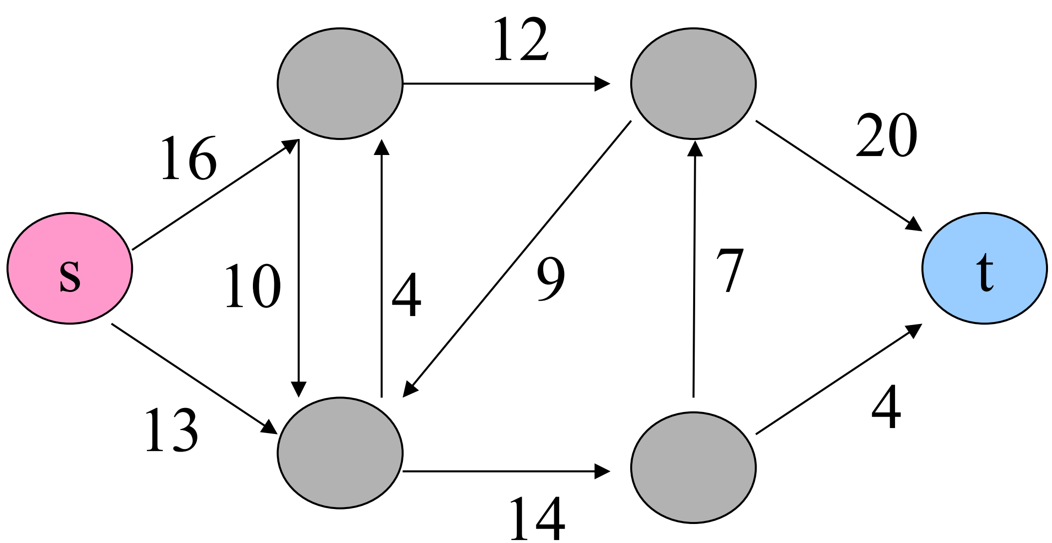
Maximum Flow

- Maximum Flow Problem
- The Ford-Fulkerson method
- Maximum bipartite matching

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.**



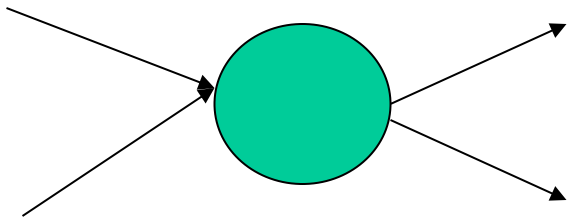
Flow:

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

  we require $f(u,v) \leq c(u,v)$.
- Skew symmetry: For all $u,v \in V$, we require

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

$$\sum_{v \in V} f(u,v) = 0$$

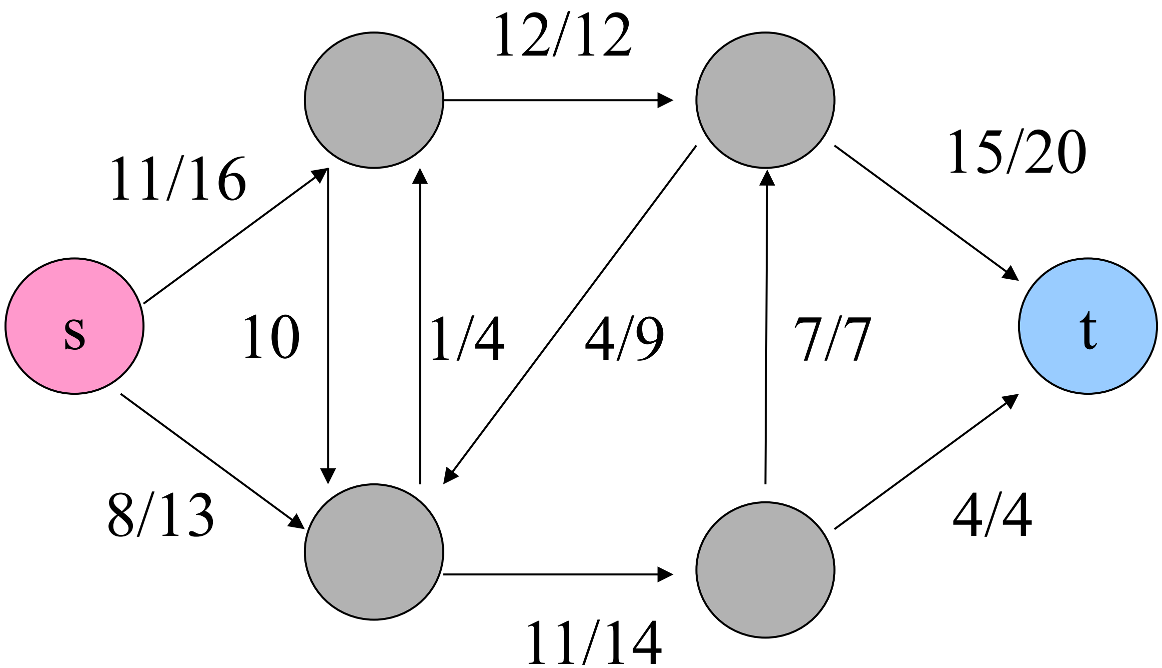


Net flow and value of a flow f:

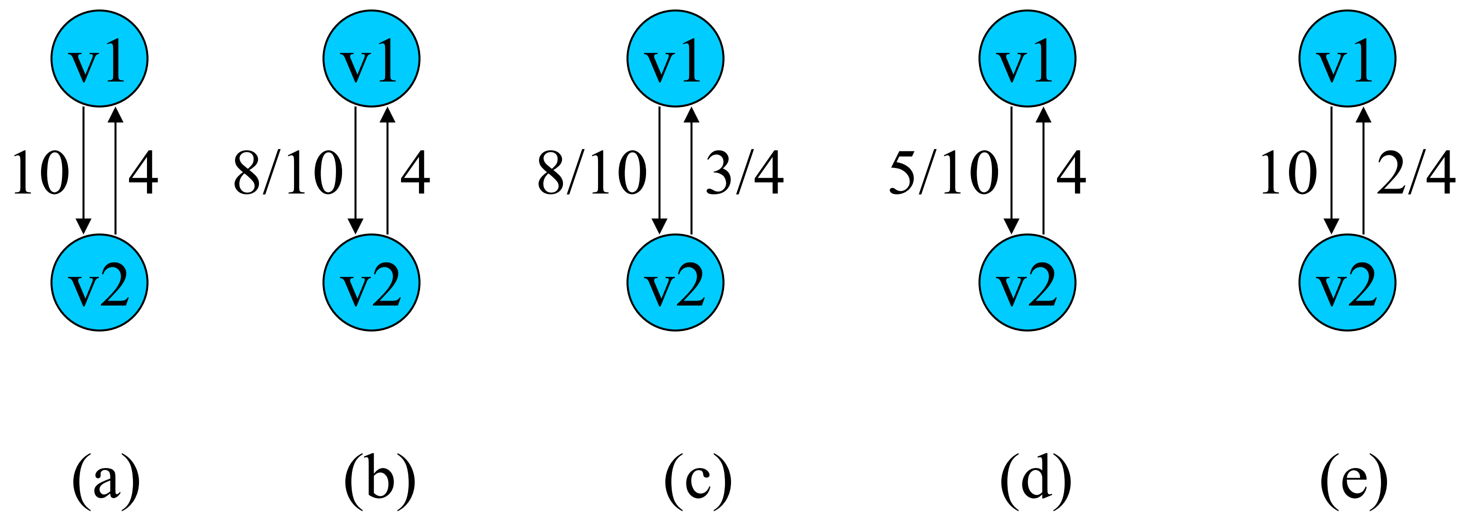
- The quantity f (u,v), which can be positive or negative, is called the net flow from vertex u to vertex v.
- The value of a flow is defined as

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

  - 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 $|f| = 19$.



Cancellation.(a) Vertices v1 and v2,with c(v1,v2)=10 and c(v2,v1) =4.(b)How we indicate the net flow 8 from v1 to v2.
(c) An additional shipment of 3 is made
from v2 to v1. (d) By cancelling flow going in opposite directions,we can represent the situation in (c) with positive net flow in one direction only.(e) Another 7 is shipped from v2 to v1.

The Ford-Fulkerson method:

- The Ford-Fulkerson method depends on three important ideas that are relevant to many flow algorithms and problems:
  - residual networks
  - augmenting paths
  - 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.

Continue:

- FORD-FULKERSON-METHOD(G,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: $c_f(u,v)=c(u,v)-f(u,v)$
- Given a flow network $G=(V,E)$ and a flow f,the residual network of G induced by f is $G_f=(V,E_f)$,where $E_f=\{(u,v)\in V*V: c_f(u,v) > 0\}$(See Slide 15, Figure (a) and (b).)

Augmenting paths:

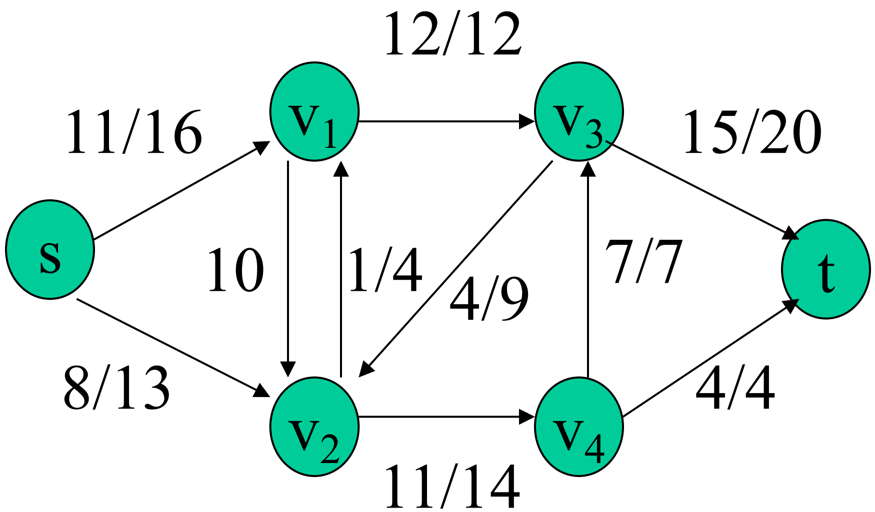
- Given a flow network $G=(V,E)$ and a flow f, an augmenting path is a simple path from s to t in the residual network G_f.

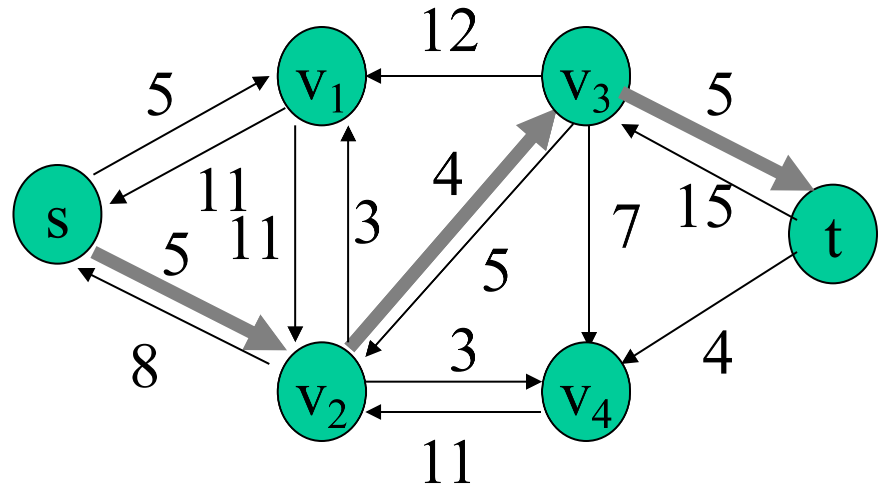
- Residual capacity of p : the maximum amount of net flow that we can ship along the edges of an augmenting path p, i.e., $c_f(p)=\min\{c_f(u,v):(u,v) \text{ is on } p\}$.

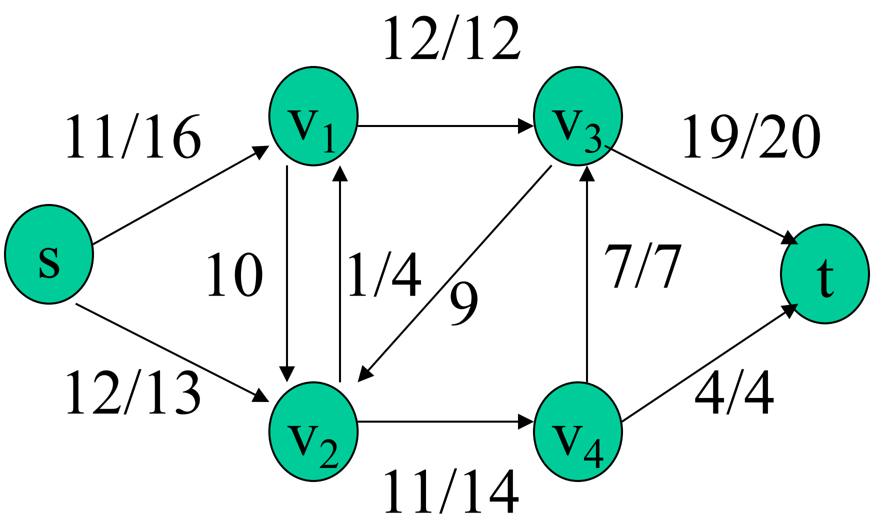
The residual capacity is 1.

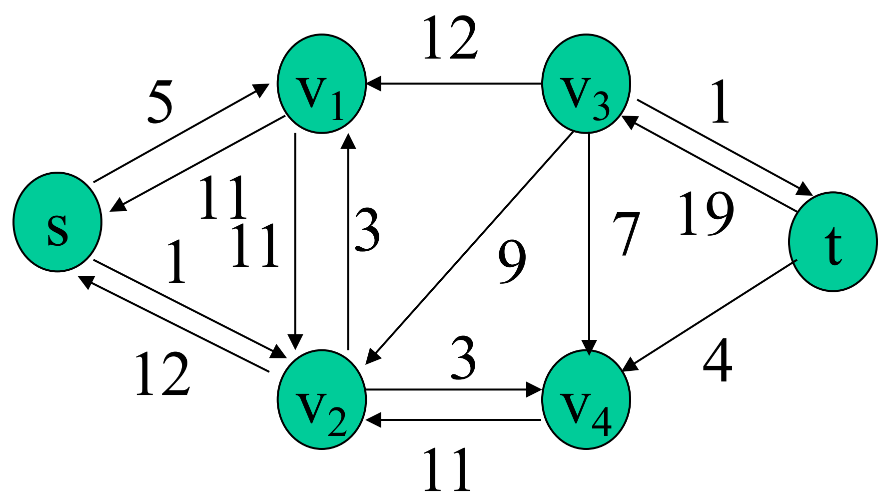
Example:

- (a)The flow network G and flow f .
- (b)The residual network G_f with augmenting path p shaded; its residual capacity is $c_f(p)=c(v_2,v_3)=4$.
- (c) The flow in G that results from augmenting along path p by its residual capacity 4.
- (d) The residual network induced by the flow in (c).


s
v_1
v_2
v_3
v_4
t
11/16
8/13
12/12
10
1/4
4/9
7/7
15/20
4/4
11/14
(a)


s
v_1
v_2
v_3
v_4
t
5
11
5
8
12
11
3
4
5
7
5
15
3
11
4
(b)


s
v_1
v_2
v_3
v_4
t
11/16
12/13
12/12
10
1/4
9
7/7
19/20
4/4
11/14
(c)

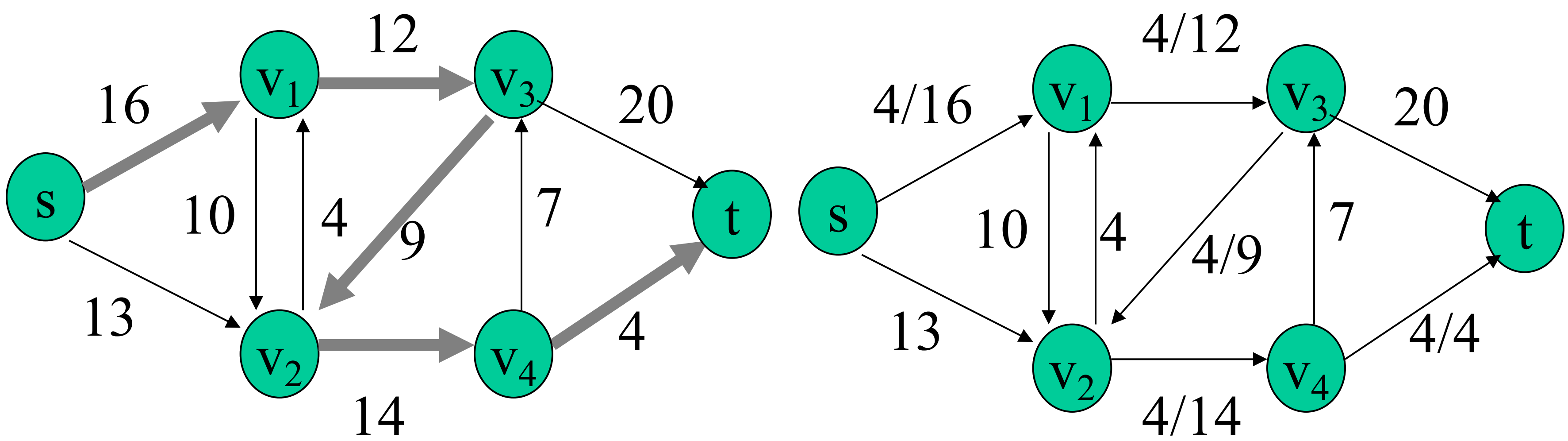

s
v_1
v_2
v_3
v_4
t
5
11
1
12
12
11
3
9
7
1
19
3
11
4
(d)

The basic Ford-Fulkerson algorithm:

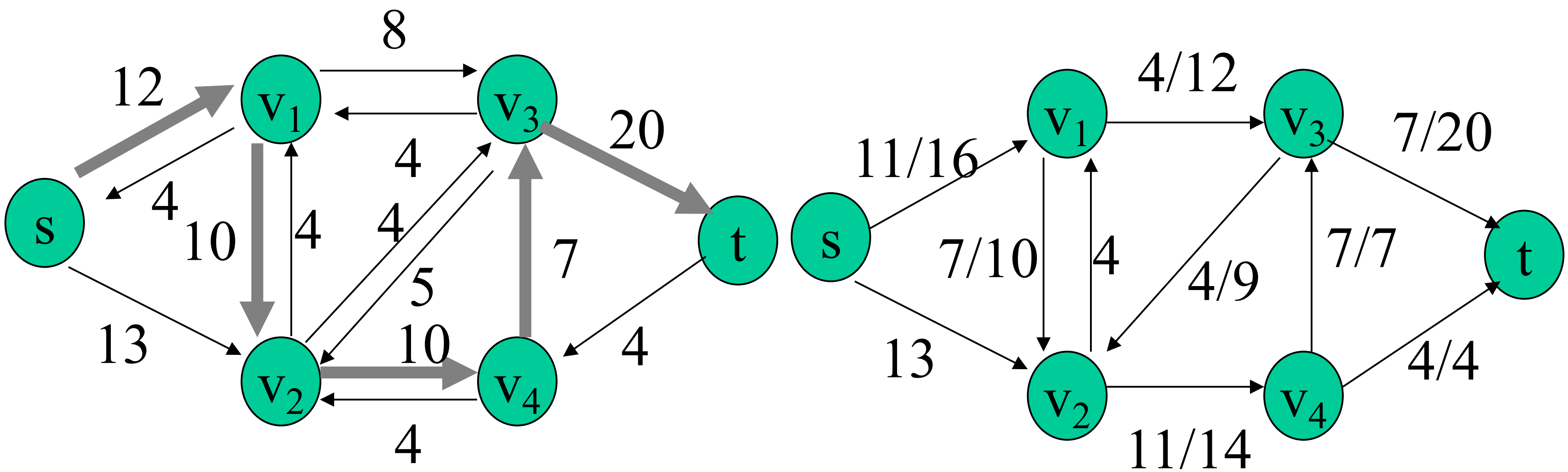
- FORD-FULKERSON(G,s,t)
- for each edge (u,v) $\in$E[G]
- do f[u,v] $\leftarrow$ 0
- f[v,u] $\leftarrow$ 0
- while there exists a path p from s to t in the residual network G_f
- do $c_f(p) \leftarrow \min\{c_f(u,v): (u,v) \text{ is in } p\}$
- for each edge (u,v) in p
- do f[u,v] $\leftarrow$ f[u,v]+$c_f(p)$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

Example:

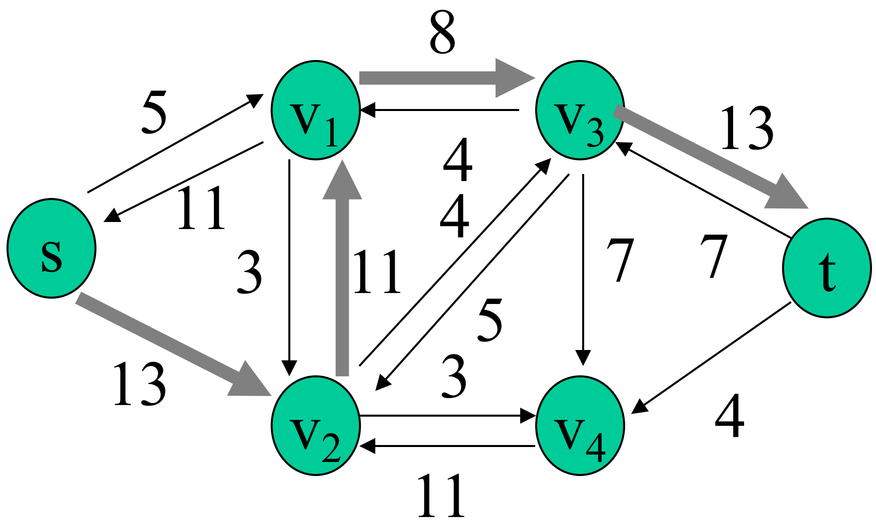
- The execution of the basic Ford-Fulkerson algorithm.(a)-(d) Successive iterations of the while loop. The left side of each part shows the residual network G_f from line 4 with a shaded augmenting path p.The right side of each part shows the new flow f that results from adding f_p to f.The residual 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.



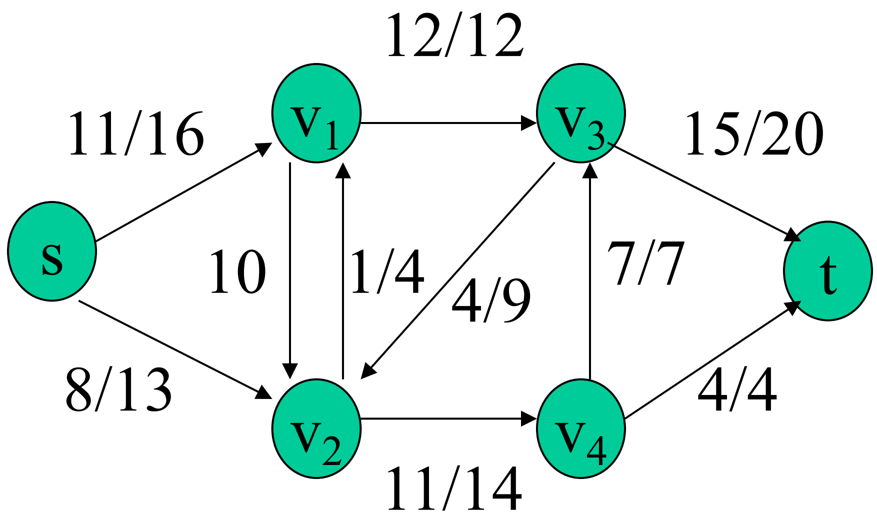
(a)


8
12
v1
v3
20
4
4
s
4
10
4
7
t
5
13
10
4
v2
v4
4
4/12
v1
v3
7/20
11/16
s
7/10
4
7/7
t
4/9
13
4/4
v2
v4
11/14

(b)

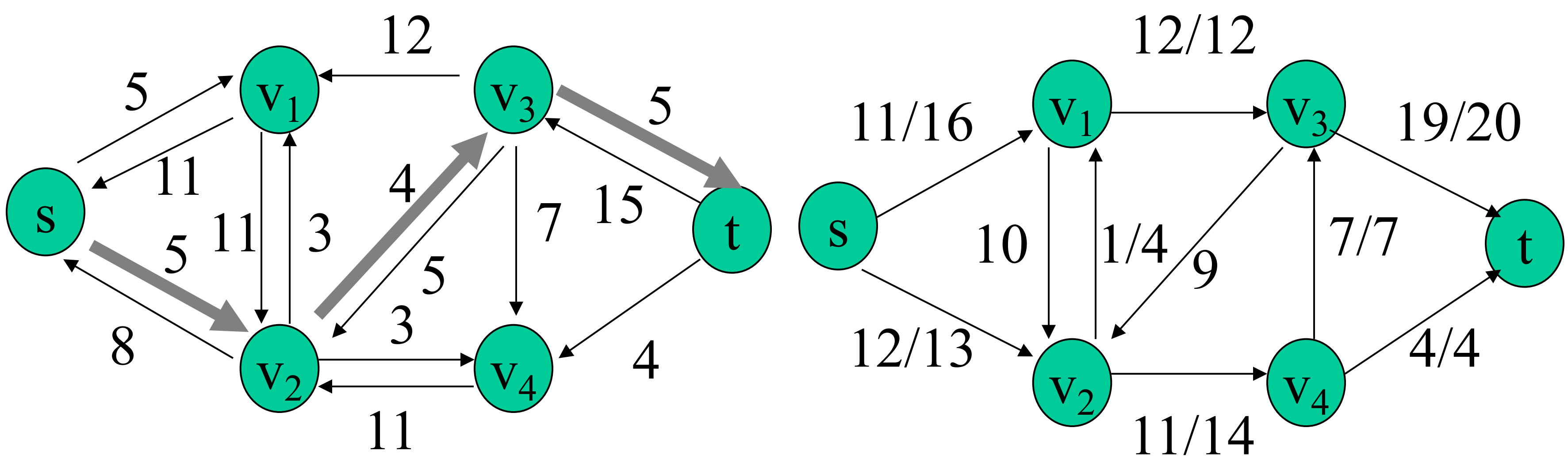


8
5
13
11
4
4
3
11
7
7
5
3
13
4
11
s
v_1
v_3
t
v_2
v_4

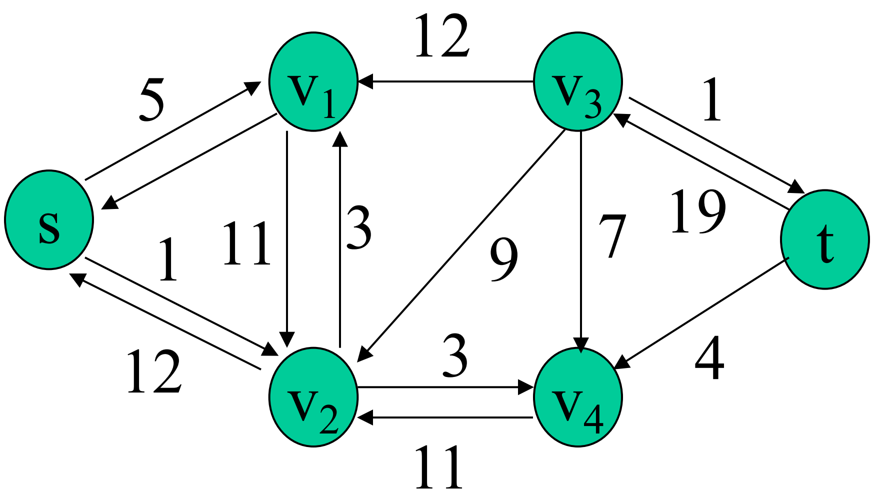


12/12
11/16
15/20
10
1/4
4/9
7/7
8/13
4/4
11/14
s
v_1
v_3
t
v_2
v_4

(c)


12
5
11
5
11
3
4
5
5
3
7
15
5
8
4
11
s
t
12/12
11/16
19/20
10
1/4
9
7/7
12/13
4/4
11/14
s
t

(d)



(e)

Time complexity:

- If each c(e) is an integer, then time complexity is O(|E|f*), where f* is the maximum flow.
- Reason: each time the flow is increased by at least one.
- This might not be a polynomial time algorithm since f* can be represented by log (f*) bits. So, the input size might be log(f*).

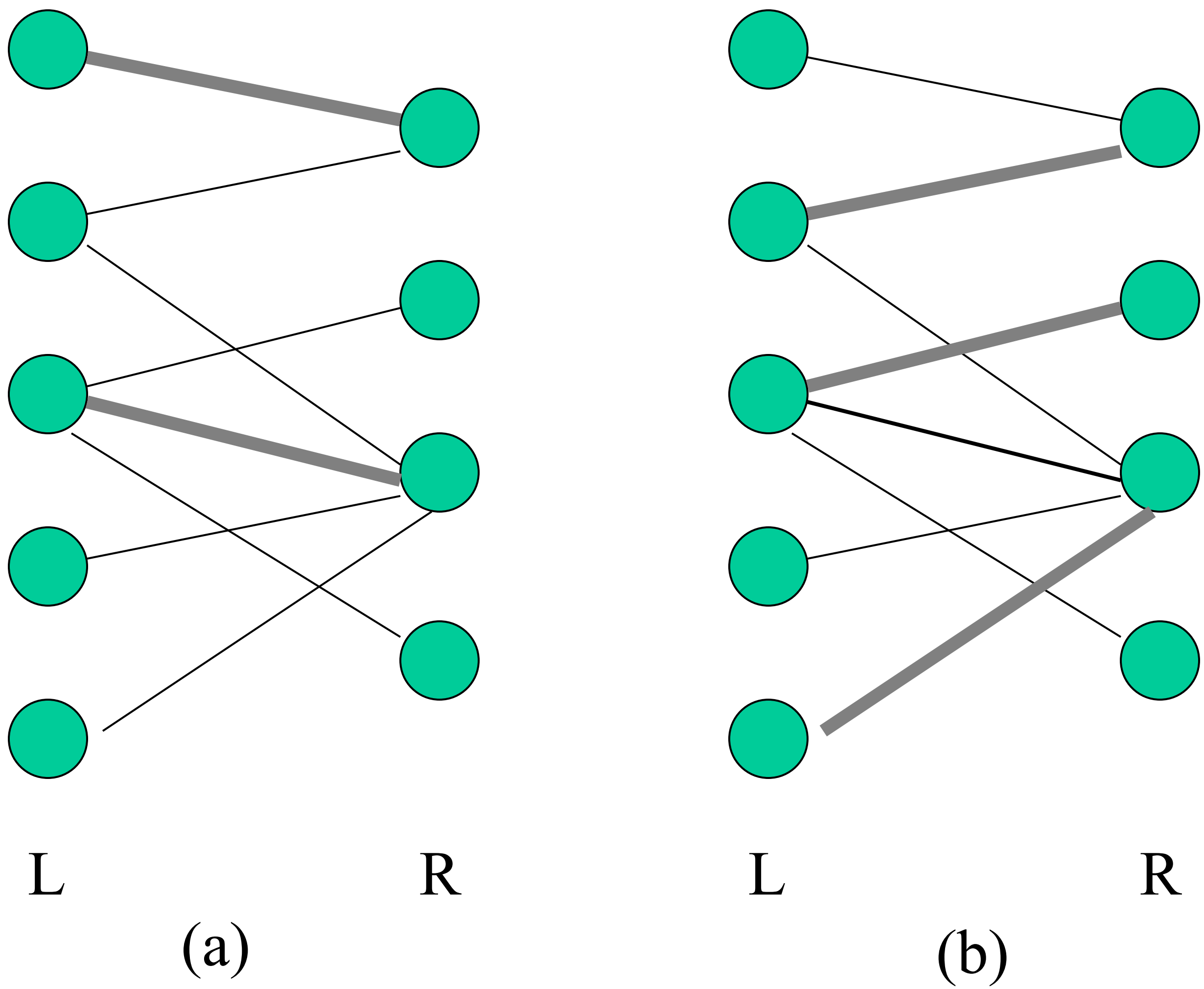
The Edmonds-Karp algorithm

- Find the augmenting path using breadth-first search.
- Breadth-first search gives the shortest path for graphs (Assuming the length of each edge is 1.)
- Time complexity of Edmonds-Karp algorithm is $O(VE^2)$.
- http://en.wikipedia.org/wiki/Edmonds-Karp_algorithm

Maximum bipartite matching:

- Bipartite graph: a graph (V, E), where $V=L\cup R$, $L\cap R$=empty, and for every $(u, v)\in E$, $u \in L$ and $v \in R$.
- Given an undirected graph $G=(V,E)$, a matching is a subset of edges $M\subseteq E$ such that for all vertices $v\in V$,at most one edge of M is incident on v.We say that a vertex $v \in V$ is matched by matching M if some edge in M is incident on v;otherwise, v is unmatched. A maximum matching is a matching of maximum cardinality,that is, a matching M such that for any matching M', we have .

$$|M| \geq |M'|$$

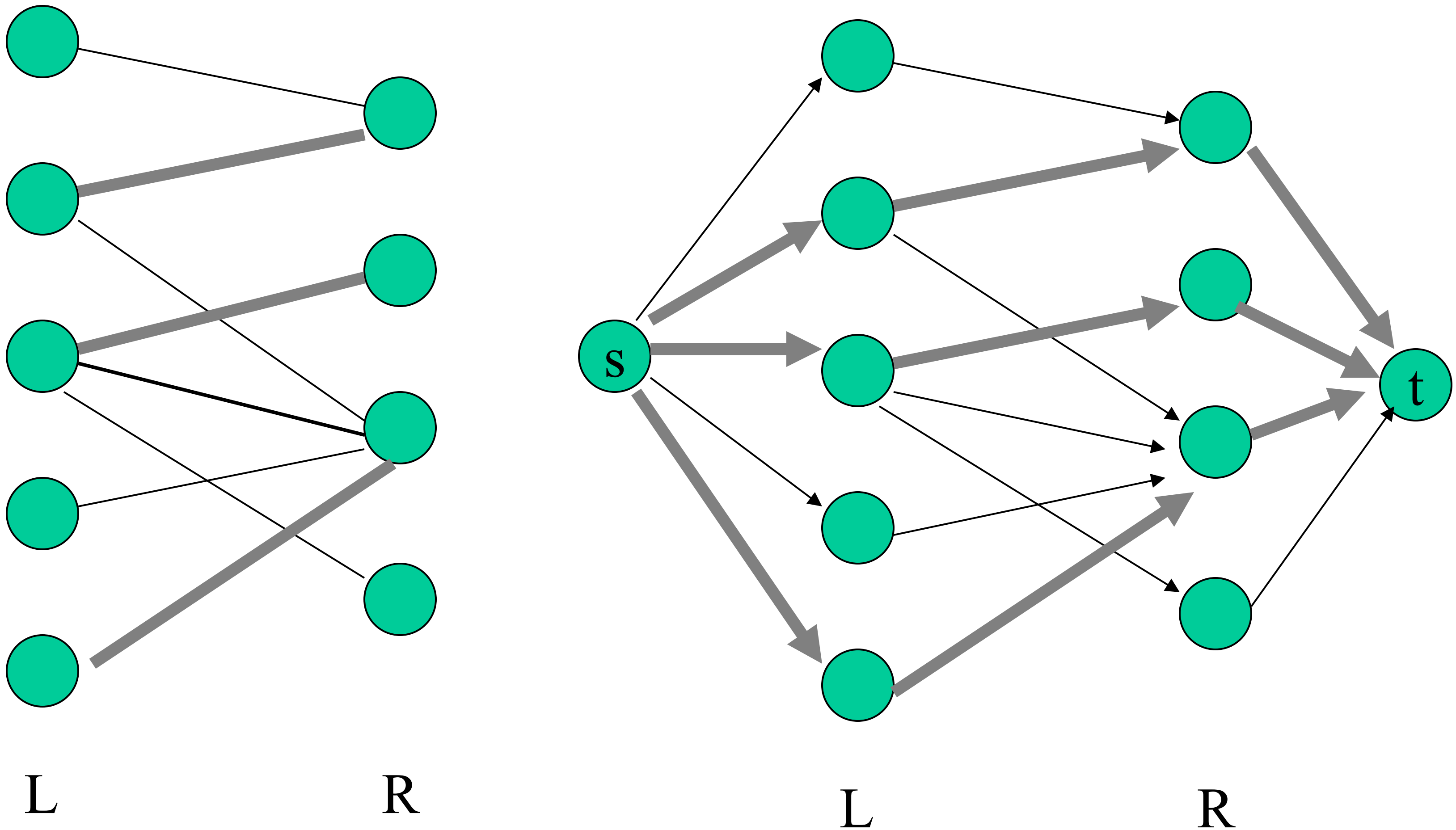


A bipartite graph G=(V,E) with vertex partition V=L∪R.(a)A matching with cardinality 2.(b) A maximum matching with cardinality 3.

Finding a maximum bipartite matching:

- We define the corresponding flow network G'=(V',E') for the bipartite graph G as follows. Let the source s and sink t be new vertices not in V, and let $V'=V\cup\{s,t\}$.If the vertex partition of G is $V=L\cup R$, the directed edges of G' are given by $E'=\{(s,u):u\in L\}\cup\{(u,v):u\in L, v\in R, \text{and } (u,v)\in E\}\cup\{(v,t):v\in R\}$.Finally, we assign unit capacity to each edge in E'.

- We will show that a matching in G corresponds directly to a flow in G's corresponding flow network G'. We say that a flow f on a flow network G=(V,E) is integer-valued if f(u,v) is an integer for all $(u,v)\in V*V$.



(a) (b)

(a)The bipartite graph G=(V,E) with vertex partition V=L∪R. A maximum matching is shown by shaded edges.(b) The corresponding flow network. Each edge has unit capacity.Shaded edges have a flow of 1,and all other edges carry no flow.

Problem

- 820 (maxflow)
- http://uva.onlinejudge.org/external/8/820.pdf
- 10092 (matching)
- http://uva.onlinejudge.org/external/100/10092.pdf
- 10080
- http://uva.onlinejudge.org/external/100/10080.pdf