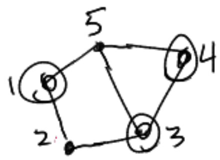
RECALL:

<u>Vertex Cover</u>: Given a graph $G = (V, E)$ find a min. size <u>vertex cover</u> — a set $U \subseteq V$ s.t. every edge has at least one endpoint in U.



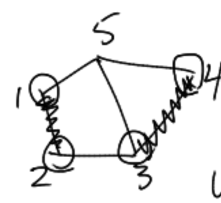
this is min. vertex cover.

RECALL

Greedy alg. finds Vertex Cover $\leq O(\log n) \cdot OPT$

A different approx. alg. for Vertex Cover

```
C ← ∅
while E ≠ ∅
    pick e = (u,v) ∈ E
    C ← C ∪ {u,v}
    remove all edges incident to u or v.
end
```



alg. finds vertex cover of size 4.

This stupid alg. is better than the greedy alg.

<u>Lemma</u> This alg. gives $|C| \leq 2 \cdot OPT$

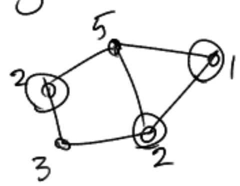
<u>Pf.</u> set of edges we pick is a matching

$|C| = 2|M|$ M is the edges we choose — matching

$|M| \leq OPT$ because every matching edge needs its own vertex in the opt vertex cover.

Best approx. factor known for Vertex Cover is 2.
[via poly. time alg.]

Weighted Vertex Cover



given weights on vertices $w: V \to \mathbb{R}^+$
find vertex cover C of min weight

$$\min \sum_{v \in C} w(v)$$

Express as integer linear program.
variable $x(v)$ for each vertex v.

$$x(v) = \begin{cases} 1 & \text{— choose } v \text{ in cover} \\ 0 & \text{— don't.} \end{cases}$$

$$\boxed{\begin{array}{l} \min \sum_{v \in V} w(v) \cdot x(v) \\ \text{constraints } \forall \text{ edge } e=(u,v) \quad x(u)+x(v) \geq 1 \\ x(v) = 0 \text{ or } 1 \end{array}}$$

Integer Linear Program — solns are exactly min. weight vertex covers.
(this is an NP-complete problem.

But Linear Programming has poly. time alg. (ellipsoid or interior point — or simplex is practical though not poly. time).

So if we relax $x(v) = 0$ or 1
to $0 \leq x(v) \leq 1$
we can solve in poly. time.

Claim Rounding the fractional soln gives a Vertex Cover.
more exactly: Suppose $\bar{x}$ is opt. solution to Linear Program
Let $C = \{ v \in V : \bar{x}(v) \geq \frac{1}{2} \}$

Lemma: C is a vertex cover and $w(C) \leq 2 \cdot OPT$

Lemma C is a vertex cover and $w(C) \leq 2 \cdot OPT$.

Pf. For edge $e = (u, v)$

$$\bar{x}(u) + \bar{x}(v) \geq 1$$

so at least one of u, v has $\bar{x}(\cdot) \geq \frac{1}{2}$
and we put that vertex in C.

$\therefore$ C is a vertex cover.

$$OPT = OPT \text{ to } \underset{\text{integer}}{I}LP \geq OPT \text{ to } LP$$

because the LP allows more solutions so OPT goes down.

$$= \sum w(v)\bar{x}(v) \quad \text{because } \bar{x} \text{ was opt sol}^{\text{n}} \text{ to LP}$$

$$\geq \sum_{v,\ \bar{x}(v) \geq \frac{1}{2}} w(v)\bar{x}(v)$$

$$\geq \sum_{v,\ \bar{x}(v) \geq \frac{1}{2}} w(v)\frac{1}{2} = \frac{1}{2} \sum_{v,\ \bar{x}(v) \geq \frac{1}{2}} w(v) = \frac{1}{2} w(C)$$

So $OPT \geq \frac{1}{2} w(C)$

$w(C) \leq 2 \cdot OPT$.

Summary: 3 approx algs for vertex cover.

1. greedy $O(\log n)$ approx. factor
2. incremental (pick edges) 2 approx factor
3. ILP relaxation & rounding 2 " " even for weighted case

Recall:

Set Cover Problem: Given a collection of sets $S_1, S_2 \cdots S_k$ $\quad S_i \subseteq \{1, \cdots n\}$

find a min. size subcollection s.t. every element $1 \cdots n$ is covered

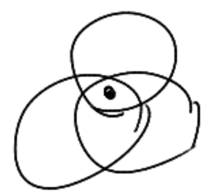
i.e $C \subseteq \{1, \cdots, k\}$

$\forall i \in \{1 \cdots n\}$ $\quad i \in S_j$ for some $j \in C$.

For Set Cover, greedy gives $\leq O(\log n) \cdot OPT$. (last day)

Is there something better? sort of.

Define $f = \max\limits_{\text{element } e} \{\#\text{ sets containing } e\}$



For Vertex Cover:

elements = edges sets  edges incident to v.

What is f for Vertex Cover? 2

Vertex Cover is a special case of Set Cover.

we can turn any vertex cover problem into Set Cover.

FACT: There is a poly. time approx. alg. for Set Cover that gives a soln $\leq f \cdot OPT$

i.e. approx factor f.

which is smaller, $\log n$ or f? Depends on the input.

can be k — all sets contain some element.

Alg. uses Linear Programming & duality.

Later: Set Cover has no constant factor approx (in poly. time) — unless P = NP

Approximating to within constant factor
(what does it mean in general)

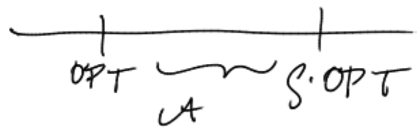
alg. returns $A(I)$ for input I

opt. is $OPT(I)$ - - -

Values of objective function, e.g. size of vertex cover

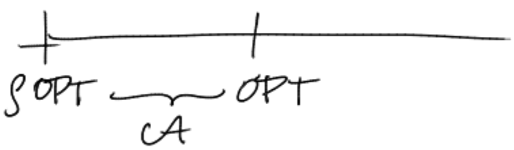
A <u>ρ-approximation</u> alg. for minimization problem guarantees $A(I) \le \rho \cdot OPT(I)$

So $\rho \ge 1$.



A <u>ρ-approx</u> alg. for maximization problem guarantees $A(I) \ge \rho \, OPT(I)$

So $\rho \le 1$



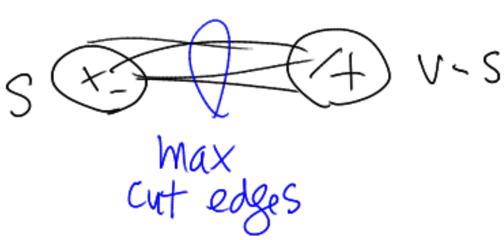
Some texts use $\frac{1}{\rho}$ for max. problems.

CLRS doesn't do max.

Vaziriani uses same convention as here.

Example of maximization problem.

<u>Max Cut</u> Given graph $G = (V, E)$, partition V into $S, V \setminus S$ to maximize #edges with one end in S and one end in $V \setminus S$.



$c(S)$ – size of cut for $S, V \setminus S$.

FACT: [decision version of] Max Cut is NP-complete - but min. cut can be found in poly. time.

Approx. Alg. — local improvement.

$S \leftarrow$ any subset of V
repeat
  for each vertex v
    if moving v to opposite set increases $c(S)$, do it
until no vertex can be moved.

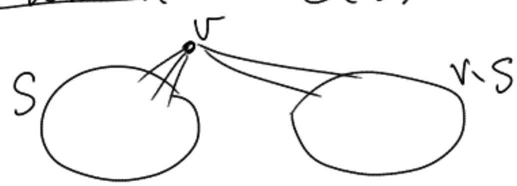
Run Time: we can increase $\underbrace{c(S)}_{\#\text{ edges in cut}}$ at most m times
$m = \#$ edges

So poly. time.

Lemma At end $c(S) \geq \frac{1}{2} m$

So $c(S) \geq \frac{1}{2} OPT$

Pf of lemma $e(S) = \#$ edges inside S



$d_S(v) = \#$ edges from v to S
$d_{V \setminus S}(v) = \text{- - - -} \; v$ to $V \setminus S$

At end of Alg:

$\forall \; v \in S \quad d_S(v) \leq d_{V\setminus S}(v)$ (otherwise we would move it)

$$\underbrace{\sum_{v \in S} d_S(v)}_{= 2e(S)} \leq \underbrace{\sum_{v \in S} d_{V\setminus S}(v)}_{= c(S)}$$

$\forall \; v \in V \setminus S \quad d_{V\setminus S}(v) \leq d_S(v)$

$$\underbrace{\sum_{v \in V\setminus S} d_{V\setminus S}(v)}_{= 2e(V\setminus S)} \leq \underbrace{\sum_{v \in V\setminus S} d_S(v)}_{= c(S)}$$

$m = e(S) + e(V\setminus S) + c(S)$

$2m = 2e(S) + 2e(V\setminus S) + c(S) \leq c(S) + c(S) + 2c(S) = 4c(S)$

$\therefore \; c(S) \geq \frac{1}{2} m$

An alternative randomized approx. alg. for Max Cut.

Pick S at random.

i.e. put each v into S with prob. $\frac{1}{2}$.

Analysis: $E[c(S)] = \frac{1}{2} m \geq \frac{1}{2} OPT$.

Pf. $E[c(S)] = E\left[\sum_{(u,v) \in E} \sigma(u,v)\right]$

$$\sigma(u,v) = \begin{cases} 1 & \text{if } (u,v) \text{ in cut} \quad \text{— prob } \frac{1}{2} \\ 0 & \text{else.} \quad \text{— prob } \frac{1}{2} \end{cases}$$

$$E[c(S)] = \sum_{(u,v) \in E} E[\sigma(u,v)] = \sum_{(u,v) \in E} \frac{1}{2} = \frac{1}{2} m.$$

expected approx factor is $\frac{1}{2}$

Best known approx factor for max Cut is .878

Next day: an approx. alg. using LP relaxation and randomization.