

5th Lecture : Modular decomposition MPRI 2013–2014

Michel Habib
 habib@liafa.univ-Paris-Diderot.fr
<http://www.liafa.univ-Paris-Diderot.fr/~habib>

Sophie Germain, 22 octobre 2013

Schedule

- Introduction
- Graph searches
- Applications of LBFS on structured graph classes
- Chordal graphs
- Cograph recognition
- A nice conjecture

Examen le mardi 26 novembre de 9h à 12h
 Salle habituelle

A hierarchy of graph models

1. Undirected graphs (graphes non orientés)
2. Tournaments (Tournois), sometimes 2-circuits are allowed.
3. Signed graphs (Graphes signés) each edge is labelled + or - (for example friend or enemy)
4. Oriented graphs (Graphes orientés), each edge is given a unique direction (no 2-circuits)
 An interesting subclass are the DAG Directed Acyclic Graphs (graphes sans circuit), for which the transitive closure is a partial order (ordre partiel)
5. Partial orders and comparability graphs an interesting particular case.
 Duality comparability – cocomparability (graphes de comparabilité – graphes d'incomparabilité)
6. Directed graphs or digraphs (Graphes dirigés)

The problem has to be defined in each model and sometimes it could be hard.

- ▶ What is the right notion for a coloration in a directed graph ?
- ▶ No directed cycle unicolorated, seems to be the good one.
- ▶ It took 20 years to find the right notion of oriented matroid
- ▶ What is the right notion of treewidth for directed graphs ?
- ▶ Still an open question. It seems that all tentative definitions loose many properties of the undirected case treewidth.

For partial orders, comparability graphs or uncomparability graphs the independant set and maximum clique problems are polynomial.

<http://math.nie.edu.sg/fmdong/Research/articles/beautiful>

Second Neighbourhoods Conjecture

P.D. Seymour 1990

Every digraph without 2-circuits has a vertex with at least as many second neighbours as first neighbours.

Second neighbours, $SN(x)$ is the set of vertices at exact distance 2 of x .

Therefore we are looking for x such that $|SN(x)| \geq |N(x)|$.

- ▶ If G has a sink then the results is true.
- ▶ So the conjecture is true for DAGs.
- ▶ The interesting case is for strongly connected graphs.

Another nice result on degrees : the politician's theorem

Characterization

Let G a connected undirected graph $|G| \geq 3$ such that for every $x, y \in N(G)$, we have $|N(x) \cap N(y)| = 1$
 then G is a star of triangles (a windmill graph).

The politician version

Suppose in a group of at least three people we have the situation that any pair of persons have precisely one common friend. Then there exists always a person (the politician) who is everybody friend.

Paul Erdős, Alfred Rényi and Vera Sòs' proof

- ▶ $|V(G)| = 3$, only the triangle which is a windmill graph satisfies the degree condition.
- ▶ G has no induced C_4 .
- ▶ For every pair x, y of non universal vertices, necessarily $d(x) = d(y) = k$.
- ▶ Using some argument from algebra, we obtain $k = 2$ and the existence of an universal vertex.
- ▶ For the complete proof, see : "Proofs from the BOOK", by Martin Aigner and Günter M. Ziegler, Springer-Verlag, \geq Second Edition.

Graph searches are very well known and often used :

1. "Fil d'ariane" in the Greek mythology.
2. Euler (1735) for solving the famous walk problem in Koenisberg
3. Tremaux (1882) and Tarry (1895) introducing DFS to solve maze problems
4. Fleury, proposed a nice algorithm to compute an Euler Tour, cited in E. Lucas, *Récréations mathématiques, Paris, 1891*.
5. Computer scientists from 1950, in particular in the 70's, R.E. Tarjan for new applications of DFS...
6. 4 points characterizations Corneil, Krueger (2008), and the definition of LDFS a new interesting basic search.

Some definitions

Graph Search

The graph is **supposed to be connected** so as the set of visited vertices. After choosing an initial vertex, a search of a connected graph visits each of the vertices and edges of the graph such that a new vertex is visited only if it is adjacent to some previously visited vertex.

At any point there may be several vertices that may possibly be visited next. To choose the next vertex we need a tie-break rule. The breadth-first search (BFS) and depth-first search (DFS) algorithms are the traditional strategies for determining the next vertex to visit.

Variations

Graph Traversal more or less equivalent to graph search

The set of visited vertices is not supposed to be connected (used for computing connected components for example)

Graph Searching for cops and robbers games on a graph

The name Graph searching is also used in this context, with a slightly different meaning. Relationships with width graph parameters such as treewidth.

Our main question

Main Problem

What kind of knowledge can we learn about the structure of a given graph via graph searching (i.e. with one or a series of successive graph searches) ?

Goals

- ▶ Building bottom up graph algorithms from well-known graph searches
- ▶ Develop basic theoretic tools for the structural analysis of graphs
- ▶ **Applications on huge graphs :**
 No need to store sophisticated data structures, just some labels on each vertex,

Lexicographic Breadth First Search (LBFS)

Data: a graph $G = (V, E)$ and a start vertex s

Result: an ordering σ of V

Assign the label \emptyset to all vertices

$label(s) \leftarrow \{n\}$

for $i \leftarrow n$ **à** 1 **do**

Pick an unnumbered vertex v **with lexicographically largest label**

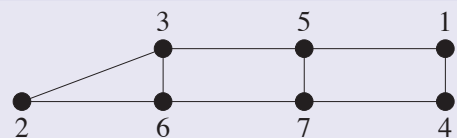
$\sigma(i) \leftarrow v$

foreach unnumbered vertex w adjacent to v **do**

$label(w) \leftarrow label(w). \{i\}$

end

end



It is just a breadth first search with a tie break rule.
 We are now considering a characterization of the order in which a LBFS explores the vertices.
 Before let us implement it using partition refinement in linear time.

Algorithm LEXBFS(G, τ)

Input : A graph $G = (V, E)$ and an initial ordering τ of the vertices.
Output : An ordering σ of the vertices of G .

1. $L \leftarrow (V)$; $i \leftarrow 1$;
2. **while** $\exists P_i \neq \emptyset$ in $L = (P_1, \dots, P_k)$ **do**
3. Let P_i be the leftmost nonempty cell
4. Remove the first vertex x (smallest with respect to τ) from P_i
5. $\sigma(x) \leftarrow i$; $i \leftarrow i + 1$;
6. **for** each cell $P_j, j \geq 1$ **do**
7. Let $P' = \{v | v \in N(x) \cap P_j\}$;
8. **if** P' is nonempty and $P' \neq P_j$, **then**
9. Remove P' from P_j
10. Insert P' to the left of P_j in L
11. **end for**
12. **end while**
13. **return** (σ)

An example

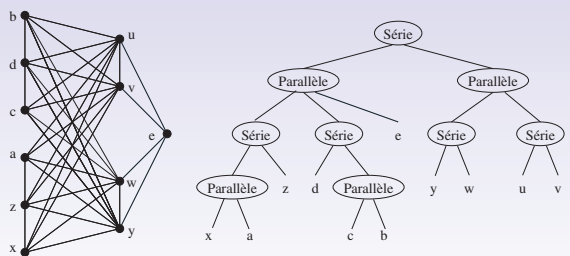


TABLE: Step by step LexBFS of G . The resulting ordering is $\sigma : x y w z u v a d c b e$.

$\sigma(\alpha)$	α	$N'(\alpha)$	Cells
			x d y u e v w c a z b
1	x	{y u v w z}	y u v w z d e c a b
2	y	{w z d e c a b}	w z u v d e c a b
3	w	{z d e c a b}	z u v d e c a b
4	z	{u v a}	u v a d e c b
5	u	{v a d e c b}	v a d e c b
6	v	{a d e c b}	a d e c b
7	a	{}	d e c b
8	d	{c b}	c b e
9	c	{}	b e
10	b	{}	e
11	e	{}	

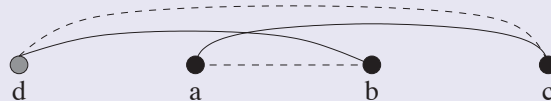
Consequences

- Using partition refinement allows to avoid the management of the labels
 The vertices with lexicographic maximum labels belongs necessarily to the right most part.
- LBFS can be implemented in $O(|V(G)| + |E(G)|)$

LBFS orderings of the vertices

Property (LexB)

For an ordering σ on V , if $a <_\sigma b <_\sigma c$ and $ac \in E$ and $ab \notin E$, then it must exist a vertex d such that $d <_\sigma a$ et $db \in E$ et $dc \notin E$.



Theorem

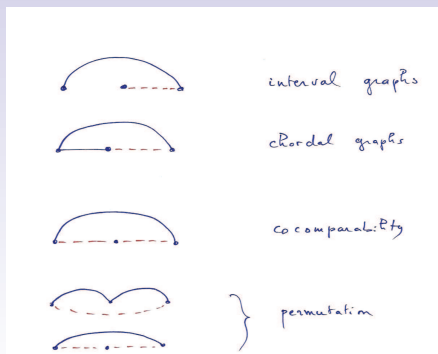
For a graph $G = (V, E)$, an ordering σ sur V is a LBFS of G iff σ satisfies property (LexB).

Importance of 4 points conditions for graph class recognition

Many classes of graphs or partial orders can be characterized by the existence of a particular ordering of the vertices with some forbidden configuration on three points.
 Examples with forbidden configuration on three points :

1. Interval graphs : ordering of the left ends of the intervals.
2. Chordal : simplicial elimination ordering.
3. Co-comparability : transitivity violation of the complement graph
4. Permutation : transitivity violation of the graph and its complement.

Forbidden 3 points suborderings



Consequences

LexBFS is involved in many recognition algorithms for these classes of graphs.

- ▶ Apply a LexBFS on \overline{G} giving an ordering σ
- ▶ If G is a comparability graph the last vertex of σ , can be taken as a source in a transitive orientation of G .
- ▶ The starting point for comparability and permutation graph recognition algorithms.

Seminal paper

D.G. Corneil et R. M. Krueger, A unified view of graph searching, SIAM J. Discrete Math, 22, Num 4 (2008) 1259-1276
 In which characterizations of the orderings yielded by well-known graph searches are provided.
 Namely : Generic Search, DFS, BFS, LBFS, LDFS

Why LBFS behaves so nicely on well-structured graphs

A nice recursive property

On every tie-break set S , LBFS operates on $G(S)$ as a legitimate LBFS.

proof

Consider $a, b, c \in S$ such that $a <_{\sigma} b <_{\sigma} c$ and $ac \in E$ and $ab \notin E$, then it must exist a vertex d such that $d <_{\sigma} a$ et $db \in E$ et $dc \notin E$. But then necessarily $d \in S$.

Remark

Analogous properties are false for other classical searches.

LexBFS versus LBFS !

Google Images query : LBFS (thanks to Fabien)
 yields :

First Answer



Definition

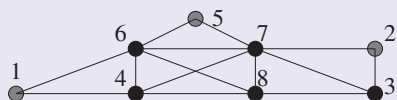
A graph is a chordal graph if every cycle of length ≥ 4 has a chord.
 Also called triangulated graphs, (cordaux in french)

1. First historical application : perfect phylogeny.
2. Many NP-complete problems for general graphs are polynomial for chordal graphs.
3. Second application : graph theory. Treewidth (resp. pathwidth) are very important graph parameters that measure distance from a chordal graph (resp. interval graph).

Two Basic facts

1. Chordal graphs are hereditary
2. Interval graphs are chordal

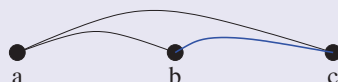
Chordal graph



A vertex is simplicial if its neighbourhood is a clique.

Simplicial elimination scheme

$\sigma = [x_1 \dots x_i \dots x_n]$ is a simplicial elimination scheme if x_i is simplicial in the subgraph $G_i = G[\{x_j \dots x_n\}]$



A characterization theorem for chordal graphs

Theorem

Dirac 1961, Fulkerson, Gross 1965, Gavril 1974, Rose, Tarjan, Lueker 1976.

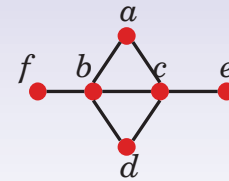
For a connected graph G the following items are equivalent :

- (i) G is chordal (every cycle of length ≥ 4 has a chord).
- (ii) G has a simplicial elimination scheme
- (iii) Every minimal separator is a clique

Minimal Separators

A subset of vertices S is a **minimal separator** if G if there exist $a, b \in G$ such that a and b are not connected in $G - S$.
 and S is minimal for inclusion with this property .

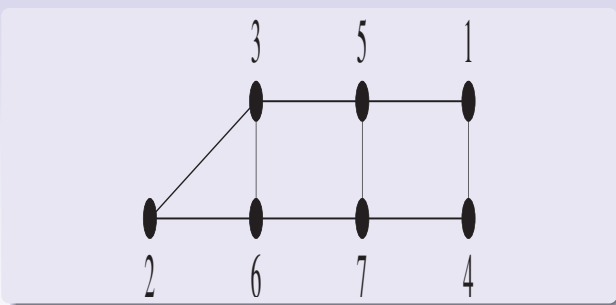
An example



3 minimal separators $\{b\}$ for f and a , $\{c\}$ for a and e and $\{b, c\}$ for a and d .

If $G = (V, E)$ is connected then for every $a, b \in V$ such that $ab \notin E$ then there exists at least one minimal separator.
 But there could be an **exponential number of minimal separators**.
 Consider 2 stars a, x_1, \dots, x_n (centered in a) and b, y_1, \dots, y_n (centered in b) and then add all the edges $x_i y_i$ for $1 \leq i \leq n$.
 There exist 2^n minimal separators for the vertices a and b .

Proof of the theorem



Theorem [Tarjan et Yannakakis, 1984]

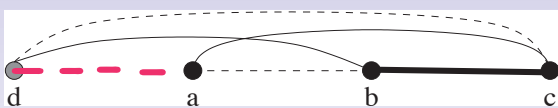
G is chordal iff every LexBFS ordering yields a simplicial elimination scheme.

Proof :

Let c be a non simplicial vertex.

There exist $a < b \in N(c)$ avec $ab \notin E$.

Using characterization of LexBFS orderings, it exists $d < a$ with $db \in E$ and $dc \notin E$. Since G is chordal, necessarily $ad \notin E$.



But then from the triple d, a, b , it exists $d' < d$ with $d'a \in E$ and $d'b \notin E$. Furthermore $d'd \notin E \dots$
 And using the triple d', d, a , we start an infinite chain

Remark

Most of the proofs based on some characteristic ordering of the vertices are like that, with no extra reference to the algorithm itself.

Chordal graphs recognition so far

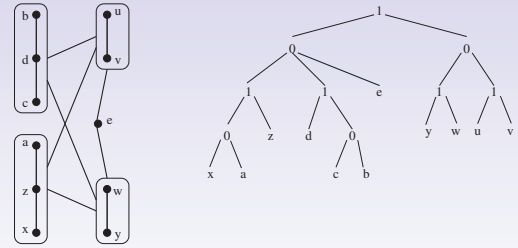
Chordal graph recognition

1. Apply a LexBFS on G $O(n+m)$
2. Check if the reverse ordering is a simplicial elimination scheme $O(n+m)$
3. In case of failure, exhibit a certificate : i.e. a cycle of length ≥ 4 , without a chord. $O(n)$

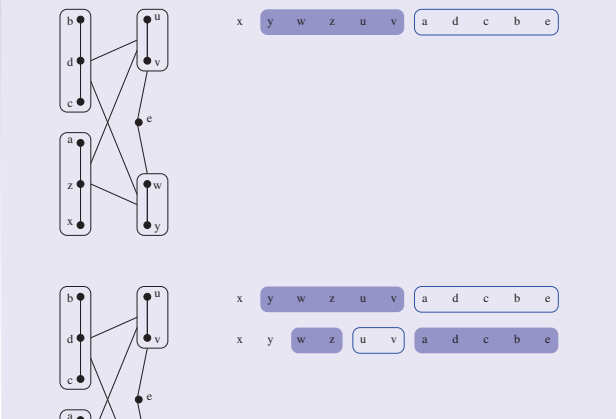
The ultimate Algorithm Using LexBFS ?

A cograph recognition algorithm [BCHP03]

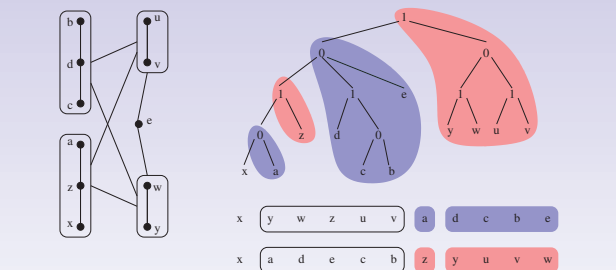
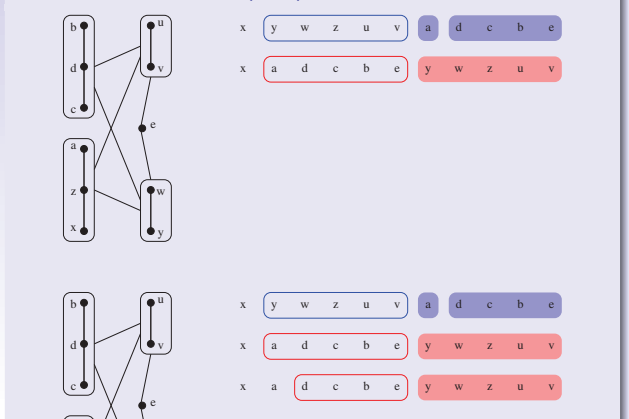
1. $\sigma \leftarrow \text{LexBFS}(G)$
2. $\bar{\sigma} \leftarrow \text{LexBFS}^-(\bar{G}, \sigma)$
3. **If** σ and $\bar{\sigma}$ both have the NS-property **then**
 - 3.1 Answer " G is a cograph"
 - 3.2 Build $MD(G)$
4. **Else** Output a P_4



Computing a LexBFS ordering σ



Computing $\bar{\sigma} = \text{LexBFS}^-(\bar{G}, \sigma)$



Lemma

$MD(G, x)$ is composed by the slices $S_i(x)$ of σ and $\bar{S}_j(x)$ of $\bar{\sigma}$.

Theorem

If G is a cograph, then $MD(G)$ can be retrieved from the slice structure of σ and $\bar{\sigma}$.

Some research problems

Generalization to arbitrary graphs ?

1. There are many similarities between two-LexBFS-sweep algorithm and the linear implementation of Ehrenfeucht et al.'s algorithm [DGM01]
2. LexBFS is useful for the transitive orientation problem. Could it lead to a simple linear time algorithm for this problem ?
3. Or another graph search, for example acting symmetrically on G and \bar{G} .
4. Certifying algorithms ?
5. Generalizations of modular decomposition (with some errors for real world graphs)

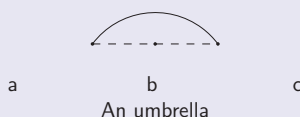
Definition

A cocomparability graph G is the complement of a comparability graph, i.e., \bar{G} admits a transitive orientation.

Definition :

For a total ordering τ of the set of vertices, an umbrella is a triple of vertices $a, b, c \in X$ such that : $a <_{\tau} b <_{\tau} c$ and $ac \in E$ and $ab, bc \notin E$. A co-comparability (co-comp for short) ordering is an umbrella-free total ordering of the vertices of G .

Forbidden triple



Characterization :

A graph is a cocomparability graph iff it admits a cocomp ordering.

Recognition status :

There exists a linear time algorithms which computes a cocomp ordering (McConnell and Spinrad).
 But the certifying step needs more $O(mn)$ or $O(MM)$.

Linear-time particular case for permutation graphs

1. Compute cocomp orderings σ and τ for G and \bar{G} .
2. Compute a representation of G as a permutation graph using σ and τ .
3. Certify the representation. This certifying step is also linear, using the geometric representation.

Since we focus on the ordering of the vertices as the result of a graph search, now we can compose graph searches in a natural way. Therefore we can denote by $M(G, x_0)$ the order of the vertices obtained by applying M on G starting from the vertex x_0 .

Definition of the + Rule

Let M be a graph search and σ an ordering of the vertices of G , $M^+(G, \sigma)$ be the ordering of the vertices obtained by applying M on G starting from the vertex $\sigma(1)$ and tie-breaking using σ in decreasing order.

Why this Rule ?

The + Rule forces to keep the ordering of the previous sweep in case of tie-break

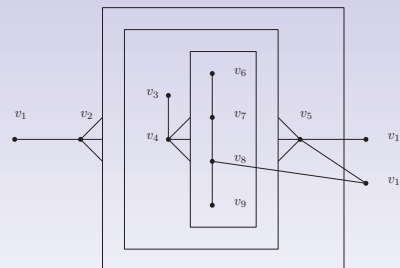
- ▶ Graph searches operate on total orderings :
 - Step 0 : $\sigma = M(G, x_0)$
 - Step 1 : $M(G, \sigma)$
 - Step 2 : $M^2(G, \sigma) = M(G, M(G, \sigma))$
 - ...
 - Step i : $M^i(G, \sigma) = M(G, M^{i-1}(G, \sigma))$
 - ...
- ▶ For which search M and graph G does there exist fixed points ?
- ▶ Unfortunately a formal study of this composition remains to be done !
- ▶ Also called multisweep algorithms.

1. Such an idea was already used for planarity testing in some algorithm (de Fraysseix and Rosentieh 1980) with 2 consecutive DFS.
2. Algorithms for strongly connected components by Kosaraju 1978, Sharir 1981
 In our framework,
 - 1) $DFS(G)$
 - 2) $DFS(G^-, post^d)$
3. To compute efficiently the diameter of a graph using successive BFS

REPEATED LBFS

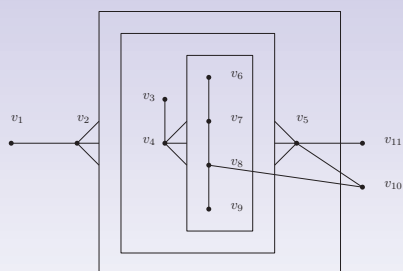
Require: $G = (V, E)$
Ensure: an ordering σ
 $\sigma \leftarrow LBFS(G)$
for $i = 2$ **to** $|V|$ **do**
 $\sigma \leftarrow LBFS^+(G, \sigma)$
end for

Algorithm 1: LBFS⁺ multi-sweep



Example

- ▶ $\tau := LBFS(G) : v_5, v_4, v_3, v_2, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_1$
- ▶ $\pi := LBFS^+(G, \tau) : v_1, v_2, v_9, v_8, v_7, v_6, v_4, v_5, v_3, v_{10}, v_{11}$
- ▶ $\theta := LBFS^+(G, \pi) : v_{11}, v_5, v_{10}, v_8, v_7, v_4, v_2, v_6, v_9, v_3, v_1$
- ▶ $\sigma := LBFS^+(G, \theta) : v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}$



Example

- ▶ $\sigma = LBFS^+(G, \theta) : v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}$
- ▶ $\theta = LBFS^+(G, \sigma) : v_{11}, v_5, v_{10}, v_8, v_7, v_4, v_2, v_6, v_9, v_3, v_1$
- ▶ ...

Known results about cocomparability and LBFS

Property (Korte, Mohring 1981)

If G is a cocomparability graph, then the last vertex of $LBFS(G)$ can be taken as a source in some transitive orientation of \overline{G} .
 Leads to good transitive orientation and interval recognition algorithms.

Property (Corneil, Olariu, Stewart 1999)

If G is an AT-free graph, 2 consecutive LBFS computes a dominating path.

Property (Corneil 1999)

For every cocomp graph G , it exists a cocomp ordering which is a LBFS ordering.

Property (Corneil, MH., Köhler 2011)

If G is a cocomparability graph and σ a cocomp ordering then $LBFS^+(G, \sigma)$ is also a cocomp ordering.

The particular case of interval graphs

Property (Corneil, Köhler 2010)

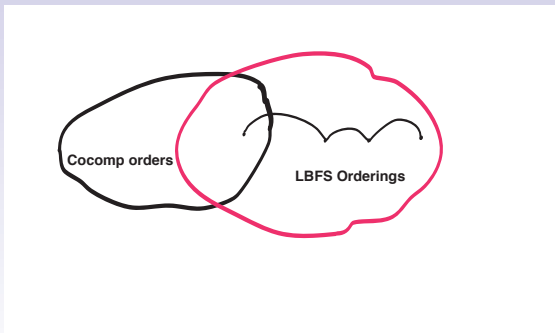
If $G = (V, E)$ is an interval graph, that REPEATED LBFS finds a interval ordering in less than $|V|$ iterations

Theorem (Corneil, Olariu and Stewart 2010)

For an interval graph, a series of **5+1 special** consecutive LBFS⁺ produces an interval ordering.

Theorem (Li, Wu 2012)

For an interval graph, a series of **4** searches produces an interval ordering.



Landscape for interval graphs

Our Results 2013

Theorem (Dusart, MH. 2013)

LBFS⁺ applied on a cocomparability graph produces in $O(n)$ steps a cocomp ordering.

Best possible

Using a Ma's family of interval graphs (2000), this result is best possible, i.e., a constant number of LBFS would not be enough for all graphs.

Consequences

Since for interval graphs a MNS ordering which is a cocomp ordering is also an interval ordering it gives the Corneil and Köhler's unpublished result.

Consequences

Dusart, MH 2013

It gives a very easy to program, $O(nm)$ cocomparability graph recognition.

Quasi fixed point

In all examples so far, Repeated LBFS⁺ on a cocomp always reaches in $O(n)$ a quasi-fixed point (i.e., a 2-loop), going back and forth on one cocomp ordering and its dual (reverse).

Is this always true?

Even for interval graphs it is still a conjecture.

Conjecture

Repeated LBFS⁺ on a cocomp always reaches in $O(n)$ a quasi-fixed point (i.e., a 2-loop), going back and forth on one cocomp ordering and its dual (reverse).

This can be the subject of a MPRI internship