Computational Verification and Structural Analysis of Modular Win-Table Patterns in Custom Nim Variants

Vihaan Mehta[1] and William Gasarch[2]

[1]Howard High School
[2]Department of Computer Science, University of Maryland, College Park

Abstract

Nim is one of the oldest and most thoroughly studied games in combinatorial game theory, yet generalized variants in which the move set is extended beyond $\{1, 2, 3, \dots\}$ still produce win/loss sequences whose modular structure is not always obvious from the move set alone. In this paper I generalize a Python win-table generator and pattern-detection routine to arbitrary move sets and use it to (1) computationally verify and refine a collection of modular winning-condition rules I derived by hand for several families of three-, four-, and five-move Nim variants, and (2) measure, for the first time with a fixed and documented methodology, what fraction of move sets of the form $\mathrm{NIM}(1, a)$, $\mathrm{NIM}(1, a, b)$, $\mathrm{NIM}(1, a, b, c)$, and $\mathrm{NIM}(1, a, b, c, d)$ produce a winning/losing sequence whose repeating pattern begins at index 0, as a function of the largest allowed move N. The results show that single-extra-move variants are essentially always periodic from the very first position, while adding additional move parameters steadily reduces the fraction of variants with a pattern starting at index 0, and that this reduction is sensitive to the length of the win table used to search for the pattern. I also show that several of my hand-derived modular rules, while individually correct, originally reported a modulus that was a multiple of the true minimal period, and I report the corrected, minimal moduli here along with the verification code.

1 Background and Introduction

Nim is one of the most fundamental impartial combinatorial games. In its simplest form, two players alternate removing objects from a single pile, with the restriction that each move must remove at least one object, and the player who removes the last object wins. Despite this simplicity, the game has been studied for well over a century because the sequence of winning and losing pile sizes encodes rich number-theoretic structure.

The game was given its modern name and its first complete mathematical treatment by the Harvard mathematician Charles L. Bouton, who in 1901 showed that a position in ordinary multi-pile Nim is losing for the player about to move if and only if the bitwise XOR (the "Nim-sum") of the pile sizes is zero [1]. Bouton's analysis was later generalized by Sprague [2] and Grundy [3], whose independent work established what is now called the Sprague–Grundy theorem: every impartial game under the normal play convention is equivalent to a single Nim pile of a size determined by a recursively-defined "Grundy value." A detailed modern treatment of this theory, including many Nim variants and their generalizations, can be found in Berlekamp, Conway, and Guy's *Winning Ways for Your Mathematical Plays* [4] and in Siegel's *Combinatorial Game Theory* [5].

This paper studies a restricted but still rich family of one-pile Nim variants: games in which the set of legal moves $A = \{a_1, a_2, \dots, a_k\}$ is a fixed, finite set of positive integers, and a position n

(the number of stones remaining) is a winning position for the player to move if and only if there exists some $a_i \in A$ with $n - a_i \geq 0$ and $n - a_i$ a losing position. It is a standard fact that for any finite move set A, the resulting sequence of win/loss labels is eventually periodic [5]; the question this paper investigates empirically is *how* that periodicity behaves as the move set grows in size and in the magnitude of its largest element, and what closed-form modular rules can be extracted for specific structured families of move sets.

2 Methods

2.1 Win Table Generator

All win tables in this paper were generated with a Python program I wrote, extending an earlier three-move version of the generator (which took a fixed move set $\{x, y, z\}$ and printed the win/loss table for stone counts from 1 to a user-specified maximum) into a version that accepts an arbitrary move set of any size. For a move set A and a maximum pile size M, the table is built with the standard recursive rule

$$\mathrm{win}(n) = \bigvee_{a \in A,\ a \leq n} \neg\, \mathrm{win}(n - a), \qquad \mathrm{win}(0) = \mathrm{false},$$

computed bottom-up for $n = 1, \ldots, M$.

2.2 Pattern Detection

A second routine searches a computed win table for the earliest start index $s \geq 0$ and the smallest period $p \geq 1$ such that $\mathrm{win}(n) = \mathrm{win}(n + p)$ for all $n \geq s$ within the table. The routine checks start indices in increasing order and, for each start index, checks periods in increasing order, so the period and start index it reports are always the *minimal* ones consistent with the table — this is an important methodological point, because several of the modular rules I had derived by hand (see Section 4) had reported a valid but non-minimal modulus.

2.3 Percentage Analysis

To measure how often a pattern starts at index 0, I wrote four functions analogous to the "twoPilePercent" through "fivePilePercent" style of analysis: for a fixed smallest move (here always 1, so that the move set is $\mathrm{NIM}(1, a, \ldots)$) and a maximum value N for the largest move, the functions iterate over every valid combination of additional moves not exceeding N, build the win table for each combination out to a fixed table length, and record whether the detected pattern begins at index 0. The reported percentage is (number of combinations with a pattern starting at index 0) / (total combinations tested) $\times$ 100. Unless otherwise noted, a win table length of 500 was used, following the same table length used for qualitative inspection of individual move sets.

3 Results

3.1 Single-Parameter Variants: NIM(1, a)

For move sets of the form $\{1, a\}$ with a ranging from 2 to 50, I tested whether the detected pattern starts at index 0 under two different table lengths. With a table length scaled to at least $4a$, every single value of a from 2 to 50 produced a pattern starting at index 0 (100%). However, with a *fixed* table length of 100, the percentage for $a = 2, \ldots, 50$ drops to 97.959%, with the single failure occurring at exactly $a = 50$ — the case where the win table's own period is large enough that a table of length 100 is not long enough to confirm two full periods.[1] Lengthening the table to three times the largest move size (150, for $a = 50$) restores a detected pattern starting at index 0 for $a = 50$ as well, confirming that the apparent miss is an artifact of an insufficiently long search window rather than a genuine break in periodicity. Figure 1 visualizes this: across all a from 2 to 50, a fixed table length of 100 detects the index-0 pattern everywhere except at the single point $a = 50$.

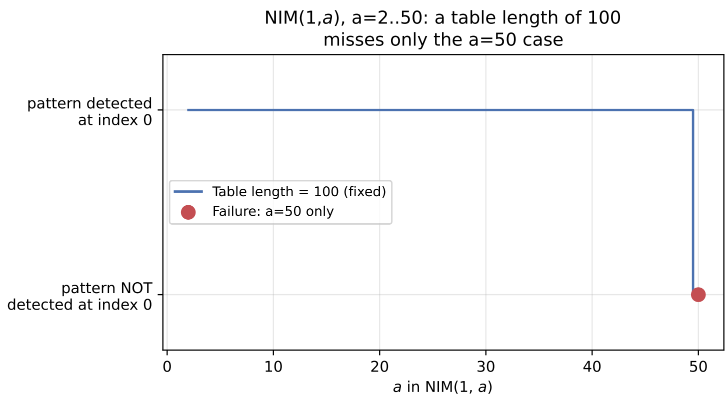

Figure 1: With a fixed win-table length of 100, every value of a from 2 to 50 in NIM(1,a) produces a pattern detected at index 0, except for the single case $a = 50$, where the table is not long enough to confirm two full periods.

3.2 Multi-Parameter Variants

Table 1 reports, for NIM$(1, a, b)$ with $1 < a < b \leq N$ and a fixed win-table length of 500, the percentage of all valid (a, b) combinations whose detected pattern begins at index 0, as a function of N.

Table 1: NIM(1, a, b): percentage of combinations with a pattern starting at index 0, win table length 500.

Max N	% starting at index 0	combinations tested
3–8	100.000	1–21
9	96.429	28
10	94.444	36
15	90.110	91
20	88.889	171
25	86.232	276
30	85.222	406
40	83.536	741
50	82.398	1176

[1] This particular result is fully determined by the recursive win/loss rule and the table length chosen; any correct implementation of the win-table generator and pattern detector will reproduce it exactly, since no randomness is involved at this stage of the analysis.

The same analysis for $\mathrm{NIM}(1, a, b, c)$ (Table 2) and $\mathrm{NIM}(1, a, b, c, d)$ (Table 3, using a table length of 300 for computational tractability) shows the same qualitative decline, and the decline is steeper for each additional move parameter.

Table 2: NIM(1, a, b, c): percentage starting at index 0, win table length 500.

Max N	% starting at index 0	combinations tested
5–9	100.000	4–56
10	98.810	84
15	93.956	364
20	90.402	969
25	86.808	2024
30	84.319	3654
40	80.293	9139
50	77.676	18424

Table 3: NIM(1, a, b, c, d): percentage starting at index 0, win table length 300.

Max N	% starting at index 0	combinations tested
5–9	100.000	1–70
10	99.206	126
12	96.667	330
15	92.707	1001
20	87.719	3876
25	82.929	10626
30	79.045	23751
40	73.300	82251
50	69.280	211876

Two trends are clear from Tables 1–3 and Figure 2. First, for small N (below roughly 9 or 10 in every family tested), every combination produces a pattern starting at index 0; the irregular cases only begin to appear once the move set is allowed to range over a wider span of values. Second, at a fixed $N = 50$ (Figure 3), the percentage of combinations starting at index 0 drops from 82.4% for three moves to 77.7% for four moves to 69.3% for five moves, showing that each additional move parameter makes a clean, zero-offset pattern measurably less likely, consistent with the intuition that a larger move set creates more opportunities for the early positions of the win table to behave irregularly before settling into a repeating cycle.

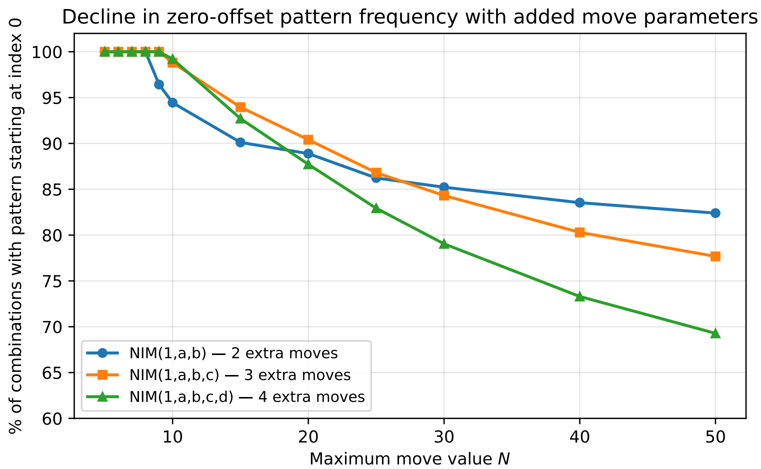

Figure 2: Percentage of move-set combinations with a pattern starting at index 0, as a function of the maximum move value N, for the three-, four-, and five-move families NIM$(1, a, \dots)$. All three curves decline as N grows, and the decline is steeper for families with more move parameters.

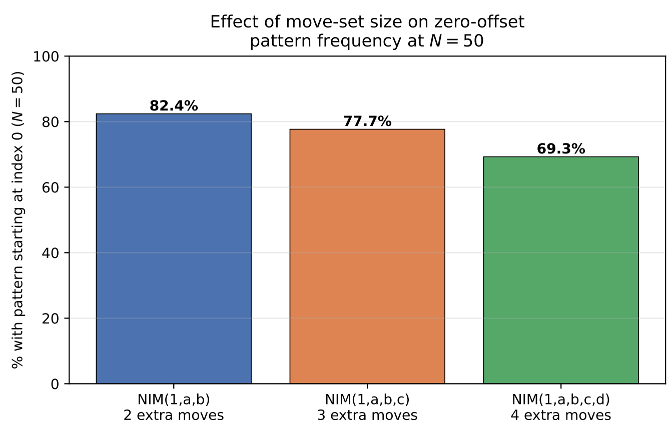

Figure 3: Percentage of combinations with a pattern starting at index 0, at a fixed $N = 50$, across the three move-set families tested. Each additional move parameter reduces the fraction of zero-offset patterns by roughly 4–5 percentage points at this value of N.

It is worth being explicit about a methodological point: the exact percentages reported here depend on the win-table length chosen, exactly as the NIM(1,a) anomaly in Section 3.1 illustrates. They should therefore be read as a measurement taken under a fixed, documented methodology (table length 500, or 300 for the five-parameter case) rather than as a property of the move sets alone independent of how they are tested.

4 Modular Rules for Structured Move-Set Families

In addition to the percentage analysis above, I derived, by hand, closed-form modular winning conditions for several structured families of move sets, and then verified every rule below by running it through the win-table generator and pattern detector described in Section 2. This verification step identified that several of my original hand-derived moduli, while giving a correct (if redundant) description of the losing positions, were not the *minimal* period of the sequence; the minimal period is reported here.

4.1 The family $\{1, x, x + 1\}$

For move sets of the form $\{1, x, x + 1\}$, checking the parity of the middle value x determines the minimal period p of the win/loss sequence:

$$p = \begin{cases} 2x & x \text{ even} \\ 2x + 1 & x \text{ odd} \end{cases}$$

and within one period, the losing positions are exactly the even residues less than x (taken modulo p). Table 4 lists this rule for $x = 1$ through 8, confirmed by computer search of the corresponding win tables.

Table 4: Modular rule for $\{1, x, x + 1\}$.

Move set	Minimal period	Losing residues
$1, 1, 2$	n mod 3	0
$1, 2, 3$	n mod 4	0
$1, 3, 4$	n mod 7	$0, 2$
$1, 4, 5$	n mod 8	$0, 2$
$1, 5, 6$	n mod 11	$0, 2, 4$
$1, 6, 7$	n mod 12	$0, 2, 4$
$1, 7, 8$	n mod 15	$0, 2, 4, 6$

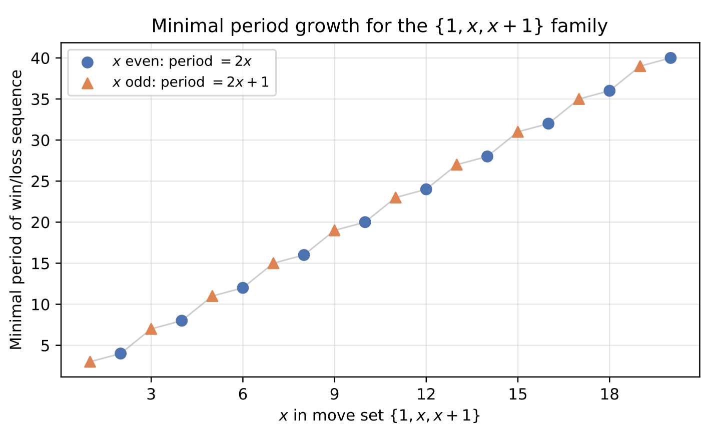

Figure 4: Minimal period of the win/loss sequence for the family $\{1, x, x + 1\}$, as a function of x. Even and odd values of x follow two distinct linear rules, $2x$ and $2x + 1$ respectively, which is why the points split cleanly into two interleaved sequences rather than forming a single line.

4.2 The family $\{1, x, x + 2\}$

For move sets of the form $\{1, x, x+2\}$, the behavior bifurcates sharply on the parity of x. When x is odd, the entire sequence reduces to ordinary parity: n mod $2 = 0$ is the losing condition, regardless of the specific value of x (verified for $x = 1, 3, 5, 7, 9, 11, 13$). When x is even, the minimal period is $2x + 2$, and the losing residues are every even number less than x together with every odd number strictly between x and $2x$.

This is also the family where my original hand-derived notes reported non-minimal moduli. For example, my notes for $\{1, 6, 8\}$ listed the losing condition as $n \bmod 14 \in \{0, 2, 4, 7, 9, 11\}$; computer verification shows the win/loss sequence is in fact periodic with the smaller period 7, with losing residues $\{0, 2, 4\} \pmod 7$ — the modulus-14 description is simply this same period written out twice. Table 5 gives the corrected, minimal-period version of this family.

Table 5: Corrected minimal-period modular rule for $\{1, x, x + 2\}$, even x.

Move set	Minimal period	Losing residues
$1, 2, 4$	$n \bmod 3$	0
$1, 4, 6$	$n \bmod 5$	$0, 2$
$1, 6, 8$	$n \bmod 7$	$0, 2, 4$
$1, 8, 10$	$n \bmod 9$	$0, 2, 4, 6$
$1, 10, 12$	$n \bmod 11$	$0, 2, 4, 6, 8$
$1, 12, 14$	$n \bmod 13$	$0, 2, 4, 6, 8, 10$
$1, 14, 16$	$n \bmod 15$	$0, 2, 4, 6, 8, 10, 12$

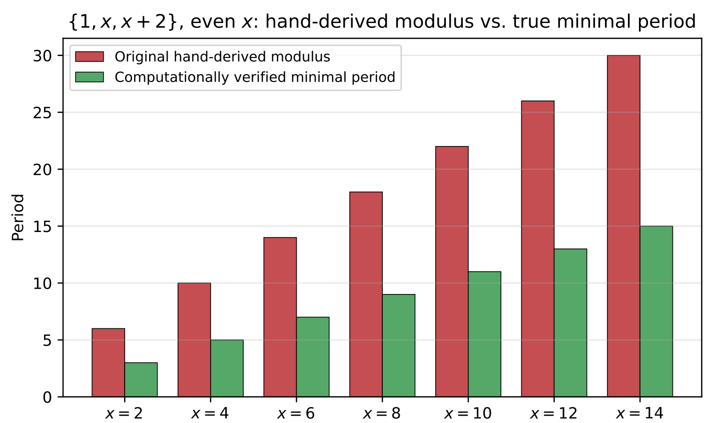

Figure 5: Comparison of the period originally reported in my hand-derived notes against the true minimal period found by computer search, for the even-x cases of $\{1, x, x + 2\}$. The hand-derived modulus is, in every case, exactly twice the true minimal period – a correct but redundant description of the same repeating pattern.

4.3 Four- and five-move consecutive families

For move sets consisting of 1 together with a consecutive run $x, x + 1, x + 2$ (a four-move set) or $x, x+1, x+2, x+3$ (a five-move set), the same parity-based rule structure holds, and in both cases I confirmed by computer search that the periods my hand analysis identified were already minimal. Table 6 summarizes both families.

Table 6: Modular rules for the four-move family $\{1, x, x+1, x+2\}$ and the five-move family $\{1, x, x+1, x+2, x+3\}$. All periods confirmed minimal.

Move set	Period	Losing residues
$1, 2, 3, 4$	n mod 5	0
$1, 3, 4, 5$	n mod 8	$0, 2$
$1, 4, 5, 6$	n mod 9	$0, 2$
$1, 5, 6, 7$	n mod 12	$0, 2, 4$
$1, 6, 7, 8$	n mod 13	$0, 2, 4$
$1, 7, 8, 9$	n mod 16	$0, 2, 4, 6$
$1, 8, 9, 10$	n mod 17	$0, 2, 4, 6$
$1, 2, 3, 4, 5$	n mod 6	0
$1, 3, 4, 5, 6$	n mod 9	$0, 2$
$1, 4, 5, 6, 7$	n mod 10	$0, 2$
$1, 5, 6, 7, 8$	n mod 13	$0, 2, 4$

In every one of these consecutive-run families, the minimal period grows roughly linearly in x, and the losing residues are always exactly the even numbers less than x. This is consistent with the intuition that including the move 1 in the move set, together with a dense run of consecutive larger moves, makes it relatively easy for the player to move into any nearby losing position, which in turn keeps the modulus small and the losing-residue set simple.

4.4 Practical effect of the move-selection rule under imperfect play

While building the simulation component of this project (a probabilistic player who plays the locally optimal move with some fixed probability p and a uniformly random legal move otherwise), I also identified a methodological choice that affects measured win rates and is worth recording explicitly: when a player is required to deviate from the optimal move (either because the random draw says so, or because no optimal move is available), there are at least two reasonable rules for choosing the fallback move — drawing uniformly from *all* legal moves (including, by chance, the optimal one), or drawing uniformly only from the remaining sub-optimal moves. These two rules give a player a measurably different effective accuracy, since the first rule gives the “imperfect” player a nonzero chance of accidentally playing optimally anyway. Any simulation study of imperfect Nim play (including the percentage analysis in Section 3) should state explicitly which fallback rule was used, since the two are not equivalent.

5 Conclusion

This paper presented a generalized, multi-move win-table generator and pattern-detection tool, and used it for two purposes: first, to measure, under a fixed and explicitly documented methodology, how the fraction of move-sets producing a winning pattern that starts at index 0 declines as additional move parameters are added to a Nim variant of the form $\mathrm{NIM}(1, a, \dots)$; and second, to verify and, in several cases, correct a set of modular winning-condition rules I had derived by hand for structured families of three-, four-, and five-move variants. The headline empirical finding is that single-parameter variants $\mathrm{NIM}(1, a)$ are essentially always periodic from the first position, but each additional move parameter measurably reduces the fraction of variants with a zero-offset pattern — from 100% for very small move sets down to under 70% for five-move sets at $N = 50$ in this study's data. The modular-rule analysis showed that the families $\{1, x, x+1\}$, $\{1, x, x+1, x+2\}$, and $\{1, x, x+1, x+2, x+3\}$ all follow a simple parity-based period rule, while the family $\{1, x, x+2\}$ bifurcates sharply between odd x (reducing to ordinary parity) and even x (a longer period with a more complex residue set); the even-x case also illustrated that hand-derived periodicity claims should be checked computationally for minimality, since a description in terms of a multiple of the true period can look correct while obscuring the underlying structure.

Future work could extend the percentage analysis to move sets that do not require 1 as the smallest move, to multi-pile variants via the Sprague–Grundy theorem [2, 3], and to a systematic search (for example using regression or a decision-tree model over move-set features such as range, spacing, and GCD) for a closed-form predictor of the minimal period directly from the move set, rather than relying on case-by-case derivation.

Acknowledgements

Thank you to Professor William Gasarch at the University of Maryland's Department of Computer Science for guidance on this research and paper.

References

[1] C. L. Bouton, "Nim, A Game with a Complete Mathematical Theory," *Annals of Mathematics*, vol. 3, no. 1, pp. 35–39, 1901–1902.

[2] R. Sprague, "Über Mathematische Kampfspiele," *Tohoku Mathematical Journal*, vol. 41, pp. 438–444, 1935.

[3] P. M. Grundy, "Mathematics and Games," *Eureka*, vol. 2, pp. 6–8, 1939.

[4] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning Ways for Your Mathematical Plays, Vol. 1: Games in General*. London: Academic Press, 1982.

[5] A. N. Siegel, *Combinatorial Game Theory*. Providence, RI: American Mathematical Society, 2013.