Extending the limits of Tractability

Chapter-10.1

Vertex cover

Given graph $G = (V, E)$ and an integer k, find vertex cover of size almost k. We find $S \subseteq V$ where $|S| \leq k$, every edge $e \in E$ has at least one end in S.

$|V| = S = \{a, b, c, d, e\}$

$k = 2$

then the no. of ~~vertexcover~~ subset $= {}^5C_2$

$= 10$.

So the no. of subset $= {}^nC_k$ where n = no. of vertices
k = size of the vertex cover.

⇒ Vertex cover has two parameters (n, k)
where n = no. of vertices in the graph G ⇒ $n = |V|$.
k = size of the vertex cover / allowable size of the vertexcover.

⇒ Typically if $k = 2$ or $k = 3$ then in this case the problem can be solved in polynomial time. i.e $O(kn)$

⇒ Now try all subsets of 'V' of size 'k' and see whether any of them is a vertexcover. There are ${}^nC_k$ i.e $\binom{n}{k}$ subsets and checking each subset ~~them~~ a vertex cover or not take $O(kn)$ time.

⇒ Total time $= O(kn) \times {}^nC_k = O(kn \cdot n^k)$

$O(k \cdot n^{k+1}) = O(k \cdot n^{k+1})$

$\because O(n^k) = O({}^nC_k)$

→ The intractability of vertex cover only sets in for real, once k grows a function of n.

→ Example

let n = 1000 and k = 10

→ this will take at least 10^{34} time units. which is very very large quantity of time (i.e longer than the life of universe)

- How to make it more tractable that is practically viable when k is small constant ?

let us plan for algorithm running time bound of $O(2^k \cdot kn)$. In this case if n=1000 and k=10 then time estimated is feasible i.e 2^k is more appealing then n^k.

- Idea behind this algorithm

consider any arbitrary edge e(u,v) in G. ~~using~~ in an ~~edge~~ any ~~vertex cover~~ k-node vertex cover s of G, one of u or v must belongs to s.

Suppose $u \in s$:

if we ~~the~~ delete u and all its incident edges then it must be possible to cover remaining edges by at most k-1 nodes.

let G-{u} be the graph after deleting node u and any its incident edges. There must be a vertex cover of size at most k-1 in a G-{u}.

→ suppose $v \in s$:

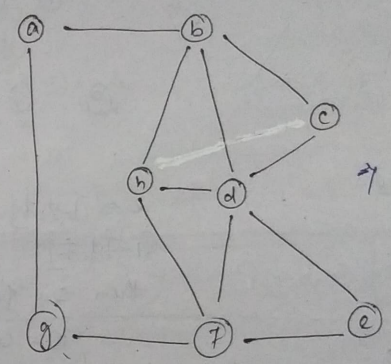
similar to above case, there must be a vertex cover of at most k-1 in G-{v}

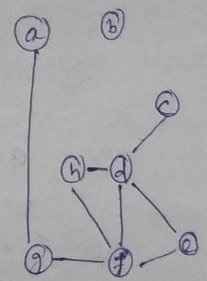
Example $O(2^k \cdot kn)$

V = {a, b, c, d, e, f, g}

S = {d, b, g, f}

k = 4





G-{b}

let (u,v) = (b,c) ∈ S

then b ∈ S

then we delete 'b' from the 's' then

s = {d, ~~b~~, g, f} = {d, g, f} = k-1

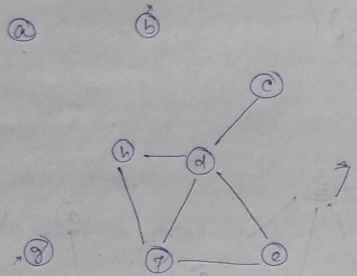
= 4-1 = 3

then S = {d, g, f} is another subgraph that is the vertex cover of the graph G.

simultaneously we delete the vertex from the vertex cover. i.e

e = (g, a) then s = {d, g, f} = {d, f}

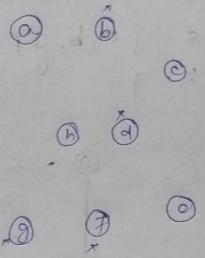
= k-1 = 3-1 = 2 is the vertex cover.



$g - \{g\}$

$K = 2$

$S = \{d, f\}$



$e = \{d, f\}$

$G - \{d, f\}$

then $S = \{d, f\} - \{\} = \{\}$

$K = 0$

Algorithm to search a K-node vertex cover in G:

(1) if in G, $|E| = 0$, then vertex cover is empty set.

(2) if in G, $|E| > K \cdot |V|$, then it has no K-node vertex cover.

(3) else

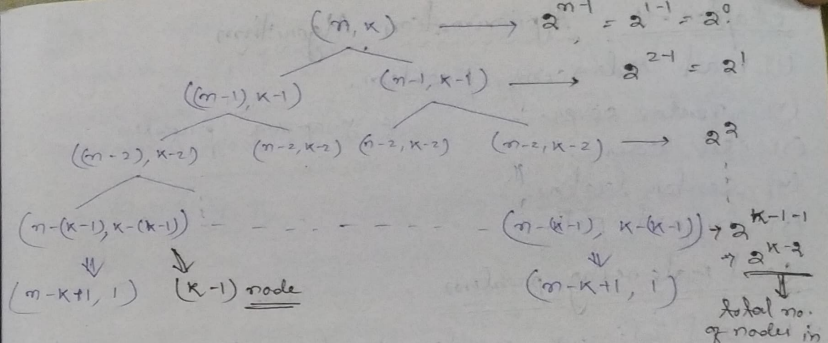
(4) let $e = (u, v) \in E$

(5) recursively check: If either of $G - \{u\}$ or $G - \{v\}$ has a vertex cover of size $K-1$;

(6) If neither of them does, then G has no K-node vertex cover.

(7) Else

(8) one of them say $(G - \{x\})$ has a $(K-1)$ node vertex cover T. So $T \cup \{x\}$ is the K-node vertex cover of G // Here x is either u, or v.



Time complexity

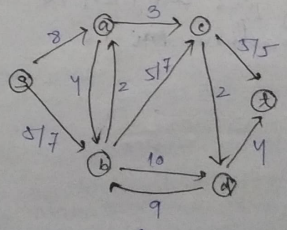
$$O(Kn \cdot 2^{K-2}) \approx O(Kn \cdot 2^K)$$

⇒ If $K = 40$ then this problem is also intractable

Challenge

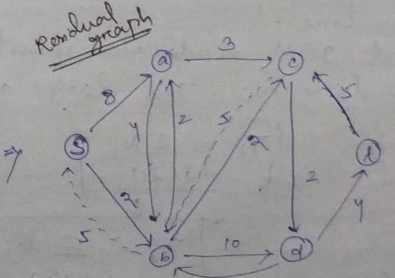
Through there is some improvement for getting a solution when $n = 1000$ and $K = 1, 2, \ldots, 10$ but when $K = 40$ finding a solution is intractable again.

Residual graph



$s \to b \to c \to t \Rightarrow 5$

Residual graph



Residual capacity = forward edge

Flow capacity = backward edge