# Tree-decompositions with metric properties on the bags

Nicolas Nisse

Inria, France

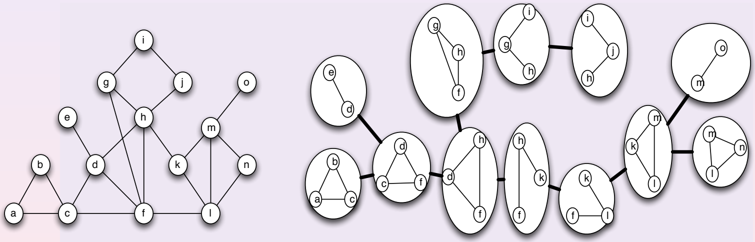
Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, Sophia Antipolis, France

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

2nd Brazilian-French Workshop GCO, Redonda, March 2016

# Tree/Path-Decompositions [Robertson and Seymour 83]

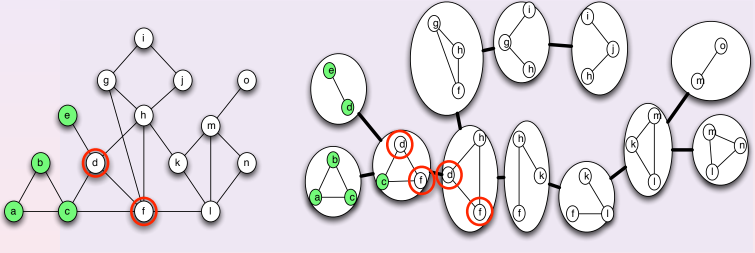
Representation of a graph as a Tree preserving connectivity properties



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]

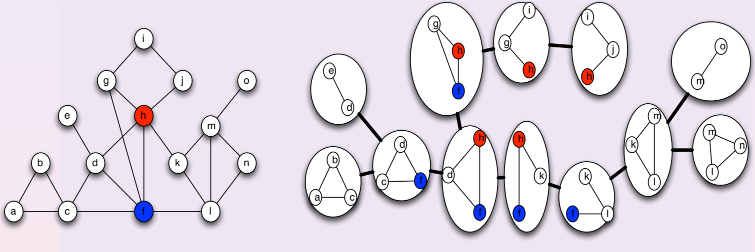
Representation of a graph as a Tree preserving connectivity properties



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

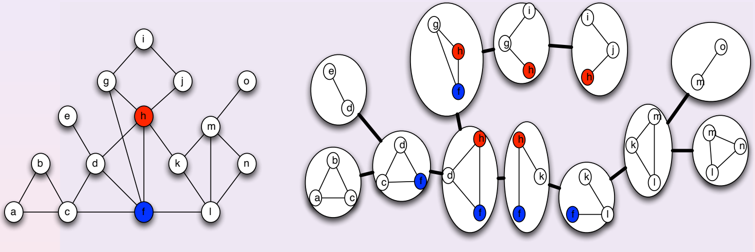



Tree $T$ + family $\mathcal{X} = (X_t)_{t \in V(T)}$ of "bags" (set of vertices of $G$)
**Important**: intersection of two adjacent bags = separator of $G$

- $\bigcup_{t \in V(T)} X_t = V(G)$;
- 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

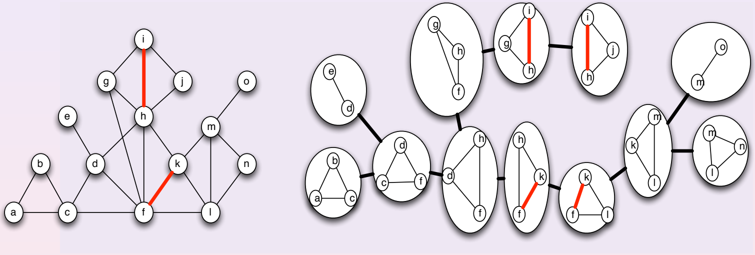



Tree $T$ + family $\mathcal{X} = (X_t)_{t \in V(T)}$ of "bags" (set of vertices of $G$)
**Important**: intersection of two adjacent bags = separator of $G$

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 Applications

- cornerstone of Graph Minors Theorem [Robertson and Seymour 1983-2004]
- problems expressible in MSOL solvable in polynomial time in graphs of bounded treewidth (dynamic programming) [Courcelle, 90]
- 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 [Arnborg,Corneil,Prokurowski 87]
- FPT: algorithm in $O(2^{k^3} n)$ [Bodlaender,Kloks 96]
- "practical" algorithms only for graph with treewidth $\leq 4$ 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

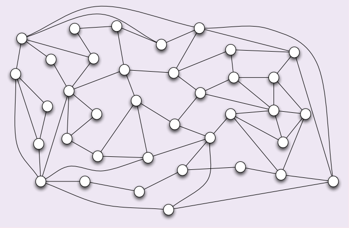
1. bags' structure
2. bags' diameter: treelength
3. bags' radius: tree-breadth

and relationships between them...

# First alternative: constraint bags' structure

## Simple algorithm [Kosowski,Li,N.,Suchan 15]

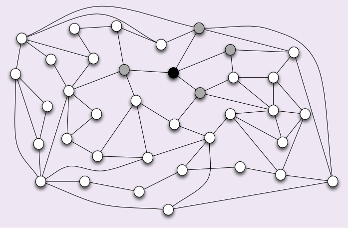
- increase an induced path $P$
  until $N[P]$ (nodes of $P$ and their neighbors) separates the graph
- "apply recursively to the connected components"



# First alternative: constraint bags' structure

## Simple algorithm [Kosowski,Li,N.,Suchan 15]

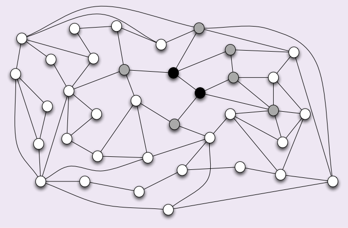
- increase an induced path $P$
  until $N[P]$ (nodes of $P$ and their neighbors) separates the graph
- "apply recursively to the connected components"



# First alternative: constraint bags' structure

## Simple algorithm [Kosowski,Li,N.,Suchan 15]

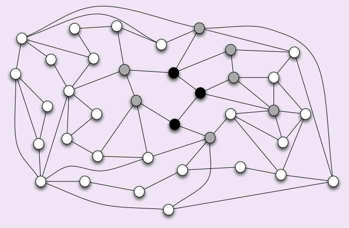
- increase an induced path $P$
  until $N[P]$ (nodes of $P$ and their neighbors) separates the graph
- "apply recursively to the connected components"



# First alternative: constraint bags' structure

## Simple algorithm [Kosowski,Li,N.,Suchan 15]

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



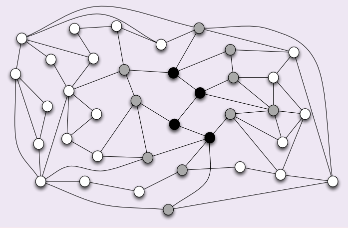
# First alternative: constraint bags' structure

## Simple algorithm

[Kosowski,Li,N.,Suchan 15]

- increase an induced path $P$

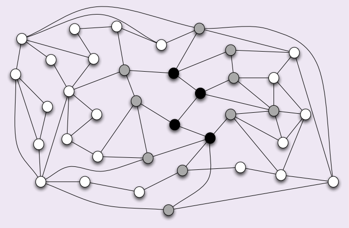
  until $N[P]$ (nodes of $P$ and their neighbors) separates the graph
- "apply recursively to the connected components"

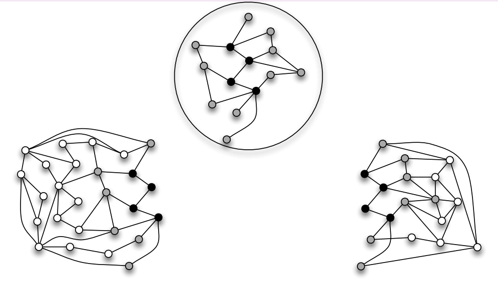


# First alternative: constraint bags' structure

## Simple algorithm [Kosowski,Li,N.,Suchan 15]

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

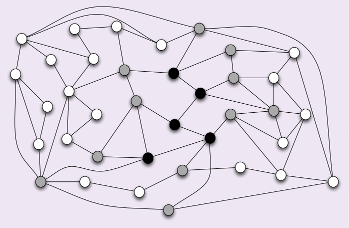


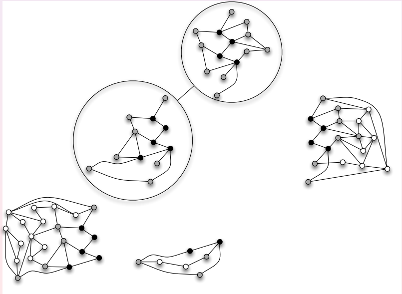


# First alternative: constraint bags' structure

## Simple algorithm [Kosowski,Li,N.,Suchan 15]

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

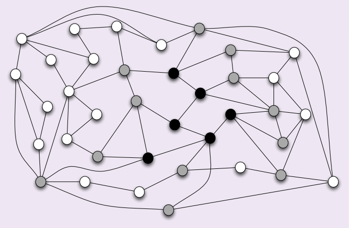


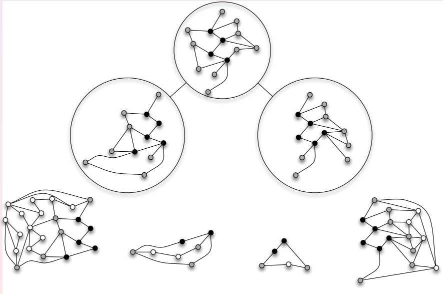


# First alternative: constraint bags' structure

## Simple algorithm [Kosowski,Li,N.,Suchan 15]

- 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 15]

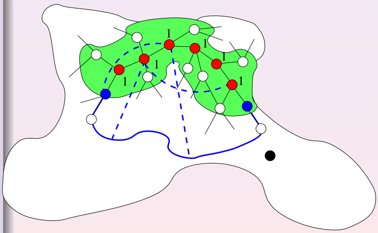
- efficient algorithm $O(m^2)$ in $m$-edge graphs

  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 15]

- 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* $\mathrm{label}(w_j) < \mathrm{label}(v)$ *for all* $j \in \{2, \ldots, p\}$ **do**
        add $i$ to label$(v)$;
        add$\{u, v\}$ to $E'$;
      **end**
  **end**
**end**

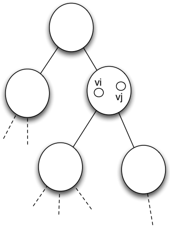
# Approximation for tree-length

[Dourisboure,Gavoille 07]

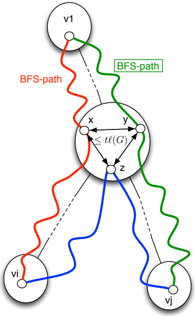
LexM "kind of LexBFS" [Rose,Tarjan,Lueker 76] with specific tie-break rules

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. $v_i v_j \in E(H) \setminus E(G)$ iff there exists a path from $v_i$ to $v_j$ with all internal nodes $> \max\{i, j\}$




LexM computes a triangulation
i.e., a tree-decomposition
**Pbm**: bound diameter of bags




Optimal tree-decomposition
i.e., all bags have
diameter $\leq t\ell(G)$

**Theorem** [Dourisboure,Gavoille 07]

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

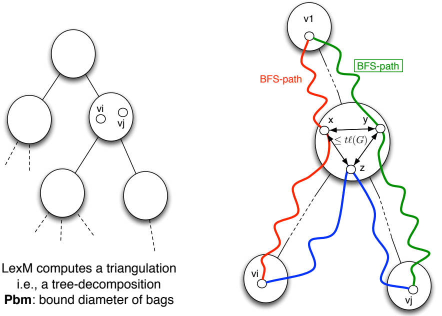
# Approximation for tree-length

[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 $> \max\{i, j\}$




LexM computes a triangulation
i.e., a tree-decomposition
**Pbm**: bound diameter of bags

Optimal tree-decomposition
i.e., all bags have
diameter $\leq t\ell(G)$

**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.

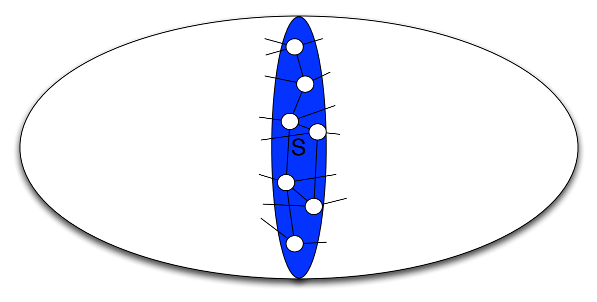
$t\ell(K_n) = 1$, $tw(K_n) = n-1$ and $tw(C_n) = 2$, $t\ell(C_n) = \lceil n/3 \rceil$

**But in some cases...** [Coudert,Ducoffe,N. 16]

- $t\ell(G) \leq \frac{\lfloor \ell(G) \rfloor}{2}(tw(G)-1)$ in graphs with no isometric cycle larger than $\ell(G)$
- $tw(G) = O(g^{3/2})t\ell(G)$ in graphs with genus $\leq g$

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

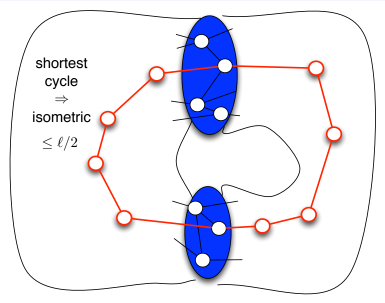
**Theorem:** For any minimal separator $S$, $diam(S) \leq \frac{\lfloor \ell(G) \rfloor}{2}(|S| - 1)$




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

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

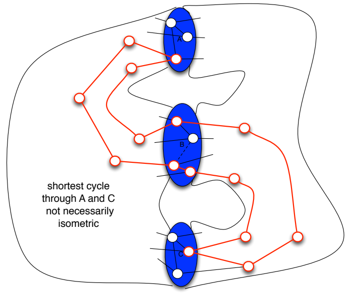
**Theorem:** For any minimal separator $S$, $diam(S) \leq \frac{\lfloor \ell(G) \rfloor}{2}(|S|-1)$




"easy" if $S$ has 2 connected components

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

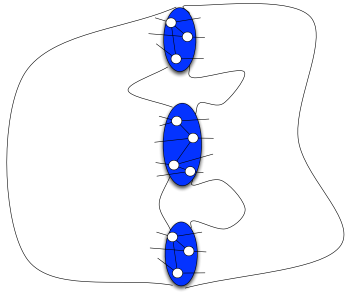
**Theorem:** For any minimal separator $S$, $diam(S) \leq \frac{\lfloor \ell(G) \rfloor}{2}(|S|-1)$




does not "work" anymore for more components :(

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

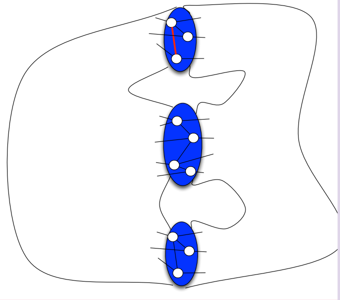
**Theorem:** For any minimal separator $S$, $diam(S) \leq \frac{\lfloor \ell(G) \rfloor}{2}(|S| - 1)$



**Tool: Cycle space:** vector space: eulerian subgraphs, symmetric difference of edges.
**Theorems:** dimension $m - n + 1$, $\exists$ a basis with cycles of length $\leq \ell(G)$

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

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



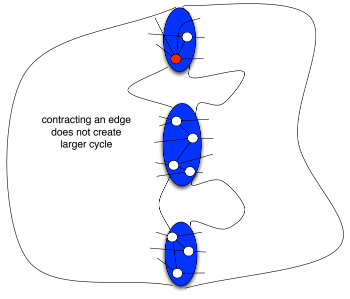
**lemma:** $\ell(G)$ stable by contraction

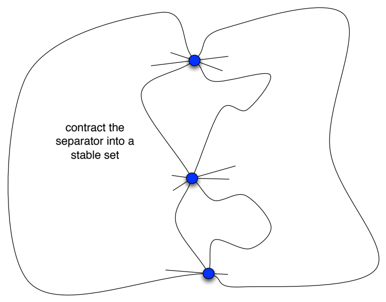
**proof:**

- $\mathcal{B} = C_1, \cdots, C_d$ 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
- $\ell(G/e) \leq \ell(G)$

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

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

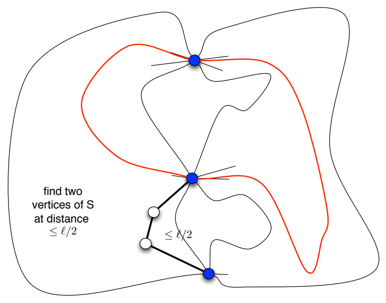



Contracting each component of $S$ does not increase $\ell(G)$
and $S$ remains a minimal separator

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

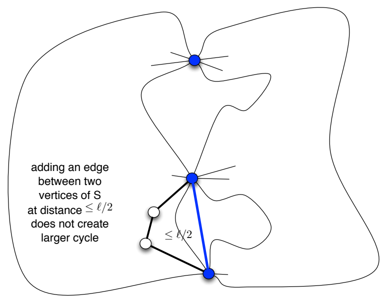
**Theorem:** For any minimal separator $S$, $diam(S) \leq \frac{\lfloor \ell(G) \rfloor}{2}(|S|-1)$




**Lemma:** If $S$ is a minimal separator ans stable, $\exists$ two vertices of $S$ at distance $\leq \ell(G)$
**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 $t\ell(G) \leq \frac{\lfloor \ell(G) \rfloor}{2}(tw(G)-1)$

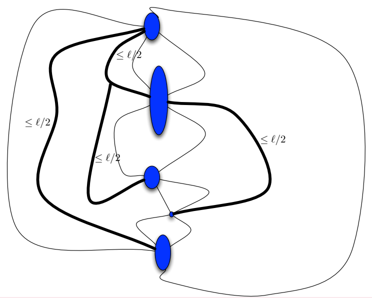
**Theorem:** For any minimal separator $S$, $diam(S) \leq \frac{\lfloor \ell(G) \rfloor}{2}(|S|-1)$




We can add an edge between 2 vertices of $S$ at distance $\leq \ell/2$.
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 $t\ell(G) \leq \frac{\lfloor \ell(G) \rfloor}{2}(tw(G)-1)$

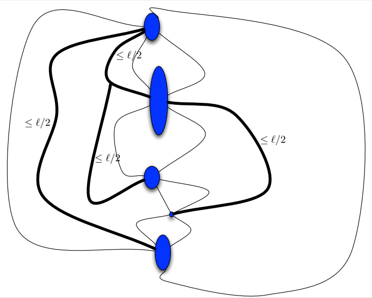
**Theorem:** For any minimal separator $S$, $diam(S) \leq \frac{\lfloor \ell(G) \rfloor}{2}(|S|-1)$




The result follows !

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

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




The result follows !

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]

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

## 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

# 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

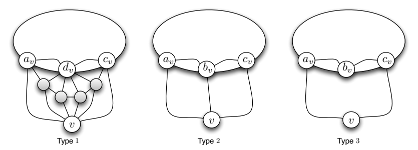
- **Good news:** efficient 3-approximation for $tb(G)$ [Dragan,Abu-Ata 14]

## Our results

[Ducoffe,Legay,N. 16]

$tb(G) = 1 \Leftrightarrow \exists$ a tree-decomposition where each bag has a dominating vertex

- Deciding $tb(G) \leq 1$ is at least as hard as deciding $tb(G) \leq r$
- **Bad news:** NP-hard to decide if $tb(G) \leq 1$ (not FPT)
- Deciding $tb(G) \leq 1$ can be solved in time $O(nm)$ in
  - bipartite graphs
  - planar graphs



# 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 $tb(G) \leq k$? 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 $tb(G) \leq k$? in planar graphs?

Other interesting bag's properties to be investigated?

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

**Muito Obrigado!**