ETH zürich

Structural Parameterization of Steiner Tree Packing

Niko Hastrich[†], Kirill Simonov[‡]

ETH Zürich [†]
University of Bergen [‡]

Def. **S**teiner **T**ree **P**acking

Def. Steiner Tree Packing

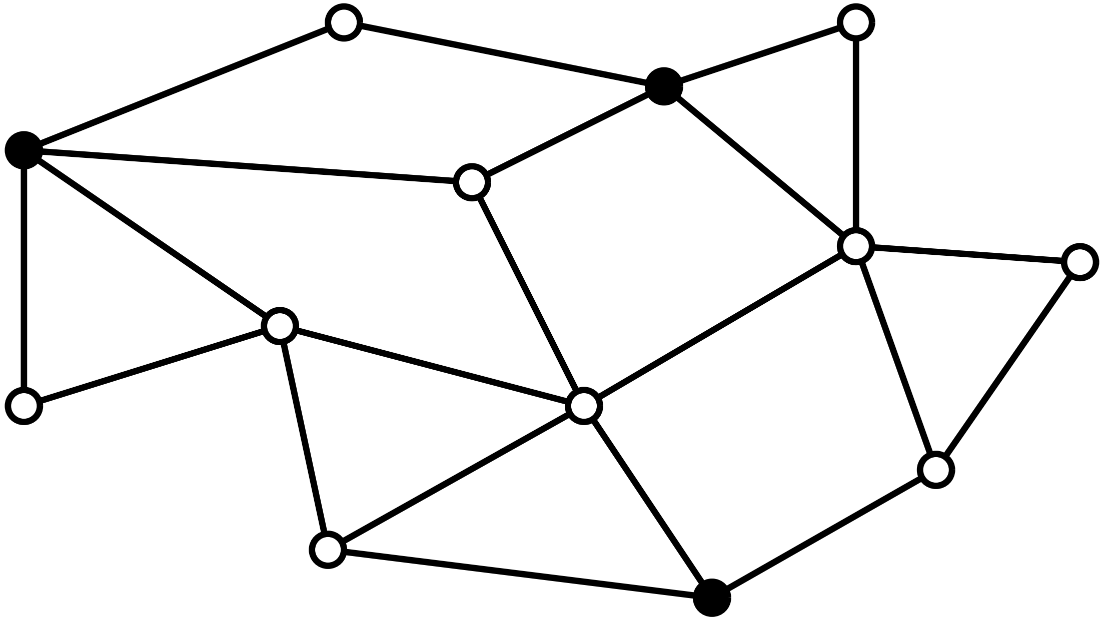
Input:

- Undirected Graph G
- Terminal set $T \subseteq V(G)$
- Demand $d \in \mathbb{N}^+$

Def. Steiner Tree Packing

Input:

- Undirected Graph G
- Terminal set $T \subseteq V(G)$
- Demand $d \in \mathbb{N}^+$

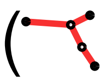
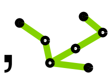


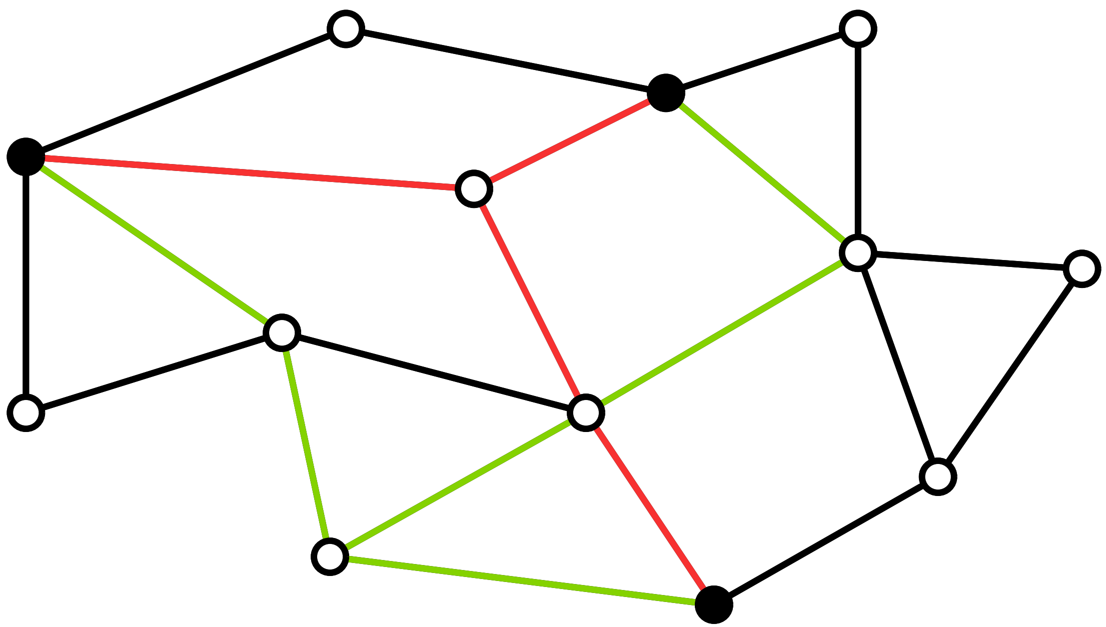
$d = 2$

Def. Steiner Tree Packing

Input:

- Undirected Graph G
- Terminal set $T \subseteq V(G)$
- Demand $d \in \mathbb{N}^+$

Output: Edge-disjoint, connected subgraphs (, ) connecting T



$d = 2$

Results for STP

Results for STP

Fact: Polynomial-time solvable if: $d = 1$ or $|T| = 2$

Results for STP

Fact: Polynomial-time solvable if: $d = 1$ or $|T| = 2$

Fact: NP-hard if: $d = 2$ or $|T| = 3$ [Aazami, Cheriyan, Jampani '12; Kaski '04]

Results for STP

Fact: Polynomial-time solvable if: $d = 1$ or $|T| = 2$

Fact: NP-hard if: $d = 2$ or $|T| = 3$ [Aazami, Cheriyan, Jampani '12; Kaski '04]

Fact: FPT by $|T| + d \rightarrow$ *highly* impractical [Robertson, Seymour '95]

Results for STP

Fact: Polynomial-time solvable if: $d = 1$ or $|T| = 2$

Fact: NP-hard if: $d = 2$ or $|T| = 3$ [Aazami, Cheriyan, Jampani '12; Kaski '04]

Fact: FPT by $|T| + d \rightarrow$ *highly* impractical [Robertson, Seymour '95]

Few results about its structural parameterization:

ETH zürich

Results for STP

Fact: Polynomial-time solvable if: $d = 1$ or $|T| = 2$

Fact: NP-hard if: $d = 2$ or $|T| = 3$ [Aazami, Cheriyan, Jampani '12; Kaski '04]

Fact: FPT by $|T| + d \rightarrow$ *highly* impractical [Robertson, Seymour '95]

Few results about its structural parameterization:

Fact: W$[1]$-hard by treewidth [Aazami, Cheriyan, Jampani '12; Bodlaender et al. '23]

Results for STP

Fact: Polynomial-time solvable if: $d = 1$ or $|T| = 2$

Fact: NP-hard if: $d = 2$ or $|T| = 3$ [Aazami, Cheriyan, Jampani ’12; Kaski ’04]

Fact: FPT by $|T| + d \rightarrow$ *highly* impractical [Robertson, Seymour ’95]

Few results about its structural parameterization:

Fact: W$[1]$-hard by treewidth [Aazami, Cheriyan, Jampani ’12; Bodlaender et al. ’23]

No known structural FPT algorithms.

Results for STP

Fact: Polynomial-time solvable if: $d = 1$ or $|T| = 2$

Fact: NP-hard if: $d = 2$ or $|T| = 3$ [Aazami, Cheriyan, Jampani '12; Kaski '04]

Fact: FPT by $|T| + d \rightarrow$ *highly* impractical [Robertson, Seymour '95]

Few results about its structural parameterization:

Fact: W$[1]$-hard by treewidth [Aazami, Cheriyan, Jampani '12; Bodlaender et al. '23]

No known structural FPT algorithms.

Theorem:
STP is FPT by tree-cut width as well as fracture number.

Generalized Steiner Tree Packing
GSTP

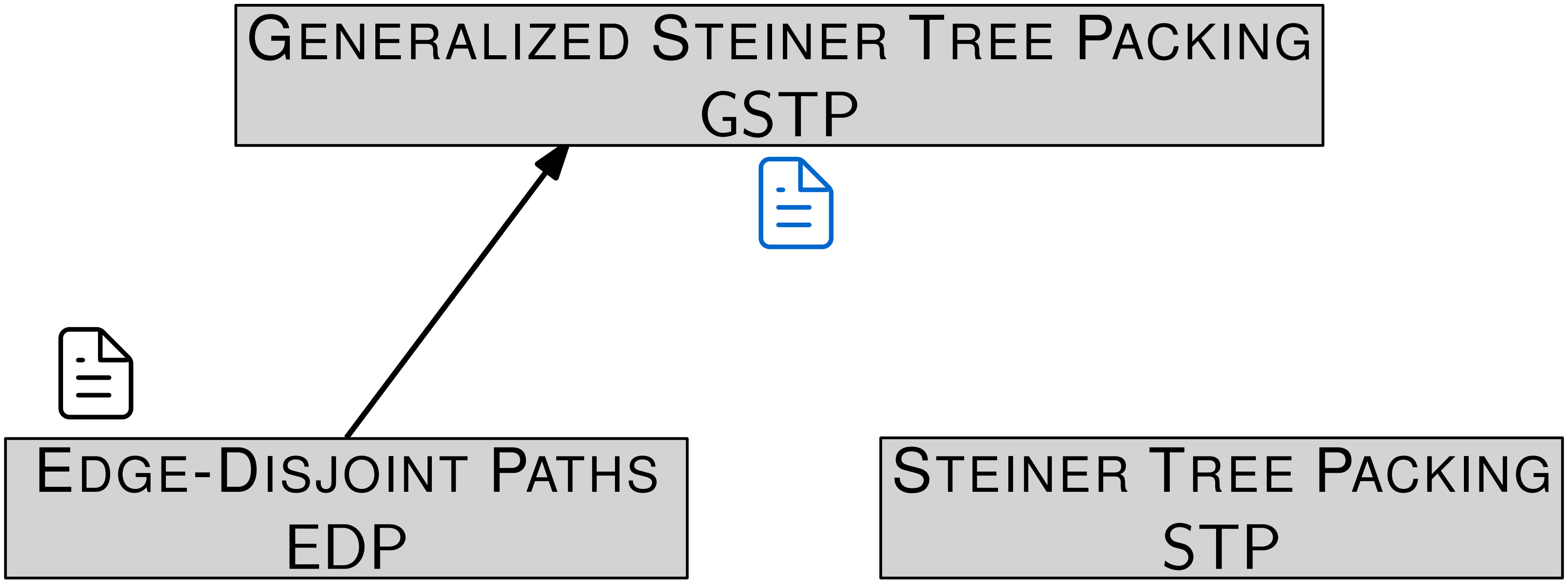
Edge-Disjoint Paths
EDP

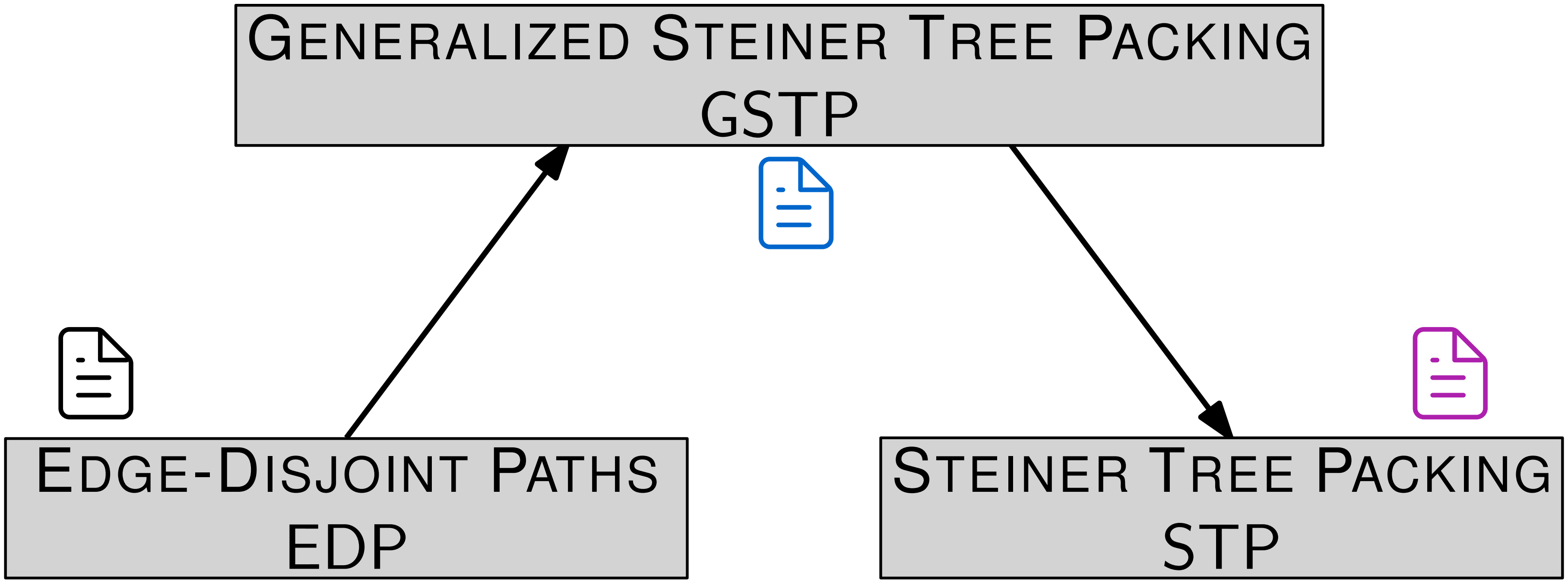
Steiner Tree Packing
STP

GENERALIZED STEINER TREE PACKING
GSTP

EDGE-DISJOINT PATHS
EDP

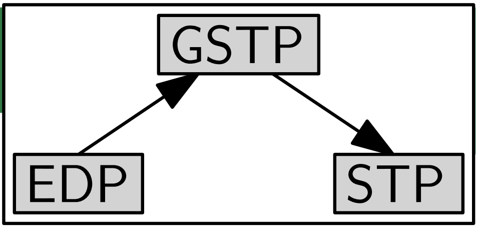
STEINER TREE PACKING
STP


GENERALIZED STEINER TREE PACKING
GSTP
EDGE-DISJOINT PATHS
EDP
STEINER TREE PACKING
STP



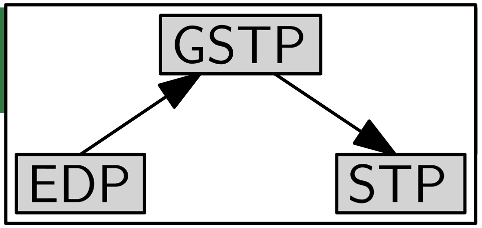
Generalized Steiner Tree Packing
GSTP
Edge-Disjoint Paths
EDP
Steiner Tree Packing
STP

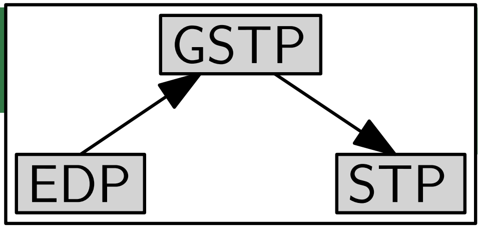
ETH zürich



ETH zürich

Def. **E**dge **D**isjoint **P**aths

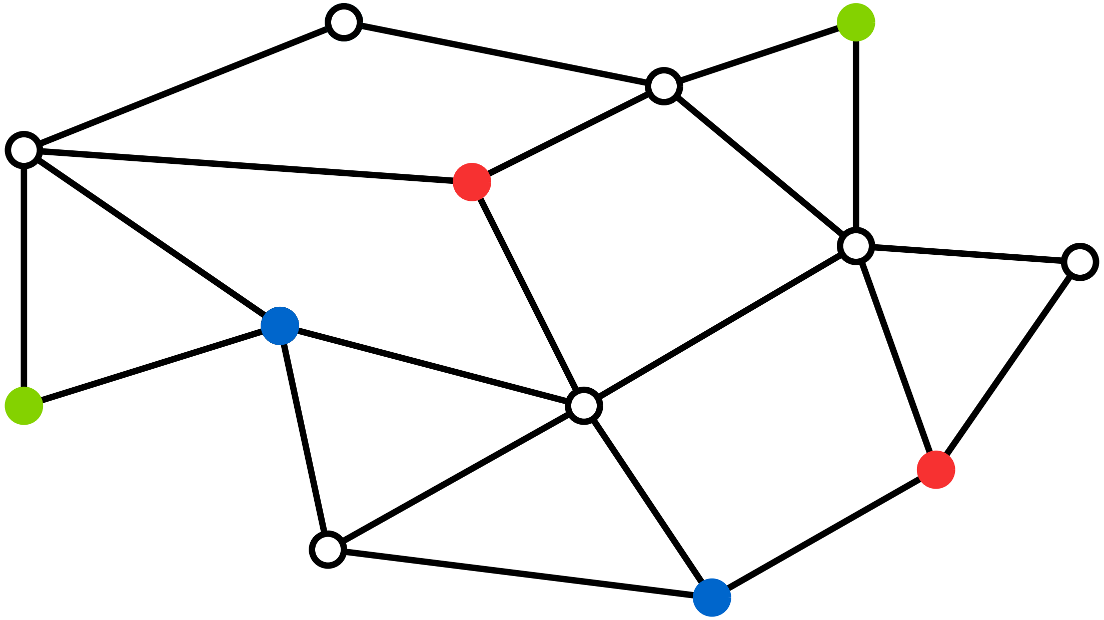


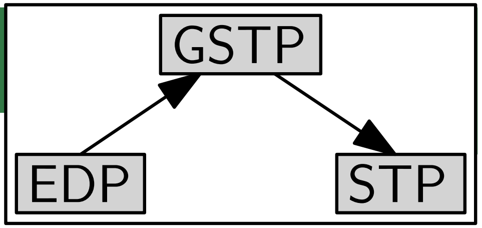


Def. **E**dge **D**isjoint **P**aths

Input:

- Undirected Graph G
- Terminal **pairs** $\mathcal{T} \subseteq \binom{V(G)}{2}$



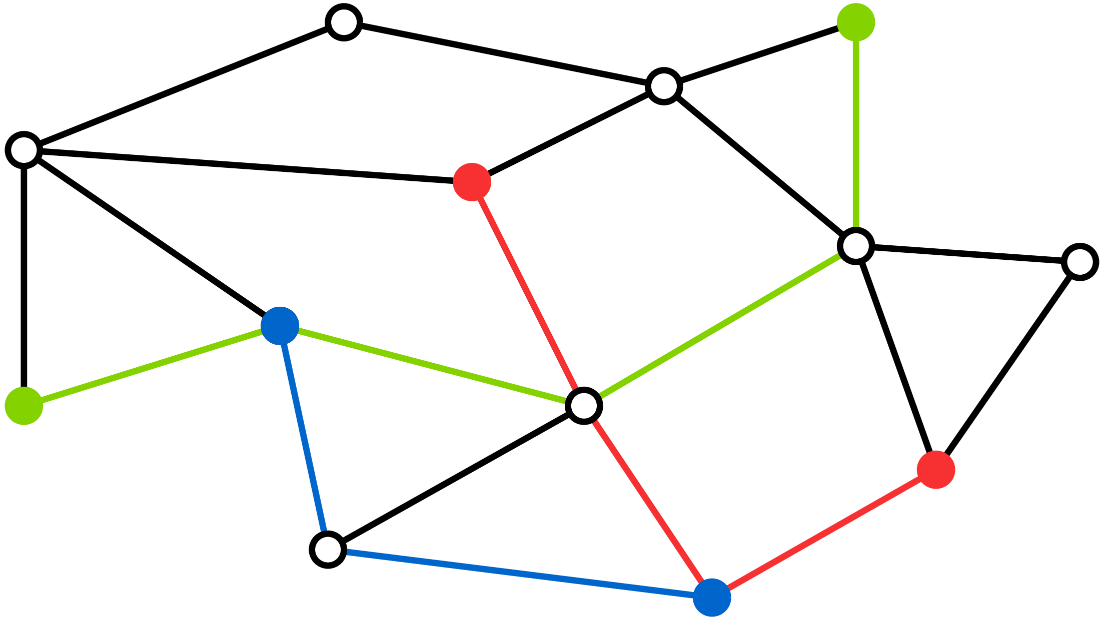


Def. **E**dge **D**isjoint **P**aths

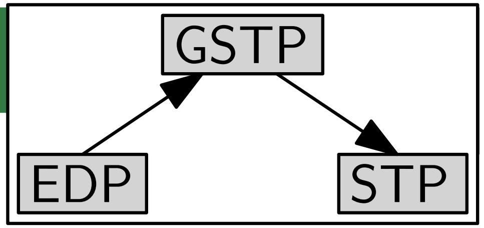
Input:

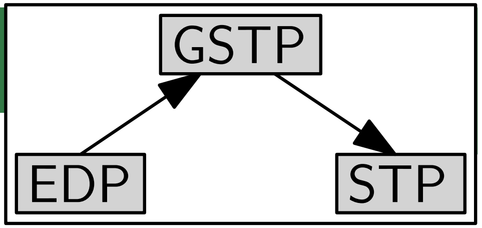
- Undirected Graph G
- Terminal **pairs** $\mathcal{T} \subseteq \binom{V(G)}{2}$

Output: Edge-disjoint **paths** (, ,) connecting **their terminal pair**



Def. **G**eneralized **S**teiner **T**ree **P**acking

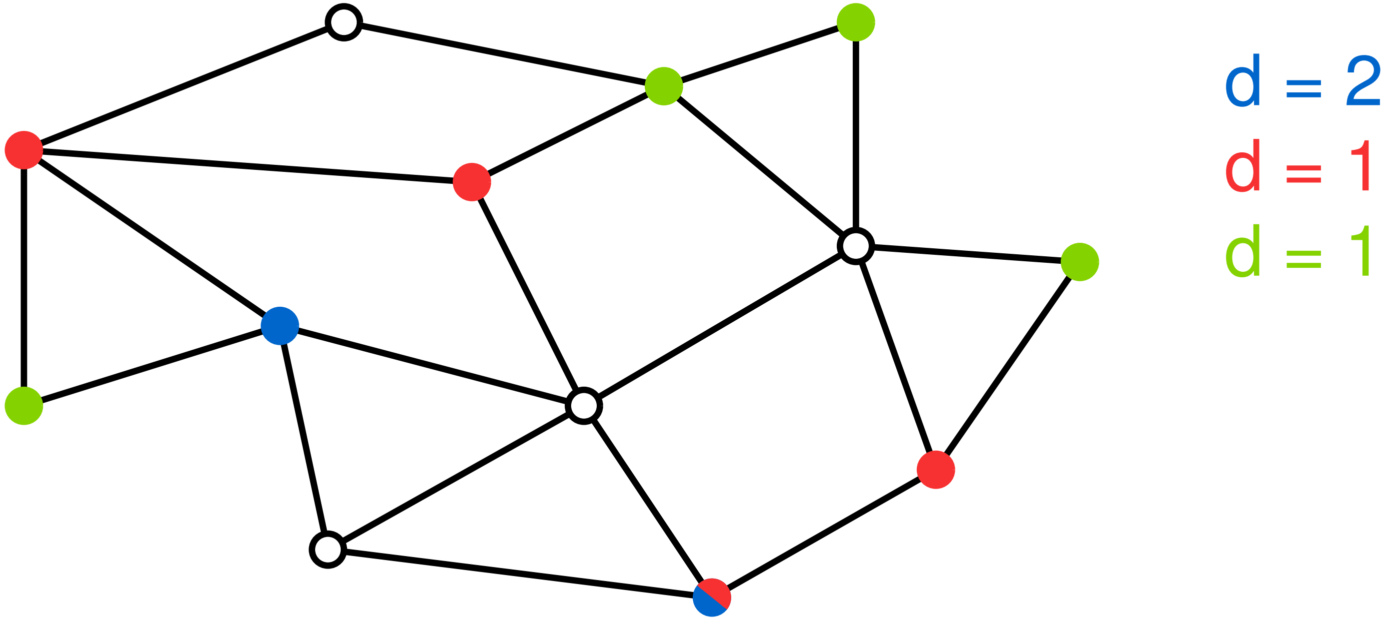


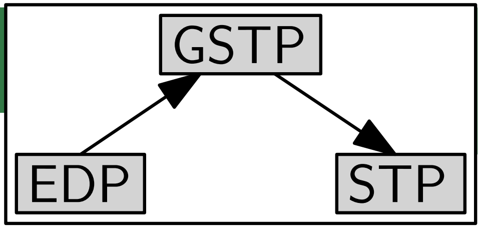


Def. Generalized Steiner Tree Packing

Input:

- Undirected Graph G
- Terminal sets $\mathcal{T} \subseteq 2^{V(G)}$
- Demands $d\colon \mathcal{T} \to \mathbb{N}^+$

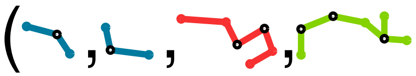


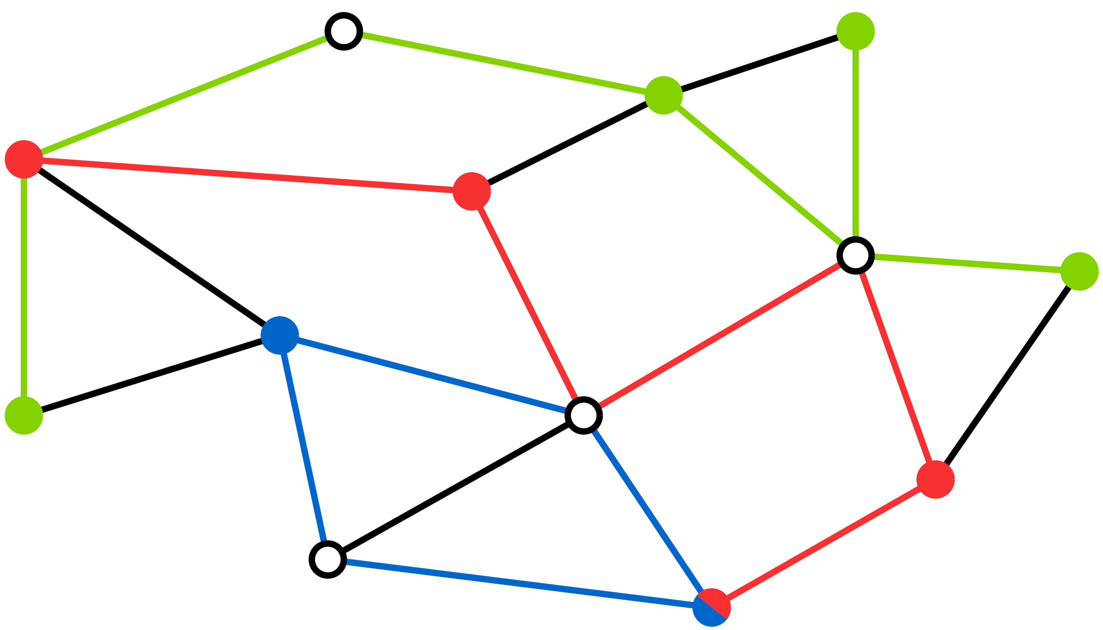


Def. **G**eneralized **S**teiner **T**ree **P**acking

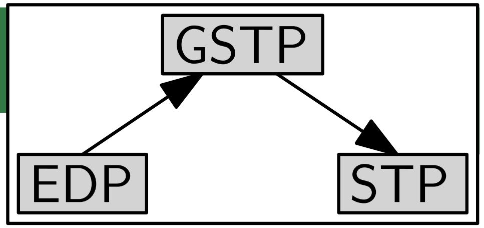
Input:

- Undirected Graph G
- Terminal sets $\mathcal{T} \subseteq 2^{V(G)}$
- Demands $d \colon \mathcal{T} \to \mathbb{N}^+$

Output: Edge-disjoint, connected subgraphs
() connecting **their terminal set**



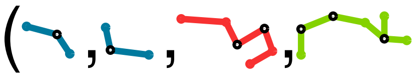
d = 2
d = 1
d = 1



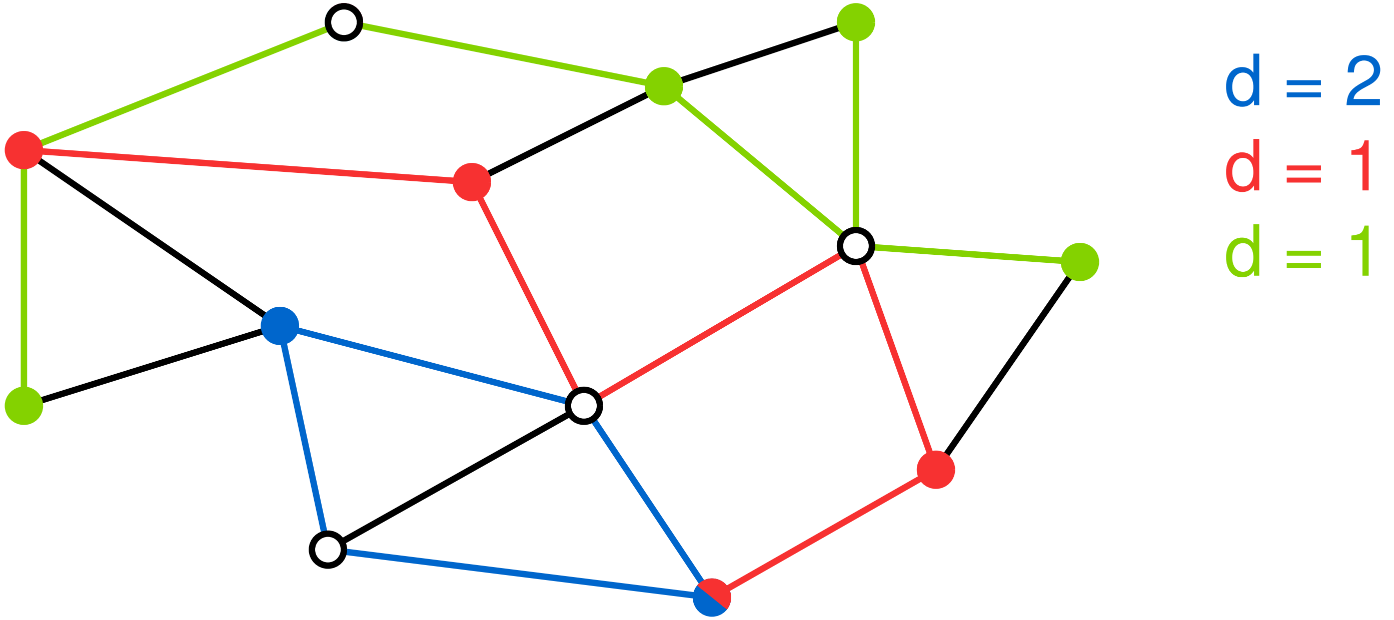
Def. Generalized Steiner Tree Packing

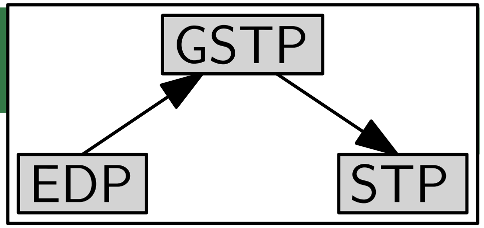
Input:

- Undirected Graph G
- Terminal sets $\mathcal{T} \subseteq 2^{V(G)}$
- Demands $d\colon \mathcal{T} \to \mathbb{N}^+$

Output: Edge-disjoint, connected subgraphs () connecting **their terminal set**

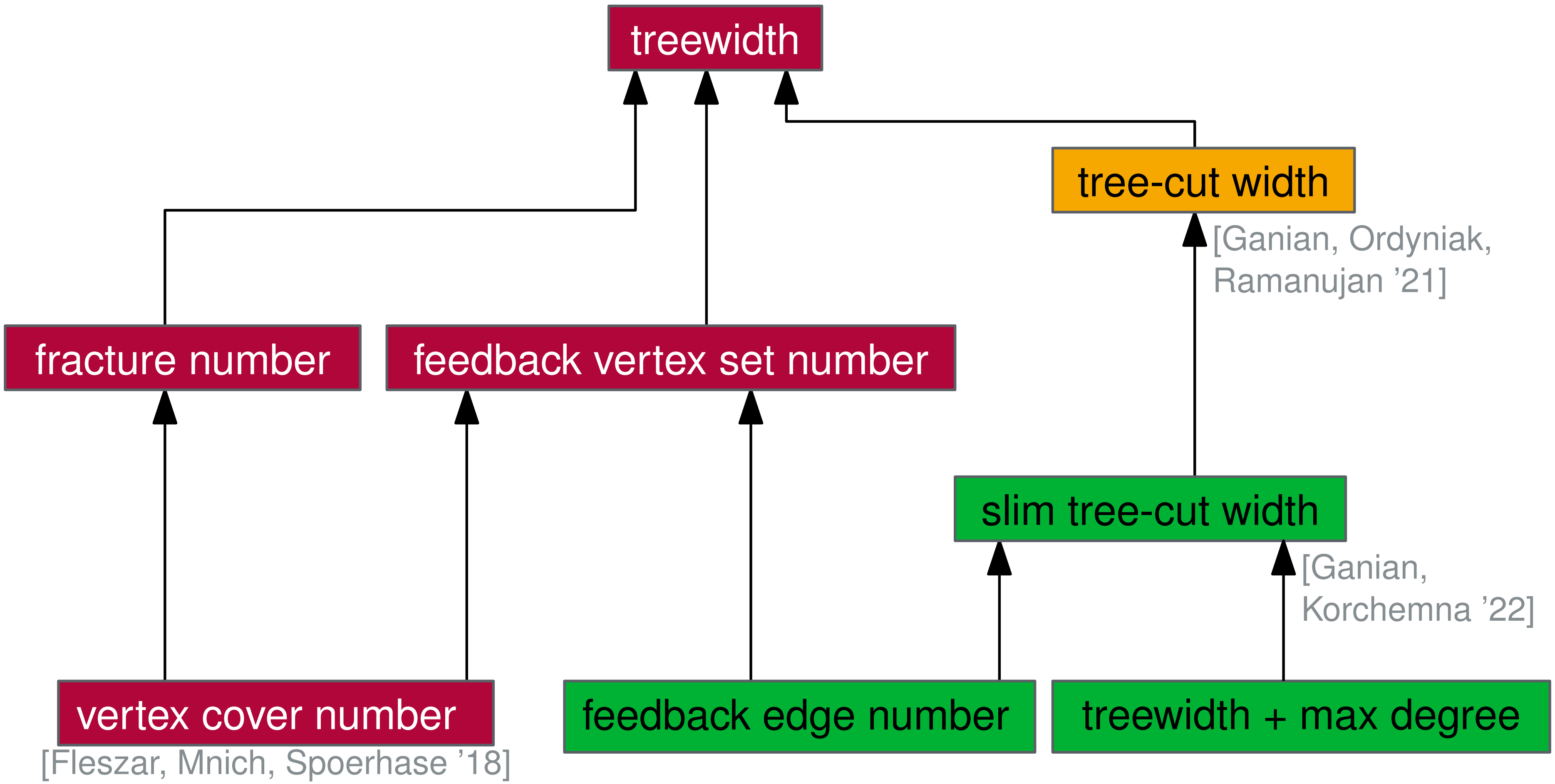
Contains EDP and STP as special cases.



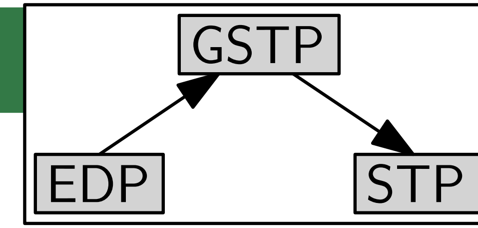


EDP by Input Graph

Known Results:

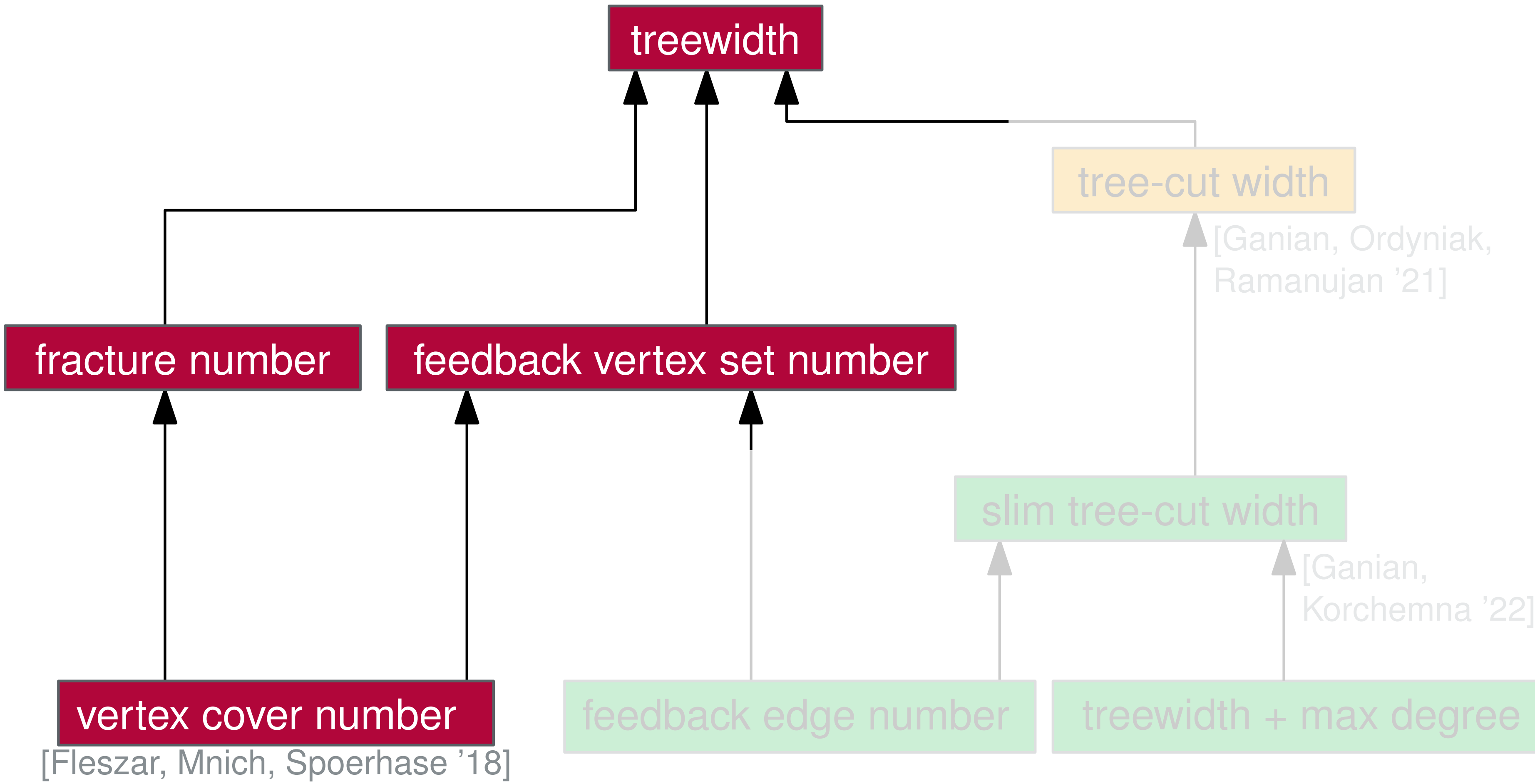


paraNP-hard W[1]-hard FPT

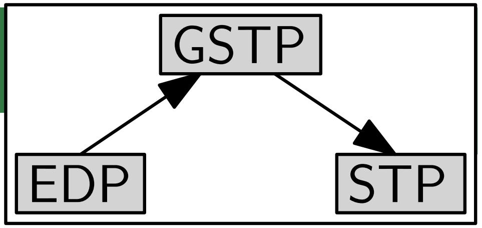


EDP by Input Graph

Known Results:

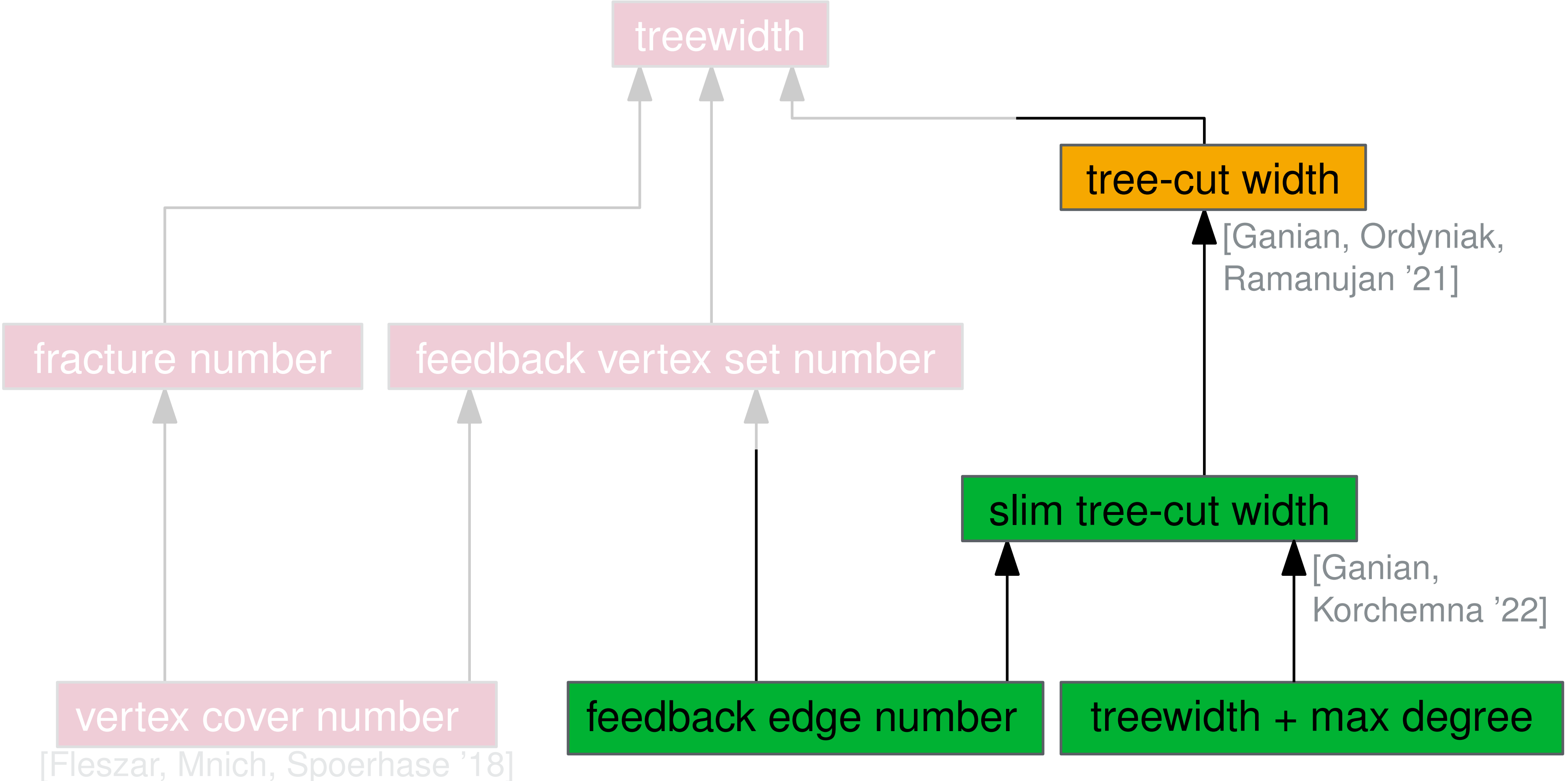


paraNP-hard W[1]-hard FPT

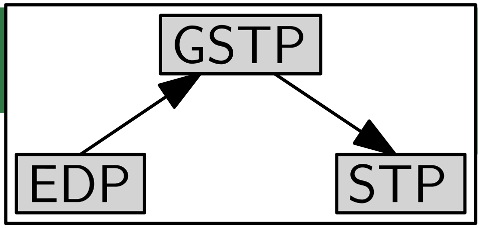


EDP by Input Graph

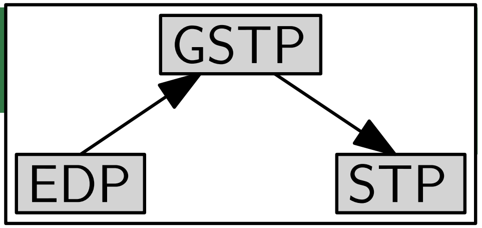
Known Results:



paraNP-hard W[1]-hard FPT

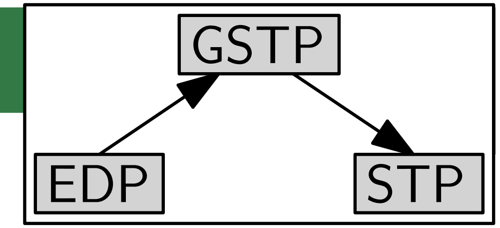


To get FPT Algorithms: Consider positions of terminal pairs.



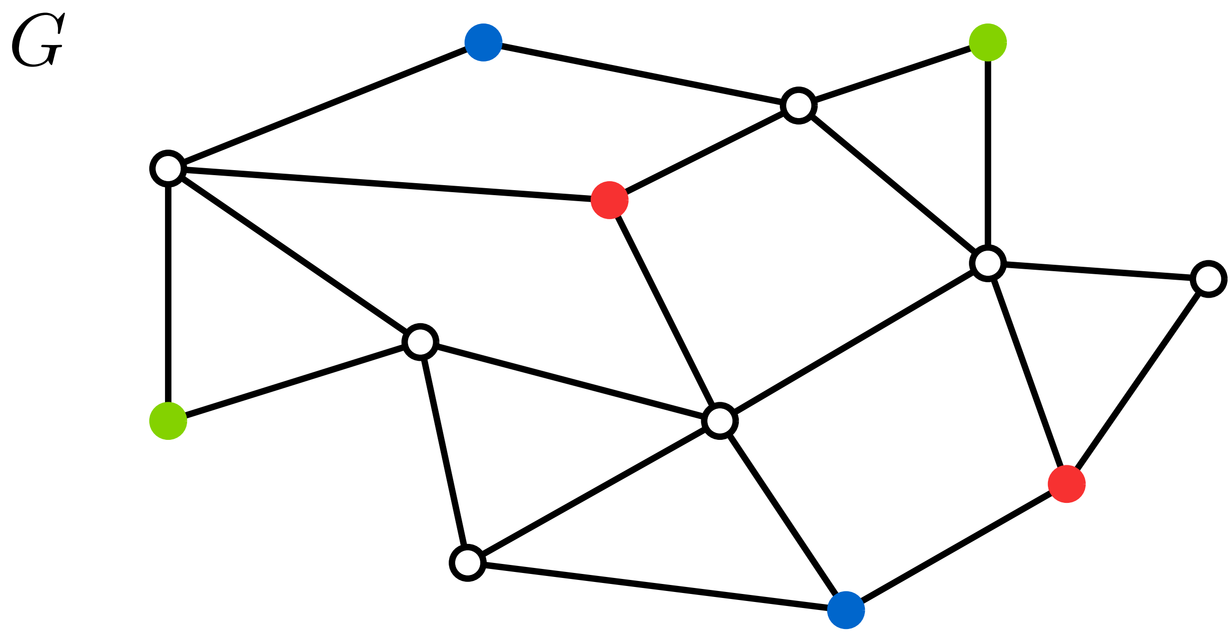
To get FPT Algorithms: Consider positions of terminal pairs.

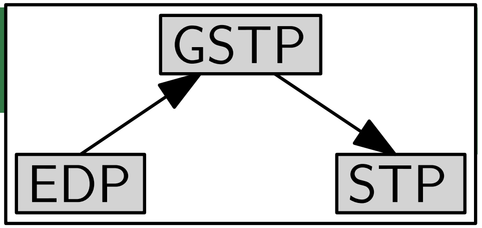
Parameterize by *augmented* graph $G^{\mathcal{T}} := G + \mathcal{T}$.



To get FPT Algorithms: Consider positions of terminal pairs.

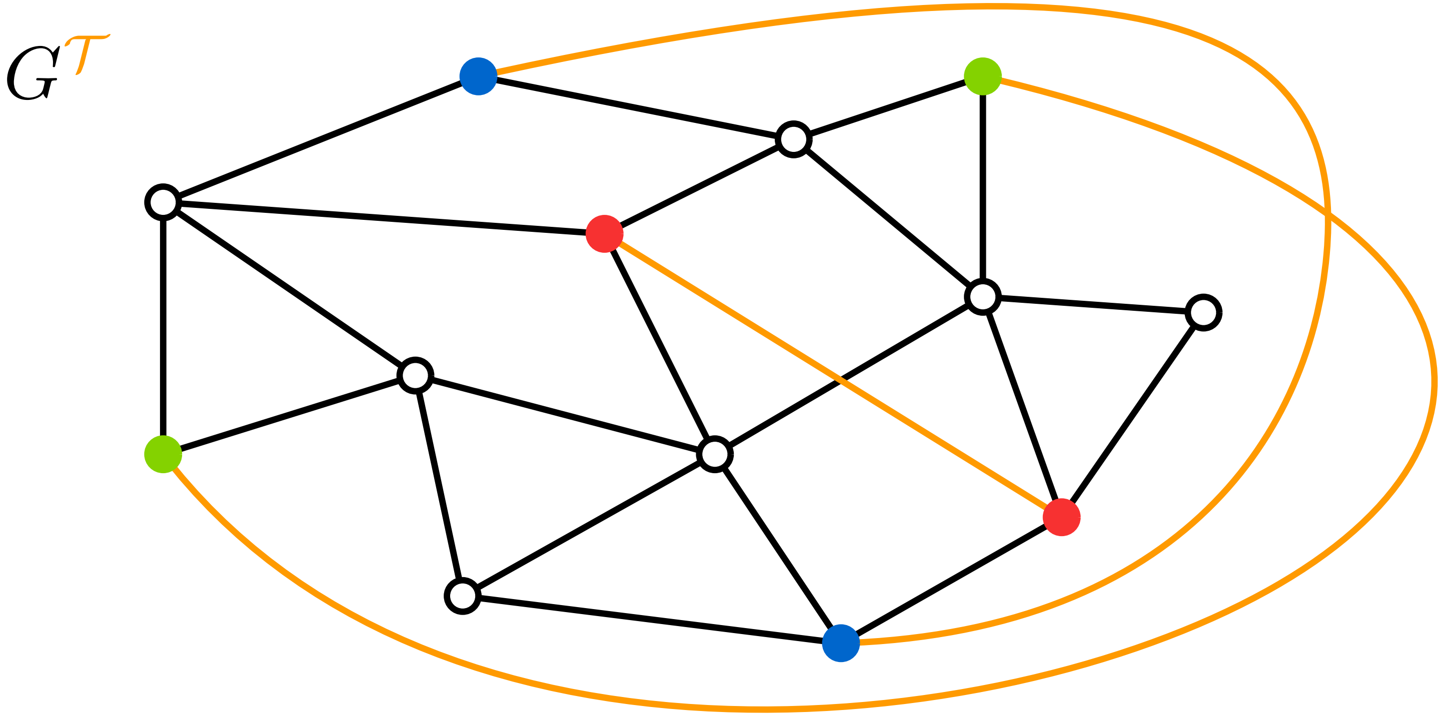
Parameterize by *augmented* graph $G^{\mathcal{T}} := G + \mathcal{T}$.

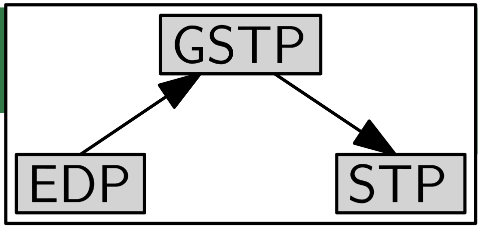




To get FPT Algorithms: Consider positions of terminal pairs.

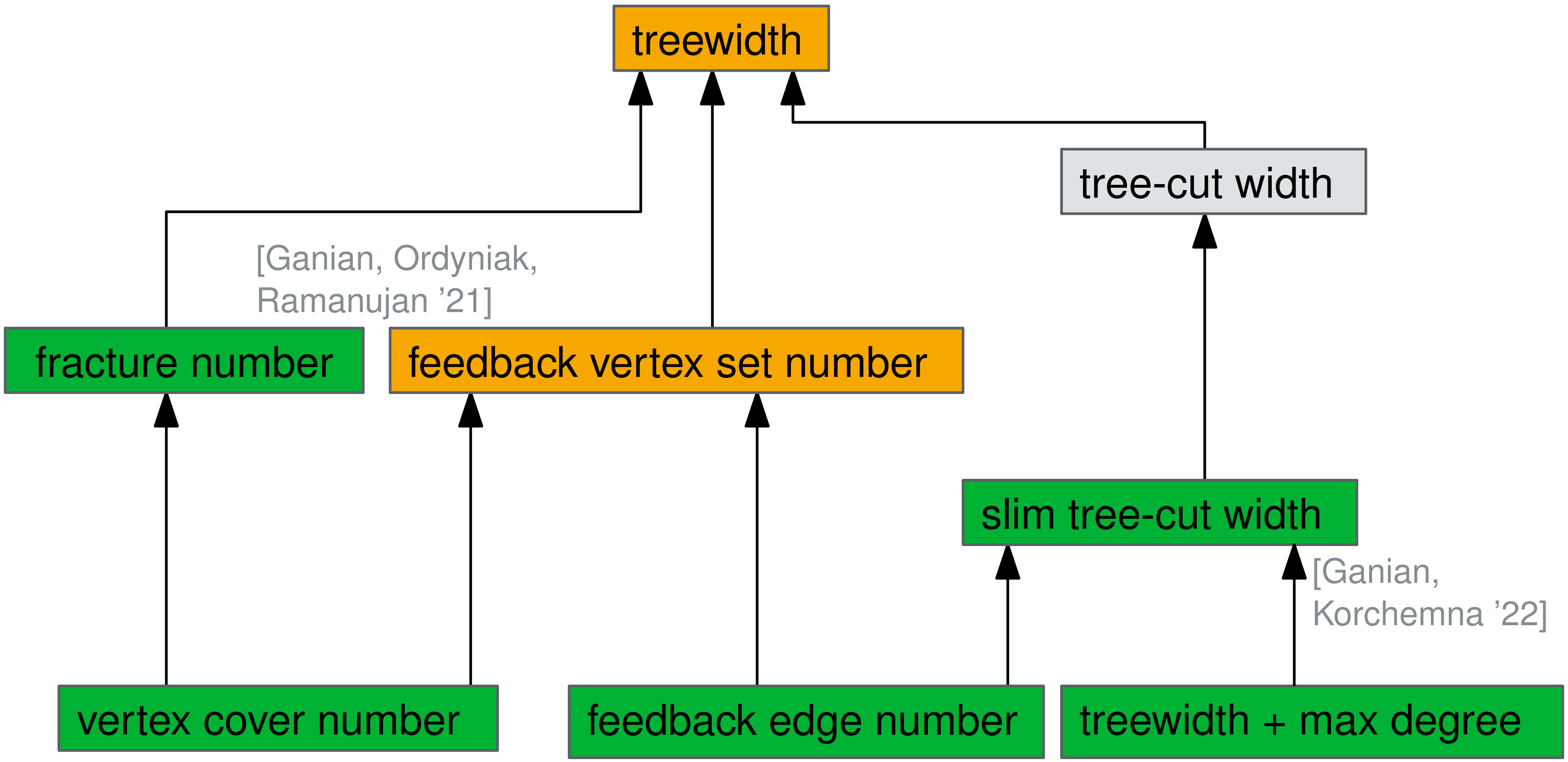
Parameterize by *augmented* graph $G^{\mathcal{T}} := G + \mathcal{T}$.





EDP by Augmented Graph

Known Results:



W[1]-hard FPT unknown

Augmentation for GSTP

ETH zürich

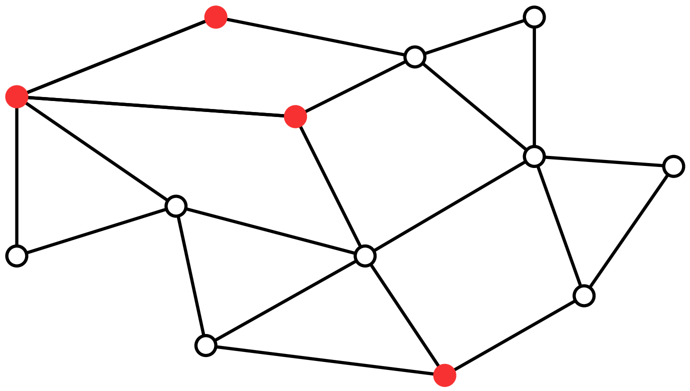
Augmentation for GSTP

Two natural approaches:

Augmentation for GSTP

Two natural approaches:

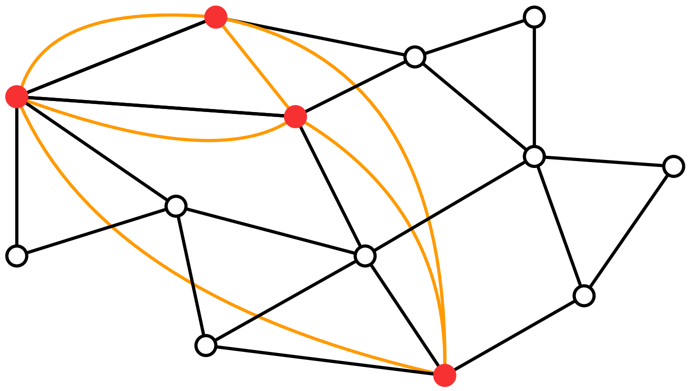
clique-augmented graph



Augmentation for GSTP

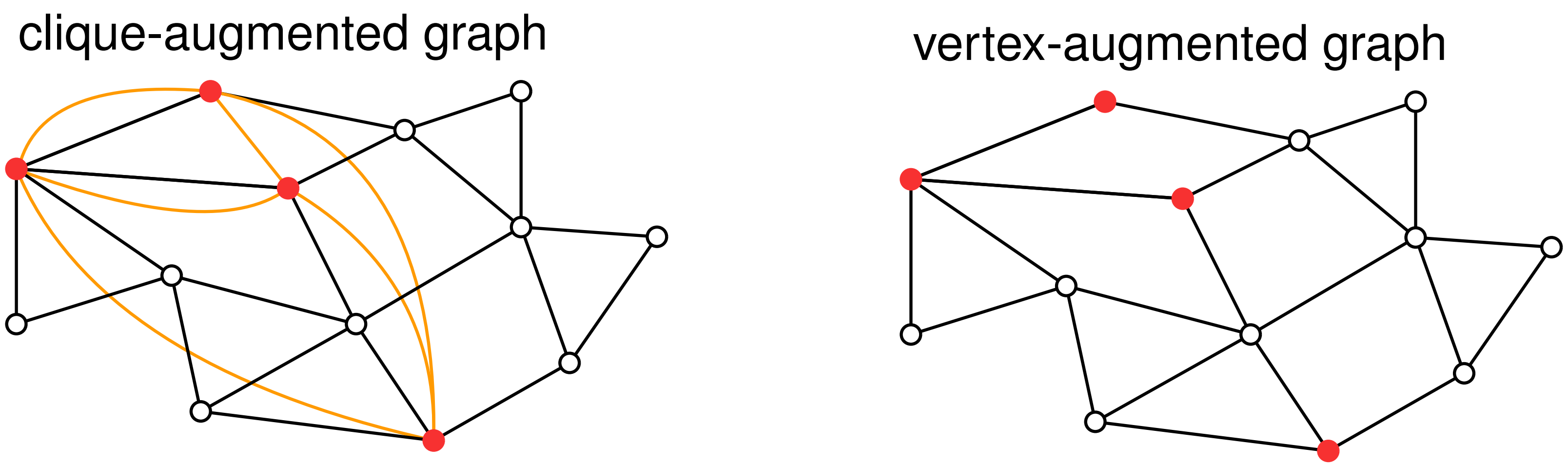
Two natural approaches:

clique-augmented graph



Augmentation for GSTP

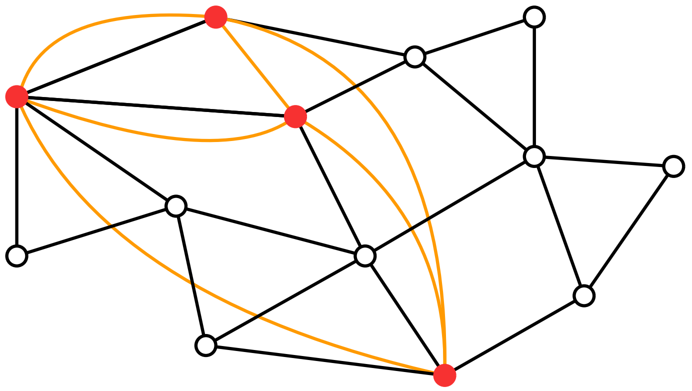
Two natural approaches:



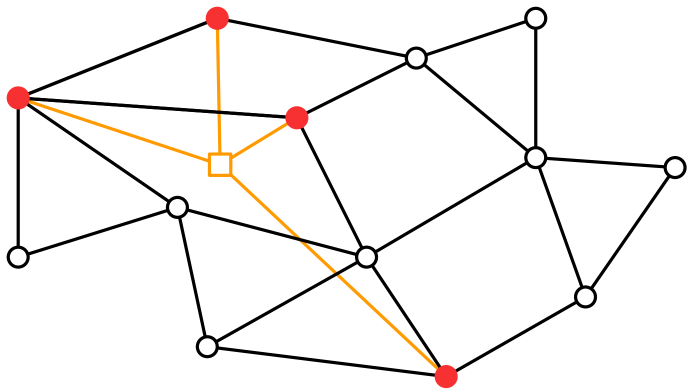
Augmentation for GSTP

Two natural approaches:

clique-augmented graph



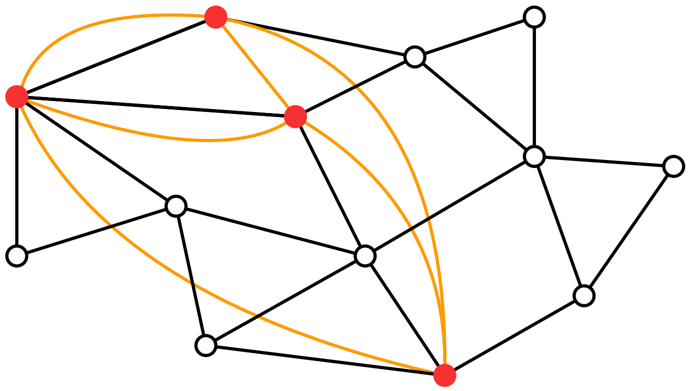
vertex-augmented graph



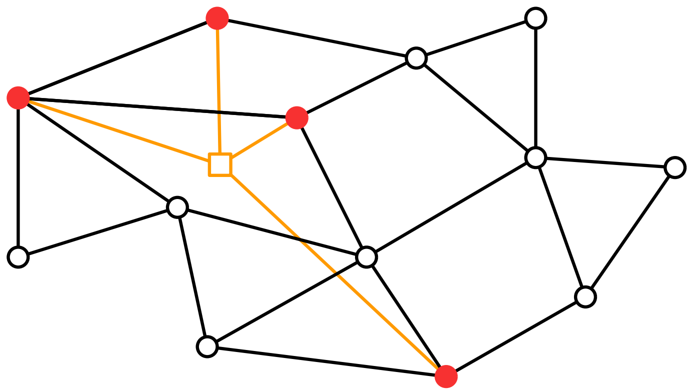
Augmentation for GSTP

Two natural approaches:

clique-augmented graph



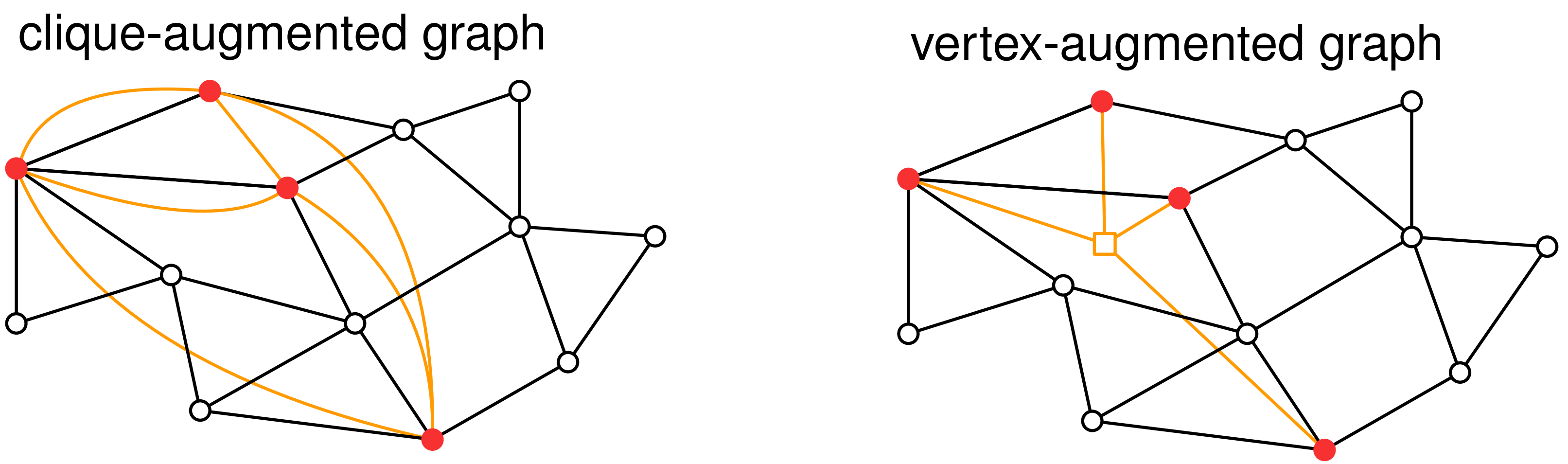
vertex-augmented graph



Lem.: Parameters of clique-augmented graph often upper bounds parameters of vertex-augmented graph.

Augmentation for GSTP

Two natural approaches:



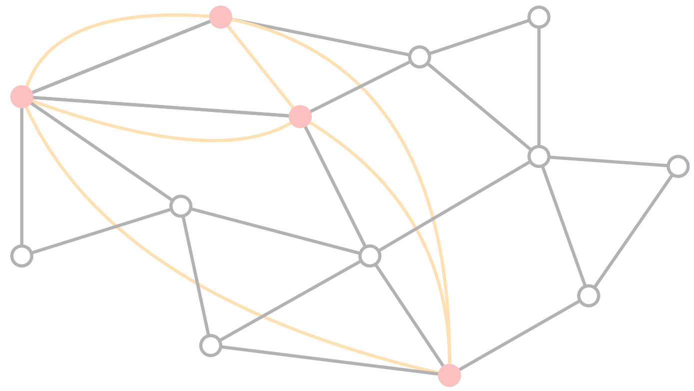
Lem.: Parameters of clique-augmented graph often upper bounds parameters of vertex-augmented graph.

FPT algorithms w.r.t. vertex-augmented graph are more general.

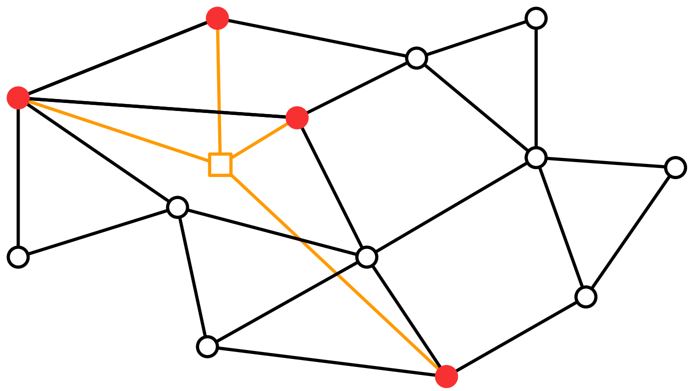
Augmentation for GSTP

Two natural approaches:

clique-augmented graph

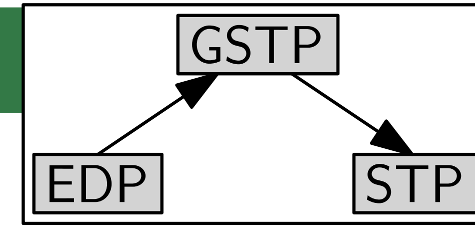


vertex-augmented graph



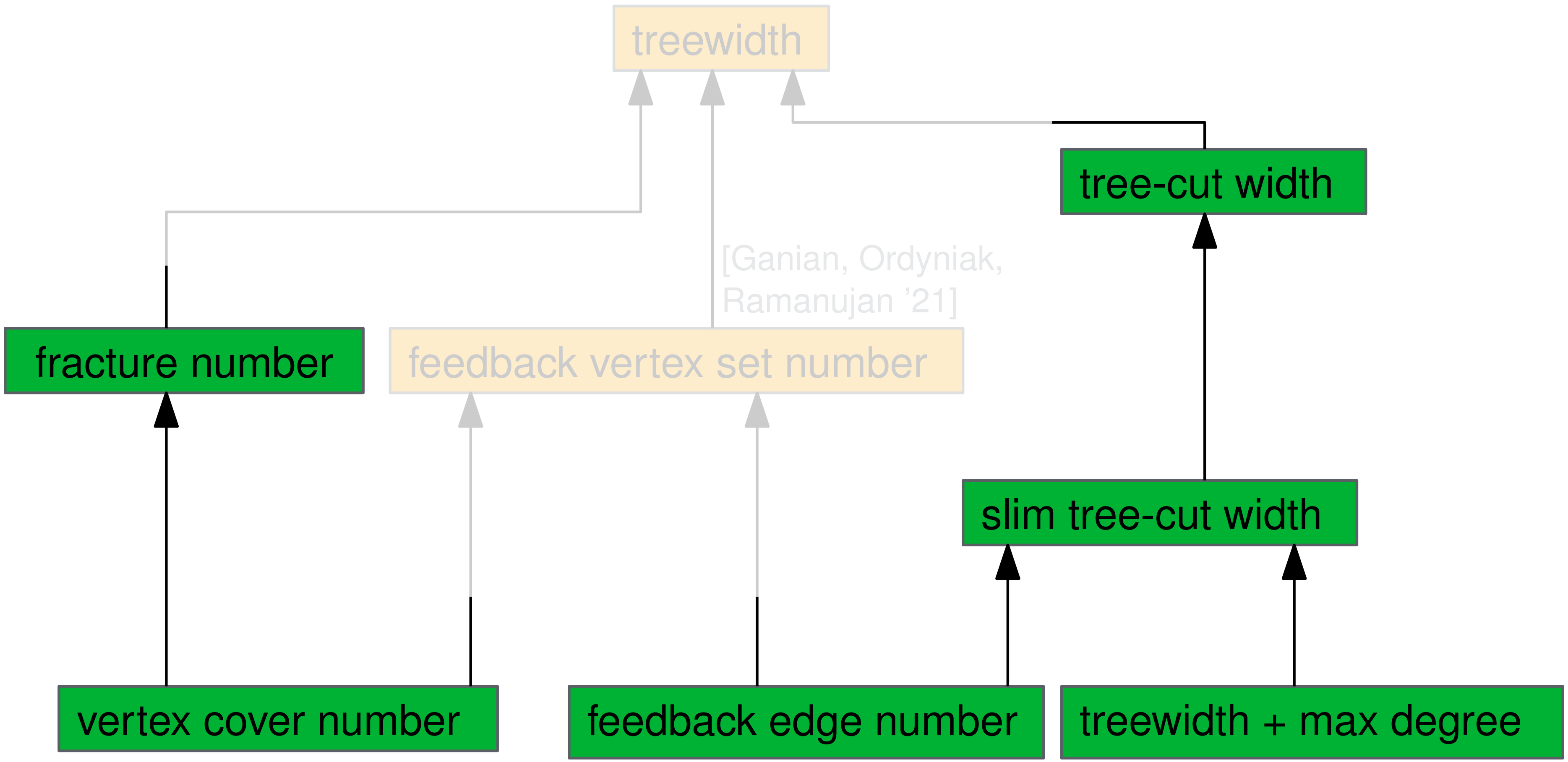
Lem.: Parameters of clique-augmented graph often upper bounds parameters of vertex-augmented graph.

FPT algorithms w.r.t. vertex-augmented graph are more general.

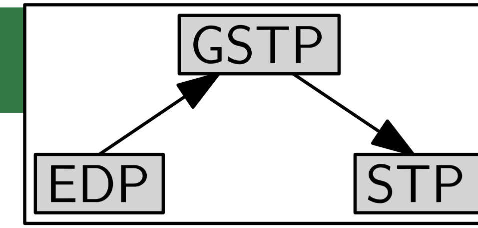


GSTP by Augmented Graph

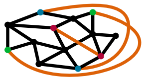
Our Results:



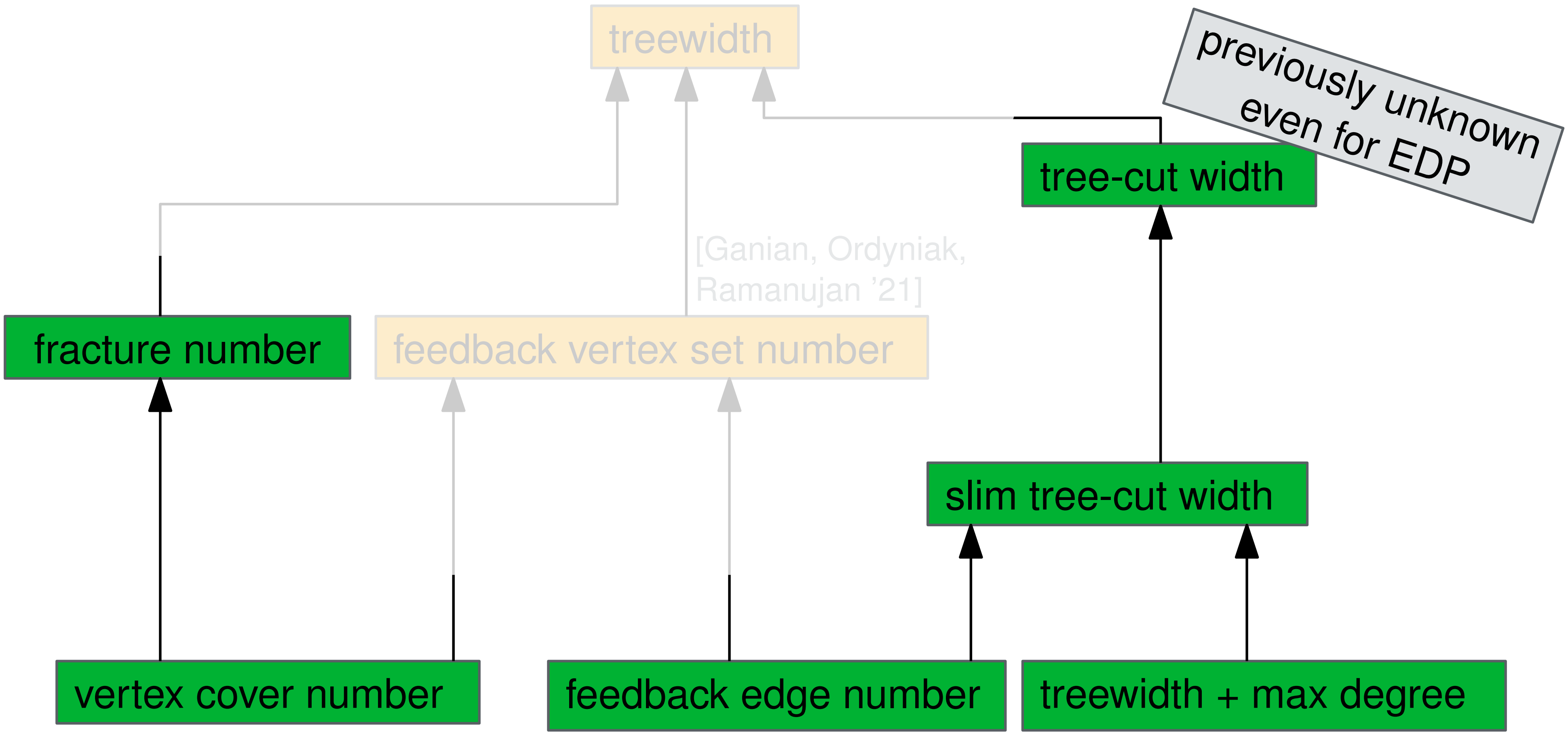
W[1]-hard FPT



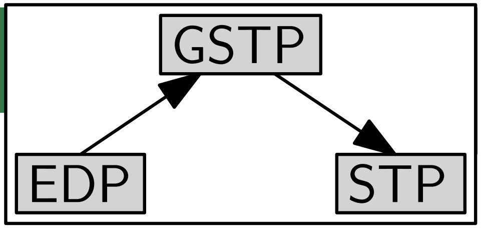
GSTP by Augmented Graph



Our Results:

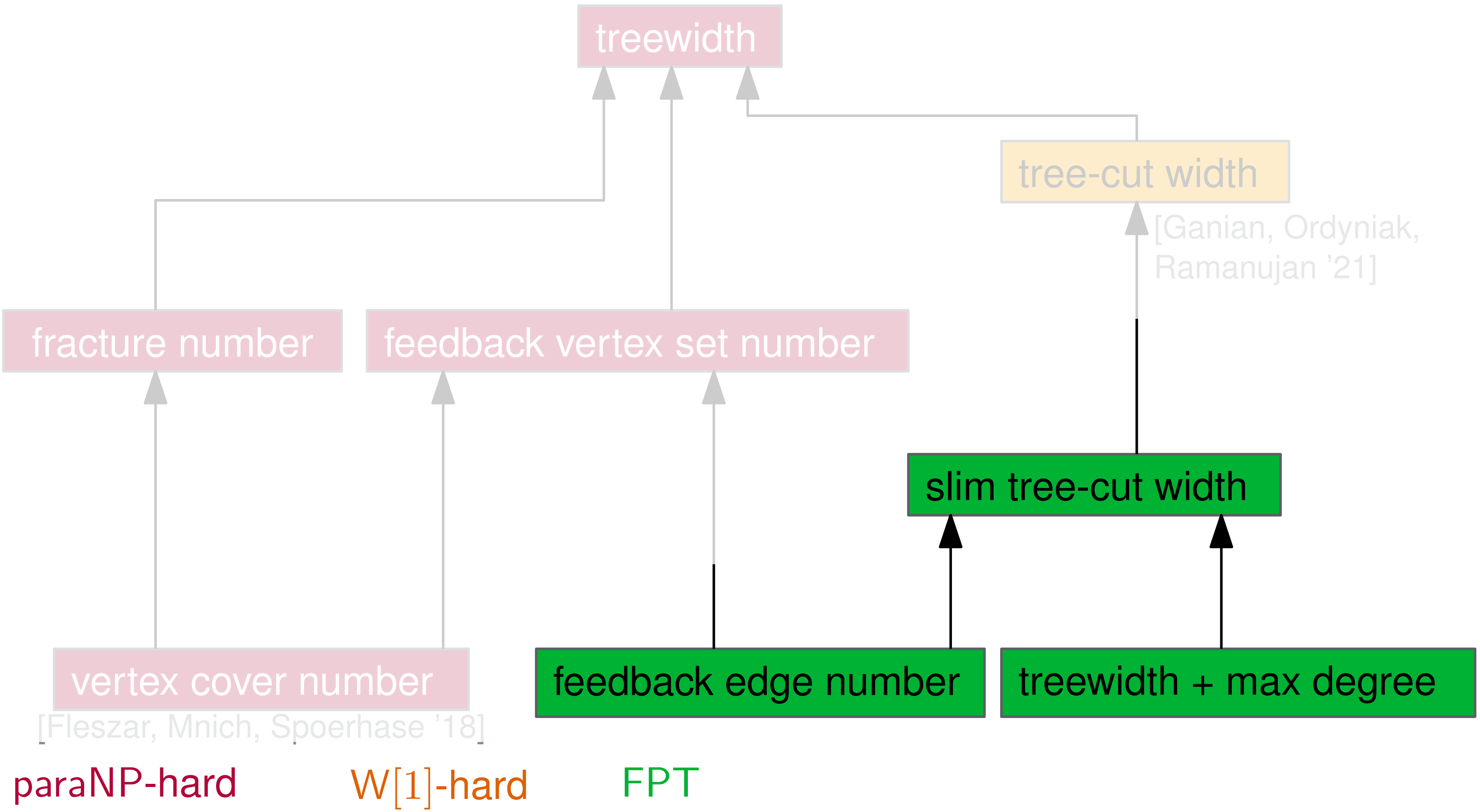


W[1]-hard FPT

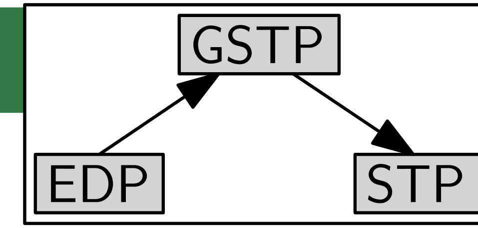


GSTP by Input Graph

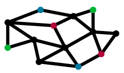
Our Results:



ETH zürich

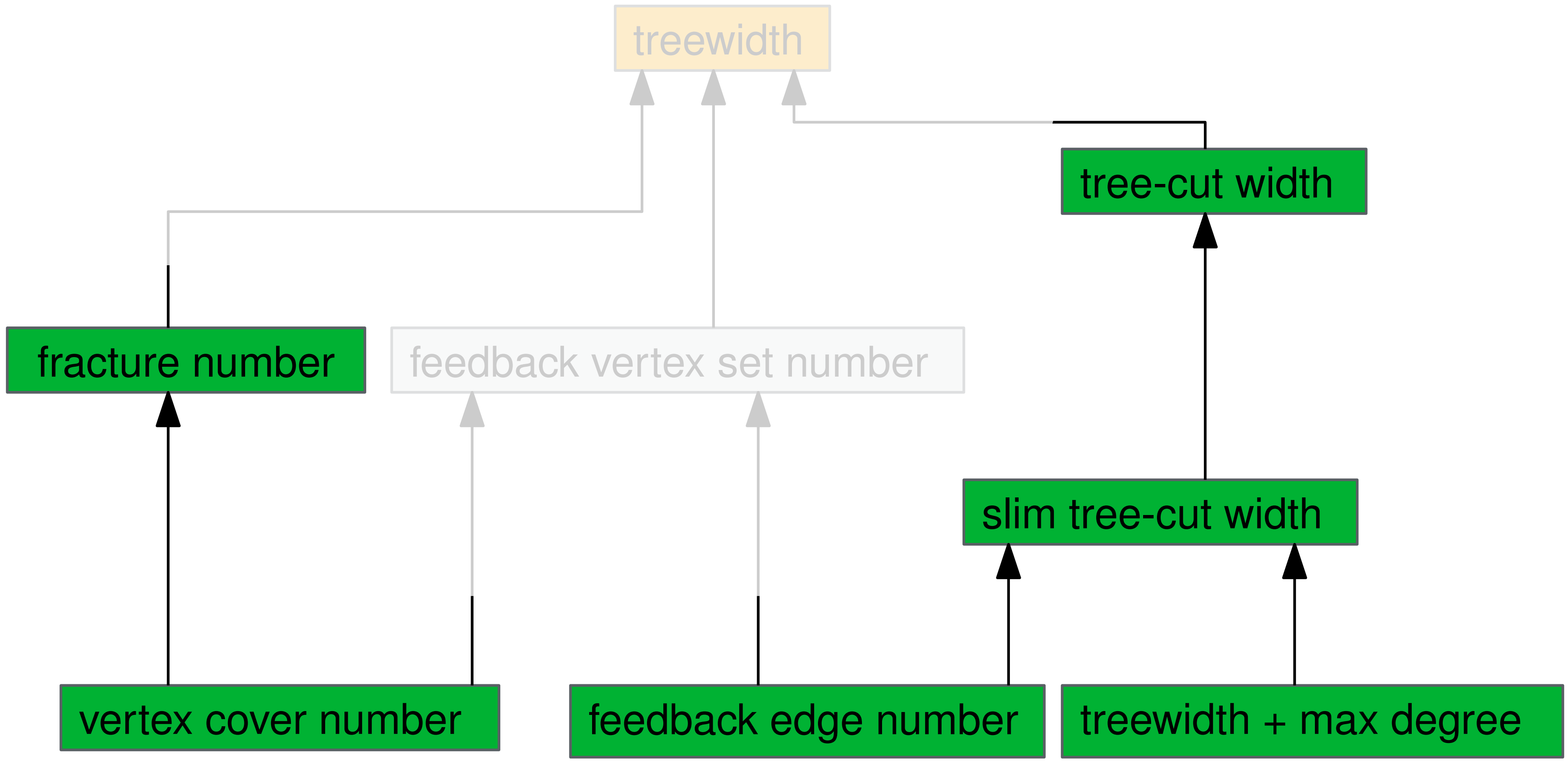


STP by Input Graph



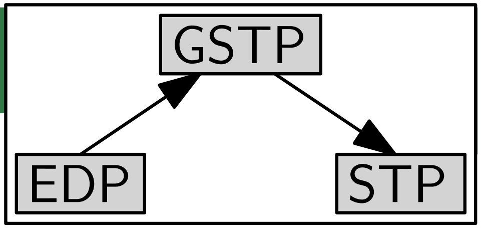
Our Results:

[Aazami, Cheriyan, Jampani '12; Bodlaender et al. '23]



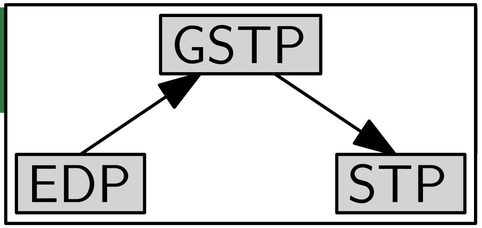
W[1]-hard FPT unknown

Recap: Our Contribution



ETH zürich

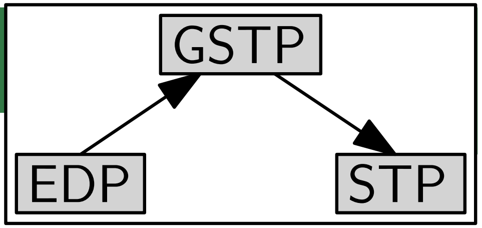
Recap: Our Contribution



- Generalize the notion of the augmented graph to GSTP

ETH zürich

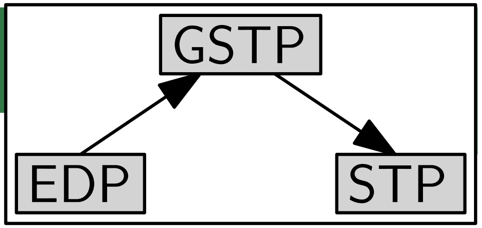
Recap: Our Contribution



- Generalize the notion of the augmented graph to GSTP
- Generalize all known structural FPT algorithms for EDP to GSTP

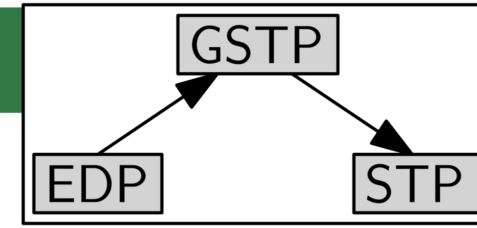
ETH zürich

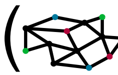
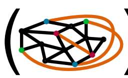
Recap: Our Contribution



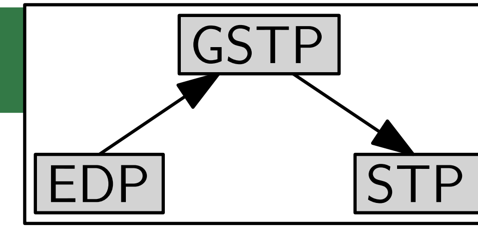
- Generalize the notion of the augmented graph to GSTP
- Generalize all known structural FPT algorithms for EDP to GSTP
- Theorem: GSTP (and EDP) is FPT by tree-cut width of augmented graph.

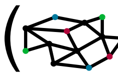
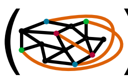
Recap: Our Contribution



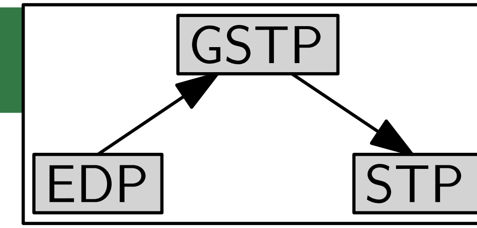
- Generalize the notion of the augmented graph to GSTP
- Generalize all known structural FPT algorithms for EDP to GSTP
- Theorem: GSTP (and EDP) is FPT by tree-cut width of augmented graph.
- For the considered parameters:
  Results for STP by input graph () coincide with GSTP by augmented graph ().

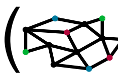
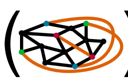
Recap: Our Contribution



- Generalize the notion of the augmented graph to GSTP
- Generalize all known structural FPT algorithms for EDP to GSTP
- Theorem: GSTP (and EDP) is FPT by tree-cut width of augmented graph.
- For the considered parameters:
  Results for STP by input graph () coincide with GSTP by augmented graph ().
- Fixed an error in previous work on nice tree-cut decompositions.

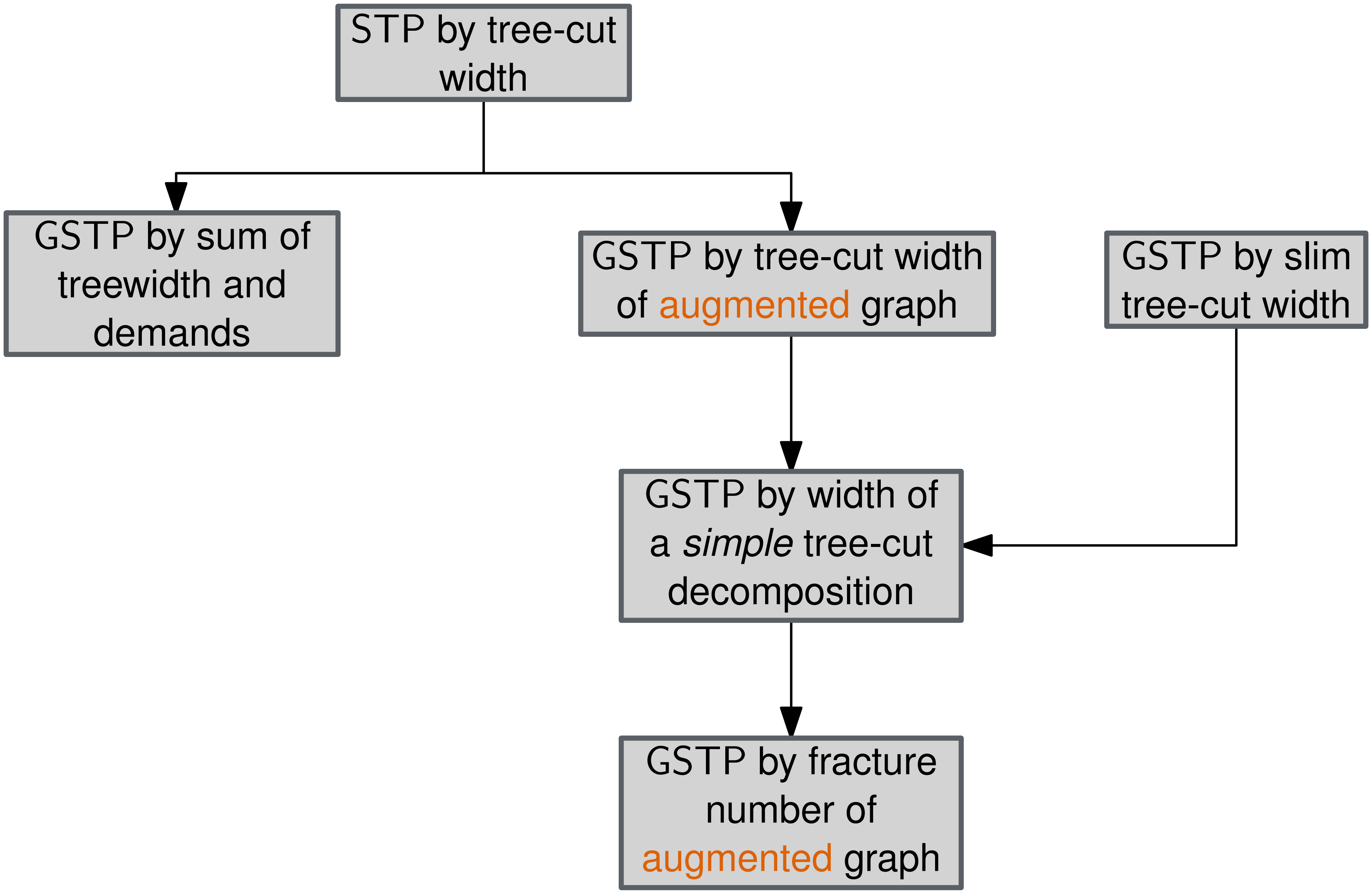
Recap: Our Contribution



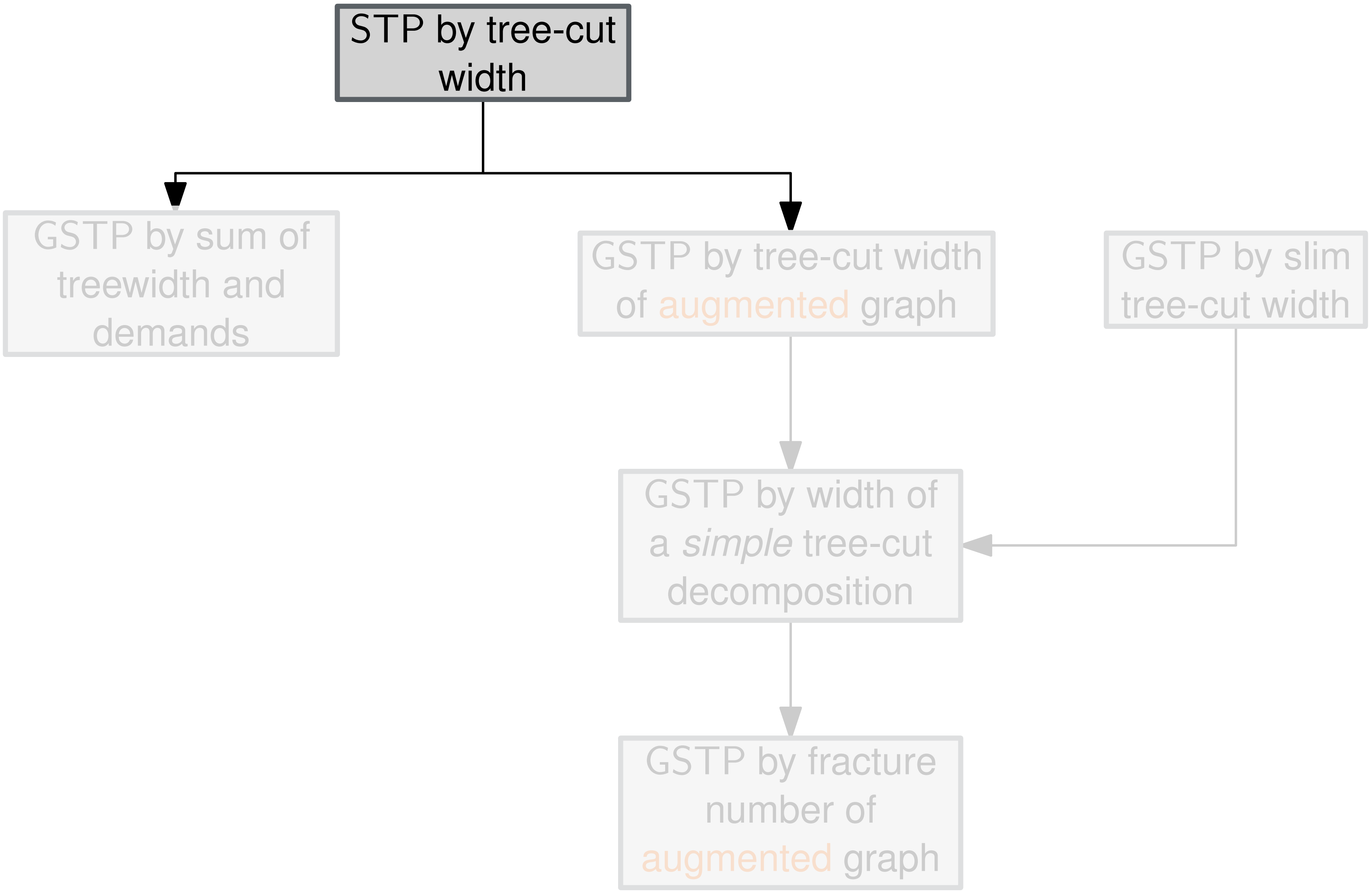
- Generalize the notion of the augmented graph to GSTP
- Generalize all known structural FPT algorithms for EDP to GSTP
- Theorem: GSTP (and EDP) is FPT by tree-cut width of augmented graph.
- For the considered parameters:
  Results for STP by input graph () coincide with GSTP by augmented graph ().
- Fixed an error in previous work on nice tree-cut decompositions.

Thanks for your attention

Our Methods


STP by tree-cut width
GSTP by sum of treewidth and demands
GSTP by tree-cut width of augmented graph
GSTP by slim tree-cut width
GSTP by width of a *simple* tree-cut decomposition
GSTP by fracture number of augmented graph

Our Methods


STP by tree-cut width
GSTP by sum of treewidth and demands
GSTP by tree-cut width of augmented graph
GSTP by slim tree-cut width
GSTP by width of a *simple* tree-cut decomposition
GSTP by fracture number of augmented graph

STP is FPT by Tree-Cut Width

Win-Win argument

STP is FPT by Tree-Cut Width

Win-Win argument

If $|T| \leq w$:

STP is FPT by Tree-Cut Width

Win-Win argument

If $|T| \leq w$:

Obs.: Tree-cut width ot $G^{\{T\}}$ is at most $2w + 3$

STP is FPT by Tree-Cut Width

Win-Win argument

If $|T| \leq w$:

Obs.: Tree-cut width ot $G^{\{T\}}$ is at most $2w + 3$

Solve using algorithm for GSTP by augmented tree-cut width.

STP is FPT by Tree-Cut Width

Win-Win argument

If $|T| \leq w$:

- Obs.: Tree-cut width ot $G^{\{T\}}$ is at most $2w + 3$
- Solve using algorithm for GSTP by augmented tree-cut width.

Otherwise ($|T| > w$):

STP is FPT by Tree-Cut Width

Win-Win argument

If $|T| \leq w$:

Obs.: Tree-cut width ot $G^{\{T\}}$ is at most $2w + 3$

Solve using algorithm for GSTP by augmented tree-cut width.

Otherwise ($|T| > w$):

Obs.: T is not contained in one bag.

STP is FPT by Tree-Cut Width

Win-Win argument

If $|T| \leq w$:

Obs.: Tree-cut width ot $G^{\{T\}}$ is at most $2w + 3$

Solve using algorithm for GSTP by augmented tree-cut width.

Otherwise ($|T| > w$):

Obs.: T is not contained in one bag.

$\rightarrow$ T is at most w-edge connected.

STP is FPT by Tree-Cut Width

Win-Win argument

If $|T| \leq w$:

Obs.: Tree-cut width ot $G^{\{T\}}$ is at most $2w + 3$

Solve using algorithm for GSTP by augmented tree-cut width.

Otherwise ($|T| > w$):

Obs.: T is not contained in one bag.

$\rightarrow$ T is at most w-edge connected.

If $d > w$: Abort!

STP is FPT by Tree-Cut Width

Win-Win argument

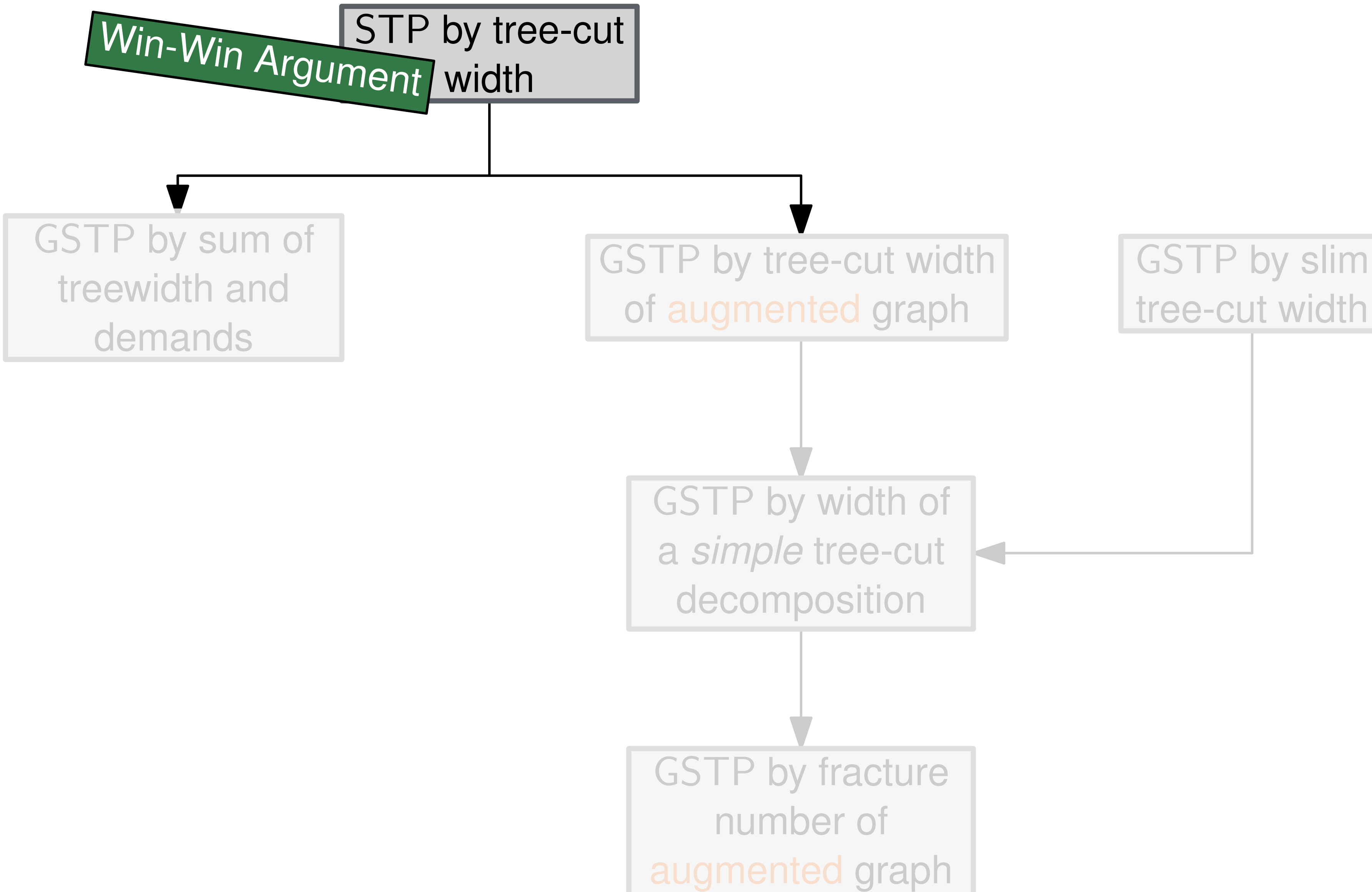
If $|T| \leq w$:

- Obs.: Tree-cut width ot $G^{\{T\}}$ is at most $2w + 3$
- Solve using algorithm for GSTP by augmented tree-cut width.

Otherwise ($|T| > w$):

- Obs.: T is not contained in one bag.
  - $\rightarrow T$ is at most w-edge connected.
- If $d > w$: Abort!
- Otherwise ($d \leq w$): solve using dynamic program w.r.t. sum of tree width and demand.

Our Methods


STP by tree-cut width
Win-Win Argument
GSTP by sum of treewidth and demands
GSTP by tree-cut width of augmented graph
GSTP by slim tree-cut width
GSTP by width of a *simple* tree-cut decomposition
GSTP by fracture number of augmented graph

Our Methods

Thanks for your attention