Structures From Chaos: In a NIM Game

Ian Cowen, Zen Nguyen, Peter Tingley
Here is our project in 1 slide

We modified NIM

And we beat it.

Well...
Basic Rules

❖ There are two players
❖ They are given one or three piles of any number of stones
❖ Each player alternates taking stones

Differences

<table>
<thead>
<tr>
<th>Original NIM</th>
<th>Our NIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>❖ There are 3 piles of stones</td>
<td>❖ There only is 1 pile of stones</td>
</tr>
<tr>
<td>❖ Take from any pile</td>
<td>❖ But, players can only pick the same amount, ± 1 stone of the last previous pick</td>
</tr>
<tr>
<td>❖ Any number of stones</td>
<td>❖ The game ends when a player cannot move and loses</td>
</tr>
<tr>
<td>❖ Last person to take a stone wins</td>
<td></td>
</tr>
</tbody>
</table>
Sample Game

- Our last pick was 3
- There is now one pile with 5 stones

- This means you now have 5 stones on your table
- So you can either take:
  • 3 stones
  • 2 stones = 3 - 1
  • or 4 stones = 3 + 1

Say You Picked 3:
- Current stones left: 2
- Now, we can take:
  ● 2, 3 or 4 stones

  We Pick 2 Again:
  - Current stones left: 0
  Remember, we just picked 2!

GAME ENDS
We win because you can’t move any more!
Visualisation of The Game

• We want to have a 2-D graphical representation of the game. So, we label the x-axis as the number of stones picked by the previous player (aka *previous pick*), and the y-axis as the number of stones left.

• So now, if I say you are now at (3, 5), it means that the other player just picked 3 stones, and there are 5 stones left.

• Let’s put our last game into an actual graph and see.
Losing and Winning States

- For each possible state \((x, y)\) of the game, one can ask: if a player starts their turn with the game in that position, can they guarantee they will win?
  - If yes, we call \((x, y)\) a **winning position**.
  - If not, we call it a **losing position**.

- Here is a recursive algorithm for determining which position wins and loses:
  - If there is no legal move, the position is losing
    - the player whose turn it is has already lost
  - If there is a legal move that puts the game in a losing position, then the current position is winning
  - If every possible move puts the game in a winning position, the current position is losing
Our Computational Results

- We wrote a Java program that implements the algorithm and produces a table labelling all positions of the game up to (250, 10000).

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>5</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>14</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

* X: losing position
*“-”: winning position

- So if you have this table with you at a casino, you can follow the “-”, and easy!
  Money $$ in your pocket!
Here is some zoom-out pictures of the table with more states...
Wedges

- Go back to the last slide, and look at the left picture.

- Do you notice these nice alternating black and white wedges to the bottom right?
  - The black wedges are losing wedges, and the white ones are winning wedges. They are the most consistent and obvious structure of the game!

  ... Aren't they beautiful?

- If the state of the game is in one of these regions, one of the players has a fairly easy winning strategy. We will now precisely describe these wedges.

(We also prove these wedges in our paper to show that they shall exist, forever!)
Wedges - Theorem

(1) Positions satisfying $2ix + i \leq y \leq (2i + 1)x - i - 2$ are losing for player A, where $i = 0, 1, 2, 3, \ldots$.

(2) Player B’s winning strategy is to take $x$ stones every play for the remainder of the game, regardless of what player A does.

Go look! ------------->
Wedges – Theorem

(3) Positions satisfying \((2i + 1)x + i \leq y \leq (2i + 2)x - i - 2\) are winning for player A. Player A’s winning strategy is to take \(x\) stones every play for the remainder of the game, regardless of what B does.

(4) Positions satisfying \((2i + 1)x - i - 1 \leq y \leq 2(i + 1)x + i\) are winning provided that \(x \geq 4i + 1\).

** \(i = 0, 1, 2, 3, \ldots\)
Repeating Patterns

- Do you notice these pretty black and white curves in the middle? (when x ranges from around 100 to 130)

- If so, can you believe that we can describe these curves by quadratic equations?

And, we can also (sort of) prove!
Formula for These Curves

- The upper curves (green):
  \[ f(x) = -\frac{2}{3}(x^2) + \frac{2}{3}i \cdot x - \frac{4 + i}{3}. \]

- The lower curves (red):
  \[ f(x) = -\frac{2}{3}(x^2) + \frac{2i - 1}{3} \cdot x + \frac{i}{3}. \]

** i = 0, 1, 2, 3, ...
How Do These Curves Happen?

Roughly Speaking…

- Remember the wedges above? We show that if $x > 4i$ ($i = 0, 1, 2, 3, \ldots$), all the positions in winning regions have at least 1 winning strategy. This means that the three winning strategies of taking $x$, $x-1$, and $x+1$ overlap.
- But what happens between $2i$ and $4i$ then?
  - The three possibles move do not overlap, and in between their gaps (dotted regions), there are actually losing regions forming.
Limit Process

- We are, unfortunately, not able to prove very much precisely about the curve patterns. However, we can make some precise statements in a limit situation.
- When we look far into the game, when both $x$ and $y$ are very large, and $y$ is much larger than $x$. When this happens, you can’t really tell the difference between $x$, $x - 1$, and $x + 1$, since they are so close together.
  - Now, imagine a game where you only always take exactly $x$. Then the recursion that generates winning and losing positions is easy: $(x, y)$ is winning exactly if $(x, y - x)$ is losing.
  - Hence, any stripe across the picture of width $x$ above each $x$ will then repeat above, with all positions reversed. Here is the visualisation, to the right!

Figure. Visualisation of the Limit Process
Our Process

Currently, we have managed to roughly finish the outline of our paper. However, the details of our proofs, description, and arguments still need to be revised and edited.

We plan to work on the paper and finalise it by the end of the Spring 2020 semester, and in the end, we hope to be able to publish this in some mathematical journal. We don’t know what is yet, but we will, so stay tuned!

We hope you enjoy the presentation! Thank you for your time!

Happy Quarantining!
Ian and Zen and Tingley
References


  https://github.com/icowen/nim/tree/master
