Min-Cuts Algorithms

Hu SiXing, Hakki Can Karaimer, Pan An,
Philipp Keck, Taehoon Kim

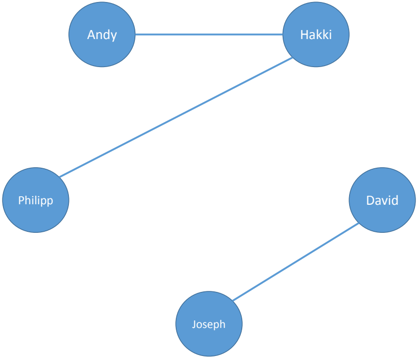
National University of Singapore

March 14, 2016

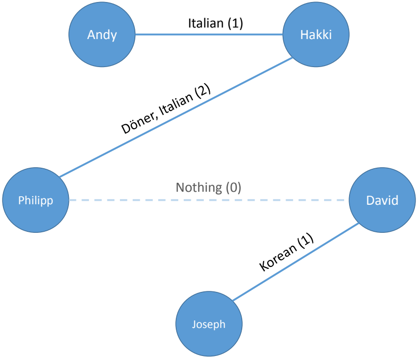
Motivation

Motivation

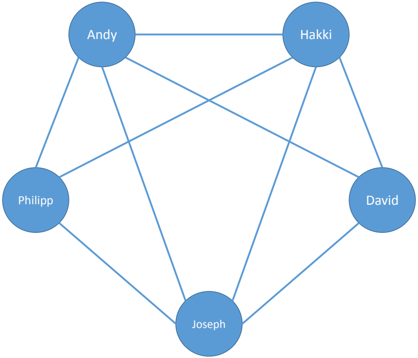
Food Tour Example



Food Tour Example



Food Tour Example



Definition

Cut

- Undirected graph $G = (V, E)$, $n = \|V\|$, $m = \|E\|$
- Cut problems can be described as partitioning V in to S and $\overline{S}$, where $S \subset V$:

Description

- Alternatively, finding a subset of E that, if removed, splits the graph into two connected componants.
- Weight: number of edges between S and $\overline{S}$:

$$w(S, \overline{S}) = \left\| \{u, v\} \in E \mid u \in S \wedge v \in \overline{S} \right\|$$

Definition

Min-Cut

A cut with minimum weight and $S, \overline{S} \neq \phi$

Other Cut Problems

- Directed Cut: Given $s, t \in V$,ensure $s \in S \wedge t \in \overline{S}$;
- K-Cut: Cuts the graph into k connected componants;
- Sparsest Cut: The sparsest cut problem is to bipartition the vertices so as to minimize the ratio of the number of edges across the cut divided by the number of vertices in the smaller half of the partition.

Generalization

Difficulty of Cuts

Minimum cut can be considered as a subset of $\mathrm{k-cut}$ where k is a fixed number 2.

- $\mathrm{k-cut}$ is $\mathrm{NP-Complete}$ problem if k is part of the input.
- Minimum cut is polynomial time calculable.

Add anything here?

Karger's Algorithm

- Contraction method is used.
- Randomized selection of Edges.
- Running multiple times of the algorithm will provide more accurate result.

Karger's Algorithm

- Basically one run of Karger's Algo takes $O(n^2)$ time;
- One time running the algorithm will make errors at the probability of $O(\frac{1}{n^2})$;
- Higher accuracy can be achieved by running the algorithm multiple times;
- It achieves error probability of $\frac{1}{\mathrm{poly}(n)}$ with $O(n^4 \log n)$ time.

Karger's Algorithm

- Improved Karger's algorithm was developed by Karger and Stein;
- Achieving $O(n^2 \log^3 n)$ running time;
- $O(\frac{1}{n})$ error probability.

Derivation will be given in the later part.

Karger's Algorithm

Comparison

- A trivial algorithm checks all x TODO possible subsets of V and computes the weight of the resulting cuts, which takes TODO time in total.
- Another algorithm(TODO the one that's based on many max-flow-computation) takes TODO time.

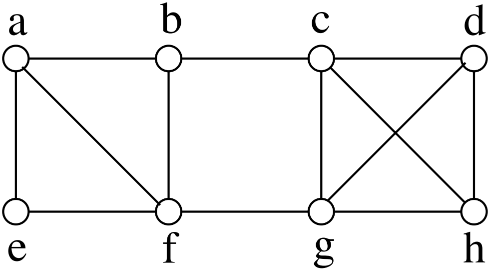
Well that's a lot of TODOS

Algorithm

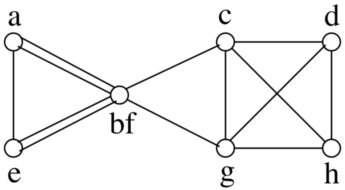
Karger's Algorithm:

repeat
  choose an edge (v, w) uniformly at random from G;
  let $G \leftarrow \frac{G}{(v,w)}$
until G has 2 vertices;

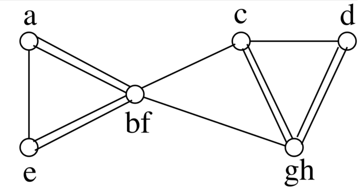
Algorithm



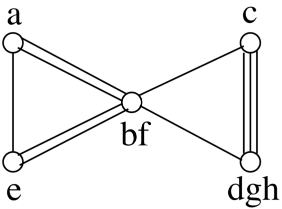
Algorithm



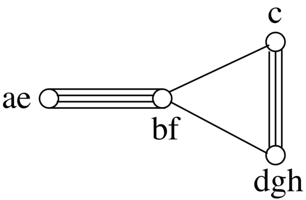
Algorithm



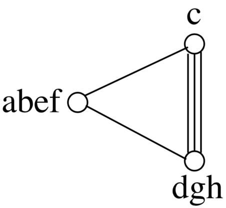
Algorithm



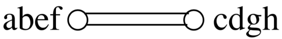
Algorithm



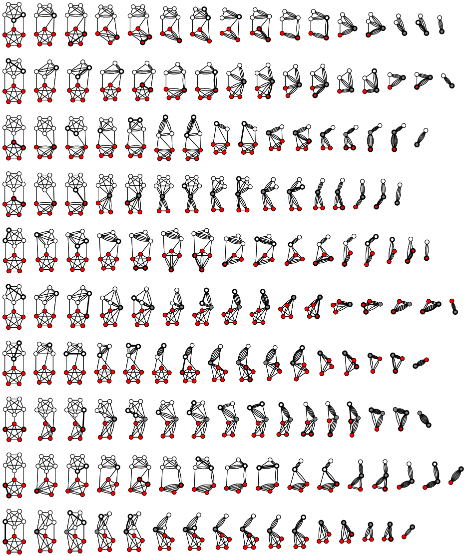
Algorithm



Algorithm



Results



Fact 1 – Sum of Degrees

$$\sum_{u \in V} \mathrm{degree}(u) = 2|E|$$

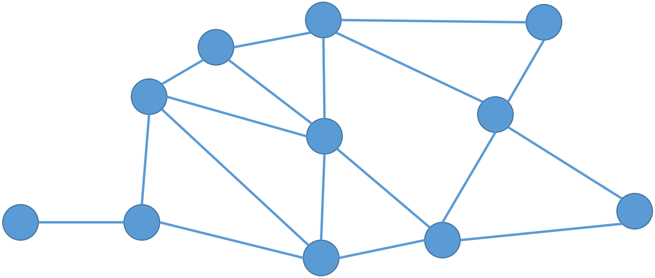
Every edge contributes exactly once to the degree of exactly two nodes.

Fact 2 – Average Degree

$$
\begin{aligned}
E(\mathrm{degree}(X)) &= \sum_{u \in V} \Pr(X = u) \cdot \mathrm{degree}(u) \\
&= \frac{1}{n} \sum_{u} \mathrm{degree}(u) = \frac{2|E|}{n}
\end{aligned} \tag{1}
$$

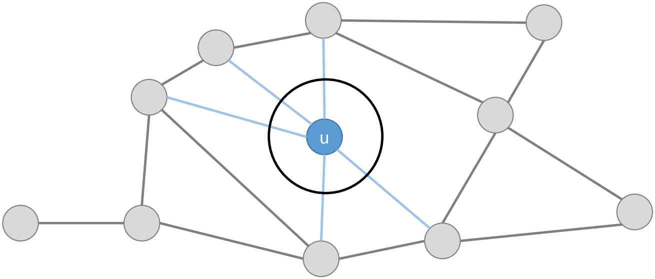
Fact 3 – Min-cut Size

The size of a min-cut is at most $\frac{2|E|}{n}$.



Fact 3 – Min-cut Size

The size of a min-cut is at most $\frac{2|E|}{n}$.



Fact 3 – Min-cut Size

The size of a min-cut is at most $\frac{2|E|}{n}$.

Proof

- For every node u, we have a cut of size $\mathrm{degree}(u)$.
- Not all nodes can have degree above average, i.e.

$$\exists u \in V: \ \mathrm{degree}(u) \leqslant \frac{2|E|}{n}$$

Fact 4 – Pr[edge across min-cut]

- Fix a certain min-cut in a graph.
- At most $2|E|/n$ of all edges are part of this min-cut.
- Choose a random edge out of all $|E|$ edges.
- $\Pr(\text{edge crosses the cut}) = \frac{2|E|/n}{|E|} = \frac{2}{n}$

Concentrating = Not Cutting

- TODO show the running example and explain how an edge that has been concentrated will never be cut.
- And the edges that remain at the end, because they have never been concentrated.

Success Probability

- Fix a certain min-cut
- We can never contract an edge from that cut
- $\Pr[firstedgeisnotinmincut] = 1 - \frac{2}{n}$
- Now $n - 1$ edges remaining, so $\Pr[secondedgeisnotinmincut] = (1 - \frac{2}{n-1})$

Success Probability

First Cut is not Fixed Cut

$$
\begin{aligned}
&\Pr[\text{fi}\,(\text{nalcutisnotthefixedcut}])\\
&\quad = (\Pr[\text{nocontractededgeisinminimuncut}])\\
&\quad \leqslant \left((1-\frac{2}{n})(1-\frac{2}{n-1})(1-\frac{2}{n-2})...(1-\frac{2}{4})(1-\frac{2}{3})\right)\\
&\quad = \left(\frac{n-2}{n}\right)
\end{aligned}
$$

Success Probability

Series Convere to e

- $e = \lim(1 + \frac{1}{n})^n$
- $\lim(1 + \frac{1}{n})^n = \lim(1 + \frac{1}{\frac{n}{a}})^{\frac{n}{a} \cdot a}$
- Let $x = \frac{n}{a}$

$$\lim(1 + \frac{1}{n})^n = \lim(1 + \frac{1}{x})^{x \cdot a} = (\lim(1 + \frac{1}{x})^x)^a = e^x$$

Boosting Success Probability

Assumn that we can get the best result by running the algorithm k times.

- Probability of at least one success: $1-(1-\frac{1}{\binom{n}{2}})^k$
- To make it an e-series, we need k to contain $\binom{n}{2}$.

Here we can use $a=-1$ and $k=\binom{n}{2}\cdot c\cdot \ln n$ Running time is thus promised:

$$O(k)\cdot O(n^2)=O(n^4\log n)$$

Boosting Success Probability

$$
\begin{aligned}
1-(1-\frac{1}{\binom{n}{2}})^k &= 1-(1+\frac{1}{\binom{n}{2}})^{-k} \\
&= 1-((1+\frac{1}{\binom{n}{2}})^{\binom{n}{2}})^{-c\cdot\ln n} \\
&= 1-e^{-c\cdot\ln n} = 1-(e^{\ln n})^{-c} \\
&= 1-\frac{1}{n^c}
\end{aligned}
$$

Boosting Success Probability

The error probability is thus:

$$O(\frac{1}{n^c})$$

Extended Karger's Analysis

One run succeeds with $\Omega(\frac{1}{\log n})$. If we run the algorithmk $= \log^2 n$ times:

$$
\begin{aligned}
Pr[atleastonerunsucceeds] &= 1-(1-\frac{1}{logn})^{\log^2 n} \\
&= 1-(1+\frac{1}{-logn})^{-\log n \cdot -\log n} \\
&= 1-e^{-\log n} = 1-\frac{1}{n} && \Rightarrow Erro
\end{aligned}
$$

References