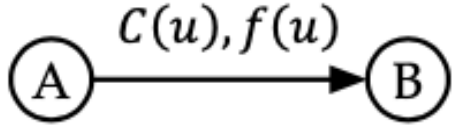
Part IV: Search for Optimal Flow

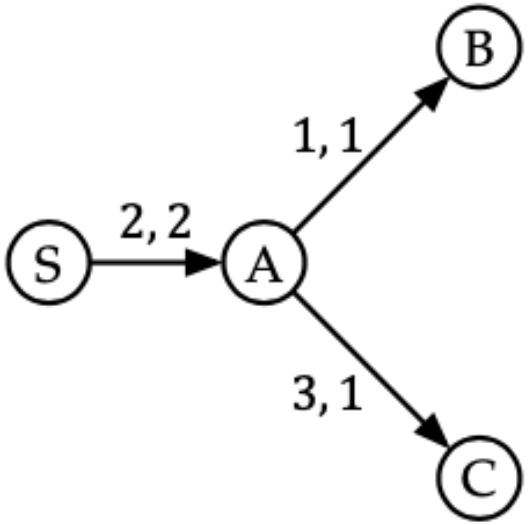
Transportation network

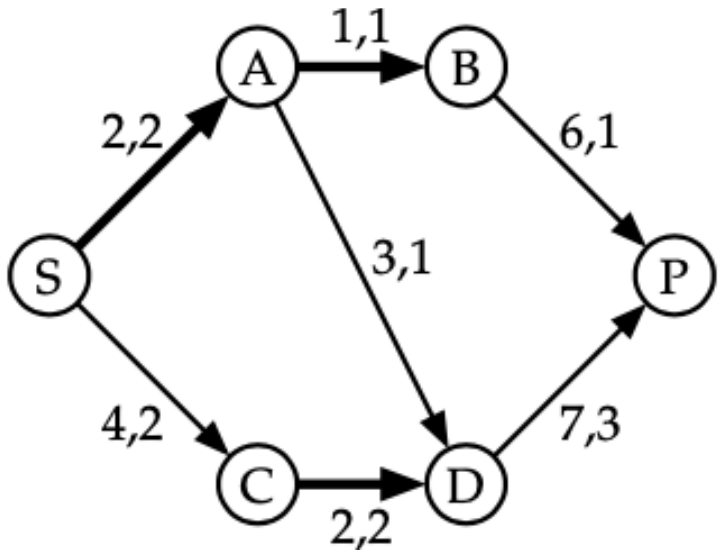
- **Transportation network (TN)** ? a directed and acyclic weighted graph used to model the movement of goods, data, or resources from one location (**source**) to another (**sink**) through intermediate nodes.
- In TN weights are positive quantity that indicates arcs **capacity** C (items could be carried / unit of time).
- TN has one vertex without any predecessor named **source** and one vertex without any successor named **sink.**
- TN may contain tow types of capacity min and max.

Flow



- **Flow** ? Function that assigns a quantity f(u) to each arc u, representing quantity of items passing through u per unit of time.
- Most flow problems deal with a **conservative flow** that follow **Kirchoff rule:** $\sum_{u \in \Gamma^-(x)} f(u) = \sum_{v \in \Gamma^+(x)} f(v)$
- A flow is **feasible** if it is conservative and does not exceed the capacity of each arc $0 \leq f(u) \leq C(u)$.
- For u∈E, u is said **saturated** if $f(u) = C(u)$.
- A flow is **maximum** if there is at least one saturated arc for each path from source to sink.





Flow problem

Solving the flow problem

Determining the optimal amount of items that can be transported through the arcs from the source to the sink

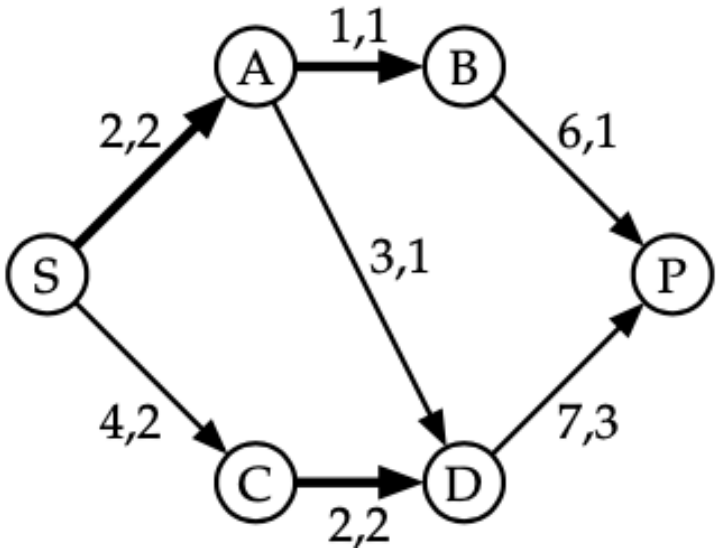
Ford – Fulkerson Algorithm

Principle

1. **Initialisation:** Begin with a feasible flow.
2. **Iterations:** find a path from source to sink in which flow could be increased.
3. **End:** if no increasable path.

Ford – Fulkerson Algorithm

- An **augmenting path** is a path (walk) from the source to the sink where additional flow can be pushed, increasing the total flow.
- A path p is augmenting if:
  - $f(u) < C(u)$, u is direct in p
  - $f(v) > 0$, v is reverse in p
- An augmenting path will increase the flow by:
  - **min($\min_{\text{u is direct}} C(u) - f(u)$, $\min_{\text{u is reverse}} f(u)$)**
- As the flow increase with t, the arcs' flows will be:
  - For direct arcs: $f(u) \leftarrow f(u)+t$
  - For reverse arcs: $f(v) \leftarrow f(v)-t$



Ford – Fulkerson Algorithm

for each arc u

f(u) = 0

Repeat

search an augmenting path p using marking algorithm

if u is direct: f(u) ← f(u) + min(augment)

if u is reverse: f(u) ← f(u) - min(augment)

Until no p is possible

mark the source

repeat until no possible marking

for each marked vertex x of arc u (x,y)

if f(u)<C(u) mark y

for each marked vertex y of arc u (x,y)

if f(u)>0 mark x

if the sink is marked

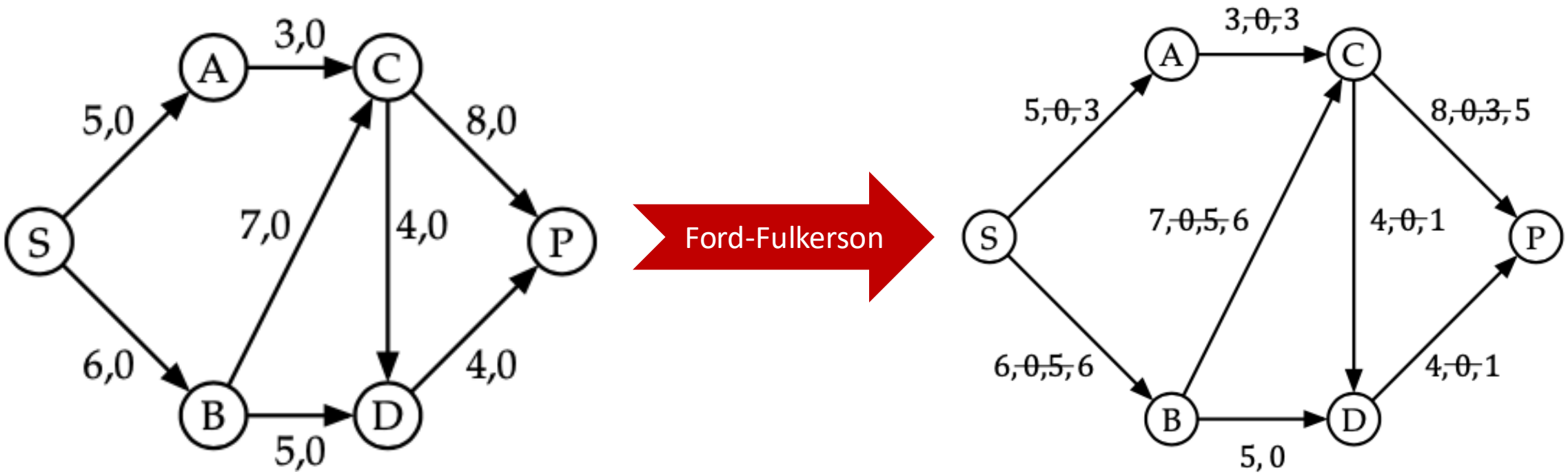
we have at least one augmenting path

else

there is no augmenting path

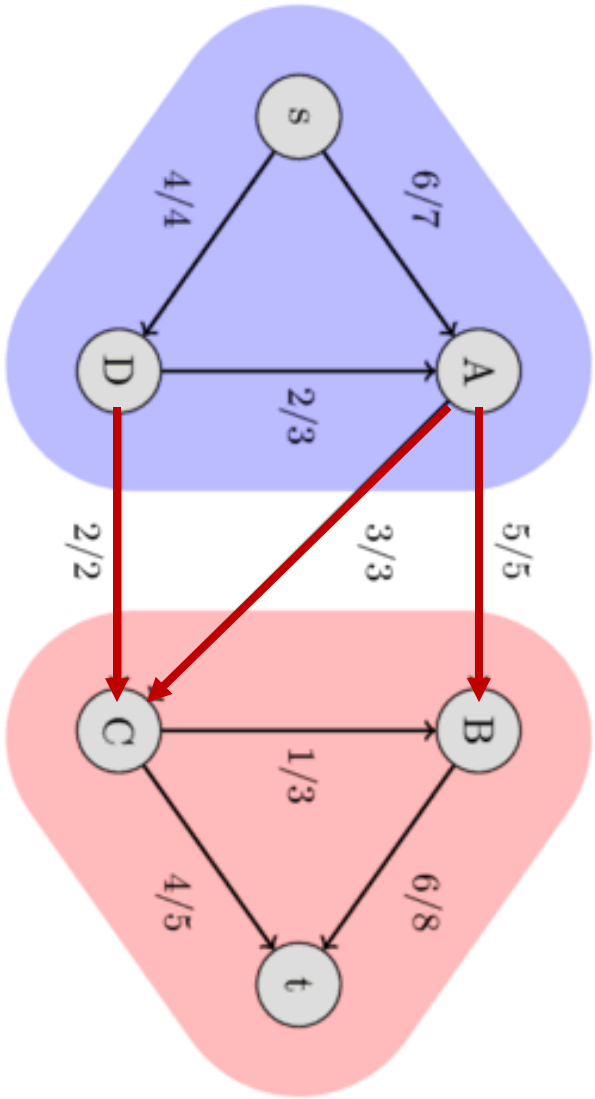
Marking algorithm

Example

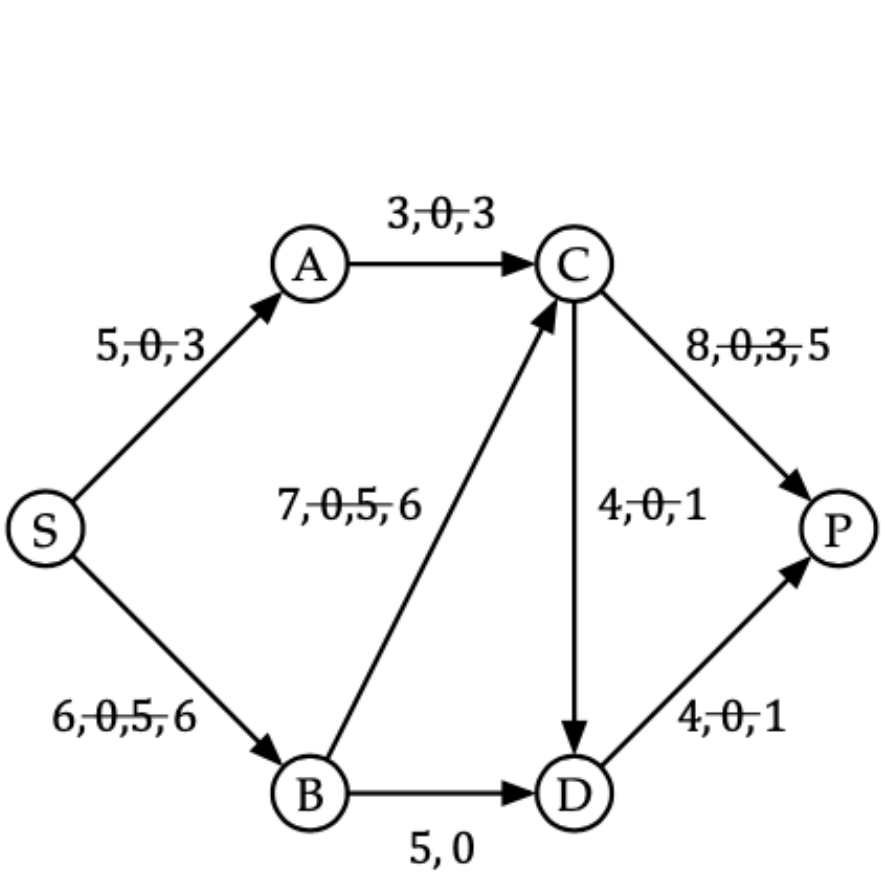

S
A
B
C
D
P
5,0
6,0
3,0
7,0
4,0
8,0
4,0
5,0
Ford-Fulkerson
S
A
B
C
D
P
5, 0, 3
6, 0, 5, 6
3, 0, 3
7, 0, 5, 6
4, 0, 1
8, 0, 3, 5
4, 0, 1
5, 0

Cut

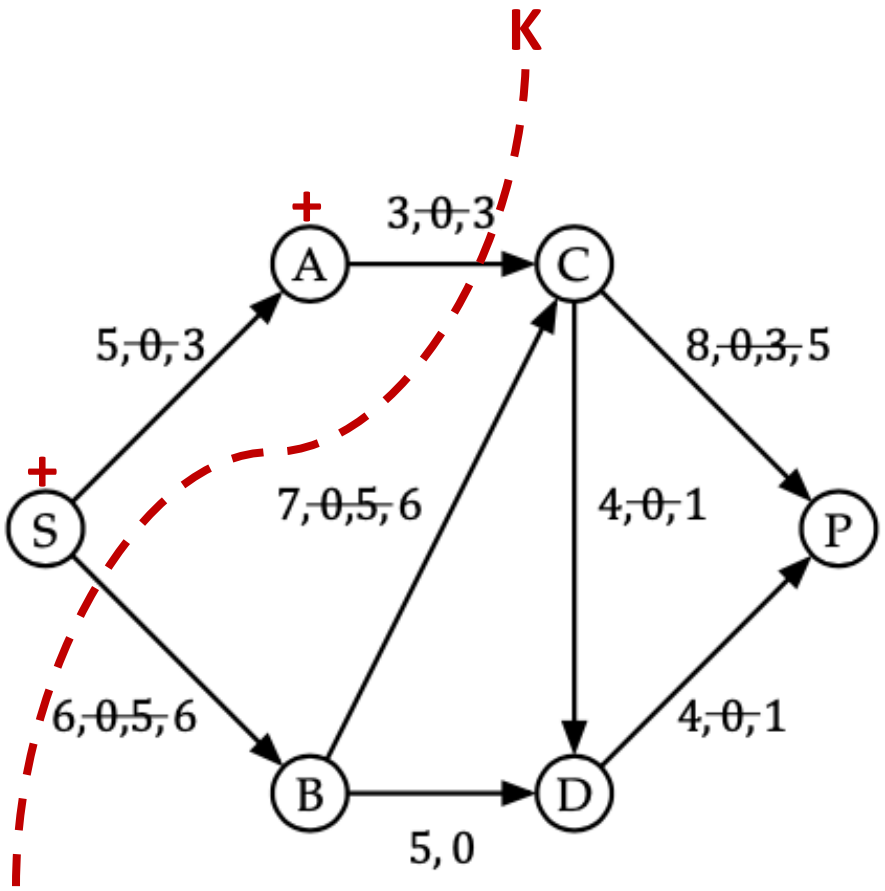
- **Cut** ? partition of the vertices V in two subsets X, Y (X∪Y=V & X∩Y=∅).
  - **It may also be defined by arcs with sources in X & targets in Y.**
- **Capacity of a cut** ? Sum of the capacities of all its arcs
  - If k = {(xi,yi)}, $C(K) = \sum_i C((xi, yi))$
- **Minimum cut** ? The cut with the min capacity
  - Marked & un-marked vertices after the last iteration of the marking algorithm form the min cut.
- **Theorem**: in a TN, max flow = capacity of min cut



Cut



Marking



Max flow = 3+6 = 8+1 = 9

Capacity(K) = 3+6 = 9

Applications - Load Balancing

Optimizing Traffic Distribution with ECMP and Path Selection

- **ECMP**: Equal-Cost Multi-Path routing splits traffic across paths with identical metrics (e.g., hop count).
- **Advanced Path Selection**: Techniques like WCMP (weighted) and dynamic load balancing for unequal paths.
- **How ?** ECMP splits traffic across multiple paths of equal cost to ensure flows stay consistent. Advanced techniques like WCMP assign traffic proportionally to link capacities. Protocols like OSPF and BGP enable multi-path discovery.

Applications - QoS Routing & RSVP

Guaranteeing Performance with QoS Routing and RSVP

- **QoS Routing**: Path selection based on constraints (bandwidth, latency).
- **RSVP**: Resource Reservation Protocol for end-to-end bandwidth guarantees.
- **How ?** QoS routing selects paths that meet constraints like bandwidth and latency using algorithms like Constrained Shortest Path First (CSPF). RSVP reserves resources by exchanging Path/Resv messages between sender and receiver, ensuring bandwidth and buffer allocation for critical flows.

Applications - Google's B4 SDN Backbone

Revolutionizing WAN Traffic with Google's B4 SDN

- **SDN-Driven WAN**: Centralized traffic engineering (TE) for global load balancing.
- **Bandwidth Allocation**: Prioritizes application classes (e.g., user data vs. backups).
- **How ?** Google's B4 uses a centralized SDN controller to optimize traffic across its global WAN. The controller employs OpenFlow to program switches based on real-time network views, partitioning bandwidth for application classes (e.g., prioritizing user-facing traffic over backups).