Algorithm HW#2

B02901178 江誠敏

April 9, 2017

1 Problem 1.

First we define some notation. $\mathcal{N} = \{1, 2, \ldots, n\}$ would be the elements and $\mathcal{S} = \{S_1, \ldots, S_m\}$ would be the sets in the set cover problem, where each $S_i \subseteq \mathcal{N}$. Let $w_i = w(S_i)$ be the cost to choose S_i. The corresponding relaxed LP is:

$$\begin{aligned} \text{minimize} \quad & \sum_{S_i \in \mathcal{S}} w_i x_i \\ \text{s.t.} \quad & \sum_{j \in S_i} x_i \geq 1, \quad \forall j \in \mathcal{N} \end{aligned} \tag{1}$$

The dual LP is

$$\begin{aligned} \text{maximize} \quad & \sum_{j \in \mathcal{N}} \alpha_j \\ \text{s.t.} \quad & \sum_{j \in S_i} \alpha_j \leq w_i, \quad \forall S_i \in \mathcal{S} \end{aligned} \tag{2}$$

The greedy algorithm mentioned in class could be extend to give a dual feasible solution.

Algorithm 1: The greedy algorithm taught in class

$C \leftarrow \mathcal{N}$ `// Elements that has not been covered`
while $C \neq \varnothing$ **do**
  Find a set S with $S \cap C \neq \varnothing$ and minimize $t = w(S)/|S \cap C|$.
  Set $\alpha_j \leftarrow t/H_k$, for each $j \in S \cap C$. `/*` j `is covered in this round */`
  Choose S. `/* The cost is` $w(S) = t|S \cap C| = \sum_{j \in S \cap C} H_k \alpha_j$ `*/`
  $C \leftarrow C \setminus S$.
end

If this algorithm indeed gives a feasible dual solution $\boldsymbol{\alpha} = (\alpha_i)$, then it is easy to see that the algorithm would give a solution of the set cover problem with cost equal to $H_k \sum \alpha_i$, and

$$H_k \sum \alpha_j \leq H_k \, \mathrm{OPT}_{\text{LP,dual}} = H_k \, \mathrm{OPT}_{\text{LP,primal}} \leq H_k \, \mathrm{OPT}_{\text{ILP,primal}}$$

Thus it is an H_k-approximation solution.

Now, for any set $S = S_i$, we shall prove that $\sum_{j \in S} \alpha_j \leq w(S)$. Assume that $S = \{e_1, e_2, \ldots, e_h\}$, $h \leq k$, and let t_i be the time (round) which e_i is covered. Reorder the element so we could assume that $t_1 \le t_2 \le \cdots \le t_h$. Just before e_i is choosen, if we choose S to cover e_i, the cost is $w(S)/(h-i+1)$, thus we must have $\alpha_{e_i} \le w(S)/((h-i+1)H_k)$. Hence

$$\sum \alpha_{e_i} \le w(S)H_k\left(\frac{1}{1}+\frac{1}{2}+\cdots+\frac{1}{h}\right) \le w(S)H_h/H_k \le w(S)$$

and the prove is completed.

2 Problem 2.

Similar to problem 1., the corresponding relaxed LP is now:

$$\begin{aligned} \text{minimize} \quad & \sum_{S_i \in \mathcal{S}} w_i x_i \\ \text{s.t.} \quad & \sum_{j \in S_i} x_i \ge k_j, \quad \forall j \in \mathcal{N} \end{aligned} \tag{3}$$

The dual LP is

$$\begin{aligned} \text{maximize} \quad & \sum_{j \in \mathcal{N}} k_j \alpha_j \\ \text{s.t.} \quad & \sum_{j \in S_i} \alpha_j \le w_i, \quad \forall S_i \in \mathcal{S} \end{aligned} \tag{4}$$

We make a small modification to the algorithm.

Algorithm 2: The modified greedy algorithm

$C \leftarrow \mathcal{N}$ `// Elements that has not been covered`
foreach $j \in \mathcal{N}$ **do** $\alpha_j \leftarrow 0$, $y_j \leftarrow 0$ `//` y_j `is the number of time` j `is covered`
while $C \ne \varnothing$ **do**
    Find a set S with $S \cap C \ne \varnothing$ and minimize $t = w(S)/|S \cap C|$.
    Define $D \triangleq S \cap C$.
    foreach $j \in D$ **do** `/* We say` j `is covered once in this round */`
        $\alpha_j \leftarrow \alpha_j + t/(H_n k_j)$
        $y_j \leftarrow y_j + 1$
        if $y_j = k_j$ **then** `/* We say` j `is fully covered in this round */`
            $C \leftarrow C \setminus \{j\}$.
        end
    end
    Choose S.
end

In each round The cost of choosing S is

$$w(S) = w(S)|D|/|D| = H_n \sum_{j \in D} k_j t/(H_n k_j) = H_n \sum_{j \in D} k_j \Delta\alpha_j,$$

where $\Delta\alpha_j$ is the amount of change of α_j in that round. Thus the algorithm give a solution of the set cover problem with cost $H_n \sum k_j\alpha_j$.

So again, if we could prove that $\boldsymbol{\alpha} = (\alpha_i)$ is a feasible solution to the dual problem (6), then

$$H_n \sum \alpha_j \le H_n \,\mathrm{OPT}_{\mathrm{LP,dual}} = H_n \,\mathrm{OPT}_{\mathrm{LP,primal}} \le H_n \,\mathrm{OPT}_{\mathrm{ILP,primal}},$$

and the solution would be an H_n-approximation solution.

Similarly, for any set $S = S_i$, Assume that $S = \{e_1, e_2, \dots, e_h\}$, and let t_i be the time (round) which e_i is **fully covered**. Reorder the element so we could assume that $t_1 \le t_2 \le \cdots \le t_h$. Notice that each element e_i would be **covered once** exactly k_{e_i} times, and these all happen before e_i is fully covered.

Consider once in this k_{e_i} times when e_i is covered once. Let $j \triangleq e_i$. The change of α_{e_i} is $\Delta\alpha_{e_i} = t/H_n k_{e_i}$ where t is defined in line 4 of algorithm 2. If we choose S, then $w(S)/|S \cap C| \le w(S)/(h-i+1)$. Thus by the minimality of t, we must have $t \le w(S)/(h-i+1)$. Hence

$$\alpha_{e_i} = \sum \Delta\alpha_{e_i} \le k_{e_i} \cdot \frac{w(S)}{(h-i+1)H_n k_{e_i}} = \frac{w(S)}{h-i+1} H_n^{-1}$$

So

$$\sum \alpha_{e_i} \le \sum_i \frac{w(S)}{h-i+1} H_n^{-1} = w(S)\frac{H_h}{H_n} \le w(S)$$

And we conclude that $\boldsymbol{\alpha}$ is indeed a feasible solution.

3 Problem 3.

The relaxed LP correspond to the vertex cover is:

$$\begin{aligned} \text{minimize} \quad & \sum_{v \in V} w_v x_v \\ \text{s.t.} \quad & x_u + x_v \ge 1, \quad \forall (u,v) \in E \end{aligned} \tag{5}$$

Let $E(v)$ be all the edges adjacent to v, the dual LP is

$$\begin{aligned} \text{maximize} \quad & \sum_{e \in E} \alpha_e \\ \text{s.t.} \quad & \sum_{e \in E(v)} \alpha_e \le w_v, \quad \forall v \in V \end{aligned} \tag{6}$$

Investigating the complementary slackness, we have

$$\sum_e \alpha_e \le \sum_e {}^1\alpha_e(x_u + x_v) = \sum_v x_v \sum_{e \in E(v)} \alpha_e \le \sum_v w_v x_v$$

[1] $(u,v) = e$ here

Clearly we only need to consider the solution which every $x_v \le 1$, so we can assume $x_v + x_u \le 2, \forall (u,v) \in E$. Then if either $x_v = 0$ or $\sum_{e\in E(v)} \alpha_e = w_v$ for each $v \in V$, we would have

$$\sum_v w_v x_v = \sum_v x_v \sum_{e\in E(v)} \alpha_e = \sum_e \alpha_e (x_u + x_v) \le 2 \sum_e \alpha_e$$

Thus if these an algorithm satisfy

1. It gives a valid vertex cover S.
2. It induces a feasible dual solution $\boldsymbol{\alpha} = (\alpha_e)$.
3. The complementary slackness relation is satisfied. That is, for each vertex v, either v is not picked into S (i.e., $x_v = 0$), or $\sum_{e\in N(v)} \alpha_e = w_v$.

Then this algorithm is a 2-approximation algorithm. So we check these 3 conditions for both algorithms.

3.1 Algorithm 1

1. **It gives a valid vertex cover**: Since eventually each edge is removed, and we only remove covered edge, all edges are covered when the algorithm ends.

2. **It induces a feasible dual solution**: Let $\alpha_e = 0$ at the beginning. Define the slackness $\tilde{r}_v$ of a vertex v as $\tilde{r}_v \triangleq w_v - \sum_{e\in N(v)} \alpha_e$, Then $\tilde{r}_v = w_v = r_v$ at the very start. Now, when reducing the residual weight r_v by $c \cdot \deg(v)$, we also increase α_e by c for all edges e. Then it is easy to see that $\tilde{r}_v$ also decrease by $c \cdot \deg(v)$. Thus $r_v = \tilde{r}_v$ in any moment. Since $c \cdot \deg(v) = \min r_v / \deg(v) \cdot \deg(v) = \min r_v$ in each round, we never decrease r_v below zero, thus $\tilde{r}_v \ge 0$ at the end. Hence $\sum_{e\in N(v)} \alpha_e \le w_v$ for each v when the algorithm ends, and $\boldsymbol{\alpha}$ is then a feasible dual solution.

3. **The complementary slackness relation is satisfied**: It is clear from the statement since we add v into S only when $r_v = \tilde{r}_v = 0$ (i.e., $w_v = \sum_{e\in N(v)} \alpha_e$).

3.2 Algorithm 2

1. **It gives a valid vertex cover**: It is clear since the algorithm stops only when each edge is covered.

2. **It induces a feasible dual solution**: Similar as in algorithm 1, let $\alpha_e = 0$ at the beginning. When reducing the residual weight r_u and r_v by $t = \min(r_u, r_v)$, also increase α_e by t. Then it is easy to see that $\tilde{r}_u, \tilde{r}_v$ also decrease by t, thus $\tilde{r}_v$

matches r_v. Also $t \leq r_u$ and $t \leq r_v$, so we never decrease r_v below zero. Hence $\sum_{e \in N(v)} \alpha_e \leq w_v$ and we know that $\boldsymbol{\alpha}$ is a feasible solution.

3. **The complementary slackness relation is satisfied**: It is clear from the statement since we add v into S only when it has zero residual weight r_v.

Collaborators: None.

4 Problem 4.

The relaxed LP correspond to the metric uncapacitated facility locating problem with penalty is:

$$\begin{array}{lrcll} \text{minimize} & \multicolumn{4}{l}{\sum c_{i,j}x_{i,j} + \sum f_i y_i + \sum p_j z_j} \\ \text{s.t.} & z_j + \sum_i x_{i,j} & \geq & 1, & \forall j \\ & y_i - x_{i,j} & \geq & 0, & \forall i,j \\ & y_i,\, x_{i,j},\, z_j & \geq & 0, & \forall i,j \end{array} \tag{7}$$

The dual LP is:

$$\begin{array}{lrcll} \text{maximize} & \multicolumn{4}{l}{\sum \alpha_j} \\ \text{s.t.} & \alpha_j - \beta_{i,j} & \leq & c_{i,j}, & \forall i,j \\ & \sum_j \beta_{i,j} & \leq & f_i, & \forall i \\ & \alpha_j & \leq & p_j, & \forall j \\ & \alpha_j, \beta_{i,j} & \geq & 0, & \forall i,j \end{array} \tag{8}$$

Thus we modify the algorithm taught in class as following, where some definition is listed below:

- If we choose to give up client j, we say j is **covered** but **unconnected**.
- If a client j is **connected**, it is simultaneously **covered**.
- If $\beta_{i,j} > 0$, then we say client j **pays** for facility i.
- If there is a client j such that j **pays** for both i, i', then we say i, i' are **conflicted** with each other.
- At the end of the algorithm, we choose the facilities that is **truely opened**, and connect each **connected** client j to a facility i such that (i, j) is **truely connected**.

Algorithm 3: The modified greedy algorithm

`// Phase 1`

Initially set $\alpha_j \leftarrow 0, \beta_{i,j} \leftarrow 0$.

while *There is still a client not covered* **do**

  repeat

    Increase all a_j such that j is not covered yet, until next event.

  until *An event e occurs*

  switch *the event e* **do**

    case $\alpha_j = c_{i,j}$ *for some i, j* **do**

      Call the (i, j) pair tight.

      Whenever α_j changes in the future, set $\beta_{i,j} \leftarrow \alpha_j - c_{i,j}$.

      if *i is already temporarily opened* **then** Connect j to i

    end

    case $\alpha_j = p_j$ *for some client j* **do**

      Choose not to connect j with penalty p_j.

      Call j covered but unconnected.

    end

    case $\sum_j \beta_{i,j} = f_i$ *for some facility i* **do**

      Temporarily open facility i.

      foreach *uncovered or unconnected facility i with (i, j) being tight* **do**

        Connect j to i

      end

    end

  end

end

`// Phase 2`

$I \leftarrow$ any maximal set of temporarily opened facilities without conflicted.

foreach *facility $i \in I$* **do** Truely open i.

foreach *connected client j* **do**

  if *there exists $i \in I$ such that (i, j) is tight* **then** truely directly connect j to i.

  else

    Let i be the facility such that j is connected to, then i is conflicted with some $i' \in I$ since I is maximal.

    Truely indirectly connect j to i'.

  end

end

Now we analyze the cost. The cost could be decomposed into three parts:

$$X \triangleq \sum c_{i,j} x_{i,j}, \quad Y \triangleq \sum f_i y_i, \quad Z \triangleq \sum p_j z_j$$

where $x_{i,j} = 1$ if (i, j) truely connected, $y_i = 1$ if facility i is **truely opened** and $z_j = 1$ if client j is uncovered (those we give up with penalties), and these value equals 0 otherwise.

Let I be the opened facilities, U be all the uncovered clients, C_d be the clients connected directly, C_i be the clients connected indirectly. Define $X_d = \sum_{j \in C_d} c_{i,j} x_{i,j}$, $X_i = \sum_{j \in C_i} c_{i,j} x_{i,j}$, then $X = X_d + X_i$.

For Z, by line 12 in algorithm 3, $\alpha_j = p_j$ for these client j, thus $Z = \sum_{j \in U} z_j = \sum_{j \in U} \alpha_j$.

For $Y + X_d$,

$$Y = \sum_{i \in I} f_i = \sum_{i \in I} \beta_{i,j} = \sum_{\substack{i \in I \\ j \text{ pays for } i}} \beta_{i,j}$$

Notice that if j pays for i, then (i, j) is tight. By line 19, j is connected and thus $j \notin U$, also by line 27, $j \notin C_i$, so $j \in C_d$. Since there is no conflict in I, each client pays for at most one facility in I, hence

$$Y = \sum_{i \in I} f_i \leq \sum_{\substack{i \in I, \\ j \in C_d}} \beta_{i,j}$$

And we have

$$Y + X_d = \sum_{\substack{i \in I, \\ j \in C_d}} \beta_{i,j} + \sum_{\substack{j \in C_d \\ i,j \text{ connected}}} c_{i,j} \leq \sum_{j \in C_d} \alpha_j$$

Since if (i, j) is connected, (i, j) tight and thus $\alpha_j = \beta_{i,j} + c_{i,j}$.

For X_i, let j be a client in C_d, i be the client j is originally connected to and $i' \in I$ be the client j is truely indirectly connected to at the very end. Since i, i' are in conflict, there is a client j' such that j' pays for both i, i', so $(i, j), (i, j'), (i', j')$ are all tight. j' would not be connected after j connected to i, or else once i is opened, j' would immediately be connected to i. Thus $\alpha_{j'} \leq \alpha_j$, so $c_{i',j} \leq c_{i',j'} + c_{i,j'} + c_{i,j} \leq \alpha_{j'} + \alpha_{j'} + \alpha_j \leq 3\alpha_j$ since these edges are tight. Hence

$$X_i = \sum_{\substack{j \in C_i \\ i',j \text{ connected}}} c_{i',j} \leq 3 \sum_{j \in C_i} \alpha_j$$

Combining all above, we have

$$X + Y + Z = X_i + X_d + Y + Z \leq \sum_{j \in U} \alpha_j + \sum_{j \in C_d} \alpha_j + 3 \sum_{j \in C_i} \alpha_j \leq 3 \sum_{j} \alpha_j$$