CMPSCI 611 Advanced Algorithms **Fall 2001**

Lecture 12: October 17

Lecturer: Micah Adler *Scribe: Jianbin Tan and Joel Sieh*

12.1 Review of Network Flow

Input:

- Directed Graph G
- Capacities of the edges $c(u,v) > 0$ for $(u,v) \in E$; and $c(u,v) = 0$ if $(u,v) \notin E$
- A source node s
- A sink node t

Output: The maximum flow from s to t with the flow function $f : V \times V \to \Re$ and the following properties:

1. Skew-symmetry: $f(u,v) = -f(v,u) \quad \forall u,v \in V$
2. Conservation of Flow: $\sum f(u,v) = 0 \quad \forall u,v \in (V - \{s,t\})$
3. Capacity Constraints: $f(u,v) \leq c(u,v) \quad \forall u,v \in V$

12.1.1 Size of a flow

We define the size of a flow as the Total flow coming out of the source:

$$|f| = \sum_v f(s,v) = \sum_v f(v,t)$$

12.1.2 The s–t cut

An s–t cut (abbreviated cut) of G is a partition of the vertices v into two sets A and B such that $s \in A$ and $t \in B$.

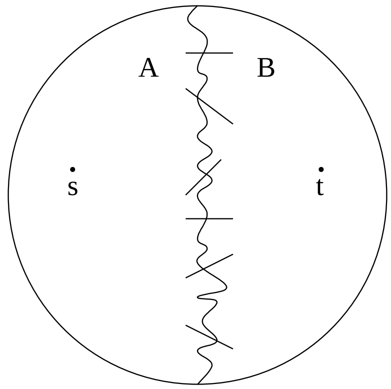

Figure 12.1: An example s–t cut

The capacity of cut (A, B) is defined as the sum of the capacities of all edges across the cut:

$$c(A, B) = \sum_{u \in A, v \in B} c(u, v)$$

The flow across a cut is defined similarly, as the flow across all edges across the cut:

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

The capacity constraints still hold when talking about cuts, and so we can say that $f(A, B) \leq c(A, B)$.

Claim 12.1 *For any flow f and for all cuts (A, B), $f(A, B) = |f|$*

Proof: by induction on the size of A.

Base Case: $|A| = 1$ ($A = \{s\}$)
In this case, the claim follows from the definition of the size of a flow.

Inductive Step: We assume that $f(A, B) = |f|$ for $\forall A$ that $|A| \leq k$. We'll show that it holds when $|A| = k + 1$.

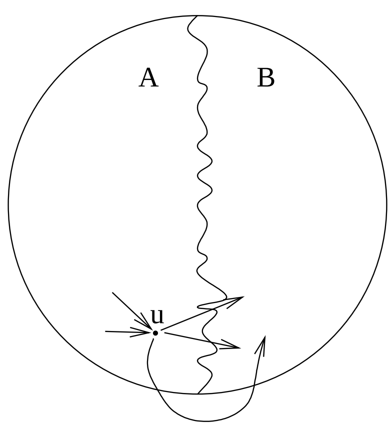

Figure 12.2: Moving u from A to B

Now assume that the size of A is $k+1$. We'll consider the cut$(A - \{u\}, B \cup \{u\})$ for any $u \in A - \{s\}$. We call it cut(A', B'). We know that $f(A', B') = |f|$ by the inductive hypothesis. If we move u over from A to B, it will increase the flow over the cut by the flow over the edges which are from $v(v \in A)$ to u; and it will decrease the flow over the cut by the flow over the edges which are from u to $v(v \in B)$, so:

$$f(A, B) = f(A', B') + \sum_{v \in A} f(v, u) - \sum_{v \in B} f(u, v) = f(A', B') = |f|$$

■

This gives us an upper bound on $|f|$. If we consider all s–t cuts in the graph, the size of the flow through the graph must be less than or equal to the capacity of each of them, including the capacity over the minimum cut.

It turns out that we can always achieve this value.

12.2 The Max Flow - Min Cut Theorem

Theorem 12.2 *For any flow network, there is a flow f and a cut (A, B) such that $|f| = c(A, B)$. Note that f must be a maximum flow (if we increase the flow, we exceed the capacity), and (A, B) must be a minimum cut.*

To prove this, we need to introduce the concept of a residual network.

12.2.1 Residual Networks

Residual networks encapsulate the leftover capacity in the network that the flow in it is not using.

Definition 12.3 *For a network G and a flow f in G, the Residual Network G_f is defined as $G_f = (V, E_f)$ where $E_f = \{(u, v) \mid c(u, v) - f(u, v) > 0\}$. Or, in other words, E_f is the set of edges in which we haven't used all the capacity. Consequently, the capacity of an edge in E_f is the amount of capacity in E that we didn't use: $c_f(u, v) = c(u, v) - f(u, v)$.*

Below are examples of a flow network and it's accompanying residual flow network. In the flow network, the numbers not in parentheses indicate capacities, while the numbers in parentheses indicate the actual flow.

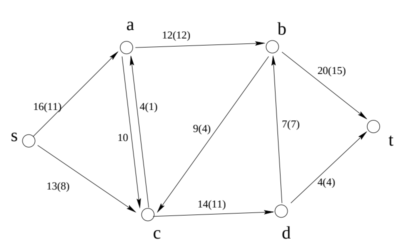

Figure 12.3: Example Flow Network

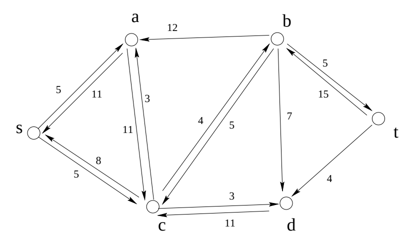

Figure 12.4: Residual Network for the Example Flow Network

In general, the residual capacity for an edge is the original capacity minus the flow (with sign to indicate direction). So the edge from a to c has $10 - (-1) = 11$ capacity and the edge from b to a has $0 - (-12) = 12$ capacity.

It's important to note that there can be more edges in the residual network than there were in the original one because we generally have residual capacities along the opposite direction of flow edges. On the other hand, edges that have been saturated need not appear in the residual network.

It is important to note that $|E_f| \leq 2 \cdot |E|$ because we can at most have a positive capacity in the direction of flow and the opposite direction for each edge.

The point of the residual networks is that we can increase the flow from s to t by finding a path between the two in the residual network.

Definition 12.4 *An augmenting path p for flow f is a path from s to t in graph G_f. This is the path we will use to increase the flow in G.*

Definition 12.5 *A bottleneck capacity $b(p)$ is the minimum capacity in G_f of any edge along that path.*

For example, in our figures above, $s \to c \to b \to t$ is a path p and $b(p) = 4$.

We can always add $b(p)$ units of flow along path p. This will always satisfy our three conditions of maximum flows.

12.2.2 Proof of the Max Flow - Min Cut Theorem

Theorem 12.6 *If f is a flow in a flow network $G = (V, E)$ with source s and sink t, then the following conditions are equivalent:*

1. *f is a maximum flow in G.*
2. *there exists some cut (A, B) of G such that $|f| = c(A, B)$.*
3. *The residual network G_f contains no augmenting paths.*

Proof:
$2 \Longrightarrow 1$: Since $|f| = f(A, B)$, and for all c(A, B), $f(A, B) \leq c(A, B)$, so the f with the size of $c(A, B)$ is a maximum flow.
$1 \Longrightarrow 3$: Otherwise, the flow could be increased by $b(p)$ units along the augmenting path p. This will contradict that f is a maximum flow.

$3 \Longrightarrow 2$: Suppose that G_f has no augmenting path, that is, that G_f contains no path from s to t. Define:
$A = \{v : v \text{ is reachable from } s \text{ in } G_f\}$
$B = V - A$

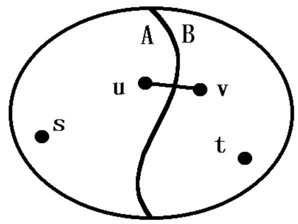

Figure 12.5: Cut (A, B)

Considering cut(A, B): we have $s \in A$ and $t \notin A$ because there is no path from s to t in G_f. For any u, v such that $u \in A$ and $v \in B$, we have $f(u, v) = c(u, v)$, since otherwise $(u, v) \in E_f$ and then v should be in set A. Therefore $c(A, B) = f(A, B) = |f|$.

■

12.3 Ford-Fulkerson Algorithm

12.3.1 Basic Ford-Fulkerson Algorithm

1. $f = 0$
2. **while** there exists an augmenting path p for f, do
3. find p
4. augment f by $b(p)$ units along p
5. **return** f

12.3.2 Analysis of basic Ford-Fulkerson

Integer Capacities: Assume all the capacities of the edges are integers.

- let f^* be the maximum flow, then $|f^*|$ is an integer
- all values in *F.F.* algorithm are integers

Most often in practice, the maximum-flow problem arises with integral capacities. If the capacities are rational numbers, an appropriate scaling transformation can be used to make them all integral. But this algorithm will not work with irrational capacities.

Running Time: With the integral capacities, a straightforward implementation of Ford-Fulkerson runs in time $O(|f^*||E|)$.

- In each iteration, the time to find a path in a residual network is $O(|E|)$

- Since the flow value increases by at least one unit in each iteration, the **while** loop of line 2 is executed at most $|f^*|$ times

Problems: When the capacities are integral and the optimal flow value $|f^*|$ is small, the running time of the Ford-Fulkerson algorithm is good. But when $|f^*|$ is large, the running time can be large too. *Figure* 12.6 and 12.7 shows such an example.

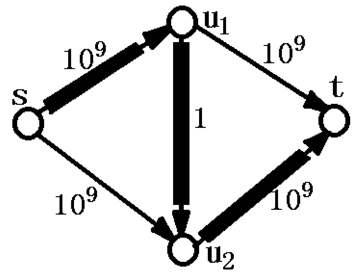

Figure 12.6: an augmenting path with residual capacity 1

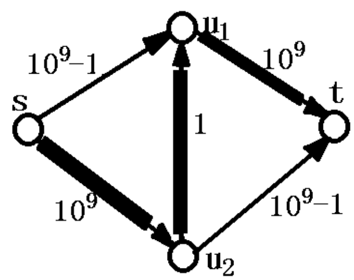

Figure 12.7: the resulting residual network and another augmenting path with residual capacity 1

12.3.3 Ford-Fulkerson Algorithm with Edmonds-Karp Heuristic

1. $f = 0$
2. **while** there exists an augmenting path p for f, do
3. find the shortest augmenting path p
4. augment f by $b(p)$ units along p
5. **return** f

Theorem 12.7 *Edmonds-Karp Heuristic algorithm finds a maximum flow in time* $O(|E|^2|V|)$

Definition 12.8 $\delta_f(s,u)$ *is the unweighted shortest path distance from* s *to* u *in the residual network* G_f*, where shortest path is defined in terms of the number of edges.*

Lemma 12.9 *If the Edmonds-Karp algorithm is run on a flow network* $G = (V, E)$ *with source* s *and sink* t*, then for all vertices* $v \in (V - \{s,t\})$*,* $\delta_f(s,v)$ *is non-decreasing as* f *changes.*

Definition 12.10 *Edge (u, v) is* **critical** *for flow f if (u, v) is on p and $c(u, v)$ in G_f is bottleneck capacity $b(p)$, where p is the augmenting path chosen from G_f.*

To show:

- For any (u, v) between two iterations where (u, v) is critical, $\delta_f(s, u)$ increases by at least 2.
- $|E|$ edges, each is critical at most $|V|/2$

Next time: We will prove lemma 12.9 and theorem 12.7 in next lecture.

References

BL00 MATTHEW BILLMERS and WEI LI, Scribe Notes for Lecture 11, UMASS course 611: Advanced Algorithms, Fall 2000

KL00 SERGEI KOVALENKO and DIMA LISIN, Scribe Notes for Lecture 12, UMASS course 611: Advanced Algorithms, Fall 2000