

Algorithms for multiflows

Gyula Pap

Department of Computer Science, Cornell University

September 29, 2008, Ithaca

Definition of a multiflow

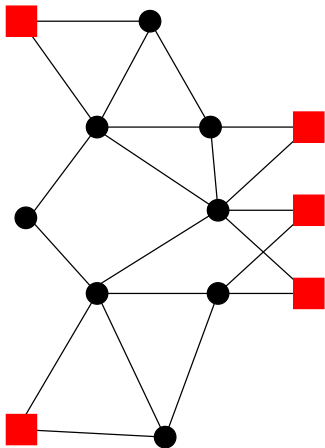
- We are given an undirected graph $G = (V, E)$ and a set of **terminals** $A \subseteq V$.

- A multiflow, denoted by

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

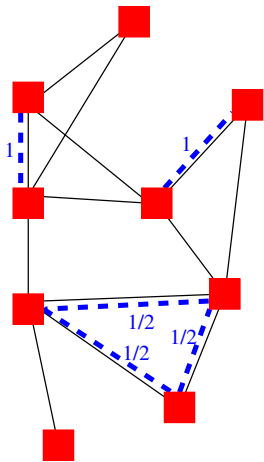
is a set of $\binom{|A|}{2}$ flows x_{ab} between all pairs of terminals $a \neq b, a, b \in A$.

- $size(x) := \sum size(x_{ab})$
- Problem: maximize the size of a multiflow subject to certain capacity and/or integrality constraints.

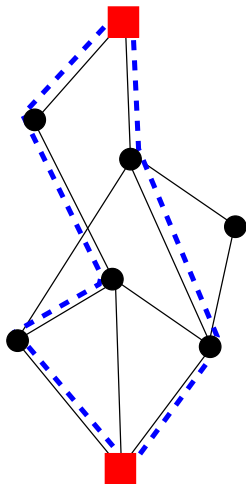


Special Cases

$A = V$:
max (fractional) matching

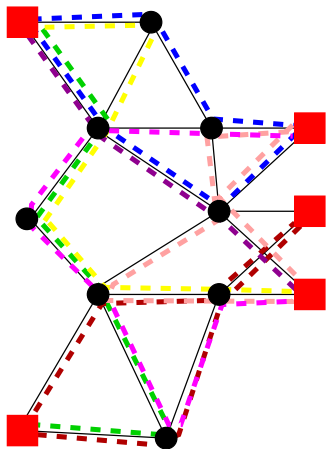


$|A| = 2$:
max s - t flow



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.



- Research done at the Egerváry Research Group, Eötvös University, Budapest, Hungary, www.cs.elte.hu/egres
- Edge-Capacitated Multiflows: Lovász-Cherkassky 1972, half-integrality of the LP, Ibaraki-Karzanov-Nagamochi 1998
- Example: LP may have a "very fractional" extreme optimum.
- Node-Capacitated Multiflows: half-integrality of the LP, a strongly polynomial time algorithm
- Useful in the construction of a strongly polynomial time algorithm: Frank and Tardos' 1986 paper showing that – with the help of the ellipsoid method – a 0,1 LP is solvable in strongly polynomial time

All-one edge-capacities, Eulerian graph

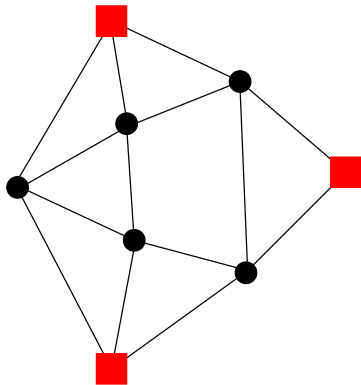
Lovász, Cherkassky, ≈ 1972

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

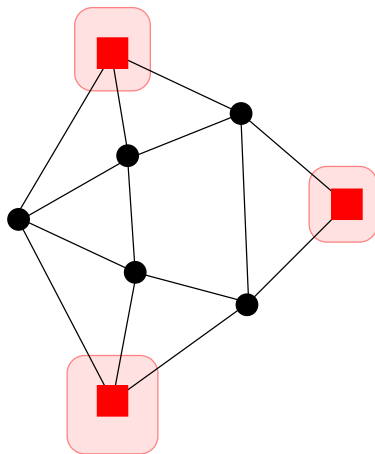
Lovász, Cherkassky, ≈ 1972

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

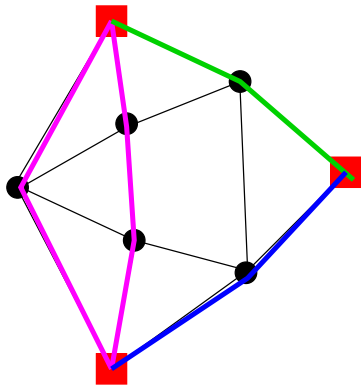
Lovász, Cherkassky, ≈ 1972

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

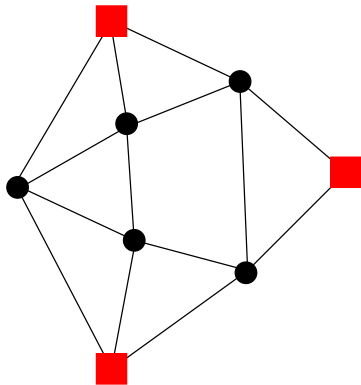
Lovász, Cherkassky, ≈ 1972

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

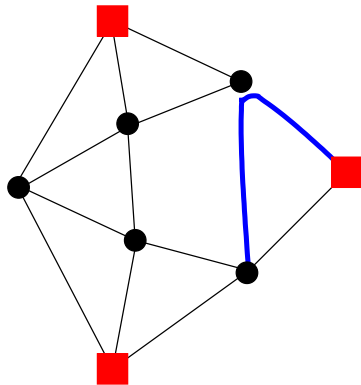
Lovász, Cherkassky, ≈ 1972

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.



All-one edge-capacities, Eulerian graph

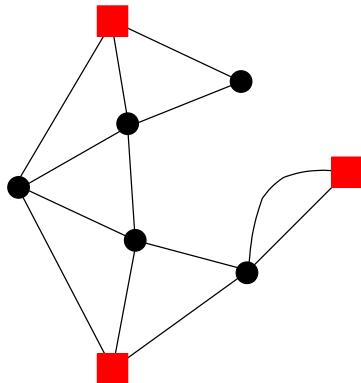
Lovász, Cherkassky, ≈ 1972

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

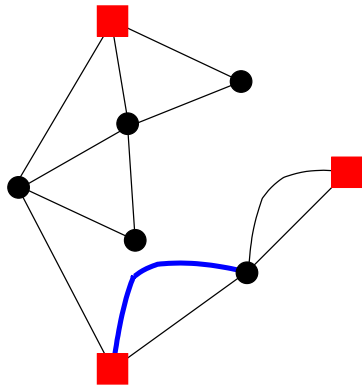
Lovász, Cherkassky, ≈ 1972

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

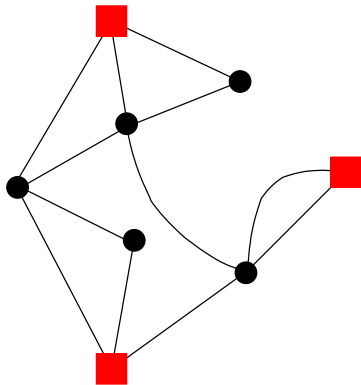
Lovász, Cherkassky, ≈ 1972

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

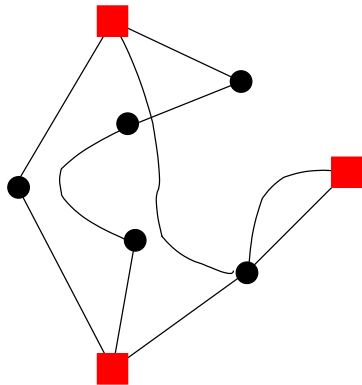
Lovász, Cherkassky, ≈ 1972

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

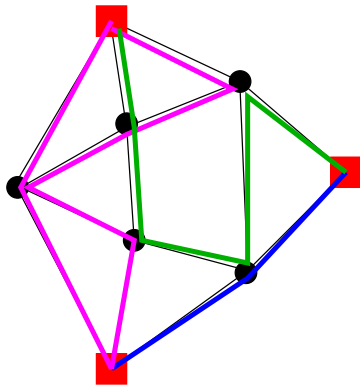
Lovász, Cherkassky, ≈ 1972

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.



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

max $size(x)$ subject to

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

LP of Edge-Capacitated Multiflows

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

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

$$x \geq 0$$

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

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

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

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

Corollary – Half-Integrality

For any $G = (V, E)$, $A \subseteq V$, $c : E \rightarrow \mathbb{N}$, 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 to find a half-integral optimum?
- Possible approach: 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)

Determining 2^k (in decimals)

700000000000=1010001011111011010000000101100000000000

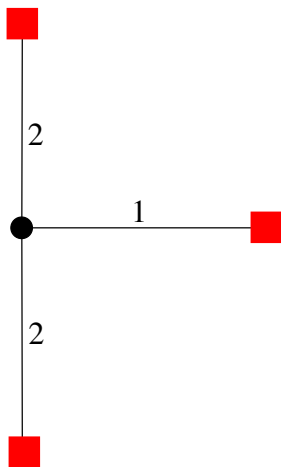
implying that

$$2^{700000000000} =$$

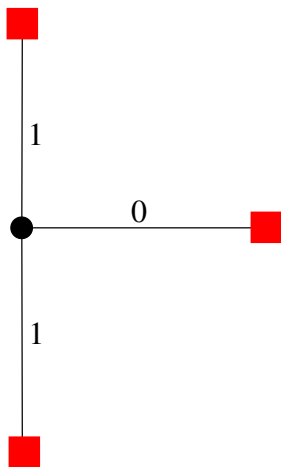
$$= 2^{2^{11}} \cdot 2^{2^{12}} \cdot 2^{2^{14}} \cdot 2^{2^{22}} \cdot 2^{2^{24}} \cdot 2^{2^{25}} \cdot 2^{2^{27}} \cdot 2^{2^{28}} \cdot 2^{2^{29}} \cdot 2^{2^{30}} \cdot 2^{2^{31}} \cdot 2^{2^{33}} \cdot 2^{2^{37}} \cdot 2^{2^{39}}$$

requiring a total of 375 arithmetic operations.

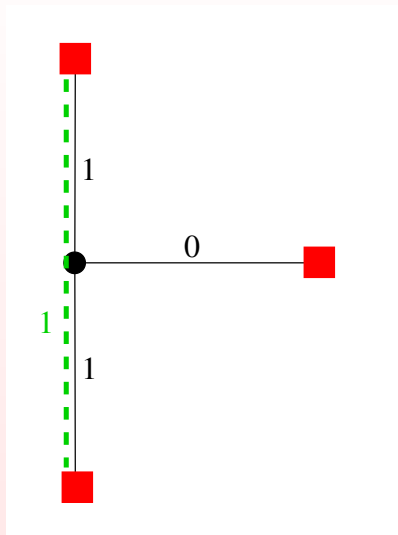
Capacity Scaling for Half-Integral Edge-Capacitated Multiflows



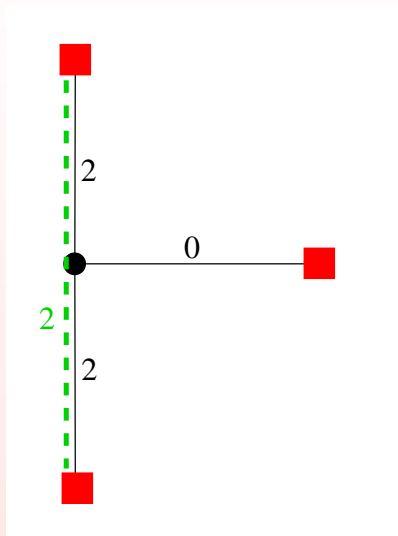
Capacity Scaling for Half-Integral Edge-Capacitated Multiflows



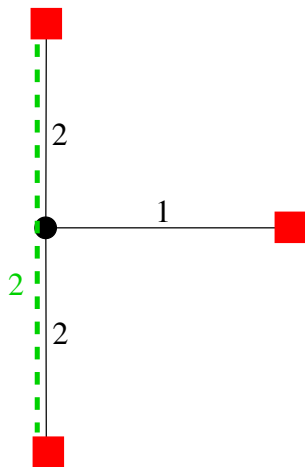
Capacity Scaling for Half-Integral Edge-Capacitated Multiflows



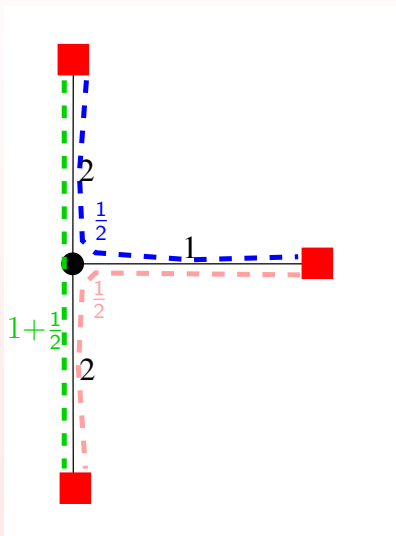
Capacity Scaling for Half-Integral Edge-Capacitated Multiflows



Capacity Scaling for Half-Integral Edge-Capacitated Multiflows



Capacity Scaling for Half-Integral Edge-Capacitated Multiflows



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, achieving a running time of

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

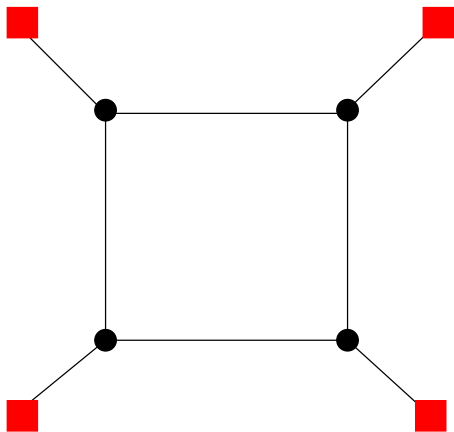
where U denotes the maximum capacity of a node.

Half-Integral Edge-Capacitated Multiflows

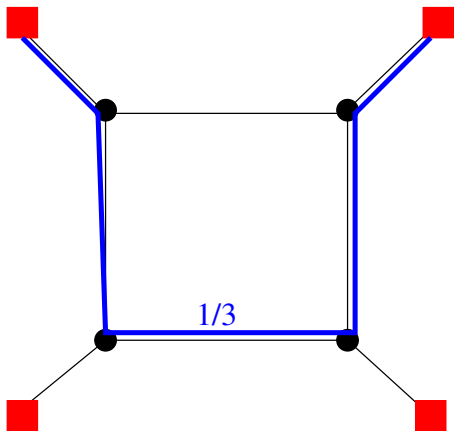
Possible approach to achieve strongly polynomial time:

- 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.

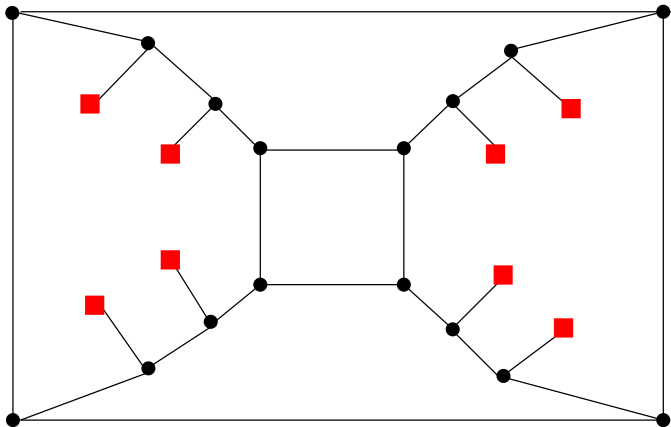
The polytope of multiflows can have a fractional extreme point:



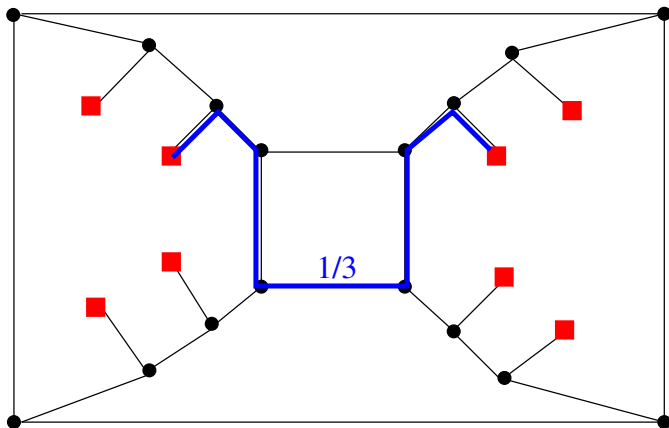
The polytope of multiflows can have a fractional extreme point:



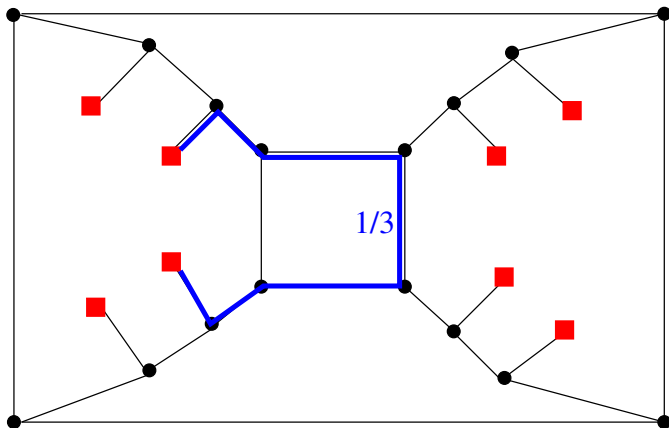
The polytope of maximum multiflows can have a fractional extreme point:



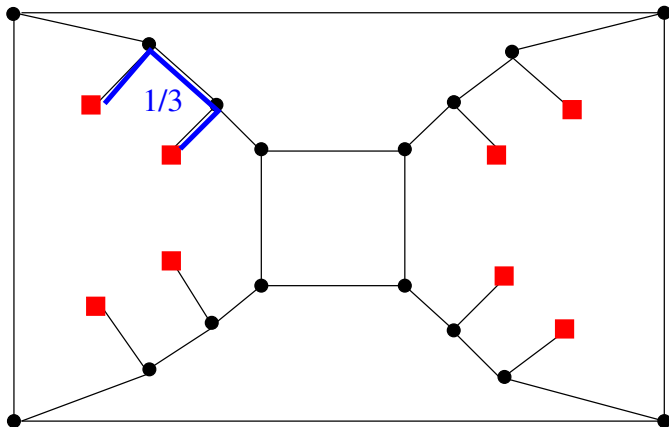
The polytope of maximum multiflows can have a fractional extreme point:



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



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

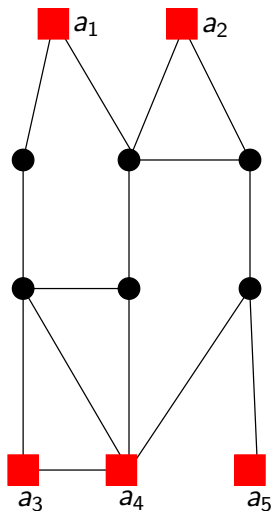


Half-Integral Edge-Capacitated Multiflows

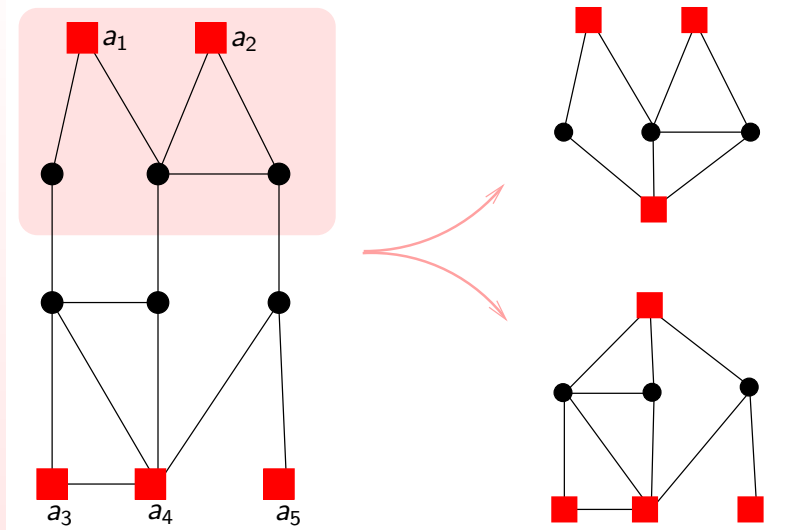
A strongly polynomial time algorithm motivated by an idea of Karzanov, Ibaraki, Nagamochi (1998):

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

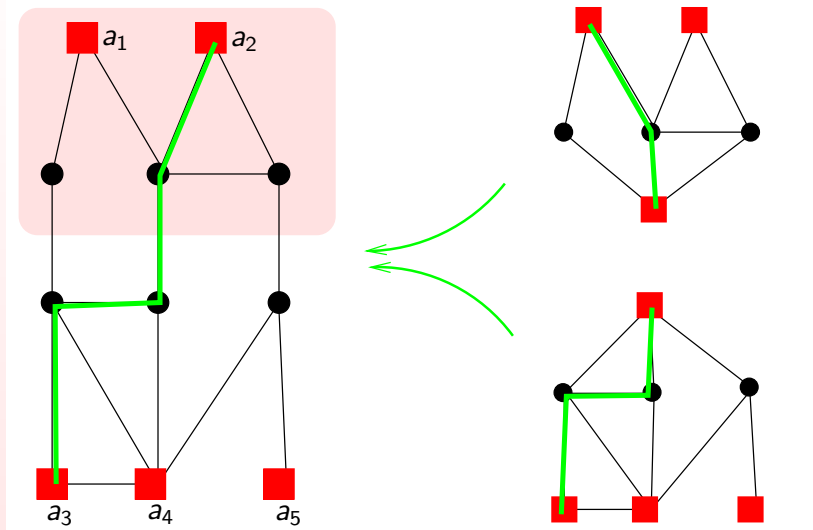
Decompose by $\{a_1, a_2\} - \{a_3, \dots, a_k\}$ cuts



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



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



Corollary – Reduction to 3 Terminals

Enough to solve for three terminals.

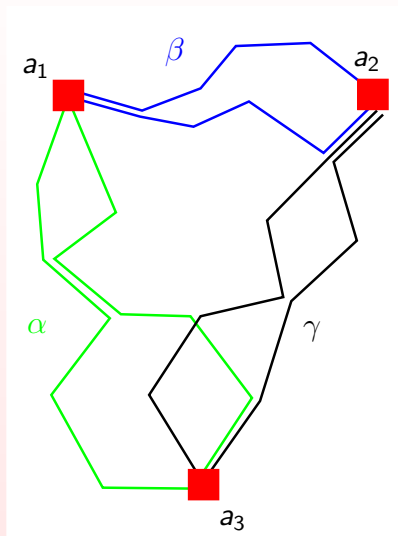
3 Terminals

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

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

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

We can compute α, β, γ !



3 Terminals

There is a circulation z satisfying

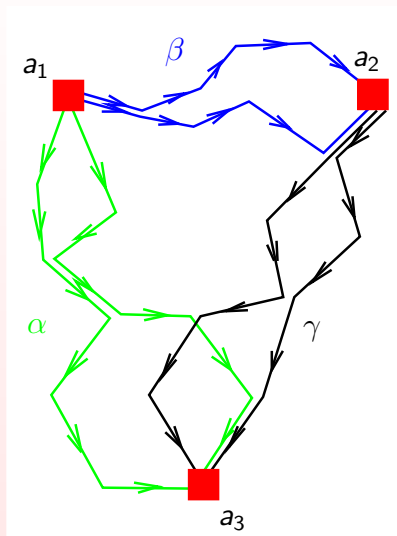
$$\text{excess}_z(a_1) = \alpha + \beta$$

$$\text{excess}_z(a_2) = \gamma - \beta$$

$$\text{excess}_z(a_3) = -\alpha - \gamma$$

$$\text{excess}_z(v) = 0 \quad \text{for } v \in V - A.$$

We construct an integral z by a flow algorithm.



3 Terminals

And there is a circulation y satisfying

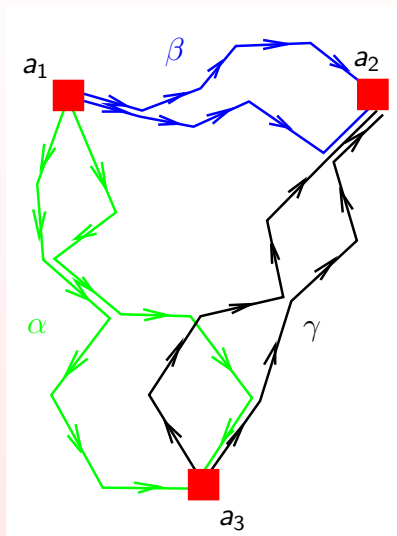
$$\text{excess}_y(a_1) = \alpha + \beta$$

$$\text{excess}_y(a_2) = -\gamma - \beta$$

$$\text{excess}_y(a_3) = \gamma - \alpha$$

$$\text{excess}_y(v) = 0 \quad \text{for } v \in V - A.$$

We construct an integral y by a flow algorithm.



3 Terminals

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

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

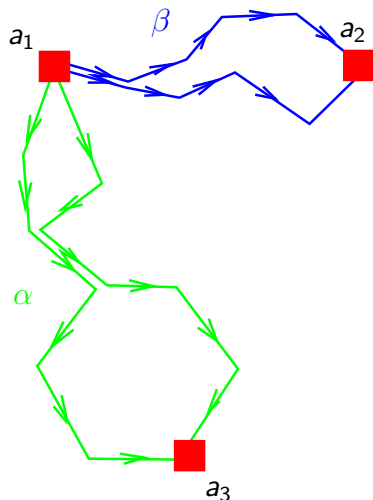
$$\text{excess}_{\frac{1}{2}(z+y)}(a_2) = -\beta$$

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

$$\text{excess}_{\frac{1}{2}(z+y)}(v) = 0$$

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 .



3 Terminals

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

$$\text{excess}_{\frac{1}{2}(z-y)}(a_1) = 0$$

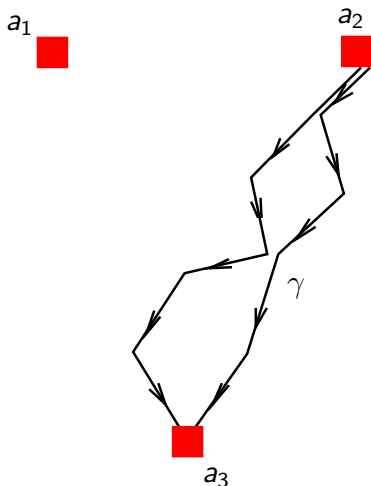
$$\text{excess}_{\frac{1}{2}(z-y)}(a_2) = \gamma$$

$$\text{excess}_{\frac{1}{2}(z-y)}(a_3) = -\gamma$$

$$\text{excess}_{\frac{1}{2}(z-y)}(v) = 0$$

for $v \in V - A$,

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



3 Terminals

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

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

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

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

we get that

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

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 $c : V \rightarrow \mathbb{N}$ denote a vector of integral node-capacities.

max $size(x)$ subject to

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

LP for Node-Capacitated Multiflows

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

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

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

$$x_{ij}(\delta^{in}(a_l)) = 0 \quad \text{for } i < j, l \neq j$$

$$x_{ij}(\delta^{out}(a_l)) = 0 \quad \text{for } i < j, l \neq i$$

$$x_{ij}(\delta^{in}(v)) - x_{ij}(\delta^{out}(v)) = 0 \quad \text{for } i < j, v \in V - A$$

$$\sum_{i < j} x_{ij}(\delta^{in}(v)) \leq c(v) \quad \text{for } v \in V - A$$

$$\sum_{i < j, i \neq l} x_{ij}(\delta^{in}(a_l)) + \sum_{l < j} x_{lj}(\delta^{out}(a_l)) \leq c(a_l) \quad \text{for } 1 \leq l \leq k.$$

LP Dual for Node-Capacitated Multiflows

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

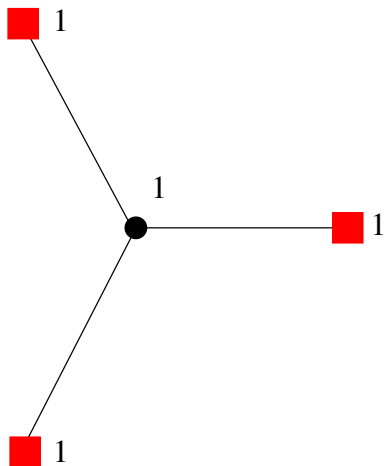
$$\min \sum_{v \in V} c(v) w(v) \quad \text{subject to}$$

$$\left\{ \begin{array}{l} w \geq 0 \\ \sum_{v \in P} w(v) \geq 1 \end{array} \right. \quad \text{for every path } P \text{ connecting two distinct } a, b \in A$$

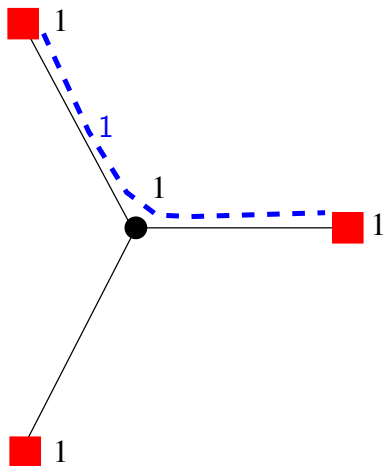
Theorem – Primal and Dual Half-Integrality

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

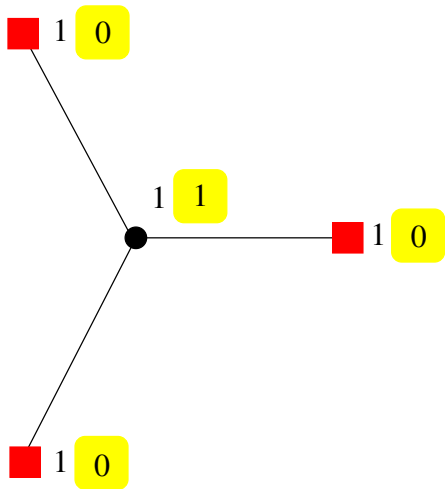
Example 1 – Node-Capacitated Multiflows



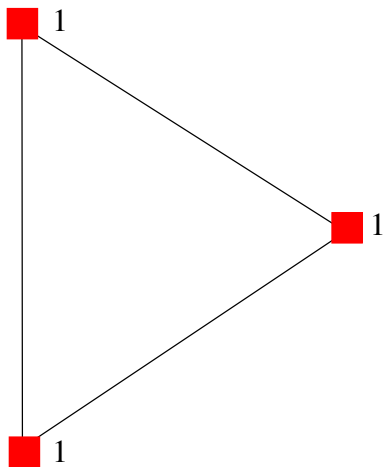
Example 1 – Node-Capacitated Multiflows



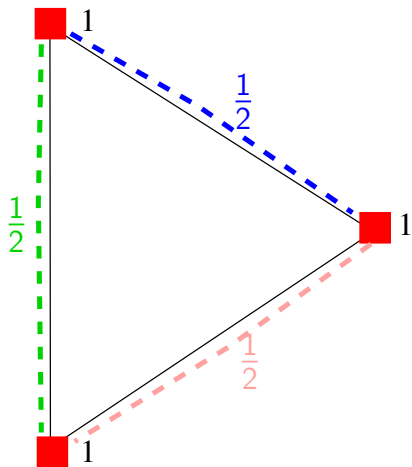
Example 1 – Node-Capacitated Multiflows



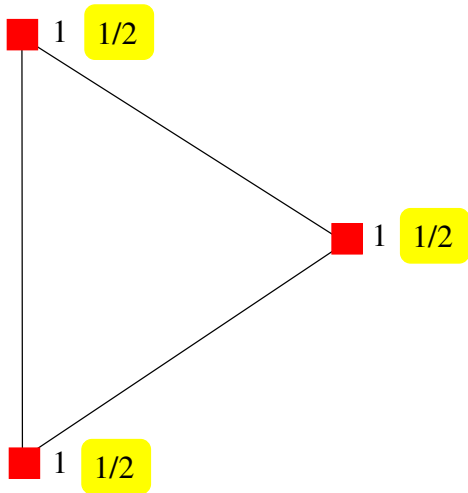
Example 2 – Node-Capacitated Multiflows



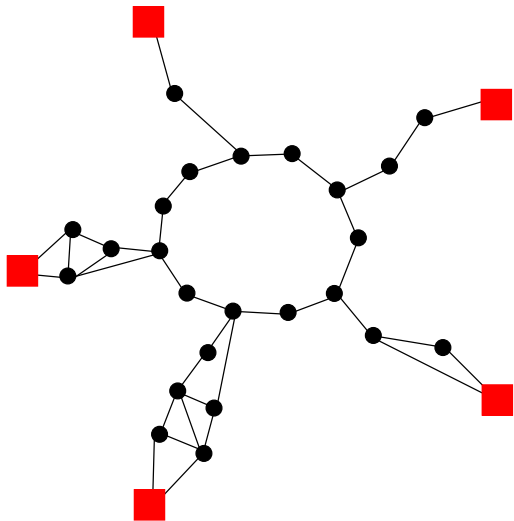
Example 2 – Node-Capacitated Multiflows



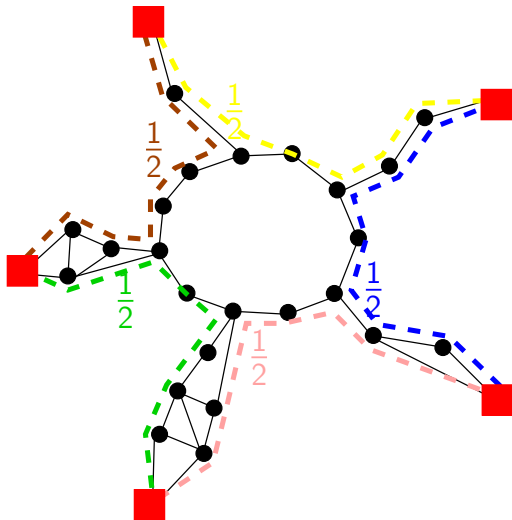
Example 2 – Node-Capacitated Multiflows



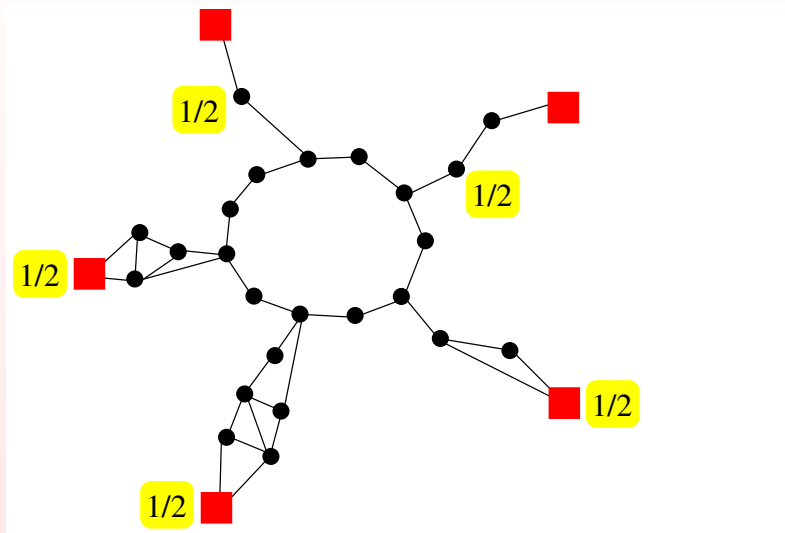
Example 3 – Node-Capacitated Multiflows



Example 3 – Node-Capacitated Multiflows



Example 3 – Node-Capacitated Multiflows



Theorem – Dual Half-Integrality

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

Sketch Proof (Vazirani). Let $w : V \rightarrow \mathbb{R}_+$ 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.

Theorem – Primal Half-Integrality

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

Lemma

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

This Lemma also implies a strongly polynomial time algorithm!

Node-Capacitated Multiflows

LP for **maximum multiflows**:

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

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

LP for **shortest maximum multiflows**:

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

$$\left\{ \begin{array}{ll} x & \text{is a multiflow} \\ \text{total.load}_x(p) \leq c(p) & \text{for all } p \in V \\ \text{size}(x) = M & \end{array} \right.$$

Polytope of shortest maximum multiflows:

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

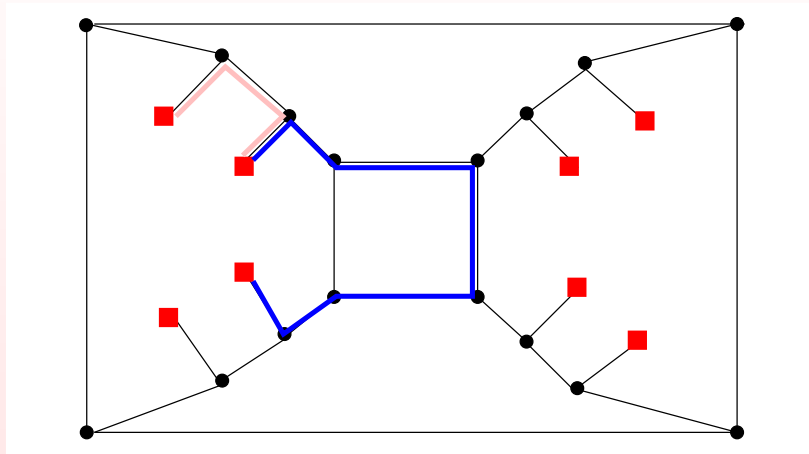
Lemma

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:



Lemma

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

Proof Sketch.

Let y be a vertex of the shortest maximum multiflow polytope. One can show that for all $a \in A$ there are certain disjoint regions R_a such that $a \in R_a$, and so that, with the notation $R := \cup R_a$ and $Q := V - R$, both of the following conditions hold:

- 1 $y_{ab}(pq) = 0$ for all $a, b \in A, pq \in E, p \in V - R_a, q \in V - R_b$
- 2 $total.load_y(q) = c(q)$ for all $q \in Q$.

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

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

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).
- Open Questions: Improve running time, simplify, get rid of LP?
- Group-Labeled Graphs (Chudnovsky, Geelen, Gerards, Goddyn, Lohman, Seymour, 2005)