

基于 **Langford** 序列超均匀性的 **$P \neq NP$** 启发式路径

Wino 研究组*

发布于 2025-09-02，修改于 2025-09-22

Langford 问题简介

将 m 组1到 n 的数字排成一个序列，使得任意两个相邻的数 k 之间恰好有 k 个数字 [1 - 5]。

设 $L(m, n)$ 表示在考虑反转对称性下的不同 Langford 序列的个数。我们有 $L(2, 3) = L(2, 4) = 1$ 以及 $L(3, 9) = 3$ ：

3 1 2 1 3 2

4 1 3 1 2 4 3 2

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

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

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

Langford 序列计数的渐近公式

猜想 1. Langford 序列计数 $L(m, n)$ 具有如下渐近公式 [6]:

$$L(m, n) \sim n! e^{-\ell(m)n},$$

其中 $\ell(m)$ 是仅依赖于 m 的指数系数, 即

$$\ell(m, n) = \frac{1}{n} \log \frac{n!}{L(m, n)}$$

在 $n \rightarrow \infty$ 时收敛到一个常数。

猜想 2. Langford 序列计数 $L(2, n)$ 的指数系数 $\ell(2, n)$ 收敛于 Taniguchi 常数:

$$\lim_{n \rightarrow \infty} \ell(2, n) = \prod_{p \in \mathbb{P}} \left(1 - \frac{3}{p^3} + \frac{2}{p^4} + \frac{1}{p^5} - \frac{1}{p^6} \right) = 0.678234491\dots,$$

其中乘积取遍所有素数 $\mathbb{P}$ 。更确切地说, 该特定值可近似为:

$$\ell(2, n) \simeq \prod_{i=1}^N \left(1 - \frac{3}{p_i^3} + \frac{2}{p_i^4} + \frac{1}{p_i^5} - \frac{1}{p_i^6} \right).$$

表 1. Langford 序列的个数 $L(2, n)$ ，参见 OEIS A014552。在参考文献 [7] 中，使用并行退火算法得到了近似值 $L(2, 31) \simeq 5.381 \cdot 10^{24}$ 和 $L(2, 32) \simeq 8.812 \cdot 10^{25}$ 。

n	exact		approximate		error
	$L(2, n)$	$\ell(2, n)$	$\ell(2, n)$	$L(2, n)$	
3	1	0.597253	0.765625	1	$\sim 0\%$
4	1	0.794513		1	$\sim 0\%$
7	26	0.752438		24	-7.7%
8	150	0.699246	0.701560	147	-2.0%
11	17792	0.701437		$1.777 \cdot 10^4$	-0.1%
12	108144	0.699666		$1.057 \cdot 10^5$	-2.2%

15	39809640	0.693310	0.687148	$4.367 \cdot 10^7$	+9.7%
16	326721800	0.691703		$3.514 \cdot 10^8$	+7.6%
19	256814891280	0.687803		$2.600 \cdot 10^{11}$	+1.3%
20	2636337861200	0.686760		$2.616 \cdot 10^{12}$	-0.8%
23	3799455942515488	0.684045	0.681745	$4.006 \cdot 10^{15}$	+5.4%
24	46845158056515936	0.683296		$4.862 \cdot 10^{16}$	+3.8%
27	111683611098764903232	0.681309		$1.104 \cdot 10^{20}$	-1.2%
28	1607383260609382393152	0.680745	0.680305	$1.627 \cdot 10^{21}$	+1.2%
31				$5.701 \cdot 10^{24}$	
32				$9.240 \cdot 10^{25}$	
35			0.679426	$4.861 \cdot 10^{29}$	
36				$8.871 \cdot 10^{30}$	

排列与 **Langford** 序列之间的关系

对于集合 $\{1, 2, \dots, n\}$ 的任意一个排列 $\sigma(n)$ ， 我们知道它的 Lehmer 编码构成一个阶乘进制数 x ， 该数可用于在字典序中索引该排列。

由于每个 Langford 序列都可以表示为一个排列， 例如序列 1 4 1 5 6 7 4 2 3 5 2 6 3 7 可以表示为 $(1, 4, 5, 6, 7, 2, 3)$ ， 因此产生了一个有趣的问题： 在索引区间 $[x, x + d)$ 内， 有多少个排列对应于 **Langford** 序列？

Langford 序列的超均匀分布

对于 Langford 配对问题 $\mathbb{L}(2, n)$, 配对比率 $r(n, d)$ 定义为

$$r(n, d) = \frac{n! \mu(n, d)}{2L(2, n)d},$$

其中 $\mu(n, d)$ 表示在索引区间 $[x, x + d)$ 上 Langford 序列数量的采样均值。从表 2 中可以看出,

$$r(n, d) \simeq 1 - \frac{1}{d} \quad \Rightarrow \quad \mu(n, d) \simeq 2(d - 1)e^{-\ell(2, n)n}.$$

表 2. 不同区间长度下的配对比率 $r(n, d)$ 。

$r(n, d)$	10^7 个样本		10^8 个样本		10^9 个样本	
	$n = 11$	$n = 12$	$n = 15$	$n = 16$	$n = 19$	$n = 20$
$d = 2$	0.497950	0.504275	0.504712	0.505264	0.504812	0.494178
$d = 5$	0.794028	0.802189	0.795189	0.798817	0.799836	0.811909
$d = 10$	0.898364	0.901406	0.894949	0.895355	0.897554	0.908530
$d = 20$	0.948910	0.950527	0.952901	0.951805	0.953246	0.943667
$d = 50$	0.979904	0.982732	0.983297	0.979322	0.976178	0.980197
$d = 100$	0.990496	0.991568	0.987725	0.986786	0.988295	0.989223
$d = 200$	0.995367	0.994959	0.993111	0.995348	0.995793	0.992925
$d = 500$	0.998519	0.998965	0.998034	1.001088	0.998281	0.997641
$d = 1000$	0.999376	0.999480	0.998831	1.000253	0.999208	0.999534

参考物理学中的概念，我们引入全局密度和局部密度，定义如下：

$$\rho_g(n) = \frac{2L(2, n)}{n!}, \quad \rho_l(n, d) = \frac{\mu(n, d)}{d}.$$

然后，我们有 $r(n, d) = \rho_l(n, d) / \rho_g(n)$ 。类似地，密度涨落定义为

$$s^2(n, d) = \frac{1}{N-1} \sum_{N \text{ 个样本}} [\rho_l(n, d) - \rho_g(n)]^2.$$

从表 3 可以看出， ρ_l 几乎保持不变，并且 s^2 随着 d 的增加而衰减，这些正是超均匀系统的主要特征。

表 3. 不同区间长度下的局部密度 $\rho_l(n, d)$ 和密度涨落 $s^2(n, d)$ 。所有计算使用的样本数量均为 $N = 10^8$ 。

$\rho_l(n, d)$	$n = 11$	$n = 12$	$n = 15$	$n = 16$	$n = 19$	$n = 20$
$s^2(n, d)$						
$d = 2$	$4.442 \cdot 10^{-4}$	$2.254 \cdot 10^{-4}$	$2.965 \cdot 10^{-5}$	$1.586 \cdot 10^{-5}$	$2.250 \cdot 10^{-6}$	$1.110 \cdot 10^{-6}$
	$2.221 \cdot 10^{-4}$	$1.127 \cdot 10^{-4}$	$1.482 \cdot 10^{-5}$	$7.927 \cdot 10^{-6}$	$1.125 \cdot 10^{-6}$	$5.550 \cdot 10^{-7}$
$d = 5$	$7.150 \cdot 10^{-4}$	$3.623 \cdot 10^{-4}$	$4.867 \cdot 10^{-5}$	$2.497 \cdot 10^{-5}$	$3.286 \cdot 10^{-6}$	$1.690 \cdot 10^{-6}$
	$1.483 \cdot 10^{-4}$	$7.533 \cdot 10^{-5}$	$1.008 \cdot 10^{-5}$	$5.172 \cdot 10^{-6}$	$6.788 \cdot 10^{-7}$	$3.476 \cdot 10^{-7}$
$d = 10$	$8.026 \cdot 10^{-4}$	$4.065 \cdot 10^{-4}$	$5.480 \cdot 10^{-5}$	$2.803 \cdot 10^{-5}$	$3.791 \cdot 10^{-6}$	$1.926 \cdot 10^{-6}$
	$8.740 \cdot 10^{-5}$	$4.418 \cdot 10^{-5}$	$5.932 \cdot 10^{-6}$	$3.021 \cdot 10^{-6}$	$4.089 \cdot 10^{-7}$	$2.046 \cdot 10^{-7}$
	$8.478 \cdot 10^{-4}$	$4.281 \cdot 10^{-4}$	$5.735 \cdot 10^{-5}$	$2.958 \cdot 10^{-5}$	$4.004 \cdot 10^{-6}$	$2.052 \cdot 10^{-6}$

$d = 20$	$4.829 \cdot 10^{-5}$	$2.435 \cdot 10^{-5}$	$3.238 \cdot 10^{-6}$	$1.666 \cdot 10^{-6}$	$2.236 \cdot 10^{-7}$	$1.128 \cdot 10^{-7}$
$d = 50$	$8.732 \cdot 10^{-4}$	$4.431 \cdot 10^{-4}$	$5.961 \cdot 10^{-5}$	$3.064 \cdot 10^{-5}$	$4.097 \cdot 10^{-6}$	$2.123 \cdot 10^{-6}$
	$2.258 \cdot 10^{-5}$	$1.130 \cdot 10^{-5}$	$1.486 \cdot 10^{-6}$	$7.560 \cdot 10^{-7}$	$9.892 \cdot 10^{-8}$	$5.089 \cdot 10^{-8}$
$d = 100$	$8.823 \cdot 10^{-4}$	$4.470 \cdot 10^{-4}$	$6.028 \cdot 10^{-5}$	$3.102 \cdot 10^{-5}$	$4.160 \cdot 10^{-6}$	$2.138 \cdot 10^{-6}$
	$1.257 \cdot 10^{-5}$	$6.221 \cdot 10^{-6}$	$8.075 \cdot 10^{-7}$	$4.122 \cdot 10^{-7}$	$5.359 \cdot 10^{-8}$	$2.739 \cdot 10^{-8}$
$d = 200$	$8.875 \cdot 10^{-4}$	$4.494 \cdot 10^{-4}$	$6.041 \cdot 10^{-5}$	$3.117 \cdot 10^{-5}$	$4.162 \cdot 10^{-6}$	$2.161 \cdot 10^{-6}$
	$7.578 \cdot 10^{-6}$	$3.694 \cdot 10^{-6}$	$4.576 \cdot 10^{-7}$	$2.330 \cdot 10^{-7}$	$2.982 \cdot 10^{-8}$	$1.539 \cdot 10^{-8}$
$d = 500$	$8.901 \cdot 10^{-4}$	$4.507 \cdot 10^{-4}$	$6.080 \cdot 10^{-5}$	$3.116 \cdot 10^{-5}$	$4.199 \cdot 10^{-6}$	$2.160 \cdot 10^{-6}$
	$3.857 \cdot 10^{-6}$	$1.852 \cdot 10^{-6}$	$2.196 \cdot 10^{-7}$	$1.091 \cdot 10^{-7}$	$1.392 \cdot 10^{-8}$	$7.056 \cdot 10^{-9}$
$d = 1000$	$8.906 \cdot 10^{-4}$	$4.512 \cdot 10^{-4}$	$6.078 \cdot 10^{-5}$	$3.118 \cdot 10^{-5}$	$4.216 \cdot 10^{-6}$	$2.171 \cdot 10^{-6}$
	$2.489 \cdot 10^{-6}$	$1.190 \cdot 10^{-6}$	$1.348 \cdot 10^{-7}$	$6.595 \cdot 10^{-8}$	$8.166 \cdot 10^{-9}$	$4.100 \cdot 10^{-9}$

我们接着考虑 Langford 序列中，其指数 x 属于模 k 的不同剩余类的个数：

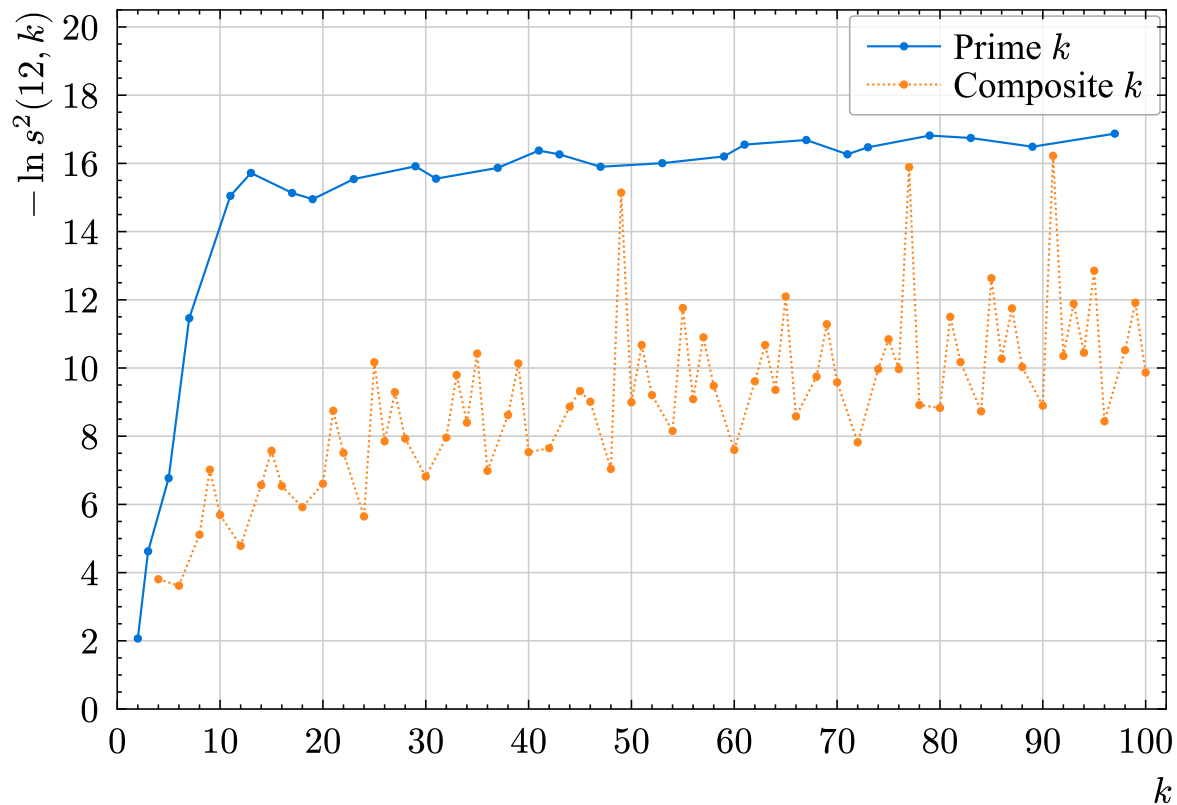
$$c(n, k, a) = \#\{x \in \mathbb{L}(2, n) : x \equiv a \pmod{k}\}.$$

在不同模数下的分布方差定义为：

$$s^2(n, k) = \frac{1}{k-1} \sum_{a=0}^{k-1} \left[\frac{c(n, k, a)}{2L(2, n)} - \frac{1}{k} \right]^2.$$

有趣的是，我们发现指数在不同剩余类中的分布均匀性与模数 k 的素因子个数密切相关。图 1 给出了 $n = 12$ 时的一个示例。

图 1. 在不同模数 k 下的负对数方差。



$P \neq NP$ 的启发式路径

对于任意给定的索引 x 和固定的长度 d , 在区间 $[x, x + d)$ 上寻找一个排列 $\sigma(n)$, 使其构成 Langford 配对问题 $L(2, n)$ 的一个解。

当 $d > 1 + \frac{1}{2}e^{\ell(2,n)n}$ 且 $n \rightarrow \infty$ 时, 我们可以预期无论 x 取何值, 在区间 $[x, x + d)$ 内至少存在一个解。验证这个解是容易的, 但由于 Langford 序列的分布具有超均匀性, 它应该不存在找到该解的高效算法, 除非我们穷举遍历该区间内所有 $O(e^{\ell n})$ 个排列。因此, 我们可以得出结论:
 $P \neq NP$ 。

参考文献

1. Langford CD. Problem 228. The Mathematical Gazette. 1958;42.
2. Miller JE. Langford's Problem. Available: <https://dialectrix.com/langford.html>
3. Walsh T. CSPLib Problem 024: Langford's Number Problem. Jefferson C, Miguel I, Hnich B, Walsh T, Gent IP, editors. Available: <https://www.csplib.org/Problems/prob024>
4. Krajecki M, Loiseau J, Alin F, Jaillet C. Many-Core Approaches to Combinatorial Problems: case of the Langford Problem. Super-

computing Frontiers and Innovations. 2016;3: 21 – 37. doi:10.14529/jsfi160202

5. Akgün Ö, Miguel I. Modelling Langford's Problem: A Viewpoint for Search. 2018. doi:10.48550/arXiv.1808.09847
6. Pan Z. Conjectures on the Number of Langford Sequences. 2021. doi:10.5281/zenodo.13824675
7. Assarpour A, Barnoy A, Liu O. Counting Skolem Sequences. 2015. doi:10.48550/arXiv.1507.00315

附录： Wino Studio 中的相关命令

- `oeis.langford.count_pairings(n, start, end)`
- `oeis.langford.find_pairings(n, start, end, count)`
- `oeis.langford.pairing_moduli(n, start, end, m)`
- `oeis.langford.estimate_pairings(n, d, samples)`
- `oeis.langford.pairing_density(n, d, samples)`
- `oeis.langford.pairing_ratio(n, d, samples)`

示例

输入: `oeis.langford.count_pairings(12, 1000000, 2000000)`

输出: **622**