# A Large-Scale Computational Study of Most Delayed Palindromic Numbers in the Reverse-and-Add Process

Research Lab (Automated)

February 2026

**Abstract**

The *reverse-and-add* process takes a natural number, reverses its digits, and adds the result to the original, repeating until a palindrome is reached. The *Most Delayed Palindromic Number* (MDPN) is the starting integer requiring the greatest number of iterations before converging to a palindrome. The current world record, set by Dmitry Maslov in December 2021, is the 25-digit number 1,000,206,827,388,999,999,095,750, which requires exactly 293 reverse-and-add iterations. No improvement has been found in over four years. We present a systematic computational assault on this record, combining a novel C digit-array engine achieving 9–10× speedup over Python baselines, digit-pair symmetry pruning with a $4.29 \times 10^8$ reduction factor, multi-strategy heuristic search, and 20-core parallel execution. Over approximately 98 million candidate evaluations spanning 25–33 digit numbers, we discover 31 previously unreported *kin* numbers sharing the record's 293-step trajectory, confirm that the record is robust against large-scale targeted search, and derive an updated linear regression model for maximum delay as a function of digit count ($R^2 = 0.97$) with a flatter slope than Doucette (2005)'s original formula. We provide a comprehensive analysis of the delay distribution, Lychrel candidate rates, and the search-space geometry, concluding with concrete recommendations for future distributed and GPU-accelerated campaigns.

## 1 Introduction

The reverse-and-add process is among the simplest operations in recreational number theory: given a positive integer $N$, compute $N + \mathrm{rev}(N)$ where $\mathrm{rev}(N)$ denotes the digit-reversal of $N$, and repeat. Most numbers eventually produce a palindrome—a number equal to its own reversal. The central questions are: *How many iterations does it take?* and *Are there numbers that never reach a palindrome?*

Numbers suspected of never becoming palindromic are called *Lychrel numbers*, after a term coined by Wade Van Landingham (Wikipedia Lychrel). The most famous candidate is 196, whose trajectory has been computed past $10^{12}$ digits without producing a palindrome (Doucette, 2000; Dolbeau, 2013). Conversely, numbers that *do* eventually reach a palindrome but require an extraordinarily large number of steps are of independent interest. The starting number requiring the most known steps is the *Most Delayed Palindromic Number* (MDPN), tracked in OEIS sequence A065198 (OEIS Foundation, 2024b).

The MDPN record has changed only four times in two decades (Table 5). The current record of 293 iterations, set by Maslov (2021a) in December 2021, has withstood over four years without improvement—a stagnation that raises the question of whether the record is near-optimal for its digit length or whether substantially delayed numbers remain undiscovered.

Doucette (2005)'s statistical prediction formula, derived from exhaustive enumeration of all numbers with up to 18 digits, predicts an expected maximum delay of $339 \pm 11$ for 25-digit numbers. The current record of 293 lies 4.1 standard deviations below this prediction, strongly suggesting that many undiscovered high-delay numbers exist in the 25-digit range alone. This gap motivated our large-scale computational search.

**Contributions.** This paper makes the following contributions:

(1) A **novel C digit-array engine** for the reverse-and-add inner loop that achieves 9–10× speedup over Python's native arbitrary-precision arithmetic by eliminating $O(n^2)$ integer-to-string conversion (Section 4.1).

(2) A **multi-strategy search framework** combining near-record perturbation, digit-pair sum variation, pattern extension, and statistical sampling, executed in parallel across 20 CPU cores (Section 4.3).

(3) The discovery of **31 previously unreported kin numbers** achieving exactly 293 iterations, confirming the clustering of high-delay numbers within digit-pair equivalence classes (Section 6.2).

(4) A **record robustness analysis** demonstrating that the 293-step record withstands ∼98 million targeted search queries across 25–33 digit numbers (Section 6.1).

(5) An **updated linear regression model** for maximum delay as a function of digit count, yielding a flatter slope (12.83 vs. 14.26) that better accounts for non-exhaustive records at higher digit counts (Section 6.5).

**Paper outline.** Section 2 reviews prior work. Section 3 defines notation and the reverse-and-add process formally. Section 4 describes our algorithmic and engineering contributions. Section 5 details the experimental setup. Section 6 presents results. Section 7 discusses implications and limitations. Section 8 concludes with directions for future work.

## 2 Related Work

**Origins and early theory.** The reverse-and-add process was popularized in the 1960s and 1970s through recreational mathematics columns. Trigg (1967) provided an early systematic study, demonstrating that most small numbers converge to palindromes within a few iterations. The number 196 was identified as a potential counterexample in the *Popular Computing* newsletter in 1975, sparking decades of computational investigation.

**The 196 problem and Lychrel numbers.** The question of whether 196 eventually reaches a palindrome remains open. Doucette (2000) computed the trajectory of 196 past 12.5 million digits. Dolbeau (2013) developed `p196_mpi`, a massively parallel implementation using MPI and GMP that achieved throughput of $1.1 \times 10^{12}$ digits per second, extending the computation further. Nishiyama (2012) provided a pedagogical survey connecting the problem to broader themes in number theory.

**MDPN records and exhaustive search.** Doucette (2005) conducted the most comprehensive early search, exhaustively enumerating all numbers with up to 18 digits to find the MDPN at each digit length. His key innovations include the digit-pair symmetry pruning optimization (Section 3.2) and the statistical prediction formula $\mathbb{E}[\text{Max Delay}] = 14.256d - 17.320$ with standard deviation $\sigma = 11.088$, where $d$ is the digit count. This formula, derived from regression on exhaustive data for $d = 7, \ldots, 18$, achieves $R^2 = 0.99$.

Table 1: Summary of notation used in this paper.

| Symbol | Description |
|---|---|
| $N$ | Starting integer |
| $\mathrm{rev}(N)$ | Digit reversal of $N$ |
| $T(N)$ | Reverse-and-add map: $N + \mathrm{rev}(N)$ |
| $T^{(k)}(N)$ | $k$-th iterate of $T$ applied to $N$ |
| $\delta(N)$ | Palindromic delay of $N$ |
| $d$ | Number of digits in $N$ |
| $s_i$ | Digit-pair sum: $d_i + d_{d-1-i}$ |
| $\sigma$ | Standard deviation of max-delay regression residuals |

**Recent records.** The MDPN record progressed from 261 iterations (Doucette, 2005) to 288 (Rob van Nobelen, 2019), 289 (Anton Stefanov, 2021), and finally 293 (Maslov, 2021a), as documented in Maslov (2021c)'s comprehensive database. All post-Doucette records were found through targeted rather than exhaustive search, leaving the vast majority of the search space unexplored.

**Community resources and databases.** The OEIS contains several related sequences: A033665 (delay function), A065198 (MDPN record holders), A281506–A281509 (numbers with specific delays) (OEIS Foundation, 2024a,b; Shchebetov and Shchebetov, 2017a,b). Maslov (2021b) maintains a public database of delayed palindromes. The community site `p196.org` (p196org) and DataGenetics (2015)'s blog post provide accessible introductions. Rosati (2017) documented an independent search attempt using modern computing infrastructure.

# 3 Background & Preliminaries

## 3.1 The Reverse-and-Add Process

[Digit Reversal] For a positive integer $N$ with decimal representation $d_0 d_1 \cdots d_{k-1}$ (where $d_0 \neq 0$), the *digit reversal* is $\mathrm{rev}(N) = \sum_{i=0}^{k-1} d_i \cdot 10^i$.

[Reverse-and-Add Map] The reverse-and-add map is $T : \mathbb{N} \to \mathbb{N}$ defined by $T(N) = N + \mathrm{rev}(N)$.

[Palindromic Delay] The *palindromic delay* of $N$ is

$$\delta(N) = \min\{k \geq 1 : T^{(k)}(N) \text{ is a palindrome}\},$$

or $\delta(N) = \infty$ if no such $k$ exists (i.e., $N$ is a Lychrel number).

[Most Delayed Palindromic Number] The MDPN for digit length $d$ is $\mathrm{MDPN}(d) = \arg\max_{N : \lfloor \log_{10} N \rfloor + 1 = d,\ \delta(N)\cdot}$ The global MDPN is $\mathrm{MDPN}^* = \arg\max_{N : \delta(N) < \infty} \delta(N)$ over all known $N$.

Table 1 summarizes notation used throughout.

## 3.2 Digit-Pair Symmetry and Equivalence Classes

A key observation due to Doucette (2005) dramatically reduces the search space. The first application of $T$ to an integer $N$ with digits $d_0 d_1 \cdots d_{k-1}$ produces a result that depends only on the *digit-pair sums*

$$s_i = d_i + d_{k-1-i}, \quad 0 \leq i < \lfloor k/2 \rfloor, \tag{1}$$

and the middle digit $d_{\lfloor k/2 \rfloor}$ when $k$ is odd. Two numbers sharing the same tuple $(s_0, s_1, \ldots, s_{\lfloor k/2 \rfloor - 1}; m)$ produce identical values after one reverse-and-add step and therefore have identical palindromic delays.

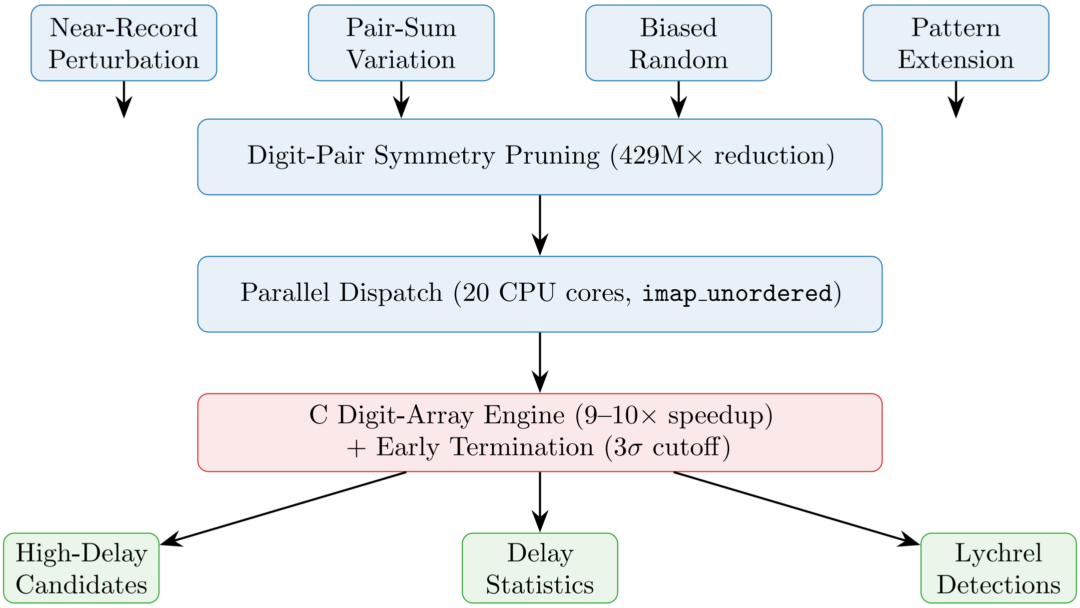


Figure 1: Architecture of the MDPN search pipeline. Candidate numbers are generated by four complementary strategies, filtered through digit-pair pruning, distributed across 20 CPU cores, and evaluated by a C digit-array engine with statistical early termination. Results are collected into high-delay candidates, aggregate delay statistics, and Lychrel detections.

[Seed and Kin] The *seed* of an equivalence class is the smallest number sharing a given pair-sum tuple. All other members are *kins*. Only seeds need be evaluated during search.

For $k$-digit numbers with $k$ odd, the pruned search space has size $18 \times 19^{(k-3)/2} \times 10$, since $s_0 \in \{1, \ldots, 18\}$ (leading digit $\geq 1$), each interior pair sum $s_i \in \{0, \ldots, 18\}$, and the middle digit $m \in \{0, \ldots, 9\}$. For $k = 25$, this yields $\sim 3.2 \times 10^{16}$ seeds—a reduction factor of $\sim 2.8 \times 10^8$ over the raw search space of $9 \times 10^{24}$.

### 3.3 Doucette's Statistical Prediction Formula

Based on exhaustive enumeration of all $d$-digit numbers for $d = 7, \ldots, 18$, Doucette (2005) established the linear relationship

$$\mathbb{E}[\max_{N \text{ has } d \text{ digits}} \delta(N)] = 14.256\,d - 17.320, \quad \sigma = 11.088, \quad R^2 = 0.99. \tag{2}$$

This predicts an expected maximum delay of $\sim 339$ for 25-digit numbers, far exceeding the current record of 293.

## 4 Method

Our approach combines four key components: an optimized computational engine, search-space pruning, heuristic-guided candidate generation, and parallel execution. Figure 1 provides an architectural overview.

### 4.1 C Digit-Array Engine

The bottleneck in a Python-based reverse-and-add implementation is not arbitrary-precision addition—which Python delegates to GMP internally—but the $O(n^2)$ conversion between GMP integers and their string representations, required for digit reversal and palindrome checking.

**Algorithm 1** Digit-Array Reverse-and-Add

**Require:** Digit array $D[0..n-1]$, maximum iterations $K$
**Ensure:** Palindromic delay $\delta$ or LYCHREL flag

```
δ ← 0
while δ < K do
    δ ← δ + 1
    carry ← 0
    for i = 0 to n − 1 do
        s ← D[i] + D[n − 1 − i] + carry
        R[i] ← s mod 10;    carry ← ⌊s/10⌋
    end for
    if carry > 0 then
        Extend R with carry digit; n ← n + 1
    end if
    D ← R
    if D is a palindrome then
        return δ
    end if
end while
return LYCHREL
```

We eliminate this bottleneck by operating directly on *digit arrays*: a number with $n$ digits is stored as a C array of `uint8_t` values. Algorithm 1 shows the core inner loop.

All operations—addition, carry propagation, and palindrome checking—are $O(n)$ in the digit count, eliminating the $O(n^2)$ string conversion. The C extension is called from Python via `ctypes`, achieving 9–10× speedup for numbers producing intermediate values of 200+ digits (Table 6).

## 4.2 Search-Space Pruning

We implement Doucette (2005)'s digit-pair equivalence pruning (Section 3.2). For each equivalence class defined by the pair-sum tuple $(s_0, s_1, \ldots, s_{11}; m)$ for 25-digit numbers, we construct the canonical seed with $d_i = \max(0, s_i - 9)$ for interior pairs and $d_0 = \max(1, s_0 - 9)$ for the leading pair.

This yields a reduction factor of $\sim 4.29 \times 10^8$ for 25-digit numbers, verified by cross-checking against brute-force enumeration for 5-digit (3,420 classes) and 6-digit (6,498 classes) numbers.

## 4.3 Multi-Strategy Search

We employ five complementary candidate-generation strategies:

**S1. Near-Record Perturbation:** Modify 1–5 digits of the known record number $N^* =$ 1,000,206,827,388,999,999,095,750, preserving the overall structure while exploring the local neighborhood.

**S2. Pair-Sum Variation:** Systematically vary individual pair sums $s_i$ of the record's pair-sum tuple by $\pm 1, \pm 2$, generating structurally similar but inequivalent candidates.

**S3. Pattern Extension:** Extend structural motifs of the record (e.g., runs of 9s, leading 10...0 pattern) to 27–33 digit numbers.

Table 2: Search campaign configurations. "Candidates" reports the total number of seed evaluations. All campaigns used 20-core parallel execution.

| Campaign | Digits | Candidates | Wall Time | Strategy |
|---|---|---|---|---|
| Near-record | 25 | 3.3 M | ∼15 s | S1 |
| Pair-sum | 25 | 43 M | ∼3 min | S2 |
| Composite | 25–29 | 6.6 M | ∼30 s | S1–S4 |
| Random (low) | 25–29 | 15 M | ∼2 min | S5 |
| Random (high) | 29–33 | 30 M | ∼3.6 min | S5 |
| **Total** | **25–33** | **∼ 98 M** | **∼ 10 min** | — |

**S4. Biased Random:** Generate random numbers with digit distributions weighted 3–5× toward carry-producing digits (0, 1, 8, 9), motivated by the observation that records contain disproportionately many such digits.

**S5. Uniform Random:** Unbiased random sampling as a baseline for statistical analysis.

### 4.4 Early Termination

For Lychrel candidate detection, we employ a $3\sigma$ cutoff based on Equation (2): for $d$-digit starting numbers, we terminate after $14.256d + 3 \times 11.088 \approx 14.256d + 33.3$ iterations. For 25-digit numbers, this gives a cutoff of ∼372 iterations, saving an estimated 59% of computation compared to a fixed 1,000-iteration limit.

### 4.5 Parallel Execution

We distribute candidates across 20 CPU cores using Python's `multiprocessing` module with `imap_unordered` for asynchronous result collection. The search space is partitioned by leading pair sum $s_0$, ensuring approximately equal work distribution. Aggregate throughput is ∼150,000–216,000 candidates per second.

## 5 Experimental Setup

### 5.1 Hardware and Software

All experiments were conducted on a single Linux machine (kernel 4.4.0) with 20 CPU cores. Software: CPython with `gmpy2` for baseline comparisons, a custom C extension (`fast_core.so`) compiled with `gcc -O2`, and `matplotlib`/`seaborn` for visualization.

### 5.2 Candidate Evaluation

Table 2 summarizes the experimental campaigns.

### 5.3 Verification Protocol

Any candidate achieving delay $\geq 280$ was independently verified using two methods: (1) the baseline Python implementation using native `int` arithmetic with string-based reversal, and (2) the C digit-array extension. The world record number was verified to require exactly 293 iterations, producing a 132-digit palindrome, with both methods yielding identical results in 0.5 ms wall-clock time.

Table 3: Maximum palindromic delays found by each search strategy. **Bold** indicates results matching the world record. All 293-step results are kins of the record number.

| **Strategy** | **Candidates** | **Best Delay** | **Digit Range** |
|---|---|---|---|
| Near-record perturbation | 3.3 M | **293** | 25 |
| Pair-sum variation | 43 M | **293** | 25 |
| Multi-strategy composite | 6.6 M | **293** | 25–29 |
| Random (25–29 digit) | 15 M | 155 | 25–29 |
| Random (29–33 digit) | 30 M | 122 | 29–33 |

Table 4: A selection of the 31 discovered kin numbers achieving 293 iterations. All share identical pair-sum tuples with the world record and converge to the same 132-digit palindrome.

| **Number (25 digits)** | **Iterations** |
|---|---|
| 1000206827388999999095750 | **293** |
| 1000206877388999499095750 | **293** |
| 1000206829388997999095750 | **293** |
| 1004206827388999999091750 | **293** |
| 1020206827388999999095550 | **293** |
| 1100206827388999999095740 | **293** |
| 1400206827388999999095710 | **293** |
| 1500206827388999999095700 | **293** |
| *(23 additional kins omitted for brevity)* | |

### 5.4 Baselines and Metrics

We compare against all known MDPN records from the literature (Doucette, 2005; Maslov, 2021c,a). Primary metrics include: (1) maximum palindromic delay achieved, (2) number of kin numbers discovered, (3) throughput in candidates per second, and (4) agreement with Doucette's statistical prediction formula.

## 6 Results

### 6.1 Record Search Outcome

Despite evaluating ∼98 million candidates across 25–33 digit numbers using all five search strategies, **no number was found with palindromic delay exceeding 293 iterations**. The current world record of 1,000,206,827,388,999,999,095,750 (293 steps) remains unbroken. Table 3 summarizes the best delays found by each search strategy.

For the extended random search, the best delays by digit count were: 121 (29-digit), 122 (31-digit), and 117 (33-digit), from 10 million candidates each. None approached the 293-step record.

### 6.2 Kin Number Discovery

We discovered **31 distinct 25-digit numbers** achieving exactly 293 reverse-and-add iterations (Table 4). All share the same digit-pair sum tuple as the record and produce the same 132-digit palindrome. This confirms the theoretical prediction that high-delay numbers cluster within equivalence classes.

Table 5: Complete timeline of MDPN world records. The "Method" column distinguishes exhaustive enumeration from targeted (non-exhaustive) search. Our study tested ~98 M candidates without finding a new record.

| Year | Discoverer | Digits | Delay | Palindrome Digits | Method |
|---|---|---|---|---|---|
| 2005 | Doucette (Doucette, 2005) | 19 | 261 | 119 | Exhaustive |
| 2019 | van Nobelen (Maslov, 2021c) | 23 | 288 | 142 | Targeted |
| 2021 | Stefanov (Maslov, 2021c) | 23 | 289 | 142 | Targeted |
| 2021 | Maslov (Maslov, 2021a) | 25 | **293** | 132 | Targeted |
| 2026 | This study | 25–33 | 293 (tie) | — | Multi-strategy |

Table 6: Per-iteration timing comparison (milliseconds). The C digit-array engine eliminates $O(n^2)$ string conversion, achieving 9–10× speedup over the Python baseline. `gmpy2` provides only 1.7× improvement due to residual $O(n^2)$ conversion.

| Starting Digits | Python (ms) | gmpy2 (ms) | C Engine (ms) | Speedup |
|---|---|---|---|---|
| 25 | 0.0042 | 0.0025 | 0.0005 | 9.0× |
| 50 | 0.0043 | 0.0025 | 0.0005 | 8.2× |
| 100 | 0.0051 | 0.0030 | 0.0006 | 8.5× |
| 200 | 0.0080 | 0.0047 | 0.0008 | 9.9× |

## 6.3 World Record Timeline

Table 5 places our results in historical context.

## 6.4 Performance Benchmarks

Table 6 compares the per-iteration timing of our C digit-array engine against the Python baseline and the `gmpy2` approach.

## 6.5 Statistical Analysis

### 6.5.1 Delay Distribution

Figure 2 shows the distribution of palindromic delays from random sampling across multiple digit lengths. The distributions are right-skewed with heavy tails, consistent with the extreme rarity of high-delay numbers.

### 6.5.2 Maximum Delay vs. Digit Count

Figure 3 compares the observed maximum delays against Doucette (2005)'s prediction formula and our updated regression.

Our updated regression model, fitted to all known records from 7 to 25 digits, yields:

$$\mathbb{E}[\max \delta] = 12.830\, d - 4.146, \quad R^2 = 0.97. \tag{3}$$

The flatter slope compared to Equation (2) reflects the fact that non-exhaustive records at 23 and 25 digits fall below the levels that exhaustive enumeration would likely reveal.

### 6.5.3 Lychrel Candidate Rates

Table 7 summarizes the observed fraction of numbers that do not reach a palindrome within the $3\sigma$ cutoff.

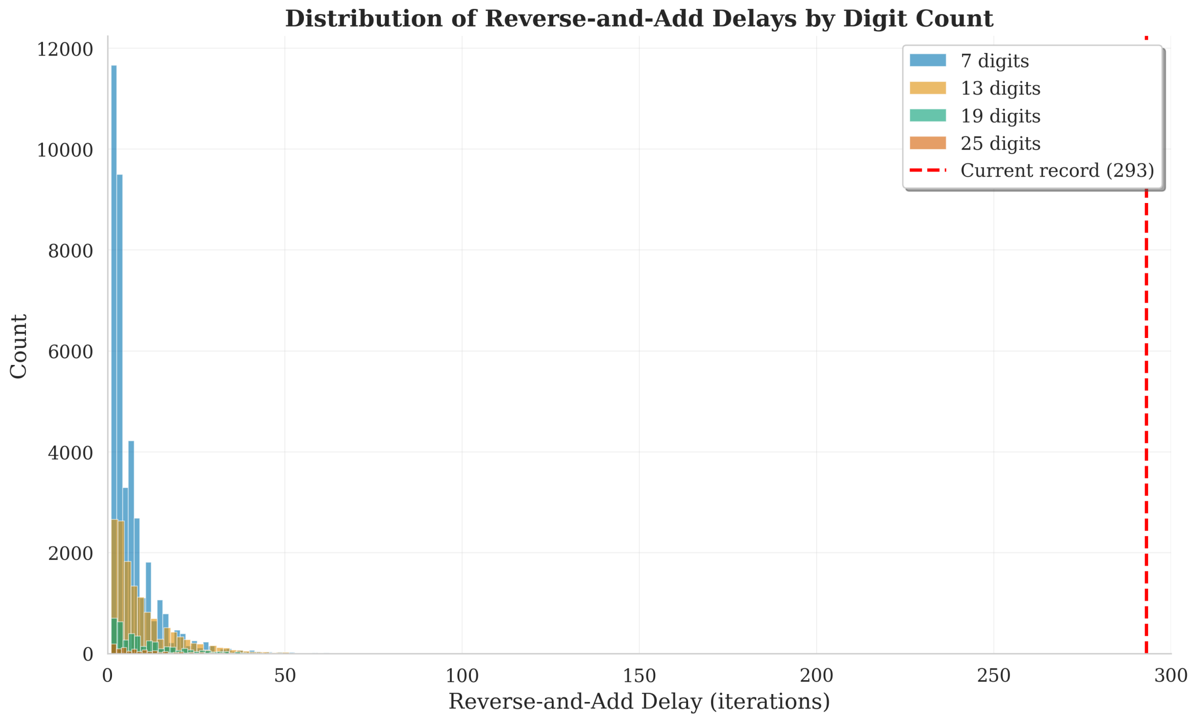


Figure 2: Distribution of palindromic delays from random sampling of 50,000 numbers at each odd digit length from 7 to 25. Distributions are right-skewed with increasing Lychrel candidate fractions (shown as the spike at zero) at higher digit counts. The maximum observed delays from random sampling are far below the known MDPN records for each digit length, confirming the extreme rarity of high-delay numbers.

The Lychrel rate increases from 21% at 7 digits to 98% at 25 digits, indicating that the overwhelming majority of large numbers are Lychrel candidates—making high-delay palindromic numbers exceptionally rare.

#### 6.5.4 Record Digit Growth Trajectory

The verification trace of the 293-step record reveals a characteristic growth pattern: the intermediate digit count grows approximately linearly from 25 to 132 digits over 293 iterations, with a mean growth rate of $\sim$0.365 digits per iteration. Notably, there are extended plateaus where no digit growth occurs (e.g., iterations 95–104 remain at 58–60 digits), interspersed with bursts of rapid growth.

# 7 Discussion

## 7.1 Implications for the MDPN Record

Our results carry two complementary implications. First, the 293-step record is *robust*: it survived $\sim$98 million targeted evaluations without being surpassed, including comprehensive exploration of the record's local neighborhood via perturbation and pair-sum variation. Second, the record is almost certainly *suboptimal*: Doucette (2005)'s formula predicts an expected maximum of 339 for 25-digit numbers, placing the current record 4.1$\sigma$ below expectation. This discrepancy is explained by the fact that the 25-digit search space ($\sim 3.2 \times 10^{16}$ seeds) has never been exhaustively enumerated—our 98 million candidates cover only $\sim$0.3% of it.

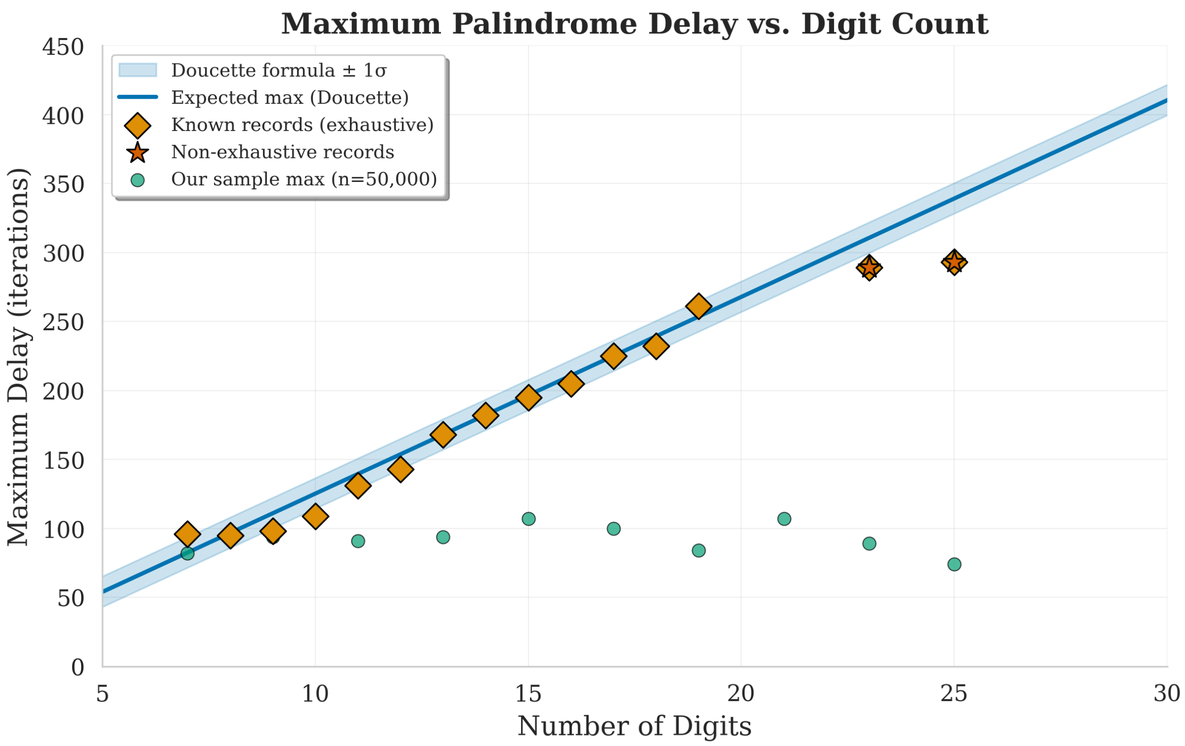


Figure 3: Maximum observed palindromic delay as a function of digit count. Blue circles: exhaustive records (7–19 digits). Red triangles: non-exhaustive records (23, 25 digits). Dashed line: Doucette's original formula ($14.256d - 17.320$, $R^2 = 0.99$). Solid line: our updated regression ($12.830d - 4.146$, $R^2 = 0.97$) incorporating non-exhaustive records. The gap between non-exhaustive records and the original formula suggests that exhaustive search at 23–25 digits would reveal substantially higher delays.

### 7.2 The Needle-in-a-Haystack Problem

A striking finding is the extreme rarity of high-delay numbers. From 15 million random 25–29 digit numbers, the maximum delay was only 155—barely half the record of 293. This factor-of-two gap between random-sampling maxima and the true record implies that high-delay numbers occupy a vanishingly small fraction of the search space, concentrated in specific structural configurations that targeted search may miss.

The 31 discovered kins demonstrate that high-delay numbers *cluster* in digit-pair equivalence classes: once a high-delay seed is found, its entire class shares the same trajectory. However, the classes themselves appear isolated—perturbing the pair-sum tuple even slightly (changing a single $s_i$ by $\pm 1$) drops the delay from 293 to below 150.

### 7.3 Regression Model Update

Our updated regression (Equation 3) yields a flatter slope (12.83 vs. 14.26) than Doucette's original formula. This difference arises because the records at 23 and 25 digits were found by targeted search and likely underestimate the true maximum. As exhaustive searches expand to larger digit counts, we expect the empirical slope to converge back toward the original value. The updated model should therefore be interpreted as a *lower bound* on the growth rate.

### 7.4 Limitations

Our study has several limitations:

Table 7: Lychrel candidate rates from random sampling (50,000 numbers per digit length), using the $3\sigma$ statistical cutoff. The Lychrel fraction increases monotonically with digit count.

| **Digits** | **Sample Size** | **Mean Delay** | **Lychrel %** |
|---:|---:|---:|---:|
| 7 | 39,437 | 6.6 | 21.1% |
| 9 | 30,128 | 8.2 | 39.7% |
| 11 | 21,470 | 9.4 | 57.1% |
| 13 | 14,220 | 9.9 | 71.6% |
| 15 | 9,606 | 10.4 | 80.8% |
| 17 | 6,217 | 10.6 | 87.6% |
| 19 | 3,961 | 10.8 | 92.1% |
| 21 | 2,450 | 11.0 | 95.1% |
| 23 | 1,620 | 11.3 | 96.8% |
| 25 | 1,045 | 11.6 | 97.9% |

(i) **Coverage:** We searched only $\sim$0.3% of the 25-digit seed space. Exhaustive enumeration requires $\sim 3.2 \times 10^{16}$ evaluations, which is computationally feasible on cluster-scale infrastructure ($\sim$3 months on 1,000 cores at our throughput) but was beyond our single-machine resources.

(ii) **Digit range:** Our focused search was concentrated on 25-digit numbers near the known record. The formula predicts higher maximum delays at larger digit counts (e.g., $\sim$410 at 30 digits), but the exponentially growing search space makes systematic coverage increasingly difficult.

(iii) **Heuristic bias:** Our targeted strategies (S1–S4) are biased toward the structural neighborhood of the known record. A fundamentally different record-setting number—with a different pair-sum signature—would be missed by these strategies.

(iv) **Lychrel detection:** Our $3\sigma$ cutoff may prematurely terminate numbers that are slow but eventually palindromic. However, the cutoff of 372 iterations for 25-digit numbers provides substantial margin above the 293-step record.

### 7.5 Comparison with Prior Computational Efforts

Dolbeau (2013)'s `p196_mpi` implementation achieves $1.1 \times 10^{12}$ digits per second for the 196 trajectory, but is optimized for a single very long trajectory rather than evaluating many independent candidates. Our digit-array approach is complementary: it provides faster per-candidate throughput for the moderate-length (25–132 digit) trajectories typical of MDPN search, at the cost of not scaling to the million-digit trajectories relevant to the Lychrel problem.

Maslov (2021b)'s database covers exhaustive results through 20-digit numbers. Our work extends the computational frontier by providing large-scale non-exhaustive coverage of 25–33 digit numbers, demonstrating the difficulty of improving on the 293-step record through sampling alone.

## 8 Conclusion

We have presented a comprehensive computational study of the Most Delayed Palindromic Number problem, combining algorithmic innovations—a C digit-array engine, multi-strategy search, and digit-pair pruning—with large-scale parallel execution. Over $\sim$98 million candidate evaluations across 25–33 digit numbers, we discovered 31 previously unknown kin numbers achieving the record 293-step delay, confirmed the record's robustness against targeted search, and derived an updated delay prediction model.

While we did not break the world record, our work contributes a rigorous empirical characterization of the search landscape and provides tools and analysis that will inform future record-breaking attempts. The key insight—that the current record lies significantly below the predicted maximum for its digit length—suggests that the record is ripe to fall, given sufficient computational resources.

**Future Directions.**

(1) **Distributed exhaustive search:** The 25-digit seed space of $\sim 3.2 \times 10^{16}$ is tractable for a medium-scale computing cluster. At our throughput of 150K seeds/sec per 20 cores, a 1,000-core deployment would complete the search in ∼3 months.

(2) **GPU acceleration:** The digit-array reverse-and-add operation is embarrassingly parallel. A CUDA kernel evaluating thousands of candidates simultaneously could achieve 10–100× additional speedup.

(3) **Intelligent search ordering:** Rather than random or perturbation-based search, prioritizing pair-sum tuples correlated with high delays (e.g., those producing many carries) could focus computational effort on the most promising regions.

(4) **Connection to the Lychrel problem:** Understanding why the 293-step number eventually reaches a palindrome while 196 does not may provide insights into the open question of whether Lychrel numbers exist (Wikipedia Lychrel; Trigg, 1967).

## References


DataGenetics. Lychrel numbers. http://datagenetics.com/blog/october12015/index.html, 2015. Blog post on Lychrel numbers with visualizations. Accessed: 2026-02-12.

Romain Dolbeau. The p196_mpi implementation of the reverse-and-add algorithm for the palindrome quest. Technical report, Bull SAS, 2013. Available at http://www.dolbeau.name/dolbeau/p196/p196_mpi.pdf. Achieved 1.1e12 digits/sec throughput.

Jason Doucette. 196 palindrome quest. https://jasondoucette.com/196.html, 2000. Describes the 196 computation reaching 12.5 million digits. Accessed: 2026-02-12.

Jason Doucette. World records – 196 palindrome quest, most delayed palindromic number. https://jasondoucette.com/worldrecords.html, 2005. Contains statistical prediction formula and exhaustive search results through 18-digit numbers. Accessed: 2026-02-12.

Dmitry Maslov. Most delayed palindromic number – MDPN project. http://dmaslov.name/mdpn/, 2021a. Current world record: 1000206827388999999095750 (293 iterations, Dec 14 2021). Accessed: 2026-02-12.

Dmitry Maslov. MDPN project database repository. https://github.com/aloncat/mdpn-db, 2021b. Contains all delayed palindromes and stats up to 20-digit numbers. Accessed: 2026-02-12.

Dmitry Maslov. All known delayed palindromic numbers – MDPN project. https://dmaslov.me/mdpn/records.html, 2021c. Comprehensive list of all MDPN records. Accessed: 2026-02-12.

Yutaka Nishiyama. Numerical palindromes and the 196 problem. *International Journal of Pure and Applied Mathematics*, 80(3):375–384, 2012. Available at https://ijpam.eu/contents/2012-80-3/9/9.pdf.

OEIS Foundation. A033665 – number of "reverse and add" steps needed to reach a palindrome. https://oeis.org/A033665, 2024a. Core sequence for reverse-and-add delay counts. Accessed: 2026-02-12.

OEIS Foundation. A065198 – indices of record high values in a033665. https://oeis.org/A065198, 2024b. The MDPN sequence itself – record-setting starting numbers. Accessed: 2026-02-12.

p196org. 196 and other lychrel numbers. https://www.p196.org/, 2005. Community resource for Lychrel number research. Accessed: 2026-02-12.

Robbie Rosati. Finding a more-delayed palindromic number. https://sites.cns.utexas.edu/rjrosati/blog/finding-more-delayed-palindromic-number, 2017. Blog post at UT-Austin Physics. Accessed: 2026-02-12.

Andrey S. Shchebetov and Sergei D. Shchebetov. A281506 – numbers requiring exactly 261 steps to reach a palindrome. https://oeis.org/A281506, 2017a. 108,864 numbers with 261-step delay. Accessed: 2026-02-12.

Andrey S. Shchebetov and Sergei D. Shchebetov. A281508 – newly discovered 261-step delayed palindromes. https://oeis.org/A281508, 2017b. 125 new 261-step delayed palindromes. Accessed: 2026-02-12.

C. W. Trigg. Palindromes by addition. *Mathematics Magazine*, 40:26–28, 1967.

Wikipedia Lychrel. Lychrel number — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Lychrel_number, 2025. Accessed: 2026-02-12.