Subexponential algorithms in non sparse classes of graphs

Édouard Bonnet

Department of Computer Science, Middlesex University, London

Lyon, May 15th, 2017
not all the NP-complete problems are as hard when one looks at their optimal running time
Subexponential algorithms

NP-hardness:

your problem is not solvable in polytime, unless 3-SAT is.
very widely believed but do not give an evidence against algorithms running in, say, $2^{n^{1/100}}$. 
Subexponential algorithms

NP-hardness:

your problem is not solvable in polytime, unless 3-SAT is.

very widely believed but do not give an evidence against
algorithms running in, say, $2^{n^{1/100}}$.

ETH-hardness:

stronger assumption than $P \neq NP$ is ETH asserting that no
$2^{o(n)}$ algorithm exists for 3-SAT

Allows to prove stronger conditional lower bounds

linear reduction from 3-SAT: no $2^{o(n)}$ algorithm for your
problem, quadratic reduction: no $2^{o(\sqrt{n})}$ algorithm, etc.
Biased viewpoint of this talk

$2^{O(\sqrt{n})}$ (or at least $2^{O(n^c)}$ with $c < 1$) vs no $2^{o(n)}$ under ETH?

For optimisation problems: $n^{O(\sqrt{k})}$ vs no $n^{o(k)}$ (or $n^{o(k/\log k)}$)?

we will not care about the log factors in the exponent.
Biased viewpoint of this talk

$2^O(\sqrt{n})$ (or at least $2^O(n^c)$ with $c < 1$) vs no $2^o(n)$ under ETH?

For optimisation problems: $n^O(\sqrt{k})$ vs no $n^o(k)$ (or $n^o(k/\log k)$)?

we will not care about the log factors in the exponent.

$n^O(\sqrt{k}) \overset{\text{often}}{\sim} 2^O(\sqrt{n \log n})$ because $k \overset{\text{often}}{\leq} n$.

Notable exception: **Chordal Completion** can even be solved in $2^O(\sqrt{k \log k})$ but not in $2^o(n)$ under ETH ($k$ can be $\Omega(n^2)$).
Biased viewpoint of this talk

$2^{O(\sqrt{n})}$ (or at least $2^{O(n^c)}$ with $c < 1$) vs no $2^{o(n)}$ under ETH?
For optimisation problems: $n^{O(\sqrt{k})}$ vs no $n^{o(k)}$ (or $n^{o(k/\log k)}$)?
we will not care about the log factors in the exponent.

We will focus on geometric non necessarily sparse graphs:
intersection graphs: disks, balls, segments, strings, etc.
visibility graphs.
Sparse classes and dense classes

sparse class $\mathcal{C}$: $\exists \alpha_{\mathcal{C}}, \forall G \in \mathcal{C}$ such that $|E(G)| \leq \alpha_{\mathcal{C}}|V(G)|$.

dense class $\mathcal{C}$: $\exists \alpha_{\mathcal{C}}$ and infinitely many graphs $G \in \mathcal{C}$ with $|E(G)| \geq \alpha_{\mathcal{C}}|V(G)|^2$.
Examples

Sparse classes:
- forests (with $\alpha_C = 1$)
- 3-regular graphs (with $\alpha_C = 3/2$)
- planar graphs (with $\alpha_C = 3$)
- graphs with average degree bounded by $d$ (with $\alpha_C = d/2$)

Dense classes:
- general graphs
- chordal graphs
- intersection graphs of disks
- cliques

Planar graphs are especially nice sparse graphs since they admit small balanced separators...
Examples

Sparse classes:
- forests (with $\alpha_C = 1$)
- 3-regular graphs (with $\alpha_C = 3/2$)
- planar graphs (with $\alpha_C = 3$)
- graphs with average degree bounded by $d$ (with $\alpha_C = d/2$)

Dense classes:
- general graphs
- chordal graphs
- intersection graphs of disks
- cliques

Planar graphs are especially nice sparse graphs since they admit small balanced separators...
Square root phenomenon on planar graphs

Many problems are solvable in $2^{O(\sqrt{n})}$ in planar graphs, and unlikely solvable in $2^{o(n)}$ in general graphs.
Many problems are solvable in $2^{O(\sqrt{n})}$ in planar graphs, and unlikely solvable in $2^{o(n)}$ in general graphs.
Square root phenomenon on planar graphs

Max Independent Set, 3-Coloring, Hamiltonian Path...
Square root phenomenon on planar graphs

Max Independent Set, 3-Coloring, Hamiltonian Path...
Square root phenomenon on planar graphs

Max Independent Set, 3-Coloring, Hamiltonian Path...
Square root phenomenon on planar graphs

Max Independent Set, 3-Coloring, Hamiltonian Path...

\[ T(n) \leq 2^{O(\sqrt{n})} T(2n/3) \leq 2^{O(\sqrt{n} \log n)} \]
Square root phenomenon on planar graphs

Max Independent Set, 3-Coloring, Hamiltonian Path...

Dynamic programming would spare a $\log n$ in the exponent.
Square root phenomenon on planar graphs

Max Independent Set, 3-Coloring, Hamiltonian Path...
Square root phenomenon on planar graphs

Max Independent Set, 3-Coloring, Hamiltonian Path...
Bidimensionality

Theorem (Robertson & Seymour, Graph Minors)

A planar graph with treewidth $\geq 5k$ admits a $k \times k$-grid minor.
Bidimensionality

**Theorem (Robertson & Seymour, Graph Minors)**

A planar graph with treewidth $\geq 5k$ admits a $k \times k$-grid minor.

If $tw < 5\sqrt{k} \leadsto$ algorithm in $2^{O(tw)}n^{O(1)} = 2^{O(\sqrt{k})}n^{O(1)}$. 
Bidimensionality

Theorem (Robertson & Seymour, Graph Minors)

A planar graph with treewidth $\geq 5k$ admits a $k \times k$-grid minor.

If $tw < 5\sqrt{k} \leadsto$ algorithm in $2^{O(tw)} n^{O(1)} = 2^{O(\sqrt{k})} n^{O(1)}$.

If $tw \geq 5\sqrt{k} \leadsto \exists \sqrt{k} \times \sqrt{k}$-grid minor $\leadsto$ always yes (always no).
Packing unit disks

Theorem (Alber & Fiala, J. Alg.’04)

Unit disks can be packed in time $n^{O(\sqrt{k})}$. 
Packing unit disks

Theorem (Alber & Fiala, J. Alg.'04)

Unit disks can be packed in time $n^{O(\sqrt{k})}$.

Imagine a solution.
Packing unit disks

Theorem (Alber & Fiala, J. Alg.'04)

Unit disks can be packed in time $n^{O(\sqrt{k})}$.

Translate by one unit $\sqrt{k}$ vertical lines distant by $\sqrt{k}$. 
Packing unit disks

Theorem (Alber & Fiala, J. Alg.’04)

*Unit disks can be packed in time* $n^{O(\sqrt{k})}$.

At some point, the lines intersect it on at most $\sqrt{k}$ disks.
Packing unit disks

Theorem (Alber & Fiala, J. Alg.’04)

Unit disks can be packed in time $n^{O(\sqrt{k})}$.

Guess that intersection in $n^{\sqrt{k}}$. 
Packing unit disks

Theorem (Alber & Fiala, J. Alg.’04)

Unit disks can be packed in time $n^{O(\sqrt{k})}$.

Remove the disks touched by the lines or this intersection.
Packing unit disks

Theorem (Alber & Fiala, J. Alg.’04)

Unit disks can be packed in time $n^{O(\sqrt{k})}$.

Those instances can be solved by DP in $n^{O(\sqrt{k})}$ due to their width.
Packing disks

Idea of Sariel Har-Peled to get geometric QPTAS: use the Voronoi diagram of an assumptive solution.

Theorem (Marx & Pilipczuk, ESA ’15)

Disks can be packed in time $n^{O(\sqrt{k})}$. 
Packing disks

Theorem (Marx & Pilipczuk, ESA ’15)
Disks can be packed in time $n^{O(\sqrt{k})}$.

The Voronoi diagram of a solution is a planar graph with $k$ faces.
Packing disks

Theorem (Marx & Pilipczuk, ESA ’15)

Disks can be packed in time $n^{O(\sqrt{k})}$.

It has a $O(\sqrt{k})$ face-balanced noose. Guess it in $n^{O(\sqrt{k})}$. 
Packing disks

Theorem (Marx & Pilipczuk, ESA ’15)

Disks can be packed in time \(n^{O(\sqrt{k})}\).

Remove the disks touched by it or by the intersected solution.
Theorem (Marx & Pilipczuk, ESA ’15)

Disks can be packed in time $n^{O(\sqrt{k})}$.

Recurse: $T(n, k) \leq n^{O(\sqrt{k})} T(n, 2k/3) \leq n^{O(\sqrt{k})}$. 
Packing disks

Theorem (Marx & Pilipczuk, ESA ’15)

Disks can be packed in time $n^{O(\sqrt{k})}$.

For non unit disks, use distance $d(c, p) := \|c - p\|_2 - r(p)$. 
Covering points with disks

Theorem (Marx & Pilipczuk, ESA ’15)

Selecting $k$ objects among a set of disks covering a set of points can be solved in time $n^{O(\sqrt{k})}$.

With more subtle rules to make the inside and outside independent.
Essentially best algorithms: **Grid Tiling**

**Theorem (Chen et al., CCC ’04)**

$k$-**Clique** cannot be solved in time $f(k)n^{o(k)}$ under ETH.

**Grid Tiling**: *Embed $k$-Clique* in a $k$-by-$k$ grid.  
In each cell select one pair among a prescribed subset of $[n] \times [n]$.  
Two horizontally (vertically) adjacent cells should agree on their first (second) coordinate.
Essentially best algorithms: Grid Tiling

Theorem (Marx, ESA ’05)

Grid Tiling cannot be solved in time $f(k)n^{o(k)}$ under ETH.

Grid Tiling: Embed $k$-Clique in a $k$-by-$k$ grid.
In each cell select one pair among a prescribed subset of $[n] \times [n]$.

Two horizontally (vertically) adjacent cells should agree on their first (second) coordinate.

The choice of vertices are made along the diagonal: $(u_i, u_i)$.
Checking the edge $u_iu_j$ is done in the cell $(i, j)$. 
Essentially best algorithms: **Grid Tiling**

**Theorem (Marx, ESA ’05)**

MIS on UDG cannot be solved in time $f(k)n^{o(\sqrt{k})}$ under ETH.

Syntactical reduction to MIS on UDG.
Smith and Wormald ’98: \( \forall n \) disks with ply \( p \),
Smith and Wormald ’98: $\forall n$ disks with ply $p$, $\exists O$ at most $3n^4$ intersected disks
Actually more general

Theorem (Smith & Wormald, FOCS ’98)

For every $d \geq 1$ and $B \geq 0$, there exists a constant $c = c(d, B)$, such that for every $B$-fat collection $S$ of $n$ $d$-dimensional convex sets with ply at most $\ell$, there exists a $d$-dimensional sphere $Q$, such that:

- at most $\frac{d+1}{d+2} n$ elements of $S$ are entirely inside $Q$,
- at most $\frac{d+1}{d+2} n$ elements of $S$ are entirely outside $Q$,
- at most $cn^{1-1/d} \ell^{1/d}$ elements of $S$ intersect $Q$. 
Simple algorithm for $\ell$-coloring disks

Win-win based on the value of the ply:

$p > \ell \implies \text{answer NO.}$
Simple algorithm for $\ell$-coloring disks

Win-win based on the value of the ply:

$p > \ell \implies$ answer NO.
$p \leq \ell \implies$ balanced separator of size $O(\sqrt{n\ell}) \implies$
treewidth $\tilde{O}(\sqrt{n\ell}) \implies$ coloring in time $2^{\tilde{O}(\sqrt{n\ell})}$
Simple algorithm for $\ell$-coloring disks

Win-win based on the value of the ply:

$p > \ell \Rightarrow$ answer NO.

$p \leq \ell \Rightarrow$ balanced separator of size $O(\sqrt{n\ell}) \Rightarrow$

treewidth $\tilde{O}(\sqrt{n\ell}) \Rightarrow$ coloring in time $2^{\tilde{O}(\sqrt{n\ell})}$

For $\ell$-coloring $d$-dimensional balls, the same argument gives
running time $2^{\tilde{O}(n^{1-1/d} \ell^{1/d})}$. 
Essentially best algorithms: **Grid Coloring**

**Theorem (Biro et al., SoCG ’17)**

For any $\alpha \in [0, 1]$, coloring $n$ unit disks with $\ell = \Theta(n^{\alpha})$ colors cannot be solved in time $2^{o\left(n^{\frac{1+\alpha}{2}}\right)} = 2^{o\left(\sqrt{n\ell}\right)}$, under the ETH.
Essentially best algorithms: **Grid Coloring**

Theorem (Biro et al., SoCG ’17)

*For any* $\alpha \in [0, 1]$, *coloring* $n$ unit disks with $\ell = \Theta(n^\alpha)$ colors *cannot be solved in time* $2^{o(n^{\frac{1+\alpha}{2}})} = 2^{o(\sqrt{n\ell})}$, *under the ETH.*

Constant number of colors $\leadsto$ square root phenomenon.
Linear number of colors $\leadsto$ no subexponential-time algorithm.
Essentially best algorithms: **Grid Coloring**

Theorem (Biro et al., SoCG ’17)

*For any* $\alpha \in [0, 1]$, *coloring* $n$ unit disks with $\ell = \Theta(n^{\alpha})$ colors *cannot be solved in time* $2^{o(n^{\left(\frac{1}{2} + \alpha\right)})} = 2^{o(\sqrt{n\ell})}$, *under the ETH.*

Constant number of colors $\leadsto$ square root phenomenon.
Linear number of colors $\leadsto$ no subexponential-time algorithm.

And everything in between (hard part).
For instance, $\sqrt{n}$-coloring cannot be done in $2^{o(n^{3/4})}$. 
Essentially best algorithms: Grid Coloring

$k$-by-$k$ Grid Tiling instance with $t$ legal pairs in each cell.

Instead of selecting one center in each small grid...
Essentially best algorithms: Grid Coloring

$k$-by-$k$ Grid Tiling instance with $t$ legal pairs in each cell.

...we color them with a different color each...
Essentially best algorithms: Grid Coloring

$k$-by-$k$ Grid Tiling instance with $t$ legal pairs in each cell.

...such that each color class corresponds to a clique
Essentially best algorithms: **Grid Coloring**

$k$-by-$k$ **Grid Tiling** instance with $t$ legal pairs in each cell.

...such that each color class corresponds to a clique

**Grid Coloring** cannot be solved in time $2^{o(tk)}$ under ETH.
Segment intersection graphs

The subexponential algorithm generalizes to other fat objects.
Segment intersection graphs

The subexponential algorithm generalizes to other fat objects.

**Theorem (Rzążewski)**

4-COLORING on 2-DIR cannot be solved in time $2^{o(n)}$.

No subexponential algorithms even for a constant number of colors.
3-COLORING on segments?
**3-COLORING** on segments?

Subexponential algorithm even on string graphs!

String graphs: intersection graphs of curves in the plane.

**Theorem**

(List) **3-COLORING** on strings can be solved in time $2^{O(n^{2/3})}$. 
Subexponential algorithms on string graphs

Important fact: string graphs have separators of size $O(\sqrt{m})$. 

Win-win: high maximum degree: good branching
low maximum degree: small separator

Theorem (Fox & Pach, SODA ’11) Max Independent Set has a subexponential algorithm on string graphs.

What about Min Dominating Set? Min Independent Dominating Set? Max Clique? No, No, No
Subexponential algorithms on string graphs

Important fact: string graphs have separators of size $O(\sqrt{m})$.

Win-win:
- high maximum degree: good branching
- low maximum degree: small separator

Theorem (Fox & Pach, SODA ’11)

**Max Independent Set** has a subexponential algorithm on string graphs.
Subexponential algorithms on string graphs

Important fact: string graphs have separators of size $O(\sqrt{m})$.

Win-win:
  - high maximum degree: good branching
  - low maximum degree: small separator

Theorem (Fox & Pach, SODA ’11)

Max Independent Set has a subexponential algorithm on string graphs.

What about Min Dominating Set? Min Independent Dominating Set? Max Clique?
Subexponential algorithms on string graphs

Important fact: string graphs have separators of size $O(\sqrt{m})$.

Win-win:
- high maximum degree: good branching
- low maximum degree: small separator

Theorem (Fox & Pach, SODA ’11)

Max Independent Set has a subexponential algorithm on string graphs.

What about Min Dominating Set? Min Independent Dominating Set? Max Clique? No, No, No
Art Gallery Problem

Simple polygon with $n$ vertices.
Art Gallery Problem

Simple polygon with $n$ vertices.
Guard the gallery with $k$ points.
Art Gallery Problem

Simple polygon with $n$ vertices.
Guard the gallery with $k$ vertices.
Art Gallery Problem

Simple polygon with $n$ vertices. 
**Dominating Set** in the visibility graph of a simple polygon.

Those three problems are NP-hard.
Art Gallery Problem

Simple polygon with $n$ vertices. **Dominating Set** in the visibility graph of a simple polygon.

Those three problems are NP-hard.

Algorithm in time $f(k)n^{O(\sqrt{k})}$? Algorithm in time $f(k)n^c$?
Intermediate problem and linker

Theorem (B. & Miltzow, ESA ’16)

*No algorithm in time $f(k)n^{o(k/\log k)}$ unless the ETH fails.*

two instances of min hitting set of *intervals*
taking a point in instance A forces a specific point in instance B
Intermediate problem and linker

Theorem (B. & Miltzow, ESA '16)

*No algorithm in time* $f(k)n^{o(k/\log k)}$ *unless the ETH fails.*

two instances of min hitting set of *intervals*
taking a point in instance A forces a specific point in instance B
Terrain Guarding

Guarding an $x$-monotone polygonal curve with $k$ vertices.

**Theorem (Ashok et al. SoCG’17)**

**Terrain Guarding** is solvable in $n^{O(\sqrt{k})}$, hence in $2^{O(\sqrt{n \log n})}$.

A planar graph with domination number $k$ has treewidth $O(\sqrt{k})$.
Terrain Guarding

Guarding an $x$-monotone polygonal curve with $k$ vertices.

**Theorem (Ashok et al. SoCG’17)**

Terrain Guarding is solvable in $n^{O(\sqrt{k})}$, hence in $2^{O(\sqrt{n \log n})}$.

A planar graph with domination number $k$ has treewidth $O(\sqrt{k})$. Non-trivial divide-and-conquer based on this separator.
Subexponential algorithms of geometric graphs

Algorithmic techniques: guessing a small separator relative to a hypothetical solution (Voronoi diagram, planar graph, etc.), separator theorems (for disk graph, string graphs; generalizing the planar separator theorem), win-win approach.
ETH-based lowerbounds: reductions from Grid Tiling, Grid Coloring, 2-Track Hitting Set.

Separator-based techniques also lead to approximation algorithms: Instead of brute-forcing on the separator, ignore it.
Subexponential algorithms of geometric graphs

Algorithmic techniques: guessing a small separator relative to a hypothetical solution (Voronoi diagram, planar graph, etc.), separator theorems (for disk graph, string graphs; generalizing the planar separator theorem), win-win approach.

ETH-based lowerbounds: reductions from Grid Tiling, Grid Coloring, 2-Track Hitting Set.

Separator-based techniques also lead to approximation algorithms: instead of brute-forcing on the separator, ignore it.

Some open questions:

Optimal complexity of MIS, 3-coloring, on string graphs?
Lowerbound or better algorithm for Terrain Guarding?
Subexponential algorithms of geometric graphs

Algorithmic techniques: guessing a small separator relative to a hypothetical solution (Voronoi diagram, planar graph, etc.), separator theorems (for disk graph, string graphs; generalizing the planar separator theorem), win-win approach.

ETH-based lowerbounds: reductions from Grid Tiling, Grid Coloring, 2-Track Hitting Set.

Separator-based techniques also lead to approximation algorithms: Instead of brute-forcing on the separator, ignore it.

Some open questions:
  Optimal complexity of MIS, 3-coloring, on string graphs?
  Lowerbound or better algorithm for Terrain Guarding?

Thanks for your attention!