Towards Multilinear Depth-3 PIT

- A depth-3 circuit is the following polynomial representation:
  \[ C(x_1, \ldots, x_n) = \sum_{i \in [k]} \prod_{j \in [d]} p_{ij} \]
  where \( p_{ij} \) is a linear polynomial in \( F[x_1, \ldots, x_n] \).

- The size of the circuit is \( ndk \) + bits required to represent the constants in \( F \).

- Depth-3 PIT Qn: Check \( C(x)^3 = 0 \) in \( \text{poly(size(c))} \) time.

- PIT question can be asked for any arithmetic circuit model.

> It has a poly-time randomized algorithm.
- In fact, the randomized algorithm suggests the existence of a poly-sized $H \subseteq \mathbb{F}_2^n$ s.t., \forall circuits $C$ of size $s$,
  \[ C \equiv 0 \Rightarrow \exists x \in H, C(x) \neq 0. \]

- For circuit families "small" hitting-sets exist.

- Blackbox PIT Qn: Can a hitting-set family be constructed in det, poly-time?

- We are mainly interested in blackbox PIT algorithms.
  (In contrast to whitebox PIT.)

- This question is related to circuit lower bound questions.
  \[ \text{PIT} \Rightarrow \text{LB}_0 : [\text{HS'80, KI'04, Apr'05, ...}] \]
  \[ \text{depth} 3 \Rightarrow \text{general: } [\text{AV'08, koi'12, Tav'13, Gkks'13}] \]
- In the last decade, several results have been shown for depth-3 covering both lower bounds & PIT. But they are with restrictions.

- In this talk we restrict to multilinear depth-3. I.e. \( C = \sum_i \prod_j C_{ij} \), where \( \forall i, \{ C_{ij}, \ldots, C_{kl} \} \) are supported on disjoint variables.

- e.g. \( x_1 x_2 \) cannot be computed by multilinear depth-3. While \( (1 + x_1) \cdots (1 + x_n) \) can.

- For this model:
  - exponential (2) lower bounds by [RY’09].
  - sub-exponential \( (n^{2.5\cdot \log n}) \) blackbox PIT by [de Oliveira, Shpilka, Volk ’15].
- We begin the study with set-multilinear depth-3.
  I.e. \( f, f_1, \ldots, f_d \) induces the same partition on the variables \([n]\).

- e.g. \((1+x_1) \cdots (1+x_n) + 1\) is set-multilinear, while \((x_1+x_2)(x_3+x_4) + (x_1+x_3)(x_2+x_4)\) is not (syntactically).

- Set-multilinear depth-3 have a special ABP (arithmetic branching program):
  all variables separated (or read-once oblivious).

- e.g. \((x_1+x_2)(x_3+x_4) + (1+x_1)(1+x_3) = C(x)\) can be expressed as a branching prog.
  of length \(\approx n\)
  width \(\approx kn\)
Further, we get the matrix product:
\[
C(x) = \overline{\pi^T} \cdot \begin{pmatrix}
1 & \left(\begin{array}{c}
1 \\
1 \\
1 \\
\end{array}\right)
\end{pmatrix} \cdot \begin{pmatrix}
1 & 0 \\
0 & 1 \\
0 & 0 \\
\end{pmatrix} \cdot \begin{pmatrix}
1 & 0 \\
0 & 1 \\
0 & 0 \\
\end{pmatrix} \cdot \begin{pmatrix}
1 & 0 \\
0 & 1 \\
0 & 0 \\
\end{pmatrix} \cdot \begin{pmatrix}
1 & 0 \\
0 & 1 \\
0 & 0 \\
\end{pmatrix} .
\]

We call such a matrix product \( \prod_{i=1}^{n} A_i(x_i) = D(x) \in M_w(F)[x] \) an \textbf{ROABP},

and it computes the polynomial \( C(x) = \overline{\pi^T} \cdot D(x) \cdot a \) for \( \overline{\pi}, a \in F^w \).

(C is said to have a width-\( w \) ROABP.)

- PIT algorithms are known (almost):
  1. whitebox \( \text{poly}(nw) \)-time \([\text{RS}'05]\)
  2. blackbox \( \text{poly}(\log n) \)-time \([\text{AGKS}'15]\).

Open:
- Remove \( \log n \)?

- (1) is based on evaluation dimension/coeff. space
  \( \text{rank concentration in } D \).
Idea of (1)

- Given \( D = A_1(x_1) \ldots A_n(x_n) \in M_{w}(\mathbb{F})[x] \)
we want to test \( C = \bar{e}^T \cdot D \cdot \bar{d} \approx 0 \).

- Algorithm sketch:
  (i) For \( i = 2 \) to \( n \)
  (ii) \[ \text{expand } D_i = D_{i-1} \cdot A_i(x_i) \]
  (iii) if \( \text{sparsity}(D_i) > w^2 \) then \( \text{coefficients in } D_i \) are \( \mathbb{F} \)-dependent.
    Keep an \( \mathbb{F} \)-basis & drop the extra monomials.
    (What remains is called \( D_i' \).)
  (iv) Test whether \( \bar{e}^T \cdot D_n \cdot \bar{d} \approx 0 \).

\[ \triangleright \text{coeff-span}_\mathbb{F}(D) = \text{coeff-span}_\mathbb{F}(D_n) \& \]
\[ \text{sparsity}(D_n) \leq w^2. \]

\[ \triangleright \text{This takes time } \text{poly}(nw), \]
\[ \text{But is white-box!} \]
Idea of (2)

- We are given only an oracle to $C(x) = c^T \cdot D \cdot \bar{d}$, where $D = \frac{1}{\sqrt{r}} A_i(x_i)$.

- The idea is to find a map $\varphi : x_i \mapsto t^{w_i}$ s.t. a least basis of coeff-span$_F(D)$ gets isolated in $\varphi(D)$.

$I \leq \omega^2$ → I.e. $\exists$ monomials $S \subseteq \text{support}(D)$ s.t.

(i) $\text{span}_F \{ \text{coeff}(m)(D) \mid m \in S \} = \text{coeff-span}_F(D)$,

& (ii) $\forall m' \notin S$, $\text{coeff}(m')(D) \in$

$\text{span}_F \{ \text{coeff}(m)(D) \mid m \in S, \varphi(m) < \varphi(m') \}$.

If $\varphi$ isolates a least basis in $D$ then: $C \neq 0 \Rightarrow \varphi(C) \neq 0$.

- Thus, all we need to do is to construct “small” weights $w_i$ s.t. $\varphi$ isolates a least basis.
- The idea for $\psi$ is to attempt a recursion on the ROABP length. (Giving a $\log n$ in the exponent.)

**Sketch:**

- Say, the two halves $L = \prod_{i=1}^{n/2} A_i$ and $R = \prod_{i>n/2} A_i$

  have a least basis isolated under a monomial map
  
  $\psi: \mathbb{F}[x_1, \ldots, x_n] \rightarrow \mathbb{F}[t_1, \ldots, t_e]$.

  $\Rightarrow \psi(L) = \text{least-basis-part} + \text{rest}$
  
  $\psi(R) = \text{least-basis-part} + \text{rest}$

- Note that by extending $\psi$ (i.e., the $w^t$ monomials in the product of the two least basis parts are preserved) we achieve least basis isolation in $L \cdot R$!
This extension of $\psi$ needs a new variable $t_{e+1}$ with the ordering $t_1 < t_2 < \ldots < t_{e+1}$ & individual degrees $\text{poly}(w)$.

$\Rightarrow$ By using this recursive step on contiguous blocks of $2, 2^2, \ldots, 2^\lg n$, we get the map $\psi$ in time $O(\lg n)$.

(We get $O(\lg n)$ candidate $\psi$, each $\lg n$-variate & individual degree $\text{poly}(w)$.)

- Recently, [GKST'15] have extended this to a sum of constantly many ROABP's.

  e.g. multilinear depth-3 with few underlying partitions.