1 Additive Combinatorics

Question [Erdos Turan 1936]: How big a set can you pick from the set $\{1, \ldots, N\}$ such that the set does not have three distinct numbers a, b, c that form an arithmetic progression? Arithmetic Progressions are triples (a, b, c) such that $a + b = 2c$ or $b - a = c - b$.

We define a set:

$$r_3(N) = \max_{S \subseteq [N]} |S| \text{ such that } S \text{ contains no 3-term arithmetic progression}$$

1.1 Early Strategies

1.1.1 Greedy

We pick the earliest number such that it does not create a 3-term arithmetic progression with the numbers already picked. For example, we select $\{1, 2, \_, 4, 5, \_, \_, ...\}$. This strategy gets roughtly $N^{0.6}$ numbers.

1.1.2 Random Set

Try selecting a random set of numbers. Formally, we select S : each element i is in S with probability P. We think about how many 3-arithmetic progressions there are in S. There are roughly N^2 arithmetic progressions in $[N]$. This is for any 2 elements a, b, c is fixed. Now, the probabiility that $(a, b, c) \in S$ is P^3. Therefore we have the following:

$$E[\#\text{ of 3-AP in S}] \leq N^2 P^3$$

If we set $P \approx N^{-1/3}$, then we have the expected number of 3-AP's in S is less than N (Where the N trivial APs are (a, a, a)). Furthermore, with this probability, the expected size of S is $NP = N^{2/3}$.

In both of these cases, the size of S is polynomially smaller than N.

1.2 History

Advances in this question are as follows:

- $r_3(N) \leq \frac{N}{\log \log N}$ - Roth 1953
- Heath, Borwn, Szemeredi 87-90: $r_3(N) \leq \frac{N}{(\log N)^c}$ for some small $c > 0$

- Bourgain 1999: $r_3(N) \leq \frac{N}{(\log N)^{2/3}}$
- Sanders 2011: $r_3(N) \leq \frac{N(\log\log N)^6}{\log N}$

Bloom-Sisask 2020: $r_3(N) \leq \frac{N}{\log N^{1+c}}$(For some small $c > 0$). Surprisingly, this implies that prime numbers contain infinitely many 3-term arithmetic progressions since the number of primes up to N is roughly $\frac{N}{\log N}$.

Beherd 1946: $r_3(N) \geq \frac{N}{2^{c\sqrt{\log N}}}$ for some small $c > 0$.

Meka Kelley 2023: $r_3(N) \leq \frac{N}{2^{c(\log N)^{1/12}}}$ for some small $c > 0$.

There are also $r_K(N)$: questions about k-term arithmetic progressions

1.3 Why 3-APs?

There are many reasons why this is an important question. There are implications on communication complexity and parallel repetition. It gives insight to two main important tools:

- Structure vs. Randomness
- Polynomial Method

2 Beherend 1946 Construction

First we will demonstrate the lemma proved by Behered in 1946.

Theorem 2.1. *For any N, $r_3(N) \geq \frac{N}{2^{c\sqrt{\log N}}}$*

We aim to find $S \subseteq [N]$ such that $|S|$ is as large as possible and for no three distinct elements $a, b, c \in S$ do they form an arithmetic progression.

Punchline: If we take a sphere and select two points on the sphere, the midpoint is inside the sphere but not on it. This is the key idea behind Beherend's construction. More formally, if you take the unit sphere in $d-$dimensions:

$$S^{d-1} = \{(x, ..., x_d) : \sum x_i^2 = 1\}$$

S^{d-1} does not contain distinct points x, y, z such that $x + y = 2z$.

The strategy is:

- Fix a range k, and dimension d.
- Pick $S \subseteq \{0, ..., k\}^d$
- $S_R = \{x \in [k]^d : x_1 + x_2 + ... + x_d^2 = R\}$

We fix a radius R and look at all points which have integer coordinates and lie on a shell of radius exactly R.

Claim: S_R has no distinct points x, y, z such that $x + y = 2z$.

Claim: One of these $R \leq k^2 d$ such that $|S_R| \geq \frac{k^d}{k^2 d}$.

Proof. We use an averaging argument. Given $[k]^d : k^d$ points, $x \leftarrow [k]^d$ has $\sum x_i^2 \leq d \cdot k^2$ (This is because the largest possible x_i is k and there are d coordinates that define a point). Furthermore, every summation must add up to an integer since squaring x_i integer values result in an integer value and summing integer values results in an integer value. So we know that for all possible radius values 1 to $d \cdot k^2$, at least one of those radius values has to have at least $k^d/(d \cdot k^2)$. This implies that:

$$\exists R \text{ such that } |S_R| \geq \frac{k^d}{k^2 \cdot d}$$

□

We now want to map these coordinates $x \in [k]^d$ to the integers such that the desired properties hold. Formally, we want to find a mapping $f : S_R \to \mathbb{Z}_N$ where the following holds:

$$f(x) + f(y) = 2 \cdot f(z) \Leftrightarrow x + y = 2z$$

2.1 Mapping the Integers to $[k]^d$

We are trying to map vectors to integers such that adding the integers corresponds to adding the vectors. To do this, we use bitvectors, in otherwords, we map the integers to binary representations of their numbers of d bits. For formally we have:

$$x = (x_1, x_2, ..., x_d)$$

Representing x as a base $2k + 1$ representation of a number. In otherwords:

$$f(x) = \sum_{i=1}^{d} x_i \cdot (2k + 1)^{i-1}$$

That's it. We prove that $x + y = 2z \iff f(x) + f(y) = 2 \cdot f(z)$

Proof. We demonstrate the equality:

$$\begin{aligned} x + y = 2 \cdot z \Rightarrow \\ f(x) + f(y) &= \sum_{i=1}^{d} (x_i + y_i)(2k+1)^{i-1} \\ 2 \cdot f(z) &= 2 \cdot \sum_{i=1}^{d} (z_i)(2k+1)^{i-1} \\ &= \sum_{i=1}^{d} (2 \cdot z_i)(2k+1)^{i-1} \end{aligned}$$

Since vector addition is isolated to each index i, it is clear that for all i, $2 \cdot z_i = x_i + y_i$, which demonstrates the correctness of the equality. □

We now know the mapping, and are tasked with finding the cardinality of the set $f(S_R)$.

2.2 Size of $f(S_R)$

We know that $f(S_R) \subseteq [N]$. Furthermore, since $x \in [k]^d$, then we find the maximum value for $f(x)$.

$$\begin{aligned} f(x) &= \sum_{i=1}^{d} x_i (2k+1)^{i-1} && (1) \\ &\leq \sum_{i=1}^{d} k \cdot (2k+1)^{i-1} && (2) \\ &= k \cdot \frac{(2k+1)^d - 1}{2k} && (3) \\ &\leq (2k+1)^d && (4) \end{aligned}$$

Since we know that all numbers are going to be less than $(2k+1)^d$, we set N to be equal to this value. Furthermore, from step 1 to step 2, we replace x_i with k since k is the max value that x_i can take. From step 2 to 3, we use the following geometric series formula:

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1} - 1}{r - 1}$$

More explicitly, we rewrite the term as:

$$k \cdot \sum_{i=0}^{d-1} (2k+1)^i$$

and use the series formula with $r = (2k+1)$. From this, we have that

$$|S_R| \geq \frac{k^d}{k^2 \cdot d} \approx 2^d \cdot k^2 \cdot d$$

The best values for k and d are to set $2k+1 = 2^{\sqrt{\log N}}$ and $d = \sqrt{\log N}$. We have the following.

$$\begin{aligned} |S_R| &\geq \frac{N}{2^d \cdot k^2 \cdot d} \\ &= \frac{N}{2^{\sqrt{\log N}} \cdot \frac{1}{4}(2^{\sqrt{\log N}} - 1)^2 \cdot \sqrt{\log N}} \\ &= \frac{N}{2^{O(\sqrt{\log N})}} \end{aligned}$$

2.3 Application To Communication Complexity

Consider two parties Alice and Bob. Alice has some integer $x \in [N]$ and Alice has integer $y \in [N]$. They try to compute some $f : [N] \times [N] \to \{0, 1\}$. Consider the function

$$cSum(x, y) = \begin{cases} 1 \text{ if } x + y = N \\ 0 \text{ otherwise.} \end{cases}$$

The communication complexity of $cSum$ must be at least $\log_2 N$, sending the entire string over.

We generalize this problem to three problems with parties Alice, Bob, and Charlie with inputs $x, y, z \in [N]$ respectively. If they want to compute some joint function we consider CheckSum defined as $f : [N] \times [N] \times [N] \to \{0, 1\}$ where:

$$cSum(x, y, z) = \begin{cases} 1 \text{ if } x + y + z = N \\ 0 \text{ otherwise.} \end{cases}$$

The communication complexity of checksum for 3 players is equal to $\Theta(\log_2 N)$, which implies you essentially have to send the entire input. However, consider this new game where each player, instead of having their inputs, have

their numbers on their forehead. This means they can see the other players inputs, but not their own (i.e Alice sees y, z but does not know her input x). We analyze the communication complexity of this new game we call Number-On Forehead or NOF model. For a function f such that

$$f : [N] \times [N] \times [N] \to \{0, 1\}$$

The NOF communication model, or $NOF(f)$ is the min # of bits needed to compute f in a NOF protocol. Firstly, it is clear that this model is stronger than the previous model since each indivdiual can just simulate one of the other players by running the protocol with the other player's input. The $NOF(cSum) = O(\sqrt{\log N})$, quadratic savings! This was done with Beherend's constructions.

2.3.1 3AP for $NOF(cSum)$

The main idea was showing the following:

$$NOF(cSum) \leq \log(\frac{N}{r_3(6N)})$$

In otherwords, if there exists a large $3AP$-free set, the NOF of $cSum$ is small. We know that by Beherend's construction:

$$r_3(6N) \approx \frac{N}{2^{c \cdot \sqrt{\log N}}}$$

Which gives us:

$$NOF(cSum) = \sqrt{\log N}$$

Quantitatively, if we show that $NOF(CheckSum) = \omega(1)$ implies that sets of constant density over $[N]$ have 3APs.

2.3.2 Construction

Consider a $3AP$ free set $S \subseteq [N]$ where $|S| = r_3(N)$.

Want: A coloring $x : \{1, ..., N\} \to [\Delta_N]$ such that no monochromatic 3APs exist:

$$\nexists a, b, c \text{ s.t } a + b = 2c \ \& \ x(a) = x(b) = x(c)$$

Claim: $\frac{N}{\Delta_N} \leq r_3(N) \leq O(1) \cdot \frac{N \cdot \log N}{\Delta_N}$

This implies that $\Delta_N \leq \frac{O(1)\cdot N\cdot \log N}{r_3(N)} = 2^{O(\sqrt{\log N})}$ So we can color the numbers that we can color the numbers with $2^{O(\log N)}$ colors such that there is no monochromatic 3AP.

2.3.3 Protocol

We say Alice has input x, Bob (y), and Charlie (z) and that they want to check if $x + y + z = N$. In other words they will check if $x + y + z - N = 0$ which we will refer to as $d = 0$. The protocol is as follows:

- Alice computes $-y - 2z$
- Bob computes $(-y - 2z) + d$
- Charlie finds $(-y - 2z) + 2d$

These numbers form a 3AP with difference d. However, these values can be negative so we add $3N$ to each value to keep that from happening. We rewrite Bob and Charlies values as follows

- Bob:

$$\begin{aligned}(-y - 2z) + d + 3N &= -y - 2z + x + y + z - N + 3N \\ &= -z + x + 2N\end{aligned}$$

- Charlie:

$$\begin{aligned}-y - 2z + 2d + 3N &= -y - 2z + 2x + 2y + 2z - 2N + 3N \\ &= y + 2x + N\end{aligned}$$

- Alice: $3N - y - 2z$

Now each player announces what color their values yield. The number of bits it takes depends on how many colors we have. There are $2^{\sqrt{\log N}}$ colors, so it takes $\sqrt{\log N}$ bits of communication. The full protocol is:

1. Announce colors as above
2. If all colors are the same, say $cSum = 1$ else say $cSum = 0$.

Why does this work? Because if $d = 0$, each player has the same number calculated from the protocol, so the color will be the same. If $d \neq 0$, then the three players have values that form a 3AP, however, we know that these values cannot have the same colors since the coloring says no three APs have the same color.

2.3.4 Summary

We showed that the $NOF(cSum)$ is $O(\sqrt{\log N})$. A lowerbound we have for this is $\Omega(\log \log \log N)$.