高等演算法 HW1

許博翔

April 11, 2024

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

Problem 1.

Equivalent ILP:

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

$\text{subjectto} : x_u + x_v \geq 1,\ \forall uv \in E.$

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

LP relaxation:

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

$\text{subjectto} : x_u + x_v \geq 1,\ \forall uv \in E.$

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

Dual LP:

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

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

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

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

$\therefore \sum_{u \in N(v)} \alpha_{uv} =$ 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).

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

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. $q_i := \sum_{j \notin S : i \in U_j} \alpha_j$ 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, $\forall i \in I,\ \{j : i \in S_j\} \subseteq B_i$.
$\Rightarrow \forall i \in I,\ \sum_{j \in S \cap B_i} (\alpha_j - c_{ij}) = f_i$.
$\Rightarrow \forall i \in I,\ f_i + \sum_{j \in S \cap B_i} c_{ij} = \sum_{j \in S \cap B_i} \alpha_j$.
$\forall j \notin S$, 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 $\alpha_k - c_{p_j k} > 0$.
$\Rightarrow c_{p_j j} \le c_{ij} + c_{ik} + c_{p_j k} < c_{ij} + 2\alpha_k \overset{\because i \in U_j}{\le} \alpha_j + 2\alpha_k \le 3\alpha_j$.
The last inequality above is because $\alpha_k =$ the time that k is connected $=$ the time that i is temporarily open $\le$ the time that j is connected $= \alpha_j$.
$\forall i \in I$:
$\sum_{j \in S \cap B_i} \alpha_j \overset{2(\mathrm{d})}{=} \sum_{j \in S^{(i)}} \alpha_j = q_i$.
$\sum_{j \in B_i \setminus S} \alpha_j \overset{\text{there's } k \in U_j \text{ s.t. } k \in A_i}{\le} \sum_{k \in A_i} \sum_{j \in B_i \setminus S : k \in U_j} \alpha_j \le \sum_{k \in A_i} \sum_{j \in S^{(i)}} \alpha_j = \sum_{k \in A_i} q_k \overset{2(\mathrm{a})}{\le} \sum_{k \in A_i} q_i = |A_i| q_i \le (K-1) q_i$.
$\Rightarrow \sum_{j \in S \cap B_i} \alpha_j = q_i = \frac{1}{K}(1 + K - 1) q_i \ge \frac{1}{K} \left(\sum_{j \in S \cap B_i} \alpha_j + \sum_{j \in B_i \setminus S} \alpha_j \right) = \frac{1}{K} \sum_{j \in B_i} \alpha_j$.
$\therefore \sum_{j \in B_i} c_{ij} + f_i = \sum_{j \in S \cap B_i} \alpha_j + \sum_{j \in B_i \setminus S} c_{ij} \le \sum_{j \in S \cap B_i} \alpha_j + 3 \sum_{j \in B_i \setminus S} \alpha_j = 3 \sum_{j \in B_i} \alpha_j - 2 \sum_{j \in S \cap B_i} \alpha_j \le (3 - \frac{2}{K}) \sum_{j \in B_i} \alpha_j$.
Also, $\forall i \notin I,\ x_i = 0$.
$\therefore$ this is a $(3 - \frac{2}{K})$-approximation.

Problem 4.

Problem 5.