JOURNAL OF EMERGING INVESTIGATORS

Comparative analysis of ability and ideal starting positions in 1 and 2 pile Nim games

Rohan Sinha[1], Rhun Henderson[2], William Gasarch[3]

[1] Richard Montgomery High School, Rockville, Maryland

[2] Montgomery Blair High School, Silver Spring, Maryland

[3] Department of Computer Science, University of Maryland, College Park, Maryland

Student Authors

Rohan Sinha – High School

Rhun Henderson – High School

SUMMARY

This research paper analyzes imperfect play in the game of Nim. More specifically, it questions whether player ability or the influence of winning starting positions is more important in game outcomes. We also research the required ability differential needed for a player to win from a losing position. We hypothesize that when the pile size is large, the initial position has minimal impact on win probability and player ability becomes the primary determinant under imperfect play. This paper presents several computational game simulations and a derived mathematical model for win percentages, both of which verify our hypothesis. Our results show that a winning starting position only benefits a player until a certain point during imperfect play, then has negligible effect. This research holds significance because, according to the Sprague-Grundy Theorem, any impartial game is equivalent to a single pile game of Nim. Additionally, it also has potential applications in the field of reinforcement learning, as it provides data on imperfect play, which is necessary to create robust models, as seen priorly.

INTRODUCTION

Nim consists of two players taking turns removing sticks from a pile containing n number of sticks. Players can remove a certain number of sticks from the pile per turn, and the player who removes the last stick wins.

There are many variations of Nim (1). However, we research a specific one in this paper. Let A be a finite set of natural numbers representing the allowed moves in a turn. For this paper, we use a common variant of the game: $|A| = 3$ and $1 \in A$. And where the *starting position* is the number of sticks in each pile at the start of the game.

To illustrate the nature of the game, consider $A = \{1, 2, 3\}$ *as an example*. It is apparent that the player with 4 sticks left at the time of their turn is always going to lose. If they remove 1, their opponent can remove 3 and win; if they remove 2, their opponent can remove 2 and win; if they remove 3, their opponent can remove 1 and win. Hence leaving the opponent with 4 sticks left is ideal. This applies to any multiple of 4 as well. Each position (remaining number of sticks in the pile) is calculated to be winning for a certain player if they play perfectly. This paper, however focuses on when both players play imperfectly. Before we explain the solution to the game under perfect play (which is still relevant), it is important to consider the context of the surrounding research.

Nim was first invented in 1905, but there is still active research on it and its variants (2). These include models incorporating economic elements like taxation (3), systems utilizing bidding for turn rights (4, 5) and other abstract versions (6). Further studies have examined Candy Nim, where a secondary goal of resource collection is introduced; analyzed the computational complexity of winning positions; and proposed novel variants such as Veto-Nim and Large Nim (7, 8, 9). A common foundational assumption uniting this extensive body of work is that all players execute perfect play. This paper addresses a significant gap in the literature by investigating scenarios where players do not play perfectly. Apart from simply answering the question of what was more relevant to game outcomes: being a better player, or having a better position, this research has an application to the field of reinforcement learning. This approach is advocated by Gleave et al. and Pinto et al., among others, who argue that robust adversarial learning requires training an AI against a diverse set of opponents, including many imperfect ones (10,11). The rationale is that this exposes the AI to a much broader set of positions; conversely, if an AI is trained solely against perfect or expert players, many positions encountered in actual play may be unfamiliar, leaving the AI unable to respond effectively. This principle is exemplified by AlphaGo and AlphaZero, which utilized self-play that, especially in its early stages, involved clearly non-expert gameplay. Additionally, this research holds significance because, according to the Sprague-Grundy Theorem, any impartial game is equivalent to a single pile game of Nim (12).

Regarding perfect play, to calculate the winner at any given position for any value of A, Grundy numbers are utilized (13). The Grundy numbers for the game of Nim are defined as

$$G(n) = \begin{cases} 0 & , \ n = 0 \\ mex\,\{G(n-a) : a \ \in A, n - a \geq 0\}, & n \geq 1 \end{cases} \qquad \text{(Equation 1)}$$

where *G(n)* is recursively defined for each position *n* and *mex* returns the smallest natural number that is not in the set (14). It indicates which player the position is winning for through a recursive piecewise function that ultimately returns either 0, 1, 2 or 3. If *G(n) = 0*, *n* is a winning position for the player who just made their move. Additionally, if *G(n) = 0* at the start of the game, the position is winning for player 2. The opposites apply respectively if $G(n) \neq 0$. Each starting position and value for *A* affects whether the position is winning for player 1 or 2.

Let $A = \{1, 2, 3\}$. $G(1) = 1$ since the set of Grundy numbers of reachable positions is $\{G(0)\} = \{0\}$, the least natural number that is not in it is 1. Since $G(1) \neq 0$, this starting position is winning for player 1. Then, $G(2) = 2$ since the set of Grundy numbers of reachable positions is $\{G(0), G(1)\} = \{0, 1\}$, and the least natural number not in it is 2. Since $G(2) \neq 0$, this starting position is winning for player 1. Similarly, $G(4) = 0$ since the set of Grundy numbers of reachable positions is $\{G(1), G(2), G(3)\} = \{1, 2, 3\}$, and the least natural number not in it is *0*. Since $G(4) = 0$, this starting position is winning for player *2*.

Grundy numbers can be expanded to multi pile Nim (15). Let $A_1, . . ., A_k$ be k finite sets of naturals representing the valid moves in pile i where $1 \leq i \leq k$. Let $G_i(n_i)$ be the Grundy function for Nim-A_i. Then the Grundy function for the Nim-$(A_1, . . ., A_k)$ is

$$G(n_1, \dots, n_k) = \bigoplus_{i=1}^{k} G_i(n_i) \qquad \text{(Equation 2)}$$

Where $\oplus$ is the bitwise XOR function. $G_i(n_i)$ is represented in base 2 (15).
The Sprague-Grundy Theorem explains the results of this:
player 1 wins the game starting from position $(n_1, n_2, . . . , n_k)$ if and only if
$G(n_1, n_2, . . . , n_k) \neq 0$ *(12).*

To illustrate the Sprague-Grundy Theorem, consider a game with two piles. Let $A_1 = \{1, 2, 3\}$ and $n_1 = 17$. Let $A_2 = \{1, 3, 4\}$ and $n_2 = 5$. The Grundy numbers for each pile are:

$$G_1(n_1) = mex\,\{G(17 - a) : a \in A_1,\, 17 - a \geq 0\} \qquad \text{[Eqn 1]}$$

$$G_2(n_2) = mex\,\{G(5 - a) : a \in A_2,\, 5 - a \geq 0\} \qquad \text{[Eqn 1]}$$

Hence, $G_1(n_1) = 1$ and $G_2(n_2) = 3$, then

$$G(n_1, n_2) = G_1(n_1) \oplus G_2(n_2) = 1 \oplus 3 = 2 \neq 0 \qquad \text{[Eqn 2]}$$

Hence Player *1* wins in this configuration. We use a simple algorithm that plays the game perfectly given a winning position for both single and multi-pile Nim.

When both players play optimally, the player in the initial winning position at the start will always win (e.g., with $A = \{1, 2, 3\}$, player 1 will always win if n is not a multiple of 4) (Table 1 & 2). Our research focuses on the game's outcomes when players make imperfect moves.

Our methodology consists of computationally simulating a large number of games with varying levels of imperfect play and starting positions to draw empirical conclusions from its results. Specifically, we examine scenarios where player 1 plays the optimal move with probability of P_1, and player 2 with probability of P_2, analyzing the percent of games each player wins at various starting positions and value of P_1 and P_2. The optimal moves are calculated using Grundy Numbers (Eqn 1 & 2). Additionally, we research how much better a player must be for them to win from a losing position. We find that when n is large the initial position has minimal impact on win probability and player ability becomes the primary determinant under imperfect play.

RESULTS

We simulated 10^6 games given the ability of players (the probability of making the optimal move), P_1 and P_2, and the set of allowed moves, A. This program then computed the percentage of games each player won (W_1 for player 1 and W_2 for player 2) at each starting position n, where $n \in \{1, . . . , 300\}$. This is also known as a Monte Carlo Simulation, and is highly effective for these tests since they involve many simulations with varying game values (16). Once analyzed, it became clear that the win percentages trended toward stable values, eventually exhibiting negligible change (Figures 1 & 2).

Single Pile

Using moving averages, we computed the stabilization points n_s at the value n where the win percentages stopped changing significantly. The data confirmed that in games where player 1 played at a higher ability than some player 2, W_1 always stabilized to a greater value than W_2, and vice versa regardless of whether n was a winning position for them or not (given that $n > n_s$) (Figure 1). We formalize these observations as the conjecture below.

Conjecture 1.1

If $P_1, P_2 \in$ (0%, 100%) and $P_1 > P_2$, then as $n \to \infty$, $W_1 > W_2$.

Conversely, if $P_2 > P_1$, then as $n \to \infty$, $W_2 > W_1$.

Empirical evidence from simulations of all considered different game configurations strongly supports this conjecture as the player with the higher ability wins a greater number of games as n increases than the other player, independent of winning and losing positions.

These stabilization values remained consistent within each game type. For example, in single pile Nim, regardless of the value of A, if $P_1 = 90\%$ and $P_2 = 80\%$, and if $n \geq n_s$, $W_1 \approx 70\%$ and $W_2 \approx 30\%$ (Figures 1.A & 1.D).

If Player 1 and Player 2 played at equivalent ability, W_1 and W_2 both stabilized at 50%, meaning that, as $n \to \infty$, both players would win approximately an equal number of games regardless of whichever values of n were ideal for each player (Figures 1.C & 1.B). We formalize these observations as the conjecture below.

Conjecture 1.2

$$\text{If } P_1, P_2 \in (0\%, 100\%) \text{ and } P_1 = P_2, \text{ then as } n \to \infty,\ W_2 = W_1.$$

Even though the values of A do not affect the fact that stabilization occurs, they do affect the value of n_s. For example, if $P_1 = 90\%$ and $P_2 = 80\%$, in (1, 2, 3)-Nim, $n_s = 61$, however in (1, 4, 5)-Nim, $n_s = 120$ (Figures 1.A & 1.D).

Incidentally, we discovered that there was a strong direct relationship between the ability of both players (P_1 & P_2) and n_s (Figure 2). These best fits had R^2 values of 0.887 (Figure 2.A) and 0.911 (Figure 2.B), respectively. Though their shape was slightly different depending on *A*, they consistently maintained a general hyperbolic paraboloid shape. We formalize these observations as the conjecture below.

Conjecture 1.3

The value of n_s if minimized, but not necessarily to 0 when $P_1 + P_2 \approx 100\%$.
Furthermore, combinations where both P_1 and P_2 are either high or both low result in higher values of n_s, i.e., n_s is maximized when $P_1, P_2 \to 0\%$ or $P_1, P_2 \to 100\%$.

Winning from a losing position

We researched the ability difference required for Player 1 to have a higher win rate ($W_1 > W_2$) in losing positions for Player 1. To achieve this, we developed a program that simulated 10^6 games for each combination of P_1 and P_2 in 1% increments, and determined the minimum difference $min(P_1 - P_2)$ for which $W_1 > W_2$ for every iteration of P_1. The results revealed that there was a decrease in minimum difference as n increased; furthermore, as n increased, the lower bound decreased (Figure 3): in summary, as *n* increased, the ideal position's influence over the game diminished.

The relationship between Player 1 ability and Minimum Difference was strong for (1, 2, 3)-Nim with an average coefficient of determination being 0.986 (Figure 3.A). And stronger for (1, 3, 4)-Nim: 0.994 (Figure 3.B).

There were visible differences between games, however. The minimum difference at the lowest value of P_1 in (1, 2, 3)-Nim were roughly equivalent to the minimum difference at the greatest value of P_1 (Figure 3.A). In (1, 3, 4)-Nim, the lowest value of P_1's minimum difference were significantly lower than that of the greatest value of P_1 (Figure 3.B). Also, while the curves' vertices in (1, 2, 3)-Nim were around the center of their domain, the curves' vertices in (1, 3, 4)-Nim were left of the center (Figure 3.B & 3.A).

2 Pile Nim

We simulated 10^5 games given the ability of players, P_1 and P_2, and the sets of allowed moves, A_1 and A_2. This program then computed the percentage of games each player won at each starting position n_1, n_2, where $n_1, n_2 \in \{1, . . ., 100\}$. As with 1 pile Nim, 2 pile Nim stabilized as n_1, n_2 increased (Figure 4). The stabilization points identified in the graphs represent the coordinates where W_1 and W_2 stop changing significantly. This implies Conjecture 1.1 can be extended to more than 1 pile.

DISCUSSION

We researched the comparative importance of player ability versus ideal starting positions in single and multi pile nim games. Our computations show that as the number of sticks increases, the significance of having an ideal starting position diminishes (given that both players play imperfectly). Notably, beyond a certain number of starting sticks, W_1 and W_2 stabilize to specific values (Figure 1, 2, 4). In particular, when $P_1 = P_2$, both W_1 and W_2 stabilize to 50%. Moreover, our analysis revealed that the player with a higher ability will always stabilize to a higher win rate, despite that player having to start in losing positions. (Figure 1). This was anticipated because as n increases, so does the probability of player mistakes. Hence, before the stabilization point, ideal positions have a greater influence on win percentages due to fewer opportunities for mistakes. This highlights the importance of player ability over positional advantage in certain scenarios, especially after n_s. Additionally, our results show that the greater the values of A, the greater the value of n_s. This is because fewer number of moves will be required at any starting position as a greater number of sticks will be removed each move which means that n_s increases to accumulate enough optimal and sub-optimal moves. Both are needed to negate the advantage of ideal starting positions.

Examining stabilization points further, we found that n_s are minimized when $P_1 + P_2 \approx 100\%$ and maximized when both abilities are either high or low (Figure 2). This is because if both players play with greater ability, a greater number of moves are required for imperfect moves to accumulate. Conversely, if both players play less accurately, a greater number of moves are required for optimal moves to accumulate.

Our data shows that the minimum ability differential required for a player in a losing position to win more than 50% games is highly dependent on their ability and is not constant (Figure 3). Furthermore, our data revealed that with each incremental increase in sticks towards non-ideal positions for Player 1, the minimum difference decreased. This is because since there are a greater number of moves to be made; it is more likely that the winning player will make a mistake, hence a lower difference in ability is required. The minimum difference happened to decrease similarly to a reverse Fibonacci sequence as n increased (Figure 3).

While our experiment effectively demonstrated various properties inherent to Nim, we encountered several limitations that warrant further consideration. Due to computational constraints, we were only able to simulate 10^5 (two pile) and 10^6 (one pile) games for each game position. A greater number of simulations would give more accurate data, with little to no significant anomalies. In addition, we were not able to compute data related to stabilization points for more than two piles, since it resulted in 10^{2p+5} game simulations for p number of piles. Access to more powerful computers would have allowed us to compute data for more piles. Our playing algorithm also used a specific strategy (see Materials and Methods) that affected the win percentages for both players.

This research provides a framework for understanding how player ability and starting positions influence win probabilities in impartial games. According to the Sprague-Grundy theorem, all impartial games are equivalent to single pile Nim (12). Hence, these results may extend to games like Tic Tac Toe, Sprouts, Kayles, Quarto, and Chomp (15).

Impartial game logic is also relevant in reinforcement learning, where agents must learn strategies in structured, turn-based environments (17). By modeling how imperfect decisions accumulate over repeated interactions, this work may offer a basis for exploring learning behavior in systems where agents are not guaranteed to act optimally (17).

Although this paper focuses on empirical evidence, we found an equation to estimate the stabilized win percentage (win percentage after n_s) for both players as n approaches ∞:

$$\frac{a}{a+b} = \frac{P_1(100-P_2)}{P_1(100-P_2)+P_2(100-P_1)} \cdot 100 \qquad \text{(Equation 3)}$$

It uses the statistical formula that calculates the probability of an event *a* happening before event *b* to do so. Since this equation does not contain any variables related to winning positions, it does directly support our hypothesis, granted it is most accurate for smaller values of *A*. For example, the stabilized win percentage for 1,2,3-Nim (P_1 =90% and P_2 = 80%) was computed to be approximately 69.29% and the equation's estimate was 69.23% (Figure 1 & Eqn. 3). Notably, it also does not involve any variables related to Nim, which may indicate its application in other impartial games such as those mentioned above. Possible future experiments could explore the similarities between imperfect play in Nim and said other games.

This paper computationally and mathematically shows that the effect of ideal starting positions fade, and player ability becomes the primary driver of performance. While our research is grounded in Nim, the findings may offer broader insights into imperfect play in other settings such as training AI models and other impartial games.

MATERIALS AND METHODS

Stabilization point of single pile Nim

To find the stabilization point in a single pile game, the derivative (represented as the ‘) of a moving average *MA* with a window size of 10 is used. The stabilization point is defined using a threshold of 0.06:

$$n_s = \arg \min_{i \in [i, i+10)} (\max(MA') - \min(MA') < 0.06) + 5 \qquad \text{(Equation 4)}$$

Argmin returns the lowest number of sticks that satisfies the condition: the range of the derivatives < 0.06 in a 10 stick window where *i* is the number of sticks at the start of the window and *i+10* is the number of sticks at the end of the window. The range is calculated by subtracting the lowest derivative *(min(MA'))* from the greatest in the window *(max(MA'))*.

Stabilization point of 2 Pile Nim

To find the stabilization point in a 2 pile game, a Gaussian smoothing with Σ = 0.1 is applied and then gradient magnitudes ($\|\nabla\|$) are computed:

$$n_s = \arg \min_{(i,j) \in W_{i,j}} (\max \| \nabla Z \| - \min \| \nabla Z \| < 1) \quad \text{(Equation 5)}$$

Argmin returns the lowest combination of number of sticks that satisfy the condition: the range of the gradient magnitudes < 1 in a window where *W* is the region from the combination of starting number of sticks *(i, j)* to the point where the starting number of sticks in both piles are 100.

Game Simulation

A Java program calculates the optimal move by testing every single possible move in every single pile until the Grundy number computes to 0 (winning). If there are no such moves it chooses the smallest move possible It makes this move in the first pile by default. If there is more than one optimal move, it chooses the biggest move. It tallies up the total number of wins for each player and divides by the total number of games played. When playing sub optimally, Player 1 choses this optimal move with a probability of P_1 and Player 2 with a probability of P_2.

Software and Packages

The Java Programming Language (Version 1.8.0 51) was used for the Monte Carlo simulations and game calculations (Grundy numbers and Optimal move calculation) (16). Java's Util Package was utilized for hash maps, hash sets, sets, scanners, and random number methods (18). Java IO was used for file writing functionality (19). For stabilization point calculations and graphing, the Python Programming Language (Version 3.10) was used. The NumPy package (Version 1.26.4) was used for mathematical calculations (20). Matplotlib (Version 3.7.1) was used for graphing (21). The Scipy package (Version 1.13.1) was used for Gaussian filters and linear algebra functions (22).

ACKNOWLEDGMENTS

We thank Rena Katz for proofreading the paper, providing significant feedback on our work (including code), and helping with the references section.

REFERENCES

1. Berlekamp, Elwyn R., et al. *Winning Ways for Your Mathematical Plays. Vol. 1 : Games in General.* London, Academic Press, 1983, pp. 55, 310, 381, 393.
2. Bouton, Charles. "Nim, A game with a Complete Mathematical Theory." *Annals of Mathematics*, vol. 3, 1901, pp. 35-39, https://doi.org/10.2307/1967631.
3. Berlekamp, Elwyn R. "The economist's view of combinatorial games." *Games of No Chance 3*, edited by Michael H Albert and Richard J Nowakowski, Cambridge University Press, 1994, pp. 365-405.
4. Kant, Prem, and Larsson, Urban. "Survey of Richman bidding combinatorial games." *Games of No Chance 6,* edited *by* Urban Larrson, Cambridge University Press, May 2025, pp. 43-54.
5. Lazarus, Andrew J., et al. "Richman games." *Games of No Chance*, edited by Richard J. Nowakowski, Cambridge University Press, 1994, pp. 439 - 450.
6. Davies, Alfie M., et al. "Combinatorial game theory monoids and their absolute restrictions: a survey." *Games of No Chance 6*, edited by Urban Larsson, Cambridge University Press, May 2025, pp. 1-24.
7. Mani, Nitya, et al. "$\wp$ play in Candy Nim." *Games of No Chance 6*, edited by Urban Larsson, Cambridge University Press, May 2025, pp. 405-432.
8. Hearn, Robert A., and Demaine, Erik D. *Game, Puzzles, and Computation*. 1st ed., A K Peters/CRC Press, July 2009.
9. den Bergh, Mark V., et al. "Nim variants." *International Computer Games Association*, vol. 44, no. 1, 2022, pp. 2-17, https://doi.org/10.3233/ICG-220206.
10. Gleave, Adam, et al. "Adversarial policies: Attacking deep reinforcement learning." *ArXiv*, Jan 2021, https://doi.org/10.48550/arXiv.1905.10615.
11. Pinto, Lerrel, et al. “Robust Adversarial Reinforcement Learning.” *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 6 Aug. 2017, pp. 2817–26
12. Sprague, Roland. “Über Mathematische Kampfspiele.” *Tohoku Mathematical Journal*, vol. 41, 1 Jan. 1935, pp. 438–444.
13. Grundy, Patrick M. “Mathematics and Games.” *Eureka*, vol. 2, 1939, pp. 6–8.

14. Kogler, Jakob, and Malek Sware. “MEX (Minimal Excluded) of a Sequence.” *Algorithms for Competitive Programming*, 2024, cp-algorithms.com/sequences/mex.html. Accessed 5 Aug. 2024. Kroese, Dirk P, et al.
15. Kroese, Dirk P., et al. “Theory of Impartial Games.” *Massachusetts Institute of Technology*, 2009, web.mit.edu/sp.268/www/nim.pdf. Accessed 4 Aug. 2024.
16. “Why the Monte Carlo Method Is so Important Today.” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, no. 6, 20 June 2014, pp. 386–392, https://doi.org/10.1002/wics.1314.
17. Zhou, Bei, and Søren Riis. “Impartial Games: A Challenge for Reinforcement Learning.” *ArXiv* , 25 May 2022, https://doi.org/10.48550/arxiv.2205.12787.
18. “Package Java.util.” *Oracle*, 6 Jan. 2020, docs.oracle.com/javase/8/docs/api/java/util/package-summary.html. Accessed 5 Aug. 2024.
19. “Package Java.io.” *Oracle*, docs.oracle.com/javase/8/docs/api/java/io/package-summary.html. Accessed 5 Aug. 2024.
20. “NumPy User Guide.” *NumPy*, numpy.org/doc/stable/user/index.html. Accessed 5 Aug. 2024.
21. “Users Guide.” *Mathplotlib*, matplotlib.org/stable/users/index.html. Accessed 5 Aug. 2024.
22. “SciPy User Guide.” *SciPy*. docs.scipy.org/doc/scipy/tutorial/index.html. Accessed 5 Aug. 2024.

Figures and Figure Captions

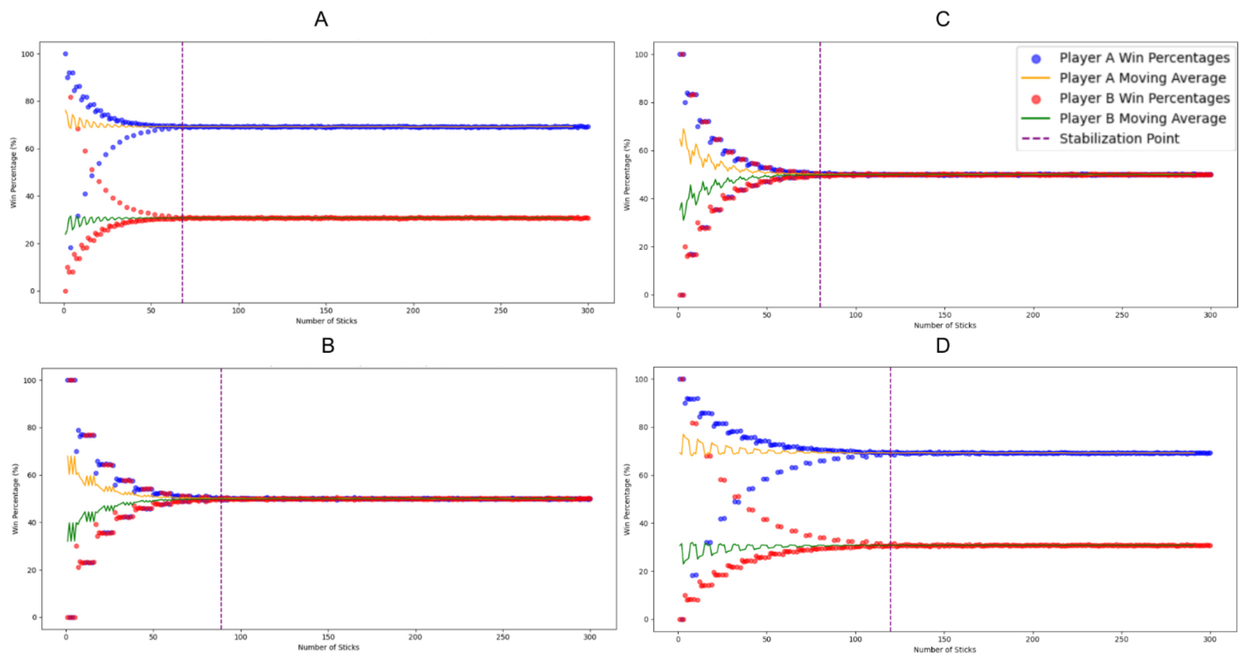

Figure 1. Win Percentages of Player 1 (W_1) and Player 2 (W_2) vs. Starting Number of Sticks (n). Scatter plot , moving averages, and stabilization points [Eqn 4] for win percentages in different game configurations: A) A = {1, 2, 3}, P_1 = 90%, and P_2 = 80%; B) A = {1, 5, 6}, P_1 = 70%, and P_2 = 70%; C) A = {1, 3, 4}, P_1 = 85%, and P_2 = 85%; D) A = {1, 4, 5}, P_1 = 90%, and P_2 = 80%. 10^6 simulations. Win percentages were calculated using a simulation algorithm that recorded the proportion of wins for each player.

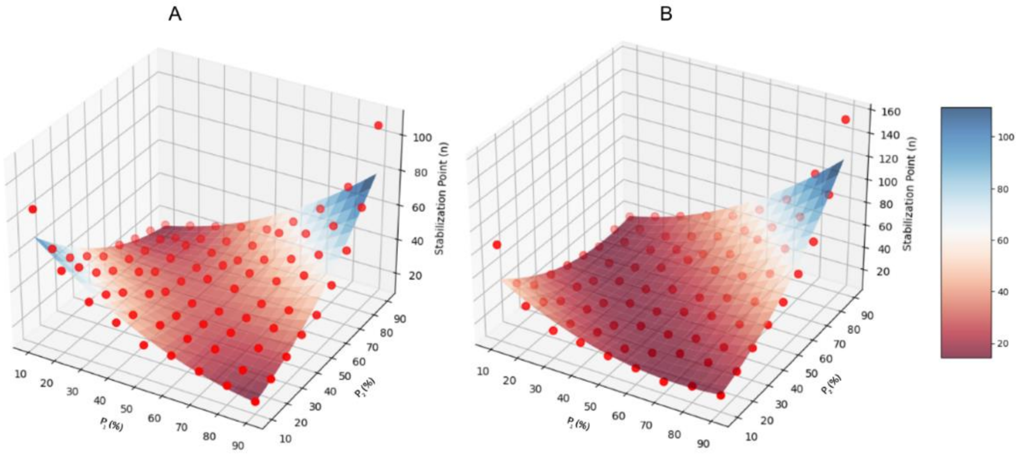

Figure 2. Player 1 ability (P_1) and Player 2 ability (P_2) vs. Number of Sticks Needed for Stabilization (n_s). Scatterplots and 2nd order 3-dimensional best-fit surfaces for n_s for 2 game configurations: A) A = {1, 2, 3}; B) A = {1, 3, 4}. Stabilization points were calculated using Equation 4 on win percentages from the game simulator.

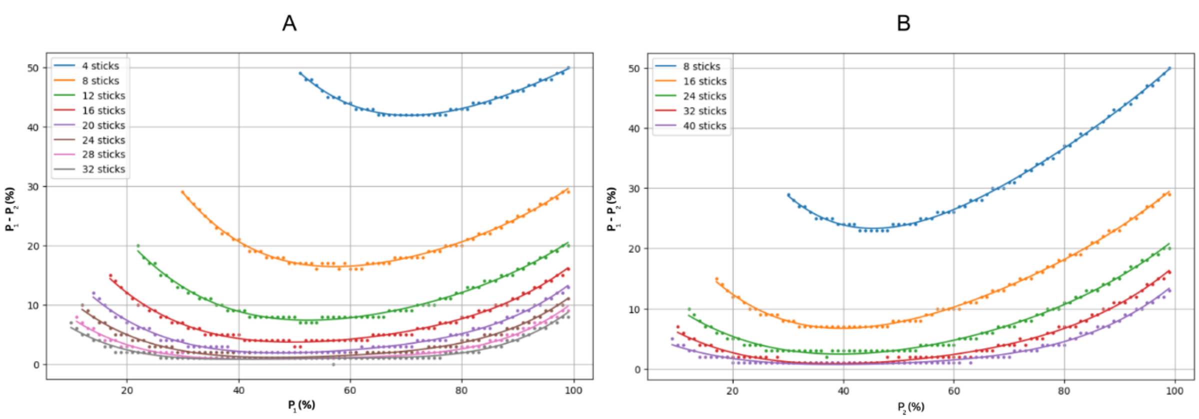

Figure 3. Minimum ability difference ($P_1 - P_2$) where $W_1 > W_2$ vs. Player 1 ability (P_1) in ideal positions for Player 2. Scatter plots and parabolic regressions for 2 game configurations: A) A = {1, 2, 3}; B) A = {1, 3, 4}. All values of n are losing positions for Player 1. Minimum difference was found by simulating games per P_1–P_2 pair (1% increments) and identifying the smallest (P1 − P2) where Player 1 outperformed Player 2.

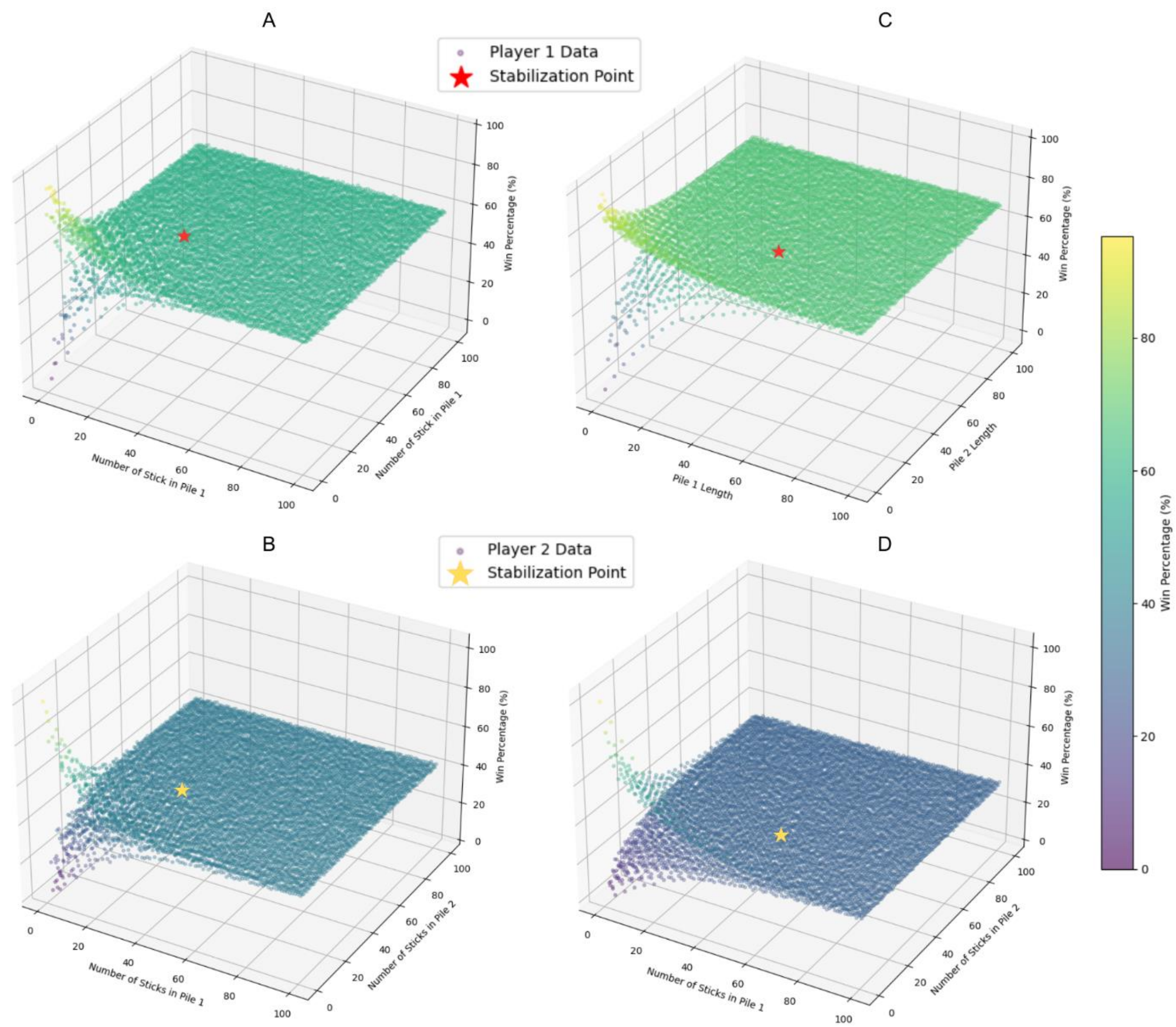

Figure 4. Number of Sticks in Pile 1 (n_1) and Number of Sticks in Pile 2 (n_2) vs. Player 1 and Player 2 Win Percentages (W_1, W_2). Scatterplots and stabilization points [Eqn 5] for 2 game configurations: A) and B) $A_1 = \{1, 3, 4\}$, $A_2 = \{1, 4, 5\}$, $P_1 = 80\%$, $P_2 = 70\%$, though A) represents W_1 while B) represents W_2. Graphs C) and D) $A_1, A_2 = \{1, 2, 3\}$, $P_1 = 90\%$, $P_2 = 80\%$, though C) represents W_1 while D) represents W_2. 10^5 simulations. Win percentages were calculated using a simulation algorithm that recorded the proportion of wins for each player.

Tables with Captions

Number of Sticks	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Winner	2	1	1	1	2	1	1	1	2	1	1	1	2	1	1	1	2

Table 1. **Win table of 0 − 16 sticks**. Computed using game logic mentioned in the introduction where A = {1, 2, 3}.

Number of Sticks	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Grundy Number	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0

Table 2: **Grundy numbers of 0 − 16 sticks.** Computed using equation 1 where A = {1, 2, 3}.

Appendix

GitHub link: github.com/newrohansinha/NIM. This repository contains all the code used in this paper and the raw data.