KAIST, School of Computing, Spring 2024 **Scribed By: Eunjung KIM**
Algorithms for NP-hard problems (CS492) **Lecture #28**
Lecture: Eunjung KIM **30 May 2024**

Contents

1 WEIGHTED SET COVER revisited: a dual feasible solution as a certificate of your solution

Consider the problem WEIGHTED SET COVER. Suppose you're a director of an engineers' team and in charge of a new project, which is to construct a set of transmission tower which collectively covers a given set of points U. One can imagine that the set of points correspond to the ground set of the WEIGHTED SET COVER instance. You're given a set of candidate locations for constructing a tower and each candidate location for a transmission tower will cover certain set of points. You already noticed that each candidate, described by the set of points it covers, is a set $S \in \mathcal{S}$. Constructing a transmission tower at a candidate location (corresponding to 'choosing a set' $S \in \mathcal{S}$ in the set cover) comes with a cost c_S. Your job as a director of engineering is to choose a set of locations for constructing towers so that the transmission spans the entire points while minimizing the cost incurred at the company.

Your boss is not a mathematician, and he is impatient about the cost. You were somehow able to persuade the boss $O(\log n)$ times the optimal cost is the best one can ever dream to achieve. But without knowing the cost of an optimal solution, how can we persuade the boss that the solution you brought up with your team is actually $O(\log n) \cdot \texttt{opt}$? Mind that your boss is not a mathematician. So explaining the greedy algorithm below (from previous lecture) and showing the proof does not work. It only makes him even more irritated.

Algorithm 1 Algorithm for WEIGHTED SET COVER

```
procedure SetCover(U, S, k)
    Mark every element of U as uncovered.
    S' ← ∅ and C ← ⋃_{S∈S'} S
    while U has an uncovered element do
        Select X ∈ S \ S' which minimizes c(X)/|X \ C|.
        price(e) ← c(X)/|X \ C| for every element e ∈ X \ C.
        S' ← S' ∪ {X}
        Mark every element of X covered; C ← C ∪ X.
    return S'.
```

We wrote in the previous lecture note:

> *Consider a sequence $X_1, \ldots, X_\ell$ of sets in $\mathcal{S}$ such that each set X_i covers at least one new element, i.e. $\tilde{X}_i := X_i \setminus \bigcup_{j<i} X_j$ is non-empty. Let us define the* price *of an element e covered in this sequence as*
>
> $$\mathsf{price}(e) := \frac{c(X_i)}{|\tilde{X}_i|}.$$

Recall that the total cost of selected sets by the algorithm equals the sum of prices over all elements.

In fact, the concept of *price* is extremely useful for convincing your boss that you did your job decently. Imagine the cost of a set S is in fact paid to the land owner. The land owner will put a price on each point $u \in U$ so as to justify being paid c_S for a set of points S. The land owner will try to maximize her profit while respecting the budget c_S imposed by the company (if she does not respect the budget constraint c_S, her pricing policy will not be taken seriously). This leads to the canonical maximization problem, which is exactly the dual linear program for the linear programming relaxation of SET COVER.

$$
\begin{aligned}
& \max \sum_{e \in U} y_u & & \\
\text{subject to} \quad & \sum_{e \in S} y_e \leq c_S & & \forall S \in \mathcal{S} \\
& y_e \geq 0 & & \forall e \in U.
\end{aligned}
$$

As a director, you can use the land owner's pricing policy (a *dual feasible solution*) as a justification that your proposed solution (a *primal integral feasible solution*, a set cover in this case) is a reasonable solution; the rationale is "Hey, look the landlady is not a fool and will do her best to maximizer her profit and we need to pay her at least that much. But the solution that our engineers' team proposed is not that far from her best attempt in pricing."

Returning to the greedy heuristic for SET COVER, we want to use the values on U distributed by $\mathsf{price(e)}$. Unfortunately, price is not a feasible dual solution. However,

$$p(e) := \mathsf{price}(e)/H_n$$

is dual feasible. Here $n = |U|$ and H_n is defined as $\sum_{i=1}^{n} 1/i$.

To see that $\{p(e)\}_{e \in U}$ is dual feasible, consider an arbitrary set $S \in \mathcal{S}$. Let $S_1, \ldots, S_\ell$ be the sets chosen by the greedy algorithm and let $W_i = \bigcup_{j \leq i} S_j$. ($W_0 = \emptyset$ by convention.) If an element e is first covered by S_i, then

$$\mathsf{price}(e) = \frac{c_{S_i}}{|S_i \setminus W_{i-1}|}$$

and it holds that

$$\mathsf{price}(e) \leq \frac{c_S}{|S \setminus W_{i-1}|}$$

for any $S \in \mathcal{S}$ with $S \setminus W_{i-1} \neq \emptyset$ by the greedy selection of the sets S_i's. Let k' is the least integer such that $S \subseteq \bigcup_{i=1}^{k'}$. Then

$$
\begin{aligned}
\sum_{e\in S}\frac{\mathsf{price}(e)}{H_n} &= \frac{1}{H_n}\cdot\sum_{i=1}^{k'}\sum_{e\in S\cap S_i}\mathsf{price}(e)\\
&= \frac{1}{H_n}\cdot\sum_{i=1}^{k'}|S\cap S_i|\cdot\frac{c_{S_i}}{|S_i\setminus W_{i-1}|}\\
&\le \frac{1}{H_n}\cdot\sum_{i=1}^{k'}|S\cap S_i|\cdot\frac{c_{S}}{|S\setminus W_{i-1}|}\\
&\le \frac{c_S}{H_n}\cdot(\underbrace{\frac{1}{|S|}+\cdots+\frac{1}{|S|}}_{|S\cap S_1|\text{ times}}+\underbrace{\frac{1}{|S\setminus S_1|}+\cdots+\frac{1}{|S\setminus S_1|}}_{|S\cap S_2|\text{ times}}+\cdots+\\
&\qquad\underbrace{\frac{1}{|S\setminus W_{k'-1}|}+\cdots+\frac{1}{|S\setminus W_{k'-1}|}}_{|S\cap S_{k'}|\text{ times}})\\
&\le \frac{c_S}{H_n}\cdot(\frac{1}{|S|}+\frac{1}{|S|-1}+\cdots\frac{1}{2}+\frac{1}{1})\\
&\le c_S.
\end{aligned}
$$

Define $p_e := \mathsf{price}(e)/H_n$ and observe that $\{p_e\}_{e\in U}$ is dual feasible. Observe that $\sum_{i=1}^{k} c_{S_i} = \sum_{e\in U}\mathsf{price}(e) = H_n\cdot\sum_{e\in U} p_e \le H_n\cdot c(\texttt{opt})$, where opt is an optimal set cover and $c(\texttt{opt})$ is the cost of an optimal set cover. Therefore, the produced solution $S_1,\ldots,S_k$ is within $H_n = O(\log n)$ factor of an optimal set cover.

2 Primal-Dual Method

We saw that a dual feasible solution can be used to bound (upper bound for minimization problem and lower bound) Primal-dual method is a way to use a dual feasible solution, which is improved over the course of the algorithm, as a guidance to find a primal feasible solution. The general idea is to maintain some primal and dual conditions while updating a dual feasible solution and a (not necessarily feasible) primal solution so that when we finally get a primal feasible solution, this is bounded by a good function of the dual solution.

2.1 2-approximation for VERTEX COVER via primal-dual

Note that while performing the procedure **VertexCover** the primal solution x (as an indicator vector of X at hand) and the dual solution at hand satisfy the following conditions.

1. **Dual feasibility:** y is a dual feasible solution.
2. **Primal condition:** if $x_u > 0$, then $\sum_{e\in\delta(u)} y_e = \omega_e$.
3. **Dual condition:** if $y_e > 0$ for $e = uv$, then $x_u + x_v \le 2$.

The dual feasibility condition holds because of line 7. The primal condition is maintained due to the choice of the vertex u at line 8 and the dual condition trivially holds because each edge has exactly two endpoints.

Algorithm 2 Algorithm for VERTEX COVER

procedure VertexCover(G, ω)
  $y_e \leftarrow 0$ for all $e \in E$.
  $X \leftarrow \emptyset$.
  $\triangleright$ y is dual feasible, $x \leftarrow 0$ (as an indicator vector of X) is primal infeasible
  while X is primal infeasible (i.e. not a vertex cover) **do**
    Select an arbitrary edge e not covered by X.
    Increase y_e to the maximum while preserving dual feasibility.
    Let u be an endpoint of e whose dual constraint is tight. (i.e. $\sum_{e\in\delta(v)} y_e = \omega_u$).
    $X \leftarrow X \cup \{u\}$.
  return X.

Let x' and y' be the primal and dual solutions when we complete the while-loop, and note that both are feasible solutions. Now

$$\begin{aligned}
\sum_{v\in V} \omega_v \cdot x'_v &= \sum_{v\in V} x'_v \cdot \Big(\sum_{e\in\delta(v)} y'_e \Big) && \because \text{primal condition} \\
&= \sum_{e=uv\in E} y'_e \cdot \big(x'_u + x'_v\big) && \because \text{rearranging the terms} \\
&\leq \sum_{e=uv\in E} 2 \cdot y'_e && \because \text{dual condition} \\
&\leq 2 \sum_{v\in V} \omega_v \cdot x^*_v && \because \text{feasibility of } y' \text{ and weak duality of LP,}
\end{aligned}$$

where x^* is a optimal fractional solution to the primal LP.

2.2 Algorithm for SHORTEST (s,t)-PATH via primal-dual

Let $\mathcal{S}$ be the collection of all vertex sets $S \subseteq V$ such that $s \in S$ and $t \notin S$. We express the problem SHORTEST (s,t)-PATH by the following integer program:

$$\begin{aligned}
\min \sum_{e\in E} \omega_v \cdot x_v & \\
\sum_{e\in\delta(S)} x_e \geq 1 & \qquad \forall S \in \mathcal{S} \\
x_e \in \{0,1\} & \qquad \forall e \in E
\end{aligned}$$

and the linear programming relaxation is obtained by replacing each constraint $x_e \in \{0,1\}$ by $x_e \geq 0$.

The dual LP is

$$
\begin{aligned}
\max \sum_{S\in\mathcal{S}} y_S & & \\
\sum_{\substack{S\in\mathcal{S}:\\ e\in\delta(S)}} y_S \le \omega_e & & \forall e\in E\\
y_S \ge 0 & & \forall S\in\mathcal{S}.
\end{aligned}
$$

One can easily verify the following observation, implying that the above integer program is indeed a formulation for SHORTEST (s,t)-PATH.

Observation 1. *Let $P \subseteq E$ and let $x^P \in \{0,1\}^E$ be an indicator vector of P, that is,*

$$x_e^P = \begin{cases} 1 & \text{if } e \in P \\ 0 & \text{otherwise.} \end{cases}$$

Then (V,P) contains an (s,t)-path if and only if x^P is a feasible solution to the above integer program.

Algorithm 3 Algorithm for SHORTEST (s,t)-PATH

1: **procedure** **ShortestPath**(G,ω)
2:     **if** s and t are in distinct connected components of G **then return** primal infeasible.
3:     $x_e \leftarrow 0$ for all $e \in E$.
4:     $F \leftarrow \emptyset$.
5:     ▷ y is dual feasible.
6:     ▷ We grow F by adding edges one by one till there is an (s,t)-path in the subgraph (V,F).
7:     **while** there is no (s,t)-path in (V,F) **do**
8:         Let C be the connected component of (V,F) containing s.
9:         Increase y_C to the maximum while preserving dual feasibility.
10:         Let e be an edge of G whose dual constraint is tight. (i.e. $\sum_{S\in\mathcal{S}:e\in\delta(S)} y_S = \omega_e$).
11:         $F \leftarrow F \cup \{e\}$.
12:     **return** the unique (s,t)-path in (V,F).

By $G[F]$ for $\emptyset \neq F \subseteq E$, we denote the subgraph of G on the vertex set

$$V(F) := \{u \in V : u \text{ is an endpoint of some edge of } F\}$$

with F as the edge set. When $F = \emptyset$, we define $G[F]$ as the graph on the singleton $\{s\}$ (with no edges).

Observation 2. *Throughout the execution of the procedure* **ShortestPath***, the graph $G[F]$ is a tree containing s.*

Proof: We prove by induction on $|F|$. When $F = \emptyset$, the claim trivially holds. If $G[F]$ is a tree containing s at the beginning of i-th execution of the while-loop, the set C chosen at line 8 coincides with $V(F)$ by induction hypothesis. Note that whichever edge e is chose at line 10, e belongs to $\delta(S)$ because changing the value of y_S only affects the dual constraint corresponding to an edge incident with S. Therefore we maintain the property that $G[F+e]$ is a tree containing s. □

Let us examine the primal and dual conditions that the procedure **ShortestPath** maintains. Consider $C \subseteq V$ found at line 8 and $v \in C$. Let P_v be the (s,v)-path in (V,F) for $v \in C$, which is unique due to Observation 2, and let $x^{P_v} \in \{0,1\}^E$ be the indicator vector of P_v. y is the current dual solution. Then the following is valid for each $v \in C$.

1. **Dual feasibility:** y is a dual feasible solution.
2. **Primal condition:** $x_e^{P_v} > 0$ implies $\sum_{S \in \mathcal{S}} y_S = \omega_e$.
3. **Dual condition:** $y_S > 0$ implies $\sum_{e \in \delta(S)} x_e^{P_v} = 1$.

The dual feasibility condition holds because of how y is constructed at line 9. The primal condition is derived from that $P_v \subseteq F$ and how the edges of F are chosen at line 10.

To see why the dual condition holds, suppose that there is $S \in \mathcal{S}$ with $y_S > 0$ and $\sum_{e \in \delta(S)} x_e^{P_v} \neq 1$. Then we have $\sum_{e \in \delta(S)} x_e^{P_v} \geq 2$; indeed, $y_S > 0$ implies that we chose S at line 8 at some point and increasing y_S made some edge e tight at line 10. And such an edge e must belong to $\delta(S)$ because otherwise the value of the left-hand side does not change. Therefore, $\sum_{e \in \delta(S)} x_e^{P_v} \geq 1$ and in particular $\sum_{e \in \delta(S)} x_e^{P_v} \geq 2$ by our assumption.

Note that $|\delta(S) \cap P_v| = \sum_{e \in \delta(S)} x_e^{P_v} \geq 2$ and let e_1, e_2 be two edges in $\delta(S) \cap P_v$. Without loss of generality, we assume that e_2 was added to F later than e_1. Let F_2 be the set named as "F" at the iteration when e_2 was added and note that $e_1 \in F_2 \subseteq F$. On the other hand, P_v contains a subpath containing e_1 and e_2 which takes at least one vertex of $V \setminus F_2$. This means that $F_2 \subseteq P_v$ contains a cycle while both F_2 and P_v are subsets of F. This contradicts Observation 2, which says that F is a tree. This completes the proof that the dual condition holds.

Lemma 1. *For each $v \in C$, the (s,v)-path P_v of F is a shortest (s,v)-path of G.*

Proof: (Proof sketch) Consider a linear program for the SHORTEST (s,v)-PATH problem and corresponding dual LP. It is an easy exercise to verify that x^{P_v} and a simple modification of y (how?) are primal and dual feasible solutions satisfying the primal and dual condition. By Complementary Slackness Condition, x^{P_v} is an optimal primal feasible solution. In particular, P_v is a shortest (s,v)-path of G. □

Finally, when the procedure terminates, the returned (s,t)-path at line 12 is a shortest (s,t)-path by Lemma 1.