

**Deterministic APSP,  
Orthogonal Vectors, and More:  
Quickly Derandomizing Razborov–Smolensky**

*Timothy Chan*

U of Waterloo

Ryan Williams

Stanford

# Problem 1: All-Pairs Shortest Paths (APSP)

Given real-weighted graph with  $n$  vertices, compute the shortest path between every pair of vertices

- textbook:  $O(n^3)$  time
- **Big Open Q:**  $O(n^{3-\varepsilon})??$
- importance: many applications, & various APSP-hardness results...

# History of APSP Alg'ms

Fredman'75	$n^3(\log \log n / \log n)^{1/3}$
Takaoka'92	$n^3 \sqrt{\log \log n / \log n}$
Dobosiewicz'90	$n^3 / \sqrt{\log n}$
Han'04	$n^3(\log \log n / \log n)^{5/7}$
Takaoka'04	$n^3 \log^2 \log n / \log n$
Zwick'04	$n^3 \sqrt{\log \log n / \log n}$
Chan'05	$n^3 / \log n$
Han'06	$n^3(\log \log n / \log n)^{5/4}$
Chan'07	$n^3(\log \log n)^3 / \log^2 n$
Han–Takaoka'12	$n^3 \log \log n / \log^2 n$

Williams [STOC'14]

$$n^3 / 2^{\Omega(\sqrt{\log n})}$$

rand. (Monte Carlo)

new

$$n^3 / 2^{\Omega(\sqrt{\log n})}$$

det.

# Problem 2: Orthogonal Vectors (OV)

Given  $n$  Boolean vectors in  $d$  dims, is there a pair with dot product = 0?

- trivial:  $O(dn^2)$  time
- improvable by rect. matrix multiplication (e.g.,  $O(n^{2+\varepsilon})$  for  $d \ll n^{0.30}$ , or  $\tilde{O}(n^2)$  for  $d \ll n^{0.17}$ )
- trivial:  $O(2^d)$  time
- **Big Open Q:**  $O(n^{2-\varepsilon})$  alg'm for some  $d = \omega(\log n)$ ??
- importance: applications e.g. to  $k$ -SAT (OV is SETH-hard), & numerous OV-hardness results . . .

# History of OV Alg'ms

For  $d = c \log n$ :

Impagliazzo–Lovett–  
Paturi–Schneider'14  $n^{2-1/c^{O(1)}}$

Abboud–Williams–Yu  
[SODA'15]  $n^{2-1/O(\log c)}$

rand. (Monte Carlo)

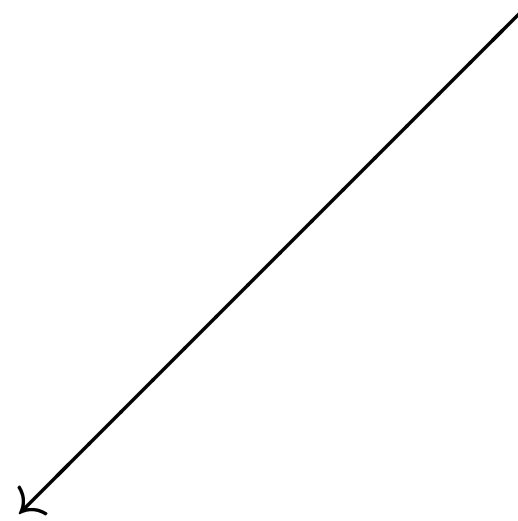
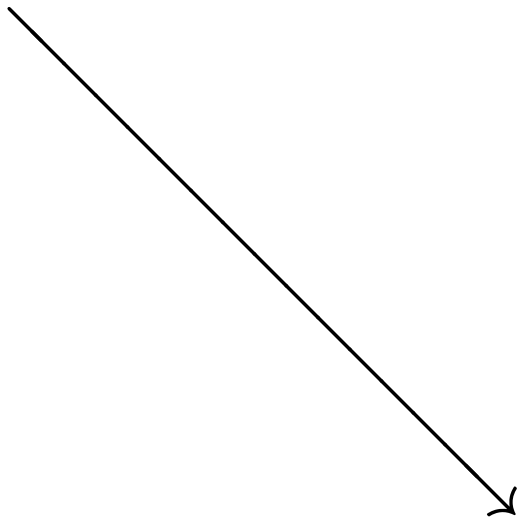
new  $n^{2-1/O(\log c)}$

det.

# Plan

APSP

OV



Rect. Matrix Multiplication (RMM)

# Plan

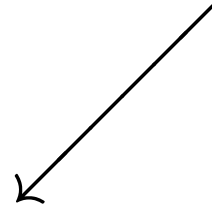
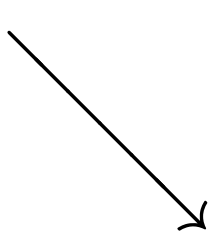
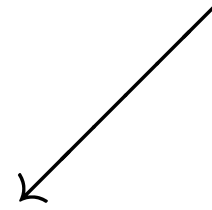
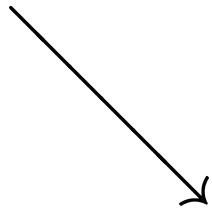
APSP

OV

“Funny” RMM

“Funny” RMM

Standard RMM



# Standard RMM

Given vectors  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}, \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}$  in  $d$  dims, compute

$$\begin{pmatrix} \text{---} \mathbf{x}^{(1)} \text{---} \\ \vdots \\ \text{---} \mathbf{x}^{(n)} \text{---} \end{pmatrix} \begin{pmatrix} | & & | \\ \mathbf{y}^{(1)} & \dots & \mathbf{y}^{(n)} \\ | & & | \end{pmatrix} = \begin{pmatrix} \text{all dot products} \\ \mathbf{x}^{(s)} \cdot \mathbf{y}^{(t)} \end{pmatrix}$$

- $\tilde{O}(n^2)$  time for  $d \ll n^{0.17}$  [Coppersmith'82]



# “Funny” RMM

Given vectors  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}, \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(n)}$  in  $d$  dims, compute

$$\begin{pmatrix} \text{---} \mathbf{x}^{(1)} \text{---} \\ \vdots \\ \text{---} \mathbf{x}^{(n)} \text{---} \end{pmatrix} \begin{pmatrix} | & & | \\ \mathbf{y}^{(1)} & \dots & \mathbf{y}^{(n)} \\ | & & | \end{pmatrix} = \begin{pmatrix} \text{all values} \\ f(\mathbf{x}^{(s)}, \mathbf{y}^{(t)}) \end{pmatrix}$$

for some “funny” function  $f(\mathbf{x}, \mathbf{y})$

- $\tilde{O}(n^2)$  time for what  $d$ ??

# APSP...

- is known to be reducible to “funny” RMM with

$$f(\mathbf{x}, \mathbf{y}) = \min_{k=1}^d (x_k + y_k) \quad (\mathbf{x}, \mathbf{y} \in \mathbb{R}^d)$$

called **min-plus RMM**

- by known techniques (Fredman’s subtraction trick, Matoušek’s dominance method, ...), can be further reduced to a “less funny” RMM with

$$f(\mathbf{x}, \mathbf{y}) = \bigwedge_{i=1}^{d'} \bigvee_{j=1}^{d''} (x_{ij} \wedge y_{ij}) \quad (\mathbf{x}, \mathbf{y} \in \{0, 1\}^{d' d''})$$

which I’ll call **AND-OR-AND RMM** ( $d', d'' = \text{poly}(d)$ )

(Williams’ original alg’m used XOR-AND-OR-AND RMM...)

# OV...

- by dividing into  $n/g$  groups of  $g$  vectors, can be reduced to “funny” RMM with

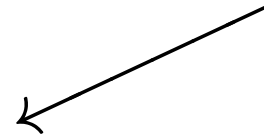
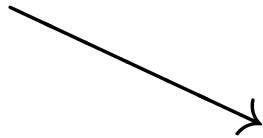
$$f(\mathbf{x}, \mathbf{y}) = \bigwedge_{i=1}^g \bigwedge_{j=1}^g \bigvee_{k=1}^d (x_{ik} \wedge y_{jk}) \quad (\mathbf{x}, \mathbf{y} \in \{0, 1\}^{dg})$$

which is again **AND-OR-AND RMM!** ( $d' = g^2, d'' = d$ )

# Plan

APSP

OV



AND-OR-AND RMM



??

Standard RMM

# Solving AND-OR-AND RMM: The “Polynomial Method”

- rewrite  $f(\mathbf{x}, \mathbf{y})$  as a multivariate polynomial
- problem is then reduced to standard RMM where  
new dim. = # monomials of  $f$
- **Ex:**  $f(\mathbf{x}, \mathbf{y}) = x_1y_2 + 4x_2y_1y_2 + 3x_1x_2y_1$   
 $= (x_1, 4x_2, 3x_1x_2) \cdot (y_2, y_1y_2, y_1)$

# Solving AND-OR-AND RMM: The “Polynomial Method”

- **New Problem:** rewrite

$$\text{AND-OR-AND}(\mathbf{x}, \mathbf{y}) = \bigwedge_{i=1}^{d'} \bigvee_{j=1}^{d''} (x_{ij} \wedge y_{ij})$$

as a polynomial with small # monomials

# Solving AND-OR-AND RMM: The “Polynomial Method”

- **New Problem:** rewrite

$$\text{AND-OR}(z) = \bigwedge_{i=1}^{d'} \bigvee_{j=1}^{d''} z_{ij}$$

as a polynomial with small # monomials

- aim for small degree...

# Razborov–Smolensky's OR Trick

Q: how to write a polynomial for  $\bigvee_{j=1}^{d''} z_j$ ?

Easy Sol'n:  $1 - \prod_{j=1}^{d''} (1 - z_j)$  (but degree =  $d''$ , too big!)



# Razborov–Smolensky's OR Trick

Q: how to write a polynomial for  $\bigvee_{j=1}^{d''} z_j$ ?

Randomized Sol'n:

- take random vector  $r \in \{0, 1\}^{d''}$
- return  $\sum_{j=1}^{d''} r_j z_j \pmod{2}$

Analysis: degree = 1!

- false  $\Rightarrow$  always correct
- true  $\Rightarrow$  error prob. =  $1/2$
- can lower error prob. to  $1/d'$  by **repeating**  $\approx \log d'$  times & taking product  $\Rightarrow$  degree  $\approx \log d'$

# Randomized AND-OR Polynomial

- just apply Razborov–Smolensky twice  
(for the top AND, use de Morgan & const error prob.)
- degree  $\approx \log d'$
- # monomials  $\approx d' \cdot \binom{d''}{\log d'}$   
 $\leq O\left(\frac{d''}{\log d'}\right)^{\log d'}$   
 $= 2^{O(\log^2 d)}$  for APSP ( $d', d'' = \text{poly}(d)$ )  
 $\ll n^{0.17}$

by setting  $d = 2^{\Theta(\sqrt{\log n})}$

# Derandomizing Razborov–Smolensky

- $2^{d''}$  choices for random vector  $\mathbf{r}$
- **Idea 1:** use a smaller sample space that behaves roughly the same

⇒  **$\varepsilon$ -biased sets**

[Naor–Naor'93, Alon–Goldreich–Hastad–Peralta'92, ...]

**Fact 1:**  $\exists$  set  $\mathcal{R}_\varepsilon$  of  $\text{poly}(d'', 1/\varepsilon)$  vectors s.t.

if  $\bigvee_{j=1}^{d''} z_j$  is true, then

$$\Pr_{\mathbf{r} \in \mathcal{R}_\varepsilon} \left[ \sum_{j=1}^{d''} r_j z_j \equiv 1 \pmod{2} \right] \in (1/2 - \varepsilon, 1/2 + \varepsilon)$$

# Derandomizing Razborov–Smolensky

- now try all random choices  $r \in \mathcal{R}_\epsilon \dots$   
but can't tally the results when working mod 2!
- **Idea 2:** use a larger modulus  
 $\Rightarrow$  **modulus-amplifying polynomials**  
[Yao'90, Beigel–Tarui'94, on ACC circuits, ...]

**Fact 2:**  $\exists$  polynomial  $F_\ell$  of degree  $O(\ell)$  s.t.

$$x \equiv 0 \pmod{2} \Rightarrow F_\ell(x) \equiv 0 \pmod{2^\ell}$$

$$x \equiv 1 \pmod{2} \Rightarrow F_\ell(x) \equiv 1 \pmod{2^\ell}$$

# Final Deterministic AND-OR Polynomial

Rewrite AND-OR( $\mathbf{z}$ ) =  $\bigwedge_{i=1}^{d'} \bigvee_{j=1}^{d''} z_{ij}$  as

$$P(\mathbf{z}) := \sum_{i=1}^{d'} \sum_{r \in \mathcal{R}_\varepsilon} F_\ell \left( \sum_{j=1}^{d''} r_j z_{ij} \right)$$

# Final Deterministic AND-OR Polynomial

Rewrite  $\text{AND-OR}(\mathbf{z}) = \bigwedge_{i=1}^{d'} \bigvee_{j=1}^{d''} z_{ij}$  as

$$P(\mathbf{z}) := \sum_{i=1}^{d'} \sum_{\mathbf{r} \in \mathcal{R}_\varepsilon} F_\ell \left( \sum_{j=1}^{d''} r_j z_{ij} \right)$$

- true  $\Rightarrow P(\mathbf{z}) \approx d' |\mathcal{R}_\varepsilon| (1/2 \pm \varepsilon) \pmod{2^\ell}$
- false  $\Rightarrow P(\mathbf{z}) \leq (d' - 1) |\mathcal{R}_\varepsilon| (1/2 \pm \varepsilon) \pmod{2^\ell}$
- set  $2^\ell \approx d' |\mathcal{R}_\varepsilon|$ ,  $\varepsilon \approx 1/d'$  (to distinguish true vs. false)
- $|\mathcal{R}_\varepsilon| = \text{poly}(d'', 1/\varepsilon) = \text{poly}(d', d'')$
- degree =  $O(\ell) = O(\log |\mathcal{R}_\varepsilon|) = O(\log(d'd''))$
- so, # monomials basically same as randomized Q.E.D.

# More Applications

- also works for **SUM-OR** instead of AND-OR  
⇒ can **count** # orthogonal pairs of vectors
- offline **dominance, orthogonal range searching, or exact  $L_\infty$  nearest neighbor search** in  $n^2 / 2^{\Omega(\sqrt{\log n})}$   
**det.** time in dim.  $d \leq 2^{O(\sqrt{\log n})}$
- can solve **# $k$ -SAT** in  $2^{(1-1/O(k))n}$  **det.** time