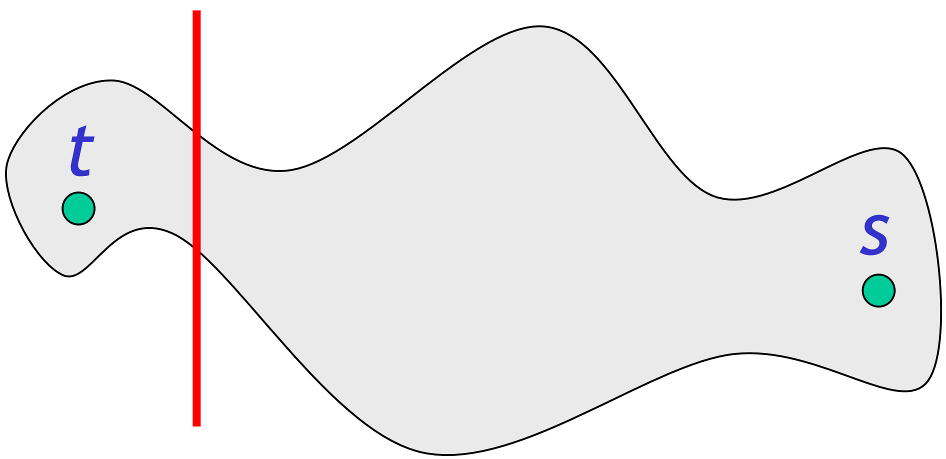
Unbalanced graph cuts

Zoya Svitkina
Cornell University

Joint work with Ara Hayrapetyan, David Kempe and Martin Pál

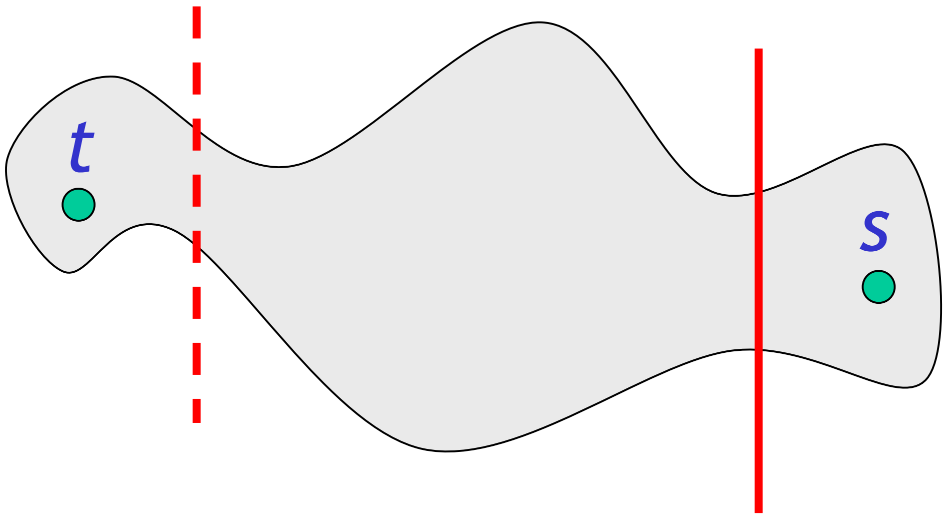
Graph cuts



Min-cut: smallest capacity

Graph cuts

Small capacity and small size of *s*-side



Min-cut: smallest capacity

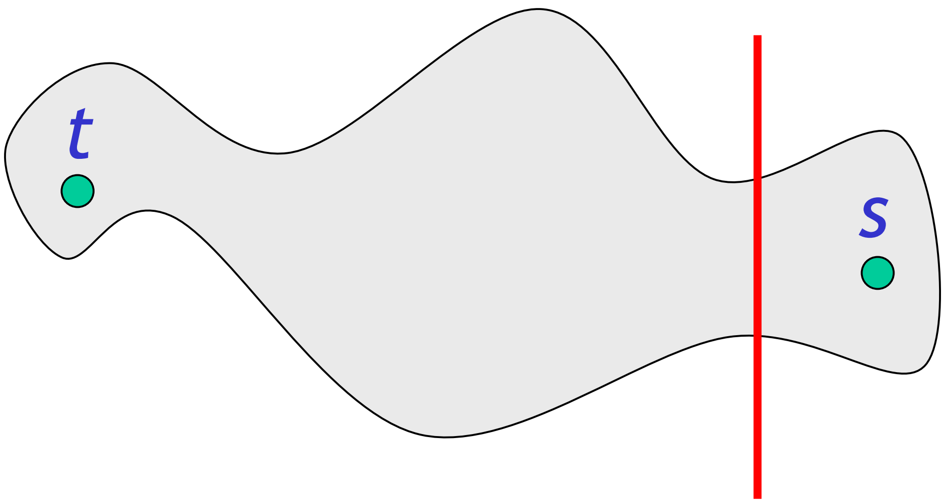
Minimum size bounded capacity cut

Given:

Graph G, nodes s,t, bound B

Find:

- s-t cut (S, T)
  - minimize $|S|$
  - capacity $\leq B$



Applications

- Direct
  - Disaster/military/crime containment
  - Epidemiology (vaccination, quarantine)
  - Computer virus containment
- Indirect (as subroutine)
  - Communities in graphs
  - Others

Related work

- Balanced/sparsest cuts
  - T. Leighton, S. Rao (1988)
  - N. Garg, V. Vazirani, M. Yannakakis (1996)
  - S. Arora, S. Rao, U. Vazirani (2004)
- Cuts of specified size
  - U. Feige, R. Krauthgamer (2000)
- Minimum size cuts
  - S. Eubank, V.S.A. Kumar, M.V. Marathe, A. Srinivasan, N.Wang (2005)

NP-hardness

- Weighted problem on trees
  - reduction from Knapsack
- Unweighted problem on general graphs
  - reduction from Clique

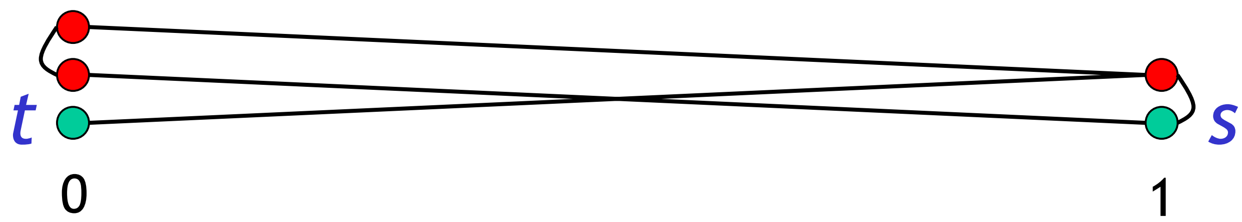
Two algorithms

- Bicriteria approximation:
  - find a cut of capacity at most $2 \cdot B$,
  - with at most $2 \cdot OPT$ nodes on s-side

- Techniques:
  1) randomized rounding of a linear program
  2) parametric minimum cuts,
     analysis based on Lagrangian relaxation

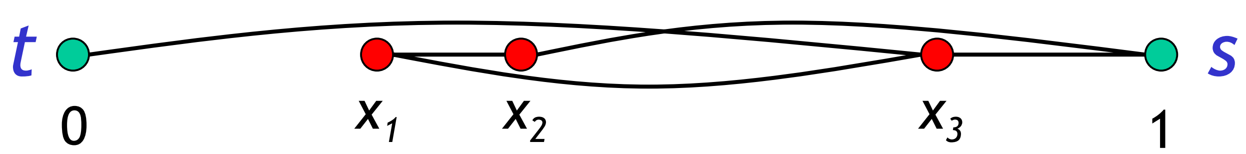
LP used for the algorithms

$$
\begin{array}{lll}
\text{Minimize} & \sum_{v \in V} x_v & \\
\text{subject to} & x_s = 1 & \\
& x_t = 0 & \\
& y_e \geq \left| x_u - x_v \right| & \text{for all } e = (u,v) \in E \\
& \sum_{e \in E} y_e \cdot c_e \leq B & \\
& x_v, y_e \geq 0 &
\end{array}
$$



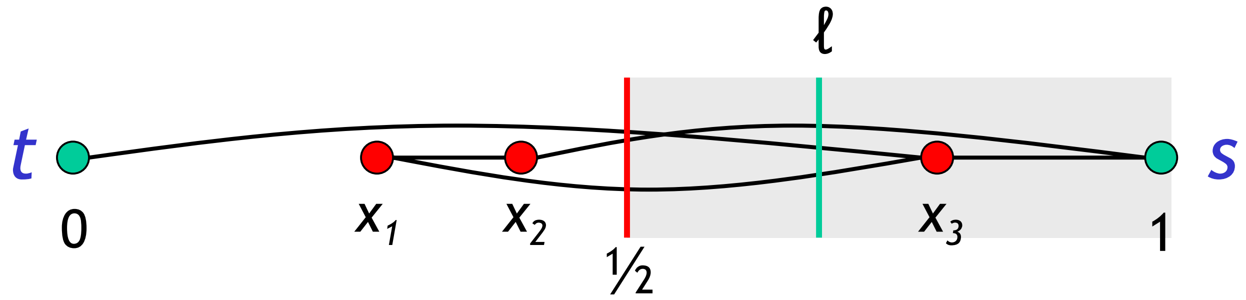
LP used for the algorithms

$$
\begin{array}{lll}
\text{Minimize} & \sum_{v \in V} x_v & \\
\text{subject to} & x_s & = 1 \\
& x_t & = 0 \\
& y_e & \geq \left| x_u - x_v \right| \quad \text{for all } e = (u, v) \in E \\
& \sum_{e \in E} y_e \cdot c_e & \leq B \\
& x_v, y_e & \geq 0
\end{array}
$$



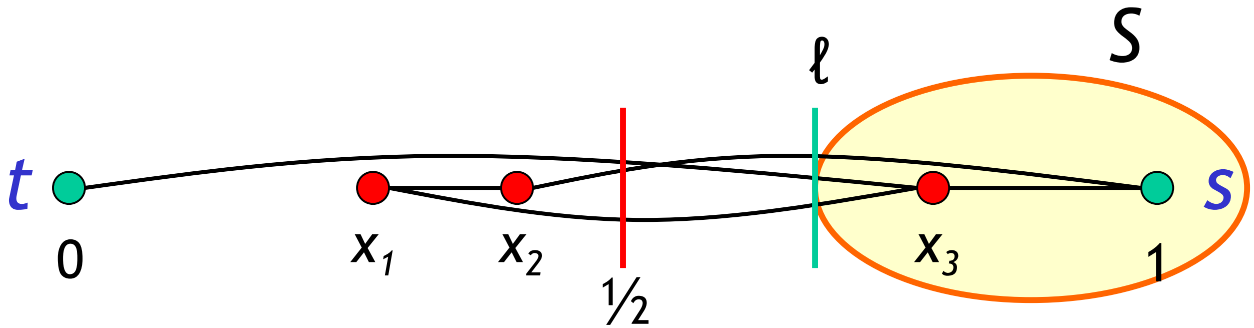
LP rounding algorithm

1. Solve the LP
2. Choose random $\ell \in [½, 1]$



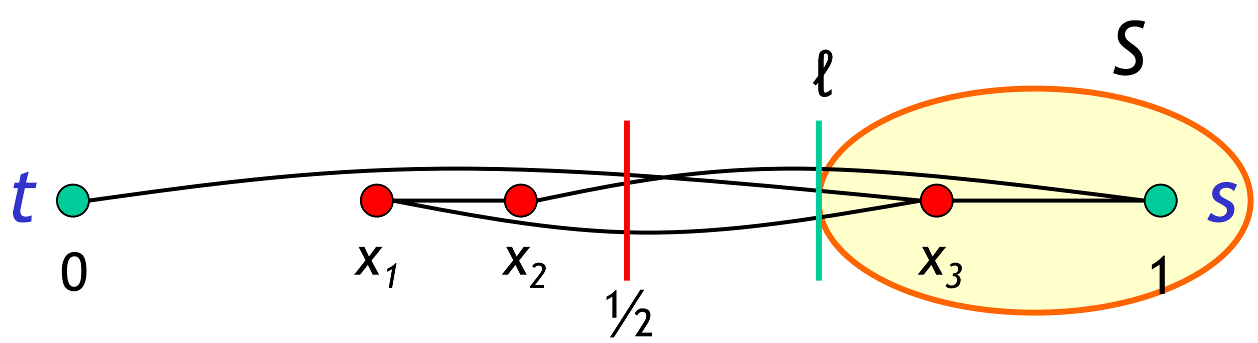
LP rounding algorithm

1. Solve the LP
2. Choose random $\ell \in [½, 1]$
3. Return set *S* of nodes with $x_i \geq \ell$



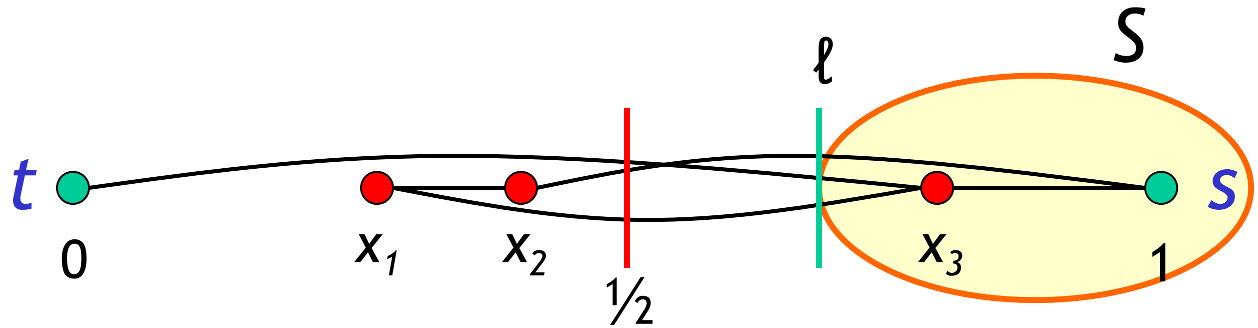
Analysis: (2,2)-bicriteria

- Claim 1: Size of $S \le 2 \cdot LP \le 2 \cdot OPT$
  - Proof: For each chosen i, $x_i \ge ½$



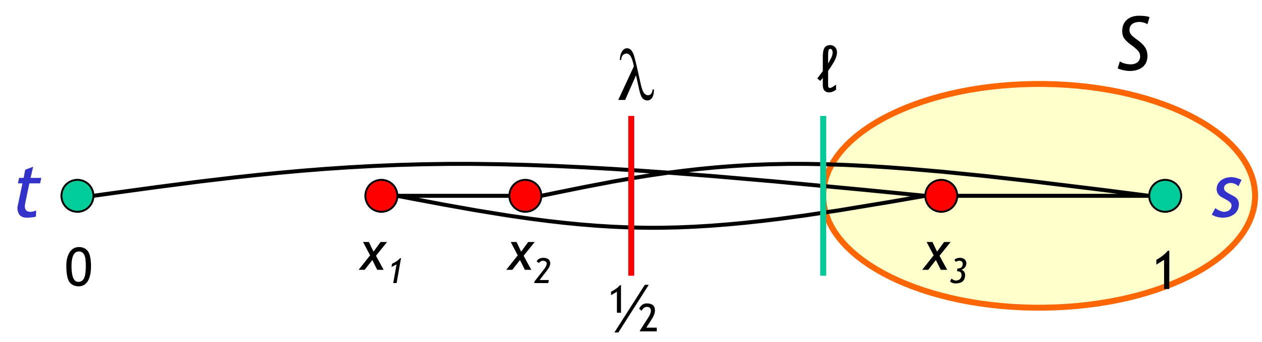
Analysis: (2,2)-bicriteria

- Claim 1: Size of $S \leq 2 \cdot LP \leq 2 \cdot OPT$
  - Proof: For each chosen i, $x_i \geq ½$
- Claim 2: Expected edges cut $\leq 2 \cdot B$
  - Proof: Choose random ℓ on segment of length ½ $\Rightarrow$ Prob. cut $e \leq 2 \cdot$ length of e



Extensions to LP rounding algorithm

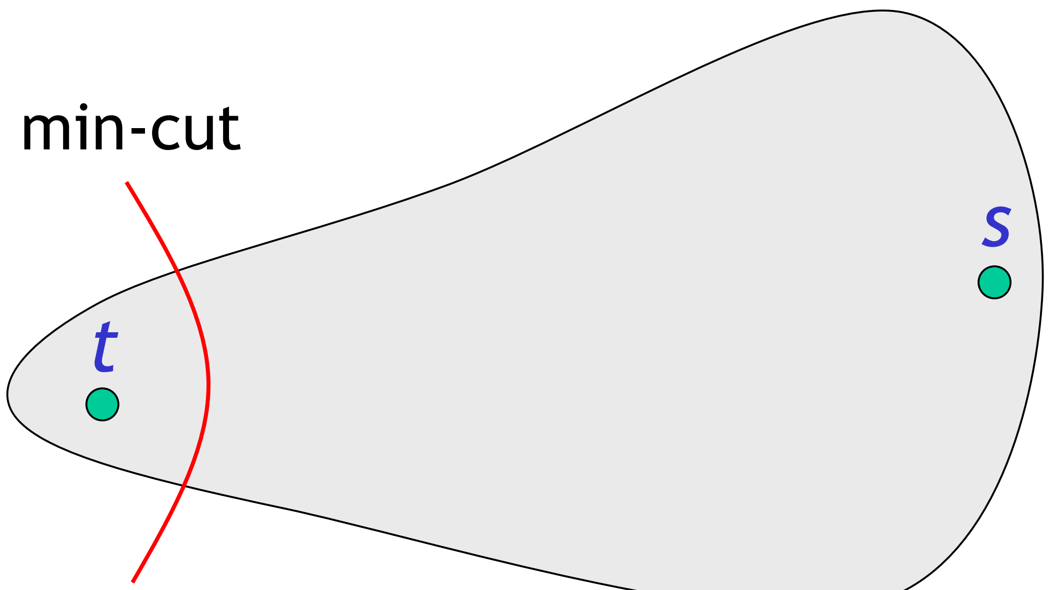
- Derandomize by trying all ℓ
- By using any $\lambda \in (0,1)$ instead of ½, get a bicriteria algorithm with
  - size of set ≤ OPT/λ
  - capacity of cut ≤ $B/(1-\lambda)$



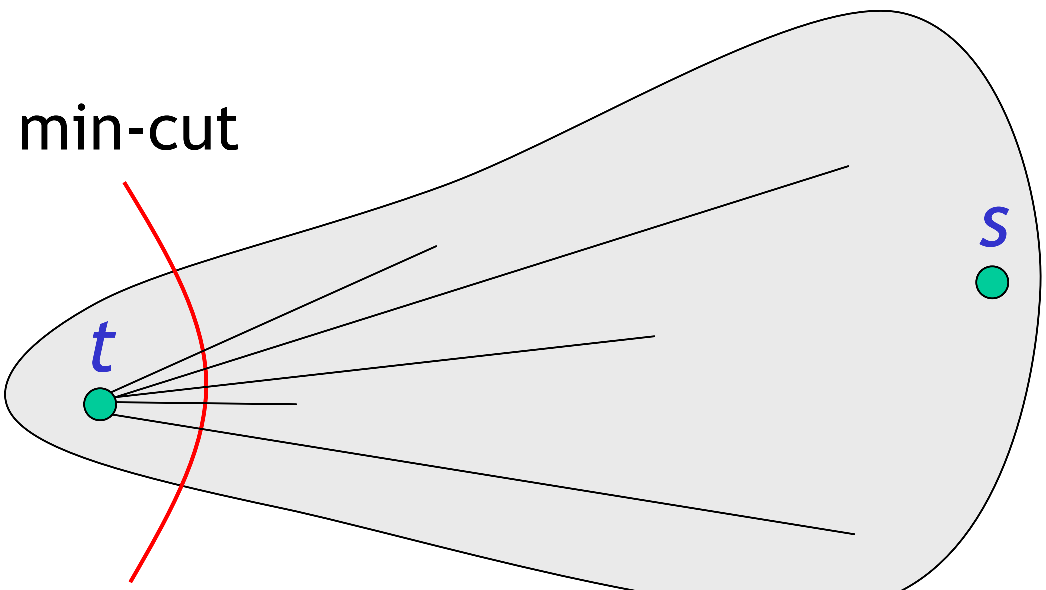
Parametric cut algorithm

- Does not solve the LP
  - running time like min-cut
  - Lagrangian relaxation used for analysis
- Approximates only one criterion: for a given $\lambda \in (0,1)$, finds a cut that
  - ***either*** has size $\leq$ OPT/λ (and capacity $\leq$ B)
  - ***or*** has capacity $\leq$ B/(1–λ) (and size optimal)

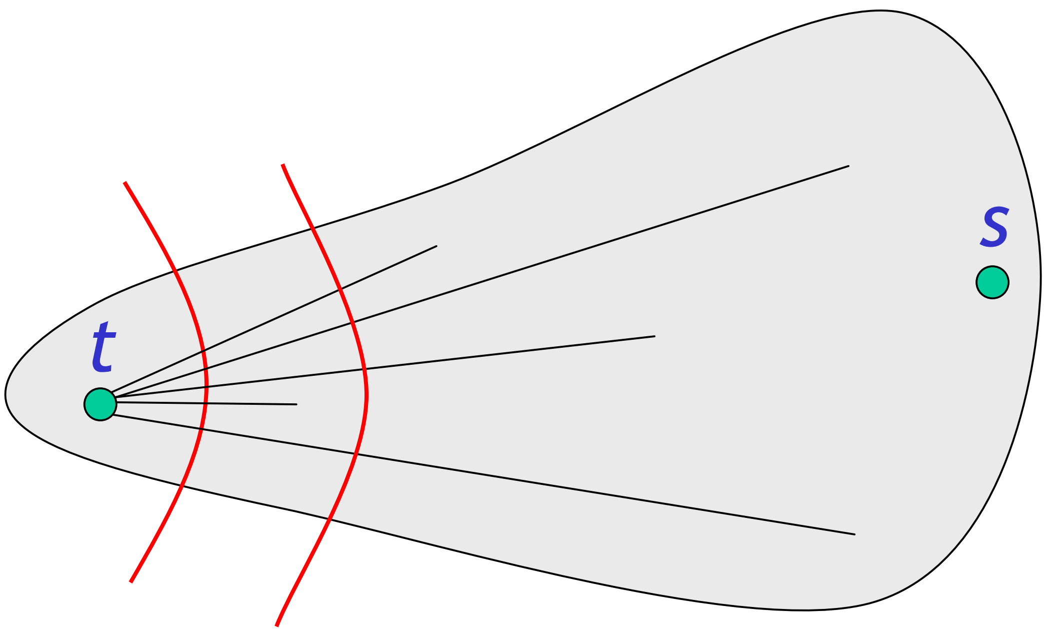
Parametric cut algorithm



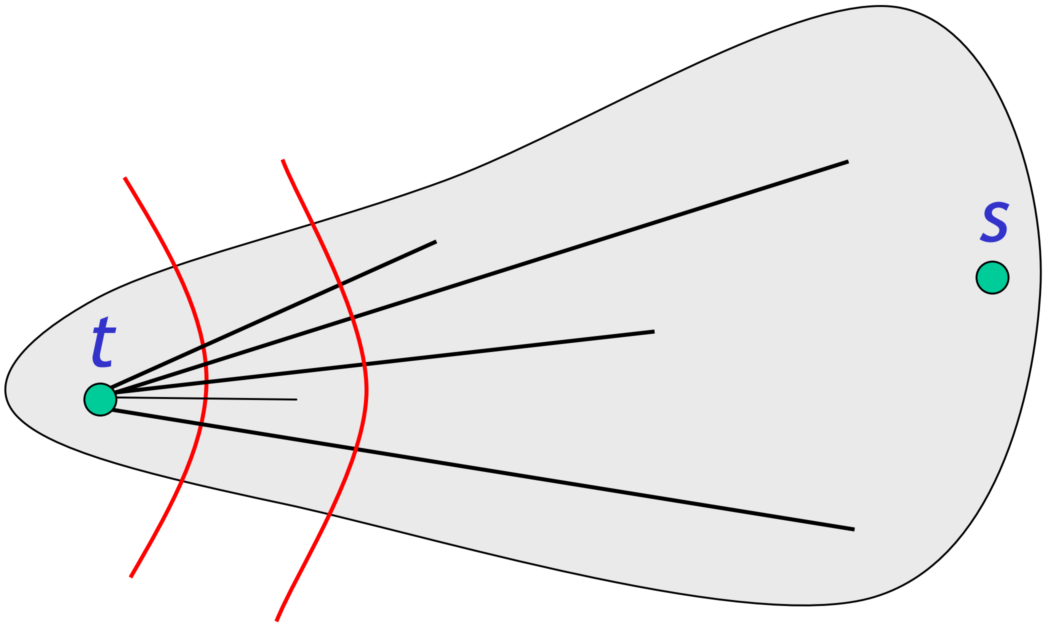
Parametric cut algorithm



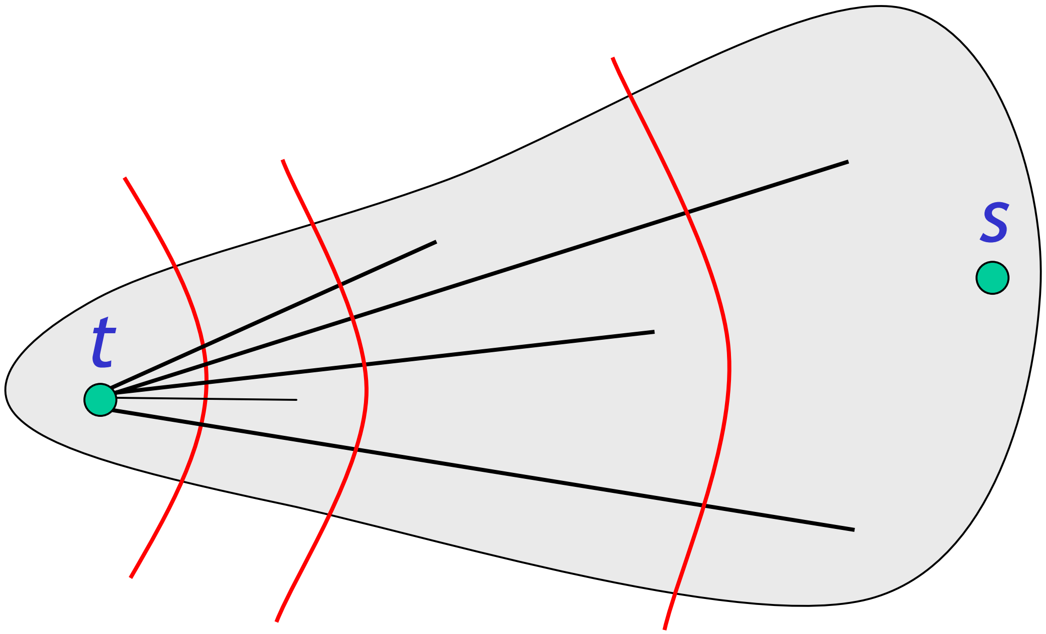
Parametric cut algorithm



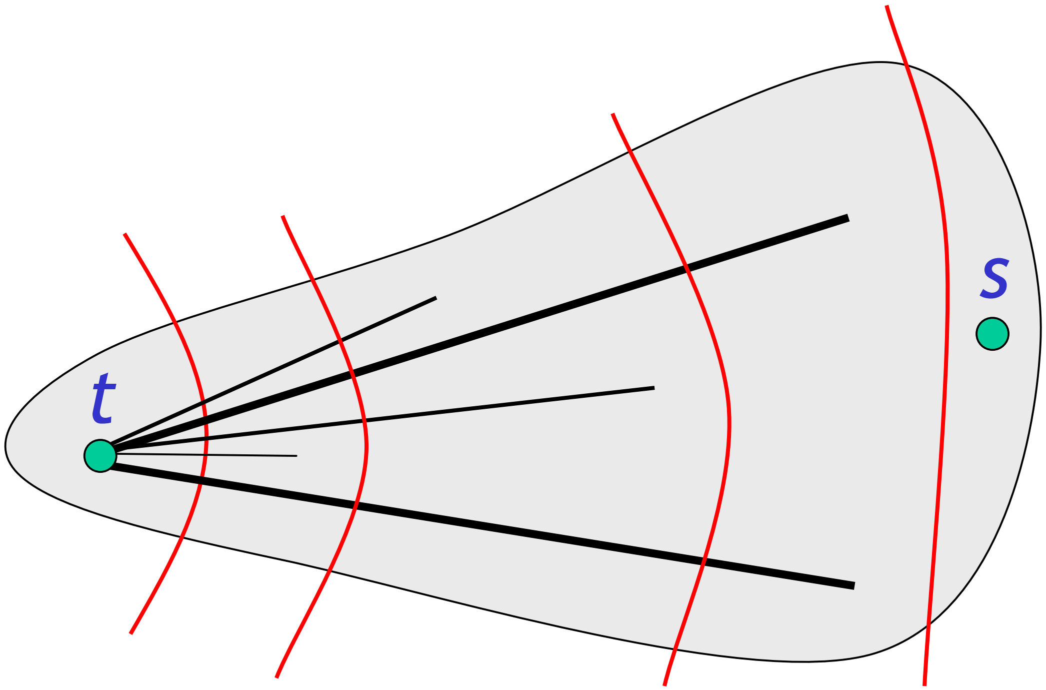
Parametric cut algorithm



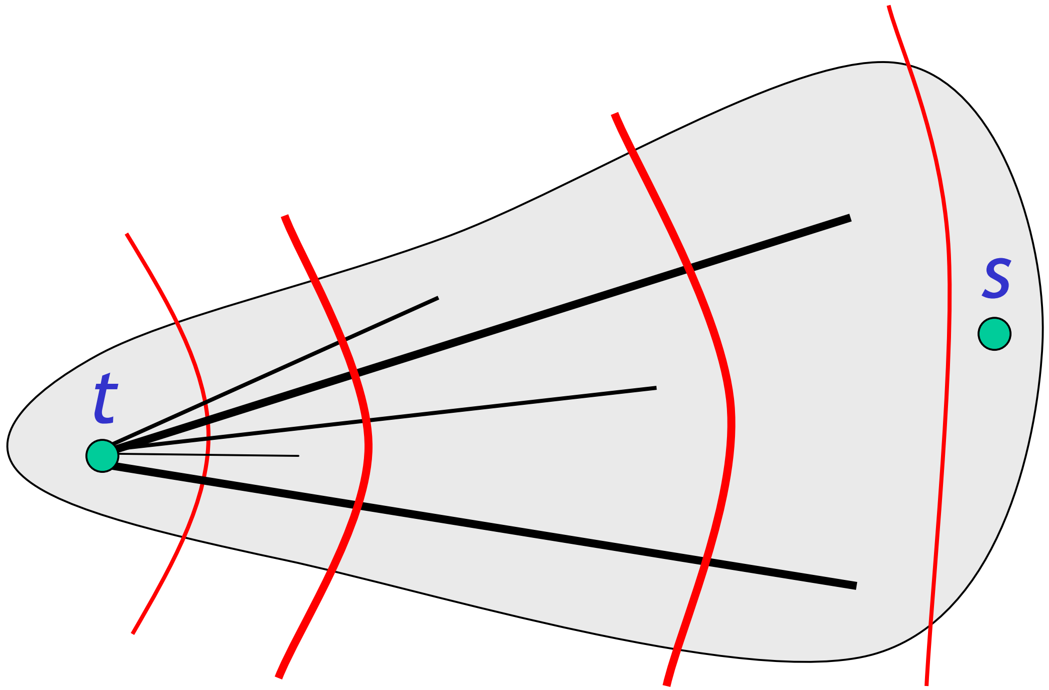
Parametric cut algorithm



Parametric cut algorithm



Parametric cut algorithm



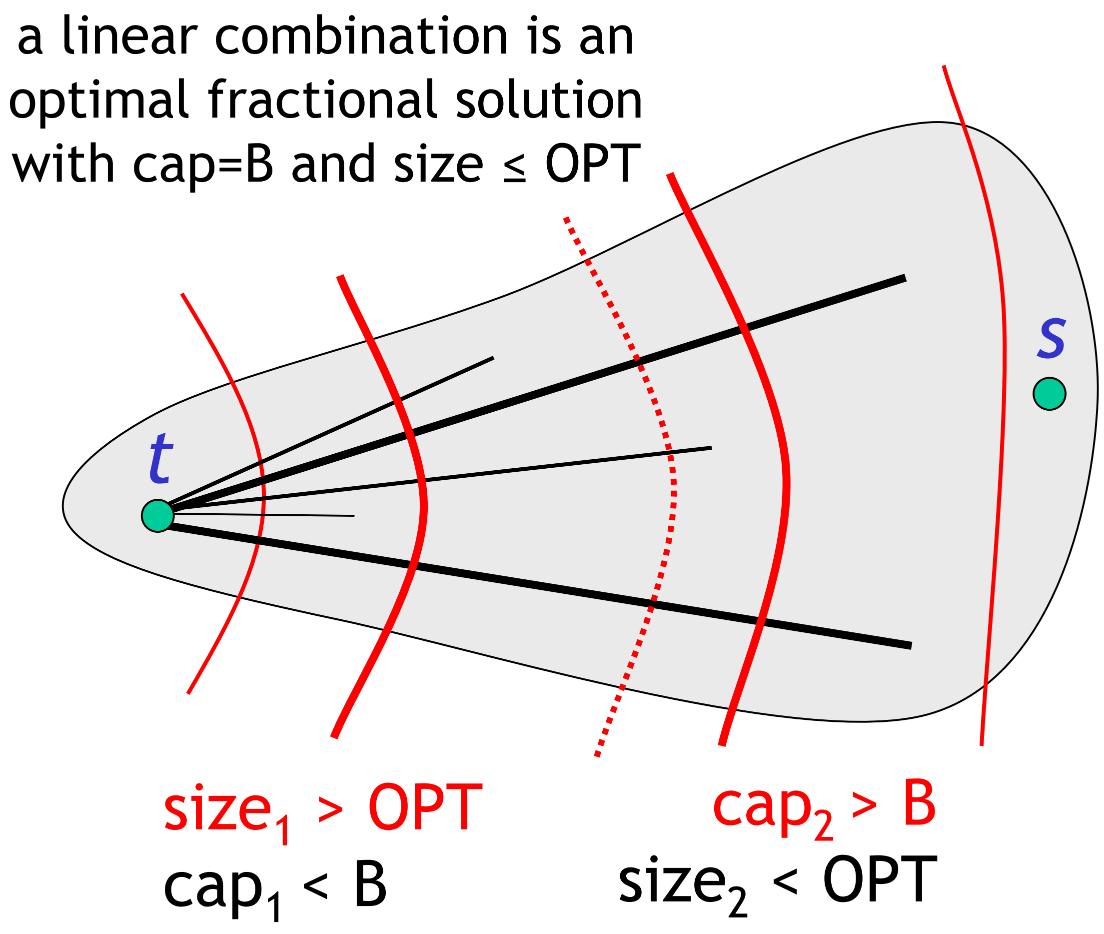
$size_1 > OPT$

$cap_1 < B$

$cap_2 > B$

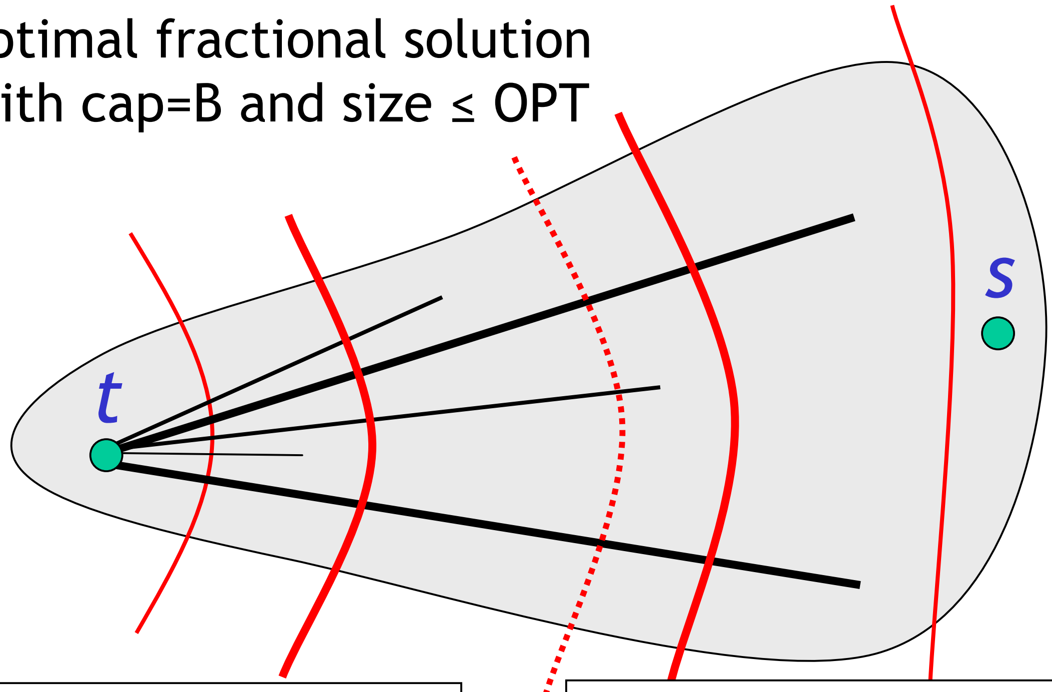
$size_2 < OPT$

Parametric cut algorithm



Parametric cut algorithm

a linear combination is an
optimal fractional solution
with cap=B and size ≤ OPT



either

OPT/λ ≥ size$_1$ > OPT
cap$_1$ < B

or

B/(1-λ) ≥ cap$_2$ > B
size$_2$ < OPT

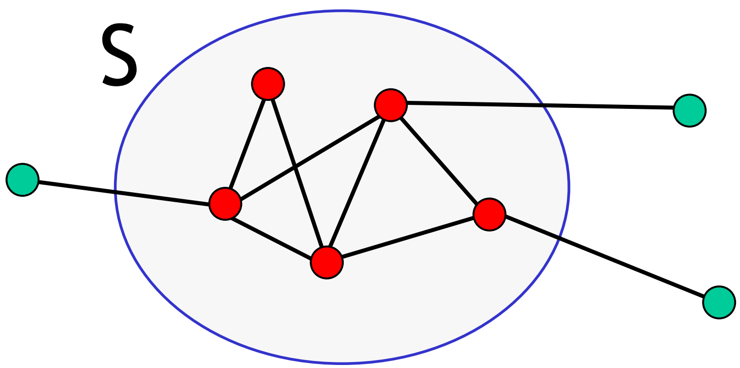
Summary of parametric cut algorithm

- Running time
  - Does not solve the LP
  - Can find all cuts in time of push-relabel algorithm
- Analysis uses Lagrangian relaxation
- Approximates only one criterion

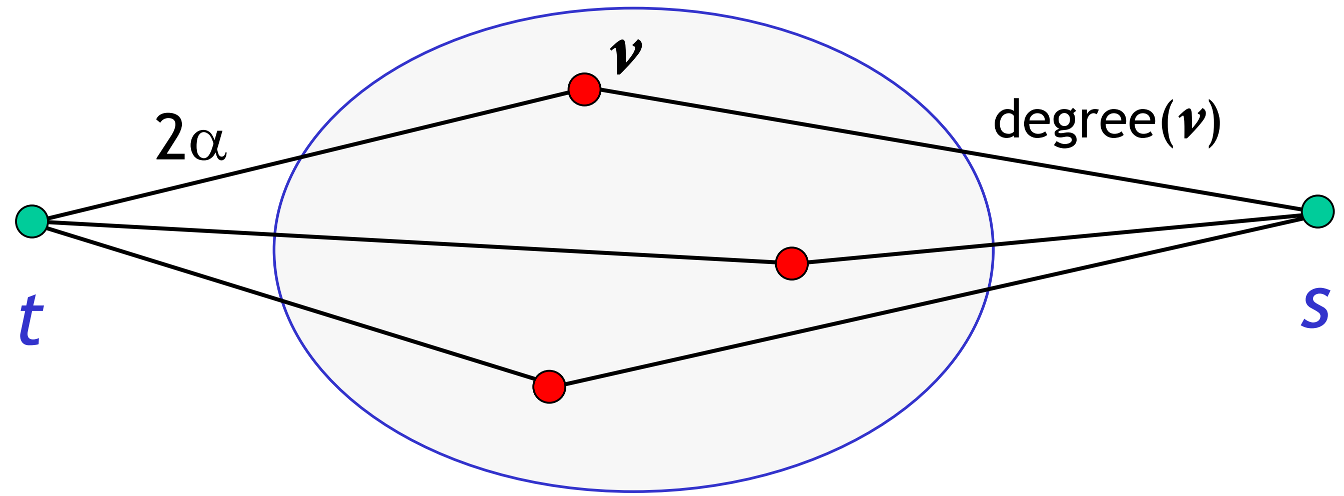
Application: Graph communities

- Define α-community = subgraph S with

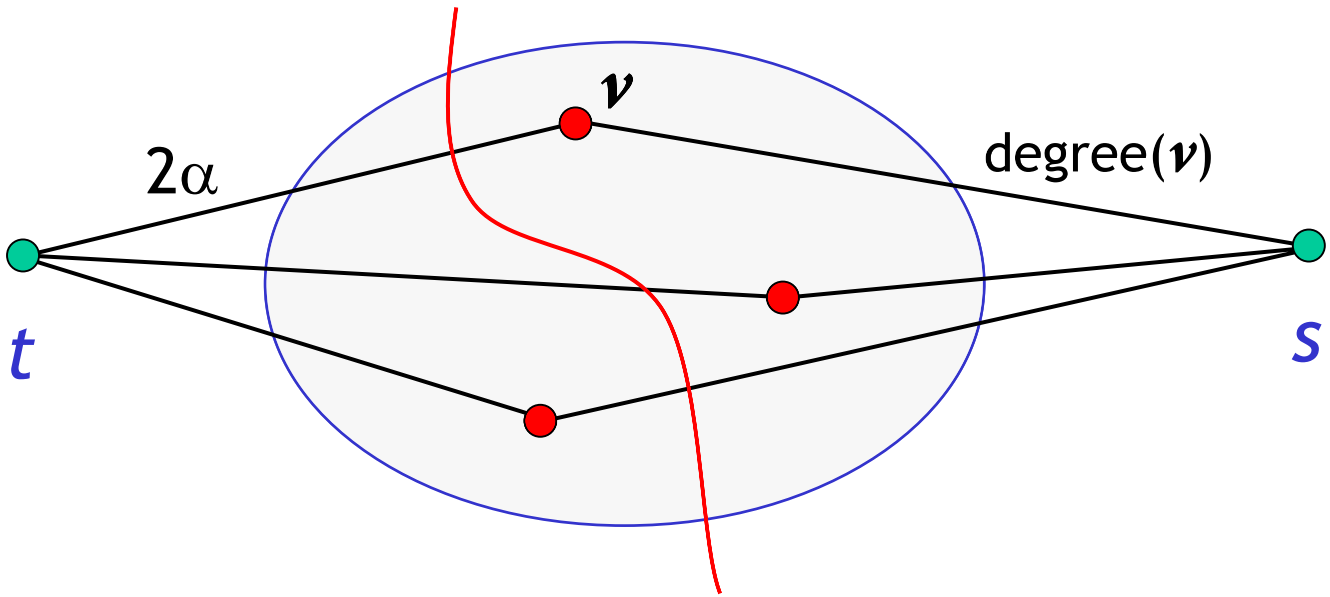
$$\frac{\text{edges inside S}}{\text{total degree in S}} \geq \alpha$$



Finding graph communities

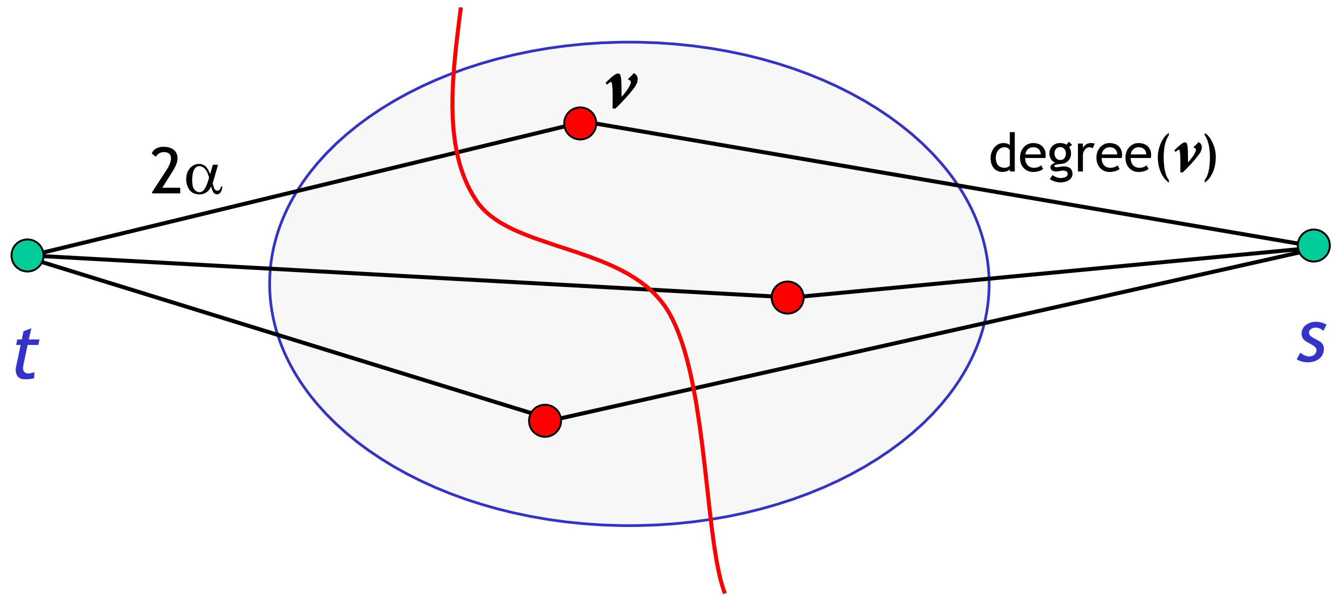


Finding graph communities



Theorem: S is an α-community iff (S ∪ {s}, V\S ∪ {t}) is a cut of capacity ≤ 2E

Finding graph communities



Theorem: S is an α-community iff (S ∪ {s}, V\S ∪ {t}) is a cut of capacity ≤ 2E

Problem: Whole graph is the best community

Goal: Given α, find the smallest α-community

Other results

- Exact algorithm for unweighted trees and graphs of bounded treewidth (PTAS if weighted)
- Extension to node cuts
- Algorithm producing optimal size and polylogarithmic violation of capacity

Open problems

- Algorithm not violating capacity bound
- Good approximation for maximization version