



**HAL**  
open science

# Complexity and algorithms for Arc-Kayles and Non-Disconnecting Arc-Kayles

Kyle Burke, Antoine Dailly, Nacim Oijid

► **To cite this version:**

Kyle Burke, Antoine Dailly, Nacim Oijid. Complexity and algorithms for Arc-Kayles and Non-Disconnecting Arc-Kayles. 2024. hal-04495881

**HAL Id: hal-04495881**

**<https://hal.science/hal-04495881v1>**

Preprint submitted on 8 Mar 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Complexity and algorithms for ARC-KAYLES and NON-DISCONNECTING ARC-KAYLES\*

Kyle Burke<sup>1</sup>, Antoine Dailly<sup>2</sup>, and Nacim Oijid<sup>3</sup>

<sup>1</sup>Florida Southern College, Lakeland, FL 33801, USA

<sup>2</sup>Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne,  
Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France

<sup>3</sup>Univ Lyon, CNRS, INSA Lyon, UCBL, Centrale Lyon, Univ Lyon 2, LIRIS,  
UMR5205, F-69622 Villeurbanne, France

## Abstract

ARC-KAYLES is a game where two players alternate removing two adjacent vertices until no move is left. Introduced in 1978, its computational complexity is still open. More recently, subtraction games, where the players cannot disconnect the graph while removing vertices, were introduced. In particular, ARC-KAYLES admits a non-disconnecting variant that is a subtraction game. We study the computational complexity of subtraction games on graphs, proving that they are PSPACE-complete even on very structured graph classes (split, bipartite of any even girth). We prove that NON-DISCONNECTING ARC-KAYLES can be solved in polynomial-time on unicyclic graphs, clique trees, and subclasses of threshold graphs. We also show that a sufficient condition for a second player-win on ARC-KAYLES is equivalent to the graph isomorphism problem.

## 1 Introduction

ARC-KAYLES is one of the many games introduced by Schaefer in his seminal paper on the computational complexity of games [20]. It is a two-player, information-perfect, finite vertex deletion game, in which the players alternate removing two adjacent vertices and all their incident edges from a graph. The game ends when the graph is either empty or reduced to an independent graph, and the first player unable to play loses.

While Schaefer proved the PSPACE-completeness of many games, the complexity of ARC-KAYLES surprisingly remains open. Some FPT and XP algorithms have been proposed, and the game has been studied on some specific graph classes[17], but the results are quite scarce, the game is still generally open on even subdivided stars with three paths.

ARC-KAYLES can also be seen as an *octal game* played on a graph. Octal games are taking-breaking games played on heaps of counters, where the octal code defines how many counters one can take from a heap, as well as how the heap can be left after the move (empty, non-empty, split in two non-empty heaps). An overview of octal games can be found in volume 1 of [4]. In [2], octal games were expanded to be played on graphs, where removing counters means removing a connected subgraph, and splitting a heap means disconnecting the graph. Under this definition, ARC-KAYLES is the octal game **0.07**.

Non-disconnecting octal games are also called *subtraction games*, and have also been studied on graphs [9]: for a non-empty set of integers  $S$ , the *connected subtraction game on  $S$* ,  $CSG(S)$ , is played by removing connected subgraphs of order  $k$  (where  $k \in S$ ) without disconnecting the graph. The non-disconnecting condition of subtraction games allows for better results, compared to

---

\*The research of the second author was supported by the International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25 and by the ANR project GRALMECO (ANR-21-CE48-0004). The research of the third author was supported by the ANR project P-GASE (ANR-21-CE48-0001-01).

ARC-KAYLES, on constrained classes such as trees. Note that NON-DISCONNECTING ARC-KAYLES is  $\text{CSG}(\{2\})$ .

The aim of this paper is to further the study of the computational complexity of ARC-KAYLES and of subtraction games, particularly NON-DISCONNECTING ARC-KAYLES. We present results in PSPACE-completeness for subtraction games, an FPT algorithm for ARC-KAYLES and several polynomial-time algorithms for NON-DISCONNECTING ARC-KAYLES on structured graph classes.

**Combinatorial Game Theory terminology.** ARC-KAYLES and subtraction games are impartial games, a subset of combinatorial games which have been extensively studied. Since both players have the same possible moves, impartial games can have two possible *outcomes*: either the first player has a winning strategy ( $\mathcal{N}$ ) or the second player does ( $\mathcal{P}$ ). A position  $G'$  that can be reached by a move from a position  $G$  is called an *option* of  $G$ , the set of options of  $G$  is denoted by  $\text{opt}(G)$ . If every option of  $G$  is  $\mathcal{N}$ , then  $G$  is  $\mathcal{P}$ ; conversely, if any option is  $\mathcal{P}$ , then  $G$  is  $\mathcal{N}$ . A refinement of outcomes are the *Sprague-Grundy values* [14, 22], denoted by  $\mathcal{G}(G)$  for a given position  $G$ . The value can be computed inductively, with  $\mathcal{G}(G) = \text{mex}(\text{opt}(G))$ , where  $\text{mex}(S)$  is the smallest non-negative integer not in  $S$ . A position  $G$  is  $\mathcal{P}$  if and only if  $\mathcal{G}(G) = 0$ . Furthermore, ARC-KAYLES and connected subtraction games are *vertex deletion games*, a family of games where the players alternate removing vertices from a graph following some fixed constraints. For more information and history about combinatorial games, we refer the reader to [1, 4, 6, 21].

**ARC-KAYLES: a difficult game.** As stated above, while ARC-KAYLES was introduced in 1978 [20], its complexity remains open to this day. However, there have been some algorithmic results. Determining the winner of ARC-KAYLES is FPT when parameterized by the number of rounds [18] and the vertex-cover number, and XP when parameterized by neighborhood diversity [25, 16]. ARC-KAYLES on paths is exactly the octal game **0.07**, and its Grundy sequence is well-known to have period 34 and a preperiod of 68 [15]. ARC-KAYLES on grid graphs is exactly the game CRAM [13], and thus there are winning strategies for the first player on even-by-odd grids and for the second player on even-by-even grids (by an easy symmetry argument). The game remains open on odd-by-odd grids; Sprague-Grundy values for grids with at most 30 squares [23] and of size  $3 \times n$  for  $n$  up to 21 and a few other odd-by-odd grids [19, 24] have been computed without any regularity appearing. Finally, the game seems to be hard to solve even on trees: an  $\mathcal{O}(2^{\frac{n}{2}})$  enumerative algorithm was proposed [16]. While the game remains open on subdivided stars with three paths, one of which has size 1 [17]—by fixing the length of the second path and having the third one change—the authors conjectured that the sequence of Sprague-Grundy values thus obtained would ultimately be periodic. However, the Sprague-Grundy values of ARC-KAYLES are unbounded in the general case [8].

**Subtraction games on graphs.** Subtraction games have been extensively studied on heaps of counters, and there are several regularity results on them [1, 4]. It is important to note that these are different rulesets than NIMG or GRAPHNIM, game where NIM-heaps are embedded on the nodes or vertices of the graph [11, 12, 5]. Subtraction games on graphs were introduced in [9]. They are of particular interest since they restrict the types of moves available on sparser classes (such as trees), and because they cannot be solved through a simple symmetry strategy started by cutting the graph in two isomorphic subgraphs. Hence, their analysis differs fundamentally from the analysis of vertex deletion games that allow disconnecting.

In [9], the authors proved that, by fixing  $G$  and a vertex  $u$ , the Sprague-Grundy values of the sequence of graphs obtained by appending a path of increasing length to  $u$  were ultimately periodic. They also found regularity results for the family  $\text{CSG}(\{1, \dots, N\})$  on subdivided stars with three paths, one of which has size 1, in contrast with ARC-KAYLES, which they used as base cases for subdivided stars with specific values of  $N$ . In [2], the authors gave polynomial-time algorithms for the game  $\text{CSG}(\{1, 2\})$  on subdivided stars and bistars.

As for the specific case of NON-DISCONNECTING ARC-KAYLES<sup>1</sup>, which is the subtraction game  $\text{CSG}(\{2\})$ , it is known to be polynomial-time solvable on trees, wheels and grids of height at most 3 [7].

**Our results and outline of the paper.** We further expand algorithmic results on ARC-KAYLES

<sup>1</sup>A playable version is available at <http://kyleburke.info/DB/combGames/nonDisconnectingArcKayles.html>.

and subtraction games on graphs. Our first contribution is a general PSPACE-completeness reduction for most subtraction games:

**Theorem 1.** *If  $S$  is finite and  $1 \notin S$ , then,  $\text{CSG}(S)$  is PSPACE-complete, even on bipartite graphs of any given even girth.*

We also prove that  $\text{CSG}(\{k\})$  is PSPACE-complete on split graphs, a very structured graph class. Note that NON-DISCONNECTING ARC-KAYLES is in this family of games.

**Theorem 2.** *For  $k \geq 2$ ,  $\text{CSG}(\{k\})$  is PSPACE-complete, even on split graphs.*

Both of those results are proved in Section 2, where we also show that deciding whether a symmetry strategy can be applied for ARC-KAYLES is as hard as the Graph-Isomorphism problem (GI-hard), even on bipartite graphs.

Finally, in Section 3, we give polynomial-time algorithms for deciding the winner of NON-DISCONNECTING ARC-KAYLES on several structured graph classes: unicyclic graphs, clique trees, and subclasses of threshold graphs.

## 2 Hardness results for vertex deletion games

Before we get to the hardness results, we point out that  $\text{CSG}(\{1\})$  is a strategy-free game. Every vertex can be removed over the course of a game, so players can count the number of moves remaining.

For our first proof, we need to define NODE-KAYLES, another impartial vertex deletion game. Each turn, the current player selects one vertex on the graph that is not adjacent to an already selected vertex (there is no distinction between the players' selected vertices). Just as in ARC-KAYLES, under normal play the last player to move wins. Note that, while NODE-KAYLES is a vertex deletion game (it can be seen as alternately removing one vertex and all its neighbors), it cannot be expressed as a subtraction or octal game on graphs, since the deletion of vertices is restricted by the graph itself and not by a specific set of integers. Recall that the *girth* of a graph is the length of its smallest cycle.

**Theorem 1.** *If  $S$  is finite and  $1 \notin S$ , then,  $\text{CSG}(S)$  is PSPACE-complete, even on bipartite graphs of any given even girth.*

*Proof.* First, note that  $\text{CSG}(S)$  is clearly in PSPACE. To prove completeness, we reduce from NODE-KAYLES, which was shown to be PSPACE-complete in [20].

Let  $G(V, E)$  be a position of NODE-KAYLES. For a given finite set  $S$  such that  $1 \notin S$ , we construct a position  $G'(V', E')$  of  $\text{CSG}(S)$  such that the outcomes of  $G$  for NODE-KAYLES and of  $G'$  for  $\text{CSG}(S)$  are the same.

Let  $M = \max(S)$ . We construct  $G'$  from  $G$  as follows:

- Create a so-called *control vertex*  $c$ , and attach  $M + 1$  leaves to it;
- For every vertex  $v \in V$ , create a star with center  $v'$  and  $M - 1$  leaves, and add the edge  $cv'$ ;
- For every edge  $uv \in E$ , create a vertex  $e_{uv}$ , attach  $M$  leaves to it, and add the edges  $u'e_{uv}$  and  $v'e_{uv}$ .

The reduction is depicted on Figure 1. It is clearly polynomial, since we have  $|V'| = M|V| + (M + 1)|E| + M + 2$ .

We now prove that playing NODE-KAYLES on  $G$  is exactly the same as playing  $\text{CSG}(S)$  on  $G'$ .

First, note that the only possible moves on  $G'$  consist of removing a vertex  $v'$  and its attached leaves. Indeed, since  $1 \notin S$ , it is impossible to play on an induced star of order more than  $M$ , except if the graph is reduced to exactly a star of order  $M + 1$ . Hence, it is always impossible to remove  $c$  or any  $e_{uv}$  for  $uv \in E$ , leaving the stars centered on vertices  $v'$  as the only possible moves.

Furthermore, if a vertex  $u'$  has been removed, then it is impossible to remove any vertex  $v'$  such that  $uv \in E$ , since we cannot remove the star centered on  $e_{uv}$ , as doing so would disconnect the graph.

Hence, the only possible moves for  $\text{CSG}(S)$  on  $G'$  are on vertices equivalent to those of  $G$ , and playing on such a vertex (removing the star it is the center of) prevents either player from playing on any of its neighbors. This proves that the moves of  $\text{CSG}(S)$  on  $G'$  are exactly the ones of  $\text{NODE-KAYLES}$  on  $G$ , and thus the game trees (and thus the outcomes and the Grundy values) of those two games are the same.

Note that  $G'$  has girth 4, since for any edge  $uv$  in  $G$  there is a 4-cycle  $u'cv'e_{uv}$  in  $G'$ , and that  $G'$  is bipartite, since any odd cycle in  $G$  now contains twice as many vertices since each edge of  $G$  has been subdivided, and no cycle containing  $c$  can be of odd length. Furthermore, the girth of  $G$  can be increased (but will remain even) by subdividing every edge enough times and adding  $M + 1$  leaves to the newly created vertices: a cycle using  $c$  will have length  $4 + 2i$  if we subdivide the edges incident with  $c$   $i$  times, afterwards we can subdivide the edges of any more smaller cycles in  $G'$  to increase the girth. This transformation does not affect the proof, since the newly created vertices can never be removed, thus proving that  $G'$  can have any even girth.  $\square$

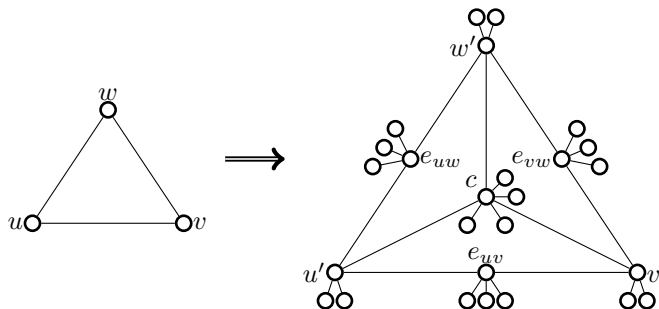


Figure 1: An illustration of the reduction from  $\text{NODE-KAYLES}$  to  $\text{CSG}(S)$  with  $S = \{2, 3\}$ .

We remark that the non-disconnecting variant of  $\text{NODE-KAYLES}^2$ , where the selected vertices and their neighbors must not induce a separating set (*i.e.*, a set of vertices that disconnects the graph), is also  $\text{PSPACE-complete}$  under this definition.

**Theorem 3.** *NON-DISCONNECTING NODE-KAYLES is PSPACE-complete.*

*Proof.* We reduce from  $\text{NODE-KAYLES}$ , and consider both  $\text{NODE-KAYLES}$  and  $\text{NON-DISCONNECTING NODE-KAYLES}$  as vertex deletion games. Let  $G(V, E)$  be a position of  $\text{NODE-KAYLES}$ . We construct a position  $G'$  of  $\text{NON-DISCONNECTING NODE-KAYLES}$  by replacing, in  $V$ , every edge  $uv$  by the following gadget  $X_{uv}$ :

- Subdivide  $uv$  twice, let  $e_{uv}^1$  and  $e_{uv}^2$  be the two vertices thusly created;
- Construct a copy of the complete bipartite graph on six vertices  $K_{3,3}$ , and connect one of its vertices to both  $e_{uv}^1$  and  $e_{uv}^2$ .

The edge gadget  $X_{uv}$  is depicted on Figure 2. Afterwards, connect all the  $e_{uv}^1$ 's and  $e_{uv}^2$ 's for every edge  $uv \in E$ .

Playing  $\text{NODE-KAYLES}$  in  $G$  is strictly equivalent to playing  $\text{NON-DISCONNECTING NODE-KAYLES}$  in  $G'$ . Indeed, note that no player can ever select a vertex in a copy of  $K_{3,3}$ , as doing so would disconnect the two other vertices in its part of the copy from the rest of  $G'$ . Furthermore, selecting  $u$  in  $G'$  prevents both players from selecting  $v$  for every  $uv \in E$  afterwards, as this would disconnect the copy of  $K_{3,3}$  in  $X_{uv}$  from the rest of  $G'$ . Finally, it is impossible to select a vertex  $e_{uv}^1$  or  $e_{uv}^2$  from any gadget  $X_{uv}$ , as doing so would disconnect the copy of  $K_{3,3}$  in the gadget from the rest of  $G'$ . Hence, every move in  $G$  has an equivalent move in  $G'$  and conversely, so the two game trees, and thus outcomes and Grundy values, are the same.  $\square$

For our third reduction, we need to define  $\text{AVOID TRUE}$ , a game played on positive (without negations) boolean formulas and an assignment of all variables such that the formula is false. (The

<sup>2</sup>A playable version is available at <http://kyleburke.info/DB/combGames/nonDisconnectingNodeKayles.html>.

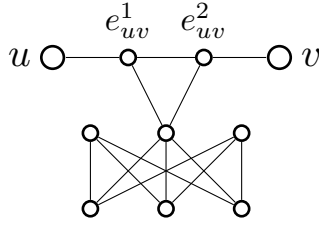


Figure 2: The edge gadget  $X_{uv}$  of the reduction of Theorem 3

variables are initially set to false.) Each turn the current player selects one false variable and switches the assignment to true. Players may not switch variables that cause the overall formula to evaluate to true. Under normal play, the last player to move wins. AVOID TRUE is known to be PSPACE-complete on formulas in disjunctive normal form[20]. Furthermore, recall that a split graph is a graph where the vertex set can be partitioned into a clique (so every possible edge within the part exists) and an independent set (so no edge exists within this part).

**Theorem 2.** For  $k \geq 2$ ,  $\text{CSG}(\{k\})$  is PSPACE-complete, even on split graphs.

*Proof.* The proof is obtained by reduction from AVOID TRUE.

Let  $\varphi = \bigvee_{i=1}^m C_i$  be an AVOID TRUE formula in disjunctive normal form over the variables  $x_1, \dots, x_n$ . We construct a graph  $G = (V, E)$  as follows:

- For any  $1 \leq i \leq n$ , we add  $k$  vertices  $v_i$  and  $v_i^1, \dots, v_i^{k-1}$  in  $V$  and all the edges  $(v_i, v_i^j)$  for  $1 \leq j \leq k-1$  in  $E$ .
- For any  $1 \leq i, j \leq n$  we add the edge  $(v_i, v_j)$  in  $E$ .
- For any  $1 \leq j \leq m$ , we add a vertex  $c_j$  in  $V$ , and if  $x_i \in C_j$ , we add the edge  $(v_i, c_j)$  in  $E$ .

The obtained graph (an example for  $k = 2$  is depicted on Figure 3) is a split graph, as the  $v_i$ 's are a clique and all the other vertices are an independent set.

First note that the only moves available are of the form  $\{v_i, v_i^1, \dots, v_i^{k-1}\}$ . Indeed, as the  $v_i$ 's are a vertex cover, any move has to contain at least one  $v_i$ , and if this move does not contain any of the  $v_i^j$  (for  $1 \leq j \leq k-1$ ), then this vertex becomes isolated and the graph is no longer connected.

Now we prove that the first (second resp.) player wins in AVOID TRUE if and only if he wins in NON-DISCONNECTING ARC-KAYLES.

Suppose that the first (second resp.) player has a winning strategy  $\mathcal{S}$  in AVOID TRUE. We define its strategy in NON-DISCONNECTING ARC-KAYLES as follows:

- Whenever  $\mathcal{S}$  wants to claim a variable  $x_i$ , he plays the set  $\{v_i, v_i^1, \dots, v_i^{k-1}\}$
- If his opponent plays a set  $\{v_i, v_i^1, \dots, v_i^{k-1}\}$ , he considers that the variable  $x_i$  has been played against  $\mathcal{S}$  in AVOID TRUE.

Following this strategy, as  $\mathcal{S}$  is a winning strategy in AVOID TRUE, the second (first resp.) player will be the first to satisfy a clause  $C_j$ . The moves satisfying  $C_j$  will therefore be removing the last vertex  $v_i \in C_j$  and will disconnect the graph. Therefore, the first (second resp.) player wins.  $\square$

Returning to ARC-KAYLES, we present a natural and sufficient winning condition for the second player, which is used in the case of grids. When the graph is highly symmetric, the second player may want to apply a symmetry strategy with respect to a given "point" in the graph. We show that, in the general case, deciding whether such a strategy is possible is as hard as the famous Graph Isomorphism problem. Note that such symmetry strategies are often used in games, and are often GI-hard to compute, see [3, 10].

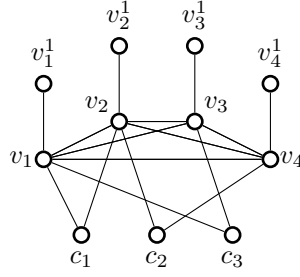


Figure 3: An illustration of the reduction from AVOID TRUE to CSG( $k$ ) with  $k = 2$ . The clauses are  $C_1 = (x_1 \wedge x_2)$ ,  $C_2 = (x_2 \wedge x_4)$  and  $C_3 = (x_1 \wedge x_3)$ . The  $v_i$ 's form a clique, and the other vertices an independent set.

**Definition 4.** An automorphism  $f$  of a graph  $G$  is said to be edge-disjoint if for any  $e \in E(G)$ , we have  $e \cap f(e) = \emptyset$ .

**Theorem 5.** Let  $G = (V, E)$  be a graph such that there exists an involutive automorphism  $f$  of  $G$  such that for all  $e \in E$ , we have  $e \cap f(e) = \emptyset$ . Then  $G$  is a  $\mathcal{P}$ -position in ARC-KAYLES.

*Proof.* Let  $G$  be a graph, and let  $f$  be an involutive automorphism of  $G$  satisfying the hypothesis of the Theorem. Consider the following strategy for the second player: each time his opponent plays an edge  $e$ , he answers by playing  $f(e)$ . This move is always available as  $e \cap f(e) = \emptyset$ , and  $G \setminus V(e, f(e))$  still satisfies the hypothesis of the theorem.  $\square$

Unlike ARC-KAYLES, in NON-DISCONNECTING ARC-KAYLES, there are graphs with such an  $f$  where the symmetry strategy does not give a winner, such as the cycle on six vertices: it has this property, but the outcome is  $\mathcal{N}$ . Indeed, in NON-DISCONNECTING ARC-KAYLES, the move  $f(e)$  can be forbidden as it could disconnect the graph.

**Theorem 6.** Let  $G$  be a graph. Verifying that a graph  $G$  admits an involutive automorphism  $f$  such that, for all  $e$ ,  $e \cap f(e) = \emptyset$  is GI-hard, even in bipartite graphs.

*Proof.* We provide a reduction from the graph isomorphism problem. Let  $G_1, G_2$  be two graphs. If  $|V(G_1)| \neq |V(G_2)|$ , let  $G' = K_2$ , two vertices connected by an edge. Otherwise, denote by  $n = |V(G_1)| = |V(G_2)|$ , and let  $G'_1, G'_2$  be the graphs obtained from  $G_1$  and  $G_2$  obtained by adding a universal vertex  $u_i$  and a leaf connected to it  $v_i$  in  $G_i$ . Let  $G' = G'_1 \cup G'_2$ . We prove that  $G_1$  and  $G_2$  are isomorphic if and only if  $G'$  admits an edge-disjoint involutive automorphism  $f$ .

Suppose first that  $G'$  admits an edge-disjoint involutive automorphism  $f$ . As  $u_1$  and  $u_2$  are the only two vertices of degree  $n + 1$ , they have to be the image of the other by  $f$ . Then, since  $v_1$  and  $v_2$  both are leaves and  $f(v_1) = u_2$ , we can suppose without loss of generality that  $f_1(v_1) = v_2$ . Now, by the hypothesis, as the automorphism conserves connectivity,  $f$  induces an isomorphism between  $G'_1$  and  $G'_2$ , and therefore between  $G_1$  and  $G_2$ .

Suppose now that  $G_1$  and  $G_2$  admits an isomorphism  $f$ . Then consider  $f' : G'_1 \rightarrow G'_2$  defined by

$$f'(v) = \begin{cases} u_2 & \text{if } v = u_1 \\ v_2 & \text{if } v = v_1 \\ f(v) & \text{otherwise.} \end{cases}$$

Now consider  $\phi : G' \rightarrow G'$  defined by

$$\phi(v) = \begin{cases} f'(v) & \text{if } v \in G'_1 \\ f'^{-1}(v) & \text{if } v \in G'_2 \end{cases}$$

By construction  $\phi$  is an involutive automorphism of  $G'$ , and for any  $e \in G'$ , we have  $e \in G'_1$ , if and only if  $\phi(e) \in G'_2$ . therefore  $e \cap \phi(e) = \emptyset$ . so this automorphism is edge-disjoint.  $\square$

Put more simply, deciding whether the second player can win by applying a symmetry strategy (such as on even-even grids) is GI-hard.

### 3 NON-DISCONNECTING ARC-KAYLES on structured graph families

While there are very few polynomial-time algorithms for ARC-KAYLES on structured graph classes, its non-disconnecting variant seems more approachable. In this section, we expand on the results in [7], where it was shown that trees are trivial to solve. Indeed, as with  $\text{CSG}(\{1\})$ , no strategy is required, as all possible moves on a tree will end up being played, and so the Sprague-Grundy value is either 0 or 1.

First, we will show that this result on trees can be extended to tree-like graphs: unicyclic graphs and clique trees. Recall that a graph is *unicyclic* if it contains exactly one cycle, and that it is a *clique tree* (or *block graph*) if every biconnected component is a clique.

**Theorem 7.** *NON-DISCONNECTING ARC-KAYLES can be solved in polynomial-time on unicyclic graphs.*

*Proof.* Let  $G$  be a unicyclic graph. We define *cycle-moves* on  $G$  as legal moves that remove at least one vertex of the cycle. Cycle-moves are of two types: they can remove one vertex from the cycle that has exactly one pendant leaf (along with this leaf), or they can remove two adjacent vertices of the cycle that have no neighbor outside of the cycle. Note that playing a cycle-move leaves a tree.

The algorithm works as follows. First, assume that there is at least one cycle-move. For every cycle-move, compute the outcome of NON-DISCONNECTING ARC-KAYLES on the remaining tree. If the outcome is  $\mathcal{P}$ , play this move. If no cycle-move leaves a tree with outcome  $\mathcal{P}$ , then, the first player loses and thus plays any move. The second player then applies the same algorithm to play their winning move. Assume now that no cycle-move exists, then the players will have to play on pendant trees until cycle-moves are open or until no move is possible. Since it is possible to know in polynomial-time if a given pendant tree will be emptied or reduced to a leaf, we can determine if a cycle-move will end up being possible. If so, compute the first player who will be able to play a cycle-move and apply the algorithm above for them. Otherwise, the graph is a disjoint union of a trees, for which the outcome can be computed in polynomial-time.

To prove the validity of the algorithm, it suffices to show that (i) if no cycle-move leads to a  $\mathcal{P}$ -position tree, then the first player has no winning move (since, if the first player can play to such a tree, they win) and (ii) it is possible to know which player will be the first to play a cycle-move.

(i) Assume that there is at least one cycle-move, but all cycle-moves lead to  $\mathcal{N}$ -positions. The first player plays on a given pendant tree. However, this has the effect of reducing the number of vertices in any tree resulting from a cycle-move by 2. Since no strategy is possible on trees, the result is a removal of one move from such trees. Hence, all cycle-moves lead to  $\mathcal{P}$ -positions, and thus the second player wins by playing any cycle-move.

(ii) For every pendant tree, we can compute in polynomial-time the number of moves that will either empty it or reduce it to a leaf. Thus, we can compute, for every vertex and pair of adjacent vertices of the cycle, the number of moves that will open a cycle-move on it or them. We can then compute which of the two players will benefit from those cycle-moves (by the same reasoning as in (i)). Hence, we can know which player will play a cycle-move first and thus win.  $\square$

**Theorem 8.** *NON-DISCONNECTING ARC-KAYLES can be solved in polynomial-time on clique trees.*

*Proof.* There are three types of vertices in a clique tree: *articulation points* (vertices that, when removed, disconnect the graph: they are the intersection of cliques), *leaves* (vertices of degree 1) and *clique vertices* (the remaining vertices). It is easy to see that the articulation points and leaves form a backbone tree-like structure that behaves like a tree for NON-DISCONNECTING ARC-KAYLES. Such vertices can only be removed if they do not disconnect the graph. For a leaf, this is when their neighbor has only clique vertices as other neighbors (and they get removed together). An articulation point can be removed with a clique vertex if it does not disconnect the graph, or along with another articulation point or leaf vertex under a similar condition. The clique vertices can be removed at any time without any consequence, and indeed they will have to be removed during the course of the game. Hence, the clique vertices are "covering" a tree-like structure, and so there is, as in trees, no strategy to follow: all possible moves will be played, and thus we can easily determine who wins.  $\square$



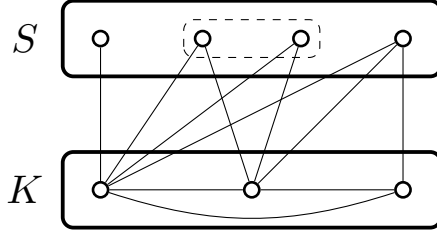


Figure 4: An example of a threshold graph with a twin-free clique.

Threshold graphs are a subclass of split graphs, where the vertices can be ordered as  $v_1, \dots, v_n$  such that  $N(v_1) \subseteq \dots \subseteq N(v_n)$ , where  $N(u)$  denotes the neighborhood of  $u$ . Alternatively, threshold graphs are graphs that can be constructed by repeatedly adding isolated or universal vertices. Note that this implies that the vertices can be partitioned into a clique and an independent set, meaning threshold graphs are split graphs. (Threshold graphs are actually exactly the intersection of split graphs and cographs.) In particular, threshold graphs can be partitioned into cliques  $K_1, \dots, K_m$  and independent sets  $S_1, \dots, S_m$  such that all the vertices of  $S_i$  (resp.  $K_i$ ) are twins (*i.e.*, vertices with the exact same neighborhood),  $\cup_{i=1}^m K_i$  is a clique, and the vertices of  $S_i$  are adjacent to the vertices of  $\cup_{j=1}^i K_j$ . The construction order is the following: for  $i$  from  $n$  to 1, add the vertices of  $S_i$  as isolated vertices, then the vertices of  $K_i$  as universal vertices. We denote a given threshold graph by  $(K \cup S, E)$ , where  $E$  is the edge set,  $K$  is the union of the cliques, and  $S$  is the union of the independent sets. If a vertex can be in both  $K$  and  $S$ , we consider that it is in  $S$  (so the independent set is *maximal* and the clique is *minimal* considering the fact that they form a partition of the vertices). An example of a threshold graph is depicted in Figure 4.

Recall that NON-DISCONNECTING ARC-KAYLES is PSPACE-complete on split graphs (Theorem 2), so the following result implies that there is a complexity gap between clique-twin-free threshold graphs and split graphs:

**Theorem 9.** *Let  $T = (K \cup S, E)$  be a threshold graph with  $|K| = n$  and such that  $K$  is twin-free. The outcome of NON-DISCONNECTING ARC-KAYLES on  $T$  can be computed in linear time. More precisely, if  $n \geq 5$ , then NON-DISCONNECTING ARC-KAYLES on  $T$  is outcome-equivalent to SUB( $\{1, 2\}$ ) on a heap of size  $n - 1$ .*

*Proof.* First, note that, as  $S$  is a stable set, it is not possible to remove two vertices of  $S$  in a single move. Therefore, the two possible moves are :

- Type 1: Removing one vertex of  $S$  and an adjacent vertex of  $K$ .
- Type 2: Removing two vertices of  $K$ .

We first deal with threshold graphs with small cliques. Note that as the clique is supposed to be minimal, any vertex  $v \in K$  has at least one neighbor in  $S$ , otherwise,  $v$  could be moved to  $S$ , and  $S$  would stay a stable, but  $K$  would be smaller.

- If  $T$  is a threshold graph with a twin-free minimum clique of order 1, then  $T$  is a star. Either  $T$  is just  $P_2$  or  $P_3$  (the path on two or three vertices), and the first player wins, or  $T$  has at least three edges, and no move is allowed on  $T$  thus the second player wins.
- If  $T$  is a threshold graph with a twin-free minimum clique of order 2, then, as we have at least two vertices in  $S$ , type 2 moves are impossible. Moreover, as  $K$  is twin free, and must have a universal vertex, the only possible move is of type 1. After this move, the graph is a star, and we are left with the previous case.
- If  $T$  is a threshold graph with a twin-free minimum clique of order 3, then the first player wins by playing a type 2 move, leaving the second player with a star or order at least 4.
- If  $T$  is a threshold graph with a twin-free minimum clique of order 4, then the second player wins. If the first player plays a move of type  $i$ , by playing move of type  $(3 - i)$ , the graph is left with one universal vertex in the clique and at least 3 leaves connected to it. Therefore, no moves are available, and the second player has won.

To prove the result for  $n \geq 5$ , we prove that the game will be finished once the clique is reduced to a single vertex.

Let  $G = (K \cup S, E)$  be a threshold graph with a maximal independent set  $S$  and the clique  $K$  of order  $n$ . Denote by  $v_1, \dots, v_n$  the vertices of  $K$  such that  $N(v_1) \supseteq N(v_2) \supseteq \dots \supseteq N(v_n)$ . Similarly, denote by  $u_1, \dots, u_N$ , with  $N \geq n$  the vertices of  $S$  such that  $N(u_1) \subseteq N(u_2) \subseteq \dots \subseteq N(u_N)$ .

Suppose that some moves have already been played. As  $|S| \geq |K|$  before the beginning of the game, and as it is not possible to remove more vertices from  $S$  than from  $K$ , at any moment of the game, there will be at least as many vertices remaining in  $S$  than in  $K$ .

Let  $i$  the largest integer such that  $v_i$  has not been removed. By inclusion of the neighborhoods, and as  $K$  is a clique and the game is non-disconnecting,  $v_i$  is a universal vertex. Therefore, any move that does not remove  $v_i$  cannot disconnect the graph, and is legal. Therefore, while  $K$  contains at least three vertices, the two types of moves are possible.

Finally, note that, even if  $|S| = |K|$  at the beginning of the game, as the optimal move for the player having a winning strategy in  $\text{SUB}(\{1, 2\})$ , is to remove the complementary (i.e. playing a move of type  $(3 - i)$  against a move of type  $i$ ) to its opponent after each move, in an optimal strategy, one of the three first moves will be a move in the clique, and therefore we will reach a position with  $|S| \geq |K| + 2$ , and thus the last vertices of the clique will not be removed.  $\square$

## 4 Conclusion

In this paper, we expanded previous PSPACE-hardness results for connected subtraction games, a very large class of vertex deletion games. In particular, we proved that, provided 1 is not in the subtraction set, then the game is PSPACE-complete even on bipartite graphs of any given girth. We also proved that  $\text{CSG}(k)$  is PSPACE-complete on split graphs, a very restricted class. However, the cases where 1 is in a subtraction set of at least two elements are still open, as well as other strong conditions on the graph (such as planarity, interval graphs, etc).

We also showed that a sufficient winning condition for the second player for ARC-KAYLES is GI-hard, which is one of the few hardness results on this game. While this is not enough to determine the computational complexity of ARC-KAYLES, such an approach might help to understand more about it. A future research direction would be to study more necessary and sufficient conditions for first and second player winning, and determine their complexity.

On the polynomial-time side, we showed that NON-DISCONNECTING ARC-KAYLES is easy on tree-like graphs, such as unicyclic graphs and clique trees. Note that this contrasts with ARC-KAYLES, where even subdivided stars with three paths are open. Future research could consider other tree-like classes, such as cacti. Furthermore, our study of clique-twin-free threshold graphs seems to imply that the complexity gap may lie between threshold and split graphs. Another direction would be to look at cographs, since threshold graphs are the intersection of split (for which the game is PSPACE-complete) and cographs.

## References

- [1] Albert, M., Nowakowski, R., Wolfe, D.: Lessons in play: an introduction to combinatorial game theory. AK Peters/CRC Press (2019)
- [2] Beaudou, L., Coupechoux, P., Dailly, A., Gravier, S., Moncel, J., Parreau, A., Sopena, E.: Octal games on graphs: The game 0.33 on subdivided stars and bistars. *Theoretical Computer Science* **746**, 19–35 (2018)
- [3] Bensmail, J., Fioravantes, F., Mc Inerney, F., Nisse, N.: The largest connected subgraph game. *Algorithmica* **84**(9), 2533–2555 (2022)
- [4] Berlekamp, E.R., Conway, J.H., Guy, R.K.: Winning ways for your mathematical plays. AK Peters/CRC Press (2001–2004)
- [5] Burke, K.W., George, O.: A pspace-complete graph nim. *CoRR* **abs/1101.1507** (2011), <http://arxiv.org/abs/1101.1507>

- [6] Conway, J.H.: On numbers and games. CRC Press (2000)
- [7] Dailly, A.: Criticalité, identification et jeux de suppression de sommets dans les graphes: Des étoiles plein les jeux. Ph.D. thesis, Université de Lyon (2018)
- [8] Dailly, A., Gledel, V., Heinrich, M.: A generalization of arc-kayles. *International Journal of Game Theory* **48**, 491–511 (2019)
- [9] Dailly, A., Moncel, J., Parreau, A.: Connected subtraction games on subdivided stars. *Integers: Electronic Journal of Combinatorial Number Theory* **19**, G3 (2019)
- [10] Duchêne, E., Oijid, N., Parreau, A.: Bipartite instances of influence. *Theoretical Computer Science* **982**, 114274 (2024)
- [11] Fukuyama, M.: A nim game played on graphs. *Theor. Comput. Sci.* **1-3**(304), 387–399 (2003)
- [12] Fukuyama, M.: A nim game played on graphs ii. *Theor. Comput. Sci.* **1-3**(304), 401–419 (2003)
- [13] Gardner, M.: Mathematical games: Cram, crosscram and quadrupage: new games having elusive winning strategies. *Scientific American* **230**(2), 106–108 (1974)
- [14] Grundy, P.M.: Mathematics and games. *Eureka* **2**, 6–8 (1939)
- [15] Guy, R.K., Smith, C.A.: The g-values of various games. In: *Mathematical Proceedings of the Cambridge Philosophical Society*. vol. 52(3), pp. 514–526. Cambridge University Press (1956)
- [16] Hanaka, T., Kiya, H., Ono, H., Yoshiwatari, K.: Winner determination algorithms for graph games with matching structures. *Algorithmica* pp. 1–17 (2023)
- [17] Huggan, M.A., Stevens, B.: Polynomial time graph families for arc kayles. *Integers* **16**, A86 (2016)
- [18] Lampis, M., Mitsou, V.: The computational complexity of the game of set and its theoretical applications. In: *LATIN 2014: Theoretical Informatics: 11th Latin American Symposium, Montevideo, Uruguay, March 31–April 4, 2014. Proceedings 11*. pp. 24–34. Springer (2014)
- [19] Lemoine, J., Viennot, S.: Computation records of normal and misère cram, <http://sprouts.tuxfamily.org/wiki/doku.php?id=records#cram>
- [20] Schaefer, T.J.: On the complexity of some two-person perfect-information games. *Journal of computer and system Sciences* **16**(2), 185–225 (1978)
- [21] Siegel, A.N.: *Combinatorial game theory*, vol. 146. American Mathematical Soc. (2013)
- [22] Sprague, R.: Über mathematische kampfspiele. *Tohoku Mathematical Journal, First Series* **41**, 438–444 (1935)
- [23] Uiterwijk, J.W.: Construction and investigation of cram endgame databases. *ICGA Journal* **40**(4), 425–437 (2018)
- [24] Uiterwijk, J.W.: Solving cram using combinatorial game theory. In: *Advances in Computer Games*, pp. 91–105. Springer (2019)
- [25] Yoshiwatari, K., Kiya, H., Hanaka, T., Ono, H.: Winner determination algorithms for graph games with matching structures. In: *International Workshop on Combinatorial Algorithms*. pp. 509–522. Springer (2022)