Motivation

• So far, mostly happy with polynomial-time algorithms
• Obstacle encountered: NP-hardness
  • In that case, considered approximation and beyond worst-case analysis

• Polynomial time doesn’t necessarily mean practical
• On large inputs/datasets, practical may mean near-linear time

• Given a problem in P, how fast can it be solved?
Example: Edit Distance

• How many insertions/deletions/substitutions to get from string \( x \) to \( y \)?

\[
\begin{align*}
\text{G A C T A A} \\
\text{A C T A A G}
\end{align*}
\]

• Classic \( O(n^2) \) dynamic programming algorithm

• Important application: aligning genetic sequences
  • With such large \( n \), \( O(n^2) \) is prohibitive
Can we get a faster algorithm for edit distance?
  - In particular, can it be solved in $O(n^{2-\epsilon})$ time?
  - Best known exact algorithm takes $O(n^2 / \log^2 n)$ (from 1980!)

If not, can we explain the “hardness”?
  - More fine-grained perspective than NP-hardness is needed

Explore relations between problems in P
Relations Between Problems in P

• Given two problems A and B:
  A solvable in time \(a(n)\) (e.g., edit distance with \(a(n) = O(n^2)\)),
  B solvable in time \(b(n)\)

• Goal is to show:
  A solvable in time \((a(n))^{1-\varepsilon}\) \(\implies\) B solvable in time \((b(n))^{1-\varepsilon'}\)

• That is, improvement for A implies improvement for B
Relations Between Problems in P

• Main tool: fine-grained reductions
  • Start from conjecture on the time needed to solve a “hard” problem B, e.g., B cannot be solved in time \( (b(n))^{1-\varepsilon'} \)
  • Reduce B to A to show:
    A solvable in time \( (a(n))^{1-\varepsilon} \) \( \Rightarrow \) B solvable in time \( (b(n))^{1-\varepsilon'} \)
  • We get a conditional lower bound for A:
    Any faster algorithm for A will refute the conjecture on B

• Conceptually, mimic NP-hardness
One Popular Conjecture

• **CNF-SAT**
  Given a CNF formula with $n$ variables and $m$ clauses, find a satisfying assignment
  
  • $(x_1 \lor x_2) \land \overline{x_2} \lor x_5 \lor x_{10}$

• **Strong Exponential Time Hypothesis (SETH)**
  For any $\varepsilon > 0$, CNF-SAT cannot be solved in time $O(2^{(1-\varepsilon)n})$
  
  • Formally, for every $\varepsilon > 0$, $\exists k \geq 3$ such that $k$-SAT with $n$ variables cannot be solved in time $2^{(1-\varepsilon)n}\text{poly}(n)$
Outline

• Introduce the Orthogonal Vectors (OV) problem
• SETH implies OV cannot be solved in $n^{2-\varepsilon}\text{poly}(d)$
• Reduction from OV to graph diameter:
  $(3/2 - \delta)$-approximation on sparse graphs not solvable in $O(n^{2-\varepsilon})$ time
Orthogonal Vectors

• One of the first cases where SETH was used to show hardness of problems in P

• Orthogonal Vectors
  Given two sets $A, B \subseteq \{0,1\}^d$ such that $|A| = |B| = n$, find if there is $a \in A, b \in B$ satisfying $a \cdot b = \sum_{i=1}^{d} a_i b_i = 0$

• Note that we sum over $\mathbb{R}$

• Naive algorithm: $O(n^2d)$

• Hypothesis: not solvable in time $n^{2-\varepsilon}\text{poly}(d)$

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,1,1,1)</td>
<td>(0,0,0,1)</td>
</tr>
<tr>
<td>(0,0,0,1)</td>
<td>(1,0,0,1)</td>
</tr>
<tr>
<td>(0,1,0,1)</td>
<td>(0,1,0,0)</td>
</tr>
<tr>
<td>(1,1,0,0)</td>
<td>(1,1,0,0)</td>
</tr>
<tr>
<td>(1,1,1,1)</td>
<td>(1,1,0,1)</td>
</tr>
</tbody>
</table>
Reducing CNF-SAT to OV

- **Theorem**: If OV on sets of $N$ vectors in $\{0,1\}^m$ solvable in $N^{2-\varepsilon} \text{poly}(m)$, then CNF-SAT on $n$ variables and $m$ clauses solvable in $2^{(1-\varepsilon')n} \text{poly}(m)$.

- **Proof**: Reduction from CNF-SAT to OV

  - For simplicity, assume $n$ is even

  - Define the OV instance:
    - $A$ corresponds to possible assignments to first $n/2$ variables,
    - $B$ to the last $n/2$ variables
Reducing CNF-SAT to OV

- $A$ includes a vector for each possible assignment $\phi$ to $x_1, \ldots, x_{n/2}$:
  \[ A_{\phi}[i] = \begin{cases} 0 & \text{clause } i \text{ satisfied under } \phi \\ 1 & \text{otherwise} \end{cases} \]
  ($2^{n/2}$ vectors of length $m$)

- $B$ defined similarly

- **Claim:** CNF-SAT instance satisfiable if and only if $A, B$ contain orthogonal vectors
Reducing CNF-SAT to OV

\[ A_\phi[i] = \begin{cases} 
0 & \text{clause } i \text{ satisfied under } \phi \\
1 & \text{otherwise}
\end{cases} \]

- CNF-SAT instance satisfiable \( \implies \) \( A, B \) contain orthogonal vectors:
- Let \( \phi \) be the satisfying assignment. Consider \( A_\phi \in A, B_\phi \in B \).
- At least one literal in clause \( i \) is true under \( \phi \).
  - If the literal is one of \( x_1, \ldots, x_{n/2} \) (or negation), \( A_\phi[i] = 0 \).
  - Otherwise, \( B_\phi[i] = 0 \).
- Then, \( A_\phi \cdot B_\phi = \sum_{i=1}^{m} A_\phi[i] \cdot B_\phi[i] = \sum_{i=1}^{m} 0 = 0 \).
Reducing CNF-SAT to OV

\[ A_\phi[i] = \begin{cases} 
0 & \text{clause } i \text{ satisfied under } \phi \\
1 & \text{otherwise} 
\end{cases} \]

- CNF-SAT instance satisfiable \( \iff \) \( A, B \) contain orthogonal vectors:
  - Assume \( A_{\phi_1} \cdot B_{\phi_2} = 0 \) and define \( \phi \) as the concatenation of \( \phi_1, \phi_2 \)
  - For each clause \( i \), either \( A_{\phi_1}[i] = 0 \) or \( B_{\phi_2}[i] = 0 \)
  - Then \( i \) must be satisfied under \( \phi \)
  - Therefore, \( \phi \) satisfies all the clauses
Reducing CNF-SAT to OV

- We proceed to analyze the run time
- Started from CNF-SAT instance with $n$ variables and $m$ clauses
- Created OV instance with $N = 2^{n/2}$ vectors per set, dimension $m$
  - Creating the instance takes at most $O(Nm)$ time
- If OV solvable in $N^{2-\epsilon} \text{poly}(m)$, we can solve CNF-SAT in time
  $$(2^{n/2})^{2-\epsilon} \text{poly}(m) = 2^{(1-\epsilon/2)n} \text{poly}(m)$$
- This proves the theorem for $\epsilon' = \epsilon/2$
- Alternatively, SETH implies the OV hypothesis
Graph Diameter

• Use OV to get conditional lower bound for a graph problem

• **Definition:** Given an undirected $G = (V, E)$, the diameter of $G$ is $\max_{u,v \in V} d(u, v)$, i.e., the maximum distance between two vertices $u, v \in V$

• On unweighted graphs, can be solved in time $O(n^\omega \text{polylog}(n))$
  • $2 \leq \omega < 2.373$ is the matrix multiplication exponent

• All known algorithms take $\Omega(n^2)$, even when $m = O(n)$
  • When $m = O(n)$, run BFS from each node to get total time $O(n^2)$
Graph Diameter

• Let $D$ denote the diameter of $G$. A $c$-approximation algorithm returns $D'$ such that \( \frac{D}{c} \leq D' \leq D \).

• $3/2$-approximation in $\tilde{O}\left(m^{3/2}\right)$ time
  • For sparse graphs, improvement over $O(n^2)$

• Can we hope to improve the run time for solving diameter exactly?
Graph Diameter

• We show that under the OV hypothesis, any \((3/2 - \varepsilon)\)-approximation for diameter is not solvable in time \(O(n^{2-\varepsilon'})\)
  • Even for unweighted, sparse graphs
  • Also implied by SETH (since SETH implies OV hypothesis)

• How do we prove hardness of approximation?
• We show it is hard to distinguish between diameter 2 and 3
  • Any \((3/2 - \varepsilon)\)-approximation returns value > 2 for input with diameter 3, and value \(\leq 2\) for input with diameter 2
Reducing OV to Diameter

- **Theorem:** Given an unweighted $G$ with $O(N)$ nodes and edges, if one can distinguish between diameter 2 and 3 in time $O(N^{2-\varepsilon})$, then OV on sets of $n$ vectors in $d$ dimensions is solvable in time $n^{2-\varepsilon}\text{poly}(d)$
  - Even on sparse, unweighted graphs
  - Also implied by SETH (since SETH implies the OV hypothesis)
- We prove by reducing OV to diameter
- Need to design an instance where
  - If there is an orthogonal pair, diameter is 3
  - Otherwise, diameter is 2

\[
\text{CNF-SAT} \rightarrow \text{OV} \rightarrow \text{Diameter}
\]
Reducing OV to Diameter

• Let $A, B \in \{0,1\}^d$ where $|A| = |B| = n$

• The nodes of $G$ are:
  • $a$ for every $a \in A$, $b$ for every $b \in B$
  • $c_1, \ldots, c_d$ and $x, y$

• The edges:
  • If $a[i] = 1$, add $(a, c_i)$,
    if $b[i] = 1$, add $(b, c_i)$
  • $(x, a), (x, c_i)$ for all $a, i$
  • $(y, b), (y, c_i)$ for all $b, i$
  • $(x, y)$

Figure based on survey by Virginia Vassilevska Williams
Reducing OV to Diameter

- If there is no orthogonal pair: 
  \( d(a, b) = 2 \) for any \( a \in A, b \in B \)
  The distance between all other nodes is at most 2, so the diameter is 2.

- If there is an orthogonal pair \( a, b \): 
  Any path from \( a \) to \( b \) must go through \( x \) or \( y \).
  Then \( d(a, b) = 3 \), and so is the diameter.

Figure based on survey by Virginia Vassilevska Williams
Reducing OV to Diameter

- Let $N = nd$
- Constructing the graph takes $O(N)$
- The graph has $O(N)$ nodes and edges
- If we can distinguish between diam. 2 or 3 in $O(N^{2-\varepsilon})$, we can solve OV in $O(N^{2-\varepsilon}) = n^{2-\varepsilon} \text{poly}(d)$

Figure based on survey by Virginia Vassilevska Williams
Other Popular Conjectures (Informal)

- **3-SUM**: Given set $S$ of $n$ integers in $\{-n^c, \ldots, n^c\}$, check if there are $x, y, z \in S$ such that $x + y + z = 0$
- **3-SUM Hypothesis**: 3-SUM cannot be solved in $O(n^{2-\epsilon})$

- **All-Pairs Shortest Path (APSP)**: compute the distance (shortest path) between all pairs of nodes in an integer-weighted graph with no negative cycles
- **APSP Hypothesis**: APSP cannot be solved in $O(n^{3-\epsilon})$

- Applied to many problems: edit distance, LCS, listing triangles, dynamic data structures, stable matching, radius...