Maximum edge-disjoint paths problem in k-sums of graphs

Chandra Chekuri, *Guyslain Naves*, Bruce Shepherd

ICALP 2013

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

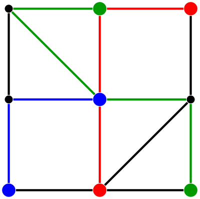
The Edge-Disjoint Paths problem (EDP)

Input:

- a capacitated graph (G, c),
- commodities $(s_i, t_i) \in V(G)^2$, for $i \in [\![1, k]\!]$. Graph $H = (T \subseteq V(G), \{(s_i, t_i)\})$.

Output: Decide whether there is:

- $(P_i)_{i \in [\![1,k]\!]}$, where P_i is a (s_i, t_i)-path in G,
- $|\{i \in [\![1, k]\!] : e \in P_i\}| \leq c(e) \quad$ for all $e \in E(G)$ (disjointness).



LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Two optimization problems

EDP is NP-hard.

Maximum Concurrent Integral Flow:
(Min Congestion Int. Flow)

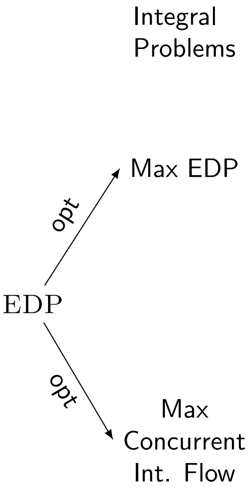
- route every commodity,
- relax disjointness: $|\{i \in [\![1,k]\!] : e \in P_i\}| \leq \frac{1}{\alpha}c(e)$,
- maximize α (minimize $\frac{1}{\alpha}$: the congestion).

Maximum Edge-Disjoint Paths: (Max EDP)

- select a subset of commodities $I \in [\![1,k]\!]$,
- route commodities in I with edge-disjoint paths,
- maximize $|I|$ (or even $w(I)$).

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

The big picture



Relaxation: multicommodity flows

Let $\mathcal{P}_i$ be the set of all $(s_i, t_i) - \text{paths}$.

IP for Min Congestion Int. Flow:

$$
\begin{array}{lll}
\max & \alpha \quad \text{subject to} & \\
 & \sum_{P \in \mathcal{P}_i} x_P = 1 & \text{(for all } i \in [\![1, k]\!]) \\
 & \sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i, e \in P} x_P \leq \frac{1}{\alpha} \cdot c_e & \text{(for all } e \in E(G)) \\
 & x \in \{0, 1\} &
\end{array}
$$

IP for Max EDP:

$$
\begin{array}{lll}
\max & \sum_{h \in H} z_h \quad \text{subject to} & \\
 & \sum_{P \in \mathcal{P}_i} x_P = z_h \leq 1 & \text{(for all } i \in [\![1, k]\!]) \\
 & \sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i, e \in P} x_P \leq c_e & \text{(for all } e \in E(G)) \\
 & x \in \{0, 1\} &
\end{array}
$$

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Relaxation: multicommodity flows

Let $\mathcal{P}_i$ be the set of all $(s_i, t_i) - \text{paths}$.

Max concurrent flow (Max Conc. Flow):

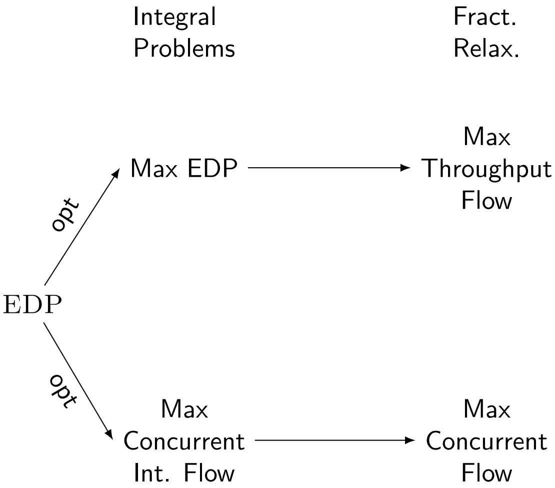
$$
\begin{aligned}
\max \quad & \alpha \quad \text{subject to} \\
& \textstyle\sum_{P \in \mathcal{P}_i} x_P = 1 && \text{(for all } i \in [\![1, k]\!]) \\
& \textstyle\sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i, e \in P} x_P \leq \frac{1}{\alpha} \cdot c_e && \text{(for all } e \in E(G)) \\
& x \geq 0
\end{aligned}
$$

Max throughput flow (Max Throughput Flow):

$$
\begin{aligned}
\max \quad & \textstyle\sum_{h \in H} z_h \quad \text{subject to} \\
& \textstyle\sum_{P \in \mathcal{P}_i} x_P = z_h \leq 1 && \text{(for all } i \in [\![1, k]\!]) \\
& \textstyle\sum_{i=1}^{k} \sum_{P \in \mathcal{P}_i, e \in P} x_P \leq c_e && \text{(for all } e \in E(G)) \\
& x \geq 0
\end{aligned}
$$

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

The big picture



LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Cut condition

Inspiration: max-flow-min-cut theorem

EDP: $c(\delta_G(X)) \geq d_H(X)$ for all $X \subseteq V(G)$

Max Conc. Flow: Sparsity of a cut X (with $d_H(X) \neq 0$):

$$\frac{c(\delta_G(X))}{d_H(X)}$$

Max Throughput Flow:
Cardinality of a multicut $E' \subseteq E(G)$:

$$E' \cap E(P) \neq \emptyset \qquad \text{for all } (s_i, t_i)\text{-path } P$$

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Cut condition

Inspiration: max-flow-min-cut theorem

EDP: $c(\delta_G(X)) \geq d_H(X)$ for all $X \subseteq V(G)$

Max Conc. Flow: Sparsity of a cut X (with $d_H(X) \neq 0$):

$$\frac{c(\delta_G(X))}{d_H(X)}$$

Max Throughput Flow:
Cardinality of a multicut $E' \subseteq E(G)$:

$$E' \cap E(P) \neq \emptyset \qquad \text{for all } (s_i, t_i)\text{-path } P$$

Any max-flow-min-cut extension? No!

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

The flow-cut gap

EDP, Min Congestion Int. Flow, Sparsest Cut:

NP-hard

Flow-Cut gap: ratio min congestion / sparsest cut
= "How far are we from sufficiency of cut condition."

- $O(\log n)$ in general (Leighton - Rao, Linial - London - Rabinovich),
- $\Omega(\log n)$ in constant degree expanders,
- $O(\sqrt{\log n})$ in planar graphs (Rao),
- $O(1)$ for SP graphs, bounded pathwidth, outerplanar,...
- Conjecture : $O(1)$ in minor-closed families (GNRS conjecture)

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Integral flow – Sparsest cut gap

Flow-cut gap: between fractional flow and sparsest cut.

Gap between integral and fractional flow ?

gap-conjecture (Chekuri, Shepherd, Weibel): there exists f such that for any family of graphs with flow-cut gap β, its integral flow-cut gap is $f(\beta)$.

Examples:

- Series-Parallel graphs (CSW). FC gap $= 2$, integrality gap $= 5$.
- Seymour K_5-free, FC gap $= 1$, integrality gap $= 2$

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

The throughput gap

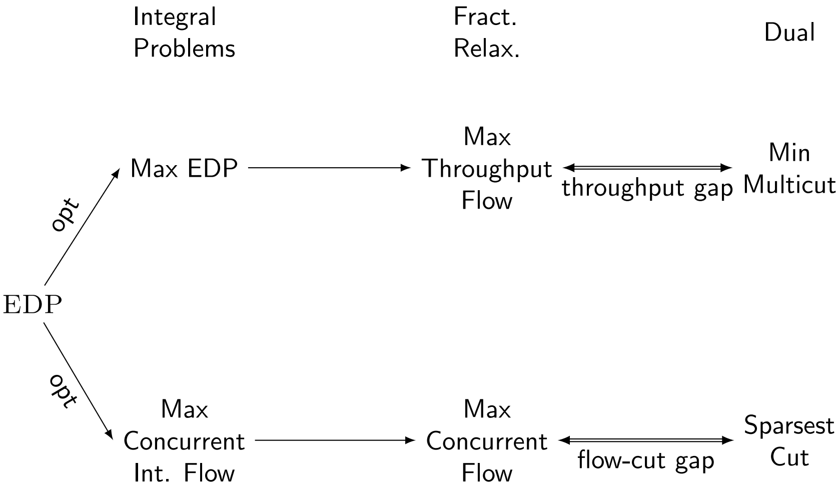
EDP, Max EDP, Min Multicut: NP-hard

Throughput gap: ratio max throughput / min multicut
= “How far are we from sufficiency of cut condition.”

- $\Theta(\sqrt{n})$ in general (Garg - Vazirani - Yannakakis, Chekuri - Khanna - Shepherd),
- $O(1)$ for minor-closed families of graphs (Klein - Plotkin - Rao, Tardos - Vazirani).

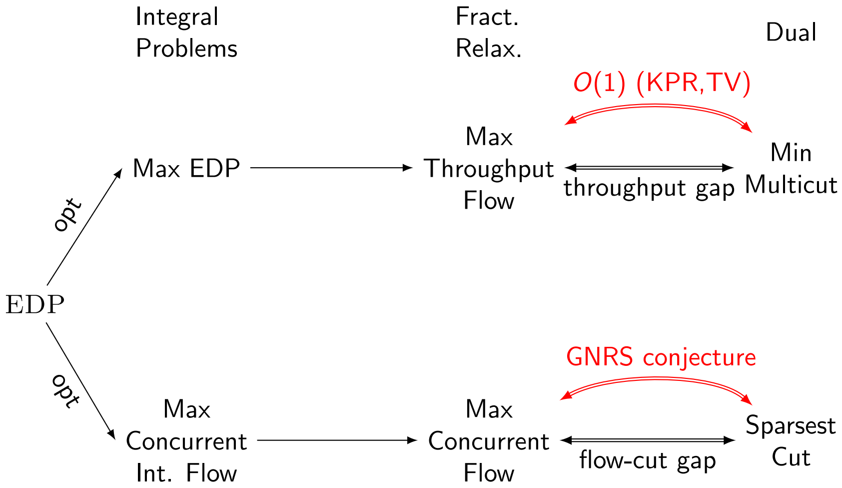
LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

The big picture



LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

The big picture



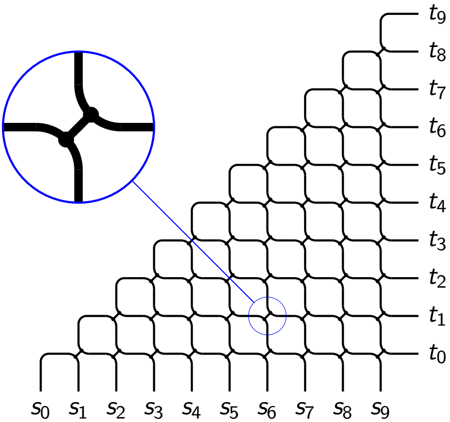
LABORATOIRE
D'INFORMATIQUE
FONDAMENTALE
de Marseille

MAX EDP– multicut gap

Throughput gap: between fractional flow throughput and multicut.

Gap between integral and fractional flow ?

Bad $O(\sqrt{n})$ (even in planar graphs):



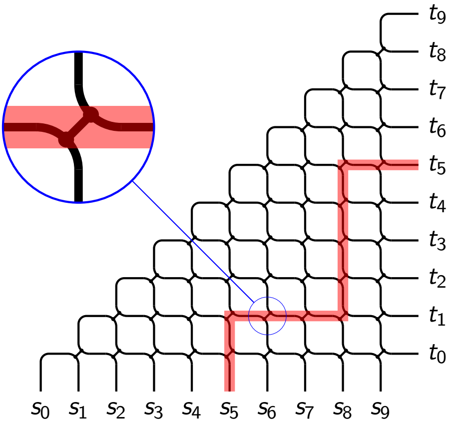
LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Max EDP– multicut gap

Throughput gap: between fractional flow throughput and multicut.

Gap between integral and fractional flow ?

Bad $O(\sqrt{n})$ (even in planar graphs):



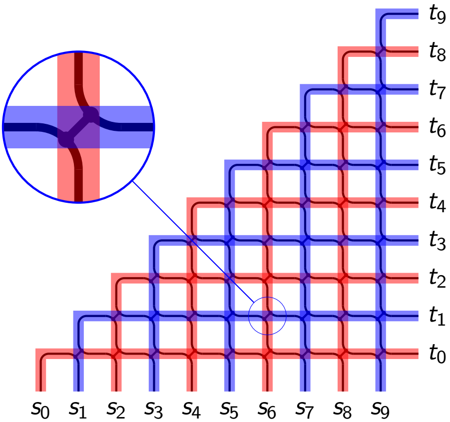
LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

MAX EDP– **multicut gap**

Throughput gap: between fractional flow throughput and multicut.

Gap between integral and fractional flow ?

Bad $O(\sqrt{n})$ (even in planar graphs):



LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Closing the integrality gap

$O(\sqrt{n})$ even in planar graphs

Idea: use constant congestion β to get a constant integrality gap. (equivalent: prove constant $\frac{1}{\beta}$-integrality gap)

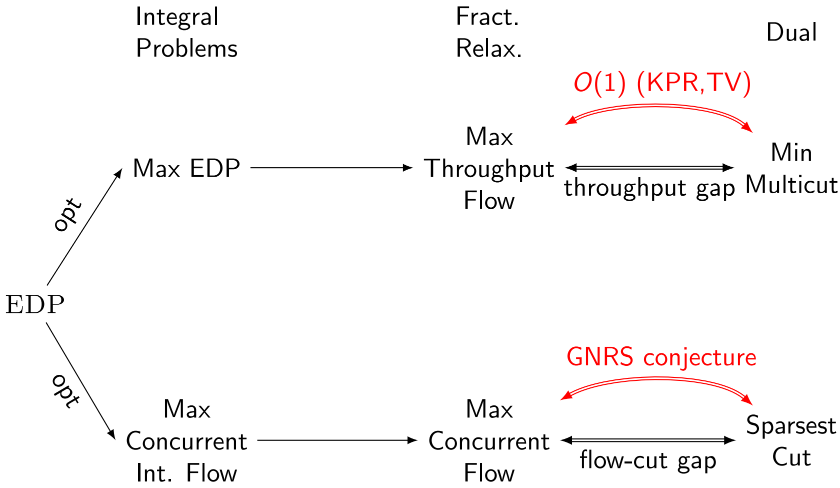
- constant gap with congestion 2 for planar graphs (Chekuri - Khanna - Shepherd, Seguin-Charbonneau - Shepherd),
- congestion 2 gives $O(\text{polylog}\, n)$ gap for any graph (Chuzhoy - Li).

Our conjecture: Constant gap with congestion 2 for any minor-closed family.

This work: Bounded treewidth and bounded genus.

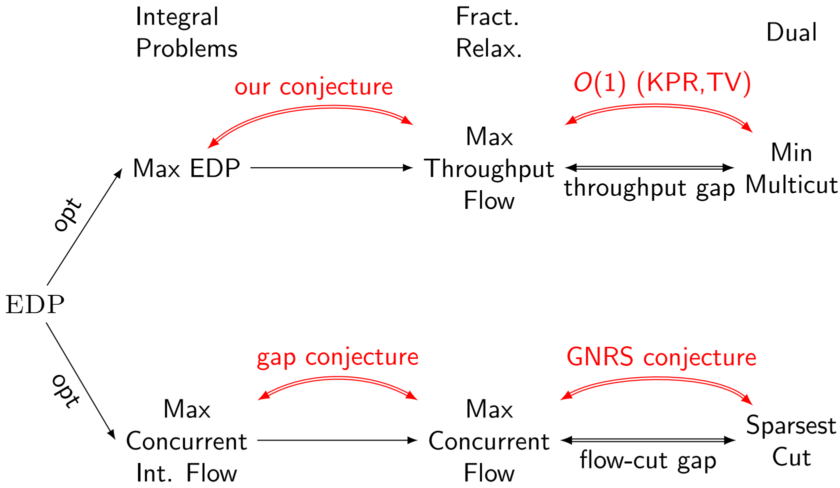
LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

The big picture



LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

The big picture



LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Minor-closed families

Theorem (Robertson-Seymour)

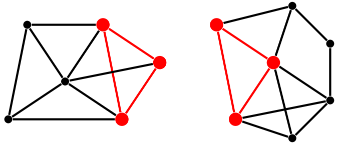
For any minor-closed family $\mathcal{G}$, there is k such that for all $G \in \mathcal{G}$, G can be obtained by:

- *draw graphs on surfaces of genus $\leq k$,*
- *add $\leq k$ vortices to each of them,*
- *add $\leq k$ apex vertices to each of them,*
- *make k-sums of these graphs.*

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

k-sums of graphs

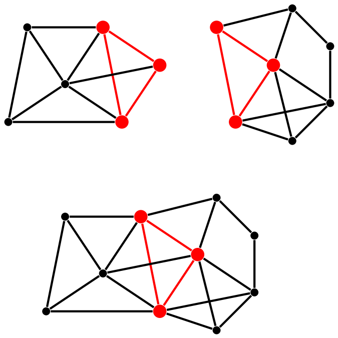
Example of 3-sum:



Treewidth k graph: k-sums of graphs on $k+1$ vertices.

k-sums of graphs

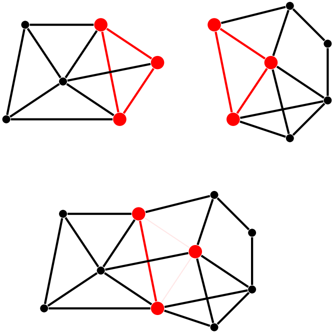
Example of 3-sum:



Treewidth k graph: k-sums of graphs on $k+1$ vertices.

k-sums of graphs

Example of 3-sum:



Treewidth k graph: k-sums of graphs on $k+1$ vertices.

Results

Theorem

If $\mathcal{G}$ is minor-closed, with integrality gap α using congestion β, then its closure by k-sums has integrality gap $2^{O(k)}\alpha$ using congestion $\beta + 3$.

Theorem

Graphs of treewidth k have integrality gap $2^{O(k)}$.

Theorem

Graphs of genus g have integrality gap $O(g \log^2(g+1))$ using congestion 3.

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Technical tools

(Partly from Chekuri - Khanna - Shepherd)

1. Moving terminals,

LABORATOIRE
D'INFORMATIQUE
FONDAMENTALE
de Marseille

Moving terminals (I)

$x(v)$: value of flow paths ending at v in fractional optimum.
$=$ marginal value at v

- If there is a flow routing $x(s), s \in S$ to $U \subset V$,
- Then, move the terminals in S to U,
- Up to approximation 5, congestion $+2$:

$$\text{flow } f \text{ in } G \xRightarrow{\text{move terminal}} \text{flow } f' \text{ in } G'$$

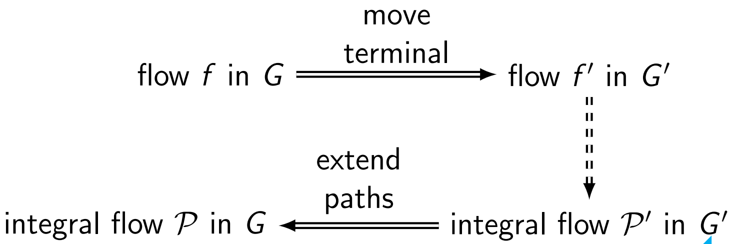
$$\text{integral flow } \mathcal{P} \text{ in } G \xLeftarrow{\text{extend paths}} \text{integral flow } \mathcal{P}' \text{ in } G'$$

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Moving terminals (I)

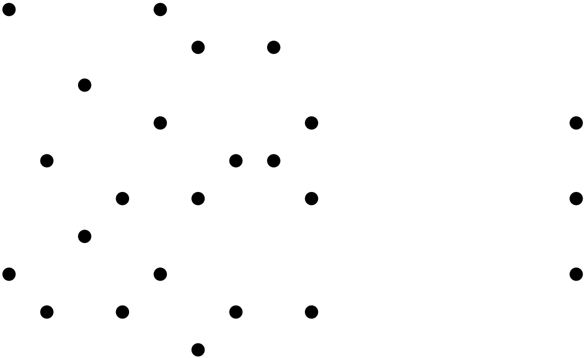
$x(v)$: value of flow paths ending at v in fractional optimum.
$= $ marginal value at v

- If there is a flow routing $x(s), s \in S$ to $U \subset V$,
- Then, move the terminals in S to U,
- Up to approximation 5, congestion $+2$:



Moving terminals (II)

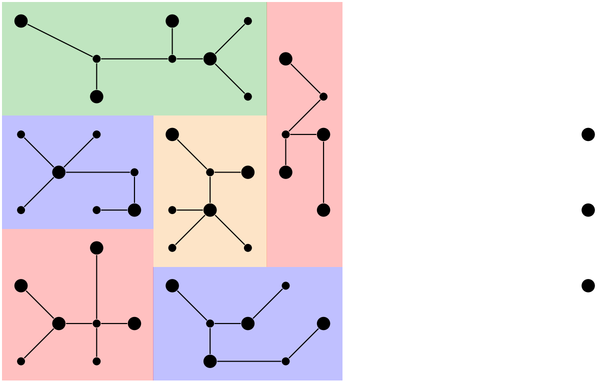
Cluster, find paths, move terminals and get back.



LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Moving terminals (II)

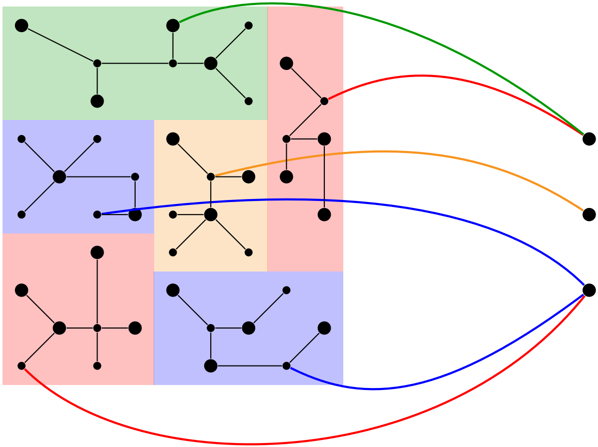
Cluster, find paths, move terminals and get back.



LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Moving terminals (II)

Cluster, find paths, move terminals and get back.



Technical tools

(Partly from Chekuri - Khanna - Shepherd)

1. Moving terminals,
2. Integral sparsifiers,

Integral sparsifiers

Let (G, c) capacitated graph, $S \subseteq V(G)$.

Goal: Get $H = (S, E_H)$ (σ, ρ)-sparsifier with

- Any multiflow between S in G is routable in H with congestion σ,
- Any integral multiflow in H is routable in G with congestion 2.

Using simple splitting-off technique:

Lemma

There is a $(|S|^2, 2)$-sparsifier.

LABORATOIRE D'INFORMATIQUE FONDAMENTALE de Marseille

Technical tools

(Partly from Chekuri - Khanna - Shepherd)

1. Moving terminals,
2. Integral sparsifiers,
3. Routing through bounded number of nodes.

Routing through one node

Lemma

If every flow path of a fractional solution goes through some vertex v, there is an integral solution rounting $\frac{1}{12}$ as much.

Easily extends to small subsets of vertices.

Solves the apex step.