Circuit Lower Bounds, Help Functions, and the Remote Point Problem

V Arvind and Srikanth Srinivasan

The Institute of Mathematical Sciences, Chennai, India.

January 7, 2010

Outline

Outline

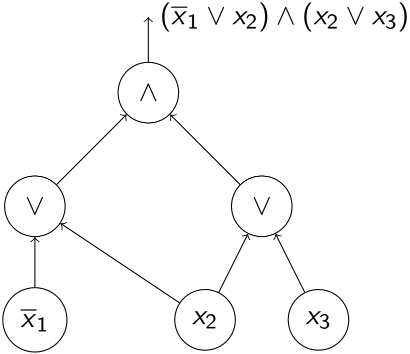
1. Boolean circuits and the Help Functions problem
   - The Help functions problem
   - An application to standard questions
   - The Remote Point Problem (RPP)
   - The connection to the RPP

2. Algebraic Branching Programs with Help polynomials
   - Noncommutative Algebraic Branching Programs
   - Towards explicit lower bounds
   - Results

3. Summary

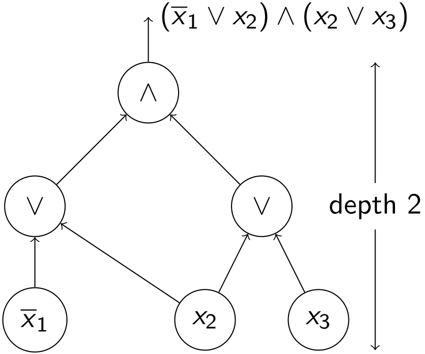
Boolean circuits

- Set of variables $X = \{x_1, x_2, \ldots, x_n\}$.
- Directed acyclic graph (DAG) with labels from $X \cup \overline{X} \cup \{\wedge, \vee\} \cup \{0, 1\}$.
- Computes a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$.



Boolean circuits – parameters

- Size of a circuit – number of vertices.
- Depth of a circuit – The length of the longest path in the circuit.
- Circuits of interest: Constant depth circuits of small size.



Boolean circuit lower bounds

- Notation: Size($s(n)$) – families of functions $\{f_n : \{0,1\}^n \rightarrow \{0,1\}\}_{n \in \mathbb{N}}$ that can be computed by circuits of size $s(n)$. Similarly SizeDepth($s(n), d(n)$).
- $\mathrm{AC}^0 = \mathrm{SizeDepth}(n^{O(1)}, O(1))$.
- AIM: To come up with an explicit (say, computable in EXP) family of boolean functions that cannot be computed by subexponential-sized boolean circuits.
- Current status: $\mathrm{EXP} \not\subseteq \mathrm{Size}(n^c)$ for any fixed $c > 0$.

Boolean circuit lower bounds

- Notation: $\mathrm{Size}(s(n))$ – families of functions $\{f_n : \{0,1\}^n \to \{0,1\}\}_{n\in\mathbb{N}}$ that can be computed by circuits of size $s(n)$. Similarly $\mathrm{SizeDepth}(s(n), d(n))$.
- $\mathrm{AC}^0 = \mathrm{SizeDepth}(n^{O(1)}, O(1))$.
- AIM: To come up with an explicit (say, computable in EXP) family of boolean functions that cannot be computed by subexponential-sized boolean circuits.
- Current status: $\mathrm{EXP} \not\subseteq \mathrm{Size}(n^c)$ for any fixed $c > 0$.

Boolean circuit lower bounds

- Notation: Size($s(n)$) – families of functions $\{f_n : \{0,1\}^n \rightarrow \{0,1\}\}_{n\in\mathbb{N}}$ that can be computed by circuits of size $s(n)$. Similarly SizeDepth($s(n), d(n)$).
- $AC^0 = \text{SizeDepth}(n^{O(1)}, O(1))$.
- AIM: To come up with an explicit (say, computable in EXP) family of boolean functions that cannot be computed by subexponential-sized boolean circuits.
- Current status: EXP $\not\subseteq$ Size(n^c) for any fixed $c > 0$.

Boolean circuit lower bounds

- Notation: Size($s(n)$) – families of functions $\{f_n : \{0,1\}^n \rightarrow \{0,1\}\}_{n \in \mathbb{N}}$ that can be computed by circuits of size $s(n)$. Similarly SizeDepth($s(n), d(n)$).
- $\mathrm{AC}^0 = \mathrm{SizeDepth}(n^{O(1)}, O(1))$.
- AIM: To come up with an explicit (say, computable in EXP) family of boolean functions that cannot be computed by subexponential-sized boolean circuits.
- Current status: EXP $\not\subseteq$ Size(n^c) for any fixed $c > 0$.

Boolean circuit lower bounds (contd.)

- Better lower bounds for restricted classes of circuits.
  - Monotone boolean circuits (Razborov, Alon-Boppana): $2^{n^{\Omega(1)}}$ lower bound for CLIQUE.
  - Constant-depth circuits (Furst-Saxe-Sipser, Yao, Håstad): Parity $\notin$ SizeDepth$(2^{n^{\Omega(1)}}, O(1))$.
  - Constant-depth circuits with Mod_p gates and a few Majority gates (Razborov, Smolensky, Aspnes-Beigel-Furst-Rudich) ...
- Currently unknown: Does all of EXP have polynomial-sized constant depth circuits with Mod_m gates (with m composite)?

Boolean circuit lower bounds (contd.)

- Better lower bounds for restricted classes of circuits.
  - Monotone boolean circuits (Razborov, Alon-Boppana): $2^{n^{\Omega(1)}}$ lower bound for CLIQUE.
  - Constant-depth circuits (Furst-Saxe-Sipser, Yao, Håstad): Parity $\notin$ SizeDepth$(2^{n^{\Omega(1)}}, O(1))$.
  - Constant-depth circuits with Mod_p gates and a few Majority gates (Razborov, Smolensky, Aspnes-Beigel-Furst-Rudich) ...
- Currently unknown: Does all of EXP have polynomial-sized constant depth circuits with Mod_m gates (with m composite)?

Boolean circuit lower bounds (contd.)

- Better lower bounds for restricted classes of circuits.
  - Monotone boolean circuits (Razborov, Alon-Boppana): $2^{n^{\Omega(1)}}$ lower bound for CLIQUE.
  - Constant-depth circuits (Furst-Saxe-Sipser, Yao, Håstad): Parity $\notin$ SizeDepth$(2^{n^{\Omega(1)}}, O(1))$.
  - Constant-depth circuits with Mod_p gates and a few Majority gates (Razborov, Smolensky, Aspnes-Beigel-Furst-Rudich) ...
- Currently unknown: Does all of EXP have polynomial-sized constant depth circuits with Mod_m gates (with m composite)?

Boolean circuit lower bounds (contd.)

- Better lower bounds for restricted classes of circuits.
  - Monotone boolean circuits (Razborov, Alon-Boppana): $2^{n^{\Omega(1)}}$ lower bound for CLIQUE.
  - Constant-depth circuits (Furst-Saxe-Sipser, Yao, Håstad): Parity $\notin$ SizeDepth$(2^{n^{\Omega(1)}}, O(1))$.
  - Constant-depth circuits with Mod_p gates and a few Majority gates (Razborov, Smolensky, Aspnes-Beigel-Furst-Rudich) ...
- Currently unknown: Does all of EXP have polynomial-sized constant depth circuits with Mod_m gates (with m composite)?

Boolean circuit lower bounds (contd.)

- Better lower bounds for restricted classes of circuits.
  - Monotone boolean circuits (Razborov, Alon-Boppana): $2^{n^{\Omega(1)}}$ lower bound for CLIQUE.
  - Constant-depth circuits (Furst-Saxe-Sipser, Yao, Håstad): Parity $\notin$ SizeDepth$(2^{n^{\Omega(1)}}, O(1))$.
  - Constant-depth circuits with Mod_p gates and a few Majority gates (Razborov, Smolensky, Aspnes-Beigel-Furst-Rudich) ...
- Currently unknown: Does all of EXP have polynomial-sized constant depth circuits with Mod_m gates (with m composite)?

The Help functions problem

- Fix $h_1, h_2, \ldots, h_m : \{0,1\}^n \to \{0,1\}$ ($m \approx n^{O(1)}$ or $2^{(\log n)^{O(1)}}$).
- What can constant-depth circuits do when given the ability to compute $H = \{h_1, h_2, \ldots, h_m\}$ (on the given input) for "free"?
- Example: Consider constant-depth boolean circuits that, along with $x_1, x_2, \ldots, x_n$, are also given $\bigoplus_{i=1}^n x_i$ as input. Can they compute $\bigoplus_{i \le n/2} x_i$?

The Help functions problem

- Fix $h_1, h_2, \ldots, h_m : \{0,1\}^n \rightarrow \{0,1\}$ ($m \approx n^{O(1)}$ or $2^{(\log n)^{O(1)}}$).
- What can constant-depth circuits do when given the ability to compute $H = \{h_1, h_2, \ldots, h_m\}$ (on the given input) for "free"?
- Example: Consider constant-depth boolean circuits that, along with $x_1, x_2, \ldots, x_n$, are also given $\bigoplus_{i=1}^{n} x_i$ as input. Can they compute $\bigoplus_{i \leq n/2} x_i$?

The Help functions problem

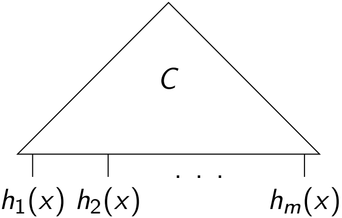
- Fix $h_1, h_2, \ldots, h_m : \{0,1\}^n \rightarrow \{0,1\}$ ($m \approx n^{O(1)}$ or $2^{(\log n)^{O(1)}}$).
- What can constant-depth circuits do when given the ability to compute $H = \{h_1, h_2, \ldots, h_m\}$ (on the given input) for "free"?
- Example: Consider constant-depth boolean circuits that, along with $x_1, x_2, \ldots, x_n$, are also given $\bigoplus_{i=1}^{n} x_i$ as input. Can they compute $\bigoplus_{i \leq n/2} x_i$?

The Help functions problem

- Fix $h_1, h_2, \ldots, h_m : \{0,1\}^n \to \{0,1\}$ ($m \approx n^{O(1)}$ or $2^{(\log n)^{O(1)}}$).
- What can constant-depth circuits do when given the ability to compute $H = \{h_1, h_2, \ldots, h_m\}$ (on the given input) for "free"?
- Example: Consider constant-depth boolean circuits that, along with $x_1, x_2, \ldots, x_n$, are also given $\bigoplus_{i=1}^n x_i$ as input. Can they compute $\bigoplus_{i \leq n/2} x_i$?

The Help functions problem (contd.)

$\text{SizeDepth}_H(s,d)$ - functions computable by circuits of size s and depth d that take functions from H as input.



The Help functions problem (contd.)

- The Help functions problem: another way of extending known circuit lower bounds.
- The $(m(n), s(n), d)$-Help function problem:
  - INPUT: A collection of boolean functions $H = \{h_1, h_2, \ldots, h_m : \{0,1\}^n \rightarrow \{0,1\}\}$.
  - QUESTION: Find a boolean function $F : \{0,1\}^n \rightarrow \{0,1\}$ such that $F \notin \mathrm{SizeDepth}_H(s, d)$.
- Interesting for $d = O(1)$, $m = n^{O(1)}$ or $2^{(\log n)^{O(1)}}$, and $s = 2^{(\log n)^a}$ or $2^{n^{\Omega(1)}}$.

The Help functions problem (contd.)

- The Help functions problem: another way of extending known circuit lower bounds.
- The $(m(n), s(n), d)$-Help function problem:
  - INPUT: A collection of boolean functions $H = \{h_1, h_2, \ldots, h_m : \{0,1\}^n \rightarrow \{0,1\}\}$.
  - QUESTION: Find a boolean function $F : \{0,1\}^n \rightarrow \{0,1\}$ such that $F \notin \text{SizeDepth}_H(s, d)$.
- Interesting for $d = O(1)$, $m = n^{O(1)}$ or $2^{(\log n)^{O(1)}}$, and $s = 2^{(\log n)^a}$ or $2^{n^{\Omega(1)}}$.

The Help functions problem (contd.)

- The Help functions problem: another way of extending known circuit lower bounds.
- The $(m(n), s(n), d)$-Help function problem:
  - INPUT: A collection of boolean functions $H = \{h_1, h_2, \ldots, h_m : \{0,1\}^n \to \{0,1\}\}$.
  - QUESTION: Find a boolean function $F : \{0,1\}^n \to \{0,1\}$ such that $F \notin \mathrm{SizeDepth}_H(s, d)$.
- Interesting for $d = O(1)$, $m = n^{O(1)}$ or $2^{(\log n)^{O(1)}}$, and $s = 2^{(\log n)^a}$ or $2^{n^{\Omega(1)}}$.

The Help functions problem (contd.)

- The Help functions problem: another way of extending known circuit lower bounds.
- The $(m(n), s(n), d)$-Help function problem:
  - INPUT: A collection of boolean functions $H = \{h_1, h_2, \ldots, h_m : \{0,1\}^n \to \{0,1\}\}$.
  - QUESTION: Find a boolean function $F : \{0,1\}^n \to \{0,1\}$ such that $F \notin \mathrm{SizeDepth}_H(s, d)$.
- Interesting for $d = O(1)$, $m = n^{O(1)}$ or $2^{(\log n)^{O(1)}}$, and $s = 2^{(\log n)^a}$ or $2^{n^{\Omega(1)}}$.

The Help functions problem (contd.)

- The Help functions problem: another way of extending known circuit lower bounds.
- The $(m(n), s(n), d)$-Help function problem:
  - INPUT: A collection of boolean functions $H = \{h_1, h_2, \ldots, h_m : \{0,1\}^n \rightarrow \{0,1\}\}$.
  - QUESTION: Find a boolean function $F : \{0,1\}^n \rightarrow \{0,1\}$ such that $F \notin \text{SizeDepth}_H(s, d)$.
- Interesting for $d = O(1)$, $m = n^{O(1)}$ or $2^{(\log n)^{O(1)}}$, and $s = 2^{(\log n)^a}$ or $2^{n^{\Omega(1)}}$.

Previous work

- Has been studied by Jin-Yi Cai (1991) and Satya Lokam (1995).
- Cai proves "almost-explicit" lower bounds when $H = \{x_1, \ldots, x_n\} \cup \{h_1, h_2, \ldots, h_k\}$, and $k \leq n^{1/5-\varepsilon}$.
- Lokam: connections to problems in communication complexity.

Previous work

- Has been studied by Jin-Yi Cai (1991) and Satya Lokam (1995).
- Cai proves "almost-explicit" lower bounds when $H = \{x_1, \ldots, x_n\} \cup \{h_1, h_2, \ldots, h_k\}$, and $k \leq n^{1/5-\varepsilon}$.
- Lokam: connections to problems in communication complexity.

Previous work

- Has been studied by Jin-Yi Cai (1991) and Satya Lokam (1995).
- Cai proves "almost-explicit" lower bounds when $H = \{x_1, \ldots, x_n\} \cup \{h_1, h_2, \ldots, h_k\}$, and $k \leq n^{1/5-\varepsilon}$.
- Lokam: connections to problems in communication complexity.

An application to standard questions

- Suspected: EXP $\not\subseteq$ Size$(n^{O(1)})$.
- Weaker statement: EXP does not polynomial-time many-one reduce to SizeDepth$(n^{O(1)}, O(1))$ (a.k.a. AC^0).
- To prove a lower bound, we want an $L \in$ EXP such that L does not polynomial-time reduce to SizeDepth$(n^{O(1)}, O(1))$.
- Define $L(x)$ by diagonalization. Defining $L_n : \{0,1\}^n \to \{0,1\}$:

An application to standard questions

- Suspected: EXP $\not\subseteq$ Size($n^{O(1)}$).
- Weaker statement: EXP does not polynomial-time many-one reduce to SizeDepth($n^{O(1)}, O(1)$) (a.k.a. AC^0).
- To prove a lower bound, we want an $L \in$ EXP such that L does not polynomial-time reduce to SizeDepth($n^{O(1)}, O(1)$).
- Define $L(x)$ by diagonalization. Defining $L_n : \{0,1\}^n \rightarrow \{0,1\}$:

An application to standard questions

- Suspected: EXP $\not\subseteq$ Size$(n^{O(1)})$.
- Weaker statement: EXP does not polynomial-time many-one reduce to SizeDepth$(n^{O(1)}, O(1))$ (a.k.a. AC^0).
- To prove a lower bound, we want an $L \in$ EXP such that L does not polynomial-time reduce to SizeDepth$(n^{O(1)}, O(1))$.
- Define $L(x)$ by diagonalization. Defining $L_n : \{0,1\}^n \rightarrow \{0,1\}$:

An application to standard questions

- Suspected: $\mathrm{EXP} \not\subseteq \mathrm{Size}(n^{O(1)})$.
- Weaker statement: EXP does not polynomial-time many-one reduce to $\mathrm{SizeDepth}(n^{O(1)}, O(1))$ (a.k.a. AC^0).
- To prove a lower bound, we want an $L \in \mathrm{EXP}$ such that L does not polynomial-time reduce to $\mathrm{SizeDepth}(n^{O(1)}, O(1))$.
- Define $L(x)$ by diagonalization. Defining $L_n : \{0,1\}^n \rightarrow \{0,1\}$:

R_1

R_2

R_3

$\vdots$

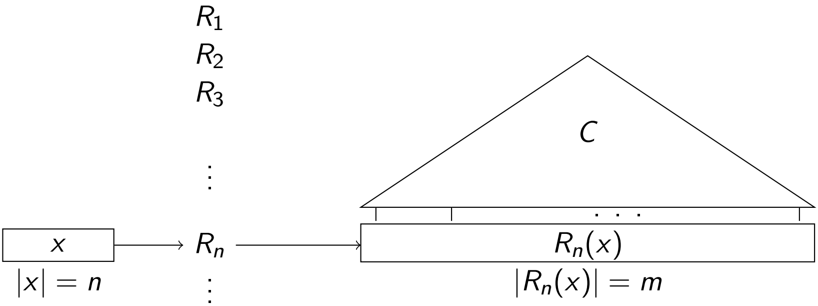
$\boxed{x}$ R_n

$|x| = n$

$\vdots$

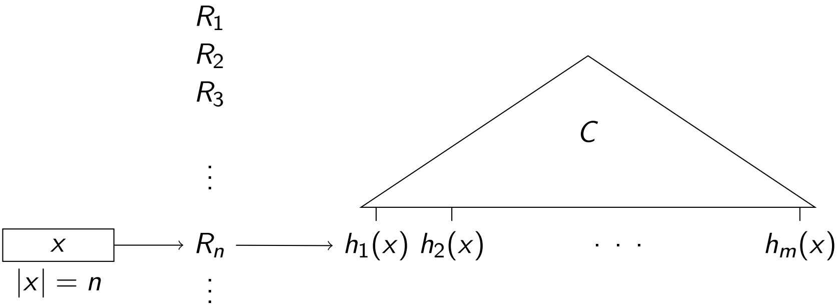
An application to standard questions

- Suspected: EXP $\not\subseteq$ Size$(n^{O(1)})$.
- Weaker statement: EXP does not polynomial-time many-one reduce to SizeDepth$(n^{O(1)}, O(1))$ (a.k.a. AC^0).
- To prove a lower bound, we want an $L \in$ EXP such that L does not polynomial-time reduce to SizeDepth$(n^{O(1)}, O(1))$.
- Define $L(x)$ by diagonalization. Defining $L_n : \{0,1\}^n \to \{0,1\}$:



An application to standard questions

- Suspected: EXP $\not\subseteq$ Size($n^{O(1)}$).
- Weaker statement: EXP does not polynomial-time many-one reduce to SizeDepth($n^{O(1)}, O(1)$) (a.k.a. AC^0).
- To prove a lower bound, we want an $L \in$ EXP such that L does not polynomial-time reduce to SizeDepth($n^{O(1)}, O(1)$).
- Define $L(x)$ by diagonalization. Defining $L_n : \{0,1\}^n \to \{0,1\}$:



Our observation

A solution to the Help Function problem (for constant-depth circuits) would follow from a "good" solution to the Remote Point Problem.

The Remote Point Problem (RPP)

- Define the $(k(N), r(N))$-Remote Point Problem (RPP) as follows:
  - INPUT: A basis for a subspace V of $\mathbb{F}_2^N$ of dimension at most $k = k(N)$.
  - SOLUTION: A vector $u \in \mathbb{F}_2^N$ such that $\Delta(u, v) \geq r(N)$ for all $v \in V$.
- Here, $\Delta(x, y)$ is the Hamming distance between x and y: that is, $|\{i \in [n] \mid x_i \neq y_i\}|$.

The Remote Point Problem (RPP)

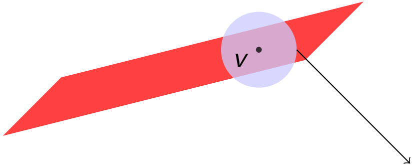
- Define the $(k(N), r(N))$-Remote Point Problem (RPP) as follows:
  - INPUT: A basis for a subspace V of $\mathbb{F}_2^N$ of dimension at most $k = k(N)$.
  - SOLUTION: A vector $u \in \mathbb{F}_2^N$ such that $\Delta(u, v) \geq r(N)$ for all $v \in V$.
- Here, $\Delta(x, y)$ is the Hamming distance between x and y: that is, $|\{i \in [n] \mid x_i \neq y_i\}|$.

The Remote Point Problem (RPP)

- Define the $(k(N), r(N))$-Remote Point Problem (RPP) as follows:
  - INPUT: A basis for a subspace V of $\mathbb{F}_2^N$ of dimension at most $k = k(N)$.
  - SOLUTION: A vector $u \in \mathbb{F}_2^N$ such that $\Delta(u, v) \geq r(N)$ for all $v \in V$.
- Here, $\Delta(x, y)$ is the Hamming distance between x and y: that is, $|\{i \in [n] \mid x_i \neq y_i\}|$.

The Remote Point Problem (RPP)

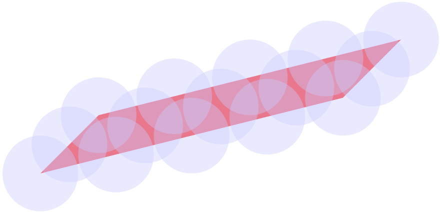
- Define the $(k(N), r(N))$-Remote Point Problem (RPP) as follows:
  - INPUT: A basis for a subspace V of $\mathbb{F}_2^N$ of dimension at most $k = k(N)$.
  - SOLUTION: A vector $u \in \mathbb{F}_2^N$ such that $\Delta(u, v) \geq r(N)$ for all $v \in V$.
- Here, $\Delta(x, y)$ is the Hamming distance between x and y: that is, $|\{i \in [n] \mid x_i \neq y_i\}|$.



$$B(v) = \{v' \mid \Delta(v, v') < r\}$$

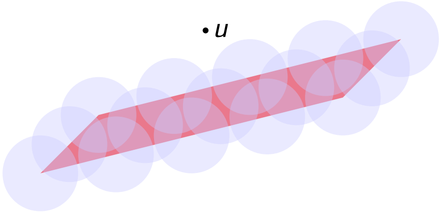
The Remote Point Problem (RPP)

- Define the $(k(N), r(N))$-Remote Point Problem (RPP) as follows:
  - INPUT: A basis for a subspace V of $\mathbb{F}_2^N$ of dimension at most $k = k(N)$.
  - SOLUTION: A vector $u \in \mathbb{F}_2^N$ such that $\Delta(u, v) \geq r(N)$ for all $v \in V$.
- Here, $\Delta(x, y)$ is the Hamming distance between x and y: that is, $|\{i \in [n] \mid x_i \neq y_i\}|$.



The Remote Point Problem (RPP)

- Define the $(k(N), r(N))$-Remote Point Problem (RPP) as follows:
  - INPUT: A basis for a subspace V of $\mathbb{F}_2^N$ of dimension at most $k = k(N)$.
  - SOLUTION: A vector $u \in \mathbb{F}_2^N$ such that $\Delta(u, v) \geq r(N)$ for all $v \in V$.
- Here, $\Delta(x, y)$ is the Hamming distance between x and y: that is, $|\{i \in [n] \mid x_i \neq y_i\}|$.



Motivation and previous work

- Introduced by Alon, Panigrahy, and Yekhanin (2008).
- An interesting "restriction" of the Matrix Rigidity question.
- The Matrix Rigidity question may be phrased in terms of small hitting sets for the RPP.
- Interesting parameters: $(k(N) = N/10, r(N) = N/10)$. Random point is a solution w.h.p.. Need a deterministic solution.
- Current best solution (Alon-Panigrahy-Yekhanin): The $(k, N\frac{\log k}{k})$-RPP has a polynomial-time algorithm for $k \leq N/2$.

Motivation and previous work

- Introduced by Alon, Panigrahy, and Yekhanin (2008).
- An interesting "restriction" of the Matrix Rigidity question.
- The Matrix Rigidity question may be phrased in terms of small hitting sets for the RPP.
- Interesting parameters: $(k(N) = N/10, r(N) = N/10)$. Random point is a solution w.h.p.. Need a deterministic solution.
- Current best solution (Alon-Panigrahy-Yekhanin): The $(k, N\frac{\log k}{k})$-RPP has a polynomial-time algorithm for $k \leq N/2$.

Motivation and previous work

- Introduced by Alon, Panigrahy, and Yekhanin (2008).
- An interesting "restriction" of the Matrix Rigidity question.
- The Matrix Rigidity question may be phrased in terms of small hitting sets for the RPP.
- Interesting parameters: $(k(N) = N/10, r(N) = N/10)$. Random point is a solution w.h.p.. Need a deterministic solution.
- Current best solution (Alon-Panigrahy-Yekhanin): The $(k, N\frac{\log k}{k})$-RPP has a polynomial-time algorithm for $k \leq N/2$.

Motivation and previous work

- Introduced by Alon, Panigrahy, and Yekhanin (2008).
- An interesting "restriction" of the Matrix Rigidity question.
- The Matrix Rigidity question may be phrased in terms of small hitting sets for the RPP.
- Interesting parameters: $(k(N) = N/10, r(N) = N/10)$. Random point is a solution w.h.p.. Need a deterministic solution.
- Current best solution (Alon-Panigrahy-Yekhanin): The $(k, N\frac{\log k}{k})$-RPP has a polynomial-time algorithm for $k \leq N/2$.

Motivation and previous work

- Introduced by Alon, Panigrahy, and Yekhanin (2008).
- An interesting "restriction" of the Matrix Rigidity question.
- The Matrix Rigidity question may be phrased in terms of small hitting sets for the RPP.
- Interesting parameters: $(k(N) = N/10, r(N) = N/10)$. Random point is a solution w.h.p.. Need a deterministic solution.
- Current best solution (Alon-Panigrahy-Yekhanin): The $(k, N\frac{\log k}{k})$-RPP has a polynomial-time algorithm for $k \leq N/2$.

Motivation and previous work

- Introduced by Alon, Panigrahy, and Yekhanin (2008).
- An interesting "restriction" of the Matrix Rigidity question.
- The Matrix Rigidity question may be phrased in terms of small hitting sets for the RPP.
- Interesting parameters: $(k(N) = N/10, r(N) = N/10)$. Random point is a solution w.h.p.. Need a deterministic solution.
- Current best solution (Alon-Panigrahy-Yekhanin): The $(k, N\frac{\log k}{k})$-RPP has a polynomial-time algorithm for $k \leq N/2$.

The connection to the Help functions problem

- The $(m(n), s(n), d)$-Help function problem:
  - INPUT: A collection of boolean functions $H = \{h_1, h_2, \ldots, h_m : \{0,1\}^n \rightarrow \{0,1\}\}$.
  - QUESTION: Find a boolean function $F : \{0,1\}^n \rightarrow \{0,1\}$ such that $F \notin \text{SizeDepth}_H(s, d)$.
- C - small constant-depth boolean circuit with m inputs.
- Using low-degree polynomial approximations to AC^0 (Razborov, Smolensky, Tarui), there is a polynomial p_0 of small degree (at most $\ell = \log^{O(1)}(m)$) such that,

$$\Pr_{x \sim \{0,1\}^n}[p_0(h_1(x), \ldots, h_m(x)) = C(h_1(x), \ldots, h_m(x))] > 1 - \varepsilon$$

The connection to the Help functions problem

- The $(m(n), s(n), d)$-Help function problem:
  - INPUT: A collection of boolean functions $H = \{h_1, h_2, \ldots, h_m : \{0,1\}^n \to \{0,1\}\}$.
  - QUESTION: Find a boolean function $F : \{0,1\}^n \to \{0,1\}$ such that $F \notin \mathrm{SizeDepth}_H(s, d)$.
- C - small constant-depth boolean circuit with m inputs.
- Using low-degree polynomial approximations to AC^0 (Razborov, Smolensky, Tarui), there is a polynomial p_0 of small degree (at most $\ell = \log^{O(1)}(m)$) such that,

$$\Pr_{x \sim \{0,1\}^n}[p_0(h_1(x), \ldots, h_m(x)) = C(h_1(x), \ldots, h_m(x))] > 1 - \varepsilon$$

The connection to the Help functions problem (contd.)

0	1	0	0	1	· · · ·	0	1	0	$C(h_1(x), \ldots, h_m(x))$

Hamming distance $< \varepsilon 2^n$.

0	0	0	0	1	· · · ·	1	1	0	$p_0(h_1(x), \ldots, h_m(x))$

- $N = 2^n$. Let V be the subspace of $\mathbb{F}_2^N$ of all degree $\leq \ell$ polynomials in $h_1, h_2, \ldots, h_m$.
- Any function F such that $\Delta(F, V) \geq \varepsilon N$ cannot be computed by a small constant-depth circuit using $h_1, h_2, \ldots, h_m$.
- An $(m^\ell, \varepsilon N)$-solution to the RPP would give such a function.

The connection to the Help functions problem (contd.)

0	1	0	0	1	$\cdot\ \cdot\ \cdot\ \cdot$	0	1	0	$C(h_1(x), \ldots, h_m(x))$

Hamming distance $< \varepsilon 2^n$.

0	0	0	0	1	$\cdot\ \cdot\ \cdot\ \cdot$	1	1	0	$p_0(h_1(x), \ldots, h_m(x))$

- $N = 2^n$. Let V be the subspace of $\mathbb{F}_2^N$ of all degree $\leq \ell$ polynomials in $h_1, h_2, \ldots, h_m$.
- Any function F such that $\Delta(F, V) \geq \varepsilon N$ cannot be computed by a small constant-depth circuit using $h_1, h_2, \ldots, h_m$.
- An $(m^\ell, \varepsilon N)$-solution to the RPP would give such a function.

The connection to the Help functions problem (contd.)

0	1	0	0	1	· · · ·	0	1	0	$C(h_1(x), \ldots, h_m(x))$

Hamming distance $< \varepsilon 2^n$.

0	0	0	0	1	· · · ·	1	1	0	$p_0(h_1(x), \ldots, h_m(x))$

- $N = 2^n$. Let V be the subspace of $\mathbb{F}_2^N$ of all degree $\leq \ell$ polynomials in $h_1, h_2, \ldots, h_m$.
- Any function F such that $\Delta(F, V) \geq \varepsilon N$ cannot be computed by a small constant-depth circuit using $h_1, h_2, \ldots, h_m$.
- An $(m^\ell, \varepsilon N)$-solution to the RPP would give such a function.

The connection to the Help functions problem (contd.)

0	1	0	0	1	· · · ·	0	1	0	$C(h_1(x), \ldots, h_m(x))$

Hamming distance $< \varepsilon 2^n$.

0	0	0	0	1	· · · ·	1	1	0	$p_0(h_1(x), \ldots, h_m(x))$

- $N = 2^n$. Let V be the subspace of $\mathbb{F}_2^N$ of all degree $\leq \ell$ polynomials in $h_1, h_2, \ldots, h_m$.
- Any function F such that $\Delta(F, V) \geq \varepsilon N$ cannot be computed by a small constant-depth circuit using $h_1, h_2, \ldots, h_m$.
- An $(m^\ell, \varepsilon N)$-solution to the RPP would give such a function.

Does this help?

- Does the connection to the RPP give us a non-trivial solution to the Help functions problem?
- Not really. The best solution currently (Alon et. al.) is a $(k, N^{\frac{\log k}{k}})$-solution. Need a $(k, N^{\frac{1}{k^{o(1)}}})$-solution.
- However, interesting that a *restriction* of the rigidity question already implies some nontrivial lower bounds.
- Also, in the *algebraic* setting, this point of view does give some non-obvious results.

Does this help?

- Does the connection to the RPP give us a non-trivial solution to the Help functions problem?
- Not really. The best solution currently (Alon et. al.) is a $(k, N^{\frac{\log k}{k}})$-solution. Need a $(k, N^{\frac{1}{k^{o(1)}}})$-solution.
- However, interesting that a *restriction* of the rigidity question already implies some nontrivial lower bounds.
- Also, in the *algebraic* setting, this point of view does give some non-obvious results.

Does this help?

- Does the connection to the RPP give us a non-trivial solution to the Help functions problem?
- Not really. The best solution currently (Alon et. al.) is a $(k, N^{\frac{\log k}{k}})$-solution. Need a $(k, N^{\frac{1}{k^{o(1)}}})$-solution.
- However, interesting that a *restriction* of the rigidity question already implies some nontrivial lower bounds.
- Also, in the *algebraic* setting, this point of view does give some non-obvious results.

Does this help?

- Does the connection to the RPP give us a non-trivial solution to the Help functions problem?
- Not really. The best solution currently (Alon et. al.) is a $(k, N^{\frac{\log k}{k}})$-solution. Need a $(k, N^{\frac{1}{k^{o(1)}}})$-solution.
- However, interesting that a *restriction* of the rigidity question already implies some nontrivial lower bounds.
- Also, in the *algebraic* setting, this point of view does give some non-obvious results.

Does this help?

- Does the connection to the RPP give us a non-trivial solution to the Help functions problem?
- Not really. The best solution currently (Alon et. al.) is a $(k, N^{\frac{\log k}{k}})$-solution. Need a $(k, N^{\frac{1}{k^{o(1)}}})$-solution.
- However, interesting that a *restriction* of the rigidity question already implies some nontrivial lower bounds.
- Also, in the *algebraic* setting, this point of view does give some non-obvious results.

Outline

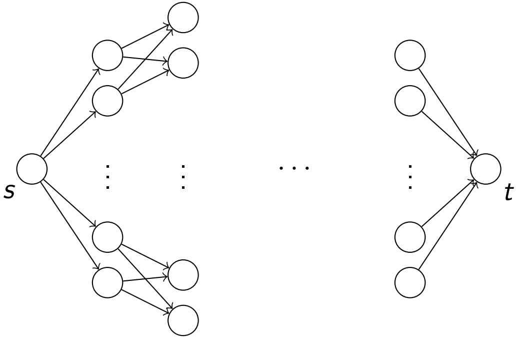
Noncommutative Algebraic Branching Programs (ABPs)

- Field $\mathbb{F}$. Set of variables $X = \{x_1, x_2, \ldots, x_n\}$.
- Noncommutative ring of polynomials $\mathbb{F}\langle X \rangle$. $x_1 x_2 \neq x_2 x_1$.

The RMP

Noncommutative Algebraic Branching Programs (ABPs)

- Field $\mathbb{F}$. Set of variables $X = \{x_1, x_2, \ldots, x_n\}$.
- Noncommutative ring of polynomials $\mathbb{F}\langle X \rangle$. $x_1 x_2 \neq x_2 x_1$.

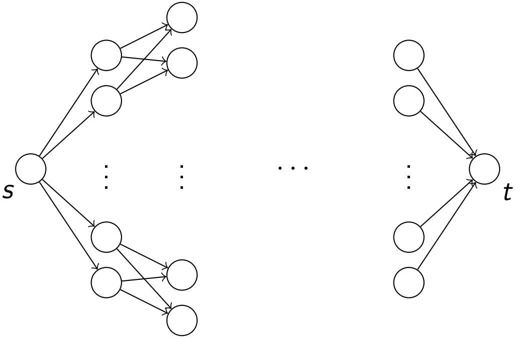


The RMP

Noncommutative Algebraic Branching Programs (ABPs)

- Field $\mathbb{F}$. Set of variables $X = \{x_1, x_2, \ldots, x_n\}$.
- Noncommutative ring of polynomials $\mathbb{F}\langle X \rangle$. $x_1 x_2 \neq x_2 x_1$.

$\ell = \sum_i \alpha_i x_i$



▸ The RMP

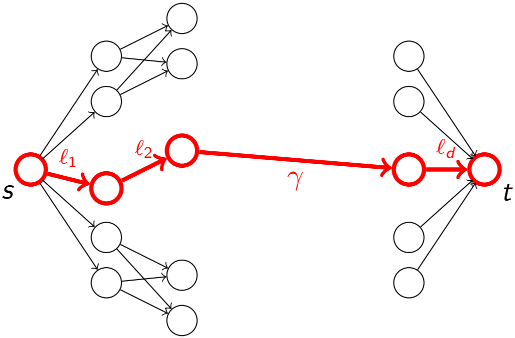
Noncommutative Algebraic Branching Programs (ABPs)

- Field $\mathbb{F}$. Set of variables $X = \{x_1, x_2, \ldots, x_n\}$.
- Noncommutative ring of polynomials $\mathbb{F}\langle X \rangle$. $x_1 x_2 \neq x_2 x_1$.

$$\bigcirc \xrightarrow{\ell} \bigcirc$$

$$\ell = \sum_i \alpha_i x_i$$

$$f_\gamma = \ell_1 \ell_2 \cdots \ell_d$$

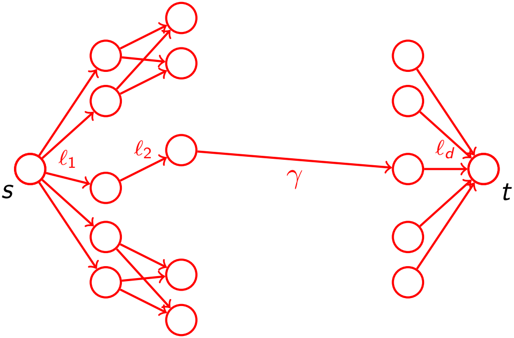


The RMP

Noncommutative Algebraic Branching Programs (ABPs)

- Field $\mathbb{F}$. Set of variables $X = \{x_1, x_2, \ldots, x_n\}$.
- Noncommutative ring of polynomials $\mathbb{F}\langle X \rangle$. $x_1 x_2 \neq x_2 x_1$.

$$\ell = \sum_i \alpha_i x_i$$



$$f_\gamma = \ell_1 \ell_2 \cdots \ell_d$$

$$f = \sum_{\gamma \in \mathcal{P}_{st}} f_\gamma$$

The RMP

Properties

- An ABP with d layers computes homogeneous (degree d) polynomials in the noncommutative ring $\mathbb{F}\langle X \rangle$.
- Size of an ABP A: the number of vertices in the underlying graph.
- ABPs at least as powerful as arithmetic formulas.
- Nisan proved exponential lower bounds for the size of ABPs computing a whole range of *noncommutative* polynomials, such as the Determinant, the Permanent, etc.
- Only explicit lower bounds for the noncommutative arithmetic model. Lower bounds for general noncommutative arithmetic circuits unknown.

The RMP

Properties

- An ABP with d layers computes homogeneous (degree d) polynomials in the noncommutative ring $\mathbb{F}\langle X\rangle$.
- Size of an ABP A: the number of vertices in the underlying graph.
- ABPs at least as powerful as arithmetic formulas.
- Nisan proved exponential lower bounds for the size of ABPs computing a whole range of *noncommutative* polynomials, such as the Determinant, the Permanent, etc.
- Only explicit lower bounds for the noncommutative arithmetic model. Lower bounds for general noncommutative arithmetic circuits unknown.

The RMP

Properties

- An ABP with d layers computes homogeneous (degree d) polynomials in the noncommutative ring $\mathbb{F}\langle X\rangle$.
- Size of an ABP A: the number of vertices in the underlying graph.
- ABPs at least as powerful as arithmetic formulas.
- Nisan proved exponential lower bounds for the size of ABPs computing a whole range of *noncommutative* polynomials, such as the Determinant, the Permanent, etc.
- Only explicit lower bounds for the noncommutative arithmetic model. Lower bounds for general noncommutative arithmetic circuits unknown.

The RMP

Properties

- An ABP with d layers computes homogeneous (degree d) polynomials in the noncommutative ring $\mathbb{F}\langle X \rangle$.
- Size of an ABP A: the number of vertices in the underlying graph.
- ABPs at least as powerful as arithmetic formulas.
- Nisan proved exponential lower bounds for the size of ABPs computing a whole range of *noncommutative* polynomials, such as the Determinant, the Permanent, etc.
- Only explicit lower bounds for the noncommutative arithmetic model. Lower bounds for general noncommutative arithmetic circuits unknown.

The RMP

Properties

- An ABP with d layers computes homogeneous (degree d) polynomials in the noncommutative ring $\mathbb{F}\langle X\rangle$.
- Size of an ABP A: the number of vertices in the underlying graph.
- ABPs at least as powerful as arithmetic formulas.
- Nisan proved exponential lower bounds for the size of ABPs computing a whole range of *noncommutative* polynomials, such as the Determinant, the Permanent, etc.
- Only explicit lower bounds for the noncommutative arithmetic model. Lower bounds for general noncommutative arithmetic circuits unknown.

The RMP

Noncommutative ABPs with help polynomials

- Fix $H = \{h_1, h_2, \ldots, h_m\}$, a set of arbitrary polynomials from the noncommutative ring $\mathbb{F}\langle X \rangle$.
- ABPs with help polynomials H - Same as standard ABPs, except we allow the h_i in the linear forms.

$$\ell = \sum_i \alpha_i x_i + \sum_j \beta_j h_j$$

- The ABP with help polynomials lower bound question: Given $H = \{h_1, h_2, \ldots, h_m\}$, compute a polynomial F such that F cannot be computed by a small ABP using H.

The RMP

Noncommutative ABPs with help polynomials

- Fix $H = \{h_1, h_2, \ldots, h_m\}$, a set of arbitrary polynomials from the noncommutative ring $\mathbb{F}\langle X \rangle$.
- ABPs with help polynomials H - Same as standard ABPs, except we allow the h_i in the linear forms.

$$\ell = \sum_i \alpha_i x_i + \sum_j \beta_j h_j$$

- The ABP with help polynomials lower bound question: Given $H = \{h_1, h_2, \ldots, h_m\}$, compute a polynomial F such that F cannot be computed by a small ABP using H.

The RMP

Noncommutative ABPs with help polynomials

- Fix $H = \{h_1, h_2, \ldots, h_m\}$, a set of arbitrary polynomials from the noncommutative ring $\mathbb{F}\langle X \rangle$.
- ABPs with help polynomials H - Same as standard ABPs, except we allow the h_i in the linear forms.

$$\ell = \sum_i \alpha_i x_i + \sum_j \beta_j h_j$$

- The ABP with help polynomials lower bound question: Given $H = \{h_1, h_2, \ldots, h_m\}$, compute a polynomial F such that F cannot be computed by a small ABP using H.

The RMP

The communication matrix $M_k(f)$

- Fix $f \in \mathbb{F}\langle X\rangle$ homogeneous of degree d.
- $\mathrm{Mon}_\ell(X)$ – monic monomials of degree ℓ.
- $f(m)$ – coefficient of monomial m in f.
- For $0 \le k \le d$, the matrix $M_k(f)$ is an $n^k \times n^{d-k}$ matrix over $\mathbb{F}$ such that:
  - The rows are labelled by elements of $\mathrm{Mon}_k(X)$.
  - The columns are labelled by elements of $\mathrm{Mon}_{d-k}(X)$.
  - The (m_1, m_2)th entry is $f(m_1 m_2)$.

The communication matrix $M_k(f)$

- Fix $f \in \mathbb{F}\langle X \rangle$ homogeneous of degree d.
- $\mathrm{Mon}_\ell(X)$ – monic monomials of degree ℓ.
- $f(m)$ – coefficient of monomial m in f.
- For $0 \leq k \leq d$, the matrix $M_k(f)$ is an $n^k \times n^{d-k}$ matrix over $\mathbb{F}$ such that:
  - The rows are labelled by elements of $\mathrm{Mon}_k(X)$.
  - The columns are labelled by elements of $\mathrm{Mon}_{d-k}(X)$.
  - The (m_1, m_2)th entry is $f(m_1 m_2)$.

The communication matrix $M_k(f)$

- Fix $f \in \mathbb{F}\langle X \rangle$ homogeneous of degree d.
- $\mathrm{Mon}_\ell(X)$ – monic monomials of degree ℓ.
- $f(m)$ – coefficient of monomial m in f.
- For $0 \leq k \leq d$, the matrix $M_k(f)$ is an $n^k \times n^{d-k}$ matrix over $\mathbb{F}$ such that:
  - The rows are labelled by elements of $\mathrm{Mon}_k(X)$.
  - The columns are labelled by elements of $\mathrm{Mon}_{d-k}(X)$.
  - The (m_1, m_2)th entry is $f(m_1 m_2)$.

The communication matrix $M_k(f)$

- Fix $f \in \mathbb{F}\langle X \rangle$ homogeneous of degree d.
- $\mathrm{Mon}_\ell(X)$ – monic monomials of degree ℓ.
- $f(m)$ – coefficient of monomial m in f.
- For $0 \leq k \leq d$, the matrix $M_k(f)$ is an $n^k \times n^{d-k}$ matrix over $\mathbb{F}$ such that:
  - The rows are labelled by elements of $\mathrm{Mon}_k(X)$.
  - The columns are labelled by elements of $\mathrm{Mon}_{d-k}(X)$.
  - The (m_1, m_2)th entry is $f(m_1 m_2)$.

The communication matrix $M_k(f)$

$$
\begin{array}{c c}
 & \mathrm{Mon}_{d-k}(X) \longrightarrow \qquad m_2 \\
\begin{array}{c} \mathrm{Mon}_k(X) \\ \downarrow \\ \\ \\ m_1 \\ \\ \\ \end{array} &
\begin{pmatrix}
. & . & . & . & & \cdots & . \\
. & . & . & . & & \cdots & . \\
. & . & . & . & & \cdots & . \\
 & & & & f(m_1 m_2) & & \\
\vdots & \vdots & \vdots & \vdots & & \cdots & . \\
. & . & . & . & & \cdots & .
\end{pmatrix}
\end{array}
$$

The approach to lower bounds

- Say we have a small ABP A computing f using H.
- Then, $M_{d/2}(f) = M' + M$, where:
  - M' small rank.
  - $M \in V(H)$, where $V(H)$ a small dimensional vector space depending *only* on H.
- Thus, for an explicit lower bound, it suffices to find M_0 such that $\mathrm{rank}(M_0 - M)$ is large for *every* $M \in V(H)$. Then, choose $F \in \mathbb{F}\langle X\rangle$ so that:

$$M_{d/2}(F) = M_0$$

- F cannot be computed by small ABPs using H.

The approach to lower bounds

- Say we have a small ABP A computing f using H.
- Then, $M_{d/2}(f) = M' + M$, where:
  - M' small rank.
  - $M \in V(H)$, where $V(H)$ a small dimensional vector space depending *only* on H.
- Thus, for an explicit lower bound, it suffices to find M_0 such that $\text{rank}(M_0 - M)$ is large for *every* $M \in V(H)$. Then, choose $F \in \mathbb{F}\langle X \rangle$ so that:

$$M_{d/2}(F) = M_0$$

- F cannot be computed by small ABPs using H.

The approach to lower bounds

- Say we have a small ABP A computing f using H.
- Then, $M_{d/2}(f) = M' + M$, where:
  - M' small rank.
  - $M \in V(H)$, where $V(H)$ a small dimensional vector space depending *only* on H.
- Thus, for an explicit lower bound, it suffices to find M_0 such that rank$(M_0 - M)$ is large for *every* $M \in V(H)$. Then, choose $F \in \mathbb{F}\langle X \rangle$ so that:

$$M_{d/2}(F) = M_0$$

- F cannot be computed by small ABPs using H.

The approach to lower bounds

- Say we have a small ABP A computing f using H.
- Then, $M_{d/2}(f) = M' + M$, where:
  - M' small rank.
  - $M \in V(H)$, where $V(H)$ a small dimensional vector space depending *only* on H.
- Thus, for an explicit lower bound, it suffices to find M_0 such that $\text{rank}(M_0 - M)$ is large for *every* $M \in V(H)$. Then, choose $F \in \mathbb{F}\langle X \rangle$ so that:

$$M_{d/2}(F) = M_0$$

- F cannot be computed by small ABPs using H.

The Remote Matrix Problem (the RPP with rank metric)

- Let $\Delta_{\text{rank}}(M_1, M_2) = \text{rank}(M_1 - M_2)$.
- The $(k(N), r(N))$-Remote Matrix Problem (RMP) is defined as follows:
  - INPUT: A collection of matrices $M_1, M_2, \ldots, M_k \in \mathbb{F}^{N \times N}$.
  - SOLUTION: A matrix $M \in \mathbb{F}^{N \times N}$ such that $\Delta_{\text{rank}}(M - M') \geq r$ for each $M' \in \text{span}(M_1, M_2, \ldots, M_k)$.
- Easy parameters: The $(k, N/(k+1))$-RMP has an easy solution.
- Interesting parameters: $k = N^2/10$, $r = N/10$. Random point is a solution w.h.p..

The Remote Matrix Problem (the RPP with rank metric)

- Let $\Delta_{\text{rank}}(M_1, M_2) = \text{rank}(M_1 - M_2)$.
- The $(k(N), r(N))$-Remote Matrix Problem (RMP) is defined as follows:
  - INPUT: A collection of matrices $M_1, M_2, \ldots, M_k \in \mathbb{F}^{N \times N}$.
  - SOLUTION: A matrix $M \in \mathbb{F}^{N \times N}$ such that $\Delta_{\text{rank}}(M - M') \geq r$ for each $M' \in \text{span}(M_1, M_2, \ldots, M_k)$.
- Easy parameters: The $(k, N/(k+1))$-RMP has an easy solution.
- Interesting parameters: $k = N^2/10$, $r = N/10$. Random point is a solution w.h.p..

The Remote Matrix Problem (the RPP with rank metric)

- Let $\Delta_{\text{rank}}(M_1, M_2) = \text{rank}(M_1 - M_2)$.
- The $(k(N), r(N))$-Remote Matrix Problem (RMP) is defined as follows:
  - INPUT: A collection of matrices $M_1, M_2, \ldots, M_k \in \mathbb{F}^{N \times N}$.
  - SOLUTION: A matrix $M \in \mathbb{F}^{N \times N}$ such that $\Delta_{\text{rank}}(M - M') \geq r$ for each $M' \in \text{span}(M_1, M_2, \ldots, M_k)$.
- Easy parameters: The $(k, N/(k+1))$-RMP has an easy solution.
- Interesting parameters: $k = N^2/10$, $r = N/10$. Random point is a solution w.h.p..

The Remote Matrix Problem (the RPP with rank metric)

- Let $\Delta_{\text{rank}}(M_1, M_2) = \text{rank}(M_1 - M_2)$.
- The $(k(N), r(N))$-Remote Matrix Problem (RMP) is defined as follows:
  - INPUT: A collection of matrices $M_1, M_2, \ldots, M_k \in \mathbb{F}^{N\times N}$.
  - SOLUTION: A matrix $M \in \mathbb{F}^{N\times N}$ such that $\Delta_{\text{rank}}(M - M') \geq r$ for each $M' \in \text{span}(M_1, M_2, \ldots, M_k)$.
- Easy parameters: The $(k, N/(k+1))$-RMP has an easy solution.
- Interesting parameters: $k = N^2/10$, $r = N/10$. Random point is a solution w.h.p..

The Remote Matrix Problem (the RPP with rank metric)

- Let $\Delta_{\text{rank}}(M_1, M_2) = \text{rank}(M_1 - M_2)$.
- The $(k(N), r(N))$-Remote Matrix Problem (RMP) is defined as follows:
  - INPUT: A collection of matrices $M_1, M_2, \ldots, M_k \in \mathbb{F}^{N \times N}$.
  - SOLUTION: A matrix $M \in \mathbb{F}^{N \times N}$ such that $\Delta_{\text{rank}}(M - M') \geq r$ for each $M' \in \text{span}(M_1, M_2, \ldots, M_k)$.
- Easy parameters: The $(k, N/(k+1))$-RMP has an easy solution.
- Interesting parameters: $k = N^2/10$, $r = N/10$. Random point is a solution w.h.p..

The Remote Matrix Problem (the RPP with rank metric)

- Let $\Delta_{\text{rank}}(M_1, M_2) = \text{rank}(M_1 - M_2)$.
- The $(k(N), r(N))$-Remote Matrix Problem (RMP) is defined as follows:
  - INPUT: A collection of matrices $M_1, M_2, \ldots, M_k \in \mathbb{F}^{N \times N}$.
  - SOLUTION: A matrix $M \in \mathbb{F}^{N \times N}$ such that $\Delta_{\text{rank}}(M - M') \geq r$ for each $M' \in \text{span}(M_1, M_2, \ldots, M_k)$.
- Easy parameters: The $(k, N/(k+1))$-RMP has an easy solution.
- Interesting parameters: $k = N^2/10$, $r = N/10$. Random point is a solution w.h.p..

Results

Lemma

The $(k, N/(k+1))$-RMP can be solved in polynomial time.

Theorem

There is an explicit lower bound F against ABPs using H if:

- *H is not too large.*
- *H is a set of help polynomials with minimum degree $\geq d(1/2+\varepsilon)$.*

Theorem

If the $(k, N/k^{1/2-\varepsilon})$-RMP can be solved in polynomial time, then there is an explicit lower bound F against ABPs using H, for any H that is not too large.

Results

Lemma

The $(k, N/(k+1))$-RMP can be solved in polynomial time.

Theorem

There is an explicit lower bound F against ABPs using H if:

- *H is not too large.*
- *H is a set of help polynomials with minimum degree $\geq d(1/2+\varepsilon)$.*

Theorem

If the $(k, N/k^{1/2-\varepsilon})$-RMP can be solved in polynomial time, then there is an explicit lower bound F against ABPs using H, for any H that is not too large.

Results

Lemma

The $(k, N/(k+1))$-RMP can be solved in polynomial time.

Theorem

There is an explicit lower bound F against ABPs using H if:

- *H is not too large.*
- *H is a set of help polynomials with minimum degree $\geq d(1/2+\varepsilon)$.*

Theorem

If the $(k, N/k^{1/2-\varepsilon})$-RMP can be solved in polynomial time, then there is an explicit lower bound F against ABPs using H, for any H that is not too large.

Other Results

Following the general proof structure of the result of Alon, Panigrahy, and Yekhanin's result on the RPP:

Theorem

The $(N, \log N)$-RMP can be solved in polynomial time, for constant-sized fields.

Outline

Summary

- We studied the computational model of constant-depth boolean circuits with help functions, and Noncommutative ABPs with help polynomials.
- We showed connections between the Help function problem and the problem of separating EXP from the polynomial-time many-one closure of SizeDepth($n^{O(1)}, O(1)$).
- We also showed connections between the Help function/polynomial problems and solving the Remote Point Problem in the Hamming and rank metrics respectively.
- The connection yields restricted lower bounds against ABPs using help polynomials.

Summary

- We studied the computational model of constant-depth boolean circuits with help functions, and Noncommutative ABPs with help polynomials.
- We showed connections between the Help function problem and the problem of separating EXP from the polynomial-time many-one closure of SizeDepth($n^{O(1)}, O(1)$).
- We also showed connections between the Help function/polynomial problems and solving the Remote Point Problem in the Hamming and rank metrics respectively.
- The connection yields restricted lower bounds against ABPs using help polynomials.

Summary

- We studied the computational model of constant-depth boolean circuits with help functions, and Noncommutative ABPs with help polynomials.
- We showed connections between the Help function problem and the problem of separating EXP from the polynomial-time many-one closure of SizeDepth$(n^{O(1)}, O(1))$.
- We also showed connections between the Help function/polynomial problems and solving the Remote Point Problem in the Hamming and rank metrics respectively.
- The connection yields restricted lower bounds against ABPs using help polynomials.

Summary

- We studied the computational model of constant-depth boolean circuits with help functions, and Noncommutative ABPs with help polynomials.
- We showed connections between the Help function problem and the problem of separating EXP from the polynomial-time many-one closure of SizeDepth($n^{O(1)}, O(1)$).
- We also showed connections between the Help function/polynomial problems and solving the Remote Point Problem in the Hamming and rank metrics respectively.
- The connection yields restricted lower bounds against ABPs using help polynomials.

Open questions

- Algorithms with better parameters for the RPP and RMP.
- Specific cases of the Help functions question:
  - Is there a small H such that $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contains all the parities?
  - If H contains only parities, then does $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contain the inner-product function?
- Connections between the ABP with help polynomials question and lower bounds against general noncommutative arithmetic circuits.

Open questions

- Algorithms with better parameters for the RPP and RMP.
- Specific cases of the Help functions question:
  - Is there a small H such that $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contains all the parities?
  - If H contains only parities, then does $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contain the inner-product function?
- Connections between the ABP with help polynomials question and lower bounds against general noncommutative arithmetic circuits.

Open questions

- Algorithms with better parameters for the RPP and RMP.
- Specific cases of the Help functions question:
  - Is there a small H such that $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contains all the parities?
  - If H contains only parities, then does $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contain the inner-product function?
- Connections between the ABP with help polynomials question and lower bounds against general noncommutative arithmetic circuits.

Open questions

- Algorithms with better parameters for the RPP and RMP.
- Specific cases of the Help functions question:
  - Is there a small H such that $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contains all the parities?
  - If H contains only parities, then does $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contain the inner-product function?
- Connections between the ABP with help polynomials question and lower bounds against general noncommutative arithmetic circuits.

Open questions

- Algorithms with better parameters for the RPP and RMP.
- Specific cases of the Help functions question:
  - Is there a small H such that $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contains all the parities?
  - If H contains only parities, then does $\mathrm{SizeDepth}_H(n^{O(1)}, O(1))$ contain the inner-product function?
- Connections between the ABP with help polynomials question and lower bounds against general noncommutative arithmetic circuits.

Thank you