Graph Theory
[6]

Maximum flows

Ford-Fulkerson method

Edmonds and Karp's algorithm

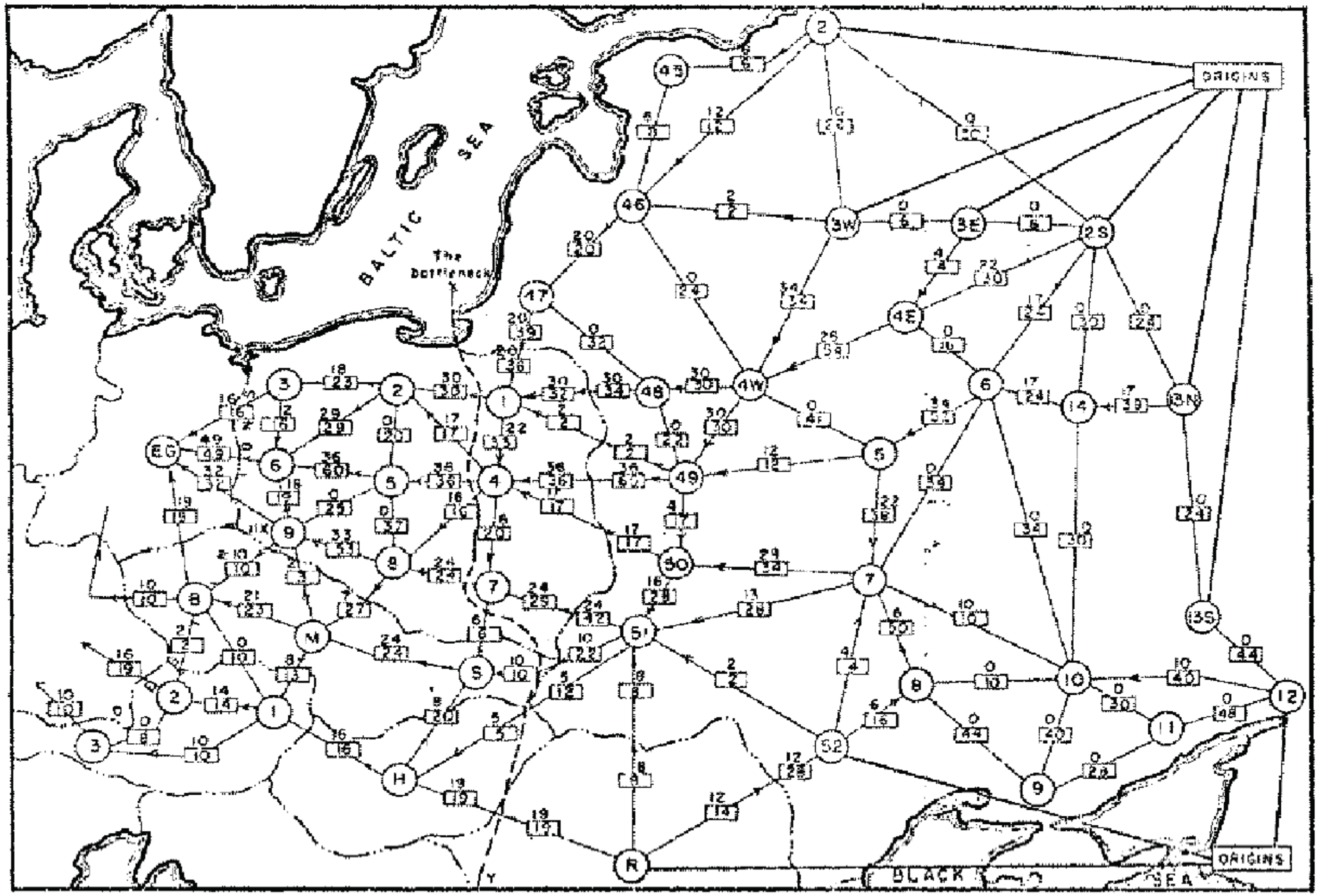
Documents are here:

https://www-l2ti.univ-paris13.fr/~viennet/ens/2024-USTH-Graphs

Emmanuel Viennet

emmanuel.viennet@univ-paris13.fr

Soviet Rail Network, 1955

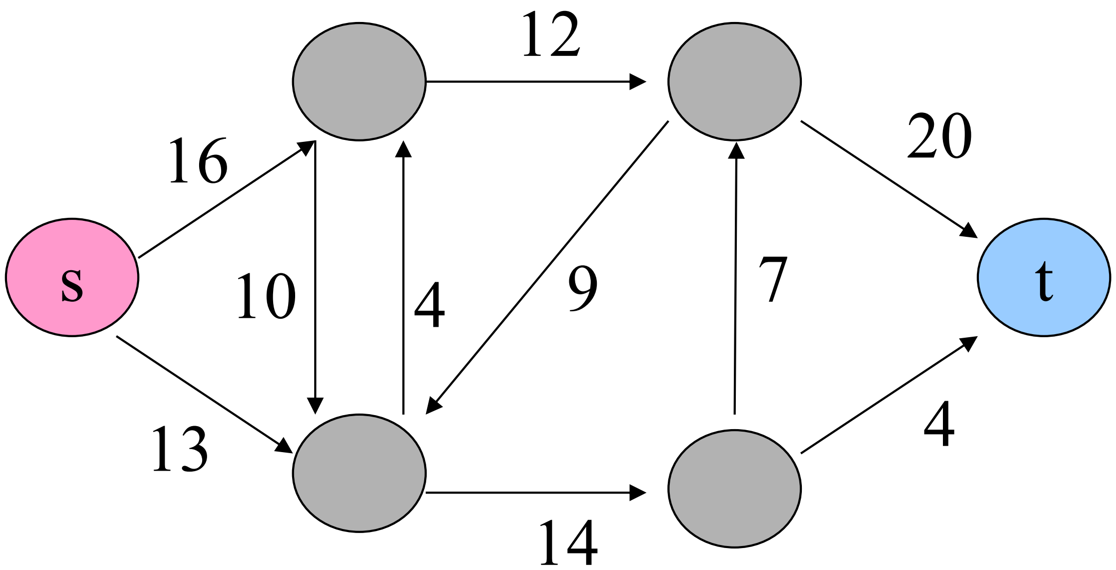


Reference: *On the history of the transportation and maximum flow problems.*
Alexander Schrijver in Math Programming, 91: 3, 2002.

•material coursing through a system from a source to a sink

Flow networks:

- A flow network G=(V,E): a directed graph, where each edge $(u,v)\in E$ has a nonnegative capacity $c(u,v)>=0$.
- If $(u,v)\notin E$, we assume that $c(u,v)=0$.
- two distinct vertices :**a source s and a sink t.**

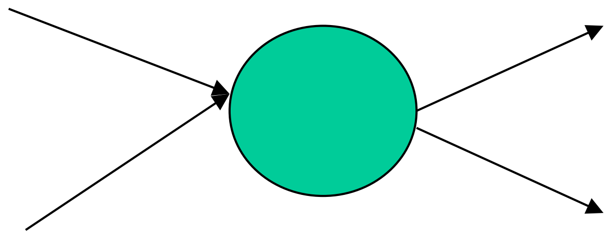


Flow:

- G=(V,E): a flow network with capacity function c.
- s-- the source and t-- the sink.
- A flow in G: a real-valued function f:V*V → R satisfying the following three properties:
- Capacity constraint: For all u,v $\in$V,

  we require $f(u,v) \leq c(u,v)$.
- Flow conservation: For all u $\in$V-{s,t}, we require

$$\sum_{e.in.v} f(e) = \sum_{e.out.v} f(e)$$

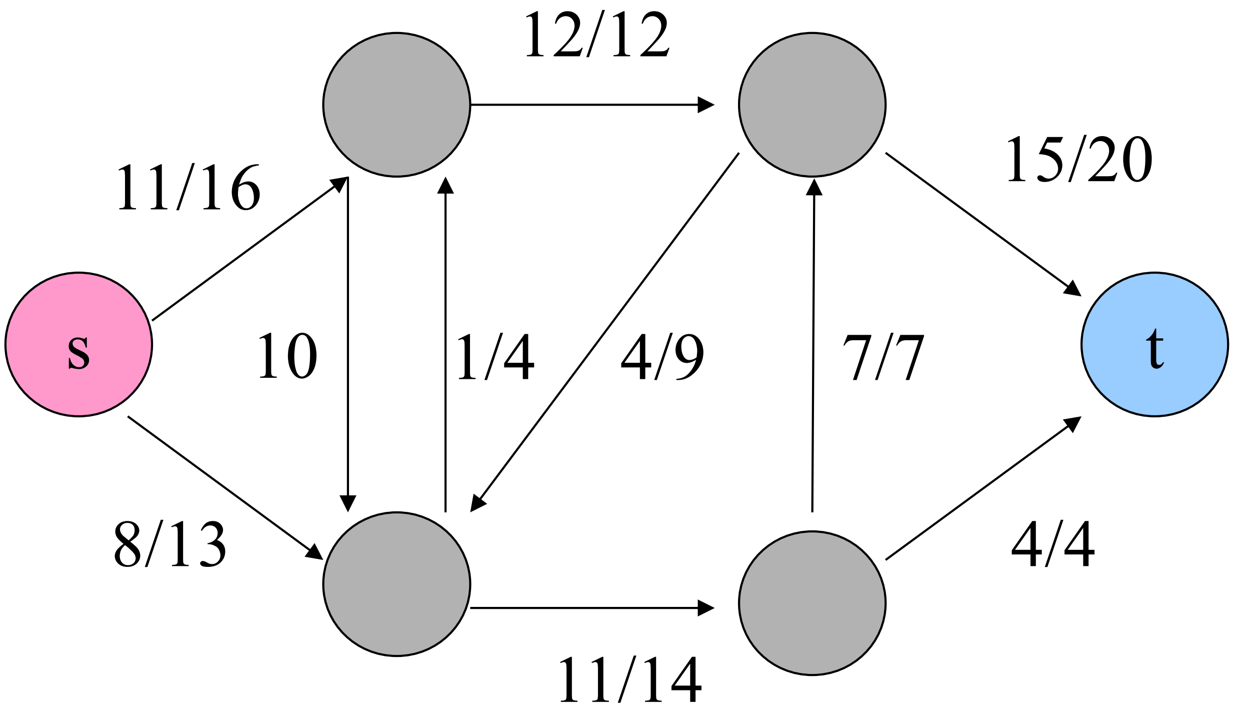


Net flow and value of a flow f:

- The quantity f (u,v), which can be positive or negative, is called the net flow from vertex u to vertex v.
- The value of a flow is defined as

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

  - The total flow from source to any other vertices.
  - The same as the total flow from any vertices to **the sink.**



A flow f in G with value $\left|f\right| = 19$.

Maximum-flow problem:

- Given a flow network G with source s and sink t
- Find a flow of maximum value from s to t.

- How to solve it efficiently?

The Ford-Fulkerson method

This section presents the Ford-Fulkerson method for solving the maximum-flow problem. We call it a “method” rather than an “algorithm” because it encompasses several implementations with different running times.
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.
These ideas are essential to the important max-flow min-cut theorem, which characterizes the value of maximum flow in terms of cuts of the flow network.

The Ford-Fulkerson method

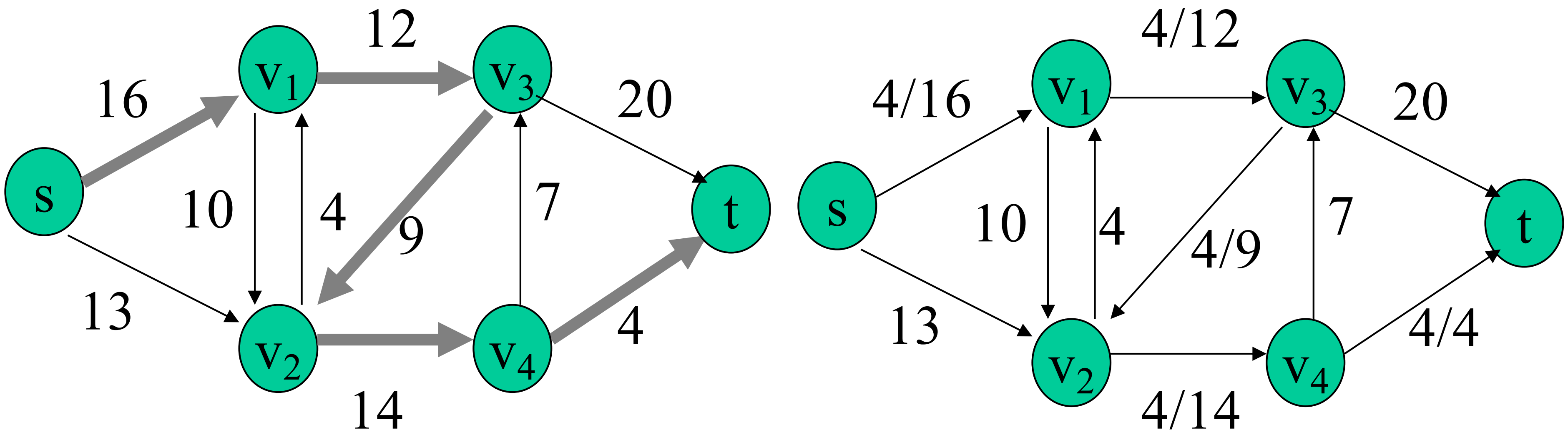
Given a graph G and two nodes (s, t)

- initialize flow *f* to *0*
- while there exists an *augmenting* path *p*
- do *augment* flow *f* along *p*
- return *f*

Residual networks

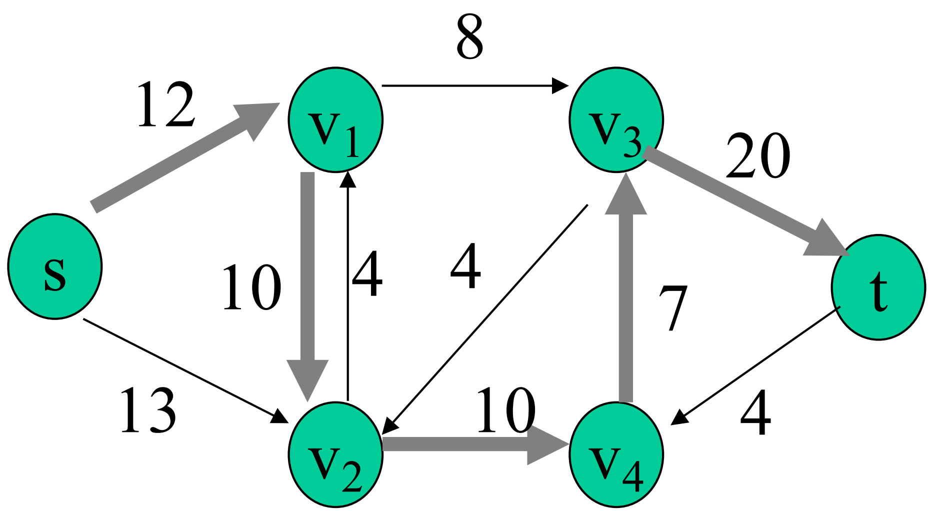
- Given a flow network and a flow, the **residual network** consists of edges that can admit more net flow.
- G=(V, E) a flow network with source s and sink t
- f: a flow in G.
- The amount of additional net flow from u to v before exceeding the capacity c(u,v) is the residual capacity of (u,v), given by: $c_f(u,v) = c(u,v) - f(u,v)$

Example of residual network



(a)

Example of Residual network (continued)



(b)

Fact 1

- Let G=(V,E) be a flow network with source s and sink t, and let f be a flow in G
- Let G_f be the residual network of G induced by f, and let f' be a flow in G_f
  Then, the flow sum f+f' is a flow in G with value

$$|f + f'| = |f| + |f'|$$

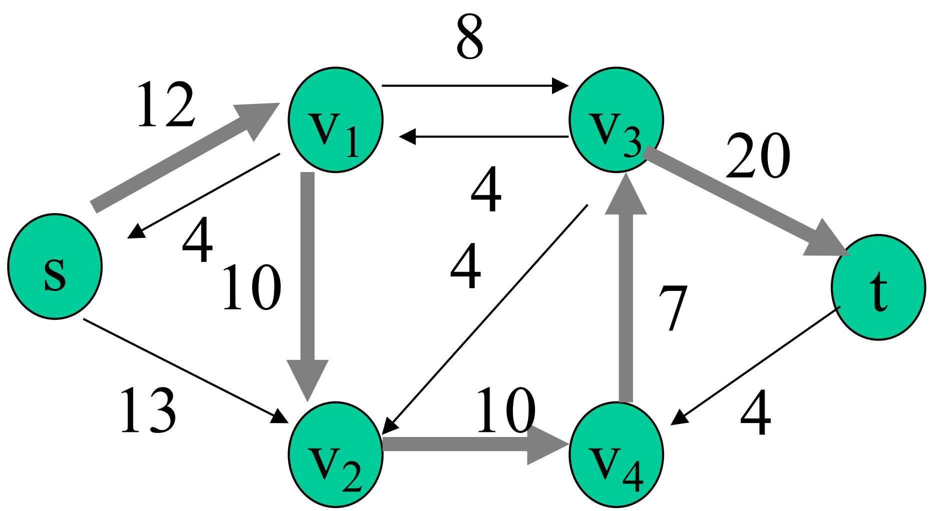
Augmenting paths

- Given a flow network G=(V,E) and a flow f, an augmenting path is a simple path from s to t in the residual network G_f.

- Residual capacity of p : the maximum amount of net flow that we can ship along the edges of an augmenting path p, i.e., $c_f(p)=\min\{c_f(u,v):(u,v) \text{ is on } p\}$.

The residual capacity is 1

Example of an augment path (bold edges)



(b)

The basic Ford-Fulkerson algorithm:

- FORD-FULKERSON(G,s,t)
- for each edge (u,v) $\in$ E[G]
-         do f[u,v] $\leftarrow$ 0
-             f[v,u] $\leftarrow$ 0
- while there exists a path p from s to t in the residual network G_f
-         do $c_f(p) \leftarrow \min\{c_f(u,v): (u,v) \text{ is in } p\}$
-             for each edge (u,v) in p
-                 do $f[u,v] \leftarrow f[u,v] + c_f(p)$
-

Example: next slides (a) to (e)

Execution of the basic Ford-Fulkerson algorithm (successive iterations of the while loop)

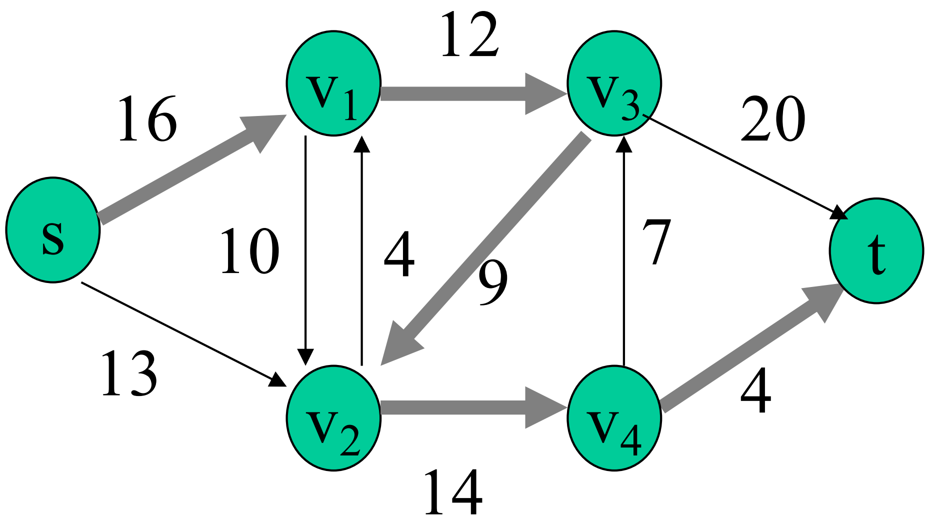
The **left side** of each part shows the **residual network** G_f with a shaded augmenting path p.

The **right side** of each part shows the **new flow** f that results from adding f_p to f.

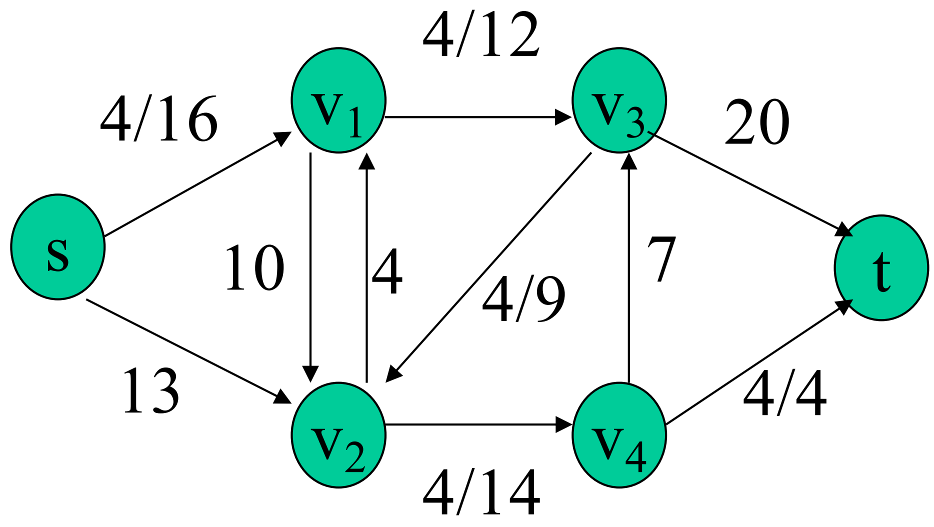
The residual network in (a) is the input network G.

(e) The residual network at the last while loop test. It has no augmenting paths, and the flow f shown in (d) is therefore a maximum flow.

residual network G_f

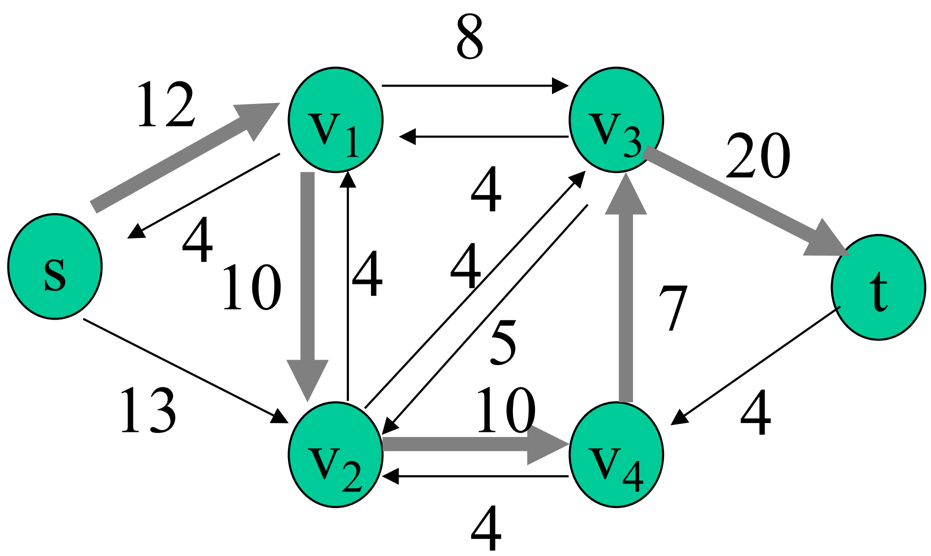


new flow f

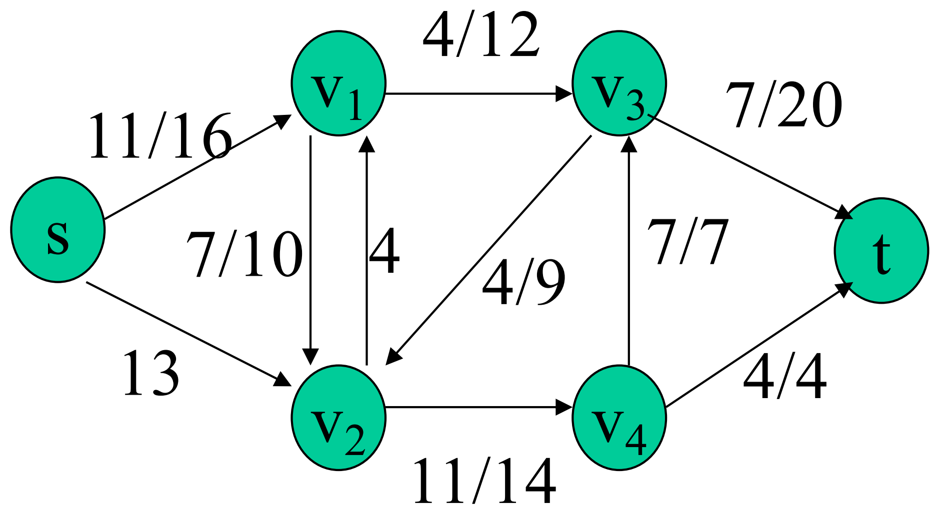


(a)

residual network G_f

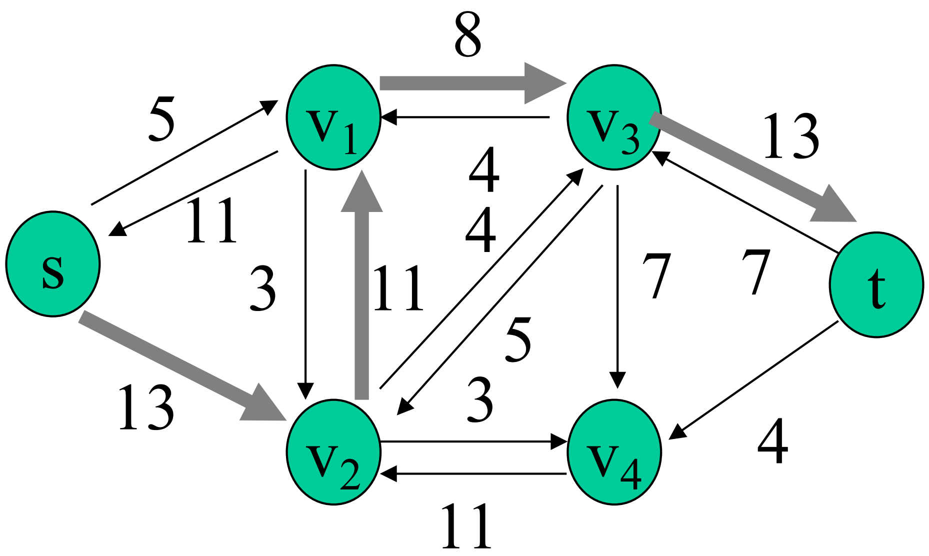


new flow f

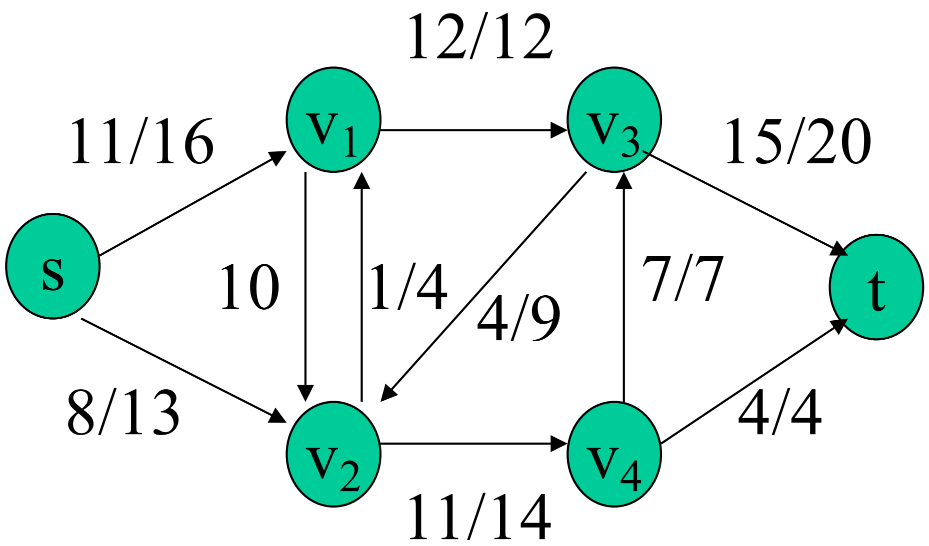


(b)

residual network G_f

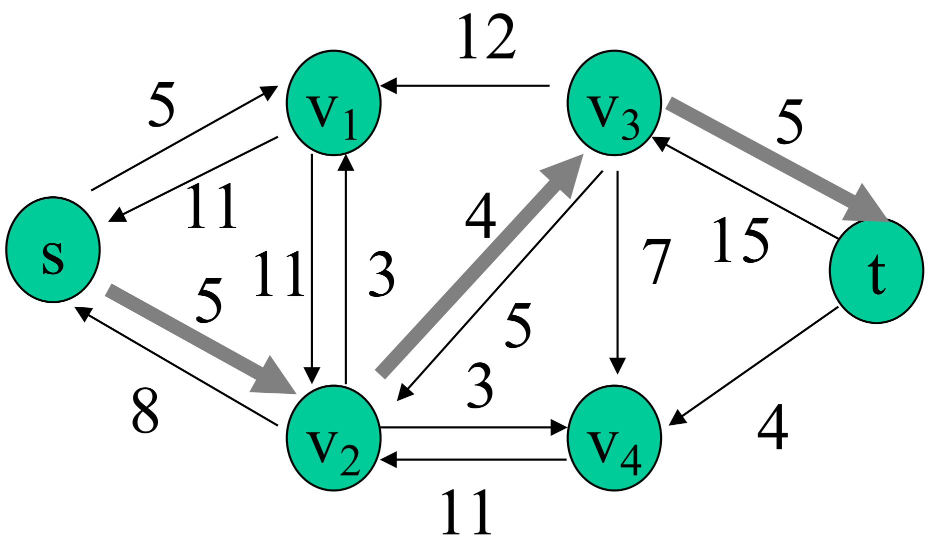


new flow f

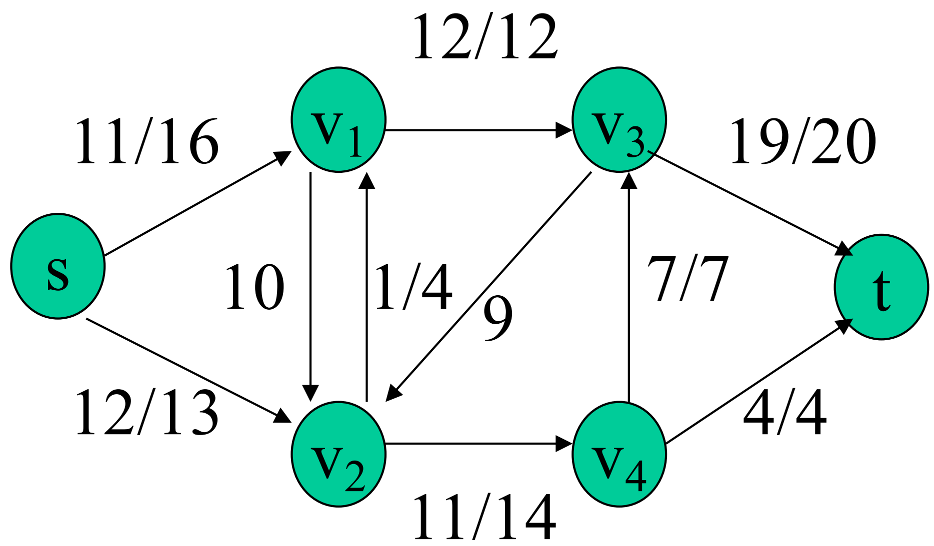


(c)

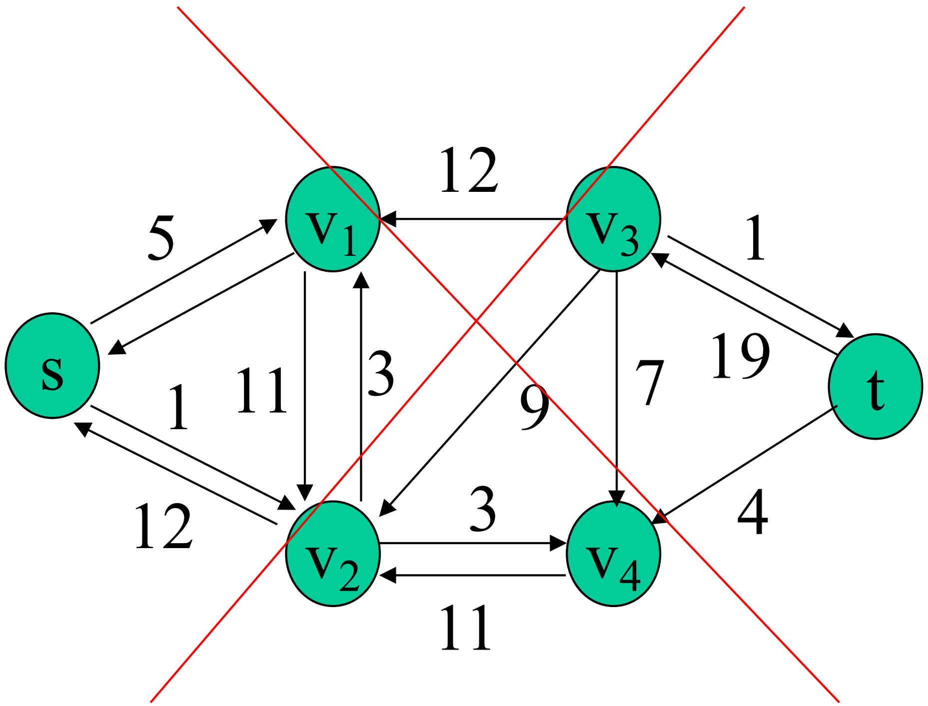
residual network G_f



new flow f



(d)



(e)

No augmenting path !

stop

Time complexity

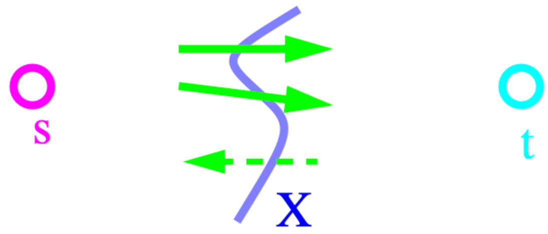
Time complexity of the Ford-Fulkerson's algorithm is

O(max_flow * E)

We run a loop while there is an augmenting path.

In worst case, we may add 1 unit flow in every iteration.
Therefore the time complexity becomes O(max_flow * E).

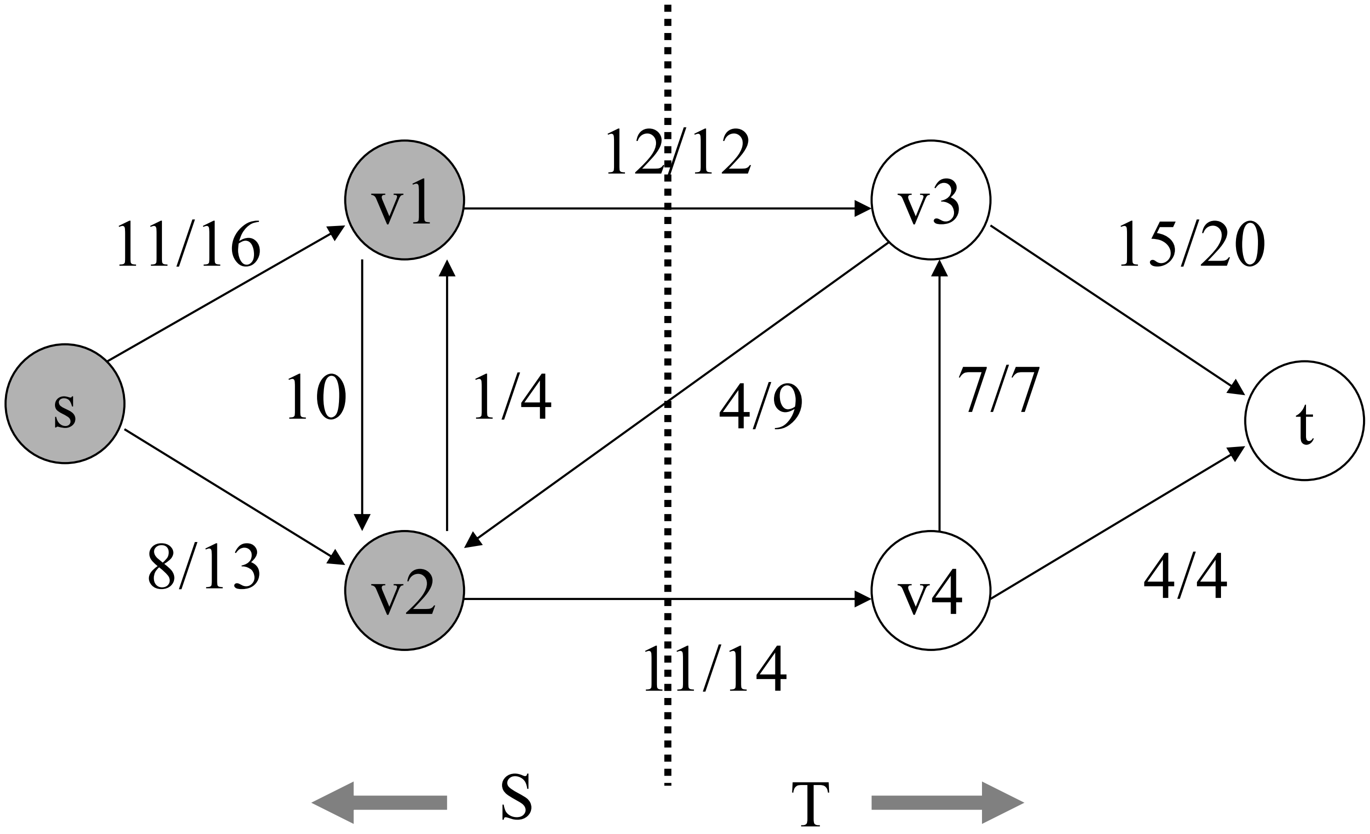
Cuts of flow networks



The proof of the correctness of the Ford-Fulkerson method depends on a concept "cut".

- A cut (S,T) of 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 across the cut (S,T) is $F(S,T) = \Sigma_{u \in S \& v \in T} \; f(u, v)$.
- The capacity of the cut (S,T) is

$$c(S, T) = \Sigma_{u \in S \& v \in T} \; c(u, v).$$



A cut (S,T), where S={s,v1,v2} and T={v3,v4,t}.

The net flow across (S,T) is f(S,T) = 12-4+11 = 19

and the capacity is c(S,T)=12+14=26.

Property of cuts

- Let f be a flow in a flow network G with source s and sink t, and let (S,T) be a cut of G.
  Then, the net flow across (S,T) is f(S,T)=$|f|$.
- Proof: 1. f(S-s, V)=0 by flow conservation.
- 2. f(S, S)=0 since f(u, v)=-f(v, u).
- f(S, T)=f(S, V)-f(S, S)=f(S, V)

  =f(s, V)+f(S-s, V)=f(s, V)=|f|.

Property of cuts (cont.)

- The value of any flow f in a flow network G is bounded from above by the capacity of any cut of G.

- Proof: $f(S, T) \le c(S, T)$.

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:

1. f is a maximum flow in G;
2. The residual network G_f contains no augmenting paths;
3. |f| = c(S,T) for some cut (S,T) of G.

Proof:

1➔2: Otherwise, if a aug. path exists, we can further increase the flow.

2➔3. If no aug. path exists, then we construct S as the set of vertices that is reachable from s. T=V-S. By construction, there is no edge (u, v) in the residual graph such that $u \in S$ and $v \in T$. Thus, |f|=f(S,T)=c(S, T).

3➔1 |f|=f(S, T)=c(S,T). Recall that |f|= f(S, T) $\leq$ c(S,T). Thus, |f| is maximum.

The Edmonds-Karp algorithm

- Find the augmenting path using **breadth-first search** (BFS)

Breadth-first search gives the shortest path for graphs (Assuming the length of each edge is 1.)

- Time complexity of Edmonds-Karp algorithm is $O(VE^2)$.

Playground:

https://visualgo.net/en/maxflow

More examples:

https://www.hackerearth.com/practice/algorithms/graphs/maximum-flow/tutorial