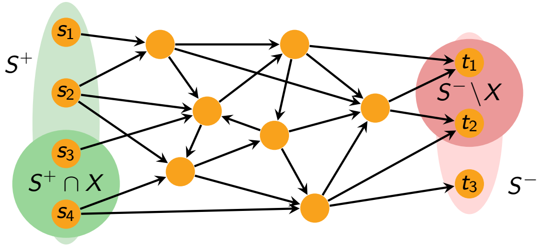
A Faster Algorithm for Quickest Transshipments via an Extended Discrete Newton Method

Miriam Schlöter   Martin Skutella   Khai Van Tran

Dutch Seminar on Optimization

September 30, 2021



Maximum Flow Over Time

Algorithm. [Ford, Fulkerson 1958]

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

Output: maximum s-t-flow over time with time horizon θ

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

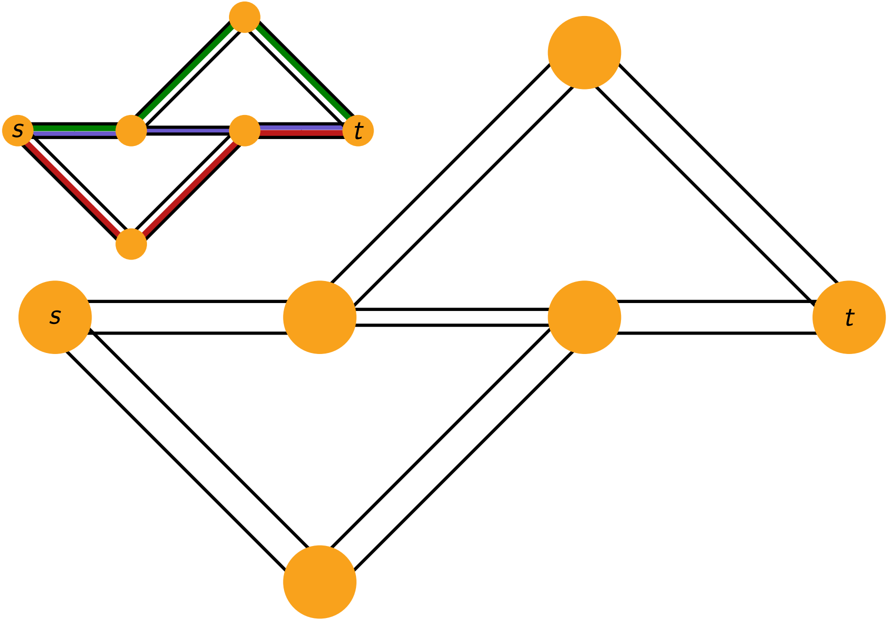
$$\text{maximizing} \quad \theta\,|x| \;-\; \sum_{a \in A} \tau_a\, x_a$$

2. determine path-decomposition

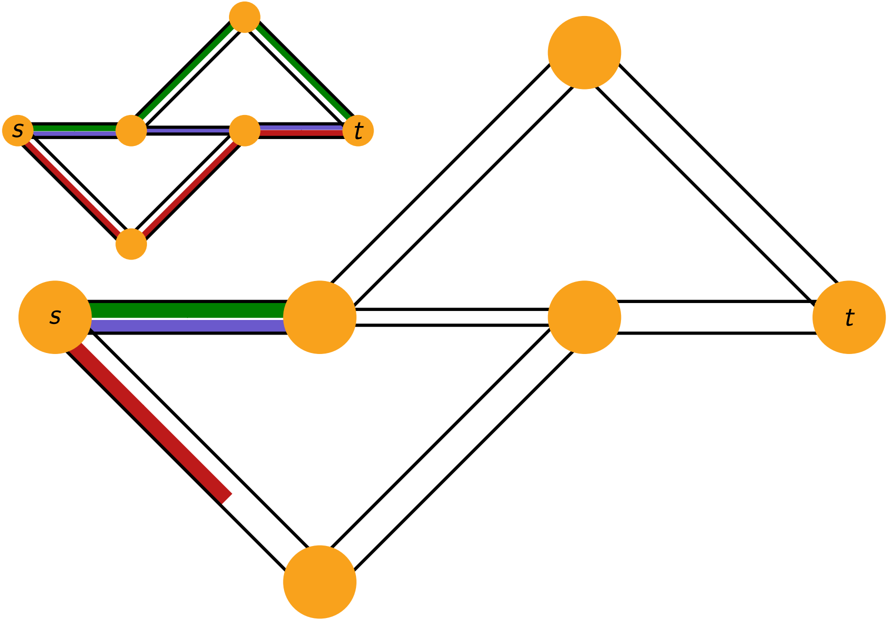
$$x_a \;=\; \sum_{P \in \mathcal{P}\colon a \in P} x_P \qquad \text{for all } a \in A$$

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 θ

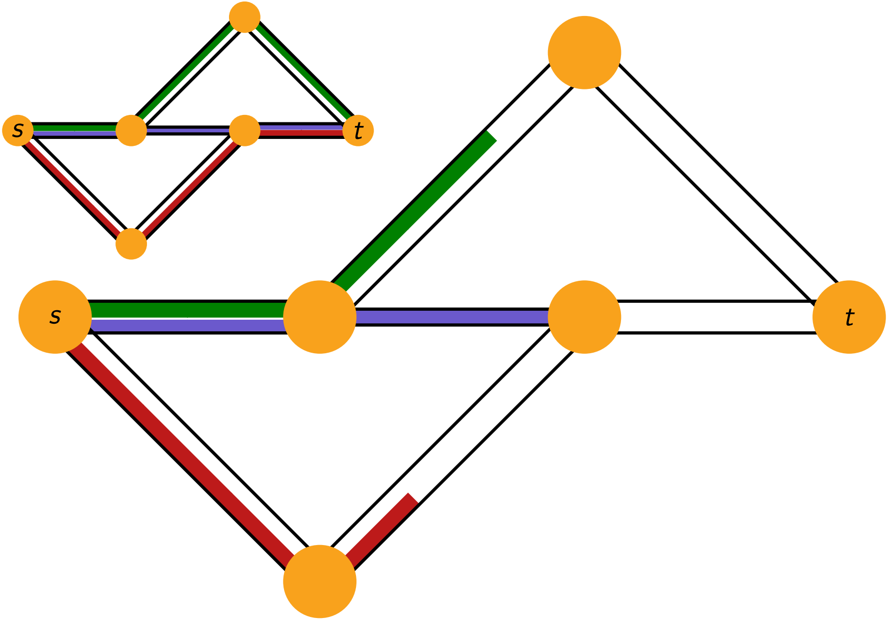
Maximum Flow Over Time: Example



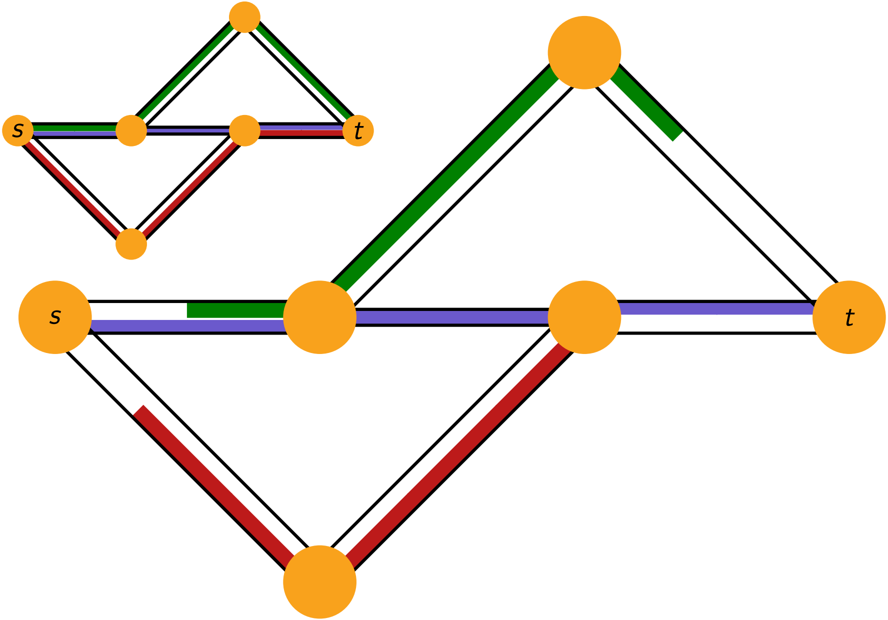
Maximum Flow Over Time: Example



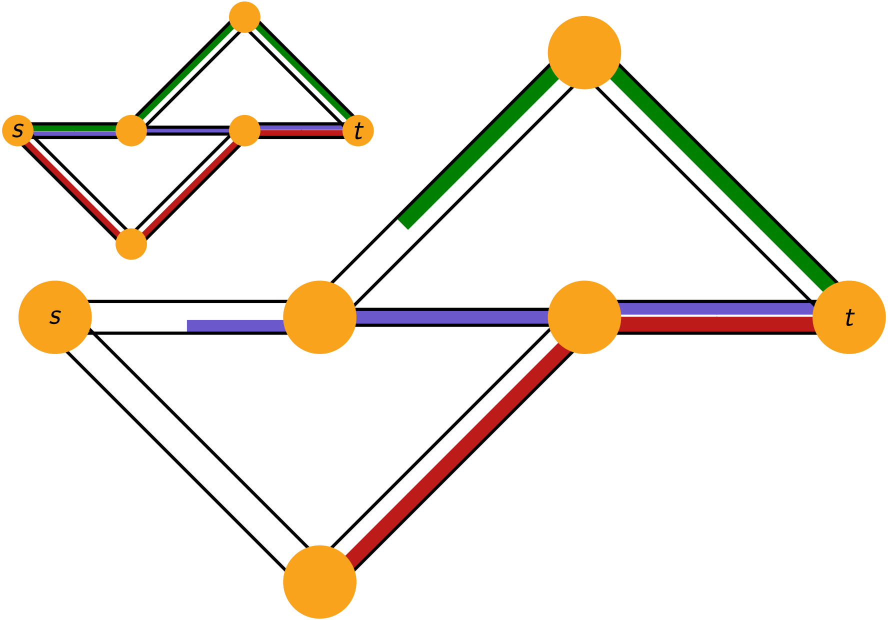
Maximum Flow Over Time: Example



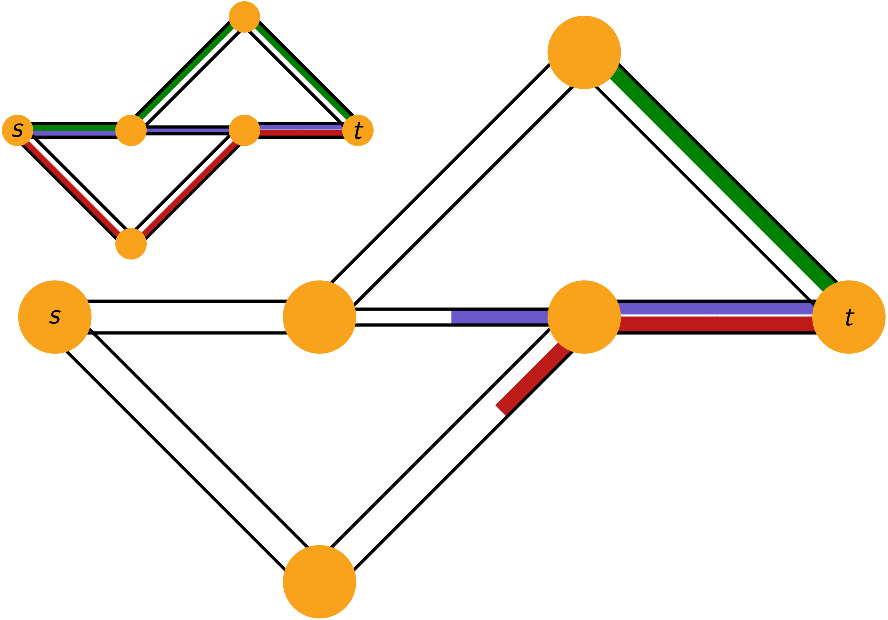
Maximum Flow Over Time: Example



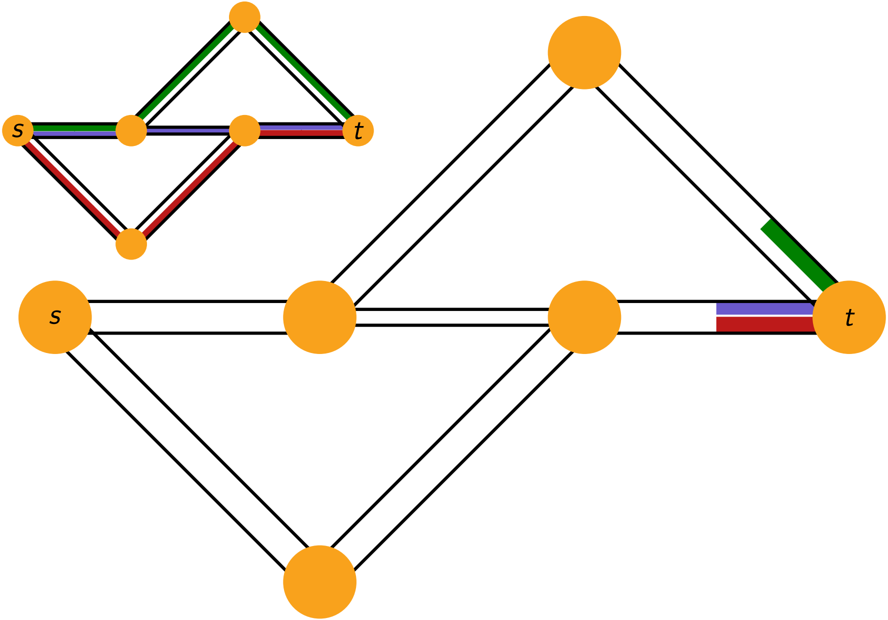
Maximum Flow Over Time: Example



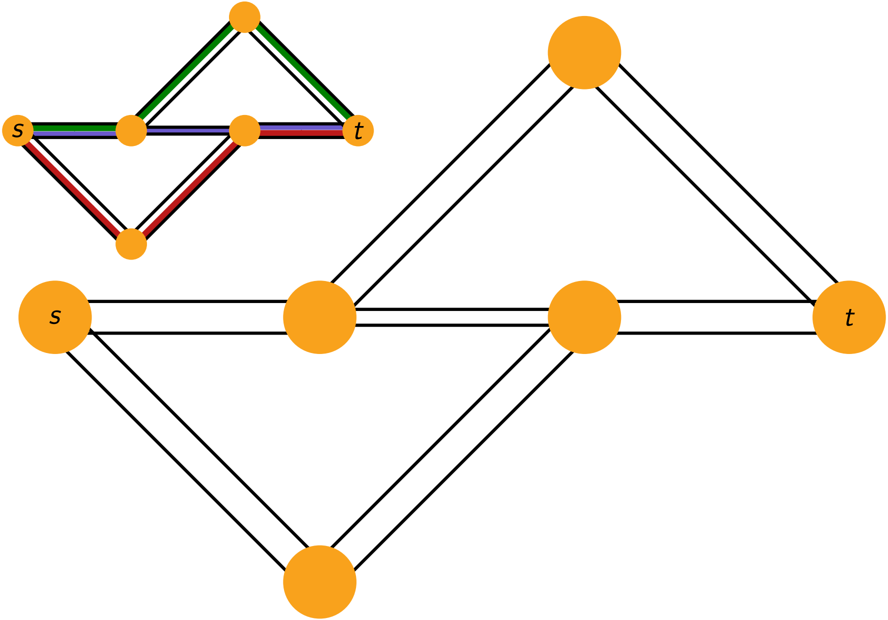
Maximum Flow Over Time: Example



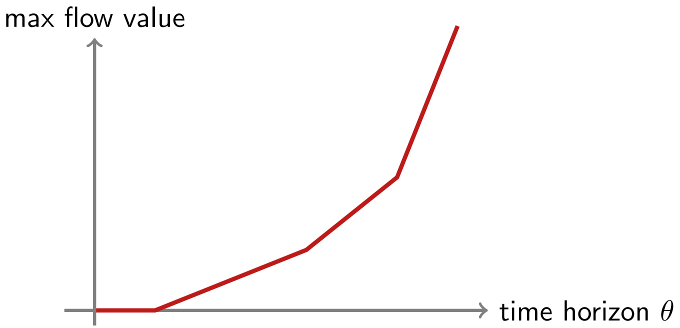
Maximum Flow Over Time: Example



Maximum Flow Over Time: Example



Parametric Maximum Flow Over Time

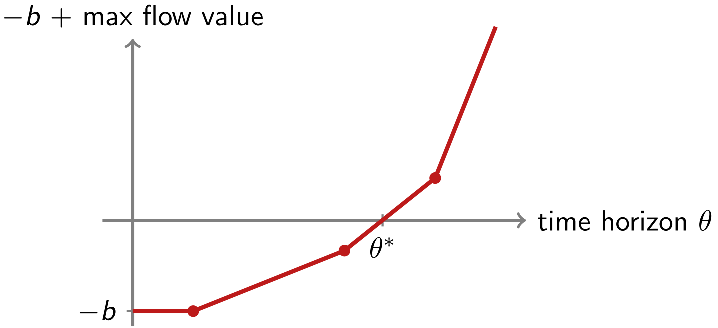


- function is piecewise linear, convex, and increasing
- every breakpoint corresponds to s-t-path P in bidirected graph
- slope of linear piece equals capacity of s-t-cut in subnetwork
- obtain value / slope at θ via static min-cost flow
- but: exponentially many breakpoints [Zadeh 1973; Disser, Sk. 2019]

Quickest Flows

Given: $D = (V, A)$, $s, t \in V$, capacities u_a, transit times τ_a, flow value b

Task: Find s-t-flow over time of value d with minimum time horizon θ^*



Solution methods:

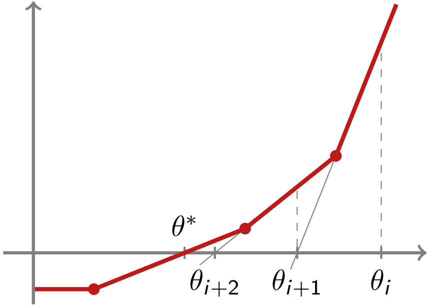
- binary search: (weakly) polynomial
- parametric search [Megiddo 1979]: strongly poly. [Burkard et al. 1993]
- cost scaling: $O(m^2 n \log^2 n)$ [Saho, Shigeno 2017; Lin, Jaillet 2015]

Discrete Newton Method

[Dinkelbach 1967]

[Radzik, *Fractional Combinatorial Optimization*, 1998]

Algorithm



Analysis

Observation.
In each iteration, function value or slope decreases by factor $\leq 1/2$

Lemma. [Goemans 1992]
Let $u \in \mathbb{R}^m$, $y_1, \ldots, y_q \in \{0,1\}^m$ with

$$0 < y_{i+1}u \leq \tfrac{1}{2} y_i u \quad \text{for all } i,$$

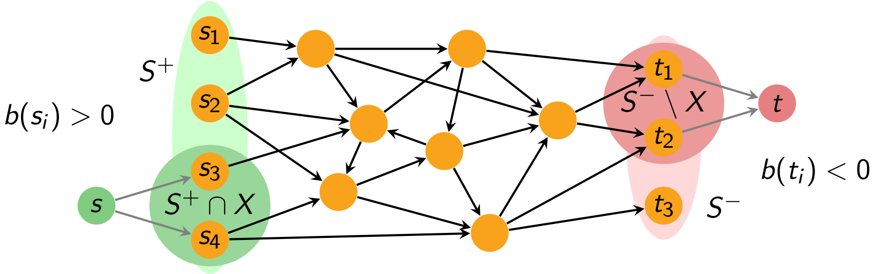
then $q \in O(m \log m)$.

With some more tricks, this yields strongly polynomial quickest flow algo.

Quickest Transshipments

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

Task: find flow over time satisfying supplies/demands in minimum time θ^*



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

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

Lemma. [Klinz 1994] $\theta \geq \theta^* \iff o^\theta(X) \geq b(X) \quad \forall X \subseteq S^+ \cup S^-$

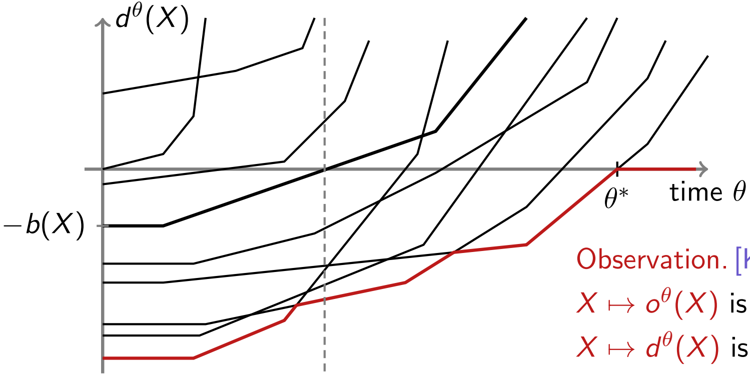
Quickest Transshipments

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

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

Lemma. [Klinz 1994] $\theta \geq \theta^* \iff o^\theta(X) \geq b(X) \quad \forall X \subseteq S^+ \cup S^-$

Corollary. $\theta \geq \theta^* \iff \min_{X \subseteq S^+ \cup S^-} \{\underbrace{o^\theta(X) - b(X)}_{d^\theta(X) :=}\} \geq 0$

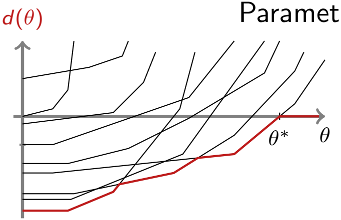


Observation. [Klinz 1994]

$X \mapsto o^\theta(X)$ is submodular

$X \mapsto d^\theta(X)$ is submodular

Quickest Transshipments: State of the Art



Parametric Submodular Function Minimization (SFM):

$$\theta^* = \min\Big\{\theta \geq 0 \mid \underbrace{\min_{X \subseteq S^+ \cup S^-} d^\theta(X)}_{=:\, d(\theta)} \geq 0\Big\}$$

$k := |S^+ \cup S^-|$ (number of terminals)

[Hoppe, Tardos 2000]:

- determine θ^* using Megiddo's parametric search
- $2k - 2$ parametric SFMs to find quickest transshipment

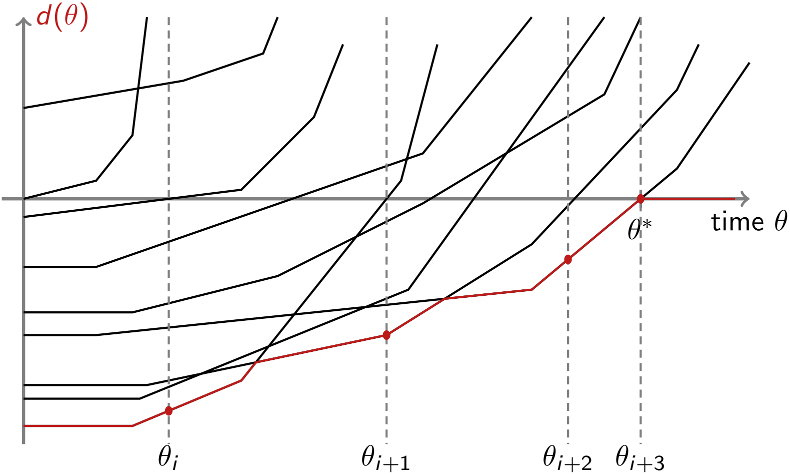
[Schlöter, Sk. 2017]: quickest transshipment with only one parametric SFM

Running time for parametric SFM: $\tilde{O}(m^4 k^{14})$

- need *fully combinatorial* SFM algorithm: $\tilde{O}(m^2 k^7)$ [Iwata, Orlin 2009]

[Schlöter, Sk., Tran 2021]: $\tilde{O}(m^2 k^5 + m^3 k^3 + m^3 n)$ via discrete Newton

First Attempt: Simple Algorithm



[Schlöter 2018; Kamiyama 2019]:

- if $|S^+| = 1$ or $|S^-| = 1$, then minimizers (subsets S_i) are nested
- thus, in this special case, at most k iterations

Extended Discrete Newton with Large Jumps

Algorithm 2: Extended Discrete Newton with Large Jumps

$i := 0, \ \ \theta_0 := 0, \ \ J := \left\{2^0, 2^1, 2^2, \ldots, 2^{\lceil \log_2(k^2/4) \rceil}\right\}$

while $d(\theta_i) < 0$ **do**

$S_i := \operatorname{argmin}\{d^{\theta_i}(S) : S \subseteq S^+ \cup S^-\}$

$\theta'_i := \min\{\theta : d^{\theta}(S_i) = 0\}$

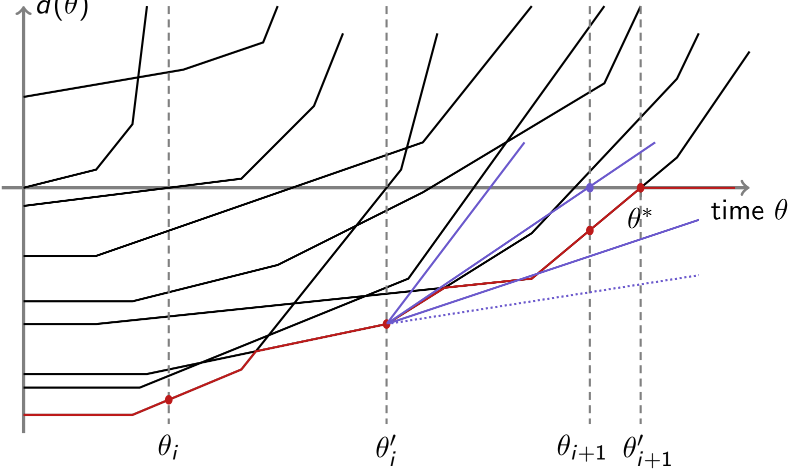
$\theta_{i+1} := \theta'_i \qquad \max\left(\{\theta'_i\} \cup \left\{\theta = \theta'_i + j \cdot \frac{|d(\theta'_i)|}{\mathrm{cut}^{\theta'_i}(S_i)} : d(\theta) < 0, \ j \in J\right\}\right)$

$i := i + 1$

end

return θ_i

Extended Discrete Newton with Larger Jumps



[Dadush, Koh, Natura, Végh 2021]:

- somewhat similar 'look-ahead' approach for classical discrete Newton

Bounding the Number of Iterations

Algorithm 2: Extended Discrete Newton with Large Jumps

$i := 0, \;\; \theta_0 := 0, \;\; J := \left\{2^0, 2^1, 2^2, \ldots, 2^{\lceil \log_2(k^2/4) \rceil}\right\}$

while $d(\theta_i) < 0$ **do**

$\quad S_i := \text{argmin}\{d^{\theta_i}(S) : S \subseteq S^+ \cup S^-\}$

$\quad \theta_i' := \min\{\theta : d^{\theta}(S_i) = 0\}$

$\quad \theta_{i+1} := \max\left(\{\theta_i'\} \cup \left\{\theta = \theta_i' + j \cdot \frac{|d(\theta_i')|}{\text{cut}^{\theta_i'}(S_i)} : d(\theta) < 0, \, j \in J\right\}\right)$

$\quad i := i + 1$

end

return θ_i

Partition iterations into three groups:

1. iterations with longest possible jump, i.e., $j = 2^{\lceil \log_2(k^2/4) \rceil} \geq k^2/4$
2. iterations with shorter jump that move over some breakpoint
3. iterations with shorter jump that do not move over breakpoint

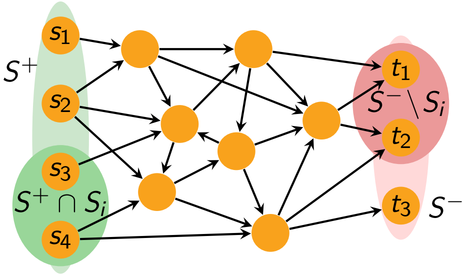
Bounding the Number of Iterations with Longest Jump

Lemma.
The number of iterations with longest possible jump is at most $k^2/4$.

Proof sketch.
Consider one such iteration:

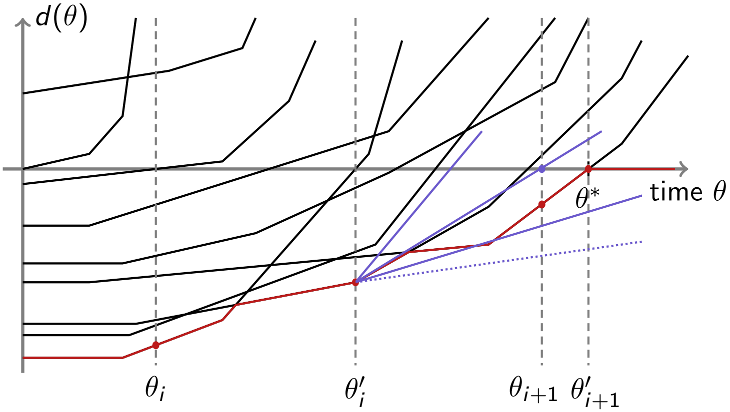
$$\theta_{i+1} \geq \theta'_i + \frac{k^2}{4} \cdot \frac{|d(\theta'_i)|}{\mathrm{cut}^{\theta'_i}(S_i)}$$

Sources $S^+ \cap S_i$ can supply at least $k^2/4$ times the necessary amount to sinks $S^- \setminus S_i$ between θ'_i and θ_{i+1}.



- As there are $\leq k^2/4$ source-sink pairs, one pair alone can do the job.
- This source-sink pair can no longer occur in later iterations! □

Bounding the Number of Iterations with Short Jumps



Lemma. For iterations with short jumps: $\theta^* - \theta_{i+1} \leq \frac{1}{2}(\theta^* - \theta_i)$. □

- Thus, by Goemans' Lemma, number of iterations with breakpoint in $[\theta_i, \theta_{i+1}]$ is at most $O(m \log m)$.
- Number of iterations without breakpoint: $O(k^2 \log k + m \log m \log k)$ Analysis uses 'ring families', similar to [Goemans, Gupta, Jaillet 2017].

Overall Running Time

Lemma. The number of iterations is at most $O(k^2 \log k + m \log m \log k)$.

Proof.

- $O(k^2)$ iterations with longest possible jump
- $O(m \log m)$ iterations with short jump and breakpoint
- $O(k^2 \log k + m \log m \log k)$ short jumps without breakpoint □

Theorem. The overall running time is in $\tilde{O}(m^2 k^5 + m^3 k^3 + m^3 n)$.

Proof. In each iteration we need to solve

- $O(\log k)$ submodular function minimizations, with running time $\tilde{O}(m^2 k^3)$ each [Lee, Sidford, Wong 2015];
- one quickest s-t-flow problem, with running time $\tilde{O}(m^2 n)$ [Saho, Shigeno 2017]. □

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 2021]	$\tilde{O}(m^2 k^5 + m^3 k^3 + m^3 n)$

`arxiv.org/abs/2108.06239`