Approximation-I

Pricing Method and LP

Fahad Panolan

Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India

8 December 2022

Outline

- Approximation algorithms.
- Greedy algorithms. Analysis using pricing method.
- Linear Programming and rounding.

Outline

- Approximation algorithms.
- Greedy algorithms. Analysis using pricing method.
- Linear Programming and rounding.

Problems:

- (Weighted) Vertex Cover
- (Weighted) Set Cover

Coping up with **NP**-completeness

Coping up with **NP**-completeness

- compromise on the running time.
- Restrict the input.
- Compromise on the quality of the output.

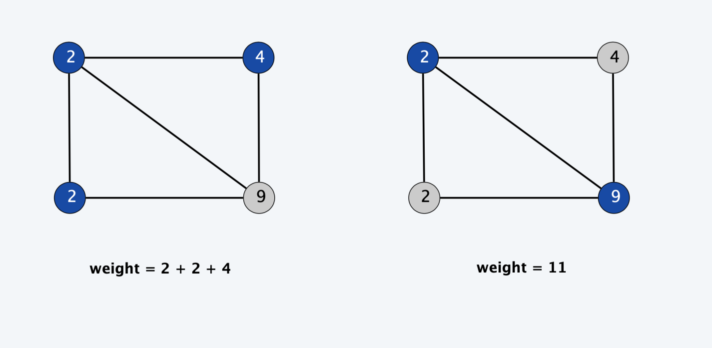
Coping up with **NP**-completeness

- compromise on the running time.
- Restrict the input.
- Compromise on the quality of the output.

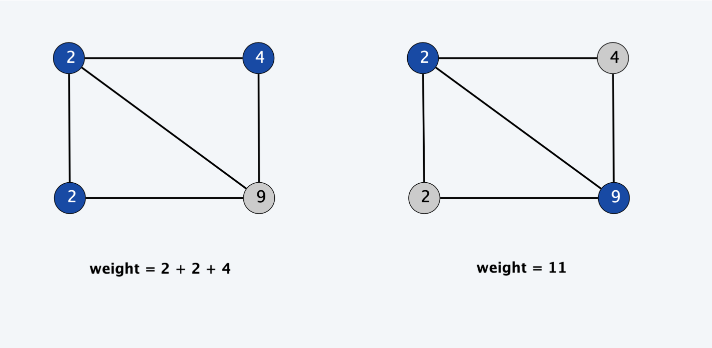
α-approximation algorithm, where $\alpha > 1$

Finds solution in polynomial time that is within ratio α of optimum.

Weighted Vertex Cover



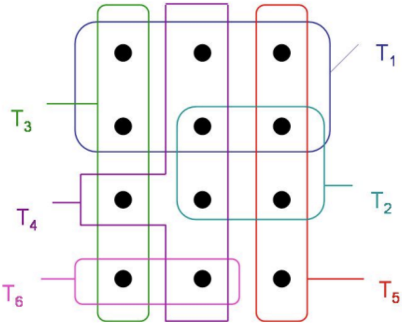
Weighted Vertex Cover



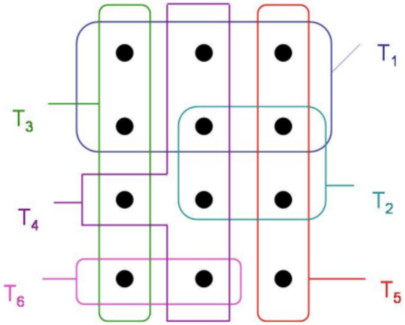
Input: A graph G and $w : V(G) \to \mathbb{R}_{\geq 0}$
Output: A minimum weight vertex cover of G.

Image credit: Algorithm Design by Kleinberg and Tardos

Set Cover



Set Cover



Input: A universe U, family $\mathcal{F}$ of subsets of U, and $w\colon \mathcal{F} \to \mathbb{R}_{\geq 0}$
Output: A minimum weight subfamily of $\mathcal{F}$ covering U.

Image credit: Lecture notes by Amit Chakrabarti

Warm up: Greedy Vertex Cover

Input: A graph G
Output: A minimum cardinality vertex cover of G.

Start with $S := \emptyset$
While $E(G) \neq \emptyset$
    Select an edge $e = (x, y)$.
    $S := S \cup \{x, y\}$.
    $G := G - \{x, y\}$
EndWhile
Return S

Tight analysis?

Tight analysis?

Khoot–Regev 2006

If Unique Games Conjecture is true, then no $(2 - \epsilon)$-approximation algorithm for Vertex Cover

Pricing Method

Excursion Problem modeled as WEIGHTED SET COVER

- Universe $U =$ we.

Excursion Problem modeled as Weighted Set Cover

- Universe $U =$ we.
- Sets in $\mathcal{F}$ are groups who like to go together in a car/bus/train etc.

Excursion Problem modeled as Weighted Set Cover

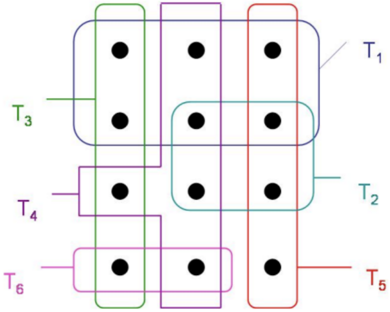
- Universe $U =$ we.
- Sets in $\mathcal{F}$ are groups who like to go together in a car/bus/train etc.
- Weight of the sets corresponds to the cost of the travel.

Excursion Problem modeled as Weighted Set Cover

- Universe $U =$ we.
- Sets in $\mathcal{F}$ are groups who like to go together in a car/bus/train etc.
- Weight of the sets corresponds to the cost of the travel.

Excursion Problem modeled as WEIGHTED SET COVER

- Universe U = we.
- Sets in $\mathcal{F}$ are groups who like to go together in a car/bus/train etc.
- Weight of the sets corresponds to the cost of the travel.



How to solve?

Greedy Algorithm

Start with $R := U$ and no sets selected

While $R \neq \emptyset$

    Select S_i that minimizes $\frac{w(S_i)}{|S_i \cap R|}$

    Each $x \in S_i \cap R$ pay a cost $c_x = \frac{w(S_i)}{|S_i \cap R|}$

    $R := R \setminus S_i$

EndWhile

Return the selected sets

Analysis: Approximation Ratio

- Let $\mathcal{C}$ be the ouput of the algorithm.

Analysis: Approximation Ratio

- Let $\mathcal{C}$ be the ouput of the algorithm.
- Then, $\sum_{S\in\mathcal{C}} w(S) = \sum_{x\in U} c_x$.

Analysis: Approximation Ratio

- Let $\mathcal{C}$ be the ouput of the algorithm.
- Then, $\sum_{S \in \mathcal{C}} w(S) = \sum_{x \in U} c_x$.
- Now, for each $S \in \mathcal{F}$, we upper bound $\sum_{y \in S} c_y$.

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.
- At the beginning of this iteration
  - $y_j, y_{j+1}, \ldots, y_d \in R$.

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.
- At the beginning of this iteration
  - $y_j, y_{j+1}, \ldots, y_d \in R$.
  - $|S \cap R| \geq d - j + 1$.

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.
- At the beginning of this iteration
  - $y_j, y_{j+1}, \ldots, y_d \in R$.
  - $|S \cap R| \geq d - j + 1$.
  - $\frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$.

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.
- At the beginning of this iteration
  - $y_j, y_{j+1}, \ldots, y_d \in R$.
  - $|S \cap R| \geq d - j + 1$.
  - $\frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$.
- Suppose Q is the set selected in this iteration.

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.
- At the beginning of this iteration
  - $y_j, y_{j+1}, \ldots, y_d \in R$.
  - $|S \cap R| \geq d - j + 1$.
  - $\frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$.
- Suppose Q is the set selected in this iteration.
- $c_{y_j} = \frac{w(Q)}{|Q \cap R|} \leq \frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.
- At the beginning of this iteration
  - $y_j, y_{j+1}, \ldots, y_d \in R$.
  - $|S \cap R| \geq d - j + 1$.
  - $\frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$.
- Suppose Q is the set selected in this iteration.
- $c_{y_j} = \frac{w(Q)}{|Q \cap R|} \leq \frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$
- $\sum_{y \in S} c_y = \sum_{j=1}^{d} c_{y_j}$

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.
- At the beginning of this iteration
  - $y_j, y_{j+1}, \ldots, y_d \in R$.
  - $|S \cap R| \geq d - j + 1$.
  - $\frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$.
- Suppose Q is the set selected in this iteration.
- $c_{y_j} = \frac{w(Q)}{|Q \cap R|} \leq \frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$
- $\sum_{y \in S} c_y = \sum_{j=1}^{d} c_{y_j} \leq \sum \frac{w(S)}{d-j+1} = w(S) \left(\frac{1}{d} + \frac{1}{d-1} + \ldots + 1 \right)$

Upper bound: $\sum_{y \in S} c_y$

- Let $S = \{y_1, \ldots, y_d\}$ and it is the order they removed from R.
- Consider the iteration where y_j is covered by the algorithm.
- At the beginning of this iteration
  - $y_j, y_{j+1}, \ldots, y_d \in R$.
  - $|S \cap R| \geq d - j + 1$.
  - $\frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$.
- Suppose Q is the set selected in this iteration.
- $c_{y_j} = \frac{w(Q)}{|Q \cap R|} \leq \frac{w(S)}{|S \cap R|} \leq \frac{w(S)}{d-j+1}$
- $\sum_{y \in S} c_y = \sum_{j=1}^{d} c_{y_j} \leq \sum \frac{w(S)}{d-j+1} = w(S) \left(\frac{1}{d} + \frac{1}{d-1} + \ldots + 1 \right) = H(d) \cdot w(S)$

Final Step

- Let $\mathcal{C}^\star$ is a optimum solution and $d^\star = \max_{S \in \mathcal{F}} |S|$.

Final Step

- Let $\mathcal{C}^\star$ is a optimum solution and $d^\star = \max_{S \in \mathcal{F}} |S|$.
- For each $S \in \mathcal{C}^\star$, $\sum_{y \in S} c_y \leq w(S) \cdot H(d^\star)$

Final Step

- Let $\mathcal{C}^\star$ is a optimum solution and $d^\star = \max_{S \in \mathcal{F}} |S|$.
- For each $S \in \mathcal{C}^\star$, $\sum_{y \in S} c_y \leq w(S) \cdot H(d^\star)$

$$\sum_{x \in U} c_x \leq \sum_{S \in \mathcal{Q}^*} \sum_{y \in S} c_y$$

Final Step

- Let $\mathcal{C}^\star$ is a optimum solution and $d^\star = \max_{S \in \mathcal{F}} |S|$.
- For each $S \in \mathcal{C}^\star$, $\sum_{y \in S} c_y \leq w(S) \cdot H(d^\star)$

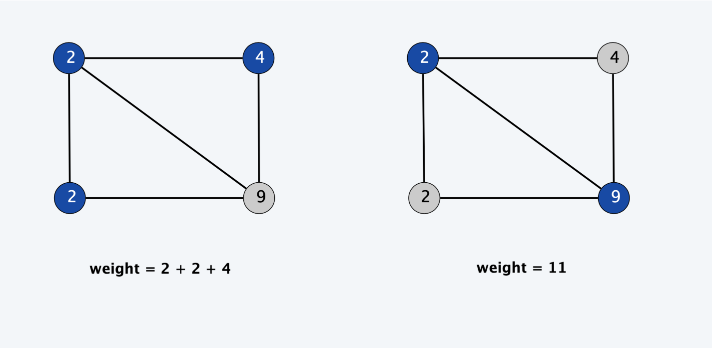
$$
\begin{aligned}
\sum_{x \in U} c_x &\leq \sum_{S \in \mathcal{Q}^*} \sum_{y \in S} c_y \\
&\leq \sum_{S \in \mathcal{Q}^*} w(S) \cdot H(d^\star)
\end{aligned}
$$

Final Step

- Let $\mathcal{C}^\star$ is a optimum solution and $d^\star = \max_{S \in \mathcal{F}} |S|$.
- For each $S \in \mathcal{C}^\star$, $\sum_{y \in S} c_y \leq w(S) \cdot H(d^\star)$

$$
\begin{aligned}
\sum_{x \in U} c_x &\leq \sum_{S \in \mathcal{Q}^*} \sum_{y \in S} c_y \\
&\leq \sum_{S \in \mathcal{Q}^*} w(S) \cdot H(d^\star) \\
&= H(d^\star) \cdot \mathsf{opt} \\
&\leq 2(\log d^\star) \cdot \mathsf{opt}
\end{aligned}
$$

Weighted Vertex Cover



Input: A graph G and $w: V(G) \to \mathbb{R}_{\geq 0}$
Output: A minimum weight vertex cover of G.

Image credit: Algorithm Design by Kleinberg and Tardos

How to model Vertex Cover as Set Cover?

Let (G, w) be the input of Weighted Vertex Cover

How to model Vertex Cover as Set Cover?

Let (G, w) be the input of Weighted Vertex Cover

- $U = E(G)$

How to model VERTEX COVER as SET COVER?

Let (G, w) be the input of WEIGHTED VERTEX COVER

- $U = E(G)$
- $\mathcal{F} = \{S_v \ : \ v \in V(G)\}$, where S_v is the set of edges incident on v.

How to model Vertex Cover as Set Cover?

Let (G, w) be the input of Weighted Vertex Cover

- $U = E(G)$
- $\mathcal{F} = \{S_v \,:\, v \in V(G)\}$, where S_v is the set of edges incident on v.

Result on Weighted Set Cover implies that there is $2 \log \Delta$-approximation algorithm for Weighted Vertex Cover, where Δ is the maximum degree.

How to model Vertex Cover as Set Cover?

Let (G, w) be the input of Weighted Vertex Cover

- $U = E(G)$
- $\mathcal{F} = \{S_v \,:\, v \in V(G)\}$, where S_v is the set of edges incident on v.

Result on Weighted Set Cover implies that there is $2 \log \Delta$-approximation algorithm for Weighted Vertex Cover, where Δ is the maximum degree.

Next: 2-approximation algorithm for Vertex Cover.