Maximum Flow Problem

CLRS Chapters 26.1–2

- An historic transportation problem
- Flow networks and maximum flow
- Properties of flow networks
- Linear programming formulation
- Ford-Fulkerson method
- Max-flow min-cut theorem

Martin Zachariasen, DIKU

November 24, 2006

An historic transportation problem

Maximum flow originally developed as a means for studying rail transportation networks (Harris, 1955):

> “Consider a rail network connecting two cities by way of a number of intermediate cities, where each link of the network has a number assigned to it representing its capacity. Assuming a steady state condition, find a maximal flow from one given city to the other.”

The Soviet rail system was studied in a classified report by Harris and Ross from 1955 entitled “Fundamentals of a Method for Evaluating Rail Net Capacities”.

Modeling of the Soviet rail network from the 1950s

Rail network not modeled exactly due to its size and (probably) due to inexact information.

Modeled with nodes as small regions that are connected to neighboring regions.

Connections between neighboring regions assigned a **capacity** which is the tonnage (in 1000 tons) that can be transported between the nodes on a **daily** basis.

US interest in network: Computation of “bottleneck” — what might that be?

Flows in networks

A flow network is a directed graph $G = (V, E)$ where each edge $(u, v) \in E$ has an associated **capacity** $c(u, v) \geq 0$. When $(u, v) \notin E$ then $c(u, v) = 0$.

Given a **source** s and a **sink** t. We assume that $s \leadsto v \leadsto t$ for all $v \in V$.

A **positive flow** is a function $x : V \times V \rightarrow \mathbf{R}$ fulfilling the following constraints:

1. *Capacity constraint:*

   For all $u, v \in V$: $0 \leq x(u, v) \leq c(u, v)$

2. *Flow conservation:*

   For all $u \in V \setminus \{s, t\}$: $\underbrace{\sum_{v \in V} x(v, u)}_{\text{flow into } u} = \underbrace{\sum_{v \in V} x(u, v)}_{\text{flow out of } u}$

Flows in networks (using net flows)

May cancel flow that runs in opposite directions along an edge.

Define **net flow** as: $f(u,v) = x(u,v) - x(v,u)$.

Properties of (net) flow $f \ : \ V \times V \rightarrow \mathbf{R}$:

1. *Capacity constraint:*

   For all $u, v \in V$: $f(u,v) \leq c(u,v)$

2. *Skew symmetry:*

   For all $u, v \in V$: $f(u,v) = -f(v,u)$

3. *Flow conservation:*

   For all $u \in V \setminus \{s,t\}$: $\sum_{v \in V} f(u,v) = 0$

   Equivalently: $\underbrace{\sum_{v \in V : f(v,u) > 0} f(v,u)}_{\text{pos flow into } u} = \underbrace{\sum_{v \in V : f(u,v) > 0} f(u,v)}_{\text{pos flow out of } u}$

Maximum flow problem

Value of flow f is $|f| = \underbrace{\sum_{v \in V} f(s,v)}_{\text{flow out of source}}$

Maximum-flow problem: Given a graph G, source s, sink t, and capacities c, find a flow f with maximum value $|f|$.

Informally: What is the maximum amount of flow that we can push from the source to the sink?

Example: In the Soviet rail network, what is the maximum tonnage that can be transported to the European satellite countries?

Maximum flow problem by linear programming

Straightforward to formulate the maximum flow problem as a linear programming problem:

$$
\begin{array}{lrcll}
\text{maximize} & \sum_{v \in V} f(s,v) & & & \\
\text{subject to} & f(u,v) & \leq & c(u,v), & \forall u, v \in V \\
 & f(u,v) & = & -f(v,u), & \forall u, v \in V \\
 & \sum_{v \in V} f(u,v) & = & 0, & \forall u \in V \setminus \{s,t\}
\end{array}
$$

Note that this formulation is not in standard form: There exist equality constraints in the formulation, and there are no non-negativity constraints.

Multiple sources and sinks

Easy to handle problems with multiple sources $\{s_1, s_2, \ldots, s_m\}$ and sinks $\{t_1, t_2, \ldots, t_n\}$:

- Add a **supersource** s and a **supersink** t to the network.
- Add edges (s, s_i) with capacities $c(s, s_i) = \infty$, $i = 1, \ldots, m$, and edges (t_j, t) with capacities $c(t_j, t) = \infty$, $j = 1, \ldots, n$.

Solving the maximum-flow problem in the extended network gives a maximum flow in the multiple source and sink problem.

Implicit summation notation and related properties

Define flows between pairs of node sets $X, Y \subseteq V$:

$$f(X,Y) = \sum_{x \in X} \sum_{y \in Y} f(x,y)$$

(Example: Flow conservation for u becomes $f(u,V) = 0$.)

A few properties:

1. For all $X \subseteq V: \ f(X,X) = 0$
2. For all $X, Y \subseteq V: \ f(X,Y) = -f(Y,X)$
3. For all $X, Y, Z \subseteq V$ such that $X \cap Y = \emptyset$:

   $f(X \cup Y, Z) = f(X,Z) + f(Y,Z)$
   $f(Z, X \cup Y) = f(Z,X) + f(Z,Y)$
4. $|f| = f(s,V) = f(s, V \setminus s) = f(V \setminus t, t) = f(V,t)$

Ford-Fulkerson method

An **augmenting path** is a path p from s to t along which it is possible to send more flow.

```
FORD-FULKERSON-METHOD($G$, $s$, $t$)
initialize flow $f$ to 0
while there exists an augmenting path $p$
    do augment flow $f$ along $p$
return $f$
```

How can we find an augmenting path for a given flow?

Residual network

The residual network G_f for a given flow f consists of edges where it is possible to send more flow.

Residual capacity of $(u,v) \in V \times V$:

$$c_f(u,v) = c(u,v) - f(u,v)$$

Residual network $G_f = (V, E_f)$:

- Vertex set: V, the same as for the flow network
- Edge set: $E_f = \{(u,v) \in V \times V \mid c_f(u,v) > 0\}$

Observations:

1. May have $(u,v) \in E_f$ even if $c(u,v) = 0$. Corresponds to pushing flow $f(v,u) > 0$ back along (u,v).
2. If $c(u,v) = c(v,u) = 0$ then neither (u,v) nor (v,u) are in E_f.
3. The size of E_f is at most twice the size of E.

Use of residual network

If f' is a flow in G_f then $f + f'$ is a flow in G with flow value $|f + f'| = |f| + |f'|$.

Proof: Need to show that the three flow properties are fulfilled.

1. *Capacity constraint:*
   For all $u, v \in V: \ (f' + f)(u,v) \le c(u,v)$
2. *Skew symmetry:*
   For all $u, v \in V: \ (f + f')(u,v) = -(f + f')(v,u)$
3. *Flow conservation:*
   For all $u \in V \setminus \{s,t\}: \ \sum_{v \in V} (f + f')(u,v) = 0$

Value of flow:

$$\begin{aligned} |f + f'| &= \sum_{v \in V} (f + f')(s,v) \\ &= \sum_{v \in V} (f(s,v) + f'(s,v)) \\ &= \sum_{v \in V} f(s,v) + \sum_{v \in V} f'(s,v) \\ &= |f| + |f'| \end{aligned}$$

Augmenting paths

Simple path p from s to t in G_f.

What is the maximum flow we can push along p in G_f?
This is the **residual capacity** of path p:

$$c_f(p) = \min\{c_f(u,v) \mid (u,v) \text{ is on } p\} > 0$$

Define $f_p : V \times V \to \mathbf{R}$ as follows:

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

Observations:

- f_p is a flow in G_f
- $f + f_p$ is a flow in G with flow value
  $$|f + f_p| = |f| + c_f(p) > |f|$$
  That is: By sending flow along p we *strictly increase* the flow value.

Basic Ford-Fulkerson algorithm

```
FORD-FULKERSON($G$, $s$, $t$)
for each edge $(u,v) \in E[G]$
    do $f[u,v] \leftarrow 0$
       $f[v,u] \leftarrow 0$
while there exists a path $p$ from $s$ to $t$ in $G_f$
    do $c_f(p) \leftarrow \min\{c_f(u,v) \mid (u,v)$ is on $p\}$
       for each edge $(u,v)$ in $p$
           do $f[u,v] \leftarrow f[u,v] + c_f(p)$
              $f[v,u] \leftarrow -f[u,v]$
```

Running time: If all capacities are **integral**, then the running time is $O(E|f^*|)$, where $|f^*|$ is the value of a maximum flow.

Hence the running time is *not* polynomial.

Cuts in a flow network

A cut (S,T) of G is a partition of V into S and $T = V \setminus S$ such that $s \in S$ and $t \in T$.

Flow across cut (S,T) is $f(S,T)$.
Capacity of cut is $c(S,T)$, and we have $f(S,T) \leq c(S,T)$.

Minimum cut: Cut with minimum capacity.

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

Proof:
Step 1. Show that $f(S \setminus s, V) = 0$.
Step 2. $f(S,T) = f(S,V) - f(S,S) = f(S,V)$
$= f(s,V) + f(S \setminus s, V) = |f|$

Max-flow min-cut theorem

If f is a flow in G then the following conditions are equivalent:

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

Proof:

$1 \Rightarrow 2$: If augmenting path p exists then there is a flow with value $|f| + c_f(p) > |f|$.

$2 \Rightarrow 3$: Suppose no augmenting path. Define cut by

$$S = \{v \in V \mid \text{there exists a path } s \rightsquigarrow v \text{ in } G_f\}$$

and $T = V \setminus S$. Must have $t \in T$ so (S,T) is a cut.

For $u \in S$ and $v \in T$ we have $f(u,v) = c(u,v)$, since otherwise $(u,v) \in E_f$ and $v \in S$.

Now we have $|f| = f(S,T) = c(S,T)$.

$3 \Rightarrow 1$: Have that $|f| = f(S,T) \leq c(S,T)$ for all cuts (S,T). Thus $|f| = c(S,T)$ and f is a maximum flow.

Corollary: The Ford-Fulkerson algorithm gives a maximum flow.