Advanced Algorithms: Notes

Author: Martin Grünbaum (martin@itsolveonline.net)

May 10, 2014

Contents

1 Max-flow

A flow network $G = (V, E)$ is a directed graph where each edge $(u, v) \in E$ has a non-negative capacity $c(u, v) \geq 0$. If there is an edge $(u, v) \in E$ then there is no edge $(v, u) \in E$. If $(u, v) \notin E$ then $c(u, v) = 0$ for convenience. When $(u, v) \notin E$, $f(u, v) = 0$.

Flow networks have a source s and a sink t. For each vertex $v \in V$, the flow network contains a path $s \leadsto v \leadsto t$. The graph is therefore connected, meaning $|E| \geq |V| - 1$.

A flow is a real-valued function $f : V \times V \to \mathbb{R}$ that satisfies two properties:

Capacity constraint: For all $u, v \in V$, $0 \leq f(u, v) \leq c(u, v)$

Flow conservation: For all $u \in V - \{s, t\}$, $\sum_{v \in V} f(u, v) = \sum_{v \in V} f(v, u)$.

The value of a flow, $|f|$, is defined as:

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

In the **maximum-flow** problem, we are given a flow network G and we wish to find a maximum flow.

Edges are anti-parallel if there is both an edge (u, v) and an edge (v, u). This is not allowed, and to get around this we instead introduce a new edge x and re-structure the edges as follows: $(u, x), (x, v), (v, u)$. The capacity of the new edges involving x is the same as the capacity from (u, v). See page 711 in the book for an example.

1.1 Multiple sources and sinks

This can be accounted for by introducing a **supersink** and **supersource** with infinite flow and capacity out to all of the sources and from all of the sinks to the supersink. See page 713.

1.2 Ford-Fulkerson

Three basic principles: **residual networks**, **augmenting paths** and **cuts**. Essential for **max-flow min-cut** theorem (Theorem 26.6).

Intuition is as follows: We have a flow network G. We iteratively alter the flow of G, by finding an augmenting path in an associated residual network G_f. Once we know the edges that belong to an augmenting path, we can identify specific edges in G to increase or decrease the flow of. Each iteration increases overall flow, but it may do so by decreasing the flow along certain edges. This is repeated until the residual network G_f has no more augmenting paths.

max-flow min-cut shows that upon termination, this yields a maximum flow.

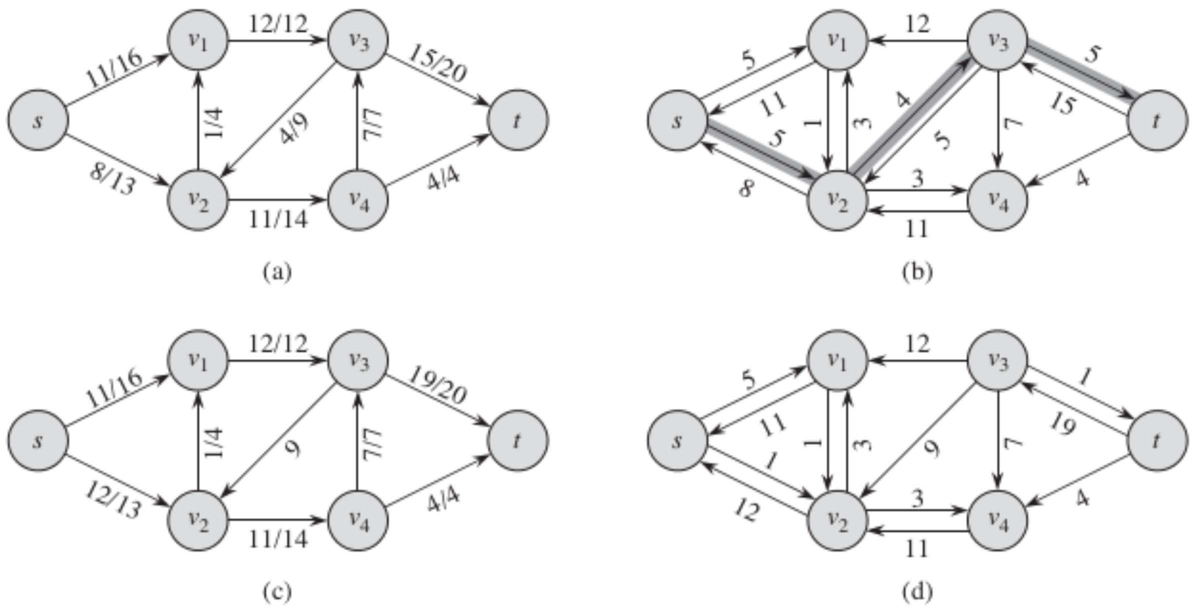
1.2.1 Residual network

Given a network $G = (V, E)$ with a flow f, the **residual network** of G induced by f is $G_f = (V, E_f)$, where

$$E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}.$$

Residual capacity $c_f(u, v)$ is defined by

$$c_f(u, v) = \begin{cases} c(u, v) - f(u, v) & \text{if } (u, v) \in E, \\ f(v, u) & \text{if } (v, u) \in E, \\ 0 & \text{otherwise} \end{cases}$$



Note: that $(u, v) \in E$ implies $(v, u) \notin E$, so there is always only one of the three above cases that applies.

Because the edges in E_f are either edges from E or an edge in the opposite direction, $|E_f| \leq 2|E|$.

Intuition: A residual network G_f consists of edges with capacities that represent how we can alter the flow on edges of G. G can admit an additional amount of flow along an edge, equal to the capacity minus the current flow. If the edge can admit more flow, that edge is placed into G_f with a value of $c_f(u, v) = c(u, v) - f(u, v)$. The residual network may also contain edges that are not in G: In order to represent a possible decrease of a flow $f(u, v)$ on an edge in G, we place an edge (v, u) into G_f with residual capacity $c_f(v, u) = f(u, v)$. In other words, an edge that can admit flow in the opposite direction, at most cancelling out flow entirely. See Figure **??** for an example.

Flows in a residual network satisfy the definition of a flow, but with respect to capacities c_f in the network G_f. If f is a flow in G and f' is a flow in the corresponding residual network G_f, we define $f \uparrow f'$, the **augmentation flow** of f by f', as a function from $V \times V$ to $\mathbb{R}$ defined by

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

Intuition: Increase the flow $(f(u, v))$ by $f'(u, v)$, but decrease it by the flow in the opposite direction $(f'(v, u))$. Pushing flow in the reverse direction is also called **cancellation**.

1.2.2 Augmenting path

An augmenting path p is a simple path from s to t in the residual network G_f. By the definition of a residual network, we may increase the flow of an edge (u, v) by up to $c_f(u, v)$ without violating the capacity constraint on whichever of (u, v) and (v, u) is in the original flow network G.

The maximum amount by which we can increase flow on each edge of an augmenting path p is the **residual capacity** of p, given by $c_f(p) = min\{c_f(u, v) : (u, v) \text{ is on p}\}$. More specifically, if p is an augmenting path in G_f, we define a function $f_p : V \times V \to \mathbb{R}$ as

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

Then f_p is a flow in G_f with value $|f_p| = c_f(p) > 0$. See Lemma 26.2, page 720. It remains to be shown that augmenting f by f_p produces a different flow in G whose value is closer to the maximum. Corollary 26.3 on page 720 shows this by immediate proof, using Lemma 26.1 and 26.2.

1.2.3 Cuts of a network

We know, based on the above, that we can augment flows in G and that doing so can produce a new flow closer to the maximum. But how do we know that when it terminates, the algorithm has in fact found a maximum flow? Max-flow min- cut tells us that a flow is maximum only if its residual network contains no augmenting paths.

A **cut** (S, T) of a flow network $G = (V, E)$ is a partition of V into S and $T = V - S$ such that $s \in S$ and $t \in T$. If f is a flow then the **net flow** $f(S, T)$ across the cut (S, T) is defined to be

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

The **capacity** of the cut (S, T) is

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