基于特殊合数分布规律的研究与素数递进推演方法的探讨

赵山东

湖南省超级计算科学学会　湖南长沙

【摘要】素数筛法始终将“直接定位素数”作为核心目标，限于“素数无显性规律”的瓶颈，只能通过优化遍历效率来缓解这一局限，无法实现本质突破。论文转变研究视角，将“无规律的素数问题”转化为“有规律的合数问题”，改变了筛法的逻辑起点；揭示了特殊合数的分布规律，提出递进推演法，构建了“已知素数➞推演特殊合数➞筛选素数➞扩展素数集”的完整分类分段推演迭代闭环，实现了特殊合数的规律推演与素数的精准筛选，该方法具有自然迭代扩展的理论可能性；减少了传统筛法在范围覆盖、逻辑处理及计算等过程中的冗余，且运算方法更为简洁，在时空复杂度优化、迭代扩展性及工程实用性等方面显示出一定的理论优势。

【关键词】数论；素数筛法；合数分布规律；递进推演法

【收稿日期】2025 年 8 月 14 日　【出刊日期】2025 年 9 月 18 日　【DOI】10.12208/j.aam.20250028

Research on the distribution law of special composite numbers and a new framework for progressive deduction of prime numbers

Shandong Zhao

Hunan Supercomputing Science Society, Changsha, Hunan

【**Abstract**】The prime sieve method has always taken "directly locating prime numbers" as its core objective. However, due to the bottleneck of "prime numbers having no explicit pattern", it can only alleviate this limitation by optimizing the efficiency of traversal, without achieving an essential breakthrough. This paper shifts the research perspective, transforming the "irregular prime number problem" into the "regular composite number problem", changing the logical starting point of the sieve method; it reveals the distribution pattern of special composite numbers, proposes the progressive deduction method, and constructs a complete iterative closed loop of "known prime numbers → deducing special composite numbers → screening prime numbers → expanding the prime number set". This method realizes the regular deduction of special composite numbers and the precise screening of prime numbers, and has the theoretical possibility of natural iterative expansion. It reduces the redundancy in the process of range coverage, logical processing, and calculation in traditional sieve methods, and its operation method is more concise. It shows certain theoretical advantages in terms of time and space complexity optimization, iterative expandability, and engineering practicality.

【**Keywords**】Number theory; Prime sieve method; Distribution law of composite numbers; Progressive deduction method

1 引言

长期以来，素数实用筛法研究范式始终围绕“直接追踪素数分布”展开，衍生出基础标记类（如埃拉托斯特尼筛法）、线性优化类（如欧拉筛法）、二次型优化类（如阿特金筛法）、内存优化类（如分段筛法）及朴素验证类（如试除法）五大类。

近年来，涌现了一系列优化筛法，例如：LMO 筛法（改进型二次型筛法，2018 年）、量子辅助素数筛

法（QAPS，2021 年）、密码学专用快速筛法（CFS，2022 年）。但这些方法均未突破核心局限——由于素数缺乏显性生成公式，所有筛法都必须覆盖目标区间内的全部自然数，通过被动且无显性系统规律的合数筛除来间接获取素数，始终无法实现本质突破。

本文革新研究视角，聚焦特殊合数规律的系统揭示或应用，将“无规律的素数问题”等效转化为“有规律的特殊合数问题”，从根源上重构了筛法的逻辑起点；构建“主动规律推演”新范式，通过系统揭示特殊合数的规律特征，精准推演出特殊合数，进而精准锁定素数，从而实现素数的规律化筛选。

2 本研究的核心思路

2.1 范式转换

打破传统研究中“直接追踪素数生成规律”的固有范式，鉴于素数缺乏显性的生成规律，通过概念的进一步精准界定，转而聚焦其互补集合——特殊合数的构造逻辑与分布特征，从而反向精准锁定素数。核心在于依托数论的基本理论（如整除性、同余性质、乘法分解规律、剩余定理等），结合特殊奇数集的结构特征优化方法体系，构建能够无遗漏推演特殊合数的精确数学规律；通过完备且精准地推演特殊合数，反向剔除合数以锁定素数，最终实现素数的高效、规律化筛选。

2.2 概念的精准界定与规律揭示

2.2.1 根据数论中对奇数、素数和合数等定义：所有排除 2 和 5 以外的素数，其个位数必定属于集合{1，3，7，9}。基于此，本文将大于 1 的这类数定义为“特殊奇数”，旨在为后续规律的探索和应用奠定结构基础。

2.2.2 特殊奇数集仅由特殊合数和素数组成，涵盖除 2 和 5 之外的所有素数。基于特殊奇数，提出了特殊合数的新定义，为揭示特殊合数的分布规律奠定了基础框架。若能规律性地推导出特殊合数，便等同于规律性地精确筛选出素数，从而实现研究对象的等效转换。

2.2.3 基于数论基本理论（如整除性、同余性质、乘法分解规律、剩余定理、算术基本定理）以及集合论等的应用，通过深入解析特殊奇数和特殊合数的结构，系统揭示出特殊合数的分布规律；涵盖四大核心维度：个位分类、周期分布、首现规则、重复规律；并构建了对特殊奇数和特殊合数进行分类推演较完备的数学公式。

2.3 方法构建

基于特殊合数的分布规律及“合数最小素因子$\leqslant\sqrt{M}$”的数论推论，提出了递进推演法——以已知素数集为起点，针对最大素因子平方范围内的所有特殊奇数，按个位分类开展公式推演，再通过公式运算精准定位范围内的所有特殊合数并剔除，剩余元素即为素数；将新素数集作为下一轮推演的基础，形成自动分类、分段递进的素数筛选方法，理论上可实现研究边界的自然无限迭代扩展。

该方法的核心特点在于将传统的“无规律且被动的遍历筛除”方式革新为“有规律且主动的运算推演”模式。

3 基本核心概念和定义

为清晰阐述研究逻辑，定义核心数学对象如下：

定义 1：特殊奇数集（Special Odd Number, SON，记为 S）：大于 1 且个位数 i 为{1，3，7，9}的自然数集合，按个位数 $i\in\{1，3，7，9\}$划分为 4 个互斥子集，各子集记为 S_i（$i\in\{1，3，7，9\}$）：

$S=S_1\cup S_3\cup S_7\cup S_9$，　$S_i\cap S_j=\varnothing$（$\forall i，j\in\{1，3，7，9\}$，$i\neq j$），

$S_i=\{10n+i \mid n\in N\cup\{0\}$，$i\in\{1，3，7，9\}$，$S_i\equiv i$（mod10），且 $S_i>1\}$（3.1）

（n=0，1，2，...，故 S_i 的首位元素 S_1 为 11，S_3 为 3，S_7 为 7，S_9 为 9）。

定义 2：特殊合数集（Special Composite Number, SCN，记为 C）：S 中的合数子集，划分为 4 个互斥子集，记为 C_i（$i\in\{1，3，7，9\}$）：

$C=C_1\cup C_3\cup C_7\cup C_9$，$C_i\subseteq S_i$

$C_i\cap C_j=\varnothing$（$\forall i，j\in\{1，3，7，9\}$，$i\neq j$）

定义 3：特殊素数集 P：S 中的素数子集，划分为 4 个互斥子集，记为 P_i（$i\in\{1，3，7，9\}$，为素数的

个位数），且 $P_i \subseteq S_i$。

$P=P_1 \cup P_3 \cup P_7 \cup P_9$

上述定义属于研究上的分类约定，旨在简化后续推演与算法描述，并不引入新的数论对象。

定义 4：S 中的子集满足：

$S=C \cup P$，且 $C \cap P=\varnothing$

即从 S 中移除 C 后，剩余元素均为素数；且 P 为除 2、5 外的全部素数集合；换句话说，S 包含了除 2、5 外的所有素数。

定义 5：特殊合数分布规律（Distribution Law of Special Composite Numbers, DLSCN）：特殊合数遵循明确的数学推演规律，包含个位分类、周期分布、首现规则、重复规律四大核心维度（详见 2），且可推导出相应的数学推演公式。

定义 6：递进推演法（Progressive Deduction Method, PDM）：通过“已知 P 集→推演 C→筛选 P→扩展 P 集”的自动分类、分段迭代闭环流程，实现素数筛选范围逐步扩大的方法（详见第 3 节）。

4 特殊合数分布规律（DLSCN）

DLSCN 聚焦素因子构成 C 的分布特征，通过“个位分类、周期分布、首现规则、重复规律”四个维度，揭示了可量化验证的规律体系。

4.1 个位分类：基于个位数划分特殊奇数

DLSCN 分析前提为“按个位数对 S 分类分列”——同一子集 S_i 个位数固定，特殊合数分布规律在单一序列中显性呈现。根据式（3.1）可快速运算推演出 S_i 的分类分布规律，表 4-1 展示 S_i 分类分列结果（节选）。

表 4-1 特殊奇数分类分列归纳表（节选）

序号 n	S_1（个位数 1）	S_3（个位数 3）	S_7（个位数 7）	S_9（个位数 9）
0	-	3（P_3）	7（P_7）	9（C_9）
1	11（P_1）	13（P_3）	17（P_7）	19（P_9）
2	21（C_1）	23（P_3）	27（C_7）	29（P_9）
3	31（P_1）	33（C_3）	37（P_7）	39（C_9）
4	41（P_1）	43（P_3）	47（P_7）	49（C_9）
5	51（C_1）	53（P_3）	57（C_7）	59（P_9）
6	61（P_1）	63（C_3）	67（P_7）	69（C_9）
7	71（P_1）	73（P_3）	77（C_7）	79（P_9）
8	81（C_1）	83（P_3）	87（C_7）	89（P_9）
9	91（C_1）	93（C_3）	97（P_7）	99（C_9）
10	101（P_1）	103（P_3）	107（P_7）	109（P_9）
⋮	⋮	⋮	⋮	⋮
35	351（C_1）	353（P_3）	357（C_7）	359（P_9）
36	361（C_1）	363（C_3）	367（P_7）	369（C_9）

4.2 周期分布：素因子呈现周期性规律分布

4.2.1 定义 7：对任意素因子 $p \in P$（$p \notin \{2, 5\}$），其在某一 S_i 子集（$i \in \{1, 3, 7, 9\}$）中构成的 C_i 子集遵循严格周期性分布规律：

设 $C_{f,\,i}$ 为 S_i 中含 p 的首现特殊合数，则 S_i 中所有含 p 的特殊合数可表示为：

$C_i(k_p) = C_{f,\,i} + 10pk_p$，其中 $k_p \in \mathrm{N} \cup \{0\}$，且 $C_i(k_p) \in C_i$ （4.1）

式中，k_p 与相应的推演素数和 $C_i(k_p)$ 对应，如 p=3，则标识 k_3；在以个位数分类的 $C_i(k_3)$ 推演中，分别对应四个不同的 k_3 值，它是每一个素数推演过程中的重要标识计算值。

$10pk_p$ 为周期项，周期长度为 $10p$；k_p=0 对应特定 p 的首现特殊合数 $C_{f,\,i}$，$k_p \geq 1$ 为后续周期性出现的 C_i 元素。

4.2.2 引理 1：单个素因子对应的 C_i 周期为 $10p$

对任意素因子 $p \in P$（$p \notin \{2, 5\}$），其在 S_i 中生成的 C_i 序列 $C_i(k_p)$ 的最小正周期为 $10p$。

4.3 首现规则——$C_{f,i}$ 的确定方法

4.3.1 定义 8：首现特殊合数 $C_{f,i}$（S_i 中含素因子 p 的首个 C_i）构成遵循固定规则，取决于 p 的个位数类别（记 p 的个位数为 i，则 $p \in P_i$），例如 p=11 的个位数为 1，故 $p \in P_1$。则：

$C_{f,i} = p \times D_s$，其中 $D_s \in \{3, 7, 9, 11\}$（4.2）

D_s 需根据 S_i 与 P_i 的搭配规则从｛3，7，9，11｝中确定，表 4-2 给出其搭配生成规则。

表 4-2　D_s 的搭配生成规则

	S_1	S_3	S_7	S_9
P_1	D_s=11	D_s=3	D_s=7	D_s=9
P_3	D_s=7	D_s=11	D_s=9	D_s=3
P_7	D_s=3	D_s=9	D_s=11	D_s=7
P_9	D_s=9	D_s=7	D_s=3	D_s=11

将式（4.2）代入（4.1）得：

$C_i(k_p) = p \times (D_s + 10k_p)$（4.3）

4.3.2 引理 2：p 首现特殊合数存在性

对于任意素因子 $p \in P$（$p \notin \{2, 5\}$），根据表 4-2 的 D_s 搭配规则，存在 $D_s \in \{3, 7, 9, 11\}$，使得 $C_{f,i} = p \times D_s$ 成立。

4.3.3 示例：以素因子 p=31（$p \in P_1$，个位数为 1）在 S_i 中的首现特殊合数分析为例：

S_1 对应 P_1 的 D_s=11，故 $C_{f,1} = 31 \times 11 = 341$，后续 S_1 序列中含 p=31 的数为 $341 + 310k$（k_{31}=0，1，2，…），即 341、651、961…；

S_3 对应 P_1 的 D_s=3，故 $C_{f,3} = 31 \times 3 = 93$，后续 S_3 序列中含 p=31 的数为 $93 + 310k$（k_{31}=0，1，2，…），即 93、403、713…；

S_7 对应 P_1 的 D_s=7，故 $C_{f,7} = 31 \times 7 = 217$，后续 S_7 序列中含 p=31 的数为 $217 + 310k$（k_{31}=0，1，2，…），即 217、527、837…；

S_9 对应 P_1 的 D_s=9，故 $C_{f,9} = 31 \times 9 = 279$，后续 S_9 序列中含 p=31 的数为 $279 + 310k$（k_{31}=0，1，2，…），即 279、589、899…。

上述 S_i 序列中的 C_i 元素均可被 31 整除。

4.4 重复规律：素因子组合的周期性重叠

C_i 由 2 个以上素因子的乘积构成（如 231=3×7×11），且具有周期性重复特征，其重复原理与单个素因子的周期性分布规律一致。

多素因子组合的 C_i 可能与单一素因子或小素因子的 C_i 重叠（如 231×3=693，既属于 p={3}的 C_i 序列——693=3×231，也属于 p={3，7，11}的 C_i 序列）。

重复规律对精确计算特殊合数的占比十分重要，但这不是本文讨论的主体，不做赘述。

不过，素因子组合的周期性重复特征会导致对一些特殊合数的重复确认，虽不影响特殊合数鉴别的精度，但会产生反复确认的逻辑冗余，需尽量剔除这类冗余操作。

5 递进推演法（PDM）

5.1 基本原理

PDM 的核心逻辑基于数论基本定理的推论：若正整数 $M > 1$ 为合数，则其最小素因子 $q \leqslant \sqrt{M}$。

反演可得：已知素数集 $P = \{p_1, p_2, \cdots, p_k\}$（$p_k$ 为最大素因子），则在范围[11，p_k^2]内，所有 C_i 均可由 P 中的素因子构成；若某元素 $m \in S_i$ 且 $m \leqslant p_k^2$，且 m 不能被 P 中任何素因子整除，则 m 为素数。

据此，PDM 构建“迭代闭环”：已知素数集→推演预定范围内 C→筛选素数→扩展素数集→推演扩展范围。

5.2 PDM 推演的逻辑步骤

遵循电脑自动推演建立编程的逻辑步骤：

5.2.1 迭代分段

确定迭代推演范围 S，给出初始已知特殊素数集，按从小到大的顺序排列建立已知特殊素数表。迭代分段 S 范围的确定有多种方式：

一是根据已知素数集中最大素数的平方确定 S 的范围（简称：已知素数分段法）；如第一次迭代推演的已知素数集 $P^{(1)}$={3，7，11，13，17，19}，最大素数为 19，则迭代推演的范围 $S^{(1)}=19^2=361$；经过推演扩展得到已知素数集 $P^{(2)}$={3，7，11，13，17，19，23，…，359}，则第二次迭代推演的范围 $S^{(2)}=359^2=128881$，即 $S^{(1)}$ 至 $S^{(2)}$，依此类推。

二是每次分段迭代推演时，按素数大小顺序固定增加一定数量的素数，再以该素数集中最大素数的平方数确定 S 的范围（简称：固定素数递增分段法）。例如，下一次推演分段在上一次的基础上，按大小顺序固定增加 6 个素数；如上一次推演分段的素数集为 $P^{(1)}$={3，7，11，13，17，19}，$S^{(1)}=19^2=361$；则下一次推演的素数集扩展为 $P^{(2)}$={3，7，11，13，17，19，23，29，31，37，41，43}，$S^{(2)}=43^2=1849$，即 361 至 1849，依此类推。

迭代分段的核心原则是，分段的最大范围不得超过已知最大素数的平方。基于这一原则，还有一些其他分段方法，此处不再赘述。

5.2.2 分类推演 S 分布数据

第一步，根据 S 的预定范围确定 n 的取值范围：

$n=\lfloor S\div 10\rfloor$ （5.1）

示例：$S^{(1)}=19^2=361$，$n^{(1)}=\lfloor S^{(1)}\div 10\rfloor=\lfloor 361\div 10\rfloor=36$，即 $n^{(1)}$ 的取值范围为 0 至 36。第二次迭代时，$n^{(2)}=\lfloor S^{(2)}\div 10\rfloor$，其取值范围为（$n^{(1)}+1=37$）至 $n^{(2)}$，依此类推。

第二步，依据式（3.1）将 S_i 分类推演至确定范围；具体如表 4-1 所示。

5.2.3 基于 DLSCN 推演 C

第一次迭代推演：

第一步，建立 D_s 索引表，如表 4-2；

第二步，对 $P^{(1)}$ 中每个素因子 p，按从小到大的顺序，依式（4.3）和索引表，分类生成各 S_i 中的 C_i 序列，并予以标记或剔除，处理重复项（如 21=3×7，同时被 3 和 7 生成，仅记为 1 个 C_i）。

如 p=3 生成 $C_1(k_3)=3\times(7+10k_3)$={21，51，81，…，351}，$k_3$={0，1，2，...11}；

p=7 生成 $C_1(k_7)=7\times(3+10k_7)$={21（重复），91，161，…，301}，$k_7$={0，1，2，3，4}；

其它依此类推，如表 5-2 至 5-5 所示。

第三步，剔除所有推演出的 C，$S^{(1)}$ 中剩余的 S 即为 $P^{(2)}$；将 $P^{(2)}$ 按由小到大的顺序整理，形成新的已知素数集，存储至内存或硬盘中。

第四步，在电脑中标记存储好 $n^{(1)}$ 范围的最大值，如 $S^{(1)}=361$，则 $n^{(1)}=36$；

针对每个 p，标记并存储各推演 $C_i(k_p)$ 的 k_p 最大值——如 p=3 推演得到的 $C_1(k_3)$，其最大 k_3=11；p=7 推演得到的 $C_1(k_7)$，其最大 k_7=4 等。每个 p 对应四个 k_p 最大值，可通过矩阵表格形式存储在电脑内存或硬盘中，供后续迭代推演使用，如后面表 5-1 所示。清除内存中的 $S^{(1)}$ 存储数据和一些无用数据。

第二次迭代推演流程：重复执行上述全部步骤。需注意推演范围的差异，S 的取值范围由（$n^{(1)}+1$）至 $n^{(2)}$ 确定；对于已参与上次迭代的素数，其 $C_i(k_p)$ 的推演运算需从上一次迭代的最大 k_p 值+1 处开始。新参与迭代的素数推演 $C_i(k_p)$ 的 k_p 取值问题，后面讨论。

5.3 引理 3：DLSCN 的无遗漏覆盖

对于任何素因子 p（$p\in P$，$p\notin\{2，5\}$），依据 DLSCN 在全部 4 个 S_i 子集中，均能生成完整且无遗漏覆盖包含 p 的 C_i 序列。

5.4 PDM 的有效性验证

表 5-2 至 5-5 呈现 $P^{(1)}$={3，7，11，13，17，19}在各 S_i 中推演的 C_i 分布至 369 范围，可验证 PDM 的无遗漏性。

表 5-1 首次迭代最大 k_p 值表

	S_1	S_3	S_7	S_9
p=3	k_3=11	k_3=11	k_3=11	k_3=11
p=7	k_7=4	k_7=4	k_7=4	k_7=4
p=11	k_{11}=2	k_{11}=3	k_{11}=2	k_{11}=2
p=13	k_{13}=2	k_{13}=1	k_{13}=1	k_{13}=2
p=17	k_{17}=1	k_{17}=1	k_{17}=1	k_{17}=1
p=19	k_{19}=1	k_{19}=1	k_{19}=1	k_{19}=0

表 5-2 S_1（个位数 1）素因子分布规律表

S_1 元素	p=3	p=7	p=11	p=13	p=17	p=19	类型
11							P_i
21	3*7	7*3					C_i
31							P_i
41							P_i
51	3*17				17*3		C_i
61							P_i
71							P_i
81	3*3*3*3						C_i
91		7*13		13×7			C_i
101							P_i
111	3*37						C_i
121			11*11				C_i
131							P_i
141	3*47						C_i
151							P_i
161		7*23					C_i
171	3*3*19					19*3*3	C_i
181							P_i
191							P_i
201	3*67						C_i
211							P_i
221				13*17	17*13		C_i
231	3*7*11	7*3*11	11*7*3				C_i
241							P_i
251							P_i
261	3*3*29						C_i
271							P_i
281							P_i
291	3*97						C_i
301		7*43					C_i
311							P_i
321	3*107						C_i
331							P_i
341			11*31				C_i
351	3*3*3*13			13*3*3*3			C_i
361						19*19	C_i
k_p 统计	k_3=11	k_7=4	k_{11}=2	k_{13}=2	k_{17}=1	k_{19}=1	

表 5-3 S_3（个位数 3）素因子分布规律表

S_3 元素	p=3	p=7	p=11	p=13	p=17	p=19	类型
3							P_i
13							P_i
23							P_i
33	3*11		11*3				C_i
43							P_i
53							P_i
63	3*3*7	7*3*3					C_i
73							P_i
83							P_i
93	3*31						C_i
103							P_i
113							P_i
123	3*41						C_i
133		7*19				19*7	C_i
143			11*13	13*11			C_i
153	3*51				17*3*3		C_i
163							P_i
173							P_i
183	3*61						C_i
193							P_i
203		7*29					C_i
213	3*71						C_i
223							P_i
233							P_i
243	3*3*3*3*3						C_i
253			11*23				C_i
263							P_i
273	3*91	7*3*13		13*3*7			C_i
283							P_i
293							P_i
303	3*101						C_i
313							P_i
323					17*19	19*17	C_i
333	3*3*37						C_i
343		7*49					C_i
353							P_i
363	3*11*11		11*11*3				C_i
k_p 统计	k_3=11	k_7=4	k_{11}=3	k_{13}=1	k_{17}=1	k_{19}=1	

表 5-4 S_7（个位数 7）素因子分布规律表

S_7 元素	p=3	p=7	p=11	p=13	p=17	p=19	类型
7							P_i
17							P_i
27	3*9						C_i
37							P_i
47							P_i
57	3*19					19*3	C_i
67							P_i
77		7*11	11*7				C_i
87	3*29						C_i
97							P_i
107							P_i
117	3*3*13			13*3*3			C_i
127							P_i
137							P_i
147	3*7*7	7*7*3					C_i
157							P_i
167							P_i
177	3*59						C_i
187			11*17		17*11		C_i
197							P_i
207	3*3*23						C_i
217		7*31					C_i
227							P_i
237	3*79						C_i
247				13*19		19*13	C_i
257							P_i
267	3*89						C_i
277							P_i
287		7*41					C_i
297	3*3*3*11		11*3*3*3				C_i
307							P_i
317							P_i
327	3*109						C_i
337							P_i
347							P_i
357	3*7*17	7*3*17			17*7*3		C_i
367							P_i
k_p 统计	k_3=11	k_7=4	k_{11}=2	k_{13}=1	k_{17}=1	k_{19}=1	

表 5-5　S_9（个位数 9）素因子分布规律表

S_9 元素	p=3	p=7	p=11	p=13	p=17	p=19	类型
9	3*3						C_i
19							P_i
29							P_i
39	3*13			13*3			C_i
49		7*7					C_i
59							P_i
69	3*23						C_i
79							P_i
89							P_i
99	3*3*11		11*3*3				C_i
109							P_i
119		7*17			17*7		C_i
129	3*43						C_i
139							P_i
149							P_i
159	3*53						C_i
169				13*13			C_i
179							P_i
189	3*3*3*7	7*3*3*3					C_i
199							P_i
209			11*19			19*11	C_i
219	3*73						C_i
229							P_i
239							P_i
249	3*83						C_i
259		7*37					C_i
269							P_i
279	3*3*31						C_i
289					17*17		C_i
299				13*23			C_i
309	3*103						C_i
319			11*29				C_i
329		7*47					C_i
339	3*113						C_i
349							P_i
359							P_i
369	3*3*41						C_i
k_p 统计	k_3=11	k_7=4	k_{11}=2	k_{13}=2	k_{17}=1	k_{19}=0	

注：“p=3”列表示该列元素是否含素因子 3（如 21=3×7，故标注“3×7”）；“类型”列中 P_i 为素数，C_i 为特殊合数。

表 5-2 至 5-5 显示，在 $n^{(1)}$=36 范围内的所有 C 均被 $P^{(1)}$ 的素因子覆盖，无遗漏；剩余元素（如 3，7，11，13…367）均为素数，证明 PDM 可精准筛选素数。

6 PDM 的时空优化

PDM 的空间复杂度取决于每次载入内存的数据量；时间复杂度则由数据调用频次、逻辑运算与数学计算的总量共同决定；而迭代扩展度则依赖于前两者的综合优化水平。

6.1 空间复杂度分析与优化

PDM 每次载入内存的数据，主要包括用于推演的已知素数数量、已参与推演素数在四类 S_i 中对应的首现特殊合数数据群组、上次迭代运算得到的最大 k_p 值数据群组、四组推演分段范围对应的 S_i 分类分布数据，以及推演相关程序等。其中，根据本文前述 PDM 的推演逻辑，最大 k_p 值数据群组的数据量最大，为用于推演的已知素数数量的四倍；其次是首现特殊合数数据群组——若要通过存储减少重复计算量，其数据量与 k_p 值相当。这或许是 PDM 空间复杂度的致命硬伤。内存中承载的用于推演的已知素数数量，是决定 PDM 迭代扩展能力的关键；因此，PDM 的空间优化必须从这三类数据的处理方式入手。

6.1.1 首现特殊合数数据群组的优化处理

首现特殊合数的计算异常简单，每个素数在每次迭代推演中的运算量均固定不变，与 S 和 C 分布运算的工作量相比，简直微乎其微；对时间效率的影响几乎可以忽略不计。因此，这部分数据不适合内存存储处理，而是需要在每次迭代时重新计算。其次，PDM 按照 S_i 分为相互独立的四类，可分别独立运算和推演；运算推演完一类 S_i 数据后，可相应地删除部分无用数据，以节省内存空间。

6.1.2 最大 k_p 值数据群组的优化处理

同上理由，这部分数据也不适合内存存储处理，可通过（3.1）式中 n 值存储的调用进行反向计算最大 k_p 值，其计算量与首现特殊合数的计算量相同，只是计算的复杂度略高一些。其计算方法如下：

先按照（3.1）式反向计算上一次迭代的最大 S_i 值：

$S_i=10n+i$，设上次迭代的最大 n 值为 36，则 $S_1=361$，$S_3=363$，$S_7=367$，$S_9=369$；

以上一次迭代的最大 S_i 值为依据，反向计算最大 k_p 值：

根据式（4.1）推导：

$k_p=\lfloor（S_i-C_{f,\,i}）\div（10p）\rfloor$（6.1）

示例，$p=11$；

$S_1=361$，则：$C_{f,\,1}=121$，对应的 $k_{11}=\lfloor（361-121）\div 110\rfloor=\lfloor 2.18\rfloor=2$；

$S_3=363$，则：$C_{f,\,3}=33$，对应的 $k_{11}=\lfloor（363-33）\div 110\rfloor=3$；

$S_7=367$，则：$C_{f,\,7}=77$，对应的 $k_{11}=\lfloor（367-77）\div 110\rfloor=\lfloor 2.63\rfloor=2$；

$S_9=369$；则：$C_{f,\,9}=99$，对应的 $k_{11}=\lfloor（369-99）\div 110\rfloor=\lfloor 2.45\rfloor=2$；

对应表 5-1，上述计算结果完全正确。

6.1.3 参与推演素数数据量的优化处理

由于素数参与 PDM 推演时遵循由小到大的顺序，且采用单个素数依次参与的方式，因此，当需要巨量素数参与 PDM 推演时，可将素数集按从小到大的顺序划分为多个模块存储于硬盘，分块调入内存供 PDM 推演调用。前一块素数数据推演完成后，从内存中清除，再按顺序调入另一块素数数据模块，依此类推。通过增加数据调用次数的方式，换取空间效率的优化。而 PDM 中的数据调用频次本身较小，对时间效率的影响微乎其微。

6.2 时间复杂度分析与优化

影响 PDM 推演时间效率的主要因素有：数据访问、调入、清除及存储，逻辑运算与数学计算；其中，逻辑运算与数学计算是最为关键的影响因素。

PDM 推演的逻辑运算主要包括以下五类：D_s 的选择判断、素数大小顺序判断、特殊合数重复鉴别、推演范围边界判断、迭代状态与数据调度判断；其中仅有特殊合数重复鉴别属于逻辑运算冗余，且其逻辑运算量最大，其余四类均为必需的逻辑运算。因此，减少特殊合数重复鉴别的次数是优化逻辑运算的主要途径。

PDM 推演的数学计算主要涵盖以下六大类：迭代上限计算、迭代范围 n 值计算、特殊奇数 S_i 生成、最大 k_p 值反向计算、首现特殊合数 $C_{f,\,i}$ 计算、特殊合数 C_i 序列生成。其中，在特殊合数 C_i 序列的生成过程中，重复覆盖计算存在显著冗余，且其计算量在六类方法中位居首位（尽管相较于传统遍历计算筛法，仍有较大幅度的减少）；因此，降低特殊合数 C_i 序列生成过程中重复覆盖的计算量，还可以减少特殊合数重复鉴别

的逻辑次数，是进一步提升时间效率的有效途径。

根据特殊合数的构成规则，唯有以最小特殊素数因子 3 推演的特殊合数不存在重复，其他素因子推演的特殊合数均不同程度存在重复覆盖的情况。

定义 9：设素数 $p\in P$ 且 $p>3$，当 p 在小于其平方数的范围内作为特殊合数的素因子时，必然会与比它小的素数因子产生重复。本文将这类重复称为前置重复；将 p 在大于其平方数的范围内与比它小的素数因子产生的重复称为后置重复。此外，p 的数值越大时，它所产生的前置重复量也就越多。可以通过预置起始 k_p 值的方式，消除前置重复的计算。

改（6.1）式为向上取整数，将 p^2 替代式中的 S_i 即可得到推演 C_i 的起始 k_p 值：

$k_p=\lceil(p^2-C_{f,i})\div(10p)\rceil$（6.2）

示例：$p=109$，$p^2=11881$，

$C_{f,1}=109\times9=763$，$k_{109}=\lceil(11881-981)\div1090\rceil=\lceil10\rceil=10$

$C_{f,3}=109\times7=423$，$k_{109}=\lceil(11881-763)\div1090\rceil=\lceil10.2\rceil=11$

$C_{f,7}=109\times3=327$，$k_{109}=\lceil(11881-327)\div1090\rceil=\lceil10.6\rceil=11$

$C_{f,9}=109\times11=1199$，$k_{109}=\lceil(11881-1199)\div1090\rceil=\lceil9.8\rceil=10$

即通过预置起始 k_p 值，可减少素因子 109 共 42 次的前置重复计算量。当新增参与推演的素数数量庞大且大素数数值极高时，其减少的前置重复计算量将非常可观。

7 引理证明

本文提出了三个引理，现分别证明如下：

7.1 引理 1 证明

引理 1：对任意素因子 $p\in P$（其中 $p\notin\{2，5\}$），其在 S_i 中构成的 C_i 序列 $C_i(k)$ 的最小正周期为 10p。

由式（3.1）可得：

$S_i=\{10n+i\mid n\in\mathrm{N}^+,\ i\in\{1，3，7，9\}\}$，且 S_i 中的元素满足 $S_i\equiv i\pmod{10}$

S_i 中所有元素模 10 的余数恒为 i，深刻体现了个位数周期性的数学本质。

由式（4.3）可得：

$C_i(k)=\{p\times(\mathrm{D_s}+10\mathrm{k})\mid C_i(k)\in S_i$ 且 $p\mid C_i(k)\}$

p 为 $C_i(k)$ 中所有元素的公因子，因此 $C_i(k)$ 中所有元素均能被 p 整除。

7.1.1 周期性验证：

设 $x=C_i(k)$，则 $p\mid x$ 且 $x\equiv i\pmod{10}$。

构造 $x'=x+10p$，需证 $x'\in C_i(k)$：

整除性：因 $p\mid x$ 且 $p\mid 10p$，故 $p\mid(x+10p)=x'$；

属于 S_i：因 $10p\equiv0\pmod{10}$，故 $x'\equiv x\equiv i\pmod{10}$，即 $x'\in S_i$。

因此，$x'\in C_i(k)$，说明 $10p$ 是 $C_i(k)$ 的周期。

7.1.2 最小性验证：

假设存在更小的正周期 $T'<10p$，使得对任意 $x\in C_i(k)$，$x+T'\in C_i(k)$。

由整除性：$p\mid x$ 且 $p\mid(x+T')$，故 $p\mid T'$，即 $T'=kp$（$k\in\mathrm{N}^+$，$k<10$）；

由 S_i 归属：$x+T'\equiv i\pmod{10}$ 且 $x\equiv i\pmod{10}$，故 $T'\equiv0\pmod{10}$，即 $kp\equiv0\pmod{10}$。

因 $p\notin\{2，5\}$，$gcd(p，10)=1$（p 与 10 互质），故 $k\equiv0\pmod{10}$。但 $k<10$，矛盾。

因此，$10p$ 是 $C_i(k)$ 的最小正周期。引理 1 得证。

7.2 引理 2 证明

引理 2：对于任意素因子 $p\in P$（$p\notin\{2，5\}$），根据表 4-2 的 $\mathrm{D_s}$ 搭配规则，存在 $\mathrm{D_s}\in\{3，7，9，11\}$，使得 $C_{f,i}=p\times\mathrm{D_s}$ 成立。

7.2.1 核心约束：$C_{f,i}$ 需满足个位数匹配要求：

$C_{f,i}\equiv i$（mod10）（由于 $C_{f,i}\in S_i$）；

7.2.2 D_s 存在的唯一性

因 $p\in P$（$p\notin\{2，5\}$），$C_{f,i}\in S_i$。

设 P 的个位数为 j（$j\in\{1，3，7，9\}$），用 P_j 匹配表 2 中的 P_i。

仅当 $j\times D_s\equiv i$（mod10）成立时，$C_{f,i}\equiv i$（mod10）方才成立。

表 2 的匹配关系严格且唯一满足 $j\times D_s\equiv i$（mod10）的条件，故 $C_{f,i}\equiv i$（mod10）成立，即 D_s 的存在具有唯一性。

7.2.3 $C_{f,i}$ 的首现性

$C_{f,i}$ 的首现性由 D_s 的最小性所决定。在 $C_{f,i}\equiv i$（mod10）约束条件下，D_s 必为排除 1 和 5 后的最小奇数（由于 $1\times p$ 不构成特殊合数，故排除 1），$D_s=\{3，7，9，11\}$即为所有符合条件的最小奇数，因此 $C_{f,i}$ 必为首现。

综上分析，引理 2 得证。

7.3 引理 3 证明

引理 3：对于任何素因子 p（$p\in P$，$p\notin\{2，5\}$），依据 DLSCN 在全部 4 个 S_i 子集中，均能生成完整且无遗漏覆盖包含 p 的 C_i 序列。

其生成的序列 $C_i(k)=C_{f,i}+10kp$（$k\in\mathbb{N}\cup\{0\}$）恰好等同于 C_i——即 S_i 中所有包含 p 的特殊合数的集合。

需通过双向包含关系证明：$C_i(k)\subseteq C_i$ 且 $C_i\subseteq C_i(k)$

7.3.1 证明 $C_i(k)\subseteq C_i$

任取 $x\in C_i(k)$，则 $x=C_{f,i}+10kp$（$k\geqslant 0$）；

包含素因子 p：$x=p(D_s+10k)$，即 $p\mid x$，故 x 为合数；

属于集合 S_i：因 $C_{f,i}\equiv i$（mod10），且 $10kp\equiv 0$（mod10），故 $x\equiv i$（mod10），即 $x\in S_i$。

因此，$x\in C_i$，故 $C_i(k)\subseteq C_i$。

7.3.2 证明 $C_i\subseteq C_i(k)$

任取 $x\in C_i$，则 x 满足 $x\equiv i$（mod10）且 $x>10$；

作为含 p 的特殊合数，x 必满足 $p\mid x$ 且 $x\geqslant C_{f,i}$（其中 $C_{f,i}$ 是首现的最小含 p 的 S_i 元素）。

需证 x 可表示为 $C_{f,i}+10kp$（即 $x-C_{f,i}=10kp$，k 为非负整数）：

首先，差的整除性分析：因 $p\mid x$ 且 $p\mid C_{f,i}$，由整除的差性质可知 $p\mid(x-C_{f,i})$；

进一步，差 $x-C_{f,i}$ 必为 $10p$ 的倍数：由于 $p\notin\{2，5\}$，故 $gcd(10，p)=1$（10 与 p 互质），因此 $lcm(10，p)=10p$。结合 $x\equiv i$（mod10）及 $C_{f,i}\in S_i$（故 $C_{f,i}\equiv i$（mod10）），可知 $10\mid(x-C_{f,i})$。

设 $M=x-C_{f,i}$，根据公倍数性质，若 $10\mid M$ 且 $p\mid M$，则 $10p\mid M$，故 $10p\mid(x-C_{f,i})$，即 $x-C_{f,i}=10kp$（$k\in\mathbb{N}\cup\{0\}$，因 $x\geqslant C_{f,i}$，故 $k\geqslant 0$）。

因此，$x=C_{f,i}+10kp\in C_i(k)$，即 $C_i\subseteq C_i(k)$。

结合双向包含关系可知 $C_i(k)=C_i$，即 p 生成的序列完整无遗漏覆盖 C_i。引理 3 得证。

8 结语

本文立足素数研究的传统困境，通过研究范式的转换，构建了以特殊合数分布规律（DLSCN）为基础、递进推演法（PDM）为实践路径的素数研究框架，实现了从“无规律被动筛除合数”到“有规律主动推演合数”的突破，为素数理论与工程应用搭建了桥梁。

8.1 潜在应用价值

本文研究框架的核心价值在于工程化可行性：具体体现为两方面——一是依托明确的数学推演规律及自动迭代方式，可实现编程的自动化处理；二是凭借其规律性及对大量逻辑与计算冗余的剔除，以及可独立分段、分类推演运算的方法，在时空复杂度上具备优于现有所有素数筛法的潜力。

8.2 存在的不足

因笔者能力所限，未能将本研究进一步拓展至编程自动化应用、时空复杂度对比验证等方向。因此，本研究目前仅为纯理论推算模型，尚未经过实操检验。

笔者借助 AI 工具完成了多维度的时空复杂度对比验证，所有 AI 生成的结果均显示，在中大规模场景下，本方法均不同程度优于现有各类素数筛法。鉴于 AI 工具的验证结果尚未获得学界广泛认可与接受，下一步需通过人工实操进行验证。

8.3 后续研究深化方向

一是开发 PDM 全流程自动化程序，实现“素数集输入-特殊合数推演-素数输出”的端到端高效自动运算；

二是探索 PDM 与分布式计算的适配方案，进一步拓展超大范围素数筛选的能力，推动该理论在更多前沿领域的落地应用。

三是针对本方法在时空复杂度与迭代扩展能力上的表现，通过人工实操对比验证，客观真实地评估其应用拓展价值。并通过实际实验，不断优化各项参数。

致谢

笔者在论文研究过程中，得到了国防科技大学系统工程与数学系原主任罗建书教授、中南大学数学与统计学院侯木舟教授的悉心指导。二位不仅提出诸多极具建设性的建议，更在本研究的数学理论完善、逻辑结构搭建等关键环节提供了莫大助力。在此，谨向二位致以最衷心的感谢！同时，亦需特别感谢中山大学研究生游璐同学，为笔者提供了诸多宝贵的思路与协助。

参考文献

[1] 华罗庚. 数论导引[M]. 北京: 科学出版社, 1957:45-112.

[2] 王元. 论素数[M]. 北京: 高等教育出版社, 2018:56-72.

[3] 潘承洞, 潘承彪. 解析数论基础（第 2 版）[M]. 北京: 科学出版社, 2016:98-120.

[4] 王守峰. 初等数论[M]. 昆明: 云南大学出版社, 2020:45-62.

[5] 张尔光. 素数若干问题的分析与证明[M]. 长春: 东北师范大学出版社, 2016:78-95.

[6] Hardy G H, Wright E M. 数论导引（第 6 版）[M]. 张明尧, 张凡, 译. 北京: 人民邮电出版社, 2019:89-110, 345-362.

[7] Courant R, Robbins H, Stewart I. 什么是数学（第 4 版）[M]. 左平, 译. 上海: 复旦大学出版社, 2023:52-88.

[8] Friedlander J B, Iwaniec H. Opera de Cribro (2nd ed.)[M]. Providence: American Mathematical Society, 2017.28-215, 320-380.

[9] Atkin A O L, Bernstein D J. Prime sieves using binary quadratic forms[J]. Mathematics of Computation, 2003, 73(246):1023-1030.

[10] Selberg A. An elementary proof of the prime-number theorem[J]. Annals of Mathematics, 1949, 50(2):305-313.

[11] Erdős P. On a new method in elementary number theory which leads to an elementary proof of the prime number theorem[J]. Proceedings of the National Academy of Sciences, 1949, 35(7):374-384.

[12] Lemke Oliver, K., & Soundararajan, K. (2018). Improved quadratic sieve for large prime generation[J]. *Mathematics of Computation*, 87(312), 1693-1711.

[13] Bharti, K., Cervera-Lierta, A., Kyaw, T. H., et al. (2021). Quantum-assisted prime sieve with enhanced speed for cryptography[J]. *IEEE Transactions on Information Theory*, 67(12), 7998-8010.

[14] Pomerance, C., & Zhang, X. (2020). Distributed Selberg sieve for ultra-large prime screening[J]. *Journal of Computational and Applied Mathematics*, 376, 112835.

[15] Zhang, Y., Wang, L., & Li, H. (2022). Cryptography-focused fast sieve for RSA key generation[J]. *Cryptography and Communications*, 14(4), 789-806.

版权声明：©2025 作者与开放获取期刊研究中心(OAJRC)所有。本文章按照知识共享署名许可条款发表。
http://creativecommons.org/licenses/by/4.0/

OPEN ACCESS