Approximation Algorithms

For a minimization problem, an $\alpha$-approximation algorithm computes a number, $\text{ALG}$, such that

$$\forall \text{input } x, \quad \text{OPT}(x) \leq \text{ALG}(x) \leq \alpha \cdot \text{OPT}(x)$$

Duh! Ahah!

In this type of guarantee, $\alpha$ is often a constant, but could be a function of $n=|x|$ or even of $x$ itself.

Duh! Ahah!

For maximization, \[ \text{ALG}(x) \leq \text{OPT}(x) \leq \alpha \cdot \text{ALG}(x) \quad (\alpha \geq 1) \]

Or sometimes, \[ \alpha \cdot \text{OPT}(x) \leq \text{ALG}(x) \leq \text{OPT}(x) \quad (\alpha \leq 1) \]

Def: A vertex cover of an undirected graph $G$ is (a) a set of vertices, $S$, such that every edge has an endpoint in $S$ or (b) the complement of an independent set.

\[ \text{Min}\text{-}\text{comparability vertex cover} \]

Theorem (König–Egerváry) If $G$ is bipartite,

$$\min \{|S|: S \text{ a vertex cover}\} = \max \{|M|: M \text{ a matching}\}$$

Proof sketch

Apply max-flow min-cut.
Greedy 2-approx alg:

\[ S = \emptyset, \quad \text{mark all edges uncovered,} \quad M = \emptyset \]

while \( E \) on uncovered edge \( e=(u,v) \):

\[ S \leftarrow S \cup \{u,v\}; \quad M \leftarrow M \cup e? \]

mark all edges incident to \( u \) or \( v \) as covered

endwhile

output \( S \)

1. \( S \) is a valid vertex cover.
   (Every edge \( e \prime \) got covered in the iteration where we marked it as covered)

2. \( |S| \leq 2 \cdot |\text{OPT}| \).
   By construction \( |S| = 2|M| \).
   By pigeonhole, \( |M| \leq |\text{OPT}| \)
   because if \( S^* \) is an \( \text{OPT} \) vertex cover,
   there is a 1-to-1 mapping \( \text{OPT} \rightarrow S^* \)
   that sends each \( e \in M \) to an endpoint of \( e \).

Randomized 2-approx alg for VC

\[ S = \emptyset, \quad \text{mark all edges uncovered} \]

while \( E \) edge \( e=(u,v) \) which is uncovered:

    toss a fair coin and:
    heads \( \rightarrow \) \( S = S \cup \{u\} \)
    tails \( \rightarrow \) \( S = S \cup \{v\} \)

mark all edges incident to the new vertex as covered

endwhile

output \( S \)
Claim. \( E|S| \leq 2 \cdot \alpha \cdot t \)

**Proof.** Let \( S^* \) = any minimum vertex cover

\[
X_t = |\{ u \in S | u \in S^* \} | \quad \text{after } t \text{ iterations}
\]
\[
Y_t = |\{ u \in S \setminus S^* \} | \quad \text{after } t \text{ iterations}
\]

\[X_t + Y_t = |S| = t \quad \forall t.\]

\[
E[X_{t+1} - X_t] = \begin{cases} 1 & \text{if } \{u,v\} \in S^* \\ 1/2 & \text{if } \{u,v\} \notin S^* \end{cases}
\]

\[
E[Y_{t+1} - Y_t] = \begin{cases} 0 & \text{if } \{u,v\} \in S^* \\ 1/2 & \text{if } \{u,v\} \notin S^* \end{cases}
\]

At termination \( E|S| = E|S \cap S^*| + E|S \setminus S^*| \)

\[
\leq 2 \cdot E|S \cap S^*| \leq 2|S^*|.
\]

**Weighted Vertex Cover** vertices have weights \( w(v) \geq 0 \).

Minimize \( \sum_{v \in S} w(v) \), subject to \( S \) being a vertex cover.
1. Reformulate as an integer program.
   Introduce "decision variables" $x_v$ such that $x_v = 1$ indicates $v \in S$
   $x_v = 0$ indicates $v \notin S$.
   
   \[
   \begin{align*}
   \min & \quad \sum_v w(v) x_v \\
   \text{st.} & \quad x_u + x_v \geq 1 \quad \forall \ e=(uv) \in E \\
   & \quad x_v \in \{0,1\} \quad \forall v
   \end{align*}
   \]

2. Relax to a linear program
   
   \[
   \begin{align*}
   \min & \quad \sum_v w(v) x_v \\
   \text{st.} & \quad x_u + x_v \geq 1 \quad \forall \ e=(uv) \in E \\
   & \quad x_v \geq 0 \quad \forall v
   \end{align*}
   \]

3. Solve the linear program.
   ("Ellipsoid algorithm" does this in poly time.)