Algorithms for integral and half-integral multiflows

Gyula Pap

Department of Computer Science, Cornell University

Oktober 19, 2009

Outline

- Main goal is to solve maximum multiflow problems subject to node-capacities (instead of edge-capacities)
- A lot is known about edge-capacitated multiflows:
  Lovász-Cherkassky 1972 (half-integrality of the LP),
  Ibaraki-Karzanov-Nagamochi 1998 (strongly polynomial time algo)
- Node-capacitated multiflows: LP has half-integral optimum
- Example: node-capacitated LP may have a "very fractional" extreme optimum
- Main result: Polytope of shortest maximum multiflows subject to node-capacities is half-integral
- Implies: a strongly polynomial time algorithm via ellipsoid method
- Best result before: a weakly-polynomial time algorithm via capacity-scaling (Babenko-Karzanov, 2008)

Motivation

Known result:

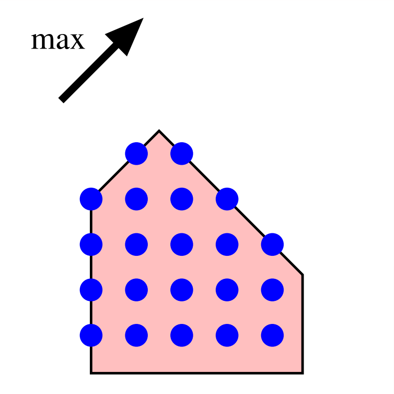
LP for maximum multiflow subject to edge- or node-capacities has as half-integral optimum.

Goal:

Find a half-integral maximum in strongly polynomial time!

Unlucky circumstance:

Very fractional basic optima.

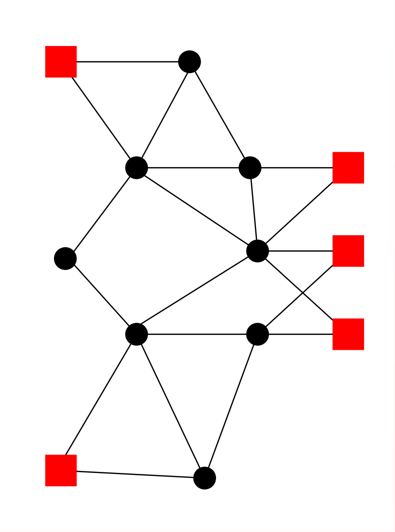


Definition of a multiflow

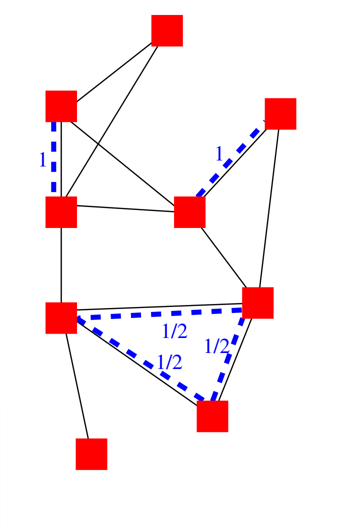
- We are given an undirected graph $G = (V, E)$ and a set of **terminals** $A \subseteq V$.
- A multiflow, denoted by

  $$x = (x_{ab} : a \neq b, a, b \in A),$$

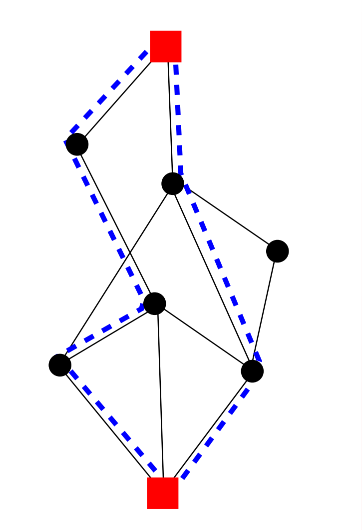
  is a set of $\binom{|A|}{2}$ flows x_{ab} between all pairs of terminals $a \neq b, a, b \in A$.
- $size(x) := \sum size(x_{ab})$
- Problem: maximize the size of a multiflow subject to certain capacity and/or integrality constraints.



$A = V$:
max (fractional) matching



$|A| = 2$:
max s–t flow

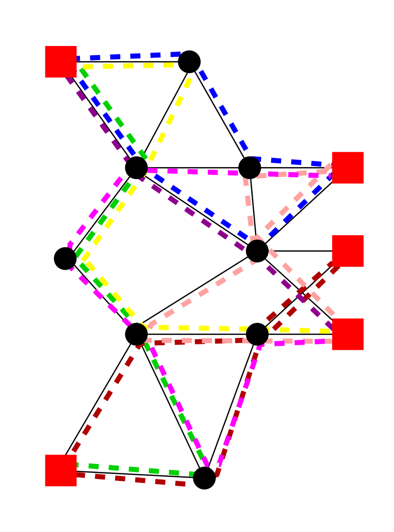


Definition of a multiflow

- We are given an undirected graph $G = (V, E)$ and a set of **terminals** $A \subseteq V$.
- A multiflow, denoted by

  $$x = (x_{ab} : a \neq b, a, b \in A),$$

  is a set of $\binom{|A|}{2}$ flows x_{ab} between all pairs of terminals $a \neq b, a, b \in A$.
- $size(x) := \sum size(x_{ab})$
- Problem: maximize the size of a multiflow subject to certain capacity and/or integrality constraints.



All-one edge-capacities, Eulerian graph

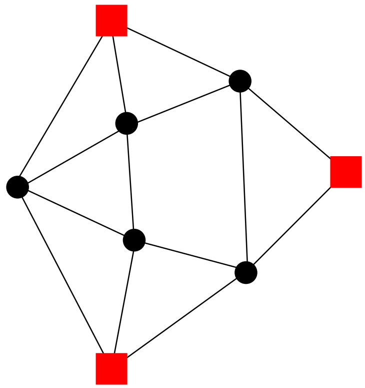
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2}\sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.

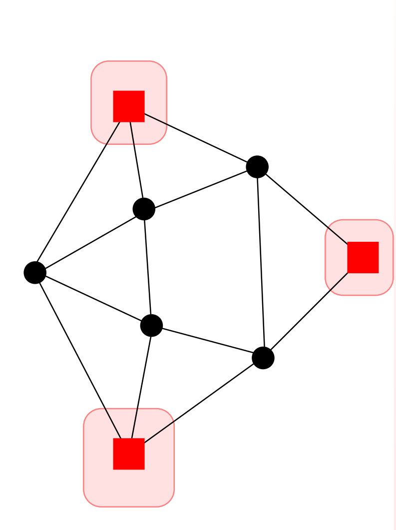


All-one edge-capacities, Eulerian graph

Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2}\sum_{a\in A}\lambda(a, A-a)$$



(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.

All-one edge-capacities, Eulerian graph

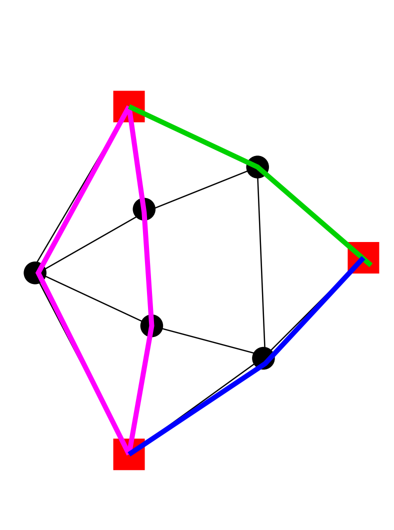
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2} \sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.



All-one edge-capacities, Eulerian graph

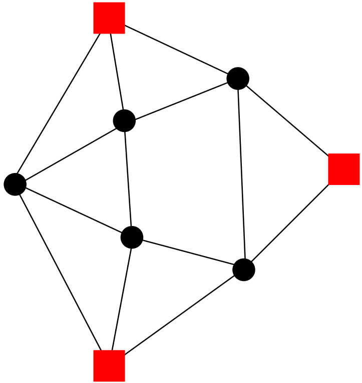
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2} \sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.



All-one edge-capacities, Eulerian graph

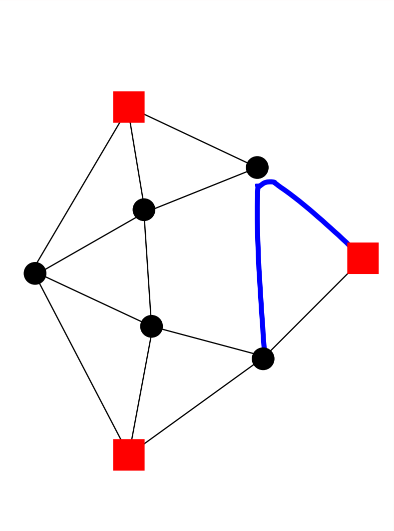
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2} \sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.



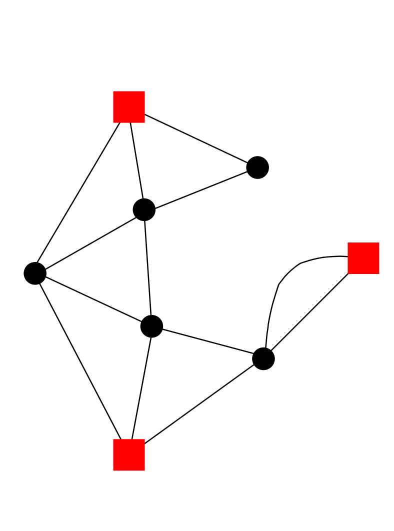
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2}\sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.



All-one edge-capacities, Eulerian graph

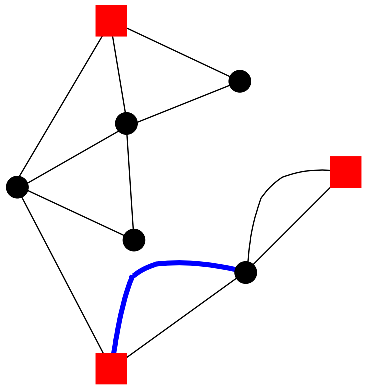
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2}\sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.



All-one edge-capacities, Eulerian graph

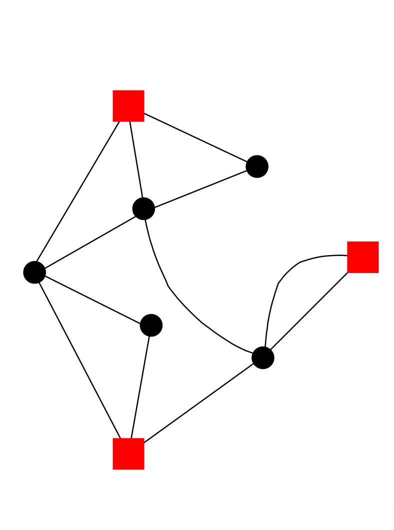
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2} \sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.



All-one edge-capacities, Eulerian graph

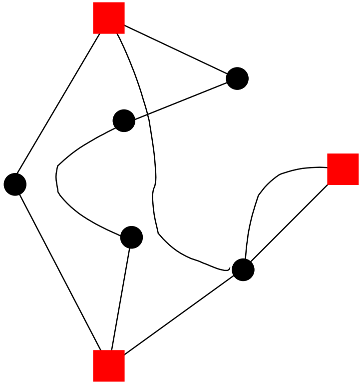
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2} \sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.



All-one edge-capacities, Eulerian graph

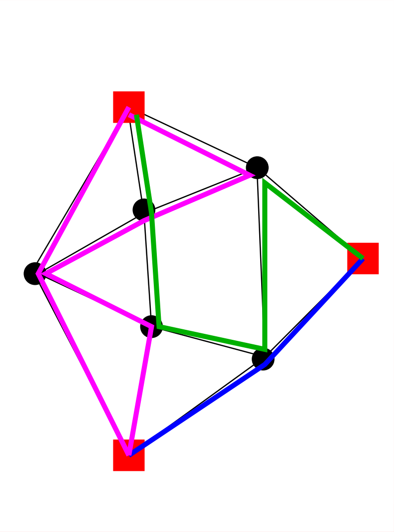
Lovász, Cherkassky, ≈1972

Let $G = (V, E)$ be eulerian, and let $A \subseteq V$. Then the maximum size of an integral multiflow subject to all-one edge-capacities is equal to

$$\frac{1}{2}\sum_{a \in A} \lambda(a, A - a)$$

(Essence: "cut condition" is necessary and sufficient.)

Proof: Split-off. Implies a polynomial time algorithm.



LP of Edge-Capacitated Multiflows (condensed)

Let $G = (V, E)$ be a graph, $A \subseteq V$ be the set of terminals, and let $c : E \to \mathbb{N}$ denote a vector of integral edge-capacities.

---

max $size(x)$ subject to

$$\begin{cases} x & \text{is a multiflow (non-negativity, flow-conservation)} \\ total.load_x(e) \leq c(e) & \text{for all } e \in E \text{ (capacity constraint)} \end{cases}$$

LP of Edge-Capacitated Multiflows

$G = (V, E)$, $c : E \to \mathbb{N}$, and denote $A = \{a_1, a_2, \cdots, a_k\} \subseteq V$.

$$\max \sum_{i<j} x_{ij}(\delta^{in}(a_j)) \quad \text{subject to}$$

$$x \geq 0$$

$$\sum_{va_l \in E} x_{ij}(va_l) = 0 \qquad \text{for } i < j, l \neq j$$

$$\sum_{av_l \in E} x_{ij}(a_l v) = 0 \qquad \text{for } i < j, l \neq i$$

$$\sum_{zv \in E} x_{ij}(zv) - \sum_{vz \in E} x_{ij}(vz) = 0 \qquad \text{for } i < j, v \in V - A$$

$$\sum_{i<j} (x_{ij}(pq) + x_{ij}(qp)) \leq c(pq) \qquad \text{for } pq \in E.$$

Half-Integral Edge-Capacitated Multiflows

Corollary – Half-Integrality

For any $G = (V, E), A \subseteq V, c : E \to \mathbb{N}$, the above LP has a half-integral optimum.

Proof:

- Construct eulerian graph G^c by replacing edge pq by a number of $2c(pq)$ paralel edges pq.
- Lovász-Cherkassky for G^c, A implies a half-integral multiflow of size equal to the min cut upper bound.
- The min cut upper bound transforms into an LP dual of same value. (Same as for s–t flows.)

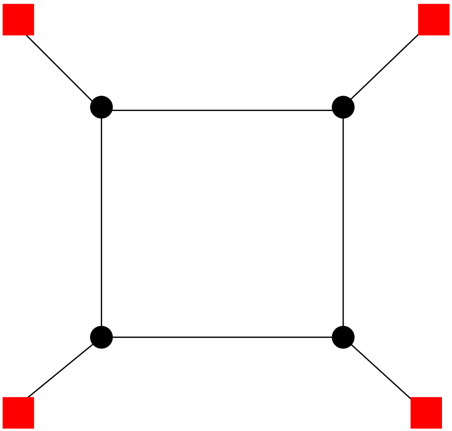
Half-Integral Edge-Capacitated Multiflows

Question: How can we find a half-integral optimum in strongly polynomial time?

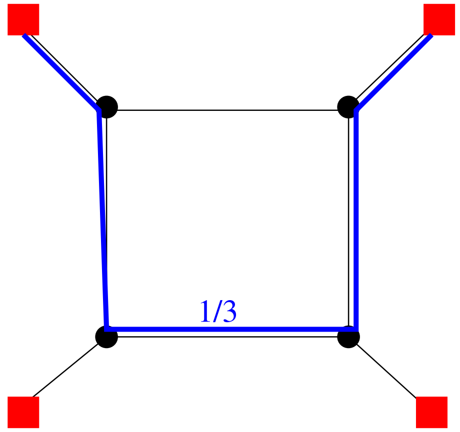
Approach 1

- Find an extreme optimum of the LP. Maybe it is guaranteed to be half-integral. This would imply a strongly polynomial time algorithm by Frank-Tardos.
- Unluckily, the following example show that the polytope of edge-capacitated multiflows may have very fractional extreme points.

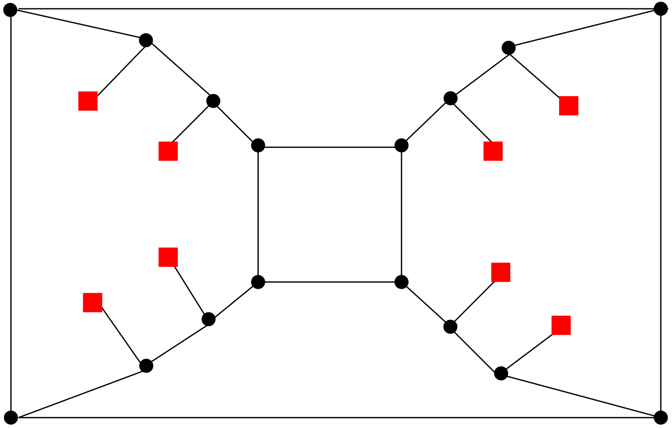
The polytope of multiflows can have a fractional extreme point:



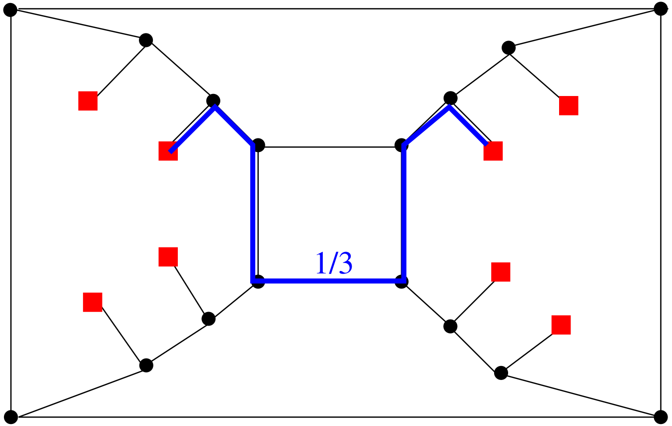
The polytope of multiflows can have a fractional extreme point:



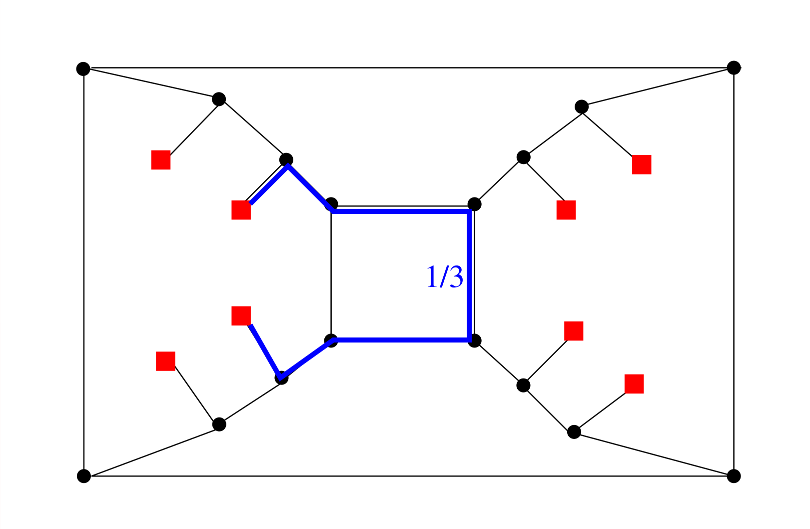
The polytope of maximum multiflows can have a fractional extreme point:



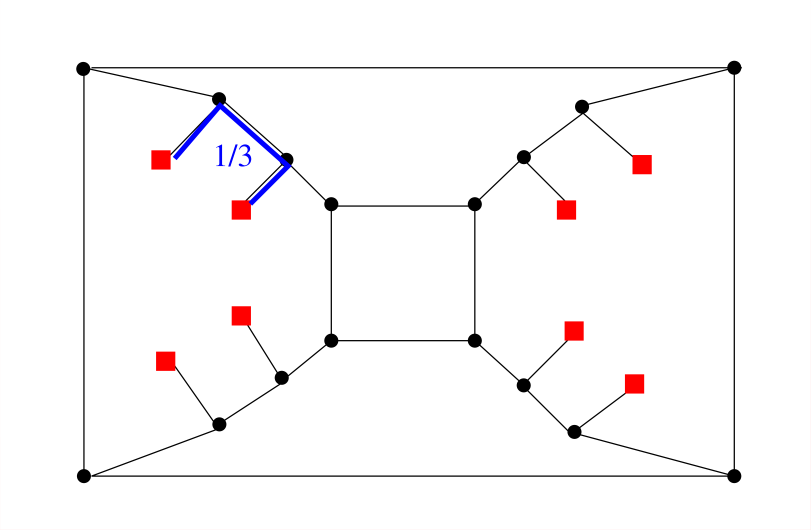
The polytope of maximum multiflows can have a fractional extreme point:



The polytope of maximum multiflows can have a fractional extreme point:



The polytope of maximum multiflows can have a fractional extreme point:



Half-Integral Edge-Capacitated Multiflows

Approach 2 — Use the integer/fractional parts of an LP optimum

- Solve LP and then "play with the fractions" until half-integral.
- Implies a polynomial time algorithm by applying the uncapacitated algorithm for the fractional part.
- Is this strongly polynomial?
- You can solve the LP in strongly polynomial time by a result of Frank and Tardos. But, ...
- ... by definition, you mustn't take the fractional part of numbers in a strongly polynomial time algorithm!

What is a strongly polynomial time algorithm?

- An algorithm – with integer/rational numbers in the input – is called strongly polynomial if it
  1. performs a polynomial number of operations $+, -, \cdot, /, \leq$
  2. size of numbers throughout the computation stays within a polynomial factor of the size of the input numbers.
- Certain natural algorithms are NOT strongly polynomial:
  1. rounding a rational number (euclidian algorithm)
  2. determining 2^k in decimals,
  3. (most) scaling algorithms.
- Strongly polynomial:
  1. maximum flow, min cost circulation, min cost perfect matching, maximum weight capacitated *b*-matching, submodular function minimization
  2. $0, \pm 1$ LP (Frank, Tardos, using ellipsoid method)

Fun Fact:
determining 2^k is not strongly polynomial, yet only requires a polynomial number of arithmetic operations

700000000000=1010001011111011010000000010110000000000000

implying that

$2^{700000000000} =$

$= 2^{2^{11}} \cdot 2^{2^{12}} \cdot 2^{2^{14}} \cdot 2^{2^{22}} \cdot 2^{2^{24}} \cdot 2^{2^{25}} \cdot 2^{2^{27}} \cdot 2^{2^{28}} \cdot 2^{2^{29}} \cdot 2^{2^{30}} \cdot 2^{2^{31}} \cdot 2^{2^{33}} \cdot 2^{2^{37}} \cdot 2^{2^{39}}$

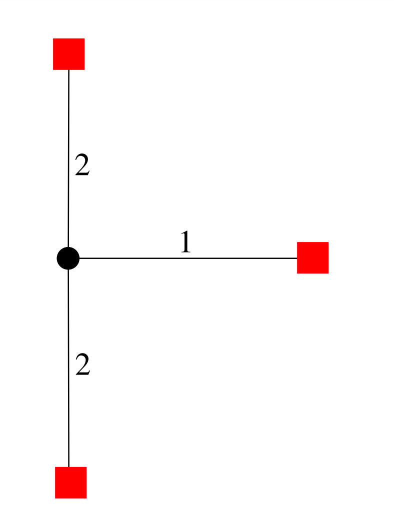
requiring a total of 375 arithmetic operations.

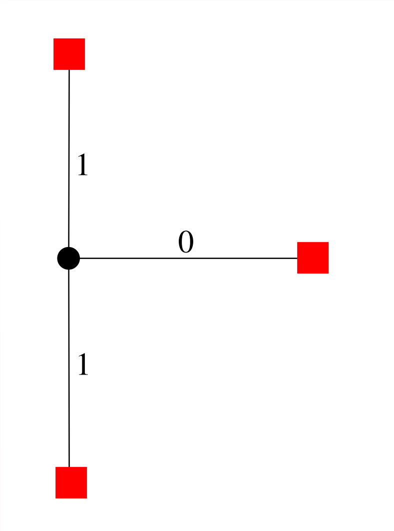
— performs a polynomial number of operations, but is not strongly polynomial, it is not even polynomial

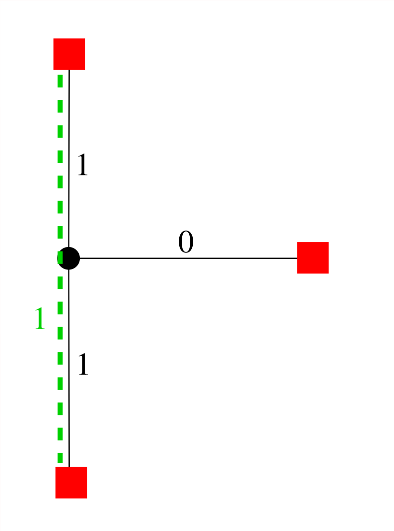
Approach 3 — Capacity splitting-off

- Solve LP to determine the maximum weight that can be split-off from edges pq, qr so that $p \in A$. (ellipsoid method)
- Perform this maximum split-off, and repeat for a different pair
- Unluckily, this doesn't maintain integrality of the capacities, so we will potentially get a badly fractional optimum, as bad as 2^{-cn}

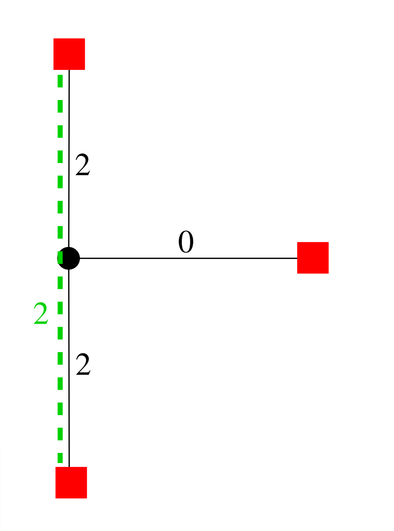
Capacity Scaling for Half-Integral Edge-Capacitated Multiflows

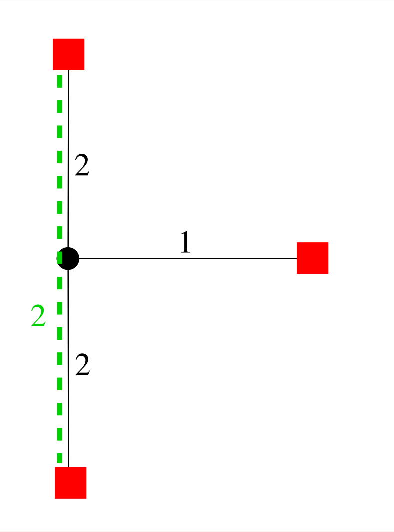


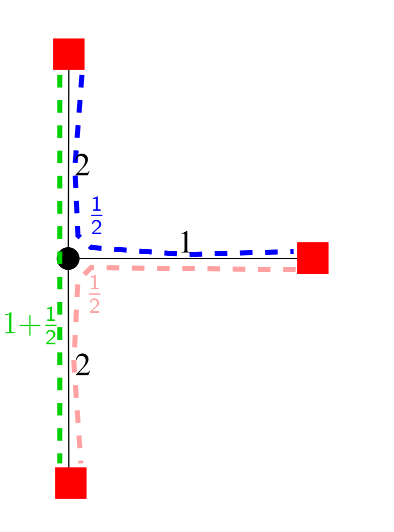

1
0
1


1
0
1
1

Capacity Scaling for Half-Integral Edge-Capacitated Multiflows




2
1
2
2


2
$\frac{1}{2}$
1
$\frac{1}{2}$
$1+\frac{1}{2}$
2

Capacity scaling + Lovász-Cherkassky $\Rightarrow$

weakly polynomial time algorithm to find a maximum half-integral multiflow. Naive running time using Goldberg-Rao for min cut:

$$O(m^3 n \min\{n^{2/3}, m^{1/2}\} \log(n^2/m) \log^2 n \log U) \approx O(n^8 \log U)$$

Remark: Babenko, Karzanov (ESA '08) constructed a faster scaling algorithm for the more general node-capacitated problem, achieving a running time of

$$O(mn^2 \min\{n^{2/3}, m^{1/2}\} \log(n^2/m) \log^2 n \log^2 U) \approx O(n^5 \log^2 U),$$

where U denotes the maximum capacity of a node.

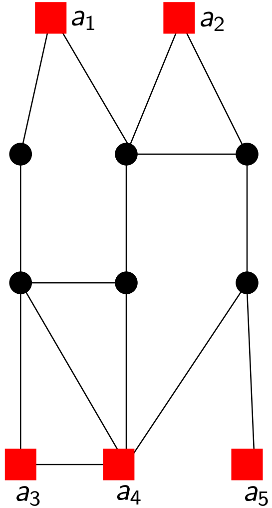
Half-Integral Edge-Capacitated Multiflows

A strongly polynomial time algorithm based on an idea of Karzanov, Ibaraki, Nagamochi (1998): ⋆

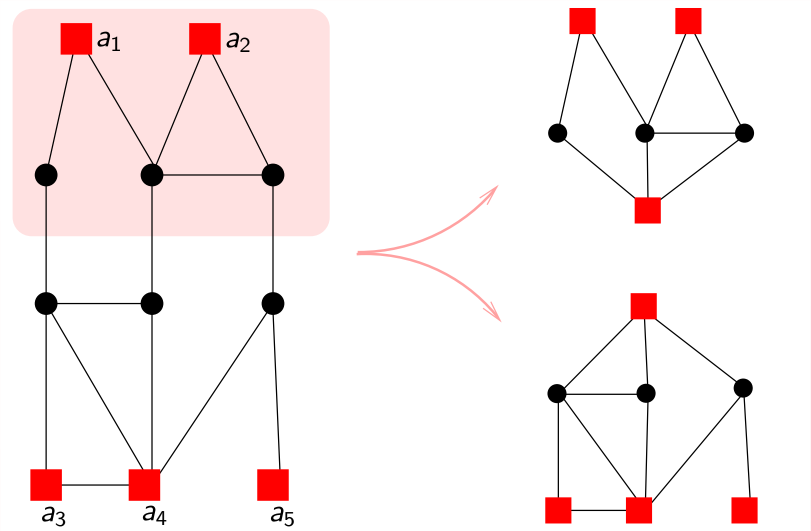
1. Decompose the graph by minimum $\{a_1, a_2\}$ - $\{a_3, .., a_k\}$ cuts into instances with three terminals.
2. Solve the three terminal case by combining two circulations.

⋆ they actually solve a more general problem on locking multiflows

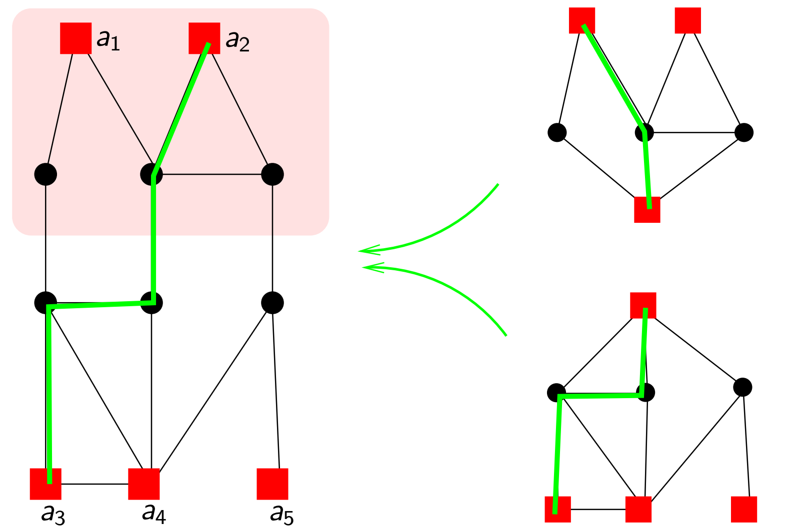
Decompose by $\{a_1, a_2\}$ - $\{a_3, .., a_k\}$ cuts



Decompose by $\{a_1, a_2\}$ - $\{a_3, .., a_k\}$ cuts



Decompose by $\{a_1, a_2\}$ - $\{a_3, .., a_k\}$ cuts



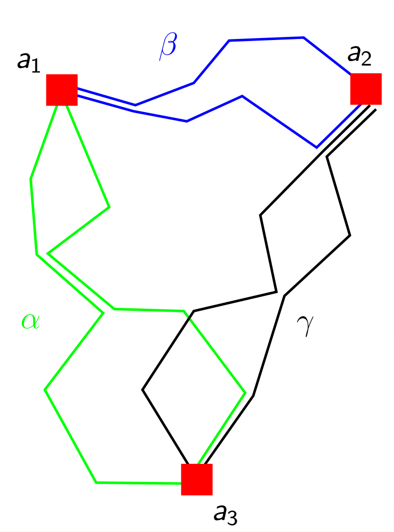
Corollary – Reduction to 3 Terminals

Enough to solve for three terminals.

3 Terminals

$$\alpha + \beta = \lambda_c(\{a_1\}, \{a_2, a_3\})$$
$$\beta + \gamma = \lambda_c(\{a_2\}, \{a_3, a_1\})$$
$$\gamma + \alpha = \lambda_c(\{a_3\}, \{a_2, a_1\})$$

We can compute α, β, γ!



There is a circulation z satisfying

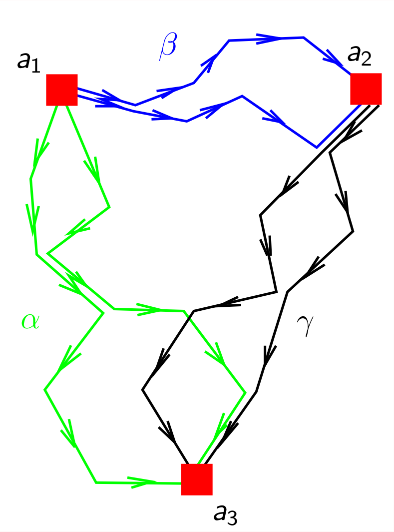
$excess_z(a_1) = \alpha + \beta$

$excess_z(a_2) = \gamma - \beta$

$excess_z(a_3) = -\alpha - \gamma$

$excess_z(v) = 0$ for $v \in V - A$.

We construct an integral z by a flow algorithm.



And there is a circulation y satisfying

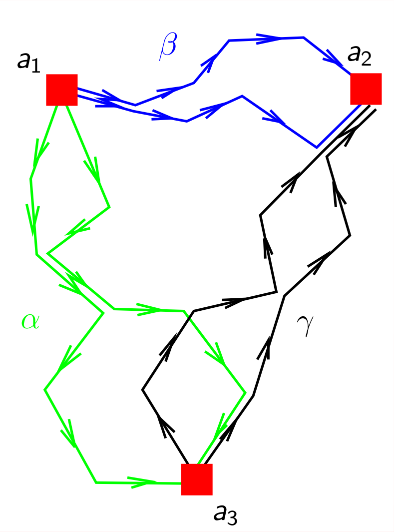
$excess_y(a_1) = \alpha + \beta$

$excess_y(a_2) = -\gamma - \beta$

$excess_y(a_3) = \gamma - \alpha$

$excess_y(v) = 0$
for $v \in V - A$.

We construct an integral y by a flow algorithm.



3 Terminals

Then $\frac{1}{2}(z+y)$ satisfies

$$excess_{\frac{1}{2}(z+y)}(a_1) = \alpha + \beta$$

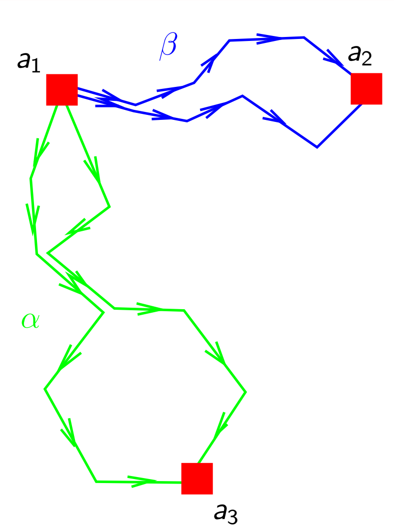
$$excess_{\frac{1}{2}(z+y)}(a_2) = -\beta$$

$$excess_{\frac{1}{2}(z+y)}(a_3) = -\alpha$$

$$excess_{\frac{1}{2}(z+y)}(v) = 0$$

for $v \in V - A$,

thus $\frac{1}{2}(z+y)$ is a flow of value $\alpha + \beta$ from a_1 to a_2, a_3.



Then $\frac{1}{2}(z-y)$ satisfies

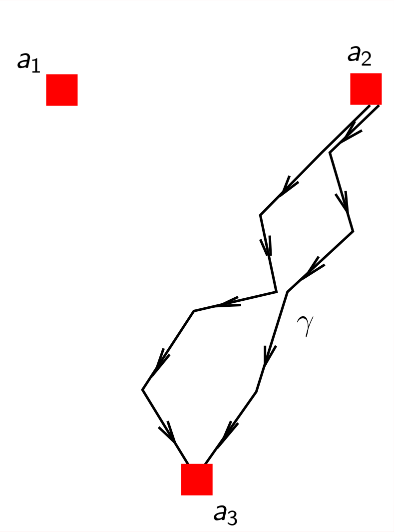
$excess_{\frac{1}{2}(z-y)}(a_1) = 0$

$excess_{\frac{1}{2}(z-y)}(a_2) = \gamma$

$excess_{\frac{1}{2}(z-y)}(a_3) = -\gamma$

$excess_{\frac{1}{2}(z-y)}(v) = 0$
for $v \in V - A$,

thus $\frac{1}{2}(z-y)$ is a flow of value γ from a_3 to a_2.



To show that the multiflow constructed from $\frac{1}{2}(z+y)$ and $\frac{1}{2}(z-y)$ satisfies the capacity constraint for an edge pq, we need that

$$|\tfrac{1}{2}(z+y)(e)| + |\tfrac{1}{2}(z-y)(e)| \leq c(e).$$

Proof: using the elemetary inequality for all $n, m \in \mathbb{R}$

$$|n+m| + |n-m| \leq \max\{2|n|, 2|m|\}$$

we get that

$$|\tfrac{1}{2}(z+y)(e)| + |\tfrac{1}{2}(z-y)(e)| \leq \max\{|y(e)|, |z(e)|\} \leq c(e).$$

Theorem (based on Ibaraki-Karzanov-Nagamochi)

There is a strongly polynomial time algorithm to find a maximum half-integral multiflow subject to edge-capacities.

LP for Node-Capacitated Multiflows (condensed)

Let $G = (V, E)$ be a graph, $A \subseteq V$ be the set of terminals, and let $c : V \to \mathbb{N}$ denote a vector of integral node-capacities.

$$\max \mathit{size}(x) \quad \text{subject to}$$

$$\begin{cases} x & \text{is a multiflow} \\ \text{total.load}_x(p) \leq c(p) & \text{for all } p \in V \end{cases}$$

.

LP for Node-Capacitated Multiflows

Let $G = (V, E)$ be a graph, $A \subseteq V$ be the set of terminals, and let $c : V \to \mathbb{N}$ denote a vector of integral node-capacities.

Moreover, let $A = \{a_1, a_2, \cdots, a_k\}$.

$$\max \sum_{i<j} x_{ij}(\delta^{in}(a_j)) \quad \text{subject to}$$

$$\begin{aligned}
x_{ij}(\delta^{in}(a_l)) &= 0 && \text{for } i < j, l \neq j \\
x_{ij}(\delta^{out}(a_l)) &= 0 && \text{for } i < j, l \neq i \\
x_{ij}(\delta^{in}(v)) - x_{ij}(\delta^{out}(v)) &= 0 && \text{for } i < j, v \in V - A \\
\sum_{i<j} x_{ij}(\delta^{in}(v)) &\leq c(v) && \text{for } v \in V - A \\
\sum_{i<j, i \neq l} x_{ij}(\delta^{in}(a_l)) + \sum_{l<j} x_{lj}(\delta^{out}(a_l)) &\leq c(a_l) && \text{for } 1 \leq l \leq k.
\end{aligned}$$

LP Dual for Node-Capacitated Multiflows

The LP dual, on variables $w : V \to \mathbb{R}$ is equivalent with:

$$\min \sum_{v \in V} c(v)\, w(v) \quad \text{subject to}$$

$$\begin{cases} w \geq 0 & \\ \displaystyle\sum_{v \in P} w(v) \geq 1 & \text{for every path } P \text{ connecting two distinct } a, b \in A \end{cases}$$

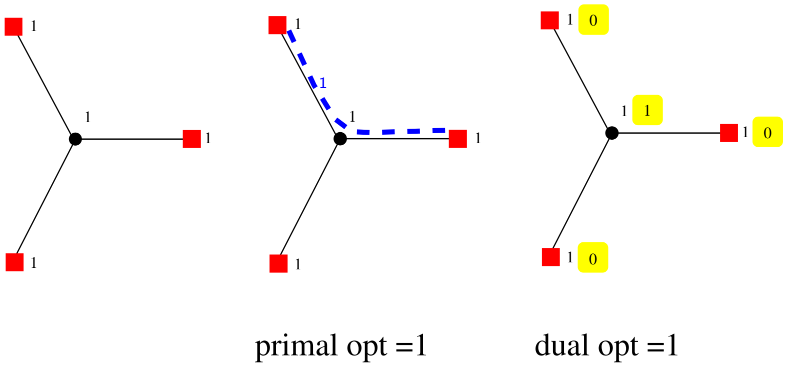
Node-Capacitated Multiflows

Theorem – Primal and Dual Half-Integrality

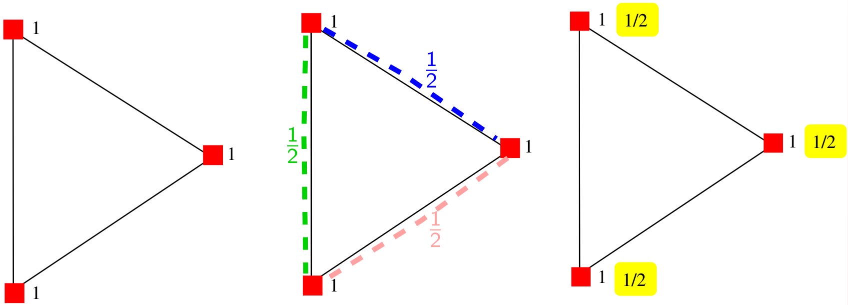
LP for maximum node-capacitated multiflow, and its dual both have a half-integral optimum.

— follows from Mader's Theorem on disjoint $\mathcal{A}$-paths
— for all weights $=1$, can be solved by Lovász' matroid matching in polynomial time

Example 1 (node-capacitated multiflow)



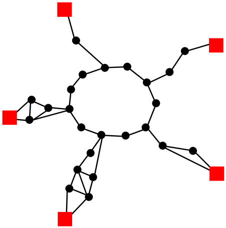
Example 2 (node-capacitated multiflow)

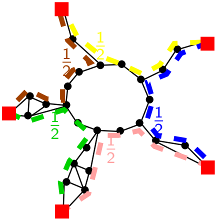


primal opt =1.5

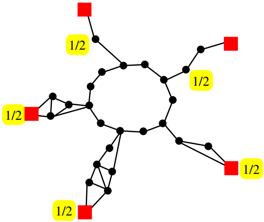
dual opt =1.5

Example 3 (node-capacitated multiflow)





primal opt =2,5



dual opt =2,5

Theorem – Dual Half-Integrality

Dual of the LP for maximum node-capacitated multiflow has a half-integral optimum.

Sketch Proof (Vazirani). Let $w : V \to \mathbb{R}_+$ be an LP dual optimum. For a node $v \in V$ we define w' as follows:

- If $w(v) = 1$, and there is a path P traversing v, connecting two nodes of A, such that $w(p) = 0$ for all $p \in P - v$, then let $w'(v) := 1$.
- Otherwise, if $w(v) > 0$, and there is a path R from a node in A to v such that $w(p) = 0$ for all $p \in R$, then let $w'(v) := 1/2$.
- Otherwise, let $w'(p) := 0$.

Consider a maximum multiflow x. Then x, w satisfy certain complementary slackness conditions:

- $w(P) = 1$ for every path P in the support of a flow in x.
- $w(v) > 0$ implies that the capacity of v is saturated.

Then one can show that x, w' also satisfy those complementary slackness conditions. This implies w' is a dual optimum.

Node-Capacitated Multiflows

Theorem – Primal Half-Integrality

LP for maximum node-capacitated multiflow has a half-integral optimum.

Theorem

Vertices of the shortest maximum multiflow polytope are half-integral.

This Lemma – by Frank and Tardos – also implies a strongly polynomial time algorithm!

Node-Capacitated Multiflows

LP for **maximum multiflows**:

$$M := \max \mathit{size}(x) \quad \text{subject to}$$

$$\begin{cases} x & \text{is a multiflow} \\ \text{total.load}_x(p) \le c(p) & \text{for all } p \in V \end{cases}$$

LP for **shortest maximum multiflows**:

$$L := \min \mathbf{1} \cdot x \quad \text{subject to}$$

$$\begin{cases} x & \text{is a multiflow} \\ \text{total.load}_x(p) \le c(p) & \text{for all } p \in V \\ \mathit{size}(x) = M & \end{cases}$$

Node-Capacitated Multiflows

Polytope of shortest maximum multiflows:

$$\begin{cases} x & \text{is a multiflow} \\ \text{total.load}_x(p) \leq c(p) & \text{for all } p \in V \\ \mathit{size}(x) = M & \\ \mathbf{1} \cdot x = L & \end{cases}$$

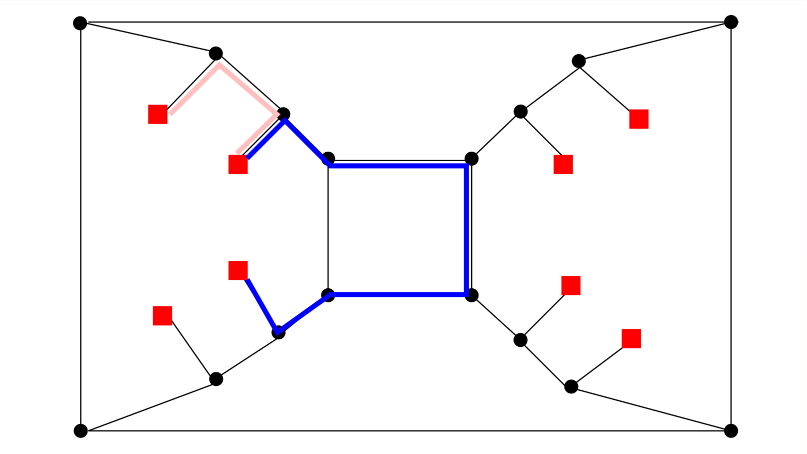
Theorem

Vertices of the shortest maximum multiflow polytope are half-integral.

Corollary

Strongly polynomial time algorithm to find a maximum half-integral node-capacitated multiflow.

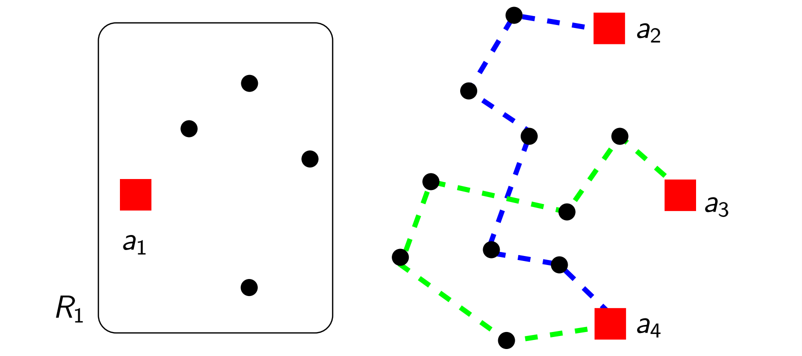
Example with all-one edge-capacities to show that the polytope of shortest maximum multiflows is half-integral, but the polytope of maximum multiflows is not:



Proof Sketch.

1. W.l.o.g. delete nodes v not traversed by any s.m.m.

2. Let $R_i \subseteq V$ be the "region", i.e. the set of nodes $v \in V$ such that there is no s.m.m. x and $r, q \neq i$ such that $x_{rq}(\delta(v)) > 0$.

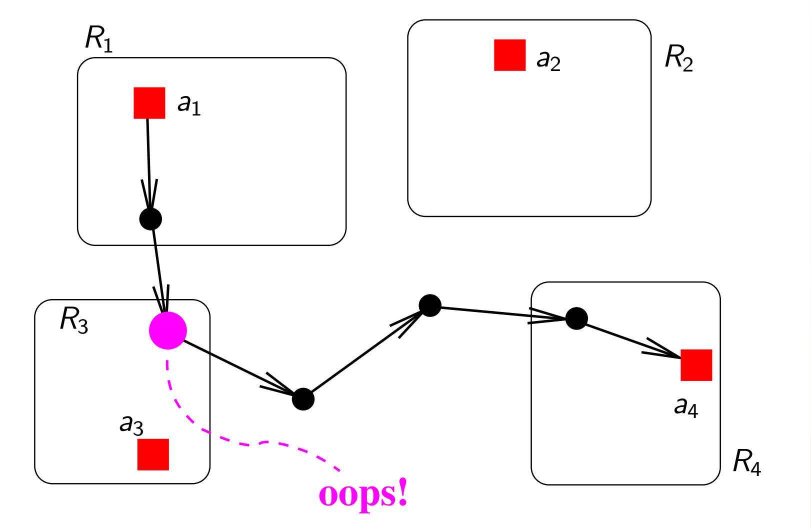


3. Then every s.m.m. x satisfies the following "region constraints":

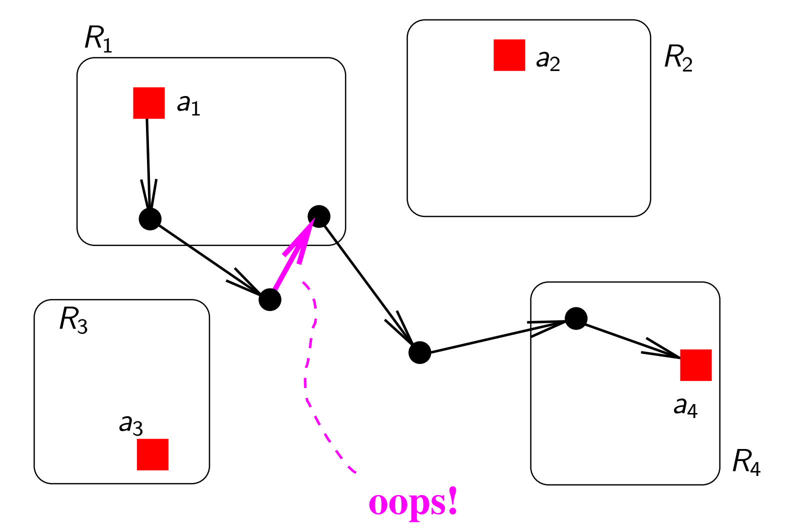
Region Constraints

$$x_{ij}(vz) = 0 \qquad \text{if } vz \in E,\ v \in V - R_i,\ z \in V - R_j.$$

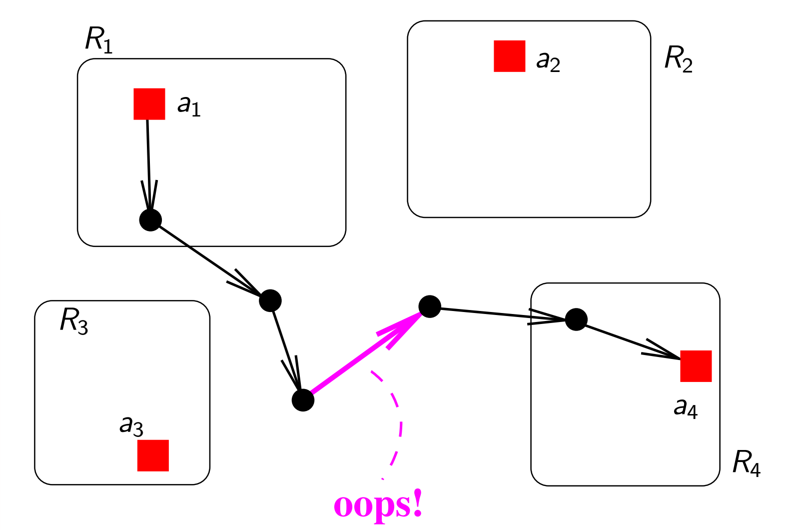
By the definition of the regions, a shortest maximum multiflow MUST NOT contain the following path with positive weight:



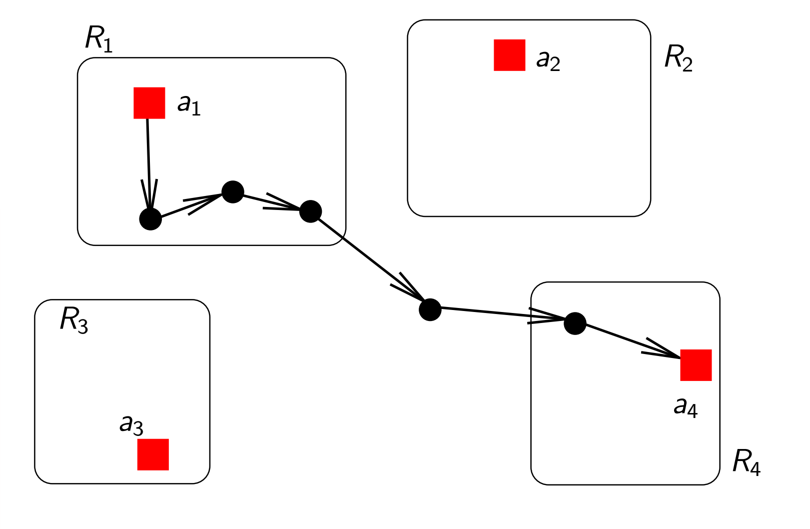
The region-constraint says that a shortest maximum multiflow MUST NOT contain the following path with positive weight:



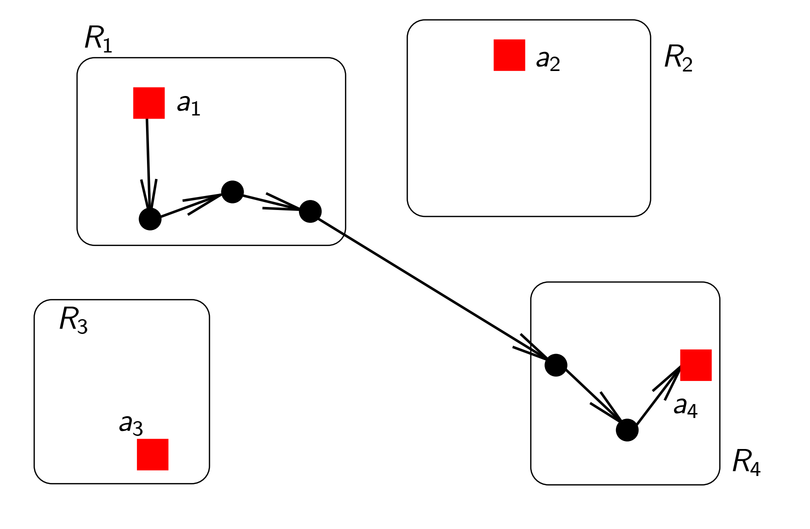
The region-constraint says that a shortest maximum multiflow MUST NOT contain the following path with positive weight:



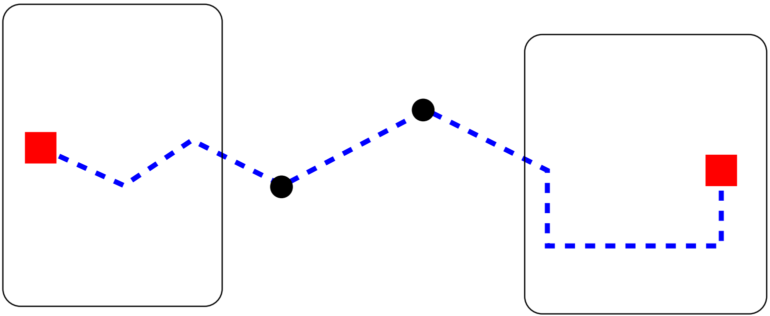
A shortest maximum multiflow MAY contain the following path with positive weight:



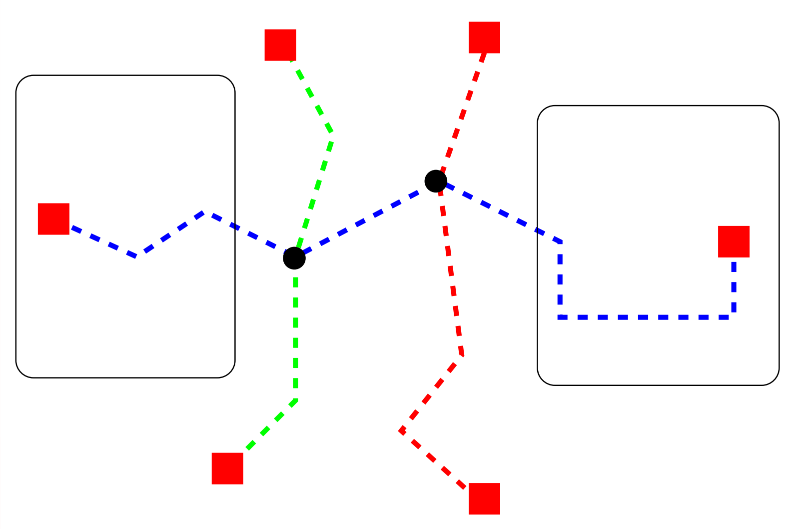
A shortest maximum multiflow MAY contain the following path with positive weight:



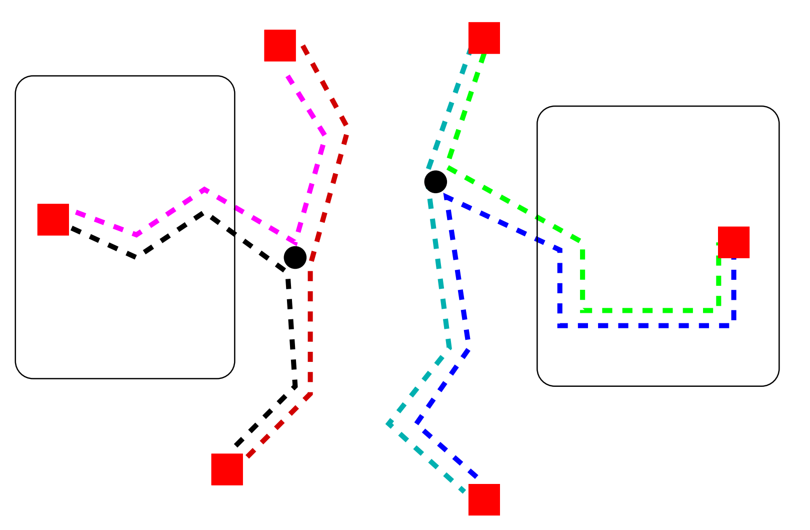
Suppose – for contradiction – that s.m.m. x contains this path with weight $\varepsilon > 0$:



Then, by definition, s.m.m. x' contains the following configuration with weight $\varepsilon' > 0$:



We replace that configuration with weight ε' by the following with weight $\varepsilon'/2$: (contradicting x being "shortest")



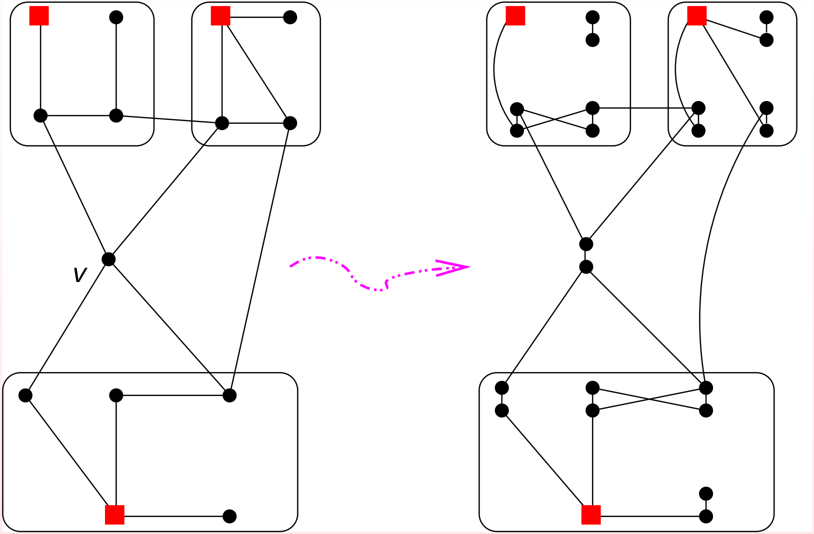
... and a similar argument shows the other case of the region constraint.

4. Thus from this point on, we can argue based on the region-constraints.

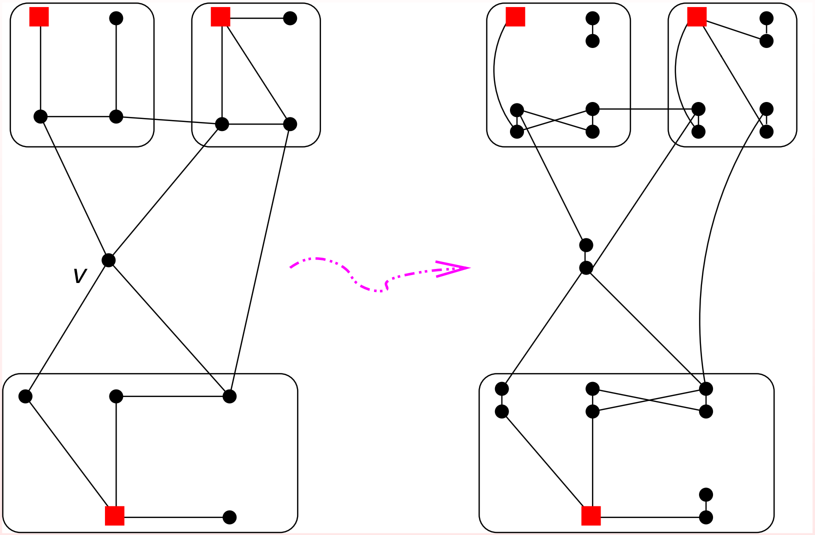
5. Let x be an extreme point of the shortest max multiflow polytope.

6. Based on x, we construct an auxiliary graph G' with node-weights b', and consider b'-matchings in G'. The construction goes as follows.

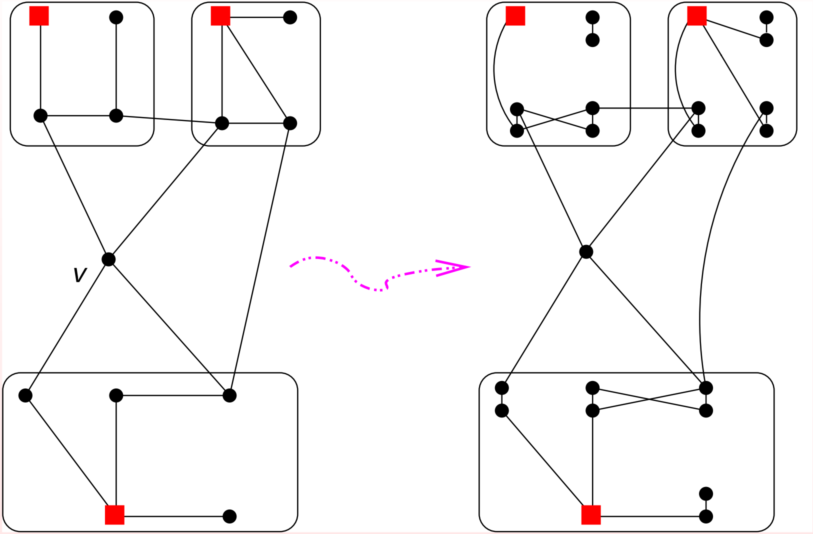
If all flow through v comes from a_3:



If all flow through v comes from a_2:



Otherwise:



7. Every b'-matching in G' can be converted into a maximum multiflow in G, A, c.

8. If x is not half-integral, then x' is not half-integral, and then x' is not a vertex of the b'-matching polytope, and then x' is the convex combination of two b'-matchings x'' and x''', which can be converted into maximum multiflows x'''' and x''''', implying that x is the convex combination of x'''' and x''''', which contradicts that x is a vertex.

This implies that y is a vertex of the following polytope:

$$\begin{cases} x & \text{is a multiflow} \\ \text{total.load}_x(p) \leq c(p) & \text{for all } p \in V \\ size(x) = M & \\ \mathbf{1} \cdot x = L & \\ x_{ab}(pq) = 0 & \text{for } a, b \in A, pq \in E, p \in V - R_a, q \in V - R_b \\ total.load_x(q) = c(q) & \text{for all } q \in Q. \end{cases}$$

Claim

This polytope is the projection of the fractional b-matching polytope of an auxiliary graph.

Lemma

Vertices of the shortest maximum multiflow polytope are half-integral.

Concluding Remarks

- Integral multiflow problems solved by Mader's min-max (1978).
- A polynomial time algorithm to find a maximum integral multiflow subject to all-one node-capacities follows from the linear matroid matching of Lovász (1981) and Schrijver's representation (2001).
- Weakly polynomial time algorithm to find a maximum integral multiflow subject to edge-capacities by Keijsper, Pendavingh, Stougie (2006)
-
- Strongly polynomial time algorithm to find a maximum integral multiflow with node-capacities by combining P (2007) and P (2008).
- Similar results are expected for group-labeled graphs of Chudnovsky, et al.
- Open Question: Improve running time, simplify, get rid of LP?
- Question: Bounded fractionality of **weighted** node-capacitated multiflow problems?