Some computational results on numbers of the form $p + 2^k$ and $p + F_n$

Joint work with Huixi Li and Xiangjun Dai

Yuda Chen

May 5, 2024

School of Mathematical Sciences, Nankai University

On numbers of the form $p + 2^k$

History of research on problems related to $p + 2^k$

$$d(N) = \frac{|\{n \le N : n = p + 2^k, p \text{ prime}\}|}{N}.$$

$$\underline{d} = \liminf_{n\to\infty} d(n) \text{ and } \overline{d} = \limsup_{n\to\infty} d(n)$$

- **In 1934**, Romanoff proved that $\underline{d} > 0$.
- **In 2004**, Chen and Sun improved it to $\underline{d} > 0.0868$.
- **In 2006**, Pintz found $\underline{d} > 0.093626$.
- **In 2010**, Habsieger and Sivak-Fischler obtained $\underline{d} > 0.0936275$.
- **In 2018**, Elsholtz and Schlage-Puchta achieved the best result so far: $\underline{d} > 0.107648$.

History of research on problems related to $p + 2^k$

$$d(N) = \frac{|\{n \leq N : n = p + 2^k, p \text{ prime}\}|}{N}.$$

$$\underline{d} = \liminf_{n\to\infty} d(n) \text{ and } \overline{d} = \limsup_{n\to\infty} d(n)$$

- **In 1849**, de Polignac found that 127 and 959 can't be written as a sum of a prime number and a power of 2.
- **In 1950**, van der Corput demonstrated in 1950 that odd integers not of the form $p + 2^k$ possess a positive density.
- **In 1950**, Erdős proved that $\overline{d} \leq 0.49999991$.
- **In 2006**, Habsieger and Roblot improved it to $\overline{d} < 0.4909$.
- **In 2024, Chen, Dai, and Li achieved the best result so far: $\overline{d} < 0.490341088858244$.**

南开大学 Nankai University

History of research on problems related to $p + 2^k$

$$d(N) = \frac{|\{n \le N : n = p + 2^k, p \text{ prime}\}|}{N}.$$

$$\underline{d} = \liminf_{n\to\infty} d(n) \text{ and } \overline{d} = \limsup_{n\to\infty} d(n)$$

Conjecture (Romani, 1983)

$d = \overline{d} = \underline{d}$.

Conjecture (Gianna del Corso, Ilaria del Corso, Dvornicich, and Romani, 2020)

If the above conjecture is true, $d \approx 0.437$.

The original algorithm by Habsieger and Roblot (2006)

$$f_M(\overline{m}) = \left\{\overline{k} \in \mathbb{Z}/\mathrm{ord}_2(M)\mathbb{Z} : \overline{m} - 2^{\overline{k}} \in (\mathbb{Z}/M\mathbb{Z})^*\right\}$$

$$g_M(I) = \{\overline{m} \in \mathbb{Z}/M\mathbb{Z} : f_M(\overline{m}) = I\}$$

$$G_M(I) = |g_M(I)| \text{ for} I \subset \mathbb{Z}/\mathrm{ord}_2(M)\mathbb{Z}$$

For a prime p, we have $I_{p,0} = \mathbb{Z}/\mathrm{ord}_2(p)\mathbb{Z}$ with $G_p(I_{p,0}) = p - \mathrm{ord}_2(p)$ and $I_{p,\bar{j}} = (\mathbb{Z}/\mathrm{ord}_2(p)\mathbb{Z}) \setminus \left\{\bar{j}\right\}$ with $G_p(I_{p,\bar{j}}) = 1$ for each $\bar{j} \in \mathbb{Z}/\mathrm{ord}_2(p)\mathbb{Z}$.

$$G_{M_1}(I_1) = \sum_{I_1 = \tilde{I}_2 \cap \tilde{I}_p} G_{M_2}(I_2)\, G_p(I_p),\ \delta_M(\nu) = \sum_{|I| = \nu} G_M(I)$$

$$\overline{d} \leq \sum_{\nu=0}^{\mathrm{ord}_2(M)} \delta_M(\nu) \min\left(\frac{1}{M}, \frac{2\nu}{\mathrm{ord}_2(M)\varphi(M)\log 2}\right).$$

The explanation of algorithm by Habsieger and Roblot (2006)

The simplest case: $3 \times 5 \quad \mathrm{ord}_2(3) = 2, \mathrm{ord}_2(5) = 4.$

$I_1 = \tilde{I}_2 \cap \tilde{I}_p$; $\tilde{I}_2$ (columns) ; $\tilde{I}_p$ (rows)	{0}	{1}	{0,1}
{0,1,2}			
{0,1,3}			
{0,2,3}			
{1,2,3}			
{0,1,2,3}			

Nankai University

The explanation of algorithm by Habsieger and Roblot (2006)

The simplest case: $3 \times 5 \quad \mathrm{ord}_2(3) = 2, \mathrm{ord}_2(5) = 4.$

$I_1 = \tilde{I}_2 \cap \tilde{I}_p$; $\tilde{I}_2$ → ; $\tilde{I}_p$ ↓	{0,2}	{1,3}	{0,1,2,3}
{0,1,2}			
{0,1,3}			
{0,2,3}			
{1,2,3}			
{0,1,2,3}			

The explanation of algorithm by Habsieger and Roblot (2006)

The simplest case: $3 \times 5 \quad \mathrm{ord}_2(3) = 2, \mathrm{ord}_2(5) = 4.$

$I_1 = \tilde{I}_2 \cap \tilde{I}_p$ ($\tilde{I}_p$ \ $\tilde{I}_2$)	{0,2}	{1,3}	{0,1,2,3}
{0,1,2}	{0,2}	{1}	{0,1,2}
{0,1,3}	{0}	{1,3}	{0,1,3}
{0,2,3}	{0,2}	{3}	{0,2,3}
{1,2,3}	{2}	{1,3}	{1,2,3}
{0,1,2,3}	{0,2}	{1,3}	{0,1,2,3}

Nankai University

The explanation of algorithm by Habsieger and Roblot (2006)

The simplest case: $3 \times 5 \quad \text{ord}_2(3) = 2, \text{ord}_2(5) = 4.$

$I_1 = \tilde{I}_2 \cap \tilde{I}_p$ ($\tilde{I}_p$ \\ $\tilde{I}_2$)	{0,2}	{1,3}	{0,1,2,3}
{0,1,2}	{0,2}	{1}	{0,1,2}
{0,1,3}	{0}	{1,3}	{0,1,3}
{0,2,3}	{0,2}	{3}	{0,2,3}
{1,2,3}	{2}	{1,3}	{1,2,3}
{0,1,2,3}	{0,2}	{1,3}	{0,1,2,3}

$$
\begin{aligned}
\overline{d} \le & \sum_{\nu=0}^{\text{ord}_2(15)} \delta_{15}(\nu) \min\left(\frac{1}{15}, \frac{2\nu}{\text{ord}_2(15)\varphi(15)\log 2}\right) \\
= & 0 + 4\min\left(\frac{1}{15}, \frac{2\times 1}{\text{ord}_2(15)\varphi(15)\log 2}\right) + 6\min\left(\frac{1}{15}, \frac{2\times 2}{\text{ord}_2(15)\varphi(15)\log 2}\right) \\
& + 4\min\left(\frac{1}{15}, \frac{2\times 3}{\text{ord}_2(15)\varphi(15)\log 2}\right) + \min\left(\frac{1}{15}, \frac{2\times 4}{\text{ord}_2(15)\varphi(15)\log 2}\right)
\end{aligned}
$$

The explanation of algorithm by Habsieger and Roblot (2006)

The simplest case: $3 \times 5 \quad \mathrm{ord}_2(3) = 2, \mathrm{ord}_2(5) = 4.$

$I_1 = \tilde{I}_2 \cap \tilde{I}_p$ ($\tilde{I}_p$ \ $\tilde{I}_2$)	{0,2}	{1,3}	{0,1,2,3}
{0,1,2}	{0,2}	{1}	{0,1,2}
{0,1,3}	{0}	{1,3}	{0,1,3}
{0,2,3}	{0,2}	{3}	{0,2,3}
{1,2,3}	{2}	{1,3}	{1,2,3}
{0,1,2,3}	{0,2}	{1,3}	{0,1,2,3}

$$
\begin{aligned}
\overline{d} \leq & \sum_{\nu=0}^{\mathrm{ord}_2(15)} \delta_{15}(\nu) \,\mathbf{min}\left(\frac{1}{15}, \frac{2\nu}{\mathrm{ord}_2(15)\varphi(15)\,\mathbf{log}\,2}\right) \\
= & 0 + 4\,\mathbf{min}\left(\frac{1}{15}, \frac{2\times 1}{\mathrm{ord}_2(15)\varphi(15)\,\mathbf{log}\,2}\right) + 6\,\mathbf{min}\left(\frac{1}{15}, \frac{2\times 2}{\mathrm{ord}_2(15)\varphi(15)\,\mathbf{log}\,2}\right) \\
& + 4\,\mathbf{min}\left(\frac{1}{15}, \frac{2\times 3}{\mathrm{ord}_2(15)\varphi(15)\,\mathbf{log}\,2}\right) + \mathbf{min}\left(\frac{1}{15}, \frac{2\times 4}{\mathrm{ord}_2(15)\varphi(15)\,\mathbf{log}\,2}\right)
\end{aligned}
$$

The explanation of algorithm by Habsieger and Roblot (2006)

Another simple case: $3 \times 5 \times 7 \quad \mathrm{ord}_2(3 \times 5) = 4, \mathrm{ord}_2(7) = 3.$

	$I_1 = \tilde{I}_2 \cap \tilde{I}_p$ \ $\tilde{I}_p$	1	1	1	4
	$\tilde{I}_2$	{0,1}	{0,2}	{1,2}	{0,1,2}
1	{0}				
1	{1}				
1	{2}				
1	{3}				
3	{0,2}				
3	{1,3}				
1	{0,1,2}				
1	{0,1,3}				
1	{0,2,3}				
1	{1,2,3}				
1	{0,1,2,3}				

The explanation of algorithm by Habsieger and Roblot (2006)

Another simple case: $3 \times 5 \times 7 \quad \mathrm{ord}_2(3 \times 5) = 4, \mathrm{ord}_2(7) = 3.$

	$I_1 = \tilde{I}_2 \cap \tilde{I}_p$; $\tilde{I}_p$; $\tilde{I}_2$	1	1	1	4
		{0,1,3,4,6,7,9,10}	{0,2,3,5,6,8,9,11}	{1,2,4,5,7,8,10,11}	{0,1,2,3,4,5,6,7,8,9,10,11}
1	{0,4,8}				Quadruple
1	{1,5,9}				Quadruple
1	{2,6,10}				Quadruple
1	{3,7,11}				Quadruple
3	{0,2,4,6,8,10}	Triple	Triple	Triple	Duodecuple
3	{1,3,5,7,9,11}	Triple	Triple	Triple	Duodecuple
1	{0,1,2,4,5,6,8,9,10}				Quadruple
1	{0,1,3,4,5,7,8,9,11}				Quadruple
1	{0,2,3,4,6,7,8,10,11}				Quadruple
1	{1,2,3,5,6,7,9,10,11}				Quadruple
1	{0,1,2,3,4,5,6,7,8,9,10,11}				Quadruple

Trying to enhancing the algorithm

The efficiency of the above algorithm is too low—calculating the result of M=$31 \times 19 \times 17 \times 13 \times 11 \times 7 \times 5 \times 3$ takes over about 40 minutes.

Firstly, we divided it into 2 groups and compute the results of them respectively. This is because we don't need to compute that in some special order. Then we used $0-1$ matrix to improve running speed. By using Python's Numpy package to enabling multi-core CPU computing, it can accelerate the speed by hundreds of times, even one thousand times in some cases.

Trying to enhancing the algorithm

By using the above methods, although solving problems of running speed, the memory usage is substantial. When computing the upper bound for $\overline{d}$ corresponding to the set of 10 primes $\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73\}$, the required memory has surpassed 64GB.

Dai suggests that, to conserve memory and expedite the process, we can introduce a new single-column matrix, known as the multiplicity matrix, to store the multiplicity of each row's corresponding set in the family of sets. Consequently, when conducting the Hadamard product on two rows, we only need to multiply the corresponding elements in the multiplicity matrix. Furthermore, after each intersection operation between clusters of sets is completed, a deduplication operation is executed.

Trying to enhancing the algorithm

With these improvements, it enables the calculation of the upper bound for $\overline{d}$ corresponding to the set of 12 primes

$$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 733, 241, 257\}$$

in just 24 minutes on a computer with 16GB of memory and an i9-13980HX CPU.

Notation: According to Habsieger and Roblot's paper (2006), they only spent 35 minutes computing it on an Intel Xeon 2.4 GHz processor with a memory stack of 2.1GB.

Utilizing GPU computing to further improve efficiency

By utilizing GPU computation implemented through Python's CuPy package, we can conserve approximately one-third of the memory and significantly increase the speed, with the exact rate depending on the GPU performance. On a computer with 16GB of memory, an i9-13980HX CPU, and an RTX4070 Laptop GPU, the computation with the set of 12 primes

$$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73, 241, 257\}$$

considered by Habsieger and Roblot can be completed in just 3 minutes.

Our results

Theorem

We have $\overline{d} < 0.490341088858244$.

With above enhanced algorithm, on a platform with 1536GB of memory, two E5-2697v2 CPUs, and a V100 GPU, it took approximately 167 hours to obtain $\overline{d} < 0.490341088858244$ when using the set

$$\{3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 61, 73\}$$

for our computation.

Why choosing this set?

Our results

We begin by analysing some data for the set

$$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73, 241, 257\}$$

used in Habsieger and Roblot's paper (2006). We list the process of expanding the set 3 to include the primes mentioned, adding them one by one, along with the corresponding decrease in the upper bound estimates for $\overline{d}$.

Our results

Set of primes	Estimates for $\overline{d}$	Improvements
$\{3\}$	0.5	
$\{3, 5\}$	0.5	0
$\{3, 5, 7\}$	0.5	0
$\{3, 5, 7, 11\}$	0.49807089	0.00192911
$\{3, 5, 7, 11, 13\}$	0.49621815	0.00185274
$\{3, 5, 7, 11, 13, 17\}$	0.49252410	0.00369405
$\{3, 5, 7, 11, 13, 17, 19\}$	0.49185782	0.00066628
$\{3, 5, 7, 11, 13, 17, 19, 31\}$	0.49143385	0.00042397
$\{3, 5, 7, 11, 13, 17, 19, 31, 41\}$	0.49115839	0.00027546
$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73\}$	0.49107930	0.00007909
$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73, 241\}$	0.49098557	0.00009373
$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73, 241, 257\}$	0.49089834	0.00008723

Our results

$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73, 241, 257\}$	0.49089834	*control group*
$\{3, 5, 7, 11, 13, 17, 19, 23, 31, 41, 73, 241, 257\}$	0.49069465	*added* 23
$\{3, 5, 7, 11, 13, 17, 19, 31, 37, 41, 73, 241, 257\}$	0.49062325	*added* 37
$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 61, 73, 241, 257\}$	0.49074788	*added* 61
$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73, 127, 241, 257\}$	0.49081180	*added* 127

To decrease $\overline{d}$, we need to choose a set with more prime numbers and make the product smaller. According to the data above, we replaced 257 with 37.

Our results

$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73, 241, 257\}$	0.49089834	
$\{3, 5, 7, 11, 13, 17, 19, 31, 37, 41, 73, 241\}$	0.49070248	*control group*
$\{3, 5, 7, 11, 13, 17, 19, 23, 31, 37, 41, 73, 241\}$	0.49060796	*added* 23
$\{3, 5, 7, 11, 13, 17, 19, 31, 37, 41, 61, 73, 241\}$	0.49056186	*added* 61

According to the data above, we added 61.

南開大學 Nankai University

Our results

$\{3, 5, 7, 11, 13, 17, 19, 31, 37, 41, 61, 73, 241\}$	0.49056186	control group
$\{3, 5, 7, 11, 13, 17, 19, 29, 31, 37, 41, 61, 73\}$	0.49041415	replaced 241 with 29
$\{3, 5, 7, 11, 13, 17, 19, 31, 37, 41, 53, 61, 73\}$	0.49060353	replaced 241 with 53
$\{3, 5, 7, 11, 13, 17, 19, 23, 31, 37, 41, 61, 73\}$	0.49062494	replaced 241 with 23

To make the product smaller, according to the data above, we replaced 241 with 29.

Our results

Since the effects of replacing 241 with 23 and 53 are similar, while $23 < 53$ and $\mathrm{ord}_2(23) < \mathrm{ord}_2(53)$, we decide to add 23 to save computation time, resulting in the following outcome.

$\{3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 61, 73\}$	0.490341088858244

Is there a general scheme for finding the set of prime numbers which can generate the best result?

南開大學 Nankai University

Some counterexamples

First, if we choose the set consisting of the first 12 odd prime numbers, then this gives the best result among all choices of sets with 12 odd primes we have tested.

$\{3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41\}$	0.49064273

This leads us to believe that the set of first m odd primes will generate the best result among all choices of sets with m odd primes.

However, we have the following counterexample.

Counterexample

For certain values of m, some set $\mathfrak{P}$ with m odd primes, which is different from the set of the first m odd primes, may yield a better upper bound estimate for $\overline{d}$.

$\{3, 5, 7, 11, 13, 17\}$	0.49252410448328
$\{3, 5, 7, 13, 17, 241\}$	0.49243452466582

The reason of this is mainly because $3, 5, 7, 13, 17, 241$ are all prime factors of $2^{24} - 1$, while $\text{ord}_2(3 \times 5 \times 7 \times 11 \times 13 \times 17) = 120$.

Some counterexamples

Therefore, we can guess if $|\mathfrak{P}_1| = |\mathfrak{P}_2|$ and $\mathrm{ord}_2(\prod_{p\in\mathfrak{P}_1} p) < \mathrm{ord}_2(\prod_{p\in\mathfrak{P}_2} p)$, then $\mathfrak{P}_1$ generates a better result than $\mathfrak{P}_2$. Unfortunately, this is still incorrect.

Counterexample

For some sets of primes $\mathfrak{P}_1$ and $\mathfrak{P}_2$ with $|\mathfrak{P}_1| = |\mathfrak{P}_2|$ and $ord_2(\prod_{p\in\mathfrak{P}_1} p) < ord_2(\prod_{p\in\mathfrak{P}_2} p)$, the upper bound for $\overline{d}$ generated from $\mathfrak{P}_1$ may not necessarily be superior to what $\mathfrak{P}_2$ produces.

$\{3, 5, 7, 11, 13, 31, 41, 61, 151, 331, 1321\}$	0.49431157054919	*base 2 order* $= 60$
$\{3, 5, 7, 11, 13, 17, 19, 31, 41, 73, 241\}$	0.49098556503467	*base 2 order* $= 360$

Some counterexamples

Finally, we found the following counterexample.

Counterexample

For two sets of primes $\mathfrak{P}$ and $\mathfrak{Q}$, if among all $q_i \in \mathfrak{Q}$, the best two results of the upper bounds generated from $\mathfrak{P} \bigcup \{q_i\}$ are $\mathfrak{P} \bigcup \{q_1\}$ and $\mathfrak{P} \bigcup \{q_2\}$, then the upper bound generated from $\mathfrak{P} \bigcup \{q_1, q_2\}$ is not necessarily the best among the upper bounds generated from $\mathfrak{P} \bigcup_{q_1, q_j \in \mathfrak{Q}} \{q_i, q_j\}$. For example,

$\{3, 5, 7, 11, 17, 19\}$	0.494609133024577	*best*
$\{3, 5, 7, 11, 17, 23\}$	0.494870288038247	*second best*
$\{3, 5, 7, 11, 17, 29\}$	0.494883239281366	*worst*

However, according to the table below, the result generated from $\{3, 5, 7, 11, 17, 19, 23\}$ is not the best.

$\{3, 5, 7, 11, 17, 19, 23\}$	0.494486144723180	
$\{3, 5, 7, 11, 17, 19, 29\}$	0.494213278918742	*best*
$\{3, 5, 7, 11, 17, 23, 29\}$	0.494618822711737	

Arithmetic progression found by Erdős

Theorem (Erdős, 1950)

$7629217 \pmod{11184810}$ *does not contain integers of the form* $p + 2^k$.

For every positive integer k, at least one of the following conditions must be satisfied:

$$\begin{cases} k \equiv 0 & \pmod{3} \\ k \equiv 1 & \pmod{4} \\ k \equiv 3 & \pmod{8} \\ k \equiv 7 & \pmod{12} \\ k \equiv 23 & \pmod{24} \\ k \equiv 0 & \pmod{2} \end{cases} \implies \begin{cases} 2^k \equiv 1 & \pmod{7} \\ 2^k \equiv 2 & \pmod{5} \\ 2^k \equiv 2^3 & \pmod{17} \\ 2^k \equiv 2^7 & \pmod{13} \\ 2^k \equiv 2^{23} & \pmod{241} \\ 2^k \equiv 1 & \pmod{3} \end{cases}$$

Arithmetic progression found by Erdős

So if integer x satisfies the following conditions, there must be $x - 2^k = mp, m \in \mathbb{Z}, p \in \{3, 5, 7, 13, 17, 241\}$:

$$\begin{cases} x - 2^0 \equiv 0 \pmod 7 \\ x - 2^1 \equiv 0 \pmod 5 \\ x - 2^3 \equiv 0 \pmod{17} \\ x - 2^7 \equiv 0 \pmod{13} \\ x - 2^{23} \equiv 0 \pmod{241} \\ x - 2^0 \equiv 0 \pmod 3 \end{cases}$$

By Chinese Reminder Theorem, we have

$$x \equiv 7629217 \pmod{11184810}$$

And after verification, $x - 2^k \neq 3, 5, 7, 13, 17, 241$, then it can't be prime.

The construction method of arithmetic progressions

Note the connection between $\mathbb{Z}-$covering systems and arithmetic progressions, we only need to find $\mathbb{Z}-$covering systems.

The following are some methods that use this method to generate arithmetic progressions:

D	$\{d_1, d_2, \cdots, d_n\}$	$\{m_1, m_2, \cdots, m_n\}$	corresponding *arithmetic progressions* $a \pmod b$
36	{2,3,4,9,12,18,36}	{1,2,3,8,11,17,35}	$309547193 \pmod{412729590}$
48	{2,4,6,8,16,24,48}	{1,2,0,0,4,4,44}	$13982215829 \pmod{21448163730}$
60	{2,3,4,5,10,12,15,20,30,60}	{0,1,3,3,5,9,11,17,29,59}	$520864019678683 \pmod{2520047004605130}$
72	{2,4,6,8,18,24,36,72}	{1,2,4,6,16,22,34,70}	$12878054009 \pmod{44153328030}$
80	{2,4,5,8,10,16,20,40,80}	{0,1,3,3,7,7,15,31,79}	$154854279578189723614177 \pmod{483570327845851669882470}$

Are there infinitely many such arithmetic progressions?

The construction method of arithmetic progressions

Theorem (Bang, 1886)

For any integer $m > 1$ and $m \neq 6$, there exists a prime p such that p divides $2^m - 1$ and p does not divide $2^{\tilde{m}} - 1$ for any $\tilde{m} < m$.

Theorem

Let $\{m_1 \pmod{d_1}, m_2 \pmod{d_2}, \cdots, m_n \pmod{d_n}\}$ be a minimal covering system with distinct moduli such that $lcm(d_1, d_2, \cdots, d_n) = D$. Then $\{1 \pmod{2}, 2m_1 \pmod{2d_1}, 2m_2 \pmod{2d_2}, \cdots, 2m_n \pmod{2d_n}\}$ is a minimal covering system with distinct moduli such that $lcm(2, 2d_1, 2d_2, \cdots, 2d_n) = 2D$.

南開大學 Nankai University

The smallest tolerance of arithmetic progressions with the property

Conjecture (Yonggao Chen, 2023)

If an arithmetic progression $a \pmod{b}$ does not contain numbers of the form $p + 2^k$, then $b \geq 11184810$.

The conjecture above has been proved by computation in our paper (2024). Moreover, if an arithmetic progression $a \pmod{11184810}$ does not contain numbers of the form $p + 2^k$, then it is one of the 48 arithmetic progressions.

Proving the minimization of tolerances by computation

Consider Dirichlet's theorem, we can shift numbers which are coprime with the modulo. For case of 10:

All odds in sense of modulo 10:

1 3 5 7 9

All numbers coprime with 10:

1 3 7 9

Shifted by 2^1:

1	3	5		9

Shifted by 2^2:

1	3	5	7	

The last 2 rows cover all odds in sense of modulo 10, so 10 is a counterexample.

Nankai University

Proving the minimization of tolerances by computation

Using this method, we checked all even integers less than 11184810. The execution time is approximately 5 hours, confirming a positive answer to Chen's.

Proving the minimization of tolerances by computation

Number even integers up to 11184808 sifted such that a given number of shifts to cover:

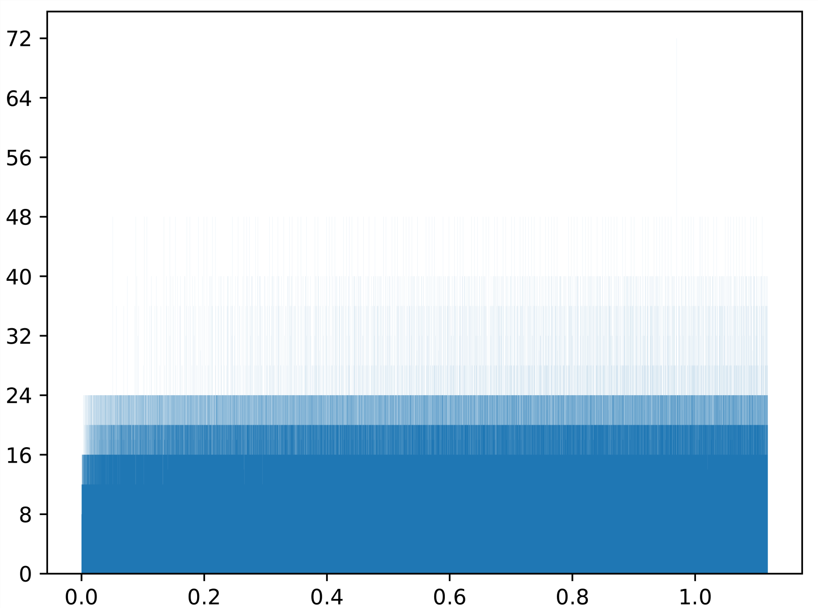
1 23	2 819307	3 1656377	4 1239765	5 295021	6 850127	7 7737	8 364486
9 6984	10 105003	11 838	12 176378	13 0	14 6195	15 5	16 27723
17 0	18 1169	19 0	20 22003	21 0	22 35	23 0	24 11441
25 0	26 0	27 0	28 578	29 0	30 7	31 0	32 130
33 0	34 0	35 0	36 496	37 0	38 0	39 0	40 434
41 0	42 0	43 0	44 0	45 0	46 0	47 0	48 141
49 0	50 0	51 0	52 0	53 0	54 0	55 0	56 0
57 0	58 0	59 0	60 0	61 0	62 0	63 0	64 0
65 0	66 0	67 0	68 0	69 0	70 0	71 0	72 1

Proving the minimization of tolerances by computation

From table we see more than 95% of the even integers from 2 to 11184818 are sifted with no more than 10 shifts, all but one even integer 9699690 are sifted with at most 48 shifts, while 9699690 is sifted with 72 shifts. For the extreme values, our data shows that the 23 numbers corresponding to the number 1 are 2^i, where $1 \leq i \leq 23$. The single number corresponding to 72 is $9699690 = 2 \times 3 \times 5 \times 7 \times 11 \times 13 \times 17 \times 19$ with 8 distinct prime divisors. The 141 numbers corresponding to 48 include $510510, 881790, \cdots, 11091990$. They are not necessarily square-free, but they all have 7 distinct prime divisors.

南開大學 Nankai University

Proving the minimization of tolerances by computation



On numbers of the form $p + F_n$

History of research on problems related to $p + F_n$

Theorem (Šiurys, 2016)

$2019544239293395 \pmod{2111872080374430}$ *does not contain integers of the form* $p^\alpha \pm F_n$.

Notice that, period of Fibonacci number mod 2111872080374430 is 360.

Our research on problems related to $p + F_n$

We start by checking numbers with period of 720. And we found the following conclusion.

Theorem

$208641 \pmod{17160990}, 218331 \pmod{17160990}, 520659 \pmod{17160990} \cdots (110 intotal)$ *does not contain integers of the form* $p + F_n$.

Our research on problems related to $p + F_n$

Soon we found that there exist arithmetic progressions with tolerance of $312018 (= 17160990/55)$.

Theorem

$208641 \pmod{312018}, 218331 \pmod{312018}$ *does not contain integers of the form* $p + F_n$.

This makes us think about whether 312018 is the smallest tolerance of arithmetic progressions with this property.

Proving the minimization of tolerances by computation

Similar to the case of $p + 2^k$, because of the modular periodicity of Fibonacci numbers, we can also use Dirichlet's theorem, deal with it by shifting numbers.

Proving the minimization of tolerances by computation

For case of 10:

All numbers in sense of modulo 10:

0 1 2 3 4 5 6 7 8 9

All numbers coprime with 10:

1 3 7 9

Shifted by $0 = F_0$:

0	1	2	3	4	5	6	7	8	9
	1		3				7		9

Shifted by $1 = F_1$:

0	1	2	3	4	5	6	7	8	9
0		2		4				8	

Shifted by $2 = F_3$:

0	1	2	3	4	5	6	7	8	9
	1		3		5				9

Shifted by $3 = F_4$:

0	1	2	3	4	5	6	7	8	9
0		2		4		6			

The last 4 rows cover all numbers in sense of modulo 10, so 10 is a counterexample.

南开大学 Nankai University

Proving the minimization of tolerances by computation

Theorem

Let $\mathcal{S}$ be the set of positive odd integers not of the form $p + F_n$. We have $208641 \pmod{312018}$ and $218331 \pmod{312018}$ are the only two arithmetic progressions in $\mathcal{S}$ with modulus 312018.

Explanation from the perspective of covering system

We claim that the arithmetic progression $208641 \pmod{312018}$ corresponds to the covering system.

Since the remainders of the Fibonacci numbers modulo 2 are $1, 1, 0, 1, 1, 0, \cdots$ with period 3, we know $n \equiv 1$ or $2 \pmod 3$ is equivalent to $F_n \equiv 1 \pmod 2$. So when $x \equiv 1 \pmod 2$ and $n \equiv 1$ or $2 \pmod 3$, we have $x - F_n \equiv 0 \pmod 2$. Similarly, when $x \equiv 0 \pmod 3$ and $n \equiv 4$ or $8 \pmod 8$, we have $x - F_n \equiv 0 \pmod 3$, etc.

Explanation from the perspective of covering system

$1, 2 \pmod{3}$	$1 \pmod{2}$
$4, 8 \pmod{8}$	$0 \pmod{3}$
$7, 9, 10, 14 \pmod{16}$	$6 \pmod{7}$
$0, 9, 18, 27 \pmod{36}$	$0 \pmod{17}$
$3, 8, 15 \pmod{18}$	$2 \pmod{19}$
$6, 18 \pmod{48}$	$8 \pmod{23}$

Therefore, for every term $x \in 208641 \pmod{312018}$ and every $n \geq 0$, we have $x - F_n$ is divisible by some prime $p \in \{2, 3, 7, 17, 19, 23\}$. This shows the arithmetic progression $208641 \pmod{312018}$ corresponds to the covering system constructed above.

Explanation from the perspective of covering system

The case of the arithmetic progression $218331 \pmod{312018}$ is similar to $208641 \pmod{312018}$, so we just present the following table and omit the explanation.

$1, 2 \pmod{3}$	$1 \pmod{2}$
$4, 8 \pmod{8}$	$0 \pmod{3}$
$1, 2, 6, 15 \pmod{16}$	$1 \pmod{7}$
$0, 9, 18, 27 \pmod{36}$	$0 \pmod{17}$
$3, 8, 15 \pmod{18}$	$2 \pmod{19}$
$30, 42 \pmod{48}$	$15 \pmod{23}$

Nankai University

Any Questions?

math.yuda.chen@gmail.com

Thanks!

math.yuda.chen@gmail.com