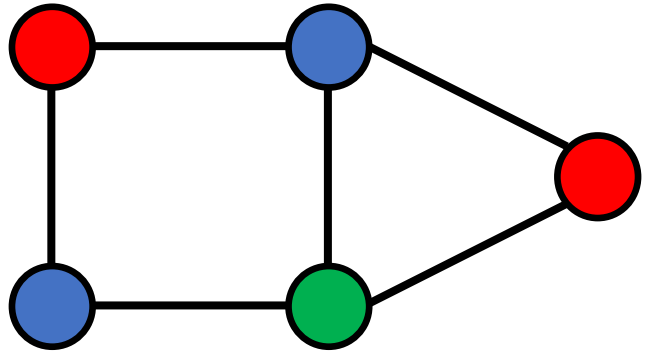
Randomized Algorithms

CSCI 532

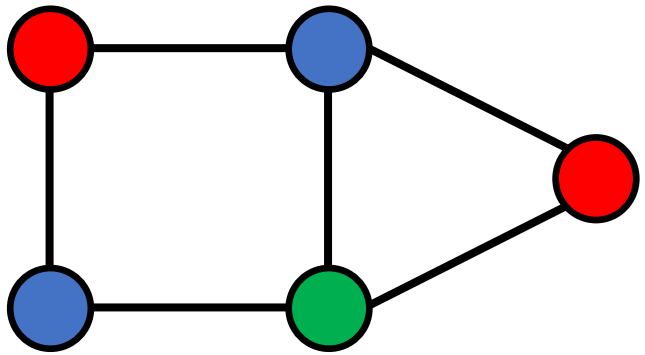
Graph Coloring

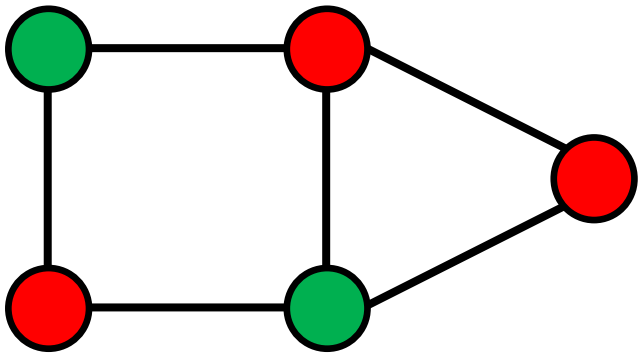
Family of graph problems that involve coloring vertices (or edges) subject to various constraints.



Graph Coloring

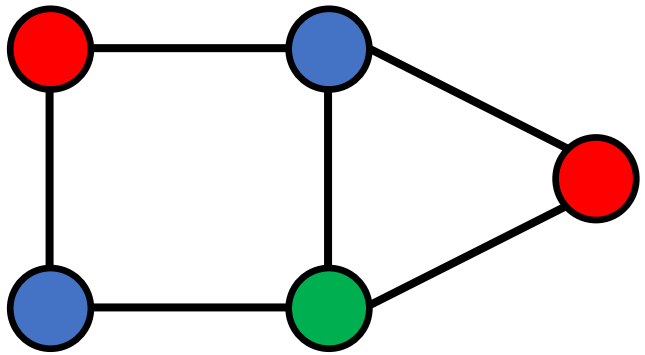
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.

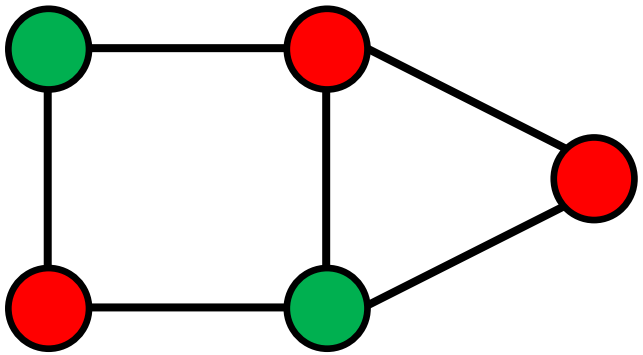




Graph Coloring

Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.

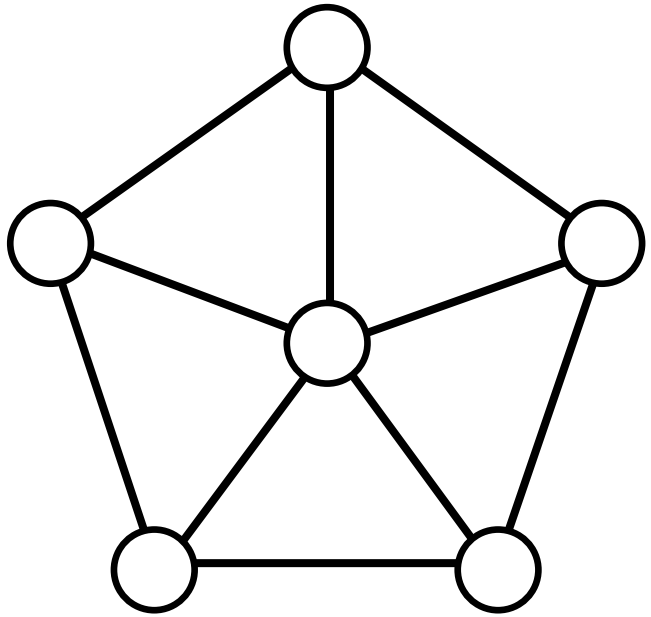




NP-Hard

Graph Coloring

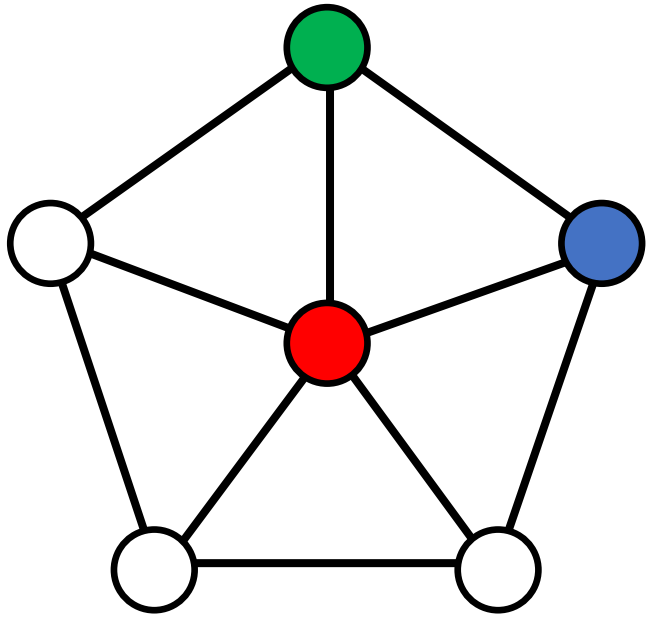
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

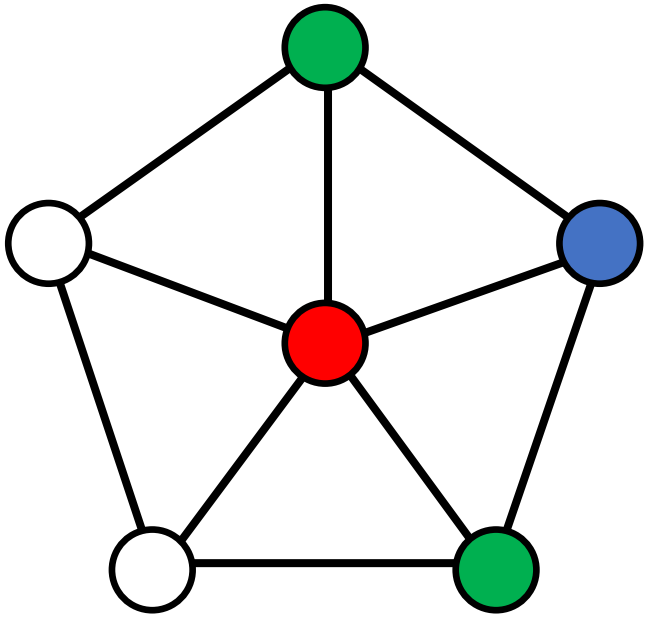
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

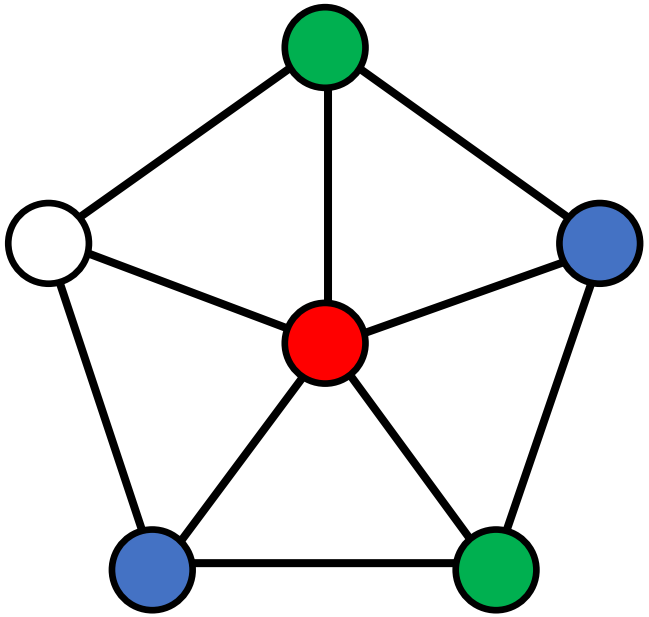
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

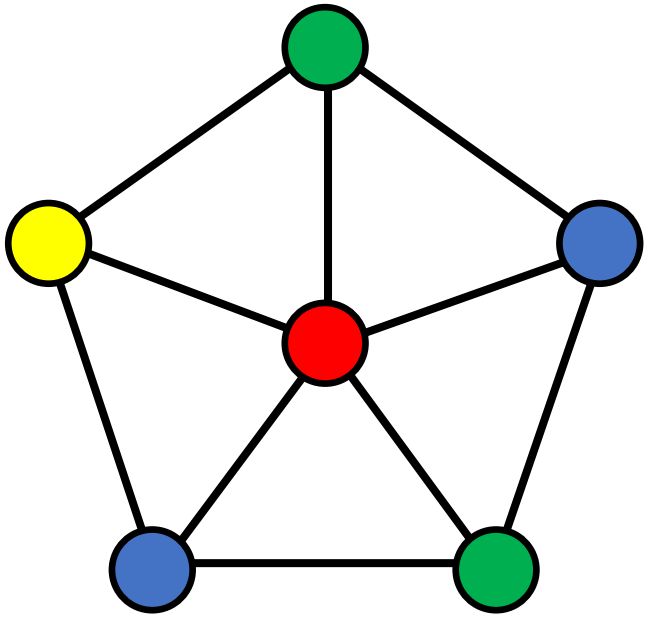
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

Graph Coloring

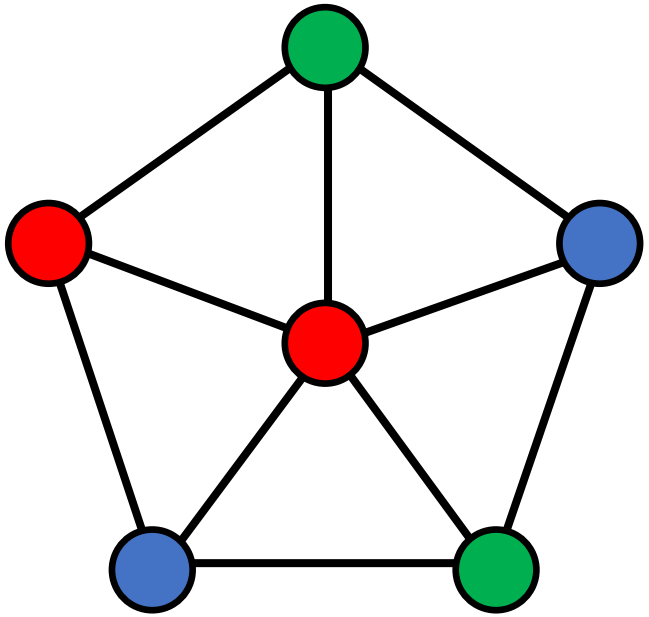
Minimum Vertex Coloring: Color the vertices of a graph using the smallest number of colors so adjacent vertices are different colors.



How many colors are needed to color this graph?

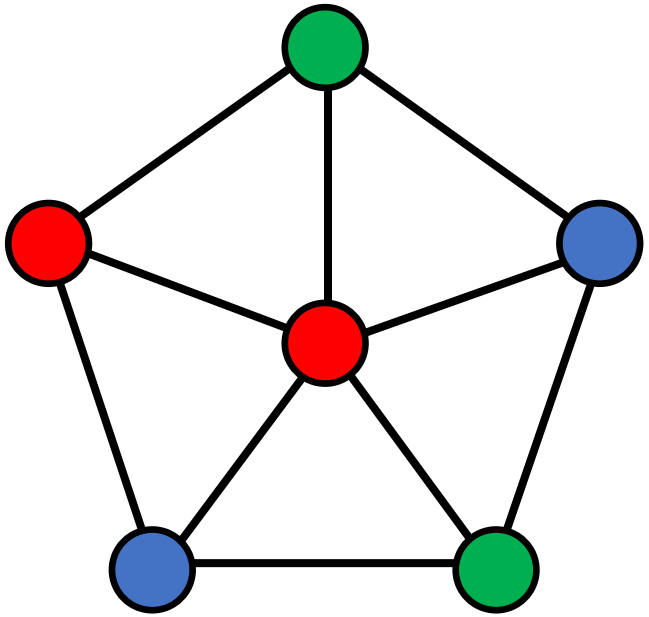
Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Max 3-Coloring

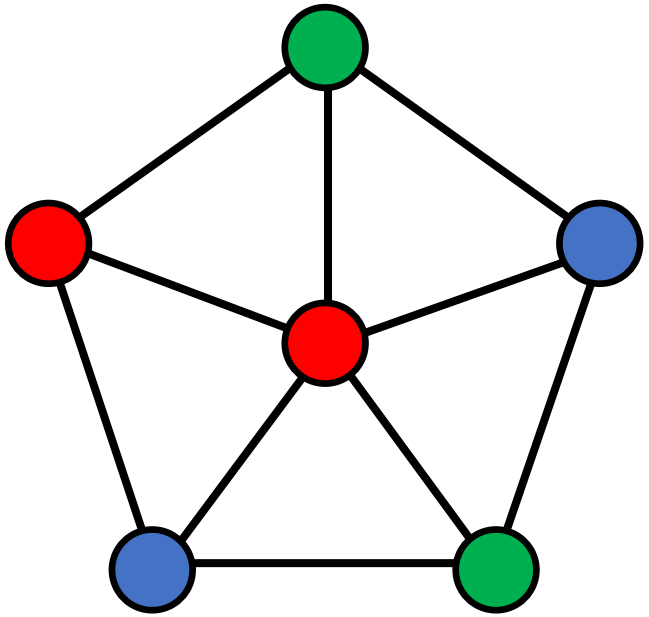
An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm?

Max 3-Coloring

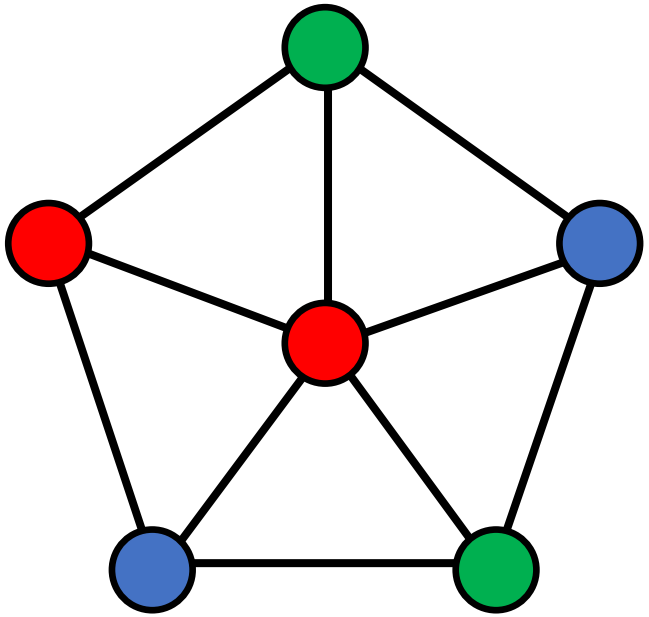
An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.

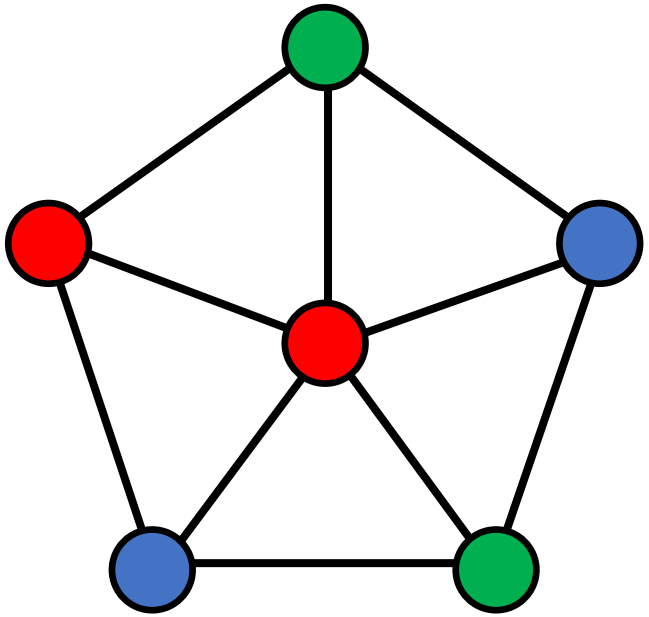


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time?

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.

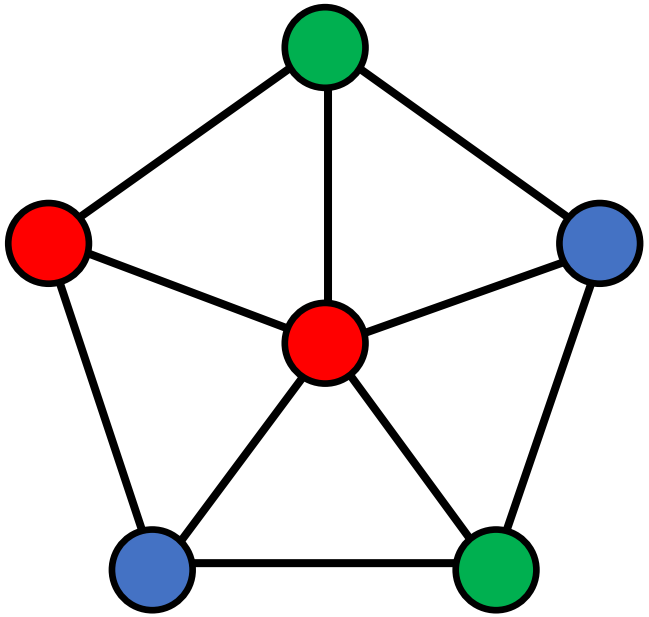


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time? $O(n)$

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



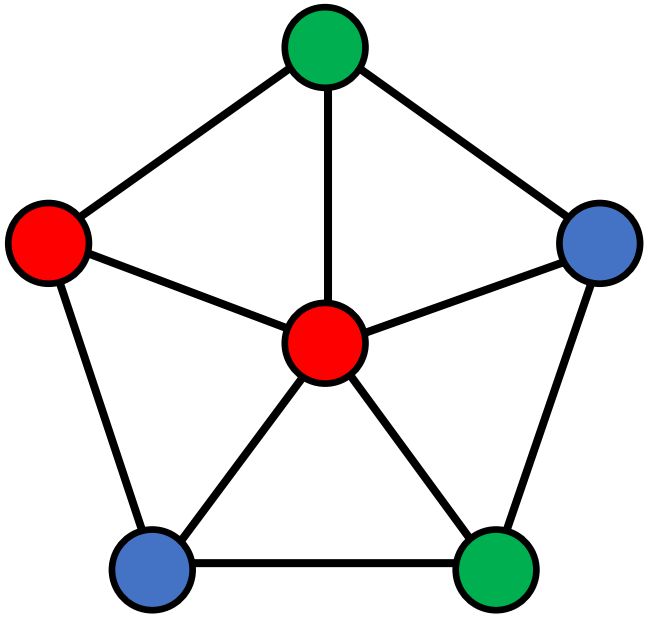
Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time? $O(n)$

Valid?

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



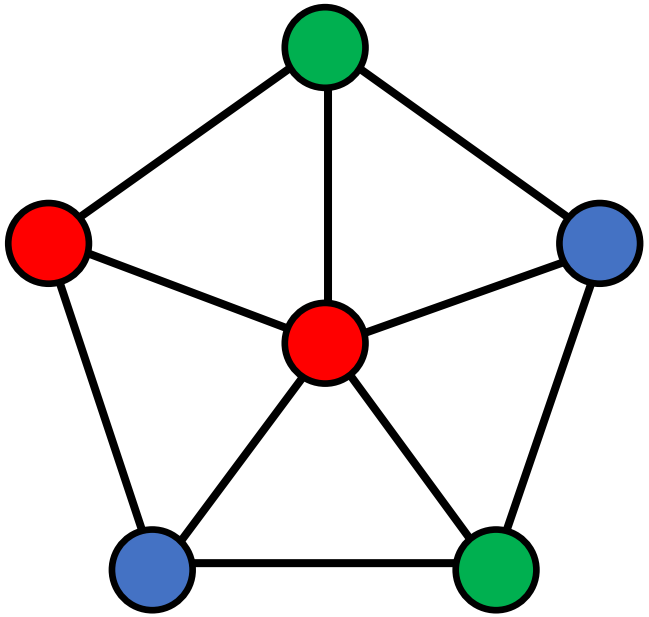
Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time? $O(n)$

Valid? Yes. Colors all vertices with ≤ 3 colors.

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

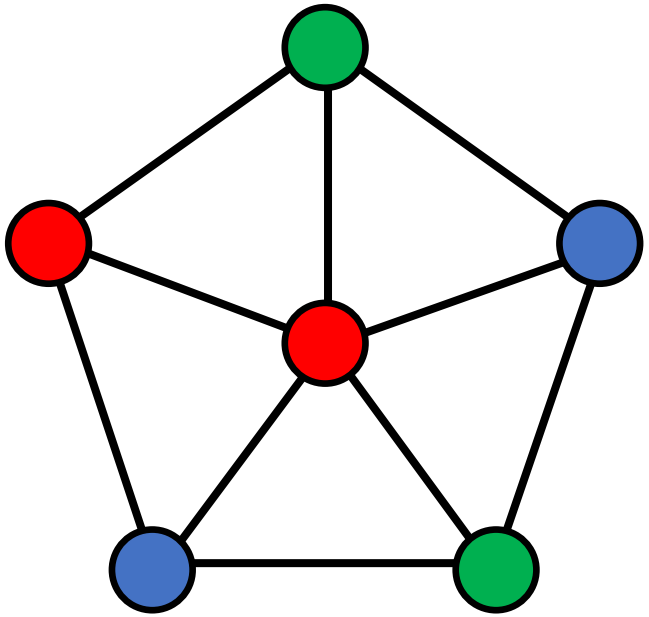
Running time? $O(n)$

Valid? Yes. Colors all vertices with ≤ 3 colors.

Optimal?

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.

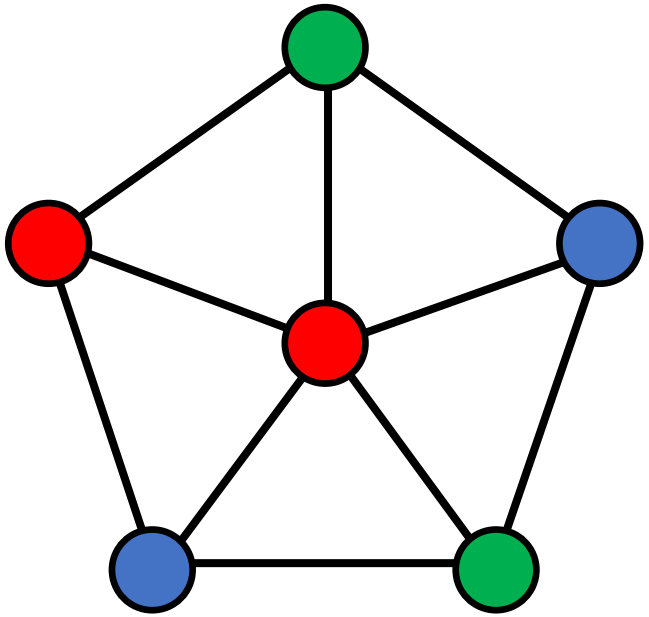


Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time? $O(n)$
Valid? Yes. Colors all vertices with ≤ 3 colors.
Optimal? Probably not…

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

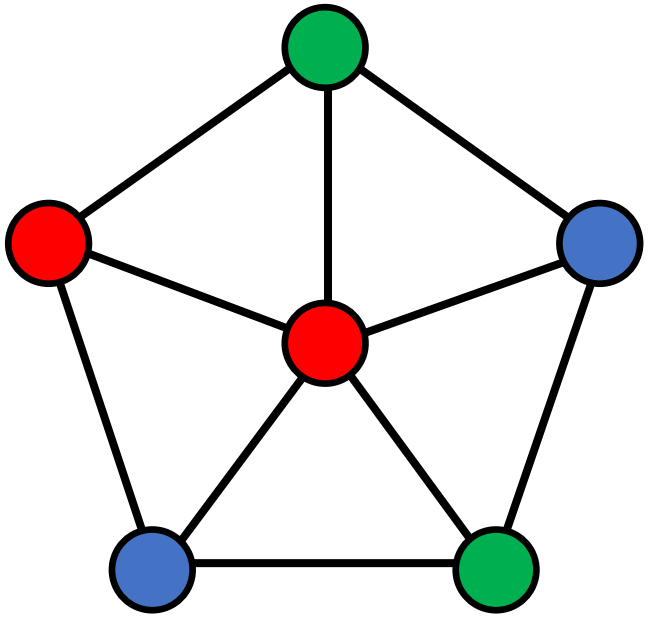
Running time? $O(n)$
Valid? Yes. Colors all vertices with ≤ 3 colors.
Optimal? Probably not…

Goal: Show this algorithm is expected to perform as a $\frac{2}{3}$ - approximation algorithm.

Max 3-Coloring

Note: if problem is a maximization problem, $ALG \geq \frac{1}{\alpha} OPT$

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is **maximized**.



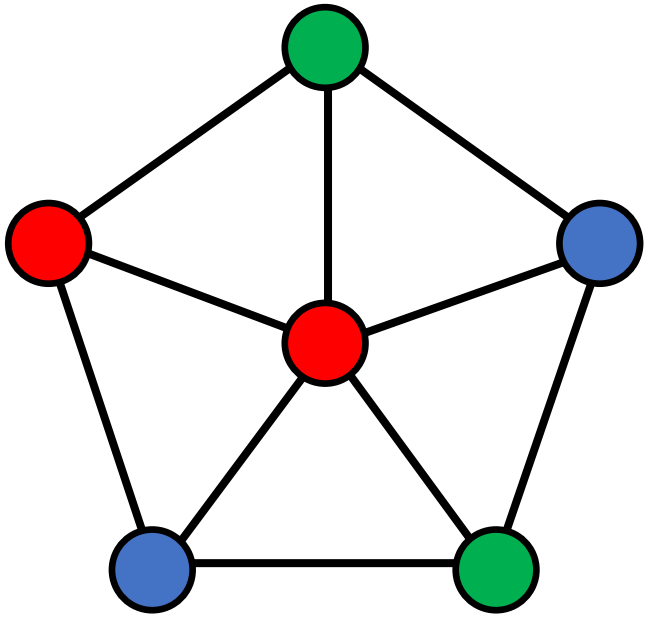
Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time? $O(n)$
Valid? Yes. Colors all vertices with ≤ 3 colors.
Optimal? Probably not…

Goal: Show this algorithm is expected to perform as a $\frac{2}{3}$ - approximation algorithm.

Max 3-Coloring

An edge in a graph is satisfied if its vertices are colored different colors. Using only three colors, color the vertices of a graph such that the number of satisfied edges is maximized.



Algorithm: For each vertex, randomly assign it one of the three colors with probability 1/3.

Running time? $O(n)$
Valid? Yes. Colors all vertices with ≤ 3 colors.
Optimal? Probably not…

Goal: Show this algorithm is **expected to perform** as a $\frac{2}{3}$ - approximation algorithm.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

$$\text{Probability of some outcome} = \frac{\text{\# events we care about}}{\text{\# all possible events}}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

All possibilities:

(b,b), (b,g), (b,r), (g,b), (g,g), (g,r), (r,b), (r,g), (r,r)

$$\text{Probability of some outcome} = \frac{\#\text{ events we care about}}{\#\text{ all possible events}} = \frac{?}{\mathbf{9}}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

All possibilities:

$$\text{Probability of some outcome} = \frac{\#\text{ events we care about}}{\#\text{ all possible events}} = \frac{\mathbf{6}}{\mathbf{9}}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

What is the probability that a single edge is satisfied?

All possibilities:

$$\text{Probability of some outcome} = \frac{\text{\# events we care about}}{\text{\# all possible events}} = \frac{6}{9} = \frac{2}{3}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

What is $\sum_e X_e$?

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

What is $\sum_e X_e$?

ALG – It's the number of satisfied edges.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

What is $\sum_e X_e$?

ALG – It's the number of satisfied edges.

What is $E[\sum_e X_e]$?

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

What is $\sum_e X_e$?

ALG – It's the number of satisfied edges.

What is $E[\sum_e X_e]$?

The most likely value for ALG.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$E[X_e] =$?

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

The expected value of a random variable ($E[Y]$) is the sum of every possible outcome (y_k) times the probability of that outcome (p_k):

$$E[Y] = \sum_k y_k p_k$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right)$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

The expected value of a random variable ($E[Y]$) is the sum of every possible outcome (y_k) times the probability of that outcome (p_k):

$$E[Y] = \sum_k y_k p_k$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

The expected value of a random variable ($E[Y]$) is the sum of every possible outcome (y_k) times the probability of that outcome (p_k):

$$E[Y] = \sum_k y_k p_k$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

$= \sum_e \frac{2}{3}$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

$$\Rightarrow E[ALG] = \frac{2}{3}|E|$$

Max 3-Coloring

Note: if problem is a maximization problem, $ALG \geq \frac{1}{\alpha} OPT$

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq$$ **OPT?**

Max 3-Coloring

Note: if problem is a maximization problem, $ALG \geq \frac{1}{\alpha} OPT$

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

How many edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e]$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Since $OPT \leq |E|$ (can't have more satisfied edges than there are edges!)

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

ϕ is a formula with clauses composed of Boolean variables connected by ORs, and clauses connected by ANDs.

Can you set the variables to **true** or **false** so that ϕ evaluates to **true**?

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

$x_1 = false$
$x_2 = true$

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

$$(F \vee F \vee T) \qquad (T \vee F \vee F) \qquad (T \vee T \vee T)$$

$x_1 = false$

$x_2 = true$

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$(F \vee F \vee T) \qquad (T \vee F \vee F) \qquad (T \vee T \vee T)$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

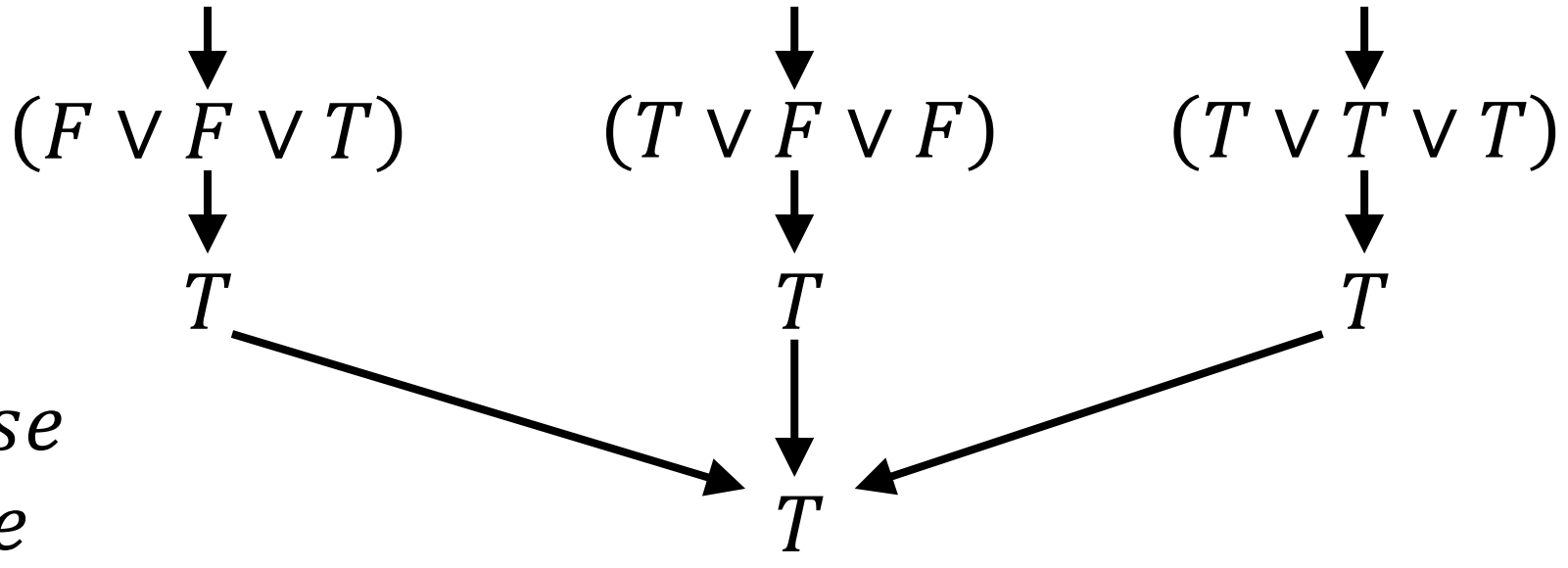
$$T \qquad\qquad T \qquad\qquad T$$

$x_1 = false$

$x_2 = true$

SAT & 3–SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$

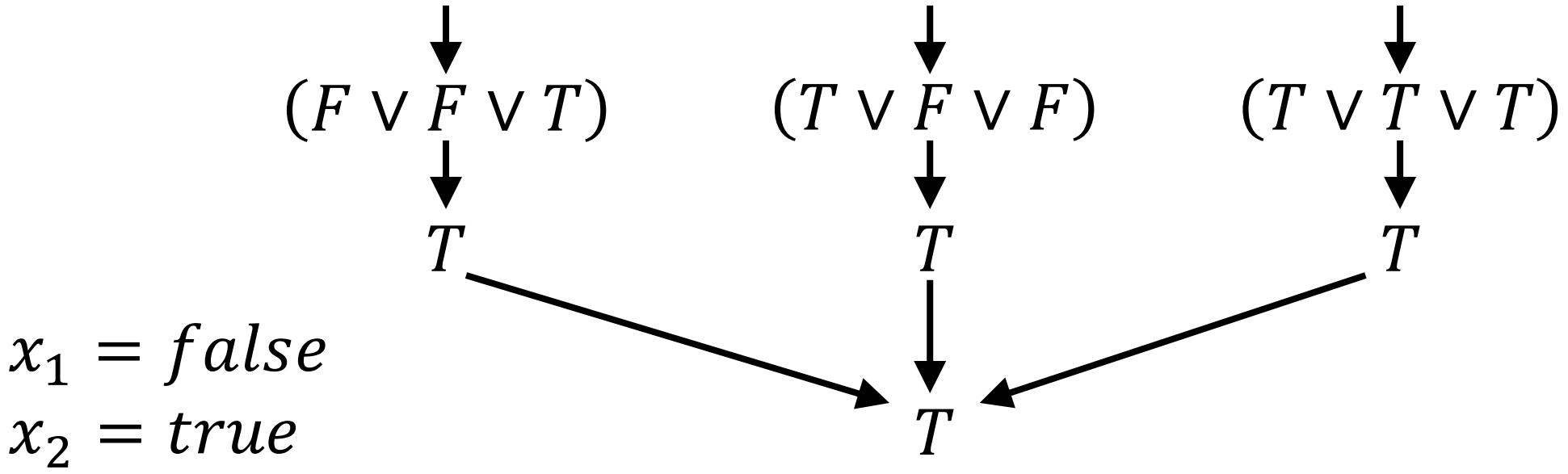


$x_1 = false$

$x_2 = true$

SAT & 3−SAT

$$\phi = (x_1 \vee x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_2)$$



$x_1 = false$
$x_2 = true$

$SAT = \{\langle\phi\rangle: \phi \text{ is a satisfiable formula}\}$
$3SAT = \{\langle\phi\rangle: \phi \text{ is a satisfiable formula with 3 variables per clause}\}$

Max 3-SAT

Given a 3-SAT instance (with unique variables in each clause), set variable values so that the number of satisfied clauses is maximized.

$$X = \{x_1, x_2, x_3, x_4\}$$

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee x_4)$$

{true, true, false, false} → true ∧ false ∧ true
{true, false, true, false} → true ∧ true ∧ true

Max 3-SAT

Given a 3-SAT instance (with unique variables in each clause), set variable values so that the number of satisfied clauses is maximized.

$$X = \{x_1, x_2, x_3, x_4\}$$

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee x_4)$$

{true, true, false, false} → true ∧ false ∧ true
{true, false, true, false} → true ∧ true ∧ true

Algorithm?

Max 3-SAT

Given a 3-SAT instance (with unique variables in each clause), set variable values so that the number of satisfied clauses is maximized.

$$X = \{x_1, x_2, x_3, x_4\}$$

$$(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee x_4) \wedge (x_1 \vee x_3 \vee x_4)$$

{true, true, false, false} → true ∧ false ∧ true
{true, false, true, false} → true ∧ true ∧ true

Algorithm: Assign x_i = true with probability 1/2.
Otherwise, set it to false.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$= \sum_e E[X_e]$

$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $=$?

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $=$?

All possibilities:

(T,T,T), (T,T,F), (T,F,T), (T,F,F)
(F,T,T), (F,T,F), (F,F,T), (F,F,F)

Given a 3-SAT instance **(with unique variables in each clause)**, set variable values so that the number of satisfied clauses is maximized.

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

All possibilities:

(T,T,T), (T,T,F), (T,F,T), (T,F,F)
(F,T,T), (F,T,F), (F,F,T), (F,F,F)

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\, OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\, OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

Probability of anything but that?

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign x_i = true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Alternate Argument:

(F,F,F) is the only failing case

Probability of **(F,F,F)**?

$$\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$$

Probability of anything but that?

$$1 - \frac{1}{8} = \frac{7}{8}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$E[Z_c] =$?

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e]$

$$= \sum_e E[X_e]$$

$$= \sum_e \frac{2}{3} = \frac{2}{3}|E|$$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1\left(\frac{7}{8}\right) + 0\left(\frac{1}{8}\right) = \frac{7}{8}$$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e] = \sum_e \frac{2}{3} = \frac{2}{3}|E|$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1\left(\frac{7}{8}\right) + 0\left(\frac{1}{8}\right) = \frac{7}{8}$$

clauses we expect to satisfy $= E[ALG] = E[\sum_c Z_c] = \sum_c E[Z_c]$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e] = \sum_e \frac{2}{3} = \frac{2}{3}|E|$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1\left(\frac{7}{8}\right) + 0\left(\frac{1}{8}\right) = \frac{7}{8}$$

clauses we expect to satisfy $= E[ALG] = E[\sum_c Z_c] = \sum_c E[Z_c] = \sum_c \frac{7}{8} = \frac{7}{8}|C|$

Max 3-Coloring

Algorithm: For each vertex, randomly assign it one of the three colors with probability $= \frac{1}{3}$.

Probability that a single edge is satisfied $= \frac{2}{3}$.

Define $X_e = \begin{cases} 1, \text{ if } e \text{ is satisfied} \\ 0, \text{ if } e \text{ is not satisfied} \end{cases}$

$$E[X_e] = 1\left(\frac{2}{3}\right) + 0\left(\frac{1}{3}\right) = \frac{2}{3}$$

edges we expect to satisfy $= E[ALG] = E[\sum_e X_e] = \sum_e E[X_e] = \sum_e \frac{2}{3} = \frac{2}{3}|E|$

$$\Rightarrow E[ALG] = \frac{2}{3}|E| \geq \frac{2}{3}\ OPT$$

Max 3-SAT

Algorithm: Assign $x_i =$ true with probability $\frac{1}{2}$. Otherwise, set it to false.

Probability that a clause is satisfied $= \frac{7}{8}$.

Define $Z_c = \begin{cases} 1, \text{ if clause } c \text{ is satisfied} \\ 0, \text{ if clause } c \text{ is not satisfied} \end{cases}$

$$E[Z_c] = 1\left(\frac{7}{8}\right) + 0\left(\frac{1}{8}\right) = \frac{7}{8}$$

clauses we expect to satisfy $= E[ALG] = E[\sum_c Z_c] = \sum_c E[Z_c] = \sum_c \frac{7}{8} = \frac{7}{8}|C|$

$$\Rightarrow E[ALG] = \frac{7}{8}|C| \geq \frac{7}{8}\ OPT$$

ILP-Based Approximations
CSCI 532

Vertex Cover ILP

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$\boldsymbol{x_i \in \{0, 1\}}$**, for each vertex** $\boldsymbol{i}$

$\in$ NP-Complete

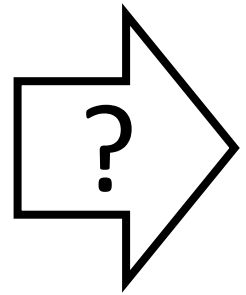
Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$\boldsymbol{0 \leq x_i \leq 1}$**, for each vertex** $\boldsymbol{i}$

$\in$ P

Vertex Cover ILP

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i



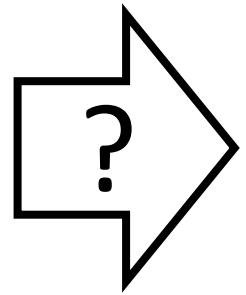
Vertex Selection

If $x_i = 1$, what should we do with vertex i?

Vertex Cover ILP

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i



Vertex Selection

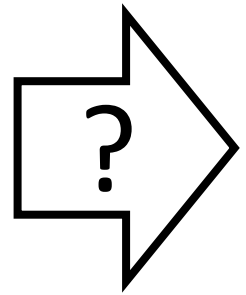
If $x_i = 1$, what should we do with vertex i? Add to subset S

If $x_i = 0$, what should we do with vertex i?

Vertex Cover ILP

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i



Vertex Selection

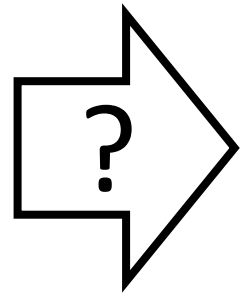
If $x_i = 1$, what should we do with vertex i? Add to subset S

If $x_i = 0$, what should we do with vertex i? Don't add to subset S

Vertex Cover ILP

Vertex Cover: Given graph, find the smallest subset of vertices such that every edge in the graph has at least one vertex in the subset.

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i



Vertex Selection

If $x_i = 1$, what should we do with vertex i? Add to subset S

If $x_i = 0$, what should we do with vertex i? Don't add to subset S

If $x_i = \frac{126}{337}$, what should we do with vertex i?

Vertex Cover ILP

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i

+

If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Vertex Cover ILP

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Is S a vertex cover?

Vertex Cover ILP

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Is S a vertex cover?

Yes. For every edge, $x_i + x_j \geq 1$.

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Is S a vertex cover?

Yes. For every edge, $x_i + x_j \geq 1$. Thus, at least one of x_i or $x_j \geq \frac{1}{2}$.

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Is S a vertex cover?

Yes. For every edge, $x_i + x_j \geq 1$. Thus, at least one of x_i or $x_j \geq \frac{1}{2}$. So for every edge, at least one of its vertices will be in S.

Vertex Cover ILP

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

What is the relationship between ALG $= |S|$ and OPT?

Vertex Cover ILP

Objective:	$\min \sum_i x_i$
Subject to:	$x_i + x_j \geq 1$, for each edge $e = (i, j)$
	$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Can we bound OPT from below?

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Can we bound OPT from below?

Let x_{ILP} and x_{LP} be the set of x values found by the ILP and LP

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Can we bound OPT from below?

Let x_{ILP} and x_{LP} be the set of x values found by the ILP and LP

Claim: $\sum x_{LP} \leq$ OPT.

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Can we bound OPT from below?

Let x_{ILP} and x_{LP} be the set of x values found by the ILP and LP

Claim: $\sum x_{LP} \leq$ OPT.

Proof: OPT $=$?

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Can we bound OPT from below?

Let x_{ILP} and x_{LP} be the set of x values found by the ILP and LP

Claim: $\sum x_{LP} \leq$ OPT.

Proof: OPT $= \sum x_{ILP}$, where $x_i \in \{0,1\}$...?

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

Can we bound OPT from below?

Let x_{ILP} and x_{LP} be the set of x values found by the ILP and LP

Claim: $\sum x_{LP} \leq$ OPT.

Proof: OPT $= \sum x_{ILP}$, where $x_i \in \{0,1\}$. When x_i is relaxed so that $0 \leq x_i \leq 1$, this gives more possibilities to further decrease $\sum_i x_i$. Thus, $\sum x_{LP} \leq$ OPT.

Vertex Cover lLP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

How does $\sum \mathrm{x_{LP}}$ relate to ALG?

$\sum \mathrm{x_{LP}} = \sum_{x_i \in \mathrm{x_{LP}}} x_i \geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} x_i$, because...?

Vertex Cover lLP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

How does $\sum \mathrm{x_{LP}}$ relate to ALG?

$\sum \mathrm{x_{LP}} = \sum_{x_i \in \mathrm{x_{LP}}} x_i \geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} x_i$, because it's a subset of $\mathrm{x_{LP}}$

Vertex Cover lLP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

How does $\sum x_{LP}$ relate to ALG?

$\sum x_{LP} = \sum_{x_i \in x_{LP}} x_i \geq \sum_{x_i \in x_{LP}: x_i \geq \frac{1}{2}} x_i$, because it's a subset of x_{LP}

$\geq \sum_{x_i \in x_{LP}: x_i \geq \frac{1}{2}} \frac{1}{2}$, because...?

Vertex Cover lLP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

How does $\sum \mathrm{x_{LP}}$ relate to ALG?

$$\sum \mathrm{x_{LP}} = \sum_{x_i \in \mathrm{x_{LP}}} x_i \geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} x_i, \text{ because it's a subset of } \mathrm{x_{LP}}$$

$$\geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} \frac{1}{2}, \text{ because each } x_i \text{ is at least } \frac{1}{2}$$

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

How does $\sum \mathrm{x_{LP}}$ relate to ALG?

$$\sum \mathrm{x_{LP}} = \sum_{x_i \in \mathrm{x_{LP}}} x_i \geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} x_i, \text{ because it's a subset of } \mathrm{x_{LP}}$$

$$\geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} \frac{1}{2}, \text{ because each } x_i \text{ is at least } \frac{1}{2}$$

$$= \frac{1}{2} \left| \left\{ x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2} \right\} \right|$$

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

How does $\sum \mathrm{x_{LP}}$ relate to ALG?

$$\sum \mathrm{x_{LP}} = \sum_{x_i \in \mathrm{x_{LP}}} x_i \geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} x_i, \text{ because it's a subset of } \mathrm{x_{LP}}$$

$$\geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} \frac{1}{2}, \text{ because each } x_i \text{ is at least } \frac{1}{2}$$

$$= \frac{1}{2} \left| \left\{ x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2} \right\} \right| = ?$$

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

How does $\sum \mathrm{x_{LP}}$ relate to ALG?

$$\sum \mathrm{x_{LP}} = \sum_{x_i \in \mathrm{x_{LP}}} x_i \geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} x_i, \text{ because it's a subset of } \mathrm{x_{LP}}$$

$$\geq \sum_{x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2}} \frac{1}{2}, \text{ because each } x_i \text{ is at least } \frac{1}{2}$$

$$= \frac{1}{2} \left| \left\{ x_i \in \mathrm{x_{LP}}: x_i \geq \frac{1}{2} \right\} \right| = \frac{1}{2} \text{ ALG}$$

Vertex Cover ILP

Objective: $\min \sum_i x_i$
Subject to: $x_i + x_j \geq 1$, for each edge $e = (i, j)$
$0 \leq x_i \leq 1$, for each vertex i

+ If $x_i \geq \frac{1}{2}$, add vertex i to our subset S.

What is the relationship between ALG and OPT?

$\sum x_{LP} \geq \frac{1}{2}$ ALG and $\sum x_{LP} \leq$ OPT

ALG ≤ 2 OPT