{a})}{\leq} \sum\limits_{k \in A_i} q_i = |A_i| q_i \leq (K-1) q_i$.

$$\Rightarrow \sum_{j \in S \cap B_i} \alpha_j = q_i = \frac{1}{K}(1 + K - 1) q_i \geq \frac{1}{K} \left(\sum_{j \in S \cap B_i} \alpha_j + \sum_{j \in B_i \setminus S} \alpha_j \right) = \frac{1}{K} \sum_{j \in B_i} \alpha_j.$$

$\therefore \sum_{j\in B_i} c_{ij}+f_i = \sum_{j\in S\cap B_i} \alpha_j + \sum_{j\in B_i\setminus S} c_{ij} \le \sum_{j\in S\cap B_i} \alpha_j + 3\sum_{j\in B_i\setminus S} \alpha_j = 3\sum_{j\in B_i} \alpha_j - 2\sum_{j\in S\cap B_i} \alpha_j \le$
$(3-\frac{2}{K})\sum_{j\in B_i} \alpha_j$.
Also, $\forall i \notin I,\ x_i = 0$.
$\therefore$ this is a $(3-\frac{2}{K})$-approximation.

Problem 4. Let OPT $= \sum_i f_i y_i = \sum_j \alpha_j$. (They're equal by the strong duality theorem).

$\Rightarrow$ all slackness conditions must hold.

$\Rightarrow \alpha_j - \beta_{ij} = c_{ij}$ or $x_{ij} = 0,\ \forall i, j$.

Let p_j denote the falicity that serves j, q_i denote the chosen client j that open i in N_j, and $B_i := \{j : p_j = i\}$.

If $p_j \notin N_j$, it means that $N_j \cap N_{q_{p_j}} \neq \emptyset$. Let r_j denote a facility that $\in N_j \cap N_{q_{p_j}}$.

Else just simply set $r_j := p_j$.

Let F denote the set of falicities that are open.

$\sum_{i\in F} f_i \le \sum_{i\in F} f_i \sum_{k\in N_{q_i}} x_{kq_i} \le \sum_{i\in F} f_i \sum_{k\in N_{q_i}} y_k \overset{f_i\le f_k,\ \forall k\in N_{q_i}}{\le} \sum_{i\in F}\sum_{k\in N_{q_i}} f_k y_k \overset{N_{q_i}\cap N_{q_k}=\emptyset,\ \forall i\neq k}{\le}$
$\sum_i f_i y_i = OPT$.

$\forall j,\ c_{p_j j} \le c_{r_j j} + c_{r_j q_{p_j}} + c_{p_j q_{p_j}}$ (by the definition of metric).

Since q_{p_j} is chosen before j, there is $\alpha_{q_{p_j}} \le \alpha_j$.

Since $r_j \in N_j, r_j \in N_{q_{p_j}}, p_j \in N_{q_{p_j}}$, by the definition of N, there is $x_{r_j j}, x_{r_j q_{p_j}}, x_{p_j q_{p_j}}$ are all nonzero.

$\therefore$:

$c_{r_j j} \le c_{r_j j} + \beta_{r_j j} = \alpha_j$.

$c_{r_j q_{p_j}} \le c_{r_j q_{p_j}} + \beta_{r_j q_{p_j}} = \alpha_{q_{p_j}} \le \alpha_j$.

$c_{p_j q_{p_j}} \le c_{p_j q_{p_j}} + \beta_{p_j q_{p_j}} = \alpha_{q_{p_j}} \le \alpha_j$.

$\Rightarrow c_{r_j j} \le 3\alpha_j$.

$\therefore \sum_{i\in F}\sum_{j\in B_i} c_{p_j j} \le \sum_{i\in F}\sum_{j\in B_i} 3\alpha_j = 3\sum_j \alpha_j = 3OPT$.

$\therefore \sum_{i\in F} f_i + \sum_{j\in B_i} c_{p_j j} \le OPT + 3OPT = 4OPT$.

Since OPT is the optimal solution of LP relaxation, which is not greater than the optimal solution of ILP (denote as OPT').

$\therefore \sum_{i \in F} f_i + \sum_{j \in B_i} c_{p_j j} \leq 4OPT \leq 4OPT'$.
$\Rightarrow$ this is a 4-approximation.

Problem 5. We'll use the term "at time t" denote when the value of α_j of unserved client j is set to t in the algorithm (that is, not performing 1. or 2. yet).

Let $U^{(t)}$ denote U at time t.

Let p_j denote the facility that serves j, and $B_i := \{j : p_j = i\}$.

Suppose that f_i is open at time a_i.

1. (a) while there are unserved clients
   i. for i in facilities
      A. if i is closed, find a set of unserved clients $S(i)$ s.t. $val(i) := \dfrac{f_i + \sum_{j \in S(i)} c_{ij}}{|S(i)|}$ is minimized. (This is equivalent to 1. in the algorithm.)
      B. if i is open, find an unserved client $S(i) = \{s(i)\}$ s.t. $val(i) := c_{is(i)}$ is minimized. (This is equivalent to 2. in the algorithm.)
   ii. Let i^* be a facility s.t. $val(i^*)$ is minimized.
   iii. Open i^* if it's closed, and serve all clients in $S(i^*)$ by i^*.

2. Lemma: if $\alpha_j > c_{ij}$, then $\alpha_j \leq a_i$.

   Proof: If $\alpha_j > a_i$, then j is served after i is open. By 2. in the algorithm, $\alpha_j \leq c_{ij}$.

   $\therefore$ the lemma holds.

   There are 2 cases:

   Case 1: $\alpha_k = 0$.

   In this case, $\alpha_j = c_{ij} = 0, \ \forall j \in \{1, 2, \ldots, k\}$.

   $\therefore \sum_{j=x}^{k} (\alpha_x - c_{ij}) = 0 \leq f_i$ holds for all $x = 1, 2, \ldots, k$.

   Case 2: $\alpha_k \neq 0$.

   $\Rightarrow \alpha_k > c_{ik}$.

   $\Rightarrow$ by the lemma, $a_i \geq \alpha_k$.

   At time α_x, $x, x+1, x+2, \ldots, k$ are unserved, by 1. in the algorithm, $\sum_{j=x}^{k} (\alpha_x -$

$c_{ij}) \leq \sum_{j \in U^{(\alpha_x)}} \max(0, \alpha_x - c_{ij}) \leq f_i$. $\therefore \sum_{j=x}^{k} (\alpha_x - c_{ij}) \leq f_i$ always holds.

3. Claim: $\alpha_j - \alpha_x \leq c_{ix} + c_{ij},\ \forall 1 \leq x \leq j \leq k$.

   Proof: If $\alpha_j = \alpha_x$, then the claim holds trivially.

   If $\alpha_j > \alpha_x$, then $\alpha_j > \alpha_x \geq a_{p_x}$ since p_x is open before x is served.

   $\Rightarrow$ by the lemma, $\alpha_j \leq c_{p_x j}$.

   $\Rightarrow \alpha_j - \alpha_x \leq c_{p_x j} - \alpha_x \overset{\text{metric}}{\leq} c_{ij} + c_{ix} + c_{p_x x} - \alpha_x \overset{x \text{ is served by } p_x}{=} c_{ij} + c_{ix}$.

   $\Rightarrow \sum_{j=x}^{k} (\alpha_j - c_{ix} - 2c_{ij}) \overset{\text{the claim}}{\leq} \sum_{j=x}^{k} (c_{ix} + c_{ij} + \alpha_x - c_{ix} - 2c_{ij}) = \sum_{j=x}^{k} (\alpha_x - c_{ij}) \leq f_i$.

4. $\sum_{j=1}^{k} (\alpha_1 - c_{ij}) \leq f_i$.

   $\sum_{j=1}^{k} (\alpha_j - c_{i1} - 2c_{ij}) \leq f_i$.

   Since $\alpha_1 - c_{i1} \geq \alpha_1 - 3c_{i1} \geq 0$.

   $\therefore \sum_{j=1}^{k} (\alpha_j - 3c_{ij}) \leq \sum_{j=1}^{k} (\alpha_1 - c_{ij} + \alpha_j - c_{i1} - 2c_{ij}) \leq 2f_i$.

5. We need to define $\alpha'_j, \beta'_{ij} := \max(\alpha'_j - c_{ij}, 0)$ so that $\sum_j \beta'_{ij} \leq f_i$ can be satisfied for all i.

   Let $\alpha'_j = \frac{1}{3}\alpha_j$, one can see that $\sum_j \beta'_{ij} = \sum_{j:\alpha_j \geq 3c_{ij}} \alpha'_j - c_{ij} = \sum_{j:\alpha_j \geq 3c_{ij}} \frac{1}{3}(\alpha_j - 3c_{ij}) \leq \frac{2}{3} f_i \leq f_i$.

   From 1. of the algorithm, $\sum_{j \in B_i} (\alpha_j - c_{ij}) = f_i$.

   $\Rightarrow \sum_{j \in B_i} c_{ij} + f_i = \sum_{j \in B_i} \alpha_j \leq 3 \sum_{j \in B_i} \alpha'_j$.

   $\therefore$ this is a 3-approximation.