Approximation Algorithms

Def: An approximation algorithm for an optimization problem is a polynomial-time algorithm that always returns a solution within a factor α of the optimal solution.

For a minimization problem this means

$$OPT \le ALG \le \alpha\, OPT$$

where OPT is the optimal solution to the problem, ALG is the value of the solution returned by the algorithm, $\alpha \geq 1$.

For a maximization problem

$$\alpha\, OPT \le ALG \le OPT \qquad \alpha \le 1$$

Example: The 2-approx algorithm for Min Vertex Cover from last class.

- Find a maximal matching $M \subseteq E$
- Let $C = \{v \in V : v \in e \text{ for some } e \in M\}$
- Return C

This can be implemented in $\underline{O(E)}$

Vertex Cover II

Recall $\{y_e\}$ $\{x_v\}$

$$\begin{array}{l} \max \sum_{e \in E} y_e \\ \text{s.t. } \forall v \in V \\ \quad \sum_{e : v \in e} y_e \le 1 \\ \quad y_e \in \{0,1\} \end{array} \le \begin{array}{l} \max \sum_{e \in E} y_e \\ \text{s.t. } \forall v \in V \\ \quad \sum_{e : v \in e} y_e \le 1 \\ \quad y_e \ge 0 \end{array} = \begin{array}{l} \min \sum_{v \in V} x_v \\ \text{s.t. } x_u + x_v \ge 1 \\ \quad \text{for } (u,v) \in E \\ \quad x_v \ge 0 \quad \forall v \in V \end{array} \le \begin{array}{l} \min \sum_{v \in V} x_v \\ \text{s.t. } x_u + x_v \ge 1 \\ \quad \text{for } (u,v) \in E \\ \quad x_v \in \{0,1\} \quad \forall v \in V \end{array}$$

(4) maximum matching — polynomial time solvable

(3) fractional matching

(2) LP relax for Min-VC

(1) Min-VC — NP-hard

Every vertex cover $C \subseteq V$ can be encoded as a feasible solution to (1) and every matching $M \subseteq E$ can be encoded as a feasible solution to (4)

Exercise: show me how. I.e.

define

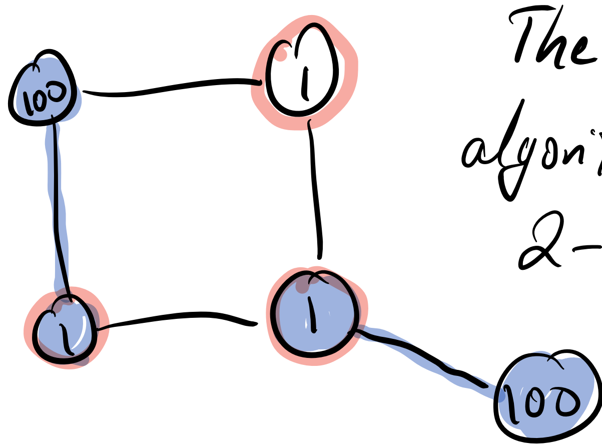
$$y_e = \begin{cases} 1 & \text{if } e \in M \\ 0 & \text{otherwise} \end{cases} \quad \text{for } M \qquad x_v = \begin{cases} 1 & \text{if } v \in C \\ 0 & \text{otherwise} \end{cases} \quad \text{for } C$$

Node-Weighted Vertex Cover

Let $G=(V,E)$ be an undirected graph with a node weight for each node: for $v \in V$ we have $w_v \geq 0$.

The weighted Min-VC problem seeks a vertex cover $C \subseteq V$ that minimizes

$$\sum_{v \in C} w_v.$$



The unweighted algorithm is not a 2-approximation.

A 2-approx for weighted Min-VC

Alg:

1. Solve the LP relaxation

$$\min \sum_{v \in V} x_v \cdot w_v$$
$$x_v + x_u \geq 1 \quad \forall \ (u,v) \in E$$
$$x_v \geq 0 \quad \forall \ v \in V$$

Let $\{\tilde{x}_v\}_{v \in V}$ be the optimal LP solution.

2. Return $\hat{C} = \{v \in V : \tilde{x}_v \geq \frac{1}{2}\}$.

Theorem: This is a 2-approximation for weighted Min-VC.

Proof: For each $v \in V$ define

$$\hat{x}_v = \begin{cases} 1 & \text{if } v \in \hat{C} \quad (\tilde{x}_v \geq \tfrac{1}{2}) \\ 0 & \text{otherwise} \end{cases}$$

and

$$x^*_v = \begin{cases} 1 & \text{if } v \in C^* \\ 0 & \text{otherwise} \end{cases}$$

where C^* is an optimal vertex cover.

Then

$$\underline{ALG} = \sum_{v \in \hat{C}} w_v = \sum_{v \in \underline{V}} \hat{x}_v \cdot w_v \geq \overbrace{\sum_{v \in V} x^*_v w_v}^{OPT}$$

$$\geq \sum_{v \in V} \boxed{\tilde{x}_v} w_v \quad (\text{why?})$$

$$\geq \sum_{v \in V} \boxed{\frac{\hat{x}_v}{2}} w_v = \frac{1}{2} ALG$$

Check: if $\underline{v \in \hat{C}}$, $\hat{x}_v = 1$ $\tilde{x}_v \geq \frac{1}{2}$

so $\tilde{x}_v \geq \frac{1}{2} \cdot 1 \geq \frac{1}{2} \cdot \hat{x}_v$

if $v \notin \hat{C}$ ... same idea.

$$ALG \geq OPT \geq \frac{ALG}{2}$$

$$\Rightarrow 2OPT \geq ALG.$$

For unweighted Min-VC, we had an $O(E)$ algorithm.

Solving an LP has a much poorer runtime

linear

We'd love an $O(E)$ algorithm for the weighted version.

$\boxed{\text{ALG: Linear-time Weighted VC}}$

- Set $p_v = 0 \quad \forall v \in V$
  $r_v = w_v \quad \forall v \in V$

invariant: $w_v = p_v + r_v$

$$\boxed{r_v = w_v - p_v}$$

$$y_{uv} = 0 \quad \forall (u,v) \in E$$

- For each $(u,v) \in E$
  - if $p_v < w_v$ and $p_u < w_u$
  - * $y_{uv} = \min\{w_v - p_v, w_u - p_u\} = \min\{r_v, r_u\}$
  - * $p_u \leftarrow p_u + y_{uv}$ $\qquad r_u \leftarrow r_u - y_{uv}$
  - * $p_v \leftarrow p_v + y_{uv}$ $\qquad r_v \leftarrow r_v - y_{uv}$

end

- Return $C = \{v : p_v = w_v\}$
  $= \{v : r_v = 0\}$

Assuming each node has at least 1 edge, $O(V+E) = O(E)$.

Thm: This is an $O(E)$-time 2-approximation for weighted min-VC.

Proof: This is a vertex cover because at every iteration I drive one of the residuals to zero.

Step 1: prove a bound on the cost of the algorithm.

Observe that $r_v \geq 0$, i.e. $p_v \leq w_v$

so

$$ALG = \sum_{v \in C} w_v = \sum_{v \in C} p_v \leq \sum_{v \in V} p_v$$

Consider the MinVC LP relaxation and its dual

$$\min \sum x_v \boxed{w_v}$$
$$\text{s.t.} \quad x_u + x_v \geq 1$$
$$x_u \geq 0$$

(dual)

$$\max \sum_{(u,v) \in E} y_{uv} \cdot 1$$
$$\text{s.t.} \quad \forall v \in V$$
$$\boxed{\sum_{u:(u,v) \in E} y_{uv} \leq \boxed{w_v}}$$
$$y_{uv} \geq 0$$

We will show the $\{y_{uv}\}$ variables from the algorithm are feasible for the dual.

Claim 1: for each $v \in V$

$$\boxed{\sum_{u:(u,v)\in E} y_{uv}} = p_v \qquad \text{Why?}$$

Because p_v starts at zero, and for each $u \in N(v)$ we increase p_v by y_{uv}.

Also $p_v \le w_v$ so for every $v \in V$

$$\boxed{\sum_{u:(u,v)\in E} y_{uv} \le w_v.}$$

So $\{y_{uv}\}_{uv \in E}$ is a feasible solution for the dual, hence

$$\boxed{\sum_{(u,v)\in E} y_{uv} \le OPT}$$

(weak duality)

Claim 2: $\sum_{v\in V} p_v = 2 \sum_{(u,v)\in E} y_{uv}$

Why? Each time we visit an edge $(u,v)\in E$ and set y_{uv}, we increase the "p" value of two nodes.

All together:

$$OPT \le \underline{ALG} \le \sum_{v\in V} p_v = 2\sum_{uv\in E} y_{uv} \le \underline{2 OPT}$$

We never had to solve the LP relaxation or its dual.