Randomnized Algorithm for Finding Min-Cut

Renming Qi Vincent Chiu

December 1, 2016

Motivation

- Ford fulkerson has a large running time for finding the minimum cut for large integer flows. It runs in pseudo polynomial time. This is a big drawback.
- There exists a randomnized algorithm that could solve the minimum cut problem that is not affected by large integer flows.

Waiting for first Success

Bob has only p=10% possibility to pass a math exam, but you can take the exam as many times as you like.

- What's possibility of taking the exam only once?
- How about 10 times, 100 times

- Using the following formula

$$\Pr\left(\text{Within N exams, Bob passes}\right) = 1 - (1-p)^N$$

With N=5, P = 0.41. With N=10, P = 0.65. With N=100, P = 0.99
So after a large number of tries, Bob will finally pass the exam for almost sure even if he know nothing.

Randomnized Algorithms

what's randomnized algorithms

Randomnized algorithm is an algorithm that makes random decision when it processes the input. So this kind of algorithm cannot always get the right answer. However, we have showed even if it can only succeed with small probability, we can get the right answer by running it many times.

Randomnized Algorithms

what's randomnized algorithms

Randomnized algorithm is an algorithm that makes random decision when it processes the input. So this kind of algorithm cannot always get the right answer. However, we have showed even if it can only succeed with small probability, we can get the right answer by running it many times.

Why we need that

1. Simpler
2. New way to analysis complex system

The Problem

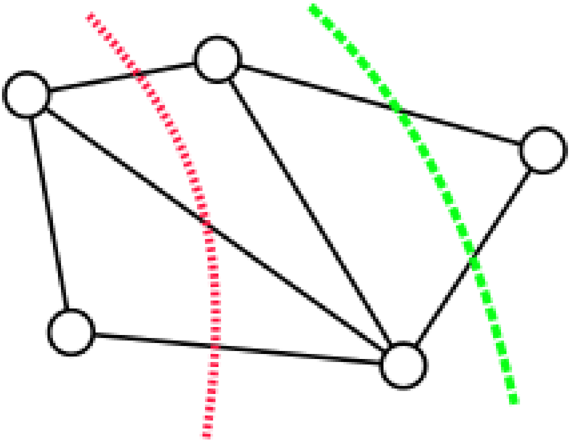
Instance

An undirected graph G(V, E)

Question

Find the global minimum cut. Note that global here refers to the fact that any cut is allowed, as there is no sink or source in our graph for this type of problem.

The minimum cut is a natural measure of how robust the graph is. If you want to disconnect the graph into 2 components, it is the least number of edges you have to cut.



What is contraction algorithm?

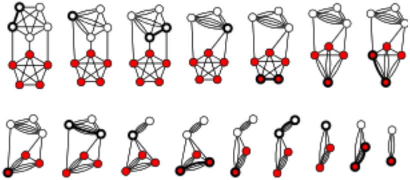
- David Karger discovered the contraction algorithm in 1992.
- Multigraph: An undirected graph that is allowed to have multiple edges between the same pair of nodes.

What is contraction algorithm?

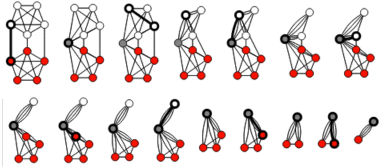
- Randomly choose edge e = (u,v) of G, we take this edge and we create a new graph G' where the nodes that edge e was connected to become one node.
- Recursively call the contraction algorithm until only 2 nodes are left.

What is contraction algorithm?

Successful run of the contraction algorithm on a graph with 10 nodes. The minimum cut is of size 3, note the 3 edges between the white node and the red node at the rightmost diagram.



An error case



Analysis of Contract Algorithm

Success Rate of Contract Algorithm

The Contraction Algorithm returns a global min-cut of G with probability at least $1/\binom{n}{2}$

Analysis of Contract Algorithm

Success Rate of Contract Algorithm

The Contraction Algorithm returns a global min-cut of G with probability at least $1/\binom{n}{2}$

Proof

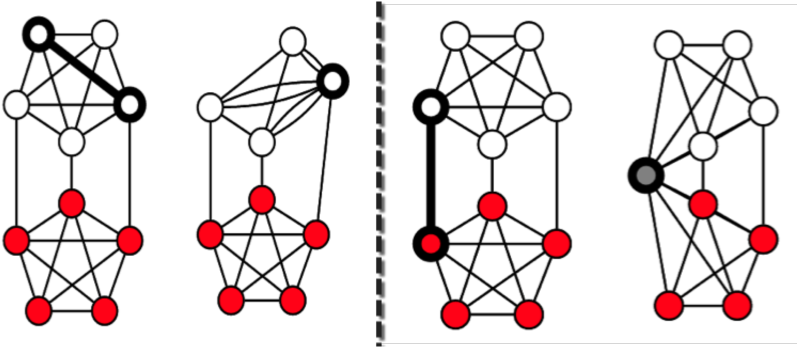
Let's assume the global min-cut in graph G(V,E) is (A,B) and it has size k. The set of cut edges is F

Case.1 An edge in F is contracted. Then game over.

Case.2 The contracted edge is not in F. We can continue contracting.

So, how large or small is the possibility p_1 that we end our game at first step?

Analysis of Contract Algorithm



Case.1 An edge in F is contracted. Then game over.

Case.2 The contracted edge is not in F. We can continue.

Analysis of Contract Algorithm

Proof Cont'd

When contracting the first edge randomly,

1. $\forall v \in V, deg(v) \geq k$
2. $|E| \geq \frac{1}{2}nk$
3. $p_1 \leq \frac{k}{\frac{1}{2}nk} = \frac{2}{n}$

After j iteration, there are n-j super-nodes in the current graph G'.

1. $\left|E'\right| \geq \frac{1}{2}(n-j)k$
2. $p_j \leq \frac{k}{\frac{1}{2}(n-j)k} = \frac{2}{n-j}$

$$\Pr(\text{Correct}) \geq \prod_{i=0}^{n-3}(1-p_n) = \prod_{i=0}^{n-3}\left(1-\frac{2}{n-i}\right) = \binom{n}{2}^{-1}$$

Analysis of Contract Algorithm

Success Rate of Contract Algorithm

The Contraction Algorithm returns a global min-cut of G with probability at least $1/\binom{n}{2}$

Time Complexity

After contracting $n^2 \log n$ times, the probability that we do not find the optimal solution

$$p_e = (1 - \binom{n}{2}^{-1})^{n^2 \log n} \approx 1/n$$

so the running time is $\mathcal{O}(n^2 \log n \times |E|)$

The End