Approximation Algorithm

PAGE NO.:

An approximation algo. is one of the ways of approach of NP-COMPLETENESS for the optimization problem & it returns near optimal soln.

Goal: To come as close as possible to the optimum value in a reasonable amount of time which is at the most polynomial time. This technique does not guarantee the best solution.

→ Approximation algo. are also called Heuristic algorithm.

For vertex cover problem:

→ Optimization problem is to find the vertex cover with fewest vertices

→ Approximation problem is to find vertex cover with few vertices

Approximation ratio

$p(n)$ = Approximation ratio
n = Input size
C = Cost of solution
C^* = Cost of optimal solution

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq p(n)$$

An algorithm that achieves $p(n)$ approximation ratio is called $p(n)$-approx. algorithm.

Minimization problem

$0 < C^* \leq C$

$\frac{C}{C^*} \geq 1$ { factor by which cost of approx. soln. is larger than optimal soln.

Maximization problem

$0 < C \leq C^*$

$\frac{C^*}{C} \geq 1$ { factor by which cost of an optimal soln. is larger than the cost of the approximate soln.

Approximation scheme

An approximation scheme is an approximation algorithm, which given any input & $\varepsilon > 0$, is a $(1+\varepsilon)$-approx. algo.

→ If is a polynomial-time approx. scheme (PTAS) if for any fixed $\varepsilon > 0$, the runtime is polynomial in n. ~~(for e.g. $O(n^{2/\varepsilon})$)~~ (for e.g. $O(n^{2/\varepsilon})$):

→ If is a fully polynomial-time approximation scheme if runtime is polynomial in both $1/\varepsilon$ & n.

Introduction:

A vertex cover of an undirected graph is $G = (V, E)$ is subset $V' \subseteq V$ such that if (u, v) is an edge of G, then either $u \in V'$ or $v \in V'$ (or both). The size of a vertex cover is the no. of vertices in it.

→ The vertex cover problem is to find a vertex cover of minimum size in a given undirected graph

→ Minimum vertex cover is called as an optimal vertex cover.

→ This problem is the optimization version of an NP-complete decision problem.

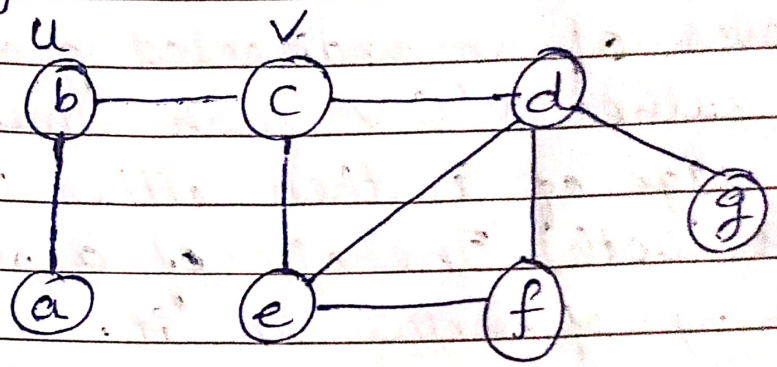
Algorithm:

S-1: Variable C vertex cover being initially constructed. Initially C is an empty set.

S-2: Set E' to be a copy of the edge set E[G] of the graph

S-3-6) Loop repeatedly,
picks an edge (u, v) from E,
add its endpoints u & v to ~~set~~ C,
deletes all edges in E' that are covered by either u or v.

E.g.

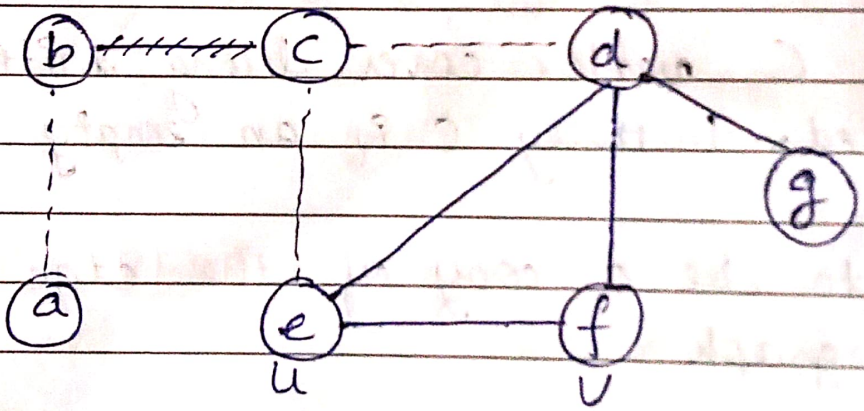


~~C = { } C = { }~~ $E' = \{(a,b), (b,c), (c,d), (c,e), (d,e), (d,f), (e,f), (d,g)\}$

Pick any arbitrary edge

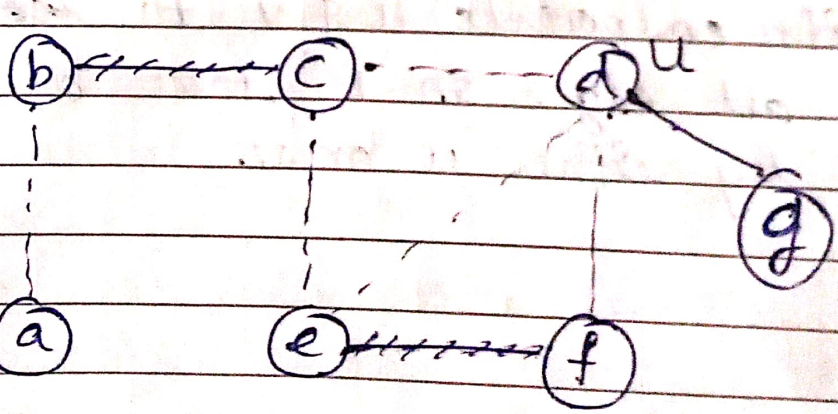
Pick (b, c)

~~C = {(b, c)}~~ $C = \{b, c\}$ $E' = \{$~~(b,c)~~, (d, e), (d, f), (e, f), (d, g)$\}$



Pick (e, f)

$C = \{b, c, e, f\}$ $E' = \{(d, g)\}$



Pick (d, g)

$C = \{b, c, e, f, d, g\}$

~~$C = \{b, c, e, d, g\}$~~ $E' = \{\}$

Vertex cover $\therefore C = 6$

But the optimal soln. is

$C = \{b, d, e\}$ $\therefore C^* = 3$

$\therefore$ For minimization problem

$$\frac{C}{C^*} \geq 1$$

$$\therefore \frac{6}{3} \geq 1 \quad \div$$

$2 \geq 1$ $\therefore$ Hence proved

T.C = O(V+E)

S.C. = O(V)

Set Cover Problem (Approximation algg.)

Problem statement: Given a universe U of n elements & a collection of S of m subsets of U, the set cover problem is to find the minimum number of subsets from S that cover all element in U.

Given: set X of size n & family of subsets F

Goal: Find a minimum size subset $C \subseteq F$ (Nq. of sets (& not elements)) s.t. $X = \bigcup_{S \in C} S$

Solvable only if $\bigcup_{S \in F} S = X$!

Steps → Initialize the set cover C to be empty
→ While there are uncovered elements in U, select the subset S_i from S that covers most uncovered elements, & do
$C = C \cup S_i$
$U = U - S_i$

E.g. $U = \{a, b, c, d, e, f, g, h\}$

$S_1 = \{a, b, c\}$ $S_2 = \{b, d, f, g\}$
$S_3 = \{a, e, f, g\}$ $S_4 = \{c, d, e, h\}$
$S_5 = \{a, h\}$ $S_6 = \{b, c, f, h\}$

→ Pick S_2
$\therefore C = C \cup S_2$ $U = \{a, c, e, h\}$
$= \{b, d, f, g\}$
$= \{S_2\}$

→ Pick S_4

$C = C \cup S_4$ $\quad U = \{a\}$

$C = \{\{S_2, S_4\}\}$

→ Pick ~~S_1~~ S_5

$C = C \cup S_5$ $\quad U = \{\}$

$= \{S_2, S_4, S_5\}$

Q2) Let $U = \{1,2,3,4,5\}$ $\quad S = \{S_1, S_2, S_3\}$

& $S_1 = \{4,1,3\}$ $\quad$ cost = 5

$S_2 = \{2,5\}$ $\quad$ cost = 10

$S_3 = \{1,4,3,2\}$ $\quad$ -11- 3

Find the min. cost set cover

Steps: 1) Let I represent set of elements added so far. Initialize $I = \{\}$

2) Do the following until I is not same as U.

a) Find the set S_i in $\{S_1, S_2, S_3 \ldots S_m\}$ whose cost effectiveness is smallest
i.e. ratio of cost $C(S_i)$ & no. of newly added elements is minimum

$\therefore$ ~~$\frac{Cost(S_i)}{|S_i|}$~~ $\quad \therefore \frac{Cost(S_i)}{|S_i - I|}$ should be minimum

b) Do, $I = I \cup S_i$

Cost effectiveness (CE) : if I is empty

$CE_1 = \frac{C(S_1)}{|S_1 - I|} = \frac{5}{3}$

$= \frac{5}{3}$

$CE_1 = 1.667$

$CE_2 = \frac{C(S_2)}{|S_2 - I|} = \frac{10}{2} = 5$

$CE_2 = 5$

$CE_3 = \frac{C(S_3)}{|S_3 - I|} = \frac{3}{4} = 0.75$

$CE_3 = 0.75$

∴ Pick S_3 first, ∵ CE_3 is minimum

∴ $I = \{1, 2, 3, 4\}$ $U = \{5\}$

Now, : $CE_2 = \frac{C(S_2)}{|S_2 - I|} = \frac{10}{1} = 10$

$CE_3 = \frac{C(S_3)}{|S_3 - I|} = \frac{3}{1}$

$CE_1 = \frac{C(S_1)}{|S_1 - I|} = \frac{5}{0} = \infty$

∴ Pick S_2

∴ $I = \{1, 2, 3, 4, S\}$ $\quad U = \{\}$

∴ $I = \{S_3, S_2\}$