Morris Games are (computationally) hard

Stefano Leucci[1,*], Michael Piccirilli[1]

[1]Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila, Italy.

Abstract

Morris Games are two-player, perfect-information board games played on planar graphs in which vertices represent positions that can host the players' pieces, called *stones*, *men*, or *cows*. Players alternate placing and moving stones trying to form *mills*, i.e., three consecutive stones of their own color along a path of three vertices along a straight line in the embedding. Whenever a player forms a mill, they choose one of their opponent's stones to be removed from the board, where stones not involved in any mill have to be removed first. The objective is either to reduce the opponent to fewer than three stones, making it impossible for them to form further mills, or to reach a configuration in which the opponent has no remaining legal moves.

While the game ends in a draw under perfect play in many variants when played on common boards (often found on the reverse side of a checkerboard), its computational complexity on generalized boards has remained open. In this paper, we prove that determining whether a given player can force a win from an arbitrary configuration is both NP-hard and co-NP-hard. This holds even when many topological and geometric features of classical boards are retained; namely, the result applies to underlying graphs that are connected and bipartite, have vertex degrees between 2 and 4, and admit planar rectilinear embeddings. Furthermore, the hardness remains regardless of whether mills are required to be collinear.

Keywords

Morris Games, Nine Men's Morris, Board games, Computational complexity

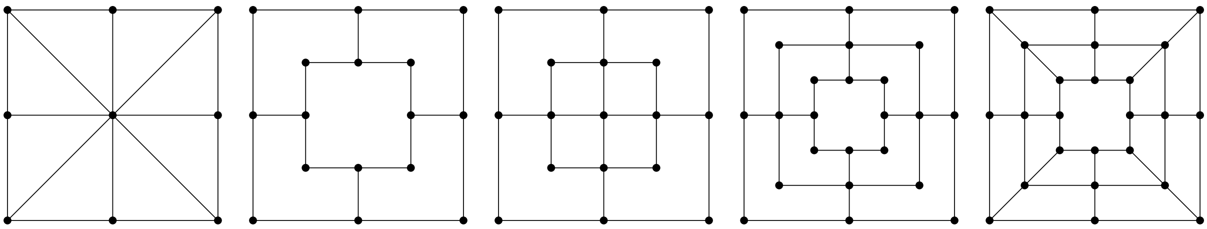
1. Introduction

Morris Games are classical two-player, perfect-information games played on a board consisting of distinguished points connected by line segments. Each player starts with a number of *stones* (also called *men* or *cows*) in their hand. Players take turns strategically placing stones on empty points of the board and, once they have no more stones in hand, they slide one of their already placed stones across a segment. When a player is reduced to three stones, they are no longer restricted to sliding, and they can instead *fly* one of their stones to any unoccupied point. Players try to get three stones of their color to form a *mill*, i.e., to occupy three points of the board that are connected by a straight line. Upon *closing* a mill, the player permanently removes one of the opponent's stones from play, where stones that are not part of any mill must be removed first. A player wins by removing all but two of their opponent's stones or by reaching a configuration in which their opponent has no legal move.

There are many variants of Morris Games, which differ mainly by the shape of the board and by the number of starting stones of each player. The most common variant is arguably Nine Men's Morris (also known as *the mill game* or *cowboy checkers*) which, as the name suggests, is played with nine stones per player on the board shown in Figure 1 (d). This variant has a long history and could have been played as early as 1400 BCE, as indicated by some boards carved into the roofing of the Ramesseum, the mortuary temple of Ramesses II in Kurna, Egypt, although the exact dating of these boards is disputed [1, 2]. Gaming pieces and what have been suggested to be fragments of a Nine Men's Morris board were also found in a Viking burial chamber in Balnakeil, and were radiocarbon dated to around 680–860 CE [3, 4]. The game was popular in ancient Rome and during the medieval period [1, 5], and the word *morris* itself comes from the Latin *merellus*, meaning "gaming piece" [6].

ICTCS 2026: 27th Italian Conference on Theoretical Computer Science, September 7–9, 2026, Udine, Italy.

*Corresponding author.
stefano.leucci@univaq.it (S. Leucci); michael.piccirilli@student.univaq.it (M. Piccirilli)
https://www.stefanoleucci.com (S. Leucci)
0000-0002-8848-7006 (S. Leucci); 0009-0006-9676-4911 (M. Piccirilli)



(a) (b) (c) (d) (e)

Figure 1: Boards of different Morris Games: (a) Three Men's Morris, (b) Five and Six Men's Morris, (c) Seven Men's Morris, (d) Nine Men's Morris, and (e) Twelve Men's Morris/Morabaraba.

Other variants include *Three Men's Morris*, which is played with three stones per player on the board shown in Figure 1 (a); *Twelve Men's Morris*, also known as *Morabaraba*, is played with twelve stones per player on the board shown in Figure 1 (e);[1] *Five Men's Morris* and *Six Men's Morris*, which are played with five and six stones per player, respectively, on the board shown in Figure 1 (b); *Seven Men's Morris*, which is played with seven stones per player on the board shown in Figure 1 (c);[2] and *Lasker Morris*, which is played with ten stones per player on the same board as Nine Men's Morris and allows players to slide stones even if they still have unplaced stones.

A reference to Three Men's Morris appears in Ovid's *Ars Amatoria* (circa 4 CE), which describes a game with twelve positions and three pieces per player in which the goal is to align the pieces in a straight line.[3]

Related work. Due to the game's popularity, there has been considerable attention devoted to devising good strategies for the players. Gasser [8] provided a strong solution for Nine Men's Morris, i.e., a perfect strategy from any configuration that can be reached from the initial empty board, and showed that the game results in a draw upon optimal play by both players. Similar results were obtained by Stahlhacke for Lasker Morris [9]. Ultrastrong solutions[4] and strong solutions for an extended set of starting configurations have later been found for Nine Men's Morris, Morabaraba, and Lasker Morris [10]. Loewer [11] showed that perfect play in Nine Men's Morris results in a draw even in some alternative rulesets with different capture mechanics. Several strategies based on machine learning, statistical analysis, state-space exploration, or heuristic approaches have also been proposed for Morris Games [12, 13, 14, 15, 16, 17, 18].

Despite such attention, Morris Games appear not to have been analyzed under the computational complexity lens. This is at odds with many other two-player board games of perfect information, such as Chess [19, 20], Go [21, 22], Checkers [23], Othello [24], TwixT [25], and Peg Duotaire [26], just to name a few, and was explicitly mentioned as an interesting problem more than fifteen years ago in [27]. Morris Games are also similar in spirit to Token Positional games, i.e., Maker-Breaker games in which two players take turns placing or moving tokens on the vertices of an input hypergraph, with the goal of filling all vertices of some hyperedge with tokens of their color. These games are known to be in EXPTIME, and they are PSPACE-hard even on 4-uniform hypergraphs, while a version allowing only token slides (between vertices belonging to the same hyperedge) is EXPTIME-complete [28].

Notably, a common theme among games and puzzles that people tend to find interesting is that their associated decision problems tend to be NP-hard (or harder) [27, 29].

[1] In this variant, if all points of the board become occupied before any sliding occurs, the game ends in a draw.
[2] In Five and Six Men's Morris, flying is not allowed.
[3] The original text is "*Est genus, in totidem tenui ratione redactum scriptula, quot menses lubricus annus habet: parva tabella capit ternos utrimque lapillos, in qua vicisse est continuasse suos*", which roughly translates to: there is a board with as many squares as months in the year, there are three pieces on each side, one wins by moving their pieces into a straight line. See, e.g., [7].
[4] Ultrastrong solutions are strategies increasing the chances to achieve more than the game-theoretic value when an opponent makes a mistake.

Our results. In this paper, we prove that Morris Games also share this feature. More precisely, we show that the problem of deciding whether a given player can force a win from an arbitrary configuration is both NP-hard and co-NP-hard.

As is customary, in order to meaningfully study the computational complexity of a game, one needs to consider an infinite class of instances. Although one can easily generalize Morris Games by modeling the board as an arbitrary graph, this naive generalization discards the geometric component of the game. Our reduction preserves many topological and geometric aspects common to many classical boards. Namely, the results hold even if the graph is required to be connected and bipartite, all vertex degrees are between 2 and 4 (inclusive), and there exists a planar rectilinear embedding in which the edges incident to any vertex of degree 2 are orthogonal. Moreover, these results hold regardless of whether mills are required to be collinear in the embedding (i.e., to lie on the same vertical or horizontal line) and regardless of whether repeating a configuration results in a draw, a rule that is sometimes in place.

2. Generalized Morris Games

Before discussing our hardness results, we define the generalized version of Morris Games used throughout the paper. We call this version GENERALIZED MORRIS GAME and we denote it by GMG for short. The two players will be referred to as *Blue* and *Red*.

In GMG, the board is modeled as an undirected graph $G = (V, E)$ along with a planar straight-line embedding of G.[5] The vertices in V represent the possible positions of the stones, while the edges in E model the connecting segments, i.e., they represent legal slides between adjacent positions.

A configuration of the game can be described by a quintuple $\langle B, R, b, r, p\rangle$, where $B, R \subseteq V$ are disjoint sets of vertices occupied by Blue and Red stones, respectively, b (resp. r) is the number of stones of Blue (resp. Red) that are yet to be placed on the board, and $p \in \{\mathrm{Blue}, \mathrm{Red}\}$ represents the next player to move. Vertices in $V \setminus (B \cup R)$ are empty.

Consider a generic configuration and suppose that the next player to move is Blue. We now describe the possible actions that Blue can take. The moves available to Red (on their turn) are defined symmetrically. Blue performs one of the following actions: if $b > 0$, then they place one of their stones on an empty vertex $u \in V \setminus (B \cup R)$ of choice. This decreases b by 1 and updates the set B to $B \cup \{u\}$. Otherwise, if $b = 0$ and $|B| > 3$, Blue chooses an edge $(u, v) \in E$ with $u \in B$ and $v \in V \setminus (B \cup R)$, and *slides* the stone on u along (u, v). Finally, if $b = 0$ and $|B| = 3$, Blue can choose any pair of vertices u and v, with $u \in B$ and $v \in V \setminus (B \cup R)$, and *fly* the stone initially on u to vertex v. In both of the latter two cases, B is updated to $(B \setminus \{u\}) \cup \{v\}$.

A mill is defined as a simple path $\pi = \langle u, v, w\rangle$ of length two in G, such that u, v, and w are collinear in the embedding of G and are occupied by stones of the same color c. If c is Blue (resp. Red), we say that π is a Blue (resp. Red) mill. A move *closes* a mill if it places or moves a stone to a vertex which becomes part of a mill as a consequence of the move.

If Blue's move closes at least one mill, then Blue chooses an opponent's stone on some vertex $x \in R$ to be permanently removed from play (i.e., R is updated to $R \setminus \{x\}$) with the following restriction: if x is part of a mill, its stone can only be removed if *all* vertices in R are each part of some mill. After a player places or moves their stone, and possibly removes an opponent's stone, the turn passes to the other player (i.e., p is updated from Blue to Red or vice versa).

A configuration is winning for Blue if $p = \mathrm{Red}$ and at least one of the following conditions holds: (i) $r = 0$ and $|R| < 3$, or (ii) Red has no legal move. The winning configurations for Red are defined symmetrically. We study the following decision problem: given an instance of GMG consisting of a graph along with its planar straight-line embedding, a starting configuration C, and a player $p^* \in \{\mathrm{Blue}, \mathrm{Red}\}$, determine whether p^* can force a win from C.

We point out that, in some rulesets, the game ends in a draw as soon as a board configuration is repeated during play [8], while other rulesets allow the game to possibly continue indefinitely. The

[5] A planar straight-line embedding of a graph G is a crossing-free drawing in the Euclidean plane in which each vertex corresponds to a distinct point, and edges are straight-line segments between (the points corresponding to) their endvertices.

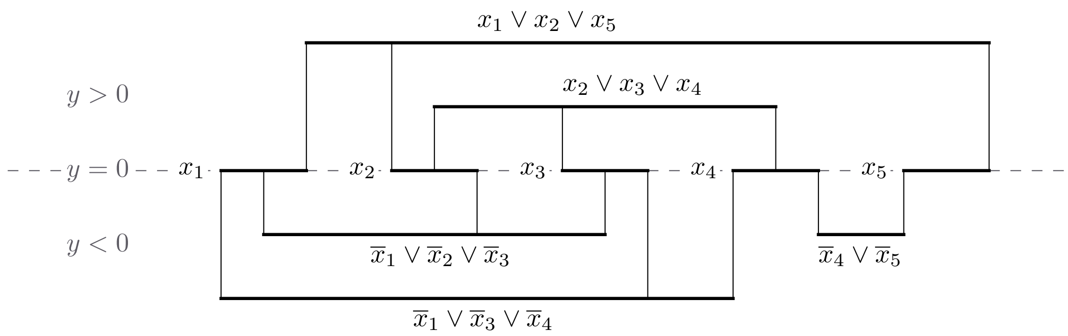


Figure 2: A rectilinear embedding for an instance of PMR3-SAT with $\varphi = (x_1 \vee x_2 \vee x_5) \wedge (x_2 \vee x_3 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_4 \vee \overline{x}_5) \wedge (\overline{x}_1 \vee \overline{x}_3 \vee \overline{x}_4)$.

above definitions do not take into account repeated configurations, yet the results described in the sequel hold regardless of whether the draw-by-repetition rule is in place.

3. Our hardness results

We are now ready to prove our hardness results, which are obtained through a Karp reduction from the PLANAR MONOTONE RECTILINEAR 3-SAT (PMR3-SAT) problem, which is known to be NP-complete [30].

In PMR3-SAT, we are given a boolean formula φ in conjunctive normal form. The formula has n variables $x_1, x_2, \dots, x_n$ and is a conjunction of m clauses $C_1, \dots, C_m$, where each clause is a disjunction of at most three literals and is either *positive* or *negative*: all literals in a positive (resp. negative) clause must be positive (resp. negative). Moreover, we are given a *rectilinear embedding* of φ: each variable corresponds to a segment on the horizontal line $y = 0$, each positive (resp. negative) clause corresponds to some horizontal segment in the half-plane $y > 0$ (resp. $y < 0$), and each occurrence of a literal ℓ in a clause C_j corresponds to a vertical segment with one endpoint on the segment of the variable associated with ℓ and the other endpoint on the segment of C_j. All such segments are pairwise non-intersecting, except for the endpoints of the vertical segments connecting variables to clauses. See Figure 2 for an example. The goal of the problem is to decide whether there exists an assignment of truth values to the variables $x_1, x_2, \dots, x_n$ that satisfies φ.

We restrict ourselves to instances of PMR3-SAT in which each variable occurs at most three times as a positive literal and at most three times as a negative literal. This can be assumed without loss of generality via a standard variable splitting argument and does not affect the NP-completeness of the problem, as we discuss in the full version of the paper.

Given an instance of PMR3-SAT, we construct an initial configuration of GMG in which Blue can force a win if and only if φ is satisfiable. In such a configuration, all the players' stones are already placed on the board.[6]

The board resulting from our reduction is organized into two areas: a *formula area* and a *threat area*. The formula area is built from the PMR3-SAT embedding by replacing each variable segment with a corresponding *variable gadget* and each clause segment with a corresponding *clause gadget*.[7] A vertical segment of PMR3-SAT connecting a variable x_i to a clause C_j corresponds to an edge connecting a vertex in x_i's gadget to a vertex in C_j. The formula area contains only Blue stones, and the Blue player is expected to slide one of these stones at each turn. Moreover, while variable and clause gadgets can interact by closing cross-gadget mills, each individual stone will be confined to its starting gadget.

The threat area contains $T = 6n + m$ *threat gadgets*, whose role is to force Blue to close a new mill in each of their first T moves. There are three states in which each threat gadget can be: *active*, *primed*, or *defused* states. Red can slide a stone in an active threat gadget to transition it to the primed state.

[6] As a consequence, the reduction also works without any modifications in the Lasker Morris ruleset.

[7] A gadget is a localized sub-structure (consisting of positions, connections among them, and a suitable placement of stones) that allows a specific behavior to be simulated, or a constraint of the PMR3-SAT problem to be enforced.

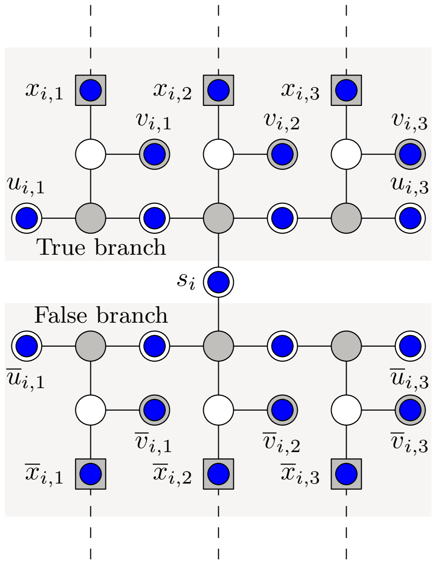


Figure 3: The variable gadget corresponding to variable x_i in its initial configuration. Square vertices can be made part of a mill if and only if their branch is activated (by sliding the stone on s_i) and allow the variable gadget to interact with the clause gadgets for the clauses containing x_i or $\overline{x}_i$ by closing a mill involving their incident dashed edges.

Initially, there is exactly one gadget in the primed state. If Red's turn starts when a threat gadget is primed, then Red will be able to force a win by repeatedly playing in a suitable gadget called *punisher gadget*. Hence, Blue must defuse the primed threat gadget by closing a mill and removing some Red stone from the gadget. If Blue manages to defuse all T threat gadgets (by closing T mills), then they will be able to force a win.

Our reduction ensures that the winning player wins by reaching a board configuration in which their opponent has no legal move. Since neither player is reduced to having exactly three stones, we do not have to be concerned with *flying* stones. Moreover, no board configuration will ever be repeated; hence, our reduction is not affected by whether repetitions result in a tie.

We now describe the gadgets in each area in more detail.

3.1. Formula area gadgets

Variable gadgets. Each variable x_i of the PMR3-Sat instance is modeled by the variable gadget shown in Figure 3. The gadget consists of a central vertex s_i, which is connected to two mirrored copies of a subgraph that we refer to as a *branch*. More precisely, the branch at the bottom (resp. top) of s_i will be referred to as the *false branch* (resp. *true branch*) of x_i.

Blue plays the gadget by sliding the stone initially on s_i downward into the false branch, or upward into the true branch. In the former (resp. latter) case, we say that the false (resp. true) branch is activated. Observe that any such move closes a mill, and how these are the only two possible moves that allow Blue to close the first mill within the gadget.

If Blue needs to close a mill after their every move in the gadget, as will be the case, then the next five moves within the gadget are forced, up to some possible reorderings. Immediately after the true branch is activated, its configuration is that shown (rotated by 90 degrees due to space constraints) in Figure 4 (a). From this configuration, three mills can be independently closed by sliding the Blue stones on $u_{i,1}$, $v_{i,2}$, and $u_{i,3}$, each along its only incident edge. Figure 4 (b) shows the configuration in which all such slides have been performed. After sliding the stone on $u_{i,1}$ (resp. $u_{i,3}$), it is possible to close one more mill by also sliding the stone on $v_{i,1}$ (resp. $v_{i,3}$). Once all the above moves have been performed, the true branch is in the configuration shown in Figure 4 (c), and the gadget does not allow for any more mills to be closed by sliding a single stone initially contained within the gadget. Observe how $x_{i,1}$, $x_{i,2}$, and $x_{i,3}$ now contain a Blue stone which is part of a vertical mill (in the original orientation

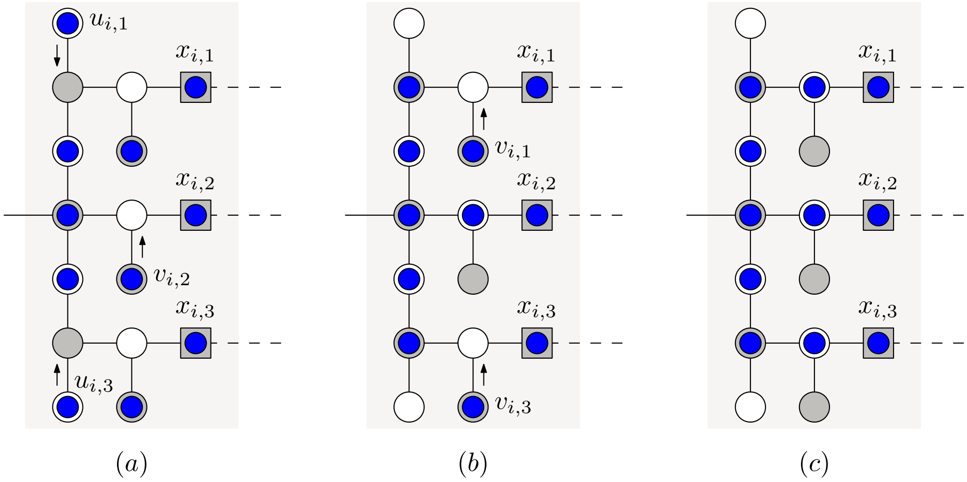


Figure 4: A sequence of mill-closing moves enabled by activating the true branch of x_i's gadget. Due to space constraints, only the true branch is depicted, and it is rotated by 90 degrees clockwise.

of the gadget).

The case in which the false branch is activated is symmetric and allows for five subsequent moves within the gadget, resulting in three disjoint vertical mills involving vertices $\overline{x}_{i,1}$, $\overline{x}_{i,2}$, and $\overline{x}_{i,3}$.

The rectilinear embedding of the PMR3-Sat instance guides the embedding of the variable gadgets in our GMG instance: we embed x_i's gadget in place of the horizontal segment corresponding to x_i.

Clause gadgets and clause–variable connections. Each clause C_j of the PMR3-Sat instance is modeled by a clause gadget, which is either the one shown in Figure 5 (a) or in Figure 5 (b) depending, respectively, on whether C_j is positive or negative.

Let $x_{j_1}, \ldots, x_{j_k}$ be the variables corresponding to the literals of C_j (recall that $2 \leq k \leq 3$) with $j_1 < \cdots < j_k$. The clause gadget consists of a vertex c_j that contains a Blue stone and is adjacent to three empty *literal vertices* $\ell_{j,1}$, $\ell_{j,2}$, and $\ell_{j,3}$. Vertex $\ell_{j,h}$ is associated with variable x_{j_h} for $h = 1, \ldots, k$ (while $\ell_{j,3}$ might not be associated with any variable).

The rectilinear embedding of the PMR3-Sat instance guides how we embed the clause gadget in our GMG instance, and how we connect it to the variable gadgets. When C_j is positive, we embed the gadget so that $\ell_{j,h}$ is vertically aligned with a vertex among $x_{j_h,1}$, $x_{j_h,2}$, and $x_{j_h,3}$ of the variable gadget for x_{j_h}. More precisely, we choose $x_{j_h,k}$ if the vertical segment corresponding to literal x_{j_h} of C_j is the k-th leftmost positive vertical segment intersecting (the segment of) variable x_{j_h} in the embedding of PMR3-Sat. When C_j is negative, the situation is symmetric: we vertically align $\ell_{j,h}$ with $\overline{x}_{j_h,k}$ if the vertical segment corresponding to literal $\overline{x}_{j_h}$ of C_j is the k-th leftmost negative vertical segment intersecting (the segment of) variable $\overline{x}_{j_h}$ in the embedding of PMR3-Sat.

The gadget is meant to be played by Blue. If C_j is positive (resp. negative) and the true branch (resp. false branch) of some variable gadget x_{j_h} has been activated, then Blue is able to close a mill by sliding the stone on c_j to $\ell_{j,h}$. After this move, no single slide of this stone can close any additional mill. Conversely, if all variable gadgets of $x_{j_1}, \ldots, x_{j_k}$ have not been activated yet, or their activated branch corresponds to a truth value that does not satisfy the corresponding literal in C_j, then there is no way of sliding the stone on c_j to close a mill.

3.2. Threat area gadgets

The punisher gadget. The punisher gadget is used in conjunction with threat gadgets to force the Blue player to close a mill on each of their first $T = 6n + m$ turns. This is enforced by guaranteeing that, whenever Blue does not obey the above prescription, Red can force a win.

The threat gadgets will be described in detail shortly, but the only relevant property needed to explain the operation of the punisher gadget is that threat gadgets do not contain any Blue stones. In other

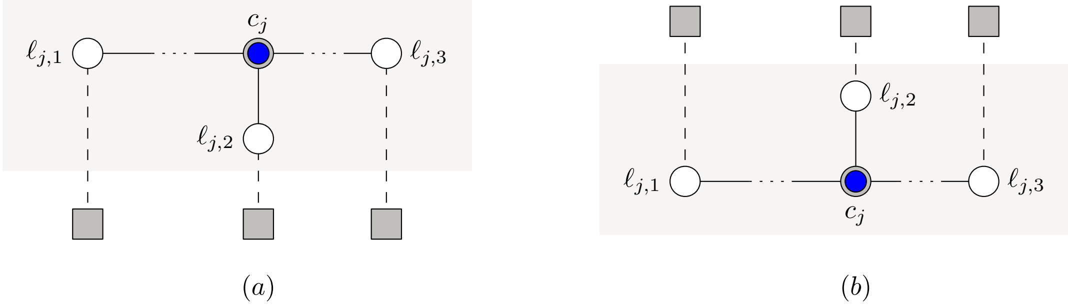


Figure 5: The possible initial configurations of the clause gadget for a generic clause C_j. (a) Depicts the gadget when C_j is positive, while (b) is a vertically-mirrored version used when C_j is negative. The square vertices belong to the appropriate branch of the variable gadgets corresponding to the incident literal vertex (whose connecting edges are drawn using dashed lines).

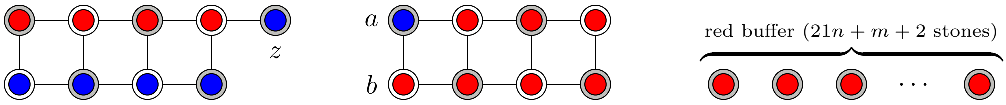


Figure 6: The punisher gadget in its initial configuration with its three parts, from left to right: the *impasse part*, the *flip-flop part*, and the *Red buffer*.

words, each Blue stone is either in the formula area or part of the punisher gadget.

The initial configuration of the punisher gadget is shown in Figure 6 and it consists of three parts: (i) an *impasse part* consisting of a grid graph with two rows and four columns in which the vertices in the top (resp. bottom) row contain Red (resp. Blue) stones, plus a vertex z with a Blue stone dangling from a Red corner; (ii) a *flip-flop part* consisting of a grid graph with two rows and four columns, where one corner contains a Blue stone and all other vertices contain Red stones; (iii) a *Red buffer* consisting of $B = 21n + m + 2$ isolated vertices containing a Red stone.

Suppose that the Blue stone on vertex a is removed, that all stones of the impasse part are still in their original positions, and that at least $B - 1$ of the buffer positions still contain a Red stone. Then, Red can force a win. Indeed, Red can alternate sliding the stone initially on position a along the edge (a, b). Each such slide closes a Red mill and allows Red to first remove the Blue stone placed on z, and then remove all Blue stones placed in the formula area.[8] This continues until the only remaining Blue stones are those in the vertices of the impasse gadget other than z. At this point, Blue has no more legal moves and Red wins. Observe that neither of the Red stones in the mills involving a and b closed by Red, nor those in the impasse part of the gadget can be removed by Blue. Indeed, they always belong to at least one Red mill; hence, they can only be deleted after all the Red stones in the buffer vertices have already been deleted. This requires at least $B - 1$ additional Blue turns, yet Red needs to delete at most $B - 1 = 21n + m + 1$ Blue stones (the stone on z, at most 21 stones per variable gadget, and at most one stone per clause gadget); hence, Red wins after playing at most $B - 1$ turns, i.e., within Blue's next $(B - 1)$-th turn.

Observe further that no Blue stone can fly, since the game ends in a configuration in which Blue still has 4 stones in the impasse part of the gadget.

Threat gadgets. Finally, we discuss threat gadgets. Figure 7 (a) and (b) show a threat gadget in its *active* and *primed* state, respectively. The initial configuration contains $T = 6n + m$ threat gadgets, namely $T - 1$ copies of a threat gadget in the active state, plus a single threat gadget in the primed state.

Notice how Red can bring one active gadget into the primed state by sliding a stone. We say that a threat gadget is *defused* if at least one of its Red stones has been removed. The idea of the gadget is as follows: if at the beginning of one of the first T Blue turns there exists a primed threat gadget,

[8]Although vertex z is not necessary to show the NP-hardness of GMG as defined, it will be useful in the sequel to enforce additional connectivity constraints.

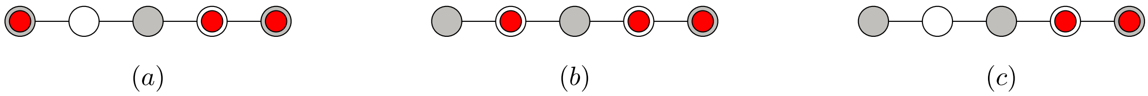


Figure 7: (a) A threat gadget in its active state. (b) A threat gadget in its primed state. (c) A threat gadget in one of its possible defused states.

then Blue must defuse it by the end of their turn. This can only be achieved by closing a Blue mill and deleting a Red stone from the gadget. If the upcoming Red turn is one of the first $T-1$ turns, then Red can prime another threat gadget which needs to be defused, and so on.

As soon as the Blue player fails to defuse a primed threat gadget, Red is able to act on the threat by sliding a Red stone in the gadget to close a Red mill. This allows Red to delete the Blue stone in position a of the punisher gadget and to eventually win the game. The goal of Blue is to defuse all threat gadgets, i.e., to close a mill in each of their first T moves.

3.3. Completing the reduction

We can now show that the GMG instance built as described above allows Blue (resp. Red) to force a win if and only if φ admits (resp. does not admit) a satisfying assignment. We also provide an upper bound on the number of moves required for Blue (resp. Red) to win, as it will be useful in the sequel.

Lemma 1. *If φ is satisfiable, then Blue has a strategy that wins the game in at most $72n + 7m + 18$ moves. Otherwise, Red has a strategy that wins the game in at most $27n + 2m$ moves.*

Proof. Suppose that φ is satisfiable, let $\pi : \{x_1, \dots, x_n\} \to \{\bot, \top\}$ be a satisfying assignment, and recall that $T = 6n + m$. Blue uses four moves for each variable x_i to activate the branch of x_i's gadget that corresponds to the truth value $\pi(x_i)$, and to perform the three subsequent forced moves in the gadget (see Figure 4 for the true branch; the false branch is symmetric). Then, for each positive (resp. negative) clause C_j, Blue picks some literal x_i of C_j with $\pi(x_i) = \top$ (resp. some literal $\overline{x}_i$ of C_j with $\pi(x_i) = \bot$) and slides the stone on c_j to a vertex in $\{\ell_{j,1}, \ell_{j,2}, \ell_{j,3}\}$ that is adjacent to one of $x_{i,1}$ or $x_{i,2}$ (resp. $\overline{x}_{i,1}$ or $\overline{x}_{i,2}$). Each of the above moves closes a Blue mill, and Blue removes Red stones by following this strategy: if there is a primed threat gadget, Blue defuses it by deleting any of its stones; otherwise, Blue chooses an arbitrary active threat gadget and defuses it.

Then, after the T-th Blue turn, all threat gadgets have been defused and Red cannot close any mill. Blue repeatedly breaks and closes a mill (e.g., by choosing a vertex $\ell_{j,h}$ that contains a Blue stone, and sliding such a stone back and forth along the edge $(c_j, \ell_{j,h})$), and deletes all Red stones except those in the impasse area of the punisher gadget. As soon as the last such Red stone is removed, Red has no remaining legal moves and Blue wins.[9] Observe that Red has $3T + B + 7$ stones in the initial configuration, excluding those in the impasse part of the punisher gadget. Since T of these stones are removed in the first T turns of Blue, and each additional Red stone is removed after at most two Blue turns from the previous removal, Blue wins in at most $T + 2(2T + B + 7) = 72n + 7m + 18$ moves.

Assume now that φ is not satisfiable. As discussed in the corresponding sections, if a mill needs to be closed at every move, then each of the n variable gadgets allows for at most 4 moves, and each clause gadget allows for at most a single move. Moreover, in order to close a mill by moving a stone in the gadget of a generic positive (resp. negative) clause C_j, the stone on c_j needs to be slid to some literal vertex $\ell_{j,h}$ adjacent to either $x_{i,1}$ or $x_{i,2}$, where x_i (resp. $\overline{x}_i$) is a literal of C_j and the true branch (resp. false branch) of x_i's gadget has been activated. Then, if Blue were able to close a mill in all of their first T moves, this would imply that the truth assignment induced by the activated variable branches satisfies φ, a contradiction. Hence, Blue can close at most $T-1$ mills in their first T moves. A winning strategy for Red is the following: if at the beginning of Red's turn there is a primed threat gadget, Red closes a mill in such a gadget, removes the Blue stone on vertex a of the punisher gadget, and

[9]This is not necessarily the most efficient winning strategy for Blue, yet it is enough to establish that Blue can force a win in the required number of moves.

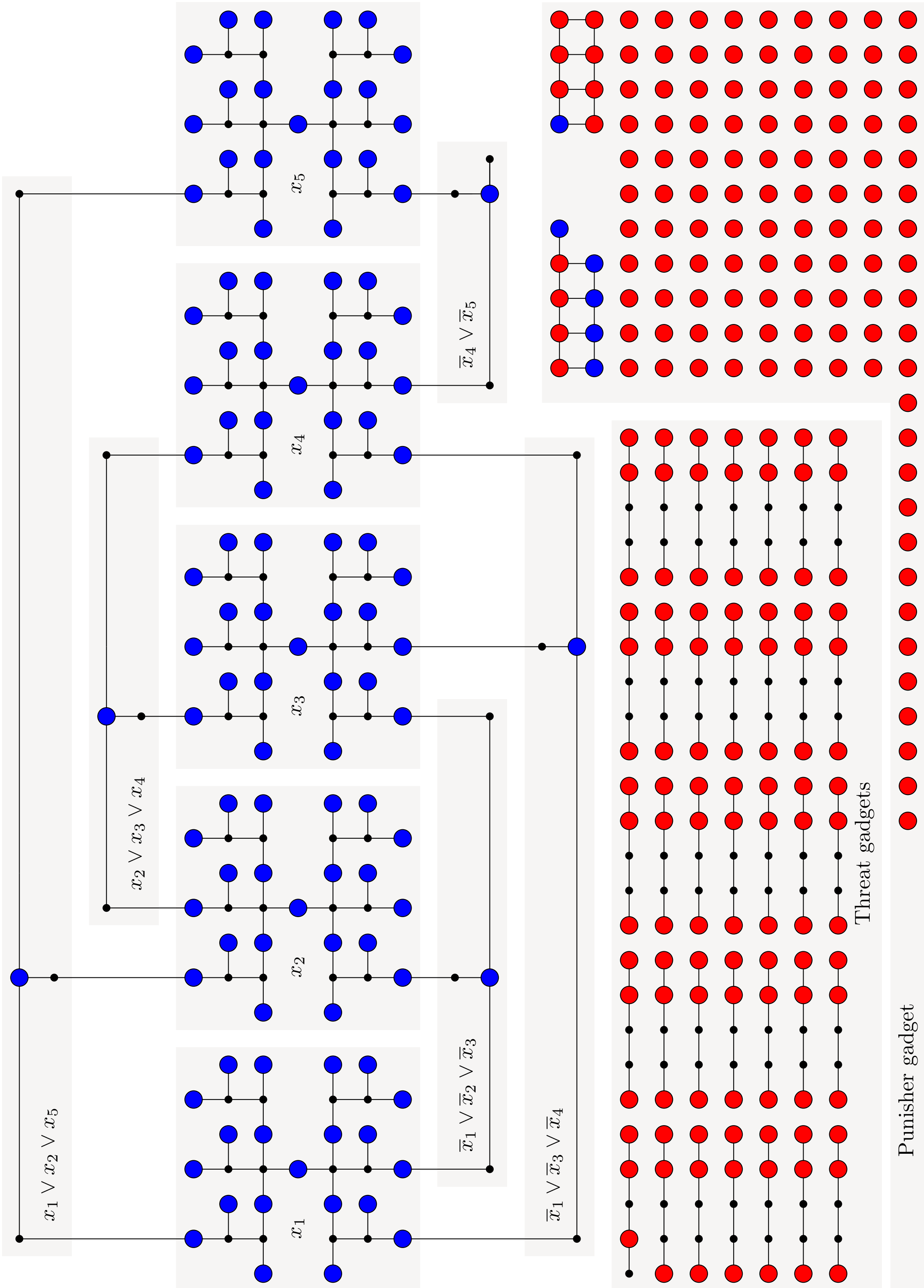


Figure 8: The instance of GMG obtained by applying our reduction to the PMR3-SAT instance of Figure 2. An interactive version is available at https://isnphard.com/g/generalized-morris-games/.

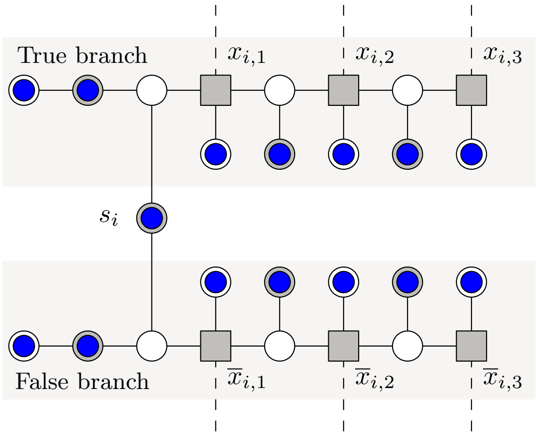


Figure 9: An alternative variable gadget to the one in Figure 3 which allows for non-collinear mills.

proceeds to win by playing the flip-flop part of the punisher gadget as discussed in the relevant section. Otherwise, Red chooses an arbitrary active threat gadget and brings it into the primed state by sliding one Red stone. Since Red has T threat gadgets, yet Blue can only defuse at most $T-1$ of them, one of the first T Red turns will necessarily start with a threat gadget in the primed state. Observe that after the T-th Red turn, the Blue stone on vertex a has been deleted, and there are at most $21n+m$ remaining Blue stones not in the impasse area, i.e., those in the formula area. Then Red wins within their first $T+21n+m=27n+2m$ moves. □

An example of the GMG instance obtained by applying our reduction to the PMR3-Sat instance of Figure 2 is shown in Figure 8 and is playable at https://isnphard.com/g/generalized-morris-games/.

3.4. Variants and additional constraints

The graphs (and their embeddings) corresponding to the classical Five/Six, Seven, and Nine Men's Morris boards exhibit additional topological and geometrical properties besides planarity. Namely, the graph is connected, bipartite, it has minimum degree 2 (and maximum degree 4), the embedding is planar and rectilinear,[10] and each vertex has at least two incident edges with orthogonal orientations.

In the rest of this section we argue that our NP-hardness result holds even if we impose all of the above restrictions on the GMG instance. Moreover, one could also wonder how the complexity of the game changes if mills are not required to be collinear, i.e., a mill consists of any path of three vertices all occupied by stones of the same color. A small modification of our reduction shows that this variant is also NP-hard and co-NP-hard, even if all the above constraints still need to hold. Observe that this variant is a special case of the generalization in which the geometric aspect of the game is removed altogether, i.e., the game is played on an arbitrary graph (without an associated embedding).

Allowing non-collinear mills. The reduction can be easily adapted to handle the case in which mills are not required to be collinear. Indeed, it suffices to replace the variable gadgets of Figure 3 with those shown in Figure 9.

Observe that no tweaks for T or B are necessary, as the replacement variable gadget allows Blue to close 6 mills, as in the original variable gadget. Moreover, it uses fewer Blue stones (15 compared to 21); hence, the buffer part of the threat gadgets still contains enough Red stones for the reduction to work.

Bipartiteness and rectilinearity. We observe that the graph resulting from our reduction is already bipartite and that it admits a rectilinear embedding. A possible bipartition is shown by the shading

[10] A planar rectilinear embedding is a planar straight-line embedding in which each edge is drawn as an axis-aligned (i.e., either a horizontal or vertical) segment.

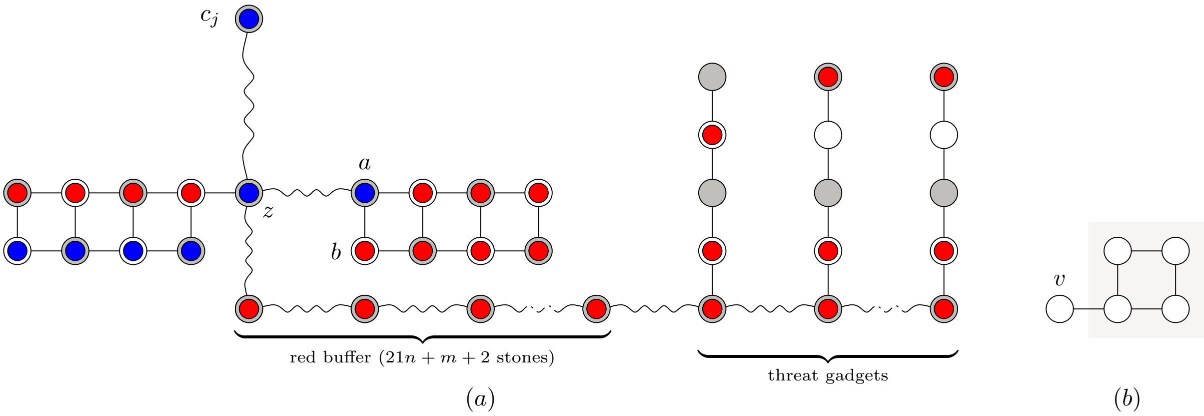


Figure 10: (a) A schematic representation of how the various parts of the punisher gadget and all the threat gadgets can be connected together and to the formula area while preserving rectilinearity. Vertex c_j belongs to the gadget of a negative clause C_j with the smallest y-coordinate in the rectilinear embedding of PMR3-SAT. (b) A gadget that can be used to locally increase the degree of a vertex v (once a suitable rotation is applied).

(either white or gray) of the vertices of the gadgets in Figures 3, 5, 6, 7, and 9. Observe that all vertices $\ell_{j,1}, \ell_{j,2}, \ell_{j,3}$ in the clause gadgets are white, while the vertices $\overline{x}_{i,1}, \overline{x}_{i,2}, x_{i,1}, x_{i,2}$ in the variable gadgets are gray; hence, the edges connecting those gadgets do not violate the bipartiteness constraint. Moreover, the drawing of the gadgets is already rectilinear.

Connectivity. To ensure that the graph resulting from our reduction is connected, we can modify our graph as follows. We first link all variable gadgets together by augmenting the graph with a collection of $n-1$ horizontal paths of length $L = 72n + 8m + 20$, where the i-th such path has s_i and s_{i+1} as its endvertices (see Figure 3 and Figure 9) and $L-1$ additional empty internal vertices. This ensures that the subgraph in the formula area is connected.

To ensure that the subgraph in the threat area is also connected, that the formula and the threat areas are connected to each other, and that the embedding remains rectilinear, we let C_j be any (negative) clause that minimizes the y-coordinate of its segment in the rectilinear embedding of φ, and we connect together all parts of the punisher gadget, all threat gadgets, and vertex c_j of C_j's gadget as shown in Figure 10 (a), where each wavy line denotes a path of length L (i.e., with $L-1$ additional empty internal vertices).[11]

Since the number of internal vertices of each path is odd, and all endvertices of the paths are on the same side of the bipartition described above, the resulting graph is also bipartite. One can verify that the arguments in Lemma 1 continue to hold once one observes that our choice of L ensures that no stone starting from an endvertex of a path can reach any neighbor of the other endpoint within $72n + 7m + 18$ moves, nor can two stones starting from opposite endvertices of a path and moving within the path ever become adjacent.

Degree and local orthogonality constraints. Even after enforcing connectivity, our reduction has some vertices of degree 1, while in the classical board all vertices have a degree of at least 2. We can always increase the degree of a vertex v to 2 by appending one or two (suitably rotated) copies of the gadget shown in Figure 10 (b).

Assume now that all vertices have degree at least two. One can further observe that, in the classical board, all vertices have two orthogonal incident edges. Since the only vertices that do not satisfy this property must have a degree of exactly two, we can augment these vertices by appending a further (suitably rotated) copy of the gadget in Figure 10 (b) to them (notice that all vertices of the gadget in Figure 10 (b) already satisfy this orthogonality property).

[11]We can assume w.l.o.g. that there exists at least one negative clause in φ, since otherwise φ is trivially satisfiable.

Reversing the roles of the players. Lemma 1 and the above discussion show that, in all the described variants (and combinations thereof), deciding whether the first player to move from an arbitrary configuration can force a win is NP-hard, while deciding whether the second player to move can force a win is co-NP-hard.

We can also reverse the roles of the players by considering the above reduction where the first player to move is Red, and the T threat gadgets are initially all in the active state (i.e., we replace the single threat gadget in the primed state with one in the active state). Our proof strategy also works in this case: if the input formula φ is satisfiable, then Blue can defuse all T threat gadgets and proceed to win the game; otherwise, Red wins by ensuring that in each of the first T Blue turns there is some primed threat gadget, at least one of which cannot be defused. This shows that deciding whether the first (resp. second) player to move can force a win is also NP-hard (resp. co-NP-hard).

The following theorem summarizes the above discussion.

Theorem 1. *Given a configuration of GMG, deciding whether the next player to move (resp. the second player to move) has a winning strategy is* NP*-hard and* co-NP*-hard, even if the graph is connected, bipartite, admits a planar rectilinear embedding with vertex degrees between* 2 *and* 4*, and all vertices with degree* 2 *have two incident orthogonal edges. The above result also holds when mills are not required to be collinear.*

4. Open problems

A consequence of our result is that GMG is unlikely to be in NP. Moreover, it is easy to see that the number of possible configurations is upper bounded by $3^n \cdot (b+1)(r+1)$, where n is the number of vertices of the graph, and b (resp. r) is the number of stones of Red (resp. Blue) that are on some vertex or have not yet been placed; hence, a standard retrograde analysis technique shows that GMG is in EXPTIME. It would therefore be interesting to narrow this complexity gap, e.g., by showing that GMG is PSPACE-complete or EXPTIME-complete.

Declaration on Generative AI

The authors have not employed any Generative AI tools for the writing of this manuscript.

References

[1] R. C. Bell, Board and table games from many civilizations, volume 1, Courier Corporation, 1979.

[2] F. Berger, From circle and square to the image of the world: a possible interpretation for some petroglyphs of merels boards, Rock Art Research 21 (2004) 11–26.

[3] D. M. Low, C. E. Batey, R. Gourlay, A viking burial at balnakeil, sutherland, Baldwin (ed.) (2000) 24–34.

[4] Highland Historic Environment Record Website, MHG11310 - Viking Burial - Balnakeil Bay, 2026. URL: https://her.highland.gov.uk/Monument/MHG11310.

[5] K. Jaworski, Rock engravings depicting nine men's morris boards from krzyżowa mountain in the strzegomskie hills, Acta Archaeologica Carpathica 60 (2025) 193–218. URL: https://www.ceeol.com/search/article-detail?id=1394877.

[6] H. J. R. Murray, A History of Board-Games Other Than Chess, Oxford University Press, Oxford, 1952.

[7] R. K. Gibson, Ovid: Ars Amatoria, Book III, volume 40, Cambridge University Press, 2003.

[8] R. Gasser, Solving nine men's morris, in: Games of No Chance, volume 29 of *MSRI Publications*, 1996, pp. 101–113.

[9] P. Stahlhacke, The game of Lasker Morris, Technical Report, FSU Jena, Tech. Rep., 2003.

[10] G. E. Gévay, G. Danner, Calculating ultrastrong and extended solutions for nine men's morris, morabaraba, and lasker morris, IEEE Transactions on Computational Intelligence and AI in Games 8 (2016) 256–267. doi:10.1109/TCIAIG.2015.2420191.

[11] W. B. Loewer, The effects of rule variations on perfect play databases for nine men's morris, ResearchGate preprint, 2016. URL: https://www.researchgate.net/publication/305413042_The_Effects_of_Rule_Variations_on_Perfect_Play_Databases_for_Nine_Men%27s_Morris. doi:10.13140/RG.2.1.4972.4407.

[12] S.-A. Petcu, S. Holban, Nine men's morris: Evaluation functions, in: Proceedings of the 9th International Conference on Development and Application Systems, Suceava, Romania, 2008, pp. 89–92. URL: https://www.dasconference.ro/papers/2008/B7.pdf.

[13] F. Chesani, A. Galassi, M. Lippi, P. Mello, Can deep networks learn to play by the rules? a case study on nine men's morris, IEEE Transactions on Games 10 (2018) 344–353. doi:10.1109/TG.2018.2804039.

[14] J. Abukhait, A. Aljaafreh, N. Al-Oudat, A multi-agent design of a computer player for nine men's morris board game using deep reinforcement learning, in: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), 2019, pp. 489–493. doi:10.1109/SNAMS.2019.8931879.

[15] D. Vrajitoru, A pattern-based bayesian network for the game of nine men's morris, in: J. P. Zagal, E. MacCallum-Stewart, J. Togelius (Eds.), Proceedings of the 10th International Conference on the Foundations of Digital Games, FDG 2015, Pacific Grove, CA, USA, June 22-25, 2015, Society for the Advancement of the Science of Digital Games, 2015. URL: http://www.fdg2015.org/papers/fdg2015_paper_09.pdf.

[16] I. Stanojkovska, Playing nine men's morris with monte carlo tree search and a neural network, 2025. URL: https://repozitorij.uni-lj.si/IzpisGradiva.php?id=167753.

[17] S. Andaloro, Monte carlo tree search applied to nine men's morris, 2016. URL: https://hdl.handle.net/10589/126365.

[18] A. Farouk, Artificial intelligence algorithms: Implementing a nine men's morris with ai, 2019. URL: http://hdl.handle.net/2437/267215.

[19] J. A. Storer, On the complexity of chess, J. Comput. Syst. Sci. 27 (1983) 77–100. URL: https://doi.org/10.1016/0022-0000(83)90030-2. doi:10.1016/0022-0000(83)90030-2.

[20] A. S. Fraenkel, D. Lichtenstein, Computing a perfect strategy for n$\times$ n chess requires time exponential in n, in: International colloquium on automata, languages, and programming, Springer, 1981, pp. 278–293.

[21] D. Lichtenstein, M. Sipser, GO is polynomial-space hard, J. ACM 27 (1980) 393–401. URL: https://doi.org/10.1145/322186.322201. doi:10.1145/322186.322201.

[22] J. M. Robson, The complexity of go, Proc. IFIP (International Federation of Information Processing) (1983).

[23] A. S. Fraenkel, M. R. Garey, D. S. Johnson, T. Schaefer, Y. Yesha, The complexity of checkers on an n × n board, in: 19th Annual Symposium on Foundations of Computer Science (sfcs 1978), 1978, pp. 55–64. doi:10.1109/SFCS.1978.36.

[24] S. Iwata, T. Kasai, The othello game on an n*n board is pspace-complete, Theor. Comput. Sci. 123 (1994) 329–340. URL: https://doi.org/10.1016/0304-3975(94)90131-7. doi:10.1016/0304-3975(94)90131-7.

[25] E. Bonnet, F. Jamain, A. Saffidine, Havannah and twixt are pspace-complete, in: H. J. van den Herik, H. Iida, A. Plaat (Eds.), Computers and Games - 8th International Conference, CG 2013, Yokohama, Japan, August 13-15, 2013, Revised Selected Papers, volume 8427 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 175–186. URL: https://doi.org/10.1007/978-3-319-09165-5_15. doi:10.1007/978-3-319-09165-5_15.

[26] D. Bilò, L. Gualà, S. Leucci, G. Proietti, M. Rossi, On the pspace-completeness of peg duotaire and other peg-jumping games, in: H. Ito, S. Leonardi, L. Pagli, G. Prencipe (Eds.), 9th International Conference on Fun with Algorithms, FUN 2018, La Maddalena, Italy, June 13-15, 2018, volume 100 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 8:1–8:15. URL: https:

//doi.org/10.4230/LIPIcs.FUN.2018.8. doi:10.4230/LIPICS.FUN.2018.8.
[27] E. D. Demaine, R. A. Hearn, Playing games with algorithms: Algorithmic combinatorial game theory, in: M. H. Albert, R. J. Nowakowski (Eds.), Games of No Chance 3, volume 56 of *Mathematical Sciences Research Institute Publications*, Cambridge University Press, 2009, pp. 3–56. doi:10.1017/CBO9780511807251.002.
[28] G. Bagan, Q. Deschamps, F. Galliot, M. Mikalacki, N. Oijid, Token positional games, in: J. Iacono (Ed.), 13th International Conference on Fun with Algorithms, FUN 2026, Porquerolles, France, May 18-22, 2026, volume 366 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026, pp. 5:1–5:22. URL: https://doi.org/10.4230/LIPIcs.FUN.2026.5. doi:10.4230/LIPICS.FUN.2026.5.
[29] A. Fraenkel, Combinatorial games: selected bibliography with a succinct gourmet introduction, Games of no Chance 3 (2009) 491–575.
[30] M. de Berg, A. Khosravi, Optimal binary space partitions for segments in the plane, International Journal of Computational Geometry & Applications 22 (2012) 187–206. doi:10.1142/S0218195912500045.