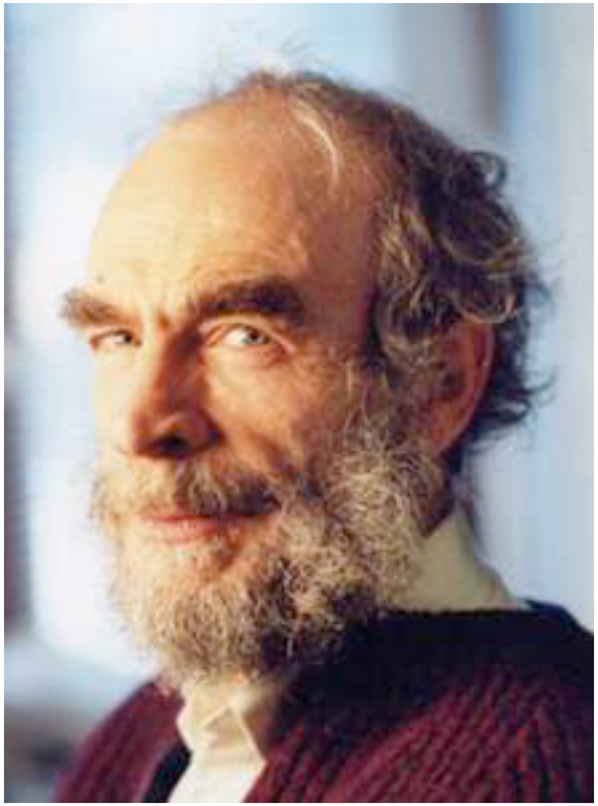
On computing the Gromov hyperbolicity

Nathann Cohen[1], David Coudert[2,3], Guillaume Ducoffe[4], Aurélien Lancin[2,3]

1. LRI, Univ. Paris Sud, France
2. Inria, France
3. Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France
4. ENS Cachan, France

Mikhail Leonidovich (Misha) Gromov



Professor at IHES

Abel Prize in 2009 *"for his revolutionary contributions to geometry"*

Hyperbolicity [Gromov 87]

Parameter: **δ**

- Originally defined for group theory
- Then extended to discrete metric spaces and graphs

Several definitions (at least 8)

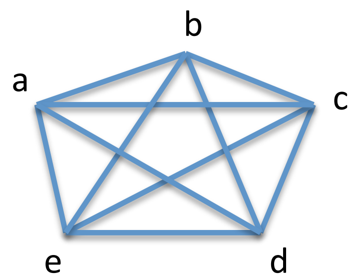
- Equivalent up to a multiplicative factor

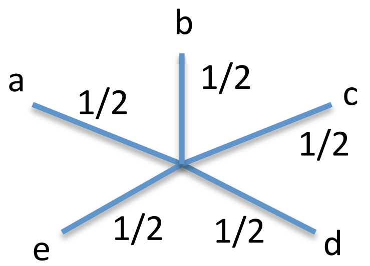
Provides information on

- How the **metric structure** of a graph looks like the metric structure of a tree
- Embedding into tree metrics with respect additive distortion

Embedding into tree metric

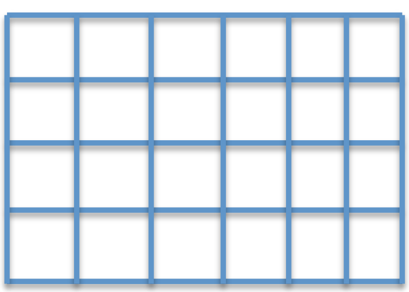
Cliques are "*similar*" to trees





Stretch
- Multiplicative = 1
- Additive = 0

Grids are "*far*" from trees



NxM grid with N ≤ M

Minimum stretch spanning tree:
Additive stretch = N-1

Metric space

Metric space (X,d)

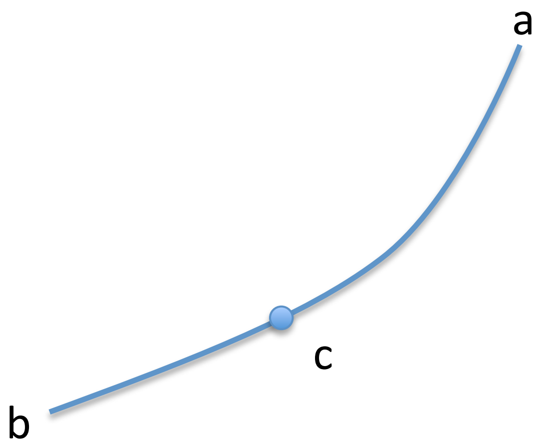
- X: set of points
- d: distance metric, $d: X \times X \longrightarrow R^+$

Properties of d:

1. Symmetry $d(x,y) = d(y,x)$
2. Separation $d(x,y) = 0 \Leftrightarrow x = y$
3. Triangular inequality $d(x, z) \leq d(x, y) + d(y, z)$

Geodesic

- Shortest *path* in X
- $d(a,b) = d(a,c) + d(b,c)$



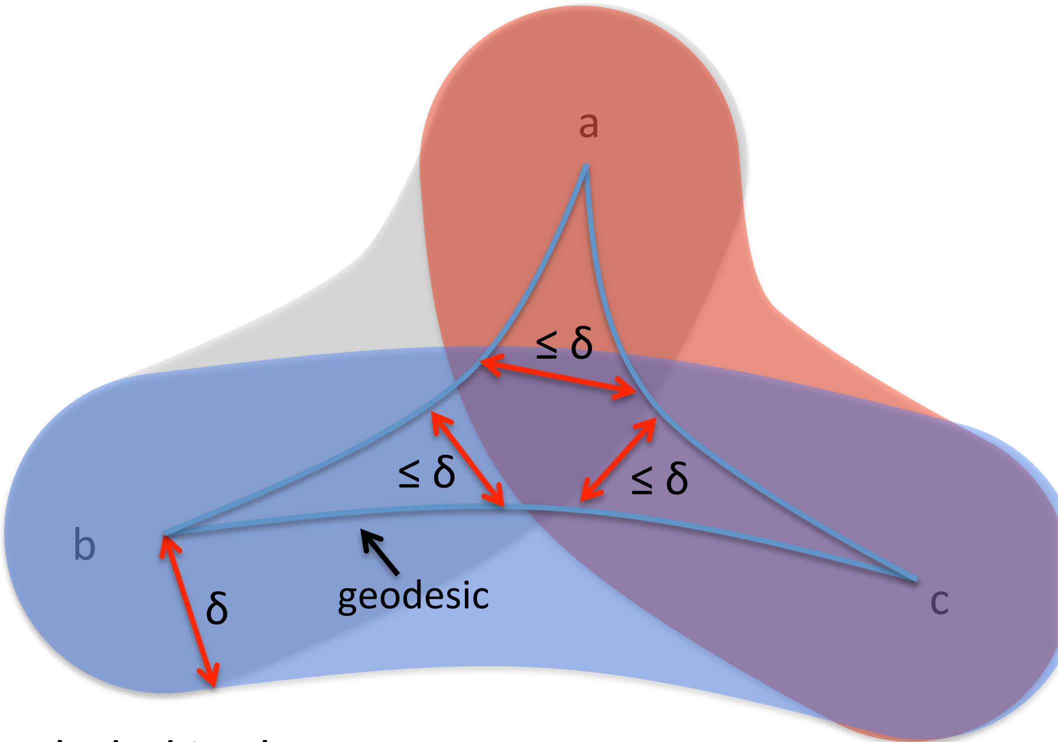
δ-hyperbolicity and δ-thin triangles

Metric space (X,d)

- X: set of points
- d: distance metric

Triangle Δ(a,b,c)

- Sides are geodesics



δ-thin triangle

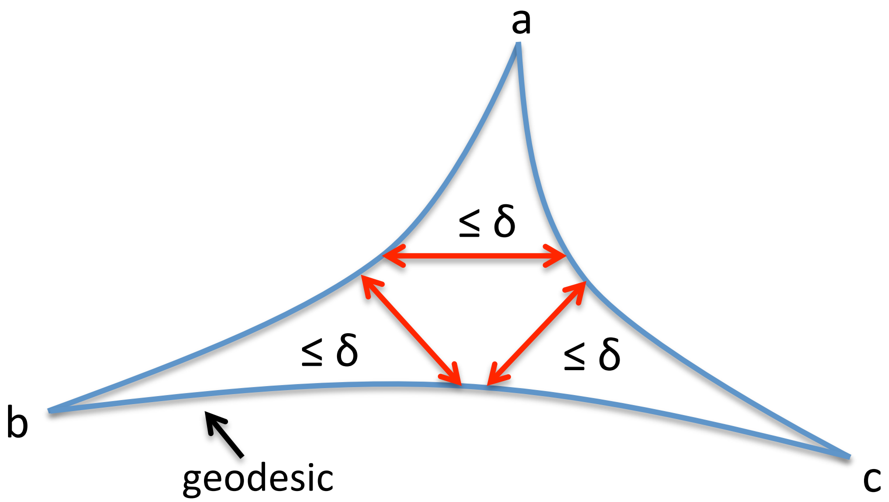
- Δ(a,b,c) is δ-thin if each side is included in the δ-neighborhood of the 2 others

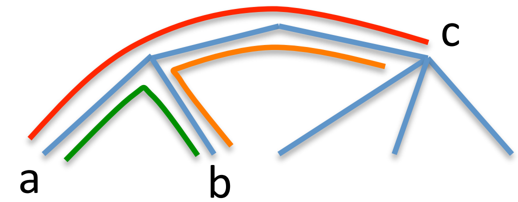
Definition:

- (X,d) is **δ-hyperbolic** if all triangles are δ-thin.

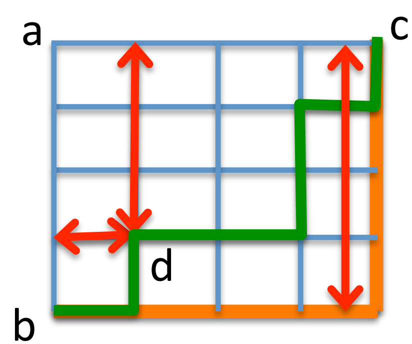
Examples

Metric space





Tree: $\delta = 0$



n x n grid

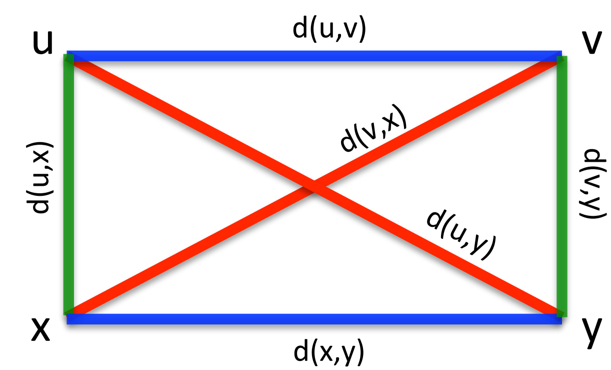
$\delta = n-1$

d is at distance ≤ δ from one of the shortest path from a to b or from a to c

Hyperbolicity: 4-points condition

$G = (V_G, E_G)$ connected graph

$d(u,v)$ shortest path distance in G



Definition:

(V_G, d) is δ-hyperbolic if for all u,v,x,y in V_G

$$d(u,v)+d(x,y) - \max\{ d(u,x) + d(v,y), d(u,y) + d(v,v)\} \leq 2\delta$$

or,

G is δ-hyperbolic if the two largest of the three sums (matchings)

$S_1 = d(u,v) + d(x,y)$ $S_2 = d(u,x) + d(v,y)$ $S_3 = d(u,y) + d(v,x)$

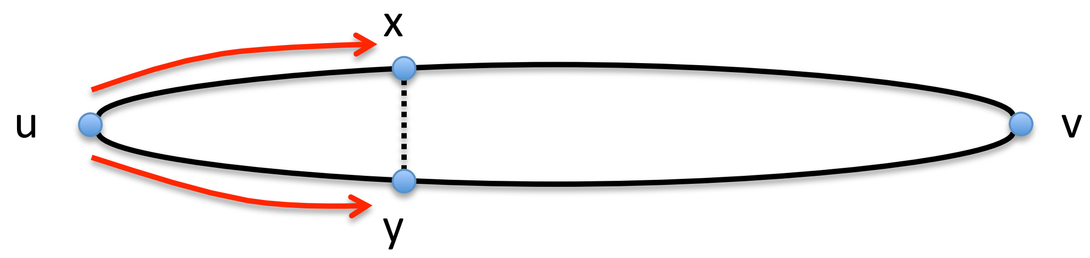
differ by at most 2δ.

If $S_1 \geq S_2 \geq S_3$
then $S_1 - S_2 \leq 2\delta$

δ(G) is the *smallest* δ such that G is δ-hyperbolic

Interpretation

2 geodesics
from u to v



Proposition: If x and y travel at the same speed, then d(x,y) ≤ 2δ

Proof:

- S_1 = d(u,v) + d(x,y) S_2 = d(u,x) + d(v,y) S_3 = d(u,y) + d(v,x)
- x and y have same speed => $S_2 = S_3$ = d(u,v)
  - d(u,x) = d(u,y) and d(v,x) = d(v,y)
- By 4-points condition: S_1-S_2 = d(x,y) ≤ 2δ

Consequence for routing:

- Distance between shortest paths bounded by 2δ

Using the Gromov product

Gromov product

- Metric space (X,d)
- Base point $u \in X$
- Gromov product of x by y with respect u: $(x|y)_u = (d(x,u) + d(y,u) – d(x,y)) / 2$

- $(x|y)_u = 0$ if u on the geodesic between x and y

Proposition: (X,d) is δ-hyperbolic if, and only if, for all u,v,x,y in X

$$(x|y)_u \geq \min \{ (x|v)_u, (y|v)_u \} - \delta$$

Equivalence with 4-points condition

Gromov product: $(x|y)_u = (d(x,u) + d(y,u) - d(x,y)) / 2$

Proposition: (X,d) is δ-hyperbolic if, and only if, for all u,v,x,y in X

$$(x|y)_u \geq \min \{ (x|v)_u, (y|v)_u \} - \delta$$

$$\begin{aligned}
(x|y)_u &\geq \min\{(x|v)_u, (y|v)_u\} - \delta \\
d(x,u) + d(y,u) - d(x,y) &\geq \min \{d(x,u) + d(v,u) - d(x,v), d(y,u) + d(v,u) - d(v,y)\} - 2\delta \\
-d(x,u) - d(y,u) + d(x,y) &\leq -d(v,u) + \max\{-d(x,u) + d(x,v), -d(y,u) + d(v,y)\} + 2\delta \\
d(x,y) + d(v,u) &\leq d(x,u) + d(y,u) + \max\{-d(x,u) + d(x,v), -d(y,u) + d(v,y)\} + 2\delta
\end{aligned}$$

$$d(x,y) + d(v,u) - \max\{d(x,v) + d(y,u), d(v,y) + d(x,u)\} \leq 2\delta.$$

Proposition: (X,d) is δ-hyperbolic if, and only if, for all u,v,x,y in X

$$d(u,v)+d(x,y) - \max\{ d(u,x) + d(v,y), d(u,y) + d(v,x)\} \leq 2\delta$$

Example

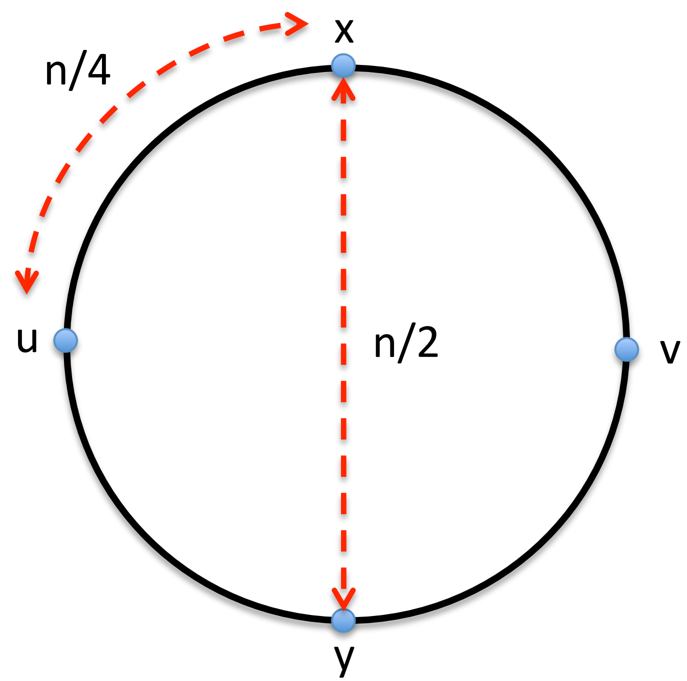
Cycle or order n, C_n

$S_1 = d(u,v) + d(x,y) = n$

$S_2 = d(u,x) + d(v,y) = n/2$

$S_3 = d(u,y) + d(v,x) = n/2$

$2\delta \geq S_1 - S_2 = n/2$



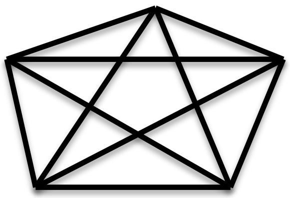
$\delta(C_n) = n/4 - \varepsilon$, $\varepsilon = 1/2$ when $n \equiv 1 \pmod 4$, and 0 otherwise

n	3	4	5	6	7	8	9	10	11
$\delta(C_n)$	0	1	½	1	1	2	3/2	2	2

Examples

0-hyperbolic graphs

- Trees
- Cliques
- Block-graphs



$S_1 = S_2 = S_3 = 2$

1-hyperbolic graphs

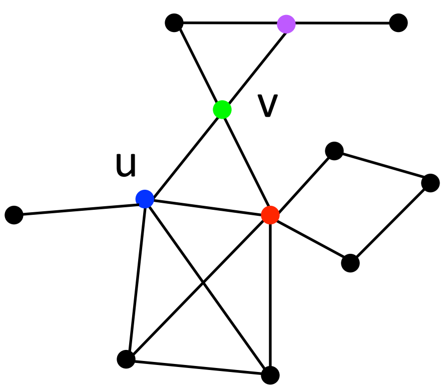
- Characterization by **[Chepoi et al. 2003]**
- Chordal graphs (every cycle of size 4 or more has a chord)

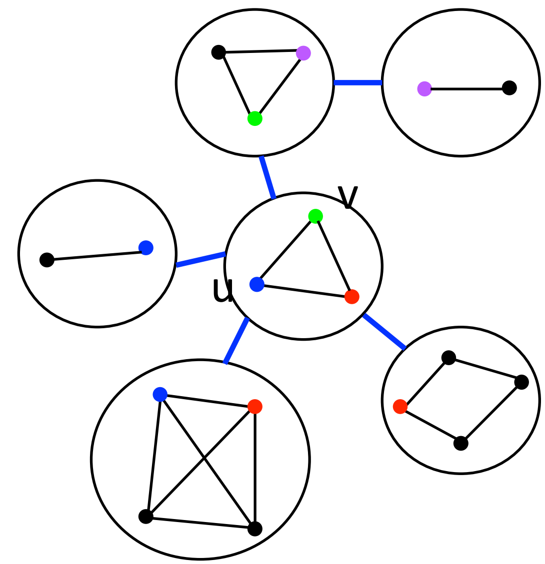
- k-chordal graphs with $k \geq 4$ are ($\leq k/4$)-hyperbolic
- NxM grid with $N \leq M$ are (N-1)-hyperbolic

Relation with graph structural properties

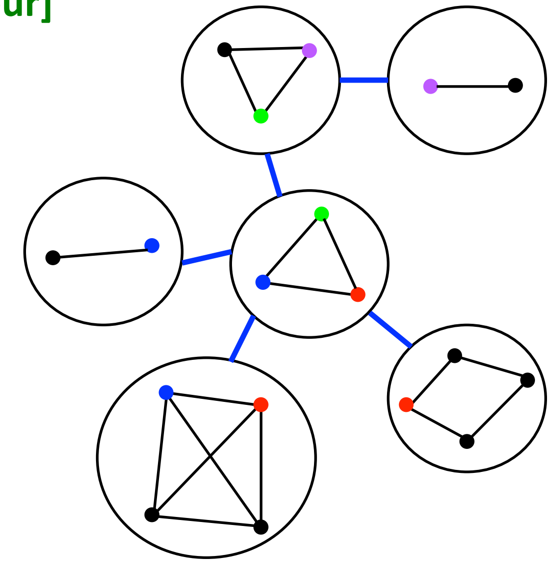
Tree-decomposition (S,T) of G=(V,E)

- $S = \{S_1, S_2, \ldots, S_k\}$ is a collection of subsets of V — S_i is a *bag* (and a separator)
- T is a tree whose vertices are elements of S such that:
  1. $\cup_i S_i = V$ — each node is in a bag
  2. $\forall e=(u,v)$ in E, $\exists S_i$ s.t. u,v in S_i — each edge is in a bag
  3. $\forall$ u in V, the elements of S containing u form a subtree of T





Relation with graph structural properties



Treewidth, tw(G) [Courcelle et al.] [Robertson, Seymour]

- Size of the largest bag
- Design of algorithms
- No relation: $tw(K_n) = n-1$, $\delta(K_n) = 0$
  - See [Soto, de Mongolfier, Viennot 2010]

Treelength, tl(G) [Dourisboure and Gavoille 2003]

- Largest diameter of a bag
  - Distances in G
- Design of routing schemes

Theorem [Chepoi, Dragan, Estellon, Habib, Vaxes 2008]

If $tl(G) = k$, then $\delta(G) <= k$

But computing treelength is NP-hard [Lokshtanov 2007]

Other relations

Diameter:

- **$\delta(G) \leq diam(G) / 2$**
- Tight for square grids, cycles

Girth and circumference:

- $g(G)$: length of the shortest cycle
- $c(G)$: length of the longest cycle
- $g(G)/4 \leq \delta(G) \leq c(G)/4$

Dominating set:

- $\gamma(G)$: size of the smallest dominating set
- $\delta(G) \leq 3\gamma(G)/2$

Why is it useful? Where is it used?

Hyperbolicity

- Provides bounds on the distortion of embedding into a tree metric

Routing

- Bound on the additive stretch of *Greedy Distance Vector* routing protocol
  - Embedded into hyperbolic space to assign coordinates to nodes
- O(δlog n)-additive routing labeling scheme with O($\delta\log_2 n$) bit labels

[Chepoi, Dragan, Estellon, Habib, Vaxes 2008]

- Compact routing schemes with small routing tables and low distortion

[Gavoille et al.]

Phylogenetic (bio-informatics) **[Chakerian and Holmes, 2010]**

- Another metric to compare *phylogenetic networks*
- Model evolutionary relationships among species, genes, etc.

Why am I studying it?

Context:

- Collaboration on *compact* routing schemes
- EU project EULER (2010-2014) with Alcatel-Lucent Bell (Belgium)

(Long) discussion with keywords “hyperbolicity”, “hyperbolic graphs”, etc.

Q: Can we easily compute this property?

A: Yes, it’s polynomial, in $O(n^4)$

Q: What???? but the AS topology has more than 40.000 nodes!?!

It will take months

A: Well…

Complexity

Basic method:

- Maximum over all 4-tuples u,v,x,y in G — $O(n^4)$
- 2-approximation: fix x — $O(n^3)$

Best known algorithm:

- Use (max,min)-product of matrices — $O(n^{3.69})$ **[Fournier et al, 12]**
- 2-approximation — $O(n^{2.69})$ **[Fournier et al, 12]**

Impossibility: **[Vigneron 2014]**

- 2-approx cannot be computed in less than $O(n^{2.05})$ unless there exists a faster (max,min)-product

Chordal graphs, $\delta \leq 1$ — $O(n+m)$

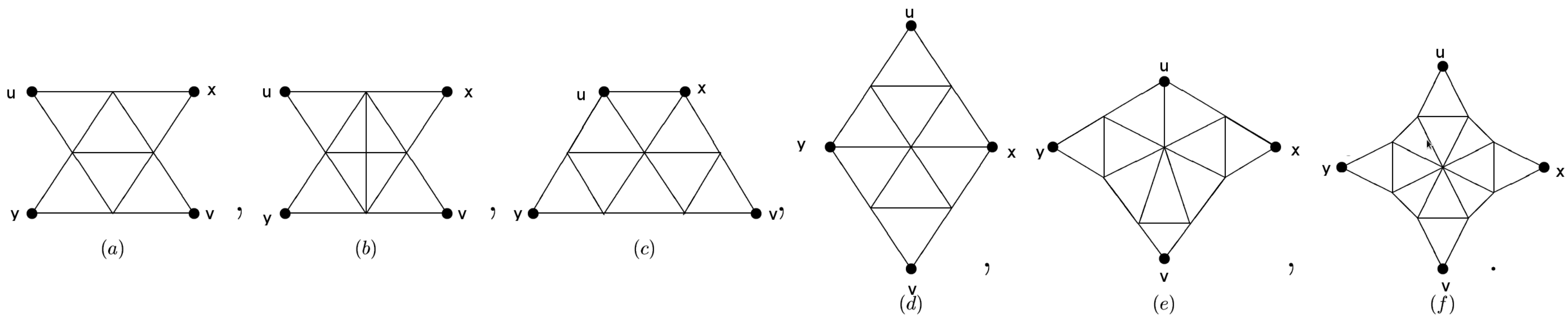
Problem: CAIDA AS map Nov 2013: n = 45 427 ~ 1.7 10^{17} months?

Graphs with low hyperbolicity

Block graphs (trees, cliques)	$\delta = 0$	O(n+m)
Chordal graphs	$\delta \leq 1$	O(n+m)

Characterization of ½ - hyperbolic graphs **[Bandelt and Chepoi, 2003]**

- No induced cycles of length different from 3 or 5
- 6 forbidden *isometric* sub-graphs



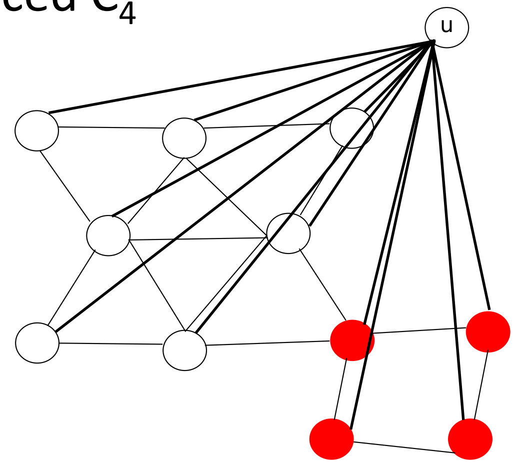
Graphs with low hyperbolicity

Use 2-approximation (fix one vertex), $O(n^{2,69})$

- Answer > 1 => $\delta \geq 1$
- Answer ≤ 1 => δ in { ½, 1 }

Theorem: [Coudert, Ducoffe 2013]

- Equivalence with the problem of searching for an induced C_4



Time complexity: $O(n^{3,37})$

Let little hope for improving worst case time complexity

Basic algorithm using massive parallelism

Table 2. Computation time (in seconds) of tasking versus workshare

Network	n		Number of CPUs						This work
			32	64	128	256	512	1015	1
Polblogs	1222	tasking	70	42	42	69	137	269	
		workshare	71	47	46	70	132	292	
CA-GrQc	4158	tasking	8989	4933	2723	2053	1916	3691	1.4 sec.
		workshare	10433	5749	5012	3260	2688	3136	
as20000102	6474	tasking	50002	37417	15308	9851	9039	11419	0.8 sec.
		workshare	47197	29309	23851	17359	12231	11491	
Gnutella09	8104	tasking	-	-	-	-	26888	13295	0.04 sec.
		workshare	-	-	-	-	28564	21456	

Source: Evaluating OpenMP Tasking at Scale for the Computation of Graph Hyperbolicity
Adcock, Sullivan, Hernandez & Mahomey -- IWOMP 2013

Goal: Compare OpenMP settings

Ideas

Pre-processing:

- Reduce the size of the instance without alteration of the hyperbolicity
- Decomposition:
  - Bi-connected components
  - Split-decomposition
  - Clique-separators

Smart ordering:

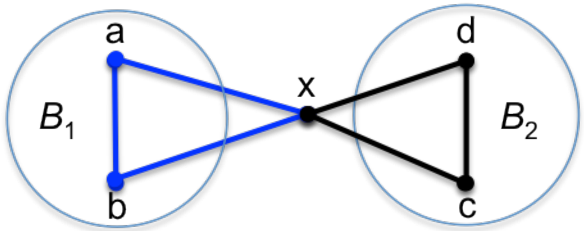
- Visit most promising 4-tuples first (with high expected hyperbolicity)

Prune search space:

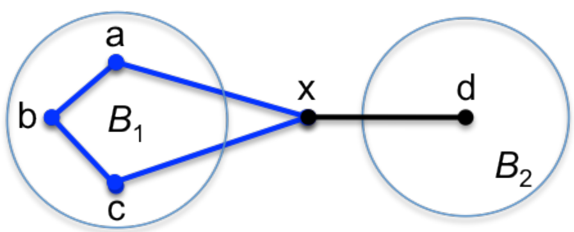
- Use bounds to stop exploration
- Prevent visiting *useless* 4-tuples

Bi-connected components

[Tarjan 73] Partition into bi-connected components O(n+m)



δ(a,b,c,d) = 0

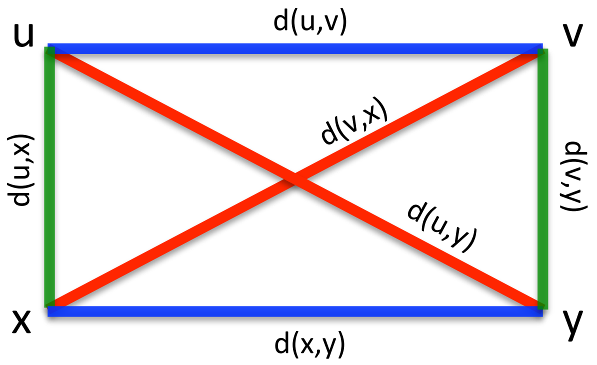


δ(a,b,c,d) = δ(a,b,c,x)

Hyperbolicity of G = maximum over all biconnected components.

CAIDA AS map Nov 2013: n = 45 427 Largest BCC = 29 432 ~ $3.1 \; 10^{16}$

Observation



$S_1 = d(u,v) + d(x,y)$
$S_2 = d(u,x) + d(v,y)$
$S_3 = d(u,y) + d(v,x)$

If $S_1 \geq S_2 \geq S_3$
then $S_1 - S_2 \leq 2\delta$

Assume $S_1 \geq \max \{ S_2 , S_3 \}$

$$\begin{aligned} S_2 + S_3 &= d(u,x) + d(v,y) + d(u,y) + d(v,x) \\ &= (d(u,x) + d(v,x)) + (d(u,y) + d(v,y)) \\ &\geq d(u,v) + d(u,v) \\ &\geq 2\, d(u,v) \end{aligned}$$

$$\begin{aligned} 2\, \delta(u,v,x,y) &\leq S_1 - \max \{ S_2 , S_3 \} \\ &\leq S_1 - (S_2 + S_3) / 2 \leq S_1 - d(u,v) \\ &\leq d(u,v) \end{aligned}$$

and

$$2\, \delta(u,v,x,y) \leq d(x,y)$$

Idea: explore large distances first *and* use bounds to cut exploration

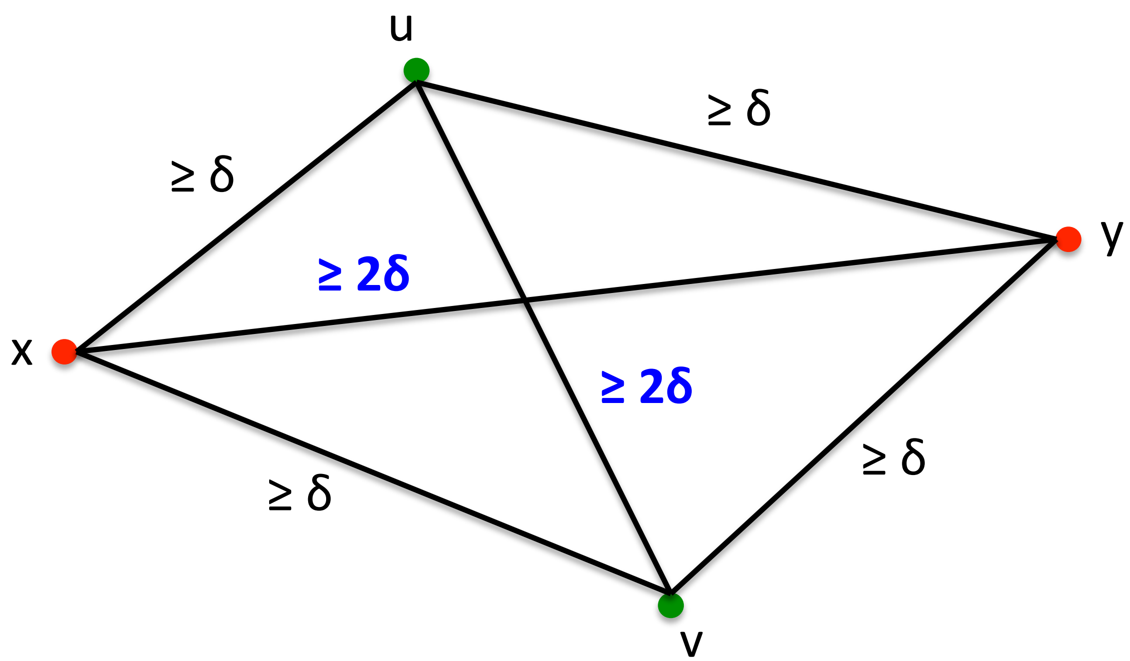
Shape of a certificate

Previous lemma:

- The pairs in largest sum are distance ≥ 2δ

Lemma:

- δ(u,v,x,y) ≤ min { d(u,v), d(u,x), d(u,y), d(v,x), d(v,y), d(x,y) }



Better algorithm

pair: list of the pairs (x,y) sorted by decreasing distances $d(\text{pair}[i]) \geq d(\text{pair}[i+1])$

```
δ := 0
for 1 ≤ i < (n choose 2) do
    (u,v) := pair[ i ]
    for 0 ≤ j < i do
        (x,y) := pair[ j ]
        δ := max { δ, δ(u,v,x,y) }
        if d(u,v) ≤ 2δ then
            return δ
```

Complexity: $O(n^4)$

But: n x n grid => only 1 test !

Running time depends on:

- Distances distribution
- Value of δ

$d(x,y) \geq d(u,v)$

$d(u,v)$ is the upper bound on $2\delta(u,v,x,y)$

Basic algorithm using massive parallelism

Table 2. Computation time (in seconds) of tasking versus workshare

Network	n		Number of CPUs 32	64	128	256	512	1015	This work
									1
Polblogs	1222	tasking	70	42	42	69	137	269	
		workshare	71	47	46	70	132	292	
CA-GrQc	4158	tasking	8989	4933	2723	2053	1916	3691	19 sec.
		workshare	10433	5749	5012	3260	2688	3136	
as20000102	6474	tasking	50002	37417	15308	9851	9039	11419	1.2 sec.
		workshare	47197	29309	23851	17359	12231	11491	
Gnutella09	8104	tasking	-	-	-	-	26888	13295	0.04 sec.
		workshare	-	-	-	-	28564	21456	

Source: Evaluating OpenMP Tasking at Scale for the Computation of Graph Hyperbolicity
Adcock, Sullivan, Hernandez & Mahomey -- IWOMP 2013

Goal: Compare OpenMP settings

	CA-GrQc	as20000102	Gnutella09
N	4 158	6 474	8 104
BCC	2 651	4 009	5 606
δ	3,5	2,5	3
1	10 480	10 101	23 510
2	50 508	1 231 628	292 458
3	209 027	3 721 215	2 101 638
4	625 349	2 569 590	7 222 588
5	1 111 919	475 366	5 420 722
6	983 190	25 648	645 387
7	418 341	485	4 490
8	91 502	3	22
9	11 332		
10	886		
11	41		
d ≥ 2*δ	522 102	501 502	649 899
d > 2*δ	103 761	26 136	4 512
Visited 4-tuples	5,38E+09	3,42E+08	1,02E+07
	19 sec.	**1.2 sec.**	**0.04 sec.**

This is not enough !

CAIDA AS map Nov 2013:

- n = 45 427
- Largest BCC = 29 432
- Total number of 4-tuples ~ $3.1\ 10^{16}$
- $\delta(G) = 2.5$
- Visited 4-tuples ~ $2.0\ 10^{14}$
- 2 weeks of computation

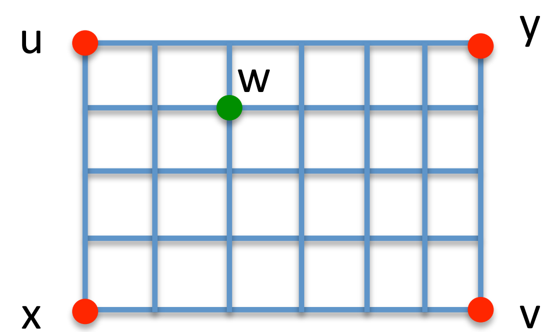
Far-apart pairs

[Soto-Gomez 2011]

Definition:

- The pair (u,v) is *far-apart* if for every w in V \ {u, v},
  $$d(w, u) + d(u, v) > d(w, v)$$
  and $d(w, v) + d(u, v) > d(w, u)$

- w on the shortest path between u and v
  ⇒ The pair (u,w) is *not* far-apart
  ⇒ The pair (v,w) is *not* far-apart



- The geodesic (u,v) is not included in another geodesic

Only 2 far-apart pairs (u,v) and (x,y)
$\delta(u,v,x,y) = N-1 = \delta(G)$

Lemma: [Soto-Gomez 2011]

- There exist two far-apart pairs (u, v) and (x, y) satisfying $\delta(u, v, x, y) = \delta(G)$

Complexity: O(nm) using BFS

With far-apart pairs

FA: list of the far-apart pairs (x,y) sorted by decreasing distances

```
δ := 0
for 1 ≤ i < (n choose 2) do
    (u,v) := FA[ i ]
    for 0 ≤ j < i do
        (x,y) := FA[ j ]
        δ := max { δ, δ(u,v,x,y) }
        if d(u,v) ≤ 2δ then
            return δ
```

Complexity:

- $O(n^4)$ in general
- O(1) for grids

Running time depends on:

- Distances distribution
- Value of δ

$d(x,y) \geq d(u,v)$

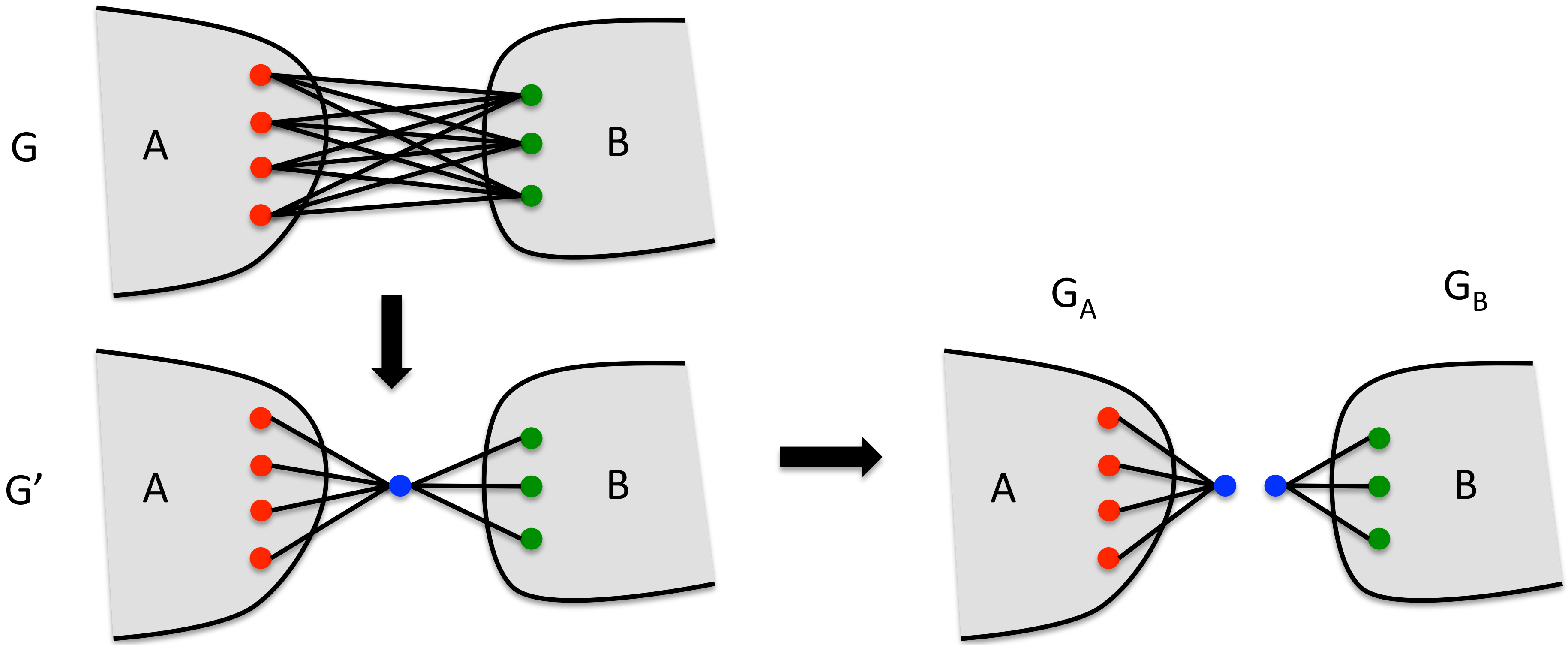
d(u,v) is the upper bound on $2\delta(u,v,x,y)$

	CA-GrQc	as20000102	Gnutella09
N	4 158	6 474	8 104
BCC	2 651	4 009	5 606
δ	3,5	2,5	3

	P	FA	P	FA	P	FA
1	10 480	1 129	10 101	--	23 510	--
2	50 508	5 538	1 231 628	519 224	292 458	3
3	209 027	24 879	3 721 215	2 235 874	2 101 638	1 999
4	625 349	109 394	2 569 590	1 943 404	7 222 588	451 789
5	1 111 919	354 964	475 366	413 824	5 420 722	3 376 888
6	983 190	512 472	25 648	24 017	645 387	629 722
7	418 341	287 452	485	472	4 490	4 404
8	91 502	73 153	3	3	22	22
9	11 332	9 800				
10	886	811				
11	41	41				
d ≥ 2*δ	522 102	371 257	501 502	438 316	649 899	634 148
d > 2*δ	103 761	83 805	26 136	24 492	4 512	4 426
Visited 4-tuples	5.38E+09	3.5E+09	3.42E+08	3.0E+08	1.02E+07	9.8E+06
	19 sec.	**12 sec.**	**1.2 sec.**	**1 sec.**	**0.044 sec.**	**0.042 sec.**

Split-decomposition

[Soto-Gomez 2011]



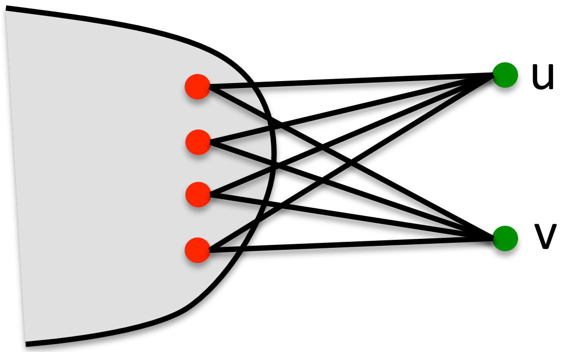
Theorem:

- $\max \{ \delta(G_A), \delta(G_B) \} \leq \delta(G) \leq \max \{ 1, \delta(G_A), \delta(G_B) \}$

Split-decomposition (cont)

Remarks:

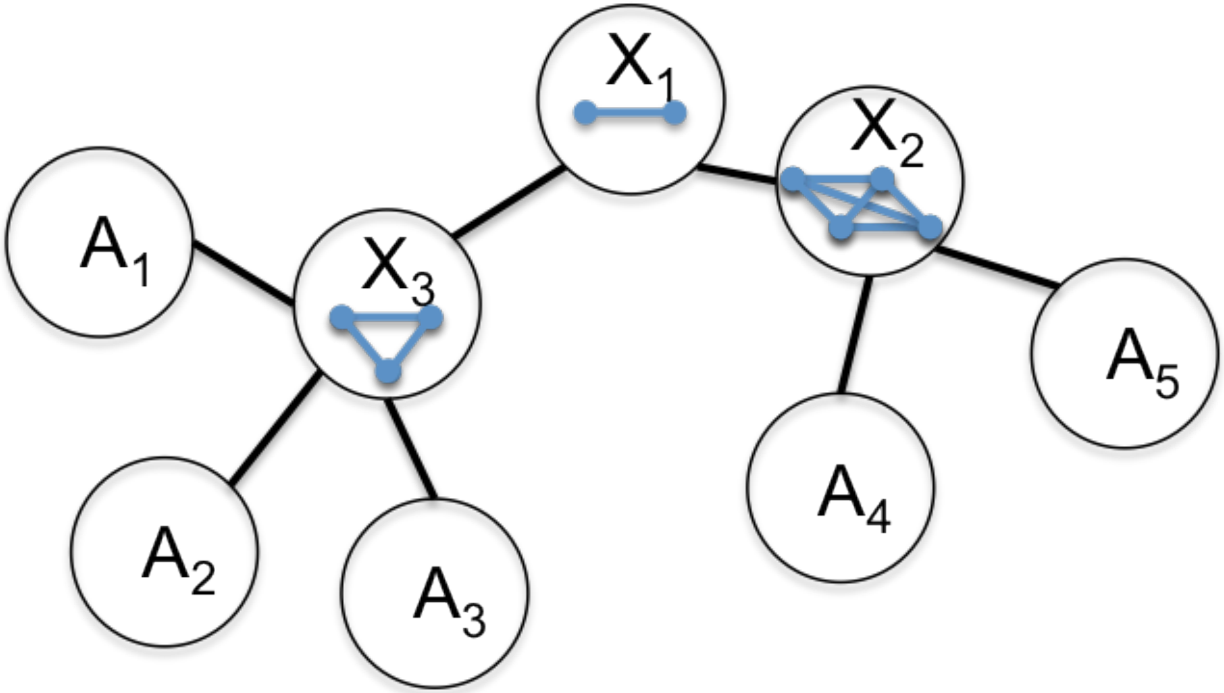
- If u and v have the same neighbors, remove one of them
  - u and v are *twins*
- Generalizes modular-decomposition



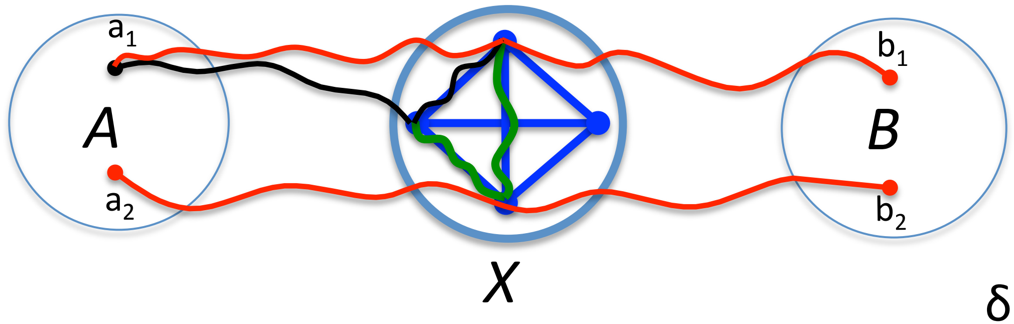
Complexity: O(n+m) **[Charbit, de Montgolfier, Raffinot 2012]**

Clique-separators

[Tarjan 85] Partition by clique-separators O(nm)



(Clique)-separators



$\delta(a_1,a_2,b_1,b_2) \leq 1$

Let

- $S_1 = d(a_1,a_2) + d(b_1,b_2)$
- $S_2 = d(a_1,b_1) + d(a_2,b_2)$
- $S_3 = d(a_1,b_2) + d(a_2,b_1)$

Assume $S_1 \geq \max \{ S_2 , S_3 \}$

$d(a_i,b_i) \geq d(a_i,X) + d(X,b_i)$ for i= 1,2

$d(a_i,a_j) \leq d(a_i,X) + d(a_j,X) + D_X$ for i,j= 1,2

$$
\begin{aligned}
S_2 &= d(a_1,b_1) + d(a_2,b_2) \\
&\geq d(a_1,X) + d(X,b_1) + d(a_2,X) + d(X,b_2) \\
&\geq [d(a_1,X) + d(a_2,X) + D_X] + [d(b_1,X) + d(b_2,X) + D_X] - 2D_X \\
&\geq \mathbf{S_1 - 2D_X}
\end{aligned}
$$

$S_3 \geq \mathbf{S_2 - 2D_X}$

If $S_2 \geq S_1$ then:

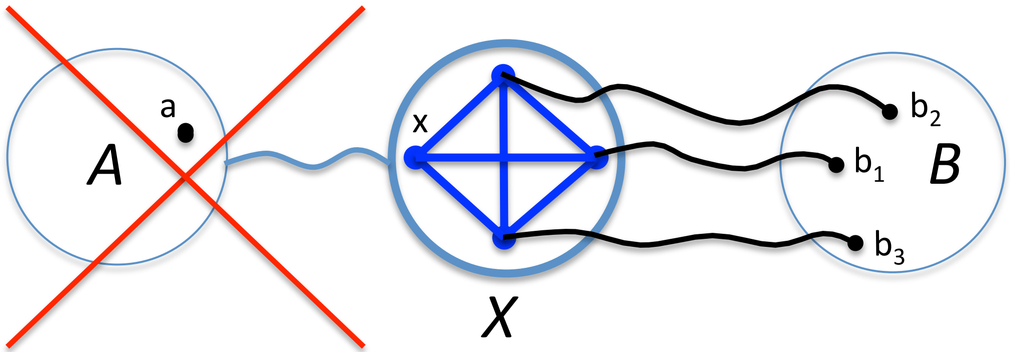
$S_1 \geq \mathbf{S_2 - 2D_X}$

Finally:

$\delta(a_1,a_2,b_1,b_2) \leq D_X$

COATI

Clique-separators



Lemma: $\delta(a,b_1,b_2,b_3) \leq \delta(x,b_1,b_2,b_3) + 1/2$

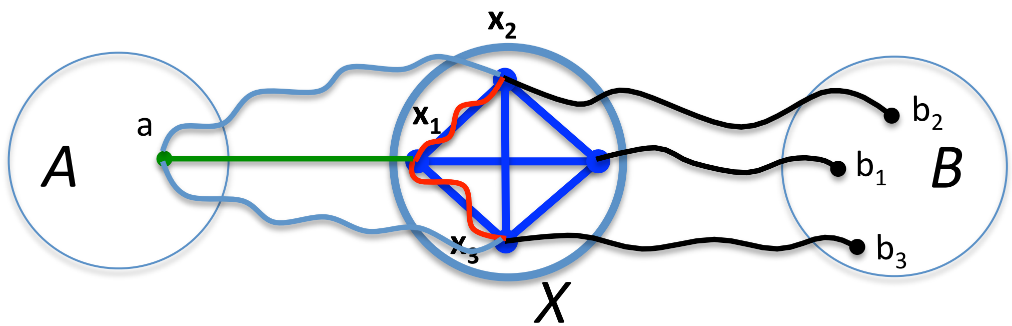
Proof: use $d(a, x) + d(x, b_i) \leq d(a, b_i) + 1$

⋮

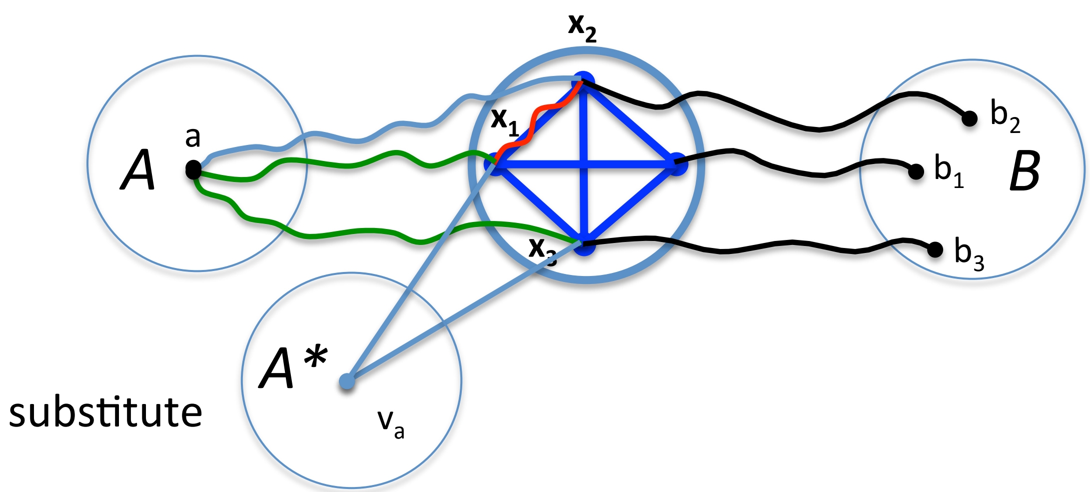
Theorem: Given the atoms A_i of the decomposition by clique-separators of G, we have:

$$\max_i \delta(A_i) \leq \delta(G) \leq \max_i \delta(A_i)+1$$

Clique-separators

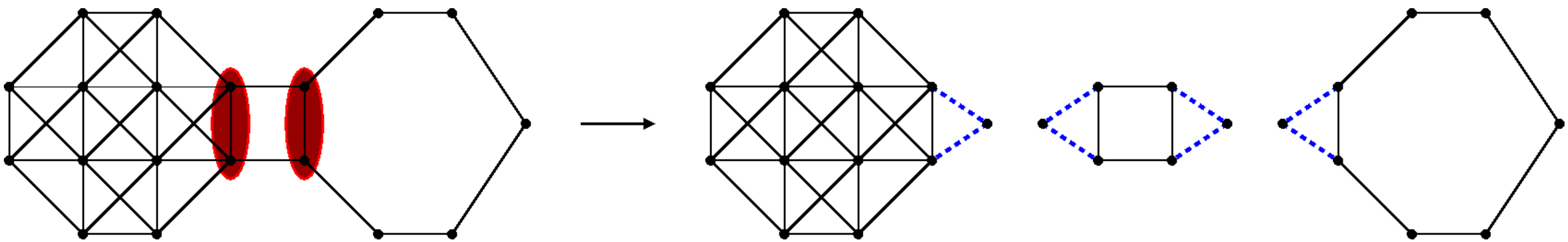


Lemma: If $d(a,x_1) = m$ and $d(a,x_2) = d(a,x_3) = m + 1$ then $\delta(\mathbf{a},b_1,b_2,b_3) = \delta(\mathbf{x_1},b_1,b_2,b_3)$



Lemma: If $d(a,x_1) = d(a,x_3) = m$ and $d(a,x_2) = m + 1$ then $\delta(\mathbf{a},b_1,b_2,b_3) = \delta(\mathbf{v_a},b_1,b_2,b_3)$

Clique-separators



Theorems:

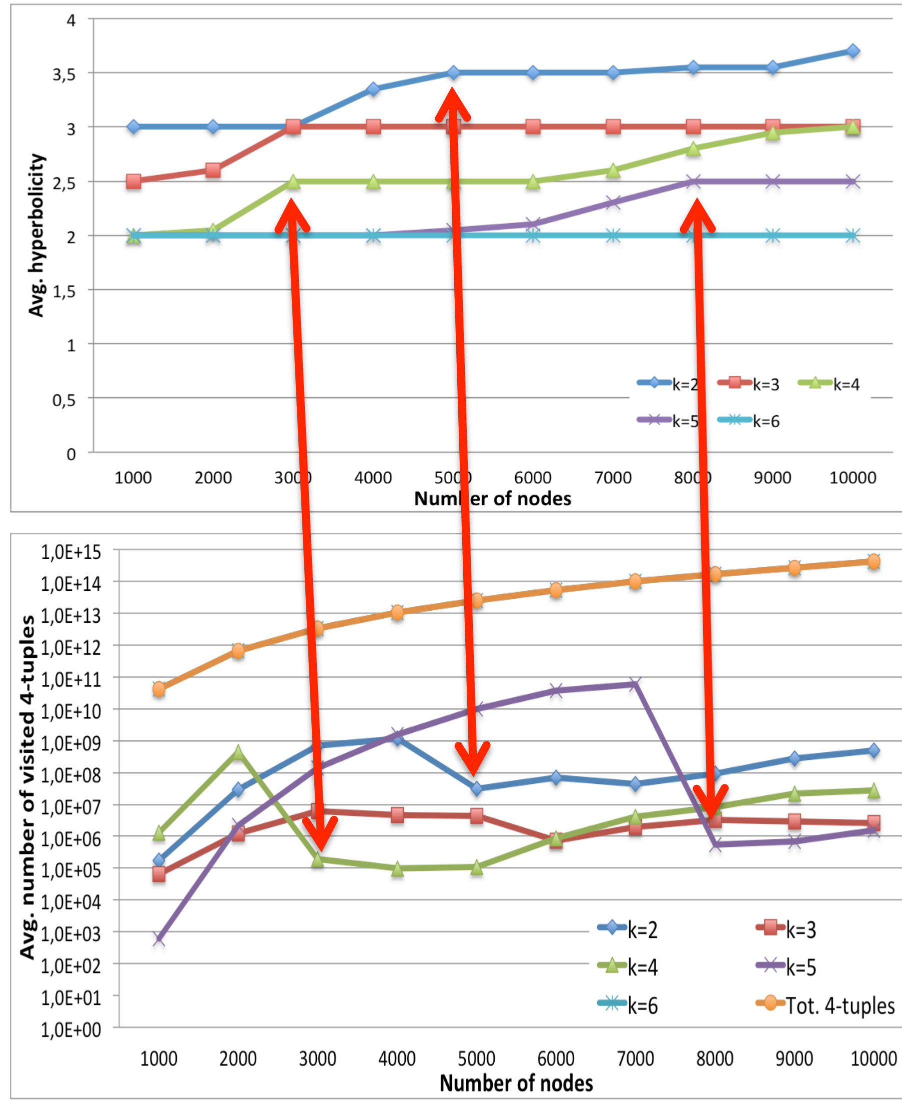
- $\max_i \delta(A_i) \leq \delta(G) \leq \max_i \delta(A_i)+1$ — +1 approximation
- $\delta(G) = \max \{ 1, \max_i \delta(A_i \cup (V \backslash A_i)^*) \}$ — using substitutes
- Construction of substitute graphs $O(nm)$ — during decomposition
- Outerplanar graphs: $O(n+m)$

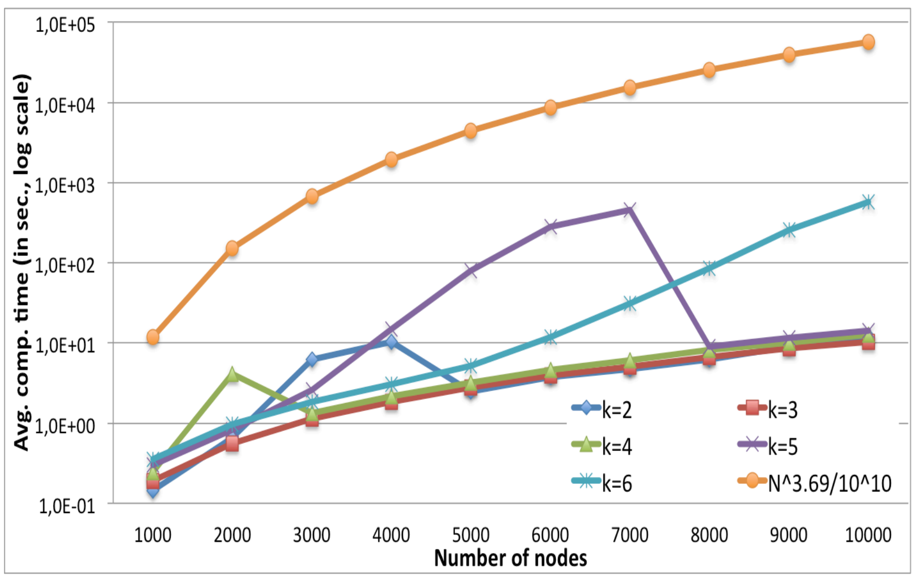
Final computation time

	CA-GrQc	as20000102	Gnutella09
N	4 158	6 474	8 104
δ	3,5	2,5	3
BCC	2 651	4 009	5 606
Visited 4-tuples	5.38 E+09	3.42 E+08	1.02E+07
Time in sec.	**19**	**1.2**	**0.044**
BCC + FA			
Visited 4-tuples	3.5 E+09	3.0 E+08	9.8 E+06
Time in sec.	**12**	**1**	**0.042**
Largest substitute	2 343	3 006	5 606
Visited 4-tuples	4.1 E+08	2.3 E+08	9.8 E+06
Time in sec.	**1.4**	**0.8**	**0.042**

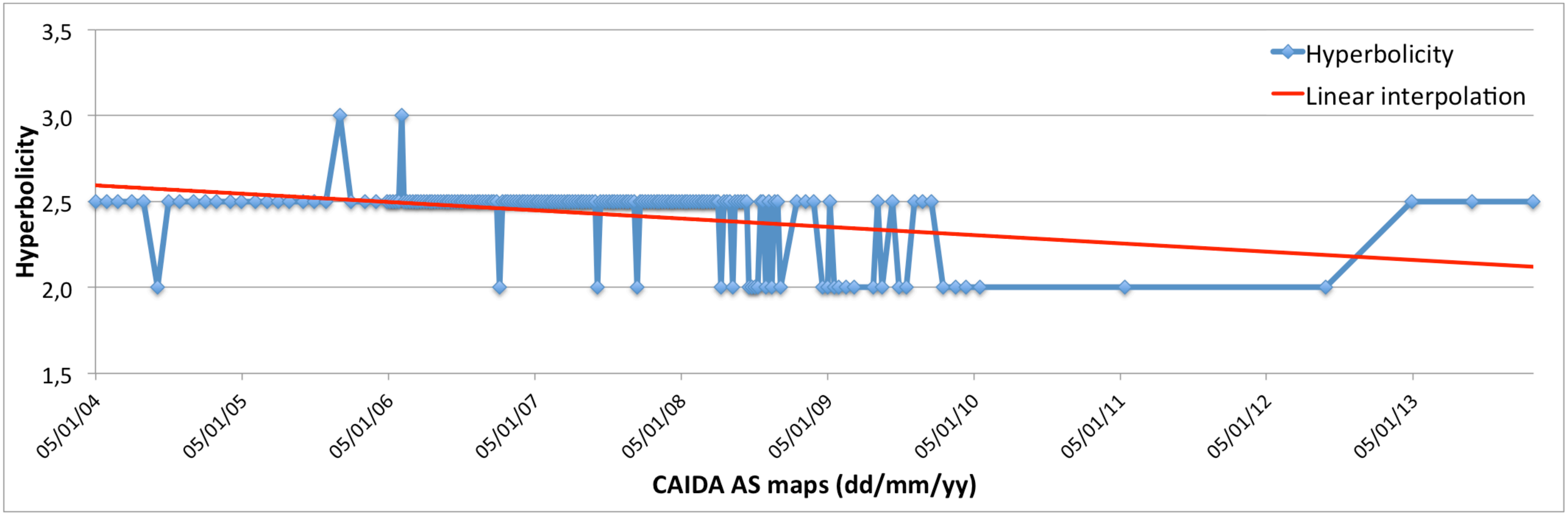
Barabasi-Albert

Add new node of degree k with preferential attachment





CAIDA AS maps



190 maps from Jan. 2004 till Nov. 2013

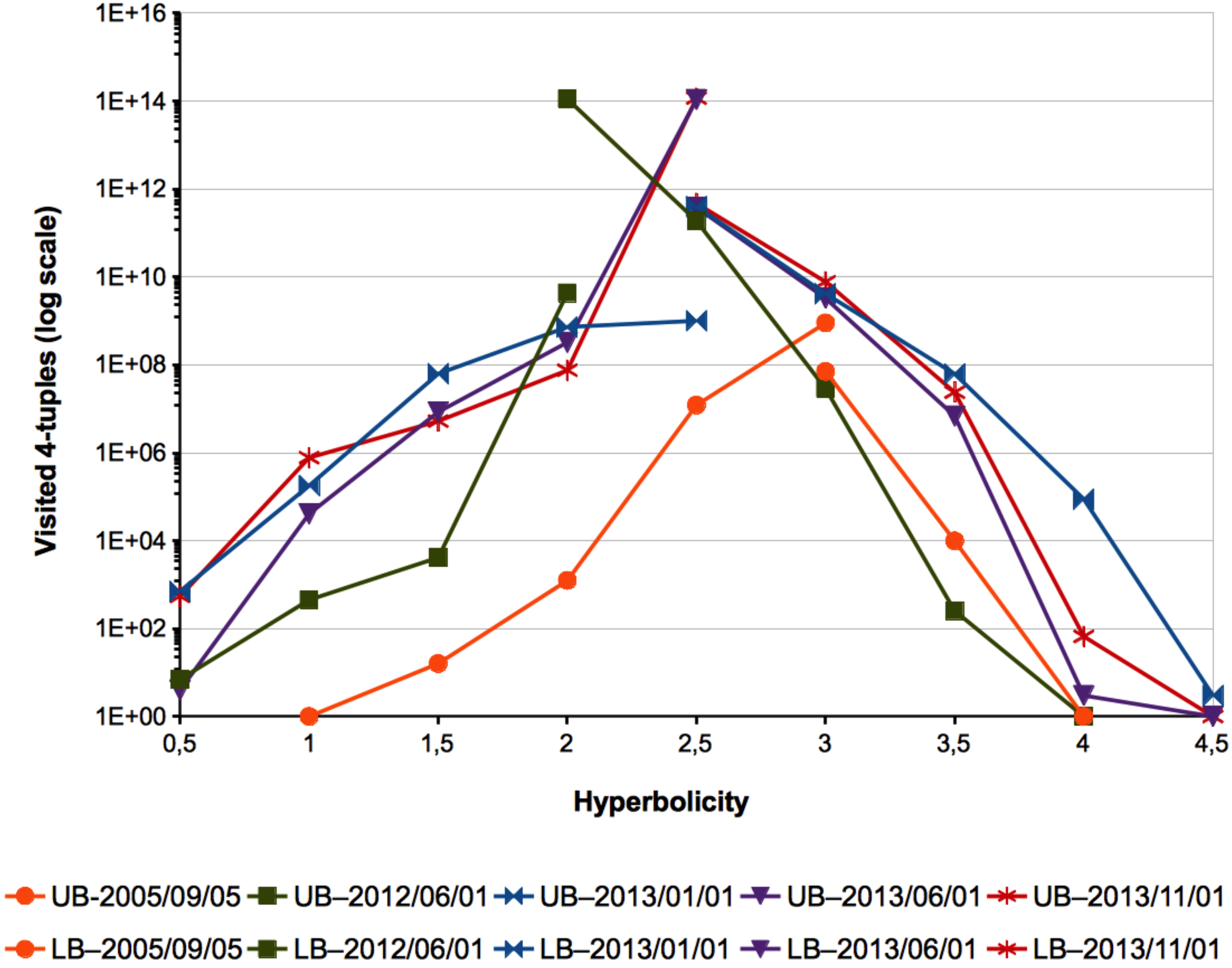
Computation time: from few seconds to 5 days

DIMES AS maps: δ = 2 for all maps

Evolution of upper and lower bounds

days

seconds



Cost of pre-processing

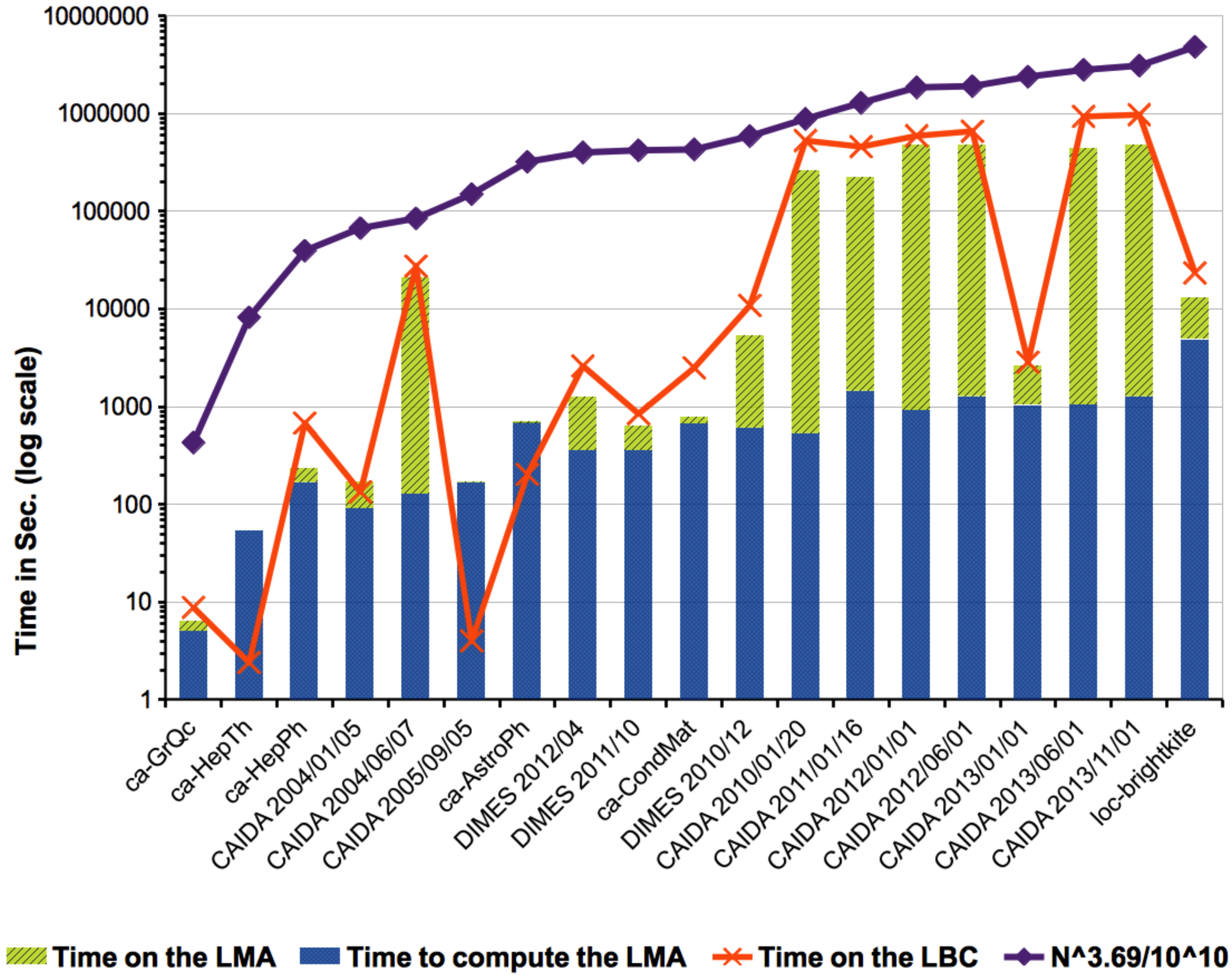
weeks

days

hours

minutes

seconds



Erdős–Rényi random graphs

G(n,p)

Lack of spectral gap **[Narayan, Saniee, Tucci 2010]**

- Threshold probability p = 1/n

  => G has a cycle of size k with high probability a.a.s.
- Large variations in the value of δ

(very) large p **[Mitsche, Pralat 2014]**

- $p \gg \log^5(n)/n.(\log\log(n))^2$
- δ(G) = diam(G)/2 with probability 1 a.a.s.
- δ(G) ~ log(n)/2

Some parallelism with OpenMP

```
δ := 0
#pragma omp parallel shared( δ, FA ) private( j )
for 1 ≤ i < (n choose 2) do
        (u,v) := FA[ i ]
        for 0 ≤ j < i do
                (x,y) := FA[ j ]
                if δ ≤ δ(u,v,x,y) then
                        #pragma omp critical
                        #pragma omp flush( δ(u,v,x,y) as δ )
                if d(u,v) ≤ 2δ then
                        break
#pragma omp barrier
return δ
```

Some parallelism with OpenMP

	δ	N	N_B	N_{LS}	T_{seq}	T_8	T_{16}
CAIDA Nov 2013	2.5	45 427	29 432	24 044	470 155	101 182	95 610
loc-brightkite	3	58 228	33 187	32 223	8 020	1 403	684
ca-AstroPh	3	18 772	15 929	15 325	29	5	3
ca-CondMat	3.4	23 133	17 234	13 451	116	18	10
ca-GrQc	**3.5**	**5 242**	**2 651**	**2 343**	**1.4**	**0.2**	**0.1**
ca-HepPh	3	12 008	9 025	8 287	66	10	5
ca-HepTh	4	9 877	5 898	5 270	0.4	0.07	0.04

PC with 2 8-cores CPUs

Summary

Pre-processing

• Decomposition into biconnected components	O(n+m)
• Split-decomposition	O(n+m)
• Decomposition by clique-separators + construction of substitutes	O(nm)
• Far-apart pairs	O(nm)

Main algorithm $O(n^4)$

Limitations:

- Overall computation time
- Space complexity (distance matrix) $O(n^2)$
  - n = 100 000 => 18 GB
  - n = 200 000 => 74 GB

Conclusion & Perspectives

It is possible to compute the hyperbolicity of AS maps !

- Combine decomposition methods and smart ordering

Open questions:

- Analysis: (average) time complexity for particular graph classes
- Improve initial lower bound, find new cuts, etc.
- Graphs with millions nodes
  - Without computing all distances (time and space constraints)

Available implementations

- Sage (http://www.sagemath.org) Python/Cython/C/C++
- Grph (http://grph.inria.fr) Java
- C + OpenMP (my web page)

Thanks

David Coudert

- david.coudert@inria.fr
- http://www-sop.inria.fr/members/David.Coudert

Project-team COATI

- Location: Sophia Antipolis, France
- Combinatorics, Optimization, Algorithms
- http://team.inria.fr/coati