G-SCOP SCIENCES POUR LA CONCEPTION, L'OPTIMISATION ET LA PRODUCTION

# Directed hypergraph connectivity augmentation by hyperarc reorientation

*Joint work with: Moritz Mühlenthaler and Zoltán Szigeti*


Benjamin Peyrille


November 23th 2023


UMR 5272

GRENOBLE INP UGA

cnrs

UGA Université Grenoble Alpes

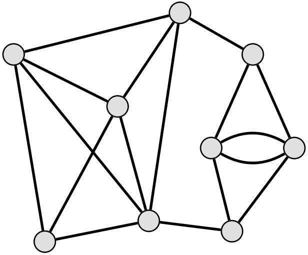
Laboratoire des Sciences pour la Conception, l'Optimisation et la production

G-SCOP

# Connectivity

## Edge connectivity

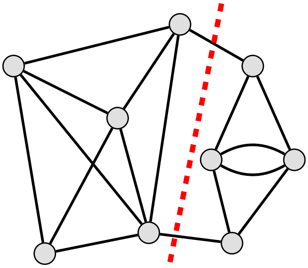
A graph $G = (V, E)$ is $k$-**edge-connected** if and only if for all non-empty vertex set $X \neq V : d(X) \geq k$.



# Connectivity

## Edge connectivity

A graph $G = (V, E)$ is $k$-**edge-connected** if and only if for all non-empty vertex set $X \neq V$ : $d(X) \geq k$.



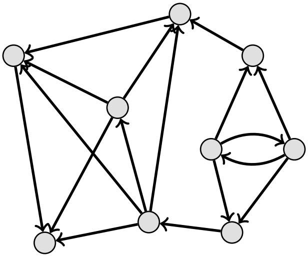
This graph is 2-edge-connected.

G-SCOP

# Connectivity

## Arc connectivity

A graph orientation $\vec{G} = (V, A)$ is $k$-**arc-connected** if and only if for all non-empty vertex set $X \neq V$: $d^{-}(X) \geq k$.

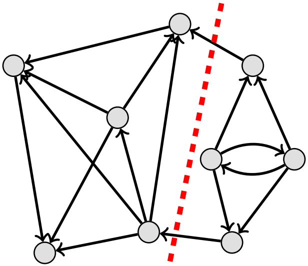


G-SCOP

# Connectivity

## Arc connectivity

A graph orientation $\vec{G} = (V, A)$ is $k$-**arc-connected** if and only if for all non-empty vertex set $X \neq V$: $d^{-}(X) \geq k$.



This orientation is 0-arc-connected.

# Augmentation results

**Weak Orientation Theorem (Nash-Williams, 1960)**

An undirected graph admits a $k$-arc-connected orientation if and only if it is $2k$-edge-connected.

G-SCOP

# Augmentation results

**Weak Orientation Theorem (Nash-Williams, 1960)**

An undirected graph admits a $k$-arc-connected orientation if and only if it is $2k$-edge-connected.

**Arc-Connectivity Augmentation (Ito et al., 2021)**

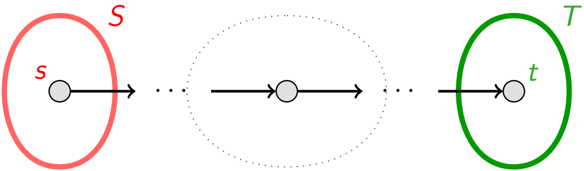
Let $G = (V, E)$ be an undirected $(2k+2)$-edge-connected graph, $D$ be a $k$-arc-connected orientation of $G$.
Then, there exist orientations $D_1, D_2, \ldots, D_\ell$ of $G$ such that

- $D_i$ is obtained from $D_{i-1}$ by reversing an arc of $D_{i-1}$,
- $\ell \leq |V|^3$,
- $\lambda(D) \leq \lambda(D_1) \leq \lambda(D_2) \leq \ldots \leq \lambda(D_\ell) = k+1$.

Furthermore, such orientations can be found in polynomial time.

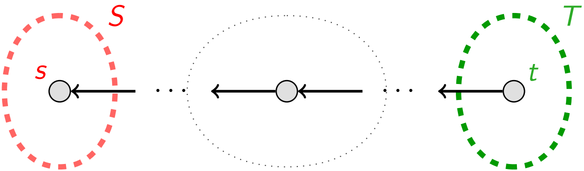
# The key idea of Ito et al.

Reversing an $(s, t)$-path only changes the connectivity of vertex sets separating $s$ and $t$.



# The key idea of Ito et al.

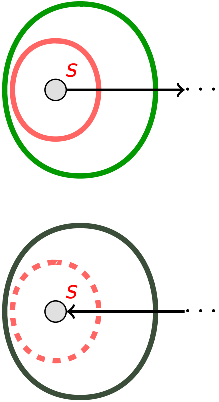
Reversing an $(s,t)$-path only changes the connectivity of vertex sets separating $s$ and $t$.




We will iteratively reverse $(s,t)$-paths connecting a minimal set $S$ of in-degree $k$ (**in-tight** $\mathcal{T}^-$) to a minimal set $T$ of out-degree $k$ (**out-tight** $\mathcal{T}^+$).
We call $s$ a **source** and $t$ a **sink**.

# The dangers

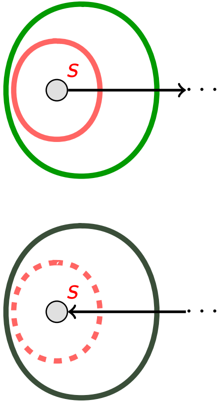
Connectivity loss by path-reversal.

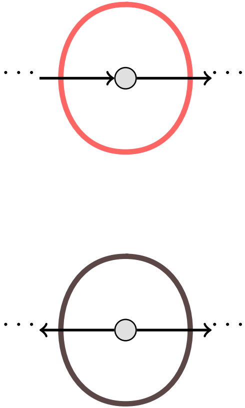


# The dangers

Connectivity loss by path-reversal.

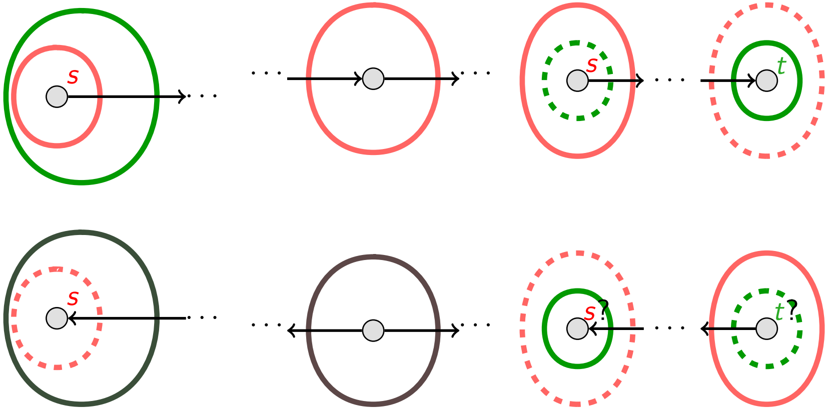
Connectivity loss by arc-reversal.



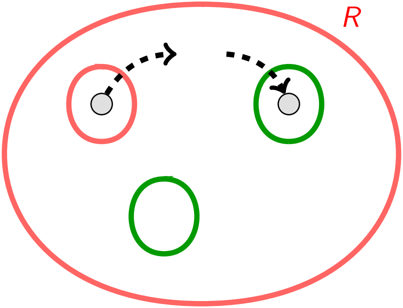
# The dangers

Connectivity loss by path-reversal. | Connectivity loss by arc-reversal. | Useless paths.



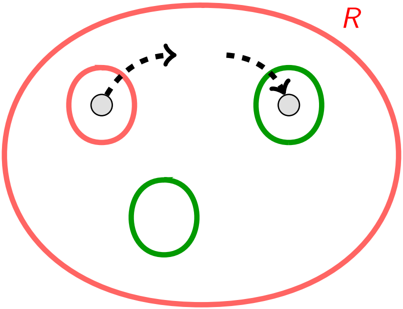
# How to preserve connectivity: path errors

We introduce a new family $\mathcal{R}^-$ containing the minimum in-tight sets containing an out-tight set.



# How to preserve connectivity: path errors

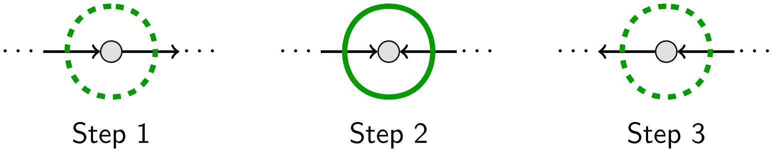
We introduce a new family $\mathcal{R}^-$ containing the minimum in-tight sets containing an out-tight set.




Restraining our paths to $R$ prevents path-reversal connectivity loss. Thus, we search for $s$ and $t$ in $R$.

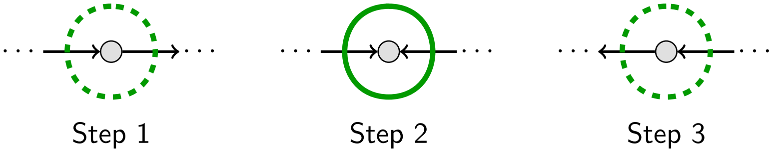
# How to preserve connectivity: arc errors

We reverse our $(s, t)$-path from end to start.
For any vertex set $X$ entered that doesn't contain $t$, $d^+(X)$ is temporarily decreased by 1.



# How to preserve connectivity: arc errors

We reverse our $(s, t)$-path from end to start.
For any vertex set $X$ entered that doesn't contain $t$, $d^+(X)$ is temporarily decreased by 1.

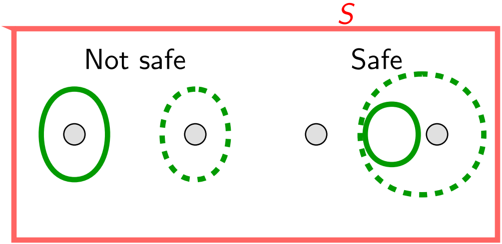



Our $(s, t)$-path must not enter any out-tight set that doesn't contain $t$.

# How to do something: safe sources

A vertex $s$ is a safe source for $S \in \mathcal{M}^-$ if:

- (Safe) If $s \in Y \in \mathcal{T}^+$ then $S \subset Y$.
- (Useful) If $s \in Z$ such that $d^+(Z) = k + 1$ and $S \not\subseteq Z$ then there exists an out-tight set in $Z$ that doesn't contain $s$.



G-SCOP

# Algorithm

- Pick a set $R \in \mathcal{R}^-$ (*If none, flip orientation*).
- Pick a safe source $s$ in a minimal set $S \in \mathcal{T}^-$ with $S \subseteq R$.
- Search for a minimum out-tight set $T$ in $R$.
  If the search enters an out-tight set, don't exit it.
- Once the search gets inside a minimum out-tight set $T$, find a safe sink $t$ in $T$.
- Reverse the search $(s, t)$-path!

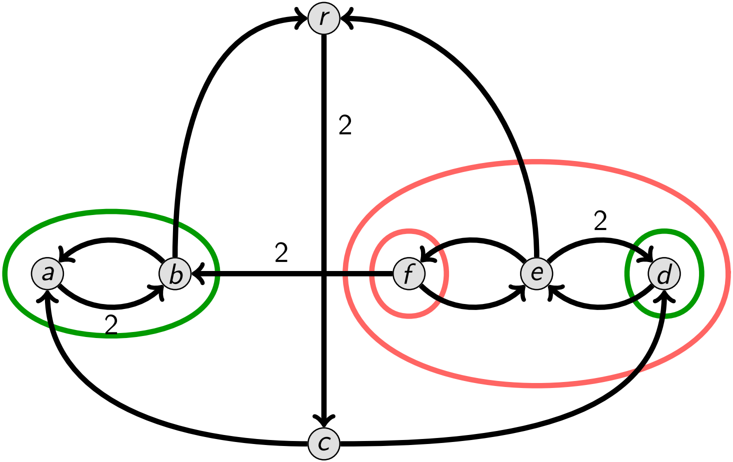
Because of the search rule, the path never leaves any out-tight set.

Repeat until no tight sets remain $\implies \lambda(D) = k + 1$.

G-SCOP

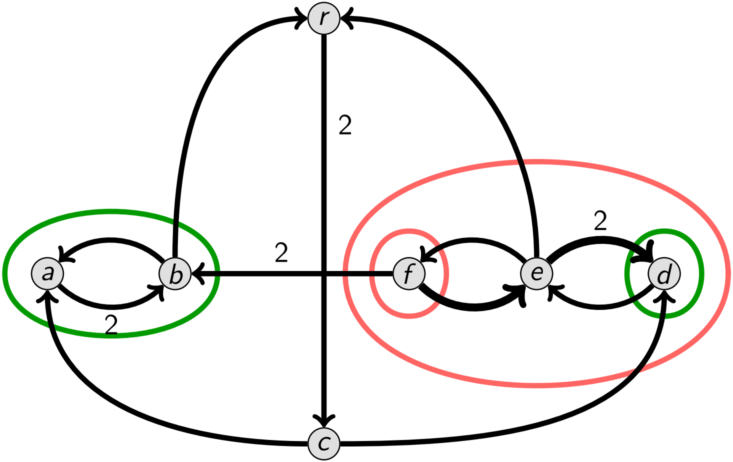
# Let's reconfigure!

Context: $G$ is 4-edge-connected and $\vec{G}$ is 1-arc-connected.



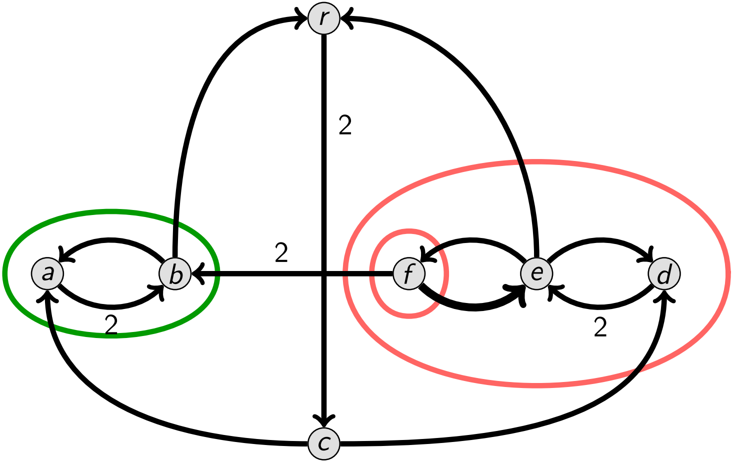
# Let's reconfigure!

Context: $G$ is 4-edge-connected and $\vec{G}$ is 1-arc-connected.



# Let's reconfigure!

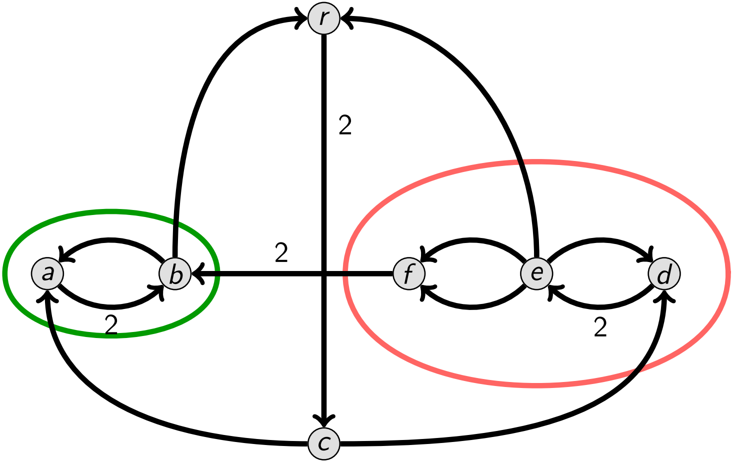
Context: $G$ is 4-edge-connected and $\vec{G}$ is 1-arc-connected.



G-SCOP

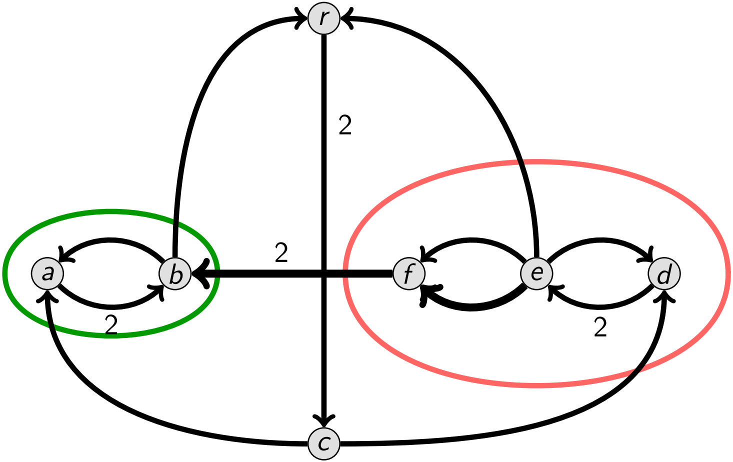
# Let's reconfigure!

Context: $G$ is 4-edge-connected and $\vec{G}$ is 1-arc-connected.



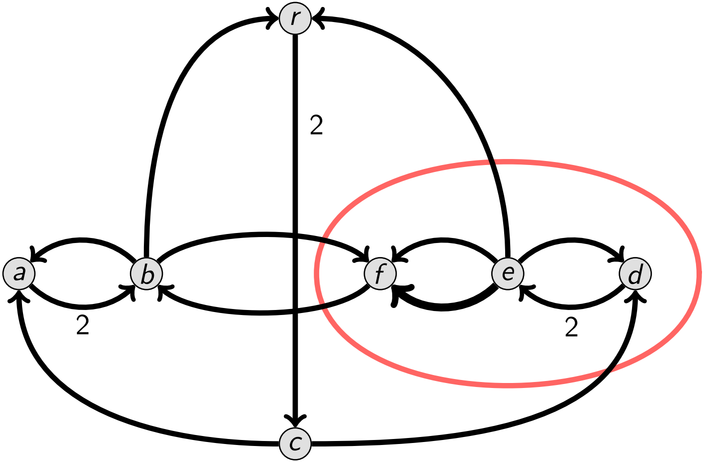
# Let's reconfigure!

Context: $G$ is 4-edge-connected and $\vec{G}$ is 1-arc-connected.



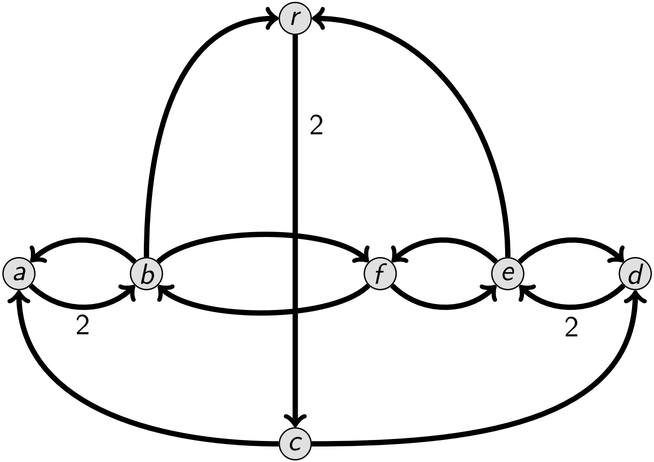
# Let's reconfigure!

Context: $G$ is 4-edge-connected and $\vec{G}$ is 1-arc-connected.



# Let's reconfigure!

Context: $G$ is 4-edge-connected and $\vec{G}$ is 1-arc-connected.

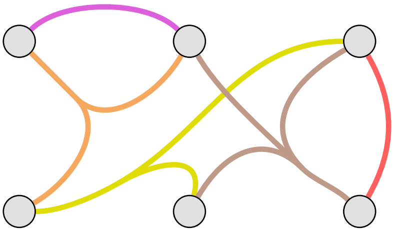


# Hypergraphs

## Hypergraph

A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is composed of:

- Vertices in $V$
- Hyperedges in $\mathcal{E}$, linking vertices together

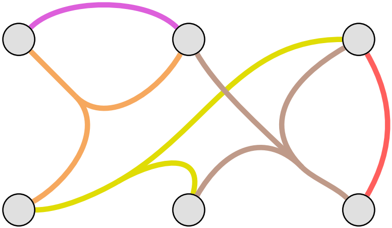


# Hypergraphs

## Hypergraph

A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is composed of:

- Vertices in $V$
- Hyperedges in $\mathcal{E}$, linking vertices together



## Partition-connectivity

$\mathcal{H}$ is $(k,k)$-**partition-connected** if for any partition $\mathcal{P}$ of $V$, at least $k|\mathcal{P}|$ hyperedges intersect at least 2 members of $\mathcal{P}$: $e_{\mathcal{H}}(\mathcal{P}) \geq k|\mathcal{P}|$.

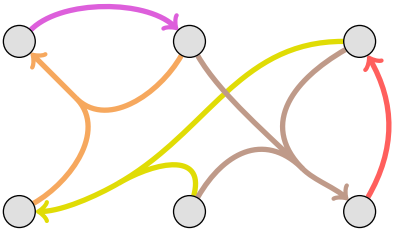
Partition-connectivity is a stronger version of edge-connectivity.

# Directed Hypergraphs

## Directed Hypergraph

A directed hypergraph $\vec{\mathcal{H}} = (V, \mathcal{A})$ is composed of:

- Vertices in $V$
- Hyperarcs in $\mathcal{A}$ with a unique head vertex

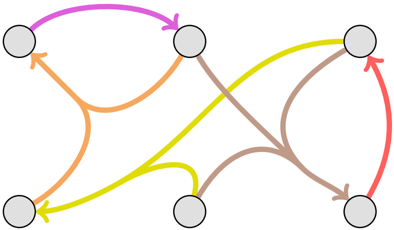


# Directed Hypergraphs

## Directed Hypergraph

A directed hypergraph $\vec{\mathcal{H}} = (V, \mathcal{A})$ is composed of:

- Vertices in $V$
- Hyperarcs in $\mathcal{A}$ with a unique head vertex



## Hyperarc-connectivity

$\vec{\mathcal{H}}$ is $k$-**hyperarc-connected** if for any non-empty vertex set $X \neq V$, at least $k$ hyperarcs enter $X$.

## Towards generalization

**Theorem on hypergraph orientations (Frank, Király, Király, 2003)**

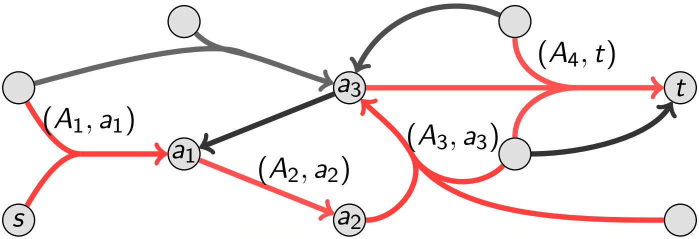
A hypergraph $\mathcal{H}$ admits a $k$-hyperarc-connected orientation if and only if it is $(k, k)$-partition-connected.

# Towards generalization

**Theorem on hypergraph orientations (Frank, Király, Király, 2003)**

A hypergraph $\mathcal{H}$ admits a $k$-hyperarc-connected orientation if and only if it is $(k, k)$-partition-connected.

Most of the previous ideas work for connectivity augmentation!
Instead of finding good paths, we find good hyperpaths and reverse them.



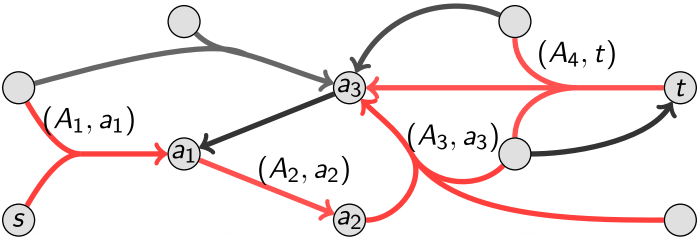
G-SCOP

# Towards generalization

**Theorem on hypergraph orientations (Frank, Király, Király, 2003)**

A hypergraph $\mathcal{H}$ admits a $k$-hyperarc-connected orientation if and only if it is $(k, k)$-partition-connected.

Most of the previous ideas work for connectivity augmentation!
Instead of finding good paths, we find good hyperpaths and reverse them.

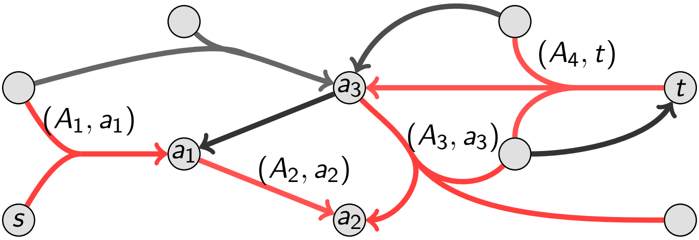


# Towards generalization

**Theorem on hypergraph orientations (Frank, Király, Király, 2003)**

A hypergraph $\mathcal{H}$ admits a $k$-hyperarc-connected orientation if and only if it is $(k, k)$-partition-connected.

Most of the previous ideas work for connectivity augmentation!
Instead of finding good paths, we find good hyperpaths and reverse them.

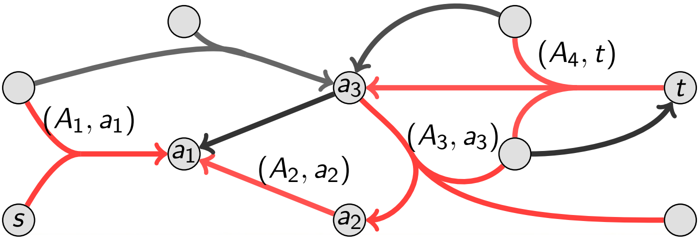


# Towards generalization

**Theorem on hypergraph orientations (Frank, Király, Király, 2003)**

A hypergraph $\mathcal{H}$ admits a $k$-hyperarc-connected orientation if and only if it is $(k, k)$-partition-connected.

Most of the previous ideas work for connectivity augmentation!
Instead of finding good paths, we find good hyperpaths and reverse them.

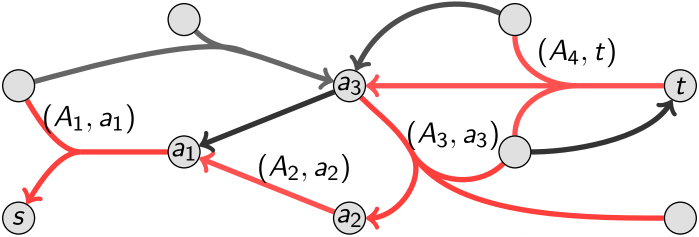


# Towards generalization

**Theorem on hypergraph orientations (Frank, Király, Király, 2003)**

A hypergraph $\mathcal{H}$ admits a $k$-hyperarc-connected orientation if and only if it is $(k,k)$-partition-connected.

Most of the previous ideas work for connectivity augmentation!
Instead of finding good paths, we find good hyperpaths and reverse them.



G-SCOP

# Our result

**Hyperarc-Connectivity Augmentation**

Let $\mathcal{H} = (V, E)$ be a $(k+1, k+1)$-partition-connected hypergraph and $\mathcal{D}$ be a $k$-hyperarc-connected orientation of $\mathcal{H}$.
Then, there exist orientations $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_\ell$ of $\mathcal{H}$ such that

- $\mathcal{D}_i$ is obtained from $\mathcal{D}_{i-1}$ by reorienting a hyperarc of $\mathcal{D}_{i-1}$,
- $\ell \leq |V|^3$,
- $\lambda(\mathcal{D}) \leq \lambda(\mathcal{D}_1) \leq \lambda(\mathcal{D}_2) \leq \ldots \leq \lambda(\mathcal{D}_\ell) = k + 1$.

Furthermore, such orientations can be found in polynomial time.

G-SCOP

# Our result

**Hyperarc-Connectivity Augmentation**

Let $\mathcal{H} = (V, E)$ be a $(k+1, k+1)$-partition-connected hypergraph and $\mathcal{D}$ be a $k$-hyperarc-connected orientation of $\mathcal{H}$.
Then, there exist orientations $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_\ell$ of $\mathcal{H}$ such that

- $\mathcal{D}_i$ is obtained from $\mathcal{D}_{i-1}$ by reorienting a hyperarc of $\mathcal{D}_{i-1}$,
- $\ell \leq |V|^3$,
- $\lambda(\mathcal{D}) \leq \lambda(\mathcal{D}_1) \leq \lambda(\mathcal{D}_2) \leq \ldots \leq \lambda(\mathcal{D}_\ell) = k+1$.

Furthermore, such orientations can be found in polynomial time.

This is the first algorithm to compute a $k$-hyperarc-connected orientation of a hypergraph.

# Frank's result : path and cycle reversing

**Reconfiguration of two $k$-arc-connected orientations (1982)**

Given two $k$-arc-connected orientations $D, D'$ of a $2k$-edge-connected graph $G$, there exist $k$-arc-connected orientations $D = D_1, D_2, \cdots, D_\ell = D'$ of $G$ such that $D_i$ is obtained from $D_{i-1}$ by reversing a path or a cycle.

Applying this theorem arc-by-arc may decrease the connectivity by one temporarily.

# Ito et al.'s result on reconfiguration

## Reconfiguration reachability of $k$-arc-connected orientations

Given two $k$-arc-connected orientations $D, D'$ of a $(2k+2)$-edge-connected graph $G$, there exist $k$-arc-connected orientations $D = D_1, D_2, \cdots, D_\ell = D'$ of $G$ such that $D_i$ is obtained from $D_{i-1}$ by reversing an arc of $D_{i-1}$.
Furthermore, such orientations can be found in polynomial time.

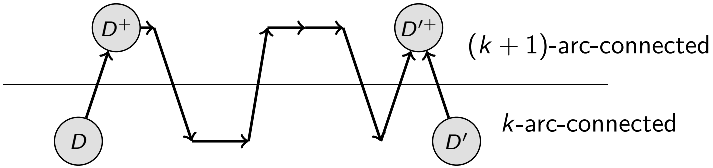
G-SCOP

# Ito et al.'s result on reconfiguration

**Reconfiguration reachability of $k$-arc-connected orientations**

Given two $k$-arc-connected orientations $D, D'$ of a $(2k+2)$-edge-connected graph $G$, there exist $k$-arc-connected orientations $D = D_1, D_2, \cdots, D_\ell = D'$ of $G$ such that $D_i$ is obtained from $D_{i-1}$ by reversing an arc of $D_{i-1}$.
Furthermore, such orientations can be found in polynomial time.

We augment $D$ and $D'$ to $(k+1)$-arc-connectivity, then we apply Frank's reconfiguration algorithm arc-by-arc.



# It works on hypergraphs

We can adapt the proof of Frank to work on hypergraph orientations, leading to the following generalization.

**Reconfiguration reachability of $k$-hyper-connected orientations**

Given two $k$-hyperarc-connected orientations $\mathcal{D}, \mathcal{D}'$ of a $(k+1, k+1)$-partition-connected hypergraph $\mathcal{H}$, there exist $k$-hyperarc-connected orientations $\mathcal{D} = \mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_\ell = \mathcal{D}'$ of $\mathcal{H}$ such that $\mathcal{D}_i$ is obtained from $\mathcal{D}_{i-1}$ by reorienting an hyperarc of $D_{i-1}$.
Furthermore, such orientations can be found in polynomial time.

# Conclusion

We generalized the results of Ito et al. to hypergraphs:

- We provided the first combinatorial algorithm for computing a $k$-hyperarc-connected orientation of a hypergraph.
- We show it is possible to reconfigure a $k$-hyperarc-connected orientation of a hypergraph into any other, if the hypergraph is $(k+1, k+1)$-partition-connected.

Open questions:

- Our upper bound on the number of reorientated hyperarcs is $|V|^3$. Can we do lower? (maybe $|V|^2$)
- The target when augmenting is $d^-(X) \geq k$. For which $f$ can we replace $k$ with $f(X)$?

# Conclusion

We generalized the results of Ito et al. to hypergraphs:

- We provided the first combinatorial algorithm for computing a $k$-hyperarc-connected orientation of a hypergraph.
- We show it is possible to reconfigure a $k$-hyperarc-connected orientation of a hypergraph into any other, if the hypergraph is $(k+1, k+1)$-partition-connected.

Open questions:

- Our upper bound on the number of reorientated hyperarcs is $|V|^3$. Can we do lower? (maybe $|V|^2$)
- The target when augmenting is $d^-(X) \geq k$. For which $f$ can we replace $k$ with $f(X)$?

Merci !