Conflict Packing
Linear kernels for FAST and Dense-RTI

Christophe Paul

CNRS - LIRMM, Montpellier France

Joint work with
A. Perez (LIFO, Orléans, France)
S. Thomassé (LIP, Lyon, France)

Berlin, December 3, 2012

"Observation" : global reduction rules yields better kernel size than local reduction rules

- Integer linear programming for Vertex Cover
- Crown decomposition for Vertex Cover, Cluster Editing
- Matching (or expansion lemma) for Feedback Vertex Set
- Region decomposition, protrusion decomposition ...

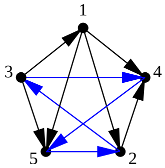
This talk : conflict packing and matching based reduction rules

Feedback Arc Set in Tournaments (FAST)

- Input: A tournament $T = (V, A)$ and a parameter k
- Question: Can T be transformed into an acyclic tournament by at most k arc reversals?

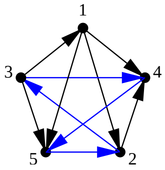
Feedback Arc Set in Tournaments (FAST)

- Input: A tournament $T = (V, A)$ and a parameter k
- Question: Can T be transformed into an acyclic tournament by at most k arc reversals?



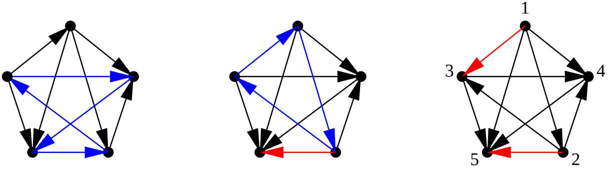
Feedback Arc Set in Tournaments (FAST)

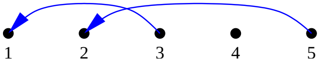
- Input: A tournament $T = (V, A)$ and a parameter k
- Question: Can T be transformed into an acyclic tournament by at most k arc reversals?



Feedback Arc Set in Tournaments (FAST)

- Input: A tournament $T = (V, A)$ and a parameter k
- Question: Can T be transformed into an acyclic tournament by at most k arc reversals?





$\rightarrow$ find a vertex ordering π with at most k backward arcs

Feedback Arc Set in Tournaments (FAST)

- NP-Complete [Alon'06] [Charbit et al.'07]
- $(1+\epsilon)$-approximation scheme [Kenyon-Mathieu, Schudy'07]

- FTP [Raman, Saurabh'06] [Alon et al.'09]

- $O(k^2)$ vertex kernel [Dom et al.'06], [Alon et al.'09]
- $O(k\sqrt{k})$ vertex kernel [Bessy et al.'10]
- $O(k)$ vertex kernel [Bessy et al.'10]

Feedback Arc Set in Tournaments (FAST)

- NP-Complete [Alon'06] [Charbit et al.'07]
- $(1+\epsilon)$-approximation scheme [Kenyon-Mathieu, Schudy'07]

- FTP [Raman, Saurabh'06] [Alon et al.'09]

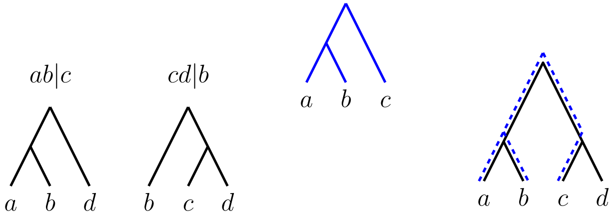
- $O(k^2)$ vertex kernel [Dom et al.'06], [Alon et al.'09]
- $O(k\sqrt{k})$ vertex kernel [Bessy et al.'10]
- $O(k)$ vertex kernel [Bessy et al.'10]

Our result

a simpler proof of an $O(k)$ vertex kenel for k-FAST

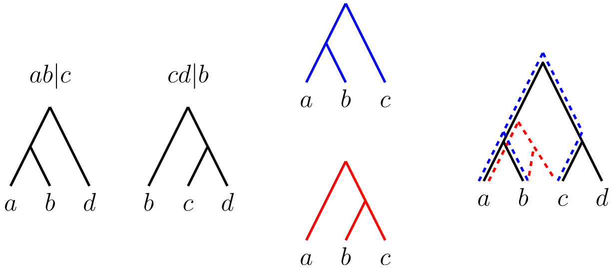
Dense Rooted Triplet Inconsistency (dense RTI)

- Input: A dense set of rooted triplets $\mathcal{R}$ on a set L of leaves and a parameter k
- Question: Does $\mathcal{R}$ contain a subset $\mathcal{R}'$ of consistent rooted triplets such that $|\mathcal{R}'| \geqslant |\mathcal{R}| - k$?



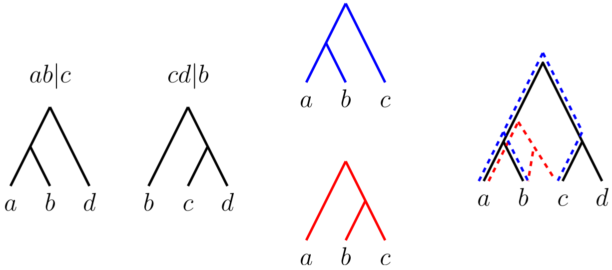
Dense Rooted Triplet Inconsistency (dense RTI)

- Input: A dense set of rooted triplets $\mathcal{R}$ on a set L of leaves and a parameter k
- Question: Does $\mathcal{R}$ contain a subset $\mathcal{R}'$ of consistent rooted triplets such that $|\mathcal{R}'| \geqslant |\mathcal{R}| - k$?



Dense Rooted Triplet Inconsistency (dense RTI)

- Input: A dense set of rooted triplets $\mathcal{R}$ on a set L of leaves and a parameter k
- Question: Does $\mathcal{R}$ contain a subset $\mathcal{R}'$ of consistent rooted triplets such that $|\mathcal{R}'| \geqslant |\mathcal{R}| - k$?



$\mathcal{R}$ is dense if it contains a rooted triplet for every $\{a, b, c\} \in L^3$

Known Results

- NP-Complete [Birka et al.'08]
- dual parameterization is known as Maximum Rooted Triplet Consistency [Birka et al.'08]
- FPT [Guillemot, Berry'07]
- subexponential FPT [Guillemot, Mnich'10]
- $O(k^2)$ kernel [Guillemot, Mnich'10]

Known Results

- NP-Complete [Birka et al.'08]
- dual parameterization is known as Maximum Rooted Triplet Consistency [Birka et al.'08]
- FPT [Guillemot, Berry'07]
- subexponential FPT [Guillemot, Mnich'10]
- $O(k^2)$ kernel [Guillemot, Mnich'10]

Our result

An $O(k)$ kernel for k-Dense RTI

Known Results

- NP-Complete [Birka et al.'08]
- dual parameterization is known as Maximum Rooted Triplet Consistency [Birka et al.'08]
- FPT [Guillemot, Berry'07]
- subexponential FPT [Guillemot, Mnich'10]
- $O(k^2)$ kernel [Guillemot, Mnich'10]

Our result

An $O(k)$ kernel for k-Dense RTI

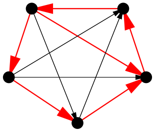
- Open : No constant approximation is known

Kernels for FAST

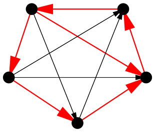
- An elementary quadratic kernel
- A subquadratic kernel
- A matching based linear kernel

Linear kernel for DENSE-RTI

Theorem: A tournament is acyclic (or transitive) iff it contains no (directed) triangle

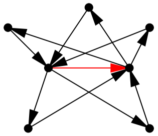


Theorem: A tournament is acyclic (or transitive) iff it contains no (directed) triangle

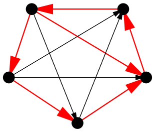


Rule 1 [irrelevant vertex] If a vertex v is not contained in any triangle, then delete v

Rule 2 [sunflower] If there is an arc belonging to more that k distinct triangles, then reverse it and decrease k by 1



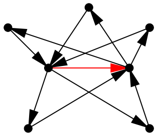
Theorem: A tournament is acyclic (or transitive) iff it contains no (directed) triangle



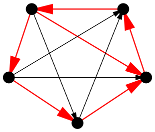
Rule 1 [irrelevant vertex] If a vertex v is not contained in any triangle, then delete v

A reduced tournament contains no source nor sink

Rule 2 [sunflower] If there is an arc belonging to more that k distinct triangles, then reverse it and decrease k by 1



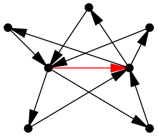
Theorem: A tournament is acyclic (or transitive) iff it contains no (directed) triangle



Rule 1 [irrelevant vertex] If a vertex v is not contained in any triangle, then delete v

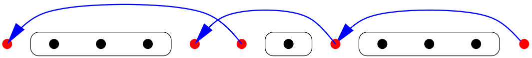
A reduced tournament contains no source nor sink

Rule 2 [sunflower] If there is an arc belonging to more that k distinct triangles, then reverse it and decrease k by 1



Observation: Rule 1 + Rule 2 yield a quadratic kernel

k-FAST $\equiv$ find a vertex ordering σ with at most k backward arcs

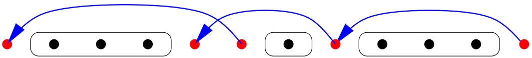


Let $\overrightarrow{uv}$ be a backward arc, then

$$span(\overrightarrow{uv}) = |\{w \in V : u <_\sigma w <_\sigma v\}|$$

Observation : Let $\overrightarrow{uv}$ be a backward arc of a reduced tournament.
By the sunflower rule, $|span(\overrightarrow{uv})| \leqslant 2k + 2$

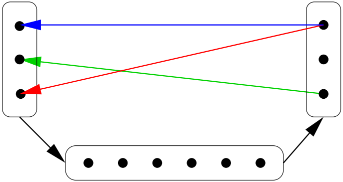
k-FAST $\equiv$ find a vertex ordering σ with at most k backward arcs



Let $\overrightarrow{uv}$ be a backward arc, then

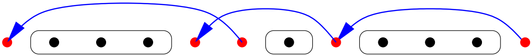
$$span(\overrightarrow{uv}) = |\{w \in V : u <_\sigma w <_\sigma v\}|$$

Observation : Let $\overrightarrow{uv}$ be a backward arc of a reduced tournament. By the sunflower rule, $|span(\overrightarrow{uv})| \leqslant 2k + 2$



Rule 3 [acyclic module] Let M be a maximal acyclic module. If there are at most $p = |M|$ arcs from $N^+(M)$ to $N^-(M)$, then reverse all these arcs and decrease k by p.

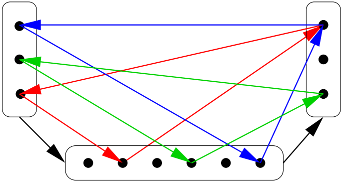
k-FAST $\equiv$ find a vertex ordering σ with at most k backward arcs



Let $\overrightarrow{uv}$ be a backward arc, then

$$span(\overrightarrow{uv}) = |\{w \in V : u <_\sigma w <_\sigma v\}|$$

Observation : Let $\overrightarrow{uv}$ be a backward arc of a reduced tournament.
By the sunflower rule, $|span(\overrightarrow{uv})| \leqslant 2k+2$

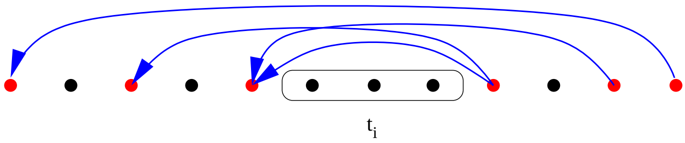


Rule 3 [acyclic module] Let M be a maximal acyclic module.
If there are at most $p = |M|$ arcs from $N^+(M)$ to $N^-(M)$,
then reverse all these arcs and decrease k by p.

Observation : at most $2k$ vertices are incident to a backward arc

By the acyclic module Rule $\Rightarrow$

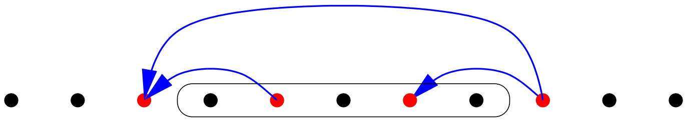
$$\sum t_i^2 \leqslant \sum s(\overrightarrow{uv})$$



Observation : at most $2k$ vertices are incident to a backward arc

By the sunflower Rule $\Rightarrow$

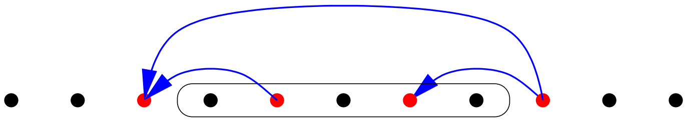
$$\sum t_i^2 \leqslant \sum s(\overrightarrow{uv}) \leqslant k(2k+2)$$



Observation : at most $2k$ vertices are incident to a backward arc

By the sunflower Rule $\Rightarrow$

$$\sum t_i^2 \quad \leqslant \quad \sum s(\overrightarrow{uv}) \quad \leqslant \quad k(2k+2)$$

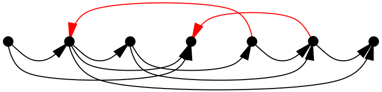


Theorem [Bessy et al.'10]

k-FAST has a vertex kernel of size $2k + \sum t_i = O(k\sqrt{k})$.

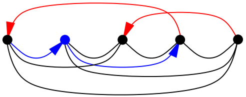
A LINEAR KERNEL FOR FAST

We denote by $T_\sigma = (V, A, \sigma)$ a tournament equipped with an ordering σ on V



A LINEAR KERNEL FOR FAST

We denote by $T_\sigma = (V, A, \sigma)$ a tournament equipped with an ordering σ on V



Let $\overrightarrow{uv}$ be a backward arc of T_σ and let $w \in span(\overrightarrow{uv})$ a vertex not incident to any backward arc, then

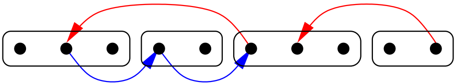
$c(\overrightarrow{uv}) = \{u, w, v\}$ is a $\overrightarrow{uv}$-certificate

Let F be a set of backward arcs, a F-certificate is a set

$c(F) = \{c(\overrightarrow{uv}) \mid \overrightarrow{uv} \in F\}$ of arc-disjoint certificates.

A linear kernel for FAST

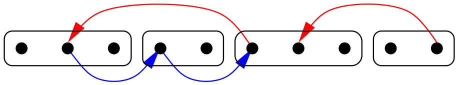
Let $\mathcal{P} = \{V_1, \dots V_l\}$ be an σ-partition of a $T_\sigma = (V, A, \sigma)$



- Denote the external arcs by $A_E = \{\overrightarrow{uv} \mid u \in V_i, v \in V_j, i \neq j\}$
- Denote the internal arcs by $A_I = A \setminus A_E$

A linear kernel for FAST

Let $\mathcal{P} = \{V_1, \ldots V_l\}$ be an σ-partition of a $T_\sigma = (V, A, \sigma)$



- Denote the external arcs by $A_E = \{\overrightarrow{uv} \mid u \in V_i, v \in V_j, i \neq j\}$
- Denote the internal arcs by $A_I = A \setminus A_E$

A σ-partition is safe if the backward arcs of A_E can be certified within A_E

Rule 4 [safe partition] Let $\mathcal{P}$ be a safe σ-partition of T_σ. Then reverse every external backward arc and decrement k accordingly

A linear kernel for FAST

- How to compute in polynomial time a safe partition?
- Prove that the irrelevant vertex rule and the safe partition rule yields a $4k$-kernel

A LINEAR KERNEL FOR FAST

- How to compute in polynomial time a safe partition?
- Prove that the irrelevant vertex rule and the safe partition rule yields a $4k$-kernel

A conflict packing is a maximal set $\mathcal{C}$ of arc-disjoint certificates. We denote by $V(\mathcal{C})$ the vertices covered by $\mathcal{C}$.

Lemma 1: If $\mathcal{C}$ is a conflict packing of a positive instance (T, k) of FAST, then $|V(\mathcal{C})| \leqslant 3k$

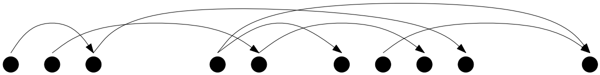
A linear kernel for FAST

Lemma 2: If $\mathcal{C}$ is a conflict packing of (T, k), then $\exists$ σ such that if $\overrightarrow{uv}$ is backward arc of T_σ, then $\{u, v\} \subseteq V(\mathcal{C})$

A linear kernel for FAST

Lemma 2: If $\mathcal{C}$ is a conflict packing of (T, k), then $\exists\ \sigma$ such that if $\overrightarrow{uv}$ is backward arc of T_σ, then $\{u, v\} \subseteq V(\mathcal{C})$

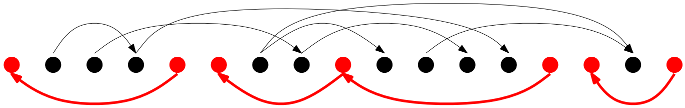
- Let $\mathcal{C}$ be a conflict packing of T. Then $T' = T - V(\mathcal{C})$ is transitive. Let π be a transitive ordering of $V \setminus V(\mathcal{C})$.



A LINEAR KERNEL FOR FAST

Lemma 2: If $\mathcal{C}$ is a conflict packing of (T, k), then $\exists\ \sigma$ such that if $\overrightarrow{uv}$ is backward arc of T_σ, then $\{u, v\} \subseteq V(\mathcal{C})$

- Let $\mathcal{C}$ be a conflict packing of T. Then $T' = T - V(\mathcal{C})$ is transitive. Let π be a transitive ordering of $V \setminus V(\mathcal{C})$.

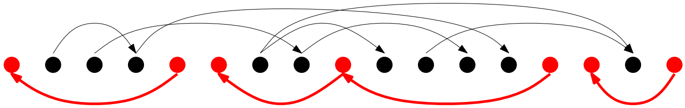


- Moreover for every vertex $x \in V(\mathcal{C})$, $T' + \{x\}$ is transitive.
- So x has a unique insertion location in π

A linear kernel for FAST

Lemma 2: If $\mathcal{C}$ is a conflict packing of (T, k), then $\exists\ \sigma$ such that if $\overrightarrow{uv}$ is backward arc of T_σ, then $\{u, v\} \subseteq V(\mathcal{C})$

- Let $\mathcal{C}$ be a conflict packing of T. Then $T' = T - V(\mathcal{C})$ is transitive. Let π be a transitive ordering of $V \setminus V(\mathcal{C})$.



- Moreover for every vertex $x \in V(\mathcal{C})$, $T' + \{x\}$ is transitive.
- So x has a unique insertion location in π

- The ordering σ is obtained by inserting every vertex of $V(\mathcal{C})$ at his location (vertices at the same location are arbitrarily ordered)

A linear kernel for FAST

Let us denote by A_B the set of backward arcs of T_σ.

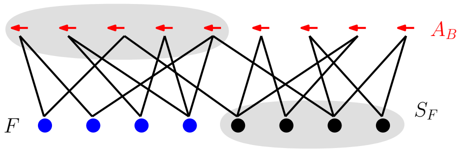
Let $\mathcal{C}$ be a conflict packing, let us denote by $F = V \setminus V(\mathcal{C})$ the set of free vertices

A linear kernel for FAST

Let us denote by A_B the set of backward arcs of T_σ.

Let $\mathcal{C}$ be a conflict packing, let us denote by $F = V \setminus V(\mathcal{C})$ the set of free vertices

- Let $B = (A_B, F, E)$ be the bipartite graph such that
  $(\overrightarrow{uv}, w) \in E \Leftrightarrow \{u, v, w\}$ is a conflict / triangle in T

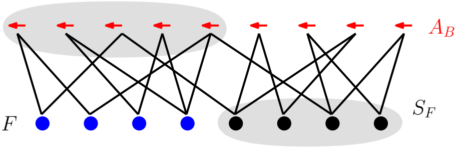


A linear kernel for FAST

Let us denote by A_B the set of backward arcs of T_σ.

Let $\mathcal{C}$ be a conflict packing, let us denote by $F = V \setminus V(\mathcal{C})$ the set of free vertices

- Let $B = (A_B, F, E)$ be the bipartite graph such that $(\overrightarrow{uv}, w) \in E \Leftrightarrow \{u, v, w\}$ is a conflict / triangle in T



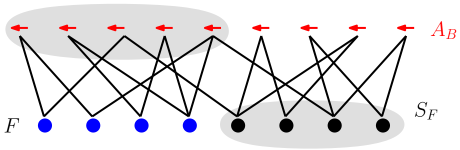
- Let S be a vertex cover of B and set $S_F = F \cap S$

A linear kernel for FAST

Let us denote by A_B the set of backward arcs of T_σ.

Let $\mathcal{C}$ be a conflict packing, let us denote by $F = V \setminus V(\mathcal{C})$ the set of free vertices

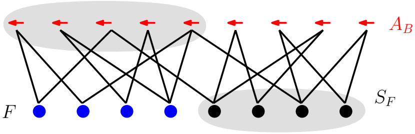
- Let $B = (A_B, F, E)$ be the bipartite graph such that $(\overrightarrow{uv}, w) \in E \Leftrightarrow \{u, v, w\}$ is a conflict / triangle in T

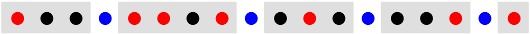


- Let S be a vertex cover of B and set $S_F = F \cap S$
- Let $\mathcal{P}$ be the σ-partition containing a part
  - $\{v\}$ for every vertex $v \notin S_F$
  - for every maximal consecutive (in σ) subset of $V(\mathcal{C}) \cup S_F$

A linear kernel for FAST

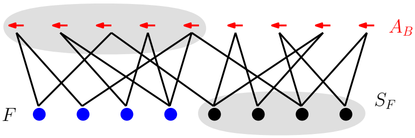
Lemma : If (T, k) is a positive instance of k-FAST of size at least $4k$ and without any irrelevant vertex, then $\mathcal{P}$ is a safe σ-partition containing at least one external backward arc.

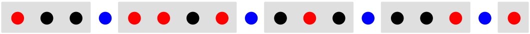




A linear kernel for FAST

Lemma : If (T, k) is a positive instance of k-FAST of size at least $4k$ and without any irrelevant vertex, then $\mathcal{P}$ is a safe σ-partition containing at least one external backward arc.

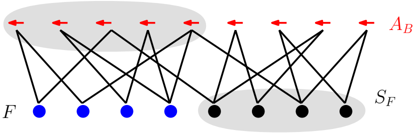


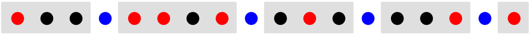


- a maximum matching $\mathcal{M}$ in B is a conflict packing
- $|\mathcal{M}| = |S| \leqslant k$, where S is a vertex cover
- as $|V(\mathcal{C})| \leqslant 3k$, $L \setminus S_F \neq \emptyset$ and $\mathcal{P}$ is non-trivial

A linear kernel for FAST

Lemma : If (T, k) is a positive instance of k-FAST of size at least $4k$ and without any irrelevant vertex, then $\mathcal{P}$ is a safe σ-partition containing at least one external backward arc.





Claim : the external backward arcs of A_B can be externally certified.

- By König-Ergevary Lemma, there is a matching $\mathcal{M}_E$ between A_B and $F \setminus S_F$ saturating $F \setminus S_F$.
- $\mathcal{M}_E$ provides the external certificate

A linear kernel for FAST

Theorem [P., Perez, Thomassé]
FAST parameterized by the solution size k has a $4k$ vertex kernel.

A linear kernel for dense-RTI

Let $\mathcal{R}$ be a set of rooted triplets on L.

Question : Can we remove at most k rooted triplets from $\mathcal{R}$ to obtain a consistent set of triplets?

A linear kernel for dense-RTI

Let $\mathcal{R}$ be a set of rooted triplets on L.

Question : Can we remove at most k rooted triplets from $\mathcal{R}$ to obtain a consistent set of triplets?

Or equiv. : can we find a binary tree hosting all but at most k rooted triplets of $\mathcal{R}$?

- we need to define what is a conflict and a certificate

A linear kernel for dense-RTI

Let $\mathcal{R}$ be a set of rooted triplets on L.

Question : Can we remove at most k rooted triplets from $\mathcal{R}$ to obtain a consistent set of triplets?

Or equiv. : can we find a binary tree hosting all but at most k rooted triplets of $\mathcal{R}$?

- we need to define what is a conflict and a certificate
- Instead of an ordering σ (as in FAST), we seek for a tree
- instead of a σ-partition, we need to define a tree-partition

A linear kernel for dense-RTI

Let $\mathcal{R}$ be a dense set of rooted triplets on L.

A conflict is a subset L' of leaves such that $\mathcal{R}/L'$ is inconsistent.
(i.e. no tree can host the rooted triplets of $\mathcal{R}/L'$)

A LINEAR KERNEL FOR DENSE-RTI

Let $\mathcal{R}$ be a dense set of rooted triplets on L.

A conflict is a subset L' of leaves such that $\mathcal{R}/L'$ is inconsistent.
(i.e. no tree can host the rooted triplets of $\mathcal{R}/L'$)

Lemma : A set $\mathcal{R}$ of rooted triplets is consistent

- iff $\mathcal{R}$ contains no conflict on four leaves
- iff $\mathcal{R}$ contains no conflict of the form

$\{ab|c,\ cd|b,\ bd|a\}$ or $\{ab|c,\ cd|b,\ ad|b\}$

A linear kernel for dense-RTI

Let $\mathcal{R}$ be a dense set of rooted triplets on L.

A conflict is a subset L' of leaves such that $\mathcal{R}/L'$ is inconsistent.
(i.e. no tree can host the rooted triplets of $\mathcal{R}/L'$)

Lemma : A set $\mathcal{R}$ of rooted triplets is consistent

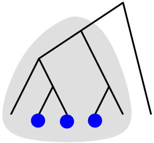
- iff $\mathcal{R}$ contains no conflict on four leaves
- iff $\mathcal{R}$ contains no conflict of the form

$\{ab|c,\ cd|b,\ bd|a\}$ or $\{ab|c,\ cd|b,\ ad|b\}$

$\longrightarrow$ it is enough to focus on subsets of four leaves
(we used triangles in FAST)

A linear kernel for dense-RTI

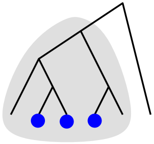
A tree-embedded instance $R_T = (\mathcal{R}, L, T)$ is formed by set of triplets $\mathcal{R}$ and a binary tree T on leaf set L.



$span_T(L') = \{\ell \in L \mid \ell \text{ is a leaf of } T_{LCA(L')}\}$

A LINEAR KERNEL FOR DENSE-RTI

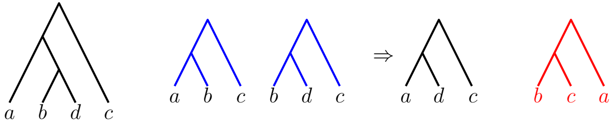
A tree-embedded instance $R_T = (\mathcal{R}, L, T)$ is formed by set of triplets $\mathcal{R}$ and a binary tree T on leaf set L.



$span_T(L') = \{\ell \in L \mid \ell \text{ is a leaf of } T_{LCA(L')}\}$

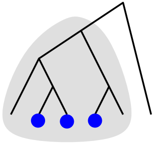
Lemma : Let $R_T = (\mathcal{R}, L, T)$ be an embedded instance. Let $t = bc|a$ be the unique inconsistent rooted triplet in $\mathcal{R}/\{a, b, c, d\}$. Then

$\{a, b, c, d\}$ is a conflict iff $d \in span(\{a, b, c\})$



A linear kernel for dense-RTI

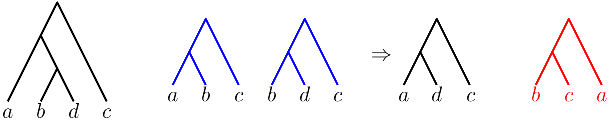
A tree-embedded instance $R_T = (\mathcal{R}, L, T)$ is formed by set of triplets $\mathcal{R}$ and a binary tree T on leaf set L.



$span_T(L') = \{\ell \in L \mid \ell \text{ is a leaf of } T_{LCA(L')}\}$

Lemma : Let $R_T = (\mathcal{R}, L, T)$ be an embedded instance. Let $t = bc|a$ be the unique inconsistent rooted triplet in $\mathcal{R}/\{a, b, c, d\}$. Then

$\{a, b, c, d\}$ is a conflict iff $d \in span(\{a, b, c\})$

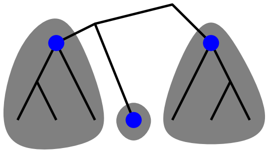


If $d \in span(\{a, b, c\})$ does not belong to any inconsistent rooted triplet with T, then $c_T(t) = \{a, b, c, d\}$ is a certificate of t.

A linear kernel for dense-RTI

$\mathcal{P} = \{T_1, \dots T_r\}$ is a tree-partition of T if T contains a set of nodes (or leaves) $x_1, \dots x_r$ such that

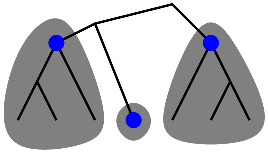
- for every $i \in [r]$, T_i is the subtree of T rooted at x_i
- L is partitionned by the sets of leaves of the T_i's



A linear kernel for dense-RTI

$\mathcal{P} = \{T_1, \ldots T_r\}$ is a tree-partition of T if T contains a set of nodes (or leaves) $x_1, \ldots x_r$ such that

- for every $i \in [r]$, T_i is the subtree of T rooted at x_i
- L is partitionned by the sets of leaves of the T_i's



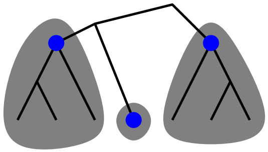
Observe that a tree-partition clearly distinguish the sets

$\mathcal{R}_E$ of external triplets and $\mathcal{R}_I$ of internal triplets

A linear kernel for dense-RTI

$\mathcal{P} = \{T_1, \ldots T_r\}$ is a tree-partition of T if T contains a set of nodes (or leaves) $x_1, \ldots x_r$ such that

- for every $i \in [r]$, T_i is the subtree of T rooted at x_i
- L is partitionned by the sets of leaves of the T_i's



Observe that a tree-partition clearly distinguish the sets

$\mathcal{R}_E$ of external triplets and $\mathcal{R}_I$ of internal triplets

A tree-partition is safe is is it possible to externally certify the set of inconstents rooted triplets of $\mathcal{R}_E$.

A linear kernel for dense-RTI

Rule 1 [irrelevant leaf] If a leaf $\ell \in L$ does not belong to any conflict, then remove ℓ.

Rule 2 [Safe partition] Let $\mathcal{P}_T$ be a tree-partition of $R_T = (\mathcal{R}, L, T)$. Then edit every inconsistent rooted triplet of $\mathcal{R}_E$ with respect to T and decrease k accordingly.

A linear kernel for dense-RTI

Rule 1 [irrelevant leaf] If a leaf $\ell \in L$ does not belong to any conflict, then remove ℓ.

Rule 2 [Safe partition] Let $\mathcal{P}_T$ be a tree-partition of $R_T = (\mathcal{R}, L, T)$. Then edit every inconsistent rooted triplet of $\mathcal{R}_E$ with respect to T and decrease k accordingly.

- How to compute in polynomial time a safe tree-partition ?

- Prove that these two rules yields a $5k$-leaf kernel.

A linear kernel for dense-RTI

Rule 1 [irrelevant leaf] If a leaf $\ell \in L$ does not belong to any conflict, then remove ℓ.

Rule 2 [Safe partition] Let $\mathcal{P}_T$ be a tree-partition of $R_T = (\mathcal{R}, L, T)$. Then edit every inconsistent rooted triplet of $\mathcal{R}_E$ with respect to T and decrease k accordingly.

- How to compute in polynomial time a safe tree-partition ?
  Use a greedy conflict packing (as in FAST).

- Prove that these two rules yields a $5k$-leaf kernel.
  Use a matching argument to show that the external inconsistent triplets can be externally certified.

A linear kernel for dense-RTI

Assuming $(R = (\mathcal{R}, L), k)$ is a YES-instance of size at least $5k$:

- a conflict packing $\mathcal{C}$ covers at most $4k$ leaves of L
- Compute in polynomial time a binary tree T such that every inconsistent rooted triplet t w.r.t. T satisfies $L(t) \subseteq L(\mathcal{C})$.

A linear kernel for dense-RTI

Assuming $(R = (\mathcal{R}, L), k)$ is a YES-instance of size at least $5k$:

- a conflict packing $\mathcal{C}$ covers at most $4k$ leaves of L
- Compute in polynomial time a binary tree T such that every inconsistent rooted triplet t w.r.t. T satisfies $L(t) \subseteq L(\mathcal{C})$.
- Build the bipartite graph B between inconsistent triplet and free leaves (non covered by $\mathcal{C}$) such that every edge represents a conflict.
- Use a vertex cover of B to construct a safe tree-partition.

A linear kernel for dense-RTI

Assuming $(R = (\mathcal{R}, L), k)$ is a YES-instance of size at least $5k$:

- a conflict packing $\mathcal{C}$ covers at most $4k$ leaves of L
- Compute in polynomial time a binary tree T such that every inconsistent rooted triplet t w.r.t. T satisfies $L(t) \subseteq L(\mathcal{C})$.
- Build the bipartite graph B between inconsistent triplet and free leaves (non covered by $\mathcal{C}$) such that every edge represents a conflict.
- Use a vertex cover of B to construct a safe tree-partition.

Theorem [P., Perez, Thomassé]
Dense-RTI paramaterized by the solution size k has a $5k$ leaf kernel.

We described

- a linear kernel for FEEDBACK ARC SET IN TOURNAMENTS
- a linear kernel for ROOTED TRIPLET INCONSISTENCY

Conflict packing also yields:

- a linear kernel for BETWEENESS IN TOURNAMENTS
- a quadratic kernel for FEEDBACK ARC SET IN BIPARTITE TOURNAMENTS

Thank you...