On the Complexity of the Maker-Breaker Happy Vertex Game

Mathieu Hilaire
Univ. Bordeaux, Bordeaux INP, LaBRI UMR CNRS 5800, F-33400, Talence, France

Perig Montfort
ENS de Lyon, France

Nacim Oijid
Umeå University, Sweden

Abstract

Given a c-colored graph G, a vertex v of G is said to be *happy* if it has the same color as all its neighbors. The notion of happy vertices was introduced by Zhang and Li [27] to compute the homophily of a graph. Eto, Fujimoto, Kiya, Matsushita, Miyano, Murao and Saitoh [11] introduced the Maker-Maker version of the Happy vertex game, where two players compete to claim more happy vertices than their opponent. We introduce here the Maker-Breaker happy vertex game: two players, Maker and Breaker, alternately color the vertices of a graph with their respective colors. Maker aims to maximize the number of happy vertices at the end, while Breaker aims to prevent her. This game is also a scoring version of the Maker-Breaker domination game introduced by Duchene, Gledel, Parreau and Renault [8], as a happy vertex corresponds exactly to a vertex that is not dominated in the domination game. Therefore, this game is a very natural game on graphs and can be studied within the scope of scoring positional games [3]. We initiate here the complexity study of this game, by proving that computing its score is PSPACE-complete on trees, NP-hard on caterpillars, and polynomial on subdivided stars. Finally, we provide the exact value of the score on graphs of maximum degree 2, and we provide an FPT-algorithm to compute the score on graphs of bounded neighborhood diversity. An important contribution of the paper is that, to achieve our hardness results, we introduce a new type of incidence graph called the *literal-clause incidence graph* for 2-SAT formulas. We prove that QMAX 2-SAT remains PSPACE-complete even if this graph is acyclic, and that MAX 2-SAT remains NP-complete, even if this graph is acyclic and has maximum degree 2, i.e. is a union of paths. We demonstrate the importance of this contribution by proving that Incidence, the scoring positional game played on a graph is also PSPACE-complete when restricted to forests.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory

Keywords and phrases Maker-Breaker game, Domination game, happy vertex game, scoring game, complexity

Digital Object Identifier 10.4230/LIPIcs.FUN.2026.24

Funding This research was supported by the ANR project P-GASE (ANR-21-CE48-0001-01).
Nacim Oijid: Kempe Foundation Grant No. JCSMK24-515 (Sweden).

1 Introduction

1.1 Framework of the study

Happy vertices have been introduced in graphs to measure their homophily: a vertex is *happy* if it has the same color as all its neighbors. This concept has been introduced by Zhang and Li [27], and has been the subject of several studies afterwards; see for instance [1, 2, 20, 28]. In this context, Eto, Fujimoto, Kiya, Matsushita, Miyano, Murao and Saitoh have introduced the happy vertex game [11] as a two-player game, in which two players, Black and White, alternately color the vertices of a graph G with their own color and aim to have more happy

vertices of their color than their opponent. We focus here on the Maker-Breaker version of the game, in which Maker and Breaker again alternately color the vertices of a graph, Maker in red, and Breaker in blue. Maker aims to maximize the number of happy red vertices, and Breaker aims to minimize it. Hence, when all the vertices are colored, the number of red happy vertices is precisely the number of vertices not dominated by Breaker, which makes this game a scoring version of the well-known *Maker-Breaker domination game*.

The Maker-Breaker domination game, introduced by Duchêne, Gledel, Parreau and Renault [8] in 2020, is one of the most studied positional games played on the vertices of graphs. In this game, two players, Dominator and Staller, alternately claim the unclaimed vertices of a given graph G until no vertices remain. Dominator wins if the set of vertices she has claimed is a dominating set; otherwise, i.e. if Staller has isolated one vertex by claiming it and all its neighbors, Staller wins. The algorithmic study of this game led to its PSPACE-completeness on bipartite graphs of bounded degree or split graphs [8, 23], and to polynomial algorithms on several classes of graphs, including paths, grids, unit interval graphs, and regular graphs [4, 7]. This game is part of the larger field of Maker-Breaker positional games, which are games played on a hypergraph with two players, Maker and Breaker, alternately claiming the vertices of the hypergraph. Maker wins if she manages to claim all the vertices of some hyperedge; otherwise, Breaker wins. Positional games were introduced by Hales and Jewett in 1963 [15], and proved to be PSPACE-complete by Schaefer in 1978 [25] for hypergraphs of rank 11, i.e. whose largest hyperedge has size 11. The study of their complexity has recently gained strong interest with the three consecutive improvements on the minimal rank for which the game remains PSPACE-complete to rank 6 by Rahman and Watson [24], to rank 5 by Koepke [17], and to rank 4 by Galliot [12]. On the positive side, Galliot, Gravier, and Sivignon proved that the game is solvable in polynomial time on hypergraphs of rank 3 [13].

Here, in the Maker-Breaker happy vertex game, the ruleset is the same as in the Maker-Breaker domination game, except that, once all the vertices have been claimed, we count the number of vertices isolated by Staller, instead of asking if one exists. This consideration makes this game in the universe of scoring positional games, introduced by Bagan, Deschamps, Duchêne, Durain, Effantin, Gledel, Oijid and Parreau [3]. They proved that scoring positional games belongs to Milnor's universe, a large framework of scoring combinatorial games, which grants some structure on the game [21]. They also proved that computing the score of a scoring Maker-Breaker positional game is PSPACE-complete, even restricted to hypergraphs of rank 2, i.e. graphs, and they provided several tools to handle this class of games.

1.2 Outline of the paper

In this paper, we first remind some useful results about Milnor's universe and the domination game in Section 2. In particular, we use Milnor's group structure to handle disconnected graphs, which can be handled through his results on sum of games, and we use the results on the domination game to focus our study on specific graph classes.

In Section 3, we introduce several variants MAX 2-SAT, together with a new representation: the literal-clause incidence graph. We prove that a quantified version of MAX 2-SAT is PSPACE-complete even when we restrict it to instances for which this graph is acyclic, and that MAX 2-SAT is NP-complete, even when we restrict it to instances for which this graph is acyclic and has maximum degree 2. As a corollary, we prove that Incidence, the scoring positional game played on a graph is PSPACE-complete, even restricted to forests, answering an open question of [3]. We then turn back to the Happy vertex game, and we prove that even though the Maker-Breaker Domination game can be solved in polynomial

time on forests, its scoring version that we study here is already PSPACE-complete on trees. This phenomenon leads us to the study of specific subclasses of trees, hence, we prove that the game is already NP-hard on caterpillars (Section 3), but can be solved in polynomial time on subdivided stars and union of paths (Section 4). Concerning the latter result, we can extend it to graphs of maximum degree 2 since isolated cycles will not change the score of the game.

Finally, in search for extending results known from scoring positional games, we adapt the so-called super-lemma to this game and manage to provide an FPT-algorithm in Section 4.3 parameterized by the neighborhood diversity, a graph parameter introduced by Lampis [18] that measures the variety of the neighborhoods of vertices in the graph.

2 Preliminaries

2.1 Definitions and Notations

Let $G = (V, E)$ be a graph, and let v be a vertex of G. We denote by $N(v)$ the *open neighborhood* of v, i.e. we have $N(v) = \{u \in V(G) \mid (u, v) \in E(G)\}$. We denote by $N[v]$ the *closed neighborhood* of v, i.e. $N[v] = \{v\} \cup N(v)$.

We denote by $Ms(G, M, B)$ the score obtained on (G, M, B) when Maker plays next, and $Bs(G, M, B)$ when Breaker plays next. Note that we will always consider that both players play optimally, i.e. this score is uniquely defined as the maximum score that Maker can ensure against all the possible strategies for Breaker. In particular, $Ms(G, \emptyset, \emptyset)$ denotes Maker's maximal score when Maker starts the game on G, and $Bs(G, \emptyset, \emptyset)$ when Breaker starts. If there is no possible confusion, they will be denoted by $Ms(G)$ and $Bs(G)$ respectively.

When the identity of the starting player is irrelevant, we denote the score by $s(G)$, omitting the prefix M or B. This allows us to state results that are valid for both Ms and Bs. However, it is important to note that this notation **does not** imply that $Ms = Bs$; it merely indicates that the statement applies to both cases independently.

We denote by SHVG the Scoring Happy Vertex Game, and we identify it with the decision problem that takes as input a graph G and an integer s and returns `true` if and only if Maker has a strategy to guarantee a score of at least s on G. Formally, the problem is defined as follows:

▶ **Problem 1** (Scoring Happy Vertex Game).
Input: *A graph G, an integer s, a first player X with $X \in \{M, B\}$.*
Question: *Do we have $Xs(G) \geq s$?*

As usual in positional game, if the first player is not specified, we suppose that it is Maker.

A current *game state* is represented by the triple (G, M, B), where $M, B \subset V$ are the sets of vertices colored by Maker (Maker) and Breaker (Breaker), respectively. We also denote by $V_F(G) = V(G) \setminus (M \cup B)$ the set of *free vertices* in G.

2.2 Scoring game

In this section, we begin by introducing some basic results about the score function associated with the game. We start by recalling the inductive definition of the score in scoring games, using the standard notations of Larsson [19].

▶ **Definition 2.** *Let (G, M, B) be a game state. Let $V_F = V \setminus (M \cup B)$ be the set of uncolored vertices in this position. Then, the scores Ms and Bs are defined inductively as follows:*

$$\begin{cases} Ms(G, M, B) = \max\limits_{x \in V_F} Bs(G, M \cup \{x\}, B), \\ Bs(G, M, B) = \min\limits_{x \in V_F} Ms(G, M, B \cup \{x\}). \end{cases}$$

If all vertices have been colored, i.e., $M \cup B = V$, the score is evaluated by:

$$s(G, M, B) = |\{v \in V(G) \,|\, N[v] \subseteq M\}|$$

We also recall that Milnor's universe is the set of games that are *dicotic* and *nonzugzwang*, i.e., the set of games in which, if a player can move, his opponent also can, and in which the players have no interest in passing their turn. Being a scoring positional game, the happy vertex game belongs to Milnor's universe [3], which implies the following inequality:

▶ **Lemma 3.** *For any game state (G, M, B), we have:*

$$Bs(G, M, B) \leq Ms(G, M, B).$$

Within Milnor's universe, we define the union between two games G and H, and we note $G \cup H$, the game in which, at each turn, a player may choose to move either in G or in H. Considering positional games played on graphs, this consists in considering that playing on a disconnected graph consists in playing on the sum of its connected components.

We also define the notion of equivalence as follows:

▶ **Definition 4.** *Two scoring games G_1 and G_2 are equivalent, and we note $G_1 = G_2$, if for every game G, the scores satisfy*

$$Ms(G \cup G_1) = Ms(G \cup G_2) \quad \text{and} \quad Bs(G \cup G_1) = Bs(G \cup G_2).$$

Moreover, the group structure in Milnor's universe makes it possible to bound the score on a union of games knowing the score of its terms. Namely, we have:

▶ **Theorem 5** (Milnor [21])**.** *Let (G_1, M_1, B_1) and (G_2, M_2, B_2) be two positions in the Happy vertex game. We have:*

$$\begin{aligned} &Bs(G_1, M_1, B_1) + Bs(G_2, M_2, B_2) \leq Bs(G_1 \cup G_2, M_1 \cup M_2, B_1 \cup B_2) \\ &\leq \min\left(Ms(G_1, M_1, B_1) + Bs(G_2, M_2, B_2), Bs(G_1, M_1, B_1) + Ms(G_2, M_2, B_2)\right) \end{aligned}$$

$$\begin{aligned} &\max\left(Ms(G_1, M_1, B_1) + Bs(G_2, M_2, B_2), Bs(G_1, M_1, B_1) + Ms(G_2, M_2, B_2)\right) \leq \\ &\qquad Ms(G_1 \cup G_2, M_1 \cup M_2, B_1 \cup B_2) \leq Ms(G_1, M_1, B_1) + Ms(G_2, M_2, B_2) \end{aligned}$$

In particular, if $Ms(G_1) = Bs(G_1)$, since $Ms \geq Bs$, the left and the right parts of the inequalities are equal, we have the following result:

▶ **Theorem 6** (Milnor 1953 [21])**.** *Let (G, M, B) be a game position such that $Ms(G, M, B) = Bs(G, M, B) = s$. Then, we have for any position (G', M', B'):*

$$s(G' \cup G, M' \cup M, B' \cup B) = s(G', M', B') + s$$

This result takes place in the more general framework of *temperature theory*, by stating that if a game G satisfies $Ms(G) = Bs(G)$, then its temperature is 0. However, the framework of temperature theory will not be required throughout this article. We refer the reader to Hanner's results [16] and to an article by Duchêne, Oijid, and Parreau [10] that summarizes these results for a general overview of this framework.

2.3 Results inherited from the Maker-Breaker domination game

The Maker-Breaker domination game can result in three different outcomes: $\mathcal{D}$, $\mathcal{S}$, and $\mathcal{N}$. The outcome $\mathcal{D}$ indicates that Dominator wins going second. The outcome $\mathcal{S}$ means that Staller wins going second. Finally, the outcome $\mathcal{N}$ means that the next (or first if no move has been played yet) player to move has a winning strategy. Note that, since in a Maker-Breaker positional game, it is always better to go first rather than second, these are the only three possible outcomes. There is a direct connection between the outcomes of the scores:

▶ Remark 7. Let G be a graph.

- The Domination Game on G results in $\mathcal{D}$ if and only if $s(G) = 0$.
- The Domination Game on G results in $\mathcal{N}$ if and only if $Bs(G) = 0$ and $Ms(G) \geq 1$.
- The Domination Game on G results in $\mathcal{S}$ if and only if $Bs(G) \geq 1$ and $Ms(G) \geq 1$.

Note that this only makes it possible to differentiate 0 scores from positive scores. For instance, if G is a star with at least two leaves, the outcome of the Maker-Breaker domination game on G is $\mathcal{N}$, but the score when Maker starts in the SHVG depends on the number of leaves.

In particular, if the outcome of a game is $\mathcal{D}$ in the Maker-Breaker domination game, it means that Breaker can ensure a score of 0 in the Happy vertex game, going either first or second. Thus, using Theorem 6, we have the following corollary:

▶ **Corollary 8.** *Let G be a graph. Then, the outcome of the Domination game on G is $\mathcal{D}$ if and only if $Ms(G) = 0$.*

A classical tool to prove that the outcome of the domination game is $\mathcal{D}$, is the *pairing strategy*. This general notion exists in all Maker-Breaker positional games, and can be expressed as follows in the Maker-Breaker domination game: if $G = (V, E)$ is a graph, a set of disjoint pairs of vertices $(x_1, y_1), \ldots, (x_k, y_k) \subset V^2$ is called a *pairing dominating set* if $V \subset \bigcup_{1 \leq i \leq k} N[x_i] \cap N[y_i]$. Duchêne *et al.* [8] proved that if G has a pairing dominating set, then its outcome is $\mathcal{D}$. In our score-based framework, this leads to the following:

▶ Remark 9. Let G be a graph. If G has a pairing dominating set, then $G = 0$.

In particular, this means that when considering disconnected graphs, removing a connected component that contains a pairing dominating set does not change the score on the game.

2.4 Simplifying instances

In this section, we aim to identify structures in a graph that can be reduced to other structures that are simpler to handle. In particular, when some vertices are already colored, we are looking for graph operations that will transform our graph into a simpler one without altering the score on the current position.

▶ **Definition 10.** *Let (G, M, B) define a position in the game on the graph G. We define the* decomposed graph *at this position, denoted by $[G]_B$, as the subgraph of G induced by the vertices $M \cup V_F$, augmented as follows: for every vertex u in the subgraph induced by $M \cup V_F$ that is adjacent to at least one vertex $v \in B$ in G, we add a new vertex u_B and an edge $\{u, u_B\}$. We denote by $[B]_B$ the set of all the vertices u_B added this way and $[M]_B = M$. This decomposition is depicted in Figure 1.*

(a) A position (G, M, B).

(b) The reduced position $([G]_B, [M]_B, [B]_B)$.

Figure 1 The reduction in Definition 10.

This transformation consists in identifying the vertices dominated by Breaker, i.e. that cannot be made happy any longer, making them dominated only by a leaf, by removing all the remaining vertices claimed by Breaker.

▶ **Lemma 11.** *Let (G, M, B) define a position in the game on the graph G. Then, $s(G, M, B) = s([G]_B, [M]_B, [B]_B)$.*

Proof. Suppose Maker has a strategy that ensures a score s_0 in G. She can apply her strategy in $([G]_B, [M]_B, [B]_B)$, which is possible since the vertices of $V_F([G]_B)$ are also free in G. At the end of the game, a vertex u is dominated in G if and only if it has a neighbor v colored by Breaker. If this neighbor v is in B, then u also has a neighbor colored by Breaker in $[G]_B$. Otherwise, v belongs to $[G]_B$ and, as the strategy in $G[B]$ mimics the one in G, v is still colored by Breaker in $[G]_B$. Consequently, any happy vertex in G is also happy in $[G]_B$, ensuring that $s([G]_B) \geq s(G)$. Similarly, by applying his strategy in G to $[G]_B$, Breaker can ensure that $s([G]_B) \leq s(G)$, which concludes the proof. ◀

We now focus on properties that can help us computing the score, by limiting the number of moves to consider: When considering scoring game, it happens that some moves will always have a higher impact on the score than others. It then seems natural to think that these moves will be played first. Note that a similar Lemma is used in [3], but the structure of the winning sets being more complex in the Maker-Breaker happy vertex game, it's stated differently. We first introduce a function h that takes a position $P = (G, M, B)$ and a free vertex $v \in V \setminus (M \cup B)$ and outputs the number of points scored instantly by playing v. Formally, we have:

$$h(P, v) = |\{w \in G \mid N[w] \subseteq M \cup \{v\}\}|$$

If there is no ambiguity on P, we will simply denote it by $h(v)$. The next lemma says that if playing v is worth more points than claiming u, even if all the neighbors of u are made happy, then, there exists an optimal strategy in which v is played before u.

▶ **Lemma 12.** *Let (G, M, B) be a position of the game. Let u, v be two vertices such that:*

$$h(v) \geq |N[u] \setminus B|$$

Then, there is an optimal strategy in which v is played before u.

Proof. We give the proof for Maker playing u before v, a similar proof provides the result if it is Breaker.

Up to consider a position of the game that is reached later in the strategy, let $P = (G, M, B)$ be a position and suppose Maker has a strategy $\mathcal{S}$ that scores s_0 in which the next move of Maker is to color u. Let $P' = (G, M \cup \{u\}, B)$. We propose the following strategy for Maker:

- Maker plays v.
- If Breaker colors a vertex $w \neq u$, Maker considers that he has played this vertex in P'. If the answer of Maker in P' is a vertex $w' \neq v$, Maker answers by playing w' in P. If Maker's answer in P' is to color v, then Maker colors u instead.
- If Breaker colors u, Maker considers that he has played v in P' and continue her strategy by playing the move that she would have played according to $\mathcal{S}$ (inverting the plays on u and v).

Following this strategy, when all the vertices are colored, either Maker has colored both u and v. In this case, the set of colored vertices by Maker and Breaker at the end of the game is exactly the same, so Maker scores s_0 by coloring v first. Otherwise, the set of colored vertices is the same, except that Maker has colored v instead of u, and Breaker has colored u instead of v. Denote by $P_u = (G, M_u, B_u)$ the resulting position if Maker has played according to $\mathcal{S}$ and $P_v = (G, M_v, B_v)$ the resulting position of the described strategy above. The score in P_v is then:

$$\begin{aligned} s(P_v) &= s_0 + |\{w \in N[v] | N[w] \subset M_v\} - |\{w \in N(u) | N[w] \subset M_u\}| \\ &\geq s_0 + |\{w \in N[v] | N[w] \subset M \cup \{v\}\} - |\{w \in N(u) | N[u] \cap B = \emptyset\}| \\ &\geq s_0 + h(v) - |N[u] \setminus B| \\ &\geq s_0 \end{aligned}$$

where the second inequality is obtained as we have $M \cup \{v\} \subset M_v$ and $M_u \subset V \setminus B$.

This proves that Maker can ensure a score at least s_0 by playing v before u. ◀

The so-called *Super Lemma* is a result that was introduced simultaneously in the PhD thesis of Nacim Oijid [22] and in the study of the game Incidence [3]. In this section, we adapt it to the *Scoring Happy Vertex Game*, thereby extending its applicability and providing a crucial tool for analyzing the strategic structure of the game. In the context of our game, this lemma says that if two vertices have the same neighborhood, among the uncolored vertices, and scores the same number of points among the colored vertices, there exists an optimal strategy in which the two players Maker and Breaker will both color one of these two vertices.

▶ **Theorem 13.** *Let (G, M, B) be a position of the game on a graph G, and let $u, v \in V_F$ be two distinct vertices such that for any $X \subset V_F \setminus \{u, v\}$, we have*

$$|\{w \in V | N[w] \subset X \cup \{u\} \cup M\}| = |\{w | w \in V, N[w] \subset X \cup \{v\} \cup M\}|$$

then

$$s(G, M, B) = s(G, M \cup \{u\}, B \cup \{v\}).$$

Proof. The proof is made by induction on $|V_F|$

If ($|V_F| = 2$), we have $V_F = \{u, v\}$. For $X = \emptyset$, the hypothesis implies:

$$|\{w \in V | N[w] \subset \{u\} \cup M\}| = |\{w | w \in V, N[w] \subset \{v\} \cup M\}|$$

Therefore, after Maker and Breaker's turns, the score is the same regardless of the choice of the vertex colored by the next player. Hence,

$$s(G, M, B) = s(G, M \cup \{u\}, B \cup \{v\}).$$

Suppose now $|V_F| \geq 3$. Assume first that Maker has a strategy $\mathcal{S}$ to ensure a score s_0 in $(G, M \cup \{u\}, B \cup \{v\})$, and consider the following strategy in (G, M, B):

- If it's Maker's turn, she colors the vertex she would have color in $(G, M \cup \{u\}, B \cup \{v\})$, following $\mathcal{S}$.
- If Breaker colors a vertex in $\{u, v\}$, Maker colors the second vertex of $\{u, v\}$, which leads to the same score as in G' by hypothesis on $\{u, v\}$.
- Otherwise, If Breaker colors $w \notin \{u, v\}$ Maker colors the vertex w' that she is supposed to color according to $\mathcal{S}$, considering that the last move of Breaker in $(G, M \cup \{u\}, B \cup \{v\})$ was w. Note that this move is always available and cannot be a vertex {u,v}, unless $|V_F| = 3$, in which case, Breaker would also color w in $(G, M \cup \{u\}, B \cup \{v\})$.

Following this strategy, if the result is not obtained directly, the resulting position is $(G, M \cup \{w'\}, B \cup \{w\})$ which has two less free vertices. Hence, by induction, it has the same score as $(G, M \cup \{u, w'\}, B \cup \{v, w\})$, and since the moves w' was supposed to be optimal answer to w in $(G, M \cup \{u\}, B \cup \{v\})$, it has a score equal or higher than $(G, M \cup \{u\}, B \cup \{v\})$, which proves that Maker can ensure s_0 in (G, M, B).

A similar argument applied to Breaker shows that if Breaker can ensure a score of at most s_0 in $(G, M \cup \{u\}, B \cup \{v\})$, he can ensure at most s_0 in (G, M, B), which concludes the proof. ◀

Similarly to the provided result in [3], this lemma enables us to directly compute the score on *complete binary trees*, i.e. trees defined by T_0 a single vertex, which is a root, and T_i is obtained by taking two copies of T_{i-1}, and adding a new root that will be connected to the previous roots of the copies of T_{i-1}. The integer i is then called the *depth* of T_i.

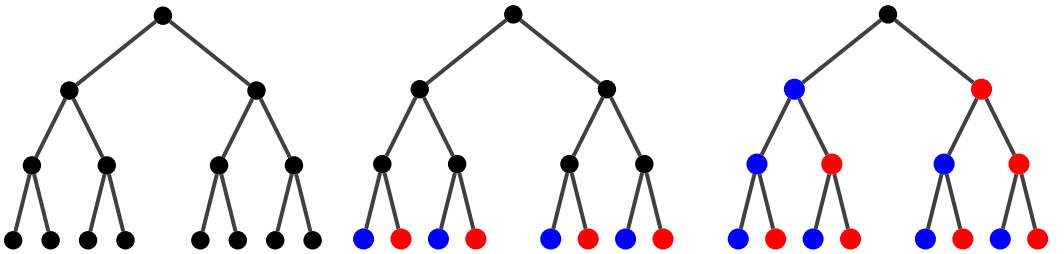
▶ **Theorem 14.** *Let T_d be the complete binary tree of depth d. We have $Ms(T_d) = 1$ and $Bs(T_d) = 0$ if $d \leq 1$, and $Ms(T_d) = Bs(T_d) = 2^{d-2}$ otherwise.*

Proof. If $d = 0$, T_d is an isolated vertex, and Maker scores one by coloring it, otherwise the score is 0. If $d = 1$, the graph is P_3, the path on 3 vertices. We can apply Theorem 13 to its two leaves, and Maker scores 1 if she starts by coloring the middle vertex; otherwise, Breaker colors it and ensures that no vertex is happy. If $d \geq 2$, each pair of leaves attached to the same vertex are twins, thus, we can apply Theorem 13 to them. Hence, Maker has colored half of them, i.e. 2^{d-1} of them. Now their parents satisfy the hypothesis of Theorem 13, as they are in the winning set of the connected leaf colored by Maker and the winning set of their common parent (which exists as $d \geq 2$). Thus, Maker will color half of them, ensuring that half of the leaves she has colored are happy, and thus scoring 2^{d-2}. We can then repeat this process until reaching the vertices of depth 1, but now, all the colored vertices by Maker already have one neighbor colored by Breaker, and thus will not be happy. Finally, regardless of whether Maker or Breaker colors the root, it has one neighbor colored by Breaker and thus will not be happy. Finally, the score is 2^{d-2}. This process is depicted in Figure 2. ◀

3 Hardness results

In this section, we concern ourselves with the hardness of the SHVG problem. Note that, since the Maker-Breaker domination game is already PSPACE-complete on bounded degree bipartite graphs, or split graphs [8, 23], we can inherit this result for $s = 1$. Namely, we have the following theorem:

▶ **Theorem 15.** *SHVG is* PSPACE*-complete even restricted to $s = 1$, and to bounded degree bipartite graphs or split graphs.*



(a) A complete binary tree. **(b)** First application of Theorem 13. **(c)** Last application of Theorem 13.

Figure 2 Computing the score on T_3, the complete binary tree of depth 3.

However, adding scores to the Maker-Breaker domination game makes it hard on smaller graph classes, and we prove that SHVG is already PSPACE-complete when restricted to trees, and that the problem is NP-hard when restricted to caterpillars.

Both hardness results will rely on reductions from two specific Quantified MAX-SAT problems, which we introduce and prove to be PSPACE-complete and at least NP-hard respectively in the next subsection. We postpone the hardness results on SHVG to Section 3.2 and Section 3.3.

3.1 Quantified MAX SAT restrictions

In this subsection, we will introduce two restrictions of the (Quantified) MAX-2-SAT problem which we will use in order to obtain our hardness results. These problems are very general, and are an important contribution of this paper since they can be used to provide other reduction, in particular, providing a natural PSPACE-complete problem on forests. Recall first the definition of Quantified MAX 2-SAT, which is a quantified version of Max 2-SAT.

▶ **Problem 16** (Quantified Max 2-SAT).
Input: (ψ, k) *where* ψ *is a QBF formula of the form* $\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n\ \varphi$ *with, for* $1 \leq i \leq n$, $Q_i \in \{\exists, \forall\}$, φ *a quantifier free* 2*-CNF formula, and* k *is an integer.*
Question: *Is it possible to satisfy simultaneously* k *clauses of* φ*?*

As usual dealing with positional games, we will consider the equivalent gaming variant of the problem [26, 3]: Two players Satisfier and Falsifier choose the value of the x_is, Satisfier chooses the existentially quantified variables, and Falsifier chooses the universally quantified variables. Satisfier wins if she manages to satisfy k clauses, otherwise Falsifier wins.

Quantified MAX 2-SAT is PSPACE-complete [3]. We will prove that this remains the case even when we impose structural properties on the formula ψ.

We now define the main tool that will be used in our reductions. Let φ be a quantifier-free 2-CNF formula, with variables $\{x_1, \dots, x_n\}$. Let G_φ be the graph in which each variable x_i is represented by two vertices, denoted x_i^+ and x_i^-, corresponding to its positive and negative literals respectively, and where for each clause $l_i \vee l_j$ in φ, there is an edge between the vertices corresponding to the literals l_i and l_j. We will call G_φ the *literal–clause incidence graph* of φ. An example of construction of this graph is provided in Figure 3. For readers used to expressing boolean formulas in graphs, note that this graph is **not** the usual incidence graph of a formula.

Let us now introduce our first specific Quantified MAX-2-SAT problem.

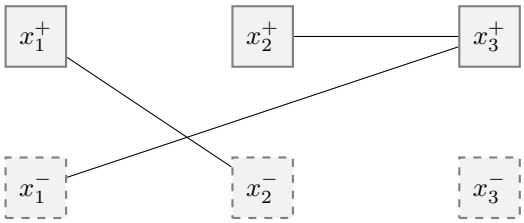


Figure 3 The literal-clause incidence graph G_φ with $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee x_3)$.

▶ **Problem 17** (Acyclic Q-MAX 2-SAT).
Input: *A QBF formula of the form* $\psi = Q_1 x_1 Q_2 x_2 \dots Q_n x_n\ \varphi$ *with* φ *a quantifier-free* 2*-CNF formula whose literal–clause incidence graph is acyclic, an integer* k*.*
Question: *Is it possible to satisfy simultaneously* k *clauses of* ψ*?*

▶ **Lemma 18.** *Acyclic Q-MAX 2-SAT is* PSPACE*-complete.*

Proof. Membership in PSPACE follows from the PSPACE-completeness of Quantified MAX 2-SAT [3]. To prove the hardness, we reduce from Quantified MAX 2-SAT without the acyclicity restriction. Let (ψ, k) be an instance of Quantified MAX 2-SAT, with $\psi = Qx_1 Qx_2 \dots Qx_n\ \varphi$ where φ is a quantifier-free formula. Let $(l_1 \vee l_2) \wedge (l_2 \vee l_3) \wedge \dots \wedge (l_m \vee l_1)$ be a cycle in φ. Without loss of generality, assume that $l_1 = x$, the case $l_1 = \neg x$ being similar. We then introduce three new existential quantifiers $\exists a \exists b \exists x'$ and the following gadget clauses:

$$(x \vee a), \quad (\neg x \vee b), \quad (\neg a \vee \neg x'), \quad (\neg b \vee x').$$

We consider the formula $\psi' = Qx_1 Qx_2 \dots Qx_n\ \exists x' \exists a \exists b\ \varphi'$ where φ' is obtained from φ by substituting $(x' \vee l_2)$ to $(x \vee l_2)$, and by adding the clauses $(x \vee a), (\neg x \vee b), (\neg a \vee \neg x'), (\neg b \vee x')$. Let $k' = k + 4$.

The intuition behind this construction is that if there is an assignment of the variables that satisfy k clauses in ψ, the same assignment putting $x' = x$, and choosing the value of a and b such that $a = b = \neg x$ will satisfy $k + 4$ clauses in ψ'.

Suppose first that Satisfier has a winning strategy in (ψ, k), by applying the same strategy in (ψ', k'), and by setting $x' = x$, she ensures that at least $k + 2$ clauses are satisfied when x' is played, as exactly one of $(x \vee a), (\neg a \vee \neg x')$ and exactly one of $(\neg x \vee b), (\neg b \vee x')$. Then, by putting $a = b = \neg x$, she ensures to satisfy the two remaining clauses, and that at least $k + 4$ clauses are satisfied.

Reciprocally, suppose that Falsifier has a winning strategy in (ψ, k). By applying the same strategy in (ψ', k'), he ensures that at most $k - 1$ clauses are satisfied when x_n is played. Then, if Satisfier puts $x = x'$, the only clauses that can be satisfied in ψ' that are not in ψ are the 4 added clauses, hence, Falsifier ensures that at most $k + 3 < k'$ are satisfied in total. If Satisfier sets $x' = \neg x$, the only clause that can be satisfied that was in ψ through this move is the clause $x' \vee l_2$, since the other clauses do not contain x'. Hence, at most k clauses are satisfied outside $\{(x \vee a), (\neg x \vee b), (\neg a \vee \neg x'), (\neg b \vee x')\}$. But, either x and $\neg x'$ are True, then, at most one of $\{(\neg x \vee b), (\neg b \vee x')\}$ can be satisfied, or x and $\neg x'$ are False, and then at most one of $\{(x \vee a), (\neg a \vee \neg x')\}$ can be satisfied. In both cases, at most three of these clauses can be satisfied, and thus, at most $k + 3 < k'$ in total, ensuring that Falsifier wins in (ψ', k'). Hence, Satisfier wins in (ψ, k) if and only if she wins in (ψ', k')

Finally, in the constructed formula ψ', the literal–clause incidence graph has at least one less cycle than the one of ψ since the variables x and x' are separated by the variables a, $\neg a$ and b, $\neg b$.

By iterating this construction we can eliminate all cycles. Note that each time this reduction is applied, the number of clauses that can be in a cycle decreases by one, since none of the added clauses can be in a cycle. Thus, the number of time this argument has to be applied is bounded by the number of clause of ψ, ensuring that this reduction is polynomial. Therefore, Acyclic Q-MAX 2-SAT is PSPACE-complete. ◀

Then, we can define our second more specific problem, where we restrict the number of times a variable can occur in a clause.

▶ **Problem 19** (Acyclic MAX 2-SAT-2-2)**.**
Input: *A 2-CNF formula φ such that*
- *the literal–clause incidence graph G_φ is acyclic,*
- *each variable occurs at most twice positively and at most twice negatively,*

an integer k.
Question: *Is it possible to satisfy simultaneously k clauses of φ?*

This problem consists in a quantified version of bounded occurrence MAX 2-SAT - but where we furthermore impose acyclicity on the literal-clause incidence graph. First, we recall that MAX 2-SAT-3, the version of MAX 2-SAT where the number of occurrences of each variable is bounded by 3 is NP-complete [5]. More formally, this means that the following problem is NP-complete:

▶ **Problem 20** (MAX 2-SAT-3)**.**
Input: *A 2-CNF formula φ such that each variable occurs at most thrice, an integer k.*
Question: *Is it possible to satisfy simultaneously k clauses of φ?*

We now prove the NP-completeness of Acyclic MAX 2-SAT-2-2 by a reduction from MAX 2-SAT-3.

▶ **Lemma 21.** *Acyclic MAX 2-SAT-2-2 is NP-complete.*

Proof. First, note that Acyclic MAX 2-SAT-2-2 is in NP as it is an instance of Max 2-SAT.

We reduce from the NP-complete problem MAX 2-SAT-3 [5]. Let φ be a 2-CNF formula such that each variable occurs at most thrice and k an integer. If a variable in φ appears more than twice positively (resp. negatively) then it must appears exactly thrice positively (resp. negatively). Then we can satisfy all three clauses by assigning the variable to `True` (resp. `False`) and no clause is satisfied by changing the truth value of the variable from `True` to `False` (resp. False to `True`), i.e. we can always satisfy these three clauses without incidence on the other clauses. Thus we can simply remove these three clauses from φ and lower k by 3. By iterating this process we construct a new formula where variables not only occur at most thrice, but also occur at most twice positively and at most twice negatively.

Furthermore, if a variable appears in a cycle in the literal-clause graph, we can apply the gadget from the previous reduction to effectively remove cycles. However, this gadget introduces a new variable x'_i for each x_i, and requires using one more occurrence of x_i if x_i appears negatively in the cycle or one more of $\neg x_i$ if it appears positively in the cycle. However, this transformation cannot make a variable appear a third time positively or negatively, since if x_i ($\neg x_i$ resp.) appears in the cycle, it means that it appears exactly twice positively (negatively resp.) and thus at most once negatively (positively resp.). Note also that this transformation ensures that all the occurences of x_i and $\neg x_i$ are either of degree 1, or neighbors of a vertex of degree 1, ensuring that they cannot appear in cycle, and

thus this reduction has to be applied at most once per variable. Hence, this transformation does not make any variable appear more than twice positively or negatively. This entire transformation can be performed in polynomial time, therefore we conclude that the problem is NP-complete. ◀

A direct consequence of this result is that, by quantifying all the vertices with ∃, it's quantified version is NP-hard and in PSPACE.

▶ **Problem 22** (Acyclic QMAX 2-SAT-2-2).
Input: *A quantified 2-CNF formula $\psi = Q_1x_1 \dots Q_nx_n\varphi$ such that*
- *the variable–clause incidence graph G_φ is acyclic,*
- *each variable occurs at most twice positively and at most twice negatively,*

an integer k.
Question: *Can Satisfier satisfy simultaneously k clauses of ψ?*

▶ **Corollary 23.** *Acyclic QMAX 2-SAT-2-2 is* NP*-hard and in* PSPACE.

Proof. The NP-hardness is directly inherited from Acyclic MAX 2-SAT-2-2 by quantifying all the variables with ∃. The problem is in PSPACE as a subproblem of QMAX 2-SAT. ◀

These new variants of Q-MAX 2-SAT are quite general and can be used in several reductions. In particular, the reduction from Q-MAX 2-SAT provided in [3], proving that Incidence, the scoring positional game on graphs is PSPACE-complete starts from the literal-clause incidence graph of a formula and only adds isolated vertices and leaves to it. Hence, we almost answer an open problem of the paper, that asks the complexity of the game on trees with the following corollary:

▶ **Corollary 24.** *Maker-Breaker Incidence is* PSPACE*-complete, even restricted to forests, and* NP*-hard, even restricted to unions of caterpillars.*

Proof. We perform the same reduction as in [3], except that the input problem is Acyclic QMAX 2-SAT (or Acyclic QMAX 2-SAT-2-2). Since this constructions consists in adding leaves to the literal-clause incidence graph of the formula, it produces either a forest or a caterpillar, depending on the problem from which we do the reduction. ◀

3.2 Trees

We now have all the necessary tools to establish the PSPACE-completeness of SHVG when restricted to trees. The proof relies on a reduction from Acyclic Q-MAX 2-SAT which we proved to be PSPACE-hard. Since SHVG belongs to PSPACE as it is a scoring positional game [3], the result follows.

▶ **Theorem 25.** *SHVG on trees is* PSPACE*-complete.*

Proof. We provide a reduction from Acyclic Q-MAX 2-SAT. Let (ψ, k) be an instance of Acyclic Q-MAX 2-SAT with $\psi = Q_1x_1Q_2x_2 \dots Q_nx_n\ \varphi$ with φ an acyclic and quantifier-free 2-CNF formula. Up to add a variable that appear in no clauses, for each pair of consecutive quantifiers of same type, we can suppose that $Q_i = \exists$ if i is odd, and $Q_i = \forall$ if i is even. Note that this addition of dummy variables do not break the acyclicity of the literal-clause incidence graph, as it only adds isolated vertices. The first step of our reduction is to duplicate our formula: we transform it into $\psi' = Q_1x_1Q_2x_2 \dots Q_nx_nQ_{n+1}x_{n+1}Q_{n+2}x_{n+2} \dots Q_{2n}x_{2n}\ \varphi \wedge \varphi'$, where φ' is obtained from φ by changing each x_i to x_{n+i}. Note that any strategy satisfying p clauses can be applied twice to satisfy $2p$ clauses in ψ'. Let m be the number of clauses of φ

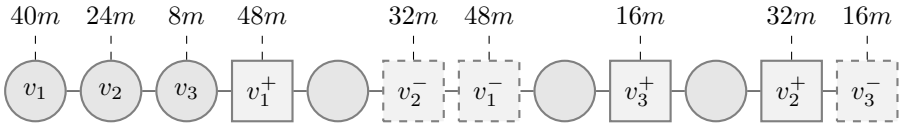


Figure 4 The resulting tree obtained from the formula $\exists x_1 \forall x_2 \exists x_3\ (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (x_3 \vee x_2)$.

We will progressively construct a tree by making modifications to the literal-clause incidence graph $G_{\varphi \wedge \varphi'}$ of $\varphi \wedge \varphi'$, in which we denote by v_i^+ (v_i^- resp.) the vertex corresponding to literal x_i ($\neg x_i$ resp.).

Since the graph is acyclic by hypothesis, it must be a disjoint union of trees $T_1, T_2, \dots T_\ell$. We now construct a tree T based on this graph. We start from $T_1 \cup T_2 \cup \dots \cup T_\ell$. For every tree T_i let us fix two of its leaves v_i^ℓ and u_i^ℓ and let us connect all trees into a single one by adding edges between u_i^ℓ and v_{i+1}^ℓ for $1 \leq i < \ell$. We denote by $\sim$ either $+$ or $-$. For each edge $\{v_i^\sim, v_j^\sim\}$, we introduce a new vertex $v_{ij}^{\sim\sim}$ and replace the edge with two edges: $\{v_i^\sim, v_{ij}^{\sim\sim}\}$ and $\{v_{ij}^{\sim\sim}, v_j^\sim\}$.

Next, to each vertex $v_i^\sim$, we attach $16(n+1-i)m$ leaves. Finally, we introduce n new vertices v_i, attach $16(n+1-i)m - 4m$ leaves to each of them, add we connect them through a path whose extremity v_n is attached to v_1^+.

An example of reduction is provided in Figure 4.

Let s_0 be defined as follows:

$$s_0 = \left(\sum_{1 \leq i \leq 2n} 8(n+1-i)m \right) + \left(\sum_{1 \leq i \leq n} 8\big(n+1-(2i-1)\big)m - 4m \right)$$

Let $s = s_0 + m - k + 1$. Both T and s can be constructed in polynomial time. We prove that Falsifier wins in (ψ, k) if and only if Maker wins in (T, s).

Firstly, each leaf can be colored using Theorem 13. Denote by M and B the vertices colored during this step. Then, if $1 \leq i \leq j \leq 2n$, we have $h(v_i^\sim) \geq |N(v_i) \setminus B|$ and $h(v_i) \geq |N(v_j^\sim) \setminus B|$. Hence, using Lemma 12, there exists an optimal strategy in which Maker and Breaker color the vertices in the following order, for $1 \leq i \leq n$, in increasing order:

- Maker colors one of $\{v_{2i-1}^+, v_{2i-1}^-\}$.
- Breaker colors the other vertex in $\{v_{2i-1}^+, v_{2i-1}^-\}$.
- Maker colors v_{2i-1}.
- Breaker colors one of $\{v_{2i}^+, v_{2i}^-\}$.
- Maker colors the other vertex in $\{v_{2i}^+, v_{2i}^-\}$.
- Breaker colors v_{2i}.

Considering this order, which is optimal, note that all the moves are forced or have to be chosen among two vertices (v_i^+, v_i^-) that are connected to the same number of leaves. Moreover, no vertex $v_i^\sim$ nor v_i can be happy, because of the leaves colored by Breaker connected to them. Hence, we know that after this step, Maker has colored exactly s_0 happy vertices.

Finally, once all the previously described vertices have been colored, only the vertices $v_{ij}^{\sim\sim}$ remain. These vertices will be colored last, and each of them can be happy if and only if both of their adjacent vertices have been colored by Maker.

Suppose first that Falsifier has a winning strategy $\mathcal{S}$ in (ψ, k). We define the following strategy for Maker:

- For $0 \leq i \leq n-1$, if Falsifier has to put x_{2i+1} to True, Maker colors v_i^-, ensuring that Breaker will color v_i^+; otherwise, she colors v_i^+, ensuring that Breaker colors v_i^-.
- For $1 \leq i \leq n$, if Breaker colors v_{2i}^+, Maker considers that Satisfier has put x_{2i} to True in $\mathcal{S}$. Otherwise, she considers that Satisfier has put x_{2i} to False.
- When x_n is played, Maker applies $\mathcal{S}$ again from scratch.

When all the vertices v_i, v_i^+ and v_i^- have been played, if C_j is a clause that has not been satisfied in $\varphi \wedge \varphi'$, then its two literals has been put to False. Thus, since $\mathcal{S}$ was a winning strategy for Satisfier in (ψ, k), we know that at most $k-1$ clauses of φ and at most $k-1$ clauses of φ' are satisfied. Thus, at least $2\big(m-(k-1)\big)$ vertices $v_{ij}^{\sim\sim}$ has their two neighbors colored by Maker. Hence, Maker can by playing only on them ensure to color at least half of them, i.e. $m-k+1$, and having colored $s_0+m-k+1=s$ happy vertices in total.

Reciprocally, suppose that Satisfier has a winning strategy $\mathcal{S}$ in (ψ, k). We define the following strategy for Breaker:

- For $0 \leq i \leq n-1$, if Satisfier has to put x_{2i+1} to True, Breaker colors v_i^+; otherwise, he colors v_i^-.
- For $1 \leq i \leq n$, if Maker colors v_{2i}^+, Breaker colors v^-2i and considers that Satisfier has put x_{2i} to False in $\mathcal{S}$. Otherwise, he considers that Satisfier has put x_{2i} to True.
- When x_n is played, Breaker applies $\mathcal{S}$ again from scratch.

Now, when all the vertices, v_i, v_i^+ and v_i^- have been played, the same argument as above ensures that any clause that was satisfied by $\mathcal{S}$ has one of its literal played by Breaker. Hence, since $\mathcal{S}$ was a winning strategy in ψ, at least $2k$ vertices $v_{ij}^{\sim\sim}$ are already dominated by Breaker, and thus, at most $2(m-k)$ of them are not. By pairing them, Breaker can ensure to claim at least $m-k$ of them ensuring that Maker scores at most $s_0+m-k<s$. Thus, Breaker wins. ◀

3.3 Caterpillars

We now consider SHVG restricted to caterpillars. A graph is called a *caterpillar* if removing all its leaves results in a path. In other words, a caterpillar consists of a central path with pendant vertices attached to some or all of its nodes.

▶ **Theorem 26.** *SHVG on caterpillars is* NP*-hard.*

Proof. The proof is identical to the one on trees, except that instead of using Acyclic Q-MAX 2-SAT, we now rely on Acyclic Q-MAX 2-SAT-2-2, which is proved to be NP-hard. In this case, the constructed graph G_φ is not only a disjoint union of trees, but in particular a disjoint union of paths, since each vertex in the graph would have degree at most 2. The same reduction carries through and produces a caterpillar by always connecting the different paths through one of their extremities. This shows that SHVG remains NP-hard on caterpillars. ◀

4 Efficient algorithms

Since the game is already PSPACE-complete on forests, and NP-hard on caterpillars, we aim to find efficient algorithms on other subclasses of trees. Namely, we focus here on union of paths and subdivided stars.

4.1 Union of paths

We first consider union of paths.

4.1.1 Single path

We begin by studying paths. A path P_n is defined by its set of vertices $\{v_1, \dots, v_n\}$ and its set of edges $\{(v_i, v_{i+1}) \mid 1 \leq i \leq n-1\}$.

▶ **Theorem 27.** *Let P_n be a path on n vertices. Then:*
- *If n is even, $Ms(P_n) = Bs(P_n) = 0$.*
- *If n is odd, then $Ms(P_n) = 1$ and $Bs(P_n) = 0$.*

Proof.
- Suppose first that n is even.
  Then, Dominator wins the Maker-Breaker domination game on P_n. Thus, by Corollary 8, we have $\mathrm{SHVG}(P_n) = 0$, and consequently, $Ms(P_n) = Bs(P_n) = 0$.
- Suppose now that n is odd, the outcome of the domination game on P_n is $\mathcal{N}$. So, $Bs(P_n) = 0$, and $Ms(P_n) \geq 1$ by Remark 7. To prove that $Ms(P_n) \leq 1$, set $n = 2k+1$, and denote by $(v_1, \dots, v_{2k+1})$ the vertices of P_n, and consider the following pairing strategy for Breaker: if Maker colors a vertex in $\{v_{2i-1}, v_{2i}\}$, for $1 \leq i \leq k$, Breaker colors the other one. This strategy ensures that no vertex in $v_1, \dots, v_{2k}$ will be happy and thus, the only vertex that Maker can make happy is v_{2k+1}, which ensures $Ms(P_n) \leq 1$. ◀

4.1.2 Union of paths

Once the score on a single path can be computed, union of paths can be handled using the group structure provided by Milnor's universe. In particular, if n is even, since, $Ms(P_n) = Bs(P_n) = 0$, $P_n = 0$, and thus for any graph G, we have $Ms(G + P_n) = Ms(G)$ and $Bs(G + P_n) = Bs(G)$.

▶ **Lemma 28.** *Let n, m be two odd integers, we have $Ms(P_n + P_m) = Bs(P_n + P_m) = 1$. Thus $P_n + P_m = 1$.*

Proof. Since $Ms \geq Bs$, it is sufficient to prove $Bs(P_n + P_m) \geq 1$ and $Ms(P_n + P_m) \leq 1$.

Suppose Breaker starts. Up to exchange n and m, suppose that his first move is a vertex v_0 in P_n. It's now Maker's turn, and thus Maker can ensure a score of 1 since $Ms(P_n + P_m, \emptyset, \{v_0\}) \geq Ms(P_m, \emptyset, \emptyset) + Bs(P_n, \emptyset, \{v_0\}) \geq Ms(P_m, \emptyset, \emptyset) = 1$ by Theorem 5.

Suppose now that Maker starts. Let $n = 2k+1$ and $m = 2l+1$. Denote by $u_1, \dots, u_{2k+1}$ the vertices of P_n and by $v_1, \dots, v_{2l+1}$ the vertices of P_m. Consider the following pairing for Breaker $\{(u_1, v_1)\} \cup \{(u_{2i}, u_{2i+1})\}_{1 \leq i \leq k} \cup \{(v_{2i}, v_{2i+1})\}_{1 \leq i \leq l}$. This strategy ensures that all the vertices u_i and v_i for $i \geq 2$ will not be happy, and that Maker will color at most one of u_1, v_1, which ensures that at most one of them will be happy. Thus, $Ms(P_n + P_m) \leq 1$. ◀

▶ **Corollary 29.** *Let $k_1, \dots, k_p$ be integers, with l of them being odd. We have $Ms(P_{k_1} + \dots + P_{k_p}) = \lceil \frac{l}{2} \rceil$ and $Bs(P_{k_1} + \dots + P_{k_p}) = \lfloor \frac{l}{2} \rfloor$*

▶ Remark 30. Since cycles have outcome $\mathcal{D}$ in the Maker-Breaker domination game, they can be added to any game without changing the score. Thus, Corollary 29 can be extended to graphs of maximum degree 2.

4.2 Subdivided Stars

Now, we focus on *subdivided stars*. These graphs are formed by connecting a finite number of paths to a central vertex. Subdivided stars represent an intermediate level of complexity between paths and more general tree structures, since they are trees with a single vertex of degree higher than 3.

▶ **Lemma 31.** *Let (G, M, B) be a position of the game. We have for any $v \in M$, $s(G, M, B) \geq s(G \setminus \{v\}, M \setminus \{v\}, B)$.*

Proof. Let $\mathcal{S}$ be a strategy for Maker in $s(G \setminus \{v\}, M \setminus \{v\}, B)$. By applying $\mathcal{S}$ in (G, M, B) which has the same set of uncolored vertices, any vertex made happy in (G, M, B) is also happy in $s(G \setminus \{v\}, M \setminus \{v\}, B)$, which proves the result. ◀

▶ **Theorem 32.** *Let S be a subdivided star obtained by connecting k paths $P_1, \ldots, P_k$ to a central vertex C. If $k \leq 2$, S is a path, and the score on S is given by Theorem 27. Otherwise, denote by l the number of odd paths attached to C. We have $Ms(S) = \lfloor \frac{l}{2} \rfloor$ and $Bs(S) = 0$, unless $l = 0$ in which case $Ms(S) = 1$.*

Proof. Let S be a subdivided star. First, since the Maker-Breaker domination game on S has outcome $\mathcal{N}$ [8], we have $Bs(S) = 0$.

Let C be its central vertex and $P_1, \ldots, P_k$ be the paths attached to it. Note that if $k \leq 2$, S is a path and its score is given by Theorem 27, i.e. its score is 1 if it has an odd number of vertices and Maker starts, and 0 otherwise.

Suppose now that Maker starts and $k > 2$. Let l be the number of odd paths in S and let $P_1, \ldots, P_l$ be these l odd-length paths. Suppose $l \geq 1$. If Maker colors the central vertex at the beginning, by coloring C, she can guarantee a score of at least $Bs(S, \{C\}, \emptyset)$. By Lemma 31, $Bs(S, \{C\}, \emptyset) \geq Bs(S \setminus C, \emptyset, \emptyset)$ and, since $S \setminus C$ is an union of paths, by Corollary 29, $Bs(S \setminus C, \emptyset, \emptyset) = \lfloor \frac{l}{2} \rfloor$. Hence $Ms(S) \geq \lfloor \frac{l}{2} \rfloor$.

Let $n_i = 2k_i + 1$ be the length of P_i and denote by $u_1^i, \ldots, u_{2k_i+1}^i$ the vertices of P_i for $1 \leq i \leq l$ such that $u_{2k_i+1}^i$ is connected to C. Breaker has a pairing strategy to prevent Maker from scoring inside the even-length paths by pairing adjacent vertices starting from their leaves. Moreover the pairing $\{(u_{2j-1}^1, u_{2j}^1)\}_{1 \leq j \leq k_1} \cup \{(u_{2k_1+1}^1, C)\}$ enables Breaker to prevent Maker from scoring inside $P_1 \cup \{C\}$. It results that the only vertices that can be made happy are in $(P_2 + \cdots + P_l)$. Hence, by applying the strategy provided in Corollary 29 in $(P_2 + \cdots + P_l)$ when Maker plays in it, and by applying his pairing strategy in $S \setminus \big(P_2 + \cdots + P_l\big)$, since $Ms(P_2 + \cdots + P_l) = \lceil \frac{l-1}{2} \rceil = \lfloor \frac{l}{2} \rfloor$, we conclude that $Ms(S) \leq \lfloor \frac{l}{2} \rfloor$. All together, we have $Ms(S) = \lfloor \frac{l}{2} \rfloor$

If $l = 0$ however, Maker can still guarantee a score of 1. We prove this result by induction. Since $k > 2$, there are at least two even paths P_1 and P_2. Let S_n be the subdivided star that is formed by connecting n paths $P_1, P_2, \ldots, P_n$ of length exactly 2 to a central vertex C. By coloring successively all the leaves' neighbors, Maker forces Breaker to answer by coloring each leaves, since otherwise by coloring one of the leaves and its neighbor Maker can score 1, which will conclude the proof. Once Maker colored all the leaves' neighbors and Breaker colored all the leaves, it is again Maker's turn. All vertices adjacent to the central vertex have been colored by Maker, thus by coloring it Maker scores 1. As a result, $Ms(S_n)$ is at least one. This sequence is illustrated in Figure 5 for $n = 3$.

If S is formed by connecting n paths $P_1, P_2, \ldots, P_n$ each of length $2k_i$ for $1 \leq i \leq n$ to a central vertex C, and assuming that P_1 has length $2k_1 > 2$. Then Maker colors the neighbor of the leaf of P_1, forcing Breaker to answer by playing on the leaf, or letting Maker score 1 in her next move. Then by Lemma 31 and Lemma 11, Maker can ensure on this position a score at least equal to that of S' formed by connecting n paths $P_1', P_2, \ldots, P_n$ to C where P_1' is the path of length $2k_1 - 2$, and, inductively, at least that of S_n.

For the other direction Breaker has a pairing strategy preventing any vertex besides the central vertex to be happy, by pairing the vertices of each path together, which is possible by again pairing adjacent vertices starting from the leaves. The only vertex that is not paired, and thus that can be happy and the end of the game is C, which proves $Ms(S) \leq 1$. All together, $Ms(S) = 1$. ◀

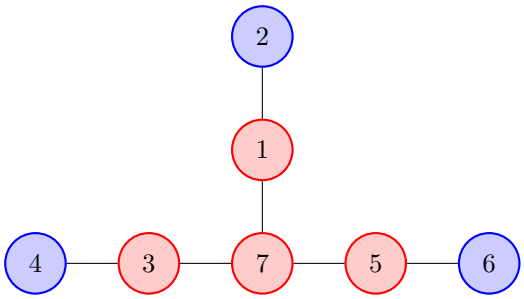


Figure 5 The strategy on a star with all branches of length 2. The number on the vertices is the order in which they are played.

4.3 An FPT algorithm for the neighborhood diversity

The parameterized complexity of positional games has recently grown a strong interest and is a promising way to obtain efficient algorithms for games that are usually hard to solve. Bonnet, Gaspers, Lambilliotte, Rümmele and Saffidine, have proved that Maker-Breaker positional games are W[1]-hard parameterized by the number of moves, but can be solved in FPT time for the same parameter in several graph classes [6]. However, these general results cannot be used when considering scores, and therefore do not apply directly in our framework. In the scoring positional games universe, the first parameter considered in [3] is the neighborhood diversity, which was introduced by Lampis [18], and measures the variety of the neighborhoods of the vertices in the graph. We propose an FPT-algorithm for this parameter, where w is the neighborhood diversity of the input graph. Note that, since the game is already NP-hard on caterpillars, one cannot hope, under classical complexity assumptions, for an FPT-algorithm for most of the general parameters, such as the pathwidth, the treewidth, the twinwidth, or the feedback edge number.

We first recall that two vertices, u and v *have the same type* if $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. If G is a graph, the neighborhood diversity of G, denoted $nd(G)$ is defined as the smallest integer w such that the vertices of G can be partitioned into k sets $V_1, \dots, V_w$ where all the vertices of each set have the same type. Such a partition is called a *neighborhood partition* of G. Note that it implies that each set V_i is either a clique or an independent set.

The main idea of the reduction is to use Theorem 13 to color all the vertices of same type but at most 1. However, contrary to the kernelization provided for the game Incidence [3], we cannot here easily remove the claimed vertices, as some of them may require several moves from Maker to contribute to the score. Hence, our FPT algorithm cannot provide us a polynomial kernel, and the question of its existence remains open.

▶ **Theorem 33.** *Let $G = (V, E)$ be a graph of neighborhood diversity w. The score in the scoring happy vertex game can be computed in* FPT *time when parameterized by the neighborhood diversity. In particular, $s(G)$ can be computed in time $\mathcal{O}(3^w(|E| + |V|))$.*

Proof. We first compute a neighborhood partition of V, which can be done in time $O(|E|+|V|)$ time [18].

Let u, v be two vertices having the same type. By definition, we have $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. Thus, for any position of the game (G, M, B) and any subset $X \subseteq V \setminus \{u, v\}$, we have:

$$|\{w \in V \mid N[w] \subseteq X \cup M \cup \{u\}\}| = |\{w \in V \mid N[w] \subseteq X \cup M \cup \{v\}\}|.$$

Therefore, Theorem 13 can be applied to each pair of vertices of the same type.

By applying this lemma to all the pairs of uncolored vertices of same type, the resulting position has at most one vertex of each type that remains uncolored and has the same score as the original instance. We can then explore all possible games on these vertices using dynamic programming. Since the number of uncolored vertices is at most w, and each vertex can be in one of three states (uncolored, colored by Maker, or colored by Breaker), this exploration can be performed in time $O(3^w)$.

Finally, given a position in which all the vertices are colored, computing the score can be done in $\mathcal{O}(m+n)$ time. Putting everything together, the overall complexity is $\mathcal{O}(3^w(|E|+|V|))$. ◀

5 Conclusion

In this paper, we initiated the study of the Maker-Breaker Scoring Happy Vertex Game, which corresponds to a scoring version of the domination game, and we focused on some simple graph classes. Among our considerations, the NP-hardness proof for caterpillars is related to a question of [4], which proved that the domination game is in NP when restricted to interval graphs, and asks whether the problem is NP-complete or not. In the case of the Happy vertex game, the NP-hardness result for caterpillar makes us wonder whether the game is in NP or PSPACE-complete for this class of graphs.

Among the other natural classes of graphs to consider, a natural extension of our results for the neighborhood diversity would be to consider graphs of bounded modularwidth, starting naturally with cographs which have modularwidth 0 and are known to be polynomial-time solvable in the Maker-Breaker Domination game. However, this study is not as simple here, since isolating a single vertex will not end the game: On playing on a union of cographs having each a single universal vertex, a move from Maker will consist in replacing one of these cographs by the union of its component, while a move of Breaker will consist in removing one of the connected component of the graph. This particular behavior makes it difficult to propose strategies.

Counting the number of winning sets claimed by Maker appears as a natural generalization of a Maker-Breaker game which was first introduced in [3]. A natural question, considering instances in which Maker wins, would be to consider other classical positional games on graphs, and to count the number of winning sets she can fill up. The study of the complexity of the other classical positional games on graphs has been studied recently [9], and observing these games with a scoring view seems to be a natural new step in their study. In particular, the study of the H-game led to the study of the number of triangles Maker can create [14], and an algorithmic study of this game would be a natural extension of our results.

References

1 Akanksha Agrawal. On the parameterized complexity of happy vertex coloring. In *International Workshop on Combinatorial Algorithms*, pages 103–115. Springer, 2017. doi:10.1007/978-3-319-78825-8_9.

2 NR Aravind, Subrahmanyam Kalyanasundaram, and Anjeneya Swami Kare. Linear time algorithms for happy vertex coloring problems for trees. In *International Workshop on Combinatorial Algorithms*, pages 281–292. Springer, 2016.

3 Guillaume Bagan, Quentin Deschamps, Eric Duchêne, Bastien Durain, Brice Effantin, Valentin Gledel, Nacim Oijid, and Aline Parreau. Incidence, a scoring positional game on graphs. *Discrete Mathematics*, 347(8):113570, 2024.

4 Guillaume Bagan, Éric Duchêne, Valentin Gledel, Tuomo Lehtilä, and Aline Parreau. Partition strategies for the maker–breaker domination game. *Algorithmica*, 87(2):191–222, 2025. doi:10.1007/s00453-024-01280-x.

5 Piotr Berman and Marek Karpinski. On some tighter inapproximability results. In *Automata, Languages and Programming: 26th International Colloquium, ICALP'99 Prague, Czech Republic, July 11–15, 1999 Proceedings*, pages 200–209. Springer, 2002.

6 Édouard Bonnet, Serge Gaspers, Antonin Lambilliotte, Stefan Rümmele, and Abdallah Saffidine. The Parameterized Complexity of Positional Games. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 90:1–90:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2017.90.

7 Pakanun Dokyeesun. Maker-breaker domination game on cartesian products of graphs. *Communications in Combinatorics and Optimization*, pages –, 2025. doi:10.22049/cco.2025.29866.2198.

8 Eric Duchene, Valentin Gledel, Aline Parreau, and Gabriel Renault. Maker–breaker domination game. *Discrete Mathematics*, 343(9):111955, 2020. doi:10.1016/J.DISC.2020.111955.

9 Eric Duchêne, Valentin Gledel, Fionn Mc Inerney, Nicolas Nisse, Nacim Oijid, Aline Parreau, and Miloš Stojaković. Complexity of maker–breaker games on edge sets of graphs. *Discrete Applied Mathematics*, 361:502–522, 2025. doi:10.1016/j.dam.2024.11.012.

10 Eric Duchêne, Nacim Oijid, and Aline Parreau. Bipartite instances of influence. *Theoretical Computer Science*, 982:114274, 2024. doi:10.1016/j.tcs.2023.114274.

11 Hiroshi Eto, Akiko Fujimoto, Hironori Kiya, Ruka Matsushita, Eiji Miyano, Yuto Murao, and Toshiki Saitoh. Happy vertex game. In *Proceedings of the International Symposium on Computing and Networking (CANDAR 2025)*, 2025.

12 Florian Galliot. 4-uniform maker-breaker and maker-maker games are pspace-complete, 2025. doi:10.48550/arXiv.2509.13819.

13 Florian Galliot, Sylvain Gravier, and Isabelle Sivignon. Maker-breaker is solved in polynomial time on hypergraphs of rank 3, 2025. arXiv:2209.12819.

14 Christian Glazik and Anand Srivastav. A new bound for the maker–breaker triangle game. *European Journal of Combinatorics*, 104:103536, 2022. doi:10.1016/j.ejc.2022.103536.

15 Alfred W Hales and Robert I Jewett. Regularity and positional games. In *Classic Papers in Combinatorics*, pages 320–327. Springer, 2009.

16 Olof Hanner. Mean play of sums of positional games. *Pacific Journal of Mathematics*, 9:81–99, 1959. URL: https://api.semanticscholar.org/CorpusID:120425420.

17 Finn Orson Koepke. Solving maker-breaker games on 5-uniform hypergraphs is pspace-complete. *arXiv preprint arXiv:2502.20271*, 2025. doi:10.48550/arXiv.2502.20271.

18 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, 2012. doi:10.1007/s00453-011-9554-x.

19 Urban Larsson. *Games of No Chance 5*, volume 5. Cambridge University Press, 2019.

20 Rhyd Lewis, Dhananjay Thiruvady, and Kerri Morgan. Finding happiness: an analysis of the maximum happy vertices problem. *Computers & Operations Research*, 103:265–276, 2019. doi:10.1016/J.COR.2018.11.015.

21 John Milnor. Sums of positional games. *Ann. of Math. Stud.(Contributions to the Theory of Games, HW Kuhn and AW Tucker, eds.), Princeton*, 2(28):291–301, 1953.

22 Nacim Oijid. *Complexité des jeux positionnels sur les graphes*. PhD thesis, Université Claude Bernard-Lyon I, 2024.

23 Nacim Oijid. Bounded degree qbf and positional games. In Irene Finocchi and Loukas Georgiadis, editors, *Algorithms and Complexity*, pages 121–135, Cham, 2025. Springer Nature Switzerland. doi:10.1007/978-3-031-92935-9_8.

24 Md Lutfar Rahman and Thomas Watson. 6-uniform maker-breaker game is pspace-complete. *Combinatorica*, 43(3):595–612, 2023. doi:10.1007/S00493-023-00026-7.

25 Thomas J Schaefer. On the complexity of some two-person perfect-information games. *Journal of computer and system Sciences*, 16(2):185–225, 1978. doi:10.1016/0022-0000(78)90045-4.

26 Larry J. Stockmeyer and Albert R. Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 1–9, 1973. doi:10.1145/800125.804029.

27 Peng Zhang and Angsheng Li. Algorithmic aspects of homophyly of networks. *Theoretical Computer Science*, 593:117–131, 2015. doi:10.1016/j.tcs.2015.06.003.

28 Peng Zhang, Yao Xu, Tao Jiang, Angsheng Li, Guohui Lin, and Eiji Miyano. Improved approximation algorithms for the maximum happy vertices and edges problems. *Algorithmica*, 80(5):1412–1438, 2018. doi:10.1007/S00453-017-0302-8.