Tree-decompositions with metric properties on the bags

Nicolas Nisse

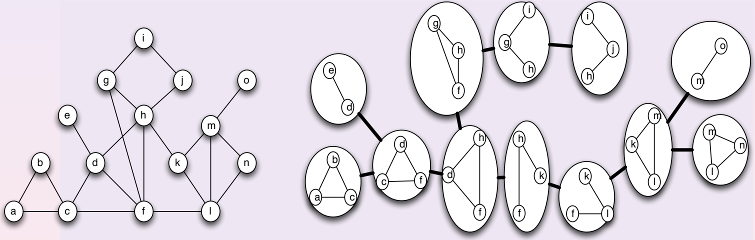
Univ. Côte d'Azur, Inria, CNRS, I3S, France

Based on works with D. Coudert, G. Ducoffe, A. Kosowski, S. Legay, B. Li and K. Suchan

Xidian University, Xi'an, September 5th, 2018

Tree/Path-Decompositions [Robertson and Seymour 83]

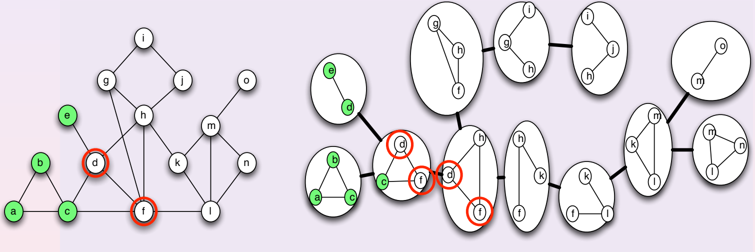
Representation of a graph as a Tree preserving connectivity properties



Tree T + family $\mathcal{X} = (X_t)_{t \in V(T)}$ of "bags" (set of vertices of G)
Important: intersection of two adjacent bags = separator of G

Tree/Path-Decompositions [Robertson and Seymour 83]

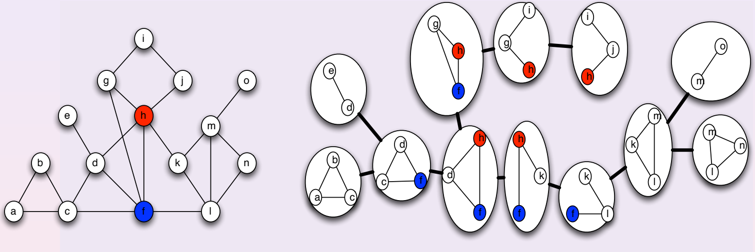
Representation of a graph as a Tree preserving connectivity properties



Tree T + family $\mathcal{X} = (X_t)_{t \in V(T)}$ of "bags" (set of vertices of G)
Important: intersection of two adjacent bags = separator of G

Tree/Path-Decompositions [Robertson and Seymour 83]

Representation of a graph as a Tree preserving connectivity properties

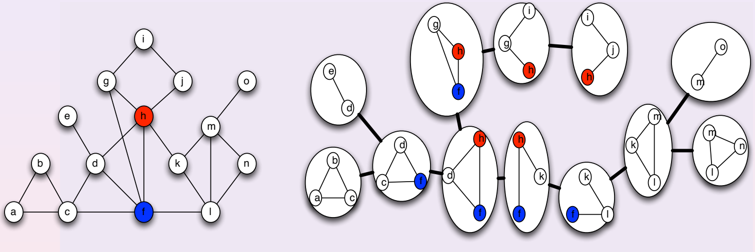


Tree T + family $\mathcal{X} = (X_t)_{t \in V(T)}$ of "bags" (set of vertices of G)
Important: intersection of two adjacent bags = separator of G

- $\bigcup_{t \in V(T)} X_t = V(G)$;
- for any $uv \in E(G)$, there exists a bag X_t containing u and v;
- for any $v \in V(G)$, $\{t \in V(T) \mid v \in X_t\}$ induces a subtree.

Tree/Path-Decompositions [Robertson and Seymour 83]

Representation of a graph as a Tree preserving connectivity properties

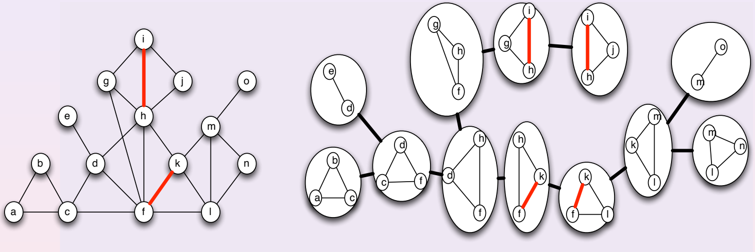


Tree T + family $\mathcal{X} = (X_t)_{t \in V(T)}$ of "bags" (set of vertices of G)
Important: intersection of two adjacent bags = separator of G

Width of $(T, \mathcal{X})$: size of largest bag (minus 1)
Treewidth of a graph G, $tw(G)$: min width over all tree-decompositions.

Tree/Path-Decompositions [Robertson and Seymour 83]

Representation of a graph as a Tree preserving connectivity properties



Tree T + family $\mathcal{X} = (X_t)_{t \in V(T)}$ of "bags" (set of vertices of G)
Important: intersection of two adjacent bags = separator of G

Tree-decomposition $\Leftrightarrow$ (clique tree of) triangulation of G (bags are maximal cliques)
Treewidth+1 $\Leftrightarrow$ Minimum $\omega(H)$ among any chordal supergraph H of G

Many important Algorithmic Applications of *tw*

Brief reminder on Computational Complexity

P. Class of Problems that can be solved by a **deterministic** algorithm in **polynomial-time** $O(n^c)$ (in the size n of the input)

NP. Class of Problems that can be solved by a **non deterministic** algorithm in **polynomial-time** $O(n^c)$ (in the size n of the input)

NP-hard. Class of Problems "**as hard as the hardest problems in NP**". Essentially: cannot be solved in polynomial-time unless P=NP

FPT. (**Fixed Parameter Tractable**) Unformally: $k \in \mathbb{N}$ be a **fixed** parameter. Let G be an input of size n, and Π a graph property

$\Pi(G) \leq n$? is FPT if it can be solved by a deterministic algorithm in time $f(k)n^c$ for some computable function f.

example: vertex cover (NP-hard). $vc(G) \leq k$? solvable in time:

- $2^{O(n)}$ if k is part of the input
- $O(2^k n)$ if k is fixed.

Many important Algorithmic Applications of *tw*

- cornerstone of Graph Minors Theorem [Robertson and Seymour 1983-2004]

  $\Rightarrow$ any graph property ($\Pi(G) \leq k$) that is closed under minor is FPT in k

- problems expressible in MSOL solvable in polynomial time in graphs of bounded treewidth (dynamic programming) [Courcelle, 90]

  any such problem is FPT in *tw*

- design of sub-exponential algorithms in some graph classes (e.g., planar, bounded genus, H-minor-free...)

  (**bi-dimensionality**) [Demaine *et al.* 04]

- design of FPT algorithms (meta-kernelization/protrusions)

  [Fomin *et al.* 09]

Main Problem: Computing tree-decomposition

Deciding if $tw(G) \leq k$? $\Rightarrow$ Very hard!

Related Work

Exact algorithms

- NP-hard if k part of the input [Arnborg,Corneil,Prokurowski 87]
- FPT: algorithm in $O(2^{k^3} n)$ [Bodlaender,Kloks 96]
- "practical" algorithms only for graph with treewidth ≤ 4 e.g., [Sanders 96]
- Branch & Bound algorithms (for small graphs) [Bodlaender *et al.* 12] [Coudert,Mazauric,N. 14]

Approximation algorithms

- 5-approximation in time $O(2^k n)$ [Bodlaender *et al.* 13]
- $\sqrt{\log OPT}$-approximation in polynomial-time (SDP) [Feige *et al.* 05]
- assuming Small Set Expansion Conjecture, no poly-time constant-ratio approximation [Wu,Austrin,Pitassi,Liu 14]
- 3/2-approximation in planar graphs (*complexity?*) [Seymour,Thomas 93]

Heuristics

- Mainly based on local complementations of edges (minimum fill-in: perfect elimination ordering of vertices) [Bodlaender *et al.*]

In this talk ⇒ We want to compute it anyway!

Instead of constraining the size of bags ⇒ constraint bags' properties

1. bags' structure
2. bags' diameter: treelength
3. bags' radius: tree-breadth

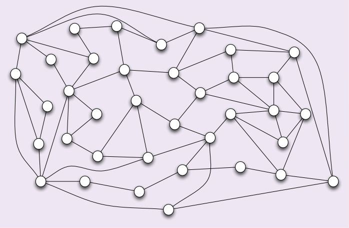
and relationships between them...

First alternative: constraint bags' structure

Simple algorithm

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"

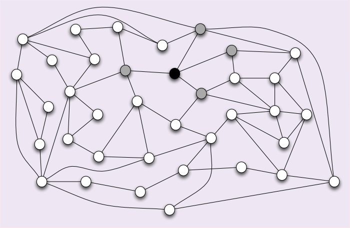


First alternative: constraint bags' structure

Simple algorithm

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"

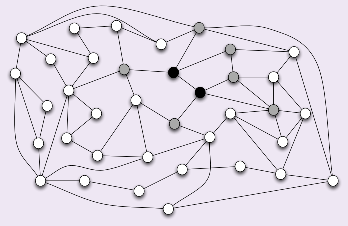


First alternative: constraint bags' structure

Simple algorithm

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"

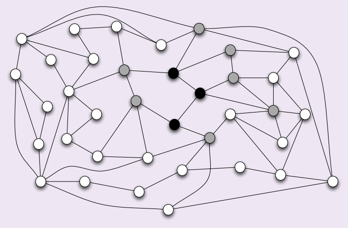


First alternative: constraint bags' structure

Simple algorithm

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"

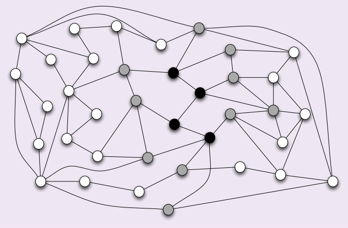


First alternative: constraint bags' structure

Simple algorithm

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

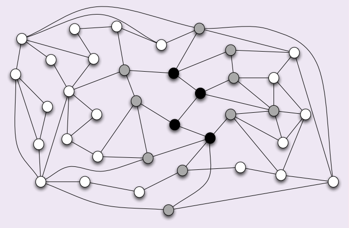
- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"

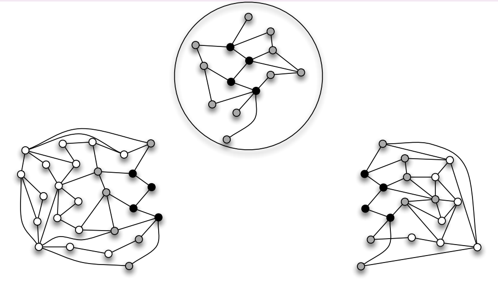


First alternative: constraint bags' structure

Simple algorithm [Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"





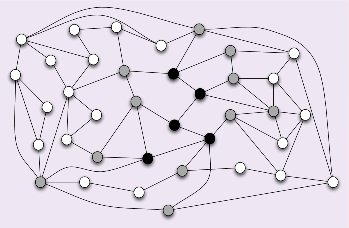
First alternative: constraint bags' structure

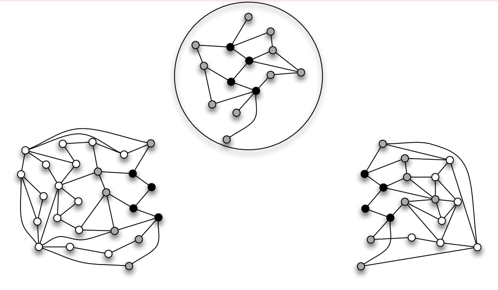
Simple algorithm

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P

  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"

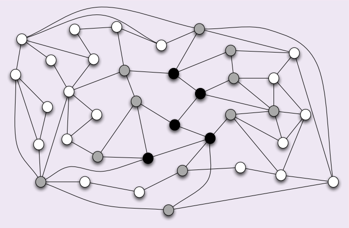


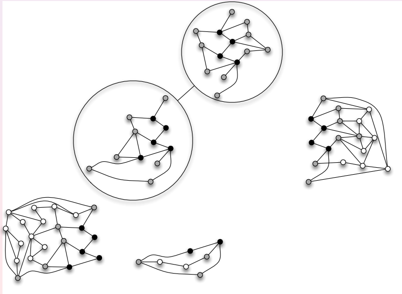


First alternative: constraint bags' structure

Simple algorithm [Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"



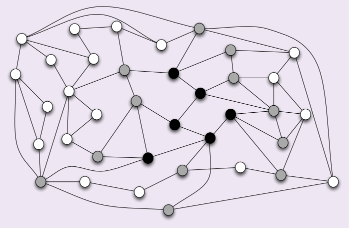


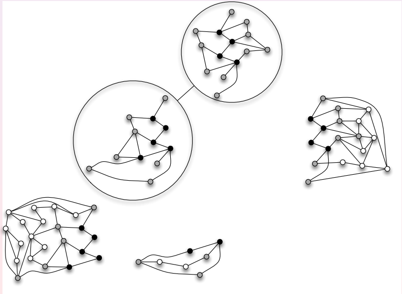
First alternative: constraint bags' structure

Simple algorithm

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"



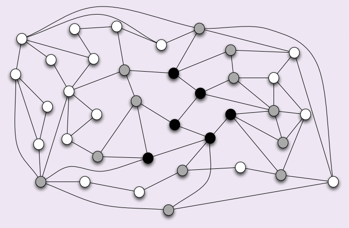


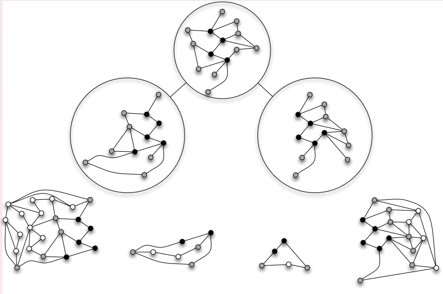
First alternative: constraint bags' structure

Simple algorithm

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

- increase an induced path P
  until $N[P]$ (nodes of P and their neighbors) separates the graph
- "apply recursively to the connected components"





First alternative: constraint bags' structure

chordality, $ch(G)$: max. length of induced cycle in G
k-chordal $\Leftrightarrow ch(G) \leq k$

Bags have *induced dominating path*

[Kosowski,Li,N.,Suchan, Algorithmica 2015]

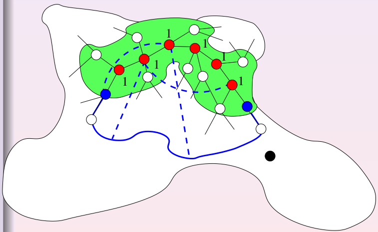
- efficient algorithm $O(m^2)$ in m-edge graphs

  based on DFS from u_0 such that all paths from u_0 are induced

  - either returns an induced cycle larger than k,
  - or compute a tree-decomposition with each bag being the closed neighborhood of an induced path of length $\leq k-1$.

- achieves a tree-decomposition of width $O(\Delta k)$ for k-chordal graphs

(improves [Bodlaender,Thilikos'97] result)



First alternative: constraint bags' structure

chordality, $ch(G)$: max. length of induced cycle in G

k-chordal $\Leftrightarrow ch(G) \leq k$

Bags have *induced dominating path* [Kosowski,Li,N.,Suchan, Algorithmica 2015]

- efficient algorithm $O(m^2)$ in m-edge graphs

  based on DFS from u_0 such that all paths from u_0 are induced

  - either returns an induced cycle larger than k,
  - or compute a tree-decomposition with each bag being the closed neighborhood of an induced path of length $\leq k-1$.

- achieves a tree-decomposition of width $O(\Delta k)$ for k-chordal graphs

  (improves [Bodlaender,Thilikos'97] result)

- **further work:** implement it and analyze its performance in practice

  How to improve it? (guide the DFS)

Bags have *bounded chromatic number* [Seymour 16]

using similar techniques: triangle-free k-chordal graphs have a tree-decomposition with bags with chromatic number $O(k)$

2nd approach: constraint bags' metric properties

Treelength [Dourisboure,Gavoille 07]

Length of $(T, \mathcal{X})$: largest diameter (in G) of a bag $\max_{u,v \in B} dist_G(u,v)$
Treelength of a graph G, $t\ell(G)$: min length over all tree-decompositions.

Applications

- upper bound on the hyperbolicity
- PTAS for TSP in bounded tree-length graphs [Krauthgamer,Lee 06]
- compact routing, greedy routing with low additive stretch [Dourisboure 05] [Boguna,Papadopoulos,Krioukov 10]

2nd approach: constraint bags' metric properties

Treelength [Dourisboure,Gavoille 07]

Length of $(T,\mathcal{X})$: largest diameter (in G) of a bag $\max_{u,v \in B} dist_G(u,v)$

Treelength of a graph G, $t\ell(G)$: min length over all tree-decompositions.

Complexity of tree-length and tree-breadth

Bad news

- NP-hard to decide if $t\ell(G) \leq 2$ (not FPT) [Lokshtanov 10]

Good news

- efficient 3-approximation for $t\ell(G)$ in time $O(nm)$ (using a kind of LexBFS: Lex M) [Dourisboure,Gavoille 07]
- **open question:** better approx?

Approximation for tree-length

[Dourisboure,Gavoille 07]

LexM "kind of LexBFS" [Rose,Tarjan,Lueker 76]
with specific tie-break rules

LexM returns:

- ordering $(v_1, \cdots, v_n)$ of V
- "BFS" rooted in v_1
  $\forall i$, $\exists$ a shortest path from v_1 to v_i with all internal nodes $< i$
- minimal triangulation H of G s.t. $v_i v_j \in E(H) \setminus E(G)$ iff there exists a path from v_i to v_j with all internal nodes $> \max\{i, j\}$

Algorithm LexM
Input: A graph $G = (V, E)$
Result: A supergraph of G: $H = (V, E' \cup E)$
begin
  Assign empty label to all vertices of G and empty set to E';
  for *i from n downto* 1 **do**
    Select:
      pick an unnumbered vertex u with largest label;
      Assign to u the number i: $\alpha(i) = u$;
    Update:
      for *each unnumbered vertex* v *such that there is a chain* $u = w_1, w_2, \ldots, w_{p+1} = v$ *wi*
      w_j *unnumbered and* $\text{label}(w_j) < \text{label}(v)$ *for all* $j \in \{2, \ldots, p\}$ **do**
        add i to label(v);
        add$\{u, v\}$ to E';
      end
  end
end

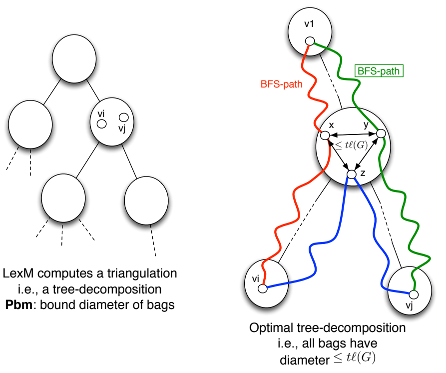
Approximation for tree-length

[Dourisboure,Gavoille 07]

LexM "kind of LexBFS" [Rose,Tarjan,Lueker 76]
with specific tie-break rules

LexM returns:

- ordering $(v_1, \cdots, v_n)$ of V
- "BFS" rooted in v_1
  $\forall i$, $\exists$ a shortest path from v_1 to v_i with all internal nodes $< i$
- minimal triangulation H of G s.t. $v_iv_j \in E(H) \setminus E(G)$ iff there exists a path from v_i to v_j with all internal nodes $> \max\{i,j\}$



Theorem [Dourisboure,Gavoille 07]

LexM returns a tree-decomposition with each bag has diameter at most $3t\ell(G)$.

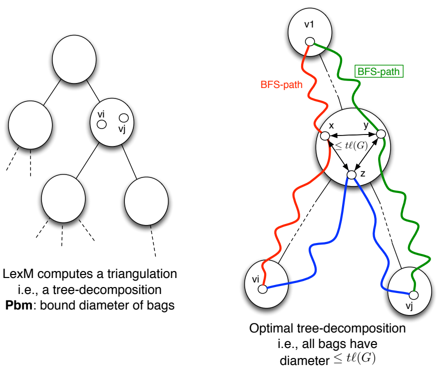
Approximation for tree-length

[Dourisboure,Gavoille 07]

LexM "kind of LexBFS" [Rose,Tarjan,Lueker 76]
with specific tie-break rules

LexM returns:

- ordering $(v_1, \cdots, v_n)$ of V
- "BFS" rooted in v_1
  $\forall i$, $\exists$ a shortest path from v_1 to v_i with all internal nodes $< i$
- minimal triangulation H of G s.t. $v_i v_j \in E(H) \setminus E(G)$ iff there exists a path from v_i to v_j with all internal nodes $> \max\{i, j\}$



Theorem [Dourisboure,Gavoille 07]

LexM returns a tree-decomposition with each bag has diameter at most $3t\ell(G)$.

Can approximation for tree-length be used for treewidth?

Relationship between treewidth and treelength

Treewidth and Treelength are not comparable in general.

- $2 = tw(C_n) < t\ell(C_n) = \lceil n/3 \rceil$ in any cycle C_n $(n > 6)$
- $n - 1 = tw(K_n) > t\ell(K_n) = 1$ in any clique K_n $(n > 2)$

Are these two cases (large cycles and large cliques) the only two problems?

Relationship between treewidth and treelength

Treewidth and Treelength are not comparable in general.

- $2 = tw(C_n) < t\ell(C_n) = \lceil n/3 \rceil$ in any cycle C_n $(n > 6)$
- $n - 1 = tw(K_n) > t\ell(K_n) = 1$ in any clique K_n $(n > 2)$

Are these two cases (large cycles and large cliques) the only two problems?

Somehow YES... [Coudert,Ducoffe,N., SIDMA 2016]

- $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor tw(G)$ with $\ell(G)$ the length of a largest **isometric** cycle in G
- $tw(G) = O(g^{3/2}) t\ell(G)$ with g the **genus** of G

A subgraph H of G is **isometric** if, for any $u, v \in V(H)$, $dist_H(u,v) = dist_G(u,v)$.
(i.e., the distances do not increase in H)

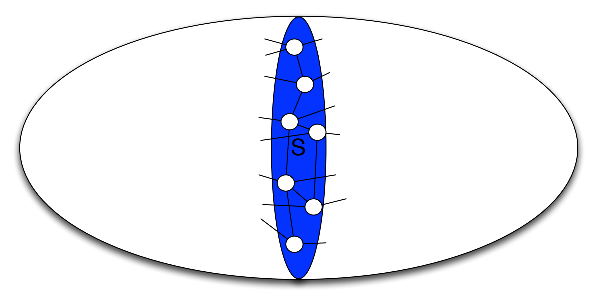
Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

To prove the Theorem, first we prove

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

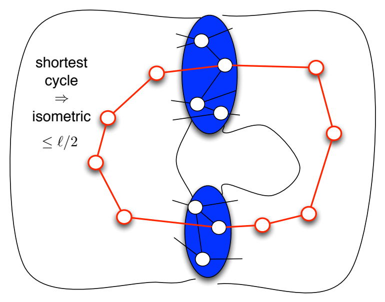


Clear if S is connected: $diam(S) \leq |S| - 1 \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

For this purpose: When at least 2 components: join them by paths of length $\leq \lfloor \frac{\ell(G)}{2} \rfloor$

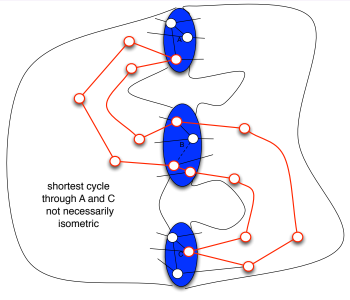


"easy" if S has 2 connected components

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

For this purpose: When at least 2 components: join them by paths of length $\leq \lfloor \frac{\ell(G)}{2} \rfloor$



does not "work" anymore for more components :(

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

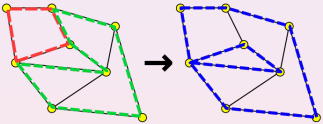
Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

For this purpose: When at least 2 components: join them by paths of length $\leq \lfloor \frac{\ell(G)}{2} \rfloor$

Tool:

Cycle space: $\mathcal{C}(G)$ **set of Eulerian subgraphs of** G

[folklore] Any Eulerian subgraph can be obtained as symmetric difference Δ of cycles.



Theorems:[?] $(\mathcal{C}(G), \Delta)$ is a vector space

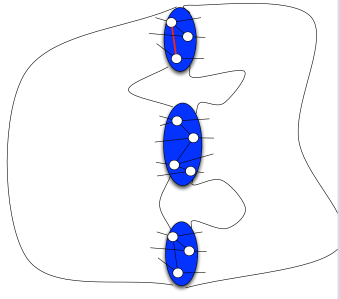
- with dimension $m - n + 1$,
- $\exists$ a basis with cycles of length $\leq \ell(G)$

Let $\mathcal{G}_\ell$ be the set of graphs such that $(\mathcal{C}(G), \Delta)$ has a basis of cycles of length $\leq \ell$.

Sketch of $t\ell(G) \leq \lfloor\frac{\ell(G)}{2}\rfloor(tw(G)-1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor\frac{\ell(G)}{2}\rfloor(|S|-1)$

For this purpose: When at least 2 components: join them by paths of length $\leq \lfloor\frac{\ell(G)}{2}\rfloor$



lemma 2: $\mathcal{G}_\ell$ stable by contraction

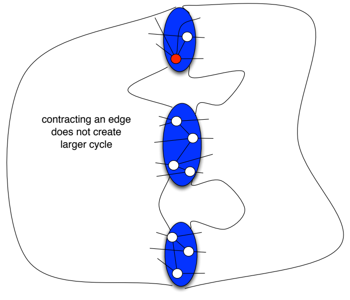
proof: $G \in \mathcal{G}_\ell$ for $\ell = \ell(G)$

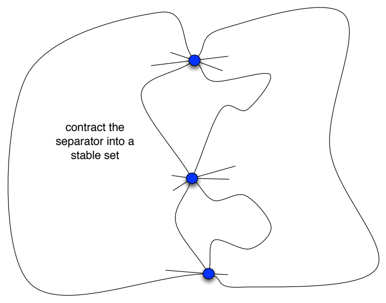
- $\mathcal{B} = C_1, \cdots, C_d$ cycle basis
  (cycles of length $\leq \ell(G)$)
  $d = dim(\mathcal{C}(G)) = m - n + 1$
- $e \in E(G)$ ($k = \#$ triangles containing e)
  $t = dim(\mathcal{C}(G/e)) = d - k$
- $C'_1, \cdots, C'_t$ obtained from $\mathcal{B}$ by contracting e and removing the triangles
- prove that $C'_1, \cdots, C'_t$ is linearly independent
  $\Rightarrow$ it is a basis of cycles of length $\leq \ell$
- $G/e \in \mathcal{G}_\ell$

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

For this purpose: When at least 2 components: join them by paths of length $\leq \lfloor \frac{\ell(G)}{2} \rfloor$



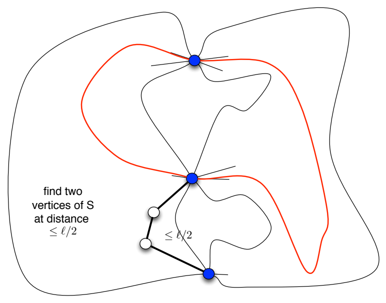


By Lemma 2, contracting each component of S, the obtained graph remains in $\mathcal{G}_\ell$ and S remains a minimal separator

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

For this purpose: When at least 2 components: join them by paths of length $\leq \lfloor \frac{\ell(G)}{2} \rfloor$



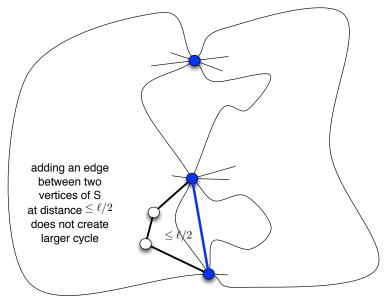
Lemma 3: If S is a minimal stable-set separator, $\exists$ 2 vertices of S at distance $\leq \lfloor \frac{\ell(G)}{2} \rfloor$

proof: take a cycle C "crossing" the separator; $C = \oplus_i C_i$, where the C_i's are cycles (of the basis) of length $\leq \ell$. One of them "crosses" the separator

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

For this purpose: When at least 2 components: join them by paths of length $\leq \lfloor \frac{\ell(G)}{2} \rfloor$



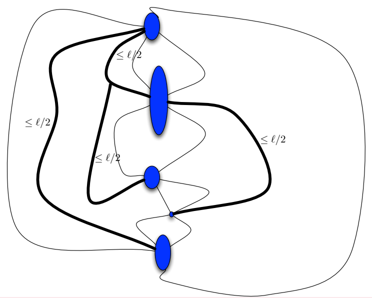
Last step of the proof

We can add an edge between 2 vertices of S at distance $\leq \ell/2$ (Lemma 3).
It does not create larger cycle (proof using cycle space) and S still a minimal separator.
# of connected components of S have been reduced $\Rightarrow$ proceed recursively

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

For this purpose: When at least 2 components: join them by paths of length $\leq \lfloor \frac{\ell(G)}{2} \rfloor$



The result follows !

Sketch of $t\ell(G) \leq \lfloor \frac{\ell(G)}{2} \rfloor (tw(G) - 1)$

Lemma 1: For any minimal separator S of G, $diam(S) \leq \lfloor \frac{\ell(G)}{2} \rfloor (|S| - 1)$

Proof of the Theorem

Now take any tee-decomposition resulting from a minimal triangulation for any bag B and $u, v \in B$, if $\{u, v\} \notin E$, u and v belong to a minimal separator

[Bouchitté, Todinca 01]

Lemma 1 $\Rightarrow$ the diameter of the bag is $\leq \lfloor \frac{\ell(G)}{2} \rfloor (|B| - 1)$

Sketch of $tw(G) = O(g^{3/2})t\ell(G)$ Apply bi-dimensionality

if $tw(G)$ "small" OK

otherwise:

Theorem [Demain,Hajiaghayi,Thilikos 06]

Let G be a graph with genus g and $tw(G) > 4k(g+1)$ with $k \geq 12g$, then G contains a $(k-12g, g)$-gridoid as a contraction.

a (k, g)-gridoid is partially triangulated $(k \times k)$-grid in with g extra edges

proof of our result

- $t\ell((k, g)$-gridoid$) \geq$ function of k and g
- hence, $t\ell(G) \geq$ function of k and g (since $t\ell$ close under contraction)
- $\Rightarrow$ if G (with genus g) has "large" treewidth, it has "large" tree-length

3rd alternative: Tree-breadth

[Dragan,Abu-Ata 14]

Breadth of $(T,\mathcal{X})$: largest excentricity (in G) of a bag $\min_{v \in G} \max u \in B dist_G(u,v)$
Tree-breadth of a graph G, $tb(G)$: min breadth over all tree-decompositions.

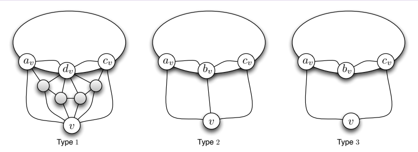
$tb(G) \leq t\ell(G) \leq 2tb(G)$ for any graph G

Complexity of tree-breadth was open

- **Good news:** efficient 3-approximation for $tb(G)$ [Dragan,Abu-Ata 14]

Our results [Ducoffe,Legay,N. 16]

- **Bad news:** NP-hard to decide if $tb(G) \leq 1$ (not FPT)
- Deciding $tb(G) \leq 1$ can be solved in polynomial-time in



- bipartite graphs (linear time)
- $K_{3,3}$ minor-free graphs ($O(n^3m)$ time)

Further work

Evaluation of our algorithms for treewidth

implement the DFS-algorithm for treewidth in real large networks

turn our results on treewidth/treelength into a constructive algorithm
i.e., how to use an algo that approx treelength to approx treewidth?

Computing Treelength and Tree-breadth

better approximation for treelength? for tree-breadth?

complexity of deciding $tb(G) \leq k$ or $t\ell(G) \leq k$ in planar graphs?

Other interesting bag's properties to be investigated?

Design an efficient algorithm to compute "good" tree-decompositions in practice

Further work

Evaluation of our algorithms for treewidth

implement the DFS-algorithm for treewidth in real large networks

turn our results on treewidth/treelength into a constructive algorithm
i.e., how to use an algo that approx treelength to approx treewidth?

Computing Treelength and Tree-breadth

better approximation for treelength? for tree-breadth?

complexity of deciding $tb(G) \leq k$ or $t\ell(G) \leq k$ in planar graphs?

Other interesting bag's properties to be investigated?

Design an efficient algorithm to compute "good" tree-decompositions in practice

谢谢