Vertex-Cover Problem

Prepared By:

M.MANOGNA VEENA
21MPC1001
SCOPE

Updated By:

Dr. Shruti Mishra
SCOPE

Contents

- ❖ Approximation Algorithm
- ❖ Problems
- ❖ Approximation Scheme
- ❖ Vertex Cover Problem
- ❖ Steps
- ❖ Example
- ❖ Performance Proofs

Approximation Algorithms

An approximation algorithm is one of the ways of approach of **NP-Completeness** for the optimization problem and it returns near-optimal solutions.

Objective :

- Here we see the polynomial-time approximation algorithms for several NP-complete problems.
- We define an optimal solution for a problem either a maximization or a minimization problem.

Approximation Ratio:

- For any input of size of n, the cost C of the solution produced by the algorithm is within a factor of ρ(n) of the cost C* of an optimal solution:

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \le \rho(n) .$$

Contd..

- If an algorithm achieves an approximation ratio of ρ(n), we call it a ρ(n)-approximation algorithm.
- The definitions of the approximation ratio and of a ρ(n) approximation algorithm apply to both minimization and maximization problems.
- These algorithms are also called as **Heuristic algorithm.**
- For vector-cover problems,

i. Optimization problem is to find the vertex cover with fewest vertices

ii. Approximation problem is to find the vertex cover with few vertices.

Problem

Minimization problem: $0 < C^* \leq C$

Ratio – C/C^* - factor by which cost of approximate solution is larger than optimal solution.

Maximization problem: $0 < C \leq C^*$

Ratio – C^*/C – factor by which the cost of an optimal solution is larger than the cost of the approximate solution.

Approximation Scheme

f(n)- approximate algorithm:

- An algorithm which generates feasible solution if and only if for every instance I of size n, $(C^*(I)-C(I)) / C^*(I) <= f(n)$.
- It is assumed that $C^*(I) > 0$.
- An approximation scheme is a polynomial time approximation scheme iff, for every $\epsilon > 0$ it has a computing time that is polynomial in the problem size.

Vector-cover Problem

- A vertex cover of an undirected graph G = (V,E) is a subset V’ ⊂ V such that if (u,v) is an edge of G, then either u ϵ V’ or v ϵ V’ (or both). The size of a vertex cover is the number of vertices in it.
- The vertex-cover problem is to find a vertex cover of minimum size in a given undirected graph.
- We call such a vertex cover an optimal vertex cover.
- This problem is the optimization version of an NP-complete decision problem.
- The following approximation algorithm takes as input an undirected graph G and returns a vertex cover whose size is guaranteed to be no more than twice the size of an optimal vertex cover.

Steps

Step 1: Variable C vertex cover being constructed . Initially, C is an empty set.

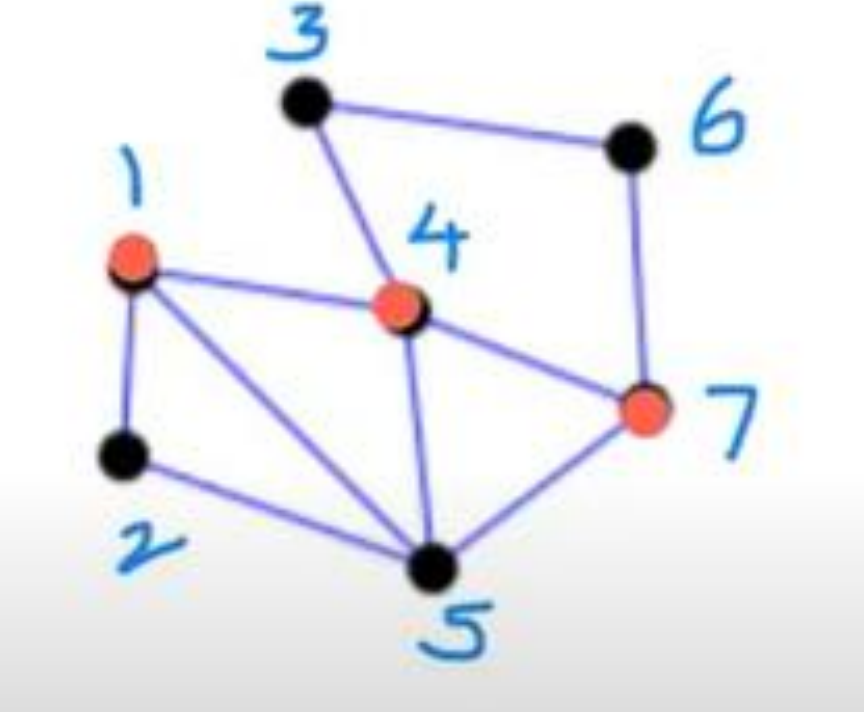
Step 2: Set E’ to be a copy of the edge set E(G) of the graph.

Step 3: From step 3, the loop is repeatedly, picks an edge (u,v) from E , adds its endpoints u and v to C , delete all edges in E’ that are covered by either u or v.

Example

- Here, initially C is an empty set.
- E' = {(1,2) (1,4) (1,5) (2,5) (3,4) (3,6) (4,7) (4,5) (5,7) (6,7)}

C={ }



[contd..]

- C={1,2}

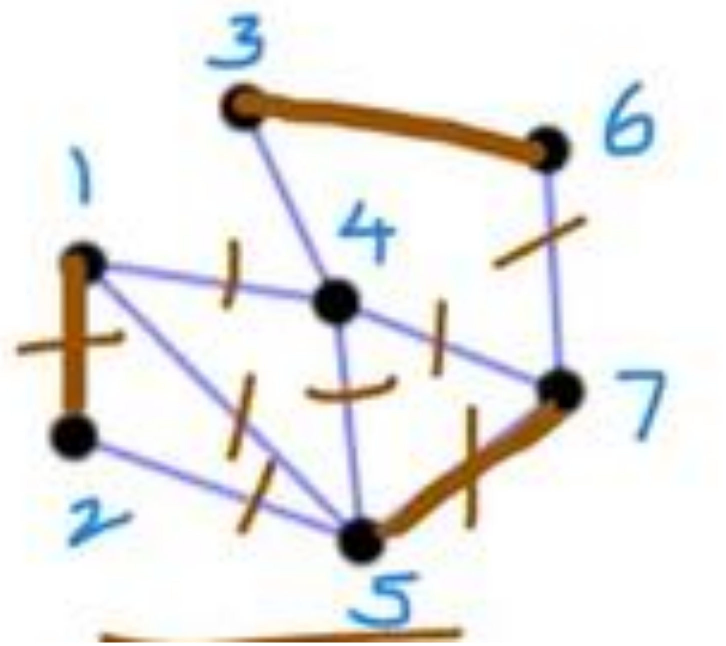
E'={(3,4) (3,6) (4,5) (4,7) (5,7) (6,7)

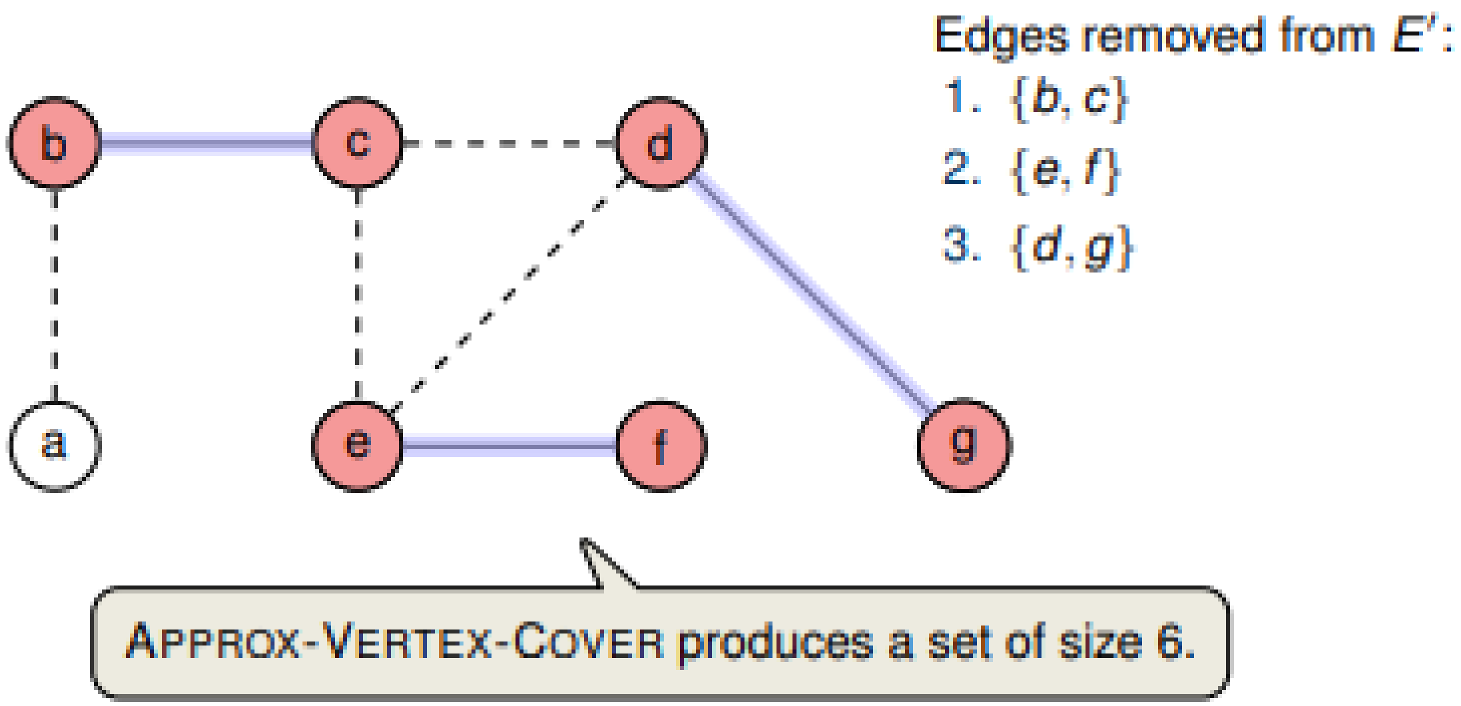
- C={1,2,5,7}

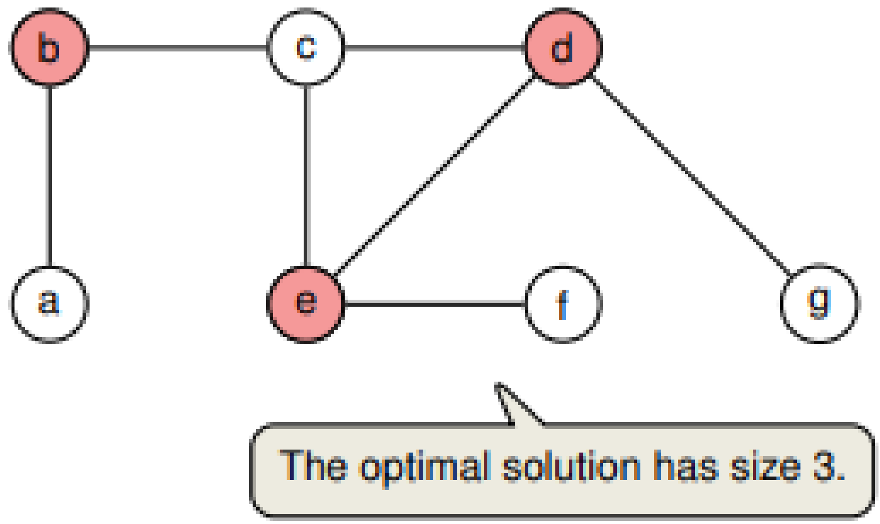
E'={(3,4) (3,6)}

- C={1,2,5,7,3,6}

E'={NULL}







Optimal is calculated by considering the min. vertex that covers max. no. of edges.

Algorithm

APPROX-VERTEX-COVER(G):

1 C = Φ

2 E’= G.E

3 **while** E’ ≠ Φ

4 let (u,v) be an arbitrary edge of E’

5 C= C U {u,v}

6 remove from E’ every edge incident on either u or v

7 **return** C

Contd..

- So, the approximate solution return by the algorithm contains 6 vertices and represented as

  |C|={1,2,5,7,3,6}

- Other observation is that the edges chosen by this algorithm are independent edges and doesn't share common vertex.

Maximal Matching:

A matching becomes a maximal matching if no more edges can be added to it.

C = {1,2,3,6,5,7}

M = {(1,2) (3,6) (5,7)}

|C| = 6 ; |M| = 3

The no.of elements in the vertex cover set given by |C| is always equal to 2 times the elements present in the maximal matching set.

|C| = 2 |M|

Performance Proofs

- The edges picked by the algorithm is a maximal matching M. Hence, C is a vertex cover.
- The algorithm runs in time polynomial of input size.
- The optimum vertex cover C* must cover every edge in M.
- Hence C* contains at least one of the end points of each edge in M,

$$|C^*| \geq |M|$$

- $|C| = 2 * |M| \leq 2 * |C^*|$ where C* is an optimal solution.
- Thus there is a 2-factor approximation algorithm for vertex cover problem.

$$|C| \leq 2\,|C^*|$$