Randomized Algorithm

(Lecture 2: Randomized Min_Cut)

Instructor: Dr. Guatam Das
Lecture note by Xin Jin

Min_Cut Porblem

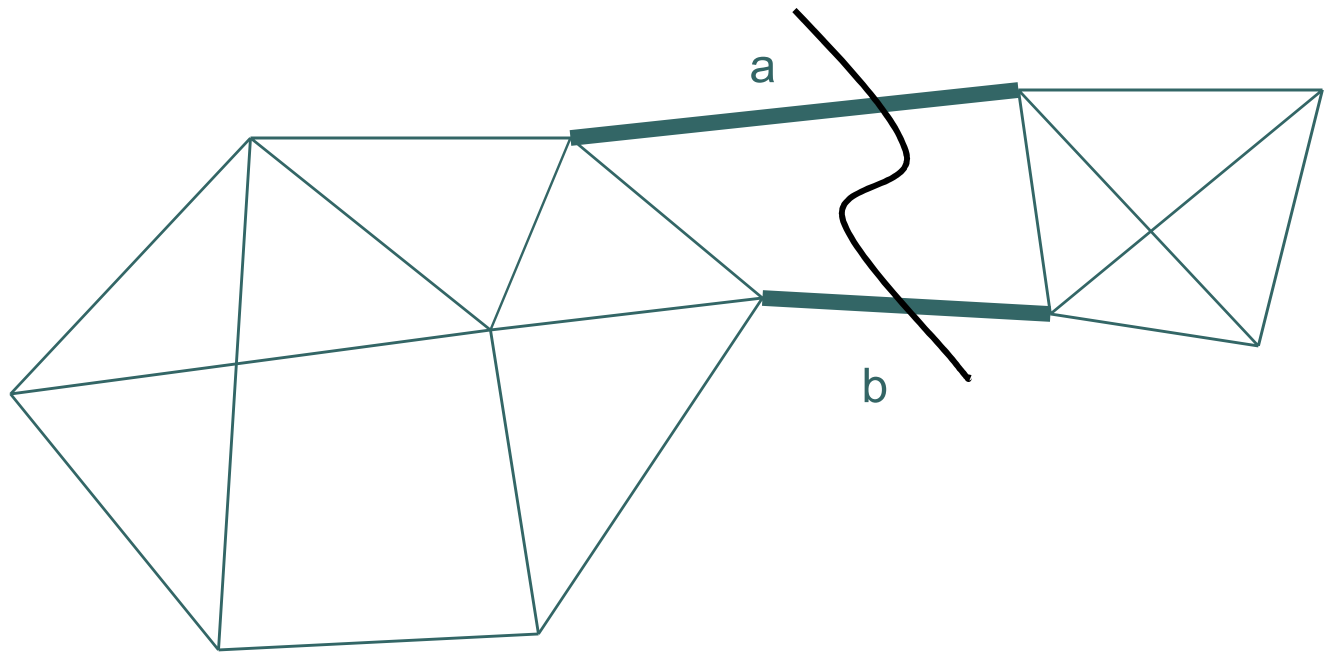
Definition:

Min_cut Problem is to find the minimum edge set C such that removing C disconnects the graph

Traditional Solution:

Max-flow: The maximum amount of flow is equal to the capacity of a minimum cut

Example of Min_Cut



e.g. Min_Cut = 2

Intuition

- Let a graph G has n nodes and size of min_cut = k, that is |C| = k

  then :

  degree for each node >= k

  total number of edges in G >= nk/2

Randomized Min_Cut

Input: a graph G(V, E), |V| = n
Output: min_cut C

Repeat:

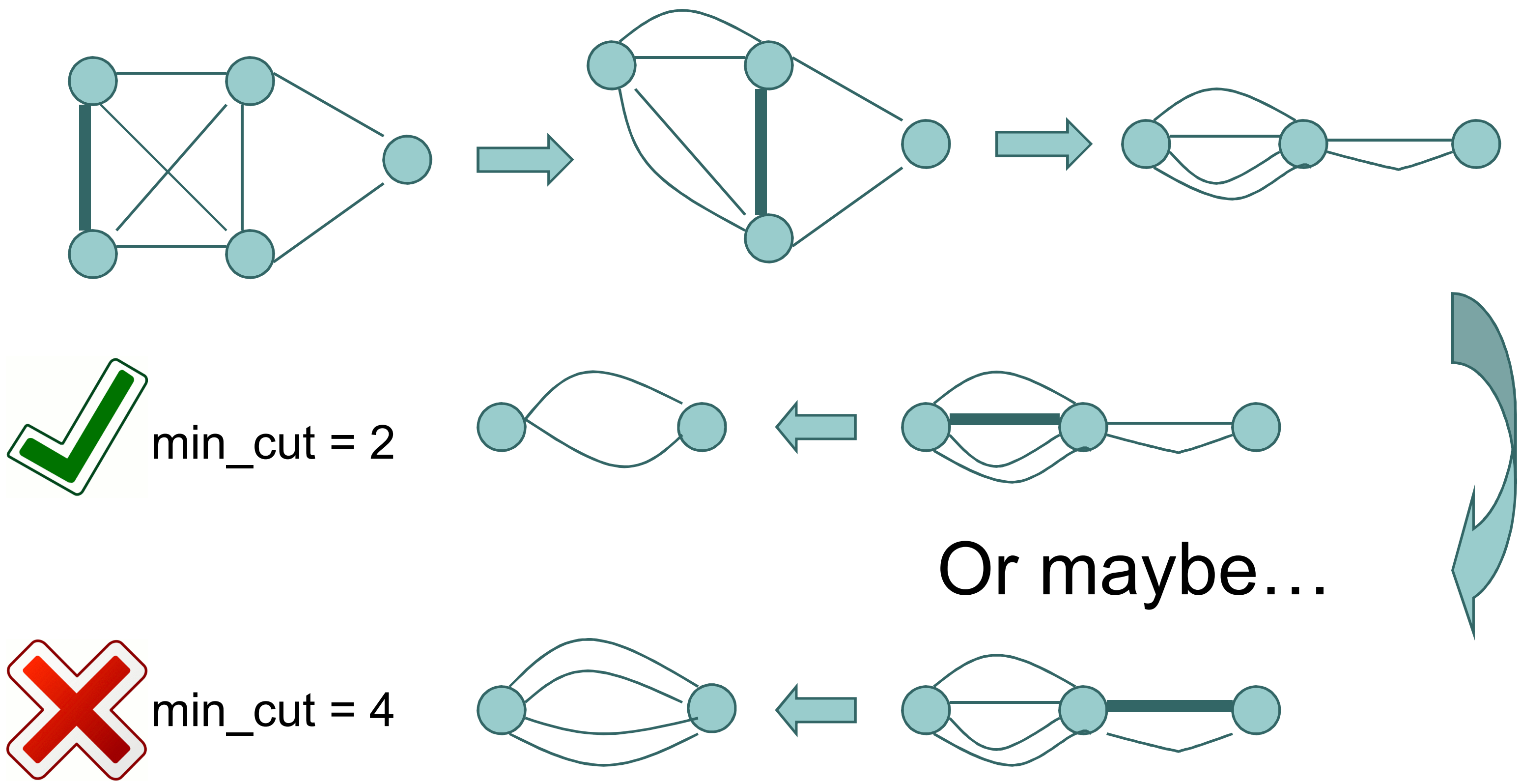
Pick any edge uniformly at random, collapse it and remove self-loops

Until:

|V| down to 2

*Running time is O(n-2)

Example of Randomized Min_Cut

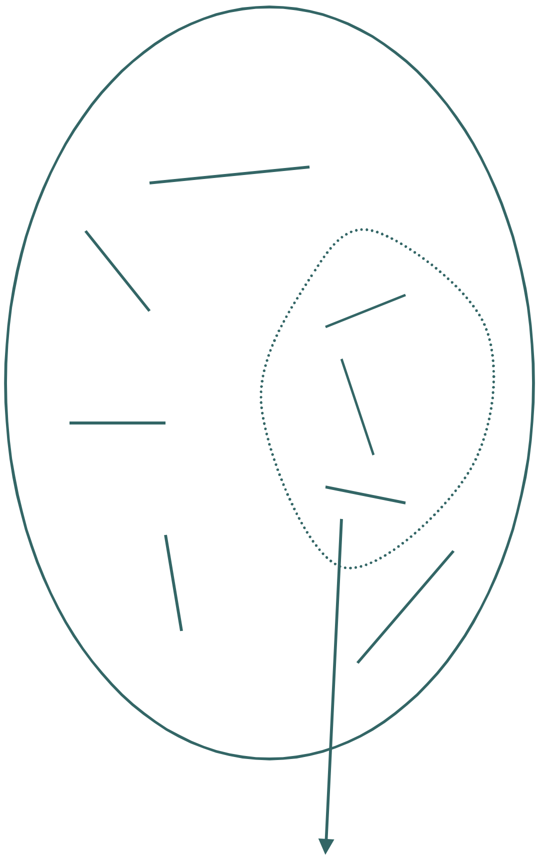


Las Vegas VS Monte Carlo

- Las Vegas Algorithm: It always produces the correct answer and the expected running time is finite (e.s.p. randomized quick sort)

- Monte Carlo Algorithm: It may produce incorrect answer but with bounded error probability (e.s.p. randomized min_cut)

Analysis



- Probability of the first edge $\notin$ C

  Prob = (kn/2 – k) / (kn/2)

  = (n-2) / n

- Probability of the second edge $\notin$ C

  Prob = (k(n-1)/2 – k) / (k(n-1)/2)

  = (n-3) / (n-1)

● ● ●

Analysis

Iteration	Probability of avoiding C
1	(n – 2) / n
2	(n – 3) / (n – 1)
3	(n – 4) / (n – 2)
4	(n – 5) / (n - 3)
…	
n - 2	1 / 3

Prob. Of outputting C:

$$\Pr \geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdots \frac{2}{4} \cdot \frac{1}{3} = \frac{2}{n(n-1)}$$

Analysis

- Probability of getting a min_cut is at least 2/n(n-1)

  Might look like small, but gets bigger after repeating the algorithm

  e.s.p. If algorithm is running twice, probability of outputting C would be:

$$Pr = 1 - \left(1 - \frac{2}{n(n-1)} \right) ^ 2$$

Analysis

- Let r be the number of running times of algorithm

Total running time = $O(n*r)$

Probability of getting C:

$$Pr \approx 1 - \left(1 - \frac{2}{n^2} \right) ^ r$$

Analysis

If $r = \frac{n^2}{2}$

then

$$T(n) = O(n*n^2 / 2) = O(n^3)$$

$$Pr = 1 - \left(1 - \frac{2}{n^2}\right) ^ {\frac{n^2}{2}}$$

$$\approx 1 - 1/e$$