

Karger's Mincut

Abdeslam El-Haman Abdeslam

October 19, 2017

1 Analysis

Karger's algorithm is a Montecarlo algorithm that estimates an answer to the mincut problem. The algorithm contracts edges randomly until there is only 2 vertices in the graph, then it's a cut.

The probability that the algorithm succeeds is of $\frac{2}{n(n-1)}$. This bound can be improved to at least $1 - \frac{1}{e}$ by repeating the algorithm $\frac{n^2}{2}$ times.

This algorithm then has been tried with 3 distinct graphs. The first one is the famous Zachary's Karate Club The second one is a "supervortex" graph where it simulates a simple graph where every node is a K_x subgraph. The third one will be a graph composed by 2 K_x subgraphs connected by $x - 2$ edges, building so a minimum cut between the 2 subgraphs.

2 Implementation

This algorithm has been developed using the networkx library. I've chosen to use this library because it has an absolute abstraction of graphs and it implements already the edge contraction. It has also graph generators to create graphs from a lot of different kind of graphs.

Networkx also has an implementation of mincut-maxflow to find the minimum edge cut, so I've used it to check the results afterwards.

3 Results

For each graph we've computed the Karger algorithm $\frac{n^2}{2}$ times, to check the $\frac{1}{e}$ factor.

The first observation is that for every graph, the percentage of success in average is:

Karate Club	65%
Supervortex	84%
Double K_x	1%

We can see here that there's a little percentage of success in the graphs where the relation between the size of the minimum cut and the size of the graph. Then 100 tests have been done with that number of repetitions and for each one the results are 100% succesful. So the above $1 - \frac{1}{e}$ of success is not only correct but also 100% correct for the graphs that I've chosen. To have better

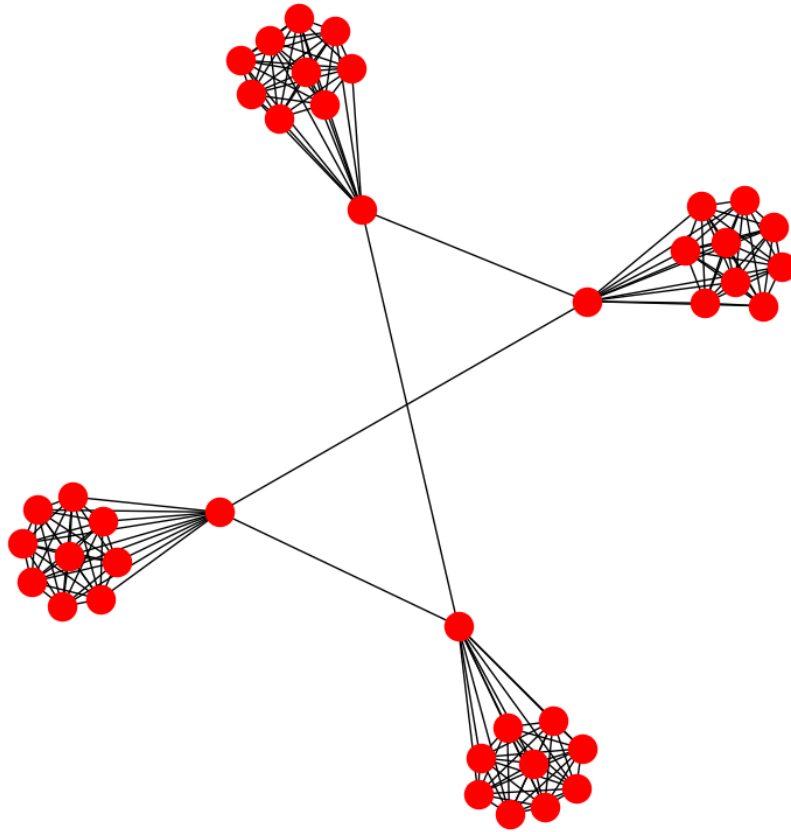


Figure 1: The "supervertex" graph

results, we should have a bigger graph with bigger cuts, but for that we'd need a better CPU.