Electronic Colloquium on Computational Complexity, Report No. 21 (2015)

Game Values and Computational Complexity: An Analysis via Black-White Combinatorial Games*

Stephen A. Fenner† University of South Carolina
Daniel Grier‡ MIT
Jochen Messner§
Luke Schaeffer¶ MIT
Thomas Thierauf‖ Aalen University

February 5, 2015

Abstract

A black-white combinatorial game is a two-person game in which the pieces are colored either black or white. The players alternate moving or taking elements of a specific color designated to them before the game begins. A player loses the game if there is no legal move available for his color on his turn.

We first show that some black-white versions of combinatorial games can only assume combinatorial game values that are numbers, which indicates that the game has many nice properties making it easier to solve. Indeed, numeric games have only previously been shown to be hard for NP. We exhibit a language of numeric games (specifically, black-white poset games) that is PSPACE-complete, closing the gap in complexity for the first time between these numeric games and the large collection of combinatorial games that are known to be PSPACE-complete.

In this vein, we also show that the game of Col played on general graphs is also PSPACE-complete despite the fact that it can only assume two very simple game values. This is interesting because its natural black-white variant is numeric but only complete for $\mathsf{P}^{\mathsf{NP}[\log]}$. Finally, we show that the problem of determining the winner of black-white Graph Nim is in P using a flow-based technique.

Keywords: combinatorial games, computational complexity, Graph Nim, poset games, black-white games, numeric games, Col

*The first author was supported by NSF grant CCF-0915948. The second author was supported by the Barry M. Goldwater Scholarship and by the NSF Graduate Research Fellowship under Grant No. 1122374. The third and fifth authors were supported by DFG grant TH472/4.

†Columbia, SC 29208, USA. fenner@cse.sc.edu
‡Cambridge, MA, USA. grierd@mit.edu
§Ulm, Germany. jochen_messner@web.de
¶Cambridge, MA, USA. lrs@mit.edu
‖Aalen, Germany. thomas.thierauf@htw-aalen.de

ISSN 1433-8092

1 Introduction

This paper considers perfect information, two-player combinatorial games. In particular, we investigate whether the value[1] of these games influences the computational complexity of deciding which player should win under optimal play. We consider games that follow the normal gameplay convention: the players alternate moves according to the rules of the game until no move is possible for some player; that player then loses the game.

A combinatorial game is *impartial* if the allowed moves depend only on the position of the game and not on which of the two players is currently moving. Examples of impartial games are NIM, poset games, and GEOGRAPHY, all of which have well-understood complexity [Bou01, Gri13, Sch78]. In contrast, *black-white* games have no options common to both players at any position.[2] Examples include games such as chess, checkers, and go. We explore simple black-white variants of well-known games in Appendix B.

There is general theory of combinatorial games developed by Conway [Con76] and Berlekamp, Conway, & Guy [BCG82] that has served as one of the major tools in the area. It can be thought of as a sort of generalization or analogy to the famous Sprague-Grundy theorem for impartial games [Gru39, Spr36], which neatly distills the properties of NIM that allow it to be solved in polynomial time. In particular, this general theory distinguishes a class of combinatorial games that correspond directly to real numbers and hence share arithmetic operations and order properties with the real numbers. We call these *numeric games*, and we review their properties in Section 2. We encourage readers who are unfamiliar with game values to read Appendix A, which gives a more thorough investigation of their properties. Numeric games are special in that it is never beneficial for either player to make a move. More specifically, if a player can win by moving, (s)he can also win by skipping a turn. Notice that this is not a property universally held by all black-white games (e.g. AMAZONS, black-white NODE KAYLES, HEX).

The standard decision problem associated with a game is to determine whether a given player has a winning strategy. As pointed out in [DH08], most two-player games with bounded length are either PSPACE-complete or in P. Until now, however, there were no classes of numeric games known (to the authors of this paper) to be PSPACE-complete. Furthermore, the few results that are known about numeric games only show NP-hardness (see Blue-Red HACKENBUSH [BCG82]). In this paper, we present a natural class of two-player bounded-length numeric games that is PSPACE-complete, namely, black-white poset games. Since numeric games have previously only been shown to be as hard as NP, there existed hope prior to this result that restricting the set of game theoretic values of a game may make the game easier to play and thus influence its complexity. By presenting a PSPACE-complete numeric game, we provide evidence that no such connection exists.

Despite the fact that numeric games have relatively simple game values, they can still assume game values that are arbitrarily large. Perhaps then it is not merely the nature of the values that affects the complexity of a game but also the number of game values that it can assume. To this end, we investigate the game of Col [BCG82] played on general graphs. We prove that although this class of games can only assume two possible game values, it is still PSPACE-complete[3]. Informally, this has the following consequence for playing many games of Col side-by-side for which we know the corresponding game values. We can perform an extremely simple computation to decide which

[1]Informally, the value of a game indicates which player will win the game and by how much. A more precise definition is given in Appendix A.

[2]These are sometimes called red-blue games in the literature.

[3]To clear possible confusion, Col was mistakenly referenced as being proven PSPACE-complete in [Cin09].

game to play in, but to decide which move to make in that game, we would need to solve a PSPACE-complete problem!

Interestingly, if you take the game of Col and convert it to a black-white version of the game in the natural way, then the game *does* become simpler to solve. In particular, we show the game becomes $\mathsf{P}^{\mathsf{NP}[\log]}$-complete. We conclude the paper with a flow-based technique for solving black-white GRAPH NIM.

1.1 Black-White poset games

Games on partially ordered sets, called *poset games*, are a class of two-player impartial games that have been widely studied. Given a partially ordered set, a player's turn consists of choosing one element e from the set. This element e is then removed along with all elements in the set that are greater than e. Well-studied subfamilies of poset games are NIM, CHOMP, DIVISORS, and HACKENDOT. In the black-white version of the game each element of the set has a color, black or white, and players are only allowed to choose elements of their own color (but choosing an element still removes everything above it, regardless of color).

Grier [Gri13] showed that (impartial) poset games are PSPACE-complete. His proof is a reduction from NODE KAYLES, showed PSPACE-complete by T. J. Schaefer, who also implicitly showed that black-white NODE KAYLES is PSPACE-complete [Sch78]. To show that black-white poset games are also PSPACE-complete, an obvious approach is to adapt Grier's reduction to use the black-white version of NODE KAYLES. However, Grier's construction crucially relies on the fact that both players can remove the same elements, and there is no obvious way to circumvent this restriction. In Section 3, we introduce novel techniques to show that black-white poset games are PSPACE-complete.

1.2 Generalized Col

The game of Col [BCG82] is a two-player combinatorial strategy game played on a simple planar graph. During the game, the players alternate coloring vertices of the graph. One player colors vertices White and the other player colors vertices Black. A player is not allowed to color a vertex neighboring a vertex of the same color. The first player unable to color a vertex loses.

A well-known theorem about Col is that the value of any game is either x or $x + *$ where x is a number. We remove the restriction that Col games be played on planar graphs and consider only those games in which no vertex is already colored. We prove that deciding whether an initially uncolored graph is a win of for the first player is a PSPACE-complete problem. Furthermore, it is easy to adapt the theorem about Col to show that the versions of Col we consider only assume the two very simply game values 0 and $*$.

1.3 NIM on graphs

The game of NIMG simultaneously generalizes the well-known game of NIM and GEOGRAPHY. A graph G is given where each vertex contains a positive number of sticks, and a token rests on a designated start vertex. In the "move-remove" variant we consider[4]—due to Stockman, Frieze, and Vera [SFV04], which we call VERTEX NIMG—each move consists of moving the token along an

[4]Other variants are possible; see Fukuyama [Fuk03] for example, where sticks are placed on edges.

edge to a vertex v then removing at least one stick from v. We will here consider the game on directed graphs and treat undirected graphs as a special case.

GEOGRAPHY and NIM are both special cases of VERTEX NIMG. Lichtenstein & Sipser [LS80] showed that GEOGRAPHY is PSPACE-complete for bipartite graphs, hence NIMG is PSPACE-hard, even in the bipartite case, i.e., the black-white version. Burke and George [BG11] consider another variant called NEIGHBORING NIMG, which corresponds to NIMG on graphs where every vertex has a self-loop. They show that NEIGHBORING NIMG is PSPACE-hard already for *undirected* graphs with ≤ 2 sticks per vertex. In contrast, GEOGRAPHY on undirected graphs is in P [FSU93].

All the considered extensions of NIM on graphs are in PSPACE when the number of sticks is polynomially bounded. However, it is an open problem whether the winner can be determined in PSPACE when we allow exponentially many sticks, i.e., where the numbers of sticks are given in binary. (Clearly, EXP is an upper bound for the general case.)

Analogously with GEOGRAPHY, the black-white version of NIMG is equivalent to the game on bipartite graphs. Since the black-white version of GEOGRAPHY remains PSPACE-complete, this holds for black-white NIMG too. As our one "easiness" result, we show that the black-white version of NIMG on *undirected* graphs is contained in P, even for an exponential number of sticks.

2 Black-white poset games are numbers

The sole purpose of this section is to establish Proposition 2.1, which asserts that black-white poset games are all numbers. In Section 3 below, we will show that determining the winner of a black-white poset game is PSPACE-complete, thus providing the first known instance of a PSPACE-complete numeric game. We start this section by briefly describing numeric games and how they fit in to the more general theory of combinatorial games. We give just enough description to define the notion of a numeric game in the sense of Conway [Con76] or Berlekamp et al. [BCG82], which then allows us to present Proposition 2.1. A more general, detailed background with formal definitions is in Appendix A.

A *black-white poset* is a partially ordered set $P = \langle P, \leq \rangle$ and partition $P = P^B \cup P^W$ with P^B and P^W pairwise disjoint. Elements of P^B are *black* and elements of P^W are *white*. For $x \in P$ let P_x denote the black-white subposet of P with domain $\{y \in P \mid x \nleq y\}$.

Every finite black-white poset P naturally corresponds to a game where, starting with P, in each move a player (Black or White) chooses an element x ($\in P^B$ or $\in P^W$, respectively) of the remaining subposet R and removes all elements $y \geq x$ from R, resulting in the new position R_x. As usual, the first player unable to move loses. When referring to a poset as a game, we always assume that it is finite.

2.1 Operations on poset games

Two poset games can be added in a natural way: If P and Q are black-white posets, then $P + Q$ denotes the disjoint union of P and Q, where no elements of P are comparable with any elements of Q. In the corresponding game, the next player can then choose to play in one poset or the other. This operation is clearly commutative and associative (at least up to isomorphism). We can also negate P by making all white nodes in P black and all black nodes white. The resulting black-white poset game, denoted $-P$, is then equivalent to P but with the roles of the players swapped. We write $Q - P$ as shorthand for $Q + (-P)$. It is not hard to see that $-(-P) = P$ and $-(P + Q) = -P - Q$

(up to isomorphism) for any black-white posets P and Q.

2.2 Comparisons on poset games

Given a black-white poset game P, we say that P is a *first-move win for White* if White has a winning strategy when she plays first. Otherwise, we say that P is a *first-move loss for White* and we denote this $P \geq 0$. Similarly, P is a *first-move win for Black* if Black has a winning strategy when he plays first. Otherwise, we say that P is a *first-move loss for Black* and we denote this $P \leq 0$. Notice that $P \geq 0$ if and only if $-P \leq 0$. It can also be shown that if P and Q are black-white poset games and $P \geq 0$ and $Q \geq 0$, then $P + Q \geq 0$. These basic relations allow us to compare any two black-white poset games P and Q:

- We define $P \leq Q$ to mean $Q - P \geq 0$ (equivalently, $P - Q \leq 0$).
- We define $P < Q$ to mean $P \leq Q$ and $Q \nleq P$.
- We define $P \approx Q$ to mean $P \leq Q$ and $Q \leq P$.

It can be shown that this $\leq$ relation is reflexive and transitive, and hence $\approx$ is an equivalence relation.

2.3 Numeric games

A black-white poset game P is a *number* if and only if, for all $x \in P^B$ and $y \in P^W$, we have $P_x < P_y$. That is, every option for Black is strictly less then every option for White.

Proposition 2.1. *All black-white poset games are numbers.*

Proof. If P is a black-white poset (game), then all options P_x of P have fewer elements than P, so we may use induction on the cardinality of P, i.e., we can assume that all smaller black-white poset games are numbers. It then suffices to show that: (i) $P_x < P$ for every $x \in P^B$; and (ii) $P < P_y$ for every $y \in P^W$. Let x be a black node of P and consider the game $G \coloneqq P_x - P = P_x + (-P)$. We just need to show that $G < 0$, that is, G is a win for White (regardless of who moves first). If White moves first, then she first chooses the white node $x \in -P$ corresponding to the black node $x \in P$. The resulting position is then $P_x - P_x$, which is a zero game, i.e., a second player win; since Black makes the next move, White wins. If Black moves first, then White responds according to the following strategy: If Black chooses an element of P_x or an element of $-P$ that is not above x, then White responds with the corresponding element (oppositely colored) in the other poset. This can continue throughout the play (resulting in a win for White) unless Black at some point chooses some $y \in -P$ such that $y \geq x$. At this point, White immediately responds by choosing $x \in -P$ (which is still present), and the resulting position is of the form $Q - Q$ for some subposet Q of P. Since $Q - Q$ is a zero game and Black makes the next move, White wins. This proves that $P_x < P$.

To show that $P < P_y$ for all $y \in P^W$, we first apply the above proof to $-P$, showing that $-P_y < -P$ for all $y \in (-P)^B$. The result follows by negating both sides, which reverses the sense of the inequality, giving $P < P_y$ for all $y \in (-P)^B = P^W$. □

Proposition 2.1 is important because it establishes a crucial part of our assertion that there is a numeric game that is PSPACE-complete.

Proposition 2.1 implies that if P is a black-white poset game, then $P_x < P$ for all $x \in P^B$, and $P < P_y$ for all $y \in P^W$. Also, any two numeric P and Q satisfy $P \leq Q$ or $Q \leq P$, and so the $\approx$-equivalence classes of numbers are totally ordered by $\leq$.

3 PSPACE-completeness of black-white poset games

Our goal in this section is to show that deciding the winner of a black-white poset game is PSPACE-complete. By standard methods the problem can be solved in polynomial space, so we will focus on the other half of this claim:

Theorem 3.1. *Black-white poset games are* PSPACE*-hard.*

The proof is by a reduction from true quantified Boolean formulas (TQBF), a PSPACE-complete problem. We give the details of the reduction, and prove its correctness in the following subsections.

3.1 Construction

Suppose we are given a fully-quantified boolean formula ϕ of the form

$$\exists x_1 \forall x_2 \exists x_3 \cdots \exists x_{2n-1} \forall x_{2n} \exists x_{2n+1} f(x_1, x_2, \ldots, x_{2n+1})$$

where $f = c_1 \wedge c_2 \wedge \cdots \wedge c_m$ is in conjunctive normal form, with clauses $c_1, \ldots, c_m$. We define a game (not a poset game) based on this formula, called the *TQBF game*, where players take turns assigning the variables either 0 or 1 in turn. That is, white chooses an assignment for x_1, black chooses an assignment for x_2, and so on. When all the variables are assigned, the game ends and white wins if f is true under that assignment, otherwise black wins.

We define our black-white poset game G based on ϕ as follows, where $(X, \leq)$ is the poset.

- The poset is divided into sections. There is a section (called a *stack*) for each variable, a section for the clauses (the *clause section*), and a section for fine-tuning the balance of the game (*balance section*).
- The ith stack consists of a set of incomparable *waiting nodes* W_i above (i.e., greater than) a set of incomparable *choice nodes* C_i. We also have a pair of *anti-cheat nodes*, α_i and β_i, on all stacks except the last stack. For odd i, the choice nodes are white, the waiting nodes are black, and the anti-cheat nodes are black. The colours are reversed for even i.
- The set of choice nodes C_i, consists of eight nodes corresponding to all configurations of three bits (i.e., $000, 001, \ldots, 111$), which we call the *left bit*, *assignment bit* and *right bit* respectively.
- The number of waiting nodes is defined to be

$$|W_i| = (2n + 2 - i)M$$

  where M is the number of non-waiting nodes in the entire game. We will use the fact that $|W_i| \geq |W_{i+1}| + M$ later in the proof.
- The anti-cheat node α_i is above nodes in C_i with right bit 0 and nodes in C_{i+1} with left bit 0. Similarly, β_i is above nodes in C_i with right bit 1 and nodes in C_{i+1} with left bit 1.
- The *clause section* contains a black *clause node* b_j for each clause c_j, in addition to a black *dummy node*. The clause nodes and dummy node are all above a single white *interrupt node*. The clause node b_j is above a choice node z in C_i if the assignment bit of z is 1 and x_i appears positively in c_j, or if the assignment bit of z is 0 and x_i appears negatively in c_j.
- The balance section or *balance game* is incomparable with the rest of the nodes. The game consists of eight black nodes below a white node, which is designed to have game-theoretic value $-7\frac{1}{2}$. All nodes in this section are called *balance nodes*.

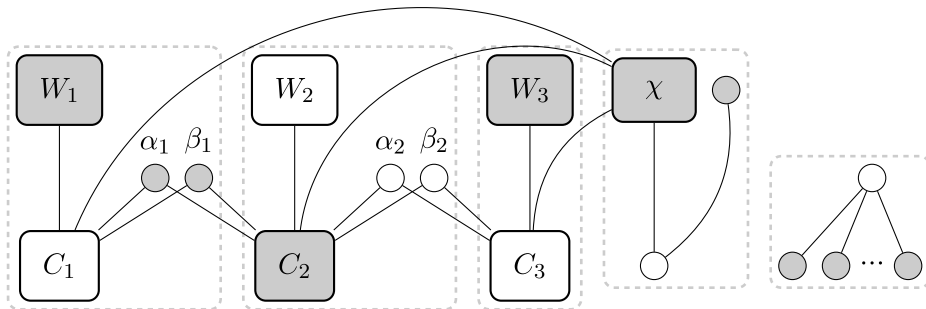

Figure 1: An example game with three variables ($n = 1$). Circles represent individual nodes, blobs represent sets of nodes, and χ is the set of clause nodes. An edge indicates that some node in the lower node set is less than some node in the upper node set. The dotted lines divide the nodes into sections (stacks, clause section and balance section).

The basic idea is that players take turns taking choice nodes, and the assignment bits of the nodes they choose constitute an assignment of the variables, $x_1, \ldots, x_{2n+1}$. The assignment destroys satisfied clause nodes, and it turns out that black can win if there remains at least one clause node. The waiting nodes and anti-cheat nodes exist to ensure players take nodes in the correct order. The interrupt node and dummy node control how much of an advantage a clause node is worth (after the initial assignment), and the balance node ensures the clause node advantage can decide whether white or black wins the game.

It is not hard to see that the number of nodes is polynomial in m and n, so the poset can be efficiently constructed from an instance of TQBF.

3.2 Strategy

We claim that white can force a win if and only if the formula is true. To show this, we need to give a strategy for white when the formula is true, and prove that it guarantees a win. We also need to show black has a winning strategy when the formula is false.

Suppose that white and black have an informal agreement to simulate the TQBF game in G by playing as follows. Suppose white's first move in the TQBF game is to assign x_1 to a_1. The corresponding move in G is to take a choice node in C_1, such that the assignment bit is a_1 and the other bits are arbitrary. Similarly, if black's reply in TQBF is to assign x_2 to a_2 then he should take a choice node in C_2 with assignment bit a_2 and arbitrary right bit. White's first move destroyed either α_1 or β_1, so black should choose the left bit of his reply to preserve the remaining black anti-cheat node in stack 1. Then white takes a node in C_3 such that the assignment bit reflects her assignment of x_3 in the TQBF game, the left bit preserves her anti-cheat node in the previous stack, and the right bit is arbitrary. This continues until white makes the final move in the TQBF game, corresponding to taking a choice node in C_{2n+1}. At this point the TQBF game ends, but there are still nodes in G; we assume the players continue under optimal play.

Assuming both players stick to the agreement, we claim (and will eventually prove) that the winner of the TQBF game is also the winner of G (under optimal play) and therefore deciding the winner of G tells us whether ϕ is true. This is complicated by the fact that players may *cheat* by taking the wrong nodes. Our goal is to show that the winner of the TQBF can also win G, even if the other player cheats.

We think of the game as having two phases. The first phase ends when the players have taken at least one node from each C_i. The second phase begins when the first phase ends, and lasts until the end of the game. If the players simulate a TQBF game as described above then the last move in first phase coincides with the last move in the TQBF game. We analyze the two phases separately.

3.2.1 Phase one strategy and analysis

In phase one, our strategy for white is the same as our strategy for black: play fair (no cheating!) until our opponent cheats. If our opponent cheats then reply according to the following rules, and continue to reply according to these rules for future moves. For the following rules, stack i is the leftmost stack containing waiting nodes of our colour (i.e., we are waiting for our opponent to play in stack i).

- If the opponent's move ends phase one, play according to the phase two strategy. This should only happen when the opponent is white.
- If the opponent moves in C_j then
  - if $j = 2n + 1$ then take a waiting node in W_i.
  - if it is their first move in C_j, reply in C_{j+1}. Choose a node that saves one of your anti-cheat nodes and destroys your opponent's anti-cheat nodes where possible. The assignment bit of your reply will not matter.
  - if it is not their first move in C_j, take a waiting node in W_i.
- If the opponent takes a waiting node in W_{j+1} then take a node in W_j.
- If the opponent takes an anti-cheat node, a clause node, the dummy node, the interrupt node, or a balance node then take a waiting node in W_i.

Observe that we take a waiting node in W_j if the opponent takes a non-waiting node (this can happen at most M times) or takes a waiting node in W_{j+1}. By construction, $|W_j| \geq M + |W_{j+1}|$, so we cannot run out of waiting nodes. Similarly, we only take a node in C_{j+1} when the opponent takes their first node from C_j, so we have all eight nodes to choose from when we play in C_{j+1}. In other words, the strategy never asks us to take a node that isn't there; the reply moves are always feasible.

Lemma 3.2. *Suppose player A (either black or white) commits to following the strategy above. Then it is optimal for player B (white or black) to avoid moves that make player A take a waiting node.*

Proof. Consider two games: in the first game, player B takes a node x, which causes player A to take a waiting node (as prescribed by the strategy) and in the second game, player B skips that move (i.e., does not take x) and plays the next move in the sequence. Since A is playing according to our strategy, her moves do not depend on whether x is present, only on B's most recent move. Therefore both players make the same moves in the same order as the first game, except that B does not take x and A does not take the corresponding waiting node.

Compare the positions in the two games at the end of phase one. If the positions are the same then B has done just as well without taking x. Otherwise, suppose the position is P in the first game and Q in the second game. Every node in P must still be present in Q. Conversely, every node in Q *except x and its ancestors* must still be present in P. In other words, taking x in position Q yields P. Recall that black-white poset games are numbers, so players always prefer *not* to play.

That is, Q is a better position for player B than any position B can move to from Q, including P. We conclude that B is in a better position in the second game, when he skips taking x.

Repeating this argument as many times as necessary, we see that B can skip some moves in an optimal sequence in such a way that optimality is preserved and A does not take any waiting nodes. □

We may therefore assume that player A never takes a waiting node, which narrows down the possible moves for player B. Specifically, player B must take a choice node from a previous untouched stack, otherwise player A will wait. Player B can still cheat by taking a choice node from the wrong stack, and we need to analyze phase two to understand how anti-cheat nodes prevent that kind of cheating.

3.2.2 Phase Two Strategy and Analysis

Let H be the black-white poset game at the start of phase two, and let k be the number of surviving clause nodes in H. Each player took exactly one choice node from each stack in phase one, and since there are more white C_i's, black has the first move in phase two. The waiting nodes in W_i are gone because some node in C_i is missing for all i. Similarly, there is at most one anti-cheat node in each stack, since at least one was destroyed by the missing choice nodes on either side.

Our analysis of phase two begins with a series of lemmas.

- We claim a player can always avoid destroying their own anti-cheat nodes in H, and therefore we may assume it is impossible for a player to destroy their own anti-cheat node. This gives us a new, equivalent game H'.
- It is optimal (in H') for white to take the interrupt node after black's first move, as long as the dummy node is intact.
- It is optimal for black to take a clause node on his first move in H'.

It follows that the clause nodes are gone by black's second move. Then every section (i.e., each stack, the clause section, or the balance section) is incomparable with the rest of the nodes. We use the following basic result about poset games to divide the game into a sum of much simpler games. Given the number of anti-cheat nodes of each colour and the number of clause nodes, we use this to rapidly determine the winner in a number of scenarios, proving the main result.

Let us start with our first result. Informally, it is a claim that a player cannot be forced to destroy their own anti-cheat nodes, so we may assume it is impossible for a player to destroy their own anti-cheat nodes.

Lemma 3.3. *Let H be as above. Construct a game H' from a copy of H by making α_i and β_i incomparable with nodes in stack $i+1$, for all i. That is, let the anti-cheat nodes be incomparable with nodes of the same colour. Then $H = H'$ in the game theoretic sense.*

Proof. For each i, either α_i or β_i is missing. We assume (without loss of generality) that all β_i's have been destroyed.

We claim that $H + (-H')$ is a win for the second player. The strategy is the same whether the second player is white or black: when your opponent takes a node in H (resp. $-H'$), take the twin node in $-H'$ (resp. H), with a few exceptions, listed below.

- Suppose our opponent takes a choice node $x = x_\ell x_a x_r$ in some C_{i+1} from $-H'$, where x_ℓ, x_a and x_r are the three constituent bits of x. The nodes $0x_a x_r$ and $1x_a x_r$ are indistinguishable in the poset for $-H'$, so we assume our opponent takes $1x_a x_r$ first and then $0x_a x_r$.

When our opponent plays $1x_ax_r$, we take the twin node. When our opponent plays $0x_ax_r$, there are no longer any nodes above its twin, except possibly α_i, since we destroyed all other nodes when we took $1x_ax_r$ (or when someone took $1x_ax_r$ in phase one). If α_i exists then we take it, otherwise we take $0x_ax_r$.

- If our opponent takes α_i in $-H'$ then we take its twin in H, if possible. If not, then at some point we took α_i instead of some choice node $0x_ax_r$ in stack $i+1$, so now we can take $0x_ax_r$.

We conclude that $H = H'$. □

Note that in H', a move in one stack does not affect any other stack directly. However, taking a node in C_i may destroy clause nodes, which will indirectly affect the optimal move in some other stack. The stacks in H' will not be completely independent until we can get rid of the clause nodes. We prove a lemma and corollary which show that clause nodes will not survive for very long.

Lemma 3.4. *Let H' be as above. It is optimal for white to take the interrupt node in H' after black's first move, as long as the dummy node exists.*

Proof. Let K be the position after black's first move and consider the alternatives for white.

- Take the interrupt node.
- Take a choice node x in some stack i.
- Take an anti-cheat node α_i (the analysis is similar for β_i).
- Take the white balance node, w. This may not be possible, depending on black's move.

Call the resulting positions K_{int}, K_{choice}, K_{cheat}, and K_{bal} respectively. We claim K_{int} is the best for white, so we need to show that $K_{\text{int}} + (-K_{\text{choice}})$, $K_{\text{int}} + (-K_{\text{cheat}})$ and $K_{\text{int}} + (-K_{\text{bal}})$ are wins for white when black goes first. We use the same kind of mirroring strategy as we did in the previous lemma: when your opponent takes a node in one subgame, you take the twin node in the other subgame, with a few exceptions.

$K_{\textbf{int}} + (-K_{\textbf{choice}})$

There are two exceptions: the interrupt node, and the anti-cheat node above x.

- When black takes the interrupt node in $-K_{\text{choice}}$ then let white take the choice node x in K_{int} (there is no interrupt node in K_{int}). Clearly x has no twin in $-K_{\text{choice}}$, and there are no nodes below x, so it will exist when black takes the interrupt node.
- If black takes α_i in K_{int} then let white take the twin in $-K_{\text{choice}}$ if possible. If the twin does not exist in $-K_{\text{choice}}$ then it must be because α_i is above x. We deduce that x still exists in K_{int}, so the interrupt node still exists $-K_{\text{choice}}$. The dummy node still exists because it has no twin and the interrupt node exists below it, so white may take the dummy node.

$K_{\textbf{int}} + (-K_{\textbf{cheat}})$

The only black node that does not have a twin is the interrupt node, but we also have to account for black nodes that could destroy α_i.

- If black takes a choice node y in K_{int} and destroys α_i then take the dummy node in $-K_{\text{cheat}}$. Note that α_i is the only node destroyed by taking y, since there are no clause nodes in K_{int}.
- If black takes the interrupt node in $-K_{\text{cheat}}$ then take α_i if possible. Otherwise α_i is missing because we took it in the previous exception, but there is a leftover twin for y in $-K_{\text{cheat}}$ instead.

$K_{\textbf{int}} + (-K_{\textbf{bal}})$

The exceptional nodes are the interrupt node and black balance nodes.

- If black takes a node z in the balance game (in K_{int}) it will destroy w and nothing else. We reply by taking the dummy node.
- If black takes the interrupt node in $-K_{\text{cheat}}$ then we take w if possible. Otherwise we must have destroyed w according to the exception above, so we can now take z's twin in $-K_{\text{bal}}$ instead.

In each game, white has a reply for every move black makes, so white will win when black eventually runs out of moves. We conclude that taking the interrupt node is optimal for white. □

If we can show that black does not take the dummy node on his first move, then the lemma tells us how white will respond, and we can use that information to determine black's optimal first move.

Corollary 3.5. *Let H' be as above. If there are any clause nodes in H' (i.e., $k \geq 1$) then it is optimal for black to take a clause node.*

Proof. Let J_{clause} be the position after black takes a clause node b_j in H'. Let J_{dummy} be the position after black takes the dummy node. We claim it is better for black to take *any* clause node than to take the dummy node, so we need to show that $J_{\text{clause}} + (-J_{\text{dummy}})$ is a win for black when white goes first. As usual, the strategy for black is to mirror white's moves, with one exception. If white takes b_j in J_{dummy} then we take the dummy node in J_{clause}. It is not hard to show that the dummy node will exist if b_j exists, since they are above the interrupt nodes in their respective games. We conclude that taking any clause node is at least as good for black as taking the dummy node, and we may assume black does not take the dummy node.

Given that black does not take the dummy node, we may assume (by Lemma 3.4) that whatever black chooses to do, white will take the interrupt node. Observe that if black takes a non-clause node x and white takes the interrupt node then we end up in the same position as if white took the interrupt node (out of turn, since black has the first move in H') and black took x. In other words, the two moves commute. On the other hand, if black takes a clause node (or a dummy node) and white takes the interrupt node, then it is the same as if black did not move at all and white took the interrupt node. The position after white takes the interrupt node is a poset game (with a number value), so it is better for black not to move. In other words, it is optimal for black to take some clause node. □

Once the clause nodes are gone, each stack in H' is completely independent of the rest of the nodes. Let H_i be the black-white poset game by restricting H' to the nodes in stack i. We present without proof the following proposition about the value of H_i.

Proposition 3.6. *Let H_i be as above. Then H_i has value ± 7 without an anti-cheat node, and $\pm 6\frac{1}{2}$ with an anti-cheat node, where the sign is $(-1)^{i+1}$. Note that the last stack, $i = 2n+1$, does not contain an anti-cheat node.*

The balance section is also independent of the rest of the nodes, and its value is $-7\frac{1}{2}$ by construction. Assuming all the anti-cheat nodes survive, the stacks and balance nodes sum up to a value of

$$6\tfrac{1}{2}\sum_{i=1}^{2n}(-1)^{i+1} + 7 - 7\tfrac{1}{2} = -\frac{1}{2}.$$

We say the *baseline value* of the game is $-\frac{1}{2}$, and adjust it up or down depending on how many anti-cheat nodes there are of each colour, and how the clause section (containing the clause nodes, interrupt node and dummy node) plays out. We can now prove our main result.

Theorem 3.7. *White has a winning strategy for G if and only if ϕ is true.*

Proof. It suffices to show that white can win G if she can win the TQBF game, and black can win G if he can win the TQBF game.

Suppose white has a winning strategy for the TQBF game. Suppose further that white plays according to our strategy (outlined in Section 3.2) and black does not cheat. That is, white takes a node in C_1, black takes a node in C_2, and so on until white takes a node in C_{2n+1} and phase two begins. Since white can win the TQBF game, she can force an assignment that satisfies all clauses in the formula, so there are no clause nodes left. The dummy node and interrupt node together have a value of $\frac{1}{2}$, so with the baseline value of $-\frac{1}{2}$ for the other nodes, the total value of the game is 0, a win for white, the second player.

Now suppose black tries to cheat to prevent white from winning. We assume that black cheats by taking choice nodes from a stack too early, since our phase one analysis rules out any other kind of cheating. Black's cheating may be enough to keep a clause node alive, since changing the order of quantifiers can easily affect the outcome of the TQBF game. We know that black will then take the clause node (by Corollary 3.5) and white takes the interrupt node (by Lemma 3.4).

The price black pays for cheating is one anti-cheat node. For instance, suppose black cheats by taking a node in C_{j+1} before white has taken C_j, destroying one of the anti-cheat nodes, α_j and β_j, in stack j. When white eventually takes a node in C_j, she can choose the right bit to destroy the other anti-cheat node in stack j. White plays according to strategy, so she will not take C_j until black takes a node in C_{j-1}, so white can also choose the left bit to preserve her own anti-cheat nodes. At the end of phase one, white will have n anti-cheat nodes and black will have at most $n-1$, leading to a $\frac{1}{2}$ advantage for white over the baseline of $-\frac{1}{2}$. The net value is 0, and therefore a win for white.

The analysis for black is similar. Suppose black has a winning strategy for the TQBF game. If white does not cheat, then black can force there to be an unsatisfied clause. Black takes the corresponding clause node at the beginning of phase two, and then white takes the interrupt node. The remaining game has no clause section and both sides have n anti-cheat nodes, so the value is the baseline value, $-\frac{1}{2}$, and a win for black. If white tries to cheat then she may be able to destroy all clause nodes, but at the expense of at least one anti-cheat node. The clause section adds $\frac{1}{2}$, but losing an anti-cheat node subtracts $\frac{1}{2}$, so we are back to the baseline, $-\frac{1}{2}$ and a win for black. □

4 Generalized Col is PSPACE-complete

Let Col be the language of Col games on uncolored general graphs where the first player has a winning strategy. Assume that the graphs are represented in some explicit manner, such as an adjacency matrix. We will show that Col is PSPACE-complete by giving a reduction from a game played on propositional formulas known to be PSPACE-complete [Sch78]. The game, $\mathrm{G}_{\mathrm{pos}}$(POS CNF), is played on a positive CNF formula. The players take turns choosing a variable that appears in the formula. Player 1 sets variables to true, and Player 2 sets variables to false. Once all the variables have been chosen, Player 1 wins if the formula evaluates to true, and Player 2 wins if the formula evaluates to false.

Theorem 4.1. Col *is* PSPACE-*complete.*

Most of the rest of this section (Section 4) is dedicated to proving Theorem 4.1. The diagrams in the proof use the interpretation of Col in which the players remove vertices from the graph, tinting their neighbors so as to reserve them for the other player. Figure 2 shows this simple coloring scheme.

●	-	Only available to Black.
○	-	Only available to White.
•	-	Available to both players.

Figure 2: Coloring Scheme for Col graph.

Let G be the graph for some Col game which may already be partially colored. Assuming that vertices x and y in G are not already colored, we will let $G^{b(x)w(y)}$ denote the graph G where x has been chosen by black and y has been chosen by white. Other game states are defined in an analogous fashion.

4.1 Preliminaries

We will first show that a slight variation of $\mathrm{G_{pos}}$(POS CNF) is also PSPACE-complete. Let $\mathrm{G^*_{pos}}$(POS CNF) be identical to $\mathrm{G_{pos}}$(POS CNF) except that Player 1 sets variables to false in an attempt to make the formula false and Player 2 sets variables to be true with the goal opposite to that of Player 1. We will show that this game is also PSPACE-complete. Let X be the set of variables in the $\mathrm{G_{pos}}$(POS CNF) game and let $c_1, c_2, \ldots, c_m$ be the clauses. Let the $\mathrm{G^*_{pos}}$(POS CNF) game be played with variables $X \cup \{u\}$ and formula $(c_1 \vee u) \wedge (c_2 \vee u) \wedge \ldots \wedge (c_m \vee u)$. Notice that if Player 1 does not make u false, then Player 2 will make u true and win the game. It is now easy to see that Player 1 wins the $\mathrm{G_{pos}}$(POS CNF) game iff Player 2 wins the $\mathrm{G^*_{pos}}$(POS CNF) game.

4.2 Main Construction

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of variables for the $\mathrm{G^*_{pos}}$(POS CNF) game played on CNF formula with clauses $C = \{c_1, c_2, \ldots, c_m\}$. We will construct a Col graph $G = (V, E)$ such that Player 1 wins the $\mathrm{G^*_{pos}}$(POS CNF) game on φ iff the second player wins the Col game on G. The elements of G are as follows:

- $V = X \cup Y \cup C \cup \{z\}$.
- X is the set of variables in the $\mathrm{G^*_{pos}}$(POS CNF) game.
- $Y = \{y_1, y_2, \ldots, y_n\}$ is a copy of the set of variables in the $\mathrm{G^*_{pos}}$(POS CNF) game such that x_i refers to the same variable as y_i for $1 \le i \le n$.
- C is the set of clauses in the $\mathrm{G^*_{pos}}$(POS CNF) game.
- $E = A \cup B \cup C \cup D$.
- $A = \{(y_i, x_i) \mid 1 \le i \le n\}$.
- $B = \{(x_i, c_j) \mid \text{ variable } x_i \text{ appears in clause } c_j\}$.
- $C = \{(c_i, c_j) \mid 1 \le i < j \le m\}$.
- $D = \{(z, c_j) \mid 1 \le j \le m\}$.

An example of this construction on the formula is given in Figure 3.

4.3 Assumptions that can be made about optimal gameplay

We will now give several lemmas that show how the Col game is to be played when constructed from some valid $\mathrm{G^*_{pos}}$(POS CNF) instance. In order to simplify the exposition of the proof, we will

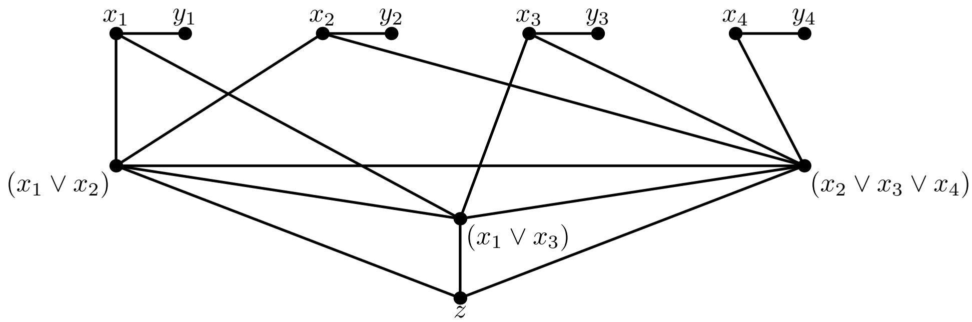

Figure 3: Example of Col construction on $(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_3 \vee x_4)$.

assume without loss of generality that White is the first player and Black is the second player.

Lemma 4.2. *Let G be the Col game constructed from some valid G^*_{pos}(POS CNF) instance. We have $G^{w(x)} > 0$ for $x \in V - \{z\}$.*

Proof. Notice first that in the Col game played on G, each player can only hope to color a maximum of $n+1$ vertices during the game. A player can color $n+1$ vertices if he colors one of the vertices of each (x_i, y_i) pair and one of either z or $c_1, c_2, \ldots, c_m$. The second player, who we are assuming is Black, wins the game if is able to choose at least as many vertices as the first player. Thus, if White chooses $x \in V - \{z\}$, then Black will respond by choosing z. Since z is not a neighbor to any (x_i, y_i) pair, Black will be able to color $n+1$ vertices and eventually win the game. So, $G^{w(x)} \nleq 0$. Using this logic, it is also clear that $G^{w(x)} \geq 0$. Since Black can win as both first and second player, $G^{w(x)} > 0$. This completes the lemma. □

Thus, White must choose the vertex z on his first turn if he hopes to win. The remaining lemmas make this assumption.

Lemma 4.3. *Let G be the partially colored graph representing the state of the Col game constructed from a G^*_{pos}(POS CNF) instance such that an (x_i, y_i) pair is not yet colored. Regardless of which player's turn it is, coloring y_i is always just as good or better than coloring x_j. That is, $G^{w(y_i)} \leq G^{w(x_j)}$ and $G^{b(x_j)} \leq G^{b(y_i)}$.*

Proof. First consider $i = j$. Intuitively, since each player can only pick at most one of x_i or y_i, y_i is the better move because it is not connected to any other vertices of G. Playing x_i potentially reserves moves for your opponent. Formally, the game $G^{w(y_i)} - G^{w(x_i)} \leq 0$ by the obvious correspondence between the vertices shown in Figure 4. A similar argument shows $G^{b(x_i)} \leq G^{b(y_i)}$. Using the dominating rule, we can assume then that under optimal gameplay each player will color the element y_i.

Next consider $i \neq j$. Suppose it is White's turn and y_j has already been colored by Black, reserving x_j for White. Using a mirroring strategy, White wins the game as the second player on $G^{w(y_i)} - G^{w(x_j)}$ shown in Figure 5. The strategy is as follows. If at any time Black plays in $G^{w(y_i)}$ then White plays the same move in $-G^{w(x_j)}$ and vice versa unless Black colors a vertex of type x_i, x_j, or y_i. Table 1 is a list of winning responses to each move of Black. These moves will necessarily leave two points of type x_i, x_j, or y_i, which natural correspond to each other. It is easy to see that this mirroring strategy allows White to win as the second player.

There are three remaining cases with proofs that are left up to the reader due to their similarity to the argument above: Black's turn and y_j has already been colored by White; White's turn and y_j has not yet been colored; Black's turn and y_j has not yet been colored. □

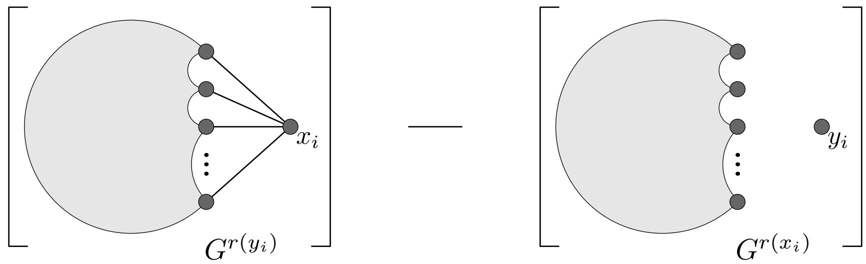

Figure 4: Visualization of the game $G^{w(y_i)} - G^{w(x_i)}$. Right wins as the second player by using simple mirroring strategy.

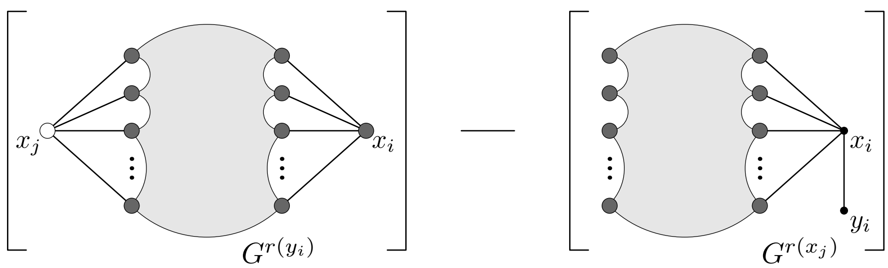

Figure 5: Visualization of the game $G^{w(y_i)} - G^{w(x_j)}$. White wins as the second player by using mirroring strategy in conjunction with the moves in Table 1.

Black's Move	$x_i \in G^{w(y_i)}$	$x_i \in -G^{w(x_j)}$	$y_i \in -G^{w(x_j)}$
White's Response	$x_i \in -G^{w(x_j)}$	$y_i \in -G^{w(x_j)}$	$x_j \in G^{w(y_i)}$

Table 1: Table of winning responses for White playing on the game depicted in Figure 5. They are not unique.

Proof of Theorem 4.1. Let G be the Col game constructed from some valid $\mathrm{G}^*_{\mathrm{pos}}$(POS CNF) instance, and assume without loss of generality that White plays first and that both players are playing optimally. By Lemma 4.2, White must take vertex z first. The players will then alternate choosing all of the y vertices due to Lemma 4.3. At this stage in the game all of vertices that remain in the graph are tinted, so no move of Black will affect White and vice versa. Furthermore, Black can only win if he can color more vertices than White, so he might as well proceed by choosing all of the x vertices now available to him. Since there are an even number of variable vertices and Black chooses the first variable vertex, White will play the last variable vertex. So Black wins the game iff there is a clause vertex available after all variable vertices are chosen.

So we can think of the game as the following. When White chooses a variable y_i, he is essentially letting x_i be true in the $\mathrm{G}^*_{\mathrm{pos}}$(POS CNF) game. And when Black chooses a variable y_i, he is essentially letting x_i be false in the $\mathrm{G}^*_{\mathrm{pos}}$(POS CNF) game. If the formula is false at the end of variable selection, then there will be some clause that is available to Black, and he will win. If however, the formula is true at the end of variable selection, then Black cannot play in any of the clauses and will lose.

Thus, Player 1 has a winning strategy on the $\mathrm{G}^*_{\mathrm{pos}}$(POS CNF) game iff Black has a winning strategy on the constructed Col game. Since the construction is clearly in polynomial time, this shows that COL on general graphs is PSPACE-hard. Furthermore, a simple enumeration of all possible game paths shows that finding the winner of any game of Col on general graphs is in PSPACE. This completes the proof. □

The complexity of COL does, in fact, stem from the vertices available to both players. First notice that the game of Col can also be thought of in the following manner. If Black chooses a vertex, delete that vertex and tint all neighboring vertices white, so that they are now only available to White. Similarly, if White chooses a vertex, delete that vertex and tint all neighboring vertices black, preserving them for Left. If a node is tinted both white and black, then it is available to neither player and can be deleted for clarity. For the purposes of displaying Col graphs in this paper, this interpretation will be used. Furthermore, this interpretation begets a natural black-white version of COL. That is, given a general graph where all nodes are initially tinted black or white, decide which player has the winning strategy.

Theorem 4.4. *Black-white Col is* $\mathsf{P}^{\mathsf{NP}[log]}$ *– complete.*

Proof. Suppose there is some edge between a vertex tinted white and a vertex tinted black. This edge can be removed without affecting the game because selecting either vertex would not affect the tint of the other vertex. The players then are simply vying to choose the largest independent set from the vertices that remain. This problem was proven $\mathsf{P}^{\mathsf{NP}[\log]}$-complete in [SV00]. □

5 Black-white NIMG on undirected graphs is in P

We consider the following problem associated with the game of NIMG.

> Undirected black-white NIMG
> *Input*: An undirected bipartite graph $G = (V, E)$, a weight function $w : V \longrightarrow \mathbb{N}$, and a node $v \in V$.
> *Question*: Is v a winning position in the NIMG game (G, w)?

We show the problem is in P even with a binary encoding of weights by providing a polynomial-time reduction to the maximum flow problem.

Let $G = (V, E)$ be a bipartite undirected graph. Let $V = B \cup W$ be the partition of V into black and white nodes, and w be the weight function such that $w(u)$ is the number of sticks on node u. Let v denote the vertex on which the token is placed currently. Remember that in a turn, a player moves the token to a neighbor u of v and removes, say, r sticks from u. The weight function w' after the move is given $w'(u) = w(u) - r$ and $w'(x) = w(x)$ for $x \neq u$.

Observe that we may assume that each player removes only $r = 1$ stick at each turn. This is because White removes sticks only from white vertices, and similarly for Black. Therefore a winning strategy for a player remains winning when the number of sticks removed in each turn is decreased to one.

We now construct a flow network G_N with capacity function c from G and w as shown in Figure 6. The following lemma characterizes the winning positions in a game. See [FF62] or some

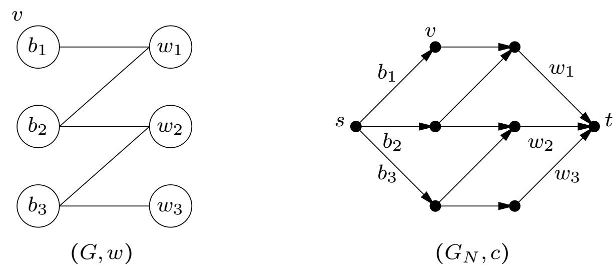

Figure 6: *A bipartite graph $G = (B \cup W, E)$ with weight function w on the vertices, and the flow network G_N constructed from G. We add s and t and connect s to all vertices of B and all vertices of W to t. As capacities define $c(s, v) = w(v)$ for $v \in B$ and $c(v, t) = w(v)$ for $v \in W$. The original edges of E (i.e., the edges going from B to W) have unbounded capacity. Edges that are not present in G_N have capacity zero.*

other standard textbook for more details on flow terminology and the properties of flows that are used in the proof.

Lemma 5.1. *Let (G, w) be a weighted undirected bipartite graph and (G_N, c) be the flow network associated with (G, w). Then a node v is a winning position for the current player if and only if he can move so that the resulting flow network has the same maximum flow value as (G_N, c).*

Notice that no move in (G, w) can ever increase the max flow value of the corresponding flow network, because edge capacities never increase. Lemma 5.1 then follows straightforwardly from the next two claims:

Claim 5.2. *Let (G, w) and (G_N, c) be as in Lemma 5.1. Any two consecutive moves starting with (G, w) result in a position whose flow network has max flow value less than that of (G_N, c).*

Proof. Let u and v be the two vertices with sticks removed in any two consecutive moves starting with (G, w). Then u and v are on opposite sides of the bipartition of G—so we assume without loss of generality that $u \in B$ and $v \in W$—and further, (u, v) is an edge of G (it is not important which vertex is chosen first). Let (G_N, c') be the corresponding flow network after the two moves. Then $c'(s, u) = c(s, u) - 1$ and $c'(v, t) = c(v, t) - 1$. Thus any feasible flow f' in (G_N, c') can be augmented through the path $s \to u \to v \to t$ to obtain a feasible flow f in (G_N, c) whose value is one plus that of f'. Thus (G_N, c') has smaller max flow value than (G_N, c). □

Claim 5.3. *Let (G, w) and (G_N, c) be as in Lemma 5.1. Suppose some move in (G, w) decreases the max flow value of the corresponding flow network. Then there exists an immediately subsequent move that does not change the max flow value of the corresponding flow network.*

Proof. Let u be the vertex that has a stick removed by the former move (coming from some vertex v, say), and let (G, w') be the resulting game position with corresponding flow network (G_N, c'). We consider the case where $u \in W$; the case where $u \in B$ is similar. We then have $c'(u, t) = c(u, t) - 1$, and all other capacities are the same in (G_N, c') as they are in (G_N, c). Let f be a maximum feasible flow in (G_N, c'). By assumption, f is feasible but not maximal in (G_N, c). Thus there exists an augmenting $s \to t$ path p of residual capacity 1 for f in (G_N, c). This path p must traverse edge (u, t), for otherwise, p would be an augmenting path for f in (G_N, c'), contradicting the maximality of f. Let $v' \in B$ be the predecessor of u along p, i.e., $p = s \to \cdots \to v' \to u \to t$.

We show next that there is a maximum feasible flow f' in (G_N, c') which does not saturate edge (s, v') in (G_N, c'), i.e., $f'(s, v') < c'(s, v')$. We distinguish two cases:

- Path p uses edge (s, v'), that is, $p = s \to v' \to u \to t$. Then $f(s, v') < c(s, v') = c'(s, v')$, and so f does not saturate (s, v') in (G_N, c'). Thus we can take f' to be f.
- Path p does not use edge (s, v'), that is, $p = s \to \cdots \to u' \to v' \to u \to t$, for some node $u' \in W$. Because $c(u', v') = 0$, an augmenting path along (u', v') decreases an actual flow in the other direction, i.e., we have $f(v', u') > 0$. A positive flow from v' can only originate from some flow via (s, v'). Hence we have $0 < f(s, v')$. We now obtain the flow f' by modifying f: first adding 1 unit of flow along p, then *subtracting* 1 unit of flow along the path $s \to v' \to u \to t$. The resulting flow f' (feasible in (G_N, c)) satisfies $f'(s, v') = f(s, v') - 1$ and has the same value as f. It is also feasible in (G_N, c'), because $f'(u, t) = f(u, t)$. Now f' is a maximum flow in (G_N, c') and does not saturate (s, v').

Since in either case, $0 \le f'(s, v') < c'(s, v') = w'(v')$, there is a stick available at v' in (G, w'), making the move from u to v' legal. Let (G_N, c'') be the flow network corresponding to the position after the move to v', that is, $c''(s, v') = c'(s, v') - 1$ and $c''(e) = c'(e)$ for all other edges e of G_N. Figure 7 illustrates the situation. Note that f' is still a feasible flow in (G_N, c''). Therefore, f' is a maximum flow in (G_N, c'') too, and the max flow values of (G_N, c') and (G_N, c'') are the same.

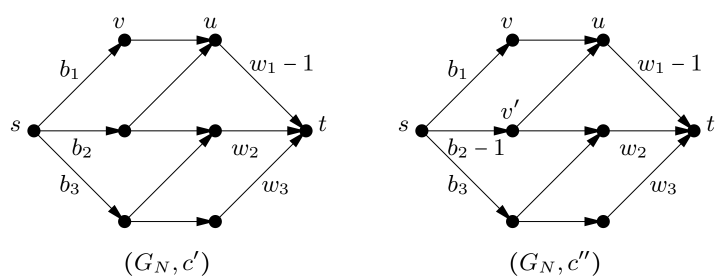

Figure 7: *After the move from v to u in the network (G_N, c), the resulting network is (G_N, c'). The next move to v' yields the network (G_N, c''). The capacities of edges (u, t) and (s, v') are decreased by one because in the corresponding* NimG*-game the players remove one stick at u and one at v' in turn.*

□

Proof of Lemma 5.1. The proof of the Lemma proceeds by induction on the total number of sticks on the vertices of the graph, noting that each move removes a stick. In the base case, where there is no stick in the graph, the statement is clearly true. The idea for the inductive step is that if the current player can move without decreasing the max flow value, then the next player has no choice but to decrease the max flow value on his subsequent move, and conversely, if the current player cannot help but to decrease the max flow value, then the next player can move afterward without decreasing the max flow value. First, suppose that from v there is a move that does not change the max flow of the associated flow network. Then by Claim 5.2, the next player cannot help but to decrease the max flow, making his a losing position by the inductive hypothesis. Thus v is a winning position.

Conversely, suppose that every legal move from v decreases the max flow. Then by Claim 5.3, after any such move—to a new vertex u, say—the next player can move to maintain the same max flow, making u a winning position by the inductive hypothesis. Thus v is a losing position. Clearly this also holds if no move is possible from v. □

Finding maximum flow values in networks with integral capacities can be done in polynomial time, even if those capacities are given in binary (see, e.g., [Din70]). To see whether v is a winning position in (G, w), we compute the maximum flow f in the associated network (G_N, c). Now we go through all neighbors u of v in G with $w(u) > 0$. Construct the networks (G_N, c_u), where $c_u(u,t) = c(u,t) - 1$ (c_u is otherwise equal to c), and compute the maximum flow f_u in (G_N, c_u). From Lemma 5.1 it follows that we find a u such that $|f| = |f_u|$ if, and only if, v is a winning position in (G, w). Thus we have the following theorem.

Theorem 5.4. *Black-white* NimG *on undirected graphs is in* P.

6 Conclusion and open problems

We have shown that it is PSPACE-hard to determine the winner of a black-white poset game, which is important in that it establishes a PSPACE-complete numeric game. We also show that a Col played on uncolored general graphs is PSPACE-complete, which is the first game known to the authors that can only assume two very simple game theoretic values and still be PSPACE-complete. These two results cast doubt on the possibility that there is some connection between the range of values that a family of games can assume and the complexity of deciding the winner of a game in that family. An interesting open question is to definitively prove that no such connection exists. For instance, given (reasonable) game values x and y, is it possible to construct a PSPACE-complete game whose value always simplifies to either x or y? More concretely, now that we have a PSPACE-completeness result for a numeric game, can we hope to use it as a template for other numeric games with longstanding open complexity (e.g. Red-Blue Hackenbush)?

For NimG, a combination of Nim and Geography, we have considered the black-white version on undirected graphs and have shown that it is decidable in P who wins even when one allows an exponential number of sticks. This is somewhat surprising given that winning gameplay may require an exponential number of moves. For all the other versions of NimG that have been considered in the literature the exact complexity of the binary encoded versions is open. Some of these games are known to be PSPACE-hard, and yet we still do not know about membership in PSPACE.

Acknowledgments

We would like to thank an anonymous referee for suggesting a simplified proof of Lemma 5.1.

References

[BCG82] E. R. Berlekamp, J. H. Conway, and R. Guy. *Winning Ways for your Mathematical Plays.* Academic Press, 1982.

[BG11] K. Burke and O. George. A PSPACE-complete graph Nim. http://arxiv.org/abs/1101.1507v2, 2011.

[Bou01] C. L. Bouton. Nim, a game with a complete mathematical theory. *The Annals of Mathematics*, 3(1/4):35–39, 1901.

[Cin09] Alessandro Cincotti. Three-player Col played on trees is NP-complete. In *International MultiConference of Engineers and Computer Scientists 2009*, pages 445–447. Newswood Limited, 2009.

[Con76] J. H. Conway. *On Numbers and Games*. Academic Press, 1976.

[DH08] E. D. Demaine and R. A. Hearn. Constraint logic: A uniform framework for modeling computation as games. In *23rd Annual IEEE Conference on Computational Complexity*, pages 149–162. IEEE, 2008.

[Din70] E. A. Dinic. Algorithm for solution of a problem of maximum flow in a network with power estimation. *Soviet Math. Doklady*, 11:1277–1280, 1970.

[FF62] L.R. Ford and D.R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[FSU93] A. S. Fraenkel, E. R. Scheinerman, and D. Ullman. Undirected edge geography. *Theoretical Computer Science*, 112:371–381, 1993.

[Fuk03] M. Fukuyama. A Nim game played on graphs. *Theoretical Computer Science*, 304:387–399, 2003.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[Gri13] D. Grier. Deciding the winner of an arbitrary finite poset game is PSPACE-complete. In *Proceedings of the 40th ICALP*, volume 7965-7966 of *Lecture Notes in Computer Science*, pages 497–503. Springer-Verlag, 2013.

[Gru39] P. M. Grundy. Mathematics and games. *Eureka*, 2:6–8, 1939.

[LS80] D. Lichtenstein and M. Sipser. Go is polynomial-space hard. *J. ACM*, 27(2):393–401, 1980.

[Sch78] T. J. Schaefer. On the complexity of some two-person perfect-information games. *Journal of Computer and System Sciences*, 16(2):185–225, 1978.

[SFV04] G. Stockman, A. Frieze, and J. Vera. The game of Nim on graphs: NimG, 2004. http://www.aladdin.cs.cmu.edu/reu/mini_probes/2004/nim_graph.html.

[Sip05] M. Sipser. *Introduction to the Theory of Computation (2nd Ed.)*. Course Technology, Inc., 2005.

[Spr36] R. P. Sprague. Über mathematische Kampfspiele. *Tohoku Mathematical Journal*, 41:438–444, 1935-1936.

[SV00] H. Spakowski and J. Vogel. θ_p^2-completeness: A classical approach for new results. *FST TCS 2000: Foundations of Software Technology and Theoretical Science: 20th Conference, New Delhi, India, December13-15, 2000 Proceedings*, 1974, 2000.

A Games and numbers: some background

In this section of the appendix we review some relevant definitions and a few facts about combinatorial games and their associated numbers. Thorough treatments of this material, with lots of examples, can be found in [BCG82, Con76] as well as other sources. Our terminology and notation is somewhat different from [BCG82, Con76], but the concepts are the same. When we say, "game," we always mean what is commonly referred to as a *combinatorial game*, i.e., a game between two players, say, Black and White, alternating moves with perfect information, where the first player unable to move loses (and the other wins). These games can be defined abstractly by what options each player has to move, given any position in the game.

Definition A.1. A *game* is an ordered pair $G = (B, W)$, where B and W are sets of games. The elements of B (respectively, W) are the *black options* (respectively, *white options*) of G. An *option* of G is either a black option or a white option of G.[5]

For this and the following inductive definitions to make sense, we tacitly assume that the "option of" relation is well-founded, i.e., there is no infinite sequence of games $g_1, g_2, \ldots$ where g_{i+1} is an option of g_i for all i.[6] A *position* of a game G is any game reachable by making a finite series of moves starting with G (the moves need not alternate colors). Formally,

Definition A.2. A *position* of a game G is either G itself or a position of some option of G.

Starting with a game G, we imagine two players, Black and White, alternating moves as follows: the initial position is G; given the current position P of G (also a game), the player whose turn it is chooses one of the options of P matching her or his color, and this option becomes the new game position. The first player faced with an empty set of options loses. The sequence of positions obtained this way is a *play* of the game G. Our well-foundedness assumption implies that every play is finite, and so there must be a winning strategy for one or the other player. We classify games by who wins (which may depend on who moves first) when the players play optimally.

Definition A.3. Let G be a game.

- G is a *first-move win for White* iff there exists a white option w of G that is a first-move loss for Black.
- G is a *first-move win for Black* iff there exists a black option b of G that is a first-move loss for White.
- G is a *first-move loss for White* (written $G \geq 0$) iff G is not a first-move win for White (equivalently, every white option of G is a first-move win for Black).
- G is a *first-move loss for Black* (written $G \leq 0$) iff G is not a first-move win for Black (equivalently, every black option of G is a first-move win for White).

[5]It is more common in the literature to use the terms *left option* and *right option* instead of black option and white option, respectively. It is also more traditional to use the notation $\{B|W\}$ or $\{b_1, b_2, \ldots | w_1, w_2, \ldots\}$ rather than (B, W), where $B = \{b_1, b_2, \ldots\}$ and $W = \{w_1, w_2, \ldots\}$.

[6]This follows from the Foundation Axiom of set theory, provided ordered pairs are implemented in the standard way: $(x, y) := \{\{x\}, \{x, y\}\}$ for all sets x and y.

Definition A.4. Let G be a game.

- G is a *zero game* (or a *first player loss*, written $G \approx 0$) iff $G \le 0$ and $G \ge 0$.
- G is *positive* (or a *win for Black*, written $G > 0$) iff $G \ge 0$ and $G \nleq 0$.
- G is *negative* (or a *win for White*, written $G < 0$) iff $G \le 0$ and $G \ngeq 0$.
- G is *fuzzy* (or a *first player win*, written $G \parallel 0$) iff $G \nleq 0$ and $G \ngeq 0$.

For example, the simplest game is the *endgame* $0 := (\varnothing, \varnothing)$, which is a zero game. The game $1 := (\{0\}, \varnothing)$ is positive, and the game $-1 := (\varnothing, \{0\})$ is negative, while the game $* := (\{0\}, \{0\})$ is fuzzy.

Games can be added and subtracted. The sum $G + H$ of two games G and H is the game where on each move, a player may decide in which of the two games to make a move. The negation $-G$ of G is the same as G but with the roles of Black and White reversed. Formally:

Definition A.5. Let $G = (B_G, W_G)$ and $H = (B_H, W_H)$ be games. We define

- $-G := (\{-w : w \in W_G\}, \{-b : b \in B_G\})$ and
- $G + H := (\{b + H : b \in B_G\} \cup \{G + b : b \in B_H\}, \{w + H : w \in W_G\} \cup \{G + w : w \in W_H\})$.

We write $G - H$ as shorthand for $G + (-H)$. One can show that + is commutative and associative when applied to games, and the endgame 0 is the identity under +. One can also show for all games G and H that $-(-G) = G$ and $-(G + H) = -G - H$. Furthermore, $G \ge 0$ iff $-G \le 0$, and if $G \ge 0$ and $H \ge 0$, then $G + H \ge 0$. It is *not* the case, however, that $G - G = 0$ for all G, although $G - G$ is always a zero game (i.e., $G - G \approx 0$).

Definition A.6. Let G and H be games.

- G and H are *equivalent* (written $G \approx H$) iff $G - H \approx 0$.
- We write $G \le H$ to mean $G - H \le 0$ (equivalently, $H - G \ge 0$).
- We write $G < H$ to mean $G \le H$ but $H \nleq G$ (equivalently, $G - H < 0$).

The $\le$ relation on games is reflexive and transitive, which makes $\approx$ an equivalence relation as the terminology suggests. The + operator respects equivalence ($G \approx G'$ and $H \approx H'$ imply $G + H \approx G' + H'$). A special subclass of games are called numbers.

Definition A.7. A game $G = (B, W)$ is a *number* (or is *numeric*) iff every option of G is a number, and in addition, $b < w$ for every $b \in B$ and $w \in W$.

One can show that $G = (B, W)$ is a number if and only if $b < G$ for every $b \in B$ and $G < w$ for every $w \in W$. If H is also a number, then either $G \le H$ or $H \le G$. The + and − operations also yield numbers when applied to numbers. Numeric games have a peculiar property: making a move only worsens your position (for Black this means having to choose a smaller number; for White, having to choose a larger number). For these games, an optimal play is then easy to describe: Black always chooses a maximum black option, and White always chooses a minimum white option.[7] This intuitive is formalized in the following theorem that is referred to in the literature as the "dominating rule".

Theorem A.8. *Let $G = (B, W)$ be a game. If $y \le b$ for some $b \in B$, then $(B, W) = (\{y\} \cup B, W)$. Similarly, if $y \ge w$ for some $w \in W$, then $(B, W) = (B, W \cup \{y\})$.*

[7]This assumes that the game has a finite number of positions. In general, Black can do OK by choosing any option $b \ge 0$, and White can do OK by choosing any option $w \le 0$.

A.1 Finite numeric games

Numeric games that are *finite*, i.e., that have a finite number of positions, correspond to dyadic rational numbers according to the following "simplicity rule":

Definition A.9. Let $G = (B, W)$ be a finite numeric game. The *value* of G, denoted $v(G)$, is the unique rational number $a/2^k$ such that

1. k is the least nonnegative integer such that there exists an integer a such that $v(b) < a/2^k$ for all $b \in B$ and $a/2^k < v(w)$ for all $w \in W$, and
2. a is the integer with the least absolute value satisfying (1.) above.

So for example, the endgame 0 has value $v(0) = 0$, the game 1 has value $v(1) = 1$, and the game -1 has value $v(-1) = -1$, as the notation suggests. In fact, for any two finite numeric games P and Q, one can show that $v(P + Q) = v(P) + v(Q)$ and that $v(-P) = -v(P)$. Also, $P \leq Q$ if and only if $v(P) \leq v(Q)$.[8] Note that the valuation map v is not one-to-one; for example, $v((\{-1\}, \{1\})) = v(0) = 0$.

B Black-White Games with Straightforward Reductions

As a first example, consider Geography. The input is a directed graph G and a designated vertex s of G on which a token initially rests. The two players alternate moving the token on G from one node to a neighboring node, trying to force the opponent to move to a node that has already been visited. Geography is a well-known PSPACE-complete game [Sch78, Sip05].

An obvious way to turn Geography into a black-white game is to color the nodes of graph G black and white. Each player is then only allowed to move the token to a node of their own color. Since moves are allowed only to neighboring nodes, the black-white version is equivalent to the impartial version on bipartite graphs. The standard method of showing that Geography is PSPACE-complete is via a reduction from True Quantified Boolean Formulas (TQBF) to Geography (see for example [Sip05]). Observe that the graph constructed in this reduction is not bipartite. That is, there are nodes that potentially may be played by both players. Hence, we cannot directly conclude that the black-white version is PSPACE-complete. However, in [LS80] Lichtenstein & Sipser show that Geography is indeed PSPACE-complete for bipartite graphs.

We now consider the game Node Kayles. This game is defined on an undirected graph G. The players alternately play an arbitrary node from G. In one move, playing node v removes v and all the direct neighbors of v from G. In the black-white version of the game, we once again color the nodes black and white. Schaefer [Sch78] showed that determining the winner of an arbitrary Node Kayles instance is PSPACE-complete. He also extended the reduction to bipartite graphs, which automatically yields a reduction to the black-white version of the game (see [GJ79]). Therefore, black-white Node Kayles is also PSPACE-complete.

[8] One can define purely game-theoretic multiplication operation on numeric games in such a way that $v(PQ) = v(P)v(Q)$ for all P and Q.

ECCC ISSN 1433-8092
http://eccc.hpi-web.de