The Min-Cut Problem

Spørgsmål 10 fra Exam Questions

Kevin Vinther

December 25, 2023

Table of Contents i

The Min-Cut Problem i

- Givet en undirected graf $G = (V, E)$, er et *cut* af G en *partition* af knuder V i to ikke-tomme sæt A og B.
- Størrelsen af et cut (A, B) er antallet af edges der har start i en partition, og ende i en anden.
- It **Global Min-cut** er et cut af minimum størrelse
- Global betyder at ethvert cut er tilladt, der er ingen source eller sink.

Theorem (13.4)

There is a polynomial-time algorithm to find a gloibal min-cut in an undirected graph G

- Vi skal først transformere grafen så der er directed kanter, og en *source* og *sink* (genkald flows)
- Måden dette bliver gjort på er følgende:
- Erstat hver $e = (u, v) \in E$ med $e' = (u, v)$ og $e'' = (v, u)$, hver med en kapacitet af 1.
- Vi kalder den resulterende graf G'.
- Herefter vælger vi arbitrært to knuder $s, t \in V$

Algoritme ii

- Vi ved at S må være i en af subsetsne, for the sake of example bruger vi A.
- Dette betyder at cuttet sepererer S fra alle knuder i B.
- VI kan så, for hver knude $\forall t \in V - s$ i grafen udregner vi et s-t cut i G'.
- Dermed må vi udregne $n - 1$ minimum $s - t$ cuts (Siden der er n knuder, og s tæller ikke med.)
- Ud af alle disse cuts, vælger vi den der er minimum.
- Slut på algoritmen. $\square$
- Trods at det lyder sådan, er denne algoritme meget simpel. Vi bruger Karger's Algoritme som eksempel.

- Algoritmen, i sin essens, er ikke super optimal. Der er dog blevet lavet optimeringer siden dens fødsel.
- Algoritmen arbejder på en *connected multigraph* $G = (V, E)$, altså, en undirected graf der tillader flere parallele kanter mellem det samme par af knuder.

Definition (Muligraph Wikipedia)

A multigraph is a graph which is permitted to have multiple edges, that is, edges that have the same end nodes. Thus two vertices may be connected by more than one edge.

- Algoritmen starter med at vælge en kant $e = (u, v) \in G$ *uniformt tilfældigt* og så **contracter** den den.

Karger's Algoritme ii

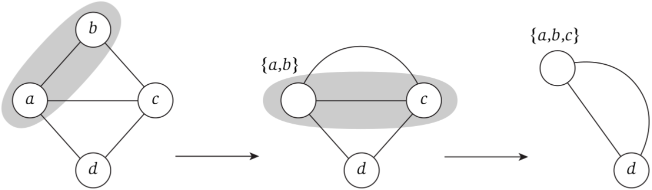

Figure 13.1 The Contraction Algorithm applied to a four-node input graph.

- Hvad dette betyder, er at der nu er en ny graf, hvori u og v er blevet til en ny knude w. Alle andre knuder forbliver sig selv.
- Alle kanter hvor $e = (u, v)$ fjernes, alle andre forbliver.
- Algoritmen kører så rekursivt på G', og bliver ved med at vælge en kant tilfældigt.

Karger's Algoritme iii

- Som disse rekursive kald fortsætter, er de konsttituerende knuder set som **superknuder**, hver superknude w er $S(w) \subseteq V$ som er blevet "slugt" og har produceret w.
- Algoritmen terminerer når den når en graf med to supernoder v_1 og v_2 (sandsynligvis med en masse parallele kanter immelem sig).
- Hver af disse super-noder v_i har en korresponderende subset $S(v_i) \subseteq V$, konsisternde af knuder som er blevet contracted ind i sig, og disse to sæt $S(v_1)$ og $S(v_2)$ returnerer vi så som svar i form af par.

Karger's Algoritme iv

```
The Contraction Algorithm applied to a multigraph G = (V, E):
  For each node v, we will record
          the set S(v) of nodes that have been contracted into v
      Initially S(v) = {v} for each v
  If G has two nodes v1 and v2, then return the cut (S(v1), S(v2))
  Else choose an edge e = (u, v) of G uniformly at random
      Let G' be the graph resulting from the contraction of e,
          with a new node z_uv replacing u and v
      Define S(z_uv) = S(u) ∪ S(v)
      Apply the Contraction Algorithm recursively to G'
  Endif
```

Analyse af algoritmen i

Theorem

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

- Vi fokuserer på et globalt min-cut (A, B) af G.
- Vi antager at det (i.e. cuttet) tager størrelsen k.
- Altså: der er et sæt F af k kanter, med en ende i A og en anden i B
- Vores mål er at finde et lower bound.
- Muligt problem: Hvis en kant i F bliver contracted, ville en knude fra både A og B blive contracted til en superknude.

Analyse af algoritmen ii

- Hvis dette sker, kan (A, B) ikke blive returneret som output.
- Omvendt, hvis $E \notin F$ bliver contracted, er der en mulighed for at (A, B) bliver returneret.
- Vi vil gerne finder et upper bound på sandsynligheden at en kant i F bliver contracted.
- For at kunne finde denne upper bound, skal vi bruge en lower bound på størrelsen af E (antal kanter).
- Antag at en knude v har *degree* (antal kanter) mindre end k (antal kanter mellem A og B).
- Så er cuttet $(\{v\}, V - \{v\})$ af størrelse mindre end k, som er imod vores antagelse at (A, B) er global min-cut. (Fordi den er for lille)

Analyse af algoritmen iii

- Derfor har enhver knude i G en degree af **mindst** k. Dermed $|E| \geq \frac{1}{2}kn$ hvor n er antal knuder. $\frac{1}{2}kn$ kommer fra en kant per knude. Dermed $\frac{1}{2}k$ i stedet for bare kn (i.e., der kan mindst være én kant fra knude til knude.)
- Dermed er sandsynligheden at en kant i F bliver contracted højest
$$\frac{k}{\frac{1}{2}kn} = \frac{2}{n}$$
- Fordi der er k af disse edges, så det er k mulige ud af upper bound $|E| = \frac{1}{2}kn$ (Den her bog gør det meget sværere end det behøver at være)

- Lad os nu kigger på situationen efter j iterationer, når der er $n-j$ superknuder i vores nuværende graf G' og antag at ingen kant i F er blevet contracted endnu.
- Så bliver udregningen

$$\frac{k}{\frac{1}{2}k(n-j)} = \frac{2}{n-j}$$

- Vi ved at cuttet (A, B) vil blive returneret hvis ingen kant i F bliver contrracted i iterationerne $a, 2, \ldots, n-2$.
- Lad ε_j være hændelsen at en kant af F **ikke** bliver contracted i iteration j, så har vi vist at $P(\varepsilon_j) \geq 1 - \frac{2}{n}$ og at $P(\varepsilon_{j+1} | \varepsilon_1 \cap \cdots \cap \varepsilon_j) \geq 1 - \frac{2}{n-j}$.

Analyse af algoritmen v

- Nu vil vi gerne finde en lower-bound på $P(\varepsilon_1 \cap \cdots \cap \varepsilon_{n-2})$.

$$
\begin{aligned}
P(\varepsilon_1) \cdot P(\varepsilon_2|\varepsilon_1) \cdots P(\varepsilon_{j+1}|\varepsilon_1 \cap \varepsilon_2 \cap \cdots \cap \varepsilon_j) \cdots P(\varepsilon_{n-2}|\varepsilon_1 \cap \cdots \cap \varepsilon_{n-2}) \\
\geq \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right) \cdots \left(1 - \frac{2}{n-j}\right) \cdots \left(1 - \frac{2}{3}\right) \\
= \left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\left(\frac{n-4}{n-2}\right) \cdots \left(\frac{2}{4}\right)\left(\frac{1}{3}\right) \\
= \frac{2}{n(n-1)} = \binom{n}{2}^{-1}. \quad \square
\end{aligned}
\tag{1}
$$

Analyse af algoritmen vi

- Vi ved nu at et enkelt *run* af Kargers Algoritme fejler med sandsynligheden $1 - \frac{1}{\binom{n}{2}}$ (dette er et tal der er meget tæt på 1).
- $1 - \frac{1}{\binom{5}{2}} = 90\%$
- $1 - \frac{1}{\binom{50}{2}} = 99.9\%$
- $1 - \frac{1}{\binom{500}{2}} = 99.999\%$
- $1 - \frac{1}{\binom{5000}{2}} = 99.999999\%$
- Vi kan dog få sandsynligheden til at blive lidt bedre ved at køre algoritmen igen og igen og ved brug af tilfældige valg.

Analyse af algoritmen vii

- Vi ved fra kapitel 1 at hvis vi kører algoritmen $\binom{n}{2}$ gange, så er sandsynligheden:

$$\left(1 - \frac{1}{\binom{n}{2}}\right)^{\binom{n}{2}} \leq \frac{1}{e}$$

(13.1)

(a) *The function* $\left(1 - \frac{1}{n}\right)^n$ *converges monotonically from* $\frac{1}{4}$ *up to* $\frac{1}{e}$ *as* n *increases from* 2.

(b) *The function* $\left(1 - \frac{1}{n}\right)^{n-1}$ *converges monotonically from* $\frac{1}{2}$ *down to* $\frac{1}{e}$ *as* n *increases from* 2.

- Vi kan gå endnu lavere. Hvis vi kører algoritmen $\binom{n}{2} \ln n$ gange, så er sandsynligheden for at vi fejler i at finde et global min-cut **højest** $e^{-\ln n} = 1/n$. (Jeg er ikke helt sikker på hvorfor?)

Theorem (13.6)

An undirected graph $G = (V, E)$ on n nodes has at most $\binom{n}{2}$ global min-cuts.

- Lad G være en graf, og lad $C_1, \ldots, C_r$ være alle dets global min-cuts.

Analyse af algoritmen ix

- Lad ε_1 være hændelsen at C_i er returneret af Contraction Algoritmen, og lad $\varepsilon = \bigcup_{i=1}^{r} \varepsilon_i$ være hændelsen at algoritmen returnerer et min-cut.
- Vi ved fra 13.5 at $P(\varepsilon) \geq \frac{1}{\binom{n}{2}}$.
- Faktisk viser beviset at for hvert i har vi $P(\varepsilon_i) \geq \frac{1}{\binom{n}{2}}$.
- Hver par af hændelser ε_i og ε_j er disjunkte, dermed ved vi fra union bound at:

$$P(\varepsilon) = P(\bigcup_{i=1}^{r} \varepsilon_i) = \sum_{i=1}^{r} P(\varepsilon_i) \geq \frac{r}{\binom{n}{2}}$$

- Ud fra dette kan vi så tydeligt se at $r \leq \binom{n}{2}$.

- 10 sider :(
- Givet en undirected multigraf $G = (V, E)$, kan edge-connectivitiy $\lambda(G)$ blive fundet ved brug af $n - 1$ max-flow udregninger.

Definition (Edge-Connectivity)

The edge connectivity, also called the line connectivity, of a graph is the minimum number of edges whose deletion from a graph disconnects. In other words, it is the size of a minimum edge cut. (Kilde: Wolfram MathWorld)

- Vi starter ud med nogle lemmaer og definitioner

Max-back orderings ii

Definition (1 (Max-back Ordering))

A max-back ordering of a given undirected multigraph G is an ordering $v_1, v_2, \ldots, v_n$ of the vertices of G, satisfying the inequality

$$d(V_i, v_{i+1}) \geq d(V_i, v_j)$$

for all indices i and j,$1 \leq i < j \leq n$. The set V_i is defined by $V_i = \{v_1, v_2, \ldots, v_i\}$.

- Ifølge ChatGPT betyder d sandsynligvis antallet af degrees.

Max-back orderings iii

Lemma (1)

A max-back ordering of a given undirected multigraph $G = (V, E)$ can be found in $O(|V| \log |V| + |E'|)$ time, where E' is the set of edges in the corresponding simple graph.