Network Flows (Cont...)

RAMESH K. JALLU

LEC-09

DT. 07/02//22

The Ford-Fulkerson method

- This method was developed by Lester Randolph Ford Jr. and Delbert Ray Fulkerson in 1956
- It is called as a method rather than an algorithm
- The method depends on 3 ideas: residual networks, augmenting paths, and cuts
- The idea behind the algorithm is as follows
  - As long as there is a path from the source to the sink, with available capacity on all edges in the path, we send flow along one of the paths
  - Then we find another path, and so on
  - A path with available capacity is called an ***augmenting path***

Ford–Fulkerson algorithm	$O(E\|f_{max}\|)$
Edmonds–Karp algorithm	$O(VE^2)$
Dinic's algorithm	$O(V^2E)$
MKM (Malhotra, Kumar, Maheshwari) algorithm[10]	$O(V^3)$
Dinic's algorithm with dynamic trees	$O(VE\log V)$
General push–relabel algorithm[11]	$O(V^2E)$
Push–relabel algorithm with *FIFO* vertex selection rule[11]	$O(V^3)$
Push–relabel algorithm with *maximum distance* vertex selection rule[12]	$O(V^2\sqrt{E})$
Push-relabel algorithm with dynamic trees[11]	$O\left(VE\log\frac{V^2}{E}\right)$
KRT (King, Rao, Tarjan)'s algorithm[13]	$O\left(VE\log_{\frac{E}{V\log V}}V\right)$
Binary blocking flow algorithm[14]	$O\left(E\cdot\min\{V^{2/3},E^{1/2}\}\cdot\log\frac{V^2}{E}\cdot\log U\right)$
James B Orlin's + KRT (King, Rao, Tarjan)'s algorithm[9]	$O(VE)$

Cont…

```
FORD-FULKERSON-METHOD(G, s, t)
initialize flow f to 0
while there exists an augmenting path p in the residual network G_f
    augment flow f along p
return f
```

- At each iteration, we increase the flow value in G by finding an "augmenting path" in an associated residual network G_f
- Although each iteration of the method increases the value of the flow, we shall see that the flow on any particular edge of G may increase or decrease
- Decreasing the flow on some edges may be necessary in order to enable an algorithm to send more flow from the source to the sink
- We repeatedly augment the flow until the residual network has no more augmenting paths

Residual capacity

- Given a flow network $G = (V, E)$ with source s and sink t
- Let f be a flow in G, and consider a pair of vertices $u, v \in V$
- We define the residual capacity $c_f(u, v)$ by

$$c_f(u,v) = \begin{cases} c(u,v) - f(u,v) & \text{if } (u,v) \in E \ , \\ f(v,u) & \text{if } (v,u) \in E \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

- Because of our assumption that $(u, v) \in E$ implies $(v, u) \notin E$, exactly one case in equation applies to each ordered pair of vertices

Residual networks

- Given a flow network G and a flow f , the residual network G_f 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 G_f with a "residual capacity" of $c_f(u, v) = c(u, v) - f(u, v)$
- The only edges of G that are in G_f are those that can admit more flow
- Those edges (u, v) whose flow equals their capacity have $c_f(u, v) = 0$, and they are not in G_f

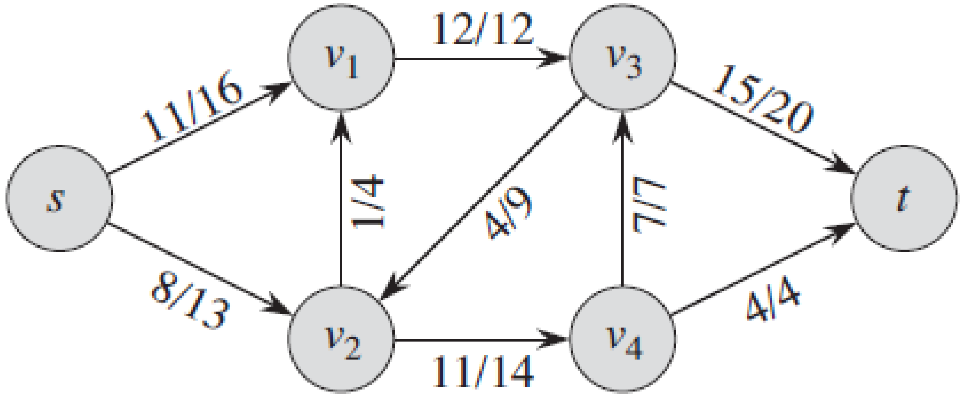
Cont...

- The residual network G_f may also contain edges that are not in G
- As an algorithm manipulates the flow, with the goal of increasing the total flow, it might need to decrease the flow on a particular edge
- In order to represent a possible decrease of a positive flow $f(u, v)$ on an edge in G, we place an edge (u, v) into G_f with residual capacity $c_f(v, u) = f(u, v)$
    - That is, an edge that can admit flow in the opposite direction to (u, v) at most canceling out the flow on (u, v)
- These reverse edges in the residual network allow to send back flow it has already sent along an edge
- Sending flow back along an edge is equivalent to decreasing the flow on the edge

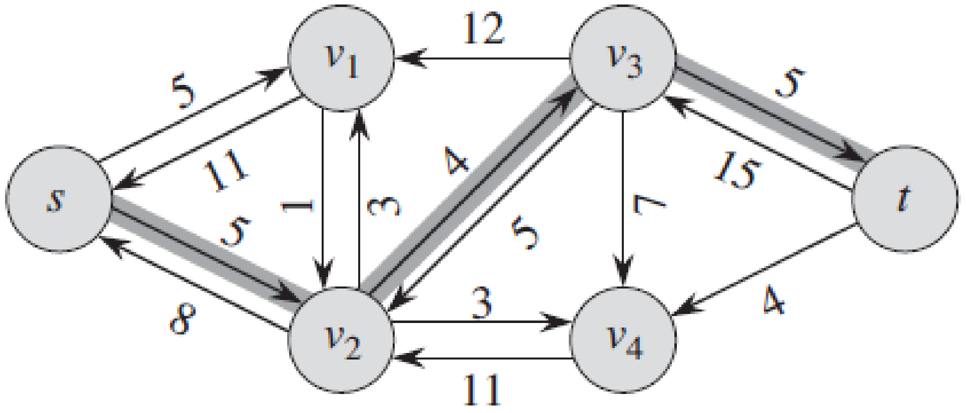
Forward and backward edges in G_f

- Given a flow network $G = (V, E)$, and a flow f on G, we define the ***residual network/graph*** $G_f = (V, E_f)$ of G with respect to f as follows
  - The node set of G_f is the same as that of G
  - For each edge (u, v) of G on which $f(u, v) < c(u, v)$, there are $c(u, v) - f(u, v)$ "leftover" units of capacity on which we could try pushing flow forward
    - We include the edge (u, v) in G_f, with a capacity of $c(u, v) - f(u, v)$
    - We will call edges included this way ***forward edges***
  - For each edge (u, v) of G on which $f(u, v) > 0$, there are $f(u, v)$ units of flow that we can "*undo*" if we want to, by pushing flow backward
    - We include the edge (v, u) in G_f, with a capacity of $f(u, v)$
    - We will call edges included this way ***backward edges***

Example



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



s
v_1
v_2
v_3
v_4
t
5
11
5
8
1
3
12
4
5
7
3
11
5
15
4

Observations

- 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 $E_f = \{(u, v) \in V \times V \mid c_f(u, v) > 0\}$
- Each edge (u, v) in G can give rise to one or two edges in G_f: if $0 < f(u, v) < c(u, v)$ it results in both a forward edge and a backward edge being included in G_f
  - Thus G_f has at most twice as many edges as G, i.e., $|E_f| \leq 2|E|$
- G_f does not satisfy our definition of a flow network because it may contain both an edge (u, v) and its reversal (v, u)
- A flow in a residual network provides a roadmap for adding flow to the original flow network

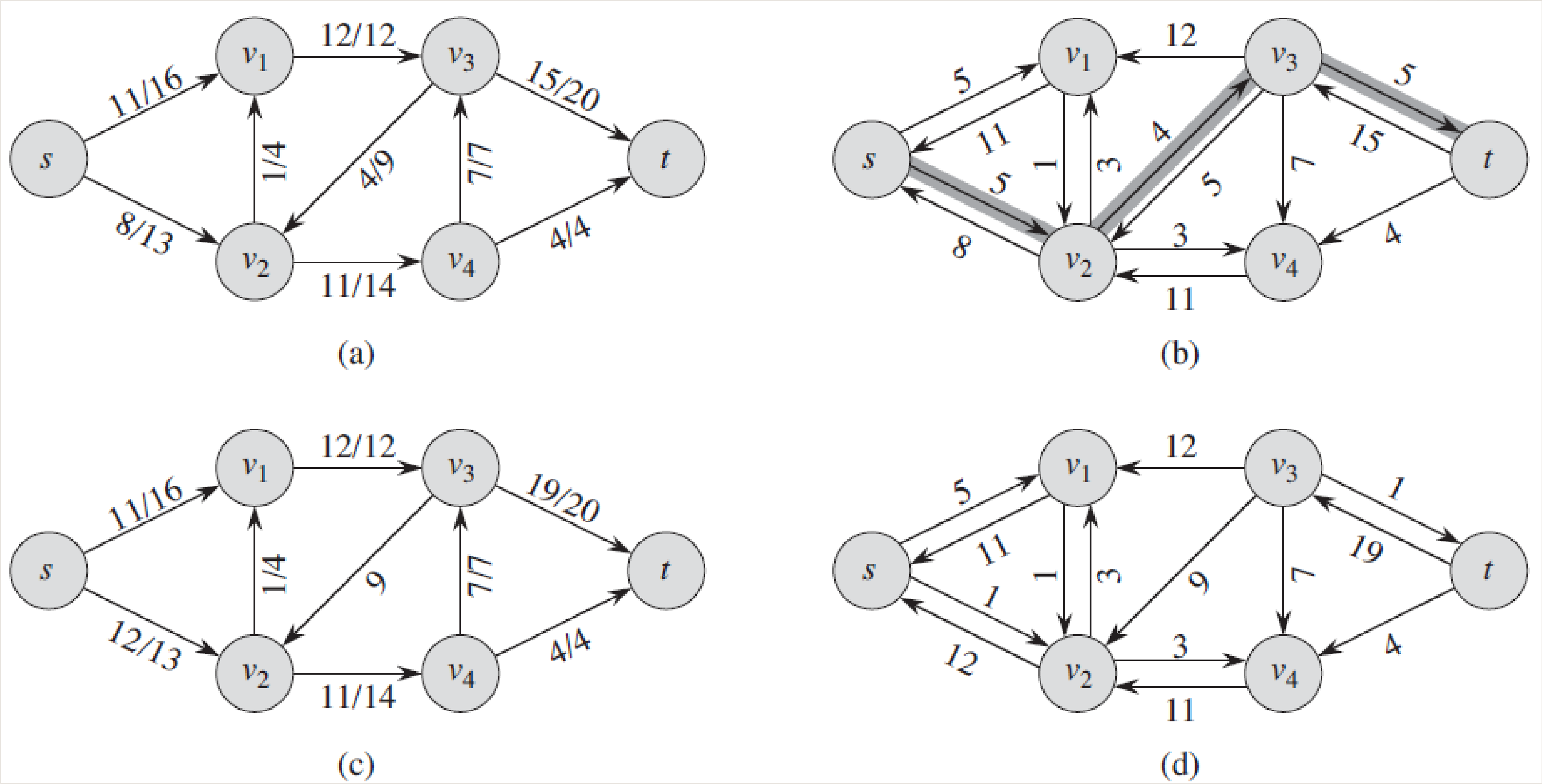
Augmenting Paths in a Residual Graph

- 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 G_f
- By the definition of the residual network, we may increase the flow on an edge (u, v) of an augmenting path by ***up to*** $c_f(u, v)$ without violating the capacity constraint on whichever of (u, v) and (v, u) is in the original flow network G
- For any augmenting path p in G_f, define ***bottleneck(p, f)*** to be the minimum residual capacity of any edge on p, with respect to the flow f
  - That is, $bottleneck(p, f) = min\{c_f(u, v) \mid (u, v) \in p\}$

Yielding a new flow in G

- Let f be a flow in G and let p be an augmenting path in G_f with $b = bottleneck(p, f)$
- Let f' be a new flow obtained by doing the following operation obtained by increasing and/or decreasing the flow values on edges of p
  - For every edge $(u, v) \in p$
    - If (u, v) is a forward edge in G_f then increase $f(u, v)$ in G by b
    - If (u, v) is a backward edge in G_f then decrease $f(u, v)$ in G by b
- We can argue that f' is indeed a flow in G

Example



- Thank you!