15-451: Algorithms Sept 5, 2019

Lecture Notes: Introduction to Max Flow

Lecturer: Gary Miller

[1]

1 Basics

A flow network is a directed graph $G = (V, A)$ with source node $s \in V$ and sink node $t \in V$. Each arc has nonnegative capacity $c_a \in \mathbb{R}^+$. For convenience, write $c_a = 0$ if $a \notin A$. And $n := |V|, m := |A|$.

Definition 1.1. Flow: $f : V \times V \to \mathbb{R}$ satisfying following constraints:

1. (capacity) $\forall u, v \in V, f_{uv} \leq c_{uv}$.
2. (conservation) $\forall u \notin \{s, t\}, \sum_{v \in V} f_{uv} = \sum_{v \in V} f_{vu}$.
3. (skew symmetry) $\forall u, v \in V, f_{uv} = -f_{vu}$.

Remark 1.2. Below I write $\sum_v$ instead of $\sum_{v \in V}$ whenever the summation is over all vertices.

- By skew symmetry, the conservation constraint is equivalent to $\sum_v f_{uv} = 0$.
- If f, g are flow, then $f + g$ preserves conservation and nonnegativeness.

Definition 1.3. value of flow $|f| = \sum_v f_{sv}$.

Definition 1.4. Max flow problem: given flow network (G, s, t, c), find flow f which maximizes $|f|$.

Below we shall explain the idea of Ford-Fulkerson algorithm, i.e. residual graph and augmenting path. 'Blindly' augmenting $s - t$ path in the residual graph, however, does not guanrantee a poly-time algorithm. We will remedy this idea in the next lecture to obtain a poly-time algorithm.

2 Residual Graph and Augumenting Path

2.1 A wrong idea

A natural idea is to try a greedy approach: find $s - t$ path in the network and try to push flow along it, until we cannot find such path. Unfortunately this does not yield an optimal solution – consider the toy example in figure 1 from [2]:

[1]Originally 15-750 notes.

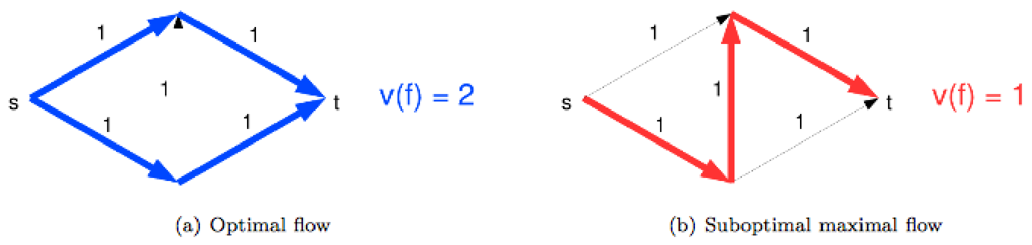

Figure 1: counterexample

The problem here is the condition that there is no $s-t$ path in G is not equivalent to the condition that the flow is optimal. In order to obtain an optimal solution in this example, we need a way to 'push back' some flows, which leads us to consider residual graph defined below.

2.2 A modified idea

Definition 2.1. Given a flow network (G, s, t, c) and a flow f, the *residual capacity* (w.r.t. flow f) is denoted by $c_f(u, v) = c_{uv} - f_{uv}$. And the *residual network* $G_f = (V, A_f)$ where $A_f = \{(u, v) : c_f(u, v) > 0\}$.

Definition 2.2. An f-augmenting path is an $s-t$ path in G_f.

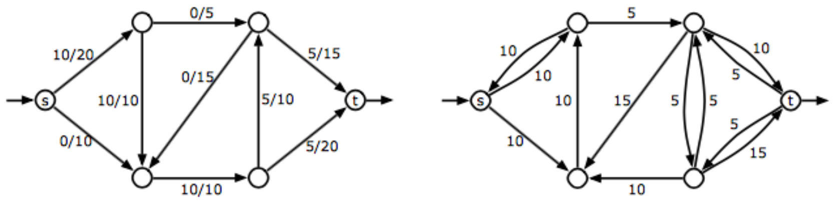

Figure 2: A flow f in a weighted graph G and the corresponding residual graph G_f

Consider figure 2 (chapter 23.3 in [1]). By the definition of residual graph, if $\exists u, v \in V$, such that $f_{uv} > 0$, then there exists a 'back arc' vu in G_f with capacity $c_f(v, u) = 0-(-f_{uv}) = f_{uv}$. This arc vu and its residual capacity captures the notion that we can push back at most $c_f(v, u)$ units of flow. To show that the residual network remedies the above wrong idea, we have the following theorem:

Theorem 2.3. *A feasible flow f is maximum if and only if there exists no f-augmenting path in the residual network.*

Note the 'only if' part is clear from the definition of residual graph. We will prove the reverse direction in the next section. But first let us see the Ford-Fulkerson algorithm that is motivated by the theorem:

Algorithm 1 Ford-Fulkerson, 1956

```
Input: Flow network $(G, s, t, c)$
Output: Maximum flow $f$
    Initialize $f$ to be 0 everywhere.
    while $\exists s - t$ path $P$ in $G_f$ do
        Let $g$ be a flow of value $\min_{a \in A(P)}$ along $P$ and 0 elsewhere
        $f \leftarrow f + g$.
        Update residual graph based on the augmented flow $f$.
    end while
    Output $f$.
```

Note at each step, due to the definition of residual graph, after we augment flow f by flow g, we still maintain a valid flow. Since DFS/BFS finds an augmenting path in $O(m+n)$, we immediately have the following runtime bound:

Theorem 2.4. *For a flow network with INTEGER capcaities and maximum flow value F, Ford-Fulkerson terminates in time $O(F(m+n))$.*

Remark 2.5. The above is not necessarily a poly-time algorithm. Consider figure 3 (chapter 23.4 in [1]), the problem has size $O(\log X)$. If we augment along the arc having capacity only 1, we will have exponential run time.

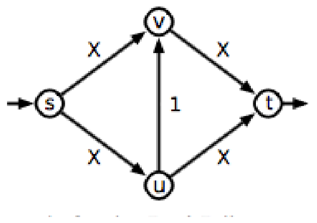

Figure 3: An exponential runtime example for Ford-Fulkerson

Remark 2.6. If all inputs are rational numbers, a similar bound still holds afer rescaling. But if we allow irrational numbers, this algorithm may not terminate, in fact it may not even converge – you can find it in chapter 23.5 of [1].

3 Max-Flow Min-Cut

We will prove theorem 2.3 in the following way: first we will show that any cut (defined below) capacity is an upper bound of any flow value, then we will exhibit a particular cut that is equal to the flow value when there is no augmenting path in the residual graph. Note the duality between flow and cut. The first step corresponds to the weak duality in linear programming.

Definition 3.1. $[S, T]$ is called an $s - t$ cut if

- $S \cap T = \emptyset, S \cup T = V$

- $s \in S, t \in T$

Let cut capacity be defined as: $C[S,T] = \sum_{u \in S, v \in T} c_{uv}$.

Claim 3.2. *Any flow value is upper bounded by any cut capacity.*

We first show a lemma:

Lemma 3.3. *Let f be any $s-t$ flow and $[S,T]$ be any $s-t$ cut. Then $|f| = f^+(S)$, where out-flow $f^+(S) := \sum_{u \in S, v \in T} f_{uv}$.*

Note the above lemma leads to the proof of our claim since $|f| = f^+(S) = \sum_{u \in S, v \in T} f_{uv} \leq \sum_{u \in S, v \in T} c_{uv} = C[S,T]$. Now we prove the above lemma:

Proof. By skew symmetry and conservation we have $\forall u \notin \{s,t\}$, the out-flow at u, $f^+(u)$, is 0. By definition we have $|f| = f^+(s)$. Adding the equality $0 = \sum_{v \in S, v \neq s} f^+(v)$ to it, we have

$$\begin{aligned} |f| &= f^+(s) + \sum_{v \in S, v \neq s} f^+(v) \\ &= \sum_{v \in S} f^+(v) \\ &= \sum_{p,q \in S} f_{pq} + \sum_{p \in S, q \in T} f_{pq} \\ &= \sum_{p \in S, q \in T} f_{pq} \\ &= f^+(S) \end{aligned}$$

where the first, second, third and last equality are from definition, the fourth equality is from skew symmetry applied to f_{pq} where $p, q \in S$. □

Recall our second step is to exhibit a cut with capacity equal to flow value. Then by our above claim, we know both of them are optimal.

Theorem 3.4. *The following statements are equivalent:*

1. *f is a max flow*
2. *G_f has no augmenting path*
3. *$\exists [S,T]$, such that $|f| = C[S,T]$*

Proof. 1 ⇒ 2: if there is an augmenting path, then we can augment flow value by a positive value, contradicting optimality of current flow.

2 ⇒ 3: Let $S := \{v \in V : \text{ v is connected to s in } G_f\}, T = V \backslash S$. We will show $|f| = C[S,T]$. Consider an arbitrary arc $a \in A(G_f), a = pq$ in $[S,T]$. It is clear that $p \in T, q \in S$ since otherwise q is connected to s in G_f. This means that in the original graph G, f_{qp} saturates arc q,p, i.e. $f_{qp} = c_{qp}$.

3 ⇒ 1: by claim 3.2. □

References

[1] Jeff Erickson. Maximum flows and minimum cuts. http://jeffe.cs.illinois.edu/teaching/algorithms/notes/23-maxflow.pdf, 2015. 2.2, 2.5, 2.6

[2] Reza Zadeh. Network flow. https://stanford.edu/~rezab/discrete/Notes/3.pdf, 2017. 2.1