# Randomized Computation

- Deterministic and (Las Vegas & Monte Carlo) Randomized Algorithms
- Probability Review
- Probabilistic Analysis of deterministic QUICK-SORT Algorithm
- RANDOMIZED-SELECT and RANDOMIZED-QUICK-SORT
- Max-Cut
- Min-Cut
- MAX-3-SAT and Derandomization
- Closest Pair
- Randomized Complexity Classes

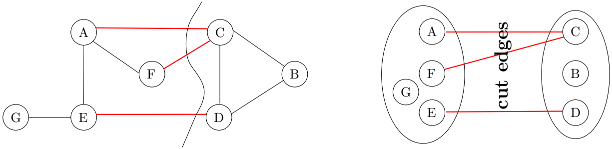
IMDAD ULLAH KHAN

## Cuts in Graphs

- Cuts in graphs are very useful structures
- It has applications in network flows, statistical physics, circuit design, complexity, and approximation theory

**A cut in $G$ is a subset $S \subset V$**

- Denoted as $\left[S, \overline{S}\right]$
- $S = \emptyset$ and $S = V$ are trivial cuts, we assume that $\emptyset \neq S \neq V$
- A graph on $n$ vertices has $2^n$ cuts
- An edge $(u, v)$ crosses the cut $\left[S, \overline{S}\right]$, if $u \in S$ and $v \in \overline{S}$

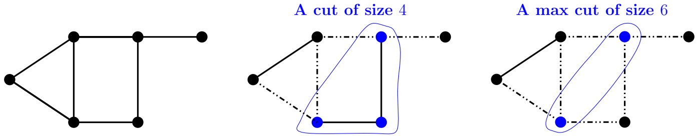


# The MAX-CUT problem

A cut in $G$ is a subset $S \subset V$

- Denoted as $[S, \overline{S}]$
- An edge $(u, v)$ crosses the cut $[S, \overline{S}]$, if $u \in S$ and $v \in \overline{S}$

Size (or cost) of a cut in the number of crossing edges




- In weighted graphs size of cut is the sum of weights of crossing edges

The MAX-CUT($G$) problem: Find a cut in $G$ of maximum size?

# The MAX-CUT problem

A cut in $G$ is a subset $S \subset V$

- Denoted as $[S, \overline{S}]$
- An edge $(u, v)$ is crossing the cut $[S, \overline{S}]$, if $u \in S$ and $v \in \overline{S}$

Size (or cost) of a cut in the number of crossing edges

- In weighted graph size of cut is the sum of weights of crossing edges

The MAX-CUT$(G)$ problem: Find a cut in $G$ of maximum size/weight?

The decision version of the MAX-CUT$(G)$ problem is NP-COMPLETE

# Randomized Approximation Algorithm for Max-Cut

The algorithm construct $[S, \overline{S}]$ by placing each vertex in $S$ or $\overline{S}$ at random

**Algorithm** RAND-MAX-CUT($G$)

$S \leftarrow \emptyset$
**for** each vertex $v \in V$ **do**
  $S \leftarrow S \cup \{v\}$ with probability $1/2$

For edge $e$, let $C_e$ be an indicator random variable, where

$$C_e = \begin{cases} 1 & \text{if } e \text{ crosses the cut } [S, \overline{S}] \\ 0 & \text{otherwise} \end{cases} \qquad size\left([S, \overline{S}]\right) = \sum_{e \in E} C_e$$

$$E\left[size\left([S, \overline{S}]\right)\right] = E\left[\sum_{e \in E} C_e\right] = \sum_{e \in E} E[C_e]$$

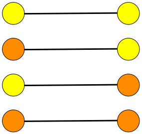
# Randomized Approximation Algorithm for MAX-CUT

$$E\left[size\left(\left[S,\overline{S}\,\right]\right)\right] \;=\; \sum_{e\in E} E[C_e]$$

$$E[C_e] = Pr\left[e = (u,v) \text{ crosses the cut } \left[S,\overline{S}\,\right]\right]$$

$$= Pr\left[(u \in S \wedge v \in \overline{S}) \vee (u \in \overline{S} \wedge v \in S)\right]$$

$$= \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$$



**Four possibilities for edge**


$$E\left[size\left(\left[S,\overline{S}\,\right]\right)\right] \;=\; \sum_{e\in E} E[C_e] \;=\; \sum_{e\in E} {}^1\!/_2 \;=\; {}^{|E|}\!/_2 \;=\; {}^{m}\!/_2$$

This cut has size at least $^1/_2$ of the max cut ▷ all cuts $\leq m$

Run time is $O(n)$ ▷ one iteration over the vertices

## Improvement over RAND-MAX-CUT($G$)

- The algorithm is good on average, but their is no (probabilistic) guarantee on the quality of the output of RAND-MAX-CUT($G$)
- We use the magic of repeated trial to amplify the probability
  - Run RAND-MAX-CUT($G$) $k$ times on $G$
  - Return the largest cut found
- For large $k$, this returns a large cut with (desired) high probability
  - We will show that we get a cut of size $m/4$ with very high probability
- Runtime is $O((m+n)k)$
  - $k$ rounds of running RAND-MAX-CUT($G$)
  - $O(n)$ to build the cut, $m$ to determine the size

# Randomized Approximation Algorithm for MAX-CUT

- Let $X_1, X_2, \cdots, X_k$ be sizes of the cuts found by each run of RAND-MAX-CUT$(G)$
- Let $\mathcal{E}$ be the event the meta algorithm produces a cut of size less than $^m/_4$

$$\mathcal{E} = \bigcap_{i=1}^{k} \left(X_i \leq \frac{m}{4}\right) \qquad \triangleright \text{ all } X_i \text{ are } \leq {}^m/_4$$

$$Pr\left[\mathcal{E}\right] = Pr\left[\bigcap_{i=1}^{k} \left(X_i \leq \tfrac{m}{4}\right)\right] = \prod_{i=1}^{k} Pr\left[X_i \leq \tfrac{m}{4}\right] \qquad \triangleright X_i\text{'s are independent}$$

- Let $Y_1, Y_2, \cdots Y_k$ be random variables, defined as

$$Y_i = m - X_i \qquad E[Y_i] = m - E[X_i] = m - {}^m/_2 = {}^m/_2$$

**Markov Inequality:** If $Z$ is a non-negative random variable and $a > 0$, then

$$Pr\left[Z \geq a\,E[Z]\right] \leq {}^1/_a \iff Pr\left[Z \geq a\right] \leq {}^{E[Z]}/_a$$

$$Pr\left[\mathcal{E}\right] = \prod_{i=1}^{k} P\left(Y_i \geq \frac{3m}{4}\right) \leq \prod_{i=1}^{k} \frac{E[Y_i]}{{}^{3m}/_4} = \prod_{i=1}^{k} \frac{{}^m/_2}{{}^{3m}/_4} = \prod_{i=1}^{k} \frac{2}{3} = \left(\frac{2}{3}\right)^k$$

## Randomized Approximation Algorithm for MAX-CUT

- If we output the largest cut out of $k$ runs of RAND-MAX-CUT$(G)$ $k$, the probability that we don't get $\geq m/4$ edges is at most $(2/3)^k$
- The probability we do get $\geq m/4$ edges is at least $1 - (2/3)^k$
- Set $k = \log_{2/3} m$, the probability we get at least $m/4$ edges is $1 - 1/m$
- So the meta algorithm is $O((m+n)\log m)$-time randomized algorithm that finds a 0.25-approximation to max-cut with probability $1 - 1/m$

We got $(\epsilon, \delta)$-approximation algorithm with $\epsilon = 1/4$ and $\delta = 1/m$