Antoine Dailly, Harmender Gahlawat, Zin Mar Myint

The Closed Geodetic Game: algorithms and strategies*

Antoine Dailly[1,2], Harmender Gahlawat[3], and Zin Mar Myint[4]

[1]Université Clermont Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France
[2]Université Clermont Auvergne, INRAE, UR TSCF, 63000, Clermont-Ferrand, France
[3]Laboratoire G-SCOP, Grenoble-INP, France
[4]Indian Institute of Technology Dharwad, India

Abstract

The *geodetic closure* of a set S of vertices of a graph G is the set of all vertices in shortest paths between pairs of vertices of S. A set S of vertices in a graph is *geodetic* if its geodetic closure contains all the vertices of the graph.

Several authors have studied variants of games around constructing geodetic sets. The most studied of those, Geodetic Game, was introduced by Harary in 1984 and developed by Buckley and Harary in 1985. It is an *achievement* game: both players construct together a geodetic set by alternately adding vertices to the set, the winner being the one who plays last. However, this version of the game allows the players to select vertices that already are in the geodetic closure of the current set.

We study the more natural variant, called Closed Geodetic Game, where the players alternate adding to S vertices that are not in the geodetic closure of S. This variant was also introduced in another, less-noticed, paper by Buckley and Harary in 1985, and only studied since then in the context of trees by Araujo *et al.* in 2024. We provide a full characterization of the Sprague-Grundy values (equivalence values for games) of graph classes such as paths and cycles, of the outcomes of several products of graphs in function of the outcomes of the two graphs, and give polynomial-time algorithms to determine the Sprague-Grundy values in cactus and block graphs.

Keywords: Geodetic Set · Geodetic Game · Polynomial-time algorithms · Graph Products · Block Graphs · Cactus Graphs · Combinatorial Games

1 Introduction

1.1 A history of geodetic games

Let $G(V, E)$ be a simple, undirected graph with vertex set V and edge set E. Given two vertices $u, v \in V$, let $\mathcal{I}(u, v)$ denote the set of vertices that lie on some shortest path with u and v as endpoints. Given a subset $S \subseteq V$ of vertices, its *geodetic closure*, denoted by (S), is defined as the set of vertices in all shortest paths between every pair of vertices of S. More formally, $(S) = \bigcup_{u,v\in S} \mathcal{I}(u, v)$.

A set S of vertices of $G(V, E)$ is called a *geodetic set of G* if $(S) = V$ (see Figure 1 for examples). The problem of finding a geodetic set of minimum size was defined in the 1980s [18, 6, 9, 17] and has been well-studied and is known to be NP-hard for general graphs [17, 3]. The problem has been also

*The research of the first author was supported by the International Research Center "Innovation Transportation and Production Systems" of the I-SITE CAP 20-25. The research of the second author was partially supported by ANR project GRALMECO (ANR-21-CE48-0004-01). The research of the third author was supported by the "Doctoral Fellowship in India for ASEAN (DIA:2020-25)".

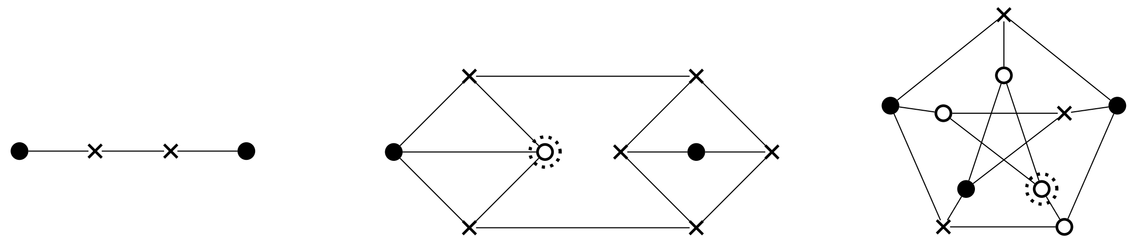


Figure 1: Black vertices belong to the set S, crossed vertices belong to the geodetic closure (S), white vertices are uncovered vertices yet. However, if we add the dotted vertices to S, then $V = (S)$, *i.e.*, S becomes a geodetic set.

well-studied on graph classes and displays quite an interesting behaviour. In particular, the problem stays NP-hard even for subcubic partial grids [11], for interval graphs [10], and admits a polynomial time algorithm for solid grids [11], outerplanar graphs [20], and proper interval graphs [13].

Interestingly, the game version of the geodetic number was introduced in parallel of its combinatorial counterpart. The first mention of a game based around geodetic sets was in 1984, by Harary [16]: two players alternate adding new vertices of a graph $G(V, E)$ to a set S, the game ends when $(S) = V$. Two winning conditions are proposed: in the *achievement* setting, the player who plays the last move wins; while they lose in the *avoidance* setting. Achievement and avoidance correspond to the *normal* and *misère* settings of combinatorial games, respectively. We will focus exclusively on achievement games.

With respect to combinatorial game theory, the first question to ask is, **given an input graph, decide which player wins**. This is called computing the *outcome* of the game. Note that, since the game is *impartial* (both players are indistinguishable), there are only two possible outcomes: either the first player wins (denoted by $\mathcal{N}$) or the second player wins (denoted by $\mathcal{P}$). Furthermore, when playing on disjoint connected components, each component is independent (selecting a vertex in a component has no influence whatsoever on the other components). If two components are $\mathcal{P}$, then, playing on their disjoint union is also $\mathcal{P}$ (the second player always applies their winning strategy on the component on which the first player played a move); if one component is $\mathcal{P}$ and the other is $\mathcal{N}$, then the first player wins by applying their winning strategy on the $\mathcal{N}$ component, becoming the second player on the disjoint union of two $\mathcal{P}$ components. However, outcomes are not enough to solve the disjoint union of two $\mathcal{N}$ components: we need to use *Sprague-Grundy values*, which are equivalence classes for games (and can be mapped to nonnegative integers). More details will be given in Section 1.3. Hence, a stronger question is, **for a given input graph, compute its Sprague-Grundy value**.

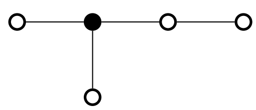
1.2 Geodetic Game, Closed Geodetic Game and other related variants

In his seminal paper [16], Harary discussed several possible rulesets for games based around the notion of geodesics. Two of those rulesets were then expanded in two 1985 papers by Buckley and Harary, and constitute the main variants. All games are played on a graph $G(V, E)$.

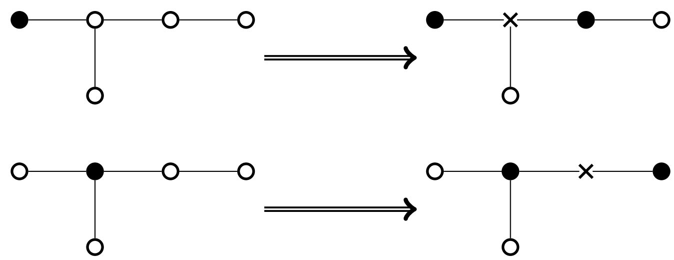
In the first paper [8], the authors define Geodetic Game. In this game, two players alternately add vertices to a set S, until $(S) = V$. They give the outcome of the game for complete graphs, cycles, (generalized) wheels, complete bipartite graphs and n-cubes. This line of research was expanded later in [21] which improved on the wheels' result, and in [19], which studied complete multipartite graphs, hypercubes, and graphs with a unique minimum-size geodetic set (such as trees, split graphs or coronas).

In the second paper [7], they introduce Closed Geodetic Game. In this game, the players alternatively add to S vertices that are not in (S) as it is at the moment they are selecting it. They study the same graph classes as in [8]. Surprisingly, this more natural variant was left unstudied for decades, probably due to the difficulty of access of [7] compared to [8]. Recently, Araujo *et al.* [2] studied several geodetic game variants, including Closed Geodetic Game (which they called *geodesic closed interval game*), for which they provided a linear-time algorithm computing the Sprague-Grundy value of a tree.

Note that Geodetic Game and Closed Geodetic Game can have different outcomes on even very simple graphs. One such example is illustrated in Figure 2. Since the players are allowed to pick covered vertices in Geodetic Game, the parity of the graph (and of covered vertices) plays a bigger role in determining the outcome compared to Closed Geodetic Game. Indeed, such is the case for Proposition 6 in [19], where graphs with a unique minimum-size geodetic set are solved due to parity, since the simili-passing moves (adding to S vertices that are in the current geodetic closure) are available.



(a) For Geodetic Game, the first player wins by picking the black vertex, ensuring that every vertex will be in S (if the second player picks a leaf, they pick the degree 2 vertex, and conversely).



(b) For Closed Geodetic Game, the second player always has an answer to the first player's first move, ensuring that exactly two moves will remain (other possible moves for the first player are treated by symmetry or similarity).

Figure 2: An example of a graph with different outcomes for Geodetic Game and Closed Geodetic Game.

Other variants were later introduced, such as the *hull game*, where the *convex hull* (instead of the geodetic closure) of S is considered [4, 14]. The outcomes for trees, cycles and complete graphs with rays were determined in [14], while [4] focused on Sprague-Grundy values for (among others) block graphs, lattice graphs, (generalized) wheels and complete multipartite graphs, as well as algebraic properties of the game.

Note that those games can be generalized as hypergraph games [22], where two players alternate selecting vertices of a hypergraph (potentially modifying it or under conditions) until those satisfy some condition related to the hyperedges. In particular, in Geodetic Game, the players add hyperedges corresponding to the shortest paths between the selected vertices, until every vertex is in at least one hyperedge.

1.3 Game definitions

Geodetic Game and Closed Geodetic Game 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 P' that can be reached by a move from a position P is called an *option* of P, the set of options of P is denoted by $\mathrm{opt}(P)$. If every option of P is $\mathcal{N}$, then P is $\mathcal{P}$; conversely, if any option is $\mathcal{P}$, then P is $\mathcal{N}$ (the winning strategy for the first player is to play to a $\mathcal{P}$ option). A refinement of outcomes are the *Sprague-Grundy values* [15, 24] (also called *nim-values* or *nimbers*), denoted by $\mathcal{G}(P)$ for a given position P and which can be mapped to nonnegative integers. The value of a position can be computed inductively, with $\mathcal{G}(P) = \mathrm{mex}(\{\mathcal{G}(P') \ : \ P' \in \mathrm{opt}(P)\})$, where $\mathrm{mex}(X)$ (for a set X of nonnegative integers) is the smallest nonnegative integer not in X. A position P is $\mathcal{P}$ if and only if $\mathcal{G}(P) = 0$. Two positions P and P' are said to be *equivalent*, denoted by $P \equiv P'$, if $\mathcal{G}(P) = \mathcal{G}(P')$.

The *disjoint sum* of two positions P and Q, denoted by $P + Q$, is a position where each player chooses to play on one of the two positions at their turn. When one of them is finished, the game continues on the other one; the last player to play wins. For Closed Geodetic Game, a disjoint sum can, for example, represent the disjoint union of two graphs. Computing the Sprague-Grundy value of a disjoint sum $P + Q$ can be done by applying the so-called *nim-sum*, which is the bitwise XOR: $\mathcal{G}(P + Q) = \mathcal{G}(P) \oplus \mathcal{G}(Q)$. An example is given below for $7 \oplus 10$.

$$\begin{array}{rcrl} 7 & \rightarrow & 1\,1\,1 & \\ 10 & \rightarrow & \underline{1\,0\,1\,0} & \\ & & 1\,1\,0\,1 & \rightarrow 13 \end{array}$$

For more information and history about combinatorial games, we refer the reader to [1, 5, 12, 23].

Due to the definition of the sum of positions, we can limit our study to connected graphs: studying a graph with several connected components is equivalent to studying each of the components independently and applying the nim-sum. Furthermore, we will always consider that both players play optimally. Optimal play is understood as, applying a winning strategy using as few rounds as possible if possible, and trying to make the game last as long as possible if no winning strategy exists. We will also make use of the following notation:

Notation 1. *Let G be a graph. We also denote the game position by G. Furthermore, let S be a subset of vertices of G, then the position denoted by G, S is the graph G where the vertices in S have already been selected.*

1.4 Structure of the paper

We will begin in Section 2 by proving general results on Closed Geodetic Game as well as studying how to compute the outcome for several graph products in function of the outcomes of its components. Then, in Section 3, we will study some graph classes where we can predict exactly which vertices will be picked. Afterwards, Section 4 will be focused on highly symmetric graphs such as paths, cycles and grids, and we compute the Sprague-Grundy values for the former two. Finally, in Section 5, we present polynomial-time algorithms based on dynamic programing computing the Sprague-Grundy values of block graphs and cacti.

2 General results and graph products

In this section, we present a few general results that help the study of Closed Geodetic Game, be it by stating that some vertices have to be selected, by helping decompose a graph, or by showing how some graph products affect the outcome of graphs.

First, we note that, in Closed Geodetic Game, due to the fact that vertices can be in the geodetic closure (S) without being in the geodetic set S only if they are on some shortest path between two vertices, some vertices always have to be selected. For example, a leaf (*i.e.*, a degree 1 vertex), can never be in a shortest path between two other vertices. More generally, consider *simplicial* vertices, that is, vertices whose neighbourhood induces a complete subgraph.

Lemma 2. *In Closed Geodetic Game, all simplicial vertices of the graph will have to be selected.*

Proof. A simplicial vertex can never be on a shortest path between two other vertices, since all its neighbours are fully connected to each other. Hence, it can never be in a geodetic closure (S) if it is not in S itself. □

Articulation points, that is, vertices whose removal disconnects the graph (also called *cut vertices* or *separating vertices*), are important for the study of Closed Geodetic Game. The following lemma will often be used to allow for decompositions:

Lemma 3. *Let G be a graph with an articulation point u, such that components $G_1, \ldots, G_k$ are attached at u. In* Closed Geodetic Game, $\mathcal{G}(G, \{u\}) = \mathcal{G}(G_1, \{u\}) \oplus \ldots \oplus \mathcal{G}(G_k, \{u\})$.

Proof. When u is selected, any selected vertex x in G_i will never help cover any vertex in G_j for $j \neq i$. Indeed, all the shortest paths from x to any vertex in G_j will go through u, so, from the point of view of G_j selecting x after u has no impact. Hence, each $G_i, \{u\}$ is an independent position, and we are playing on their disjoint union. This is depicted on Figure 3. □

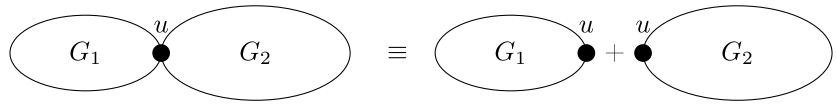


Figure 3: Selecting an articulation point is equivalent to splitting the graph and playing independently on the components.

We also focus on a classical question whenever studying games: if we know the outcomes of two graphs, can we obtain the outcome of some composition of those graphs? This question stems from the disjoint sum, which is managed through the nim-sum of Sprague-Grundy values. We study how some products of graphs interact with the outcomes for Closed Geodetic Game.

Given two graphs G and H, their *Cartesian product* $G \square H$ has vertices of the form (u, v) where $u \in V(G)$ and $v \in V(H)$ such that two vertices (u_1, v_1) and (u_2, v_2) are adjacent in $G \square H$ if

- $u_1 = u_2$ and v_1 is adjacent to v_2 in H, or
- $v_1 = v_2$ and u_1 is adjacent to u_2 in G.

Theorem 4. *If G and H are $\mathcal{N}$ for* Closed Geodetic Game, *then their Cartesian product $G \square H$ is also $\mathcal{N}$.*

Proof. Assume that the first player has a winning strategy for Closed Geodetic Game on both G and H. We will transfer the winning strategies to $G \square H$.

Initial Move: The first player selects (u_1, v_1) in $G \square H$, where u_1 (resp. v_1) is their winning move in G (resp. H).

Response to the second player's moves: After the second player makes a move in $G \square H$, selecting a vertex (u_2, v_2), we can analyze this move in terms of G and H:

- If either $u_2 = u_1$ or $v_2 = v_1$, this corresponds to replying on either G or H (respectively). In this case, the first player follows their winning strategy on the same graph by keeping the same fixed vertex. For example, say the second player played (u_1, v_2) and the winning strategy in H would be to select v_3, the first player selects (u_1, v_3). Note that finishing this strategy will not necessarily end the game on $G \square H$, but the first player always has such an answer to the second player's move (since they have winning strategies on both G and H).
- The second player cannot select (u_1, v_1).
- If both $u_2 \neq u_1$ and $v_2 \neq v_1$, this corresponds to replying on both G and H. The first player can now use their winning strategies in both G and H and choose the vertex (u_3, v_3) such that u_3 (resp. v_3) is the winning move in G (resp. H).

Consequently, the first player selects the vertices in $G \square H$ by translating their winning strategies in G and H. If one of the components of the vertex selected by the second player has already been covered in G or H (both cannot be covered: otherwise, the selected vertex itself would be covered too), then the first player uses the strategy from the first bullet, else it uses the strategy from the third bullet. Finally, we prove that the game ends when both G and H have a geodetic set constructed:

Claim 4.1. *If S_1 and S_2 are geodetic sets for G and H, respectively, then $S = \{(u,v) : u \in S_1 \text{ and } v \in S_2\}$ is a geodetic set for $G\square H$.*

Proof of Claim. Assume by contradiction that there is some (x,y) in $G\square H$ such that $(x,y) \notin (S)$. Since S_1 and S_2 cover in G and H, respectively, there exists some $u', u'' \in S_1$ and some $v', v'' \in S_2$ such that $x \in \mathcal{I}(u', u'')$ and $y \in \mathcal{I}(v', v'')$. We will show that $(x,y) \in \mathcal{I}((u',v'),(u'',v''))$, leading to a contradiction. Let the length of shortest u', u''-path (resp. v', v''-path) in G (resp. H) be α (resp. β). Then, observe that the length of a shortest $(u',v'),(u'',v'')$-path in $G\square H$ is $\alpha+\beta$. Hence, the path can be constructed as follows: a shortest $(u',v'),(x,v')$-path, followed by a shortest $(x,v'),(x,y)$-path, followed by a shortest $(x,y),(x,v'')$-path, followed by a shortest $(x,v''),(u'',v'')$-path is a path of length $\alpha+\beta$. Hence, $(x,y) \in \mathcal{I}((u',v'),(u'',v''))$, a contradiction. □

Therefore if all vertices in G and H are covered by geodetic sets, then the game of $G\square H$ will be completed. Hence, if the first player has a winning strategy in G and H, this strategy can be extended to $G\square H$, ensuring that the game ends on $G\square H$ when it ends on both G and H, and thus the outcome for $G\square H$ is also $\mathcal{N}$. This completes the proof. □

Reversing the roles of first and second players easily gives the following:

Theorem 5. *If G and H are $\mathcal{P}$ for* Closed Geodetic Game*, then their Cartesian product $G\square H$ is also $\mathcal{P}$.*

We have proved that if the graphs G and H each have the same outcome, then their Cartesian product also has the same outcome. We now observe that if G and H have different outcomes, then there is a second player winning strategy in $G\square H$. This implies that $\mathcal{P}$ is absorbing for the outcomes of the Cartesian product of graphs.

Theorem 6. *If G and H have different outcomes for* Closed Geodetic Game*, then their Cartesian product $G\square H$ has outcome $\mathcal{P}$.*

Sketch. If G is $\mathcal{N}$ and H is $\mathcal{P}$, the second player will simulate the game on both graphs, selecting (u,v) in $G\square H$ where u is either the winning move (if the first player played a non-optimal move) or the same vertex that the first player selected, and v is their winning move in H for replying to the first player. The game will end in $G\square H$ when both G and H will be over, and thus the second player will always have an answer to the first player's moves. □ □

Proof. Assume the first player has a winning strategy in G, and the second player has a winning strategy in H. We will convert these strategies to construct a winning strategy for the second player in $G\square H$. Consider the initial move by the first player in $G\square H$, selecting a vertex (u_1, v_1):

Initial Move and analysis: Depending on whether u_1 is optimal or not according to the winning strategy for G, and similarly for v_1 in H, the second player will respond as follows:

- *If u_1 is an optimal move in G:* Let v_2 be a vertex in H that is an optimal move according to the winning strategy of the second player (in particular, we have $v_2 \neq v_1$). The second player selects the vertex (u_1, v_2) in $G\square H$.
- *If u_1 is not an optimal move in G:* Let u_2 be a vertex in G that is a winning move in $G, \{u_1\}$, and let v_2 be a vertex in H that is an optimal move according to the winning strategy of the second player (again, we have $v_2 \neq v_1$). The second player selects the vertex (u_2, v_2) in $G\square H$.

Response to the following first player's moves: Each move by the first player in $G\square H$ can be analyzed by translating it into moves in the individual graphs G and H. In the subsequent rounds, let S be the set of selected vertices in $G\square H$ at the end of the second player's turn, with S_1 (resp.

S_2) being the set of vertices $x \in G$ (resp. $y \in H$) such that there is $y \in H$ (resp. $x \in G$) such that $(x, y) \in S$.

Assume that the first player selects $(u_i, v_i) \notin (S)$. If either $u_i \in (S_1)$ or $v_i \in (S_2)$, then the second player analyzes whether $u_i \in G$ is an optimal move or not. If $u_i \in (S_1)$, u_i is not an optimal move in G, and $v_i \in (S_2)$, then, the second player selects (u_j, v_j), where u_j (such that $u_j \neq u_i$ and $u_j \notin (S_1)$) is the optimal move according to the winning strategy of the second player in the current situation in G, and v_j (such that $v_j \neq v_i$ and $v_j \in (S_2)$) is the optimal move in H according to their strategy on H. If $u_i \notin (S_1)$ is an optimal move in G, and $v_i \in (S_2)$, then, the second player selects (u_i, v_j), where v_j (such that $v_j \neq v_i$ and $v_j \in (S_2)$) is the optimal move in H, in order to block the first player's winning strategy in G. If $u_i \in (S_1)$ and $v_i \notin (S_2)$, then, the second player selects (u_j, v_j), where $u_j \in (S_1)$ and $v_j \notin (S_2)$ are the optimal responses according to their winning strategies in G and H.

If neither $u_i \in (S_1)$ nor $v_i \in (S_2)$, then the second player analyzes whether $u_i \in G$ is optimal or not. Then, the second player follows the above-mentioned strategies according to their winning strategies and will select the optimal move on both games G and H, respectively. The second player will pick the vertex (u_j, v_i) in $G \square H$ by translating the moves of G and H.

End of the game: Throughout the game, it is crucial for the second player to analyze the first player's move in G before responding in $G \square H$. Given that the second player has a winning strategy in H, the optimality of the first player's move in H becomes irrelevant (they will always be able to follow their winning strategy). The game in $G \square H$ will be completed when both G and H are covered entirely as per the above-mentioned strategies of the second player: since the second player is using strategies derived from their winning strategy on H and effectively blocking the first player's moves in G, all vertices in $G \square H$ will be covered when the independent strategies have covered G and H individually. Hence, the second player will always have an answer to the first player's move, and thus will win the game. □

We also study the tensor and strong products for Closed Geodetic Game, where $\mathcal{P}$ is also absorbing for the outcomes.

Given two graphs G and H, their *tensor product* $G \times H$ and *strong product* $G \boxtimes H$ have vertices of the form (u, v) where $u \in V(G)$ and $v \in V(H)$. In $G \times H$, two vertices (u_1, v_1) and (u_2, v_2) are adjacent if and only if u_1 is adjacent to u_2 in G and v_1 is adjacent to v_2 in H. In $G \boxtimes H$, two vertices (u_1, v_1) and (u_2, v_2) are adjacent if and only if (u_1, v_1) and (u_2, v_2) are adjacent in at least one of the Cartesian product $G \square H$ and the tensor product $G \times H$.

Theorem 7. *The tensor product $G \times H$ of two graphs G and H is $\mathcal{N}$ if and only if both G and H are $\mathcal{N}$.*

Proof. First, assume that the first player has a winning strategy for Closed Geodetic Game on both G and H. We will show how to use the winning strategies on $G \times H$. The first player begins by selecting (u_0, v_0), where u_0 and v_0 are winning moves in G and H, respectively (see Figure 4a).

We then construct a matrix M with the rows representing the copies of vertices of G, and the columns representing the copies of vertices of H; hence each cell of M is a vertex of $G \times H$. Note that the first player selected the cell corresponding to (u_0, v_0), fixing a row and a column, according to their winning strategies in G and H. We will now associate rows and columns together, explaining the strategy of the first player. We will begin by detailing how to answer to the first move of the second player.

Assume first that the second player selected (u_0, v_1). The first player selects (u_0, v_2) where v_2 is the answer to v_1 in H, and associates the columns of v_1 and v_2. Apply the same strategy for H if the second player selected (u_1, v_0), associating the rows of u_1 and u_2. Note that $u_2 \neq u_0$ and $v_2 \neq v_0$. See Figures 4b and 4c for an illustration.

Assume now that the second player selected (u_1, v_1), where $u_1 \neq u_0$ and $v_1 \neq v_0$. The second player selects (u_2, v_2) such that they are winning moves in $G, \{u_0, u_1\}$ and $H, \{v_0, v_1\}$, respectively, so

$u_2 \neq u_0$ and $v_2 \neq v_0$. Associate the rows of u_1 and u_2 and the columns of v_1 and v_2. See Figure 4d for an illustration.

We now consider a move later in the game. The second player selects (u, v) in $G \times H$. If neither u nor v is in an associated row or column, and $u \neq u_0$ and $v \neq v_0$, then do the same: apply the winning strategy on G and H and associate rows and columns. This is always possible since such winning moves exist by the outcomes of G and H and some vertex (w, x) that corresponds to these answers will not be in the already-constructed geodetic closure. However, if u is in an associated row, then the answer will necessarily be a w in said associated row (and the same for v, with an associated column). If $u = u_0$ (resp. $v = v_0$), then, the answer will necessarily be in the u_0 row (resp. the v_0 column). This ensures symmetry and correspondences between rows and columns, and that such an answer always exists: if the answer did not exist, then the move (u, v) would not have been possible in the first place. See Figure 4e for an illustration.

By doing so, the first player ensures that they will always have an answer to the second player's move, and thus they will win in $G \times H$. Note that the individual simulated games G and H may end several times during the full game, but the first player will always be the last to play (since both G and H are $\mathcal{N}$), and hence they will win, so $G \times H$ is also $\mathcal{N}$.

Now, we assume that at least one of the outcomes of G and H is $\mathcal{P}$. Assume without loss of generality that G has $\mathcal{P}$, so the first player does not have a winning strategy on G. Thus, whatever move the first player makes in $G \times H$, there exists a counter-strategy in G that ensures the second player can eventually win. We detail the second player's winning strategy in response to the first player's moves, ensuring they can always win in $G \times H$ by again constructing a matrix $M = G \times H$ as in the case above. Let (u_0, v_0) be the first player's move in $G \times H$.

- If u_0 is a winning move in G (see Figure 5a), then the second player selects (u_0, v_1) where $v_1 \neq v_0$ is the answer to v_0 in H, and associates the columns of v_0 and v_1. See Figure 5a for an illustration.
- If u_0 is not a winning move in G (see Figure 5b), then the second player selects (u_1, v_1) such that they are winning moves in $G, \{u_0\}$ and $H, \{v_0\}$ where $u_1 \neq u_0$ and $v_1 \neq v_0$ in G and H, respectively. See Figure 5b for an illustration.

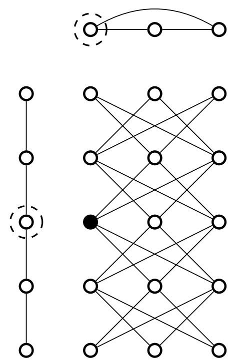
In these cases, after the first player moves, the second player becomes the new first player for the remaining graphs $G, \{u_0\}$ and $H, \{v_0\}$. In the first case, since u_0 is a winning move in G, the second player copied this vertex to steal as their winning strategy in $G, \{u_0\}$. Hence, now the second player becomes the new first player in the remaining graph $G, \{u_0\}$ and $H, \{v_0\}$ by stealing $\{u_0\}$ from G. Then we can observe that the new first player has winning strategies in the remaining graph $G, \{u_0\}$ and $H, \{v_0\}$ by selecting the vertex $\{u\}$ in G. In the second case, since u_0 was not a winning move in G, then $G, \{u_0\}$ and $H, \{v_0\}$ have the new first player's winning strategies.

Now, we will apply the above proof of the first player's winning strategy to get the new first player winning strategy in the remaining graph $G \times H, \{u_0, v_0\}$ by analyzing whether the vertex on the same row of $\{u_0\}$ or not depend on the cases. So, the new first player ensures to win in $G \times H$. In reality, since the new first player is the second player, ensuring the second player's winning strategy in $G \times H$. Therefore, at least one of the outcomes of G and H is $\mathcal{P}$, then the outcomes of $G \times H$ is $\mathcal{P}$.

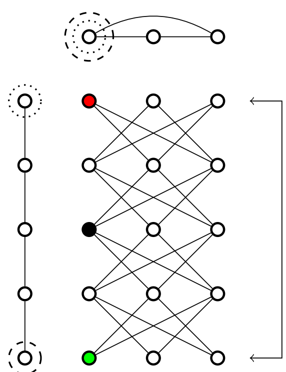
This completes the proof that $\mathcal{P}$ is absorbing for the outcomes of the tensor product of graphs for Closed Geodetic Game. □

Theorem 8. *The strong product $G \boxtimes H$ of two graphs G and H is $\mathcal{N}$ if and only if both G and H are $\mathcal{N}$.*

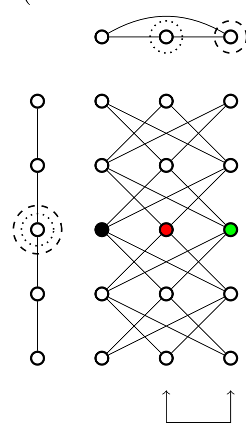
Proof. Let S, S_1 and S_2 be the sets constructed by the players for $G \boxtimes H$, $G \square H$ and $G \times H$, respectively (they contain the same vertices, that have been selected by the two players). Now, we assume by contradiction that the outcome of $G \boxtimes H$ is different from both $G \times H$ and $G \square H$ for Closed Geodetic



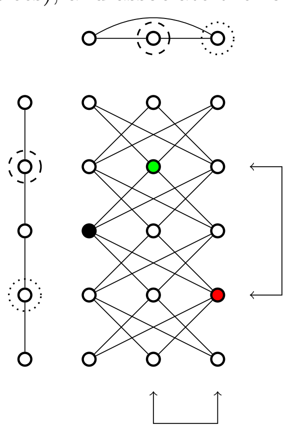
(a) The first move: applying the winning strategy on both graphs (the vertices circled with dashes).



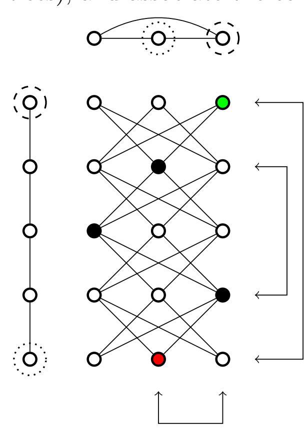
(b) If the second player plays in the same column (red, dotted vertices), answer on the same (green, dashed vertices), and associate the rows.



(c) If the second player plays in the same row (red, dotted vertices), answer on the same (green, dashed vertices), and associate the columns.

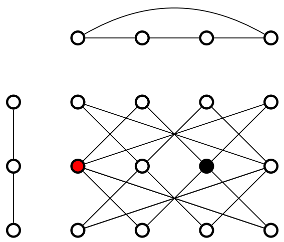


(d) If the second player plays distinctly (red, dotted vertices), answer on both graphs (green, dashed vertices), and associate the rows and columns.

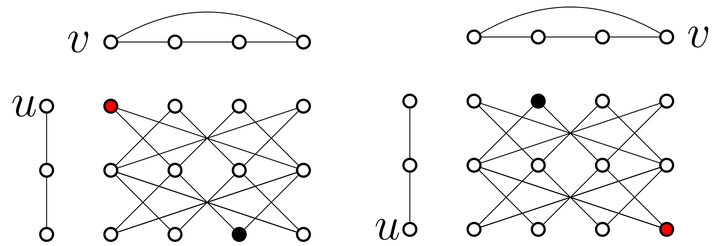


(e) After Figure 4d, apply the strategy on the associated rows and/or columns (here, columns), and add new associations (second player is red and dotted, first player is green and dashed).

Figure 4: A depiction of the first player's strategy on $P_5 \times K_3$, the Tensor product of two $\mathcal{N}$ graphs.



(a) When the first player applies the winning strategy of G, the second player copies to the vertex which is the first player plays in G. In H the second player answers by his winning strategy.



(b) When the first player does not apply the winning strategy in G, the second player answers by their winning strategies in $G, \{u\}$ and $H, \{v\}$.

Figure 5: A depiction of the second player's strategy on $P_3 \times C_4$, the Tensor product of an $\mathcal{N}$ graph and a $\mathcal{P}$ graph.

Game. Recall that $V(G \boxtimes H) = V(G \square H) = V(G \times H)$ and $E(G \boxtimes H) = E(G \square H) \cup E(G \times H)$. We first define the two following mappings for the image of the first player (resp. the second player) on $G \boxtimes H$ and the image of the second player (resp. the first player) on $G \square H$, denoted by I_1 and I_2, respectively. Those are, respectively, a function from $V(G \boxtimes H) \to V(G \square H)$ such that:

$$I_1((u,v)) = \begin{cases} (u',v') & \text{where } u' = u \text{ and } v' = v\text{, if } (u',v') \notin (S_1), \\ I_2((u',v')) & \text{otherwise;} \end{cases}$$

and a function from $V(G \square H) \to V(G \times H)$ such that:

$$I_2((u',v')) = \begin{cases} (u'',v'') & \text{where } u'' = u' \text{ and } v'' = v'\text{, if } (u'',v'') \notin (S_2), \\ I^{-1}((u'',v'')) & \text{otherwise;} \end{cases}$$

with $I = I_1.I_2 \,:\, V(G \boxtimes H) \to V(G \times H)$ a composition function. Note that $I_1^{-1}((u',v') = (u,v) \in G \boxtimes H$ and $I^{-1}((u'',v'') = (u,v) \in G \boxtimes H$ such that $(u,v) \notin (S)$ yet. Similarly, $I_1((u,v)) = (u',v') \in G \square H$ and $I_2((u',v')) = (u'',v'') \in G \times H$ such that either $(u',v') \notin (S_1)$ or $(u'',v'') \notin (S_2)$ yet.

We will now play Closed Geodetic Game on $G \boxtimes H$, and also apply the moves on both $G \square H$ and $G \times H$. The winning player will answer to their opponent's moves by playing the winning move in $G \square H$ if possible, and $G \times H$ otherwise. We will show that at least one of those two moves will be available.

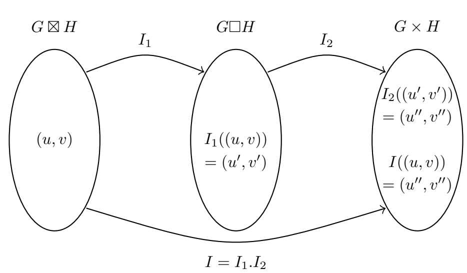


Figure 6: When the first player (resp. the second player) will play on $G \boxtimes H$ by their winning strategies, then the image of the first player (resp. the second player) will move in $G \square H$ and $G \times H$ and when the second player (resp. the first player) on either $G \square H$ or $G \times H$, then the image of the second player (resp. the first player) will move in $G \boxtimes H$.

We consider the winning player on both $G\Box H$ and $G \times H$ (since they have the same outcomes), and will show that they can emulate strategies for $G \boxtimes H$ using the previously-defined mappings.

Initial Move: The moves of the first player on $V(G \boxtimes H)$ are emulated via the image function $I_1((u_1, v_1)) = (u_1', v_1')$ and $I((u_1, v_1)) = (u_1'', v_1'')$ on $G\Box H$ and $G \times H$, respectively. The second player (resp. the first player) then responds to the first player's move (resp. the second player's move) by selecting the vertex, say (u_2', v_2') as per their strategy in $G\Box H$. Then this response from $G\Box H$ is emulated via the image function $I_2(u_2', v_2') = (u_2'', v_2'')$ in $G \times H$ and then the second player moves on $G \boxtimes H$ as the exact same moves by the inverse mapping $I_1^{-1}(u_2', v_2') = (u_2, v_2)$ in $G \boxtimes H$.

Response to the next moves in $G \boxtimes H$: After the first two moves, if $V(G \boxtimes H) = (S)$, then all games are over and we are done (the winning player on $G\Box H$ won on $G \boxtimes H$). If not, due to the winning player's strategy in $G \boxtimes H$, they can respond to the image of the other player's move by selecting a vertex in $G \boxtimes H$ as per their winning strategy on either $G\Box H$ or $G \times H$. Each move of the players is emulated via the image functions I_1 and I_2 on $G\Box H$ and then on $G \times H$. Assume that the losing player on $G\Box H$ (and $G \times H$) selected a vertex (u_k, v_k) in $G \boxtimes H$. The winning player then considers the answer (u_{k+1}, v_{k+1}) in $G\Box H$. If this vertex can be selected (*i.e.*, $(u_{k+1}, v_{k+1}) \notin (S_1)$), they select it and the game continues. Otherwise, they can select $I_2((u_{k+1}, v_{k+1}))$ in $G \times H$, since if we also had $(u_{k+1}, v_{k+1}) \in (S_2)$, then we would have $(u_k, v_k) \in (S)$ by construction of S, S_1 and S_2 and due to the fact that the winning player has a winning strategy in at least one of G and H. The winning move is then translated to $G \boxtimes H$ using I^{-1}.

Therefore, the winning player on $G\Box H$ and $G \times H$ plays as per their winning strategies on the three graph products through the two mappings. They always have an answer to their opponent, and thus will also play the last move of the game $G \boxtimes H$, ensuring a win.

Hence, $G \boxtimes H$ has the same outcomes as $G\Box H$ and $G \times H$, which proves the statement. □

3 She Loves Me-She Loves Me Not situations

She Loves Me-She Loves Me Not is one of the simplest combinatorial games: the two players alternate removing blossoms from a flower until it is empty. It has become a shorthand for easy, binary combinatorial games where every possible move from the initial position will be played. Note that, in those cases, the Sprague-Grundy values alternate between 0 (for an even number of moves) and 1 (for an odd number of moves), since no real strategy is involved. Some complex combinatorial games can be equivalent to She Loves Me-She Loves Me Not depending on the initial position. It is in particular the case for Closed Geodetic Game in some cases:

Proposition 9. *A game of Closed Geodetic Game on the complete graph K_n or a star $K_{1,n}$ will see all its vertices selected.*

Proof. Since every vertex in K_n is simplicial, each vertex of K_n will be selected in Closed Geodetic Game (Lemma 2). Hence, K_n is a She Loves Me-She Loves Me Not, and thus $\mathcal{G}(K_n) = n \bmod 2$.

For a star $K_{1,n}$ with center c and n leaves, note that every leaf will need to be selected (each leaf is a simplicial vertex). If n is even, then, the first player will select c and win (being the second player with an even number of remaining moves). If n is odd, then, there are two possible moves for the first player: either selecting c, in which case every vertex will be selected and they will lose, or selecting a leaf, in which case the second player will select c and leave the first player in a losing position where every vertex will be selected. Hence, $K_{1,n}$ is a She Loves Me-She Loves Me Not, and thus $\mathcal{G}(K_{1,n}) = 1 - (n \bmod 2)$. □

For other graphs, the game is not fully reducible to a She Loves Me-She Loves Me Not, but is still close since, under optimal play, we can determine exactly which vertices will be selected.

Proposition 10. *Let $n \geq 2$. A game of* Closed Geodetic Game *on the complete bipartite graph $K_{2,n}$ will see all the vertices of its larger part selected.*

Proof. Selecting the two vertices in the part of size 2 will end the game (the whole other part will be in the geodetic closure). Hence, no player wants to select the first of those two vertices: by doing so, they would let their opponent win. Thus, the losing player will always select vertices in the larger part, and the winning player also will, until the larger part has been wholly selected (note that selecting two vertices in one part prevents from selecting vertices in the other part, ensuring that once the first player picks a part, the second player will be able to enforce that it remains in that part). Doing so will also end the game. □

We can also compute the Sprague-Grundy values of complete bipartite graphs, for which one part will always be fully selected:

Proposition 11. *Let m and n be two integers such that $m, n \geq 2$. If m and n have the same parity, then, $\mathcal{G}(K_{m,n}) = 0$. Otherwise, $\mathcal{G}(K_{m,n}) = 2$.*

Proof. Note that, as for the proposition above, selecting two vertices in one part prevents from selecting vertices in the other part, and thus leaves a fixed number of moves that will all be played (selecting all the vertices in the part where two vertices have been selected).

Assume first that m and n are both even. The first player will select a vertex in a part, and the second player will select a vertex in the same part, ensuring that an even number of moves will remain. Hence, $K_{m,n}$ is $\mathcal{P}$ and thus $\mathcal{G}(K_{m,n}) = 0$.

Assume now that m and n are both odd. The first player will select a vertex in a part, and the second player will select a vertex in the other part. Now, whichever part the first player selects a vertex in, there will be an odd number of moves remaining, thus the second player will win. Hence, $K_{m,n}$ is $\mathcal{P}$ and thus $\mathcal{G}(K_{m,n}) = 0$.

Assume finally (without loss of generality) that m is even and n is odd. Let us analyze the options of the first player.

1. Selecting a vertex u_{odd} in the odd part. In this case, the second player has two answers:
   (a) selecting another vertex v_{odd} in the odd part leaves an odd number of moves, and thus $\mathcal{G}(K_{m,n}, \{u_{odd}, v_{odd}\}) = 1$;
   (b) selecting a vertex u_{even} in the even part, in which case there are again two options for the first player: selecting a vertex v_{odd} in the odd part leaves an odd number of moves and thus $\mathcal{G}(K_{m,n}, \{u_{odd}, u_{even}, v_{odd}\}) = 1$, while selecting a vertex v_{even} in the even part leaves an even number of moves and thus $\mathcal{G}(K_{m,n}, \{u_{odd}, u_{even}, v_{even}\}) = 0$. Hence, $\mathcal{G}(K_{m,n}, \{u_{odd}, u_{even}\}) = \text{mex}(\{0, 1\}) = 2$.

   This implies that $\mathcal{G}(K_{m,n}, \{u_{odd}\}) = \text{mex}(\{1, 2\}) = 0$.

2. Selecting a vertex u_{even} in the odd part. In this case, the second player has two answers:
   (a) selecting another vertex v_{even} in the even part leaves an even number of moves, and thus $\mathcal{G}(K_{m,n}, \{u_{even}, v_{even}\}) = 0$;
   (b) selecting a vertex u_{odd} in the odd part, and as seen above $\mathcal{G}(K_{m,n}, \{u_{odd}, u_{even}\}) = 2$.

   This implies that $\mathcal{G}(K_{m,n}, \{u_{even}\}) = \text{mex}(\{0, 2\}) = 1$.

Altogether, we have $\mathcal{G}(K_{m,n}) = \text{mex}(\{0, 1\}) = 2$. □

4 Symmetry strategies

The outcome of cycles can be fully derived from a symmetry strategy. Furthermore, we can also characterize the Sprague-Grundy value of a cycle:

Theorem 12. *For any positive integer n, $\mathcal{G}(C_n) = n \bmod 2$.*

Proof. Denote the vertices of the cycle by $u_0, \dots, u_{n-1}$, in cyclic order.

If $n = 2k$, then the second player has a winning strategy: when the first player selects a vertex u_i, they can select the opposite vertex (which is either u_{i+k} or u_{i-k}), which ends the game. Hence, $\mathcal{G}(C_n) = 0$.

Now, let $n = 2k+1$. The first player will select a vertex, say u_0. Note that, for $i \in \{0, \dots, k-1\}$, selecting u_{k+1+i} is equivalent as selecting u_{k-i} for the second move, as can be seen on Figure 7. Now, assume that the second player selects a vertex u_i for $i \in \{1, \dots, k\}$, then the first player can select u_{k+i}, which ends the game. Hence, $\mathcal{G}(C_n) = 1$ (since all possible first moves are equivalent, up to renaming the vertices, there is only one option for the first player). □

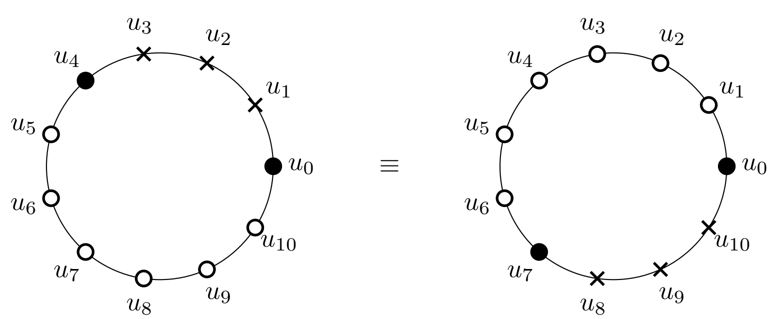


Figure 7: Selecting u_{k-i} and u_{k+1+i} are equivalent moves on $C_{2k+1}, \{u_0\}$ (here, with $k = 5$ and $i = 1$).

We can also characterize the Sprague-Grundy value of a cycle with one or two selected vertices:

Proposition 13. *For any positive integer k:*

- $\mathcal{G}(C_{2k}, \{u\}) = k$ *and* $\mathcal{G}(C_{2k+1}, \{u\}) = 0$ *for any vertex u;*
- $\mathcal{G}(C_{2k+1}, \{u_0, u_i\}) = k + 1 - i$ *for* $i \in \{1, \dots, k\}$ *(denoting the vertices by $u_0, \dots, u_{2k}$ in the cyclic order).*

Proof. Let k be a positive integer. Note that, for the first move, all the vertices are equivalent. Clearly, since $\mathcal{G}(C_{2k+1}) = 1$, we have $\mathcal{G}(C_{2k+1}, \{u\}) = 0$. For C_{2k} ($k \geq 2$), we have to analyze the moves of the second player. Denote the vertices by $u_0, \dots, u_{2k-1}$ in cyclic order, and assume, without loss of generality, that the first player selected u_0. Note that selecting u_{k+i} and u_{k-i} are equivalent moves for $i \in \{1, \dots, k-1\}$, and that selecting u_k is a winning move (and thus $\mathcal{G}(C_{2k}, \{u_0, u_k\}) = 0$). We can prove by induction on i that $\mathcal{G}(C_{2k}, \{u_0, u_{k-i}\}) = i$. If $i = 1$, then, any move of the first player will end the game, so $\mathcal{G}(C_{2k}, \{u_0, u_{k-1}\}) = 1$. Now, for $i \geq 2$, assume that the second player selected u_{k-i}. The first player can select u_{k-j} for $j < i$, so there are options with Sprague-Grundy values $1, \dots, i-1$ by induction hypothesis, and they can select u_k, which ends the game, so $\mathcal{G}(C_{2k}, \{u_0, u_{k-i}\}) \geq i$. Furthermore, the first player can select u_j for $j \in \{k+1, \dots, 2k-i\}$; those options end the game. Finally, the first player can select u_{2k-i+j} for $j \in \{1, \dots, i-1\}$; and we can clearly by a symmetry argument see that $\mathcal{G}(C_{2k}, \{u_0, u_{k-i}, u_{2k-i+j}\}) = \mathcal{G}(C_{2k}, \{u_0, u_{k-j}\}) = j$ by induction hypothesis. Hence, $\mathcal{G}(C_{2k}, \{u_0, u_{k-i}\}) = \text{mex}(\{0, 1, \dots, i-1\}) = i$ for $i \in \{1, \dots, k-1\}$, which implies that $\mathcal{G}(C_{2k}, \{u\}) = \text{mex}(\{0, \dots, k-1\}) = k$. We can apply a similar reasoning to obtain $\mathcal{G}(C_{2k+1}, \{u_0, u_i\}) = k + 1 - i$ for $i \in \{1, \dots, k\}$. □

Theorems 4 to 6 combined with Theorem 15 already give us the outcome for multidimensional grids, but we can also prove it with a symmetry strategy:

Theorem 14. *Closed Geodetic Game has outcome $\mathcal{N}$ for a multidimensional grid if and only if all its dimensions are odd.*

Proof. Let $n_1, \dots, n_k$ be the dimensions of the grid, we can associate each vertex to a k-vector of coordinates $(x_1, \dots, x_k)$ with $x_i \in \{1, \dots, n_i\}$. The central point of the grid is the point of coordinates $(\frac{n_1+1}{2}, \dots, \frac{n_k+1}{2})$. If this point corresponds to a vertex (that is, all dimensions are odd), then, the first player can select it. Now, if the second player selects the vertex of coordinates $(x_1, \dots, x_k)$, the first player can select the opposite vertex of coordinates $(n_1+1-x_1, \dots, n_k+1-x_k)$. This move is always possible, and thus the first player always has an answer to the second player's move, hence the game is $\mathcal{N}$. Conversely, if the central point is not a vertex (that is, any dimension is even), the first player can only select a vertex and the second player can select its opposite vertex, as described above, and hence the game is $\mathcal{P}$. □

Paths are another graph class where the outcome can be derived from a symmetry strategy. Going further, we hereby give a full characterization of the Sprague-Grundy values of paths (which complements the possible computation using the algorithm for trees from Araujo *et al.* [2]). Note that, while the value is expected, the proof itself is more involved.

Theorem 15. *For any positive integer n, $\mathcal{G}(P_n) = n \bmod 2$.*

Proof. Denote the vertices of P_n by $u_1, \dots, u_n$. Note that if the first two moves consisted in selecting two vertices u_i and u_j (with $1 \leq i < j \leq n$), the game on P_n with u_i and u_j selected is equivalent to the game on $P_{n-(j-i)}$ with u_i selected since none of the $j-i-1$ vertices between u_i and u_j can be selected anymore.

If n is even, then, the second player has a winning strategy by symmetry: whenever the first player selects u_i, they can always select u_{n+1-i}, which implies that P_n is $\mathcal{P}$ and thus that $\mathcal{G}(P_n) = 0$.

Assume now that n is odd. First, note that the first player has a winning strategy, by selecting the middle vertex and then applying the symmetry strategy described above. Hence, $\mathcal{G}(P_n) > 0$. We now prove that no option of P_n has Sprague-Grundy value 1. In order to do this, we will prove that every option of P_n has an option with Sprague-Grundy value 1. We first prove the following claim:

Claim 15.1. *For every nonnegative integer m, $\mathcal{G}(P_{4m+2}, \{u_{2m+1}\}) = 1$.*

Proof of Claim. The result is proved by induction. It trivially holds if $m = 0$, since only one move is possible. Assume now that the result holds for every $k < m$. Note that the first player has a winning move, by selecting u_{2m+2}, so $\mathcal{G}(P_{4m+2}, \{u_{2m+1}\}) > 0$. The other possible moves of the first player are to select u_x for either $1 \leq x \leq 2m$ or $2m+3 \leq x \leq 4m+2$. If $x \leq 2m$, then, the resulting position is P_{2m+1+x}, u_x; the second player can select either u_{2m+1} (resulting in P_{2x}, u_x) or u_{2m+3} (resulting in $P_{2x-2}, u_x \equiv P_{2x-2}, u_{x-1}$), depending on whether or not $2x$ is a multiple of 4. Both options are always available, and thus at least one of them is of the form $P_{4m'+2}, u_{2m'+1}$ with $m' < m$, thus having Sprague-Grundy value 1 by induction hypothesis. A similar reasoning can be applied if $x \geq 2m+3$, and thus every option of $P_{4m+2}, \{u_{2m+1}\}$ has an option with Sprague-Grundy value 1, which implies that $\mathcal{G}(P_{4m+2}, \{u_{2m+1}\}) = 1$. □

Going back to P_n, the non-winning options consist in selecting a vertex u_i with $i \neq \frac{n+1}{2}$. If $2i$ is a multiple of 4, then, the second player can select u_{n-i+2}, resulting in $P_{2(i-1)}, \{u_i\} \equiv P_{2(i-1)}, \{u_{i-1}\}$, which has Sprague-Grundy value 1 by Claim 15.1. Otherwise, $2i$ is not a multiple of 4, and the second player can select u_{n-i}, resulting in P_{2i}, u_i, which has Sprague-Grundy value 1 by Claim 15.1. Hence, every option of P_n has an option with Sprague-Grundy value 1, thus no option of P_n can have Sprague-Grundy value 1, which proves that $\mathcal{G}(P_n) = 1$. □

5 Sprague-Grundy values of block graphs and cacti

In their paper [2, Algorithm 2], Araujo *et al.* presented a linear-time algorithm for computing the Sprague-Grundy value of a given tree for Closed Geodetic Game. In order to do this, they use dynamic programing with subtrees that decrease in size at each step: for any given selected vertex u, each of the subtrees rooted at a neighbour of u are completely independent in Closed Geodetic Game, since by Lemma 3 selecting a vertex in a subtree will never impact any of the other subtrees. The base case happens when the subtree is either empty or a single vertex. Hence, it is possible to define a recursive algorithm computing, for each vertex of the tree, the Sprague-Grundy value of the subtrees rooted at each of its neighbours, and then apply the mex operation to the resulting set. Since each subtree is independent, and the intermediate values can be stored, the algorithm returns the result in linear-time.

In this section, we extend this idea to other tree-like graph classes: block graphs and cacti. Indeed, the building blocks of those graphs are simple graphs for which we can easily compute the Sprague-Grundy values: complete graphs and cycles. We will present polynomial-time algorithms for those classes, adapted from the trees' algorithm in [2]. Note that the building blocks have specific properties, and are close to the class of *geodetic graphs*, that is, graphs for which there is a unique shortest path between any pair of vertices (this class includes trees, block graphs, and cacti with only odd cycles); the only non-geodetic class is cacti with even cycles, for which the shortest paths between two vertices will share common articulation points and thus are almost as constrained.

5.1 Block graphs

A graph is a *block graph* if every biconnected component is a clique. Informally, we can see a block graph as a tree-like structure in which cliques are connected to each other by one of their vertices.

Theorem 16. *There is a linear-time algorithm computing the Sprague-Grundy value of a block graph for Closed Geodetic Game.*

Proof. The algorithm is based on dynamic programing. First, for a given block graph G and one of its vertices u, we present how to recursively compute $\mathcal{G}(G, \{u\})$, that is, we consider that the first player selects u. We will show that some clique vertices are equivalent for this first move, and that we can store the intermediary results to reuse them for other vertices (thus avoiding polynomial-time when applying the algorithm for all the non-equivalent vertices of G). Note that the base case is when the graph is a complete graph, since we can compute its Sprague-Grundy value using Proposition 9 (each selected vertex within a complete graphs flips the Sprague-Grundy value between 0 and 1).

There are two types of vertices in G: those that are articulation points between several cliques, and those that are not. First, assume that the selected vertex u is an articulation point, then we can compute $\mathcal{G}(G, \{u\})$ by using Lemma 3 and store the intermediate values for future computations.

Assume now that the selected vertex u is not an articulation point, hence u is a vertex in a given clique A of G. Note that selecting any other non-articulation point of A is equivalent to selecting u. Now, consider the components that are attached to A in G (that is, they share one vertex with A), denote them by $B_1, \dots, B_k$ and the articulation point between A and B_i by v_i. Note that selecting any other non-articulation point in A will not affect any of the B_i's, and that selecting v_i can only affect B_i. Hence, we can decompose the game into almost-independent subgraphs: $A \setminus \{v_1, \dots, v_k\}$ with the vertex u selected, and each component containing B_i with a leaf attached to v_i, which is selected too. This is depicted on Figure 8. Note that this decomposition works since A is a clique, and so all its vertices are at distance 1 from each other.

Now, we need to see how we can decompose the graph with one selected vertex u when another vertex v is selected. Again, if v is an articulation point, we can use Lemma 3. If v is also in A, then we do not need to decompose further. So, assume that v is in another clique B and is not an articulation point. We can apply the same principle of decomposition, after covering the vertices in the shortest paths between u and v. Note that if A and B are not adjacent cliques, then let C be a clique with two

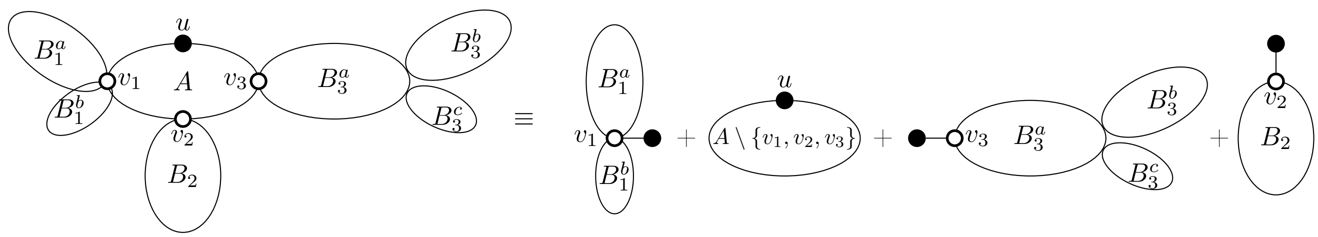


Figure 8: Selecting a vertex in a clique in a block graph allows us to decompose the graph as follows (here, each B_i^x is a clique).

vertices on the shortest path between u and v, two vertices of C are covered and the other ones will have no impact on A or B for the rest of the game. This decomposition is depicted on Figure 9.

Note that both decompositions have the property that a subgraph that is not a clique has at most one selected vertex. This covers all possible cases, and thus the algorithm will further decompose the input block graph until it is a disjoint sum of cliques, for which the Sprague-Grundy value can be computed in constant time. By using the nim-sum and the mex, we can then go back and compute the Sprague-Grundy value of the block graph. The algorithm ends in linear time for a given vertex, and allows us to store the intermediary results for future computations. In fact, the number of block components being linear, any future computation will see similar decompositions, and thus reuse the intermediary results, hence allowing the algorithm to remain linear-time when considering all possible first moves. □

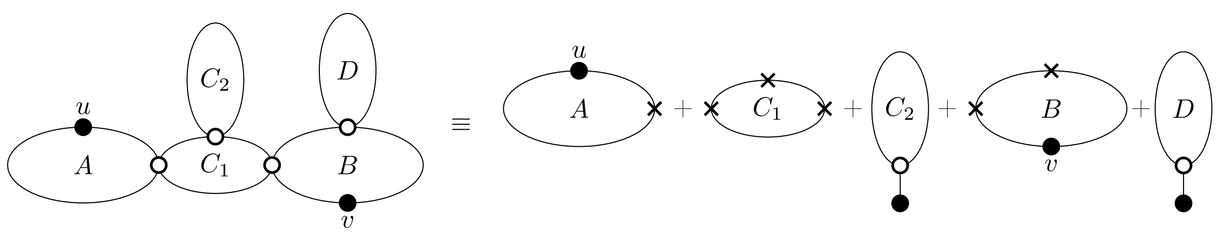


Figure 9: Selecting a second vertex in a block graph allows us to decompose it further.

5.2 Cacti

A *cactus* (plural: cacti) is a graph where every edge is in at most one simple cycle. Alternately, cacti have a tree-like structure in which cycles and trees are connected to each other by one of their vertices.

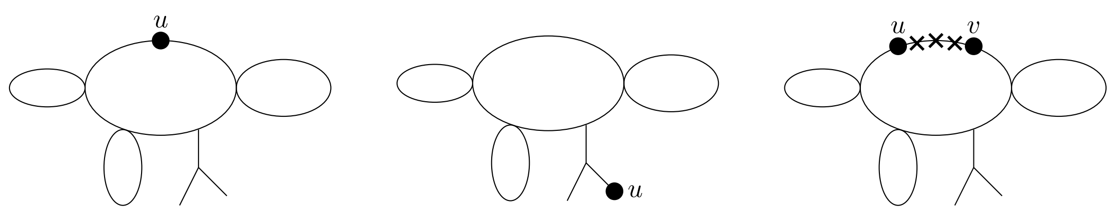
Theorem 17. *There is a quadratic-time algorithm computing the Sprague-Grundy value of a cactus for* Closed Geodetic Game*.*

Proof. As for block graphs, the algorithm is based on dynamic programing. Similarly, we are going to show how to recursively compute $\mathcal{G}(G, \{u\})$ for a given cactus G and selected vertex u by decomposing it into independent subcomponents; we can then repeat this computation for every vertex. However, unlike with block graphs, we may also have to consider $\mathcal{G}(G, \{u, v\})$, where u and v are two distinct selected vertices. We are also going to store the intermediary results in order to reuse them in future steps or iterations of the algorithm.

Let G be a cactus, and u and v be two vertices of G (u is the vertex selected by the first player, v the vertex selected by the second player). If G is a tree or a cycle, then, we can compute its Sprague-Grundy value using either the algorithm in [2] and Theorem 15 or Theorem 12 and Proposition 13, so assume G is neither a tree nor a cycle. If u (resp. v) is an articulation point, then we can compute $\mathcal{G}(G, \{u\})$ (resp. $\mathcal{G}(G, \{u, v\})$) using Lemma 3. So, assume that u (resp. v) is not an articulation point.

We identify three types of cactus graphs with either one or two selected vertices, and every further move will leave the remaining cactus to be further decomposed into subcomponents of those types. Those are depicted on Figure 10.

- A Type I cactus has a selected vertex u on a simple cycle (see Figure 10a);
- A Type II cactus has a selected vertex u on a leaf (see Figure 10b);
- A Type III cactus has two selected vertices u and v on the same simple cycle (see Figure 10c). Note that the vertices on a shortest path between u and v are covered in the geodetic closure, and thus cannot be selected.



(a) A Type I cactus. (b) A Type II cactus. (c) A Type III cactus.

Figure 10: The three types of cacti with one or two selected vertices into which we will decompose the cactus during the game.

Now, assume that we are in a Type I cactus. There are three possible kinds of moves.

1. Selecting a vertex v on the same simple cycle as u, and such that no articulation point is on a shortest path between u and v. In this case, we are simply left with a Type III cactus.
2. Selecting a vertex v on the same simple cycle C as u, and such that there are articulation points $a_1, \ldots, a_k$ separating C from $C_1, \ldots, C_k$ on a shortest path between u and v. In this case, we can decompose the cactus into the sum of a Type III cactus (the component containing C, with u and v selected) and Types I or II cacti (the components $C_1, \ldots, C_k$, with each of $a_1, \ldots, a_k$ selected). Note that the Type III cactus might be empty if C is even and v is diametrically opposed to u, since there will be two shortest paths between u and v covering all vertices of C.
3. Selecting a vertex v not in the simple cycle containing u. In this case, the shortest paths between u and v will go through at least one articulation point, so we can decompose the cactus following those articulation points. Let C_u be the component containing u, C_v be the component containing v, denote by x and y the articulation points on a shortest path from u to v separating C_u and C_v from the rest of the shortest path. The decomposition we obtain is the sum of two Types III cacti (C_u and C_v with u and x, and v and y selected, respectively) and any Types I, II and III cacti that are separated along the shortest path with their articulation point with the path also selected.

This is depicted on Figure 11. As we can see, playing on a Type I cactus leaves either a Type III cactus or the disjoint sum of Types I, II and III cacti, so we can recursively apply the algorithm and apply the nim-sum and the mex to compute $\mathcal{G}(G, \{u\})$.

Types II and III work similarly. Without detailing as much, here are the possible moves on a Type II cactus:

1. Selecting a vertex in the same subtree, leaving the disjoint sum of a Type II cactus and a tree with two selected vertices.

2. Selecting a vertex elsewhere, leaving the disjoint sum of Types I, II and III cacti and a tree with two selected vertices.

And the same can be done for a Type III cactus, with decompositions similar to the Type I:

1. Selecting a vertex w in the same simple cycle such that there is no articulation point in a shortest path between u and w and between v and w, leaving a Type III cactus.
2. Selecting a vertex in the same simple cycle but with articulation points in the shortest paths, leaving the disjoint sum of a Type III cactus and Types I and II cacti disconnected from it.
3. Selecting a vertex not in the same simple cycle, again leaving the disjoint sum of two Types III cacti, and the Types I, II and III cacti along the shortest paths.

Again, some Types III cacti might actually be empty games along the way, where every vertex is collected, in the case of an even simple cycle with two diametrically opposed vertices selected (or articulation points on a shortest path between two selected vertices).

All decompositions have the property that a component of the disjoint sum is either a tree, a cycle, or has at most two selected vertices. Furthermore, note that any sequence of decompositions will lead to a disjoint sum of trees, cycles and empty graphs, for which we can individually compute the Sprague-Grundy value in linear or constant time, and then go back and use the nim-sum and mex to compute $\mathcal{G}(G, \{u\})$. The number of different subcomponents is linear, and thus the algorithm ends in quadratic time. Since we can store the Sprague-Grundy value of each component, further applications of the algorithm with different starting vertices will reuse some of the computations, thus ensuring a quadratic-time algorithm even when considering all possible first moves. □

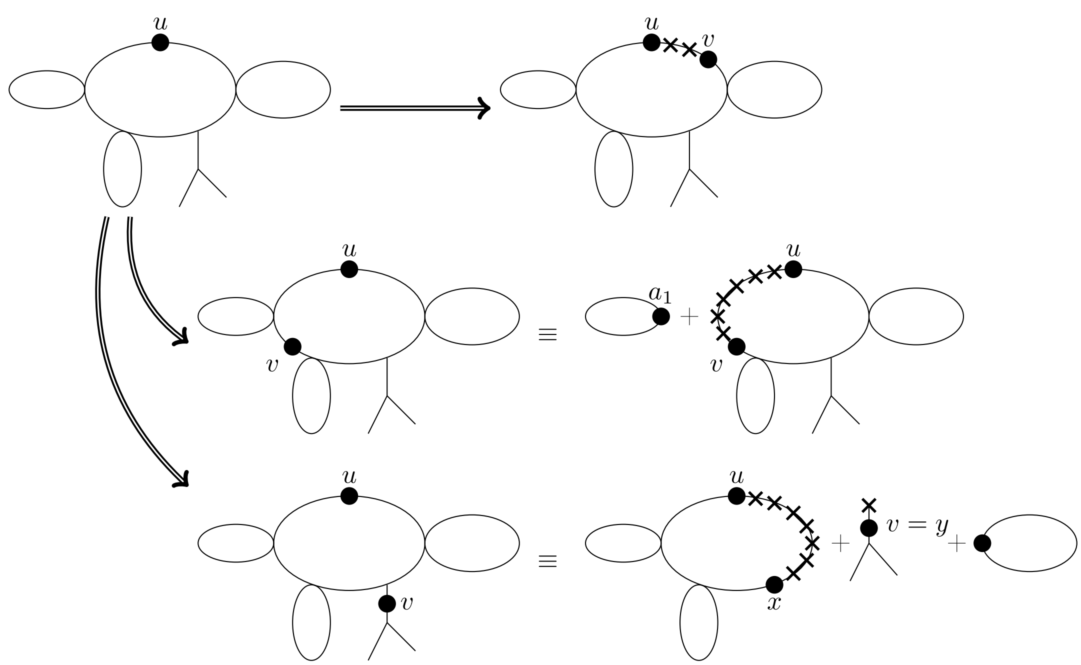


Figure 11: The three possible moves on a Type I cactus.

6 Future research

In this paper, we explored Closed Geodetic Game in three directions: general results such as graph operations (Cartesian product) and decomposition through articulation points, characterization of outcomes and Sprague-Grundy values, and polynomial-time algorithms based on dynamic programming for constrained graph classes. However, many more general graph classes remain open, algorithmically as well as structurally.

A first interesting research direction would be to fully characterize the graphs for which Closed Geodetic Game is a She Loves Me-She Loves Me Not game, or those that are close to such a situation. We already exhibited complete graphs and stars as She Loves Me-She Loves Me Not games in Section 3, but there might be other very structured classes with many simplicial vertices that are in the same situation.

Another idea would be to further the work initiated in Section 4 and to define a type of graph symmetry adapted to Closed Geodetic Game, that would allow us to obtain the outcomes of other structured graph classes. Expansions of cycles could be good candidates.

As explained in Section 5, the dynamic programing framework we used for block graphs and cacti requires strong constraints for shortest paths between vertices, which do not hold for more general graph classes such as outerplanar graphs. A research direction would be to try to express weaker constraints allowing a dynamic programing framework based on decomposition into independent subcomponents to apply for other classes. Outerplanar graphs are a natural candidate, since the geodetic set problem is polynomial for them but NP-hard for planar graphs.

Finally, other graph products than those studied in Section 2 could be explored, such as the lexicographical, modular or rooted products.

Acknowledgements.

The authors would like to thank Sagnik Sen, who initiated this research project; as well as the universities' libraries, without which the seminal paper [7] wouldn't be obtainable.

References

[1] Albert, M., Nowakowski, R., Wolfe, D.: Lessons in play: an introduction to combinatorial game theory. AK Peters/CRC Press (2019)

[2] Araújo, S.N., Brito, J.M., Folz, R., de Freitas, R., Sampaio, R.M.: Graph convexity impartial games: Complexity and winning strategies. Theoretical Computer Science **998**, 114534 (2024)

[3] Atici, M.: Computational complexity of geodetic set. International journal of computer mathematics **79**(5), 587–591 (2002)

[4] Benesh, B.J., Ernst, D.C., Meyer, M., Salmon, S., Sieben, N.: Impartial geodetic building games on graphs. arXiv preprint arXiv:2307.07095 (2023)

[5] Berlekamp, E.R., Conway, J.H., Guy, R.K.: Winning ways for your mathematical plays. AK Peters/CRC Press (2001–2004)

[6] Buckley, F., Harary, F., Quintas, L.: Extremal results on the geodetic number of a graph. Scientia **2A**, 17–26 (1988)

[7] Buckley, F., Harary, F.: Closed geodetic games for graphs. Congr. Numer **47**, 131–138 (1985)

[8] Buckley, F., Harary, F.: Geodetic games for graphs. Quaestiones Mathematicae **8**(4), 321–334 (1985)

[9] Buckley, F., Harary, F.: Distance in graphs. New York: Addison and Wesley (1990)

[10] Chakraborty, D., Das, S., Foucaud, F., Gahlawat, H., Lajou, D., Roy, B.: Algorithms and complexity for geodetic sets on planar and chordal graphs. In: 31st International Symposium on Algorithms and Computation (ISAAC 2020). Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2020)

[11] Chakraborty, D., Gahlawat, H., Roy, B.: Algorithms and complexity for geodetic sets on partial grids. Theoretical Computer Science **979**, 114217 (2023)

[12] Conway, J.H.: On numbers and games. CRC Press (2000)

[13] Ekim, T., Erey, A., Heggernes, P., van't Hof, P., Meister, D.: Computing minimum geodetic sets of proper interval graphs. In: Latin American Symposium on Theoretical Informatics. pp. 279–290. Springer (2012)

[14] Fraenkel, A.S., Harary, F.: Geodetic contraction games on graphs. International Journal of Game Theory **18**, 327–338 (1989)

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

[16] Harary, F.: Convexity in graphs: achievement and avoidance games. In: North-Holland Mathematics Studies, vol. 87, p. 323. Elsevier (1984)

[17] Harary, F., Loukakis, E., Tsouros, C.: The geodetic number of a graph. Mathematical and Computer Modelling **17**(11), 89–95 (1993)

[18] Harary, F., Nieminen, J.: Convexity in graphs. Journal of Differential Geometry **16**(2), 185–190 (1981)

[19] Haynes, T.W., Henning, M., Tiller, C.A.: Geodetic achievement and avoidance games for graphs. Quaestiones Mathematicae **26**(4), 389–397 (2003)

[20] Mezzini, M.: Polynomial time algorithm for computing a minimum geodetic set in outerplanar graphs. Theoretical Computer Science **745**, 63–74 (2018)

[21] Nečásková, M.: A note on the achievement geodetic games. Quaestiones Math. **12**, 115–119 (1988)

[22] Sieben, N.: Impartial hypergraph games. The Electronic Journal of Combinatorics pp. P2–13 (2023)

[23] Siegel, A.N.: Combinatorial game theory, vol. 146. American Mathematical Soc. (2013)

[24] Sprague, R.: Über mathematische kampfspiele. Tohoku Mathematical Journal, First Series **41**, 438–444 (1935)