

Randomized Algorithms

- 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
- Hashing, Bloom filters, Streams, Sampling, Reservoir sampling, Sketch

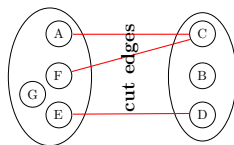
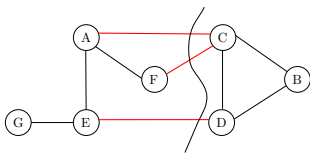
IMDAD ULLAH KHAN

Cuts in Graphs

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

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

- Denoted as $[S, \bar{S}]$
- $S = \emptyset$ 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** $[S, \bar{S}]$, if $u \in S$ and $v \in \bar{S}$

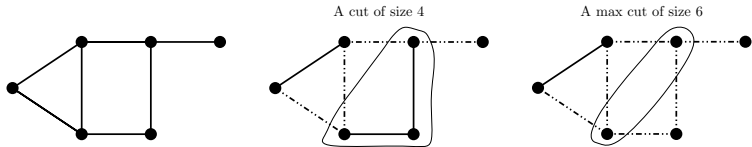


The MAX-CUT(G) problem

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

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

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(G) problem

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

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

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?

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

Randomized Approximation Algorithm for MAX-CUT(G)

The algorithm constructs $[S, \bar{S}]$ by placing each vertex in S or \bar{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

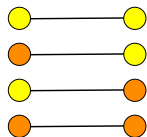
$$C_e = \begin{cases} 1 & \text{if } e \text{ crosses the cut } [S, \bar{S}] \\ 0 & \text{otherwise} \end{cases} \quad \text{size}([S, \bar{S}]) = \sum_{e \in E} C_e$$

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

Randomized Approximation Algorithm for MAX-CUT(G)

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

$$\begin{aligned} E[C_e] &= \Pr \left[e = (u, v) \text{ crosses the cut } [S, \bar{S}] \right] \\ &= \Pr \left[(u \in S \wedge v \in \bar{S}) \vee (u \in \bar{S} \wedge v \in S) \right] \\ &= \frac{1}{4} + \frac{1}{4} = \frac{1}{2} \end{aligned}$$



Four possibilities for edge e

● : $\in S$ ● : $\in \bar{S}$

$$E \left[\text{size} \left([S, \bar{S}] \right) \right] = \sum_{e \in E} E[C_e] = \sum_{e \in E} 1/2 = |E|/2 = m/2$$

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

Randomized Approximation Algorithm for MAX-CUT(G)

Improvement over RAND-MAX-CUT(G)

- The algorithm is good on average, but there 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(G)

- Let X_1, X_2, \dots, 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 $m/4$

$$\mathcal{E} = \bigcap_{i=1}^k \left(X_i \leq \frac{m}{4} \right)$$

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

- Let Y_1, Y_2, \dots, 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

$$Pr[Z \geq a] \leq E[Z]/a \iff Pr[Z \geq a E[Z]] \leq 1/a$$

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

Randomized Approximation Algorithm: Max-Cut

- If we output the largest cut out of k runs of $\text{RAND-MAX-CUT}(G)$, 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 $1 - 1/m$

Called (ϵ, δ) -approximation algorithm