Constrained Flows in Networks

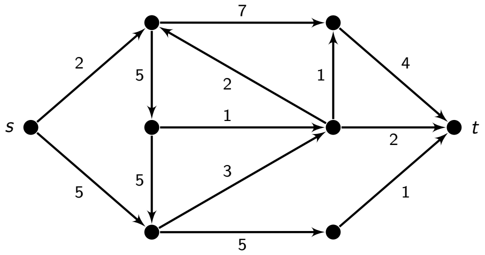
Jørgen Bang-Jensen[1], Stéphane Bessy[2], Lucas Picasarri-Arrieta[3]

[1] University of southern Denmark, Denmark
[2] Université de Montpellier, LIRMM, France
[3] Université Côte d'Azur, France

Networks

A **Network** is a quadruplet $\mathcal{N} = (D, s, t, c)$ where:

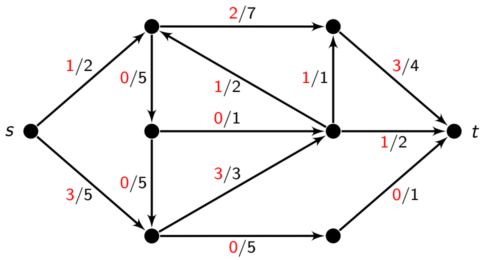
- $D = (V, A)$ is a **digraph**,
- $s \in V$ is a **source**,
- $t \in V$ is a **sink**, and
- $c : A \to \mathbb{N}$ is a **capacity function**.



Flows in networks

In a network $\mathcal{N} = (D = (V, A), s, t, c)$, a **flow** is a function $x : A \to \mathbb{N}$ such that:

- $\forall uv \in A, \quad f(uv) \leq c(uv)$, and
- $\forall v \in V \setminus \{s, t\}, \displaystyle\sum_{u \in N^-(v)} f(uv) = \sum_{w \in N^+(v)} f(vw)$.

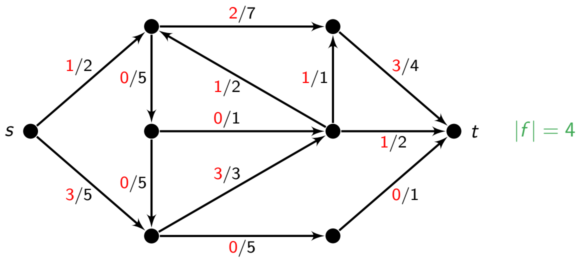


Flows in networks

In a network $\mathcal{N} = (D = (V, A), s, t, c)$, a **flow** is a function $x : A \to \mathbb{N}$ such that:

- $\forall uv \in A, \quad f(uv) \leq c(uv)$, and
- $\forall v \in V \setminus \{s, t\}, \displaystyle\sum_{u \in N^-(v)} f(uv) = \sum_{w \in N^+(v)} f(vw)$.

The **value** $|f|$: amount of flow leaving s (= entering t).

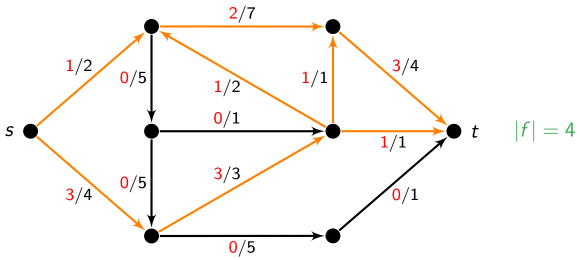


Flows in networks

In a network $\mathcal{N} = (D = (V, A), s, t, c)$, a **flow** is a function $x : A \to \mathbb{N}$ such that:

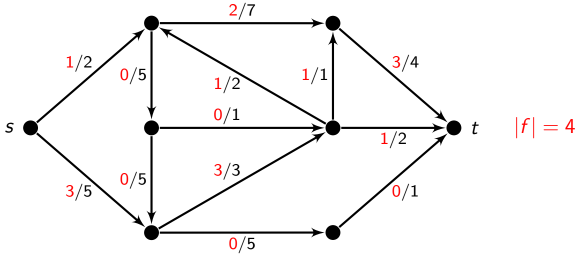
- $\forall uv \in A, \quad f(uv) \leq c(uv)$, and
- $\forall v \in V \setminus \{s, t\}, \displaystyle\sum_{u \in N^-(v)} f(uv) = \sum_{w \in N^+(v)} f(vw)$.

The **value** $|f|$: amount of flow leaving s (= entering t).
The **support** D_f : subdigraph of D with the arcs uv s.t. $f(uv) \geq 1$.



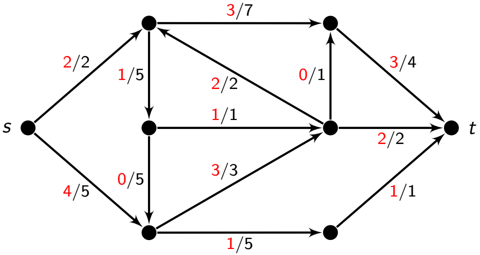
Max-flow Min-cut theorem

A **maximum flow** f^* is a flow with maximum value $|f^*|$.



Max-flow Min-cut theorem

A **maximum flow** f^* is a flow with maximum value $|f^*|$.

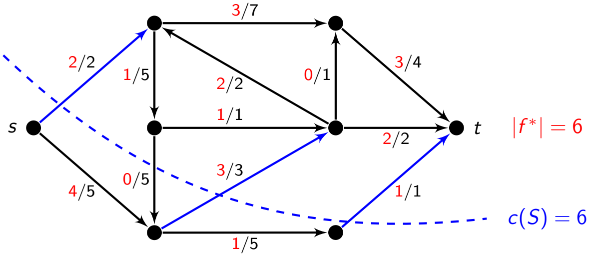


$|f^*| = 6$

Max-flow Min-cut theorem

A **maximum flow** f^* is a flow with maximum value $|f^*|$.

The value of a **maximum flow** is equal to the capacity of a **minimum cut** (Ford and Fulkerson 1962), and it can be computed in **polynomial time** (Edmonds and Karp 1972).



Constrained Flows

Given a property $\mathcal{P}$ on flows, we can consider the following problem.

$\mathcal{P}$-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$ and an integer ℓ

Question : Does there exist a flow $f \in \mathcal{P}$ such that $|f| \geq \ell$?

Our contributions:

- $f \in \mathcal{P}$ iff D_f has bounded out-degree,
- $f \in \mathcal{P}$ iff D_f is highly connected,
- $f \in \mathcal{P}$ iff it is persistent (*i.e.* $\mathcal{N}_f = (D_f, s, t, c_f)$ has a large flow),
- $f \in \mathcal{P}$ iff it is decomposable into few path-flows, and
- $f \in \mathcal{P}$ iff each arc belongs to few path-flows.

Constrained Flows

Given a property $\mathcal{P}$ on flows, we can consider the following problem.

$\mathcal{P}$-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$ and an integer ℓ

Question : Does there exist a flow $f \in \mathcal{P}$ such that $|f| \geq \ell$?

Our contributions:

- $f \in \mathcal{P}$ iff D_f has bounded out-degree,
- $f \in \mathcal{P}$ iff D_f is highly connected,
- $f \in \mathcal{P}$ iff it is persistent (*i.e.* $\mathcal{N}_f = (D_f, s, t, c_f)$ has a large flow),
- $f \in \mathcal{P}$ iff it is decomposable into few path-flows, and
- $f \in \mathcal{P}$ iff each arc belongs to few path-flows.

Constrained Flows

Given a property $\mathcal{P}$ on flows, we can consider the following problem.

$\mathcal{P}$-MAXIMUM-FLOW

Input	:	A network $\mathcal{N} = (D, s, t, c)$ and an integer ℓ
Question	:	Does there exist a flow $f \in \mathcal{P}$ such that $\|f\| \geq \ell$?

Our contributions:

- $f \in \mathcal{P}$ iff D_f has bounded out-degree,
- $f \in \mathcal{P}$ iff D_f is highly connected,
- $f \in \mathcal{P}$ iff it is persistent (*i.e.* $\mathcal{N}_f = (D_f, s, t, c_f)$ has a large flow),
- $f \in \mathcal{P}$ iff it is decomposable into few path-flows, and
- $f \in \mathcal{P}$ iff each arc belongs to few path-flows.

An easy example: acyclic flows

Theorem

Given a network $\mathcal{N}$, for every flow f there exists a flow f' s.t. $|f'| = |f|$ and $D_{f'}$ is acyclic.

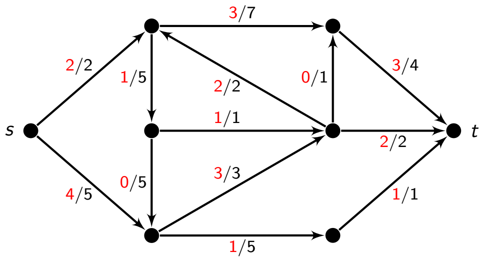
Proof: Every flow f decomposes into **path-flows** and **cycle-flows**. Remove the cycle-flows to obtain f'.

An easy example: acyclic flows

Theorem

Given a network $\mathcal{N}$, for every flow f there exists a flow f' s.t. $|f'| = |f|$ and $D_{f'}$ is acyclic.

Proof: Every flow f decomposes into **path-flows** and **cycle-flows**. Remove the cycle-flows to obtain f'.



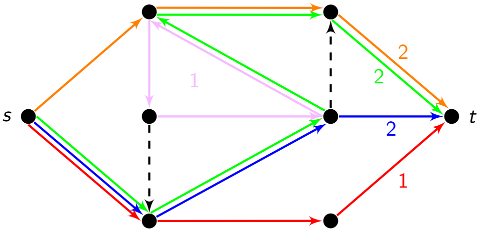
$\square$

An easy example: acyclic flows

Theorem

Given a network $\mathcal{N}$, for every flow f there exists a flow f' s.t. $|f'| = |f|$ and $D_{f'}$ is acyclic.

Proof: Every flow f decomposes into **path-flows** and **cycle-flows**. Remove the cycle-flows to obtain f'.



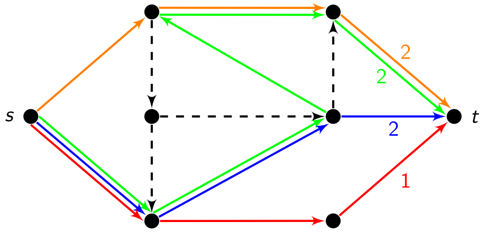
□

An easy example: acyclic flows

Theorem

Given a network $\mathcal{N}$, for every flow f there exists a flow f' s.t. $|f'| = |f|$ and $D_{f'}$ is acyclic.

Proof: Every flow f decomposes into **path-flows** and **cycle-flows**. Remove the cycle-flows to obtain f'.



□

Degree constrained flows

$(\Delta^+ \leq k)$-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$ and an integer ℓ

Question : Does there exist a flow f such that $\Delta^+(D_f) \leq k$ and $|f| \geq \ell$?

Trivial when $k = 1$.

Theorem

For every fixed $k \geq 2$, $(\Delta^+ \leq k)$-Maximum-Flow is NP-complete even when restricted to acyclic networks.

Degree constrained flows

$(\Delta^+ \leq k)$-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$ and an integer ℓ

Question : Does there exist a flow f such that $\Delta^+(D_f) \leq k$ and $|f| \geq \ell$?

Trivial when $k = 1$.

Theorem

For every fixed $k \geq 2$, $(\Delta^+ \leq k)$-Maximum-Flow is NP-complete even when restricted to acyclic networks.

Degree constrained flows

$(\Delta^+ \leq k)$-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$ and an integer ℓ

Question : Does there exist a flow f such that $\Delta^+(D_f) \leq k$ and $|f| \geq \ell$?

Trivial when $k = 1$.

Theorem

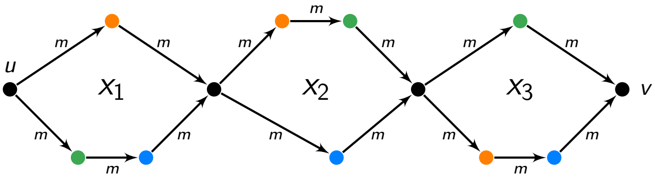
For every fixed $k \geq 2$, $(\Delta^+ \leq k)$-Maximum-Flow is NP-complete even when restricted to acyclic networks.

NP-hardness of $(\Delta^+ \leq 2)$-Maximum-Flow

Reduction from 3-SAT. $\mathcal{F} = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$

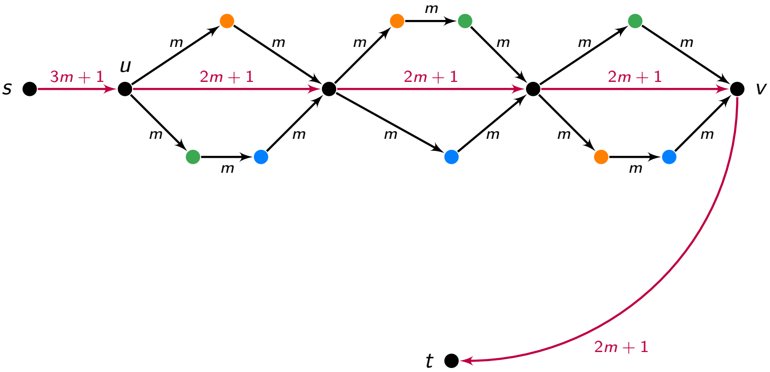
NP-hardness of $(\Delta^+ \leq 2)$-Maximum-Flow

Reduction from 3-SAT. $\mathcal{F} = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$



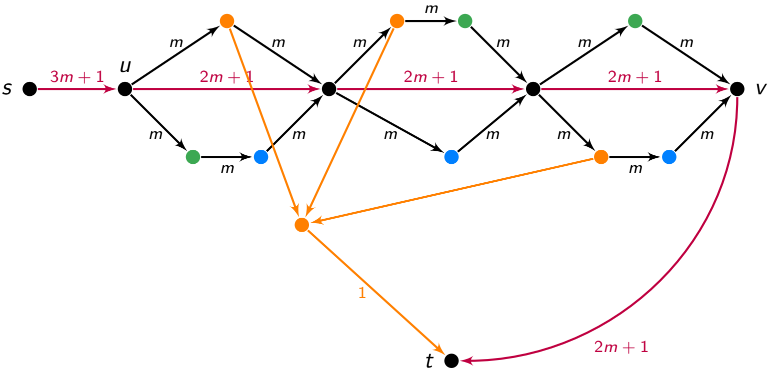
NP-hardness of $(\Delta^+ \leq 2)$-MAXIMUM-FLOW

Reduction from 3-SAT. $\mathcal{F} = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$



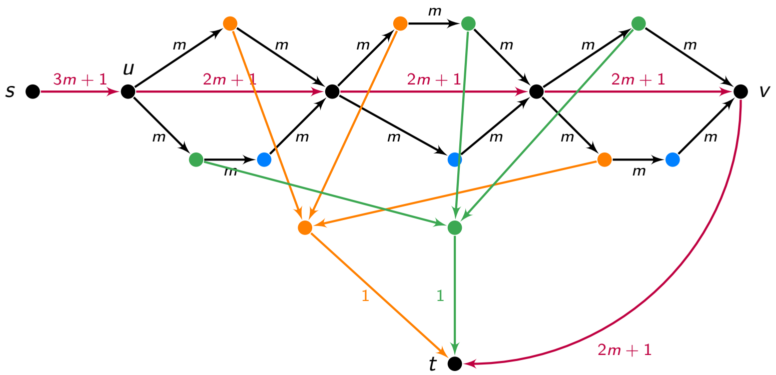
NP-hardness of $(\Delta^+ \leq 2)$-Maximum-Flow

Reduction from 3-SAT. $\mathcal{F} = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$



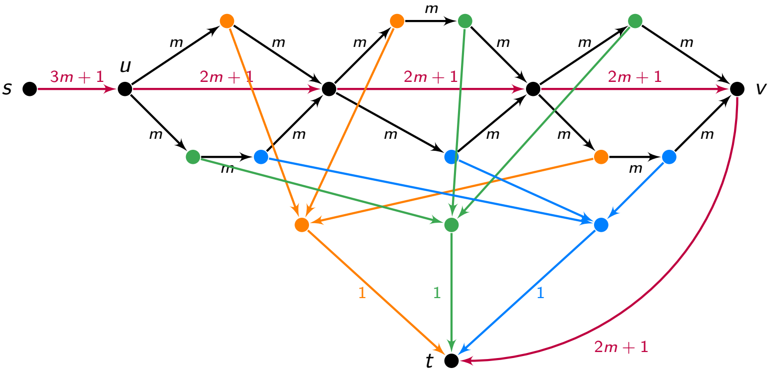
NP-hardness of $(\Delta^+ \leq 2)$-Maximum-Flow

Reduction from 3-SAT. $\mathcal{F} = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$



NP-hardness of $(\Delta^+ \leq 2)$-Maximum-Flow

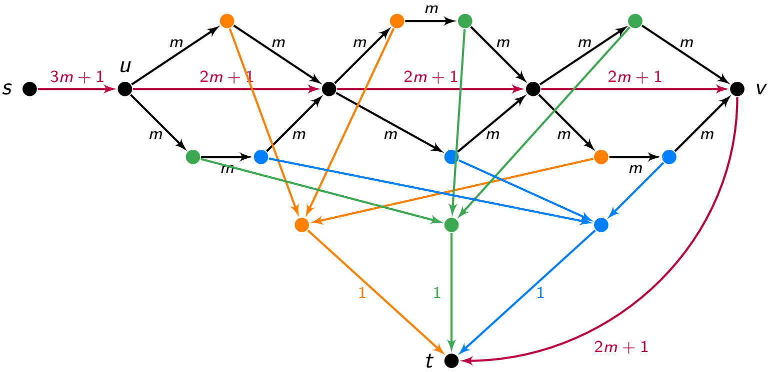
Reduction from 3-SAT. $\mathcal{F} = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$



F satisfiable iff there exists a flow f with $\Delta^+(D_f) \geq 2$ and $|f| \geq 3m+1$.

NP-hardness of $(\Delta^+ \leq 2)$-Maximum-Flow

Reduction from 3-SAT. $\mathcal{F} = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$



F satisfiable iff there exists a flow f with $\Delta^+(D_f) \geq 2$ and $|f| \geq 3m+1$.

Question: What if we have bounded capacities?

$(\Delta^+ \leq k)$-Maximum-Flow when $\ell \notin$ input

$(\Delta^+ \leq k)$-Flow of Value $k + p$

Input : A network $\mathcal{N} = (D, s, t, c)$.
Question : Does there exist a flow f such that $\Delta^+(D_f) \leq k$ and $|f| \geq k + p$?

Theorem

$(\Delta^+ \leq k)$-Flow of Value $k + p$ is solvable in time $O(n^{g(k,p)})$.

Corollary

$(\Delta^+ \leq k)$-Maximum-Flow is solvable in polynomial time on networks with bounded capacities (in time $O(n^{g(k, k \cdot c_{\max})})$).

$(\Delta^+ \leq k)$-Maximum-Flow when $\ell \notin$ input

$(\Delta^+ \leq k)$-Flow of Value $k+p$

Input : A network $\mathcal{N} = (D, s, t, c)$.
Question : Does there exist a flow f such that $\Delta^+(D_f) \leq k$ and $|f| \geq k+p$?

Theorem

$(\Delta^+ \leq k)$-Flow of Value $k+p$ *is solvable in time* $O(n^{g(k,p)})$.

Corollary

$(\Delta^+ \leq k)$-Maximum-Flow *is solvable in polynomial time on networks with bounded capacities (in time* $O(n^{g(k,k\cdot c_{\max})})$*).*

$(\Delta^+ \leq k)$-Maximum-Flow when $\ell \notin$ input

$(\Delta^+ \leq k)$-Flow of Value $k+p$

Input : A network $\mathcal{N} = (D, s, t, c)$.
Question : Does there exist a flow f such that $\Delta^+(D_f) \leq k$ and $|f| \geq k+p$?

Theorem

$(\Delta^+ \leq k)$-Flow of Value $k+p$ *is solvable in time* $O(n^{g(k,p)})$.

Corollary

$(\Delta^+ \leq k)$-Maximum-Flow *is solvable in polynomial time on networks with bounded capacities (in time* $O(n^{g(k,k\cdot c_{\max})})$*).*

$(\Delta^+ \leq k)$-Flow of Value $k + p$ is solvable in polynomial time

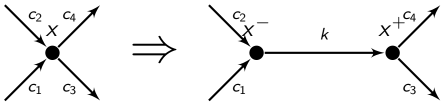
Sketch of proof for $p = 1$

1. Check that $\mathcal{N}$ has a flow of value $k + 1$ (if not, $\mathcal{N}$ is a negative instance).

$(\Delta^+ \leq k)$-Flow of Value $k + p$ is solvable in polynomial time

Sketch of proof for $p = 1$

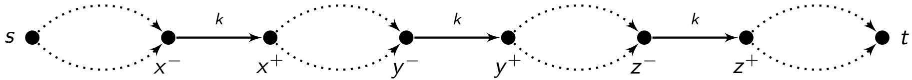
1. Check that $\mathcal{N}$ has a flow of value $k+1$ (if not, $\mathcal{N}$ is a negative instance).
2. Split every vertex (except s, t) to ensure $\Delta^+(D_f) \leq k$ for every flow f.



$(\Delta^+ \leq k)$-Flow of Value $k+p$ is solvable in polynomial time

Sketch of proof for $p = 1$

1. Check that $\mathcal{N}$ has a flow of value $k+1$ (if not, $\mathcal{N}$ is a negative instance).
2. Split every vertex (except s, t) to ensure $\Delta^+(D_f) \leq k$ for every flow f.
3. Using the cuts of capacity k, partition into smaller networks.



$(\Delta^+ \leq k)$-FLOW OF VALUE $k + p$ is solvable in polynomial time

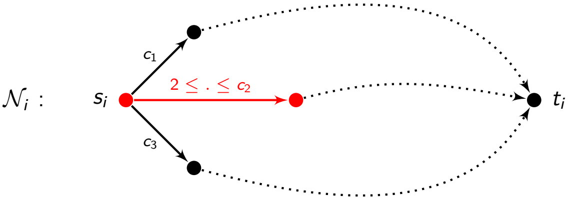
Sketch of proof for $p = 1$

1. Check that $\mathcal{N}$ has a flow of value $k + 1$ (if not, $\mathcal{N}$ is a negative instance).
2. Split every vertex (except s, t) to ensure $\Delta^+(D_f) \leq k$ for every flow f.
3. Using the cuts of capacity k, partition into smaller networks.

$(\Delta^+ \leq k)$-Flow of Value $k+p$ is solvable in polynomial time

Sketch of proof for $p = 1$

1. Check that $\mathcal{N}$ has a flow of value $k+1$ (if not, $\mathcal{N}$ is a negative instance).
2. Split every vertex (except s, t) to ensure $\Delta^+(D_f) \leq k$ for every flow f.
3. Using the cuts of capacity k, partition into smaller networks.
4. $\mathcal{N}$ is a positive instance iff for every sub-network $\mathcal{N}_i$, there exists a flow f_i of value $k+1$ with $f_i(s_i u) \geq 2$ for some arc $s_i u$.



Flows decomposable into few path-flows

p-Decomposable-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$
Output : The maximum value of a flow f s.t. f decomposes into at most p path-flows.

Theorem (Baier, Köhler, and Skutella 2005)

2-Decomposable-Maximum-Flow *is NP-hard and cannot be approximated by any ratio larger than* $\frac{2}{3}$.

Theorem (Baier, Köhler, and Skutella 2005)

p-Decomposable-Maximum-Flow *can be approximated by a ratio* $\rho = \frac{2}{3}$ *when* $p \in \{2, 3\}$ *and by a ratio* $\rho = \frac{1}{2}$ *when* $p \geq 4$.

Flows decomposable into few path-flows

p-Decomposable-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$
Output : The maximum value of a flow f s.t. f decomposes into at most p path-flows.

Theorem (Baier, Köhler, and Skutella 2005)

2-Decomposable-Maximum-Flow *is NP-hard and cannot be approximated by any ratio larger than* $\frac{2}{3}$.

Theorem (Baier, Köhler, and Skutella 2005)

p-Decomposable-Maximum-Flow *can be approximated by a ratio* $\rho = \frac{2}{3}$ *when* $p \in \{2, 3\}$ *and by a ratio* $\rho = \frac{1}{2}$ *when* $p \geq 4$.

Flows decomposable into few path-flows

p-Decomposable-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$
Output : The maximum value of a flow f s.t. f decomposes into at most p path-flows.

Theorem (Baier, Köhler, and Skutella 2005)

2-Decomposable-Maximum-Flow *is NP-hard and cannot be approximated by any ratio larger than* $\frac{2}{3}$.

Theorem (Baier, Köhler, and Skutella 2005)

p-Decomposable-Maximum-Flow *can be approximated by a ratio* $\rho = \frac{2}{3}$ *when* $p \in \{2, 3\}$ *and by a ratio* $\rho = \frac{1}{2}$ *when* $p \geq 4$.

Flows decomposable into few path-flows : Hardness

Theorem

For every fixed $p \geq 2$*, the* p-Decomposable-Max-Flow *problem is NP-hard. Moreover, unless P=NP, it cannot be approximated by any ratio larger than* $\rho(p) = \min(\rho_1(p), \rho_2(p))$*, where* $\rho_1(p), \rho_2(p)$ *are defined as follows:*

$$\rho_1(p) = \begin{cases} \frac{5}{6} & \text{if } p = 0 \mod 4 \\ \frac{5p-1}{6p-2} & \text{if } p = 1 \mod 4 \\ \frac{5p-2}{6p} & \text{if } p = 2 \mod 4 \\ \frac{5p-3}{6p-2} & \text{if } p = 3 \mod 4 \end{cases}$$

$$\rho_2(p) = \begin{cases} \frac{4}{5} & \text{if } p \text{ is even} \\ \frac{4p-2}{5p-3} & \text{otherwise.} \end{cases}$$

In particular, $\rho(2) = \frac{2}{3}$, $\rho(3) = \frac{3}{4}$, $\rho(p) \xrightarrow[p \to +\infty]{} \frac{4}{5}$*, and* $\rho(p) \leq \frac{9}{11}$ *in general.*

Flows decomposable into few disjoint path-flows.

p-Vertex-Decomposable-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$
Output : The maximum value of a flow f s.t. f decomposes into at most p path-flows intersecting exactly on $\{s, t\}$.

Theorem

For every fixed $p \geq 2$, p-Vertex-Decomposable-Maximum-Flow is NP-hard and cannot be approximated by any ratio larger than $\frac{2}{3}$.

Theorem

For every fixed $p \geq 2$, p-Vertex-Decomposable-Maximum-Flow can be approximated by a ratio $\rho = \frac{1}{H(p)}$ where $H(p) = \sum_{i=1}^{p} \frac{1}{i} \sim_p \ln(p)$.

Flows decomposable into few disjoint path-flows.

p-Vertex-Decomposable-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$
Output : The maximum value of a flow f s.t. f decomposes into at most p path-flows intersecting exactly on $\{s, t\}$.

Theorem

For every fixed $p \geq 2$, p-Vertex-Decomposable-Maximum-Flow is NP-hard and cannot be approximated by any ratio larger than $\frac{2}{3}$.

Theorem

For every fixed $p \geq 2$, p-Vertex-Decomposable-Maximum-Flow can be approximated by a ratio $\rho = \frac{1}{H(p)}$ where $H(p) = \sum_{i=1}^{p} \frac{1}{i} \sim_p \ln(p)$.

Flows decomposable into few disjoint path-flows.

p-Vertex-Decomposable-Maximum-Flow

Input : A network $\mathcal{N} = (D, s, t, c)$
Output : The maximum value of a flow f s.t. f decomposes into at most p path-flows intersecting exactly on $\{s, t\}$.

Theorem

For every fixed $p \geq 2$, p-Vertex-Decomposable-Maximum-Flow is NP-hard and cannot be approximated by any ratio larger than $\frac{2}{3}$.

Theorem

For every fixed $p \geq 2$, p-Vertex-Decomposable-Maximum-Flow can be approximated by a ratio $\rho = \frac{1}{H(p)}$ where $H(p) = \sum_{i=1}^{p} \frac{1}{i} \sim_p \ln(p)$.

$\frac{1}{H(p)}$-approximation for p-Vertex-Decomposable-Maximum-Flow

Algorithm:

1. for every $i \in \{1, \ldots, p\}$, find the largest capacity c_i s.t. $D \setminus \{uv \in A \mid c(uv) < c_i\}$ contains i disjoint (s,t)-paths.
2. return $\max\{i \cdot c_i \mid i \in \{1, \ldots, p\}\}$.

Proof:

- Let f^* be an optimal solution with path-flows $P_1^*, \ldots, P_p^*$ of values respectively $c_1^* \leq \cdots \leq c_p^*$ and f be the flow computed by the algorithm above.
- For every $i \in \{1, \ldots, p\}$, $|f| \geq i \cdot c_i \geq i \cdot c_i^*$.
- Summing the inequalities above for every i we obtain:

$$|x| \cdot \sum_{i=1}^{p} \frac{1}{i} \geq \sum_{i=1}^{p} c_i^* = |f^*|.$$

$\frac{1}{H(p)}$-approximation for p-Vertex-Decomposable-Maximum-Flow

Algorithm:

1. for every $i \in \{1, \dots, p\}$, find the largest capacity c_i s.t. $D \setminus \{uv \in A \mid c(uv) < c_i\}$ contains i disjoint (s,t)-paths.
2. return $\max\{i \cdot c_i \mid i \in \{1, \dots, p\}\}$.

Proof:

- Let f^* be an optimal solution with path-flows $P_1^*, \dots, P_p^*$ of values respectively $c_1^* \leq \dots \leq c_p^*$ and f be the flow computed by the algorithm above.
- For every $i \in \{1, \dots, p\}$, $|f| \geq i \cdot c_i \geq i \cdot c_i^*$.
- Summing the inequalities above for every i we obtain:

$$|x| \cdot \sum_{i=1}^{p} \frac{1}{i} \geq \sum_{i=1}^{p} c_i^* = |f^*|.$$

$\frac{1}{H(p)}$-approximation for p-Vertex-Decomposable-Maximum-Flow

Algorithm:

1. for every $i \in \{1, \dots, p\}$, find the largest capacity c_i s.t. $D \setminus \{uv \in A \mid c(uv) < c_i\}$ contains i disjoint (s, t)-paths.
2. return $\max\{i \cdot c_i \mid i \in \{1, \dots, p\}\}$.

Proof:

- Let f^* be an optimal solution with path-flows $P_1^*, \dots, P_p^*$ of values respectively $c_1^* \leq \dots \leq c_p^*$ and f be the flow computed by the algorithm above.
- For every $i \in \{1, \dots, p\}$, $|f| \geq i \cdot c_i \geq i \cdot c_i^*$.
- Summing the inequalities above for every i we obtain:

$$|x| \cdot \sum_{i=1}^{p} \frac{1}{i} \geq \sum_{i=1}^{p} c_i^* = |f^*|.$$

$\frac{1}{H(p)}$-approximation for p-Vertex-Decomposable-Maximum-Flow

Algorithm:

1. for every $i \in \{1, \ldots, p\}$, find the largest capacity c_i s.t. $D \setminus \{uv \in A \mid c(uv) < c_i\}$ contains i disjoint (s, t)-paths.
2. return $\max\{i \cdot c_i \mid i \in \{1, \ldots, p\}\}$.

Proof:

- Let f^* be an optimal solution with path-flows $P_1^*, \ldots, P_p^*$ of values respectively $c_1^* \leq \cdots \leq c_p^*$ and f be the flow computed by the algorithm above.
- For every $i \in \{1, \ldots, p\}$, $|f| \geq i \cdot c_i \geq i \cdot c_i^*$.
- Summing the inequalities above for every i we obtain:

$$|x| \cdot \sum_{i=1}^{p} \frac{1}{i} \geq \sum_{i=1}^{p} c_i^* = |f^*|.$$

□

$\frac{1}{H(p)}$-approximation for p-Vertex-Decomposable-Maximum-Flow

Algorithm:

1. for every $i \in \{1, \dots, p\}$, find the largest capacity c_i s.t. $D \setminus \{uv \in A \mid c(uv) < c_i\}$ contains i disjoint (s,t)-paths.
2. return $\max\{i \cdot c_i \mid i \in \{1, \dots, p\}\}$.

Proof:

- Let f^* be an optimal solution with path-flows $P_1^*, \dots, P_p^*$ of values respectively $c_1^* \leq \dots \leq c_p^*$ and f be the flow computed by the algorithm above.
- For every $i \in \{1, \dots, p\}$, $|f| \geq i \cdot c_i \geq i \cdot c_i^*$.
- Summing the inequalities above for every i we obtain:

$$|x| \cdot \sum_{i=1}^{p} \frac{1}{i} \geq \sum_{i=1}^{p} c_i^* = |f^*|.$$

$\square$

p-Vertex-Decomposable-Maximum-Flow on acyclic networks

Theorem

*When p is part of the input, p-*Vertex-Decomposable-Maximum-Flow *on acyclic networks is NP-hard, even when the capacities are in $\{1,2\}$.*

Theorem

p-Vertex-Decomposable-Maximum-Flow *on acyclic networks is solvable in time $O\left(n^{f(p)}\right)$ for some computable function f.*

Theorem

*When parameterized by p, p-*Vertex-Decomposable-Maximum-Flow *on acyclic networks is $W[1]$-hard.*

p-Vertex-Decomposable-Maximum-Flow on acyclic networks

Theorem

When p is part of the input, p-Vertex-Decomposable-Maximum-Flow on acyclic networks is NP-hard, even when the capacities are in $\{1, 2\}$.

Theorem

p-Vertex-Decomposable-Maximum-Flow on acyclic networks is solvable in time $O\left(n^{f(p)}\right)$ for some computable function f.

Theorem

When parameterized by p, p-Vertex-Decomposable-Maximum-Flow on acyclic networks is $W[1]$-hard.

p-Vertex-Decomposable-Maximum-Flow on acyclic networks

Theorem

When p is part of the input, p-Vertex-Decomposable-Maximum-Flow on acyclic networks is NP-hard, even when the capacities are in $\{1, 2\}$.

Theorem

p-Vertex-Decomposable-Maximum-Flow on acyclic networks is solvable in time $O\left(n^{f(p)}\right)$ for some computable function f.

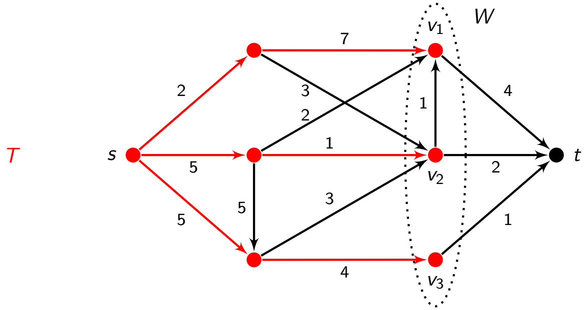
Theorem

When parameterized by p, p-Vertex-Decomposable-Maximum-Flow on acyclic networks is $W[1]$-hard.

Notion of W-tricots

$W \subseteq V(D) \setminus \{s,t\}$: an ordered set of p vertices $(v_1, \dots, v_p)$.

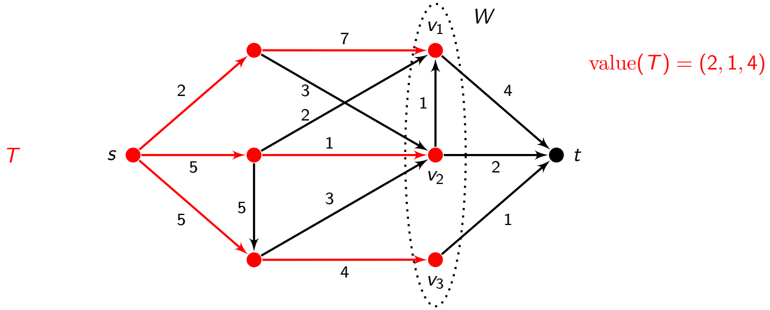
- A W-tricot T is a sequence of paths $(Q_1, \dots, Q_p)$ pairwise intersecting exactly on $\{s\}$ s.t. $\mathrm{end}(Q_i) = v_i$.



Notion of W-tricots

$W \subseteq V(D) \setminus \{s,t\}$: an ordered set of p vertices $(v_1, \dots, v_p)$.

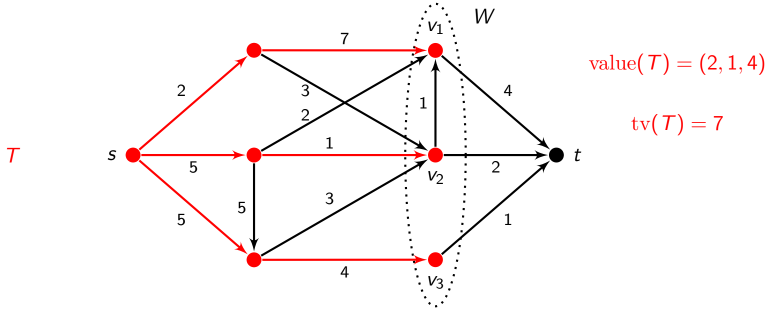
- A W-tricot T is a sequence of paths $(Q_1, \dots, Q_p)$ pairwise intersecting exactly on $\{s\}$ s.t. $\mathrm{end}(Q_i) = v_i$.
- The value $\mathrm{tv}(T)$ of T is $(c_1, \dots, c_p)$ where c_i is the minimum capacity along Q_i.



Notion of W-tricots

$W \subseteq V(D) \setminus \{s, t\}$: an ordered set of p vertices $(v_1, \dots, v_p)$.

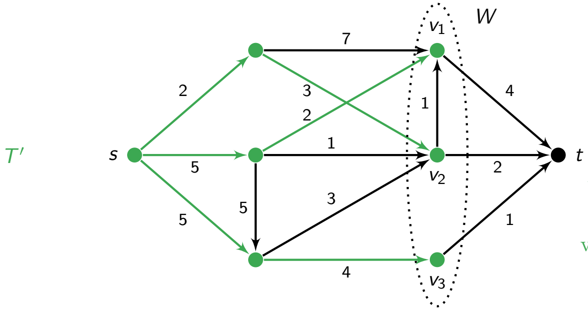
- A W-tricot T is a sequence of paths $(Q_1, \dots, Q_p)$ pairwise intersecting exactly on $\{s\}$ s.t. $\mathrm{end}(Q_i) = v_i$.
- The value $\mathrm{tv}(T)$ of T is $(c_1, \dots, c_p)$ where c_i is the minimum capacity along Q_i.
- The total value of T is $\sum_{i=1}^{p} c_i$.



Notion of W-tricots

$W \subseteq V(D) \setminus \{s,t\}$: an ordered set of p vertices $(v_1, \dots, v_p)$.

- A W-tricot T is a sequence of paths $(Q_1, \dots, Q_p)$ pairwise intersecting exactly on $\{s\}$ s.t. $\mathrm{end}(Q_i) = v_i$.
- The value $\mathrm{tv}(T)$ of T is $(c_1, \dots, c_p)$ where c_i is the minimum capacity along Q_i.
- The total value of T is $\sum_{i=1}^{p} c_i$.
- We have $\mathrm{value}(T) \preceq \mathrm{value}(T')$ iff $\forall i \in \{1, \dots, p\},\ c_i \leq c'_i$.



$\mathrm{value}(T) = (2,1,4)$

$\mathrm{tv}(T) = 7$

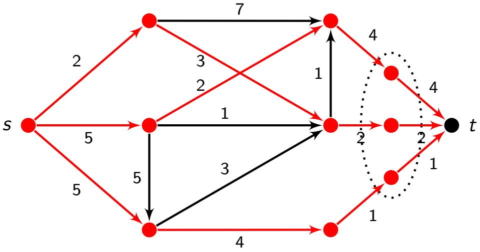
$\mathrm{value}(T') = (2,2,4)$

$\mathrm{value}(T) \preceq \mathrm{value}(T')$

Properties of the W-tricots

- After subdividing every arc vt, the optimal solution of p-Vertex-Decomposable-Maximum-Flow is exactly:

$$\max_{W \subseteq N^-(t),\ |W| \leq p} \max\{\mathrm{tv}(T) \mid T \text{ is a } W\text{-tricot}\}. \quad (\star)$$

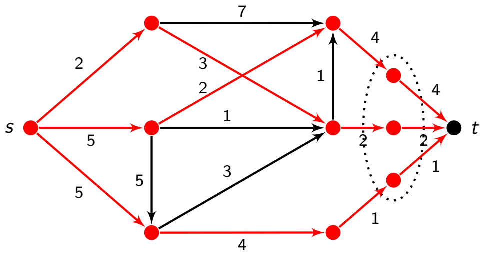


Properties of the W-tricots

- After subdividing every arc vt, the optimal solution of p-VERTEX-DECOMPOSABLE-MAXIMUM-FLOW is exactly:

$$\max_{W \subseteq N^-(t),\ |W| \leq p} \max\{\mathrm{tv}(T) \mid T \text{ is a } W\text{-tricot}\}. \quad (\star)$$

- The size of $\{\mathrm{value}(T) \mid T \text{ is a } W\text{-tricot}\}$ is bounded by $O(m^p)$.

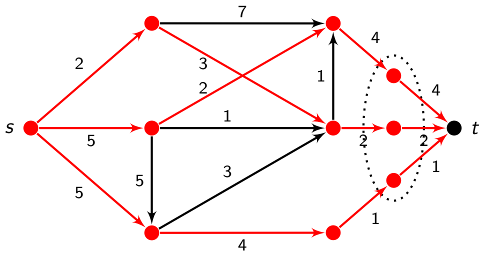


Properties of the W-tricots

- After subdividing every arc vt, the optimal solution of p-VERTEX-DECOMPOSABLE-MAXIMUM-FLOW is exactly:

$$\max_{W \subseteq N^-(t),\ |W| \leq p} \max\{\mathrm{tv}(T) \mid T \text{ is a } W\text{-tricot}\}. \quad (\star)$$

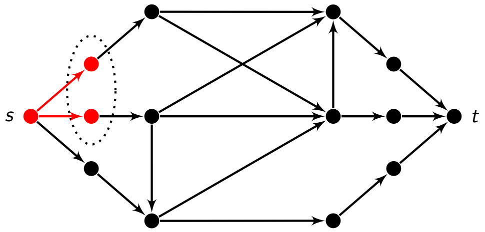
- The size of $\{\mathrm{value}(T) \mid T \text{ is a } W\text{-tricot}\}$ is bounded by $O(m^p)$.
- **Goal:** compute $\{\mathrm{value}(T) \mid T \text{ is a } W\text{-tricot}\}$ for every W, and return $(\star)$.



Polynomial-Time Algorithm for exactly p path-flows:

L : list indexed by the p-tuples of $V(D) \setminus \{s\}$. Each cell $L[W]$ is a set of W-tricots.

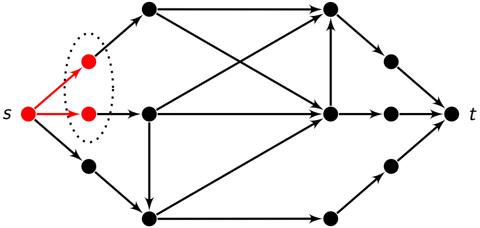
1. $\forall W \subseteq N^+(s)$, $L[W] \leftarrow \{$the only W-tricot$\}$.



Polynomial-Time Algorithm for exactly p path-flows:

L : list indexed by the p-tuples of $V(D) \setminus \{s\}$. Each cell $L[W]$ is a set of W-tricots.

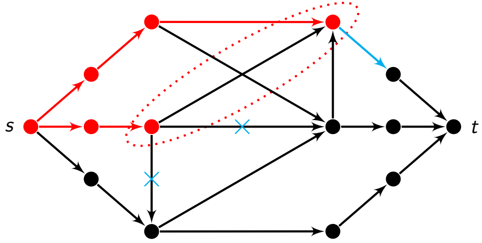
1. $\forall W \subseteq N^+(s)$, $L[W] \leftarrow$ {the only W-tricot}.
2. Fix an acyclic ordering $v_1, \ldots, v_n$ with initial vertices $N^+[s]$, and the corresponding lexicographic ordering of the p-tuples $W_1, \ldots, W_r$ where $r = \binom{n}{p} \cdot p!$.



Polynomial-Time Algorithm for exactly p path-flows:

L : list indexed by the p-tuples of $V(D) \setminus \{s\}$. Each cell $L[W]$ is a set of W-tricots.

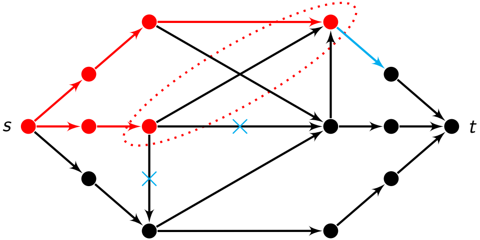
1. $\forall W \subseteq N^+(s)$, $L[W] \leftarrow \{\text{the only } W\text{-tricot}\}$.
2. Fix an acyclic ordering $v_1, \ldots, v_n$ with initial vertices $N^+[s]$, and the corresponding lexicographic ordering of the p-tuples $W_1, \ldots, W_r$ where $r = \binom{n}{p} \cdot p!$.
3. $\forall W_i$ in this order, $\forall T = (Q_1, \ldots, Q_p) \in L[W_i]$, consider every extension $T' = (Q_1, \ldots, Q_j \cup \{y\}, \ldots, Q_p)$ of T s.t. $\forall k$, $\mathrm{end}(Q_k) \prec y$.



Polynomial-Time Algorithm for exactly p path-flows:

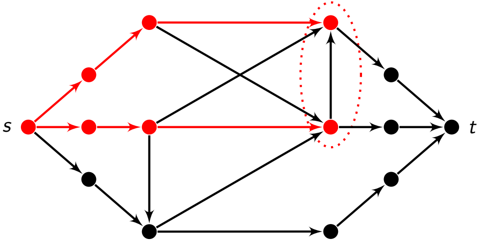
L : list indexed by the p-tuples of $V(D) \setminus \{s\}$. Each cell $L[W]$ is a set of W-tricots.

1. $\forall W \subseteq N^+(s)$, $L[W] \leftarrow \{$the only W-tricot$\}$.
2. Fix an acyclic ordering $v_1, \dots, v_n$ with initial vertices $N^+[s]$, and the corresponding lexicographic ordering of the p-tuples $W_1, \dots, W_r$ where $r = \binom{n}{p} \cdot p!$.
3. $\forall W_i$ in this order, $\forall T = (Q_1, \dots, Q_p) \in L[W_i]$, consider every extension $T' = (Q_1, \dots, Q_j \cup \{y\}, \dots, Q_p)$ of T s.t. $\forall k,\ \mathrm{end}(Q_k) \prec y$.
4. If for all W'-tricot $\tilde{T} \in L[W']$, $\mathrm{value}(T') \not\leq \mathrm{value}(\tilde{T})$, then $L[W'] \leftarrow L[W'] \cup \{T'\}$.



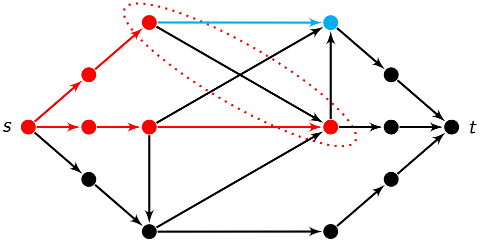
Validity of the algorithm

Invariant: when W_i is considered, $\forall W_i$-tricot T, $L[W_i]$ contains a tricot T' s.t. $\mathrm{value}(T) \preceq \mathrm{value}(T')$.



Validity of the algorithm

Invariant: when W_i is considered, $\forall W_i$-tricot T, $L[W_i]$ contains a tricot T' s.t. $\mathrm{value}(T) \preceq \mathrm{value}(T')$.



Open questions

Question: Is there a way to approximate the $(\Delta^+ \leq k)$-Maximum-Flow problem?

Question: What is the best approximation guarantee one can obtain for the p-Decomposable-Maximum-Flow problem when $p > 2$?

$$\frac{2}{3} \leq \rho(3) \leq \frac{3}{4}$$

Thank you!

Open questions

Question: Is there a way to approximate the $(\Delta^+ \leq k)$-MAXIMUM-FLOW problem?

Question: What is the best approximation guarantee one can obtain for the p-DECOMPOSABLE-MAXIMUM-FLOW problem when $p > 2$?

$$\frac{2}{3} \leq \rho(3) \leq \frac{3}{4}$$

Thank you!

Open questions

Question: Is there a way to approximate the $(\Delta^+ \leq k)$-$\textsc{Maximum-Flow}$ problem?

Question: What is the best approximation guarantee one can obtain for the p-$\textsc{Decomposable-Maximum-Flow}$ problem when $p > 2$?

$$\frac{2}{3} \leq \rho(3) \leq \frac{3}{4}$$

Thank you!