Chomp the Graph

Sam Magura, Vitchyr Pong, Elliot Cartee, Kevin Valakuzhy

Introduction

Chomp the Graph is a two player game played on a finite graph G=(V,E). On a player's turn, they remove part of the graph, either removing a vertex and all edges incident to it or just a single edge; such a removal is called a move. The player who cannot make any more moves (i.e. is faced with an empty graph) loses.

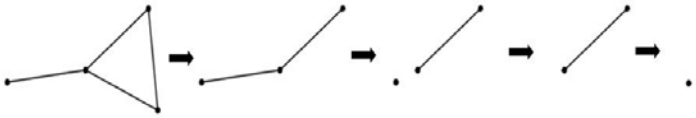


Figure 1. An example progression of Chomp the Graph. The initial graph is the leftmost. The player that makes a move on the rightmost graph, which contains a single vertex, wins the game.

We say that a graph is an N-position the player who plays next will win if using optimal strategy and a P-position otherwise. Our goal is to determine whether a given graph is an N-position or P-position. From these definitions, we see that, from an N-position, there must be a move that will send the game to a P-position; otherwise, the first position would not be an N-position. Likewise, from a P-position, there is no move that will send the game to another P-position.

Background

Chomp the Graph is an impartial game; that is, the available moves depend only on the state of the game, the same moves are available for each player, and the payoffs for winning the game are the same for each player. In other words, the only difference between the two players is that one goes first while the other goes second. Additionally, Chomp the Graph is played under the normal play convention; the player who cannot move loses. Finally, Chomp the Graph is terminating because there are no infinite lines of play. Indeed, for each move, the total number of edges and vertices must always decrease by at least one, making an infinite line of moves impossible.

We can now relate the Chomp the Graph game to another game, Nim. In the game Nim, there are several heaps of "stones". On a player's turn, they choose a heap and remove one or more stones from it.

Figure 2. A progression of a game of Nim with one heap.

Any game of Nim, along with any other impartial game, has an associated nimber, that describe whether the position in N or P. By the Sprague-Grundy theorem, any impartial game that is terminating and played under the normal play convention is equivalent to a Nim heap of a certain size. Therefore, any possible game of Chomp the Graph is equivalent to some Nim heap. Nimbers are defined recursively. Let G={G0, G1, G2, … } be a Nim-equivalent game with options G0, G1, G2, … .That is, a single move can change G to G0 or G1 or G2, and so on. Additionally, let *H denote the nimber of a game H. A Nim heap of size *H can sent to any other Nim heap with a size that is less than *H. Therefore,

$$*G = \text{mex}(\{*G0, *G1, *G2, \dots\}) \quad [1]$$

Here mex(S) denotes the least nonnegative integer not contained in the set S. The mex of an empty set is 0, so a game with no available moves has a nimber of 0. Nimbers are useful because they provide an easy way to predict the outcome of games and sums of game (e.g. a game of Nim with two heaps, each with a known nimber). Nimbers can be added, though not in the same way as integers. Instead, finite nimbers are added with a nim sum as if they were normal integers using the bitwise exclusive-or operator.

EV Parity

In this section, we prove a relationship between the nimber of a game of Chomp the Graph and the number of vertices and edges in the graph. The nimber of the empty graph is 0, since there are no moves that can be made on it. Now, we can recursively determine the nimbers of more complicated graphs using Equation 1.

Through casework, we discovered the following relationship that holds in bipartite graphs between the number of vertices and edges of a graph and its nimber.

A graph with an even number of edges and an even number of vertexes has a nimber of 0.

A graph with an even number of edges and an odd number of vertexes has a nimber of 1.

A graph with an odd number of edges and an even number of vertexes has a nimber of 2.

A graph with an odd number of edges and an even number of vertexes has a nimber of 3.

We call this system of using the parity of the edges and vertices to predict the nimber of a graph EV parity. The EV parity of a graph can be thought of either as a two-digit binary number $(ev)_2$, with e and v each representing single digits, or an ordered pair (e, v). The value of e is 1 if the graph as an odd number of edges, or 0 if it has an even number of edges. The value of v is 1 if the graph as an odd number of vertices, or 0 if it has an even number

of vertices. The binary representation of EV parity is the nimber of the graph.

Proof for Bipartite Graphs

In this section, we prove that a bipartite graph with EV parity (e, v) is equivalent to a nimber ev_2. We do this by establishing that, for a graph G with nimber *G, there exist moves to change the graph's nimber to *G - 1, *G - 2, … 1, 0, but that there is no move that will leave the graph's nimber at *G.

Propostion 1: For any graph, there is no move that will not change the graph's EV parity.

Proof. The player has two options:

Remove a vertex, and all edges connected to it. This changes the vertex parity, and may or may not change the edge parity.

Remove an edge. This changes the edge parity.

Lemma 1: If a graph has an odd number of vertices --- that is, a vertex parity v of 1 --- then the graph contains at least one vertex of even degree.

Proof. Proceed by contradiction; suppose a graph with an odd number of vertices has only vertices of odd degree. However, because there are an odd number of vertices, all of which have odd degree, then the sum of the degrees would be odd. This is impossible since the sum of degrees is the number of edges times two, and therefore must be even. Thus, at least one vertex has even degree.

Proposition 2: If presented with a graph of (0, 1) EV parity, there exists a single move that will change the graph's parity to (0, 0).

Proof. To reset this position to (0, 0), we must remove a vertex of even degree. By Lemma 1, this graph --- which has a vertex parity of 1 --- has such a vertex.

Proposition 3: If presented with a graph of (1, 0) EV parity, there exists a single move that will change the graph's parity to (0, 1) and a single move that will change the graph's parity to (0, 0).

Proof. If the graph's EV parity is (1, 0), there is at least one edge. Remove this edge to change the graph's parity to (0, 0). To change the parity to (0, 1), a vertex of odd degree must be removed. This is always possible. Proceed by contradiction; suppose that in a graph of (1, 0) EV parity, all vertices are of even degree. Then the sum of degrees of all vertices in the graph is divisible by four since there are an even number of vertices in the graph each with an even degree. Because twice the number of edges is the sum of the degrees, the number of edges in this graph must also by even. This is a contradiction, so there is at least one vertex of odd degree in a graph with (1, 0) parity, and the parity can be changed to (0, 1).

Proposition 4: If presented with a bipartite graph of (1, 1) EV parity, there exist moves to change the graph to (1, 0), (0, 1), or (0, 0) parity.

Proof. To change the parity to (1, 0), a vertex of odd degree must be removed. By Lemma 1, since this graph has an odd number of vertices, there is at least one vertex of even degree. To change the parity to (0, 1), remove any edge. Since the edge parity of the graph is 1, there is at least one edge. To reset the parity to (0, 0), a vertex of odd degree must be removed. This is always possible. Proceed by contradiction; suppose the EV parity of a graph is (1,1) and every vertex has even degree. Define such a graph to be special. However, there are no graphs that are both bipartite and special cases, as proven below.

Lemma 3: It is impossible for a graph to be both bipartite and a special case.

Proof. Let G be bipartite and have an EV parity of (1,1). Let its vertices be divided into sets A and B. Starting from just these two groups of vertices, we will attempt to add an odd number of edges in a way such that:

Each edge is incident on one vertex in A and one vertex in B.

Each vertex has even degree.

Let $\beta(A)$ be the sum of the degrees of vertices in A. Let $\beta(B)$ be the sum of the degrees of vertices in B. Before any edges are added $\beta(A) = 0$ and $\beta(B) = 0$. Each time an edge is added, $\beta(A)$ and $\beta(B)$ each increase by one, because a vertex of a bipartite graph can only be adjacent to vertices in the other set. So when m edges have been added, $\beta(A) = m$ and $\beta(B) = m$.

Because the EV parity of the final graph is (1, 1), the final value of m is odd. Therefore, both the sum of the degrees of every vertex in A and the sum of the degrees of every vertex in B are odd. Because the sum of a set of even numbers cannot be odd, at least one vertex in each set must have odd degree.

Theorem 1: A bipartite graph with EV parity (e, v) has a nimber of $(ev)_2$.

Proof. First, let us impose an order on EV parities such that if the EV parity of graph G is greater than the EV parity of graph H, then the proposed nimber of G is also greater than the proposed equivalent nimber for H. Thus, the EV parities, from least to greatest, are: (0, 0), (0, 1), (1, 0), and (1, 1).

The nimber of a game is defined as the smallest nimber not present in the set of the game's options. Our theorem is true if and only if, for all bipartite graphs, the graph's EV parity is the smallest EV parity not contained in the set of its options. We prove this restatement of the theorem by considering it for each of the four EV parities.

A non-empty graph of EV parity (0, 0) cannot have another graph of EV parity (0, 0) in its set of options, by Proposition 1, which states that there is no move that will leave a graph's EV parity unchanged. Thus the smallest EV parity not contained in the set of options is (0, 0), and the theorem holds for this case.

The empty graph, which also has EV parity (0, 0), has no options. The smallest EV parity that is not in the empty set is the smallest EV parity, (0, 0). The theorem holds for this case.

A graph of EV parity (0, 1) has a graph with EV parity (0, 0) in its set of options, by Proposition 2. It does not a graph of parity (0, 1) in its options by Proposition 1. Thus, the smallest EV parity not contained by its set of options is (0, 1).

A graph with EV parity (1, 0) has graphs of EV parity (0, 1) and (0, 0) in its set of options by Proposition 3. It does not have a graph of EV parity (1, 0) in its set of options by Proposition 1, thus the smallest EV parity not contained by its set of options is (1, 0).

A bipartite graph with EV parity (1, 1) has graphs of EV parity (1, 0), (0, 1), and (0, 0) in its set of options by Proposition 4. It does not have a graph of EV parity (1, 1) in its set of options by Proposition 1, thus the smallest EV parity not contained by its set of options is (1, 1).

Thus the theorem holds for all bipartite graphs.

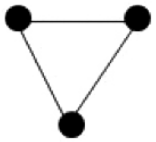
This theorem allows us to predict whether a given bipartite graph is an N-position or a P-position, since:

If the bipartite graph has non-(0, 0) EV parity, it has a nonzero nimber. A Nim heap with a nonzero number of stones is a N-position, so bipartite graphs with non-(0, 0) EV parity are N-positions.

If the bipartite graph has (0, 0) EV parity, the graph has a nimber of 0. A Nim heap with 0 stones is a P-position, so graphs with (0, 0) EV parity are P-positions.

Odd Cycles and EV Parity

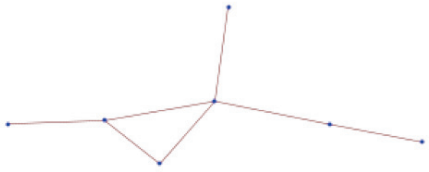
Most graphs can be solved using this idea of EV parity. However, an example of a graph that cannot be solved using EV parity is a triangle.



In this case, the EV parity is (1, 1). However, this graph cannot be sent to a (0, 0) position because there is no vertex of odd degree. (Note: this does not contradict our previous application of EV parity because it is not a bipartite graph.) Even cycles are bipartite graphs, so their positions are easy to calculate. In general, all odd cycles display this property: they have EV parities of (1,1) and have no vertices of odd degree. Still, some graphs containing odd cycles still have predictable positions.

Predictable Odd Cycle Graphs

Some graphs containing odd cycles can be won through a technique called "cycle killing." We focused on cycle killing for graphs that contain a single odd-cycle. In this technique, the player attempts to make a move that both follows EV parity and removes the odd cycle. By doing this, the player changes the graph into a (0, 0) bipartite graph, a losing position for their opponent. The cycle killing strategy works for any single odd cycle graph that has an EV parity of (1, 0). To win, the first player removes any edge from the cycle. We also investigated single odd cycle graphs where a central odd cycle has trees branching off from some or all of its vertexes. For example:



We classified these graphs based on their EV parity and the nature of the odd cycle's vertexes.

O: Each of the cycle's vertexes is incident to an odd number of edges

E: Each of the cycle's vertexes is incident to an even number of edges

M_E: One of the cycle's vertexes is incident to an odd number of edges. The rest are incident to an even number of edges.

M_O: One of the cycle's vertexes is incident to an even number of edges. The rest are incident to an odd number of edges.

M_M: The graph does not fall into any of the previously listed cases.

Through casework, we investigated these categories of graphs and created the following table. Blank spots could be either winning positions or losing positions, or are unknown. N represents a N position, and P represents a P position.

	O	E	M_O	M_E	M_M
(1,1)	N		N	N	N
(1,0)	N	N	N	N	N
(0,1)		N	N	N	N
(0,0)					L

Non-Planar Graphs

It is important to note that the argument for EV parity only requires the graph to be bipartite; the graph can be non-planar. Though planarity could potentially play a role in complex odd cycle graphs, it has no effect on graphs that can be predicted by EV parity.

Identical Subgraph Theorum

Consider a graph G that can be categorized into three distinct subgraphs H_1, H_2, and H_3 where H_1 and H_3 are isomorphic and do not share an edge on G, and H_3 is equivalently connected to H_1 and H_2, which means that for every edge between a vertex v3 on H_3 and vertex v1 on

H_1, there is an edge between v3 on H_3 and vertex v2 on H_2, in which v2 is the isomorphic counterpart to v1. If a graph satisfies these criteria, it is said to be reducible.

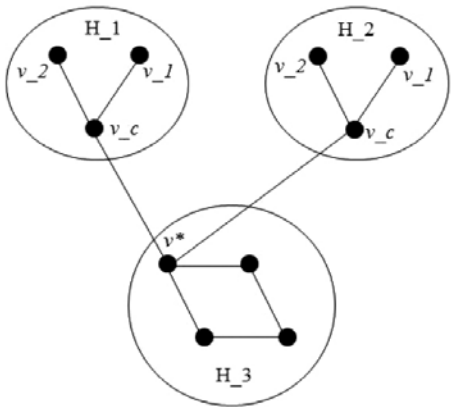


Figure 3. A graph that is reducible. H_1 and H_2 are isomorphic and do not share and edge. H_3 is equivalently connected to both H_1 and H_2.

In the example above, H_1 and H_2 are isomorphic: they are both have a center vertex vc that is connected to two other vertices. Also, H_1 and H_2 and do not share an edge. H_3 is equivalently connected to both H_1 and H_2. In both cases, the edge is connected to v* and vc. So, this graph is reducible.

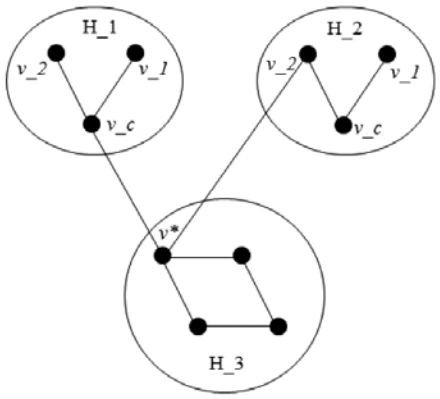


Figure 4. A graph that is not reducible given its current categorizations.

In the example above, H_3 is not equivalently connected to H_1 and H_2. Between H_1 and H_3, there is an edge between v* and v_c, whereas between H_2 and H_3, there is an edge between v* and v_2, H_3.

The Identical Subgraph Theorem (IST) states that reducible graphs has the same position (N or P) as the subgraph H_3.

Proof: Suppose a graph G, with subgraphs H_1, H_2, and H_3 is reducible.

Without loss of generality, suppose H_3 is N-position and that it is Player 1's turn. No matter what Player 1 does, Player 1 cannot avoid playing a game that is equivalent to H_3. Any moves involving H_1 or H_2 are stalling moves that do not affect H_3.

Player 1 has three stalling moves:

If Player 1 removes an edge connecting H_1 and H_3, Player 2 can remove the equivalent edge connecting H_2 and H_3. This end position is also reducible, and H_3 is maintained.

If Player 1 removes a vertex in H_1, Player 2 can remove the equivalent vertex in H_2. This end position is also reducible, and H_3 is maintained.

If Player 1 removes an edge within H_1, Player 2 can remove the equivalent edge within H_2. This end position is also reducible, and H_3 is maintained.

In each of these cases, H_3 is maintained and the graph remains reducible. Since H_1 and H_2 are finite, there **are** a finite number of stalling moves Player 1 can make, so H_1 and H_2 will eventually be removed from the overall graph without affecting H_3. If Player 1 ever plays on H_3, then Player 2 plays on H_3 as if H_1 and H_2 are not part of the game. To show why this does not affect H_1 or H_2, examine the moves on H_3 that Player 1 can make.

If Player 1 removes an edge in H_3, this will not affect H_1 or H_2.

If Player 1 removes a vertex v* in H_3, this will not uniquely affect H_1 or H_2. Since this graph is reducible, v* must be connected to the same vertices in H_1 and H_2, so H_1 and H_2 are identically affected (e.g. both v_c on H_1 and H_2 lose a degree).

Since moves involving H_1 and H_2 are only stalling moves, and moves involving H_3 identically affect H_1 and H_2, the overall graph has the same position as H_3. The IST is especially useful when the position of a graph is not easily predictable using EV parity or a cycling-killing strategy.

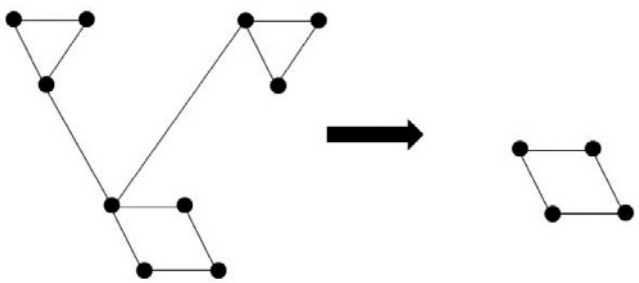


Figure 5. An example use of the Identical Subgraph Theorem to predict a graph.

In this example, the IST is used to simplify a graph containing two odd cycles. Since the graph on the right is a bipartite graph with an EV parity of (0, 0), we know that the original graph had a P-position.

Conclusion

Chomp the Graph is a terminating impartial game that adheres to normal play convention. By the Sprague-Grundy theorem, Chomp must have a nimber, which determines if a position can be won if played optimally. In bipartite graphs, casework proves that this nimber is equal to $(ev)_2$, where e is the number of edges and v is the number of vertices in the graph. This approach is limited by the odd cycles. A graph with only one odd cycle can be predicted by eliminating the cycle and keeping the edge and vertex parity in mind. When a graph goes beyond the scope of cycle-killing, the Identical Subgraph Theorem (IST) provides a method for simplifying the graphs that adhere to a more general standard.