Structural Properties of Move Sets and Modular Patterns in the Game of Nim

Niyanth Chandrasekaran

August 2025

1 Introduction

Nim is one of the most fundamental impartial combinatorial games. Despite its simple rules, the game exhibits rich mathematical structure. A central object of study in Nim is the sequence of winning and losing positions, which often exhibits a periodic or modular pattern depending on the chosen move set.

In its most basic form, it is played with one or more piles of objects (such as stones or matches). Two players take turns removing objects from a pile, with the restriction that each move must remove at least one object. The player who takes the last object wins.

The game is important in mathematics and computer science because its structure can be analyzed using modular arithmetic, parity, and recurrence relations. For one-pile Nim with custom move sets (for example, the allowed moves might be $\{1, 3, 4\}$), the sequence of winning and losing positions becomes eventually periodic. The repeating cycle of this sequence is called the *modular pattern*, and the length of this cycle (the *mod pattern length*) is the central focus of this research.

Research Question. What structural properties of a move set influence the mod pattern length in the game of Nim?

I approach this problem both computationally and theoretically. Using brute-force simulations, we generate the win/loss sequences for a wide variety of move sets and detect their repeating modular patterns. We then analyze how properties such as the greatest common divisor (GCD), least common multiple (LCM), spacing between moves, and the presence of "small" moves (such as 1 or 2) influence the periodicity.

My results suggest that several structural properties strongly affect mod pattern length:

- Higher GCD values correlate with shorter modular patterns.
- Larger LCM values often correlate with longer modular patterns.
- Move sets with evenly spaced values exhibit shorter, predictable cycles, while sets with wide gaps tend to produce longer, irregular cycles.
- Including very small moves (such as 1 or 2) typically shortens the cycle.

The importance of this research lies in connecting empirical simulations with number-theoretic structure and in opening the door for algorithmic and machine learning methods to predict pattern length with higher accuracy.

2 Background and Definitions

2.1 Nim Basics

The game of Nim begins with a pile of stones and a fixed move set

$$M = \{m_1, m_2, \ldots, m_k\}.$$

Two players alternate turns; on a turn a player removes m stones for some $m \in M$. The player unable to move loses.

A position (pile size n) is *winning (W)* if the player to move can force a win, and *losing (L)* otherwise.

2.2 Modular Patterns

For a given move set, the sequence of W/L outcomes eventually becomes periodic. That is, there exists some p such that for all sufficiently large n, the sequence repeats with period p. We refer to this repeating cycle as the *mod pattern* and to p as the *mod pattern length*.

2.3 Structural Properties Considered

We define the following numerical properties of a move set M:

- **GCD:** $\gcd(M)$, the greatest common divisor of all moves.
- **LCM:** $\mathrm{lcm}(M)$, the least common multiple of the moves.
- **Range:** $\max(M) - \min(M)$.
- **Average spacing:** the mean difference between consecutive moves when sorted.
- **Presence of 1 or 2:** whether the set contains 1 or 2, the "smallest" possible moves.

My study focuses on how these properties influence the mod pattern length.

3 Methodology

I implemented a Java program to:

1. Generate win/loss tables for pile sizes up to 1000 using dynamic programming.
2. Detect repeating cycles by finding the earliest start point and the smallest repeating length p.
3. Collect statistics (start index, cycle length) across many move sets, primarily of the form $(1, a, b)$ with $1 < a < b \leq 20$, and also broader sets of random combinations.
4. Analyze heuristics by correlating measured mod pattern lengths with properties such as GCD, LCM, and spacing.

Additionally, I experimented with AI/ML models (decision trees and random forests) trained on this data, with the goal of automatically predicting mod pattern length given a move set.

3.1 AI Model Design

The model was trained on thousands of generated move sets, each labeled with its computed mod pattern length. Each move set was transformed into a feature vector consisting of:

- GCD of the moves
- LCM of the moves
- Maximum, minimum, and range
- Average spacing between moves
- Presence of "small" moves (indicator features for 1 and 2)

The decision tree and random forest classifiers were then trained on these features, learning to predict the modular cycle length directly from the structural properties of the move set.

3.2 Training and Evaluation

The dataset was split into training and test subsets to evaluate generalization. The random forest model in particular showed strong performance, achieving an R^2 value of approximately 0.81, meaning it explained over 80% of the variance in pattern lengths. While not perfect, this demonstrates that the chosen structural features capture most—but not all—of the complexity governing Nim periodicity. Outliers tended to arise when irregular alignments of moves produced cycles not easily captured by arithmetic properties alone.

4 Results

4.1 General Trends

- **GCD effect.** For sets with $\gcd(M) > 1$, mod pattern length was consistently shorter. Example:

$$M = \{4, 8, 16\} \text{ (gcd} = 4) \rightarrow \text{short repeating cycle.}$$

$$M = \{3, 5, 7\} \text{ (gcd} = 1) \rightarrow \text{longer, less predictable cycle.}$$

- **LCM effect.** Larger LCM values tended to correlate with longer cycles. Example:

$$M = \{1, 2, 20\},\ \text{lcm} = 20 \quad \text{produced a much longer cycle than } M = \{1, 2, 3\}.$$

- **Spacing effect.** Move sets with small gaps (e.g., $\{1, 2, 3\}$) consistently produced short cycles, while widely spaced sets (e.g., $\{1, 10, 20\}$) produced longer cycles.

- **Presence of 1 or 2.** Including 1 (or even 2) almost always reduced cycle length, as it forces dense coverage of losing positions.

4.2 Example Data

$$\{1, 2, 3\} \to \text{pattern length 4.}$$
$$\{1, 3, 4\} \to \text{pattern length 3.}$$
$$\{1, 2, 20\} \to \text{pattern length 30.}$$
$$\{4, 8, 16\} \to \text{pattern length 4.}$$

5 Analysis

I attempted to formalize correlations:

- **Hypothesis 1.** Higher GCD $\to$ shorter cycles. Supported by data.

- **Hypothesis 2.** Larger LCM $\to$ longer cycles. Supported but with exceptions.

- **Hypothesis 3.** Average spacing strongly predicts cycle length. Supported.

- **Hypothesis 4.** Presence of 1 or 2 shortens cycles. Strongly supported.

Interestingly, while these heuristics explain many cases, there are outliers where irregular alignments of moves disrupt the expected pattern. This suggests that purely arithmetic properties (like GCD/LCM) are not sufficient; structural alignment also matters.

6 Discussion

My findings suggest that mod pattern length in Nim is governed by a combination of arithmetic and structural properties:

- GCD acts as a divisor constraint on possible cycles.
- LCM sets an upper bound or scaling factor for cycle length.
- Spacing governs the density of losing positions.
- Presence of 1 or 2 ensures coverage, reducing length.

However, predicting exact pattern length remains challenging. Outliers occur because losing positions are not distributed uniformly—they depend on recursive coverage by previous winning moves.

I tested a machine learning approach (decision trees, random forests) trained on thousands of combinations, which improved predictive accuracy over hand-crafted formulas. This suggests an avenue for future research: a hybrid of mathematical structure and data-driven prediction.

7 Conclusion and Open Problems

Conclusion. Structural properties such as GCD, LCM, spacing, and presence of small moves strongly influence mod pattern length in Nim. While higher GCD and small moves shorten cycles, larger LCM and wider spacing lengthen them. However, irregular alignments make exact prediction nontrivial.

Open Problems.

1. Can a closed-form formula be proven for mod pattern length given any move set?
2. What is the precise relationship between LCM and the maximum possible cycle length?
3. Can machine learning fully predict mod pattern length, or are there inherent limitations?
4. How do these results extend to 4+ move sets, or multi-pile variants of Nim?