Approximation Algorithms for NP-Complete Problems

Mr. Ajaya Kumar Dash

ajaya@iiit-bh.ac.in
http://www.iiit-bh.ac.in/

Department of Computer Science and Engineering

IIIT, Bhubaneswar

April 6, 2017

Outline of This Lecture

- Basic Concepts of Approximation Algorithm.
- Approximation Algorithm for *Vertex Cover Problem*.

The Idea Behind Approximation Algorithms

- **NP-Complete problems** $\Rightarrow$ Very likely no polynomial-time algorithm to find an optimal solution.

- **Idea :** Develop polynomial-time algorithms to find near-optimal solutions.

- **E.g.** Develop a greedy algorithm without proving the greedy choice property and optimal substructure.
- Are those solution found near-optimal?
- How **near** are they?

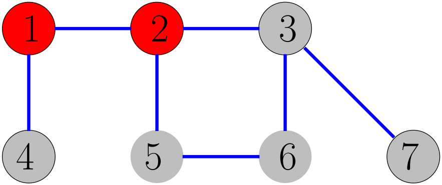
- **Approximation Ratio** $\rho(n)$
  - Define the cost of the optimal solution as C^*.
  - The cost of the solution produced by a approximation algorithm is C.
  - $\rho(n) \geq max(\frac{C}{C^*}, \frac{C^*}{C})$.
- This approximation algorithm is then called a $\rho(n)$-approximation algorithm.

- **E.g.**
  - A given graph G has a MST of weight 20.
  - An algorithm can produce some spanning trees, and they are not MSTs, but their total weights are always smaller than 25.
  - What is the *approximation ratio*?
    - $\frac{25}{20} = 1.25$
  - What this algorithm is called?
    - A 1.25-approximation algorithm.

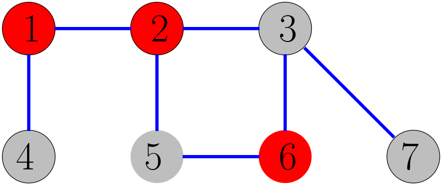
- What if $\rho(n) = 1$?

- It is an algorithm that can always find a optimal solution.

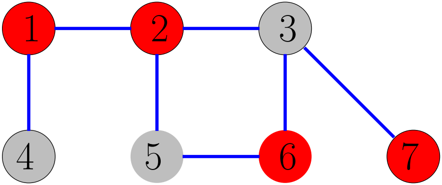
- **Definition of vertex cover**
  - Given a undirected graph $\boldsymbol{G = (V, E)}$. The vertex cover $\boldsymbol{V'}$ of $\boldsymbol{G}$ is:
    - $V' \subseteq V$.
    - For each edge $\{u, v\} \in E$ either $u \in V'$ or $v \in V'$ or both.
- The size of the vertex cover is $|v'|$.
- The aim is to find a vertex cover with minimum size.



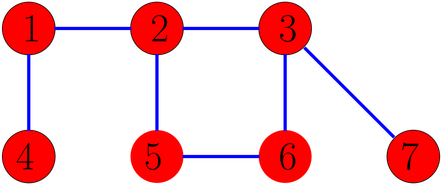
- Are the red vertices form a vertex cover ?
- No!! **Why**?
- Edges $\{5, 6\}$, $\{3, 6\}$ and $\{3, 7\}$ are not covered by it.



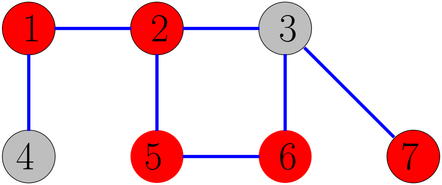
- Are the red vertices form a vertex cover ?
- No!! **Why**?
- Edge $\{3, 7\}$ is not covered by it.



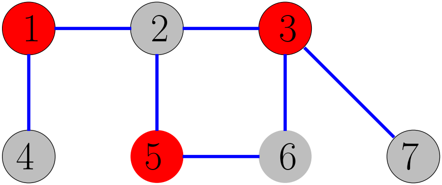
- Are the red vertices form a vertex cover ?
- Yes!!
- What is the size ?? **4**



- Are the red vertices form a vertex cover ?
- Yes!!
- What is the size ?? **7**



- Are the red vertices form a vertex cover ?
- Yes!!
- What is the size ?? $\mathbf{5}$



- Are the red vertices form a vertex cover ?
- Yes!!
- What is the size ?? $\mathbf{3}$

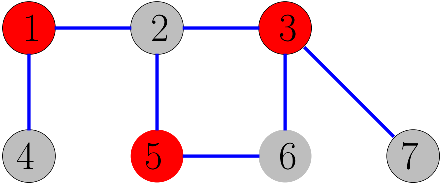


Figure: A minimum vertex cover

- **Problem statement**
  - Given a undirected graph, find a vertex cover with minimum size.

- Vertex cover problem is NP-Complete.
- A polynomial time 2-approximation algorithm is as follows

APPROX-VERTEX-COVER(G)

$C \leftarrow \phi$
$E' \leftarrow E[G]$
while $E' \neq \phi$
  do let (u, v) be an arbitrary edge of E'
    $C \leftarrow C \cup \{u, v\}$
    remove from E' every edge incident on either u or v.
return C

The Approximate Vertex Cover

```
APPROX-VERTEX-COVER(G)
C ← φ
E' ← E[G]
while E' ≠ φ
      do let (u, v) be an arbitrary edge of E'
         C ← C ∪ {u, v}
         remove from E' every edge incident on either u or v.
return C
```

- The running time is $O(V + E)$ using adjacency list representation for E'.
- The above algorithm is a 2 - approximation algorithm.
- It means when size of minimum vertex cover is S, the size of Approx-Vertex-Cover(G) is bounded above by $\mathbf{2}S$.

THE