Approximation algorithm for Diameter, or lack thereof

Édouard Bonnet

ENS Lyon, LIP

July 19th, 2021, LIS seminar
Diameter

\[ \text{diam}(G) = \text{largest distance between a pair of vertices of } G \]

- In weighted graphs, no better known than APSP
- In unweighted graphs, solvable in \( \tilde{O}(n^\omega) \)
Diameter

diam\( (G) \) = largest distance between a pair of vertices of \( G \)

\[
\begin{array}{ccc}
\text{u} & \text{largest}_{u,v} d(u, v) ? & \text{v}
\end{array}
\]

- In weighted graphs, no better known than APSP
- In unweighted graphs, solvable in \( \tilde{O}(n^\omega) \)

**Scope of the talk:** Time vs Approximation trade-offs
Pareto front of \((x, y)\), \( \exists x\)-approximation running in time \( \tilde{O}(|G|^y) \)
Directed (un)weighted Diameter

- **Runtime Exponent**: 
  - $k + 1$ for $k \leq 3/2$
  - $k + 1$ for $k > 3/2$

- **Approximation Factor**: 
  - $3/2$
  - $5/4$
  - $4/3$
  - $8/5$

- **References**: 
  - RV13
  - CLRSTV14
  - Li20
  - Li21
  - BRSVW18
  - DW21
  - B21
Directed (un)weighted Diameter

The diagram illustrates the relationship between runtime exponent and approximation factor for various algorithms. The axes represent:

- Runtime exponent (y-axis): 1, 3/2, 2
- Approximation factor (x-axis): 1, 3/2, 5/3, 2

Key algorithms and their approximation factors include:

- RV13: Approximation factor 1
- CLRSTV14: Approximation factor 3/2
- BRSVW18: Approximation factor 5/3
- Li20: Approximation factor 2

The diagram highlights the performance of these algorithms in terms of their approximation factors and runtime exponents.
Directed (un)weighted Diameter

Approximation factor  
1  3/2  4/3  5/4  6/5  2

Runtime exponent  
1  2  2

RV13  CLRSTV14  BRSVW18  Li20  B21  RV13
Directed (un)weighted Diameter

Directed graphs

- Directed diameter
- Approximation factors
- Runtime exponents

Graphs and Algorithms

- RV13
- CLRSTV14
- Li21
- BRSW18
- Li20
- B21
- DW21

Approximation factor

- 3/2
- 4/3
- 5/4
- 6/5
- k+1/k

Runtime exponent

- 1
- 2

Graphs and Algorithms

- RV13
- CLRSTV14
- Li21
- BRSW18
- Li20
- B21
- DW21
Undirected unweighted **Diameter**

- **Approximation factor**
  - $k + 1/k$
  - $k$
  - $6/5$
  - $5/4$
  - $4/3$
  - $3/2$
  - $2$

- **Runtime exponent**
  - $1$
  - $3/2$
  - $4/3$
  - $5/4$
  - $6/5$

---

**RV13**

**CLRSTV14**

**BRSVW18**

**CGR16**

**Li20**

**Li21**
Undirected unweighted Diameter

$\frac{k+1}{k}$

$\frac{3}{2}$

$\frac{5}{3}$

$\frac{7}{4}$

$2 - \frac{1}{2^{k-1}}$
Undirected unweighted Diameter

- Approximation factor: $\frac{k+1}{k}$
- Runtime exponent: $\frac{2k-1}{k}$

- RV13
- CLRSTV14
- Li20
- B21
- CGR16
- BRSVW18

- RV13
- CLRSTV14
- Li21
Undirected unweighted Diameter

\[ \frac{k+1}{k} \]

\[ \frac{5}{4} \]

\[ \frac{6}{5} \]

\[ \frac{3}{2} \]

\[ \frac{7}{4} \]

\[ 2 - \frac{1}{2k-1} \]

1

approximation factor

\[ \frac{3}{2} \]

\[ \frac{5}{3} \]

\[ \frac{7}{4} \]

\[ 2 - \frac{1}{2k-1} \]

k

runtime exponent

RV13

CLRSTV14

RV13

BRSVW18

Li20

B21

LDV21+

CGR16

Li21
Algorithms
3/2-approximation algorithm in time $\tilde{O}(m\sqrt{n})$

Sample $100\sqrt{n}\log n$ vertices uniformly at random $\rightarrow S$
3/2-approximation algorithm in time $\tilde{O}(m\sqrt{n})$

Run Dijkstra from each vertex of $S$
3/2-approximation algorithm in time $\tilde{O}(m\sqrt{n})$

Let $w$ be the furthest vertex to $S$
3/2-approximation algorithm in time $\tilde{O}(m\sqrt{n})$

Compute $N_{\sqrt{n}}(w)$: the set of $\sqrt{n}$ closest vertices from $w$
3/2-approximation algorithm in time $\tilde{O}(m\sqrt{n})$

Run Dijkstra from each vertex of $N_{\sqrt{n}}(w)$
3/2-approximation algorithm in time $\tilde{O}(m\sqrt{n})$

Output $\max\{\max(d(x, y), d(y, x)) \mid x \in V(G), y \in S \cup N_{\sqrt{n}}(w)\}$
Correctness of the $3/2$ approximation factor

Say $a$ and $b$ realizes the diameter $d = \text{diam}(G) = d(a, b)$
Correctness of the $3/2$ approximation factor

We can assume that $d(a, S) > d/3$. Why?
Correctness of the $3/2$ approximation factor

This implies that $d(w, S) > d/3$. 

\[ N\sqrt{n}(w) \leq d/3 + w(c, c') \geq 2d/3 - w(c, c') \]

\[ w > d/3 \]

\[ a > d/3 \]

\[ b > d/3 \]
Correctness of the $3/2$ approximation factor

Similarly we can assume that $d(w, b) < 2d/3$. 
Correctness of the $3/2$ approximation factor

With high probability $N_{\sqrt{n}}(w)$ intersects $S$
Correctness of the $3/2$ approximation factor

So there is $c \in N_{\sqrt{n}}(w)$ along a shortest path $w - cc' - b$...
Correctness of the $3/2$ approximation factor

\[ N_{\sqrt{n}}(w) \]

...with $d(w, c) \leq d/3$ and $d(w, c') > d/3$
Correctness of the $3/2$ approximation factor

Thus $d(w, c) > d/3 - w(c, c')$, hence $d(c, b) \leq d/3 + w(c, c')$
Correctness of the $3/2$ approximation factor

Finally $d(a, c) \geq 2d/3 - w(c, c')$
Lower bounds
∀k, ∃ε > 0, no classical algorithm solves n-var k-SAT in \((2 - \varepsilon)^n\)

In 1999, Impagliazzo and Paturi introduce ETH and mention a stronger version of it in their conclusion

\[ \text{SETH} \implies \text{ETH} \implies \text{P} \neq \text{NP} \]

ETH and SETH are then mainly used for NP-hard problems.
∀k, ∃ε > 0, no classical algorithm solves \( n\)-var \( k\)-\text{SAT} in \((2 - \varepsilon)^n\)

In 1999, Impagliazzo and Paturi introduce ETH and mention a stronger version of it in their conclusion

\[
\text{SETH} \Rightarrow \text{ETH} \Rightarrow P \neq NP
\]

- ETH and SETH are then mainly used for NP-hard problems
- In 2005, SETH is used for the first time for a problem in P

Orthogonal Vectors,
∀k, ∃ε > 0, no classical algorithm solves \( n \)-var \( k \)-SAT in \((2 − \varepsilon)^n\)

In 1999, Impagliazzo and Paturi introduce ETH and mention a stronger version of it in their conclusion

\[ \text{SETH} \Rightarrow \text{ETH} \Rightarrow \text{P} \neq \text{NP} \]

- ETH and SETH are then mainly used for NP-hard problems
- In 2005, SETH is used for the first time for a problem in P
- 2014-, dozens of papers show SETH-hardness of problems in P

**Orthogonal Vectors, Diameter, Fréchet Distance, Edit Distance, Longest Common Subsequence, Furthest Pair, dynamic problems, problems from Machine Learning, Model Checking, Language Theory etc.**
Reduction from SAT to a problem in P

2-ORTHOGONAL VECTORS (2-OV):
Are there two orthogonal vectors in a given set of $N$ 0, 1-vectors?

\[
\begin{align*}
\text{problem SAT} & \quad \text{instance } \phi \\
\text{instance } & \quad n \text{ variables}
\end{align*}
\]

\[
\begin{align*}
\text{reduction} & \quad \text{time } \leq 1.99^n \\
\text{problem 2-OV} & \quad \text{instance } V \\
N & \quad = 2^{n/2} \text{ vectors}
\end{align*}
\]
Reduction from $\text{SAT}$ to a problem in $\text{P}$

2-Orthogonal Vectors (2-OV):
Are there two orthogonal vectors in a given set of $N$ 0, 1-vectors?

\[
\begin{array}{c}
\text{problem SAT} \\
\text{instance } \phi \\
n \text{ variables}
\end{array} \xrightarrow{\text{reduction}} \begin{array}{c}
\text{problem 2-OV} \\
\text{instance } \mathcal{V} \\
N = 2^{n/2} \text{ vectors}
\end{array}
\]

\[
\text{time } \leq 1.99^n
\]

→ Solving 2-OV in $N^{1.99}$ solves SAT $1.99^n + 2^{1.99n/2}$, refuting SETH
arbitrary equipartition of $X$: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- $A$ of the red variables and
- $B$ of the blue variables

such that all the clauses are satisfied by $A$ or by $B$
**Sat → 2-Orthogonal Vectors [W05]**

arbitrary equipartition of $X$: $x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n$

Find an assignment

- $A$ of the red variables and
- $B$ of the blue variables

such that all the clauses are satisfied by $A$ or by $B$

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$B_1$</td>
<td>$C_1$</td>
<td>$C_2$</td>
<td>$C_3$</td>
<td>$C_4$</td>
<td>$C_5$</td>
<td>$C_6$</td>
<td>$C_7$</td>
</tr>
</tbody>
</table>
Sat $\rightarrow$ 2-Orthogonal Vectors [W05]

arbitrary equipartition of $X$: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- $A$ of the red variables and
- $B$ of the blue variables

such that all the clauses are satisfied by $A$ or by $B$

<table>
<thead>
<tr>
<th></th>
<th>$R$</th>
<th>$B$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
<th>$C_7$</th>
<th>$C_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$A_1$ assigns red variables
**Sat → 2-Orthogonal Vectors [W05]**

arbitrary equipartition of \( X \): \( x_1, x_2, \ldots, \frac{x_n}{2}, \frac{x_n}{2} + 1, \frac{x_n}{2} + 2, \ldots, x_n \)

Find an assignment
- \( A \) of the red variables and
- \( B \) of the blue variables

such that all the clauses are satisfied by \( A \) or by \( B \)

<table>
<thead>
<tr>
<th></th>
<th>( R )</th>
<th>( B )</th>
<th>( C_1 )</th>
<th>( C_2 )</th>
<th>( C_3 )</th>
<th>( C_4 )</th>
<th>( C_5 )</th>
<th>( C_6 )</th>
<th>( C_7 )</th>
<th>( C_8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_1 )</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A_3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( A_4 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( B_1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( B_2 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( B_3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( B_4 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( A_1 \) does not satisfy \( C_1 \)
Sat $\rightarrow$ 2-Orthogonal Vectors [W05]

arbitrary equipartition of $X$: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

$\blacktriangleright$ $A$ of the red variables and

$\blacktriangleright$ $B$ of the blue variables

such that all the clauses are satisfied by $A$ or by $B$

\[
\begin{array}{cccccccccc}
R & B & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \\
A_1 & 1 & 0 & 1 & 0 \\
A_2 &  &  &  &  &  &  &  &  &  \\
A_3 &  &  &  &  &  &  &  &  &  \\
A_4 &  &  &  &  &  &  &  &  &  \\
B_1 &  &  &  &  &  &  &  &  &  \\
B_2 &  &  &  &  &  &  &  &  &  \\
B_3 &  &  &  &  &  &  &  &  &  \\
B_4 &  &  &  &  &  &  &  &  &  \\
\end{array}
\]

$A_1$ satisfies $C_2$
**Sat \rightarrow 2\text{-Orthogonal Vectors} [W05]**

arbitrary equipartition of $X$: $x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n$

Find an assignment
- $A$ of the red variables and
- $B$ of the blue variables

such that all the clauses are satisfied by $A$ or by $B$

<table>
<thead>
<tr>
<th></th>
<th>$R$</th>
<th>$B$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
<th>$C_7$</th>
<th>$C_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$A_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

first vector $(1, 0, 1, 0, 0, 1, 1, 0, 1, 0)$
Sat \rightarrow \textbf{2-Orthogonal Vectors [W05]}

arbitrary equipartition of $X$: $x_1, x_2, \ldots, x_{n/2}, x_{n/2+1}, x_{n/2+2}, \ldots, x_n$

Find an assignment

- $A$ of the red variables and
- $B$ of the blue variables

such that all the clauses are satisfied by $A$ or by $B$

\[
\begin{array}{cccccccccc}
R & B & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 & C_8 \\
A_1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
A_2 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
A_3 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
A_4 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
B_1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\
B_2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
B_3 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
B_4 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
\end{array}
\]
Sat $\rightarrow$ 2-Orthogonal Vectors [W05]

arbitrary equipartition of $X$: $x_1, x_2, \ldots, x_{\frac{n}{2}}, x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$

Find an assignment

- $A$ of the red variables and
- $B$ of the blue variables

such that all the clauses are satisfied by $A$ or by $B$

<table>
<thead>
<tr>
<th></th>
<th>$R$</th>
<th>$B$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
<th>$C_6$</th>
<th>$C_7$</th>
<th>$C_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$A_2$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$A_3$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$A_4$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$B_1$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$B_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$B_3$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$B_4$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Consequence for 2-Orthogonal Vectors

From a SAT-instance on $n$ variables and $m$ clauses, we created $N := 2^{\frac{n}{2} + 1}$ vectors in dimension $d := m + 2$
Consequence for 2-Orthogonal Vectors

From a SAT-instance on $n$ variables and $m$ clauses, we created $N := 2^{\frac{n}{2} + 1}$ vectors in dimension $d := m + 2$

An algorithm solving 2-OV in time $2^{o(d)} N^{2-\varepsilon}$ would solve SAT in $2^{o(m)} 2^{n(1-\varepsilon/2)} \rightarrow$ breaking SETH
Consequence for 2-Orthogonal Vectors

From a SAT-instance on $n$ variables and $m$ clauses, we created $N := 2^{\frac{n}{2}+1}$ vectors in dimension $d := m + 2$.

An algorithm solving 2-OV in time $2^{o(d)} N^{2-\varepsilon}$ would solve SAT in $2^{o(m)} 2^{n(1-\varepsilon/2)} \rightarrow$ breaking SETH.

Most useful consequence here:
$N^{2-o(1)}$-time is required even if $d = \log^{O(1)} N$.
Same for $k$-OV and $N^{k-o(1)}$-time.
So far, all the pairs but of $A \times B$ are at distance $\leq 2$
2-Orthogonal Vectors $\rightarrow$ Diameter [RV13]

We put an edge between vector $v$ and index $i$ iff $v[i] = 1$
A pair \((a_4, b_2)\) is at distance 2 \(\Leftrightarrow \langle a_4, b_2 \rangle \neq 0\)
2-Orthogonal Vectors $\rightarrow$ Diameter [RV13]

$\exists (a_i, b_j)$ at distance 3 $\Leftrightarrow$ orthogonal pair
2-Orthogonal Vectors $\rightarrow$ Diameter [RV13]

If no orthogonal pair, diam($G$) = 2
3 vs 5 undirected Diameter

Theorem (Li ’20)

Approximating sparse undirected unweighted Diameter within factor better than $\frac{5}{3}$ requires time $n^{3/2-o(1)}$, unless SETH fails.
Theorem (Li '20)

Approximating sparse undirected unweighted \textsc{Diameter} within factor better than $\frac{5}{3}$ requires time $n^{3/2-o(1)}$, unless SETH fails.

Plan: hardness of 3 vs 5 \textsc{Diameter} from $N$-vector 3-OV to $O(N^2)$-vertex $\tilde{O}(N^2)$-edge \textsc{Diameter}-instances.
3 vs 5 undirected Diameter

Theorem (Li ’20)

Approximating sparse undirected unweighted Diameter within factor better than $\frac{5}{3}$ requires time $n^{\frac{3}{2}-o(1)}$, unless SETH fails.

Plan: hardness of 3 vs 5 Diameter from $N$-vector 3-OV to $O(N^2)$-vertex $\tilde{O}(N^2)$-edge Diameter-instances.

2 vs 3 hardness from 2-OV [RV13]
3 vs 5 undirected Diameter

Theorem (Li ’20)

Approximating sparse undirected unweighted Diameter within factor better than $\frac{5}{3}$ requires time $n^{\frac{3}{2}-o(1)}$, unless SETH fails.

Plan: hardness of 3 vs 5 Diameter from $N$-vector 3-OV to $O(N^2)$-vertex $\tilde{O}(N^2)$-edge Diameter-instances.

2 vs 3 hardness from 2-OV [RV13]

Vectors $a, b, c, \ldots$, Indices $i, j, k, \ldots$, 
ind($a, b, c$) = $i, with $a[i] = b[i] = c[i] = 1$ (exists if $a, b, c$ not $\perp$) 
index $i$ contradicts $a, b, c \perp$
3-Orthogonal Vectors $\rightarrow$ 3 vs 5 Diameter [Li20]

\[
\begin{array}{ccc}
S & & (c, i, j) \\
a[i] = a[j] = 1 & (a, i, j) & (a, i', j') \\
\end{array}
\]

\[
\begin{array}{ccc}
\circ & & \circ \\
(a, b) & P & (c, b) \\
\circ & & (c, d) \\
\end{array}
\]
3-Orthogonal Vectors $\rightarrow$ 3 vs 5 Diameter [Li20]

$S$

$a[i] = a[j] = 1$
$(a, i, j)$

$(c, i, j)$

$(a, i', j')$

$b[i] = 1$ or $b[j] = 1$

$(a, b)$

$(c, b)$

$(c, d)$

$P$
$3$-Orthogonal Vectors $\rightarrow 3$ vs $5$ Diameter $[\text{Li20}]$

$S$

$a[i] = a[j] = 1$

$(a, i, j)$

$(c, i, j)$

$(c, i', j')$

$b[i] = 1$ or $b[j] = 1$

$(a, b)$

$P$

$(c, b)$

$(c, d)$
$$\begin{align*}
S & \quad (c, i, j) \\
a[i] = a[j] = 1 & \quad (a, i, j) \\
(b[i] = 1 \text{ or } b[j] = 1) & \quad (a, i', j') \\
(a, b) & \quad P \quad (c, b) \quad (c, d)
\end{align*}$$
No orthogonal triple $\Rightarrow$ diam$(G) = 3,$

$i = \text{ind}(a, b, c), \ j = \text{ind}(a, c, d)$
3-Orthogonal Vectors $\rightarrow$ 3 vs 5 Diameter [Li20]

Orthogonal triple $(a,b,c) \Rightarrow d(((a, b), (b, c)) = 5$, $(a, b) - (a, i, j) - (x, i', j') - (c, i'', j'') - (c, b)$
Orthogonal triple \((a, b, c) \Rightarrow d(((a, b), (b, c))) = 5,
(a, b) - (a, i, j) - (a, i', j') - (c, i'', j'') - (c, b)\)
Orthogonal triple \((a,b,c)\) \(\Rightarrow d(((a, b), (b, c)) = 5, (a, b) - (a, i, j) - (a, i', j') - (c, i', j') - (c, b))\)
Orthogonal triple \((a, b, c)\) \(\Rightarrow\) \(d((a, b), (b, c)) = 5\),
i' or j' contradicts \(a, b, c \perp\)
Theorem (B. ’21)

Approximating sparse undirected unweighted Diameter within factor better than $\frac{7}{4}$ requires time $n^{\frac{4}{3} - o(1)}$, unless SETH fails.
4 vs 7 undirected Diameter

Theorem (B. ’21)

Approximating sparse undirected unweighted \textsc{Diameter} within factor better than $\frac{7}{4}$ requires time $n^{\frac{4}{3} - o(1)}$, unless SETH fails.

Plan: hardness of 4 vs 7 \textsc{Diameter} from $N$-vector 4-OV to $O(N^3)$-vertex $\tilde{O}(N^3)$-edge \textsc{Diameter}-instances.
4 vs 7 undirected Diameter

Theorem (B. ’21)

Approximating sparse undirected unweighted Diameter within factor better than $\frac{7}{4}$ requires time $n^{\frac{4}{3} - o(1)}$, unless SETH fails.

Plan: hardness of 4 vs 7 Diameter from $N$-vector 4-OV to $O(N^3)$-vertex $\tilde{O}(N^3)$-edge Diameter-instances.

2 vs 3 hardness from 2-OV [RV13]
3 vs 5 hardness from 3-OV [Li20]
Construction with weights

\[ d[i] = d[j] = d[k] = e[i] = e[j] = e[k] = 1 \]
\[ \{ d, e \}, i, j, k \]  \( \bigcirc \)  \( P \)

\[ a[i] = a[j] = a[k] = 1 \]
\[ (a, b, i, j, k) \]  \( \bigcirc \)  \( \bigcirc \)  \( C \)
\[ \text{maj}(b[i], b[j], b[k]) = 1 \]
\[ (a, b, i', j', k') \]

\[ (a, b, c) \]  \( \bigcirc \)  \( T \)
Construction with weights

\[ d[i] = d[j] = d[k] = e[i] = e[j] = e[k] = 1 \]
\[ \{d, e\}, i, j, k \]
\[ P \]

\[ a[i] = a[j] = a[k] = 1 \]
\[ (a, b, i, j, k) \]
\[ C \]
\[ \text{maj}(b[i], b[j], b[k]) = 1 \]
\[ (a, b, i', j', k') \]

\[ T \]
\[ (a, b, c) \]
Construction with weights

\[
d[i] = d[j] = d[k] = e[i] = e[j] = e[k] = 1^{(2,3)}
\]

\[
\{d, e\}, i, j, k)
\]

\[
\exists h \in \{i, j, k\},
\]

\[
c[h] = b[h] = 1
\]

\[
(a, b, c)
\]
Construction with weights

\[ d[i] = d[j] = d[k] = e[i] = e[j] = e[k] = 1 \]
\[ \{d, e\}, i, j, k \]  
\[ p_1, p_2, i, j, k \]

\[ a[i] = a[j] = a[k] = 1 \]
\[ \{a, b, i, j, k\} \]
\[ \text{maj}(b[i], b[j], b[k]) = 1 \]
\[ (a, b, i', j', k') \]
\[ \exists h \in \{i, j, k\}, \]
\[ c[h] = b[h] = 1 \]

\[ (a, b, c) \]
Construction with weights

\[ d[i] = d[j] = d[k] = e[i] = e[j] = e[k] = 1 \]
\[ \{d, e\}, i, j, k \]
\[ P \]

\[ a \in \{d, e\} \]

\[ a[i] = a[j] = a[k] = 1 \]
\[ (a, b, i, j, k) \]
\[ \text{maj}(b[i], b[j], b[k]) = 1 \]
\[ (a, b, i', j', k') \]
\[ C \]

\[ a[p_1] = b[p_1] = c[p_1] = 1 \]
\[ \exists h \in \{i, j, k\}, \]
\[ a[p_2] = b[p_2] = c[p_2] = 1 \]
\[ c[h] = b[h] = 1 \]

\[ (a, b, c) \]
\[ T \]
Construction with weights

\[ d[i] = d[j] = d[k] = e[i] = e[j] = e[k] = 1 \]
\[ \{d, e\}, i, j, k \]

\[ d[p_1] = e[p_2] = 1 \quad \text{or} \quad d[p_2] = e[p_1] = 1 \]
\[ a \in \{d, e\} \]

\[ a[i] = a[j] = a[k] = 1 \]
\[ (a, b, i, j, k) \]
\[ \text{maj}(b[i], b[j], b[k]) = 1 \]
\[ (a, b, i', j', k') \]

\[ a[p_1] = b[p_1] = c[p_1] = \quad \exists h \in \{i, j, k\}, \]
\[ a[p_2] = b[p_2] = c[p_2] = 1 \quad c[h] = b[h] = 1 \]

\[ (a, b, c) \]
Construction with weights

\[ \text{d}[i] = \text{d}[j] = \text{d}[k] = \text{e}[i] = \text{e}[j] = \text{e}[k] = 1 \]

\( \{d, e\}, i, j, k \)

\[ \text{d}[p_1] = \text{e}[p_2] = 1 \quad \text{or} \quad \text{d}[p_2] = \text{e}[p_1] = 1 \]

\( a \in \{d, e\} \)

\[ \text{a}[i] = \text{a}[j] = \text{a}[k] = 1 \]

\( \{a, b, i, j, k\} \)

\[ \text{maj}(\text{b}[i], \text{b}[j], \text{b}[k]) = 1 \]

\( \{a, b, i', j', k'\} \)

\[ \exists h \in \{i, j, k\}, \quad \text{c}[p_1] = \text{a}[p_1] = \text{b}[p_1] = \text{c}[p_1] = \]

\[ \exists h \in \{i, j, k\}, \quad \text{c}[p_2] = \text{a}[p_2] = \text{b}[p_2] = \text{c}[p_2] = 1 \]

\[ \text{c}[h] = \text{b}[h] = 1 \]

\( \{a, b, c\} \)

\( (a, b, c) \)

\( (p_1, p_2, i, j, k) \)

\( (p'_1, p'_2, i', j', k') \)
Construction with weights

\[\begin{align*}
&\{d, e\}, i, j, k
\end{align*}\]
No orthogonal quadruple $\Rightarrow$ diameter at most 4

Automatic paths of length at most 4, except for T-T, T-C, T-P, and C-C
No orthogonal quadruple $\Rightarrow$ diameter at most 4

$\text{T-T, T-C, C-C: } (a, b, c) \text{ or } (a, b, i', j', k') - (a, b, i, j, k) - (\{a, d\}, i, j, k) - (d, e, i, j, k) - (d, e, f) \text{ or } (d, e, i'', j'', k'')$

with $i = \text{ind}(a, b, c, d), j = \text{ind}(a, b, d, e), k = \text{ind}(a, d, e, f)$
No orthogonal quadruple $\implies$ diameter at most 4

$\{a, b, c, d, e\}$

$T-P: (a, b, c) - (p_1, p_2, i, j, k) - (\{d, e\}, i, j, k)$

with $p_1 = \text{ind}(a, b, c, d), p_2 = \text{ind}(a, b, c, e)$
\( a, b, c, d \) orthogonal \( \Rightarrow d((a, b, c), (d, c, b)) \geq 7 \)

Set \( I \) cannot help for a path of length 6
$a, b, c, d$ orthogonal $\Rightarrow d((a, b, c), (d, c, b)) \geq 7$

$(a, b, i, j, k)$ and $(d, c, i, j, k)$ have to be part of the path
Removing the weights

\[ d[i] = d[j] = d[k] = e[i] = e[j] = e[k] = 1 \]
\[ \{d, e\}, i, j, k \]

\[ d[p_1] = e[p_2] = 1 \]
\[ a \in \{d, e\} \]

\[ a[i] = a[j] = a[k] = 1 \]
\[ (a, b, i, j, k) \]
\[ \text{maj}(b[i], b[j], b[k]) = 1 \]
\[ (a, b, i', j', k') \]

\[ a[p_1] = b[p_1] = c[p_1] = 1 \]
\[ a[p_2] = b[p_2] = c[p_2] = 1 \]
\[ \exists h \in \{i, j, k\}, \]
\[ c[h] = b[h] = 1 \]

\[ a[i] = b[i] = 1 \]
\[ (a, b, c)_{T''} \]

\[ (a, b, c)_{T'} \]

\[ (a, b, c)_T \]
Thank you for your attention!