Advanced algorithms
Maximum flow noter

Søren Dahlgaard

June 15, 2011

1 Disposition

1. Flow network.
2. Flows in flow networks - cap constraint, flow conservation.
3. Antiparallel edges / multiple sinks/sources
4. Residual networks
5. Augmenting paths - they are flows in R_f.
6. $(f \uparrow f')$ is a flow in G - don't prove
7. Cuts - duality. $f(S, V \setminus S) = |f|$ for all cuts.
8. $|f| \leq c(S, V \setminus S)$
9. Duality. maxflow $\Rightarrow$ no augment $\Rightarrow$ mincut $\Rightarrow$ maxflow.
10. Bad time
11. Edmonds-Karp. Edge can be critical at most $|V|/2$ times.
12. Maximum bipartite matching. Define and show similarity.

2 Summary

Vi skal nu kigge på et super interessant problem, Maximum Flow. Det er et problem jeg kiggede på første gang i 1. g efter at have mødt følgende problem: `http://www.boi2007.de/tasks/escape.pdf` - Det er nok ingen hemmelighed at løsningen består i at bruge maximum flow. Ved mange andre konkurrencer er vi blevet stillet problemer der kunne løses med max-flow (se f.eks. F - Risk fra dette års NWERC `http://2010.nwerc.eu/results/nwerc2010-problemset.pdf`)

3 Flow netværk

For at kunne finde det maksimale flow er vi nødt til at definere hvad et flow er.Lad $G = (V, E)$ være en graf og lad der være en kapacitetsfunktion $c(u, v) \geq 0$ for alle kanter (u, v). For alle $(u, v) \neq E$ definerer vi $c(u, v) = 0$. Vi har også to særlige knuder i grafen s, t hhv. source og sink. Vi antager også, at alle knuder ligger på en sti mellem s og t. Yderligere kræver vi, at hvis $(u, v) \in E$ så er $(v, u) \notin E$

For hver kant definerer vi yderligere et "flow" $f : V \times V \to \mathbb{R}$. Dette flow skal overholde følgende to krav:

Capacity constraint For alle $u, v \in V$ skal det gælde at $0 \leq f(u, v) \leq c(u, v)$

Flow conservation For alle $u \in V \setminus \{s, t\}$ skal det gælde, at:

$$\sum_{v \in V} f(v, u) = \sum_{v \in V} f(u, v)$$

Vi definerer værdien af et flow, $|f|$ som overskudet der efterlade vores source (se herunder). Alternativt kunne vi definere det som overskudet af flow der løber ind i vores sink.

$$\sum_{v \in V} f(s, v) - \sum_{v \in V} f(v, s)$$

3.1 At transformere diverse tilfælde til flow netværk

Antiparallele kanter Et meget normalt tilfælde er at der er kanter i begge retninger mellem knuder. Dvs. vi har både $(u, v), (v, u) \in E$. I dette tilfælde erstatter vi den ene kant (u, v) med to kanter (u, w) og (w, v). Vi sætter $c(u, w) = c(w, v) = c(u, v)$.

Super source/sink Hvis der er mere end en source eller sink laver vi en ny "super source" hhv. sink. Vi forbinder super source til alle sources og lader kapaciteten mellem disse være ∞.

TODO: add proofs.

4 Ford-Fulkerson method

General algorithm:

```
FORD-FULKERSON(G, s, t)
initialize flow f to 0
while there exists augmenting path from s to t in residual network
      Augment flow along the path
return f
```

4.1 Residual network

For a graph G and flow f we define a residual network G_f, where the capacity of an edge is $c_f(u,v) = c(u,v) - f(u,v)$. We also define an edge in the opposite direction with capacity $c(v,u) = f(u,v)$. Now it is nice that we cannot have both (u,v) and (v,u) in E.

It is clear that $|E_f| = O(|E|)$. Or more exactly $|E_f| \leq 2|E|$.

Other than the fact that G_f can contain both edges (u,v) and (v,u) we can talk about it as a flow network with respect to c_f.

We talk about flow augmentation. Define $f \uparrow f'$ as the augmentation of f by f'. Define it as:

$$(f \uparrow f') = \begin{cases} f(u,v) + f'(u,v) - f'(v,u) & (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$

We claim that $f \uparrow f'$ is a flow in G with $|f \uparrow f'| = |f| + |f'|$.

Proof. First show that the capacity constraint is satisfied:

$$\begin{aligned} (f \uparrow f')(u,v) &= f(u,v) + f'(u,v) - f'(v,u) && (1) \\ &\geq f(u,v) + f'(u,v) - f(u,v) && (2) \\ &= f'(u,v) && (3) \\ &\geq 0 && (4) \end{aligned}$$

and

$$\begin{aligned} (f \uparrow f')(u,v) &= f(u,v) + f'(u,v) - f'(v,u) && (5) \\ &\leq f(u,v) + f'(u,v) && (6) \\ &\leq f(u,v) + c_f(u,v) && (7) \\ &= f(u,v) + c(u,v) - f(u,v) && (8) \\ &= c(u,v) && (9) \end{aligned}$$

Flow conservation:

$$\begin{aligned} \sum_{v \in V} (f \uparrow f')(u,v) &= \sum_{v \in V} (f(u,v) + f'(u,v) - f'(v,u)) && (10) \\ &= \sum_{v \in V} f(u,v) + \sum_{v \in V} f'(u,v) - \sum_{v \in V} f'(v,u) && (11) \\ &= \sum_{v \in V} f(v,u) + \sum_{v \in V} f'(v,u) - \sum_{v \in V} f'(u,v) && (12) \\ &= \sum_{v \in V} (f(v,u) + f'(v,u) - f'(u,v)) && (13) \\ &= \sum_{v \in V} (f \uparrow f')(v,u) && (14) \end{aligned}$$

Third line follows from flow conservation. Ask someone about the last two summations being equal. This shows that it is a flow in G. We now compute the value.

For any vertex $v \in V$ we can have either (s, v) or (v, s) but never both. Define two sets: $V_1 = \{v : (s, v) \in E\}$ and $V_2 = \{v : (v, s) \in E\}$. Clearly $V_1 \cup V_2 \subseteq V$ and $V_1 \cap V_2 = \emptyset$. We have

$$
\begin{aligned}
|f \uparrow f'| &= \sum_{v \in V} (f \uparrow f')(s, v) - \sum_{v \in V} (f \uparrow f')(v, s) && (15)\\
&= \sum_{v \in V_1} (f \uparrow f')(s, v) - \sum_{v \in V_2} (f \uparrow f')(v, s) && (16)\\
&= \sum_{v \in V_1} (f(s, v) + f'(s, v) - f'(v, s)) - \sum_{v \in V_2} (f(v, s) + f'(v, s) - f'(s, v)) && (17)\\
&= \sum_{v \in V_1} f(s, v) + \sum_{v \in V_1} f'(s, v) - \sum_{v \in V_1} f'(v, s) - \sum_{v \in V_2} f(v, s) - \sum_{v \in V_2} f'(v, s) + \sum_{v \in V_2} f'(s, v) && (18)\\
&= \sum_{v \in V_1} f(s, v) - \sum_{v \in V_2} f(v, s) + \sum_{v \in V_1 \cup V_2} f'(s, v) - \sum_{v \in V_1 \cup V_2} f')v, s) && (19)
\end{aligned}
$$

and so on... $\square$

4.2 Augmenting paths

Augmenting path is a simple path, p, from s to t in G_f.

The capacity of an augmenting path is the of a critical edge on p. We can write this as $c_f(p) = \min\{c_f(u, v) : (u, v) \in p\}$.

We can define a flow in G_f from a path p as:

$$
f_p(u, v) = \begin{cases} c_f(p) & \text{if } (u, v) \in p \\ 0 & \text{otherwise} \end{cases}
$$

It is now obvious that for any augmenting path p we have $f \uparrow f_p$ as a flow in G with $|f \uparrow f_p| = |f| + |f_p| > |f|$.

4.3 Cuts of flow networks

A cut is almost the classic definition except we require that $s \in S$ and $t \in V \setminus S = T$. We define both the flow across a cut and the capacity of a cut:

$$
f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u)
$$

$$
c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)
$$

A minimum cut is a cut with minimal capacity of all cuts.

For any cut (S, T) we have $f(S, T) = |f|$

Proof. First we look at the flow definition:

$$|f| = \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) \tag{20}$$

$$= \sum_{v\in V} f(s,v) - \sum_{v\in V} f(v,s) + \sum_{u\in S\setminus\{s\}} \left(\sum_{v\in V} f(u,v) - \sum_{v\in V} f(v,u) \right) \tag{21}$$

See p. 722. Main ideas:

1. Expand the rightmost summation
2. Regroup into summations over $v \in V$ with edges going into v. And another group with edges going out of v.
3. Split all $v \in V$ into $v \in S$ and $v \in T$. Uses $S \cup T = V$.
4. Cancel out terms. and we get what we want.

□

This means that $|f|$ is bounded above by the capacity of ALL cuts. Especially the flow is bounded above by the capacity of a minimum cut.

Max-flow Min-cut *duality* (pawel likes this word) is that the following are equivalent:

1. f is a maximum flow in G
2. There is no augmenting path in G_f
3. $|f| = c(S,T)$ for some cut (S,T) in G.

Proof. $1 \Rightarrow 2$: If there was an augmenting path f_p, then $f \uparrow f_p$ would be a flow in G with bigger value. Contradiction

$2 \Rightarrow 3$: Look at the cut $S = \{v \in V :$ There exists a path from s to v in $G_f\}$. Look at any $u \in S$ and $v \in T$. if $(u,v) \in E$ we must have $f(u,v) = c(u,v)$ (otherwise $v \in S$). Conversely if $(v,u) \in E$ we must have $f(v,u) = 0$ or $c_f(u,v) = f(v,u)$ would mean $v \in S$. Now check that $f(S,T) = c(S,T)$.

$3 \Rightarrow 1$: We cannot have $|f| > c(S,T)$. □

4.4 Analysis

If capacities are irrational we have $T(n) = O(\infty)$. If capacities are rational we can scale them to be integers. If capacities are integers we have $T(n) = O(E|f^*|$. The while loop is executed at most $|f^*|$ times because we always get a strictly better flow. The stuff in the while loop can be done in linear time.

5 Edmonds-Karp

Using a BFS for ford-fulkerson gives $O(VE^2)$ running time.

First step on proving this is to see that the minimum path length (amount of edges) in G_f increases monotonically. Let $\delta_f(s,v)$ be the minimum path length from s to v in G_f.

Proof. Look at the smallest $\delta_{f'}(s,v)$ that changed when augmenting f with f'. Let u be the vertex before v in the path from s to v. We must have $\delta_{f'}(s,u) = \delta_{f'}(s,v) - 1$. We also have $\delta_f(s,u) = \delta_{f'}(s,u)$. Therefore $(u,v) \neq E_f$.

We must therefore have had $\delta_f(s,v) = \delta_f(s,u) - 1$ which leads to contradiction. □

We can use this to show that an edge can at most be critical $|V|/2$ times.

Simply look at $\delta_f(s,v) = \delta_f(s,u) + 1$ when (u,v) is critical. In order for the edge to return to the residual network we must have (v,u) be critical. This can only happen when $\delta_{f'}(s,u) = \delta_{f'}(s,v) + 1$. Since $\delta_{f'}(s,v) + 1 \geq \delta_f(s,v) + 1 = \delta_f(s,u) + 2$ the result follows.

Each augmenting path has at least one critical edge and only $O(VE)$ times can there be a critical edge. This gives the running time.

6 Maximum bipartite matching

A matching is a subset $M \subseteq E$ such that for all vertices $v \in V$ at most one edge in M is incident to v. A maximum matching is a matching M that has $|M| \geq |M'|$ for any other matching M'.

We can create a graph G' with nodes $V' = V \cup \{s,t\}$ and edges

$$E' = \{(s,u) : u \in L\} \cup \{(u,v) : (u,v) \in E\} \cup \{(u,t) : u \in R\}$$

All capacities $c(u,v) = 1$. It is clear that $|E'| = O(|E|)$ because we assume each node $v \in L \cup R$ has one edge incident in E.

We wanna show that a flow f in G' corresponds to a matching M in G. First we say that a flow is integer-valued if $f(u,v)$ is integer for all $(u,v) \in V \times V$. The claim is that if M is a matching in G it corresponds to a flow f in G' with $|f| = |M|$ and the other way around.

Proof. For a matching M create a flow $f(u,v) = f(s,u) = f(u,t) = 1$ for all $(u,v) \in M$. It is easy to see that the constraints are satisfied and that $|f| = |M|$.

For a flow f create a matching

$$M = \{(u,v) : u \in L, v \in R, \text{ and } f(u,v) > 0$$

Because f is integer-valued this is okay. To see that this is a matching use that $(s,u) = 1$ and flow conservation means that the sum over $(u,v) = 1$ (or 0). We can also use this to show that $|M| = |f|$. □

We also need to show that when $c(u,v)$ is integral for all (u,v) the maximum flow found by Ford-Fulkerson will be integer-valued. This is done easily by induction over the iteration. Base case is trivial cause $|f| = 0$.

We can now proof by contradiction that a maximum flow corresponds to a maximum matching.