COMTEK3, CCT3 & ESD3: ALGORITHMS
Flow Networks

Ramoni Adeogun
Associate Professor and Head of the AI for Communication Group
Wireless Communication Networks Section (WCN)
Email: ra@es.aau.dk
October, 2023

AALBORG UNIVERSITET

Outline

- Flow Networks
- Flow in a Network
  - Flow value and maximal flow
  - Modelling antiparallel edges
  - Modelling networks with multiple sources and sinks
- Maximum Flow Algorithms
  - Ford-Fulkerson Algorithm
  - Edmonds-Karps Algorithm
- Summary

Flow Networks

- Graphs can be used to model various systems where a “material” is transported from a **source** to a **sink** through a network
  - Such networks are called flow networks or transport networks
  - Examples:
    - Road transportation
    - Utility (electricity, water, gas) distribution systems
    - Communication networks
  - Flow of “material” through the network is typically limited => finite **capacity**
  - Material can often be assumed to be consumed by the sink => the material is **conserved.**

Flow Networks

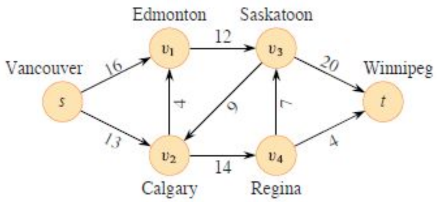
- A flow network $G = (V, E)$ is a digraph in which:
  - each edge $(u, v) \in E$ has nonnegative capacity $c(u, v) \geq 0$
  - there are no anti-parallel edges or loops
  - for each $(u, v) \notin E$, $c(u, v) = 0$

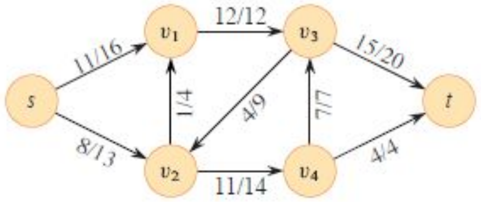
- We distinguish two vertices in a flow network: a **source** s and a **sink** t.

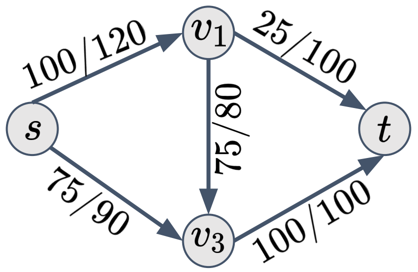
- Assumption: the flow network contains a path $s \rightsquigarrow v \rightsquigarrow t$
  - the graph is therefore connected and $|E| \geq |V| - 1$??

Flow of Network

- Let $G = (V, E)$ be a flow network with a capacity function c.
- A flow in G is a real-valued function $f : V \times V \rightarrow \mathbb{R}$ satisfying the following properties
  - **Capacity constraint**: $0 \leq f(u,v) \leq c(u,v), \forall u,v \in V$
  - **Flow conservation:** $\sum_{v \in V} f(v,u) = \sum_{v \in E} f(u,v), \forall u \in V - \{s,t\}$
- Representation







Is the flow on the network valid? If not, how can we make it valid?

Flow Value and Maximal Flow

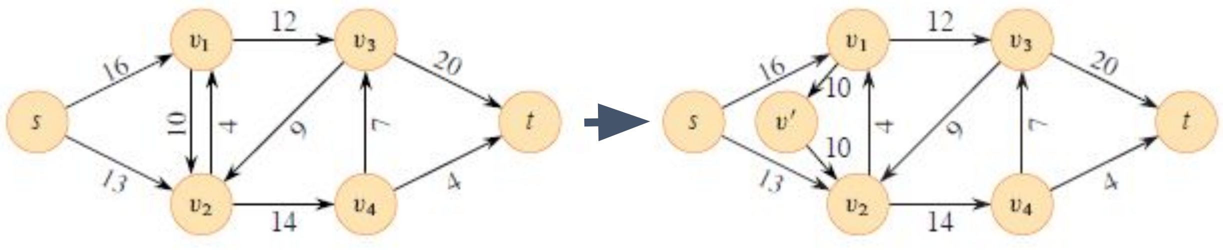
- $f(u,v)$ is called the value with **flow value**

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

- Typically equal to **zero**
- A flow f is **maximal** if there exist no other flows with flow value greater than $|f|$
- Maximal flow problem
  - To find a maximal flow of a given flow network

Modelling Antiparallel Edges

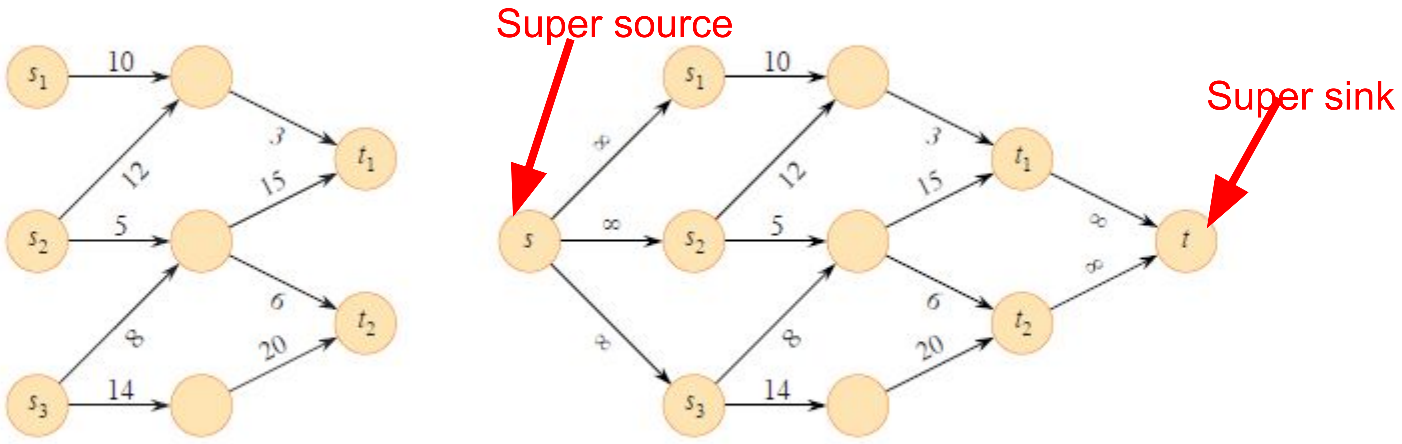
- A network with anti-parallel edges can be converted into an equivalent one with parallel edges as below:



- Can you think of an application where anti-parallel edges may arise in real-life?

Network with Multiple Sources

❖ A **multiple-source, multiple-sink maximum-flow problem** into a problem with a **single source and a single sink** using the procedure illustrated in the diagram below.



Residual Capacity

- For a flow network $G = (V, E)$ with flow f
  - it is possible to change the flow by changing that of individual edges
  - edge $(u, v) \in E$ with capacity $c(u, v)$ can carry an additional flow of $c(u, v) - f(u, v)$
  - alternatively, we can decrease the flow on the edge by up to $f(u, v)$
    - Decreasing the flow corresponds to sending flow "backwards" in the graph
      - To allow for this we define the **residual capacity** as:

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

- Considering the assumption that $(u, v) \in E \implies (u, v) \notin E$
  - exactly one case applies to each ordered pair of vertices in G.

Residual Network

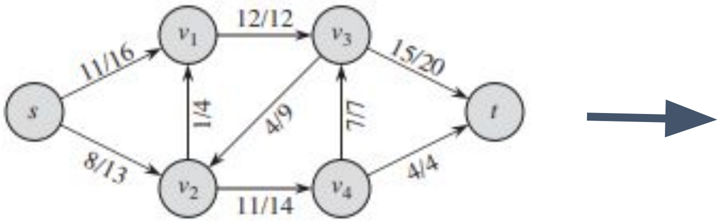
- Given a flow network $G = (V, E)$ and a flow f, we define the **residual network** as

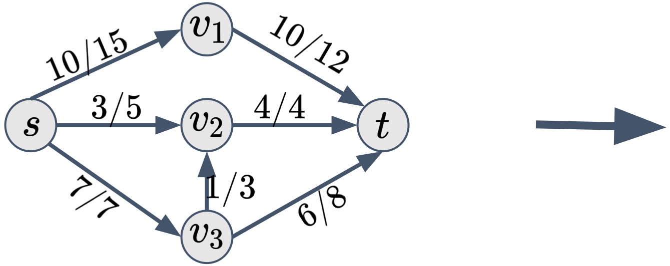
$$G_f = (V, E_f)$$
$$E_f = \{(u, v) \in V \times V : c_f(u, v) > 0\}$$

- This network includes the edges that admit more flow and edges for which the flow may be reduced
- No edges are included in E_f if $c_f(u, v) = 0$
  - This happens if the flow on (u, v) is at capacity
- The residual network clearly changes with the flow.
  - By updating the flow, we also update the residual network

Residual Network - Example

Draw the residual network for each of the following flow networks





Augmenting Path

- A path from $s \to t$ in the residual network G_f is an **augmenting path** in G with respect to the flow, f**.**

- If an augmenting path exists, the flow along it is not maximum but can be increased by the **residual capacity**:

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

- We may think of $c_f(p)$ as the capacity of the "bottle-neck" edge of the augmenting path p.

Augmenting Path - Example

- Consider a flow network and its residual:

$$G: \; s \xrightarrow{1/2} v_1 \xrightarrow{1/4} t$$

$$G_f: \; \begin{tikzcd} s \arrow[r, shift left, "1"] & v_1 \arrow[l, shift left, "1"] \arrow[r, shift left, "3"] & t \arrow[l, shift left, "1"] \end{tikzcd}$$

- G_f contains one augmenting path: $p: s \to a \to t$ with residual capacity $c_f(p) = 1$
- The flow can then be updated as:

$$G: \; s \xrightarrow{2/2} v_1 \xrightarrow{2/4} t$$

- Sketch the residual network for the updated flow.

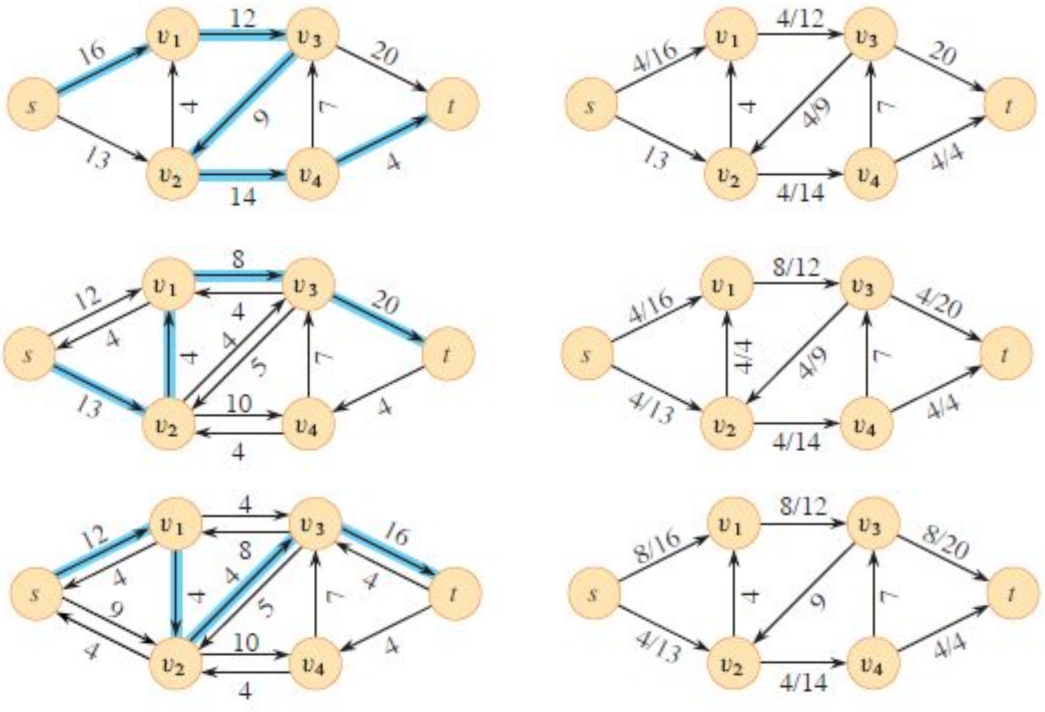
Maximum Flow Algorithms

- Since the maximum flow problem is relevant in many practical situations, many algorithms have been invented
- The Ford-Fulkerson method was the first that appeared.

```
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
```

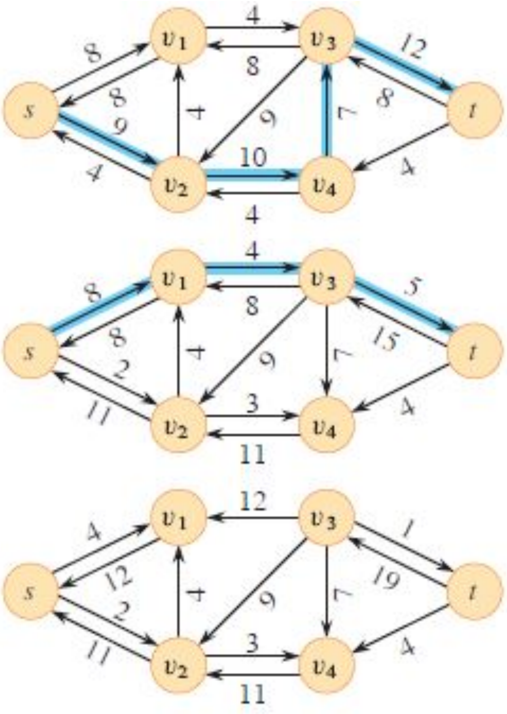
```
FORD-FULKERSON(G, s, t)
for each edge (u, v) ∈ G.E
    (u, v).f = 0
while there exists a path p from s to t in the residual network G_f
    c_f(p) = min{c_f(u, v) : (u, v) is in p}
    for each edge (u, v) in p
        if (u, v) ∈ G.E
            (u, v).f = (u, v).f + c_f(p)
        else (v, u).f = (v, u).f − c_f(p)
return f
```

FORD-FULKERSON EXAMPLE

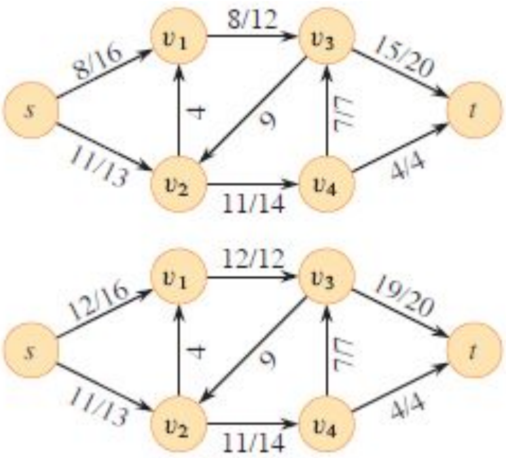


G_f G

FORD-FULKERSON EXAMPLE 2



G_f



G

The Edmonds-Karps Algorithm

- The Ford-Fulkerson algorithm can be improved by choosing the augmenting path as the shortest path from s to t in the residual network
  - Each edge in the residual network has unit distance (weight)
  - This can be done using the BFS algorithm

- The modified algorithm is the Edmonds-Karps algorithm
  - has time complexity of

$$O\left(|V| \cdot |E|^2\right)$$

Summary

- Flow (or transportation) networks appear in many real-life applications - road traffic, utility distribution, computer networks, etc
- Flows on the network have finite capacity and materials are conserved
- Maximum flow problem involve finding a flow with the largest value.
  - Many algorithms have exists for solving the problem
    - Ford-Fulkerson method
    - Edmonds-Karps method