Mathematical Tools for Engineering and Management

Lecture 9

14 Dec 2011

- Models, Data and Algorithms
- Linear Optimization
- Mathematical Background: Polyhedra, Simplex-Algorithm
- Sensitivity Analysis; (Mixed) Integer Programming
- MIP Modelling
- MIP Modelling: More Examples; Branch & Bound
- Cutting Planes; Combinatorial Optimization: Examples, Graphs, Algorithms
- TSP-Heuristics
- Network Flows
- Shortest Path Problem, Complexity Theory
- Nonlinear Optimization
- Scheduling, Lot Sizing
- Multicriteria Optimization
- Oral exam

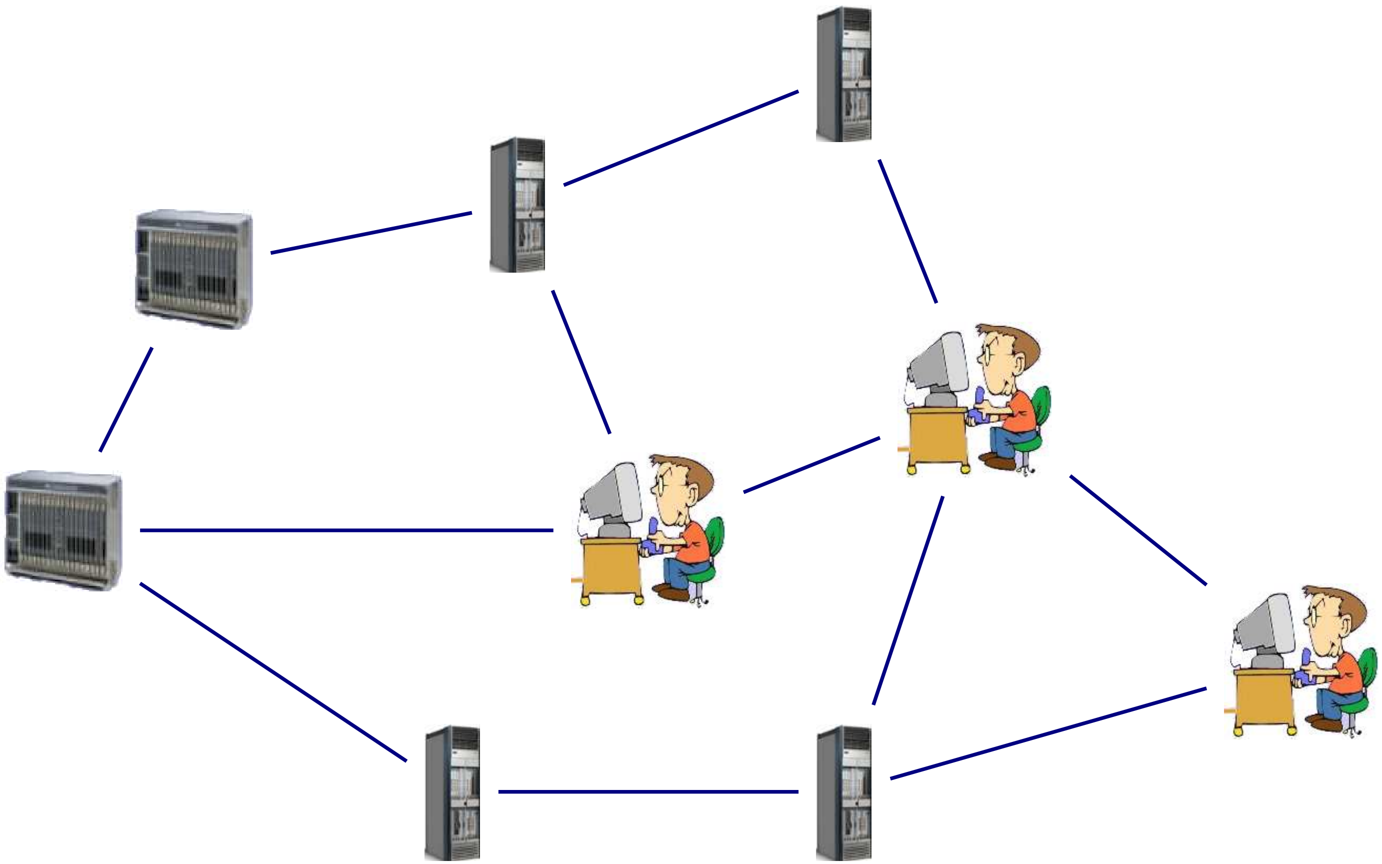
- Consider a data network with

Example: data flow

- Consider a data network with central offices , routers  and users 

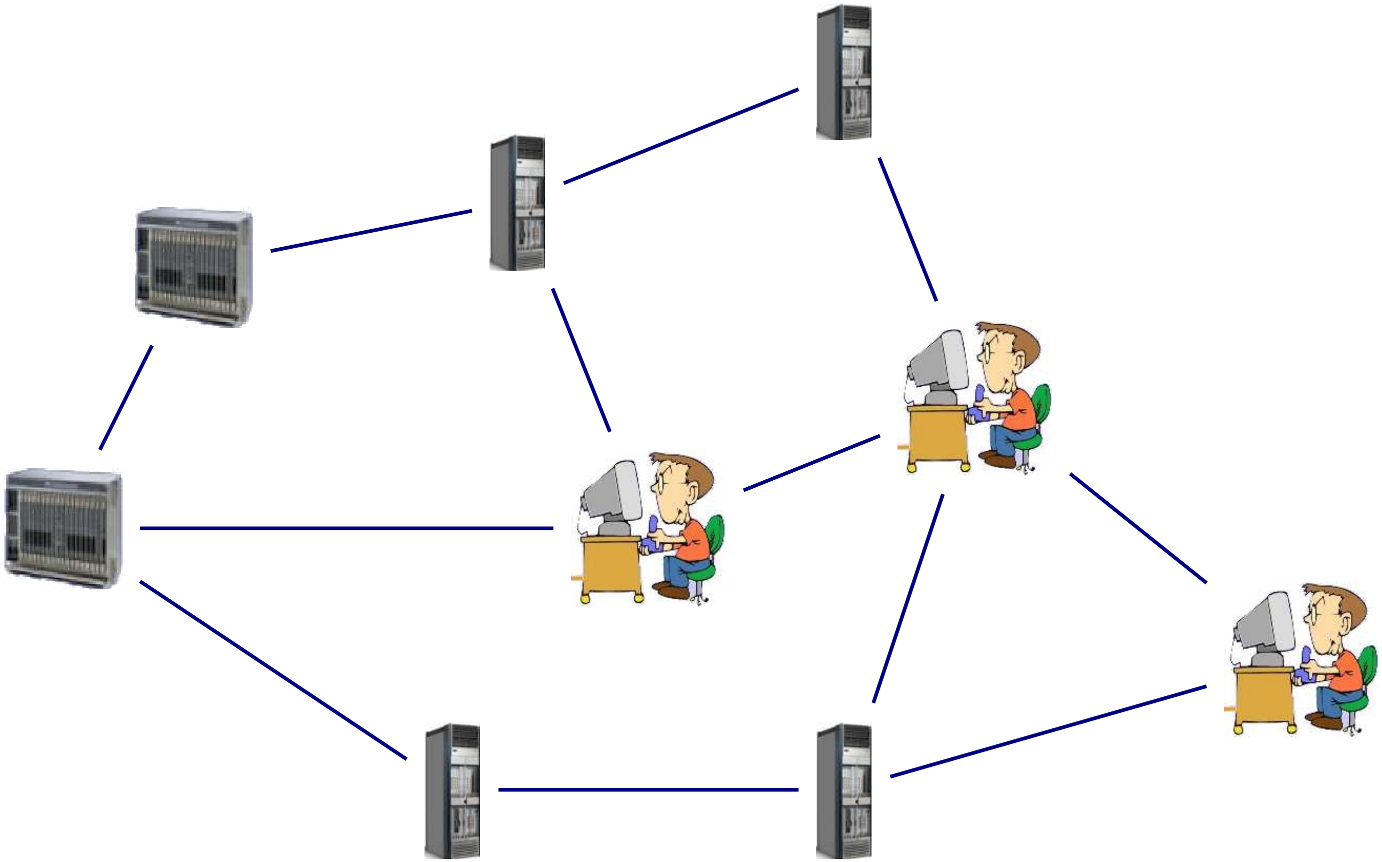
Example: data flow

- Consider a data network with central offices , routers  and users 



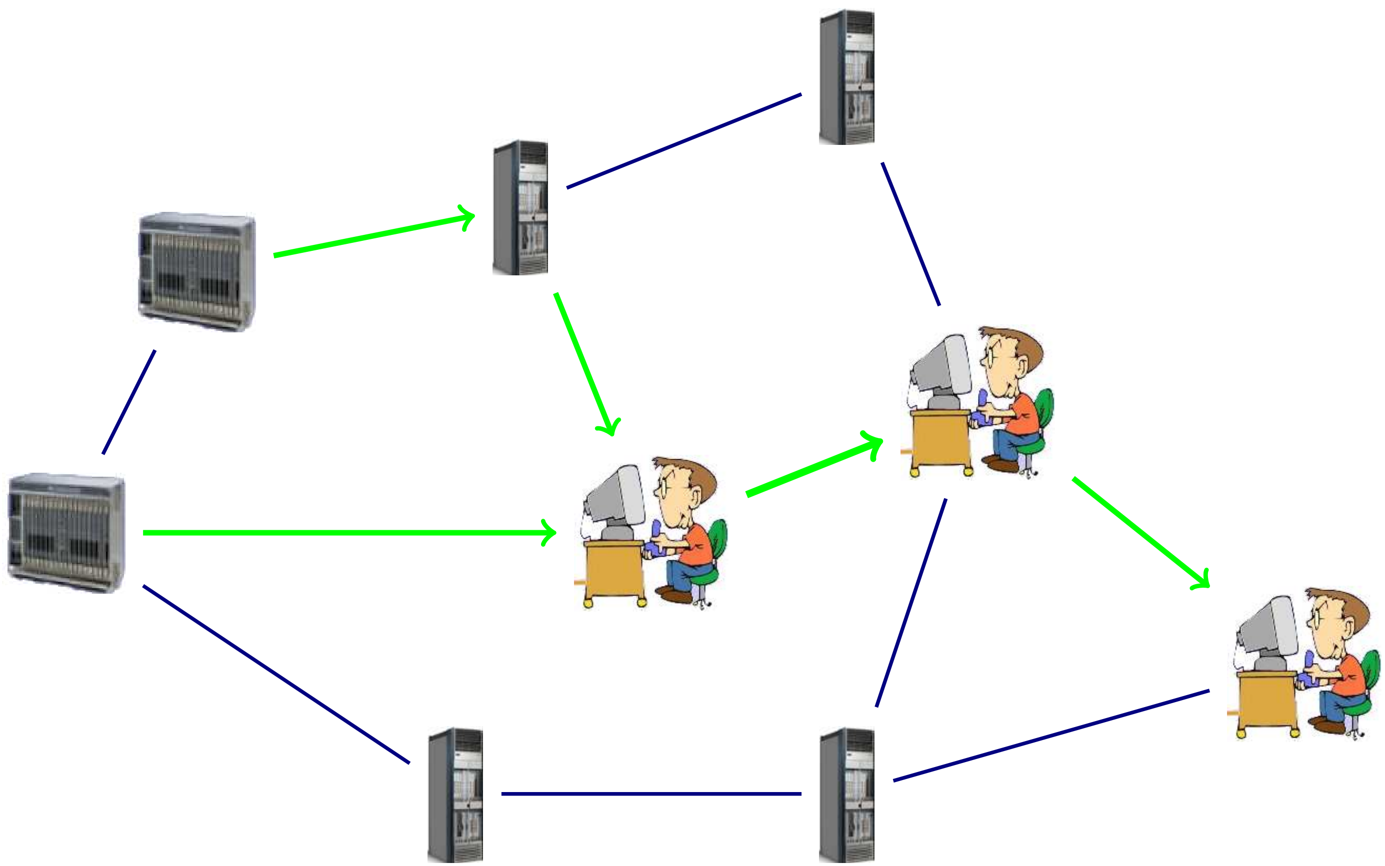
Example: data flow

- Consider a data network with central offices, routers and users
- Data has to be sent from the central offices to the users via the network



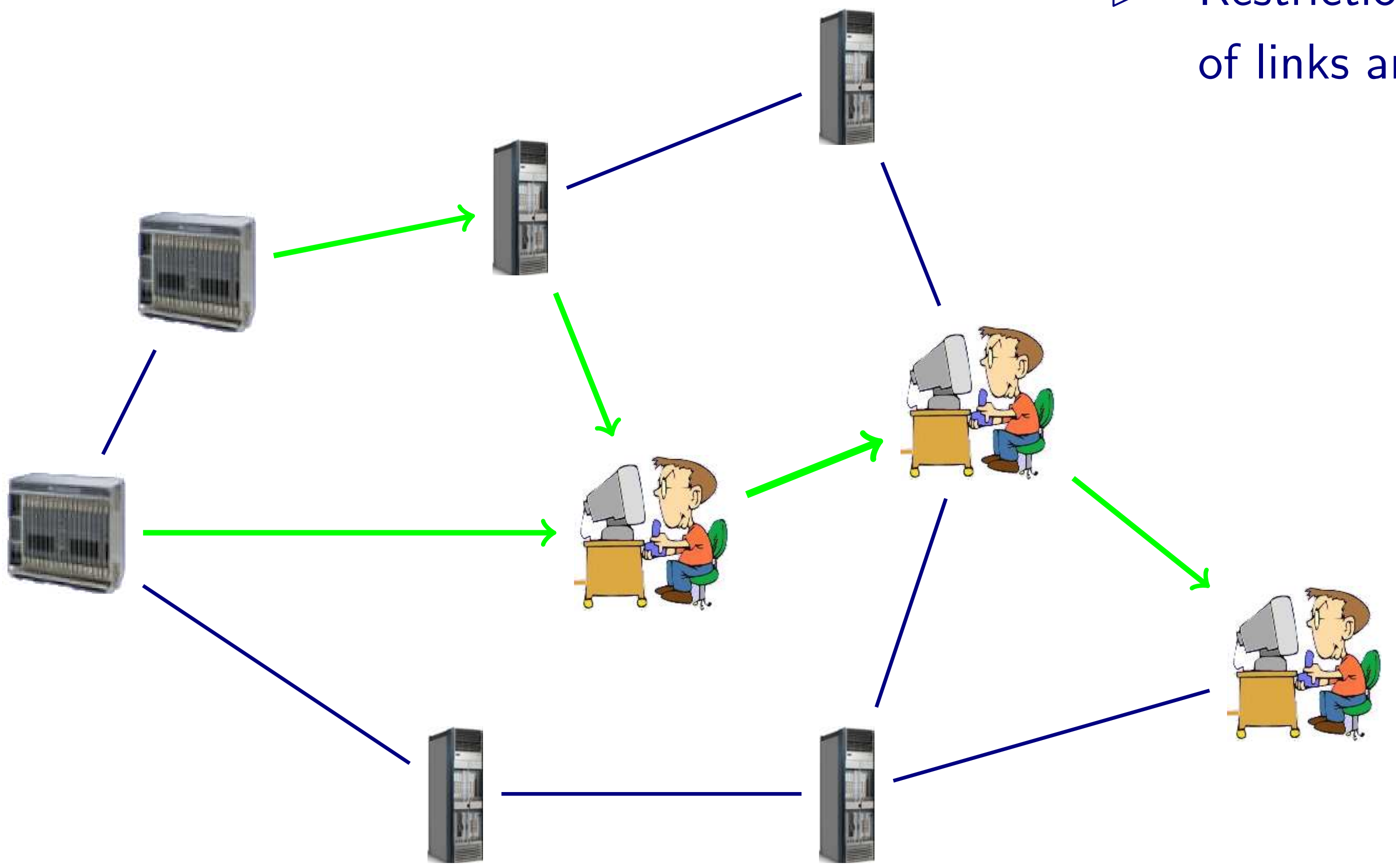
Example: data flow

- Consider a data network with central offices , routers  and users 
- Data has to be sent from the central offices to the users via the network



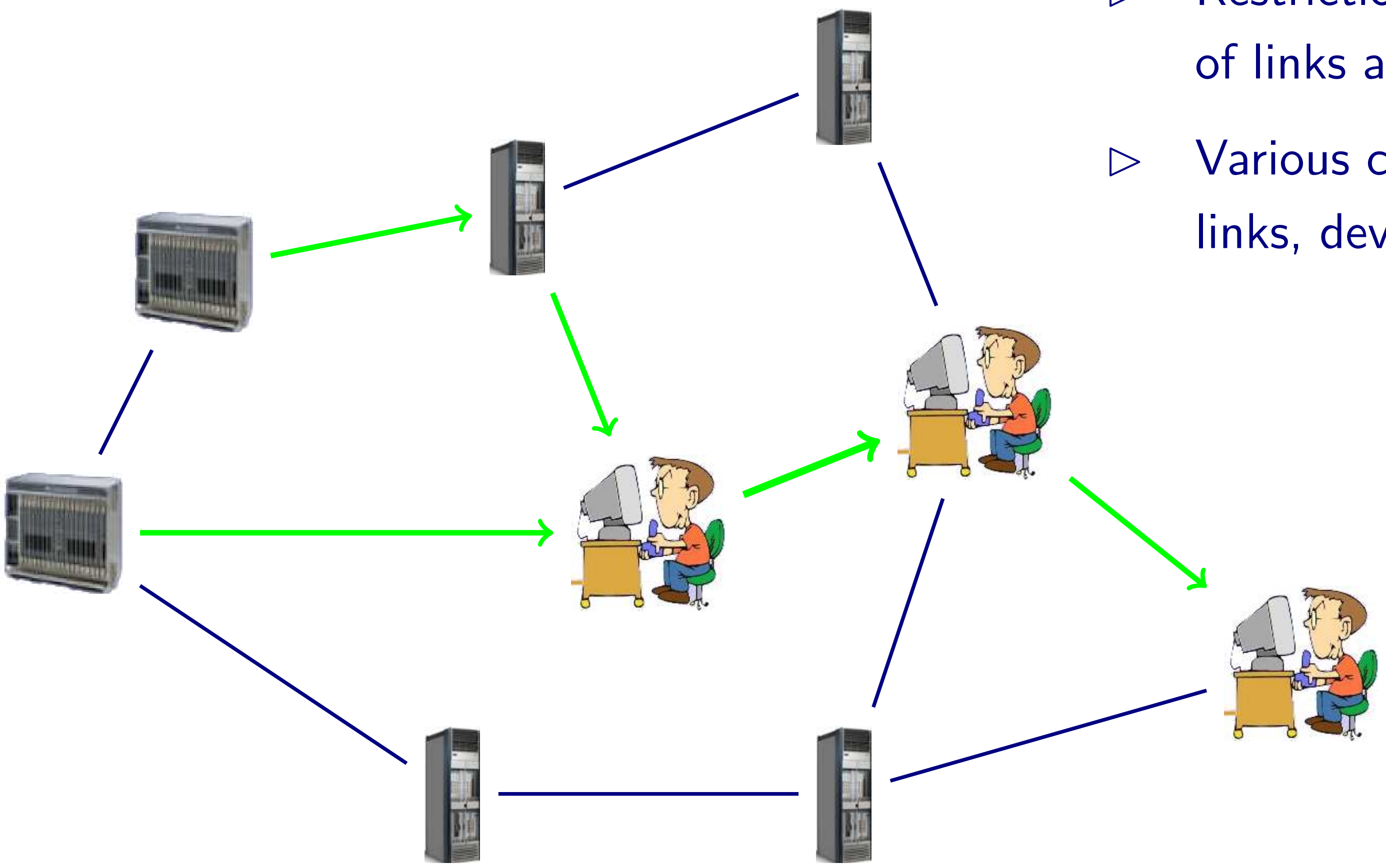
Example: data flow

- Consider a data network with central offices , routers  and users 
- Data has to be sent from the central offices to the users via the network



- Restrictions given by capacities of links and devices

- ▷ Consider a data network with central offices , routers  and users 
- ▷ Data has to be sent from the central offices to the users via the network

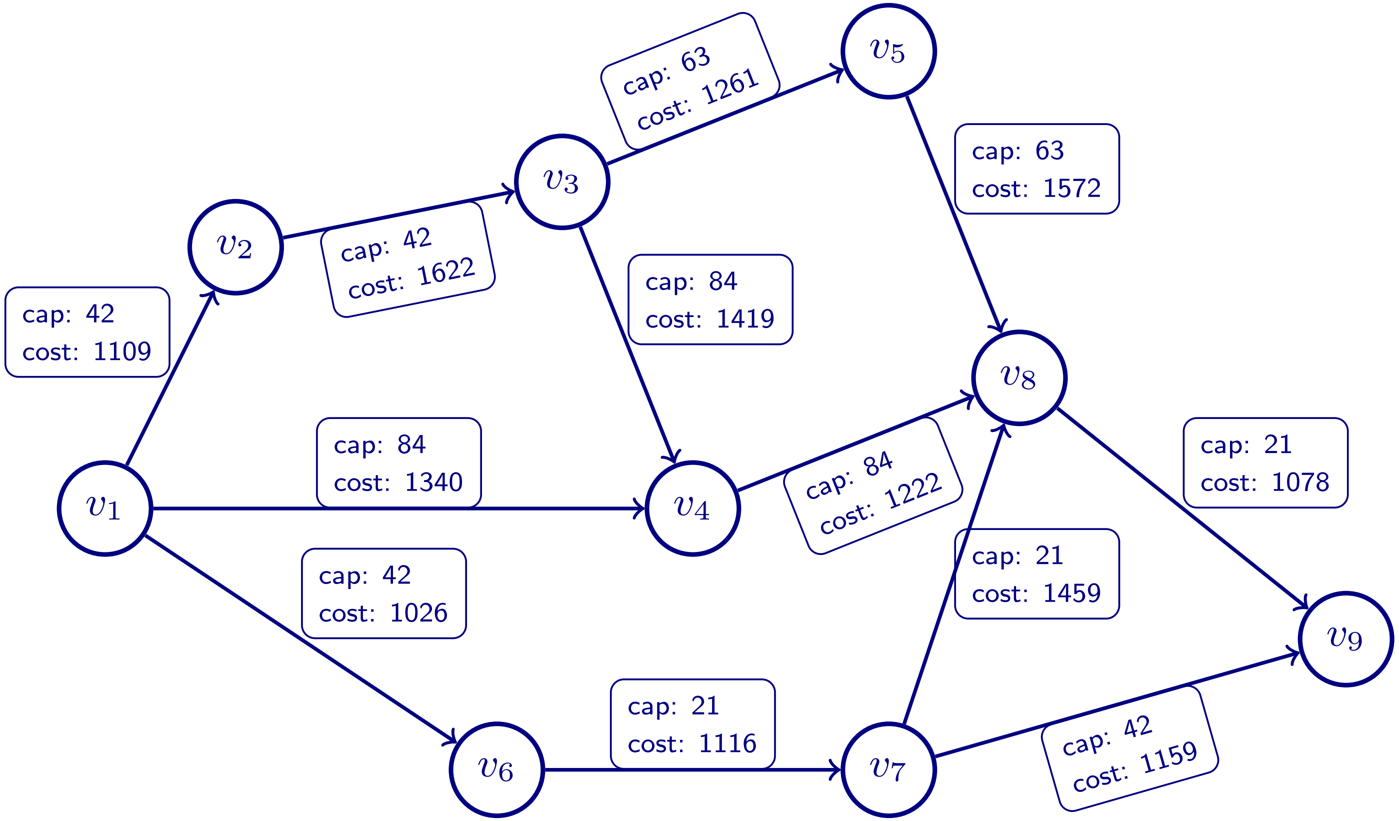


- ▷ Restrictions given by capacities of links and devices
- ▷ Various costs depending on links, devices and data volume

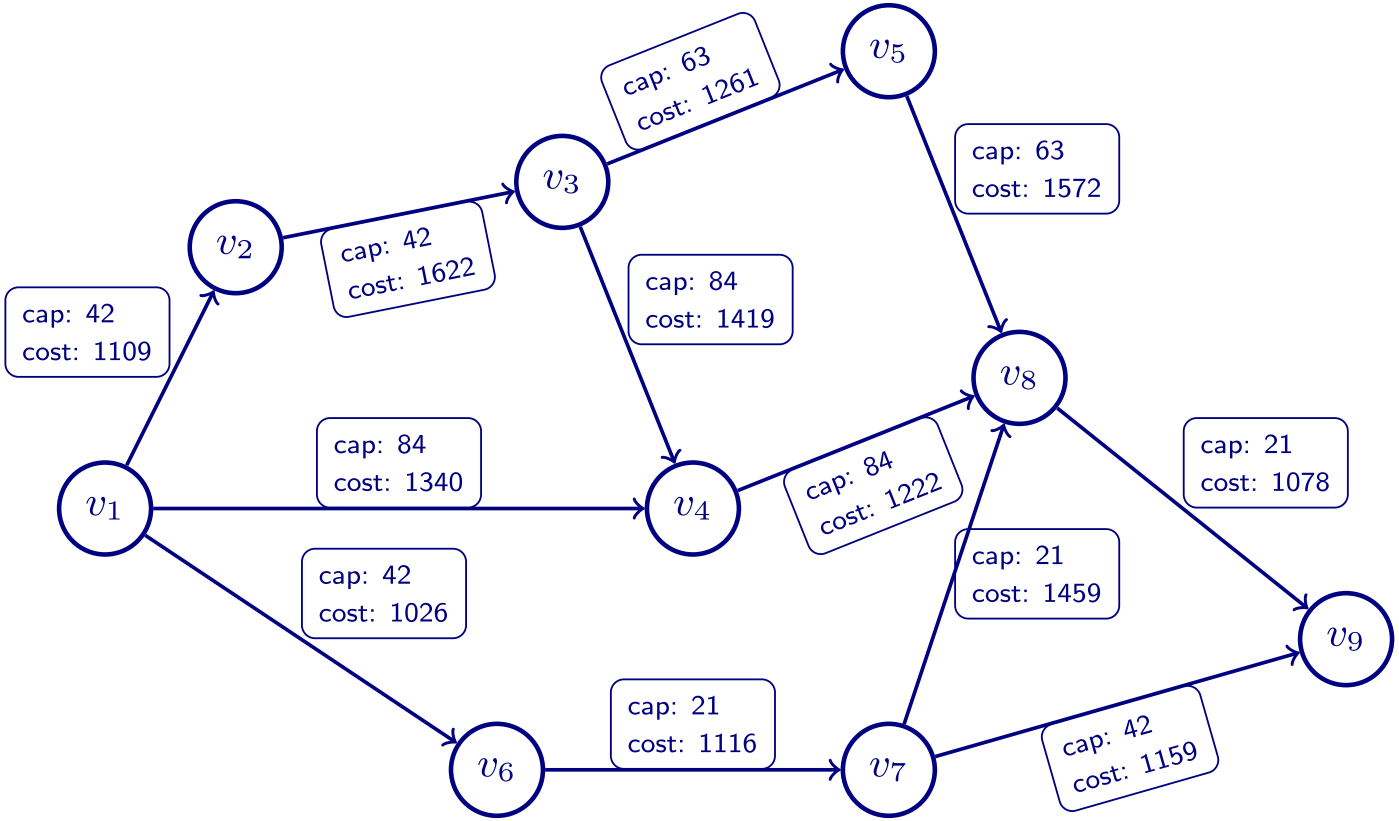
- Given a network – i.e. a directed graph

- ▷ Given a network – i.e. a directed graph, possibly with more parameters, such as capacities and costs for nodes and arcs

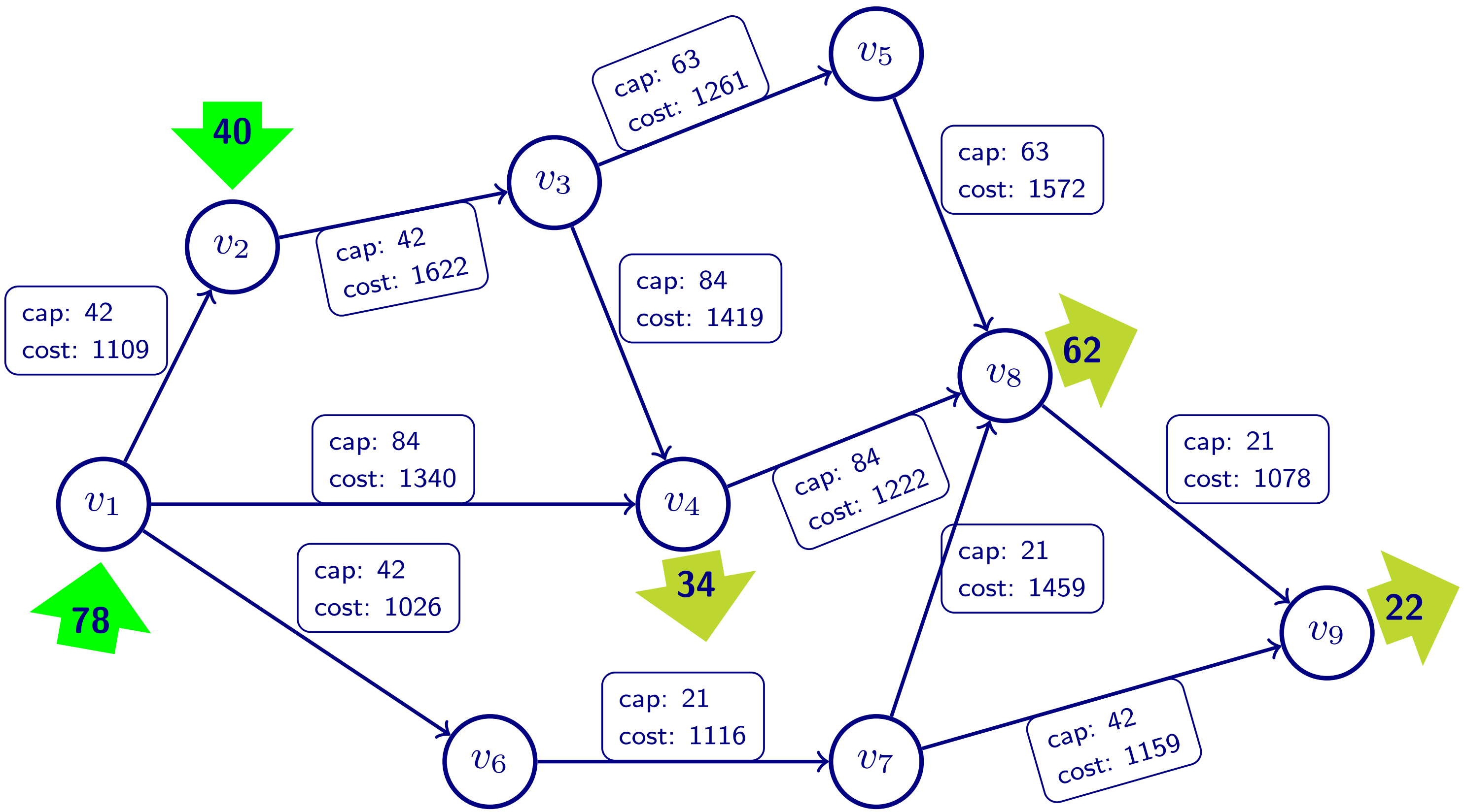
- Given a network – i.e. a directed graph, possibly with more parameters, such as capacities and costs for nodes and arcs



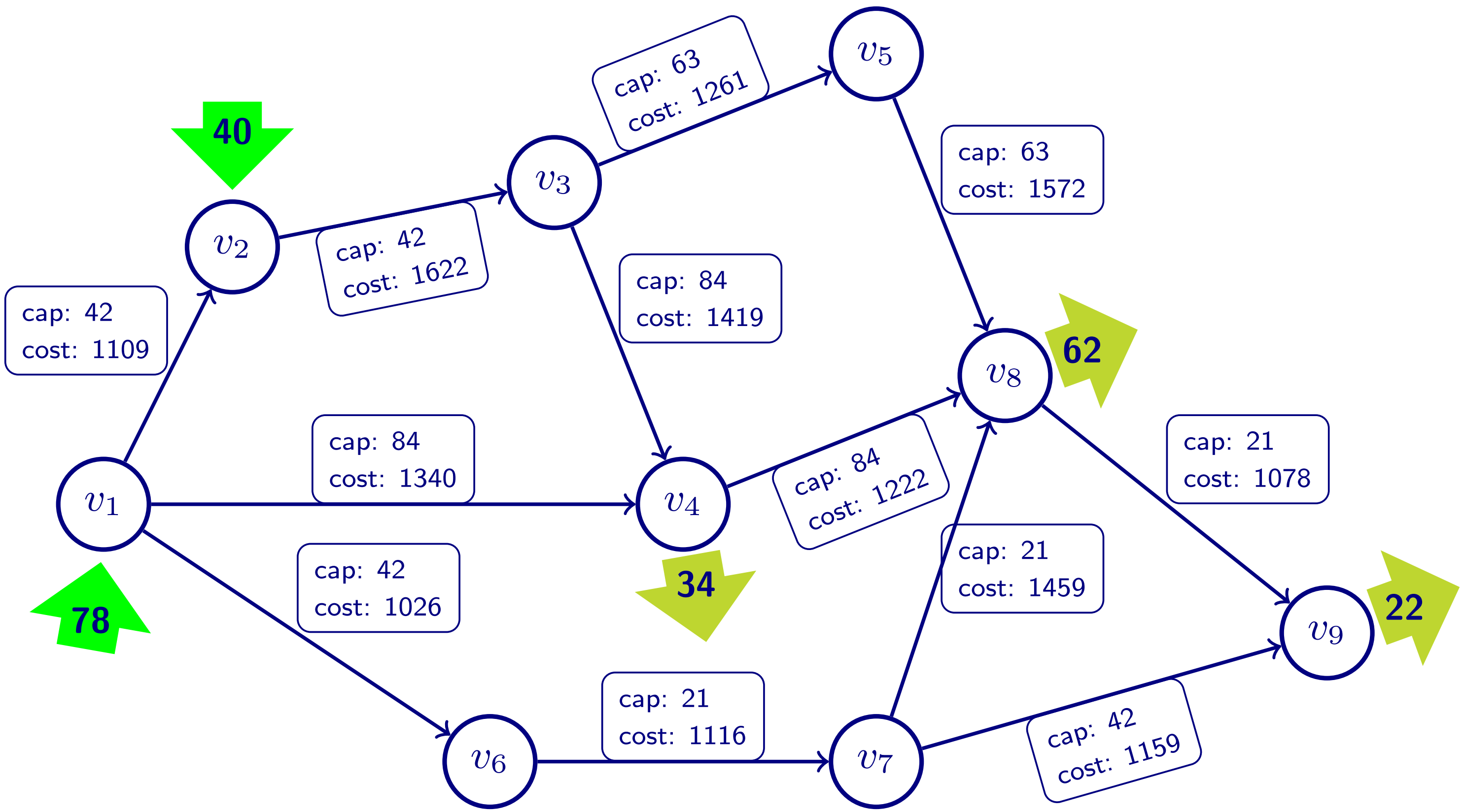
- Given a network – i.e. a directed graph, possibly with more parameters, such as capacities and costs for nodes and arcs – and certain demands...



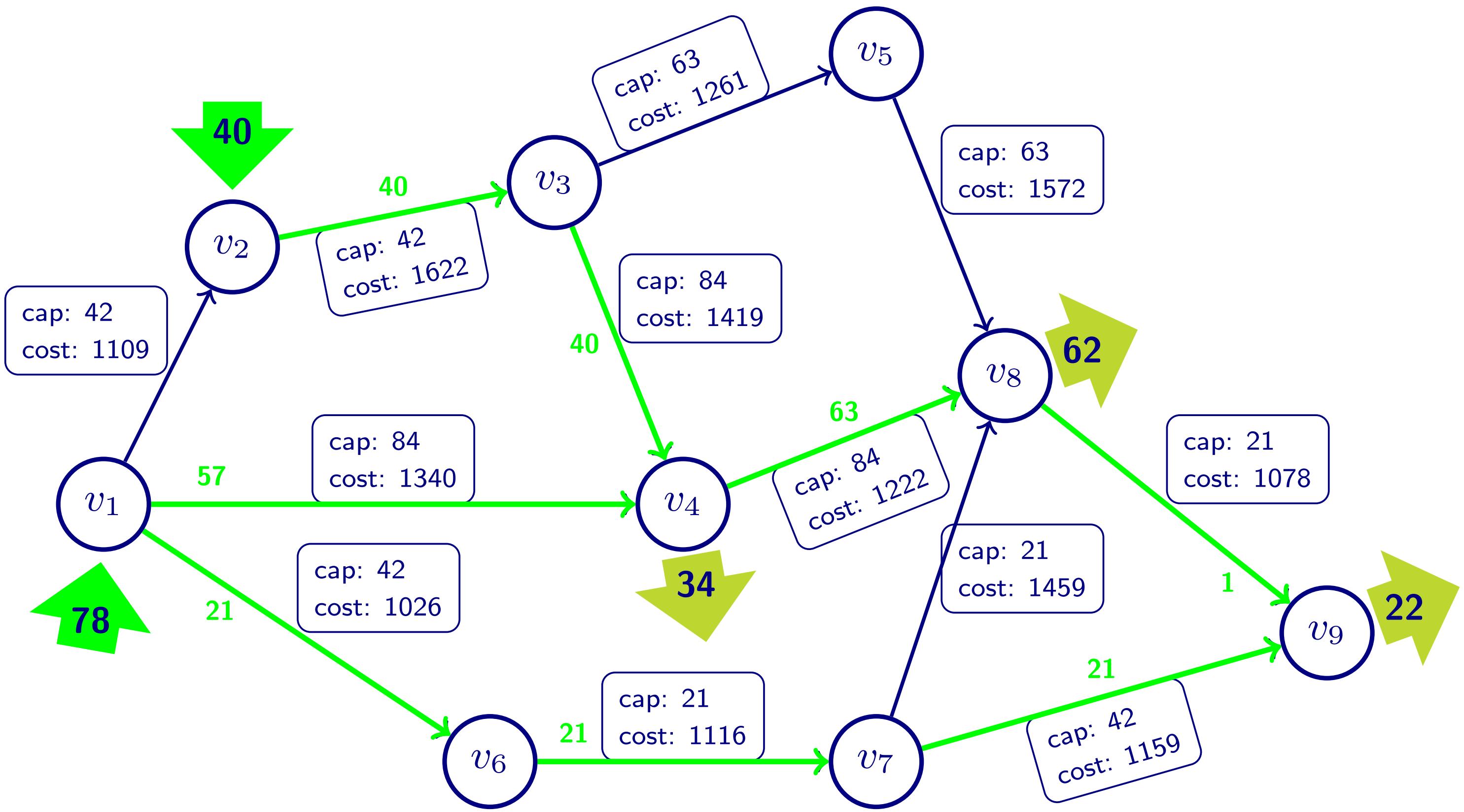
- Given a network – i.e. a directed graph, possibly with more parameters, such as capacities and costs for nodes and arcs – and certain demands...



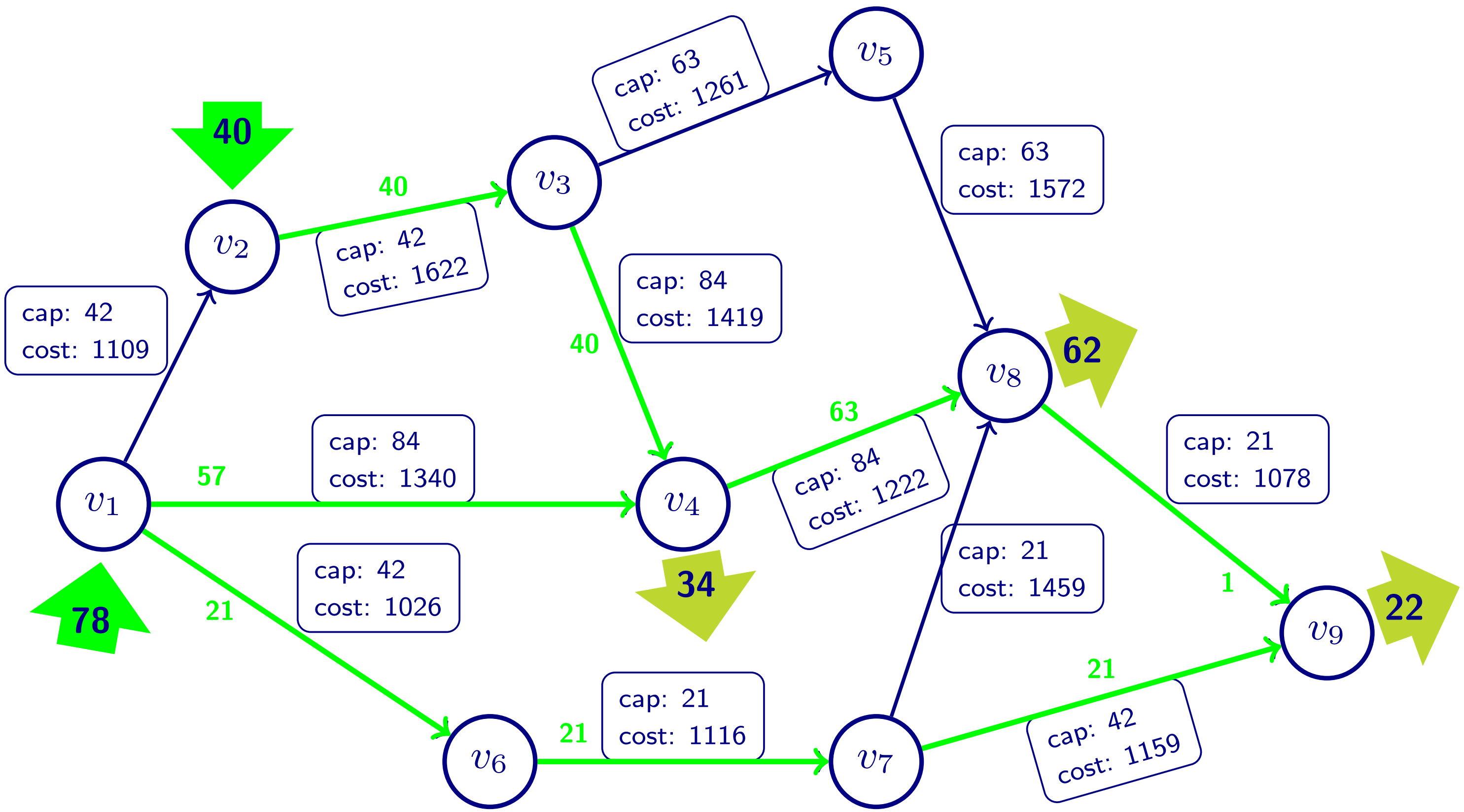
- Given a network – i.e. a directed graph, possibly with more parameters, such as capacities and costs for nodes and arcs – and certain demands...
- ...compute a flow through the network satisfying the demand



- Given a network – i.e. a directed graph, possibly with more parameters, such as capacities and costs for nodes and arcs – and certain demands...
- ...compute a flow through the network satisfying the demand



- Given a network – i.e. a directed graph, possibly with more parameters, such as capacities and costs for nodes and arcs – and certain demands...
- ...compute a flow through the network satisfying the demand, respecting the capacities, with minimal total cost

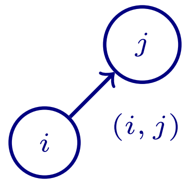


- Network: directed graph (V, A)

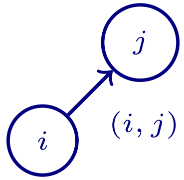
- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)

Flow variables & capacities

- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)
- Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)

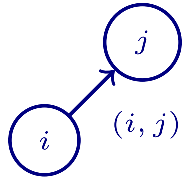


- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)
- Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)



S

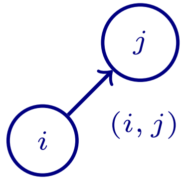
- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)
- Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)



S

- Variables: $f_{(i,j)} \geq 0$ ➡ amount of flow along the arc (i, j)

- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)
- Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)
- Variables: $f_{(i,j)} \geq 0$ ➡ amount of flow along the arc (i, j)

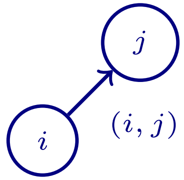


S

V

Flow variables & capacities

- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)
- Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)
- Variables: $f_{(i,j)} \geq 0$ ➡ amount of flow along the arc (i, j)
- Capacity constraints: $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i, j) \in A$



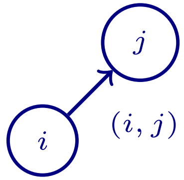
S

V

C

Flow variables & capacities

- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)
- Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)



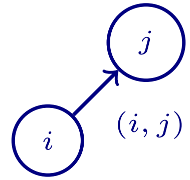
- Variables: $f_{(i,j)} \geq 0$ ➡ amount of flow along the arc (i, j)
- Capacity constraints: $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i, j) \in A$

$u_{(i,j)}$ ➡ capacity of link (i, j)

S V C P

Flow variables & capacities

- ▷ Network: directed graph (V, A)
- ▷ Nodes of the network: $i \in V$ (vertices of the graph)
- ▷ Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)



S

- ▷ Variables: $f_{(i,j)} \geq 0$ ➡ amount of flow along the arc (i, j)

V

- ▷ Capacity constraints: $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i, j) \in A$

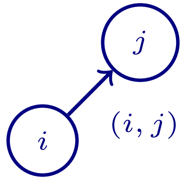
C

$u_{(i,j)}$ ➡ capacity of link (i, j)

P

- ▷ Possibly: constraints given by node capacities...

- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)
- Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)



S

V

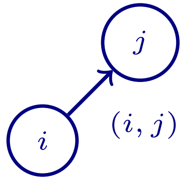
C

P

- Variables: $f_{(i,j)} \geq 0$ ➡ amount of flow along the arc (i, j)
- Capacity constraints: $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i, j) \in A$

  $u_{(i,j)}$ ➡ capacity of link (i, j)
- Possibly: constraints given by node capacities...
- **Objective**: minimize total cost $\sum_{(i,j) \in A} c_{(i,j)} \cdot f_{(i,j)}$

- Network: directed graph (V, A)
- Nodes of the network: $i \in V$ (vertices of the graph)



S

- Links of the network: $(i, j) \in A$ (arcs (edges) of the graph)
- Variables: $f_{(i,j)} \geq 0$ ➡ amount of flow along the arc (i, j)

V

- Capacity constraints: $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i, j) \in A$

C

$u_{(i,j)}$ ➡ capacity of link (i, j)

P

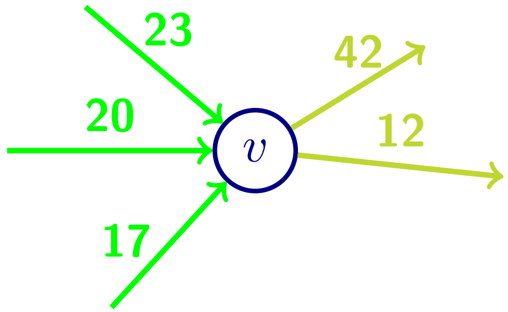
- Possibly: constraints given by node capacities...
- **Objective**: minimize total cost $\displaystyle\sum_{(i,j)\in A} c_{(i,j)} \cdot f_{(i,j)}$

$c_{(i,j)}$ ➡ cost per flow unit on arc (i, j)

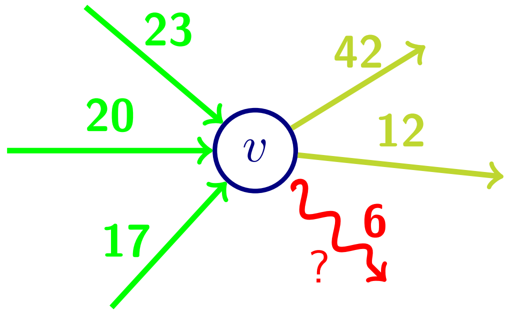
P

- Flow may not "leak" from the network!

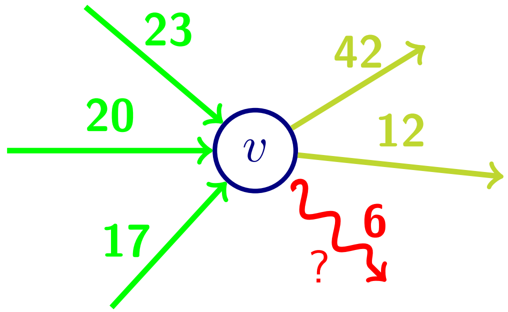
- Flow may not "leak" from the network!



- Flow may not “leak” from the network!



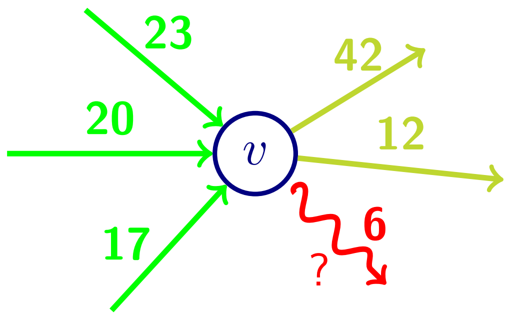
- Flow may not "leak" from the network!
  - ➡ outgoing flow must equal incoming flow

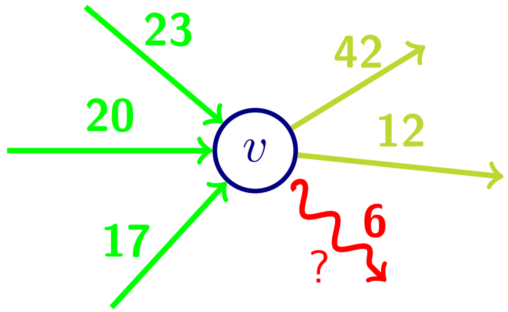


▷ Flow may not "leak" from the network!

➡ outgoing flow must equal incoming flow

➡ $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = 0 \quad \text{for all } v \in V$$





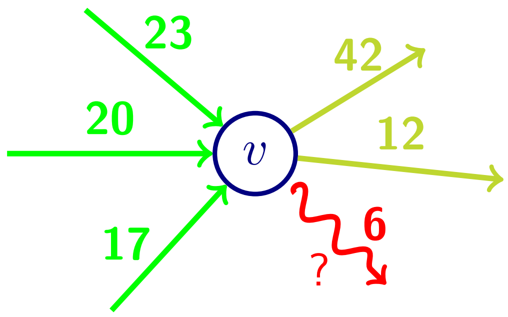
▷ Flow may not "leak" from the network!

➡ outgoing flow must equal incoming flow

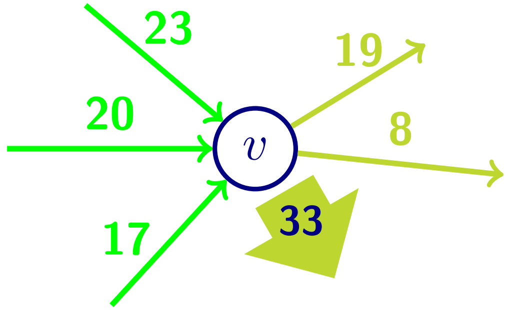
➡ $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = 0 \quad \text{for all } v \in V$$

▷ ...except for nodes with a demand d_v

- ▷ Flow may not "leak" from the network!
  - ➡ outgoing flow must equal incoming flow
  - ➡ $\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = 0$ for all $v \in V$



- ▷ ...except for nodes with a demand d_v

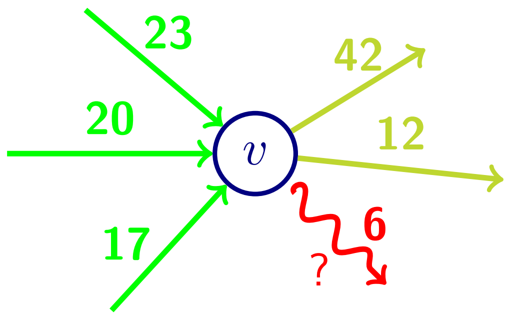


Flow conservation constraints

▷ Flow may not "leak" from the network!

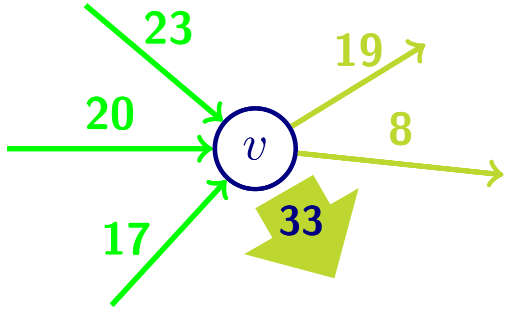
➡ outgoing flow must equal incoming flow

➡ $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = 0 \quad \text{for all } v \in V$$



▷ ...except for nodes with a demand d_v

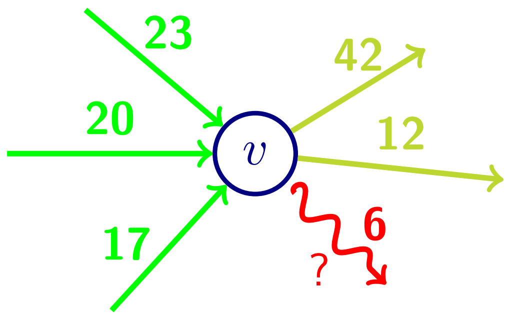
➡ $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = -d_v \quad \text{for all } v \in V$$



- Flow may not "leak" from the network!

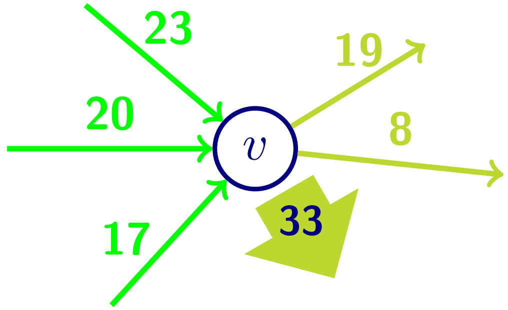
  ➡ outgoing flow must equal incoming flow

  ➡ $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = 0 \quad \text{for all } v \in V$$



- ...except for nodes with a demand d_v

  ➡ $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = -d_v \quad \text{for all } v \in V$$

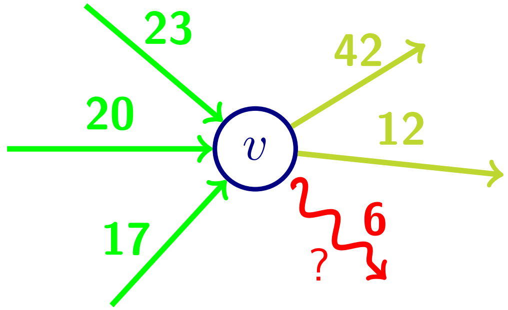


- In general:

$$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = \begin{cases} -d_v & \text{if } v \text{ is a demand node} \\ s_v & \text{if } v \text{ is a supply node} \\ 0 & \text{otherwise} \end{cases}$$

C

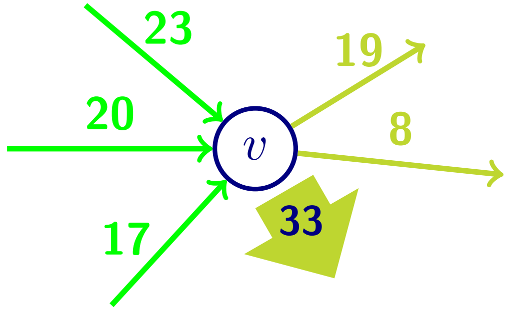
- Flow may not "leak" from the network!



  ➡ outgoing flow must equal incoming flow

  ➡ $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = 0 \quad \text{for all } v \in V$$

- ...except for nodes with a demand d_v



  ➡ $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = -d_v \quad \text{for all } v \in V$$

- In general:

$$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = \begin{cases} -d_v & \text{if } v \text{ is a demand node} \\ s_v & \text{if } v \text{ is a supply node} \\ 0 & \text{otherwise} \end{cases}$$

C

d_v ➡ demand of node v

s_v ➡ supply of node v

P

Objective: minimize (total cost) $\sum_{(i,j)\in A} c_{(i,j)} \cdot f_{(i,j)}$

Objective: minimize (total cost) $\sum_{(i,j) \in A} c_{(i,j)} \cdot f_{(i,j)}$

C (link capacity) $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i,j) \in A$

Objective: minimize (total cost) $\sum_{(i,j)\in A} c_{(i,j)} \cdot f_{(i,j)}$

C

(link capacity) $f_{(i,j)} \le u_{(i,j)}$ for all arcs $(i,j) \in A$

(flow conservation) $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = \begin{cases} -d_v & \text{if } v \text{ is a demand node} \\ s_v & \text{if } v \text{ is a supply node} \\ 0 & \text{otherwise} \end{cases}$$

Objective: minimize (total cost) $\sum_{(i,j)\in A} c_{(i,j)} \cdot f_{(i,j)}$

C (link capacity) $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i,j) \in A$

(flow conservation) $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = \begin{cases} -d_v & \text{if } v \text{ is a demand node} \\ s_v & \text{if } v \text{ is a supply node} \\ 0 & \text{otherwise} \end{cases}$$

V (flow on arcs) $f_{(i,j)} \geq 0$ for all arcs $(i,j) \in A$

Objective: minimize (total cost) $\sum_{(i,j)\in A} c_{(i,j)} \cdot f_{(i,j)}$

C (link capacity) $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i,j) \in A$

(flow conservation) $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = \begin{cases} -d_v & \text{if } v \text{ is a demand node} \\ s_v & \text{if } v \text{ is a supply node} \\ 0 & \text{otherwise} \end{cases}$$

V (flow on arcs) $f_{(i,j)} \geq 0$ for all arcs $(i,j) \in A$

S Nodes: V

Arcs: $A \subseteq \{(i,j) \mid i,j \in V,\, i \neq j\}$

Objective: minimize (total cost) $\sum_{(i,j)\in A} c_{(i,j)} \cdot f_{(i,j)}$

C (link capacity) $f_{(i,j)} \leq u_{(i,j)}$ for all arcs $(i,j) \in A$

(flow conservation) $$\sum_{(v,j)\in A} f_{(v,j)} - \sum_{(i,v)\in A} f_{(i,v)} = \begin{cases} -d_v & \text{if } v \text{ is a demand node} \\ s_v & \text{if } v \text{ is a supply node} \\ 0 & \text{otherwise} \end{cases}$$

V (flow on arcs) $f_{(i,j)} \geq 0$ for all arcs $(i,j) \in A$

S Nodes: V

Arcs: $A \subseteq \{(i,j) \mid i,j \in V, i \neq j\}$

P Link capacities: $u_{(i,j)} \geq 0$ for all arcs $(i,j) \in A$

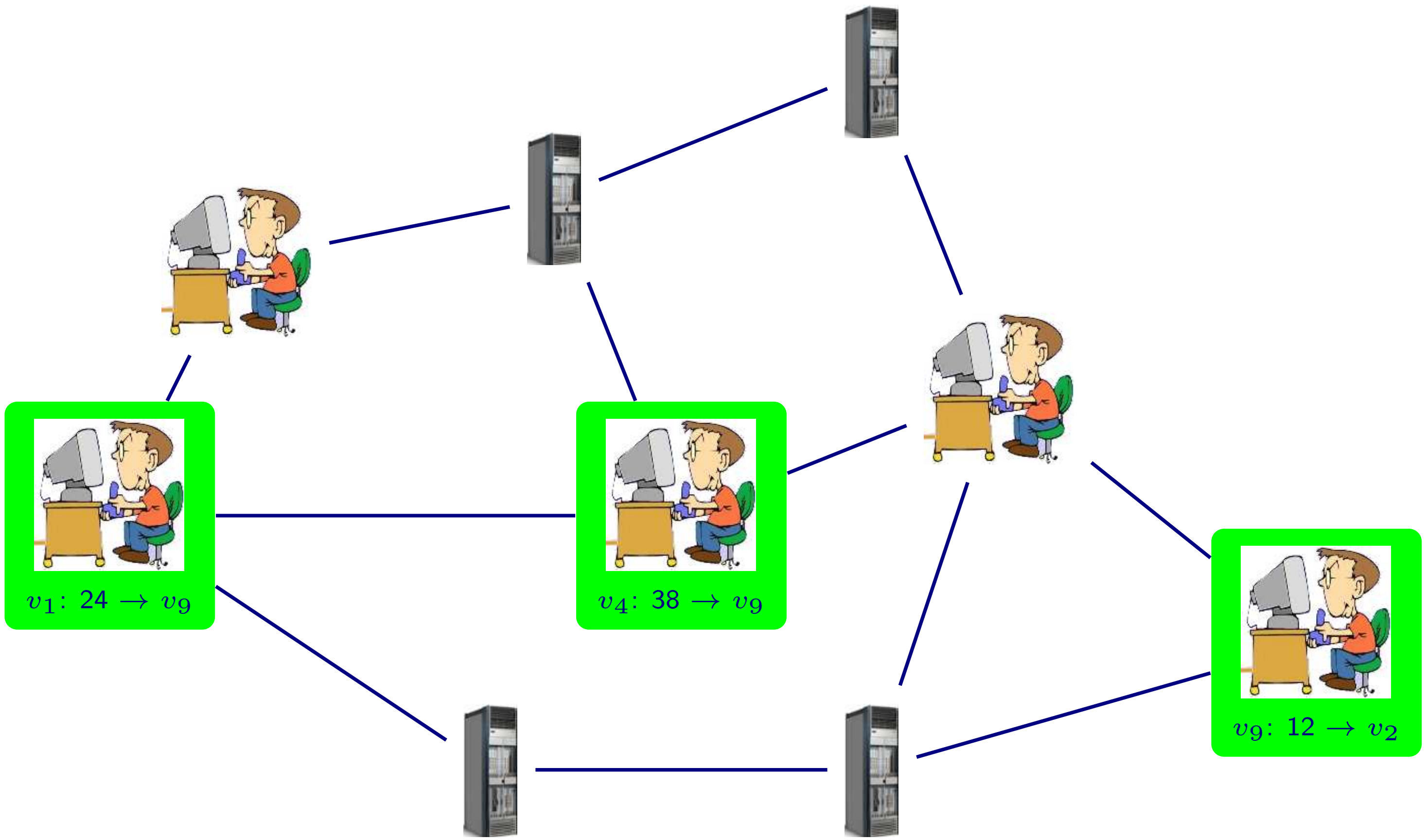
Costs: $c_{(i,j)}$ for all arcs $(i,j) \in A$

Demand of demand nodes: $d_v \geq 0$

Supply of supply nodes: $s_v \geq 0$

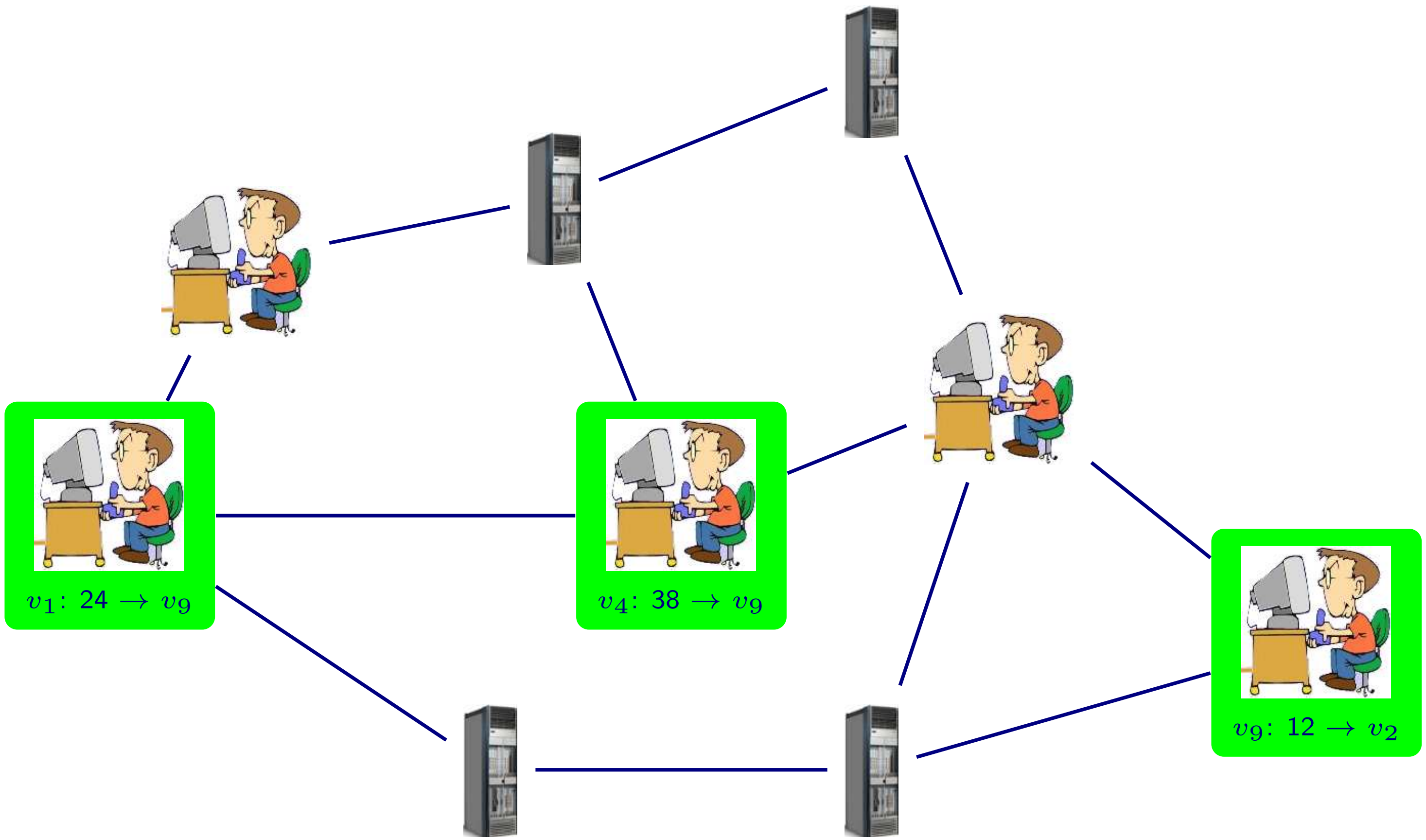
- Data are IP packets with a source and a target node for each packet

- Data are IP packets with a source and a target node for each packet

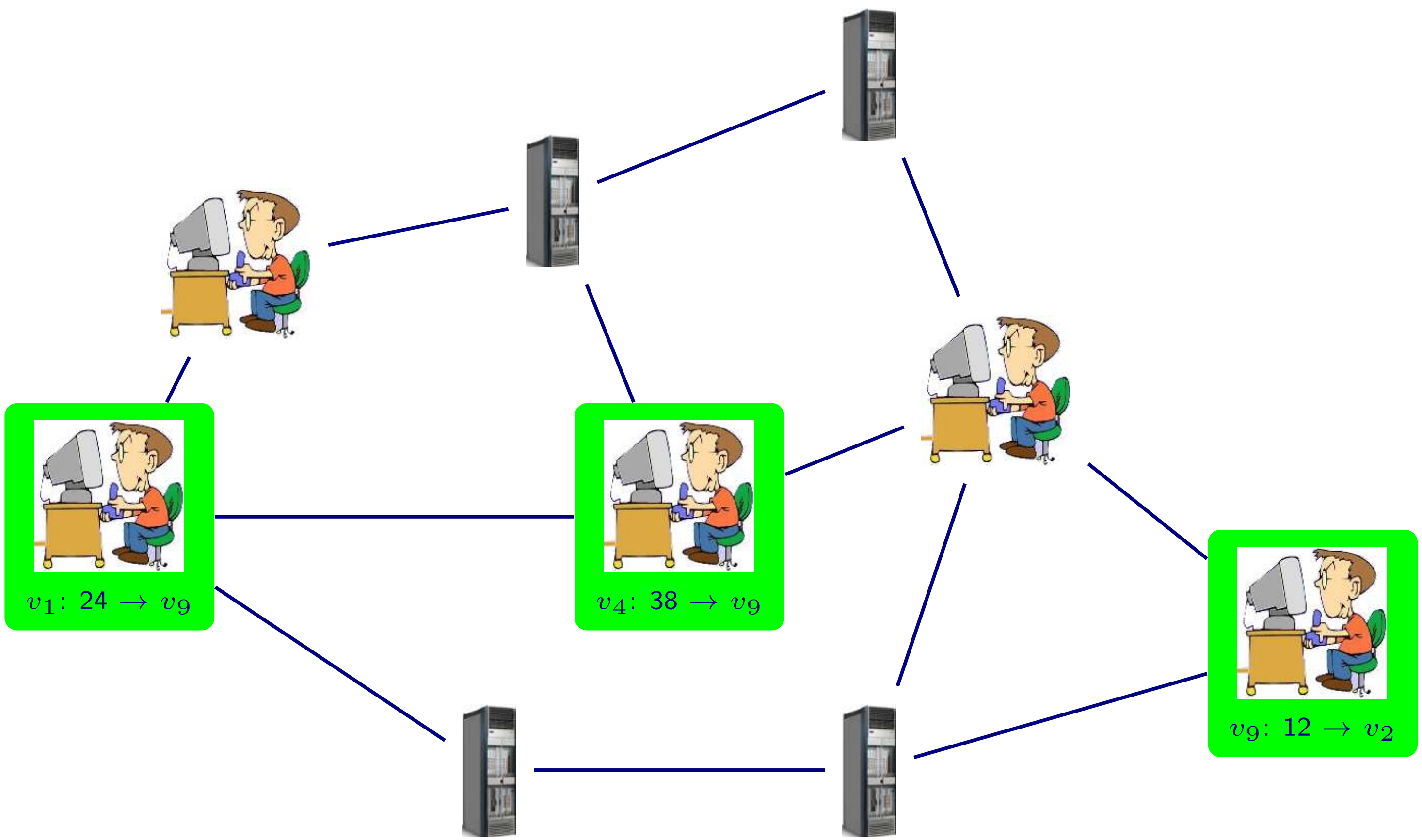


▷ Data are IP packets with a source and a target node for each packet

➡ Separate (disaggregated) flow for every pair of nodes



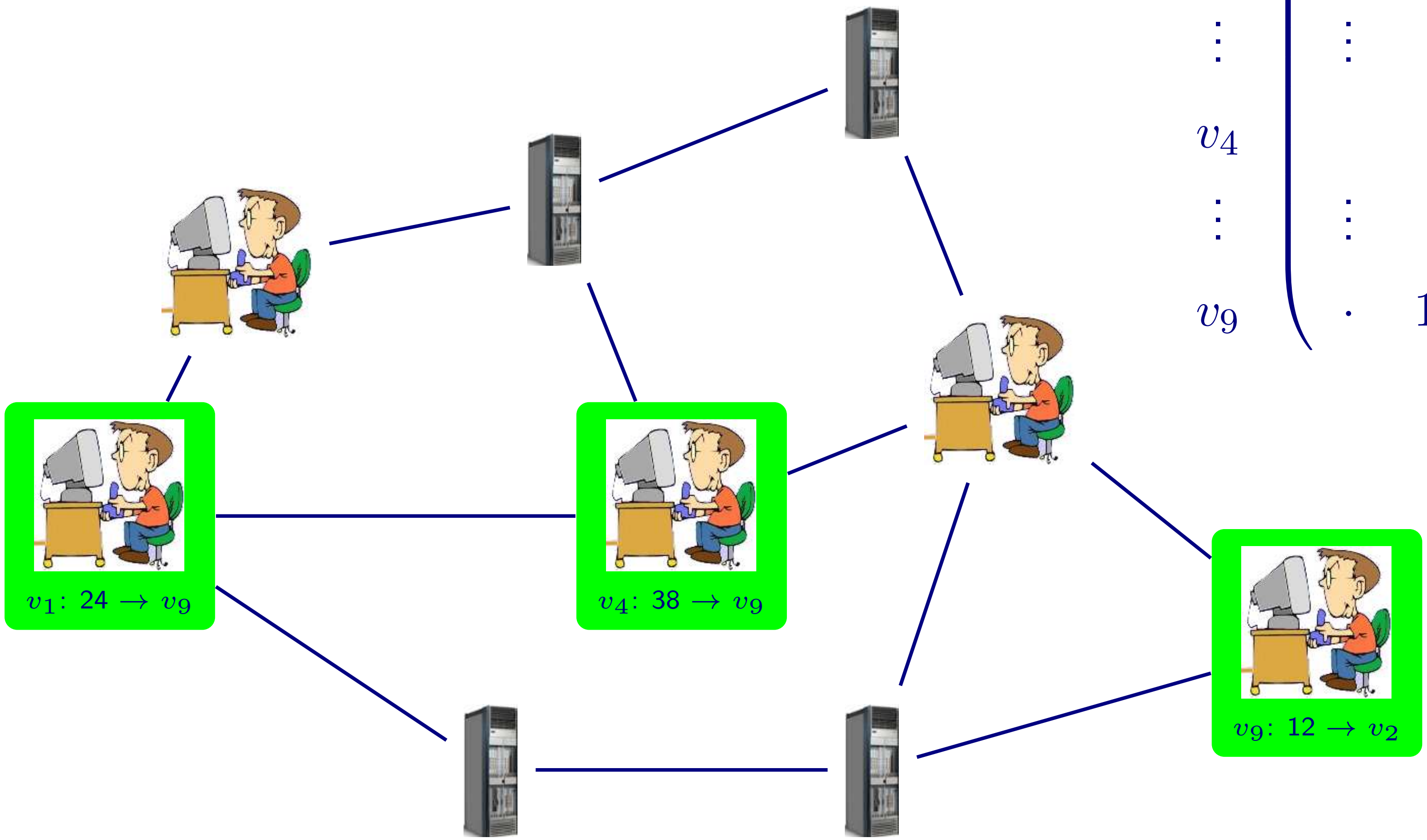
- ▷ Data are IP packets with a source and a target node for each packet
  - ➡ Separate (disaggregated) flow for every pair of nodes
  - ➡ Demand matrix with entries $d_{u,v} \geq 0$



Example: IP routing

- Data are IP packets with a source and a target node for each packet
  - ➡ Separate (disaggregated) flow for every pair of nodes
  - ➡ Demand matrix with entries $d_{u,v} \geq 0$

$$
\begin{array}{c} \\ v_1 \\ \vdots \\ v_4 \\ \vdots \\ v_9 \end{array}
\begin{array}{c}
\begin{array}{ccccc} v_1 & v_2 & \cdots & & v_9 \end{array} \\
\begin{pmatrix}
0 & \cdots & & \cdots & 24 \\
\vdots & 0 & & & \vdots \\
 & & & & 38 \\
\vdots & & & \ddots & \vdots \\
\cdot & 12 & & \cdots & 0
\end{pmatrix}
\end{array}
$$



- Disaggregated flow variables: $f_{(i,j),u\to v} \geq 0$

- ▷ Disaggregated flow variables: $f_{(i,j),u \to v} \geq 0$

  ➡ amount of flow for demand $u \to v$ along the arc (i, j)

V

- Disaggregated flow variables: $f_{(i,j),u \to v} \geq 0$

  ➡ amount of flow for demand $u \to v$ along the arc (i, j)

V

- **Objective**: minimize total cost $\sum_{(i,j) \in A} c_{(i,j)} \cdot \sum_{u \neq v} f_{(i,j),u \to v}$

- Disaggregated flow variables: $f_{(i,j),u \to v} \geq 0$ **V**
  - ➡ amount of flow for demand $u \to v$ along the arc (i,j)
- **Objective**: minimize total cost $\sum_{(i,j) \in A} c_{(i,j)} \cdot \sum_{u \neq v} f_{(i,j),u \to v}$
- Capacity constraints: $\sum_{u \neq v} f_{(i,j),u \to v} \leq u_{(i,j)}$ for all arcs $(i,j) \in A$ **C**

- Disaggregated flow variables: $f_{(i,j),u \to v} \geq 0$

  ➡ amount of flow for demand $u \to v$ along the arc (i,j)

V

- **Objective**: minimize total cost $\sum_{(i,j) \in A} c_{(i,j)} \cdot \sum_{u \neq v} f_{(i,j),u \to v}$

- Capacity constraints: $\sum_{u \neq v} f_{(i,j),u \to v} \leq u_{(i,j)}$ for all arcs $(i,j) \in A$

C

- Flow conservation constraints: for all $w \in V$, $u, v \in V, u \neq v$,

$$\sum_{(w,j) \in A} f_{(w,j),u \to v} - \sum_{(i,w) \in A} f_{(i,w),u \to v} = \begin{cases} d_{u,v} & \text{if } w = u \\ -d_{u,v} & \text{if } w = v \\ 0 & \text{otherwise} \end{cases}$$

- Disaggregated flow variables: $f_{(i,j),u \to v} \geq 0$ **V**

  ➡ amount of flow for demand $u \to v$ along the arc (i,j)

- **Objective**: minimize total cost $\sum_{(i,j) \in A} c_{(i,j)} \cdot \sum_{u \neq v} f_{(i,j),u \to v}$

- Capacity constraints: $\sum_{u \neq v} f_{(i,j),u \to v} \leq u_{(i,j)}$ for all arcs $(i,j) \in A$ **C**

- Flow conservation constraints: for all $w \in V$, $u, v \in V, u \neq v$,

$$\sum_{(w,j) \in A} f_{(w,j),u \to v} - \sum_{(i,w) \in A} f_{(i,w),u \to v} = \begin{cases} d_{u,v} & \text{if } w = u \\ -d_{u,v} & \text{if } w = v \\ 0 & \text{otherwise} \end{cases}$$

- NOTE: flow for a given demand might be split at nodes!

V

▷ Disaggregated flow variables: $f_{(i,j),u\to v} \geq 0$

➡ amount of flow for demand $u \to v$ along the arc (i,j)

▷ **Objective**: minimize total cost $\sum_{(i,j)\in A} c_{(i,j)} \cdot \sum_{u\neq v} f_{(i,j),u\to v}$

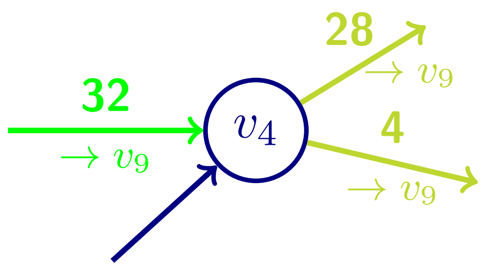
▷ Capacity constraints: $\sum_{u\neq v} f_{(i,j),u\to v} \leq u_{(i,j)}$ for all arcs $(i,j) \in A$

C

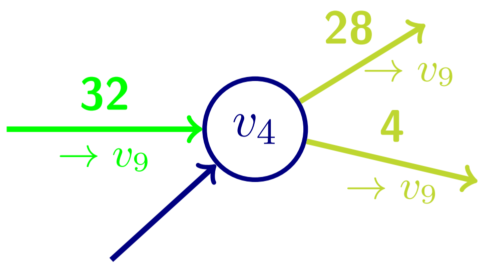
▷ Flow conservation constraints: for all $w \in V$, $u,v \in V, u \neq v$,

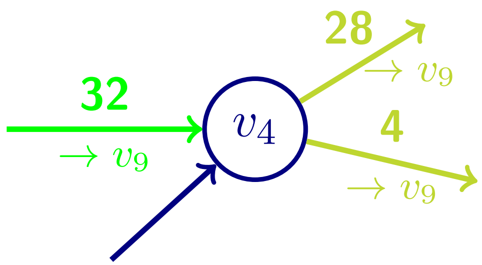
$$\sum_{(w,j)\in A} f_{(w,j),u\to v} - \sum_{(i,w)\in A} f_{(i,w),u\to v} = \begin{cases} d_{u,v} & \text{if } w = u \\ -d_{u,v} & \text{if } w = v \\ 0 & \text{otherwise} \end{cases}$$

▷ NOTE: flow for a given demand might be split at nodes!



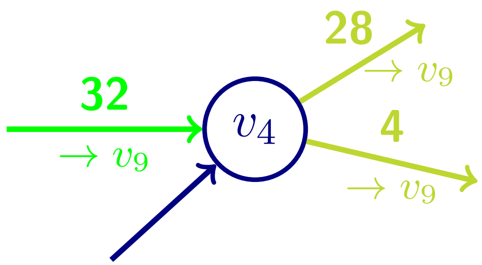
▷ How to avoid splitting flow at nodes?





- How to avoid splitting flow at nodes?
- Idea: replace flow variables $f_{(i,j),u\to v}$ with binary decision variables:

$y_{(i,j),u\to v} \in \{0,1\}$ ➡ $y_{(i,j),u\to v} = 1 \Leftrightarrow$ complete flow from u to v uses arc (i,j)



- How to avoid splitting flow at nodes?

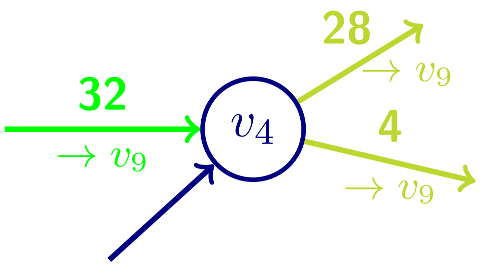
- Idea: replace flow variables $f_{(i,j),u\to v}$ with binary decision variables:

  $y_{(i,j),u\to v} \in \{0,1\}$ ➡ $y_{(i,j),u\to v} = 1 \Leftrightarrow$ complete flow from u to v uses arc (i,j)

➡ Set partitioning constraints for outgoing flows at each node w:

C

$$\sum_{(w,j)\in A} y_{(w,j),u\to v} \;=\; 1 \qquad \text{for all demands } u \to v,\ u \neq v$$



- ▷ How to avoid splitting flow at nodes?

- ▷ Idea: replace flow variables $f_{(i,j),u\to v}$ with binary decision variables:

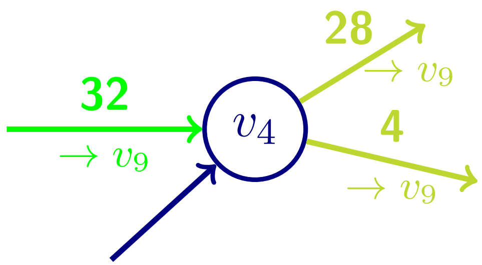
  $y_{(i,j),u\to v} \in \{0,1\}$ ➡ $y_{(i,j),u\to v} = 1 \Leftrightarrow$ complete flow from u to v uses arc (i,j)

➡ Set partitioning constraints for outgoing flows at each node w:

C

$$\sum_{(w,j)\in A} y_{(w,j),u\to v} = 1 \quad \text{for all demands } u \to v,\ u \neq v$$

➡ Replace flow $f_{(i,j),u\to v}$ with $d_{u,v} \cdot y_{(i,j),u\to v}$ in all other constraints and in the objective function...



- ▷ How to avoid splitting flow at nodes?

- ▷ Idea: replace flow variables $f_{(i,j),u\to v}$ with binary decision variables:

  $y_{(i,j),u\to v} \in \{0,1\}$ ➡ $y_{(i,j),u\to v} = 1 \Leftrightarrow$ complete flow from u to v uses arc (i,j)

➡ Set partitioning constraints for outgoing flows at each node w:

C

$$\sum_{(w,j)\in A} y_{(w,j),u\to v} = 1 \quad \text{for all demands } u \to v,\ u \neq v$$

➡ Replace flow $f_{(i,j),u\to v}$ with $d_{u,v} \cdot y_{(i,j),u\to v}$ in all other constraints and in the objective function...

- ▷ Similarly: k-splittable flow – allow splitting into at most/exactly/at least k parts at each node

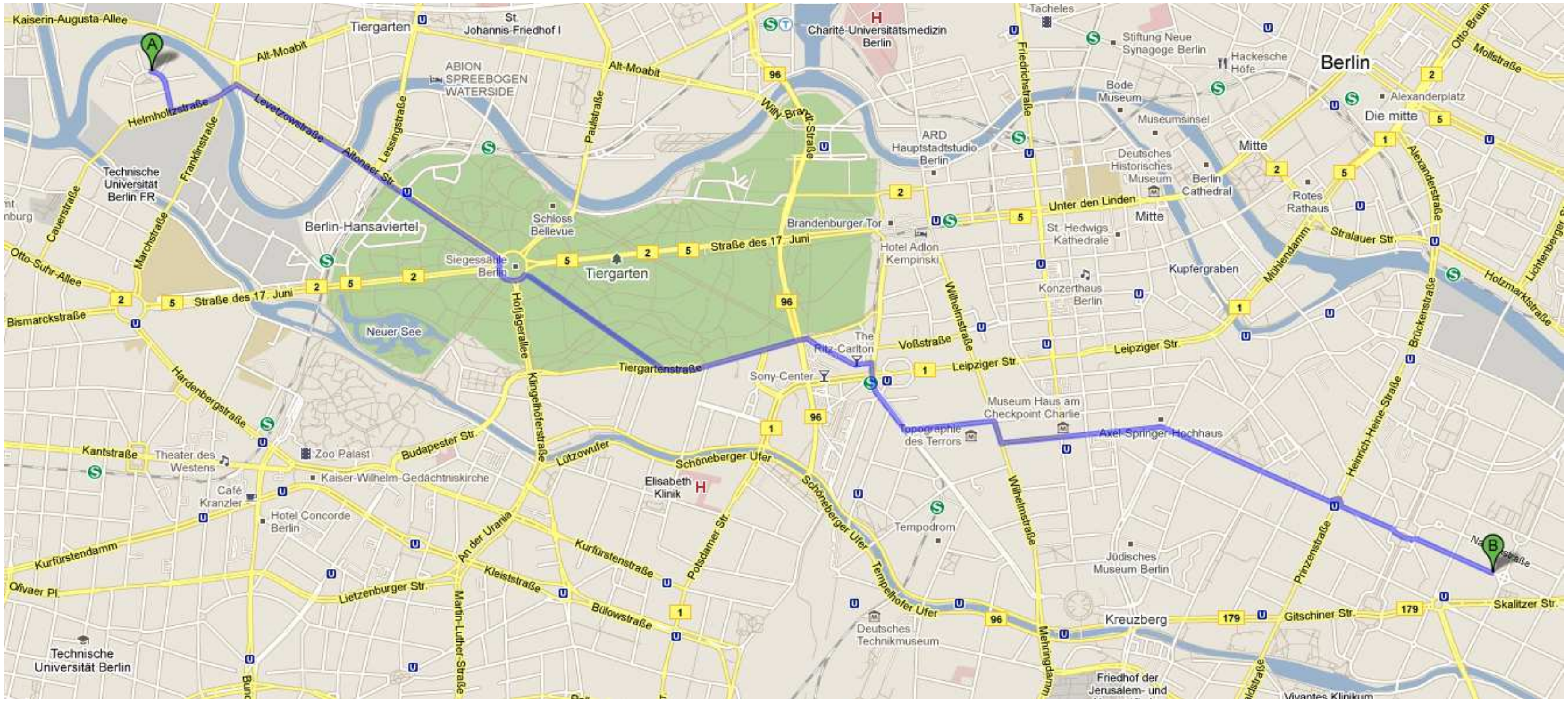
- Problem: find a shortest path from PTZ to Kreuzburger

$\rightsquigarrow$

Example: Shortest Path

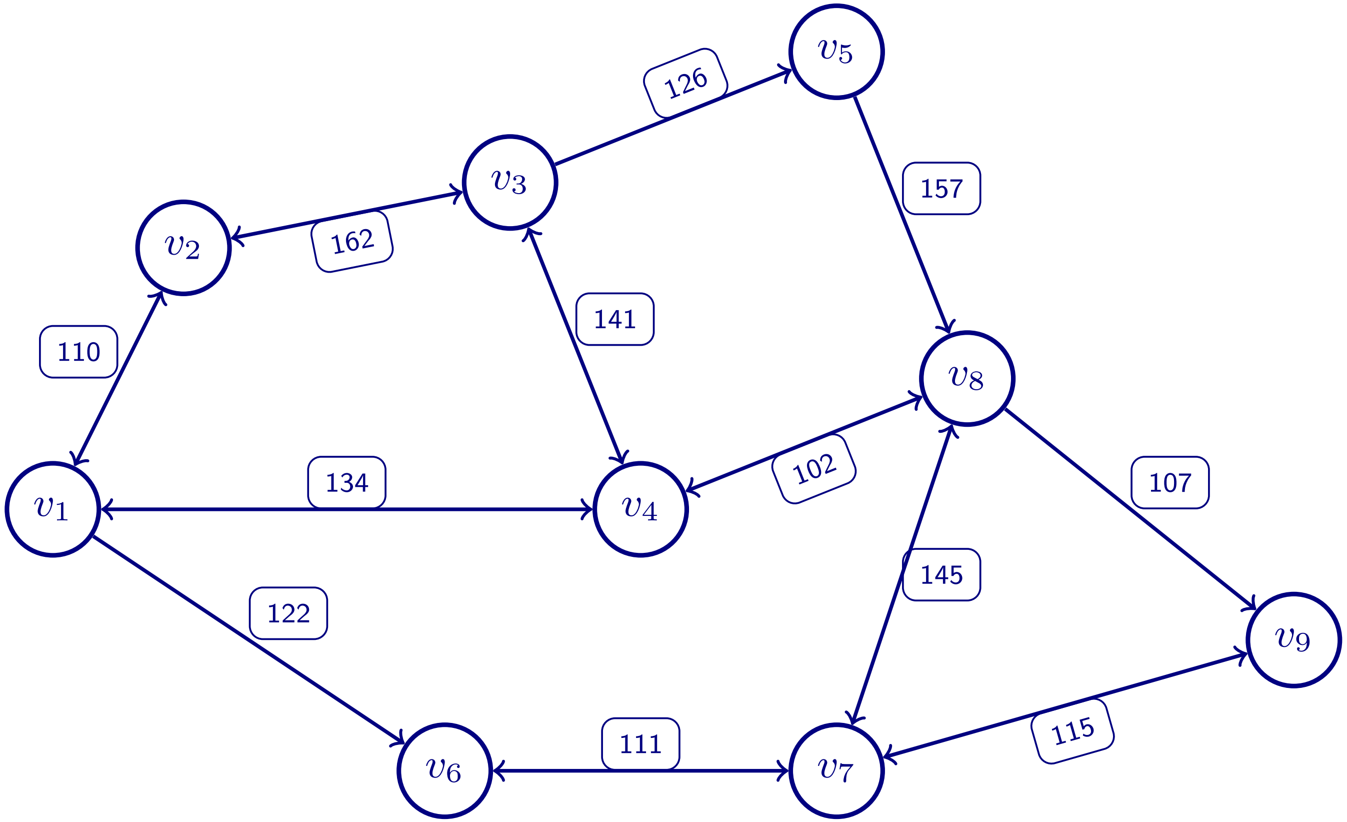
- Problem: find a shortest path from PTZ to Kreuzburger

⇝

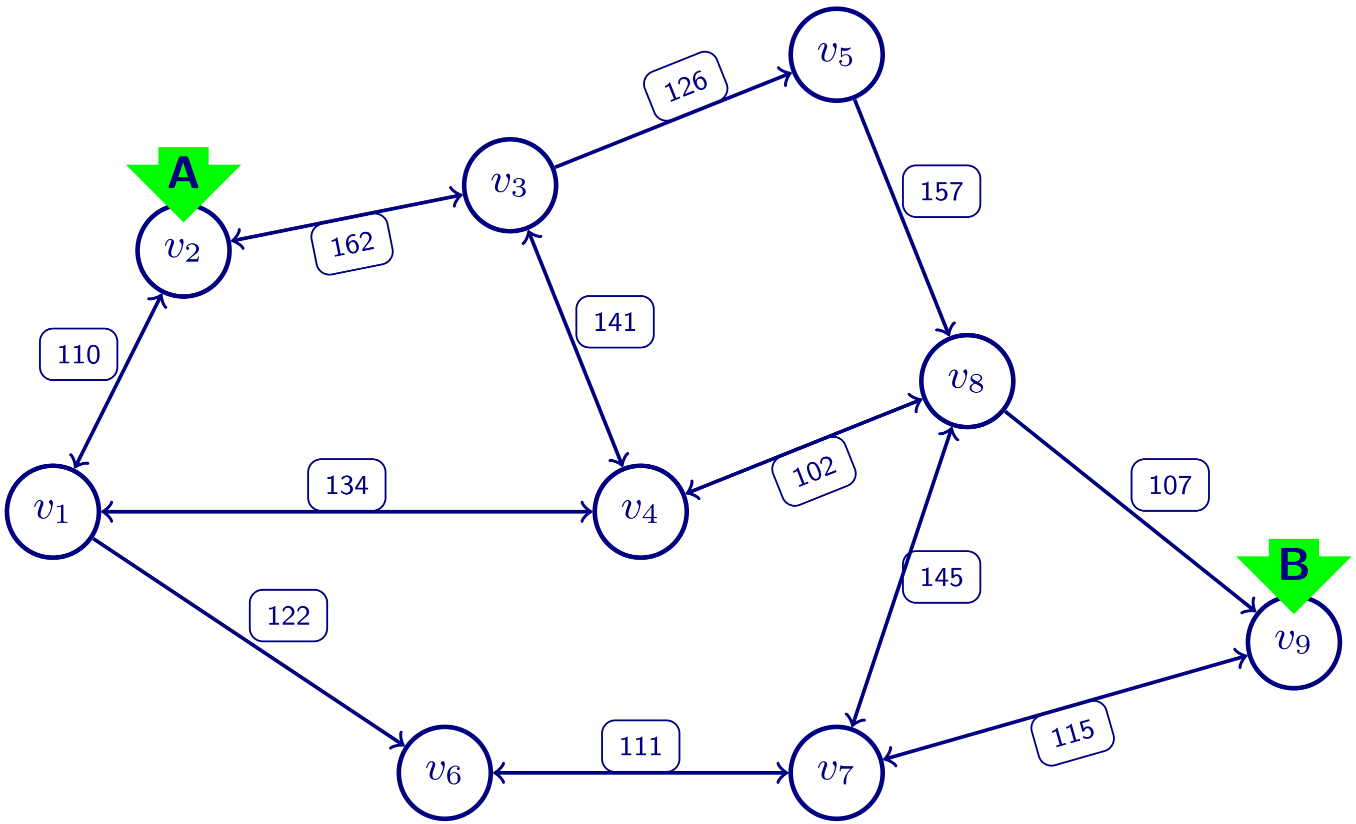


- Given a network – i.e. a directed graph – with a length for each arc, a start node A and a destination B...

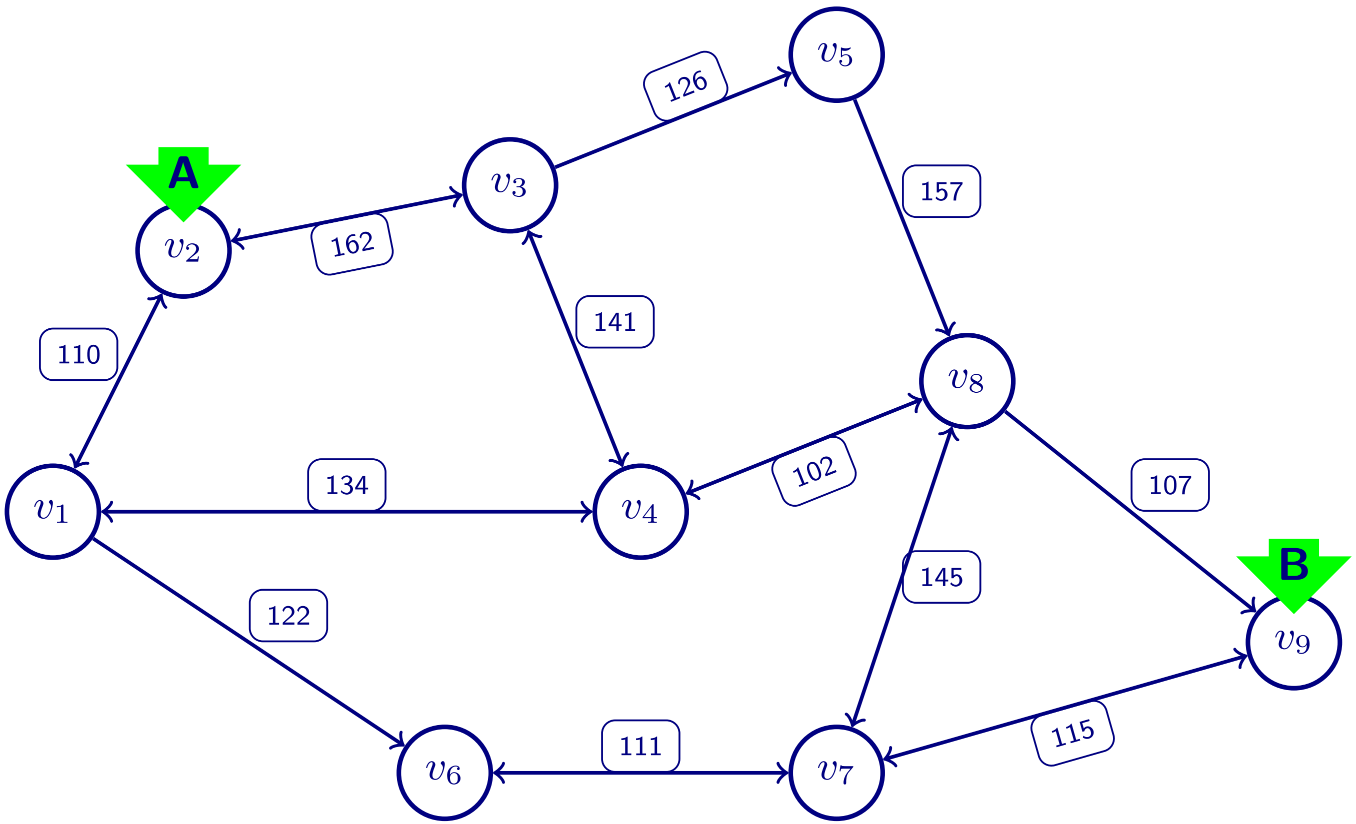
- Given a network – i.e. a directed graph – with a length for each arc, a start node A and a destination B...



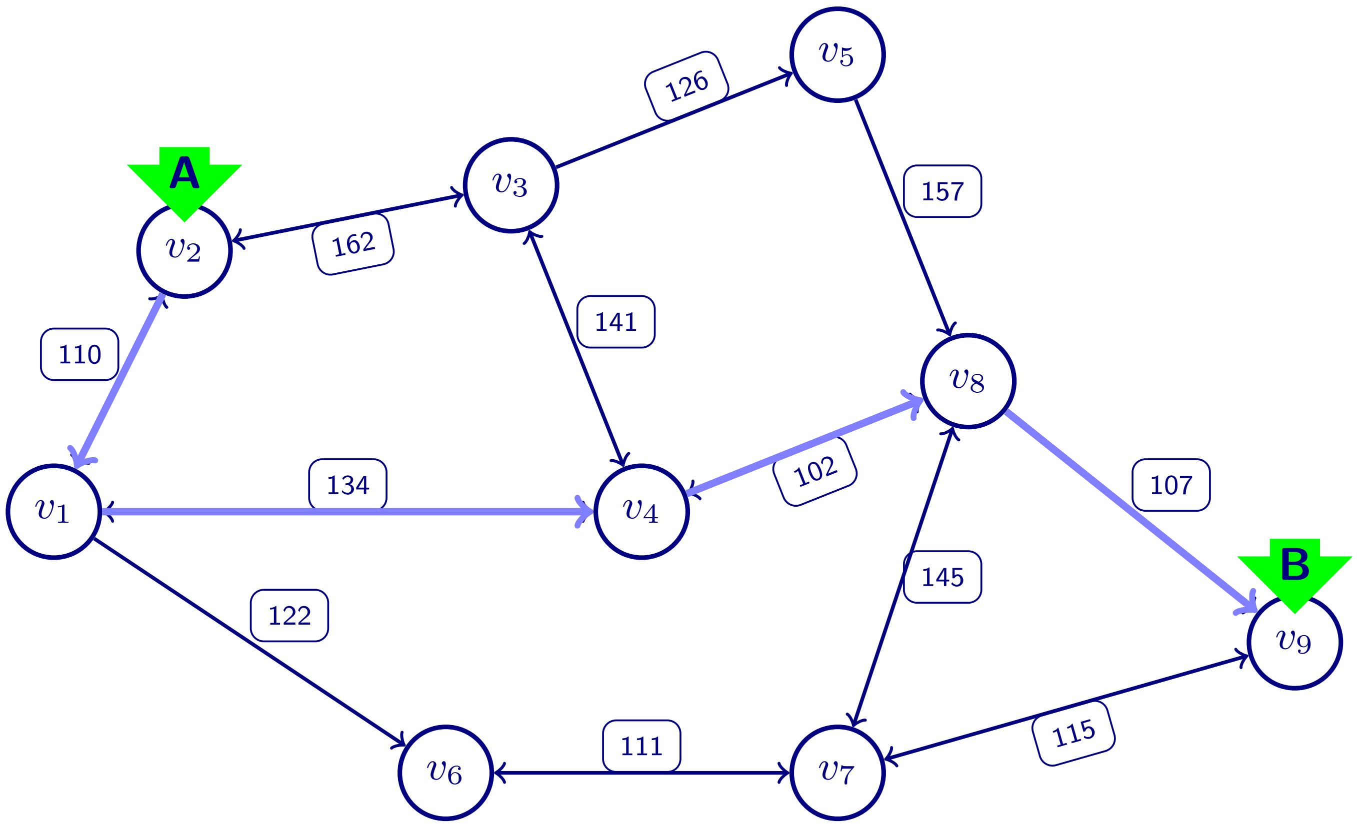
- Given a network – i.e. a directed graph – with a length for each arc, a start node A and a destination B...



- Given a network – i.e. a directed graph – with a length for each arc, a start node A and a destination B...
- ...compute a shortest path through the network from A to B

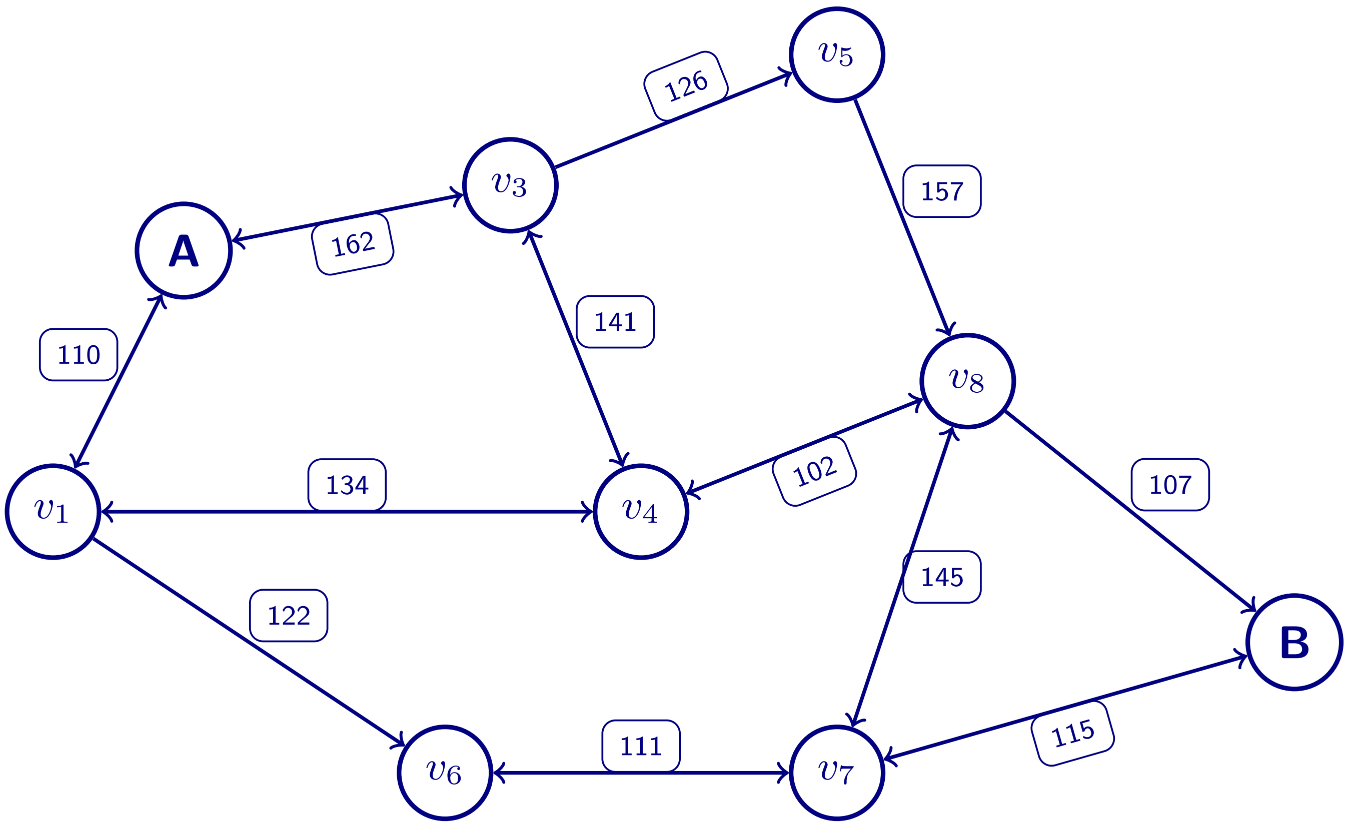


- Given a network – i.e. a directed graph – with a length for each arc, a start node A and a destination B...
- ...compute a shortest path through the network from A to B

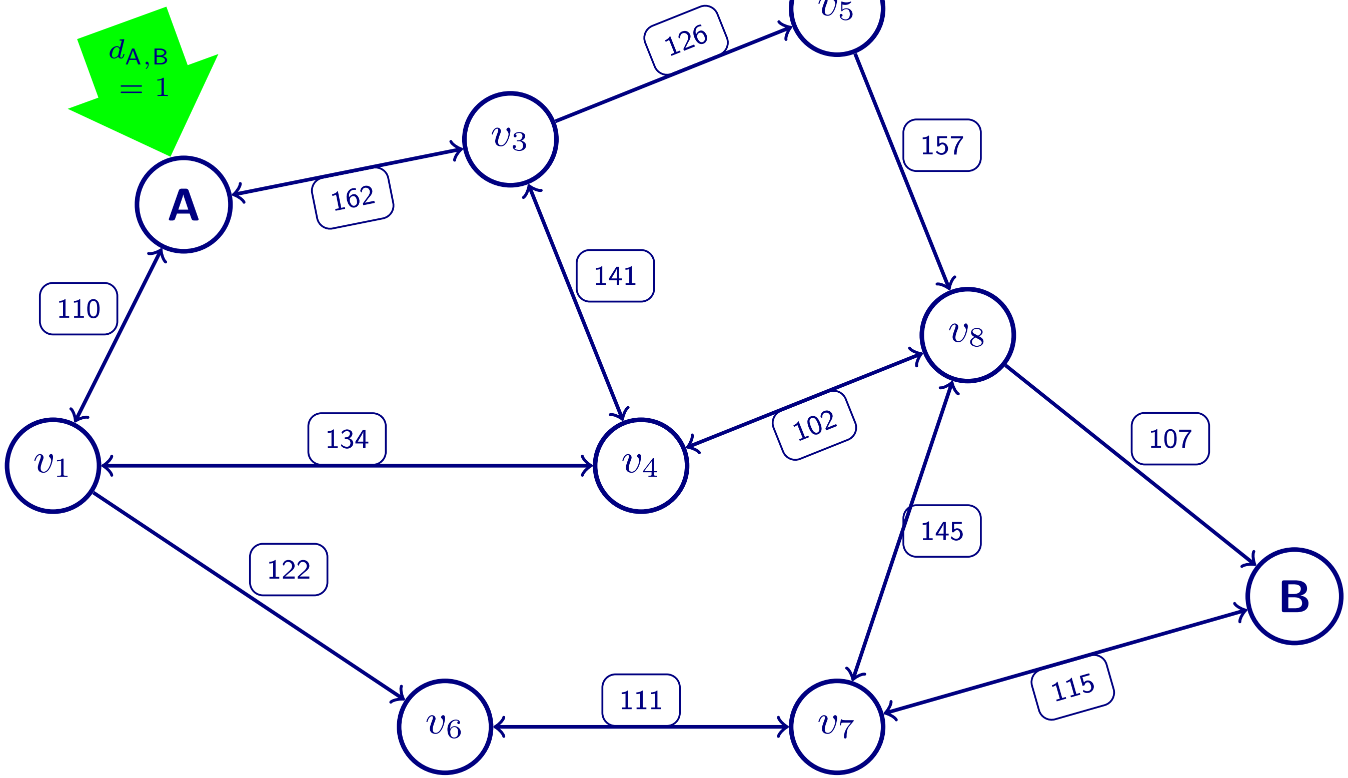


- Shortest Path Problem can be formulated as a network flow problem:

▷ Shortest Path Problem can be formulated as a network flow problem:

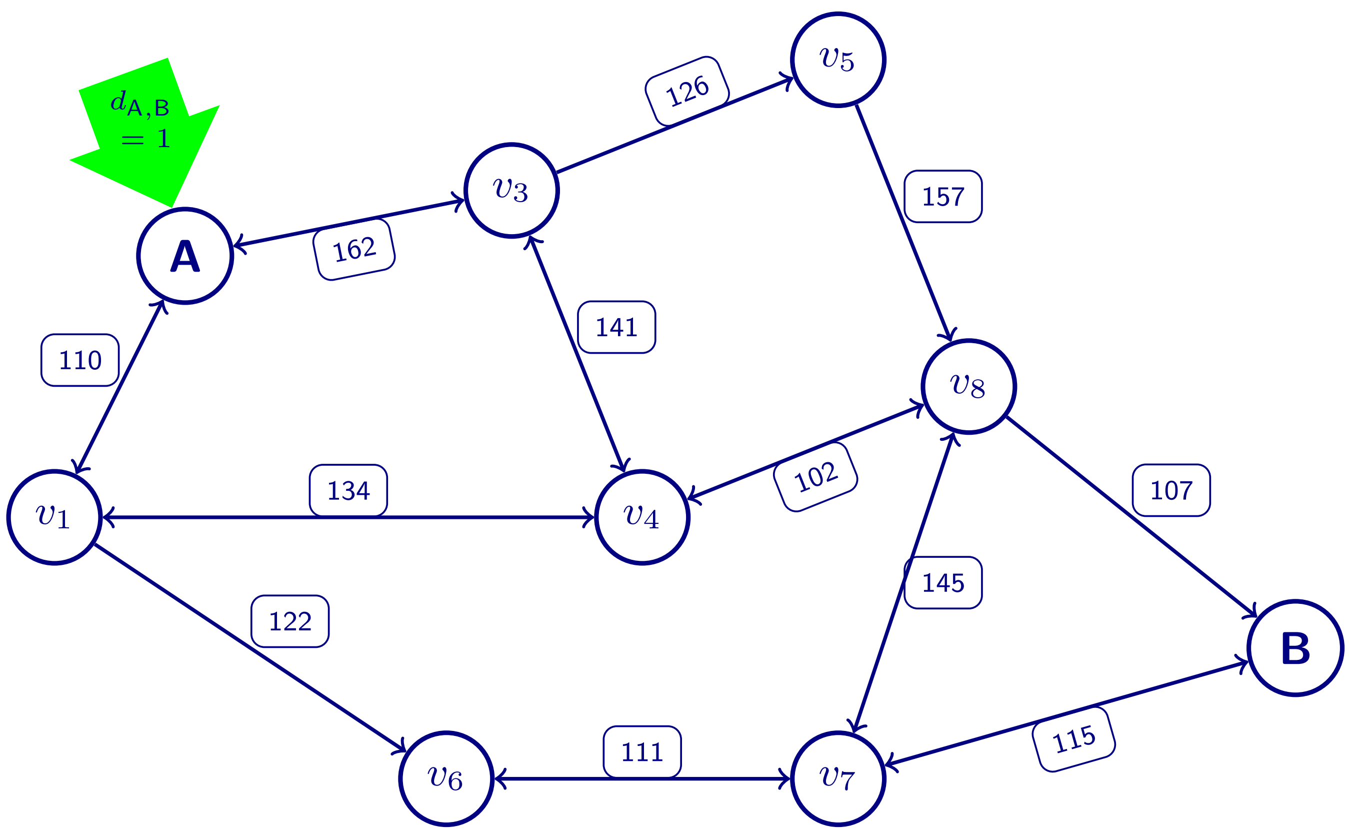


▷ Shortest Path Problem can be formulated as a network flow problem:



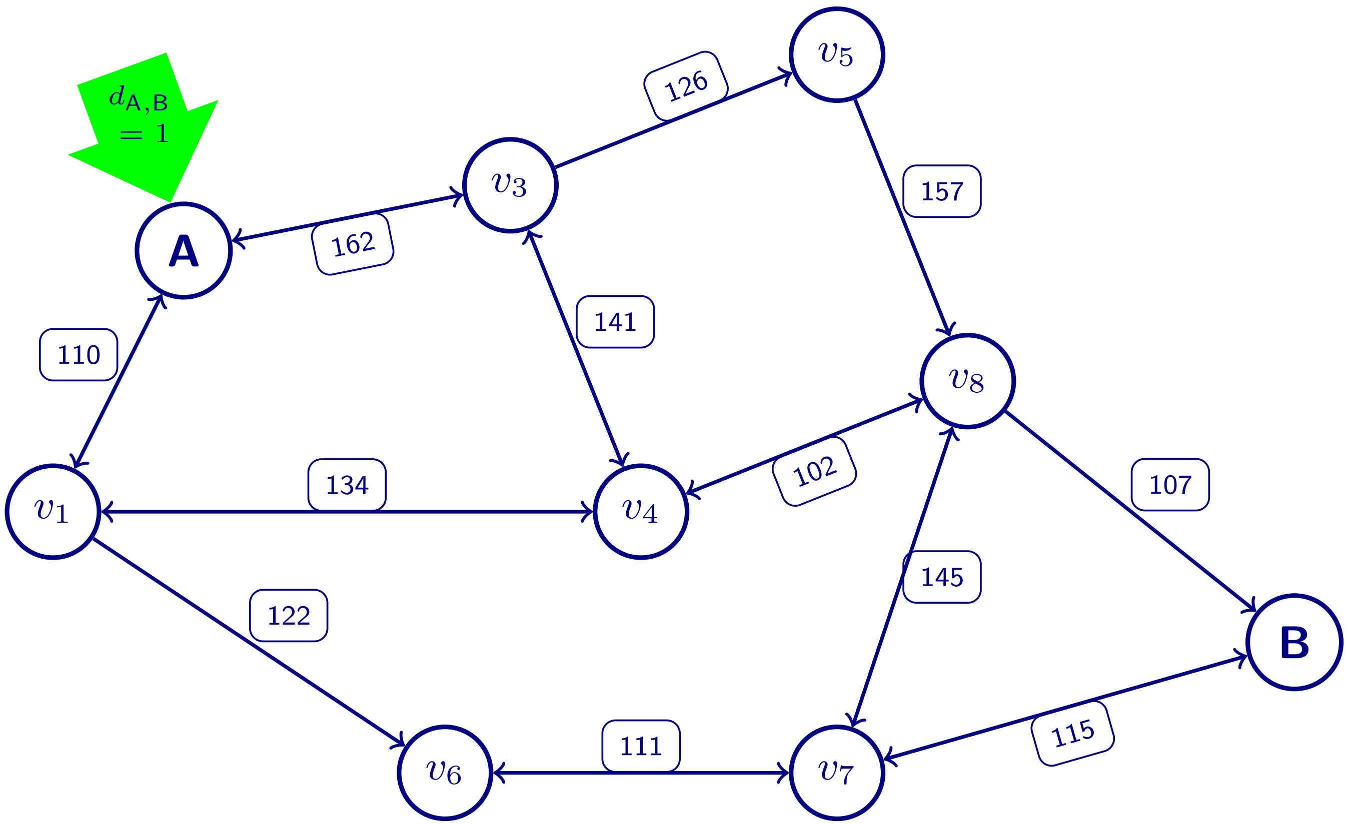
▷ Shortest Path Problem can be formulated as a network flow problem:

- Only one demand: $d_{\mathsf{A},\mathsf{B}} = 1$, all other demands $d_{u,v} = 0$



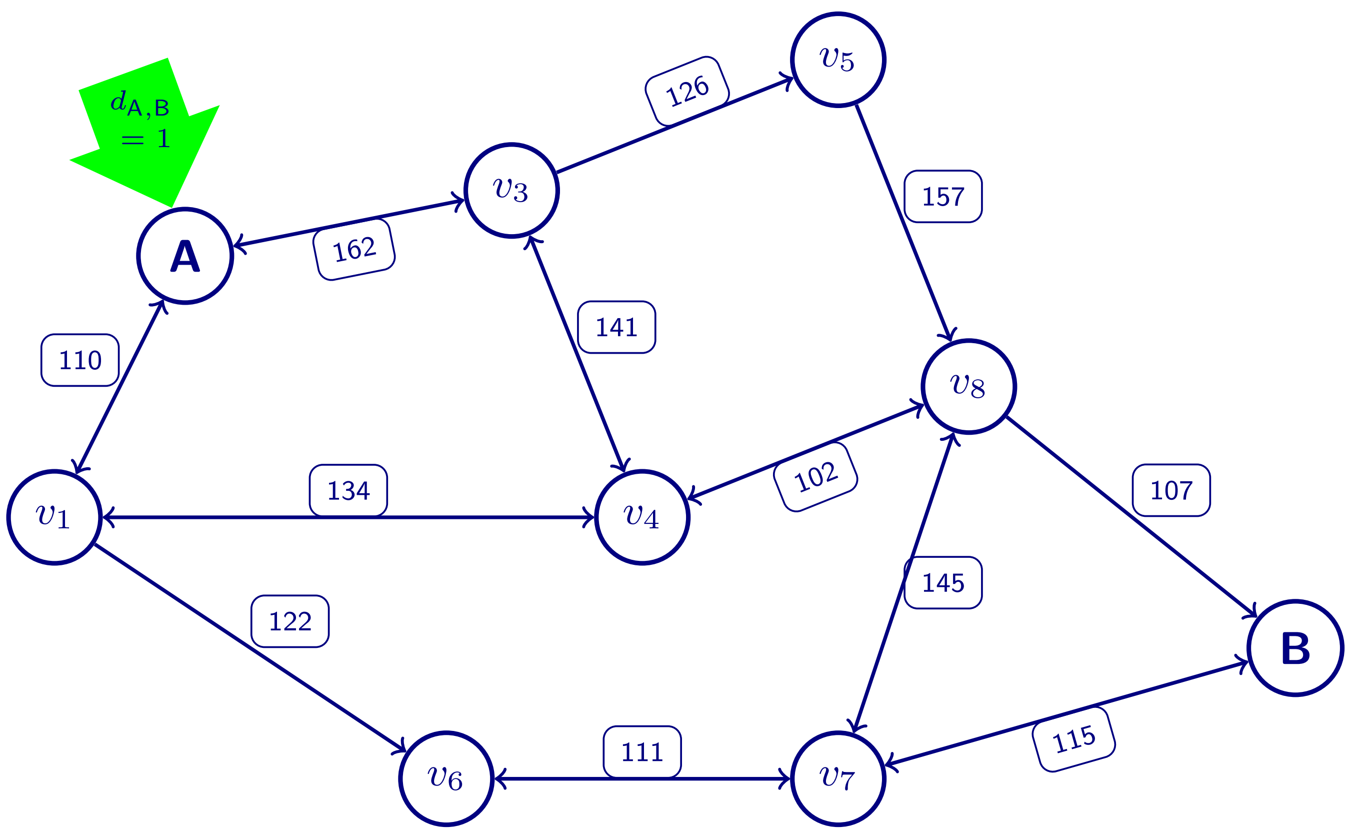
▷ Shortest Path Problem can be formulated as a network flow problem:

- Only one demand: $d_{\mathsf{A},\mathsf{B}} = 1$, all other demands $d_{u,v} = 0$
- Unsplittable flow



▷ Shortest Path Problem can be formulated as a network flow problem:

- Only one demand: $d_{\text{A,B}} = 1$, all other demands $d_{u,v} = 0$
- Unsplittable flow
- No capacities



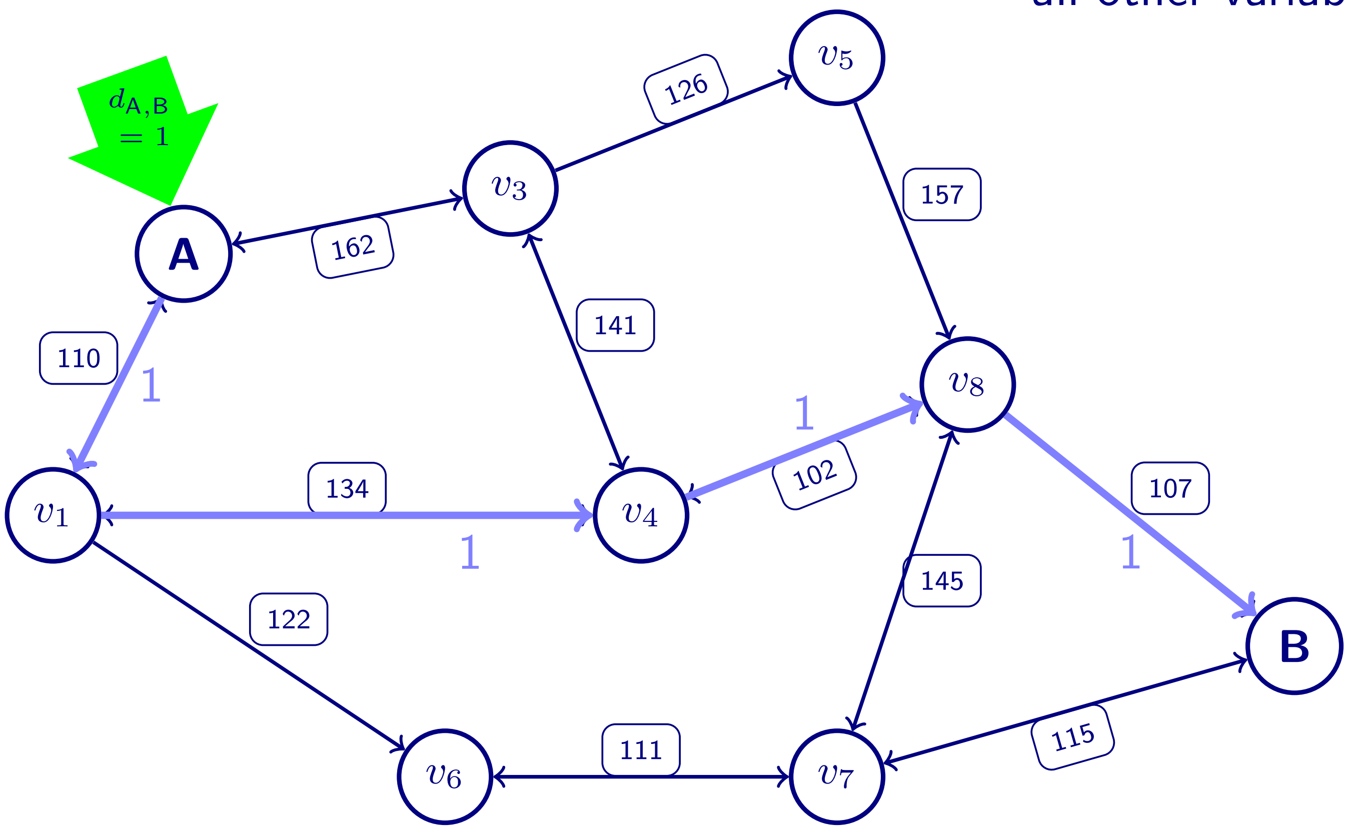
▷ Shortest Path Problem can be formulated as a network flow problem:

- Only one demand: $d_{\mathsf{A},\mathsf{B}} = 1$, all other demands $d_{u,v} = 0$
- Unsplittable flow
- No capacities

➡ Optimal solution:

$y_{(\mathsf{A},v_1)} = y_{(v_1,v_4)} = y_{(v_4,v_8)} = y_{(v_8,\mathsf{B})} = 1,$

all other variables 0



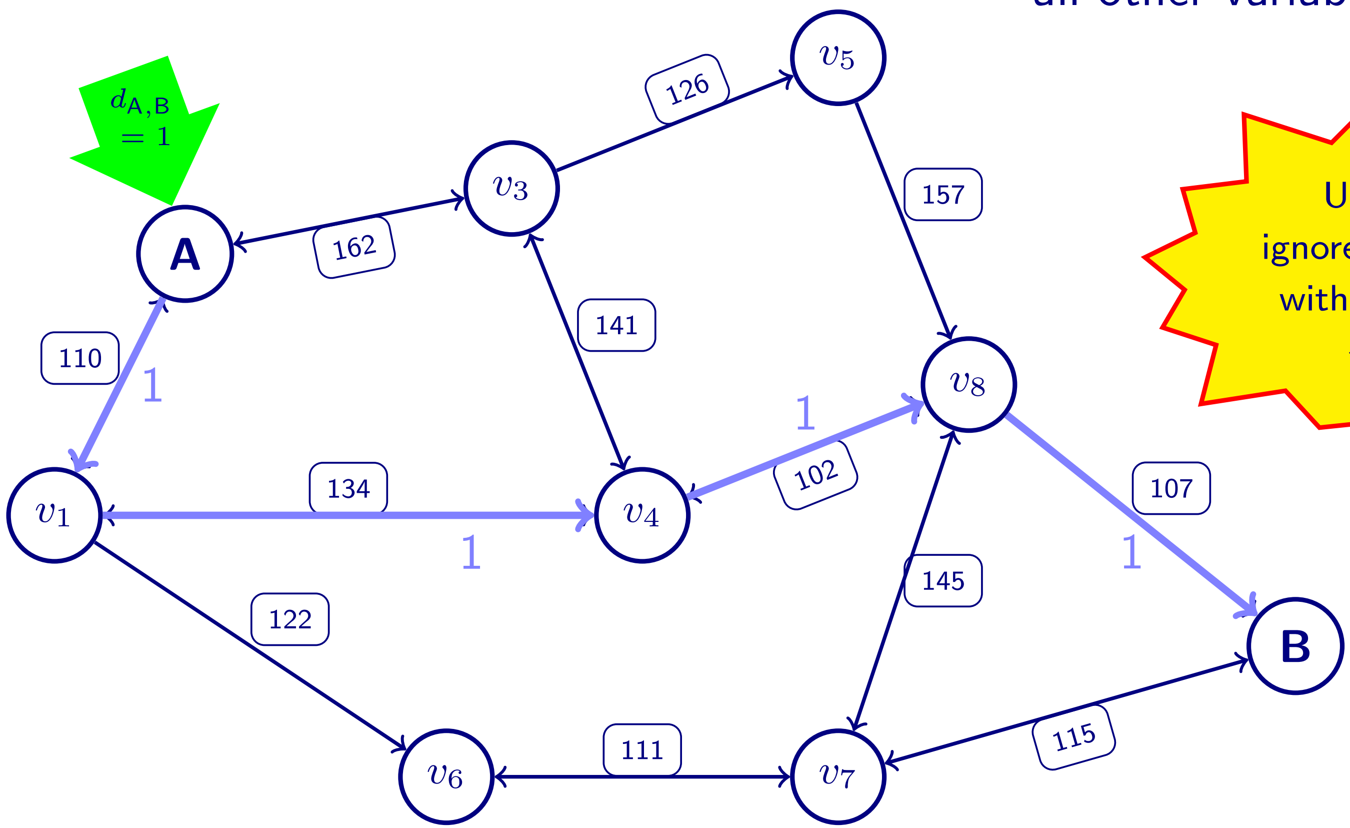
▷ Shortest Path Problem can be formulated as a network flow problem:

- Only one demand: $d_{\mathrm{A,B}} = 1$, all other demands $d_{u,v} = 0$
- Unsplittable flow
- No capacities

➡ Optimal solution:

$y_{(\mathrm{A},v_1)} = y_{(v_1,v_4)} = y_{(v_4,v_8)} = y_{(v_8,\mathrm{B})} = 1$,

all other variables 0



Unsplittability can be ignored if the model is solved with the simplex algorithm → linear program!

- Models, Data and Algorithms
- Linear Optimization
- Mathematical Background: Polyhedra, Simplex-Algorithm
- Sensitivity Analysis; (Mixed) Integer Programming
- MIP Modelling
- MIP Modelling: More Examples; Branch & Bound
- Cutting Planes; Combinatorial Optimization: Examples, Graphs, Algorithms
- TSP-Heuristics
- Network Flows
- Shortest Path Problem, Complexity Theory
- Nonlinear Optimization
- Scheduling, Lot Sizing
- Multicriteria Optimization
- Oral exam