Computing Tree Decompositions with Small Independence Number

Tuukka Korhonen

UNIVERSITY OF BERGEN

based on joint work with
Clément Dallard[1], Fedor V. Fomin, Petr A. Golovach, and Martin Milanič[1]

[1]FAMNIT and IAM, University of Primorska

Vertex Partitioning in Graphs: From Structure to Algorithms

28 November 2022

Tree-independence number

- Independence number of a tree decomposition = maximum independence number of a bag

Tree-independence number

- Independence number of a tree decomposition = maximum independence number of a bag
  - $\alpha(T) = \max_{v \in V(T)} \alpha(\mathtt{bag}(v))$

Tree-independence number

- Independence number of a tree decomposition = maximum independence number of a bag
  - $\alpha(T) = \max_{v \in V(T)} \alpha(\mathtt{bag}(v))$
- Tree-independence number of a graph = minimum independence number of a tree decomposition of the graph

Tree-independence number

- Independence number of a tree decomposition = maximum independence number of a bag
  - $\alpha(T) = \max_{v \in V(T)} \alpha(\mathtt{bag}(v))$
- Tree-independence number of a graph = minimum independence number of a tree decomposition of the graph
  - tree-$\alpha(G) = \min_{T \text{ a TD of } G} \alpha(T)$

Tree-independence number

- Independence number of a tree decomposition = maximum independence number of a bag
  - $\alpha(T) = \max_{v \in V(T)} \alpha(\mathtt{bag}(v))$
- Tree-independence number of a graph = minimum independence number of a tree decomposition of the graph
  - tree-$\alpha(G) = \min_{T \text{ a TD of } G} \alpha(T)$
- Introduced by [Yolov '18] and [Dallard, Milanič & Storgel '21]

Tree-independence number

- Independence number of a tree decomposition = maximum independence number of a bag
  - $\alpha(T) = \max_{v \in V(T)} \alpha(\mathtt{bag}(v))$
- Tree-independence number of a graph = minimum independence number of a tree decomposition of the graph
  - tree-$\alpha(G) = \min_{T \text{ a TD of } G} \alpha(T)$
- Introduced by [Yolov '18] and [Dallard, Milanič & Storgel '21]
- Generalizes chordal graphs, treewidth, and various prior generalizations of chordal graphs

Algorithmic applications of tree-independence number

Given an n-vertex graph G and a tree decomposition T of G with $\alpha(T) = k$

Algorithmic applications of tree-independence number

Given an n-vertex graph G and a tree decomposition T of G with $\alpha(T) = k$

- $\mathcal{O}(n^{k+2})$ time algorithm for maximum weight independent set

Algorithmic applications of tree-independence number

Given an n-vertex graph G and a tree decomposition T of G with $\alpha(T) = k$

- $\mathcal{O}(n^{k+2})$ time algorithm for maximum weight independent set
- $\mathcal{O}(n^{|H|\cdot(k+2)})$ time maximum weight H-packing [Dallard, Milanič & Storgel '21]

Algorithmic applications of tree-independence number

Given an n-vertex graph G and a tree decomposition T of G with $\alpha(T) = k$

- $\mathcal{O}(n^{k+2})$ time algorithm for maximum weight independent set
- $\mathcal{O}(n^{|H|\cdot(k+2)})$ time maximum weight H-packing [Dallard, Milanič & Storgel '21]
- $f(k) \cdot n^{\mathcal{O}(k)}$ time algorithms for
  - feedback vertex set
  - longest induced path
  - maximum weight induced subgraph with bounded chromatic number satisfying a CMSO property [Milanič & Rzążewski '22]

Our results on computing tree-independence number

Theorem

There is a $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(k)}$ time 8-approximation algorithm for tree-independence number, which also outputs the corresponding tree decomposition.

Our results on computing tree-independence number

Theorem

There is a $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(k)}$ time 8-approximation algorithm for tree-independence number, which also outputs the corresponding tree decomposition.

- Improves over $n^{\mathcal{O}(k^3)}$ time $\mathcal{O}(k^2)$-approximation by [Yolov '18]

Our results on computing tree-independence number

Theorem

There is a $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(k)}$ time 8-approximation algorithm for tree-independence number, which also outputs the corresponding tree decomposition.

- Improves over $n^{\mathcal{O}(k^3)}$ time $\mathcal{O}(k^2)$-approximation by [Yolov '18]

Hardness results:

Our results on computing tree-independence number

Theorem

There is a $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(k)}$ time 8-approximation algorithm for tree-independence number, which also outputs the corresponding tree decomposition.

- Improves over $n^{\mathcal{O}(k^3)}$ time $\mathcal{O}(k^2)$-approximation by [Yolov '18]

Hardness results:

- Assuming Gap-ETH, no $f(k) \cdot n^{o(k)}$ time $g(k)$-approximation algorithm

Our results on computing tree-independence number

Theorem

There is a $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(k)}$ time 8-approximation algorithm for tree-independence number, which also outputs the corresponding tree decomposition.

- Improves over $n^{\mathcal{O}(k^3)}$ time $\mathcal{O}(k^2)$-approximation by [Yolov '18]

Hardness results:

- Assuming Gap-ETH, no $f(k) \cdot n^{o(k)}$ time $g(k)$-approximation algorithm
- For every constant $k \geq 4$, NP-hard to decide if tree-$\alpha(G) \leq k$

Our results on computing tree-independence number

Theorem

There is a $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(k)}$ time 8-approximation algorithm for tree-independence number, which also outputs the corresponding tree decomposition.

- Improves over $n^{\mathcal{O}(k^3)}$ time $\mathcal{O}(k^2)$-approximation by [Yolov '18]

Hardness results:

- Assuming Gap-ETH, no $f(k) \cdot n^{o(k)}$ time $g(k)$-approximation algorithm
- For every constant $k \geq 4$, NP-hard to decide if tree-$\alpha(G) \leq k$

Open problems:

Our results on computing tree-independence number

Theorem

There is a $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(k)}$ time 8-approximation algorithm for tree-independence number, which also outputs the corresponding tree decomposition.

- Improves over $n^{\mathcal{O}(k^3)}$ time $\mathcal{O}(k^2)$-approximation by [Yolov '18]

Hardness results:

- Assuming Gap-ETH, no $f(k) \cdot n^{o(k)}$ time $g(k)$-approximation algorithm
- For every constant $k \geq 4$, NP-hard to decide if tree-$\alpha(G) \leq k$

Open problems:

- Complexity of deciding tree-$\alpha(G) \leq k$ for $k = 2, 3$?

Our results on computing tree-independence number

Theorem

There is a $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(k)}$ time 8-approximation algorithm for tree-independence number, which also outputs the corresponding tree decomposition.

- Improves over $n^{\mathcal{O}(k^3)}$ time $\mathcal{O}(k^2)$-approximation by [Yolov '18]

Hardness results:

- Assuming Gap-ETH, no $f(k) \cdot n^{o(k)}$ time $g(k)$-approximation algorithm
- For every constant $k \geq 4$, NP-hard to decide if tree-$\alpha(G) \leq k$

Open problems:

- Complexity of deciding tree-$\alpha(G) \leq k$ for $k = 2, 3$?
- Is deciding tree-$\alpha(G) \leq k$ for unbounded k in NP or $\sum_2^P$-hard?

Outline of the algorithm

1. Bounded tree-$\alpha \Rightarrow \alpha$-balanced separators with bounded α

Outline of the algorithm

1. Bounded tree-$\alpha \Rightarrow \alpha$-balanced separators with bounded α
2. Recursive construction in Robertson-Seymour fashion

Outline of the algorithm

1. Bounded tree-$\alpha \Rightarrow \alpha$-balanced separators with bounded α
2. Recursive construction in Robertson-Seymour fashion
3. Reduction from finding balanced separators to finding separators

Outline of the algorithm

1. Bounded tree-$\alpha \Rightarrow \alpha$-balanced separators with bounded α
2. Recursive construction in Robertson-Seymour fashion
3. Reduction from finding balanced separators to finding separators
4. 2-approximation algorithm for separators

Outline of 2-approximation of separators

Input: Graph G, integer k, and two sets of vertices V_1, V_2

Task: Find an (V_1, V_2)-separator S with $\alpha(S) \leq 2k$, or conclude that no (V_1, V_2)-separators with $\alpha(S) \leq k$ exist

Outline of 2-approximation of separators

Input: Graph G, integer k, and two sets of vertices V_1, V_2

Task: Find an (V_1, V_2)-separator S with $\alpha(S) \leq 2k$, or conclude that no (V_1, V_2)-separators with $\alpha(S) \leq k$ exist

1. Guess a container $R \supseteq S$ with $\alpha(R) \leq \mathcal{O}(k^2)$

Outline of 2-approximation of separators

Input: Graph G, integer k, and two sets of vertices V_1, V_2

Task: Find an (V_1, V_2)-separator S with $\alpha(S) \leq 2k$, or conclude that no (V_1, V_2)-separators with $\alpha(S) \leq k$ exist

1. Guess a container $R \supseteq S$ with $\alpha(R) \leq \mathcal{O}(k^2)$
2. By branching make sure that $R \subseteq N(V_1 \cup V_2)$

Outline of 2-approximation of separators

Input: Graph G, integer k, and two sets of vertices V_1, V_2

Task: Find an (V_1, V_2)-separator S with $\alpha(S) \leq 2k$, or conclude that no (V_1, V_2)-separators with $\alpha(S) \leq k$ exist

1. Guess a container $R \supseteq S$ with $\alpha(R) \leq \mathcal{O}(k^2)$
2. By branching make sure that $R \subseteq N(V_1 \cup V_2)$
3. Round a linear program

Optimality of tree-independence number

Optimality of tree-independence number

- Let λ be a monotone graph parameter and tree-λ defined analogously to tree-α

Optimality of tree-independence number

- Let λ be a monotone graph parameter and tree-λ defined analogously to tree-α
  - tree-$\lambda(G) = \min_{T \text{ a tree decomposition of } G} \max_{v \in V(T)} \lambda(G[\mathtt{bag}(v)])$

Optimality of tree-independence number

- Let λ be a monotone graph parameter and tree-λ defined analogously to tree-α
  - tree-$\lambda(G) = \min_{T \text{ a tree decomposition of } G} \max_{v \in V(T)} \lambda(G[\mathtt{bag}(v)])$

Theorem

Either exists f s.t. $\alpha(G) \leq f(\lambda(G))$ for all G, or independent set is para-NP-hard parameterized by tree-$\lambda(G)$

Optimality of tree-independence number

- Let λ be a monotone graph parameter and tree-λ defined analogously to tree-α
  - tree-$\lambda(G) = \min_{T \text{ a tree decomposition of } G} \max_{v \in V(T)} \lambda(G[\mathtt{bag}(v)])$

Theorem

Either exists f s.t. $\alpha(G) \leq f(\lambda(G))$ for all G, or independent set is para-NP-hard parameterized by tree-$\lambda(G)$

- Proof: Independent set is NP-hard on twice-subdivided graphs, but they have a tree decomposition where every bag induces either independent set or P_4

Minor-matching hypertree width

- Minor-matching hypertree width of G defined by [Yolov '18]:
  - tree-$\mu(G) = $ tree-$\alpha(L(G)^2)$

Minor-matching hypertree width

- Minor-matching hypertree width of G defined by [Yolov '18]:
  - tree-$\mu(G) = $ tree-$\alpha(L(G)^2)$
- Alternatively, for $S \subseteq V$, let $\mu(S)$ be maximum size of induced matching in G whose every edge has at least one endpoint in S

Minor-matching hypertree width

- Minor-matching hypertree width of G defined by [Yolov '18]:
  - tree-$\mu(G) = \text{tree-}\alpha(L(G)^2)$
- Alternatively, for $S \subseteq V$, let $\mu(S)$ be maximum size of induced matching in G whose every edge has at least one endpoint in S
  - $\mu(T) = \max_{v \in V(T)} \mu(\texttt{bag}(v))$

Minor-matching hypertree width

- Minor-matching hypertree width of G defined by [Yolov '18]:
  - tree-$\mu(G) = \text{tree-}\alpha(L(G)^2)$
- Alternatively, for $S \subseteq V$, let $\mu(S)$ be maximum size of induced matching in G whose every edge has at least one endpoint in S
  - $\mu(T) = \max_{v \in V(T)} \mu(\texttt{bag}(v))$
  - tree-$\mu(G) = \min_{T \text{ a TD of } G} \mu(T)$

Minor-matching hypertree width

- Minor-matching hypertree width of G defined by [Yolov '18]:
  - tree-$\mu(G) = $ tree-$\alpha(L(G)^2)$
- Alternatively, for $S \subseteq V$, let $\mu(S)$ be maximum size of induced matching in G whose every edge has at least one endpoint in S
  - $\mu(T) = \max_{v \in V(T)} \mu(\texttt{bag}(v))$
  - tree-$\mu(G) = \min_{T \text{ a TD of } G} \mu(T)$
- tree-$\mu(G) \leq$ tree-$\alpha(G)$, but e.g. $K_{n,n}$ has bounded tree-μ but unbounded tree-α

Minor-matching hypertree width

- Minor-matching hypertree width of G defined by [Yolov '18]:
  - tree-$\mu(G) = $ tree-$\alpha(L(G)^2)$
- Alternatively, for $S \subseteq V$, let $\mu(S)$ be maximum size of induced matching in G whose every edge has at least one endpoint in S
  - $\mu(T) = \max_{v \in V(T)} \mu(\texttt{bag}(v))$
  - tree-$\mu(G) = \min_{T \text{ a TD of } G} \mu(T)$
- tree-$\mu(G) \leq$ tree-$\alpha(G)$, but e.g. $K_{n,n}$ has bounded tree-μ but unbounded tree-α
- Given a decomposition T with $\mu(T) = k$, we can solve [Yolov '18]:

Minor-matching hypertree width

- Minor-matching hypertree width of G defined by [Yolov '18]:
  - tree-$\mu(G) = $ tree-$\alpha(L(G)^2)$
- Alternatively, for $S \subseteq V$, let $\mu(S)$ be maximum size of induced matching in G whose every edge has at least one endpoint in S
  - $\mu(T) = \max_{v \in V(T)} \mu(\texttt{bag}(v))$
  - tree-$\mu(G) = \min_{T \text{ a TD of } G} \mu(T)$
- tree-$\mu(G) \leq$ tree-$\alpha(G)$, but e.g. $K_{n,n}$ has bounded tree-μ but unbounded tree-α
- Given a decomposition T with $\mu(T) = k$, we can solve [Yolov '18]:
  - Maximum weight independent set in $n^{\mathcal{O}(k)}$ time
  - r-coloring in $n^{\mathcal{O}(kr)}$ time

Minor-matching hypertree width

- Minor-matching hypertree width of G defined by [Yolov '18]:
  - tree-$\mu(G) = $ tree-$\alpha(L(G)^2)$
- Alternatively, for $S \subseteq V$, let $\mu(S)$ be maximum size of induced matching in G whose every edge has at least one endpoint in S
  - $\mu(T) = \max_{v \in V(T)} \mu(\texttt{bag}(v))$
  - tree-$\mu(G) = \min_{T \text{ a TD of } G} \mu(T)$
- tree-$\mu(G) \leq$ tree-$\alpha(G)$, but e.g. $K_{n,n}$ has bounded tree-μ but unbounded tree-α
- Given a decomposition T with $\mu(T) = k$, we can solve [Yolov '18]:
  - Maximum weight independent set in $n^{\mathcal{O}(k)}$ time
  - r-coloring in $n^{\mathcal{O}(kr)}$ time
- Based on a theorem that a maximal independent set of G can intersect S in at most $n^{\mathcal{O}(\mu(S))}$ different ways

Conclusion

- Similar optimality argument for tree-μ: Optimal among tree decomposition based parameters where width of $\texttt{bag}(v)$ depends on $G[N[\texttt{bag}(v)]]$ with $\texttt{bag}(v)$ labeled

Conclusion

- Similar optimality argument for tree-μ: Optimal among tree decomposition based parameters where width of $\texttt{bag}(v)$ depends on $G[N[\texttt{bag}(v)]]$ with $\texttt{bag}(v)$ labeled
- Open problems:

Conclusion

- Similar optimality argument for tree-μ: Optimal among tree decomposition based parameters where width of $\texttt{bag}(v)$ depends on $G[N[\texttt{bag}(v)]]$ with $\texttt{bag}(v)$ labeled
- Open problems:
  - Feedback vertex set parameterized by tree-μ?

Conclusion

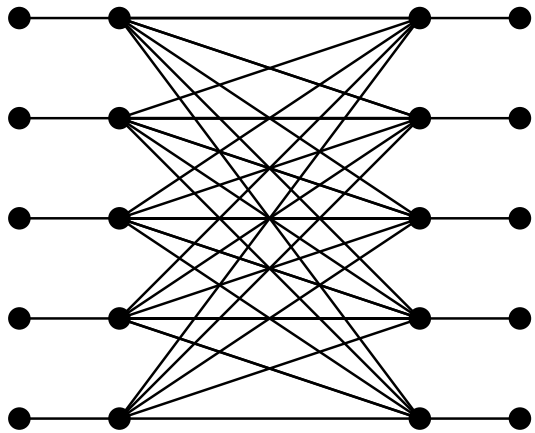
- Similar optimality argument for tree-μ: Optimal among tree decomposition based parameters where width of $\texttt{bag}(v)$ depends on $G[N[\texttt{bag}(v)]]$ with $\texttt{bag}(v)$ labeled
- Open problems:
  - Feedback vertex set parameterized by tree-μ?
  - General framework for $n^{\mathcal{O}(\text{tree-}\mu(G))}$ time algorithms for finding sparse induced subgraphs?

Conclusion

- Similar optimality argument for tree-μ: Optimal among tree decomposition based parameters where width of $\texttt{bag}(v)$ depends on $G[N[\texttt{bag}(v)]]$ with $\texttt{bag}(v)$ labeled
- Open problems:
  - Feedback vertex set parameterized by tree-μ?
  - General framework for $n^{\mathcal{O}(\text{tree-}\mu(G))}$ time algorithms for finding sparse induced subgraphs?
  - Even more general tree decomposition based parameters for which independent set is XP?

Conclusion

- Similar optimality argument for tree-μ: Optimal among tree decomposition based parameters where width of $\texttt{bag}(v)$ depends on $G[N[\texttt{bag}(v)]]$ with $\texttt{bag}(v)$ labeled
- Open problems:
  - Feedback vertex set parameterized by tree-μ?
  - General framework for $n^{\mathcal{O}(\text{tree-}\mu(G))}$ time algorithms for finding sparse induced subgraphs?
  - Even more general tree decomposition based parameters for which independent set is XP?



Thank you!