Requêtes d'appartenance et autres méthodes "géométriques" pour trouver le diamètre d'un graphe

G. Ducoffe

Université de Bucarest & I.C.I. Bucarest, Roumanie

18 Novembre 2020

Travail réalisé en collaboration avec David Coudert, Feodor Dragan, Michel Habib, Alexandru Popa et Laurent Viennot.

Problem considered

Problem (DIAMETER)

Input: *A connected unweighted graph $G = (V, E)$.*

Output: *$diam(G) = max.\ \#$ of edges on a shortest path.*

- In $\mathcal{O}(nm)$ using n BFS **[folklore]**

Seems to require $\Theta(n)$ BFS to be solved on general graphs

- In $\tilde{\mathcal{O}}(n^{2.373})$ time using fast matrix multiplication **[Seidel, STOC'92]**

In both cases it is a reduction to `APSP`: $\Omega(n^2)$ time, even on sparse graphs.

Can we do better?

Lower bound

[Roditty and V. Williams, *STOC'13*]

Under SETH we cannot solve $\textsc{Diameter}$ **in** $\mathcal{O}(n^{2-\epsilon})$ **time**

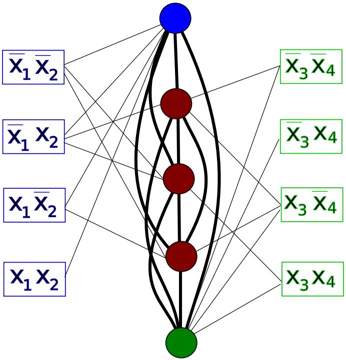
Partition variables in two halves A, B

S_A: all $2^{n/2}$ truth tables for A

S_B: all $2^{n/2}$ truth tables for B

Graph $G = (S_A \cup S_B \cup C, E)$

- C: clauses
- $c \in C$ and $a \in S_A$ adjacent iff a does not satisfy c



$diam(G) = 3 \iff$ satisfiable

$(x_1 \vee \bar{x}_2 \vee x_4) \wedge (x_1 \vee \bar{x}_3 \vee \bar{x}_4) \wedge (x_2 \vee \bar{x}_3 \vee x_4)$

$\textsc{Diameter}$ in $\mathcal{O}(N^{2-\epsilon}) \Longrightarrow \mathrm{SAT}$ in $\mathcal{O}((2^{n/2})^{2-\epsilon}) = \mathcal{O}(2^{(1-\epsilon')n})$

Lower bound cont'd

- We may reinterpret this construction as a "'disjoint sets" problem, sometimes called ORTHOGONAL VECTORS (OV).

  longtime-known relationship with DIAMETER **[Chepoi and Dragan, 1992]**

- Hardness results obtained for <u>constant</u> diameter (2 vs. 3)

- The construction works for **bipartite graphs** and **split graphs**

  [Borassi et al., *ETCS'16*]

- (Using Sparsification Lemma) $\forall \varepsilon, \exists c$ s.t. we cannot solve DIAMETER in $\mathcal{O}(n^{2-\varepsilon})$-time on split graphs with clique-number $\leq c \cdot \log n$.

Breaking the quadratic barrier

- (Almost) all conditional lower bounds for $\textsc{Diameter}$ start with a reduction from OV

(Open problem: find other obstructions to fast diameter computation)

- There are several graph classes where we can solve OV efficiently.

$\Longrightarrow$ *Does it imply fast diameter computation?*

- **Can we find a** *unifying framework* **in order to address this question for many graph classes at once?**

(based on a few methods or properties)

State of the art

Special graph classes – *quest for (almost)* **linear-time** *algorithms*

- Trees **[Jordan, 1869]**
- Outerplanar graphs **[Farley and Proskurowski., DAM'80]**
- Interval graphs **[Olariu, IJCM'90]**
- Dually chordal graphs **[Brandstädt et al., DAM'98]**
- Distance-hereditary graphs **[Dragan and Nicolai, DAM'00]**
- {claw,AT}-free graphs **[Corneil et al., DAM'01]**
- Plane triangulations and quadrangulations **[Chepoi et al., SODA'02]**
- Cactii **[Ben-Moshe et al., TCS'07]**

State of the art

Special graph classes – *quest for (almost)* **linear-time** *algorithms*

More recently: algorithms in $\mathcal{O}(n^{2-\varepsilon})$ and $f(k)n^{1+o(1)}$ time – *"FPT in P"*

- Trees **[Jordan, 1869]**
- Outerplanar graphs **[Farley and Proskurowski., DAM'80]**
- Interval graphs **[Olariu, IJCM'90]**
- Dually chordal graphs **[Brandstädt et al., DAM'98]**
- Distance-hereditary graphs **[Dragan and Nicolai, DAM'00]**
- {claw,AT}-free graphs **[Corneil et al., DAM'01]**
- Plane triangulations and quadrangulations **[Chepoi et al., SODA'02]**
- Cactii **[Ben-Moshe et al., TCS'07]**
- Graphs of constant treewidth **[Abboud et al., SODA'16]**
- Planar graphs **[Cabello, TALG'18]**
- Graphs of constant modular-width/split-width **[Coudert et al., SODA'18]**
- $(q, q-3)$-graphs **[Coudert et al., SODA'18]**
- Graphs of diameter $\Theta(n)$ **[Ducoffe, SOSA'19]**

State of the art

Special graph classes – *quest for (almost)* **linear-time** *algorithms*

More recently: algorithms in $\mathcal{O}(n^{2-\varepsilon})$ and $f(k)n^{1+o(1)}$ time – *"FPT in P"*

- Trees **[Jordan, 1869]**
- Outerplanar graphs **[Farley and Proskurowski., DAM'80]**
- Interval graphs **[Olariu, IJCM'90]**
- Dually chordal graphs **[Brandstädt et al., DAM'98]**
- Distance-hereditary graphs **[Dragan and Nicolai, DAM'00]**
- {claw,AT}-free graphs **[Corneil et al., DAM'01]**
- Plane triangulations and quadrangulations **[Chepoi et al., SODA'02]**
- Cactii **[Ben-Moshe et al., TCS'07]**
- Graphs of constant treewidth **[Abboud et al., SODA'16]**
- Planar graphs **[Cabello, TALG'18]**
- Graphs of constant modular-width/split-width **[Coudert et al., SODA'18]**
- $(q, q-3)$-graphs **[Coudert et al., SODA'18]**
- Graphs of diameter $\Theta(n)$ **[Ducoffe, SOSA'19]**

Is there a common (exploitable) property to all/most positive cases?

State of the art

Special graph classes – *quest for (almost)* **linear-time** *algorithms*

More recently: algorithms in $\mathcal{O}(n^{2-\varepsilon})$ and $f(k)n^{1+o(1)}$ time – *"FPT in P"*

- **Trees [Jordan, 1869]**
- **Outerplanar graphs [Farley and Proskurowski., DAM'80]**
- **Interval graphs [Olariu, IJCM'90]**
- Dually chordal graphs **[Brandstädt et al., DAM'98]**
- **Distance-hereditary graphs [Dragan and Nicolai, DAM'00]**
- {claw,AT}-free graphs **[Corneil et al., DAM'01]**
- **Plane triangulations and quadrangulations [Chepoi et al., SODA'02]**
- **Cactii [Ben-Moshe et al., TCS'07]**
- **Graphs of constant treewidth [Abboud et al., SODA'16]**
- **Planar graphs [Cabello, TALG'18]**
- **Graphs of constant modular-width/split-width [Coudert et al., SODA'18]**
- $(q, q-3)$**-graphs [Coudert et al., SODA'18]**
- Graphs of diameter $\Theta(n)$ **[Ducoffe, SOSA'19]**

Is there a common (exploitable) property to all/most positive cases?

State of the art

Special graph classes – *quest for (almost)* **linear-time** *algorithms*

More recently: algorithms in $\mathcal{O}(n^{2-\varepsilon})$ and $f(k)n^{1+o(1)}$ time – *"FPT in P"*

- **Trees [Jordan, 1869]**
- **Outerplanar graphs [Farley and Proskurowski., DAM'80]**
- Interval graphs **[Olariu, IJCM'90]**
- Dually chordal graphs **[Brandstädt et al., DAM'98]**
- **Distance-hereditary graphs [Dragan and Nicolai, DAM'00]**
- {claw,AT}-free graphs **[Corneil et al., DAM'01]**
- Plane triangulations and quadrangulations **[Chepoi et al., SODA'02]**
- **Cactii [Ben-Moshe et al., TCS'07]**
- **Graphs of constant treewidth [Abboud et al., SODA'16]**
- Planar graphs **[Cabello, TALG'18]**
- **Graphs of constant modular-width/split-width [Coudert et al., SODA'18]**
- $(q,q-3)$**-graphs [Coudert et al., SODA'18]**
- **Graphs of diameter** $\Theta(n)$ **[Ducoffe, SOSA'19]**

Is there a common (exploitable) property to all/most linear-time solvable cases?

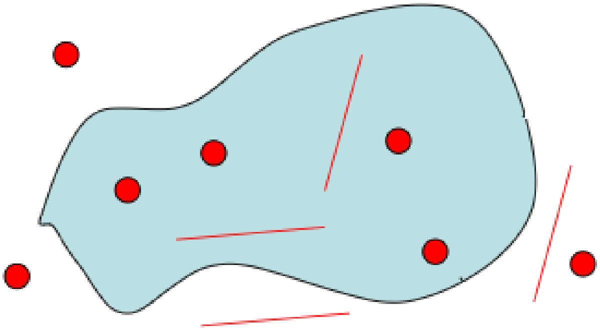
Range searching

Geometrical point of view

Global input: a collection P of objects in some space E^d

(points, lines, ...)

Range query: for some region $q \subseteq E^d$, report/count the objects in $P \cap q$

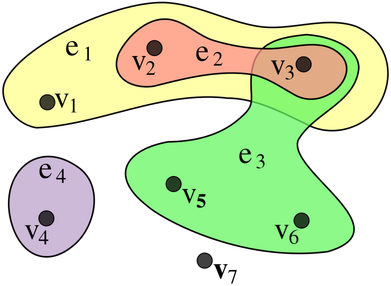


Range searching

Combinatorial point of view

Global input: hypergraph $\mathcal{H} = (V, E)$

- possibly infinite
- region $=$ hyperedge



Vertex-weights $(w_v)_{v \in V}$ taken in a (semi)group $(A, \oplus)$

Range query: for $e \in E$, compute the sum $\sum_{v \in e} w_v = w_{v_1} \oplus w_{v_2} \oplus \ldots$

Classic examples of range queries

- **Counting**: for $e \in E$, compute $|e|$
  - $\forall v \in V,\ w_v = 1$.
  - Semigroup $(\mathbb{N}, +)$.
- **Sum**: for $e \in E$, compute $\sum_{v \in e} w_v$
  - Semigroup $(\mathbb{N}, +)$.
- **Maximum**: for $e \in E$, compute $v \in E$ s.t. w_v is maximized
  - Semigroup $(\mathbb{N}, \max)$.

DIAMETER $\propto$ Range searching

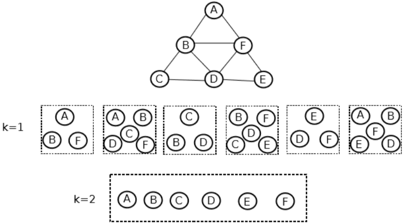
Define: $N^k[v] = \{u \in V \mid dist_G(v,u) \leq k\}$.

The **ball hypergraph** of G is: $\mathcal{B}(G) = (V, \{N^k[v] \mid v \in V, k \geq 0\})$.

- Decide whether $diam(G) \leq k$?

$\longrightarrow$ $\forall v \in V$, check if $|N^k(v)| = n$

$\longrightarrow$ reduces to n **counting** range queries

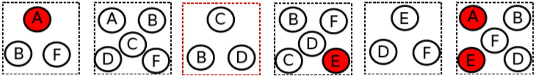


Remark: no direct access to $\mathcal{B}(G)$ (only to G)

VC-dimension

Let $\mathcal{H} = (X, \mathcal{R})$ be a hypergraph/range space/set family (possibly infinite)

- $S \subseteq X$ is **shattered** if $\{S \cap R \mid R \in \mathcal{R}\} = 2^S$
- The **VC-dimension** of $\mathcal{H}$ is the largest cardinality of a shattered subset **[Vapnik and Chervonenkis, 1971]**.



$S = \{A, E\}$ is shattered, VC-dim $= 2$

- Several applications in Learning Theory and Computational Geometry.

Theorem (Chazelle and Welzl, D & CG 1989)

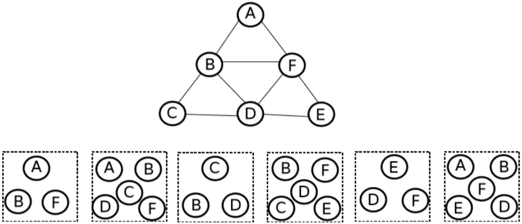
If $VCdim(\mathcal{H}) = d$ then, after polytime pre-processing, range queries can be answered in $\tilde{\mathcal{O}}(n^{1-2^{-d}})$ time.

VC-dimension **of graphs**

A graph is a hypergraph of VC-dimension $\leq 2 \ldots$

The **neighbourhood hypergraph** of $G = (V, E)$ is defined as $\mathcal{N}(G) = (V, \{N_G[v] \mid v \in V\})$.

$$\text{VC-dim}(G) =^{def} \text{VC-dim}(\mathcal{N}(G))$$



VC-dimension **of graphs**

Related work

- First introduced by **[Haussler and Welzl,DCG'87]**
- At most $\log n$ for any n-vertex graph.
- *Computation is LogNP-hard* **[Papadimitriou and Yannakakis, JCSS'96]** *and $W[1]$-hard* **[Downey et al., COLT'93]**.
- **Constant** for many graph classes:
  - **proper minor-closed [Anthony et al., DM'95]**
  - interval, girth ≥ 5, unit disk, ... **[Bousquet et al., SIDMA'15]**
  - graphs of constant interval number **[Ducoffe et al., COCOA'19]**
- Also studied for other remarkable subsets such as cliques, matchings, cycles, etc. **[Kranakis et al., DAM'87]**

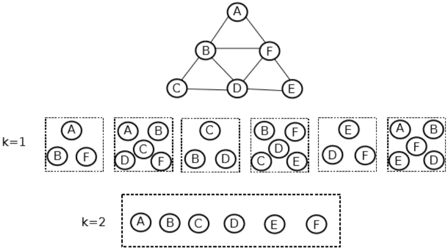
(more on that on next slide...)

Generalization: **distance VC-dimension**

Recall: $N^k[v] = \{u \in V \mid dist_G(v,u) \leq k\}$.

- The **k-neighbourhood hypergraph** of G is: $\mathcal{N}_k(G) = (V, \{N^k[v] \mid v \in V\})$.

k-dist-VC-dim$(G) =^{def}$ VC-dim$(\mathcal{N}_k(G))$



- The **ball hypergraph** of G is: $\mathcal{B}(G) = (V, \{N^k[v] \mid v \in V, k \geq 0\})$.

dist-VC-dim$(G) =^{def}$ VC-dim$(\mathcal{B}(G))$

Generalization: **distance VC-dimension**

Related work

Notion first studied by **[Chepoi et al., DCG'07]**

- Constant distance VC-dimension for:
  $\longrightarrow$ *proper minor-closed* graph classes **[Chepoi et al., DCG'07]**.

  $\longrightarrow$ *bounded clique-width* graph classes **[Bousquet and Thomassé, DM'15]**.

- A monotone graph class $\mathcal{G}$ is *nowhere dense* iff
$\forall k,\ \sup_{G \in \mathcal{G}} k\text{-dist-VC-dim}(G) < +\infty$ **[Nešetřil and Ossona de Mendez, RMS'16]**

- *Erdös-Posa* property for the hereditary classes $\mathcal{G}$ of constant distance VC-dimension **[Bousquet and Thomassé, DM'15]**. In particular, if $G \in \mathcal{G}$ and $diam(G) \leq 2R$, then $V(G)$ can be covered by $\mathcal{O}(1)$ balls of radius R.

- Applications in distributed distance computations in planar graphs **[Li and Parter, STOC'19]**.

Our results

[D., Habib, Viennot; SODA'20]

Theorem

For every $d > 0$, there exists a constant $\varepsilon_d \in (0;1)$ such that in deterministic time $\tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$ we can decide whether a graph of **VC-dimension** *at most d has diameter two.*

Our results

[D., Habib, Viennot; SODA'20]

Theorem

For every $d > 0$, there exists a constant $\varepsilon_d \in (0;1)$ such that in deterministic time $\tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$ we can decide whether a graph of **VC-dimension** *at most d has diameter two.*

Theorem

There exists a Monte Carlo algorithm such that, for every positive integers d and k, we can decide whether a graph of **distance VC-dimension** *at most d has diameter at most k. The running time is in $\tilde{\mathcal{O}}(k \cdot mn^{1-\varepsilon_d})$, where $\varepsilon_d \in (0;1)$ only depends on d.*

Our results

[D., Habib, Viennot; SODA'20]

Theorem

For every $d > 0$, there exists a constant $\varepsilon_d \in (0;1)$ such that in deterministic time $\tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$ we can decide whether a graph of **VC-dimension** *at most d has diameter two.*

Theorem

There exists a Monte Carlo algorithm such that, for every positive integers d and k, we can decide whether a graph of **distance VC-dimension** *at most d has diameter at most k. The running time is in $\tilde{\mathcal{O}}(k \cdot mn^{1-\varepsilon_d})$, where $\varepsilon_d \in (0;1)$ only depends on d.*

Theorem (simplified)

*The diameter of a H-**minor free** graph can be computed in time $\tilde{\mathcal{O}}(n^{2-\varepsilon_H})$, with a Monte Carlo algorithm, where $\varepsilon_H \in (0;1)$ is a constant that only depends on H.*

Some Remarks

- All our results also hold for **radius computation**.

- Our algorithms are "combinatorial" (basically, only using BFS and binary research trees).

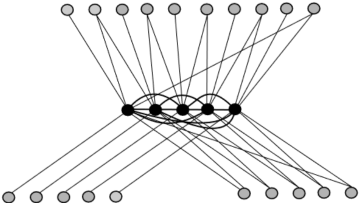
- **We do not need to know the (distance) VC-dimension**!

- We only use randomization for computing ε-nets (more on that later).

- $\varepsilon_d \to 0$ *exponentially fast* (*e.g.*, $\varepsilon_4 \leq 1/287 \ll 1/3$).

Warm-up: Split graphs with small clique

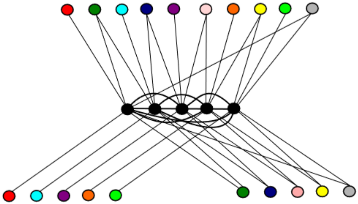
Consider a split graph $G = (K \cup S, E)$ of clique-number $|K| = \log^{\mathcal{O}(1)}(n)$.



Warm-up: Split graphs with small clique

Consider a split graph $G = (K \cup S, E)$ of clique-number $|K| = \log^{\mathcal{O}(1)}(n)$.

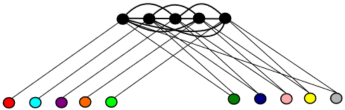
- Two vertices $s, t \in S$ are *twins* if $N(u) = N(v)$.



Warm-up: Split graphs with small clique

Consider a split graph $G = (K \cup S, E)$ of clique-number $|K| = \log^{\mathcal{O}(1)}(n)$.

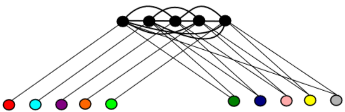
- Two vertices $s, t \in S$ are *twins* if $N(u) = N(v)$.
- We can keep one vertex per twin class **[Coudert et al., SODA'18]**.



Warm-up: Split graphs with small clique

Consider a split graph $G = (K \cup S, E)$ of clique-number $|K| = \log^{\mathcal{O}(1)}(n)$.

- Two vertices $s, t \in S$ are *twins* if $N(u) = N(v)$.
- We can keep one vertex per twin class **[Coudert et al., SODA'18]**.



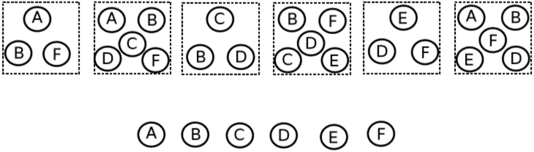
(Sauer-Shelah-Perles) VC-dim$(\mathcal{H})$ = d $\Longrightarrow \#\{Y \cap R \mid R \in \mathcal{R}\} = \mathcal{O}(|Y|^d)$.

$\longrightarrow$ There are only $\mathcal{O}(|K|^d) = \log^{\mathcal{O}(d)}(n)$ twin classes!

Stabbing Number

For a hypergraph $\mathcal{H} = (X, \mathcal{R})$, a spanning path = a total order over X.

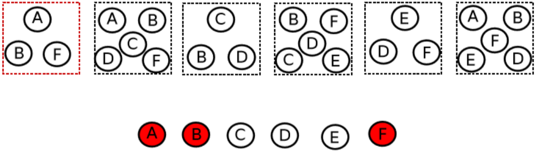
- stabbing number $\sim$ max. # of intervals to represent a hyperedge



Stabbing Number

For a hypergraph $\mathcal{H} = (X, \mathcal{R})$, a spanning path = a total order over X.

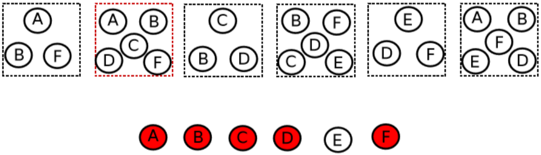
- stabbing number $\sim$ max. # of intervals to represent a hyperedge



Stabbing Number

For a hypergraph $\mathcal{H} = (X, \mathcal{R})$, a spanning path = a total order over X.

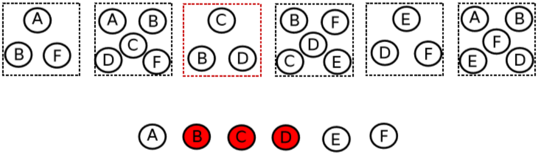
- stabbing number $\sim$ max. # of intervals to represent a hyperedge



Stabbing Number

For a hypergraph $\mathcal{H} = (X, \mathcal{R})$, a spanning path $=$ a total order over X.

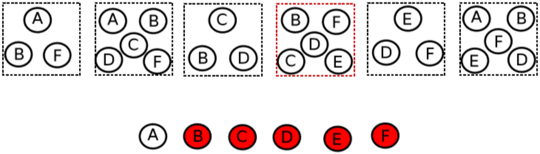
- stabbing number $\sim$ max. $\#$ of intervals to represent a hyperedge



Stabbing Number

For a hypergraph $\mathcal{H} = (X, \mathcal{R})$, a spanning path = a total order over X.

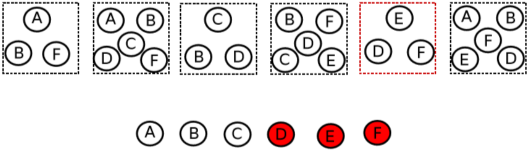
- stabbing number $\sim$ max. # of intervals to represent a hyperedge



Stabbing Number

For a hypergraph $\mathcal{H} = (X, \mathcal{R})$, a spanning path = a total order over X.

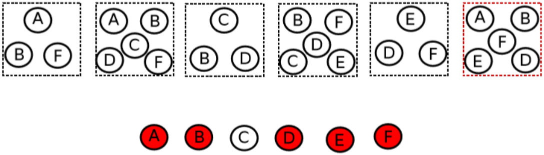
- stabbing number $\sim$ max. # of intervals to represent a hyperedge



Stabbing Number

For a hypergraph $\mathcal{H} = (X, \mathcal{R})$, a spanning path $=$ a total order over X.

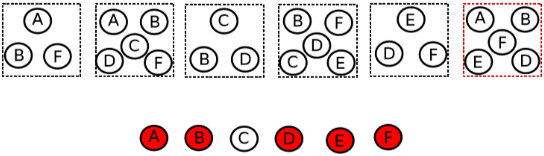
- stabbing number $\sim$ max. # of intervals to represent a hyperedge



Stabbing Number

For a hypergraph $\mathcal{H} = (X, \mathcal{R})$, a spanning path = a total order over X.

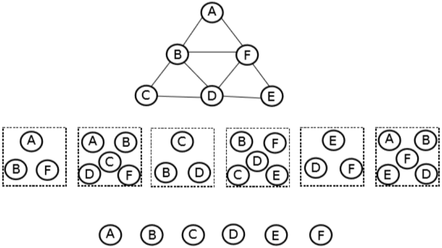
- stabbing number $\sim$ max. # of intervals to represent a hyperedge



Every hypergraph of VC-dim $\leq d$ has a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-1/2^d})$ **[Chazelle and Welzl, DCG'89]**

Application to the Diameter-Two Problem

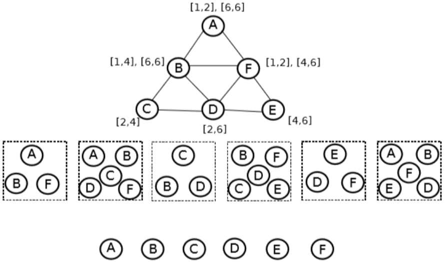
- Compute a spanning path of stabbing number t for $\mathcal{N}(G)$.



Application to the Diameter-Two Problem

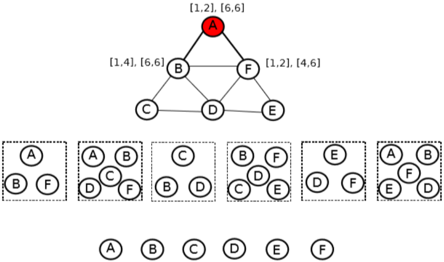
- Compute a spanning path of stabbing number t for $\mathcal{N}(G)$.

- Every vertex v collects the ends $I(v)$ of the $\mathcal{O}(t)$ intervals for $N[v]$.



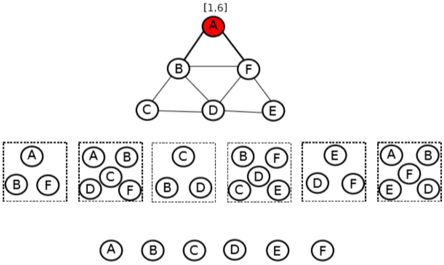
Application to the Diameter-Two Problem

- Compute a spanning path of stabbing number t for $\mathcal{N}(G)$.
- Every vertex v collects the ends $I(v)$ of the $\mathcal{O}(t)$ intervals for $N[v]$.
- Every vertex v collects the sets $I(u),\ \forall u \in N[v]$.



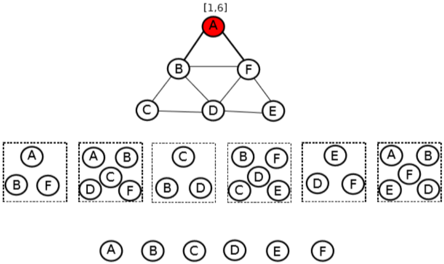
Application to the Diameter-Two Problem

- Compute a spanning path of stabbing number t for $\mathcal{N}(G)$.
- Every vertex v collects the ends $I(v)$ of the $\mathcal{O}(t)$ intervals for $N[v]$.
- Every vertex v collects the sets $I(u),\ \forall u \in N[v]$.
- Check whether $\forall v \in V,\ \bigcup_{u \in N[v]} I(u) = V$.



Application to the Diameter-Two Problem

- Compute a spanning path of stabbing number t for $\mathcal{N}(G)$.
- Every vertex v collects the ends $I(v)$ of the $\mathcal{O}(t)$ intervals for $N[v]$.
- Every vertex v collects the sets $I(u),\ \forall u \in N[v]$.
- Check whether $\forall v \in V,\ \bigcup_{u \in N[v]} I(u) = V$.



Complexity: Spanning Path Computation $+ \sum_v (deg(v)+1) \cdot \mathcal{O}(t)$

$$= \text{Spanning Path Computation} + \mathcal{O}(tm).$$

Fast Computation of a Suboptimal Spanning Path

For the neighbourhood hypergraph

- The proof of **[Chazelle and Welzl, DCG'89]** is constructive but leads to an $\tilde{\mathcal{O}}(n^3 m)$-time algorithm.

- A "classic" trick: arbitrarily partition V into $\mathcal{O}(n^\eta)$-size subsets $V_1, V_2, \ldots, V_p$, for some $p = \mathcal{O}(n^{1-\eta})$.

$\longrightarrow$ Apply the previous algorithm to each of the subhypergraphs $\mathcal{H}_i =^{def} (V_i, \{N_G[v] \cap V_i \mid v \in V\})$.

- Every $\mathcal{H}_i$ has VC-dim $\leq d \Longrightarrow$ Stabbing number in $\mathcal{O}(n^{1-\eta}) \times \tilde{\mathcal{O}}(n^{\eta(1-1/2^d)}) = \tilde{\mathcal{O}}(n^{1-\eta/2^d})$.
- (**Sauer's Lemma**) Every $\mathcal{H}_i$ has only $\mathcal{O}(n^{\eta \cdot d})$ *distinct* hyperedges $\Longrightarrow$ lower running time.

Generalization to k-neighbourhood hypergraphs

Main Issue: we cannot compute $\mathcal{N}_k(G)$.

Theorem (ε-net)

If VC-dim$(\mathcal{H}) \leq d$, then any random subset of size $\approx \frac{d}{\varepsilon} \log n$ intersects all hyperedges of cardinality $\geq \varepsilon \cdot n$.

Algorithm:

- Use a random subset S as above in order to partition the vertices into equivalence classes: $u \sim v \Longleftrightarrow^{def} N^k[u] \cap S = N^k[v] \cap S$.
- **(Sauer's Lemma)** There are only $\tilde{\mathcal{O}}(\varepsilon^{-d})$ equivalence classes $V_1, V_2, \ldots, V_q$.
- **(ε-net)** $u \sim v \Longrightarrow |N^k[u] \Delta N^k[v]| = \mathcal{O}(\varepsilon n)$. *We keep one representative $v_i \in V_i$ per equivalence class.* Let $\mathcal{H}_k =^{def} (V, \{N^k[v_i] \mid 1 \leq i \leq q\})$.
- Deduce a spanning path for $\mathcal{N}_k(G)$ from $\mathcal{H}_k$ **and** $\mathcal{N}_{k-1}(G)$.

Some final Remarks

- The stabbing numbers of $\mathcal{H}_k$ and $\mathcal{N}_k(G)$ are the same up to an additive $\mathcal{O}(\varepsilon n)$ error term.

$\longrightarrow$ A "good" spanning path of $\mathcal{H}_k$ also works for $\mathcal{N}_k(G)$.

- The hard part consists in computing the interval representation of all the k-neighbourhoods.

$\longrightarrow$ We need to compute $\forall i, \forall u \in V_i \setminus \{v_i\},\ N^k[v_i] \Delta N^k[u]$.

- This can be done efficiently using a "good" spanning path for $\mathcal{N}_{k-1}(G)$ (same trick as for the diameter-two problem).
- For H-minor free graphs, we can use instead an r-division.
  [Federickson, J. of Computing'87]

Take-home

- A new geometric framework for “fast” diameter computation on unweighted graphs.

- Generalizes many previously known cases + extends to all new classes (*e.g.*, proper minor-closed)

Conjecture: The diameter can be computed in $\tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$ time in the class of graphs of VC-dimension at most d, for some absolute constant ε_d.

$\longrightarrow$ True for chordal graphs **[Ducoffe & Dragan, Networks 2020]** and classes of polynomial expansion.

Toward (almost) linear-time algorithms

- Some drawbacks of the VC-dimension approach:
  - Running time is barely sub-quadratic
  - It does not cover all the positive cases

Example: It only considers graphs of treewidth $\mathcal{O}(1)$. But there exists a quasi-linear-time algorithm for computing the diameter within graphs of treewidth $o(\log n)$ **[Abboud et al., SODA'16]**.

Can we still use range searching in order to explain (and generalize) these results?

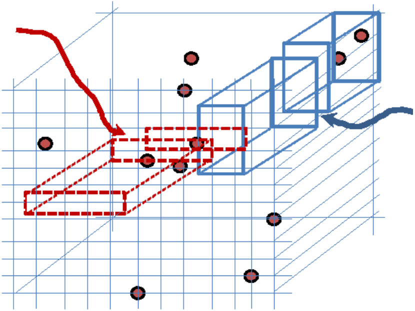
Orthogonal Range searching

The space has a fixed dimension k: $\mathbb{Z}^k$

Hyperedge = **box** = Cartesian product of k intervals

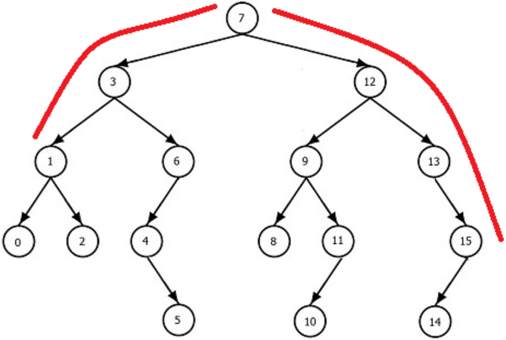
$$(l_1, u_1) \times (l_2, u_2) \times \ldots \times (l_k, u_k)$$

- Each interval can be closed/open/half-open
- We allow $l_i = -\infty$, resp. $u_i = +\infty$.



Allows faster resolution than general range searching

An old friend: Binary research trees



1) Preprocess each node v for having $\sum_{u \in T_v} w_u$

2) For (l_1, u_1), s.t. $l_1 \leq root \leq u_1$

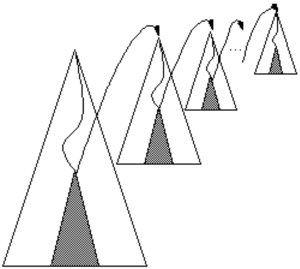
- Compute the least $x \geq l_1$ (left)
- Compute the largest $y \leq u_1$ (right)
- Sum all the right/left subtrees on the path from x/y to the root

Pre-processing time: $\mathcal{O}(n \log n)$

Query time: $\mathcal{O}(\log n)$

Generalization: k-range trees

Global input: static set V of k-dimensional points $\overrightarrow{q} = (q_1, q_2, \ldots, q_k)$



For every $i = 1 \ldots k$ do the following:

- Find $\overrightarrow{q^i} \in V$ s.t. q^i_i is a median of $\{q_i \mid \overrightarrow{q} \in V\}$;

  $\Longrightarrow V_{i,\text{Left}}, \overrightarrow{q^i}, V_{i,\text{Right}}$
- Recurse!

  Construction of right and left $(k-i+1)$-range trees

$$\begin{cases} C(n,1) = \mathcal{O}(n \log n) \text{ (balanced binary search tree)} \\ C(n,k) = C(n,k-1) + C(\lfloor n/2 \rfloor, k) + C(\lceil n/2 \rceil, k) + \mathcal{O}(n) \end{cases}$$

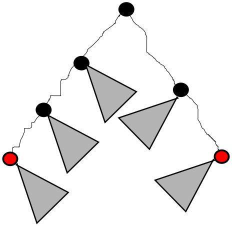
$\Longrightarrow$ in $\mathcal{O}(n \log^k(n))$-time

better analysis: $n\binom{k+\lceil \log n \rceil}{k} = 2^{\mathcal{O}(k)} n^{1+o(1)}$ **[Bringmann et al., IPEC'18]**

Answering to a query

For the ranges (l_i, u_i), $1 \leq i \leq k$.

- Find $\overrightarrow{x}$ s.t. $x_1 \geq l_1$ minimized;
- Find $\overrightarrow{y}$ s.t. $y_1 \leq u_i$ maximized;
- Locate the nearest common ancestor of $\overrightarrow{x}, \overrightarrow{y}$. Along the $\overrightarrow{x}\overrightarrow{y}$-path, do a $(k-1)$-range query for each right/left subtrees.



$\Longrightarrow$ in $\mathcal{O}(\log^k n)$-time

better analysis: $\binom{k+\lceil \log n \rceil}{k} = 2^{\mathcal{O}(k)} n^{o(1)}$ **[Bringmann et al., IPEC'18]**

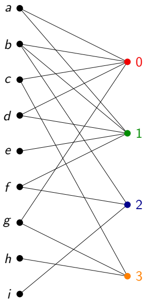
Applications to DIAMETER: **Tree decompositions**

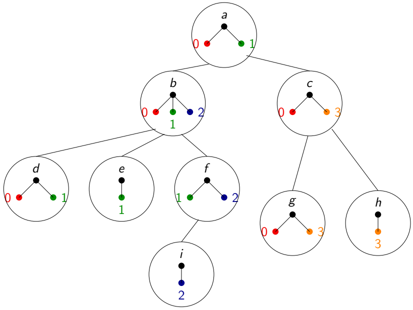
Representation of a graph as a tree preserving connectivity properties.

nodes of the tree $\sim$ subgraphs of G (*bags*)

the decomposition spans all the vertices and all the edges

edges of the tree $\sim$ separators of G





The maximum distance between disconnected vertices

S a small-size separator.

1) Fix $s \in S$.

2) Define $\forall v \notin S, \overrightarrow{p_s}(v) = (dist_G(v, s') - dist_G(v, s))_{s' \in S \setminus \{s\}}$.

3) $\forall v \notin S$, compute $u \notin S$ s.t.:

- S is an uv-separator;
- $\overrightarrow{p_s}(u) \geq -\overrightarrow{p_s}(v)$ ($\Longleftrightarrow dist_G(u, v) = dist_G(u, s) + dist_G(s, v)$);
- $dist_G(u, s)$ is maximized.

$\Longrightarrow$ textbook application of $|S|$-range tree!

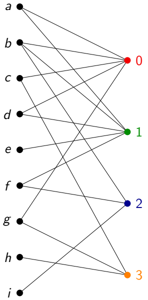
maximum range queries

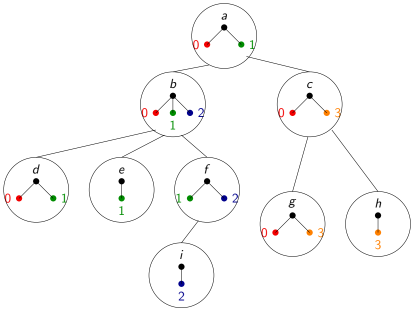
Application: **Treewidth**

- minimizing the **size** of bags

  width = max size of bags -1

  treewidth = min width of tree decompositions





$tw = 3$

5-approximation in $2^{\mathcal{O}(k)} \cdot n$-time **[Bodlaender et al., SICOMP 2016]**

Optimal diameter computation within bounded-treewidth graph classes

1) Compute a tree-decomposition of width $\mathcal{O}(tw(G))$ – in $2^{\mathcal{O}(k)}n$ time

2) Find a balanced separator of size $\mathcal{O}(k)$ – centroid in the tree-decomposition

3) Compute the maximum distance between disconnected vertices – in $2^{\mathcal{O}(k)}n^{1+o(1)}$ time with range trees

4) Recurse on each component – add weighted edges to preserve distances

Complexity: $2^{\mathcal{O}(k)}n^{1+o(1)}$ time

Optimal diameter computation within bounded-treewidth graph classes

1) Compute a tree-decomposition of width $\mathcal{O}(tw(G))$ – in $2^{\mathcal{O}(k)}n$ time

2) Find a balanced separator of size $\mathcal{O}(k)$ – centroid in the tree-decomposition

3) Compute the maximum distance between disconnected vertices – in $2^{\mathcal{O}(k)}n^{1+o(1)}$ time with range trees

4) Recurse on each component – add weighted edges to preserve distances

Complexity: $2^{\mathcal{O}(k)}n^{1+o(1)}$ time

This is sharp! DIAMETER **is SETH-hard for split graphs of clique-number** $\mathcal{O}(\log n)$**.**

Another application: **Giant diameter** graphs

Problem (h-Diameter)

Input: *A graph $G = (V, E)$; a constant $h \in (0; 1)$.*

Output: *The exact diameter of G if it is at least hn (otherwise, any value $< hn$).*

Some motivations:

- A common topic in Extremal Graph Theory
- Linear structure of some chemical/biological networks
- The OV construction applies to "small" diameter

Related work

The problem was introduced by **[Damaschke, *IWOCA'16*]**

Conjecture: For any h, we can solve h-DIAMETER in quasi linear time.

$\longrightarrow$ Partial progress:

- Linear-time algorithm for $h > 1/2$ **[Damaschke, *IWOCA'16*]**.

  (Heavily relies on biconnected decomposition)

- In randomized $\mathcal{O}(m + n \log n)$-time if $h > 1/3$.

- In randomized $\mathcal{O}(n^2/h)$-time and in deterministic $\Omega(n^2/h^3)$-time.
  Disclaimer: I am completely unsure about the constants. . .

Results

[D., *SOSA'19*]

- A Monte Carlo algorithm in time $\mathcal{O}(\frac{1}{h} \cdot (m + 2^{\mathcal{O}(\frac{1}{h})} n^{1+o(1)}))$.

  (1 BFS from a random vertex + orthogonal range queries)

- A deterministic algorithm in time $\mathcal{O}(\frac{1}{h^2} \cdot (m + 2^{\mathcal{O}(\frac{1}{h})} n^{1+o(1)}))$.

  ($\mathcal{O}(\frac{1}{h})$ BFS + orthogonal range queries)

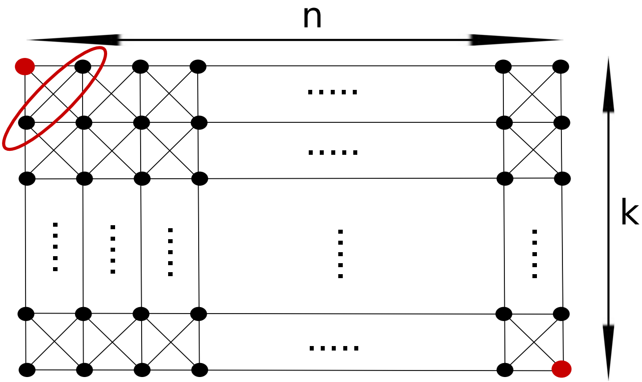
Our algorithms are both simpler and faster than previous work (but have practicality issues).

- **Conditional LB for $h = o(1/\log n)$.**

Overall Approach

- **Observation**: $\exists S,\ |S| \leq 1/h$ disconnecting a diametral pair.

(simple counting on BFS layers)



$\Longrightarrow$ *how to find S?*

A major source of difficulty in **[Damaschke, *IWOCA'16*]**

A (Simple!) Observation

- We can relax the constraint on S: we only need $|S| = \mathcal{O}(1/h)$.

Overall strategy: For an (unknown) diametral pair (x, y) compute $v \in V$ s.t. $dist_G(v, x) \leq hn/3$ (and so, $dist_G(v, y) \geq 2hn/3$).

$\Longrightarrow$ We can choose S between layers $hn/3$ and $2hn/3$ in BFS(v).

$$\mathbb{P}r_v[dist_G(v, x) \leq hn/3] \geq h/3$$

Result #1: Monte Carlo Algorithm

Algorithm GiantDiameter

```
Let v ∈ V picked u.a.r.
if ecc_G(v) < 2hn/3 then
    return ecc_G(v).
Find a layer i ∈ {⌈hn/3⌉, ..., ⌊2hn/3⌋} s.t. |L_i(v)| ≤ 3/h.
Compute D_i := max{dist_G(x, y) | L_i(v) is an xy-separator}.
return max{D_i} ∪ {ecc_G(u) | u ∈ L_i(v)}.
```

- Correct if $dist_G(v, x) \leq hn/3$ or $dist_G(v, y) \leq hn/3$;
- The bottleneck is the computation of D_i (solved using $3/h$ BFS + **orthogonal range queries**)

Remark: Before running the algorithm, we may compute a 2-approx for h using a single BFS.

Result #2: Derandomization

Definition

S is a k-distance dominating set $\iff \forall v \notin S, dist_G(v, S) \leq k$.

- If S is a $(hn/3)$-distance dominating set then, we can solve h-Diameter by running $|S|$ times Algorithm GiantDiameter.

- (Meir and Moon, *J. of Math. 1975*) Every graph has a k-distance dominating set of size $\leq \left\lceil \frac{n}{k+1} \right\rceil$.

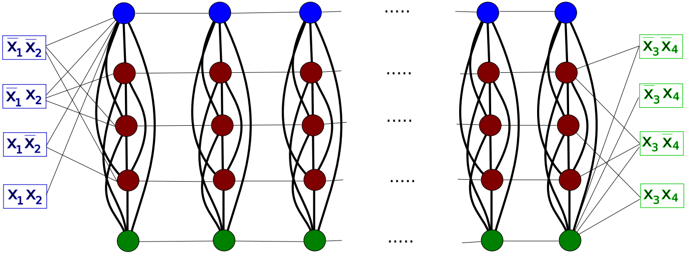
1) Compute a spanning tree T and a diametral pair x, y of T

2) Take one layer over $k+1$ in a BFS starting at x.

Result #3: A Conditional Lower-bound

Repeat n times the clique of an arbitrary split graph

(matching between every two consecutive copies)



$\forall \varepsilon, \exists c$ s.t. we cannot solve $(\frac{1}{c \cdot \log n})$-Diameter in $\mathcal{O}(n^{2-\varepsilon})$-time.

Computing diameters below $\Theta(n/\log n)$ is hard!

More applications of orthogonal range searching

Ongoing work

A first observation: Small (balanced) separators are often used in the design of **distance-labeling schemes**

$\Longrightarrow$ **How about applying orthogonal range searching to more general distance oracles?**

Partial results:

- For **hub labels** of size $\leq k$

  application to graph classes of bounded expansion

- For embeddings in the Cartesian product of $\leq k$ trees

  application to subclasses of median graphs

- For some variation of edge-hub labeling

  application to the graphs of **clique-width** *at most* k

In $\mathcal{O}(2^{\mathcal{O}(k)}(n+m)^{1+o(1)})$ time **and** this is sharp under SETH

Short conclusion

- Range searching techniques allow to handle the DIAMETER problem on **many** important graph classes, in a unifying and "efficient" way.

- Improved special cases of range searching lead to a better understanding of the (almost) linear-time solvable instances for DIAMETER

- Some interesting connections exist between faster diameter computation and important geometric properties

(VC-dimension, **Helly-type properties**)

Merci de votre attention!