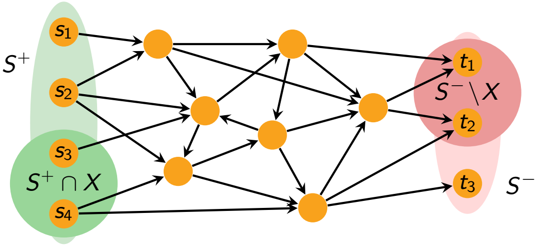
# Transshipments over time, submodular functions, and discrete Newton

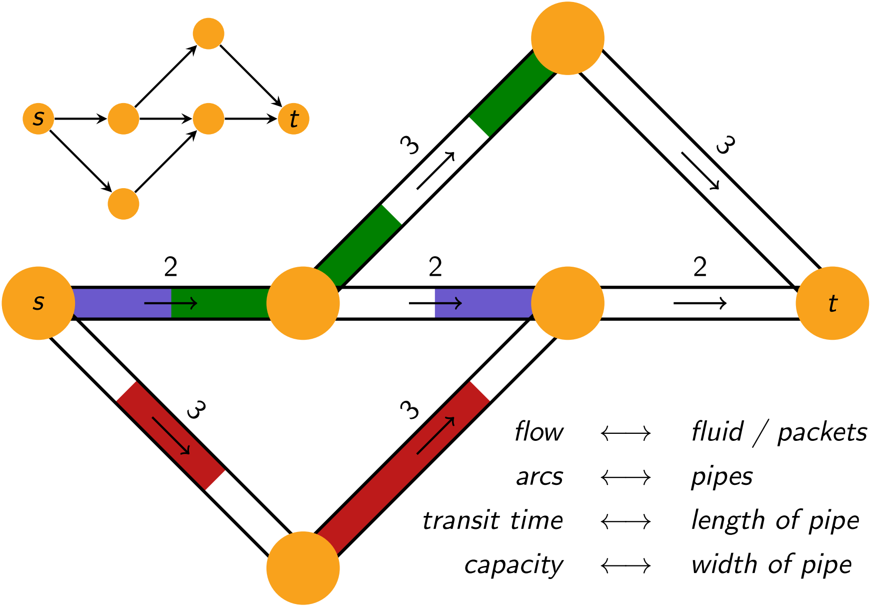
Miriam Schlöter (ETH Zürich)
Martin Skutella
Khai Van Tran
(TU Berlin)

ICERM Workshop on Combinatorics and Optimization



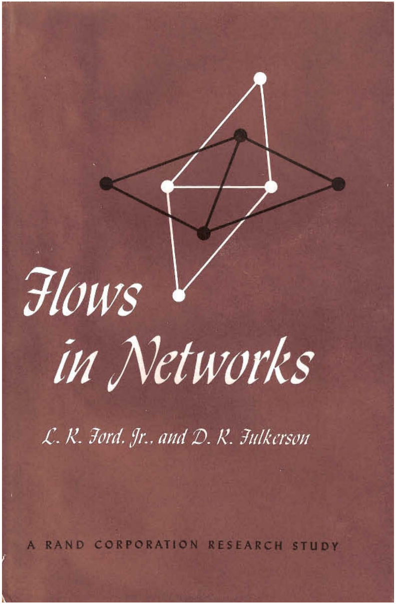

Berlin Mathematics Research Center MATH+

# Flows Over Time: Example




| | | |
|---|---|---|
| *flow* | ⟷ | *fluid / packets* |
| *arcs* | ⟷ | *pipes* |
| *transit time* | ⟷ | *length of pipe* |
| *capacity* | ⟷ | *width of pipe* |

# Flows Over Time: History



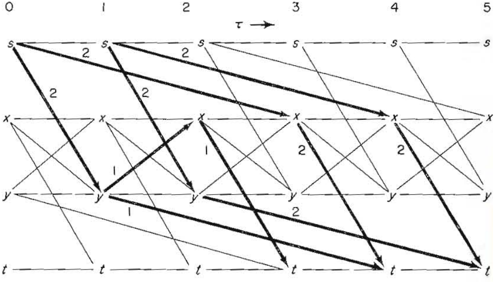



Figure 9.7

# 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] | NP-hard [3] | NP-hard [4] |

References.


[1] Ford, Fulkerson (1958)

[2] Hoppe, Tardos (1995)

[3] Klinz, Woeginger (1995)

[4] Hall, Hippler, Sk. (2007)

# Computing Quickest Transshipments Efficiently

| | |
|---|---|
| *Quickest s-t-Flow Problem* | |
| [Saho, Shigeno 2017] | $\tilde{O}(m^2 n)$ |
| *Evacuation Problem (single source or single sink)* | |
| [Schlöter 2018; Kamiyama 2019] | $\tilde{O}(m^2 k^5 + m^2 nk)$ |
| *Quickest Transshipment Problem (multiple sources and sinks)* | |
| [Hoppe, Tardos 2000] | $\tilde{O}(m^4 k^{15})$ |
| [Schlöter, Sk. 2017] | $\tilde{O}(m^4 k^{14})$[1] |
| [Schlöter, Sk., Tran 2022] | $\tilde{O}(m^2 k^5 + m^3 k^3 + m^3 n)$[1] |
| [Schlöter, Sk., Tran 2023+] | $\tilde{O}(m^2 k^6 + m^3 k^4 + m^3 n)$ |

$m := \#$ arcs $\quad n := \#$ nodes $\quad k := \#$ terminals

[1] fractional solutions only

# Maximum $s$-$t$-Flows Over Time

Algorithm. [Ford, Fulkerson 1958]

Input: $D = (V, A)$, $s, t \in V$, capacities $u_a$, transit times $\tau_a$, time $\theta \geq 0$

Output: maximum $s$-$t$-flow over time with time horizon $\theta$

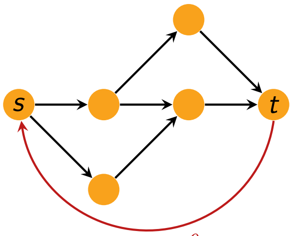
1. compute *static s-t-flow* $x$ in $D$

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

2. determine path-decomposition

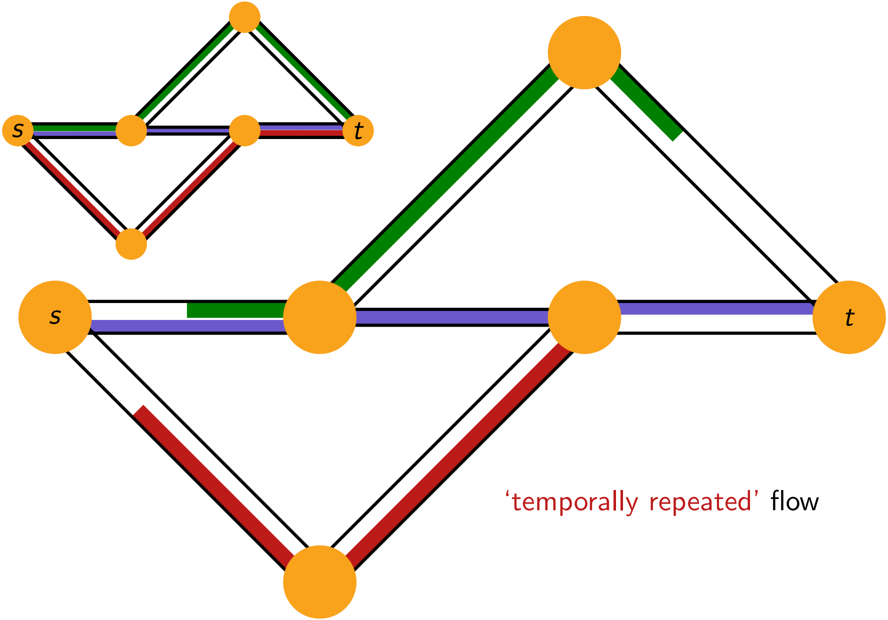
   $$x_a \;=\; \sum_{P \in \mathcal{P}:\, a \in P} x_P \qquad \text{for all } a \in A$$

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




$\tau_{ts} = -\theta$
$u_{ts} = \infty$

# Maximum $s$-$t$-Flow Over Time: Example




'temporally repeated' flow

# Transshipment Over Time Problem

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

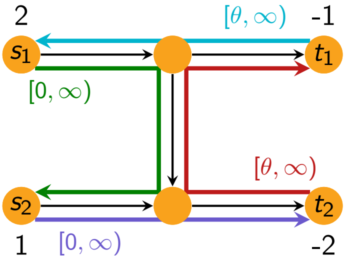
Task: find flow over time satisfying supplies/demands in time $\theta$

Example:

$u \equiv 1$

$\tau \equiv 1$

$\theta = 4$



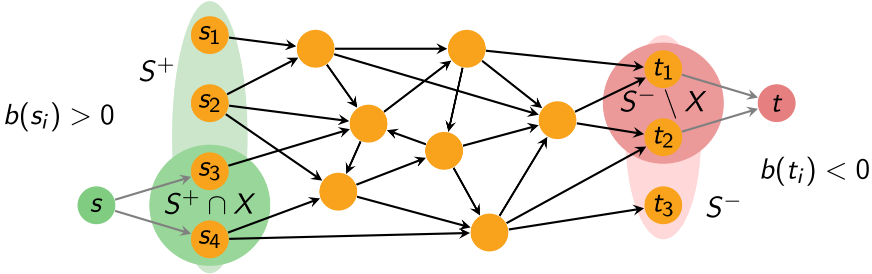

Observations:

- not clear how to use super-source / super-sink
- no temporally repeated solution

# Transshipment Over Time Problem

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

Task: find flow over time satisfying supplies/demands in time $\theta$



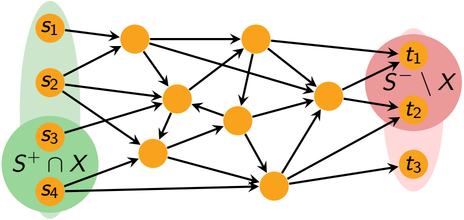

Definition. Let $o : 2^{S^+ \cup S^-} \to \mathbb{R}$ be defined as follows: for $X \subseteq S^+ \cup S^-$

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

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

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

# Sufficiency of Criterion: $b(X) \leq o(X)$ for all $X \subseteq S^+ \cup S^-$




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

**Klinz criterion:** $b \in \mathcal{B}(o)$

Base polytope of submodular function $o$:

$$\mathcal{B}(o) := \{y \in \mathbb{R}^{S^+ \cup S^-} \mid y(X) \leq o(X)\ \forall X \subseteq S^+ \cup S^-,\ y(S^+ \cup S^-) = 0\}$$

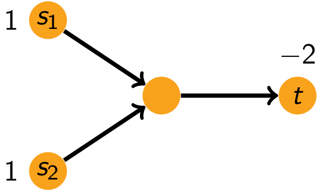
[Edmonds '70]: vertices of $\mathcal{B}(o)$ $\longleftrightarrow$ linear orders $\prec$ of $S^+ \cup S^-$

That is, each vertex is greedy solution $y^\prec$ for some order $r_1 \prec \cdots \prec r_k$:

$$y^\prec_{r_i} := o(\{r_1, \ldots, r_i\}) - o(\{r_1, \ldots, r_{i-1}\}), \quad \text{for } i = 1, \ldots, k.$$

Note: $y^\prec$ is supply/demand vector satisfied by lex-max flow over time $f^\prec$;
if $b \in \mathcal{B}(o)$, then a convex combination of lex-max flows satisfies $b$. □

# Convex Combination of Lex-Max Flows Over Time

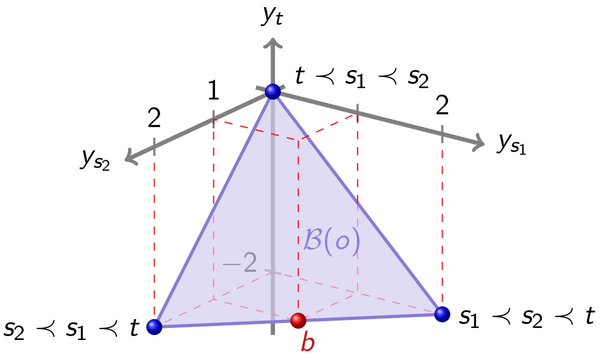



$u \equiv 1 \quad \tau \equiv 1 \quad \theta = 4$

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

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

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

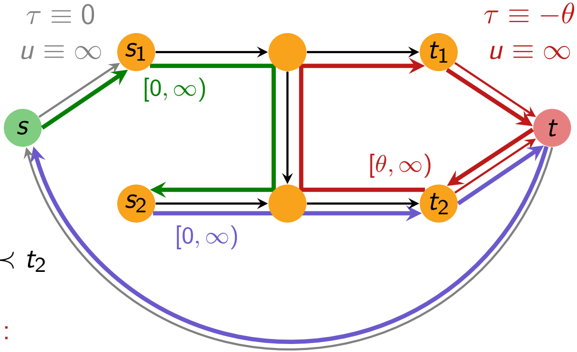


# Computing Lex-Max Flows Over Time [Hoppe, Tardos 2000]

$u \equiv 1$

$\tau \equiv 1$

$\theta = 4$




given order:
$s_1 \prec t_1 \prec s_2 \prec t_2$

Outline of algorithm:

- start with zero flow
- always maintain static min-cost flow with path decomposition
- consider terminals in reverse order
  - for sink $t_i$, add arc $t_i t$, find min-cost circulation in residual graph
  - for source $s_i$, delete arc $ss_i$, find min-cost maximum $s$-$s_i$-flow

Observation. If $u \in \mathbb{Z}^A$, the computed lex-max flow over time is integral.

# Finding Convex Combination of Lex-Max Flows Over Time

Consider submodular function given by

$$g(X) := o(X) - b(X) \qquad \text{for } X \subseteq S^+ \cup S^-.$$

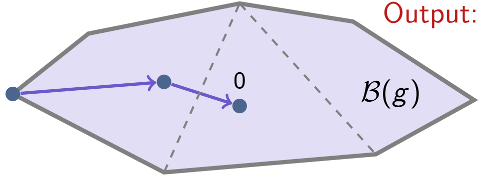
Then,

$$\mathcal{B}(g) = \mathcal{B}(o) - b,$$

and thus

$$b \in \mathcal{B}(o) \quad \Longleftrightarrow \quad 0 \in \mathcal{B}(g) \quad \Longleftrightarrow \quad \min_{X \subseteq S^+ \cup S^-} g(X) = 0.$$

Idea of SFM algorithms:



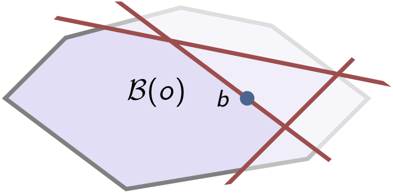

Output: $0 = \operatorname{argmax}\{y^-(U) \mid y \in \mathcal{B}(g)\}$ as convex combination of vertices

$\Longrightarrow$ $b$ as convex combination of vertices of $\mathcal{B}(o)$

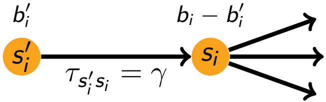
# Finding Integral Flow Over Time similar to [Hoppe, Tardos 2000]

$$\mathcal{B}(o) := \{y \in \mathbb{R}^{S^+ \cup S^-} \mid y(X) \le o(X)\ \forall X \subseteq S^+ \cup S^-,\ y(S^+ \cup S^-) = 0\}$$

Idea: Carefully tighten constraints on $\mathcal{B}(o)$ until $b$ is a vertex of $\mathcal{B}(o)$.




Combinatorial implementation: Split one terminal at a time and add delay



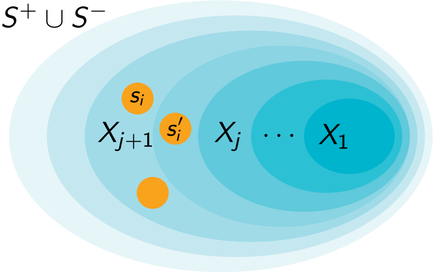

How to choose terminal?

Maintain chain of tight subsets

$$\emptyset \subset X_1 \subset \cdots \subset S^+ \cup S^-$$

with $o(X_j) = b(X_j)$ for all $j$. Set

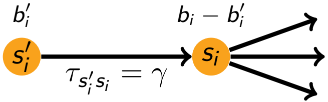
$$b_i'(\gamma) := o^\gamma(X_j \cup \{s_i'\}) - o(X_j).$$



$$\mathcal{B}(o) := \{y \in \mathbb{R}^{S^+ \cup S^-} \mid y(X) \le o(X)\ \forall X \subseteq S^+ \cup S^-,\ y(S^+ \cup S^-) = 0\}$$

Idea: Carefully tighten constraints on $\mathcal{B}(o)$ until $b$ is a vertex of $\mathcal{B}(o)$.

Combinatorial implementation: Split one terminal at a time and add delay




How to choose terminal?

Maintain chain of tight subsets

$$\emptyset \subset X_1 \subset \cdots \subset S^+ \cup S^-$$

with $o(X_j) = b(X_j)$ for all $j$. Set

$$b_i'(\gamma) := o^\gamma(X_j \cup \{s_i'\}) - o(X_j).$$

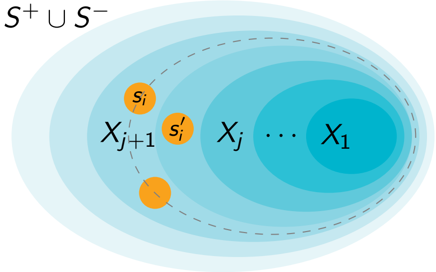
How to choose $\gamma$?

- $\gamma = 0$ yields infeasible problem
- $\gamma = \theta$ is feasible

choose minimum feasible $\gamma$

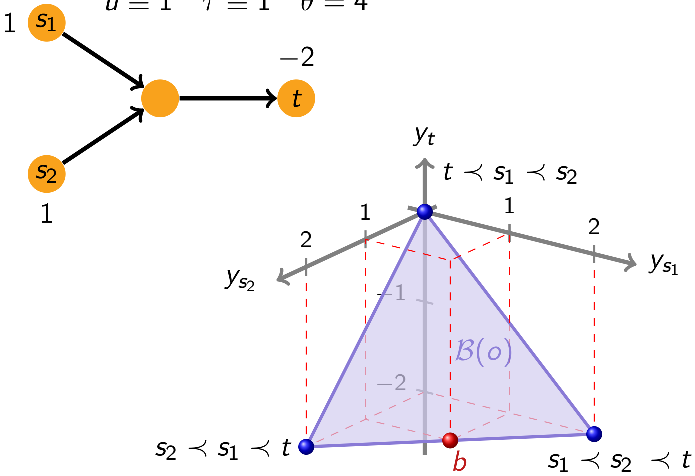
(parametric SFM problem!)

This yields another tight subset $Q$ with $X_j \cup \{s_i'\} \subset Q \subset X_{j+1}$.

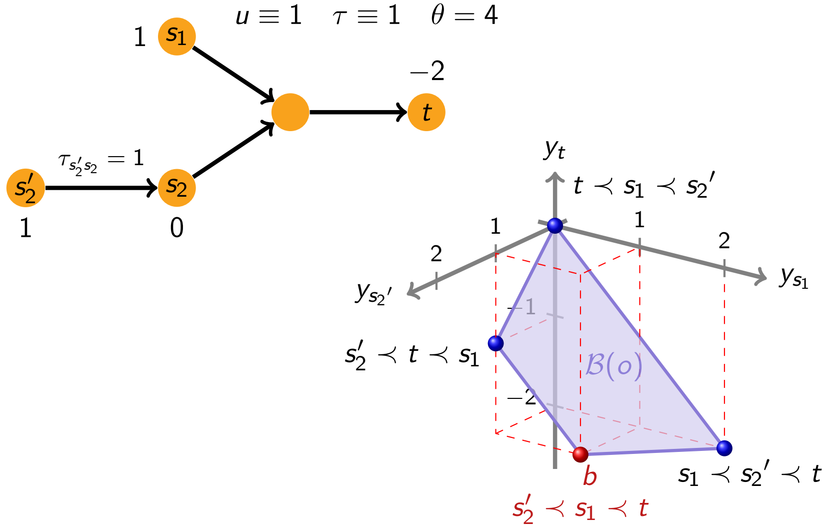


# Computing Integral Transshipment Over Time: Example

$u \equiv 1 \quad \tau \equiv 1 \quad \theta = 4$



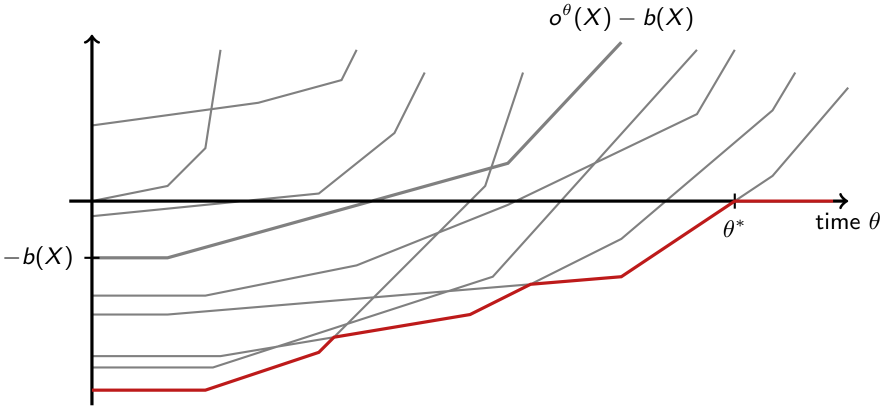
# Computing Integral Transshipment Over Time: Example



# Quickest Flows, Parametric SFM, and Discrete Newton

How to find the minimum feasible time horizon $\theta^*$?

$$\theta^* = \min\{\theta \mid o^\theta(X) - b(X) \geq 0 \text{ for all } X \subseteq S^+ \cup S^-\}$$




Related recent work: [McCormick, Oriolo, Peis 2014] [Goemans, Gupta, Jaillet 2017] [Schlöter 2018] [Kamiyama 2019] [Dadush, Koh, Natura, Végh 2021]

# Conclusion

| | |
|---|---|
| *Quickest s-t-Flow Problem* | |
| [Saho, Shigeno 2017] | $\tilde{O}(m^2 n)$ |
| *Evacuation Problem (single source or single sink)* | |
| [Schlöter 2018; Kamiyama 2019] | $\tilde{O}(m^2 k^5 + m^2 nk)$ |
| *Quickest Transshipment Problem (multiple sources and sinks)* | |
| [Hoppe, Tardos 2000] | $\tilde{O}(m^4 k^{15})$ |
| [Schlöter, Sk. 2017] | $\tilde{O}(m^4 k^{14})^*$ |
| [Schlöter, Sk., Tran 2022] | $\tilde{O}(m^2 k^5 + m^3 k^3 + m^3 n)^*$ |
| [Schlöter, Sk., Tran 2023+] | $\tilde{O}(m^2 k^6 + m^3 k^4 + m^3 n)$ |

Main Open Problem:

How to minimize specific submodular functions more efficiently?