Network Flows Over Time

(IMA Tutorial)

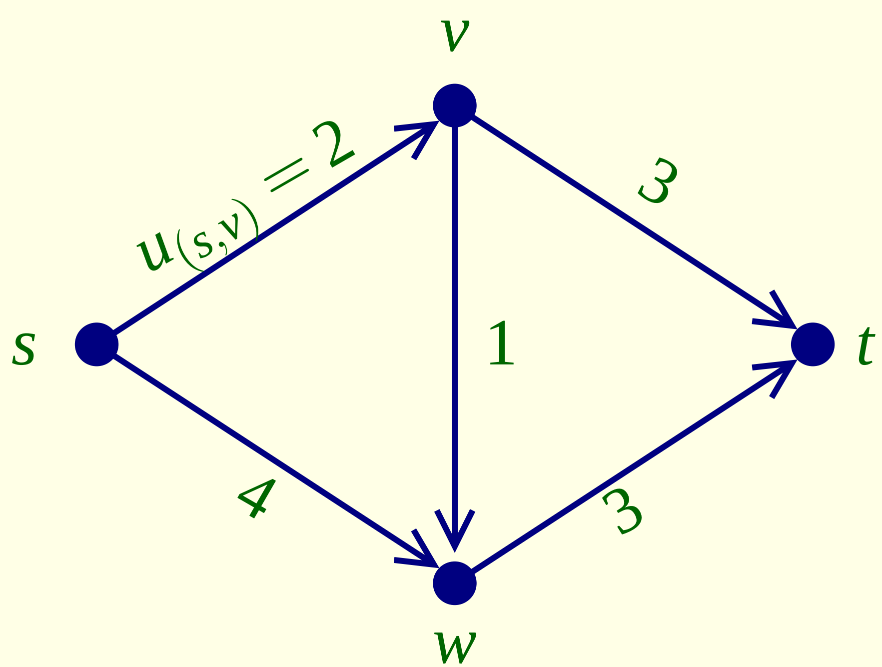
Martin Skutella

Technische Universität Berlin

(currently at M.I.T.)

Network Flows

Given: network $\mathcal{N} = (V, A)$ with capacities $u_e \in \mathbb{Z}_{\geqslant 0}$ on the arcs $e \in A$; source $s \in V$ and sink $t \in V$.



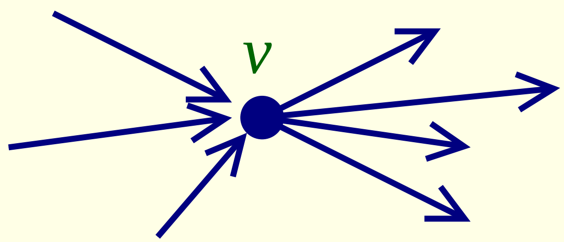
Task: Send as much flow as possible from the source s to the sink t.

Static Maximum Network Flows

Wanted: static s-t-flow $x = (x_e)_{e \in A} \in \mathbb{R}_{\geqslant 0}{}^A$ obeying

- capacity constraints: $x_e \leqslant u_e$ for all $e \in A$,
- flow conservation constraints

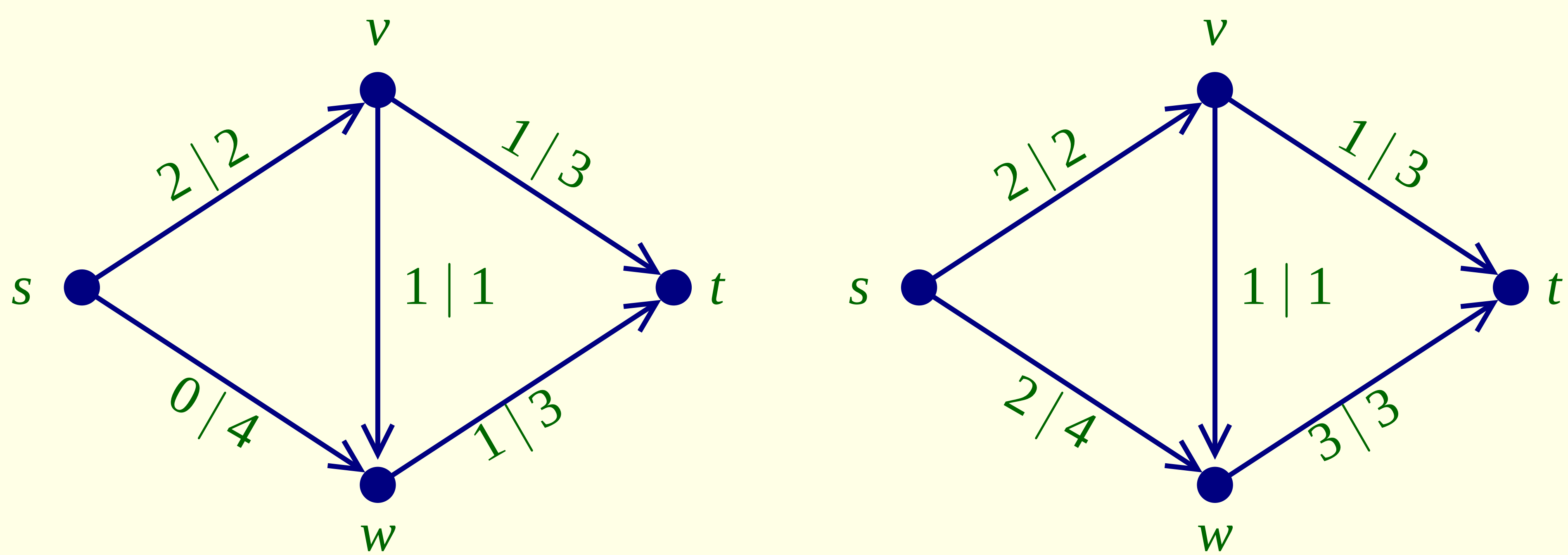
$$\sum_{e \in \delta^-(v)} x_e = \sum_{e \in \delta^+(v)} x_e \qquad \text{for all } v \neq s,t,$$



- maximizing $|x| := \sum_{e \in \delta^+(s)} x_e - \sum_{e \in \delta^-(s)} x_e$.

An Example

The value of the following flow is not yet optimal since



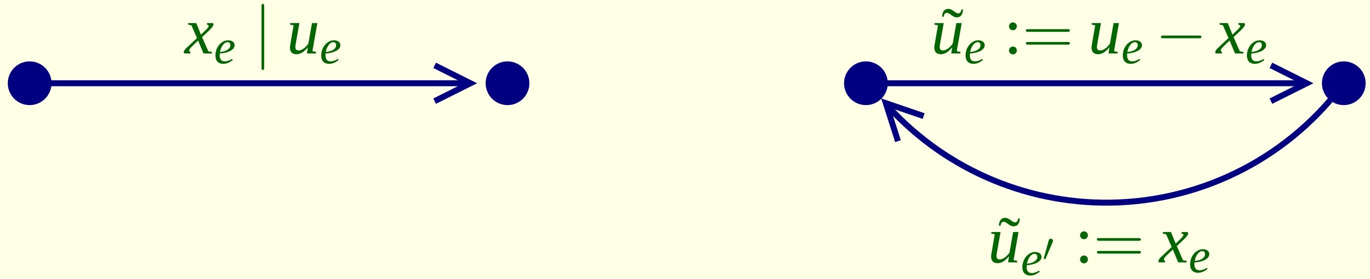
we can send 2 more units of flow from s via w to t:

The flow on the right hand side is still not optimal.

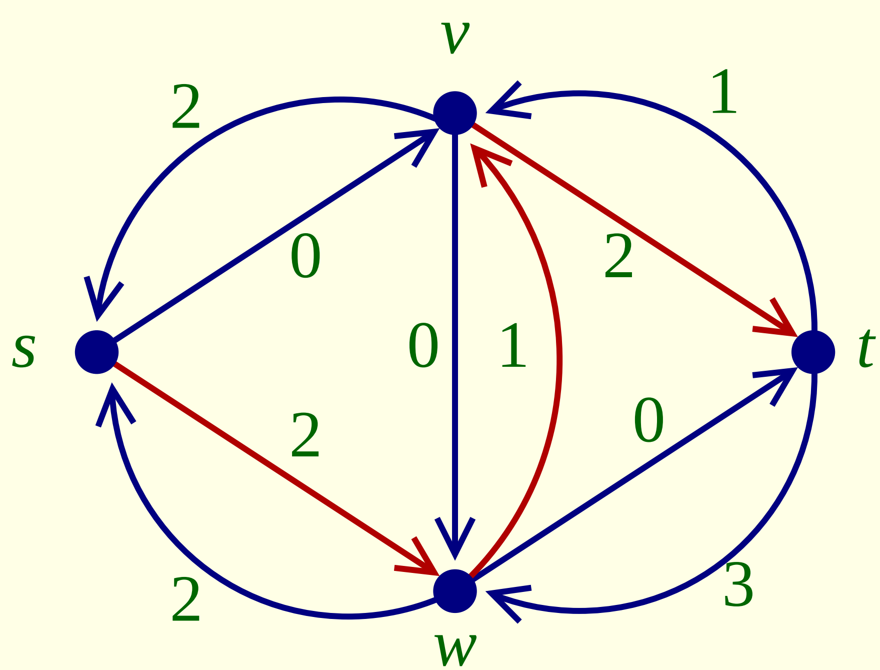
A proof for this claim can be found in the residual network.

The Residual Network

Consider the flow x_e on an arc e:

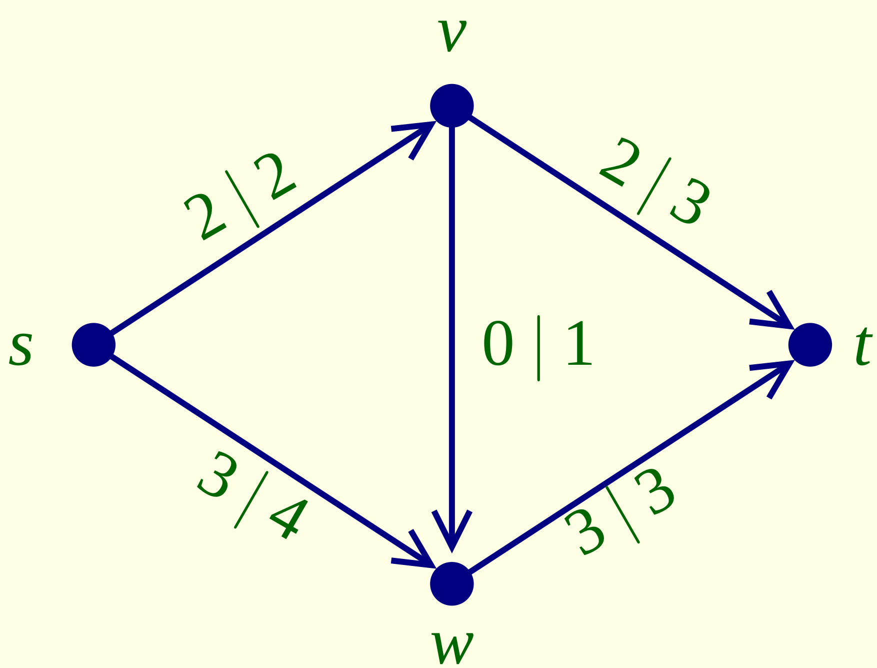


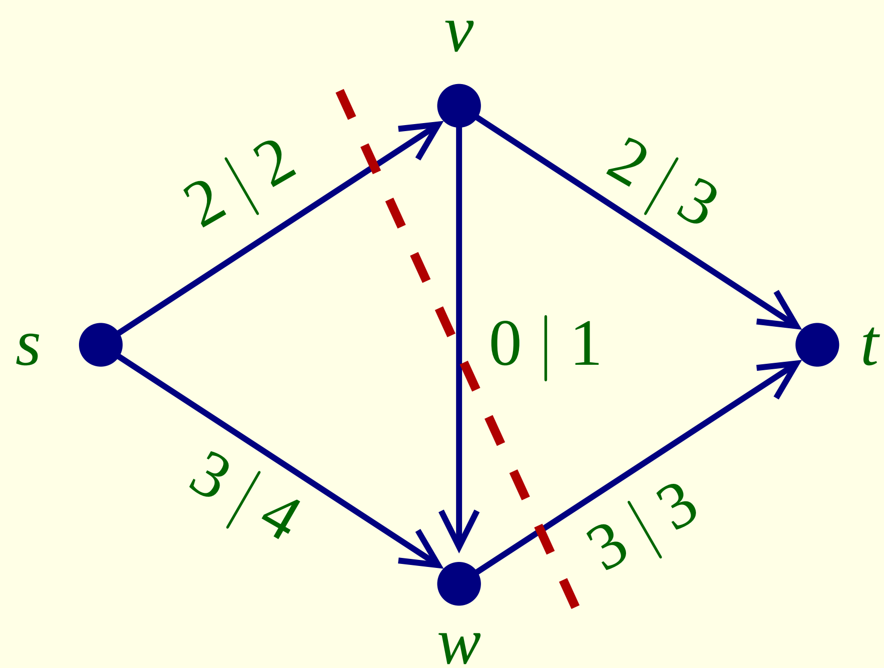
The flow can either be raised by at most $u_e - x_e$ or decreased by at most x_e. Resulting residual network:



Cuts in Networks

The resulting flow of value 5 seems to be optimal.





Proof. The value of any s-t-flow is bounded by the capacity of the red cut (certificate of optimality). $\square$

An s-t-cut $\delta^+(S)$ is given by a subset $S \subseteq V \setminus \{t\}$ with $s \in S$:

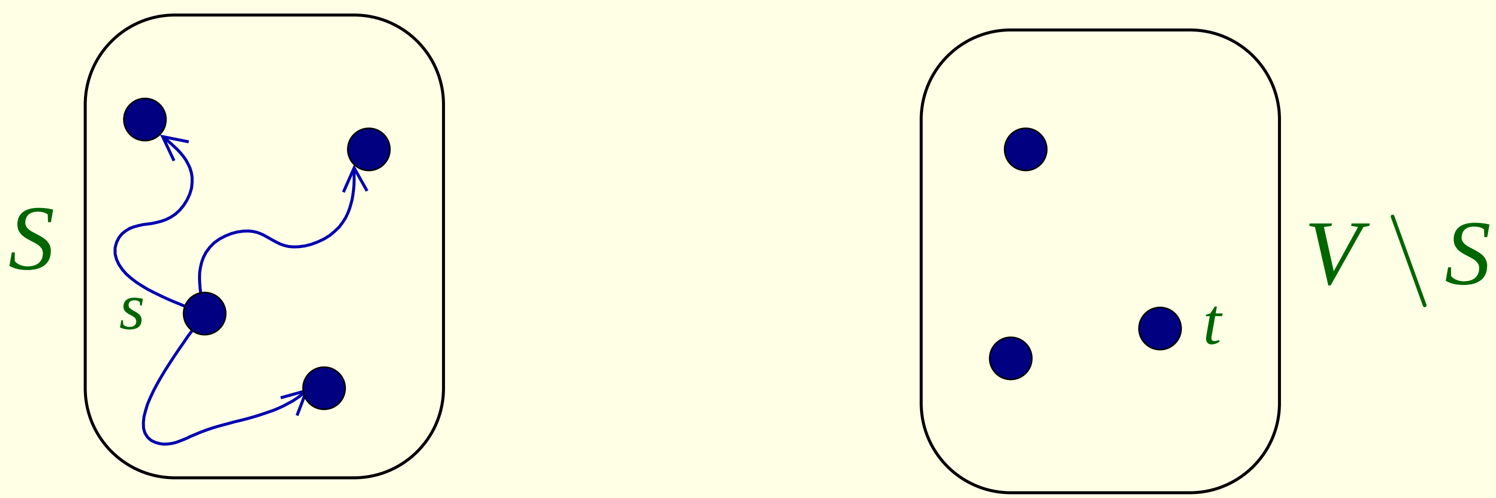
$$\delta^+(S) \;:=\; \{(v,w) \in A \mid v \in S \text{ and } w \notin S\} \;.$$

The MaxFlow-MinCut Theorem

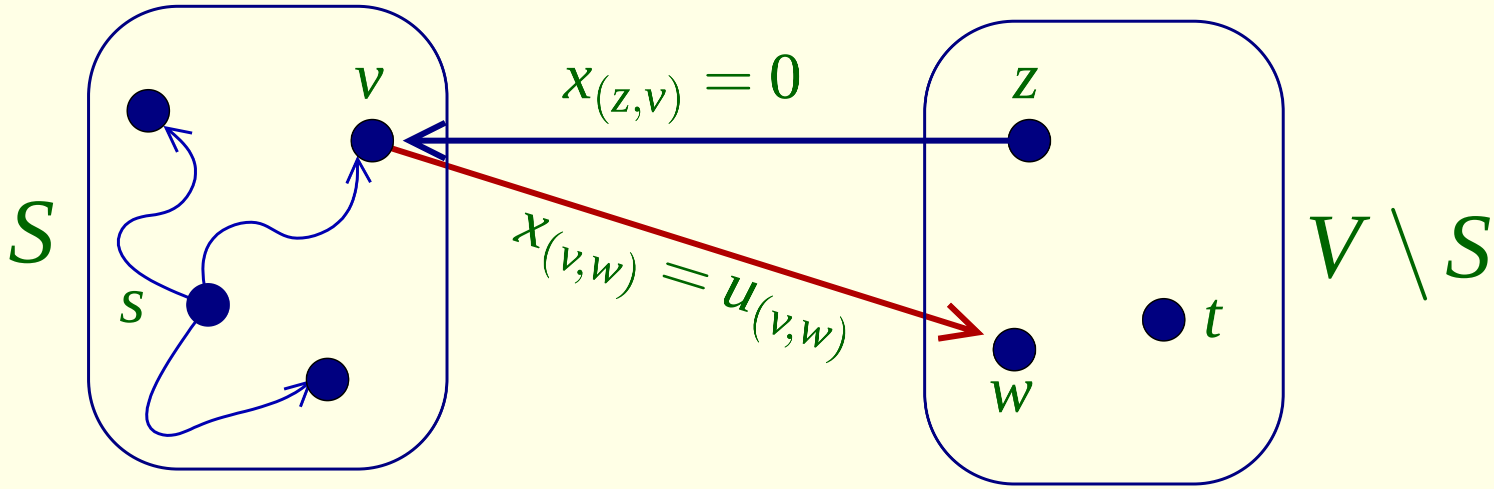
Theorem. (Ford & Fulkerson 1956)
The value of a maximum s-t-flow equals the capacity of a minimum s-t-cut.

Proof. "$\leqslant$" is clear.

"$\geqslant$": Let x be a maximum flow and let $S \subseteq V$ be the set of nodes v for which an augmenting s-v-path exists in the residual network.



Notice: $s \in S$ by construction and $t \notin S$ by optimality of x.



Moreover,

$$x_e \;=\; \begin{cases} u_e & \text{for all } e \in \delta^+(S), \\ 0 & \text{for all } e \in \delta^-(S). \end{cases}$$

Thus, $\quad |x| \;=\; u\big(\delta^+(S)\big) \;\geqslant\; u\big(\delta^+(S^*)\big).$ $\square$

Flow Decomposition

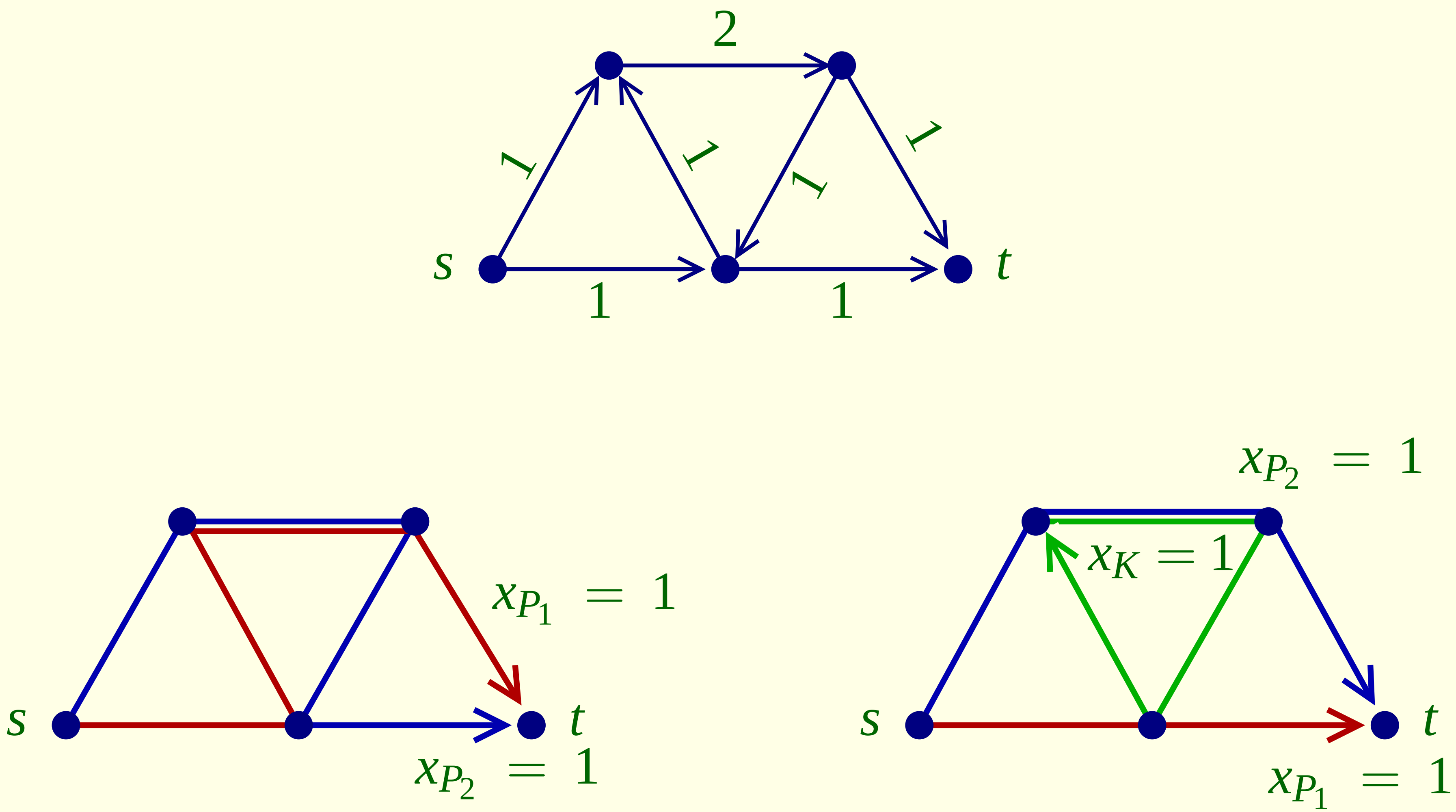
Theorem. (Fulkerson 1962)

An s-t-flow x can be decomposed into flows on s-t-paths $\mathcal{P}$ and and flows on cycles $\mathcal{C}$, that is

$$x_e = \sum_{\substack{P \in \mathcal{P} \\ e \in P}} x_P + \sum_{\substack{C \in \mathcal{C} \\ e \in C}} x_C \qquad \text{for all } e \in A.$$

Moreover, the number of paths and cycles can be bounded by the number of arcs, i. e., $|\mathcal{P}| + |\mathcal{C}| \leqslant |A|$.

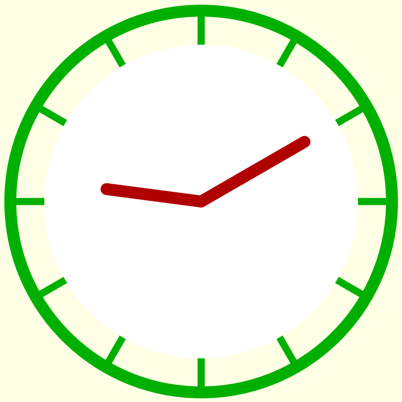
Example



Adding Temporal Dimension

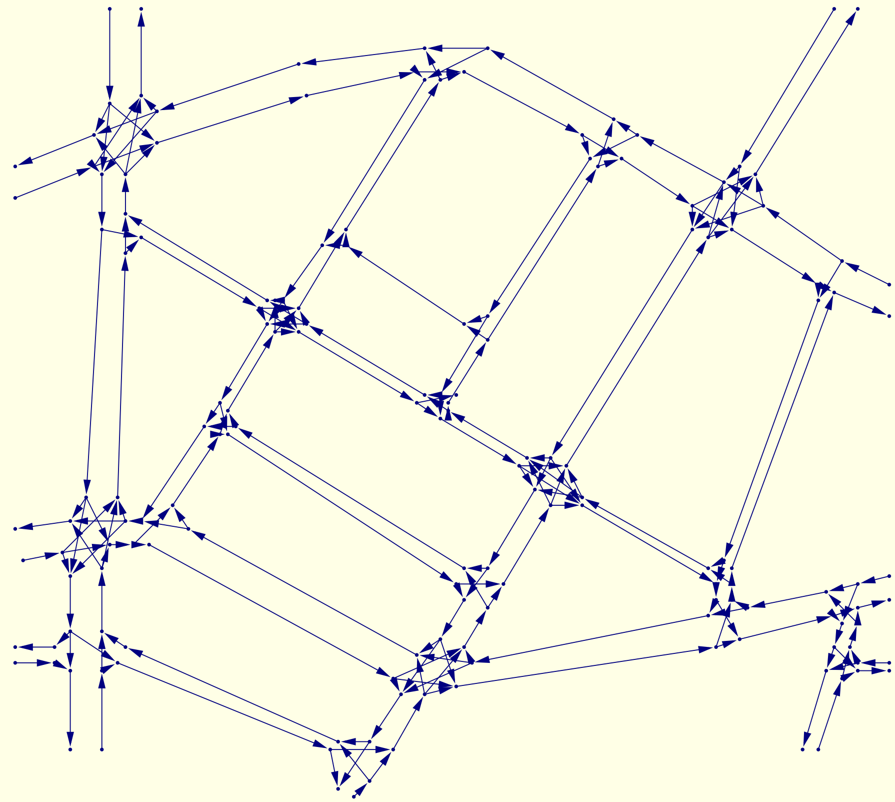
So far, we have only considered steady state flows.

However, in many applications, time plays a vital role!



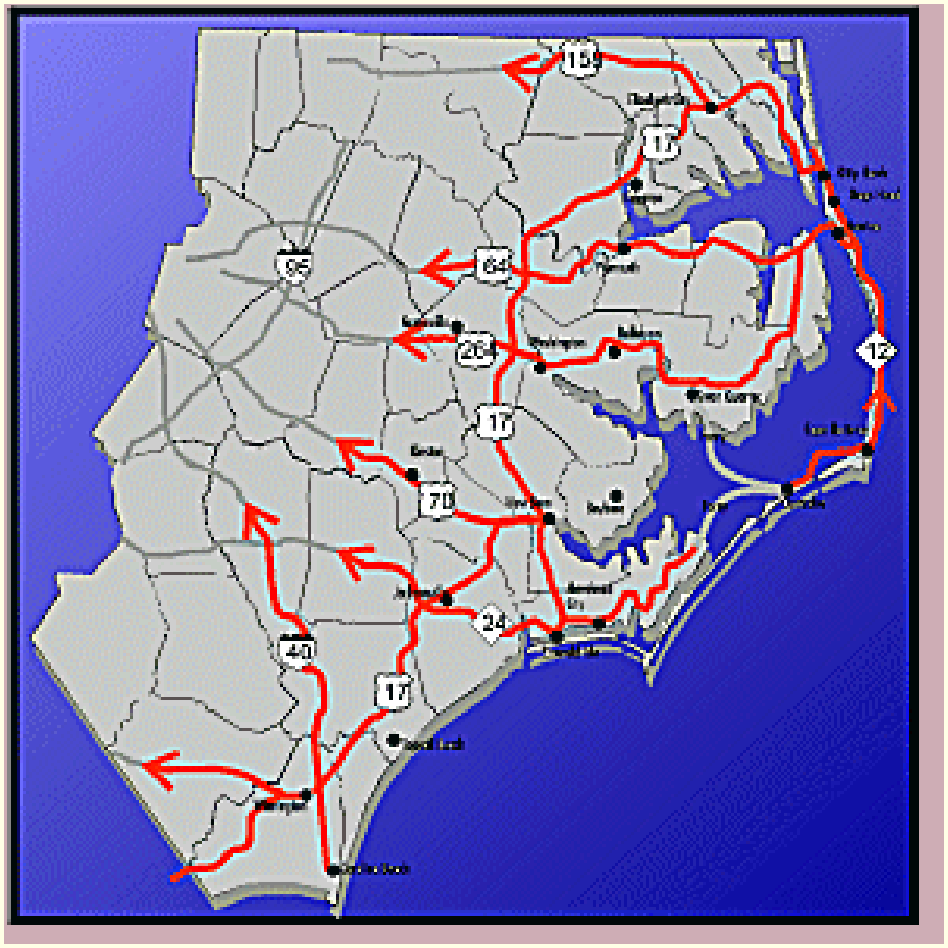
- Flow variation over time due to seasonal altering demands, supplies, and/or arc capacities.
- Flow travels only at a certain pace through the network, that is, there are transit times on the arcs.

Modeling Road Traffic as Network Flow



Other Applications

- evacuation plans
- communication networks (e. g., Internet)
- financial flows
- production systems
- air traffic



Literature: For surveys and more applications see, e. g.:

- Aronson (1989); Powell, Jaillet, & Odoni (1995).

The Maximum ‘Dynamic’ Flow Problem

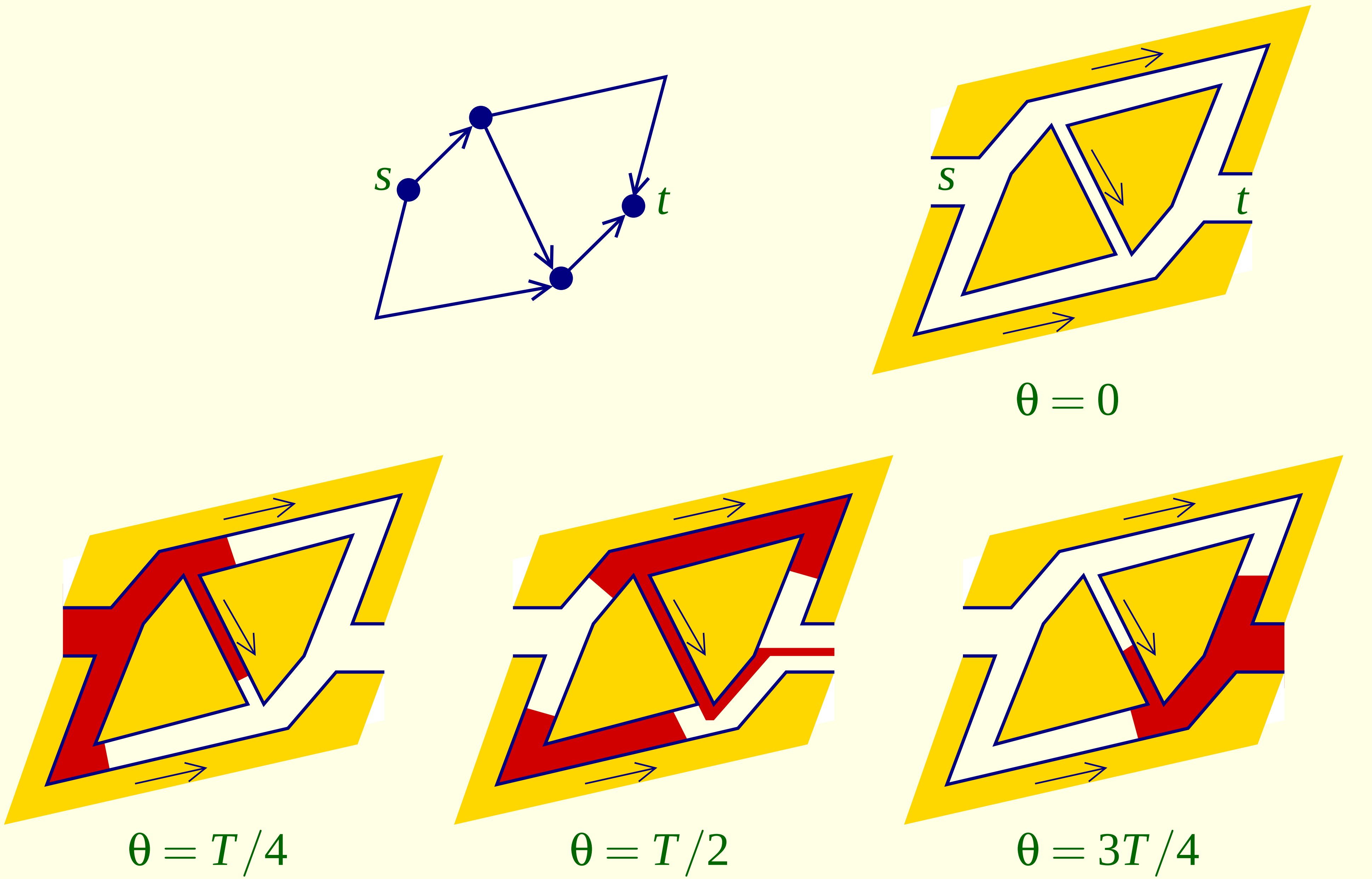
Ford & Fulkerson (1958) study the following problem:

Given: network $\mathcal{N} = (V, A)$ with capacities and transit times $\tau_e \in \mathbb{Z}_{\geqslant 0}$ on the arcs $e \in A$; source node $s \in V$ and sink node $t \in V$; time horizon $T \in \mathbb{Z}_{\geqslant 0}$.

Interpretation: The transit time τ_e of an arc $e = (v, w)$ specifies the amount of time it takes for flow to travel from the tail v to the head w of arc e.

Aim: determine the maximal amount of flow that can be sent from source s to sink t within time horizon T.

Example



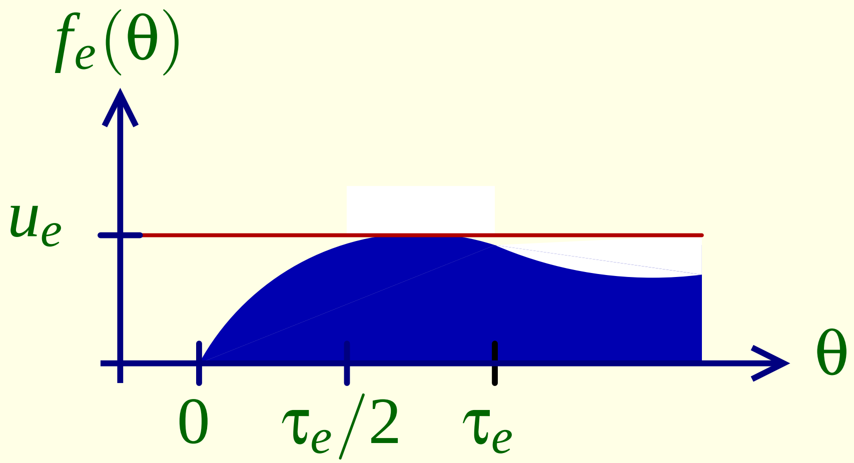
Flows Over Time

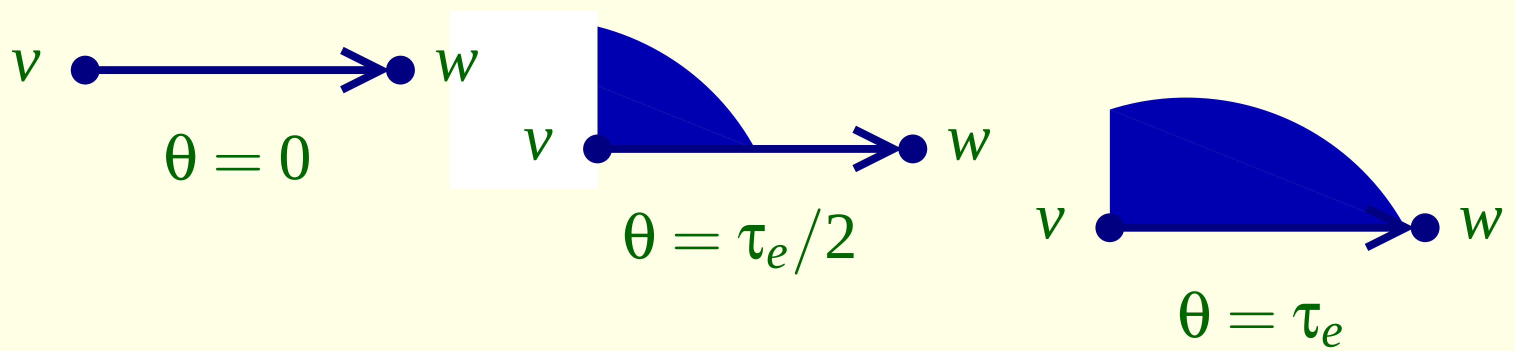
Definition: A flow over time f with time horizon T is given by functions $f_e : [0,T) \to \mathbb{R}_{\geqslant 0}$ for all arcs $e \in A$.

Interpretation: For $\theta \in [0,T)$, the value $f_e(\theta)$ determines the rate of flow into arc e at time θ.

Example

Consider an arc $e = (v, w)$.





Thus, the rate of flow arriving at the head w of arc e at time $\theta \geqslant \tau_e$ is $f_e(\theta - \tau_e)$.

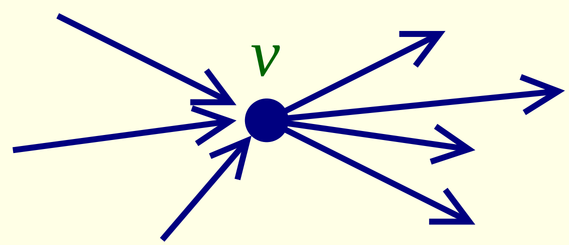
Capacities and Flow Conservation

In the setting of flows over time . . .

. . . capacities of arcs limit the rate of flow into arcs.

. . . flow conservation constraints prohibit deficit at nodes $v \neq s$ at any time $\theta \in [0, T)$:

$$\sum_{e \in \delta^-(v)} \int_{\tau_e}^{\theta} f_e(\xi - \tau_e)\ d\xi \ \geqslant \ \sum_{e \in \delta^+(v)} \int_0^{\theta} f_e(\xi)\ d\xi \ .$$



If "$>$" holds at some time, flow is stored at node v.

We require that "=" holds at time T and

$$f_e(\theta) = 0 \qquad \text{for } \theta \geqslant T - \tau_e, \text{ for all } e \in A.$$

That is, no flow remains in the network after time T.

The value of a flow over time f is

$$\begin{aligned} |f| \;:=\; & \sum_{e \in \delta^+(s)} \int_0^T f_e(\theta)\, d\theta \;-\; \sum_{e \in \delta^-(s)} \int_{\tau_e}^T f_e(\theta - \tau_e)\, d\theta \\ =\; & -\sum_{e \in \delta^+(t)} \int_0^T f_e(\theta)\, d\theta \;+\; \sum_{e \in \delta^-(t)} \int_{\tau_e}^T f_e(\theta - \tau_e)\, d\theta \end{aligned}$$

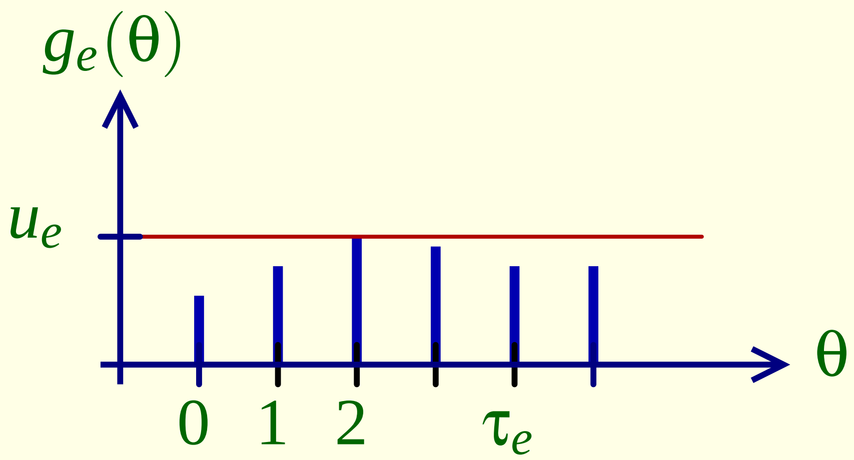
The Discrete Time Setting

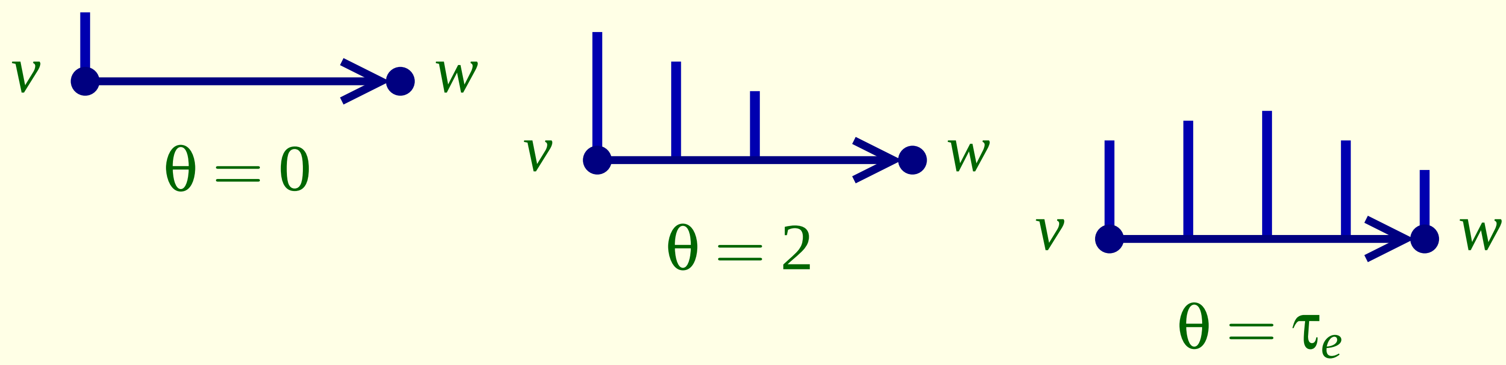
Definition: A discrete flow over time g with time horizon T is given by functions $g_e : \{0, 1, \ldots, T-1\} \to \mathbb{R}_{\geqslant 0}$ for all arcs $e \in A$.

Interpretation: For $\theta \in \{0, \ldots, T-1\}$, the value $g_e(\theta)$ describes the amount of flow entering arc e at time θ.

Example

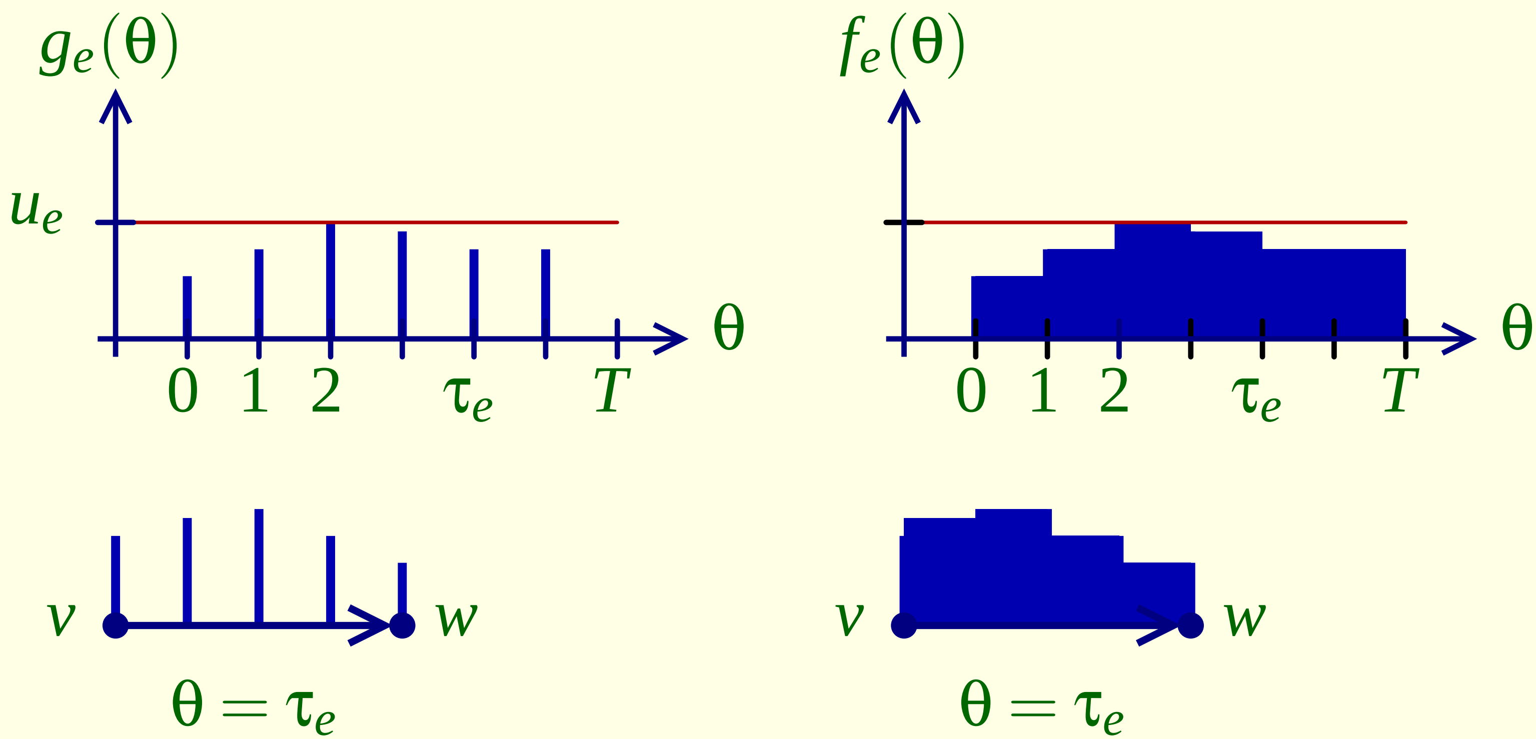
Consider an arc $e = (v, w)$.





Discrete vs. Continuous Time

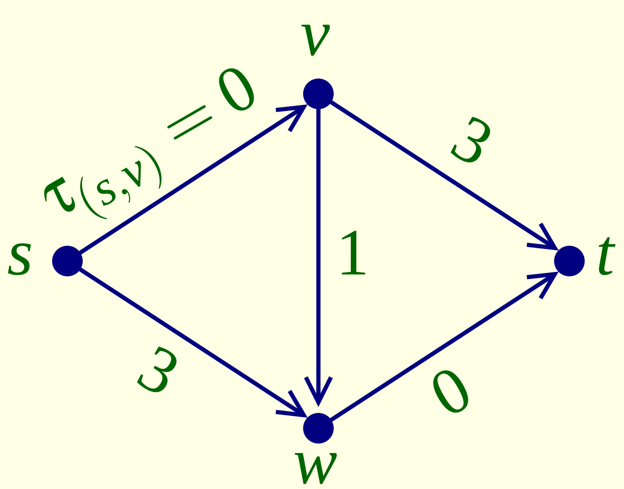
Observation. Every discrete flow over time g can be interpreted as a continuous flow over time f and vice versa (see Fleischer & Tardos 1998).

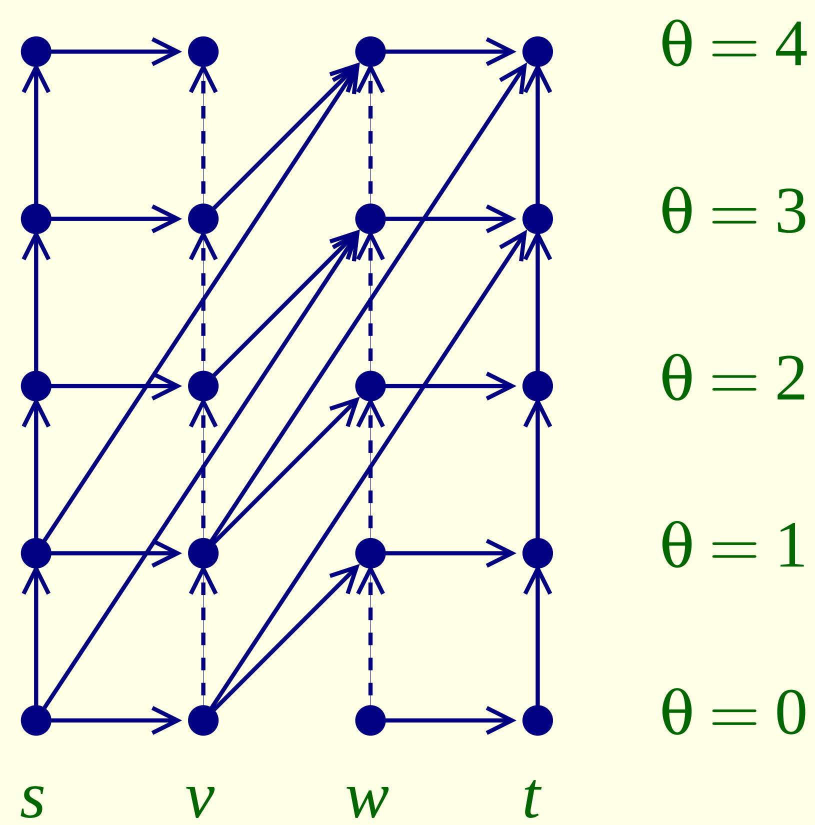


Time-Expanded Networks

Observation. Discrete flows over time correspond to static flows in time-expanded networks.

Example: $T = 5$.





Pros and Cons of Time-Expanded Networks

Pros:

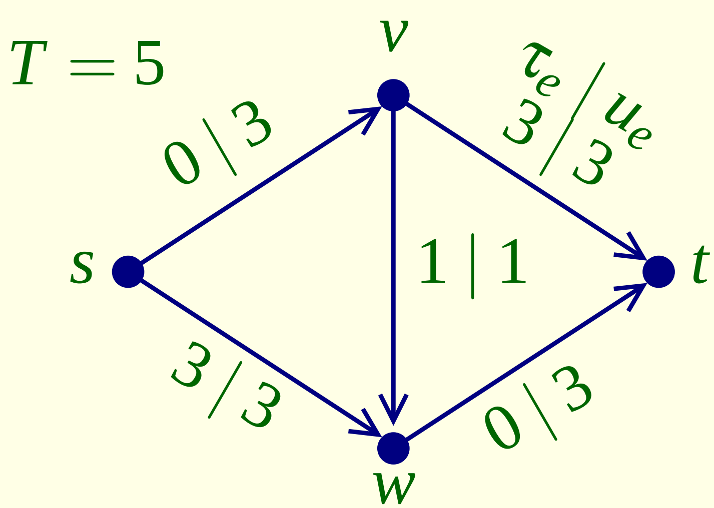
- A maximum s-t-flow over time with time horizon T can be obtained by computing a maximum static s_0-t_{T-1}-flow in the time-expanded network $\mathcal{N}^T$.
- Various more complicated flow-over-time problems can be solved by static flow algorithms in time-expanded networks.
- Thus, the entire algorithmic toolbox developed for static flows is also available for flows over time.

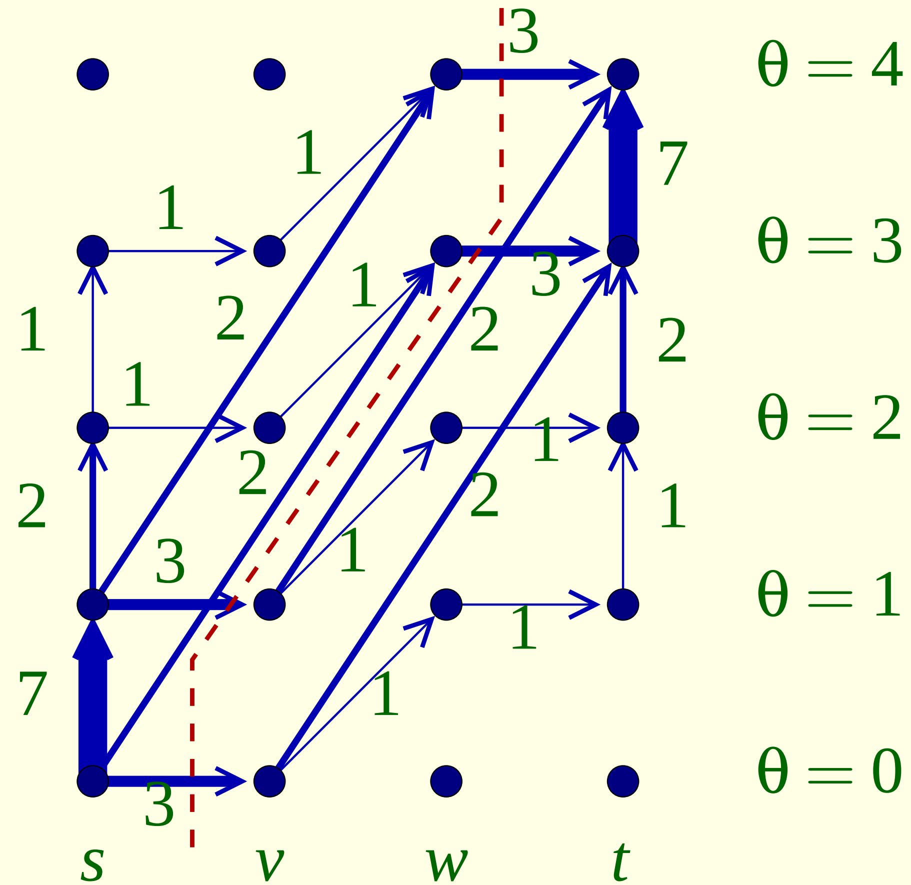
Pros and Cons of Time-Expanded Networks

Cons:

- In practice: Size of the time-expanded network $\mathcal{N}^T$ leads to huge memory requirement for computations (depending on T).
- In theory: Only pseudo-polynomial algorithms, since the size $O(|A|\,T)$ of $\mathcal{N}^T$ is pseudo-polynomial in the input size.

Example


T = 5
v
s
w
t
0 | 3
3 | 3
1 | 1
0 | 3
τe / ue
3 / 3


θ = 4
θ = 3
θ = 2
θ = 1
θ = 0
s
v
w
t
3
7
1
2
1
1
1
2
3
1
2
2
2
1
1
2
1
3
1
1
1
1
2
7
3

Ford & Fulkerson's Algorithm

Algorithm. (Ford & Fulkerson 1958)

- Compute a static flow x in $\mathcal{N}$

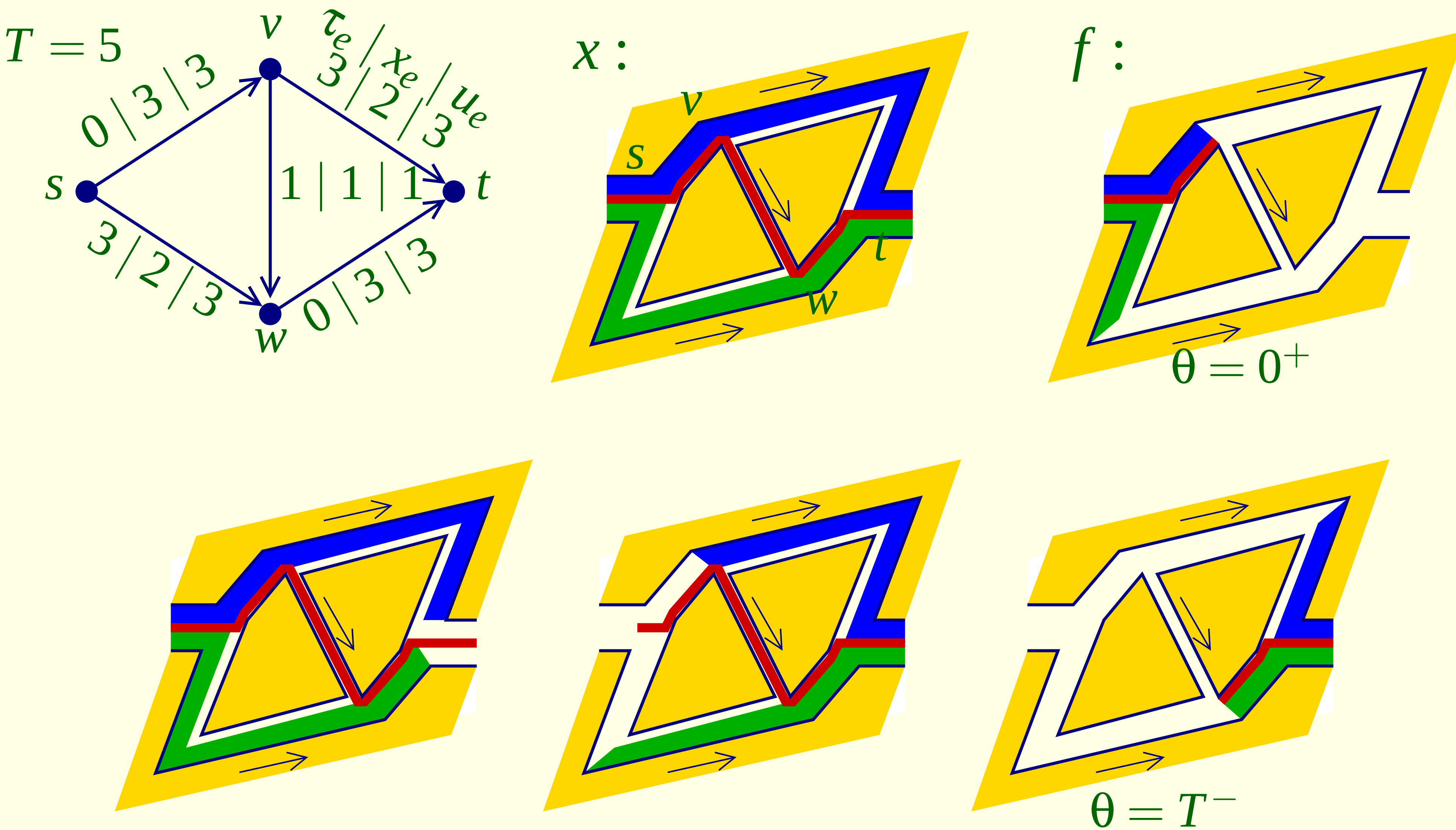
$$\text{maximizing} \quad T\,|x| - \sum_{e \in A} \tau_e x_e \ .$$

- Decompose x into flows x_P on s-t-paths $P \in \mathcal{P}$ such that

$$x_e = \sum_{\substack{P \in \mathcal{P} \\ e \in P}} x_P \qquad \text{for all } e \in A.$$

- Resulting temporally repeated flow f sends flow at constant rate x_P into paths $P \in \mathcal{P}$, as long as there is enough time left to arrive at the sink before T.

Example



Feasibility

- By construction, the temporally repeated flow f fulfills flow conservation constraints.
- Notice that f does not store flow at intermediate nodes.
- Moreover, f obeys arc capacities, since at any time θ and for any arc $e \in A$

$$f_e(\theta) \leqslant \sum_{\substack{P \in \mathcal{P} \\ e \in P}} x_P = x_e \leqslant u_e .$$

Flow value

The temporally repeated solution f sends flow at rate x_P into path $P \in \mathcal{P}$ during the time interval $[0, T - \tau_P)$, where $\tau_P := \sum_{e \in P} \tau_e$.

Thus,

$$|f| \;=\; \sum_{P \in \mathcal{P}} (T - \tau_P)\, x_P \;=\; T\,|x| \;-\; \sum_{e \in A} \tau_e\, x_e \quad .$$

Notice that x maximizes the right hand side. In particular, $\tau_P \leqslant T$ for all $P \in \mathcal{P}$.

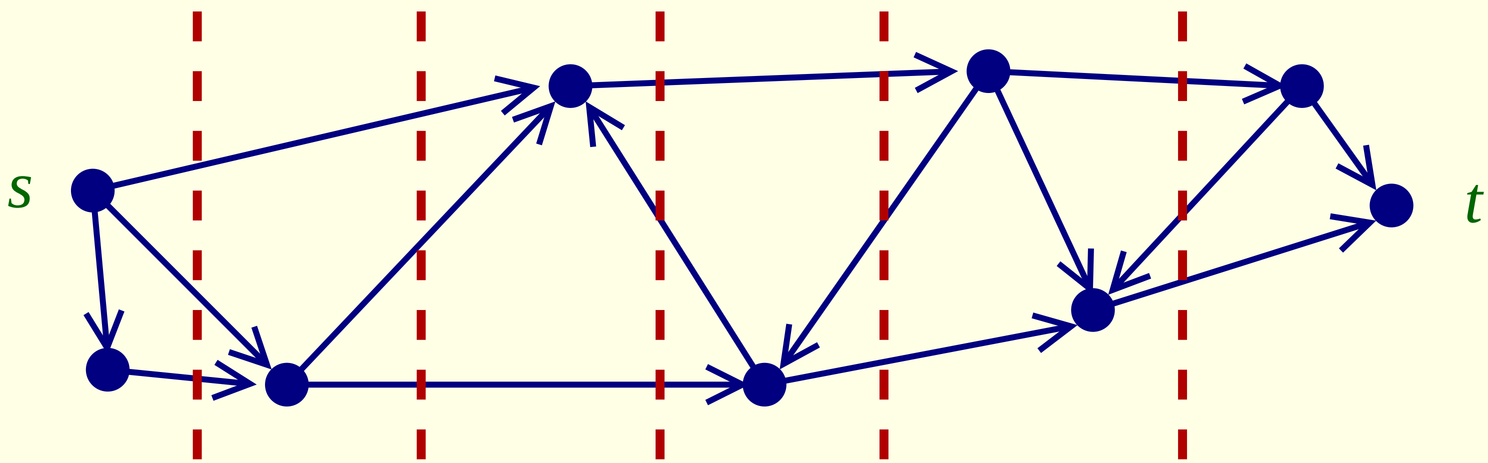
Optimality

Theorem. (Ford & Fulkerson 1958)

The temporally repeated flow f sends the maximum amount of flow from s to t within time T.

Basic idea of the proof:

Construct a matching minimum s-t-cut over time.

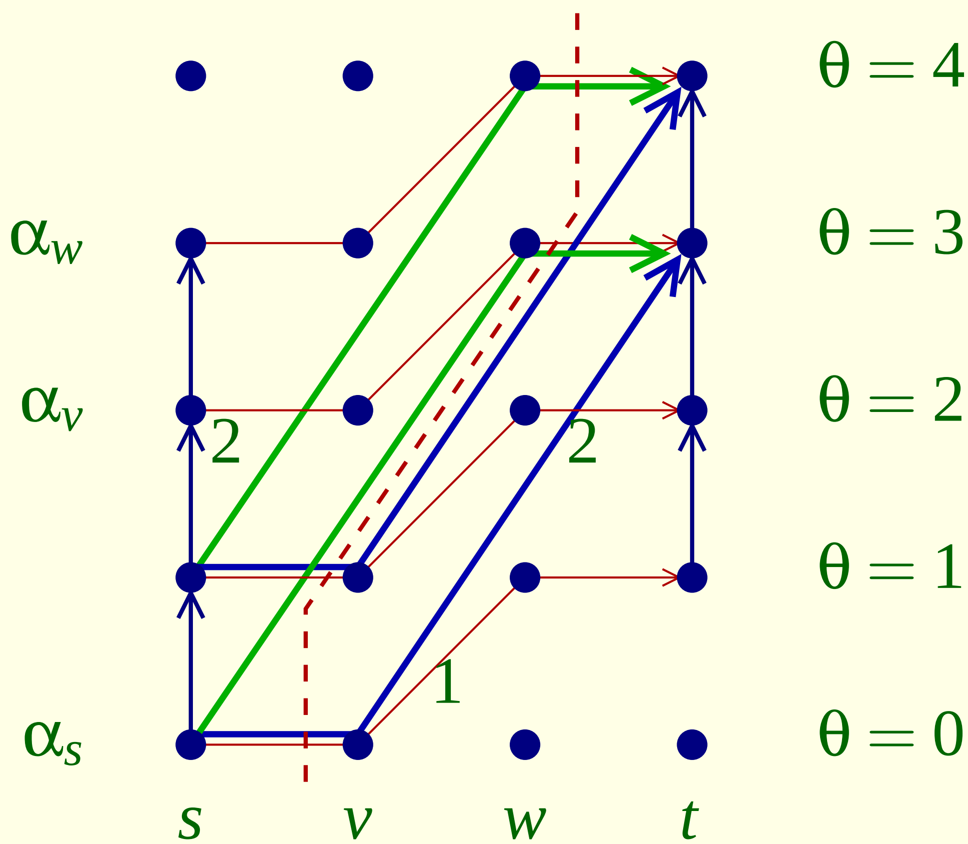


$$0 = \theta_1 < \theta_2 < \theta_3 < \theta_4 < \theta_5 < T$$

The source s is always on the left hand side of the cut and the sink t is always on the right hand side. □

Example (Cont.)

Temporally repeated flow represented in the time-expanded network together with matching minimum cut:

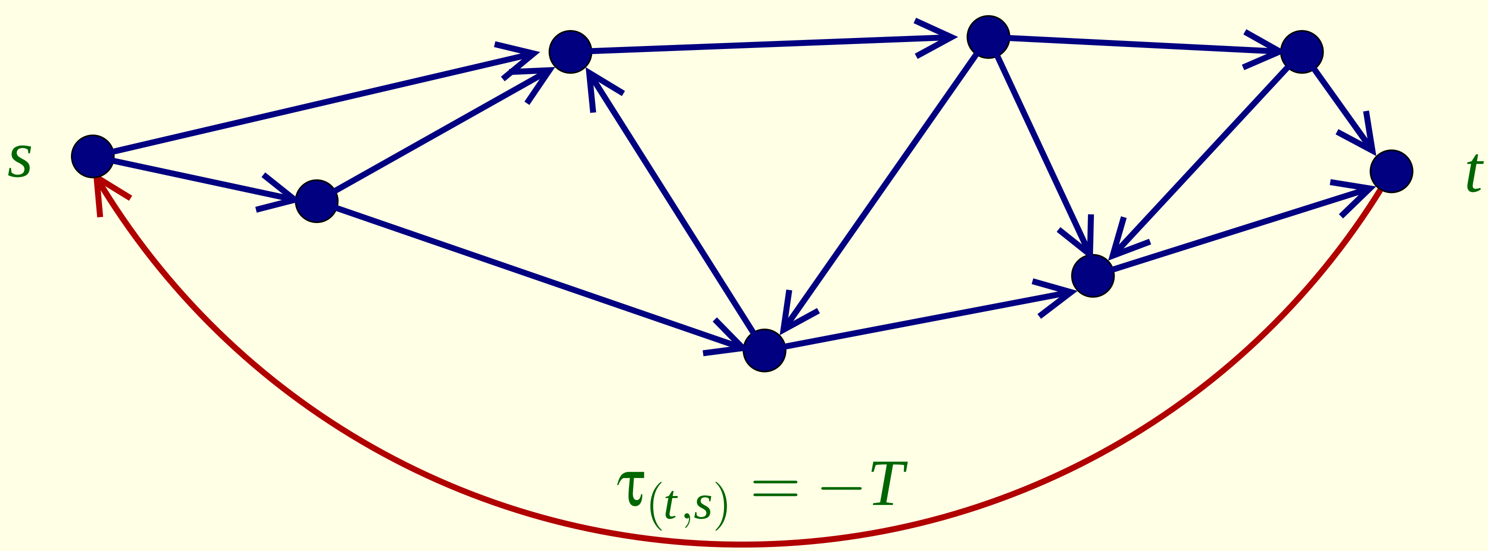


Running Time

The running time of the algorithm is dominated by the static flow computation with objective function

$$\max \quad T\,|x| \; - \; \sum_{e \in A} \tau_e x_e \quad .$$

Computing a minimum cost circulation in the following extended network with arc weights τ_e yields x.



The Complexity Landscape of Network Flows

	s-t-flow	trans-shipment	min-cost	multi-commodity
static	poly		poly	poly (LP)
dyn.	poly (static min-cost flow)	poly (minimize submodular functions)	pseudo-poly NP-hard	pseudo-poly (LP) NP-hard

Static Average Flows

Given a flow over time f with time horizon T, consider the corresponding static average flow x given by

$$x_e := \frac{1}{T}\int_0^T f_e(\theta)\ d\theta \qquad \text{for all } e \in A.$$

Then, x fulfills capacity and flow conservation constraints since f does.

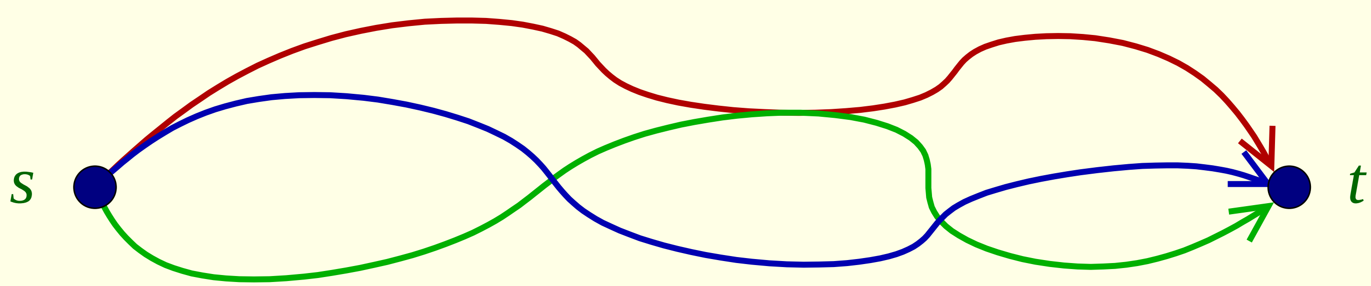
Moreover,

$$|x| = \frac{1}{T}|f| \qquad \text{and} \qquad c(x) = \frac{1}{T}c(f) \ .$$

Length-Bounded Flows

Since f has time horizon T, an s-t-path P taken by an arbitrary flow unit has length $\tau_P \leqslant T$.

Observation. The average flow x is T-length-bounded, i. e., there exists a path decomposition $(x_P)_{P \in \mathcal{P}}$ of x such that $\tau_P \leqslant T$ for all $P \in \mathcal{P}$.



A Simple Algorithm

Problem: Find a quickest s-t-flow (i. e., minimize T) satisfying demand D with cost bounded by C.

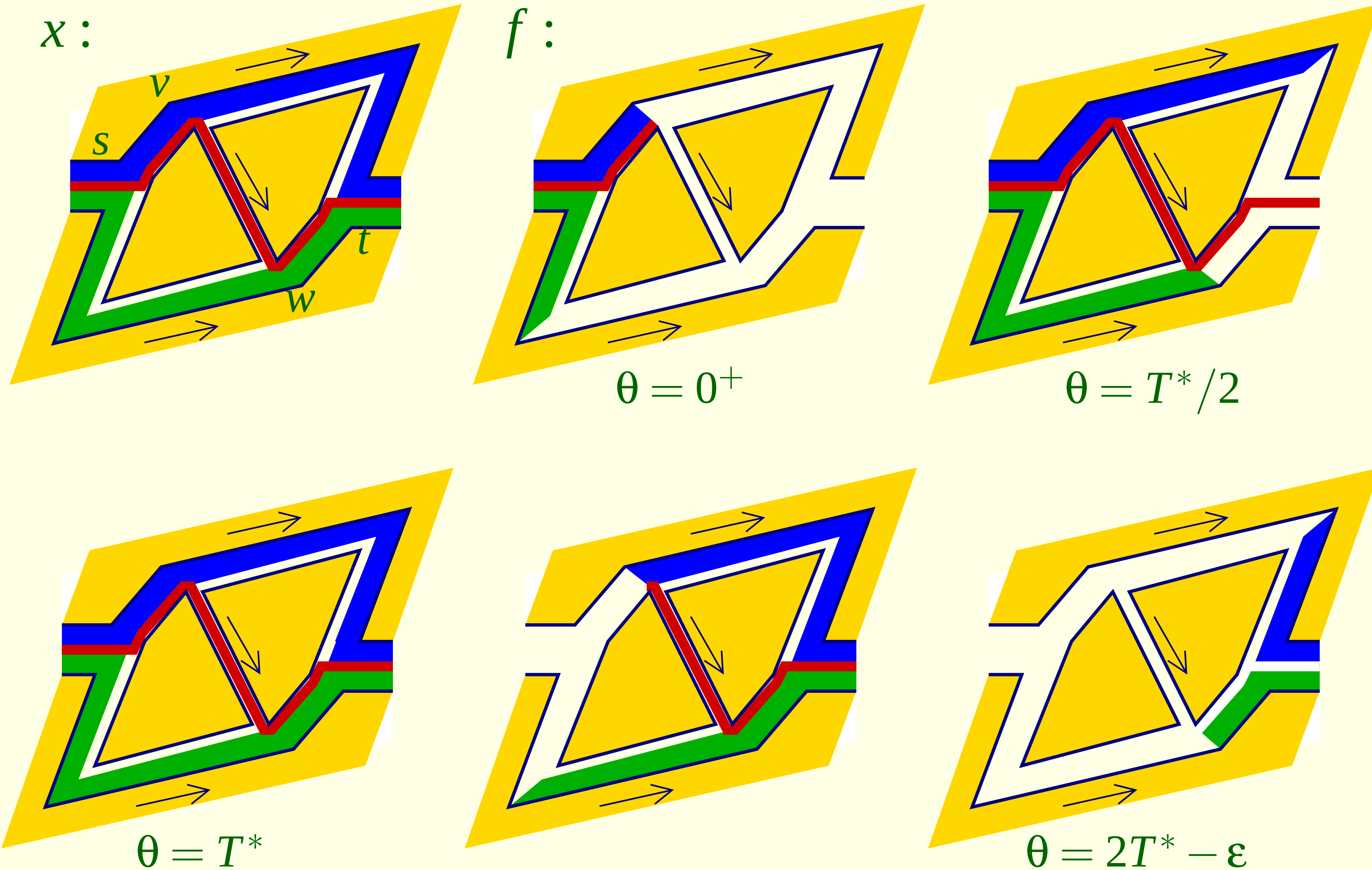
Algorithm. (Fleischer & Sk. 2002)

- Guess the optimal time horizon T^* (binary search).
- Compute a T^*-length-bounded static flow $(x_P)_{P\in\mathcal{P}}$ with

$$|x| = \frac{1}{T^*}D \qquad \text{and} \qquad c(x) \leqslant \frac{1}{T^*}C \ .$$

- Construct flow over time f by sending flow at constant rate x_P into paths $P \in \mathcal{P}$ during the time interval $[0, T^*)$. Then wait until all flow has arrived at the sink.

Example



Analysis

Feasibility:

– By construction, the flow over time f fulfills flow conservation and capacity constraints.

– Notice that f does not store flow at intermediate nodes.

Flow value:

The solution f sends flow at rate x_P into path $P \in \mathcal{P}$ for T^* time units. Thus,

$$|f| = \sum_{P \in \mathcal{P}} T^* x_P = T^* |x| = D \ .$$

Analysis (Cont.)

Cost:

$$c(f) = \sum_{P \in \mathcal{P}} c_P T^* x_P = \sum_{P \in \mathcal{P}} \sum_{e \in P} c_e T^* x_P$$

$$= T^* \sum_{e \in E} c_e \sum_{\substack{P \in \mathcal{P} \\ e \in P}} x_P = T^* \sum_{e \in E} c_e x_e = T^* c(x) \leqslant C$$

Time horizon: The flow over time f sends flow into paths $P \in \mathcal{P}$ until time T^*. Since $\tau_P \leqslant T^*$ for all $P \in \mathcal{P}$, the last unit of flow arrives at the sink before time $2T^*$.

Theorem: (Fleischer & Sk. 2002)

The algorithm achieves performance ratio 2.

Concluding Remarks

- Flows over time are of great practical importance.
- Our theoretical and practical understanding of the subject is not satisfactory:
  - examine the structure of ‘good’ flows over time;
  - develop models for flow-dependent transit times;
- Research in this direction is performed within project

  “Time-Dependent Multi-Commodity Flows: Theory and Applications”

  as part of the DFG Research Center “Mathematics for Key Technologies” in Berlin.