**Network Flow**

**Flow networks and flows**

A *flow network* G=(V, E) is a directed graph in which each edge (u, v) Є E has a nonnegative *capacity* c(u, v) ≥0. We further require that if E contains an edge (u, v), then there is no edge (v, u) in the reverse direction. We distinguish two vertices in a flow network: a *source* s and a *sink* t. The graph is therefore connected and, since each vertex other than s has at least one entering edge, |E| ≥ |V| - 1. A *flow* in G is satisfies the following two properties:

Capacity constraint: For all (u, v) Є V, we require 0 ≤ f (u, v) ≤ c(u, v).

Flow conservation: For all u Є V -{s, t}, we require

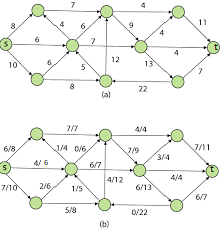
ΣvЄV f(v, u) = ΣvЄV f(u, v)

When(u, v)Ɇ E, there can be no flow from u to v, and f(u, v) = 0.

The *value* |f| of a flow f is defined as

|f| = ΣvЄV f(s, v) - ΣvЄV f(v, s)

that is, the total flow out of the source minus the flow into the source. Typically, a flow network will not have any edges into the source, and the flow into the source, given by the summation ΣvЄV f(v, s) will be 0.



**Ford Fulkerson Method**

The Ford-Fulkerson method depends on three important ideas that transcend the method and are relevant to

many flow algorithms and problems: residual networks, augmenting paths, and cuts.

The Ford-Fulkerson method iteratively increases the value of the flow. We start with f(u, v) = 0 for all (u, v) Є V, giving an initial flow of value 0. At each iteration, we increase the flow value in G by finding an “augmenting path” in an associated “residual network” Gf. Once we know the edges of an augmenting path in Gf, we can easily identify specific edges in G for which we can change the flow so that we increase the value of the flow.

FORD-FULKERSON-METHOD(G, s, t)

1 initialize flow f to 0

2 while there exists an augmenting path p in the residual network Gf

3 augment flow f along p

4 return f

In order to implement and analyze the Ford-Fulkerson method, we need to introduce several additional concepts.

**Residual networks**

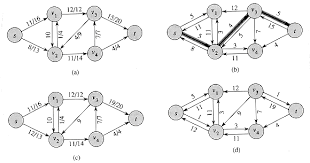
Intuitively, given a flow network G and a flow f, the residual network Gf consists of edges with capacities that represent how we can change the flow on edges of G. An edge of the flow network can admit an amount of additional flow equal to the edge’s capacity minus the flow on that edge. If that value is positive, we place that edge into Gf with a “residual capacity” of cf(u, v) = c(u, v) – f(u, v). The only edges of G that are in Gf are those that can admit more flow; those edges (u, v) whose flow equals their capacity have cf(u, v) = 0, and they are not in Gf.

cf (u, v) = c(u, v) – f(u, v) if (u, v) Є E

= f(v, u) if (v, u) Є E

= 0 otherwise

Given a flow network G = (V, E) and a flow f, the *residual network* of G induced by f is Gf = (V, Ef) where Ef ≤ 2|E|



**Augmenting paths**

Given a flow network G = (V, E) and a flow f, an *augmenting path* p is a simple path from s to t in the residual network Gf . By the definition of the residual network, we may increase the flow on an edge (u, v) of an augmenting path by up to cf(u, v) without violating the capacity constraint on whichever of (u, v) and (v, u) is in the original flow network G. i.e in the augmenting path(p) which edge contain smallest capacity that will be the residual capacity of p. Flow value of this path will be updated by residual capacity.

cf(p) = min{cf(u, v) where (u, v) is on p}

**Cuts of flow networks**

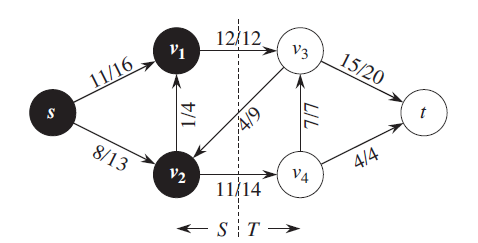
The Ford-Fulkerson method repeatedly augments the flow along augmenting paths until it has found a maximum flow. How do we know that when the algorithm terminates, we have actually found a maximum flow? The max-flow min-cut theorem, which we shall prove shortly, tells us that a flow is maximum if and only if its residual network contains no augmenting path. To prove this theorem, though, we must first explore the notion of a cut of a flow network.

A *cut (*S, T) of flow network G = (V, E) is a partition of V into S and T = V - S such that s Є S and t Є 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) = ΣuЄSΣvЄT f(u, v) - ΣuЄSΣvЄT f(v, u)

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

c(S, T) = ΣuЄSΣvЄT c(u, v)



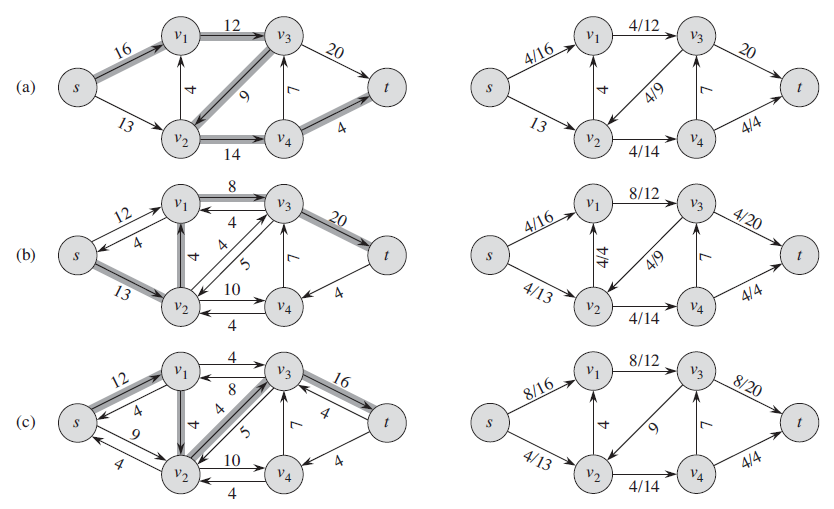
A cut (S, T) in the flow network of above Figure, where S = {s, v1, v2} and T = {v3, v4, t}. The vertices in S are black, and the vertices in T are white. The net flow across this cut is

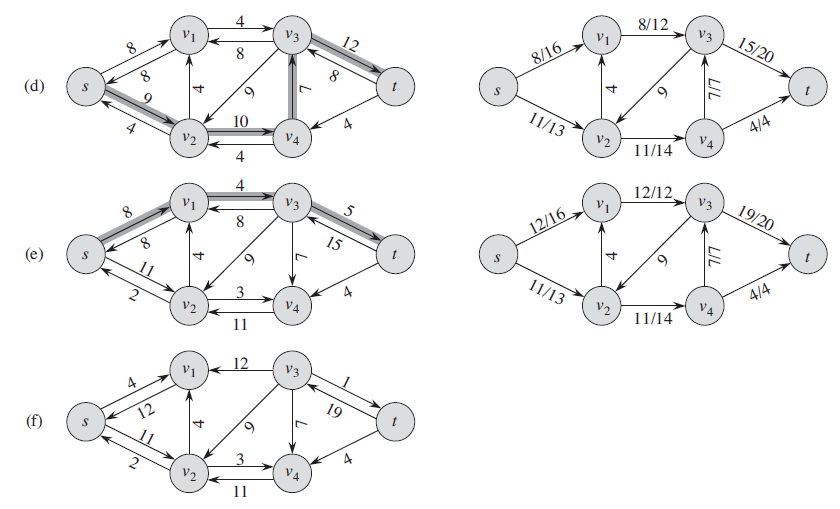
f(v1, v3) + f(v2, v4) – f(v3, v2) = 12 + 11 – 4 = 19

and the capacity of this cut is

c(v1, v3) + c(v2, v4) = 12 + 14 = 26

**Analysis of Ford-Fulkerson Algorithm**





**Ford-Fulkerson algorithm**

FORD-FULKERSON(G, s, t)

1 for each edge (u, v) Є G.*E*

2 (u, v).*f =* 0

3 while there exists a path p from s to t in the residual network Gf

4 cf(p) = min {cf (u, v): (u, v) is in p}

5 for each edge (u, v) in p

6 if (u, v) Є E

7 (u, v).*f =*  (u, v).*f +*  cf(p)

8 else (v, u).*f =*  (v, u).*f -*  cf(p)