Theory of Practical Graph Algorithms – Part II

Laurent Viennot – Mauro Sozio

MPRI – laurent.viennot@inria.fr – Inria (rue Barrault)

Part II : Outline

- Diameter/radius computation – Hardness in P (1 lecture)
- Shortest path queries – Distance labeling (1.5 lectures)
- Temporal graphs (1 lecture)
- Graph neural networks (0.5 lecture)

Diameter/radius computation

Hardness of diameter computation and fine-grained complexity

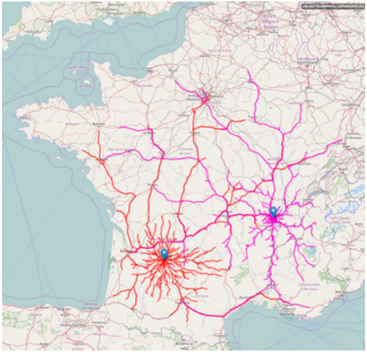
Practical algorithms and why they work

Shortest path queries

A line of practical algorithms

Some theoretical insight in their efficiency

Links with the theory of distance oracles



Temporal graphs

A hierarchy of models

Classical graph problems revisited

Graph neural networks

Expressive power and isomorphism tests

(under construction)

Diameter

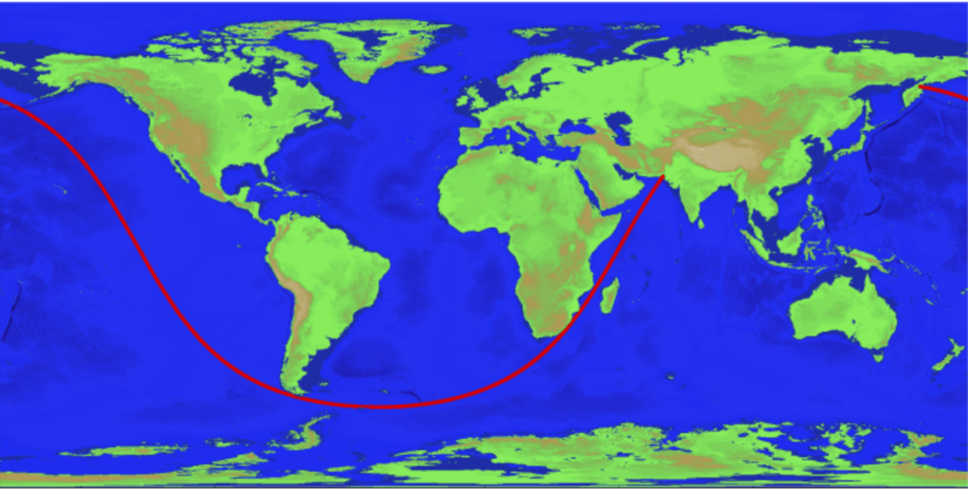
How big is the world?

Longest straight line on water

Reddit user kepleronlyknows 2012 (in r/MapPorn) :

Longest straight line on water

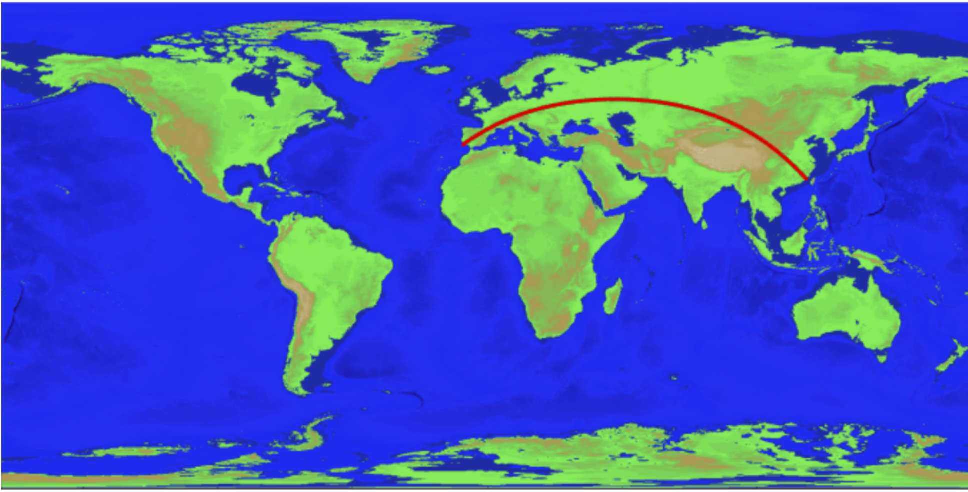
[Chabukswar, Mukherjee 2018] :



“The problem was approached as a purely mathematical exercise. The authors do not recommend sailing or driving along the found paths.”

Longest straight line on land

[Chabukswar, Mukherjee 2018] :



“The problem was approached as a purely mathematical exercise. The authors do not recommend sailing or driving along the found paths.”

World longest road trips

`http://gang.inria.fr/road/`

Graph notations

A (weighted directed) graph G is defined by :

a set $V(G)$ of nodes/vertices (or simply V),

a set $E(G)$ of arcs/edges (or simply E),

a length function $\ell : E \to \mathbb{R}^+$: $\ell(uv)$ is the length/weight of arc $uv \in E$ (default is 1).

Size : $n(G) = |V(G)|$ (or simply n) is the number of nodes, and $m(G) = |E(G)|$ (or simply m) is the number of edes.

G is undirected when $uv \in E \iff vu \in E$.

G is unweighted when $\ell(uv) = 1$ for all $uv \in E$.

A uv-path with k hops is a sequence of arcs $P = x_0x_1, \ldots, x_{k-1}x_k$ with $x_0 = u$ and $x_k = v$, its length is $\ell(P) = \sum_{i=1}^{k} \ell(x_{i-1}x_k)$.

The distance $d_G(u, v)$ (or simply $d(u, v)$) from u to v is the minimum length of a uv-path.

Graph parameters

Practical graphs : `https://networkrepository.com/`

General shape :

size, connectivity, average distance (Milgram exp.),...

Six degrees of separation

Node measures :

degree, pagerank, xx-centrality,...

Periodic Table of Network Centrality [Schoch 15]

Computing graph parameters

Average distance (degree of separation) :

- All pairs distances : $O(nm)$ or $O(n^\omega \log n)$ unweighted [Alon et al 91 ; Seidel 95] or $O(W^{(\omega-1)/2} n^{(3+\omega)/2} \log n)$ [Galil Margalit 97] (with $O(n^\omega)$ matrix multiplication)
- Approximation with sampling and empirical average.

Centralitiy measures : what nodes are "central" ?

- betweenness $b(v) = \sum_{s,t} \frac{\sigma_{st}(v)}{\sigma_{st}}$ [Freeman 77 ; Anthonisse 71]

σ_{st} : number of st-shortest paths, $\sigma_{st}(v)$: that pass through v

- stress $s(v) = \sum_{s,t} \sigma_{st}(v)$ [Shimbel 53]
- (harmonic) closeness $c(v) = \sum_t \frac{1}{d(v,t)}$ [Marchiori, Latora 2000]
- graph closeness $g(v) = \frac{1}{\max_t d(v,t)}$ [Hage Harary 95]
- eccentricity $e(v) = \max_t d(v,t)$ [Hage Harary 95]
- Often $O(nm)$ (for all v) [Brandes 01]

Diameter : maximum distance

- Diameter $diam(G) = \max_v e(v)$
- Radius $rad(G) = \min_v e(v)$

Diameter computation

Naive algorithm : compute $e(v)$ through a BFS for all $v \in V$.

Complexity : $\Theta(nm) = O(m^2)$.

Can we do better ?

Should we search for $O(m^c)$-time with $c < 2$?

The $O(nm)$ barrier

Theorem [Roditty Vassilevska-Williams 13] : There is no $O(m^{2-\varepsilon})$-time diameter algorithm under SETH (for any $\varepsilon > 0$).

Proof idea : reduce SAT (satisfiability of a CNF formula with N variables) to Diameter > 2.

Definition :
$s_k = \inf \left\{ \delta \mid \text{k-SAT has a } O(2^{\delta N})\text{-time algorithm} \right\}$

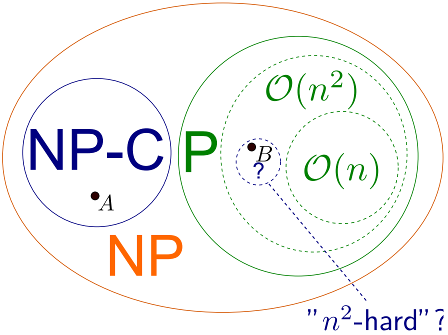
Exponential time hypothesis (ETH) [Impagliazzo Paturi 01] : $s_3 > 0$. (ETH $\Rightarrow$ P $\neq$ NP)

Strong ETH (SETH) (informal) : $\forall \varepsilon > 0$, there is no $O(2^{(1-\varepsilon)N})$-time algorithm for SAT.

Formal SETH : $\lim_{k \to \infty} s_k = 1$, i.e. for all $\varepsilon > 0$, there exists k s.t. k-SAT cannot be solved in $O(2^{(1-\varepsilon)N})$-time.

Knwown upper bounds : 3-SAT in $O(2^{0.39N})$ and k-SAT in $O(2^{(1-c/k)N})$ for some $c > 0$. [PaturiPSZ 05, Hertli 14]

Hardness in P : fine-grained complexity



Hardness in P : fine-grained complexity

What is the true c in $O(n^c)$ algorithms ?

Quasi-linear reduction : $P \leq_c Q$ if any $O(n^{c-\varepsilon})$ algorithm for Q can be turned into an $O(n^{c-\varepsilon'})$ algorithm for P.

Exemple : OV=DisjSet $\leq_2$ Diameter as the existence of two disjoint sets in a collection of n sets in a universe of size $\widetilde{O}(1)$ can be reduced to a diameter problem on $\widetilde{O}(n)$ nodes in $\widetilde{O}(n)$ time.
($\widetilde{O}(.)$ hide poly-log factors.)

Exercise : prove HitSet $\leq_2$ Radius.

Hardness in P : fine-grained complexity

Tentative hard problems :

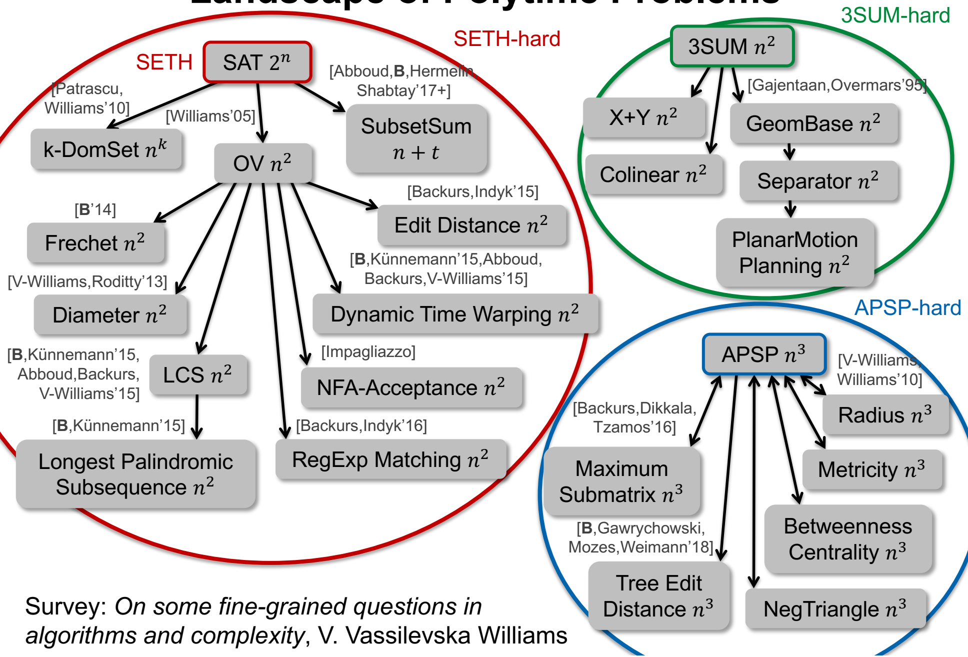
- HitSet for $O(n^2)$: given two collections A, B, does a set of A hits all sets of B ?
- 3SUM for $O(n^2)$: given n integers, do 3 of them sum to 0 ?
- All Pairs Shortest Paths (APSP) for $O(nm)$.
- d-dim OV for $O(n^2)$ ($d = c \log n$) : given two sets A, B of binary vectors of dimension d, does there exists $a \in A$ and $b \in B$ such that a and b are orthogonal.
- k-OV for $O(n^k)$: given n vectors of dimension d, does there exists k of them with product zero ?

Conjectures :

- HitSet hypothesis : HitSet requires time $n^{2-o(1)}$.
- OVH : OV requires time $n^{2-o(1)}$.
- k-OVH : k-OV requires time $n^{k-o(1)}$.

Exercise : prove HitSetHyp $\Rightarrow$ OVH.

Landscape of Polytime Problems



Survey: *On some fine-grained questions in algorithms and complexity*, V. Vassilevska Williams

Hardness of approximation

Diameter can be 2-approximated in $O(m)$-time.

$(2 - 1/k - \delta)$-approximation requires time $m^{1+1/(k-1)-o(1)}$ under SETH. [Dalirrooyfard, Li, Williams 21]

There exists a 3/2-approximation in $\widetilde{O}(m^{3/2})$-time. [Chechik LRSTW 14]

No $(3/2 - \delta)$-approximation in $m^{2-\varepsilon}$-time under SETH. [Roditty, Williams 13]

No $3/2$-approximation (even $5/3 - \delta$-approx) in $m^{3/2-\varepsilon}$-time under SETH. [Backurs RSWW 18]

Practical diameter

Diameter of Facebook graph is ?41 in 2011 [Backstrom et al 12] (720m nodes, 69g edges).

iFUB algorithm [Crescenzi et al 10] : carefully choose BFS sources and improve lower/upper bounds on diam(G) (17 BFSs for Facebook2011).

iFUB (iterative Fringe Upper Bound)

Given a BFS from s, F_j are the nodes at distance j from s.

Lemma : If a node x in F_i with $i \leq j$ has eccentricity $e(x) > 2j$, then some node $y \in F_k$ with $k > j$ has eccentricity $e(y) \geq e(x) > 2j$.

Corollary : If $D_L := \max_{y \in F_k, k>j} e(y)$, $D_L \leq diam(G) \leq \max(2j, D_L)$.

Algorithm : Scan nodes in F_j for $j = e(s), \ldots, 1$ and update $D_L := \max_{y \in F_j} e(y)$ until $2j \leq D_L$.

Analysis : a BFS for each node in F_k with $k > diam(G)/2$.

Exercise : choice of s ?

Exercise : hard graph for iFub ?

Efficient in random power-law graphs [Borassi et al 17].

Heuristic diameter

Two sweeps : Given G and $u \in V(G)$, do BFS(u) and BFS(v) where $v = A(u)$ is the antipode of u (last visited node in BFS(u)), return $d(v, A(v))$.

Theorem : correct if G is a tree [Handler 73], and $+1$ approximation if G is chordal [Corneil et al 03].
Chordal : no induced cycle of length > 3.

Often within $+1$ in practice.

Tentative center : the middle node in-between v and $A(v)$.
Center : a node with minimum eccentricity.

Bound eccentricities [Takes, Kosters 2011] (revisited)

A BFS from s provides bounds on eccentricities :

$$\forall u,\ d(s,u) \leq e(u) \leq d(s,u) + e(s)$$

$X := \emptyset$; scanned nodes
Maintain $e_L(u) := \max_{s \in X} d(u,s)$
and $e^U(u) := \min_{s \in X} d(u,s) + e(s)$ for all $u \in V(G)$.
While $\min_{u \in V} e_L(u)$ or $\max_{u \in V} e^U(u)$ is not tight **do**
  Select $u \in V$ s.t. $e_L(u)$ is minimal and add it to X.
  Select $u \in V$ s.t. $e^U(u)$ is maximal and add it to X.

Last generation : exact sum-sweep [Borassi et al. 2017].

Why such algorithms work?

What graph property enables fast diameter computation?

iFUB :

- $\mathrm{diam}(G)$ close to $2\,\mathrm{rad}(G)$,
- few far nodes.

Takes and Kosters algorithm?

Practical graphs have small certificates.

Small certificate implies $O(n^{2-\varepsilon})$ algorithm

X is a diameter certificate if
$\forall u, \min_{x \in X} d(u,x) + ecc(x) \leq diam(G)$

Certificate of size $\ell(n)$ implies a $O(m\ell(n))$ non-deterministic algorithm.

Theorem [Dragan et al. 2024] For any class of graphs with diameter certificates of size $\ell(n)$, there exists a randomized algorithm for diameter running in $O(m\sqrt{\ell(n)n}\log^{3/2} n)$ time.

Diameter parameterized by certificate size

X is a diameter certificate if
$\forall u, \min_{x \in X} d(u,x) + ecc(x) \leq diam(G)$

Equivalently : $V \subseteq \cup_{x \in X} B[x, diam(G) - ecc(x)]$
where $B[x,r] = \{u : d(u,x) \leq r\}$ (ball of radius r centered at x).

X corresponds to a covering with certain balls!

Diameter parameterized by certificate size

Main idea : binary search the value D of the diameter, while solving set cover with $\{B[y, D - ecc(y)] : y \in V\}$.

Problem : The set-cover instance may have size $\Omega(n^2)$.

Workaround : Greedy set-cover with an implicit representation through set/containment queries [Sen, Muralidhara 2010].

Problem : Still requires to know $ecc(y)$ for all y.

Workaround :

- Find for all y, an estimate $r(y) \leq ecc(y)$ s.t. $|B[y, r(y)]| \geq (1 - \varepsilon)n$ using random sampling ($O(\varepsilon^{-1} \log n)$ BFSs);
- run greedy set-cover using $\{B[y, D - r(y)] : y \in V\}$ and selecting at most $q = \ell(n) \log n$ balls with centers $x_1, \ldots, x_q$;
- $\forall i$, check $ecc(v) \leq D$ for all $v \in V \setminus B[y, D - r(x_i)]$ ($q\varepsilon n$ queries).

Plugging $\varepsilon = \frac{1}{\sqrt{n\ell(n) \log n}}$ yields the result.

Open questions

Certificates for diameter approximation?

Certificates for other hard problems in P?

Few far nodes [Borassi, Crescenzi, Trevisan 2017]

Random power law graph :

- sequence of degrees s.t. $\Theta(n/d^\beta)$ nodes with degree d,
- create edges according to the configuration model.

For $2 < \beta < 3$,

- $D = (1 + o(1))c(\beta) \log n$,
- $dist_{avg} = (2 + o(1)) \log_{1/(\beta-1)} \log n$,
- nb vertices at dist $\geq D/2$ from a random v is at most $n^{O(\varepsilon)}$ for $n > n_\beta$ ($\varepsilon = 0.2$ experimentally).

Diameter certificate [Dragan et al. 2018]

For any $L, U \subseteq V(G)$ and $u \in V(G)$:

$$e_L(u) \leq e(u) \leq e^U(u), \text{ where } \begin{cases} e^U(u) = \min_{x \in U} d(u,x) + e(x) \\ e_L(u) = \max_{x \in L} d(u,x) \end{cases}$$

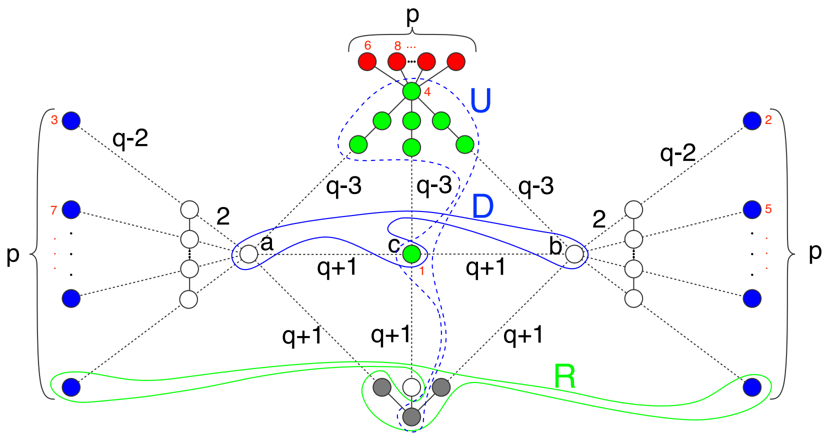
L is a radius certificate if $e_L(u) \geq rad(G)$ for all $u \in V$.

For all $u \in V$, $\exists x \in L$ such that $d(x,u) \geq rad(G)$:
L is a covering with $\{\overline{B}(u, rad(G)) : u \in V\}$.

U is a diameter certificate if $e^U(u) \leq diam(G)$ for all $u \in V$.

For all $u \in V$, $\exists x \in U$, such that $d(x,u) \leq diam(G) - e(x)$:
U is a covering with $\{B[u, diam(G) - e(u)] : u \in V\}$.

Example



D : certificate for diam$(G) = 4q - 2$, R for rad$(G) = 2q + 1$

Exercice : what about a path ?, a grid ?

Diameter computation as a primal-dual alg. [Dragan et al. 2018]

$U := \emptyset$; $K := \emptyset$
Do
  Select u such that $e^U(u)$ is maximal.
  $K := K \cup \{u\}$
  Select x such that $d(u,x) + e(x) = e(u)$.   E.g. $x := u$.
  $U := U \cup \{x\}$
while $\max_{u \in K} e(u) < \max_{u \in V} e^U(u)$
Return $e(b)$, b and U where $e(b) = \max_{u \in K} e(u)$.

U is a covering with $\{B[u, \mathrm{diam}(G) - e(u)] : u \in V\}$.

K is a packing for $\{B(u, \frac{1}{3}(\mathrm{diam}(G) - e(u))) : u \in V\}$.

Approximation ratio $\frac{\pi_{1/3}}{\pi_{[1]}}$, $O(\pi_{1/3})$ BFS.

Most practical graphs have diameter certificate $|U| < 30$.

Radius computation : iterate two-sweeps as a primal-dual alg. [Dragan et al. 2018]

```
L := ∅ ; K := ∅
Do
  | Select u ∈ V such that e_L(u) is minimal.
  | Compute distances from u (BFS from u).
  | a := argmax_{v∈V}(d(u,v), π(v))   /* Antipode of u.   */
  | K := K ∪ {u}
  | L := L ∪ {a} (BFS from a)
while min_{u∈V} e_L(u) < min_{u∈K} e(u).
Return e(c), c and L where c = argmin_{u∈K} e(u).
```

$\mathrm{rad}(G) = \min_{u \in V} e(u) \geq \min_{u \in V} e_L(u)$ and $\min_{u \in K} e(u) \geq \mathrm{rad}(G)$

K is a packing for $\{\mathrm{Antipode}_r^{-1}(u) : u \in V\}$.

$O(|\mathrm{Antipode}_r(V)|)$ BFSs : 2 per round (from u and a).

Most practical graphs have < 40 antipodes and radius certificate with $|L| \leq 5$.

Best radius algorithm using one-to-all distances queries

One-to-all distances algorithm :

- query distances from a node,
- use triangle inequality.

Theorem [Dragan et al. 2018] : If a one-to-all distances algorithm queries only k nodes to obtain the radius of a graph G, then G has a lower certificate of 2k nodes.

Further reading

[Vassilevska Williams 2018]
On some fine-grained questions in algorithms and complexity.
ICM 2018.
https://people.csail.mit.edu/virgi/eccentri.pdf

Exercize for next week

Prove HitSet $\leq_2$ Radius.

Send a drawing and a short proof to `laurent.viennot@inria.fr`.