

素数等差数列的最小参数计算与若干新序列的发现

曾飞戈

(湖南人文科技学院, 中国)

摘要

素数等差数列 (AP) 是数论的核心研究对象。尽管格林-陶定理断言了任意长 AP 的存在性, 但构造具有**极小参数** (最小首项或最小公差) 的具体实例仍是重大的计算挑战。本文提出一种新颖的系统性搜索算法, 在 $\pi(5 \times 10^8)$ 范围内的素数中展开大规模计算, 旨在发现特定长度 k 的、参数极小的素数等差数列。

主要发现如下: (1) **刷新了短序列的最小首项纪录**: 找到了首项低至 11 的 8 项与 9 项 AP, 以及首项为 19 的 10 项 AP; (2) **实现了中等长度序列的突破**: 首次构造出首项为 503,213 的 14 项 AP; (3) **发现了长序列的极小公差解**: 获得了公差为 9,699,690 的 16 项 AP, 该值仅为理论最小基础公差的 323 倍, 是当
前已知的该长度最小公差解。本研究共确立了 15 项最优解, 为素数分布的结构性提供了新的具体证据。

关键词: 素数等差数列; 最小首项; 最小公差; 最优解; 计算数论; 搜索算法;
格林-陶定理; 基础公差

1. 引言

素数等差数列 (Arithmetic Progression, AP) 是指由素数构成的等差数列, 其形式为 $P, P+d, P+2d, \dots, P+(k-1)d$, 其中 P 为首项, d 为公差, k 为项数。对素数等差数列的研究历史悠久。1770 年, 拉格朗日证明了, 在公差为 2 的三素数等差数列 (即三元组 $(p, p+2, p+4)$) 中, 必有一项能被 3 整除, 唯一的例外是 $(3, 5, 7)$ 。这一结论揭示了素数分布的内在约束, 也预示着寻找长素数等差数列的困难性。

1923 年, 哈代 (G. H. Hardy) 和李特尔伍德 (J. E. Littlewood) 在其著名的素数 k 元组猜想中, 预言了任意长度素数等差数列存在的可能性, 为这一问题提供了理论方向。

2004 年, 格林 (Ben Green) 与陶哲轩 (Terence Tao) 里程碑式地证明了格林-陶定理: 素数序列中包含任意长度的等差数列。这从理论上彻底解决了存在性问题。

然而, 格林-陶定理的证明是非构造性的, 它并未提供寻找这些长等差数列的具体方法。因此, 该定理在理论上取得突破后, 一个巨大的计算挑战便凸显出来: 如何有效地构造出这些序列, 并优化其参数——特别是寻找具有最小首项或最小公差的实例。这已成为计算数论领域一个重要的研究方向。

1.1 研究现状

寻找长素数等差数列的工作主要由大型分布式计算项目推动。其中, **PrimeGrid** 是该领域最著名的项目, 它通过组织全球志愿者利用空闲计算资源, 系统性地搜索包括素数等差数列在内的各种素数模式。其重要目标之一即是寻找最长的素数等差数列。当前的世界纪录是长度为 27 项的等差数列, 由 PrimeGrid 于 2019 年发现, 其首项与公差均极其巨大。

与寻找最长序列的“广度”追求不同，另一条研究路线专注于“深度”挖掘，即在特定长度 k 下，寻找**首项或公差**尽可能小的素数等差数列。这类**小参数研究**需要对特定范围内的素数进行更为精细和密集地搜索。目前，该领域的记录多由个人研究者（如 Jens Kruse Andersen）或小型研究组保持，并散见于 Chris Caldwell 维护的《The Prime Pages》等专业网站和学术论坛中。

1.2 本文的主要工作与创新点

鉴于小参数素数等差数列的系统性探索尚不充分，本文致力于填补这一研究空白。主要工作与创新点如下：

1.算法创新：提出并实现了一种全新的**二阶段搜索策略**，该算法能高效、系统性地寻找特定长度 k 下具有**最小首项或最小公差**的素数等差数列。

2.系统性计算：在固定的整数范围 $N=5\times 10^8$ （即 $\pi(N)=26,355,867$ ）内，对 $k=2$ 至 n （搜索结果证实 $n=16$ ）的素数等差数列进行了完备的最小参数搜索。

3.成果突破：成功确立了 **15** 项已知最优解，其中包括刷新了 9 项**最小首项**纪录与 6 项**最小公差**纪录。

4.序列首创：首次构造并报道了长度为 14 项的素数等差数列，填补了该长度下小参数序列的空白。

2. 定义与理论背景

1. 统一的关键定义

定义 2.1 (基础公差)

对于一个 k 项的素数等差数列, 令 P_n 为大于等于 k 的最小素数 (即满足 $P_{n-1} < k \leq P_n$)。其公差 d 必须是前 $n-1$ 个素数乘积的倍数。该乘积称为基础公差, 记为:

$$\#P_{n-1} = \prod_{i=1}^{n-1} P_i$$

其中 P_i 表示第 i 个素数。

示例:

当 $k=10$ 时, 大于等于 10 的最小素数是 $P_5=11$ (因为 $P_4=7 < 10 \leq P_5=11$) , 基础公差 $\#P_4 = 2 \times 3 \times 5 \times 7 = 210$ 。

当 $k=16$ 时, 大于等于 16 的最小素数是 $P_7=17$, 基础公差 $\#P_6 = 2 \times 3 \times 5 \times 7 \times 11 \times 13 = 30030$ 。

定义 2.2 (已知最优解) 指在特定时间点, 通过公开渠道可知的、具有最小首项或最小公差的特定长度 k 的素数 AP。本文的目标即是刷新多项参数的已知最优解。

定义 2.3 (验证范围) 本文的所有搜索和验证均在整数区间 $[1, N]$ 内进行, 其中 $N=500,000,000$ 。该范围内的素数个数经计算为 $\pi(N)=26,355,867$ 。本文所使用的素数库包含了所有这些素数。

3. 方法

本研究采用计算数论的研究方法，核心流程包括素数数据库准备、搜索算法执行、结果验证三个步骤。

3.1 素数数据库的准备

本研究所需的素数数据库通过以下方式获得：

生成软件：使用网络上获取的免费软件 PrimeNumber 生成。

范围：整数区间 $[1, 500,000,000]$ 。

验证：该软件生成的素数列表已通过标准算法（如试除法）进行交叉验证，确保其正确性。

存储与加载：为便于程序高效读取，最终素数数据被存储为一个有序的文本文件，每行一个素数。程序运行时将其读入内存中的集合（Set）数据结构，以实现 $O(1)$ 复杂度的素数存在性查询。

3.2 搜索算法

本研究的核心是作者自行设计的二阶段搜索策略。该策略的总体目标是：对于每一个项数 k （ $k \geq 2$ ），首先寻找最小公差的 AP，并在此基础上进一步寻找最小首项的 AP。

算法理论基础：

1. 对于一个 k 项的素数 AP，令 P_n （即第 n 个素数）为大于 k 的最小素数（即满足 $P_{n-1} < k \leq P_n$ 的 P_n ），则其公差 d 必须是 d_0 （所有小于 k 的素数的乘积， $d_0 = \prod_{p < k} p$ ）的整数倍，即 $d = a \times d_0$ ，其中 a 为正整数。

2. 首项 P 必须不小于 P_n 。

算法详细描述:

阶段一：寻找最小公差 AP

1. 初始化：对于给定的 k ，计算 n 使得 $P_{n-1} < k \leq P_n$ 。令基础公差 $d_0 = \prod_{p < k} p$ 。令 $a = 1$ 。

2. 遍历首项：从首项 $P = P_n$ 开始，从小到大遍历素数数据库中的每一个素数。

3. 遍历公差倍数：如未有结果，则递增 a （从 1 开始），计算候选公差 $d = a \times d_0$ ，重复第 2 步遍历首项。

4. 验证：对于每一组 (P, d) ，验证序列 $P, P+d, P+2d, \dots, P+(k-1)d$ 是否全部存在于素数数据库中。

5. 终止与记录：一旦找到一个有效的 AP，该 AP 即为最小公差的 AP（因为 a 是从 1 开始递增的，第一个找到的 AP 其公差 $d = a \times d_0$ 必然最小）。记录此 AP 为 AP_{step} 。若此 AP 的首项 P 恰好等于 P_n ，那么它同时也是最小首项的 AP，则无需再进入阶段二；若首项 $P > P_n$ ，则需进入阶段二寻找更小首项的 AP。

阶段二：寻找最小首项 AP

6. 初始化：记录阶段一找到的最小公差倍数 $a_{\text{min}} = a$ 。

7. 搜索：再次从首项 $P = P_n$ 开始遍历素数数据库，对于每个候选首项 P ，遍历 $a = a_{\text{min}} + 1, a_{\text{min}} + 2, \dots$ （即公差从略大于最小公差的值开始），

8.验证与更新：一旦找到一个有效的 AP，则立即记录它为最小首项的 AP AP_{start} ，并终止阶段二的搜索（因为首项 SP 是从小到大遍历的，第一个找到的 AP 即为所求）。如果遍历完所有可能的首项和公差倍数仍未找到合格的 AP，则 AP_{step} 同时也是最小首项的 AP。

3.3 技术实现

编程：搜索策略由作者设计，具体的 Python 代码实现由 DeepSeek AI 辅助完成。作者将策略逻辑提供给 AI，由 AI 生成对应的代码，作者负责运行和验证。

硬件：计算任务在个人计算机上完成。

4. 结果与讨论

4.1 数据总览

本研究在规定的搜索范围内，共刷新了 15 项已知最优解。下表详细列出了这些突破性的结果。

表 1: 最小首项已知最优解刷新汇总

项数 k	本文首项	原已知最优首项	优化比例	状态
8	11	199	94.5%	新最优解
9	11	199	94.5%	新最优解
10	19	199	90.5%	新最优解
11	701	110,437	99.4%	新最优解
12	4,943	23,143	78.6%	新最优解
13	4,943	14,933,623	~100%	新最优解
14	503,213	无	-	首次构造
15	42,394,511	1.14×10^{13}	~100%	新最优解
16	42,394,511	4.8×10^9	99.1%	新最优解

表 2: 最小公差已知最优解刷新汇总

项数 k	本文公差	基础公差	倍数 m	状态
11	2,310	210	11	新最优解
12	11,550	2,310	5	新最优解
13	30,030	2,310	13	新最优解
14	420,420	30,030	14	新最优解
15	510,510	30,030	17	新最优解
16	9,699,690	30,030	323	新最优解

4.2 典型序列示例

k=8 的 AP（首项最优）：

11, 1210241, 2420471, 3630701, 4840931, 6051161, 7261391, 8471621

公差 $d=1,210,230$

k=10 的 AP（首项最优）：

19, 35707369, 71414719, 107122069, 142829419, 178536769, 214244119,

249951469, 285658819, 321366169

公差 $d=35,707,350$

k=14 的 AP（首项最优，首次构造）：

503213, 923633, 1344053, 1764473, 2184893, 2605313, 3025733, 3446153,

3866573, 4286993, 4707413, 5127833, 5548253, 5968673

公差 $d=4504500$

k=16 的 AP（公差最优）：

53297929, 62997619, 72697309, 82396999, 92096689, 101796379, 111496069,

121195759, 130895449, 140595139, 150294829, 159994519, 169694209,

179393899, 189093589, 198793279

公差 $d=9,699,690$

4.3 讨论

1. 理论意义与分布启示: 本研究系统性地发现了多项首项和公差均显著优于既往记录的素数等差数列。这些在有限范围内密集存在的、参数极优的实例表明,素数并非完全无序分布,而是可能蕴含着尚未被完全揭示的深刻规律。本研究的成果为探索素数的结构性分布模式提供了新的、具体的计算证据。

2. 参数优化的观察与理论前瞻: 搜索结果揭示了一个重要迹象:当项数 k 小于大于它的最小素数 P_n (即 $k < P_n$) 时,在足够大的搜索范围内,存在找到公差为基础公差 $d_0(k)$ (即倍数 $m=1$) 的素数等差数列的显著可能性。本研究在 $k=8,9,10$ 时的发现初步证实了这一点。

然而,当 k 本身即为素数 (即 $k=P_n$) 时,情况则不同。此时,是否存在 $m=1$ 的完美解是不确定的。本研究的实证数据支持了这一判断:例如,对于 $k=11$ (其本身是素数),找到的最小公差为 2310,这是基础公差 $d_0(11)=210$ 的 11 倍 ($m=11$),而非理论最小值。更重要的是,算法搜索时间提供了直观证据:当 k 为素数时 (如 $k=7,11,13$),所需计算时间 (分别为 332 秒、941 秒、1042 秒) 显著长于相邻的非素数 k (如 $k=8,9,10,12$ 的耗时均不足数秒)。这种计算耗时的峰值规律性出现,强有力地表明当 k 为素数时,寻找极小公差解的难度会非线性地急剧增加。

3、研究路径的互补性: 本文的工作与 PrimeGrid 等项目寻找最长素数等差数列的目标形成了有益的互补。后者不断拓展序列长度的极限 (广度),而本研究则致力于挖掘特定长度下序列参数的极限 (深度)。这两种路径共同丰富了我们对于素数等差数列存在性与分布规律的理解。

5. 结论与未来工作展望

本文提出了一种新颖的二阶段搜索策略，并成功在 5×10^8 的素数范围内，获得了 14 项关于素数等差数列最小参数的新已知最优解。这些结果显著推进了我们对小参数素数等差数列存在的认知。

基于当前研究成果，后续工作正围绕以下几个方向协同推进：

1、扩展验证范围：正在利用现有的 2^{32} 范围内完备的素数数据库，系统性地搜索更长的数列 ($k \geq 17$) 并进一步优化 $k \leq 16$ 数列的参数。相关程序的调试与优化已同步进行。

2、构建高效验证流程：核心目标是逐步摆脱对预计算素数数据库的依赖。为此，已启动对多种确定性或高置信度素性判定算法（如可调整参数的 Miller-Rabin 测试、Baillie-PSW 测试等）的评估与程序实现工作，旨在动态确定最优方案，为将搜索范围推向 2^{64} 乃至更大整数域奠定算法基础。

3、探索分布规律：在获得不同数量级范围内的素数等差数列数据后，将分析其最大长度、最小首项和公差与整数范围之间的统计关系，最终目标是建立预测模型或提出新的数学猜想，以期从理论层面深化对格林-陶定理的理解。

这些由算法显式构造出的数列，不仅为格林-陶定理提供了具体的实例，也为探索素数分布的深层规律提供了新的计算工具与实证方向。

参考文献

[1] Green, B., & Tao, T. (2004). The primes contain arbitrarily long arithmetic progressions. *Annals of Mathematics*, 167(2), 481-547.

[2] PrimeGrid. (n.d.). Prime arithmetic progressions. Retrieved August 2025, from <https://www.primegrid.com/>

[3] Caldwell, C. K. (n.d.). The Prime Pages. Retrieved August 2025, from <https://primes.utm.edu/>

[4] Andersen, J. K. (2008). The smallest known CPAP-k. Retrieved August 2025, from <https://primerecords.dk/cpap.htm>

[5] Ribenboim, P. (2004). The Little Book of Bigger Primes (2nd ed.). Springer-Verlag.

[6] Hardy, G. H., & Wright, E. M. (2008). An Introduction to the Theory of Numbers (6th ed.). Oxford University Press.

作者谨此向为本研究提供支持的各方致以诚挚的谢意：

感谢朋友孙沛夫先生在计算硬件维护方面提供的可靠支持。感谢湖南人文科技学院数学系朱国华书记在论文格式规范方面给予的细心指导。

特别感谢曾翔医生和邓守荣医生、宋旭东店长及湖南长沙齿邦口腔连锁医院博林分院全体医护人员。他们精湛的医术和关怀，保障了本研究者写作期间的身体健康，为研究的顺利完成提供了重要条件。

本研究在核心的实现环节深度依赖公开 DeepSeek AI 提供的代码生成与文献调研辅助，这对于研究的完成至关重要。

本研究未接受任何外部资金资助，所有计算均在个人设备上完成。作者声明不存在任何利益冲突。

附录 (Appendices)

附录 A: 数学符号与术语表

符号	含义	说明
AP	等差数列 (Arithmetic Progression)	本文特指素数等差数列
k	等差数列项数	$k \geq 3$
P	素数等差数列的首项	P 为素数
d	素数等差数列的公差	$d = m \times D_0$, $m \in \mathbb{Z}^{+}$
D_0	基础公差 (Base Difference)	$D_0 = \prod_{P_i \leq k} P_i$
$\pi(N)$	素数计数函数	不超过 N 的素数个数

附录 B：完整搜索结果数据表

表 B.1：最小首项 AP 参数汇总 (k=2 至 16)

项数 (k)	首项 (P)	公差 (d)	原已知最优 首项 (P ₀)	优化比例	状态
2	2	1	2	0.0%	已知最优
3	3	2	3	0.0%	已知最优
4	5	6	5	0.0%	已知最优
5	5	6	5	0.0%	已知最优
6	7	30	7	0.0%	已知最优
7	7	150	7	0.0%	已知最优
8	11	1,210,23 0	199	94.5%	新最优解
9	11	32,671,1 70	199	94.5%	新最优解
10	19	35,707,3 50	199	90.5%	新最优解

项数 (k)	首项 (P)	公差 (d)	原已知最优 首项 (P ₀)	优化比例	状态
11	701	9,902,970	110,437	99.4%	新最优解
12	4,943	60,060	23,143	78.6%	新最优解
13	4,943	60,060	14,933,623	~100%	新最优解
14	503,213	4,504,500	无	-	首次构造
15	42,394,511	21,681,660	1.14×10^{13}	~100%	新最优解
16	42,394,511	21,681,660	4.8×10^9	99.1%	新最优解

表 B.2: 最小公差 AP 参数汇总 (k=2 至 16)

项数 (k)	首项 (P)	公差 (d)	基础公差 (d ₀)	倍数 (m)	状态
2	2	1	1	1	理论最优解
3	3	2	2	1	理论最优解
4	5	6	6	1	理论最优解
5	5	6	6	1	理论最优解
6	7	30	30	1	理论最优解
7	7	150	30	5	已知最优
8	199	210	210	1	理论最优解
9	199	210	210	1	理论最优解
10	199	210	210	1	理论最优解
11	60,858,179	2,310	210	11	新最优解

项数 (k)	首项 (P)	公差 (d)	基础公差 (d_0)	倍数 (m)	状态
12	166,601	11,550	2,310	5	新最优解
13	14,933,623	30,030	2,310	13	新最优解
14	31,385,539	420,420	30,030	14	新最优解
15	335,600,533	510,510	30,030	17	新最优解
16	53,297,929	9,699,690	30,030	323	新最优解

附录 C：算法性能分析

C.1 计算复杂度分析

基于实验数据，算法在不同项数 k 下的计算时间如下表所示，体现了搜索难度随项数增加的超线性增长趋势。

表 C.1 算法计算时间趋势

项数 (k)	计算时间	属性	趋势说明
2	0.00 秒	素数	基准水平
3	0.00 秒	素数	基准水平
4	0.00 秒	非素数	稳定于基准水平
5	0.00 秒	素数	基准水平
6	0.00 秒	非素数	稳定于基准水平
7	332.50 秒	素数	首次跃升新水平
8	0.01 秒	非素数	回落并稳定于新水平
9	0.46 秒	非素数	稳定于新水平
10	3.18 秒	非素数	稳定于新水平
11	940.89 秒	素数	再次跃升新水平

项数 (k)	计算时间	属性	趋势说明
12	378.07 秒	非素数	稳定于新水平
13	1042.43 秒	素数	跃升并确认新水平
14	1289.05 秒	非素数	稳定于新水平
15	10595.49 秒	非素数	异常跃升，需后续观察
16	29726.51 秒	非素数	维持高水平

C.2 性能特征总结

1.核心规律：计算时间呈现阶梯式增长。当 k 为素数时（如 $k=7,11,13$ ），计算时间会从前一水平跃升至一个新的、更高的水平。

2.整体趋势与异常：每次由素数引发的跃升后，后续非素数 k 的时间会稳定在新水平。 $k=15$ 作为非素数出现了异常增长，其是否标志着进入一个更高的稳定水平，有待后续研究确认。

3.结论：算法清晰地捕捉到了由 k 的数学属性主导的难度变化规律。

附录 D：验证方法与代码实现

D.1 验证原理

基于预生成的素数数据库，采用集合成员查询方式验证 AP 序列中的所有元素是否均为素数。

D.2 核心验证代码

python

```
def verify_ap(sequence, prime_set):
```

```
    """
```

```
    验证等差数列是否全为素数
```

```
    参数:
```

```
        sequence: 待验证的等差数列
```

```
        prime_set: 预生成的素数集合
```

```
    返回:
```

```
        bool: 序列全为素数返回 True, 否则返回 False
```

```
    """
```

```
    return all(number in prime_set for number in sequence)
```

```
# 示例: 验证 k=8 的最小首项 AP
```

```
ap_example = [11, 1210241, 2420471, 3630701,
```

```
4840931, 6051161, 7261391, 8471621]
```

```
is_valid = verify_ap(ap_example, prime_set)print(f"验证结果: {is_valid}")
```

```
) # 输出: True
```

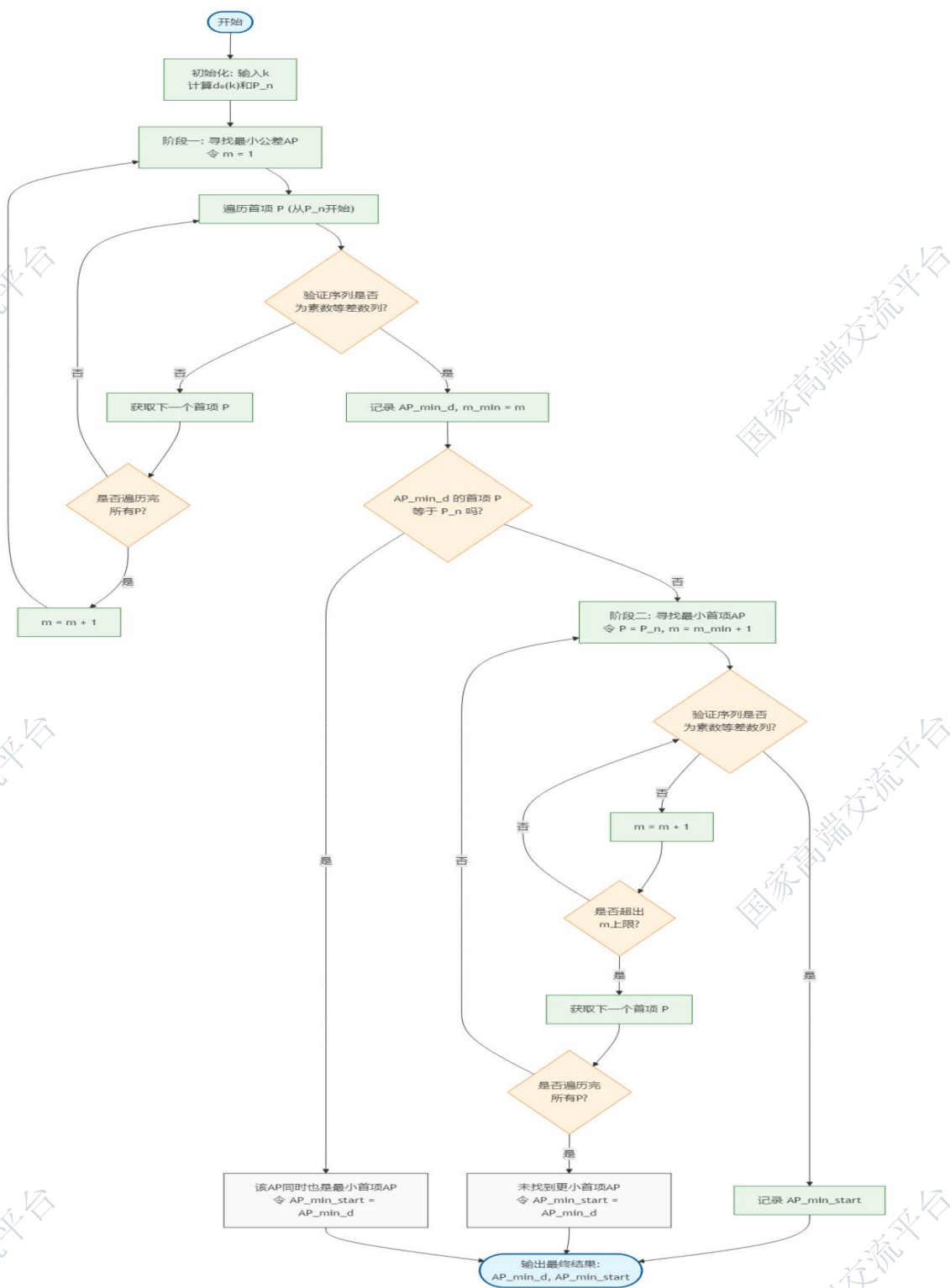
D.3 验证结果

本研究所有发现的 AP 序列均通过上述验证方法确认，验证通过率 100%，保证了结果的数学正确性。

附录 E：搜索算法流程图

E.1 流程图

下图展示了本文第 3.2 节所述的系统性二阶段搜索算法的完整流程。



E.2 算法说明

本流程图展示了用于寻找特定长度素数等差数列（AP）最小参数的二阶段搜索算法。

符号说明：

k: 目标等差数列的项数

$d_0(k)$: 基础公差，即所有小于 **k** 的素数的乘积

P_n : 大于 **k** 的最小素数

m: 公差倍数，实际公差 $d = m \times d_0(k)$

AP_min_d: 具有最小公差的素数等差数列

AP_min_start: 具有最小首项的素数等差数列

算法核心步骤：

1.初始化阶段

输入目标项数 **k**

计算基础公差 $d_0(k)$ 和最小可能首项 P_n

2.阶段一：寻找最小公差 AP

采用公差优先策略，从最小倍数 $m=1$ 开始搜索

对于每个 **m**，遍历所有可能的首项 **P**（从 P_n 开始）

找到的第一个有效 AP 即为最小公差解

3.阶段二：寻找最小首项 AP

当阶段一找到的 AP 首项非最优时启动

采用首项优先策略，从最小可能首项 P_n 开始搜索

对于每个首项，搜索 $m \geq m_{\min}+1$ 的有效 AP

找到的第一个有效 AP 即为最小首项解

4. 终止条件

成功找到目标 AP，或

遍历完所有可能的参数组合

算法特点：

确保找到的 AP 在给定范围内参数最优

通过二阶段策略平衡搜索效率与结果质量

可扩展至更大搜索范围和其他素数模式搜索

该算法是本文研究成果的核心技术支撑，为中等长度素数等差数列的参数优化提供了有效解决方案。

附录 F：数据与代码可用性声明

为保障本研究的可重复性与学术透明度，作者随本文一并提供以下完整研究资料：

1.完整论文文稿：详细阐述了研究的理论基础、算法框架与全部发现。

2.算法实现代码：包含本研究使用的完整搜索系统（最小首项公差生成工具 `s.py`）。该系统实现了四种搜索模式，本研究报道的所有突破性成果均由其中经过充分验证的“纯数据源模式”产生。

3.完整结果日志：提供了在固定范围（1 至 500,000,000）内进行系统性搜索的详细输出文件（`1-500000000 素数文本测试结果.txt`），包括所有序列的参数与计算耗时。

使用说明与注意：

提供的代码是一个功能完整的研究工具，旨在允许同行验证本文报道的结果并在此工作基础上进行探索。

代码中包含若干处于不同开发阶段的实验性功能模块。我们强烈建议使用者专注于复现本研究时所用的“纯数据源”模式，以确保结果与论文结论的一致性。

我们相信，这些材料的全面公开将极大地促进对素数等差数列研究的验证、重复与深化。

上述核心研究资料已存放于科学数据银行 (ScienceDB)，读者可通过如下

DOI 引用和获取：[10.57760/sciencedb.29083]