Flows Over Time and Submodular Function Minimization

Martin Skutella

MATHEON @ Technische Universität Berlin

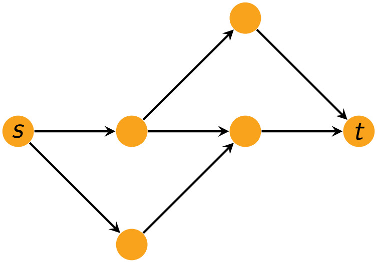
CALDAM 2017 School on Algorithms and Combinatorics

February 14th, 2017

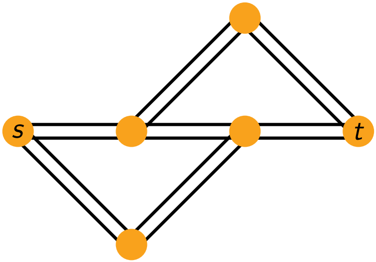
Outline

1. Short introduction to network flows over time
2. Maximum s-t-flow over time problem [Ford, Fulkerson 1958]
3. Transshipment over time and submodular functions [Schlöter, Sk. 2017]
4. Conclusion

Flows Over Time: Intuition



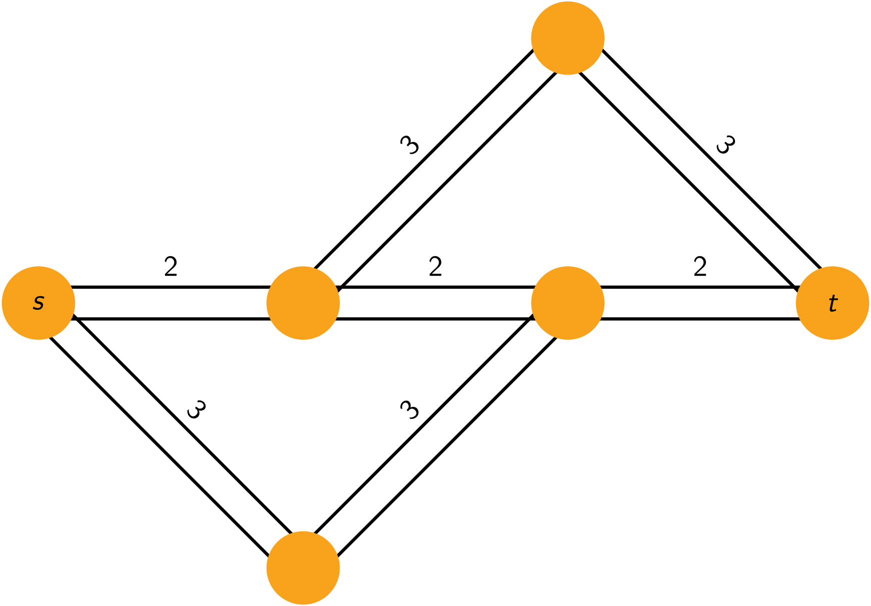
graph / network



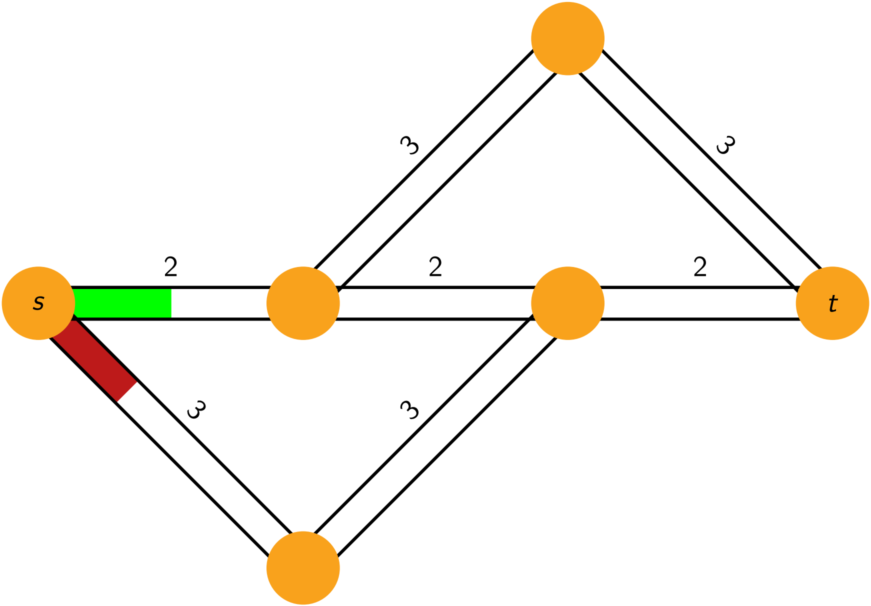
network of pipelines

flow	$\longleftrightarrow$	*fluid*
arcs	$\longleftrightarrow$	*pipes*
transit time	$\longleftrightarrow$	*length of pipe*
capacity	$\longleftrightarrow$	*width of pipe*

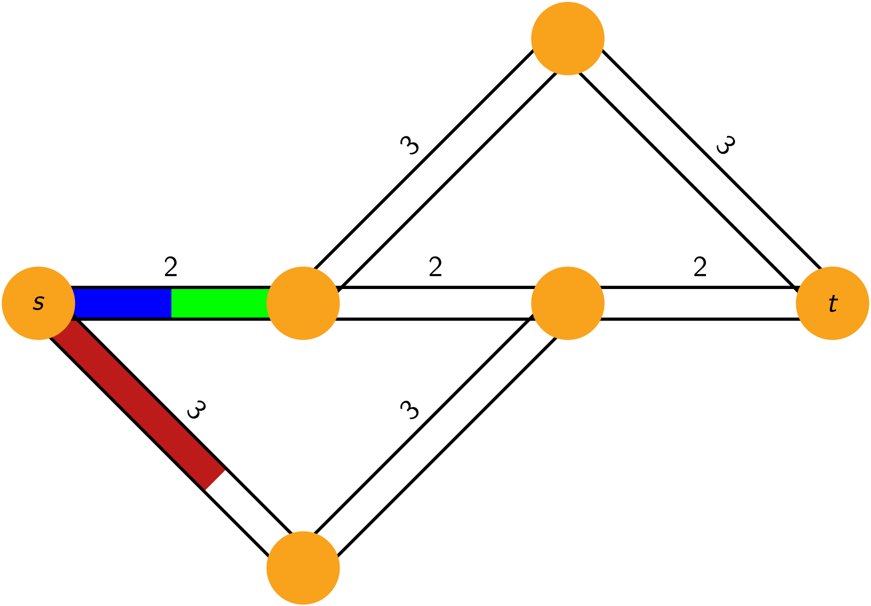
Example of an s-t-Flow Over Time



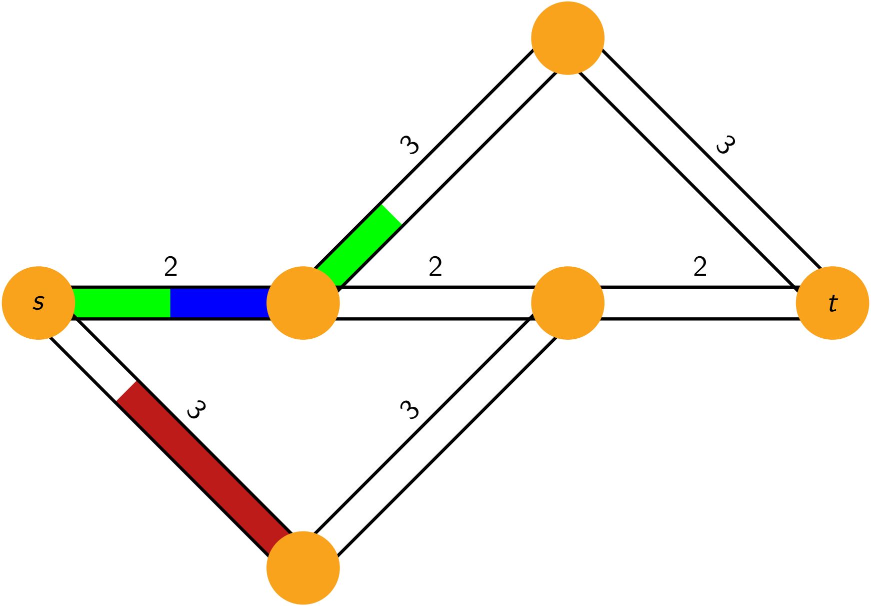
Example of an s-t-Flow Over Time



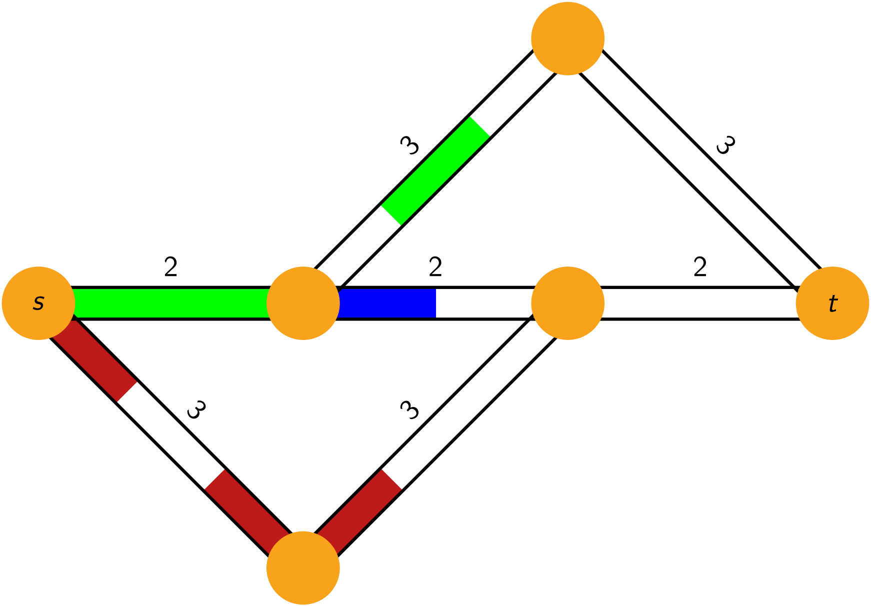
Example of an s-t-Flow Over Time



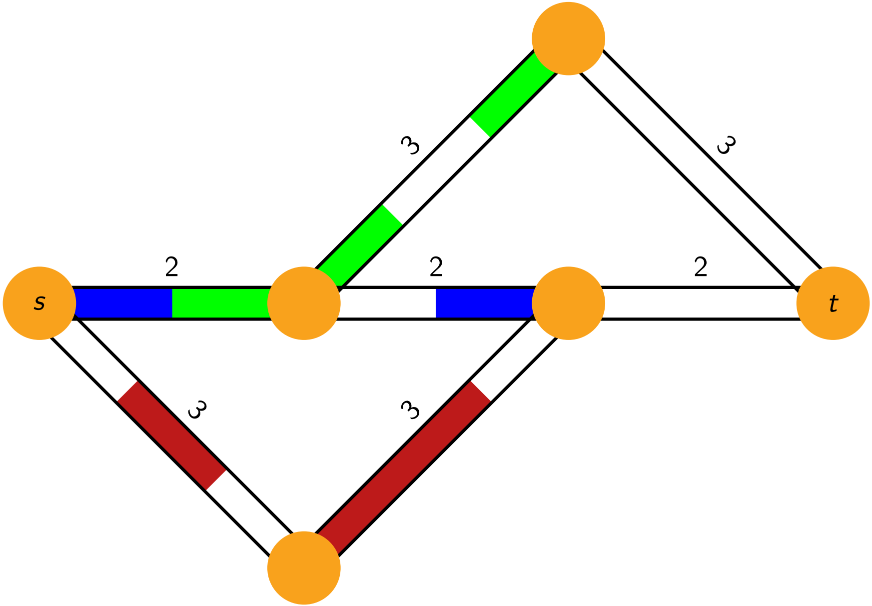
Example of an s-t-Flow Over Time



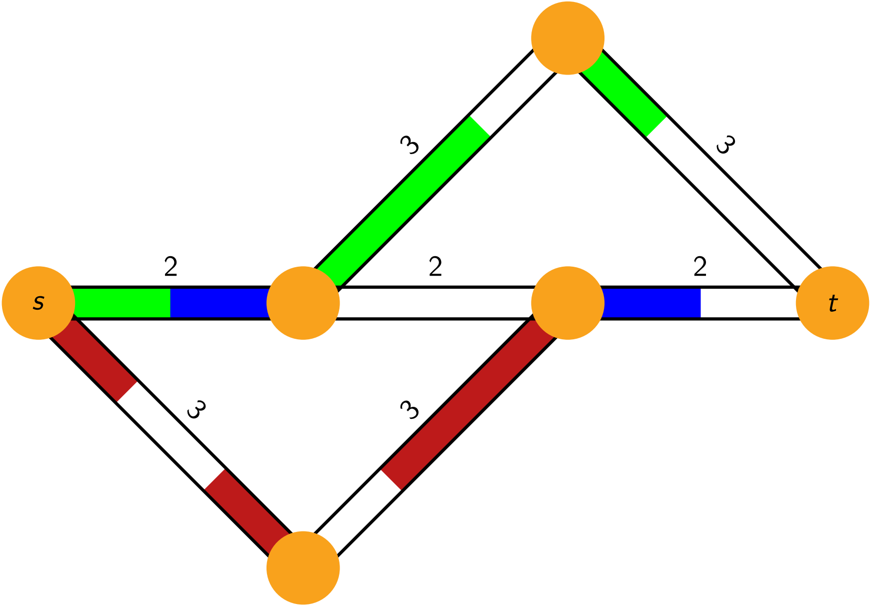
Example of an s-t-Flow Over Time



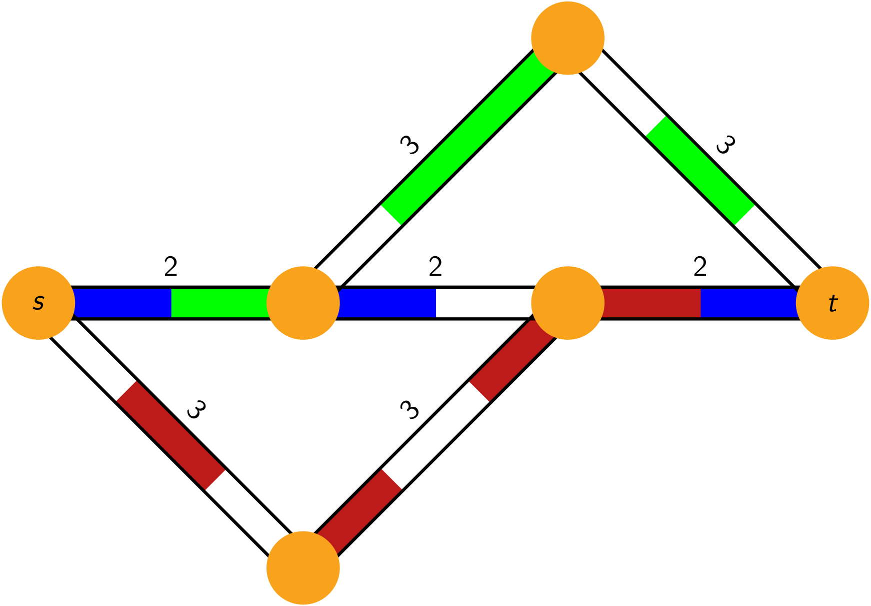
Example of an s-t-Flow Over Time



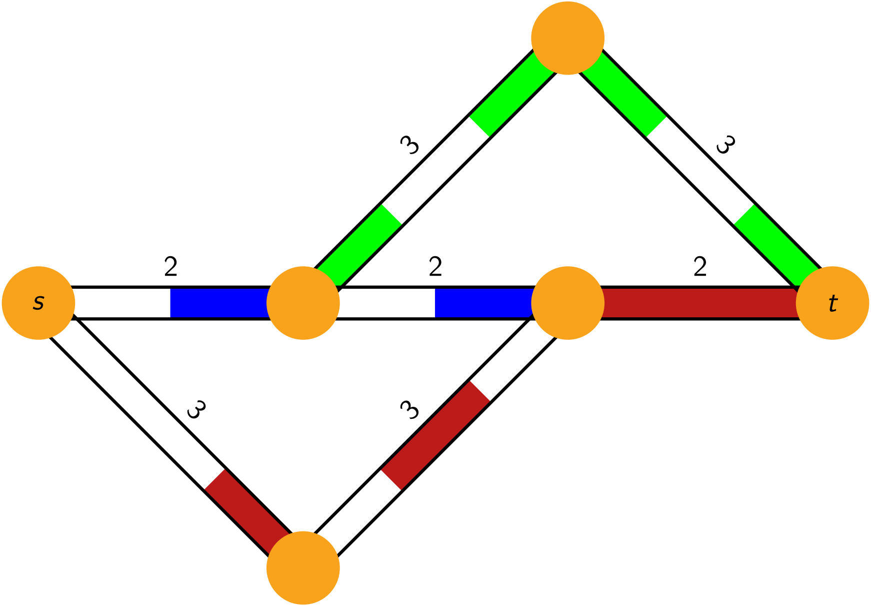
Example of an s-t-Flow Over Time



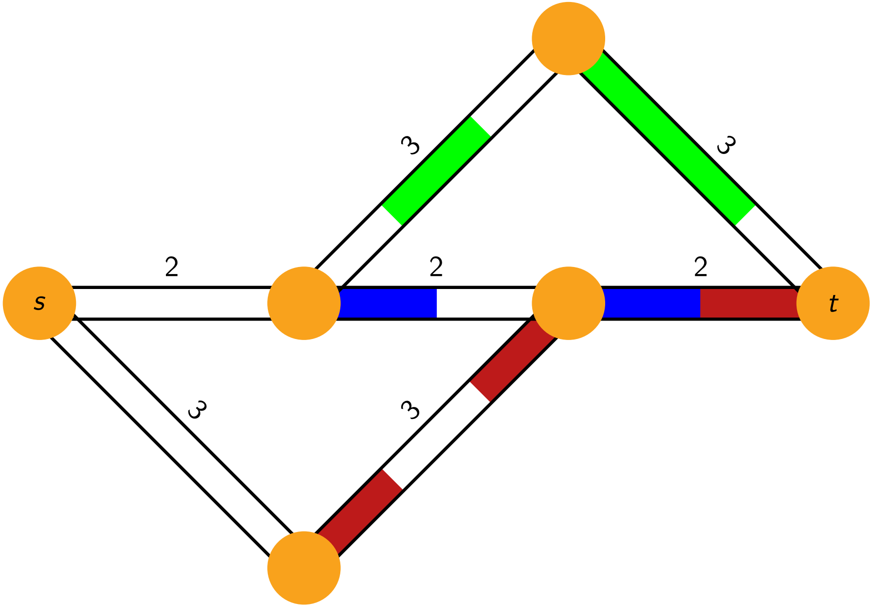
Example of an s-t-Flow Over Time



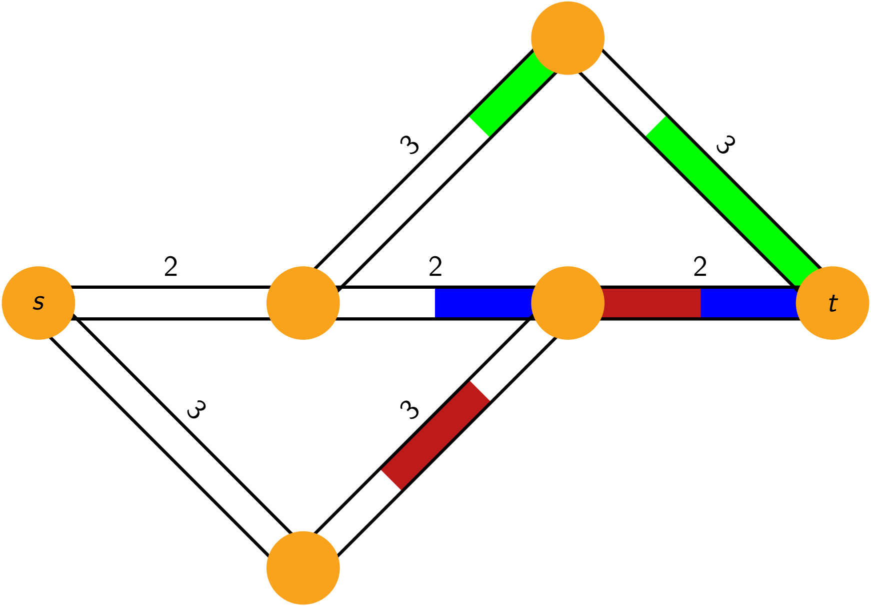
Example of an s-t-Flow Over Time



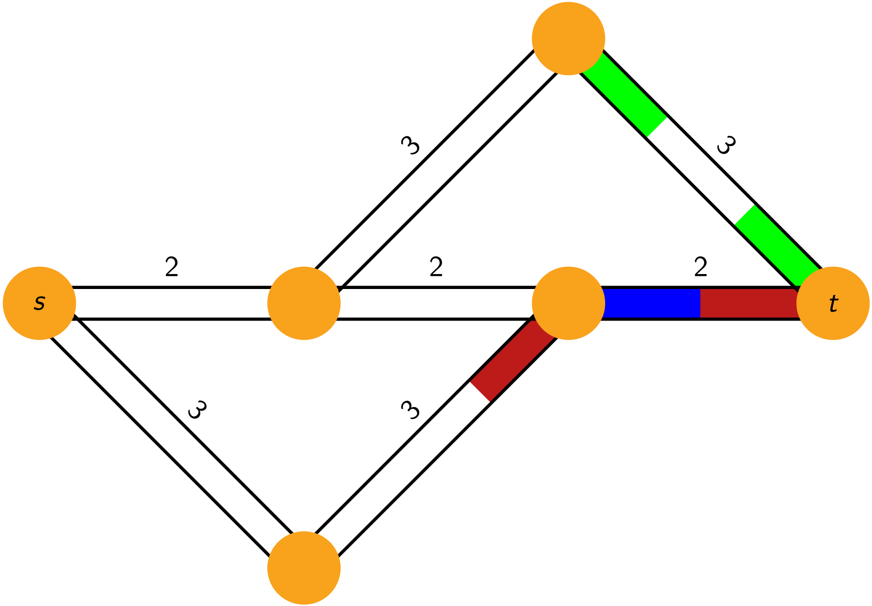
Example of an s-t-Flow Over Time



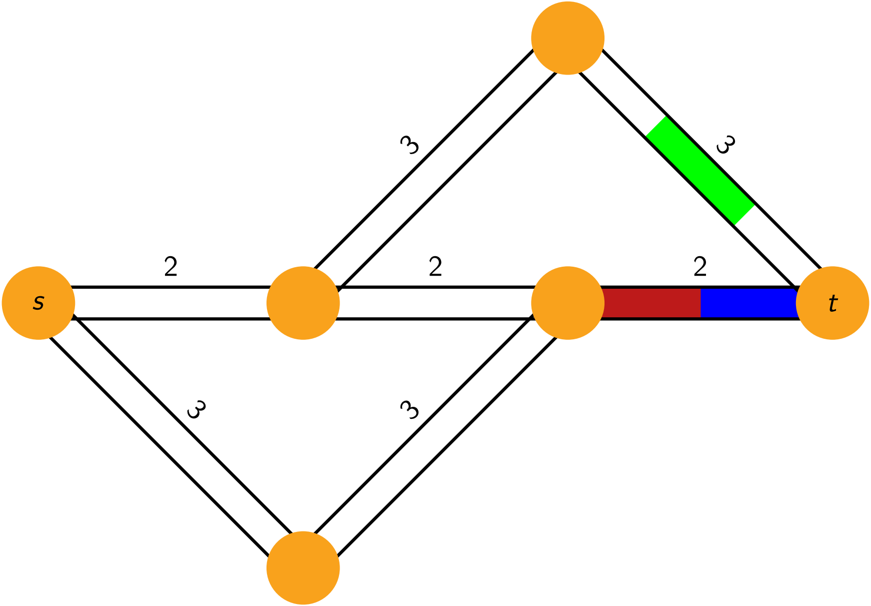
Example of an s-t-Flow Over Time



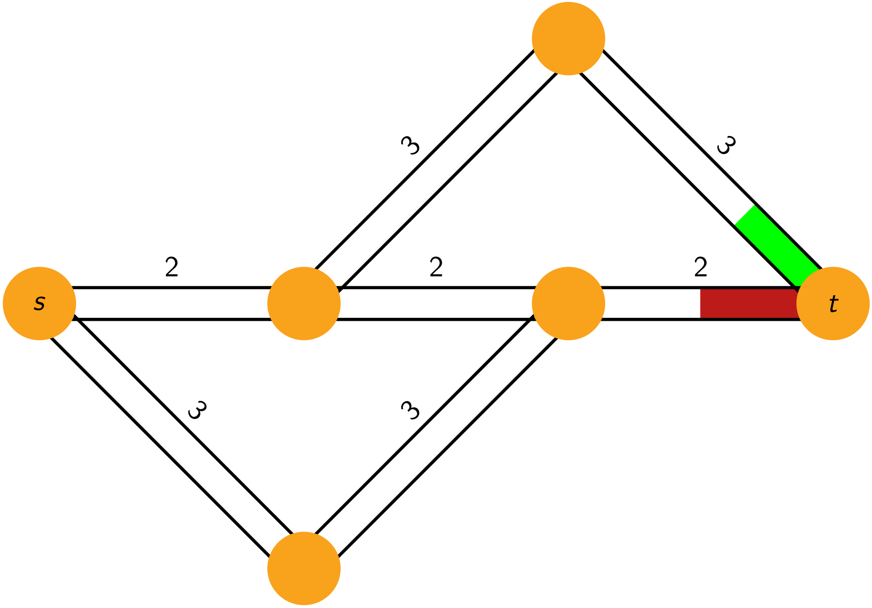
Example of an s-t-Flow Over Time



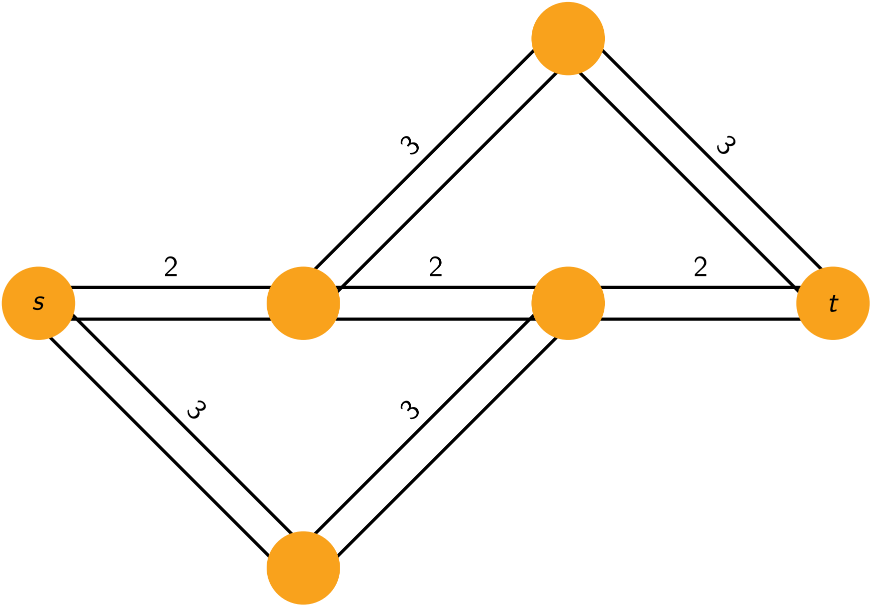
Example of an s-t-Flow Over Time



Example of an s-t-Flow Over Time



Example of an s-t-Flow Over Time



Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

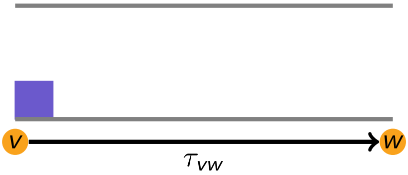
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 1$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

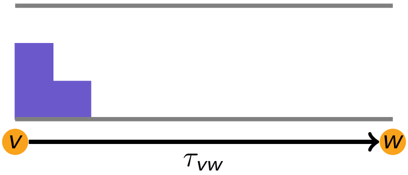
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 2$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \dots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

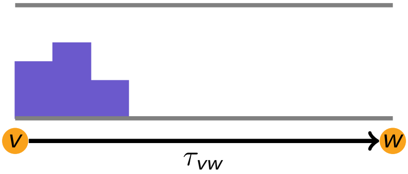
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 3$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

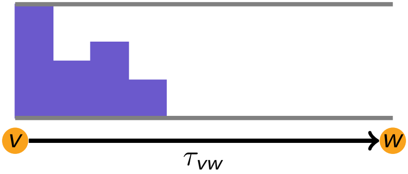
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 4$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

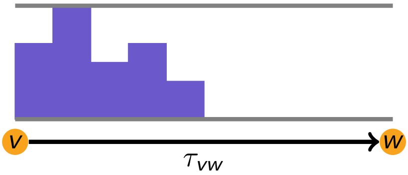
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 5$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

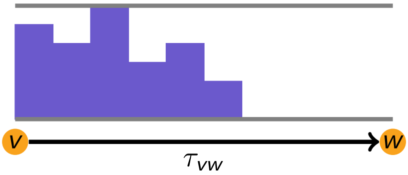
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 6$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

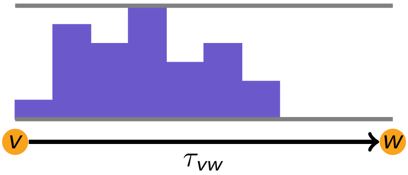
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 7$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

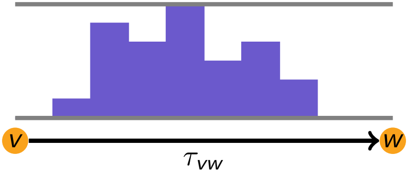
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 8$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

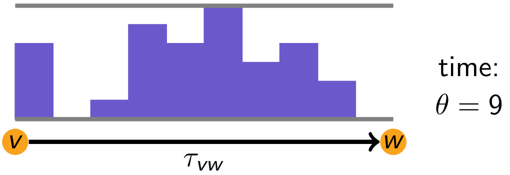
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

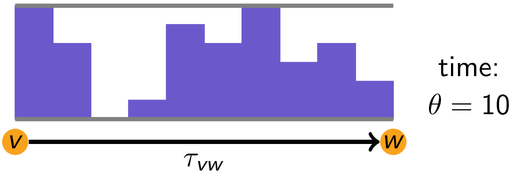
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

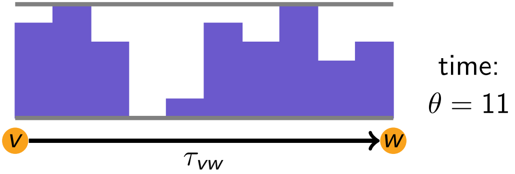
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

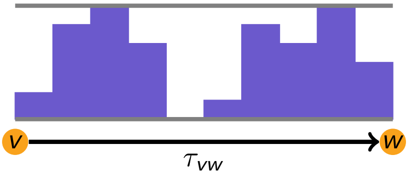
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 12$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \dots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

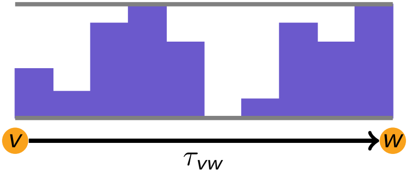
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



time:
$\theta = 13$

Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

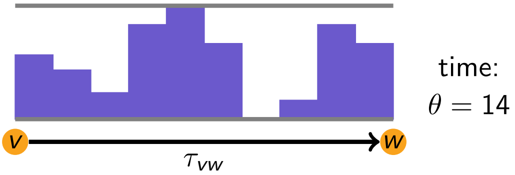
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \ldots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

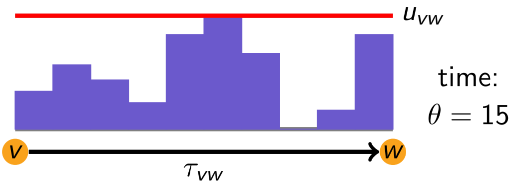
subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Flows Over Time: Definition

Given:

- digraph $D = (V, A)$
- capacities u_a, $a \in A$
- transit times τ_a, $a \in A$
- time horizon T



Definition.

A flow over time with time horizon T is a family of functions

$$f_a : \{1, \dots, T\} \to \mathbb{R}_{\geq 0}, \qquad \text{for } a \in A,$$

subject to

- $f_a(\theta) \leq u_a$ for all a, θ (capacity constraints),
- flow conservation at nodes.

Outline

1. Short introduction to network flows over time

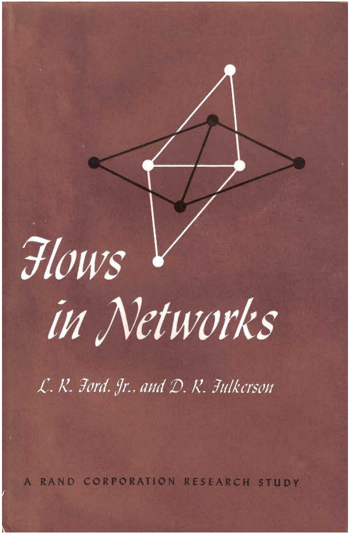
2. Maximum s-t-flow over time problem [Ford, Fulkerson 1958]

3. Transshipment over time and submodular functions [Schlöter, Sk. 2017]

4. Conclusion

The Maximum s-t-Flow Over Time Problem

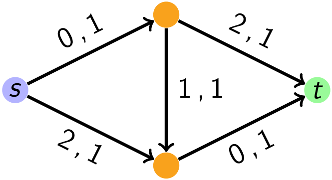
Ford & Fulkerson (1958/62) introduce *flows over time* ("dynamic flows").



The Maximum s-t-Flow Over Time Problem

Ford & Fulkerson (1958/62) introduce *flows over time* ("dynamic flows").

Given: $D = (V, A)$, $s, t \in V$, capacity u_a, transit time τ_a, time horizon T



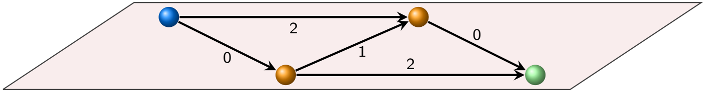
Aim: send max amount of flow from source s to sink t within time T

Time Expanded Networks

Observation. (Ford & Fulkerson 1958/62)

Flows over time correspond to static flows in time-expanded networks.

Example:

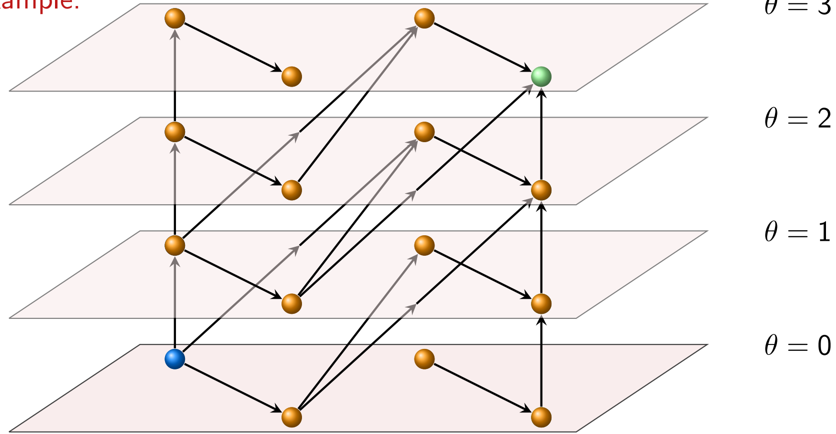


Time Expanded Networks

Observation. (Ford & Fulkerson 1958/62)

Flows over time correspond to static flows in time-expanded networks.

Example:



Pros and Cons of Time Expanded Networks

Pros:

- Many 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.

Cons:

- In practical applications: Size of the time-expanded network leads to huge memory requirement for computations (depending on T).
- In theory: Only pseudo-polynomial algorithms, since the size of the time-expanded network is pseudo-polynomial in the input size.

Fleischer & Sk. (2007), . . . :
Small 'condensed' time-expanded networks of provable quality.

Computing Maximum s-t-Flows Over Time Efficiently

Algorithm. [Ford, Fulkerson 1958]

1. compute *static s-t-flow* x in G

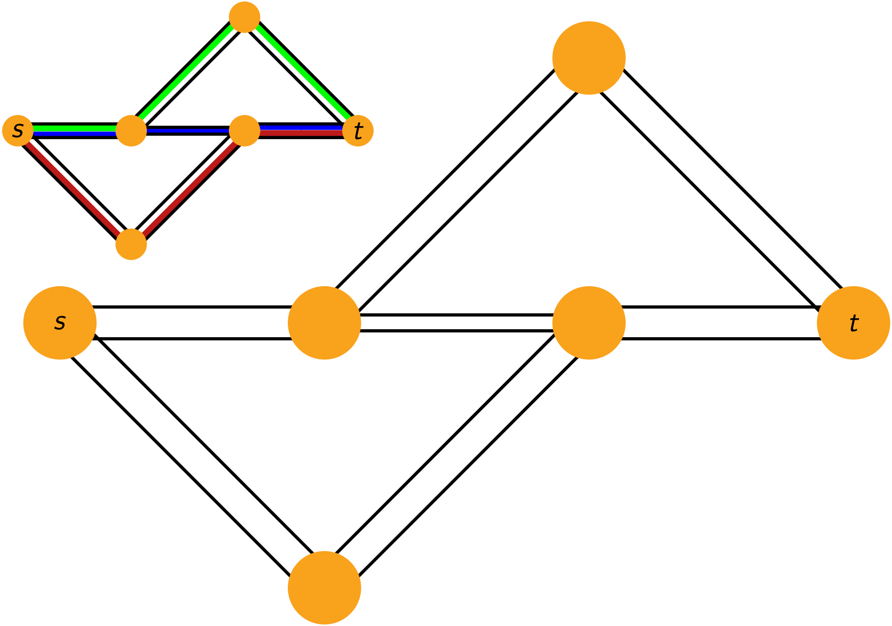
$$\text{maximizing} \quad T\,|x| \; - \; \sum_{a \in A} \tau_a \, x_a$$

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

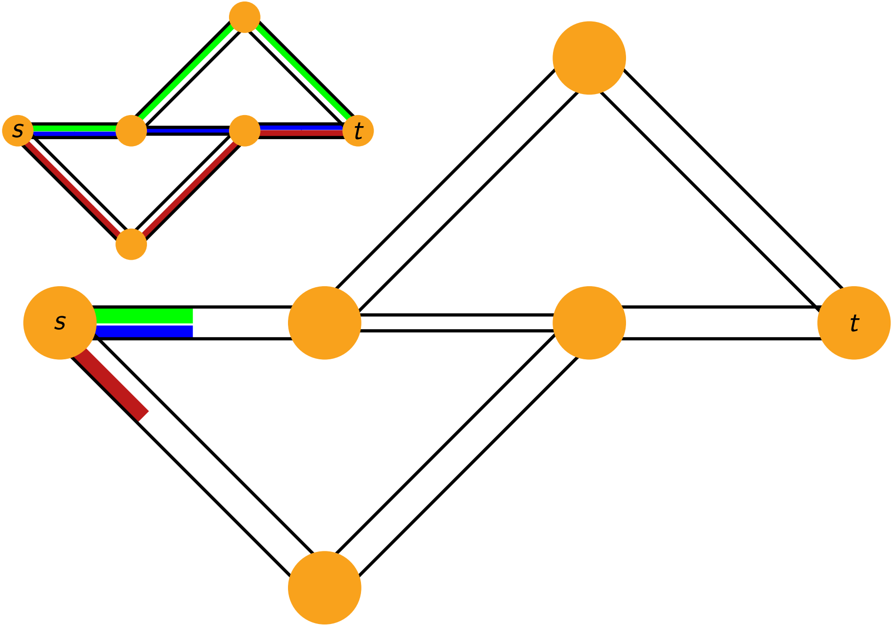
$$x_a \;=\; \sum_{\substack{P \in \mathcal{P} \\ a \in P}} x_P \qquad \text{for all } a \in A$$

3. send flow at rate x_P into paths $P \in \mathcal{P}$, as long as there is enough time left to arrive at the sink before time T

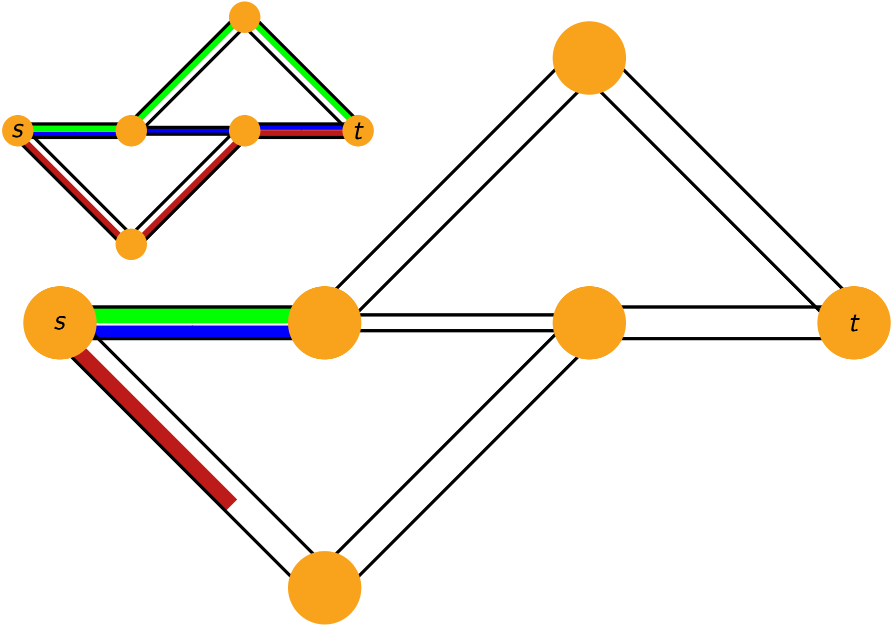
Maximum s-t-Flow Over Time



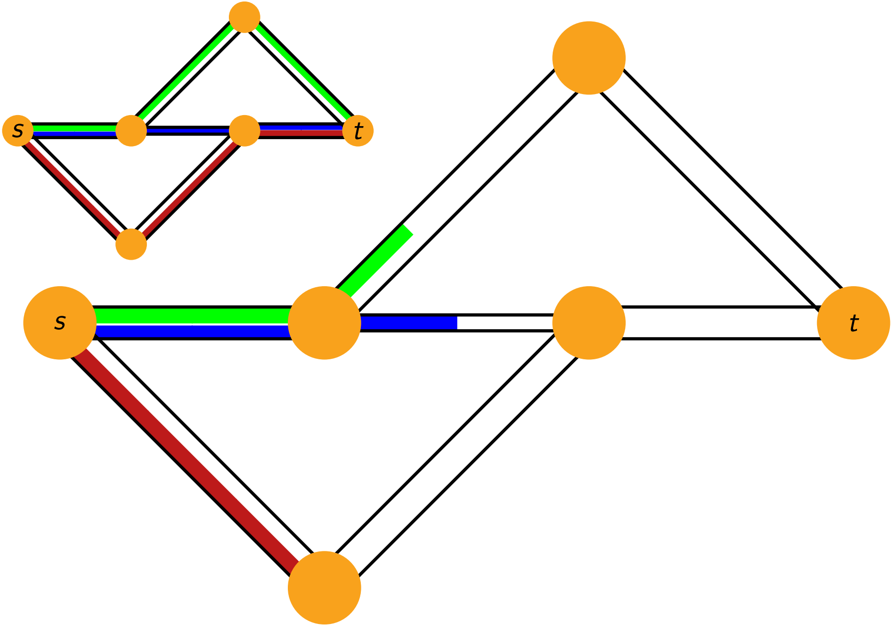
Maximum s-t-Flow Over Time



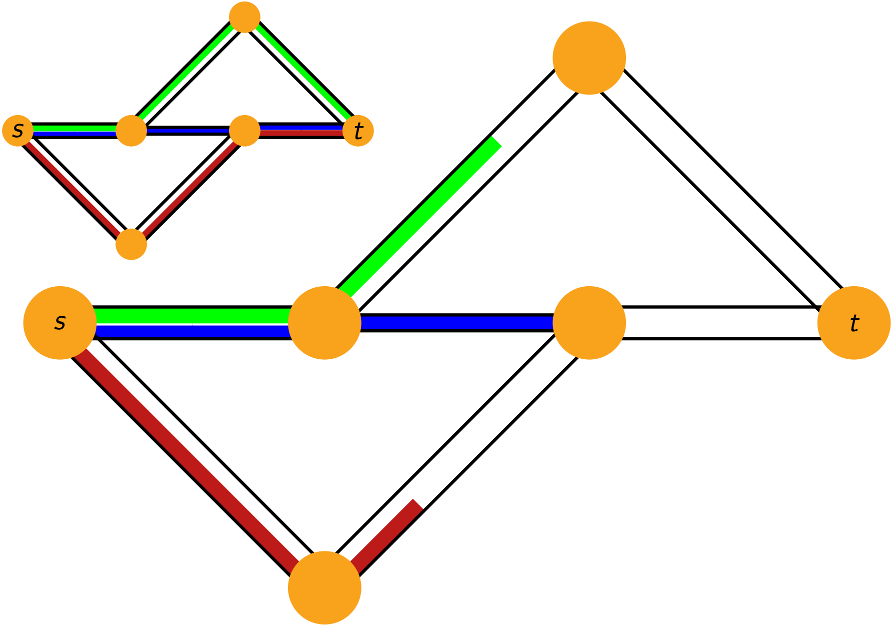
Maximum s-t-Flow Over Time



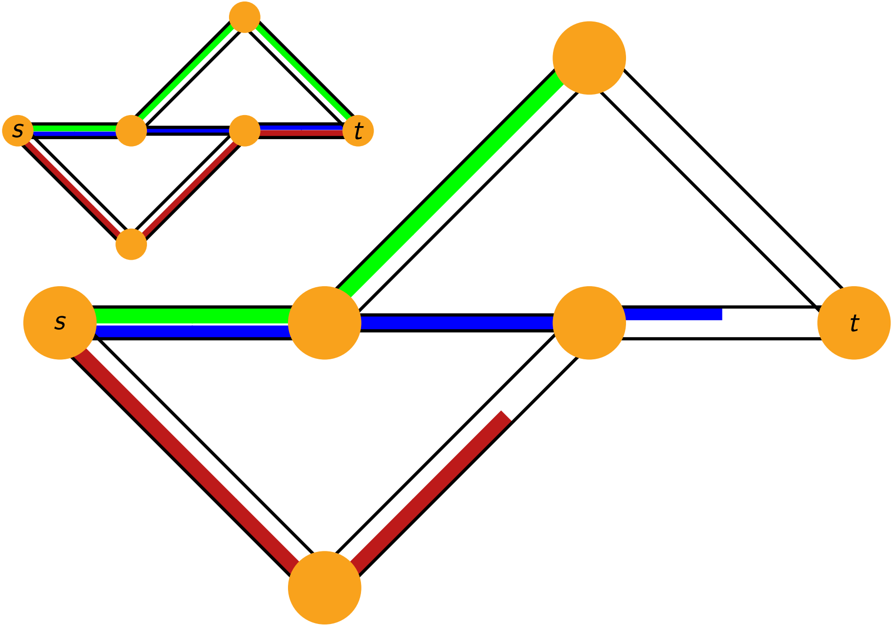
Maximum s-t-Flow Over Time



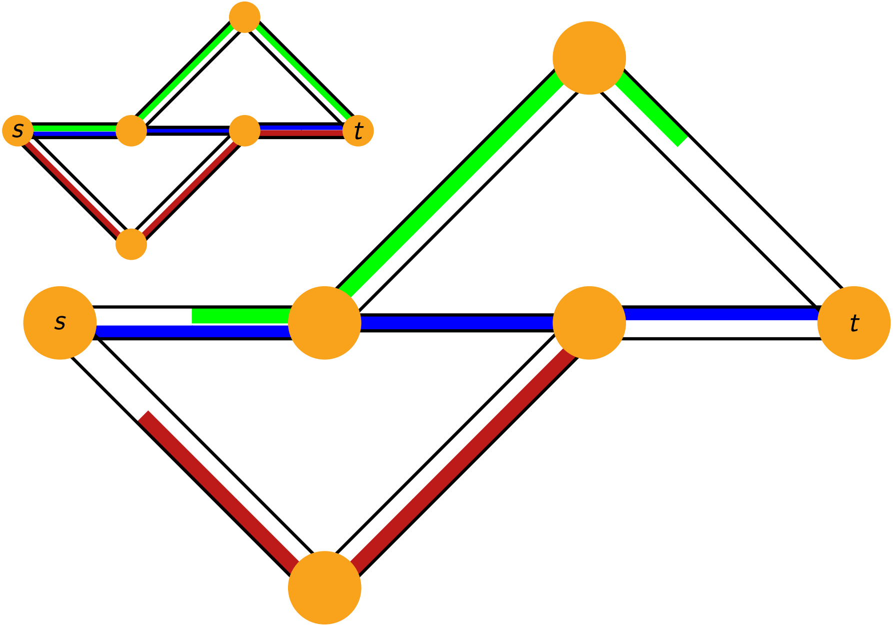
Maximum s-t-Flow Over Time



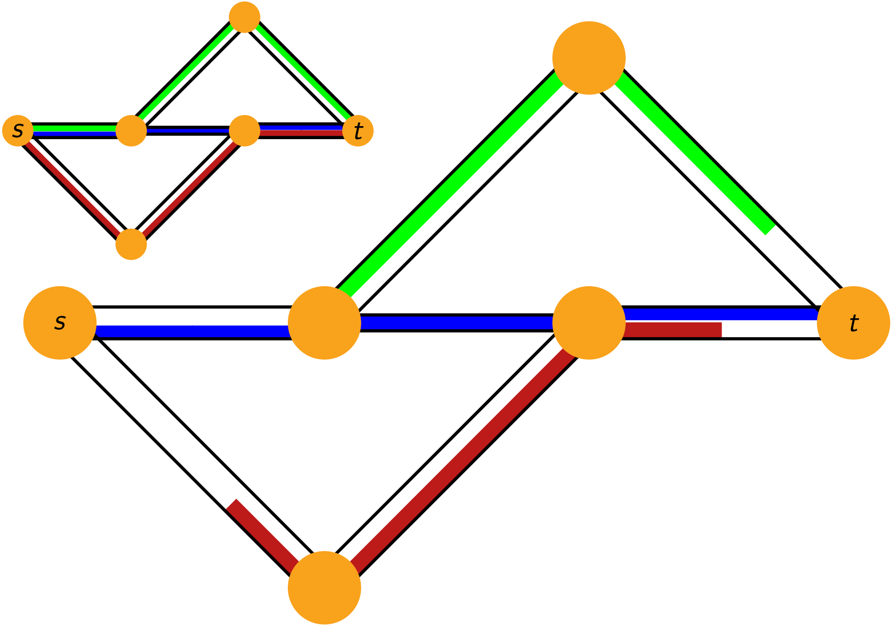
Maximum s-t-Flow Over Time



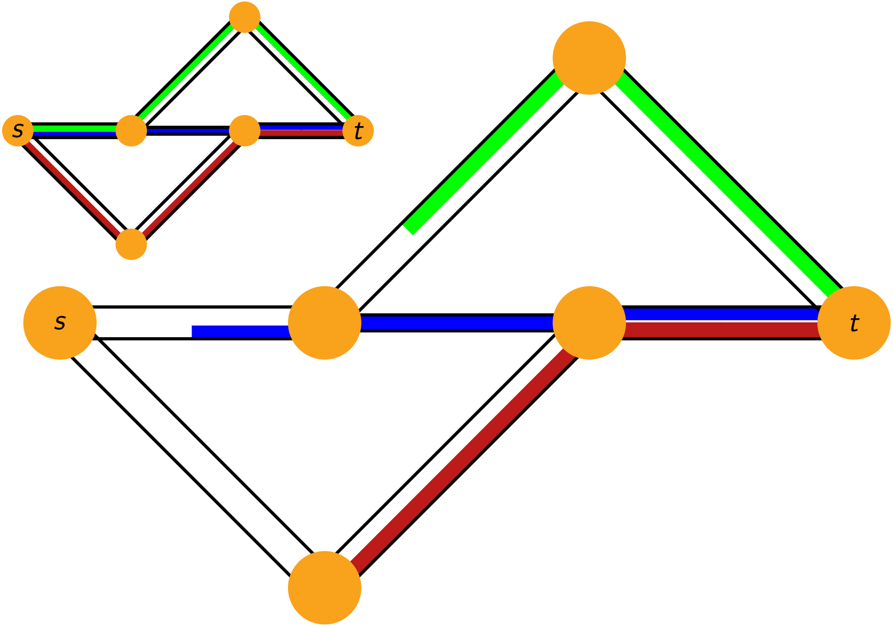
Maximum s-t-Flow Over Time



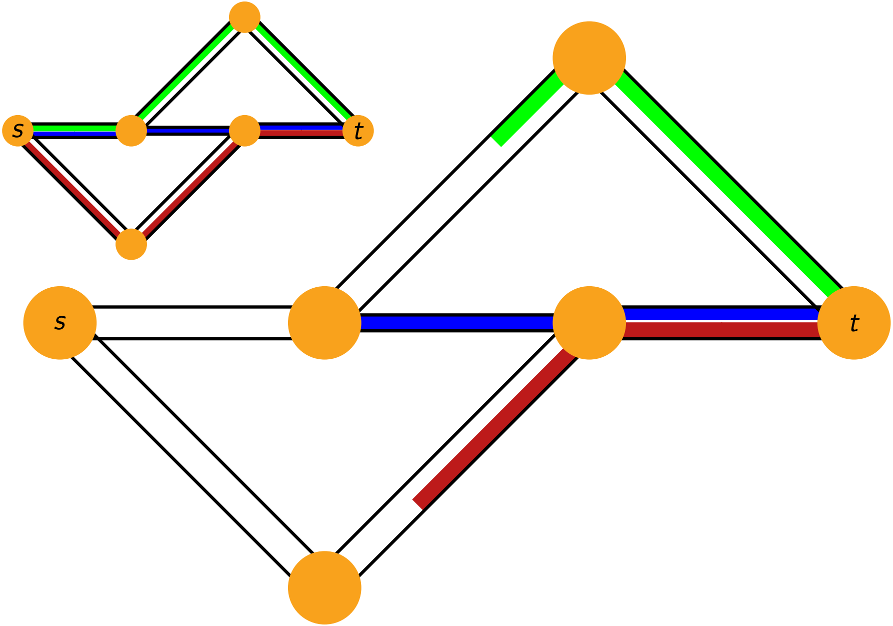
Maximum s-t-Flow Over Time



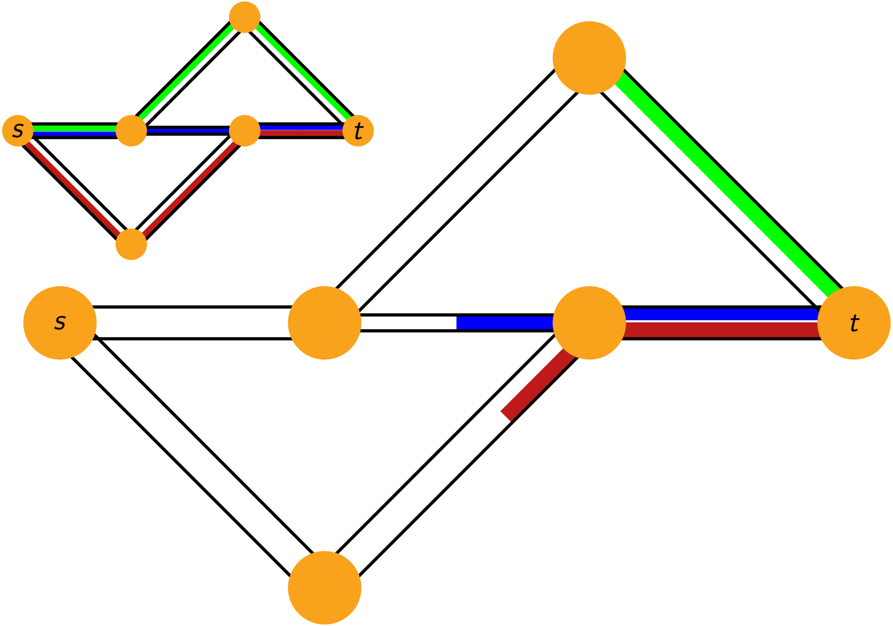
Maximum s-t-Flow Over Time



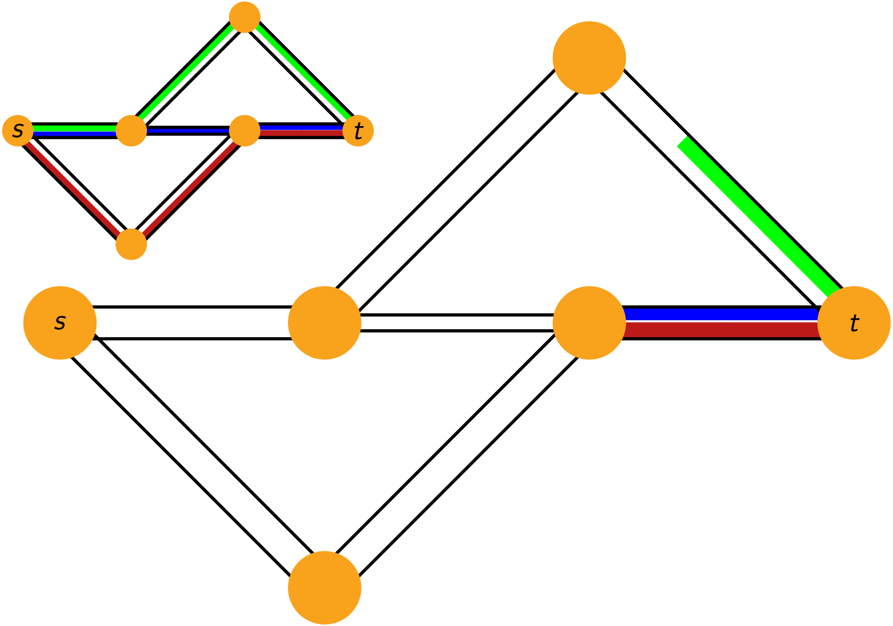
Maximum s-t-Flow Over Time



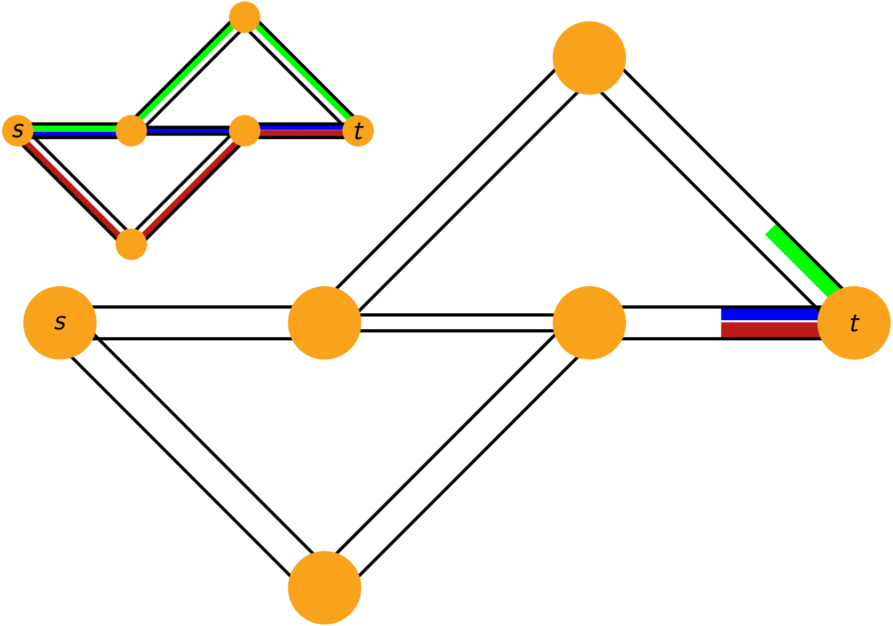
Maximum *s*-*t*-Flow Over Time



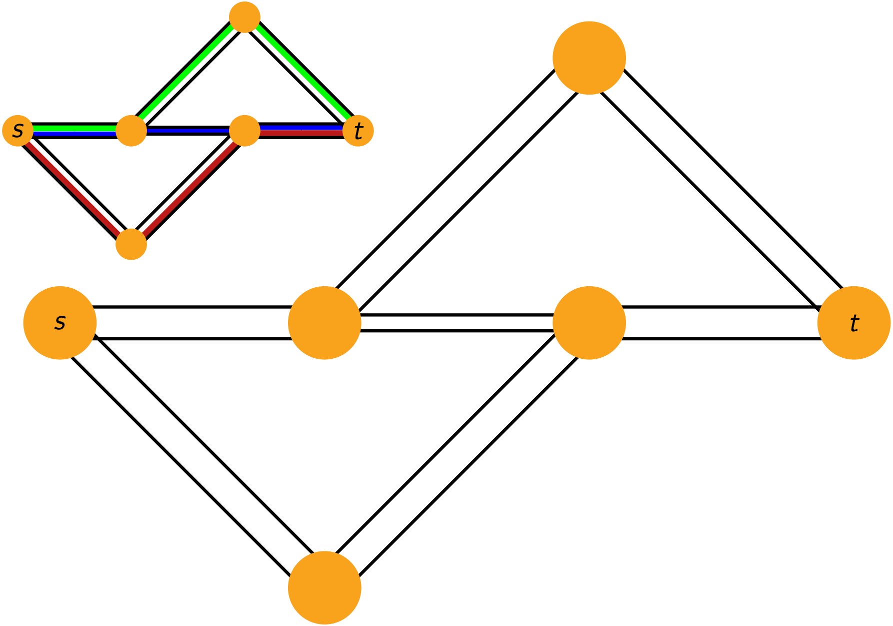
Maximum s-t-Flow Over Time



Maximum s-t-Flow Over Time



Maximum s-t-Flow Over Time



Proof of Optimality

Theorem. [Ford, Fulkerson 1958]

1. The resulting s-t-flow over time f has maximum value.
2. The running time of the algorithm is dominated by the (static) min-cost flow computation in step 1.

Proof: f has flow value

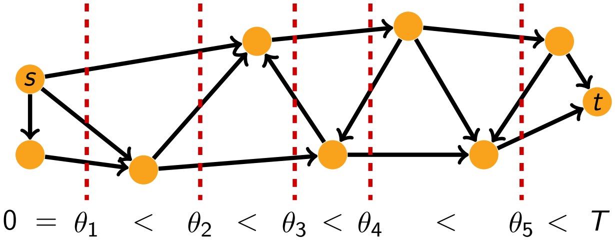
$$|f| = \sum_{P \in \mathcal{P}} (T - \tau_P)\, x_P = T\,|x| - \sum_{a \in A} \tau_a\, x_a .$$

Notice that x maximizes the right hand side (step 1)...

Proof (cont.): *s*-*t*-Cuts Over Time

Definition.

An *s-t-cut over time* is given by threshold values $\alpha_v \in [0, \infty)$ for all $v \in V$ with $\alpha_s = 0$ and $\alpha_t \geq T$. A node $v \in V$ belongs to the *right hand side* until time α_v, and afterwards to the *left hand side* of the cut.

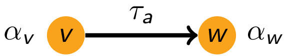


Proof (cont.): s-t-Cuts Over Time

Definition.

An *s-t-cut over time* is given by threshold values $\alpha_v \in [0, \infty)$ for all $v \in V$ with $\alpha_s = 0$ and $\alpha_t \geq T$. A node $v \in V$ belongs to the *right hand side* until time α_v, and afterwards to the *left hand side* of the cut.

Observation. Flow can cross the cut from left to right on arc $a = (v, w)$ during time interval $[\alpha_v, \alpha_w - \tau_a)$.



Definition and Lemma.

The *capacity* of an s-t-cut over time is

$$\sum_{a=(v,w)\in A} u_a \max\{0, \alpha_w - \tau_a - \alpha_v\} \ .$$

This is an upper bound on the maximum flow over time value.

Max Flow Over Time = Min Cut Over Time

The value of the s-t-flow over time computed above equals

$$\begin{aligned}
&\max\ T\cdot|x| - \sum_{a\in A}\tau_a\cdot x_a && = \min \sum_{a\in A} u_a\cdot y_a \\
&\sum_{a\in\delta^+(v)} x_a - \sum_{a\in\delta^-(v)} x_a = 0 \quad \forall v && \alpha_t - \alpha_s \geq T \\
& && y_a \geq \alpha_w - \tau_a - \alpha_v \quad \forall a=(v,w) \\
&x_a \leq u_a \quad \forall a && y_a \geq 0 \quad \forall a \\
&x_a \geq 0 \quad \forall a &&
\end{aligned}$$

Notice that $y_a = \max\{0, \alpha_w - \tau_a - \alpha_v\}$ in an optimal dual solution and, w.l.o.g., $\alpha_s = 0$, $\alpha_t \geq T$.

Theorem. [Ford, Fulkerson 1958]

Maximum flow over time value equals minimum cut over time capacity.

The Complexity Landscape of Flows Over Time

	s-*t*-flow	trans-shipment	min-cost	multi-commodity
static flow	*polynomial*		*polynomial*	*polyn.* (LP)
flow over time	*polynomial* static min-cost flow [1]	*polynomial* minimize submodular functions [2,3]	NP-hard [4]	NP-hard [5]

References.

[1] Ford & Fulkerson (1958)

[2] Hoppe & Tardos (1995)

[3] Schlöter & Sk. (2017)

[4] Klinz & Woeginger (1995)

[5] Hall, Hippler & Sk. (2007)

Outline

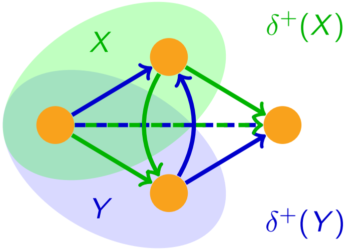
Submodular Function Minimization (SFM)

Definition. For a finite set U, function $g : 2^U \to \mathbb{R}$ is submodular if

$$g(X) + g(Y) \geq g(X \cup Y) + g(X \cap Y) \qquad \text{for all } X, Y \subseteq U.$$

Example. Cut function of network $D = (V, A)$ with capacities $u : A \to \mathbb{R}$:

$$X \mapsto u\big(\delta^+(X)\big) \qquad \text{for } X \subseteq V$$



Submodular Function Minimization (SFM)

Definition. For a finite set U, function $g : 2^U \to \mathbb{R}$ is submodular if

$$g(X) + g(Y) \geq g(X \cup Y) + g(X \cap Y) \qquad \text{for all } X, Y \subseteq U.$$

Example. Cut function of network $D = (V, A)$ with capacities $u : A \to \mathbb{R}$:

$$X \mapsto u\big(\delta^+(X)\big) \qquad \text{for } X \subseteq V$$

Submodular function minimization (SFM).

$$\min\ g(X) \text{ s.t. } X \subseteq U$$

Theorem. SFM can be solved in strongly polynomial time.

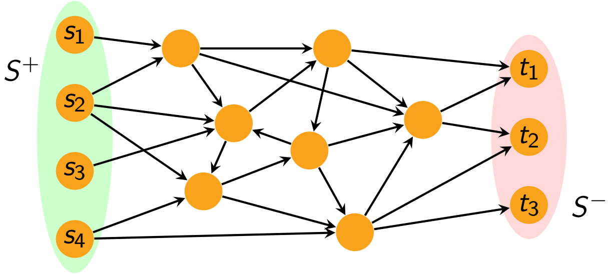
- ellipsoid method [Grötschel, Lovàsz, Schrijver 1982,1988]
- combinatorial algorithm [Schrijver 2000, Iwata et al. 2000, Orlin 2009]
- currently fastest [Lee, Sidford, Wong 2015]

Transshipment Over Time Problem

Given: $D = (V, A)$, u_a, τ_a for $a \in A$, sources/sinks $S^+, S^- \subset V$ with supplies/demands $v : S^+ \cup S^- \to \mathbb{R}$, time horizon T.

Task: Find flow over time f satisfying supplies/demands within time T.

$v(s_i) > 0$



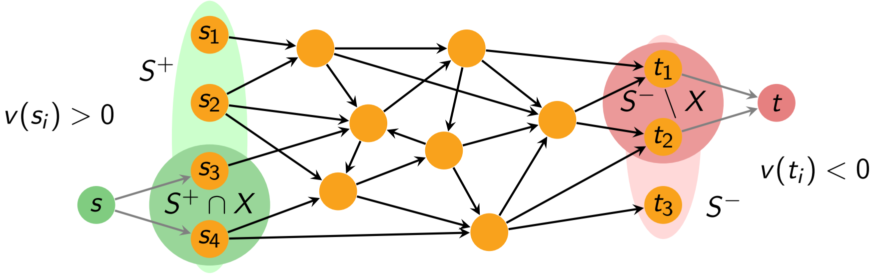
$v(t_i) < 0$

$$v(S^+ \cup S^-) = v(s_1) + \cdots + v(s_4) + v(t_1) + \cdots + v(t_3) = 0$$

Transshipment Over Time Problem

Given: $D = (V, A)$, u_a, τ_a for $a \in A$, sources/sinks $S^+, S^- \subset V$ with supplies/demands $v : S^+ \cup S^- \to \mathbb{R}$, time horizon T.

Task: Find flow over time f satisfying supplies/demands within time T.



Definition. Let $o : 2^{S^+ \cup S^-} \to \mathbb{R}$ be defined as follows: for $X \subseteq S^+ \cup S^-$

$$o(X) := \text{ value of max flow over time from } S^+ \cap X \text{ to } S^- \setminus X$$

Lemma. [Klinz 1994] The problem is feasible if and only if

$$o(X) \geq v(X) \qquad \text{for all } X \subseteq S^+ \cup S^-.$$

Transshipments Over Time via SFM

Definition. Let $o : 2^{S^+ \cup S^-} \to \mathbb{R}$ be defined as follows: for $X \subseteq S^+ \cup S^-$

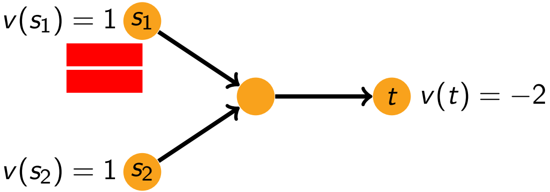
$$o(X) := \text{ value of max flow over time from } S^+ \cap X \text{ to } S^- \setminus X$$

Lemma. [Klinz 1994] The problem is feasible if and only if

$$o(X) \geq v(X) \qquad \text{for all } X \subseteq S^+ \cup S^-.$$

Example. capacities $u \equiv 1$, transit times $\tau \equiv 1$, $T = 4$

$\theta = 0$



$$0 = o(\emptyset) = o(\{t, \dots\}) \qquad 2 = o(\{s_1\})$$

Transshipments Over Time via SFM

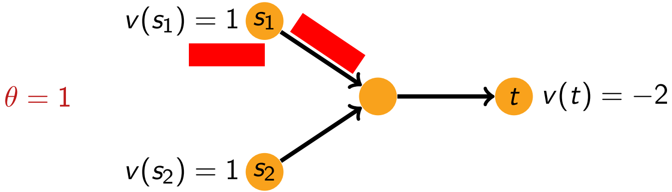
Definition. Let $o : 2^{S^+ \cup S^-} \to \mathbb{R}$ be defined as follows: for $X \subseteq S^+ \cup S^-$

$$o(X) := \text{ value of max flow over time from } S^+ \cap X \text{ to } S^- \setminus X$$

Lemma. [Klinz 1994] The problem is feasible if and only if

$$o(X) \geq v(X) \qquad \text{for all } X \subseteq S^+ \cup S^-.$$

Example. capacities $u \equiv 1$, transit times $\tau \equiv 1$, $T = 4$



$$0 = o(\emptyset) = o(\{t, \dots\}) \qquad 2 = o(\{s_1\})$$

Transshipments Over Time via SFM

Definition. Let $o : 2^{S^+ \cup S^-} \to \mathbb{R}$ be defined as follows: for $X \subseteq S^+ \cup S^-$

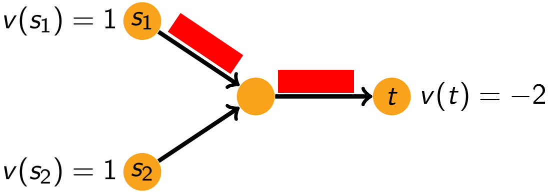
$$o(X) := \text{ value of max flow over time from } S^+ \cap X \text{ to } S^- \setminus X$$

Lemma. [Klinz 1994] The problem is feasible if and only if

$$o(X) \geq v(X) \qquad \text{for all } X \subseteq S^+ \cup S^-.$$

Example. capacities $u \equiv 1$, transit times $\tau \equiv 1$, $T = 4$

$\theta = 2$



$$0 = o(\emptyset) = o(\{t, \dots\}) \qquad 2 = o(\{s_1\})$$

Transshipments Over Time via SFM

Definition. Let $o : 2^{S^+ \cup S^-} \to \mathbb{R}$ be defined as follows: for $X \subseteq S^+ \cup S^-$

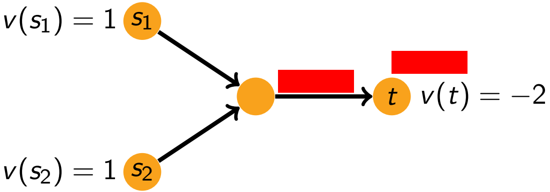
$$o(X) := \text{ value of max flow over time from } S^+ \cap X \text{ to } S^- \setminus X$$

Lemma. [Klinz 1994] The problem is feasible if and only if

$$o(X) \geq v(X) \qquad \text{for all } X \subseteq S^+ \cup S^-.$$

Example. capacities $u \equiv 1$, transit times $\tau \equiv 1$, $T = 4$

$\theta = 3$



$$0 = o(\emptyset) = o(\{t, \dots\}) \qquad 2 = o(\{s_1\})$$

Transshipments Over Time via SFM

Definition. Let $o : 2^{S^+ \cup S^-} \to \mathbb{R}$ be defined as follows: for $X \subseteq S^+ \cup S^-$

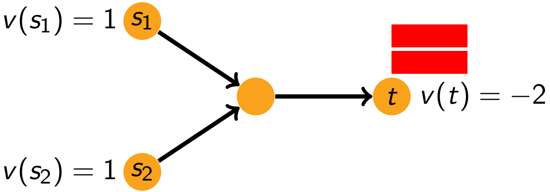
$$o(X) := \text{ value of max flow over time from } S^+ \cap X \text{ to } S^- \setminus X$$

Lemma. [Klinz 1994] The problem is feasible if and only if

$$o(X) \geq v(X) \qquad \text{for all } X \subseteq S^+ \cup S^-.$$

Example. capacities $u \equiv 1$, transit times $\tau \equiv 1$, $T = 4$

$\theta = 4$



$$0 = o(\emptyset) = o(\{t, \dots\}) \qquad 2 = o(\{s_1\}) = o(\{s_2\}) = o(\{s_1, s_2\})$$

Transshipments Over Time via SFM

Definition. Let $o : 2^{S^+ \cup S^-} \to \mathbb{R}$ be defined as follows: for $X \subseteq S^+ \cup S^-$

$$o(X) := \text{ value of max flow over time from } S^+ \cap X \text{ to } S^- \setminus X$$

Lemma. [Klinz 1994] The problem is feasible if and only if

$$o(X) \geq v(X) \qquad \text{for all } X \subseteq S^+ \cup S^-.$$

Observations.

- $X \mapsto o(X)$ is submodular (cut function in time-expanded network)
- also $X \mapsto o(X) - v(X)$ is submodular (as v is modular)

$\Longrightarrow$ Check existence of feasible transshipment over time via one SFM.

Theorem. [Hoppe, Tardos 1995]
Compute transshipment over time via $O(|S^+ \cup S^-|)$ calls to SFM oracle.

Theorem. [Schlöter, Sk. 2017] Only one SFM necessary.

Opening the Blackbox of SFM Algorithms

Let $g : 2^U \to \mathbb{R}$ submodular. For $y \in \mathbb{R}^U$, $X \subseteq U$ let $y(X) = \sum_{r \in X} y_r$.

Definition. (Base Polytope)

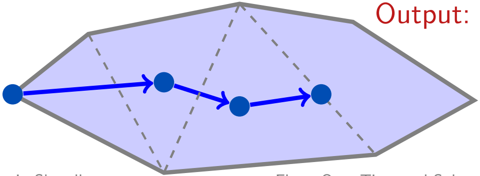
$$\mathcal{B}(g) := \{ y \in \mathbb{R}^U \mid y(X) \leq g(X) \text{ for all } X \subseteq U,\ y(U) = g(U) \}$$

Theorem. [Edmonds 1970]

$$\min\{ g(X) \mid X \subseteq U \} \;=\; \max\{ y^-(U) \mid y \in \mathcal{B}(g) \},$$

where $y^-(U) :=$ sum of all negative coordinates of vector y.

Idea of SFM algorithms:



Output: $y^* = \operatorname{argmax}\{ y^-(U) \mid y \in \mathcal{B}(g) \}$ as convex combination of vertices

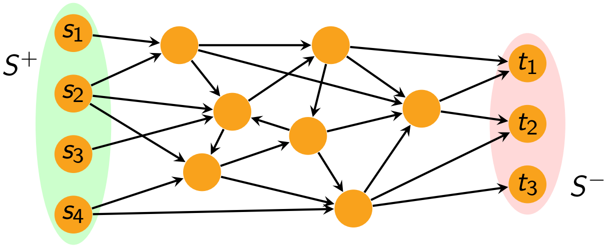
Vertices of the Base Polytope $\mathcal{B}(g)$

Theorem. [Edmonds 1970] Vertices of $\mathcal{B}(g)$ $\longleftrightarrow$ linear orders $\prec$ of U,
i.e., each vertex y is greedy solution $y = y^{\prec}$ for some order $r_1 \prec \cdots \prec r_k$:

$$y^{\prec}_{r_i} := g\big(\{r_1, \ldots, r_i\}\big) - g\big(\{r_1, \ldots, r_{i-1}\}\big), \quad \text{for } i = 1, \ldots, k. \qquad (*)$$

Apply to transshipment function $g = o : 2^{S^+ \cup S^-} \to \mathbb{R}$

Definition. A lex-max flow over time $f^{\prec}$ w.r.t. order $\prec$ on $S^+ \cup S^-$ lexicographically maximizes flow leaving each terminal in given order.



$s_3 \prec s_1 \prec t_2 \prec s_4 \prec t_1 \prec s_2 \prec t_3$

Vertices of the Base Polytope $\mathcal{B}(g)$

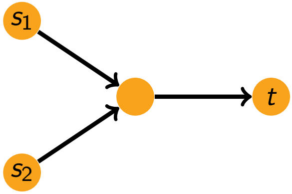
Theorem. [Edmonds 1970] Vertices of $\mathcal{B}(g)$ $\longleftrightarrow$ linear orders $\prec$ of U,
i.e., each vertex y is greedy solution $y = y^{\prec}$ for some order $r_1 \prec \cdots \prec r_k$:

$$y^{\prec}_{r_i} := g\big(\{r_1, \dots, r_i\}\big) - g\big(\{r_1, \dots, r_{i-1}\}\big), \quad \text{for } i = 1, \dots, k. \qquad (*)$$

Apply to transshipment function $g = o : 2^{S^+ \cup S^-} \to \mathbb{R}$

Definition. A lex-max flow over time $f^{\prec}$ w.r.t. order $\prec$ on $S^+ \cup S^-$ lexicographically maximizes flow leaving each terminal in given order.

Example. $u \equiv \tau \equiv 1$, $T = 4$



$t \prec s_1 \prec s_2$: no flow
$t \prec s_2 \prec s_1$: no flow
$s_1 \prec s_2 \prec t$: two units of flow from s_1 to t
$s_1 \prec t \prec s_2$: two units of flow from s_1 to t
$s_2 \prec s_1 \prec t$: two units of flow from s_2 to t
$s_2 \prec t \prec s_1$: two units of flow from s_2 to t

Vertices of the Base Polytope $\mathcal{B}(g)$

Theorem. [Edmonds 1970] Vertices of $\mathcal{B}(g)$ $\longleftrightarrow$ linear orders $\prec$ of U,
i.e., each vertex y is greedy solution $y = y^{\prec}$ for some order $r_1 \prec \cdots \prec r_k$:

$$y^{\prec}_{r_i} \; := \; g\big(\{r_1, \ldots, r_i\}\big) - g\big(\{r_1, \ldots, r_{i-1}\}\big), \quad \text{for } i = 1, \ldots, k. \qquad (*)$$

Apply to transshipment function $g = o : 2^{S^+ \cup S^-} \to \mathbb{R}$

Definition. A lex-max flow over time $f^{\prec}$ w.r.t. order $\prec$ on $S^+ \cup S^-$ lexicographically maximizes flow leaving each terminal in given order.

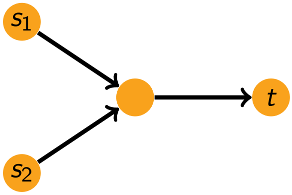
Theorem. [Hoppe, Tardos 1995]
Strongly polynomial algorithm computing lex-max flow over time exists.

Observation.
If $S^+ \cup S^- = \{r_1, \ldots, r_k\}$ and $r_1 \prec \cdots \prec r_k$, flow leaving r_i is given by (*),

i.e., vertex $y^{\prec}$ of polytope $\mathcal{B}(o)$ corresponds to lex-max flow over time $f^{\prec}$!

Example: Base Polytope $\mathcal{B}(o)$ and its Vertices

Example. $u \equiv \tau \equiv 1,\ T = 4$



$t \prec s_1 \prec s_2$: no flow

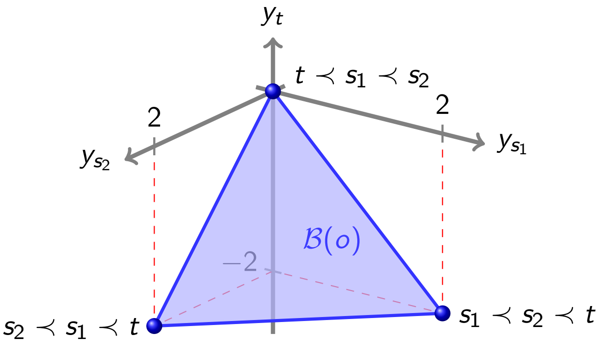
$t \prec s_2 \prec s_1$: no flow

$s_1 \prec s_2 \prec t$: two units of flow from s_1 to t

$s_1 \prec t \prec s_2$: two units of flow from s_1 to t

$s_2 \prec s_1 \prec t$: two units of flow from s_2 to t

$s_2 \prec t \prec s_1$: two units of flow from s_2 to t



Computing a Transshipment Over Time via One SFM

Remember: There is transshipment satisfying demands $v \in \mathbb{R}^{S^+ \cup S^-}$

$$\iff \quad \min\{o(X) - v(X) \mid X \subseteq S^+ \cup S^-\} \geq 0 \qquad \text{[Klinz 1994]}$$

$$\iff \quad \max\{y^-(S^+ \cup S^-) \mid y \in \mathcal{B}(o - v)\} \geq 0 \qquad \text{[Edmonds 1970]}$$

$$\iff \quad \mathbf{0} \in \mathcal{B}(o - v)$$

For $o - v$, SFM algorithm finds representation of $\mathbf{0}$ as convex combination of vertices of $\mathcal{B}(o - v)$.

Observation. $\mathcal{B}(o) = \mathcal{B}(o - v) + v$

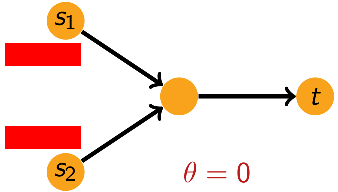
$\Longrightarrow$ representation of v as convex combination of vertices $y^{\prec}$ of $\mathcal{B}(o)$:

$$v = \sum_{\prec} \lambda^{\prec} y^{\prec} \qquad (**)$$

Summary.
Convex combination $\sum_{\prec} \lambda^{\prec} f^{\prec}$ of lex-max flows over time $f^{\prec}$ satisfies given demands v by (**) and thus solves transshipment over time problem.

Example: Transshipment Over Time as Convex Combin.

Example. $u \equiv \tau \equiv 1,\ T = 4$



$t \prec s_1 \prec s_2$: no flow

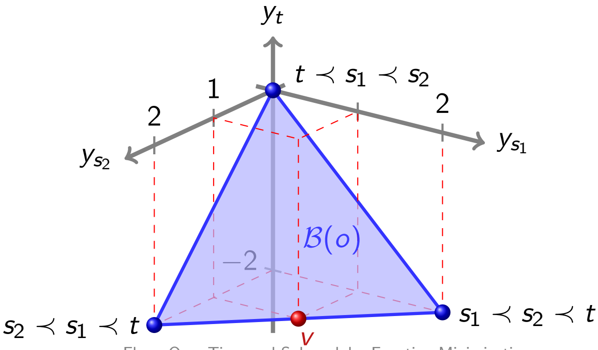
$t \prec s_2 \prec s_1$: no flow

$s_1 \prec s_2 \prec t$: two units of flow from s_1 to t

$s_1 \prec t \prec s_2$: two units of flow from s_1 to t

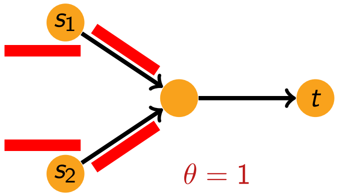
$s_2 \prec s_1 \prec t$: two units of flow from s_2 to t

$s_2 \prec t \prec s_1$: two units of flow from s_2 to t



Example: Transshipment Over Time as Convex Combin.

Example. $u \equiv \tau \equiv 1,\ T = 4$



$t \prec s_1 \prec s_2$: no flow

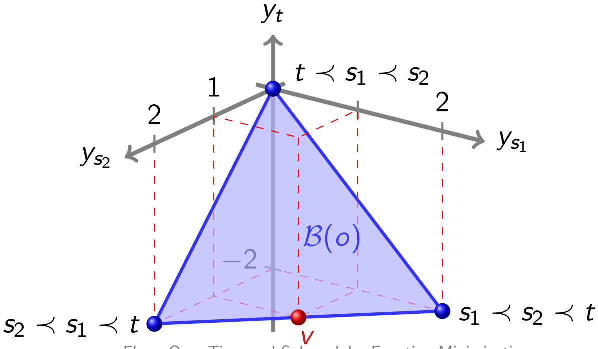
$t \prec s_2 \prec s_1$: no flow

$s_1 \prec s_2 \prec t$: two units of flow from s_1 to t

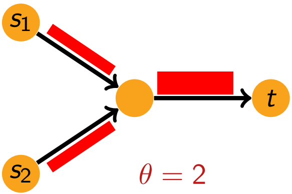
$s_1 \prec t \prec s_2$: two units of flow from s_1 to t

$s_2 \prec s_1 \prec t$: two units of flow from s_2 to t

$s_2 \prec t \prec s_1$: two units of flow from s_2 to t



Example. $u \equiv \tau \equiv 1,\ T = 4$



$t \prec s_1 \prec s_2$: no flow

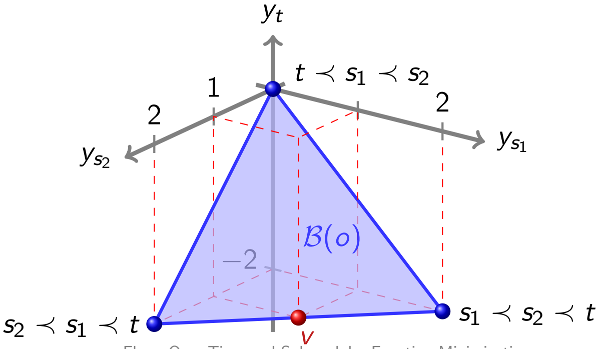
$t \prec s_2 \prec s_1$: no flow

$s_1 \prec s_2 \prec t$: two units of flow from s_1 to t

$s_1 \prec t \prec s_2$: two units of flow from s_1 to t

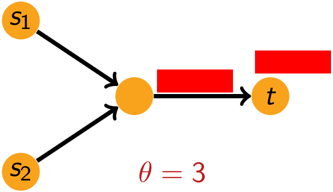
$s_2 \prec s_1 \prec t$: two units of flow from s_2 to t

$s_2 \prec t \prec s_1$: two units of flow from s_2 to t



Example: Transshipment Over Time as Convex Combin.

Example. $u \equiv \tau \equiv 1,\ T = 4$



$t \prec s_1 \prec s_2$: no flow

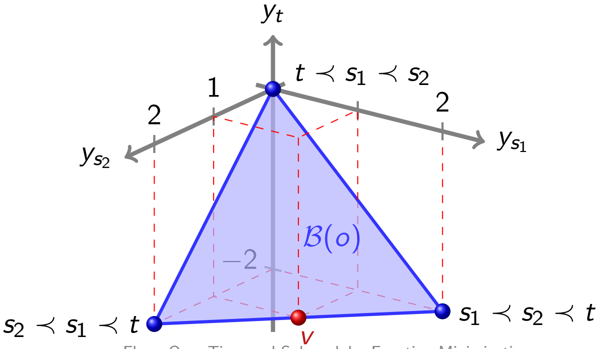
$t \prec s_2 \prec s_1$: no flow

$s_1 \prec s_2 \prec t$: two units of flow from s_1 to t

$s_1 \prec t \prec s_2$: two units of flow from s_1 to t

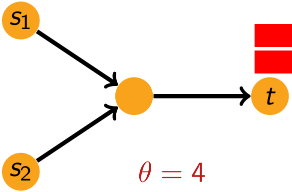
$s_2 \prec s_1 \prec t$: two units of flow from s_2 to t

$s_2 \prec t \prec s_1$: two units of flow from s_2 to t



Example: Transshipment Over Time as Convex Combin.

Example. $u \equiv \tau \equiv 1,\ T = 4$



$t \prec s_1 \prec s_2$: no flow

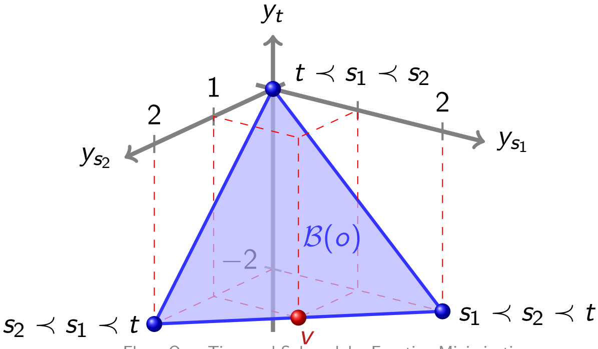
$t \prec s_2 \prec s_1$: no flow

$s_1 \prec s_2 \prec t$: two units of flow from s_1 to t

$s_1 \prec t \prec s_2$: two units of flow from s_1 to t

$s_2 \prec s_1 \prec t$: two units of flow from s_2 to t

$s_2 \prec t \prec s_1$: two units of flow from s_2 to t

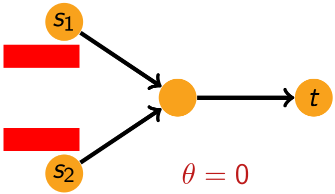


Conclusion

- Flows over time are considerably more complex than static flows.
- Transshipments over time rely on submodular function minimization

Open problems

- Make use of particular network structure behind submodular function
- Compute integral transshipment over time (like Hoppe & Tardos)

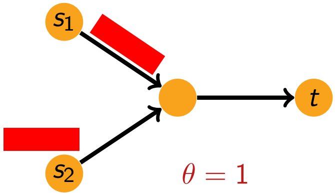


Conclusion

- Flows over time are considerably more complex than static flows.
- Transshipments over time rely on submodular function minimization

Open problems

- Make use of particular network structure behind submodular function
- Compute integral transshipment over time (like Hoppe & Tardos)

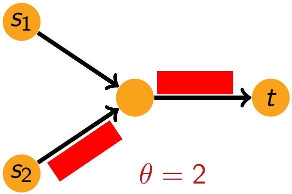


Conclusion

- Flows over time are considerably more complex than static flows.
- Transshipments over time rely on submodular function minimization

Open problems

- Make use of particular network structure behind submodular function
- Compute integral transshipment over time (like Hoppe & Tardos)

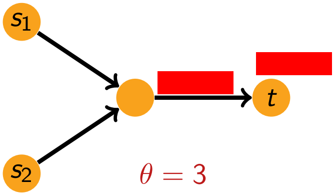


Conclusion

- Flows over time are considerably more complex than static flows.
- Transshipments over time rely on submodular function minimization

Open problems

- Make use of particular network structure behind submodular function
- Compute integral transshipment over time (like Hoppe & Tardos)



Conclusion

- Flows over time are considerably more complex than static flows.
- Transshipments over time rely on submodular function minimization

Open problems

- Make use of particular network structure behind submodular function
- Compute integral transshipment over time (like Hoppe & Tardos)