

# Chapter 13

## Network Flow

By Sarel Har-Peled, November 28, 2018<sup>①</sup>

Version: 0.27

### 13.1. Network Flow

We would like to transfer as much “merchandise” as possible from one point to another. For example, we have a wireless network, and one would like to transfer a large file from  $s$  to  $t$ . The network has limited capacity, and one would like to compute the maximum amount of information one can transfer.

Specifically, there is a network and capacities associated with each connection in the network. The question is how much “flow” can you transfer from a source  $s$  into a sink  $t$ . Note, that here we think about the flow as being splittable, so that it can travel from the source to the sink along several parallel paths simultaneously. So, think about our network as being a network of pipe moving water from the source to the sink (the capacities are how much water can a pipe transfer in a given unit of time). On the other hand, in the internet traffic is packet based and splitting is less easy to do.

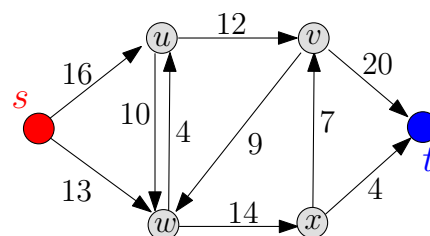
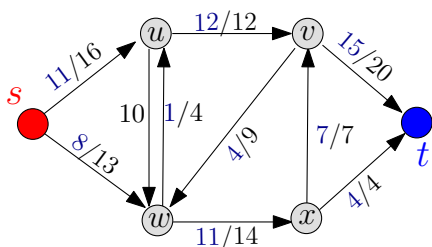


Figure 13.1: A network flow.

**Definition 13.1.1.** Let  $G = (V, E)$  be a *directed* graph. For every edge  $(u, v) \in E(G)$  we have an associated edge *capacity*  $c(u, v)$ , which is a non-negative number. If the edge  $(u, v) \notin G$  then  $c(u, v) = 0$ . In addition, there is a *source* vertex  $s$  and a target *sink* vertex  $t$ .

The entities  $G$ ,  $s$ ,  $t$  and  $c(\cdot)$  together form a *flow network* or simply a *network*. An example of such a flow network is depicted in [Figure 13.1](#).



We would like to transfer as much flow from the source  $s$  to the sink  $t$ . Specifically, all the flow starts from the source vertex, and ends up in the sink. The flow on an edge is a non-negative quantity that can not exceed the capacity constraint for this edge. One possible flow is depicted on the left figure, where the numbers  $a/b$  on an edge denote a flow of  $a$  units on an edge with capacity at most  $b$ .

We next formalize our notation of a flow.

**Definition 13.1.2 (flow).** A *flow* in a network is a function  $f(\cdot, \cdot)$  on the edges of  $G$  such that:

- (A) **Bounded by capacity:** For any edge  $(u, v) \in E$ , we have  $f(u, v) \leq c(u, v)$ .

Specifically, the amount of flow between  $u$  and  $v$  on the edge  $(u, v)$  never exceeds its capacity  $c(u, v)$ .

<sup>①</sup>This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

- (B) **Anti symmetry:** For any  $u, v$  we have  $f(u, v) = -f(v, u)$ .
- (C) There are two special vertices: (i) the **source** vertex  $s$  (all flow starts from the source), and the **sink** vertex  $t$  (all the flow ends in the sink).
- (D) **Conservation of flow:** For any vertex  $u \in V \setminus \{s, t\}$ , we have  $\sum_v f(u, v) = 0$ .<sup>②</sup> Namely, for any internal node, all the flow that flows into a vertex leaves this vertex.

The amount of flow (or simply **flow**) of  $f$ , called the **value** of  $f$ , is  $|f| = \sum_{v \in V} f(s, v)$ .

Note, that a flow on an edge can be negative (i.e., there is a positive flow flowing on this edge in the other direction).

**Problem 13.1.3 (Maximum flow).** Given a network  $G$  find the **maximum flow** in  $G$ . Namely, compute a legal flow  $f$  such that  $|f|$  is maximized.

## 13.2. Some properties of flows and residual networks

For two sets  $X, Y \subseteq V$ , let  $f(X, Y) = \sum_{x \in X, y \in Y} f(x, y)$ . We will slightly abuse the notations and refer to  $f(\{v\}, S)$  by  $f(v, S)$ , where  $v \in V(G)$ .

**Observation 13.2.1.** *By definition, we have  $|f| = f(s, V)$ .*

**Lemma 13.2.2.** *For a flow  $f$ , the following properties holds:*

- (i)  $\forall u \in V(G)$  we have  $f(u, u) = 0$ ,
- (ii)  $\forall X \subseteq V$  we have  $f(X, X) = 0$ ,
- (iii)  $\forall X, Y \subseteq V$  we have  $f(X, Y) = -f(Y, X)$ ,
- (iv)  $\forall X, Y, Z \subseteq V$  such that  $X \cap Y = \emptyset$  we have that  $f(X \cup Y, Z) = f(X, Z) + f(Y, Z)$  and  $f(Z, X \cup Y) = f(Z, X) + f(Z, Y)$ .
- (v) For all  $u \in V \setminus \{s, t\}$ , we have  $f(u, V) = f(V, u) = 0$ .

*Proof:* Property (i) holds since  $(u, u)$  it not an edge in the graph, and as such its flow is zero. As for property (ii), we have

$$f(X, X) = \sum_{\{u, v\} \subseteq X, u \neq v} (f(u, v) + f(v, u)) + \sum_{u \in X} f(u, u) = \sum_{\{u, v\} \subseteq X, u \neq v} (f(u, v) - f(u, v)) + \sum_{u \in X} 0 = 0,$$

by the anti-symmetry property of flow (**Definition 13.1.2 (B)**).

Property (iii) holds immediately by the anti-symmetry of flow, as

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

(iv) This case follows immediately from definition.

Finally (v) is a restatement of the conservation of flow property. ■

**Claim 13.2.3.**  $|f| = f(V, t)$ .

---

<sup>②</sup>This law for electric circuits is known as Kirchoff's Current Law.



Figure 13.2: (i) A flow network, and (ii) the resulting residual network. Note, that  $f(u, w) = -f(w, u) = -1$  and as such  $c_f(u, w) = 10 - (-1) = 11$ .

*Proof:* We have:

$$\begin{aligned}
 |f| &= f(s, V) = f(V \setminus (V \setminus \{s\}), V) \\
 &= f(V, V) - f(V \setminus \{s\}, V) \\
 &= -f(V \setminus \{s\}, V) = f(V, V \setminus \{s\}) \\
 &= f(V, t) + f(V, V \setminus \{s, t\}) \\
 &= f(V, t) + \sum_{u \in V \setminus \{s, t\}} f(V, u) \\
 &= f(V, t) + \sum_{u \in V \setminus \{s, t\}} 0 \\
 &= f(V, t),
 \end{aligned}$$

since  $f(V, V) = 0$  by Lemma 13.2.2 (i) and  $f(V, u) = 0$  by Lemma 13.2.2 (iv). ■

Definition 13.2.4. Given capacity  $c$  and flow  $f$ , the **residual capacity** of an edge  $(u, v)$  is

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

Intuitively, the residual capacity  $c_f(u, v)$  on an edge  $(u, v)$  is the amount of unused capacity on  $(u, v)$ . We can next construct a graph with all edges that are not being fully used by  $f$ , and as such can serve to improve  $f$ .

Definition 13.2.5. Given  $f$ ,  $G = (V, E)$  and  $c$ , as above, the **residual graph** (or **residual network**) of  $G$  and  $f$  is the graph  $G_f = (V, E_f)$  where

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

Note, that by the definition of  $E_f$ , it might be that an edge  $(u, v)$  that appears in  $E$  might induce two edges in  $E_f$ . Indeed, consider an edge  $(u, v)$  such that  $f(u, v) < c(u, v)$  and  $(v, u)$  is not an edge of  $G$ . Clearly,  $c_f(u, v) = c(u, v) - f(u, v) > 0$  and  $(u, v) \in E_f$ . Also,

$$c_f(v, u) = c(v, u) - f(v, u) = 0 - (-f(u, v)) = f(u, v),$$

since  $c(v, u) = 0$  as  $(v, u)$  is not an edge of  $G$ . As such,  $(v, u) \in E_f$ . This states that we can always reduce the flow on the edge  $(u, v)$  and this is interpreted as pushing flow on the edge  $(v, u)$ . See Figure 13.2 for an example of a residual network.

Since every edge of  $G$  induces at most two edges in  $G_f$ , it follows that  $G_f$  has at most twice the number of edges of  $G$ ; formally,  $|E_f| \leq 2|E|$ .

**Lemma 13.2.6.** Given a flow  $f$  defined over a network  $G$ , then the residual network  $G_f$  together with  $c_f$  form a flow network.

*Proof:* One need to verify that  $c_f(\cdot)$  is always a non-negative function, which is true by the definition of  $E_f$ . ■

The following lemma testifies that we can improve a flow  $f$  on  $G$  by finding a any legal flow  $h$  in the residual network  $G_f$ .

**Lemma 13.2.7.** Given a flow network  $G = (V, E)$ , a flow  $f$  in  $G$ , and a flow  $h$  in  $G_f$ , where  $G_f$  is the residual network of  $f$ . Then  $f + h$  is a (legal) flow in  $G$  and its capacity is  $|f + h| = |f| + |h|$ .

*Proof:* By definition, we have  $(f + h)(u, v) = f(u, v) + h(u, v)$  and thus  $(f + h)(X, Y) = f(X, Y) + h(X, Y)$ . We need to verify that  $f + h$  is a legal flow, by verifying the properties required to it by **Definition 13.1.2**.

Anti symmetry holds since  $(f + h)(u, v) = f(u, v) + h(u, v) = -f(v, u) - h(v, u) = -(f + h)(v, u)$ .

Next, we verify that the flow  $f + h$  is bounded by capacity. Indeed,

$$(f + h)(u, v) \leq f(u, v) + h(u, v) \leq f(u, v) + c_f(u, v) = f(u, v) + (c(u, v) - f(u, v)) = c(u, v).$$

For  $u \in V - s - t$  we have  $(f + h)(u, V) = f(u, V) + h(u, V) = 0 + 0 = 0$  and as such  $f + h$  comply with the conservation of flow requirement.

Finally, the total flow is

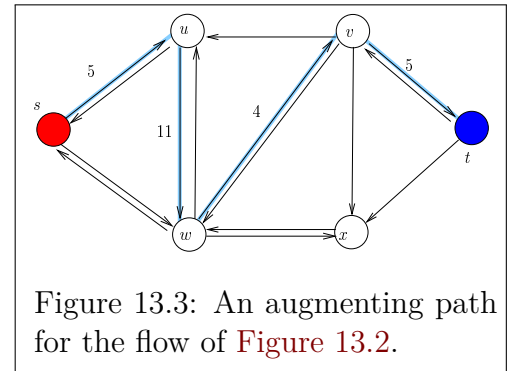
$$|f + h| = (f + h)(s, V) = f(s, V) + h(s, V) = |f| + |h|. \quad \blacksquare$$

**Definition 13.2.8.** For  $G$  and a flow  $f$ , a path  $\pi$  in  $G_f$  between  $s$  and  $t$  is an **augmenting path**.

Note, that all the edges of  $\pi$  has positive capacity in  $G_f$ , since otherwise (by definition) they would not appear in  $E_f$ . As such, given a flow  $f$  and an augmenting path  $\pi$ , we can improve  $f$  by pushing a positive amount of flow along the augmenting path  $\pi$ . An augmenting path is depicted on the right, for the network flow of **Figure 13.2**.

**Definition 13.2.9.** For an augmenting path  $\pi$  let  $c_f(\pi)$  be the maximum amount of flow we can push through  $\pi$ . We call  $c_f(\pi)$  the **residual capacity** of  $\pi$ . Formally,

$$c_f(\pi) = \min_{(u,v) \in \pi} c_f(u, v).$$



We can now define a flow that realizes the flow along  $\pi$ . Indeed:

$$f_\pi(u, v) = \begin{cases} c_f(\pi) & \text{if } (u, v) \text{ is in } \pi \\ -c_f(\pi) & \text{if } (v, u) \text{ is in } \pi \\ 0 & \text{otherwise.} \end{cases}$$

**Lemma 13.2.10.** For an augmenting path  $\pi$ , the flow  $f_\pi$  is a flow in  $G_f$  and  $|f_\pi| = c_f(\pi) > 0$ .

We can now use such a path to get a larger flow.

**Lemma 13.2.11.** Let  $f$  be a flow, and let  $\pi$  be an augmenting path for  $f$ . Then  $f + f_\pi$  is a “better” flow. Namely,  $|f + f_\pi| = |f| + |f_\pi| > |f|$ .

Namely,  $f + f_\pi$  is flow with larger value than  $f$ . Consider the flow in Figure 13.4.

Can we continue improving it? Well, if you inspect the residual network of this flow, depicted on the right. Observe that  $s$  is disconnected from  $t$  in this residual network. So, we are unable to push any more flow. Namely, we found a solution which is a local maximum solution for network flow. But is that a global maximum? Is this the maximum flow we are looking for?

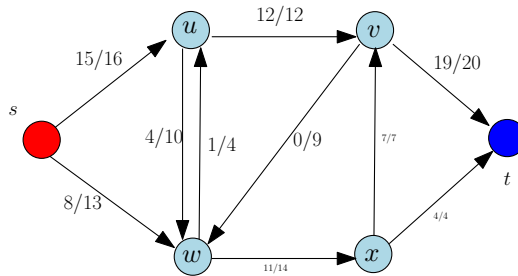
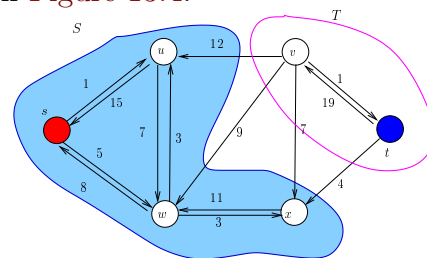


Figure 13.4: The flow resulting from applying the residual flow  $f_p$  of the path  $p$  of Figure 13.3 to the flow of Figure 13.2.



### 13.3. The Ford-Fulkerson method

```

mtdFordFulkerson(G, c)
begin
  f ← Zero flow on G
  while (G_f has augmenting
         path p) do
    (* Recompute G_f for
       this check *)
    f ← f + f_p
  return f
end

```

Given a network  $G$  with capacity constraints  $c$ , the above discussion suggest a simple and natural method to compute a maximum flow. This is known as the **Ford-Fulkerson** method for computing maximum flow, and is depicted on the left, we will refer to it as the **mtdFordFulkerson** method. It is unclear that this method (and the reason we do not refer to it as an algorithm) terminates and reaches the global maximum flow. We address these problems shortly.

### 13.4. On maximum flows

We need several natural concepts.

**Definition 13.4.1.** A **directed cut**  $(S, T)$  in a flow network  $G = (V, E)$  is a partition of  $V$  into  $S$  and  $T = V \setminus S$ , such that  $s \in S$  and  $t \in T$ . We usually will refer to a directed cut as being a **cut**.

The net **flow of  $f$  across a cut**  $(S, T)$  is  $f(S, T) = \sum_{s \in S, t \in T} f(s, t)$ .

The **capacity** of  $(S, T)$  is  $c(S, T) = \sum_{s \in S, t \in T} c(s, t)$ .

The **minimum cut** is the cut in  $G$  with the minimum capacity.

**Lemma 13.4.2.** *Let  $G, f, s, t$  be as above, and let  $(S, T)$  be a cut of  $G$ . Then  $f(S, T) = |f|$ .*

*Proof:* We have

$$f(S, T) = f(S, V) - f(S, S) = f(S, V) = f(s, V) + f(S - s, V) = f(s, V) = |f|,$$

since  $T = V \setminus S$ , and  $f(S - s, V) = \sum_{u \in S - s} f(u, V) = 0$  by **Lemma 13.2.2 (v)** (note that  $u$  can not be  $t$  as  $t \in T$ ). ■

**Claim 13.4.3.** *The flow in a network is upper bounded by the capacity of any cut  $(S, T)$  in  $G$ .*

*Proof:* Consider a cut  $(S, T)$ . We have  $|f| = f(S, T) = \sum_{u \in S, v \in T} f(u, v) \leq \sum_{u \in S, v \in T} c(u, v) = c(S, T)$ . ■

In particular, the maximum flow is bounded by the capacity of the minimum cut. Surprisingly, the maximum flow is exactly the value of the minimum cut.

**Theorem 13.4.4 (Max-flow min-cut theorem).** *If  $f$  is a flow in a flow network  $G = (V, E)$  with source  $s$  and sink  $t$ , then the following conditions are equivalent:*

(A)  *$f$  is a maximum flow in  $G$ .*

(B) *The residual network  $G_f$  contains no augmenting paths.*

(C)  *$|f| = c(S, T)$  for some cut  $(S, T)$  of  $G$ . And  $(S, T)$  is a minimum cut in  $G$ .*

*Proof:* (A)  $\Rightarrow$  (B): By contradiction. If there was an augmenting path  $p$  then  $c_f(p) > 0$ , and we can generate a new flow  $f + f_p$ , such that  $|f + f_p| = |f| + c_f(p) > |f|$ . A contradiction as  $f$  is a maximum flow.

(B)  $\Rightarrow$  (C): Well, it must be that  $s$  and  $t$  are disconnected in  $G_f$ . Let

$$S = \left\{ v \mid \text{Exists a path between } s \text{ and } v \text{ in } G_f \right\}$$

and  $T = V \setminus S$ . We have that  $s \in S$ ,  $t \in T$ , and for any  $u \in S$  and  $v \in T$  we have  $f(u, v) = c(u, v)$ . Indeed, if there were  $u \in S$  and  $v \in T$  such that  $f(u, v) < c(u, v)$  then  $(u, v) \in E_f$ , and  $v$  would be reachable from  $s$  in  $G_f$ , contradicting the construction of  $T$ .

This implies that  $|f| = f(S, T) = c(S, T)$ . The cut  $(S, T)$  must be a minimum cut, because otherwise there would be cut  $(S', T')$  with smaller capacity  $c(S', T') < c(S, T) = f(S, T) = |f|$ . On the other hand, by **Lemma 13.4.3**, we have  $|f| = f(S', T') \leq c(S', T')$ . A contradiction.

(C)  $\Rightarrow$  (A) Well, for any cut  $(U, V)$ , we know that  $|f| \leq c(U, V)$ . This implies that if  $|f| = c(S, T)$  then the flow can not be any larger, and it is thus a maximum flow. ■

The above max-flow min-cut theorem implies that if **mtdFordFulkerson** terminates, then it had computed the maximum flow. What is still allusive is showing that the **mtdFordFulkerson** method always terminates. This turns out to be correct only if we are careful about the way we pick the augmenting path.

## Bibliography