RECAP

- Definition of $\rho(n)$-approximation algorithm A_Π for optimization problem Π:

$$\forall i : |i| = n : \max\left\{ \underbrace{\frac{c(A_\Pi(i))}{c(S^*(i))}}_{\text{MIN}} , \underbrace{\frac{c(S^*(i))}{c(A_\Pi(i))}}_{\text{MAX}} \right\} \le \rho(n)$$

- Definition of PTAS, FPTAS $A_\Pi(i,\varepsilon)$

$$\rho(n) = 1+\varepsilon \ , \quad T_{A_\Pi} = O(\mathrm{poly}(n)) \quad T_{A_\Pi} = O\left(\mathrm{poly}\left(n, \tfrac{1}{\varepsilon}\right)\right)$$

- A maximal matching-based algorithm A-V-C ($\langle G \rangle$) with $\rho = 2$ (tight) to approximate MINIMUM VERTEX COVER in linear time

Can A-V-C(G) be used to obtain an approximation algorithm for other problems?

We know that any (maximum) clique V' in G is associated to a (minimum) vertex cover $V-V'$ in G^c, and vice versa.

Possible algorithm for approximating MAX-CLIQUE:

APPROX_CLIQUE(G)
  $V' \leftarrow$ APPROX_VERTEX_COVER(G^c)
  return $V-V'$

The algorithm correctly returns a clique of G. What about ρ_{AC}?

Assume that $G=(V,E)$ has a maximum clique of size $\frac{|V|}{2}+1$. Thus the minimum vertex cover in G^c has size $|V| - \frac{|V|}{2} - 1 = \frac{|V|}{2} - 1$.

Since $\rho_{A\_V\_C} = 2$, $A\_V\_C(G^c)$ may return a vertex cover V' of size $2\left(\frac{|V|}{2} - 1\right) = |V| - 2$.

Therefore we return a (trivial) clique $V - V'$ of size $|V| - (|V|-2) = 2$!

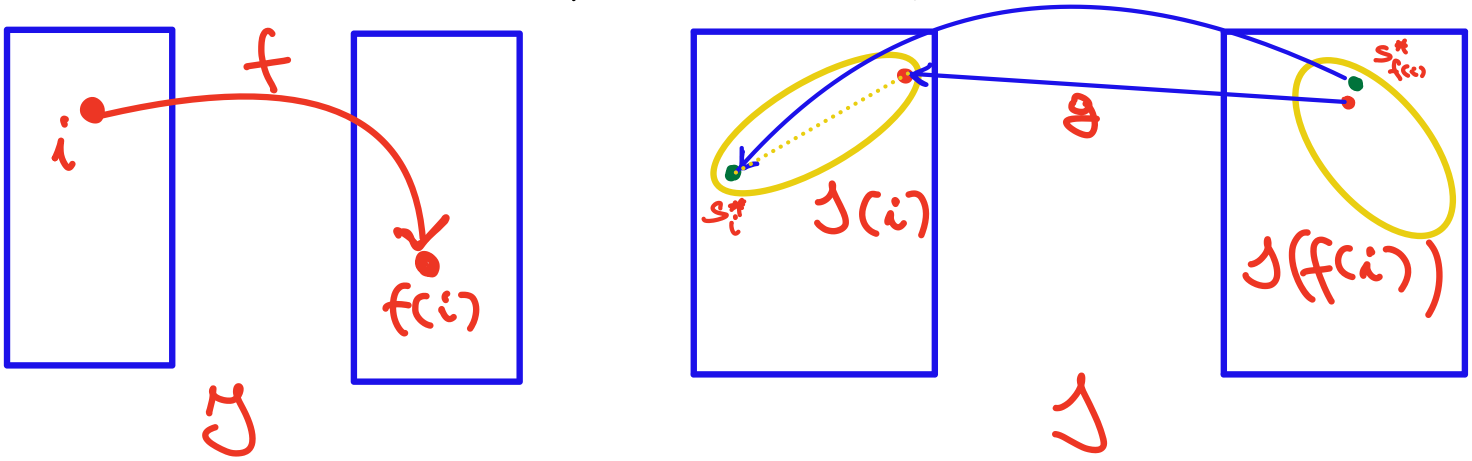
APPROXIMATION RATIO:

$$\boxed{\rho_{AC} \geq \frac{\frac{|V|}{2}+1}{2} \approx \frac{|V|}{4}}$$

Very bad quality algorithm!

This is a clear example that the reduction $f(\langle G\rangle_{\text{CLIQUE}}) = \langle G^c\rangle_{\text{VERTEX COVER}}$ with inverse function $g(V') = V - V'$ does not preserve the approximation!

ottimo $\to$ ottimo, ma quasi-ottimo $\not\to$ quasi-ottimo



HARDNESS OF APPROXIMATING CLIQUE

[Feige et al, 1991] $\forall \varepsilon > 0$, there cannot be a polynomial-time $O(|V|^{1-\varepsilon})$-approximation algorithm for MAXIMUM CLIQUE unless P=NP

MORALE: CLIQUE is essentially inapproximable within any reasonable quality bound.

NOT ALL GREEDY APPROACHES LEAD TO A GOOD APPROXIMATION

Alternative greedy approach:

$V' \leftarrow \emptyset$; $E' \leftarrow E$

repeat

Greedy choice - select arbitrary edge $e = \{u, v\}$

- $V' \leftarrow V' \cup \{u\}$ (only one endpoint)

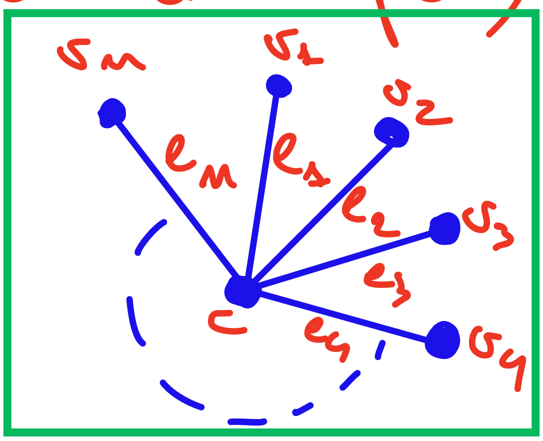
Clean up - $E' \leftarrow E' - \{e \in E' : u \in e\}$

until $(E' = \emptyset)$

Algorithm correctly returns a VC V' (all edges are covered when they are eliminated)

but (counterexample)

"wheel graph"



The algorithm might select $s_1, s_2, \ldots, s_n$! $\Rightarrow$ $\boxed{\rho(|V|) \geq (|V|-1)/1 = |V|-1}$

- "improved" strategy: select vertex with larger degree (natural!)

It can be shown that

$$\boxed{\rho(|V|) = \Omega(\log |V|)}$$

EXTENSION: **Weighted Vertex Cover**

DEFINITION Given a node-weighted undirected graph:

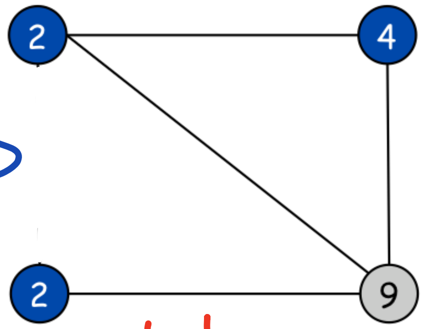
$$\langle G=(V,E), w: V \to \mathbb{N}^+ \rangle$$

(each node is associated to a positive weight), determine a vertex cover $V' \subseteq V$ minimizing

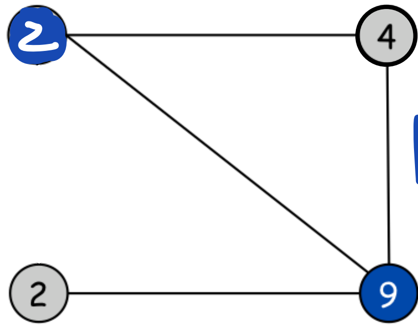
$$\boxed{C(V') = \sum_{u \in V'} w(u)}.$$

OBSERVATION Letting $w(u)=1 \ \forall u \in V$ we obtain MINIMUM VERTEX COVER

$|V'| = 3$



weight: 8



$|V'| = 2$

weight: 11

We will illustrate two different approximation algorithms for MIN-WEIGHT VERTEX-COVER

- The first algorithm is a nontrivial extension of the maximal matching algorithm

- The second algorithm introduces an important technique based on linear programming

Fair Pricing Method

As in A-V-C we proceed by covering edges. In the weighted variant, each selected $e=\{u,v\}$ pays a price to get covered. The price is a portion of the weights $w(u)$ and $w(v)$ and has to be positive:

- the price payed is

$$\boxed{p_e = \min\{w(u), w(v)\} > 0}$$

- the weights $w(u), w(v)$ are decremented:

$$\boxed{\begin{aligned} w'(u) &= w(u) - p_e \\ w'(v) &= w(v) - p_e \end{aligned}}$$

- After the selection, one of the weights becomes 0 and the corresponding node becomes tight ($\rightarrow$ will go into V')
- We proceed until there are no edges between nontight nodes
- V' = set of tight nodes

FAIR_PRICING $(\langle G=(V,E), w \rangle)$

for each $e \in E$ do $p_e = 0$

for each $v \in V$ do $w'(v) = w(v)$

while $(\exists\, e=\{u,v\} : (w'(u)>0) \wedge (w'(v)>0))$

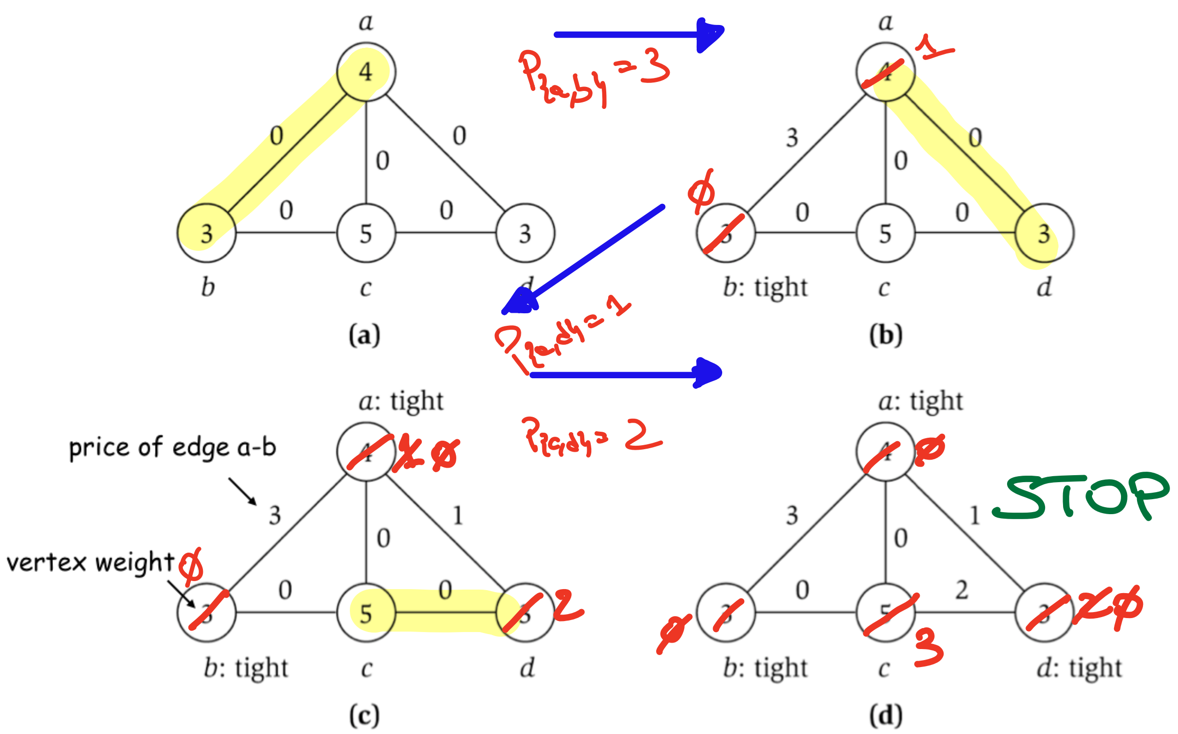
do $p_e \leftarrow \min\{w'(u), w'(v)\}$

$w'(u) \leftarrow w'(u) - p_e$

$w'(v) \leftarrow w'(v) - p_e$

{one among u and v becomes tight}

return $V' = \{v \in V : w'(v) = 0\}$



Running Time: can be implemented in linear time (single scan of adjacency list - as A-V-C)

Correctness: The set V' returned by FP is a vertex cover:

PROOF: all edges must be adjacent to a tight node, or otherwise

$$\boxed{\exists\, e = \{u,v\} \in E : (w'(u) > 0) \wedge (w'(v) > 0)}$$

and FP could not have finished.

OBSERVATION: For $\boxed{w(v) = 1\ \forall v \in V}$, we have

$$\boxed{FP \approx A\text{-}V\text{-}C}$$

(each selected edge makes both of its endpoints tight)

ANALYSIS OF APPROXIMATION RATIO

FAIR PROPERTY: At the end of FP:

$$\boxed{\forall u \in V : \sum_{e \ni u} p_e \le w(u)}$$

PROOF $w'(u) \ge p_e > 0$ is decremented by p_e after each edge $e = \{u,v\}$ is selected and no edge $\{u,v'\}$ can be selected if $\boxed{w'(u) = 0}$

COROLLARY: $\boxed{u \text{ tight} \Rightarrow \sum_{e \ni u} p_e = w(u)}$

LEMMA: For any vertex cover $\bar{V}$:

$$\boxed{\sum_{e \in E'} p_e \leq \sum_{v \in \bar{V}} w(v)} \qquad (|V'| \geq |A|)$$

PROOF

$$\boxed{\sum_{e \in E} p_e \leq \sum_{u \in \bar{V}} \sum_{e \ni u} p_e}$$

(since $\bar{V}$ is a VC, each p_e appears at least once in the right-hand-side summations)

$$\boxed{\leq \sum_{u \in \bar{V}} w(u)}$$

(by the fair property)

This is a crucial lemma that provides the lower bound on $c(V^*)$, since it must also hold for $\bar{V} = V^*$ (min-weight vertex cover)

$$\boxed{\sum_{e \in E} p_e \leq \sum_{v \in V^*} w(v) = c(V^*)} \qquad (1)$$

We are ready to prove a bound on

$$\boxed{\rho = \frac{c(V')}{c(V^*)}}$$

← V' returned by algorithm

$$\boxed{\begin{aligned} c(V') &= \sum_{u \in V'} w(u) \\ &= \sum_{u \in V'} \sum_{e \ni u} p_e \end{aligned}}$$

all nodes in V' are tight (corollary)

$$\leq \sum_{u \in V} \sum_{e \ni u} p_e$$

since $V' \subseteq V$ and $p_e \geq 0$

$$= 2 \sum_{e \in E} p_e$$

each p_e counted twice ($e = \{u, v\}$, once w.r.t. u and once w.r.t. v)

$$\leq 2\, c(V^*)$$

by (1)

Thus $\boxed{\rho = \dfrac{c(V')}{c(V^*)} \leq \dfrac{2\,c(V^*)}{c(V^*)} \leq 2}$

OBSERVATION: When $w(u) = 1 \ \forall u \in V$ (unweighted case) the analysis becomes the one used for A-V-C:

1. The fair property implies that the selected edges are a matching A

2. The lemma says that any VC must have size $\geq$ any matching A

(VERIFY AS HOMEWORK)

Integer linear programming formulation of weighted vertex cover

Given $\langle G=(V,E), w \rangle$, we use:

1. $|V|$ 0/1 variables $x_u : u \in V$
   The vertex cover is $\{u \in V : x_u = 1\}$
2. $|E|$ covering constraints:
   $\forall e = \{u,v\} : x_u + x_v \geq 1 \; (x_u, x_v \in \{0,1\})$
3. Objective function:

$$\min \sum_{u \in V} w(u) x_u$$

INTEGER LINEAR PROGRAM (ILP)

$$\begin{cases} \min \sum_{u \in V} w(u) x_u \\ \text{s.t.} \\ x_u + x_v \geq 1 : \{u,v\} \in E \\ x_u \in \{0,1\} \quad u \in V \end{cases}$$

The optimal integer solution is a minimum-weight vertex cover

(PROVE IT: each vertex cover is feasible and each feasible solution is a vertex cover)

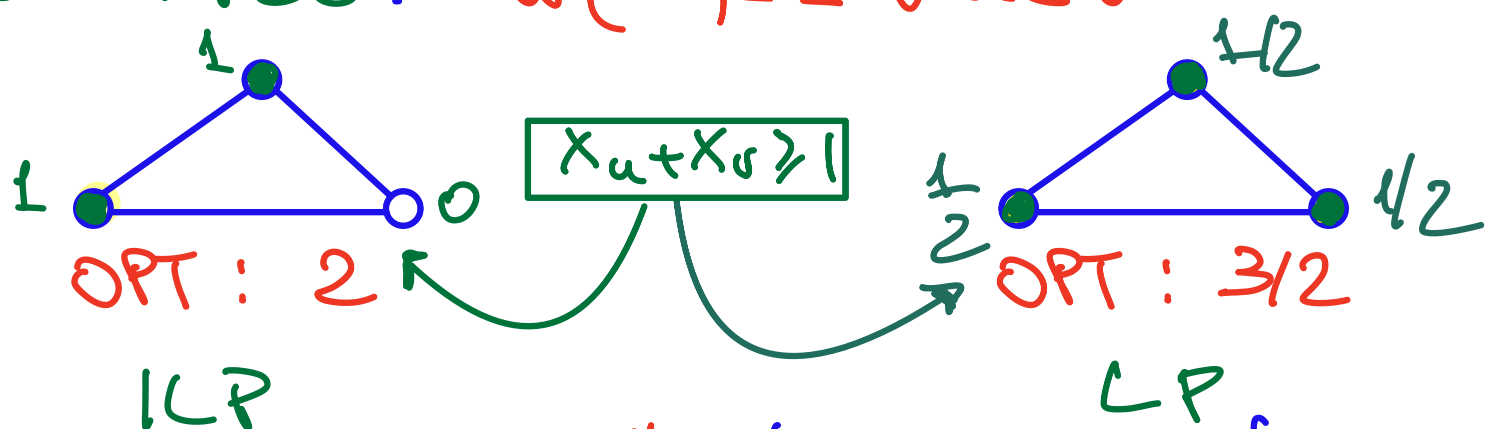
Let us consider the continuous relaxation of ILP

LINEAR PROGRAM (LP)

$$\begin{cases} \min \sum_{u \in V} w(u) x_u \\ \text{s.t.} \\ x_u + x_v \geq 1 : \{u,v\} \in E \\ 0 \leq x_u \leq 1 \qquad u \in V \end{cases}$$

Clearly, the optimal solution x^* to LP is not necessarily integer and its cost is $\leq$ cost of the optimal integer solution

EXAMPLE: $w(u) = 1 \ \forall u \in V$



We can use x^* of LP to find a "good" solution $\hat{x}$ to ILP via ROUNDING!

Also, $c(x^*)$ lower bound c_{OPT}!

Theorem If x^* is an optimal solution of the LP relaxation of the vertex-cover ILP, then

$$\boxed{V' = \{u \in V : x^*_u \geq \tfrac{1}{2}\}}$$

is a vertex cover of weight at most twice the minimum weight.

Proof. Let us prove that V' is a VC:

$\forall e = \{u, v\} \in E$, we have

$x^*_u, x^*_v \geq 0$

$$\boxed{x^*_u + x^*_v \geq 1}, \text{ therefore :}$$

$$\boxed{\left(x^*_u \geq \tfrac{1}{2}\right) \vee \left(x^*_v \geq \tfrac{1}{2}\right)}$$

$$\Rightarrow \boxed{(u \in V') \vee (v \in V')}$$

Each edge is covered by V'!

Let's prove the bound on the weight:

$$\boxed{\sum_{u \in V^*} w(u) \overset{\text{LB!}}{\geq} \sum_{u \in V} w(u) x^*_u}$$

V^* feasible for ILP

optimal cost of LP

$$\boxed{\geq \sum_{u \in V'} w(u)\, x_u^*}$$

(eliminate terms with $x_u^* < 1/2$)

$$\boxed{\geq \frac{1}{2} \sum_{u \in V'} w(u)} \qquad (u \in V' : x_u^* \geq \tfrac{1}{2})$$

Therefore:

$$\boxed{\rho = \frac{\sum_{u \in V'} w(u)}{\sum_{u \in V^*} w(u)} \leq 2}$$

LESSON LEARNED LP rounding is a powerful technique to obtain quality-controlled approximations to hard problems that admit a "good" ILP representation.

- Randomized rounding: even more powerful
- CONS: Cost of solving LP