COMPUTING STRATEGIES FOR GRAPHICAL NIM

NEIL J. CALKIN, KEVIN JAMES, JANINE E. JANOSKI, SARAH LEGGETT,
BRYCE RICHARDS, NATHAN SITARAMAN, STEPHANIE THOMAS

Abstract. In this paper, we use the Sprague-Grundy theorem to analyze modified versions of Nim played on various graphs. We also describe the periodic behavior of the Sprague-Grundy numbers for games played on paths, and caterpillars. A brief hueristic analysis of the distribution of Sprauge-Grundy numbers for Nim played on graphs of order n is discussed. This research was completed during the Clemson University Math REU which was funded by the NSF Grant: DMS-0552799.

1. Introduction

Nim is a simple impartial two player game in which players take turns removing rocks from disjoint piles until there are no rocks remaining. The player to pick up the last rock (or group of rocks) is the winner. We introduce and analyze the version of Graph Nim where players take turns removing edges from vertices in graphs instead of rocks from piles.

In Graph Nim, the players take turns removing edges that are incident to a given vertex. The number of edges incident to a given vertex is said to be the degree of the vertex; hence, the maximum number of edges that can be removed in a player's turn is equal to the degree of a specific vertex. For example, if a vertex has degree 4, a player can remove 1, 2, 3, or all 4 edges from that vertex. The object of Graph Nim is to be the person to remove the last set of edges from a given vertex.

2. Sprague-Grundy

2.1. Sprague-Grundy Function.

Definition. *A follower is a position a player can obtain in one move in a game.*

Definition. *Given a finite set of integers S, x is the minimum excluded value if it is the smallest non-negative integer such that $x \notin S$.*

For example, for the set $K = \{0, 1, 3, 5, 6, 7\}$ the mex is 2.

Let $F(x)$ denote the followers of a given position x. The Sprague-Grundy function, $g(x)$, is defined as

$$g(x) = \text{mex } \{g(y) : y \in F(x)\} \ .$$

Positions in impartial games, such as Nim, are either N-positions or P-positions. An P-position is winning for the Previous player, while the N-position is winning for the Next player. For the traditional game of Nim, the winning strategy is to finish every move leaving the game's Sprague-Grundy value at zero because of the following theorem.

Theorem. *A position in Nim is a P-position if and only if the nim-sum of its components is zero.*

It follows from the definition of the Sprague-Grundy function that once a player is given a position in a game with a Sprague-Grundy number equal to zero, then any move that player makes will change the value of the game to some non-zero Sprague-Grundy number. It is also known that a player can force a Sprague-Grundy value of zero onto the next player only if the player was not handed a Sprague-Grundy value of zero at the beginning of their turn.

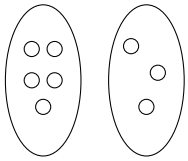
Sprague-Grundy function values are helpful when analyzing Nim played on graphs. The Sprague-Grundy theorem explains the reason why we take interest in computing Sprague-Grundy numbers.

2.2. **Sprague-Grundy Theorem.**

Theorem. *The Sprague-Grundy (S-G) value of a game consisting of many disjoint games is the nim-sum of the Sprague-Grundy values of those components.*

In the case of traditional Nim, the S-G number of the entire game is the addition of each pile's S-G number. In other words, we can consider each pile as a distinct game with its own S-G number. To calculate a nim-sum with traditional Nim, we note that the S-G number for a pile of rocks is simply the number of rocks in the pile. We then take the S-G numbers from all piles and convert them to binary. The nim-sum is then found by the addition of all converted S-G numbers mod 2.

For example, in a game with two piles below,



we take the number of rocks in each pile, 5 and 3, and convert to binary. We then have the nim-sum below.

$$\begin{array}{rcccc} & 1 & 0 & 1 & = 5 \\ + & & 1 & 1 & = 3 \\ \hline & 1 & 1 & 0 & = 6 \end{array}$$

Thus, since the S-G number for this game is greater than zero, player one has a winning strategy.

With the nim sum calculated for a given game, it is possible for a player to determine whether a game win is achievable. We know that a game with a nim-sum of zero is in P-position. Otherwise, if the nim-sum is non-zero, the game is in N-position. So in order for a player to win a game of Nim from an N-position, he should remove enough rocks to force the nim-sum to zero.

3. Graph Nim on Paths

3.1. **Simple Paths.**

Definition. *A path with n edges, denoted P_n, is a tree with two vertices of degree one and all other vertices of degree two.*

An example of P_5 is shown below.

When playing Path Nim, we consider disjoint paths instead of disjoint piles of rocks. A move is made by by removing either one edge, or two edges connected to the same vertex.

From P_5, we take away one edge to create two disjoint paths P_1 and P_3.

Also from P_5, we can take away two edges creating disjoint paths P_1 and P_2.

We can also take away one edge, therefore creating two paths of equal length. Creating two paths of equal length turns out to be the winning strategy for Player 1.

By creating two paths of equal length, we force each path to have the same Sprauge-Grundy number. It is easy to see that when we nim-sum two equal numbers, we get zero. Therefore, if we are faced with P_{2c}, where $c \in \mathbb{Z}^+$, we know we need to remove the two inner edges incident with the center vertex. Similarly, if we are faced with P_{2c+1}, we only need to remove the center edge. Therefore, since we know that Player 1 can always force the nim-sum to zero at the end of their turn, paths can always be won by Player 1. Since cycles are simply paths where the begining vertex is also the terminating vertex, the first movement would result in a path; hence, Player 2 wins Nim played on cycles.

For paths, the Sprague-Grundy numbers are as follows:

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	1	2	3	1	4	3	2	1	4	2	6
12	4	1	2	7	1	4	3	2	1	4	6	7
24	4	1	2	8	5	4	7	2	1	8	6	7
36	4	1	2	3	1	4	7	2	1	8	2	7
48	4	1	2	8	1	4	7	2	1	4	2	7
60	4	1	2	8	1	4	7	2	1	8	6	7
72	4	1	2	8	1	4	7	2	1	8	2	7

The column labels $0 - 11$ represent the least residues of the congruence $l(\text{mod}12)$. The row labels represent the length of path in intervals of 12.

We notice that, a periodic behavior occurs, so that a path of length greater than 72 has an easily computable S-G number. Say that we have a path of length l, where $l \geq 72$, we can use modular arithmetic to calculate its S-G number. For example, for $l = 87$, we have

$$87 \equiv 3 \mod 12 \Rightarrow g(87) = 8.$$

The difficulty with computing S-G numbers for paths of length less than 72 is that there are multiple exceptions before the S-G numbers become become periodic.

4. Graph Nim on Caterpillars

Definition. *A caterpillar, C_n, is defined as a path consisting of n edges with one or more edges appended to t vertices of the path, $t \geq 1$. Any vertex not in the path has degree one and distance one from the main path.*

Definition. *A caterpillar, $C_{n,k}$ is defined as a caterpillar of length n, where n is the number of edges, and consisting of one extra edge (or a leg) on index k, where the leftmost vertex is index zero.*

The gameplay of nim on caterpillars is similar to that on paths. Since there are legs attached to the main path, there are more possible moves available to a player. For example, consider $C_{5,2}$ shown below:

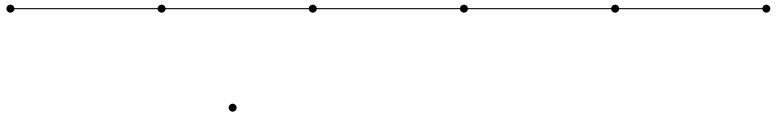
All moves available to a player on a path of length five are also available on this caterpillar (though with different results). In addition, the following four moves are also available for each extra edge:

Removing all three edges connected at index 2, we get P_2 and P_1:

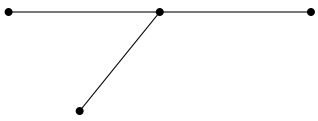
Removing the left edge and the appended edge connected to index 2, we get P_1 and P_3:

Removing the right edge and the appended edge connected to index 2, we get P_2 and P_2:

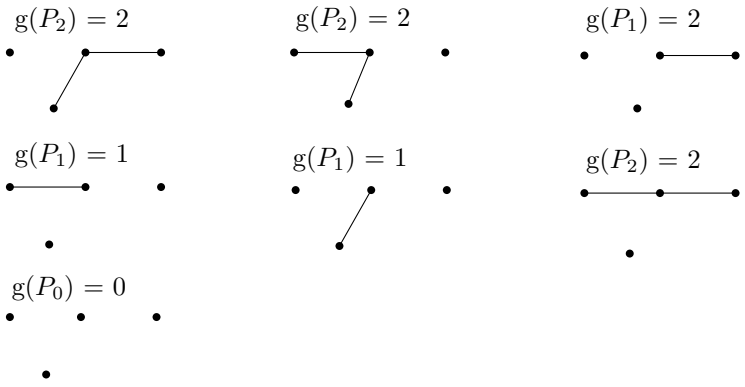
Removing only the appended edge from index 2, we get P_5:



To begin computing S-G values for caterpillars with one leg, we consider the simplest caterpillar $C_{2,1}$:

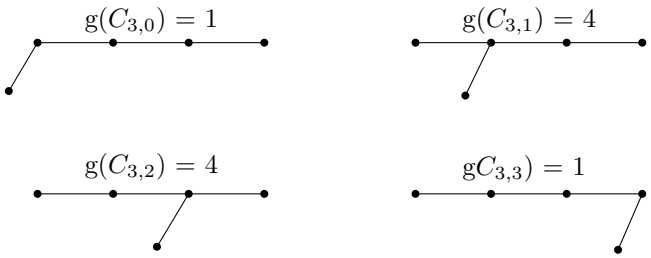


We can obtain the following graphs in one move:

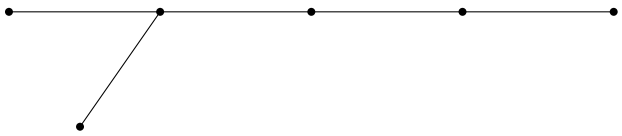


We see that the set of S-G values for followers of $C_{2,1}$ is $\{0, 1, 2\}$ because the resulting moves consist only of the empty graph and paths of lengths one and two. Taking the mex of this set, we conclude that the S-G number of $C_{2,1}$ is 3.

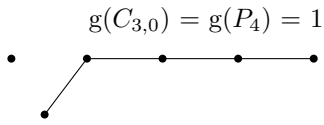
We can continue to analyze caterpillars in this way, obtaining the following S-G numbers for caterpillars of length three:



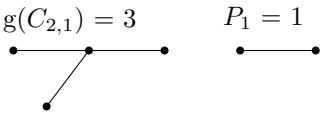
To further understand this process, we can look at a few followers of $C_{4,1}$:



By removing the left most edge from this caterpillar, we obtain the following graph and corresponding S-G number:



Another interesting follower results when we remove the 3rd edge from the caterpillar:



To get the S-G number of this, we must nim-sum 3 and 1. From this nim-sum we obtain 2, which is the S-G number of this follower. We see that the set of the followers' S-G numbers for $C_{4,1}$ is $\{0, 1, 2, 3, 4\}$ and thus the S-G value of $C_{4,1}$ is 5.

When we fix the index of the extra edge and increase the length of the main path, the S-G numbers of caterpillars become perodic. These S-G values are similar to that of paths in that we have discovered periods of 12 in both.

Figure 1 shows the S-G values of caterpillars of the form $C_{n,1}$ and figure 2 shows the S-G values of caterpillars of the form $C_{n,16}$. Note that the periodic behavior of caterpillars does not always begin at the same length. The periodic behavior of $C_{n,1}$ caterpillars begins at length 156, while the periodic behavior of $C_{n,16}$ caterpillars does not begin until length 204.

FIGURE 1. A table of S-G numbers for $C_{n,1}$. The column labels $0-11$ represent the least residues of the congruence $n(\text{mod}12)$. The row labels represent the length of path in intervals of 12. We see a periodic behavior starting at length 156.

	0	1	2	3	4	5	6	7	8	9	10	11
0	$*$	2	3	4	5	6	2	1	0	8	6	0
12	1	2	3	8	5	12	7	1	0	8	9	14
24	1	2	3	11	4	7	12	14	0	16	2	4
36	12	2	3	10	4	7	15	1	16	9	18	16
48	12	2	3	10	16	7	12	1	16	18	11	16
60	12	2	22	11	16	7	12	1	20	24	16	26
72	12	13	22	11	16	24	15	14	16	22	19	16
84	12	13	19	11	16	24	15	14	16	25	11	16
96	12	13	22	11	16	7	15	1	20	25	19	11
108	12	13	22	11	32	19	22	14	20	22	19	11
120	12	13	22	11	25	19	22	14	16	22	19	11
132	21	13	22	11	25	19	22	14	21	25	19	11
144	21	13	22	11	25	19	22	14	20	22	19	11
156	21	13	22	11	25	19	22	14	21	22	19	11

FIGURE 2. A table of S-G numbers for $C_{n,16}$. The column labels $0-11$ represent the least residues of the congruence $n(\text{mod}12)$. The row labels represent the length of path in intervals of 12. We see a periodic behavior starting at length 204.

	0	1	2	3	4	5	6	7	8	9	10	11
0	*	*	*	*	*	*	*	*	*	*	*	*
12	*	*	*	*	4	12	13	0	16	10	7	9
24	0	10	7	9	4	7	16	1	15	10	7	9
36	16	2	10	5	4	12	2	1	0	6	7	12
48	1	16	24	5	19	7	16	0	10	21	7	4
60	24	3	16	27	19	7	27	0	22	10	7	4
72	29	16	26	27	16	7	12	0	21	9	7	32
84	29	12	21	5	36	7	12	0	32	10	7	4
96	28	3	16	9	29	7	12	0	21	9	7	31
108	32	3	21	9	38	7	12	0	21	9	7	40
120	28	3	16	9	29	7	12	0	16	10	7	29
132	22	3	32	9	29	7	12	0	32	10	7	22
144	28	3	35	9	29	7	12	0	16	10	7	22
156	28	3	32	9	37	7	12	0	16	10	7	37
168	28	3	32	9	29	7	12	0	16	10	7	22
180	28	3	32	9	37	7	12	0	16	10	7	22
192	28	3	32	9	37	7	12	0	16	10	7	22
204	28	3	32	9	29	7	12	0	16	10	7	22

5. Nim on Trees

Definition. *A tree is a connected graph that contains no cycles. A tree with n vertices will be denoted* T_n

We can define a caterpillar as a tree such that when all leaves and incident edges are removed the remaining graph is a path. Thus the game of nim played on a tree is similar to that played on a caterpillar. However, the number of trees expands exponentially, as does the number of possible moves.

Sometimes a move on a tree results in a forest. When this situation occurs, we must nim sum the S-G numbers of the trees in the forest to obtain the S-G number of this follower.

Below is a description of the program used to find the S-G numbers of trees.

Program 1. *We first generate a list of unlabled trees with a given number of vertices.*

Suppose for a given number of vertices n, we are given all isomorphism classes of trees of n vertices. Our set of unlabled trees consists of only one tree from each isomorphism class, significantly decreasing the runtime of our program. With this set of trees, we can calculate the S-G number of each non-isomorphic tree.

Iterating through each tree on n vertices, we obtain all possible moves for each tree. This is done by analyzing each vertex of the tree individually and obtaining the set of moves for the specific vertex. We iterate through this set of moves to obtain all followers. The S-G numbers are then calculated for these followers and added to a set of all S-G numbers for the tree being analyzed. We find the minimal excluded value of this set, which is the S-G number of the original tree. We calculate this value for all trees with k edges and then we then repeat the process for trees with k+1 edges.

Thus far, we have noticed that the first player 1 loss occurs at 10 vertices. Of the 106 trees of 10 vertices, 16 are player 1 losses. We have also noticed that there are player 1 losses after 10 vertices, but more computation will be required to extend our analysis.

6. Graph Nim on Graphs

6.1. Winning and Losing Complete Graphs.

Definition. *A complete graph on n vertices, denoted K_n, is the simple graph in which every pair of distinct vertices is connected by an edge.*

Note that

$$e(K_n) = \binom{n}{2}$$

At the outset of this project, it was known that K_1 (trivially), K_3, and K_5 were losing graphs and K_2, K_4, and K_6 were winning. We want to determine if this pattern continues. The proposition below allows us to just consider K_{2n+1}.

Proposition 1. *There are infinitely many winning complete graphs.*

Proof. If K_n is losing then K_{n+1} will be winning, since on K_{n+1} Player 1 can delete the n edges incident to one vertex, leaving the Player 2 with K_n. Thus at least half of the complete graphs must be winning. □

We give the following definitions to allow us to talk about the relationship between two graphs.

Definition. *We call an addition of edges that are all incident to the same vertex an anti-move.*

Definition. *We say that a graph G' is a child of the graph G if G' may be obtained from G in a single move. Equivalently, if G may be obtained from G' in a single anti-move. We call G the parent of G'.*

In order to compute whether K_7 is a win or loss, we need to know the status of all 1043 subgraphs of K_7. Below we describe the program we wrote to find all n-vertex losing graphs, which we were able to run for $1 \leq n \leq 11$.

Program 2. *We use the fact that every parent of a losing graph is a winning graph. We start by creating a list of losing graphs; initially this list consists only of the empty graph on n vertices, letting n = 2k+1 where k ∈ Z. We then iterate through the possible edge-numbers of n-vertex graphs, generating lists of all nonisomorphic graphs with* $1, 2, 3, ..., \binom{n}{2}$ *edges, which we will call* $L_1, L_2, L_3, ..., L_{\binom{n}{2}}$.

After L_m has been generated, we make anti-moves on each of the losing graphs, which have fewer than m edges, in every way possible that leaves an m-edge graph. Since all m-edge graphs obtained in this manner are the parents of losing graphs, they must be winning; we delete them from L_m. After performing all possible anti-moves on the losing graphs and deleting the resulting graphs from L_m, we will have eliminated all of the winning graphs from L_m. Thus, we add any graphs remaining in L_m to our list of losing graphs, generate L_{m+1}, and repeat the above process. In this fashion, we determine whether every graph on n vertices is winning or losing.

This program computed that K_7, K_9, and K_{11} are losing graphs (and, by implication, that K_8, K_{10}, and K_{12} are winning graphs). Thus, we have extended the pattern of complete graphs alternating between winning and losing from $n = 6$ to $n = 12$. We hope to extend this to find a winning strategy for player 2.

6.2. **The Sprague-Grundy Approach.** Rather than directly computing a list of the losing graphs on n vertices, we now discuss the approach of computing the S-G numbers of all n-vertex graphs. Since those graphs with S-G number 0 are the losing graphs, this approach encompasses the approach of the previous section. However, what is gained in information is lost in efficiency; computing the S-G numbers of every n-vertex graph is a more demanding computational task than determining which graphs are losing. The program that finds the S-G numbers of n-vertex graphs operates similarly to the program that finds the S-G numbers for trees, and

so we will provide just a brief description of how it works.

Program 3. *Assuming we have calculated the S-G numbers for all graphs with fewer than m edges, we describe how we will calculate the S-G numbers of the m-edge graphs. For each graph G with m edges, we generate all the children of G. We then find the minimal excluded number of the set of the S-G numbers of these child graphs; this will be the S-G number of G. After computing the S-G numbers of all m-edge graphs in this manner, we move up to $(m+1)$-edge graphs, and continue likewise until the S-G number of every graph on n vertices has been computed.*

Figure 3 shows the distribution of the S-G numbers for 7-vertex graphs.

FIGURE 3. The number in column m and row s indicates the percentage of 7-vertex graphs of size m whose S-G number is s. For instance, 50% of size-2 graphs have S-G number 2, 0% have S-G number 1, and 50% have S-G number 0.

*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	100	0	50.0	20.0	20.0	19.0	12.2	12.3	8.2	9.2	4.1	5.4	8.4	6.2	0	4.9	4.8	10.0	20.0	0	0	100
1	*	100	0	20.0	20.0	23.8	9.8	4.6	17.5	6.1	2.0	3.4	12.2	4.1	0	7.3	0	20.0	0	0	0	0
2	*	*	50.0	0	30.0	9.5	4.9	13.8	5.2	5.3	4.1	8.1	3.1	5.2	6.2	2.4	0	0	0	0	0	0
3	*	*	*	60.0	0	0	12.2	16.9	4.1	4.6	7.4	4.1	3.1	12.4	6.2	4.9	0	0	0	0	0	0
4	*	*	*	*	30.0	4.8	4.9	9.2	15.5	5.3	2.7	2.0	6.9	4.1	6.2	0	23.8	10.0	0	0	0	0
5	*	*	*	*	*	42.9	0	0	5.2	18.3	8.1	1.4	1.5	2.1	9.2	12.2	4.8	0	0	50.0	0	0
6	*	*	*	*	*	*	56.1	0	1.0	0.8	8.1	19.6	3.8	0	0	9.8	0	0	0	50.0	0	0
7	*	*	*	*	*	*	*	43.1	0	0.8	0.7	9.5	9.9	5.2	0	0	0	10.0	40.0	0	0	0
8	*	*	*	*	*	*	*	*	43.3	0.8	0	0	3.1	12.4	6.2	2.4	4.8	0	0	0	100	0
9	*	*	*	*	*	*	*	*	*	48.9	0	0	0	3.1	10.8	17.1	4.8	0	0	0	0	0
10	*	*	*	*	*	*	*	*	*	*	62.8	0	0	0	0	7.3	9.5	10.0	0	0	0	0
11	*	*	*	*	*	*	*	*	*	*	*	46.6	0	0	0	0	14.3	0	0	0	0	0
12	*	*	*	*	*	*	*	*	*	*	*	*	48.1	1.0	0	0	0	20.0	0	0	0	0
13	*	*	*	*	*	*	*	*	*	*	*	*	*	44.3	1.5	0	0	0	0	0	0	0
14	*	*	*	*	*	*	*	*	*	*	*	*	*	*	53.8	0	0	0	0	0	0	0
15	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	31.7	0	0	0	0	0	0
16	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	33.3	0	0	0	0	0
17	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	20.0	0	0	0	0
18	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	40.0	0	0	0
19	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0
20	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0
21	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0

A few observations regarding the S-G number distributions can be made immediately. For instance, as was true with trees, graphs have the maximum possible S-G number (i.e., $e(G) = g(G)$) roughly half of the time. Also, for graphs of size m, the percentage of graphs with a particular S-G number, s, tends to peak at $s \approx 0.6m$, before bottoming out to nearly zero for $0.6m < s < m$. In order to understand these and other patterns in the S-G number distribution, we conducted a heuristic analysis of the data.

6.3. **Heuristic Analysis of S-G Number Distribution.** Given the distribution of the S-G numbers for n-vertex graphs of size $0, 1, \ldots, m-1$, we want to predict what the distribution should be for graphs of size m. Here we present a heuristic method for making this prediction.

We will try to predict the percentage distribution of the S-G numbers for labeled graphs of order n and size m, given that we know the distributions for graphs of order n and size $< m$. Working with labeled graphs has the advantage of allowing us to consider the deletion of any two distinct subsets of edges to be distinct moves, regardless of graph isomorphism.

Definition. *We define a typical labeled graph on n vertices with m edges, $G_{n,m}$, to be a graph whose degree sequence is the average of the degree sequences of all n-vertex, m-edge labeled graphs when the degrees are ordered from least to greatest. We write the degree sequence of $G_{n,m}$ as $(d_1, d_2, \ldots, d_n)$, where $d_1 \leq d_2 \leq \ldots \leq d_n$.*

In order to predict the S-G number distribution for labeled graphs, we ask ourselves the question: For the typical n-vertex, m-edge labeled graph, $G_{n,m}$, what is the probability that the S-G number will be $0, 1, \ldots, m$?

Once we have calculated the degree sequence of $G_{n,m}$, we will know how many children of size $(m-1), (m-2), \ldots, (m-d_n)$ $G_{n,m}$ has. We make the assumption that each of these children of size $(m-k)$ for $1 \leq k \leq d_n$ will be a random labeled graph of size $(m-k)$. That is, a child graph of size $(m-k)$ will have the S-G number $0, 1, \ldots, (m-k)$ with probability equal to the percentage of labeled $(m-k)$-edge graphs whose S-G numbers are $0, 1, \ldots, m-k$. Since we know the number of children, we can can calculate the probability that $G_{n,m}$ will have S-G number $0, 1, \ldots, m$.

In order to run this heuristic, we need the distribution of S-G numbers for labeled graphs of size $m-k$. We find this by counting each graph's S-G number N times, where N is the number of unique relabelings of the graph. Figure 4 shows the distribution of S-G numbers for labeled 7-vertex graphs:

FIGURE 4. The number in column m and row s indicates the percentage of 7-vertex graphs of size m whose S-G number is s.

*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
0	100	0	50.0	2.6	12.3	14.1	9.0	6.7	6.9	4.2	2.1	4.2	6.1	3.9	0	5.2	6.2	10.5	2.6	0	0	100
1	*	100	0	7.9	24.6	13.9	4.7	4.3	16.8	3.8	0.3	0.8	9.4	3.0	0	2.0	0	10.5	0	0	0	0
2	*	*	50.0	0	33.3	8.3	0.6	18.4	3.1	1.7	2.2	7.3	3.6	3.0	2.3	1.2	0	0	0	0	0	0
3	*	*	*	89.5	0	0	9.5	15.1	2.8	3.7	6.1	3.1	1.0	9.1	3.5	0.8	0	0	0	0	0	0
4	*	*	*	*	29.8	12.4	3.5	6.7	12.2	2.6	2.0	1.5	6.6	1.0	1.7	0	11.2	1.8	0	0	0	0
5	*	*	*	*	*	51.3	0	0	3.4	22.4	5.1	0.5	0.2	1.9	5.1	14.5	0.5	0	0	50.0	0	0
6	*	*	*	*	*	*	72.6	0	0.6	0.0	5.3	16.0	0.9	0	0	9.4	0	0	0	50.0	0	0
7	*	*	*	*	*	*	*	48.8	0	0.1	0.7	12.7	10.1	1.8	0	0	0	1.8	39.5	0	0	0
8	*	*	*	*	*	*	*	*	54.1	0.9	0	0	2.3	15.1	3.3	0.1	6.2	0	0	0	100	0
9	*	*	*	*	*	*	*	*	*	60.7	0	0	0	4.3	9.6	14.3	1.5	0	0	0	0	0
10	*	*	*	*	*	*	*	*	*	*	76.1	0	0	0	0	3.4	7.2	21.1	0	0	0	0
11	*	*	*	*	*	*	*	*	*	*	*	53.9	0	0	0	0	10.3	0	0	0	0	0
12	*	*	*	*	*	*	*	*	*	*	*	*	59.9	0.2	0	0	0	12.3	0	0	0	0
13	*	*	*	*	*	*	*	*	*	*	*	*	*	56.8	2.2	0	0	0	0	0	0	0
14	*	*	*	*	*	*	*	*	*	*	*	*	*	*	72.4	0	0	0	0	0	0	0
15	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	49.1	0	0	0	0	0	0
16	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	56.8	0	0	0	0	0
17	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	42.1	0	0	0	0
18	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	57.9	0	0	0
19	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0	0
20	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0	0
21	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0

We need the following definitions to describe our heuristic method.

Definition. *Let $P_e(s)$ to be the probability that a random labeled graph with e edges will have S-G number s. Similarly, let $P_{G_{n,m}}(s)$ to be the probability that our typical graph $G_{n,m}$ has S-G number s.*

Definition. *Let c_k to be the number of children of $G_{n,m}$ with $m-k$ edges.*

Definition. *Let $N_e(s,t_e)$ to be the probability that given t_e random labeled graphs with e edges, none of them will have S-G number s. Note that $N_e(s,t_e) = (1-P_e(s))^t$. Also note that $N_e(s,t_e) = 0$ if $s > e$, since $g(H) \leq e(H)$ for all graphs H.*

Definition. *Lastly, let $N_{G_{n,m}}(s)$ to be the probability that $G_{n,m}$ has no children with S-G number s.*

With this notation, we can describe the exact heuristic method used. Assume that the probability that a child of $G_{n,m}$ with $(m-k)$ edges has S-G number s, is equal to $P_{(m-k)}(s)$. Take $1 \leq j \leq n$ to be the maximum index s.t. $m - d_j \geq s$. Then

$$N_{G_{n,m}}(s) = N_{m-1}(s, c_1) \times \ldots \times N_{m-d_j}(s, c_{d_j})$$

We know that $P_{G_{n,m}}(s)$ is equal to the probability that $G_{n,m}$ has children with S-G numbers $0, 1, \ldots, (s-1)$ and no children with S-G number s. Thus,

$$P_{G_{n,m}}(s) = \{1 - N_{G_{n,m}}(0)\} \times \ldots \times \{1 - N_{G_{n,m}}(s-1)\} \times N_{G_{n,m}}(s)$$

From the above equations, we see that if we can find the number of children of $G_{n,m}$, $c_1, \ldots, c_{d_n}$, we will know the probability that $G_{n,m}$ has S-G number 1 through m. Since $G_{n,m}$ is labeled, computing $c_1, \ldots, c_{d_n}$ is a simple task.

This method yields reasonably accurate predictions of the S-G number distributions, despite our assumptions. Figure 5 shows a table of our heuristic predictions of the S-G number distributions compared to the actual distributions for $n = 7$ and $m = 4, 5, \ldots, 18$.

FIGURE 5. The number in column m and row s indicates the percentage of 7-vertex graphs of size m whose S-G number is s.

edges	*type*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
4	*heuristic*	22.5	55.8	5.4	0	16.3	*	*	*	*	*	*	*	*	*	*	*	*	*	*
4	*computed*	12.3	24.6	33.3	0	29.8	*	*	*	*	*	*	*	*	*	*	*	*	*	*
5	*heuristic*	22.7	12.5	4.3	0	10.3	50.2	*	*	*	*	*	*	*	*	*	*	*	*	*
5	*computed*	14.1	13.9	8.3	0	12.4	51.3	*	*	*	*	*	*	*	*	*	*	*	*	*
6	*heuristic*	17.8	5.7	4.0	7.6	3.5	0.8	60.6	*	*	*	*	*	*	*	*	*	*	*	*
6	*computed*	9.0	4.7	0.6	9.5	3.5	0	72.6	*	*	*	*	*	*	*	*	*	*	*	*
7	*heuristic*	10.1	9.5	15.7	32.2	3.8	0	0	28.7	*	*	*	*	*	*	*	*	*	*	*
7	*computed*	6.7	4.3	18.4	15.1	6.7	0	0	48.8	*	*	*	*	*	*	*	*	*	*	*
8	*heuristic*	5.4	10.4	6.0	5.2	8.1	0.9	0	0.3	63.7	*	*	*	*	*	*	*	*	*	*
8	*computed*	6.9	16.8	3.1	2.8	12.2	3.4	0.6	0	54.1	*	*	*	*	*	*	*	*	*	*
9	*heuristic*	7.2	5.2	1.8	2.0	5.5	27.9	0	0	0	50.3	*	*	*	*	*	*	*	*	*
9	*computed*	4.2	3.8	1.7	3.7	2.6	22.4	0.0	0.1	0.9	60.7	*	*	*	*	*	*	*	*	*
10	*heuristic*	6.4	0.9	5.4	5.4	2.2	3.2	5.1	0.1	0	0	71.3	*	*	*	*	*	*	*	*
10	*computed*	2.1	0.3	2.2	6.1	2.0	5.1	5.3	0.7	0	0	76.1	*	*	*	*	*	*	*	*
11	*heuristic*	2.9	0.5	5.8	2.4	1.4	0	11.6	0.6	0	0	0	74.7	*	*	*	*	*	*	*
11	*computed*	4.2	0.8	7.3	3.1	1.5	0.5	16.0	12.7	0	0	0	53.9	*	*	*	*	*	*	*
12	*heuristic*	7.0	9.0	7.6	2.2	7.0	0	1.4	5.2	0.2	0	0	0	60.4	*	*	*	*	*	*
12	*computed*	6.1	9.4	3.6	1.0	6.6	0.2	0.9	10.1	2.3	0	0	0	59.9	*	*	*	*	*	*
13	*heuristic*	2.3	7.9	1.1	2.1	6.1	0.6	0	0.1	11.2	0	0	0	0	68.7	*	*	*	*	*
13	*computed*	3.9	3.0	3.0	9.1	1.0	1.9	0	1.8	15.1	4.3	0	0	0.2	56.8	*	*	*	*	*
14	*heuristic*	0.6	0.6	0.7	2.3	1.8	16.9	0.1	0	2.7	6.2	0	0	0	0	68.1	*	*	*	*
14	*computed*	0	0	2.3	3.5	1.7	5.1	0	0	3.3	9.6	0	0	0	2.2	72.4	*	*	*	*
15	*heuristic*	0.1	0.1	0.2	0	0.6	6.2	0.3	0	0	0.7	0	0	0	0	0	91.9	*	*	*
15	*computed*	5.2	2.0	1.2	0.8	0	14.5	9.4	0	0.1	14.3	3.4	0	0	0	0	49.1	*	*	*
16	*heuristic*	0.4	0.5	0.5	0	1.9	0.1	2.9	0.3	0	0	12.8	0.1	0	0	0	0	80.5	*	*
16	*computed*	6.2	0	0	0	11.2	0.5	0	0	6.2	1.5	7.2	10.3	0	0	0	0	56.8	*	*
17	*heuristic*	0	1.2	0.8	0	0.7	0	0	6.4	0	0	2.1	3.0	0	0	0	0	0	85.6	*
17	*computed*	10.5	10.5	0	0	1.8	0	0	1.8	0	0	21.1	0	12.3	0	0	0	0	42.1	*
18	*heuristic*	0	1.3	6.2	1.6	0	0	0	39.6	0	0	0	0	0.8	0	0	0	0	0	50.5
18	*computed*	2.6	0	0	0	0	0	0	39.5	0	0	0	0	0	0	0	0	0	0	57.9

7. Conclusion

We have shown a periodic behavior for paths and caterpillars of type $C_{n,k}$. We want to continue this research to all graphs whwere we fix a main graph, G, and attach a path of increasing length to one vertex.

We have also shown that K_{2n+1} is a losing graph for $0 \leq$ n ≤ 5. We want to determine if $K_1 3$ will be winning or losing.

Finally we want to improve our heuristic analysis and extend our method to analyze trees.

References

[1] Theory of impartial games. Lecture Notes, MIT, 2009.

[2] Julius G. Baron. The game of nim-a heuristic approach. Mathematics Magazine, 1974.

[3] Richard K. Guy Elwyn R. Berlekamp, John H. Conway. Winning ways for your mathematical plays, second edition. A K Peters Ltd., 2001.

[4] Thomas S. Ferguson. Game theory. Lecture Notes, Math 167, School, 2000.

[5] Masahiko Fukuyama. A nim game played on graphs. Theoretical Computer Science.