Theoretical Computer Science 411 (2010) 3971–3978

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

# Phutball is PSPACE-hard

Dariusz Dereniowski [*]

*Department of Algorithms and System Modeling, Gdańsk University of Technology, Poland*

ARTICLE INFO

*Article history:*
Received 7 September 2007
Received in revised form 17 August 2010
Accepted 23 August 2010
Communicated by A. Fraenkel

*Keywords:*
Computational complexity
Games
Phutball
Pspace hardness

ABSTRACT

We consider the $n \times n$ game of Phutball. It is shown that, given an arbitrary position of stones on the board, it is a PSPACE-hard problem to determine whether the specified player can win the game, regardless of the opponent's choices made during the game.

## 1. Introduction

There is a deep mathematical theory developed for analyzing combinatorial games [1,8]. The researchers work on the algorithmic techniques which are useful for finding good game strategies for many board games, including Phutball [4,5]. Paper [6] introduces the notion of generalized threads and this technique is used to solve some Go instances and the author suggests that this approach could be effective for other board games, like Phutball. T. Cazenave used an approach called Gradual Abstract Proof Search to show that 11 × 11 Phutball is a win for the first player [7]. The game is *loopy*, i.e. it is possible to obtain a configuration of stones which already appeared in one of the previous turns — some combinatorial aspects of loopy games were considered in [25]. In this paper we are interested in the complexity of the game rather than in manipulating and analyzing the rooted tree describing the game. Several generalizations of one-player games turn out to be NP-complete: Peg Solitaire [27], Minesweeper (the problem of testing consistency) [20], Same Game [3]. However, most of the board games (especially two-player games) appear to be harder: Checkers [23], Hex (a generalization to graphs) [13], Othello (Reversi) [19], Sokoban [9], Go [21,22,28], Dyson Telescopes [12], Rush Hour [14] or Amazons [16].

The Phutball [2] game is usually played on a 19 × 19 Go board. Initially a black stone is placed in the middle of the board. The players make their moves alternately. A player makes his move by either placing a white stone in an unoccupied position, or makes a sequence of *jumps* over horizontal, vertical or diagonal sequences of white stones. Each jump is performed by moving the black stone, called the *ball*, over a line of white stones (no empty space between the ball and the line is allowed if we want to make a jump) and placing the ball on the board on the first unoccupied position after the last white stone in the line. The white stones are removed from the board immediately after the jump. Each player tries to move the ball on or over the opponent's *goal line*. The goal lines are two opposite edges of the board. We consider a natural generalization with an arbitrary size of the board and initially a black stone placed in the middle of the board.

As indicated in [2], Phutball is not the kind of game where you can expect a complete analysis. The authors considered in [17] a simplified version of the game – the case where there is only one dimension – and it turns out that according to the presented examples, the one-dimensional version still seems to be hard to analyze. Moreover, given an arbitrary position

[*] Tel.: +48 583471956; fax: +48 583471766.
*E-mail address:* deren@eti.pg.gda.pl.

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.08.019

in the 2-dimensional Phutball game, it is an NP-complete problem to determine whether the current player can win the game in his next move [11]. However, as indicated in several papers [7,10,11,18], the complexity of the Phutball game is still open. In this paper we place the problem of determining whether the current player has a winning strategy in the class of PSPACE-hard problems.

The paper is organized as follows. In Section 2 we define, for the purposes of our main result, a graph game and we prove that this game is PSPACE-hard. Then, in Section 3 we argue that we can set up a configuration of stones in the Phutball game, so that the latter simulates the special instances of the graph game used in the reduction in Section 2, which proves that Phutball is PSPACE-hard.

## 2. A graph game

We start this section by describing the rules of a game played on a graph. Then we prove that this game is PSPACE-hard. The graph constructed on the basis of a problem known to be PSPACE-complete is defined in such a way that its topology allows us to code it as a configuration of stones in the Phutball game.

The game described in the following is played on a directed graph. For completeness we list here some basic definitions. A directed graph $G$ is a pair $G = (V(G), E(G))$ with a vertex set $V(G)$ and a set of directed edges $E(G)$ (each $e \in E(G)$ is an ordered pair of two vertices). We say that $H$ is a *subgraph* of $G$, $H \subseteq G$, if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A *directed path* $P = (\{v_1, \dots, v_n\}, E(P))$ from $v_1$ to $v_n$ is a graph with edge set $E(P) = \{(v_i, v_{i+1}) : i = 1, \dots, n-1\}$. The vertices $V(P) \setminus \{v_1, v_n\}$ are the *internal vertices* of $P$.

The input of the game is a directed graph $G = (V(G), E(G))$, a set $C \subseteq V(G)$, a vertex $s \in C$, and a relation $R \subseteq V(G) \times E(G)$ between the vertices and the edges of $G$. If $(v, e) \in R$ then we say that a vertex $v$ is *pointing* at edge $e$. Denote by $R^{-1}(E(G))$ the set of vertices $v$ for which there exists $e \in E(G)$, such that $(v, e) \in R$. The players of the game will be called $\exists$-*player* and $\forall$-*player*. If $X$ refers to one of the players then $\overline{X}$ is the other player.

At each point of the game there is a unique *active vertex*. The players must follow the rules:

Rule 1 (*initialization*). The $\exists$-player starts the game. Initially $s$ is the active vertex.

Rule 2 (*a move*). Let $u \in C$ be the active vertex. The current player $X$ selects a vertex $v \in C \cup R^{-1}(E(G))$ and a directed path $P \subseteq G$ from $u$ to $v$ such that all internal vertices of $P$ are in $V(G) \setminus (C \cup R^{-1}(E(G)))$. The edges of $P$ are removed from $G$, $v$ becomes the active vertex, and $\overline{X}$ becomes the current player. We say that $X$ *moves from* $u$ *to* $v$.

Rule 3 (*game end conditions*). If the current player cannot make a move, i.e. there is no directed path $P$ from the active vertex to a vertex $v \in C \cup R^{-1}(E(G))$, then the current player loses the game. If the current player moves from $u$ to $v \in R^{-1}(E(G))$ then he wins the game.

Let us recall the PSPACE-complete Quantified Boolean Formula (*QBF*) problem [26]. Given a formula $Q$ in the form

$$Q_1 x_1 \cdots Q_n x_n F(x_1, \dots, x_n),$$

decide whether the formula is true, where $Q_i \in \{\exists, \forall\}$ for $i = 1, \dots, n$. In our case we use a restricted case of this problem where $Q_1 = \exists$, $Q_{i+1} \neq Q_i$ for $i = 1, \dots, n-1$, the integer $n$ is even, and $F$ is a 3CNF formula, i.e. $F = F_1 \wedge F_2 \wedge \dots \wedge F_m$, where $F_i = (l_{i,1} \vee l_{i,2} \vee l_{i,3})$ and each literal $l_{i,j}$ is a variable or the negation of a variable, $i = 1, \dots, m$, $j = 1, 2, 3$. (A negation of a variable $x_i$ is denoted by $\overline{x_i}$.) The restricted problem is still PSPACE-complete [24].

Given $Q$, we create a directed graph $G$. For each variable $x_i$ define the corresponding *variable component* $G(x_i)$:

$$V(G(x_i)) = \{a_i, b_i, c_i, d_i, e_i\},$$
$$E(G(x_i)) = \{(a_i, b_i), (a_i, c_i), (b_i, d_i), (c_i, d_i), (d_i, e_i)\},$$

for $i = 1, \dots, n$ (see Fig. 1(a)). We connect the variable components in such a way that $(e_i, a_{i+1}) \in E(G)$ for each $i = 1, \dots, n-1$.

Define the *formula component* $G(F)$ as follows

$$V(G(F)) = \{x_i, y_i, z_i : i = 1, \dots, m\} \cup \{w_{i,j} : i = 1, \dots, m, j = 1, 2, 3\},$$
$$E(G(F)) = \{(x_i, y_i), (y_i, z_i), (z_i, w_{i,1}), (w_{i,1}, w_{i,2}), (w_{i,2}, w_{i,3}) : i = 1, \dots, m\} \cup \{(x_i, x_{i+1}) : i = 1, \dots, m-1\}.$$

Fig. 1(b) shows the formula component. To finish the construction of $G$ let $(e_n, x_1) \in E(G)$ and introduce a vertex $e_0$ connected to the graph in such a way that $(e_0, a_1) \in E(G)$.

The input to our graph game is the directed graph $G$ defined above, the set $C = \{e_0, \dots, e_{n-1}\} \cup \{z_1, \dots, z_m\}$, $s = e_0$ and $R$ containing a pair $(w_{i,j}, (b_l, d_l))$ (respectively $(w_{i,j}, (a_l, c_l))$) for $i \in \{1, \dots, m\}$, $j \in \{1, 2, 3\}$, $l \in \{1, \dots, n\}$, iff $l_{i,j} = x_l$ ($l_{i,j} = \overline{x_l}$, resp.). Observe that initially $R^{-1}(E(G))$ contains all the vertices $w_{i,j}$, $i = 1, \dots, m$, $j = 1, 2, 3$, because a vertex $w_{i,j}$ corresponds to the literal $l_{i,j}$, which equals $x_l$ or $\overline{x_l}$ for some $l \in \{1, \dots, n\}$. However, during the game the set $R^{-1}(E(G))$ gets smaller due to the fact that some of the edges of $G$ are removed from $G$.

Let us consider the following complete example of our reduction. Given a formula $Q$

$$\exists x_1 \forall x_2 \exists x_3 \forall x_4 (x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4), \tag{1}$$

Fig. 2 depicts the corresponding graph $G$. The dashed arrows represent the elements of the relation $R$, the vertices in $C$ are denoted as white nodes, while the vertices in $V(G) \setminus C$ are the black nodes. Recall that each move 'starts' at a vertex in $C$

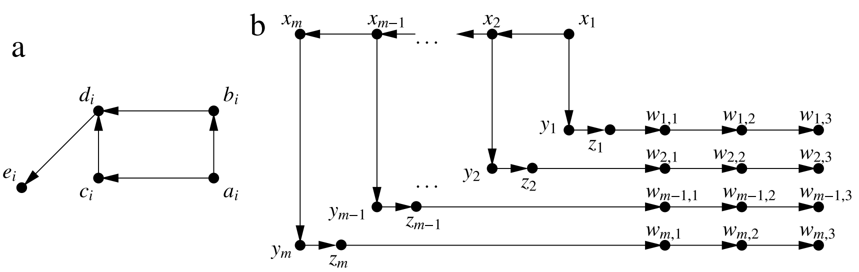

**Fig. 1.** The graphs (a) $G(x_i)$ and (b) $G(F)$.

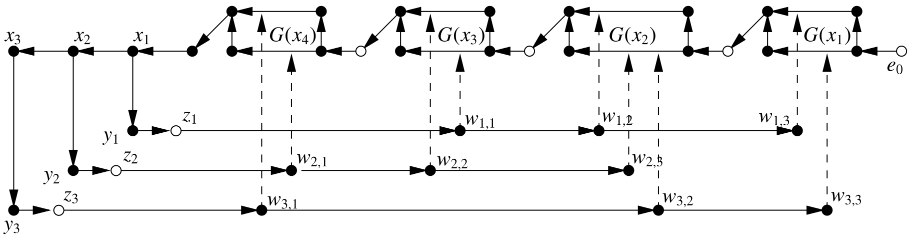

**Fig. 2.** A complete instance of the graph $G$ corresponding to the formula in (1).

and 'ends' at a vertex in $C$ or in $R^{-1}(E(G))$. In the former case the opponent makes a move, while in the latter case the game ends. Since the $\exists$-player starts the game and the active vertex is $e_0$, two moves are possible, i.e. the selected path contains the vertices $e_0, a_1, b_1, d_1, e_1$ or $e_0, a_1, c_1, d_1, e_1$. After the move $w_{1,3}$ or $w_{3,3}$ does not belong to the set $R^{-1}(E(G))$, respectively, because either $(b_1, d_1)$ or $(a_1, c_1)$ is removed from $E(G)$ as a result of the move.

The game obtained in the reduction has a special structure, which makes it quite easy to analyze. Here we list three straightforward facts describing the structure of the game.

**Fact 1.** If the active vertex is $e_{i-1}, i \in \{1, \ldots, n-2\}$ then the current player makes a move from $e_{i-1}$ to $e_i$ and the directed path $P$ removed from $G$ contains one of the following sequences of edges:

$$(e_{i-1}, a_i), (a_i, b_i), (b_i, d_i), (d_i, e_i), \tag{2}$$

$$(e_{i-1}, a_i), (a_i, c_i), (c_i, d_i), (d_i, e_i). \tag{3}$$

Furthermore, the $\exists$-player makes such a move for $i = 1, 3, 5 \ldots, n-1$ while the $\forall$-player makes this move for $i = 2, 4, 6, \ldots, n-2$ (recall that $n$ is even). □

**Fact 2.** Let $e_{n-1}$ be the active vertex. The $\forall$-player is the current player and he makes a move from $e_{n-1}$ to a vertex $z_j$ and the path $P$ contains the edges in (2) or in (3) with $i = n$, followed by the edges

$$(e_n, x_1), (x_1, x_2), \ldots, (x_{j-1}, x_j), (x_j, y_j), (y_j, z_j), \tag{4}$$

where $j \in \{1, \ldots, m\}$. □

**Fact 3.** Assume that $z_j, j \in \{1, \ldots, m\}$, is the active vertex. The $\exists$-player is the current player. If there exist a vertex $w_{j,k}$ and an edge $e \in E(G)$ such that $(w_{j,k}, e) \in R$, then $\exists$-player wins the game. Otherwise he cannot make a move and he loses the game. □

**Theorem 1.** *The above graph game is* PSPACE-*hard.*

**Proof.** The QBF problem can be described as a game between two players. The players alternate their moves choosing the variable assignments. Player 1 starts the game. The appropriate player selects in the $i$th turn of the game the value of the variable $x_i$, $i = 1, \ldots, n$. Player 1 wins if and only if $F$ is satisfied after the $n$th turn under the selected variable assignment.

To prove that the graph game is PSPACE-hard it is enough to show that the $\exists$-player has a winning strategy in the graph game if and only if Player 1 has a winning strategy in the QBF game (equivalently, $Q$ is true).

First, assume that Player 1 has a winning strategy. The $\exists$-player mirrors the moves of Player 1 in such a way that when Player 1 sets the value of $x_i$, where $i$ is odd, then the $\exists$-player makes a move from $e_{i-1}$ to $e_i$. Moreover, if $x_i$ is true, then the $\exists$-player traverses the sequence of edges as stated in (3). On the other hand, if Player 1 decides $x_i$ to be false, then the $\exists$-player traverses the edges listed in (2). If $i$ is even, then the $\forall$-player selects his path in (2) or (3) arbitrarily, just as Player 2 selects the value of the corresponding variable $x_i$. By Fact 2, the $\forall$-player chooses in his last move an index $j \in \{1, \ldots, m\}$ and makes a move from $e_{n-1}$ to a vertex $z_j$. Since $F$ is true, $F_j$ is true, and consequently, there exists a true literal $l_{j,k}, k \in \{1, 2, 3\}$. If $l_{j,k} = x_l$ for some $l \in \{1, \ldots, n\}$, then $x_l$ is true which implies that $(b_l, d_l) \in E(G)$ when the active vertex is $z_j$. Moreover,

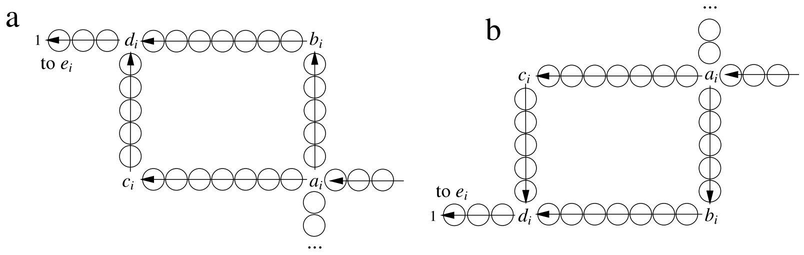

**Fig. 3.** The variable component $G_P(x_i)$ for (a) $i = 1, 3, 5, \ldots, n-1$, and (b) $i = 2, 4, 6, \ldots, n$.

by the definition of $R$, $w_{j,k}$ is pointing at $(b_l, d_l)$. Similarly, if $l_{j,k} = \overline{x_l}$ for $l \in \{1, \ldots, n\}$, then $x_l$ is false, so $(a_l, c_l) \in E(G)$ and $(w_{j,k}, (a_l, c_l)) \in R$. So, for each choice of $j$ by the ∀-player there exists a vertex $w_{j,k}$ pointing at an edge of $G$. So, the ∃-player has a win.

Assume now that ∃-player has a winning strategy. We use it to prove that Player 1 has a winning strategy as well, i.e. $Q$ is true. Similarly to the previous case, a move by the ∃-player of traversing (and removing) the edges in (2) in order to reach $e_i$ is mirrored by Player 1 by setting the value of $x_i$ to be false, otherwise $x_i$ becomes true. Then the ∀-player chooses any of the paths (2) or (3) which corresponds to setting an arbitrary Boolean value to the corresponding variable quantified by ∀. When the Boolean values have been assigned to the variables then, by Fact 2, the ∀-player chooses a vertex $z_j$ and makes it the active vertex. Since the ∃-player has a winning strategy, by Fact 3, at least one of the vertices $w_{j,k}$, $k \in \{1, 2, 3\}$, is pointing at an edge which still belongs to $G$. By Fact 3, the literal $l_{j,k}$ of $F_i$ is true. Since $j$ has been chosen arbitrarily, the formula $F$ is true. Thus, Player 1 has a winning strategy. □

## 3. Transformation of *G* to the Phutball game

In the following we transform the input to the graph game, i.e. a directed graph $G$, a set $C \subseteq V$, a starting vertex $s$ and a relation $R$, defined in Section 2, into a configuration of stones of the Phutball game. Note that we do not give a reduction between the two problems, but we only show how to code the structured instances of the graph game described in the previous section. This, together with Theorem 1, will give a desired reduction from the QBF problem to the Phutball game. The vertices in $C$ will be coded using special gadgets, while the remaining vertices of $G$ correspond to points on the board. We will re-use the vertex labels from the previous section to refer to points on the board (see e.g. Fig. 3(a)). It will be clear from the context whether we refer to a vertex or to a point on the board. We again refer to the players as the ∃-player and the ∀-player.

Let the upper (respectively lower) edge of the board be the ∀-player's (∃-player's, resp.) goal line. We will choose the empty points on the board corresponding to the vertices in $V(G) \setminus C$ in such a way that if there is an edge $(u, v) \in E(G)$ then the points corresponding to $u$ and $v$ will have the same horizontal or vertical coordinates. The edges of the graph correspond to the (horizontal or vertical) sequences of stones. The starting vertex is also coded as an empty spot and it initially contains the ball. The configuration of stones corresponding to the variable component $G(x_i)$ for $i = 1, 3, 5, \ldots, n-1$ is given in Fig. 3(a) while Fig. 3(b) gives the variable component for $i = 2, 4, 6, \ldots, n$. In all the figures of this section, the dots ending a vertical line of stones indicate that the line ends at the appropriate (upper or lower) goal line. We will use two types of configurations corresponding to the vertices $e_i$, $0 < i < n$. Fig. 4(a) (Fig. 4(b)) presents the configuration corresponding to $e_i$ for $i = 1, 3, 5, \ldots, n-1$ ($i = 2, 4, 6, \ldots, n-2$, respectively). In order to make the analysis consistent we will use the label $e_i$, $0 < i < n$, to mark a point on the board as shown in Fig. 4. Roughly speaking, such a gadget forces the following sequence of events: one player moves the ball to $e_i$ (according to an arrow on the right-hand side), then two stones are placed at points 8 and 18 (each by one player) and finally the other player (as indicated by the second arrow) moves the ball to point 19 and then directly to the point $a_{i+1}$. To obtain the configuration of stones corresponding to $G(x_i)$, denoted in the following by $G_P(x_i)$, for $i = 1, 3, 5, \ldots, n-1$ (respectively for $i = 2, 4, 6, \ldots, n-2$) we connect the gadgets in Figs. 3(a) and 4(a) (Figs. 3(b) and 4(b), resp.) in such a way that the points marked by 1 in both pictures refer to the same place on the board.

We will use the following correspondence between the edges of $G$ and the lines of stones in $G_P$: an edge of $G$

$$(x, y) \in \{(e_0, a_1), (e_n, x_1)\} \cup \bigcup_{1 \le i \le n} E(G(x_i)) \setminus \{(e_i, a_{i+1})\}$$

corresponds to a line of stones between the points $x$ and $y$ on the board, while $(e_i, a_{i+1})$, $i = 1, \ldots, n-1$, corresponds to two lines of stones: between $e_i$ and 19 of $G_P(x_i)$ and between 19 of $G_P(x_i)$ and $a_{i+1}$ of $G_P(x_{i+1})$. Note that two points in a line between $e_i$ and 19 are by the definition unoccupied, but the game is set up in such a way that when the ball is about to move from $e_i$ then there is a line of white stones between $e_i$ and 19 of $G_P(x_i)$, which we are going to prove later.

Note that we used Facts 1–3 listed in the previous section to obtain Theorem 1. Those facts gave the necessary and sufficient conditions for the graph game to simulate the QBF game. Now we prove these facts for the corresponding configurations of stones on the board. Then, we may conclude that the game of Phutball simulates the graph game which

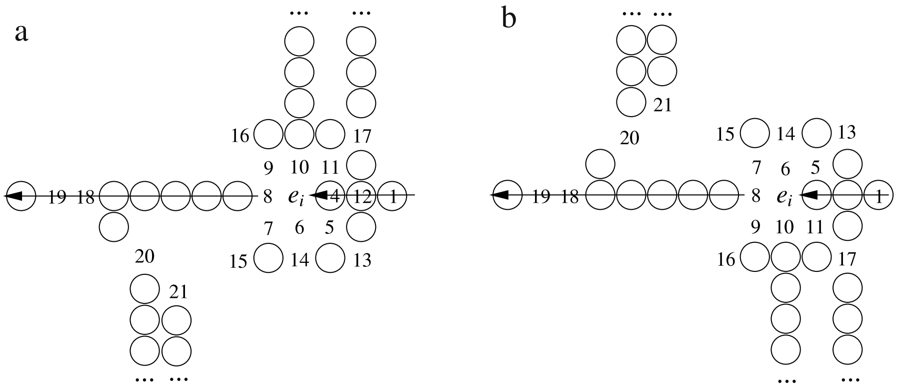

**Fig. 4.** Configuration of white stones corresponding to vertices (a) $e_i$, where $i$ is odd, (b) $e_i$, where $i$ is even, $i < n$.

will give us a desired reduction from the QBF problem to Phutball. If $x$ and $y$ are two points on the board then $x \to y$ denotes a *jump*, that is moving the ball from $x$ to $y$ and removing all the stones between $x$ and $y$ (we will use this symbol in such a way that all the conditions required by the rules of the game for making a jump will be satisfied). $x \to y \to z$ denotes two consecutive jumps $x \to y, y \to z$. Recall that in general one move in the Phutball game may consist of many consecutive jumps.

**Proof of Fact 1.** In the terms of the Phutball game we are going to prove, by an induction on $i$, that if a ball is at $e_{i-1}$, $1 < i < n$, and $X$ is the player making the next move then the following sequence of moves occurs:

(i) $X$ places a white stone at 8 of $G_P(x_{i-1})$,
(ii) $\overline{X}$ places a white stone at 18 of $G_P(x_{i-1})$,
(iii) $X$ makes a sequence of jumps over the lines of stones corresponding to the edges given in (2) or (3).

Moreover, $X$ is the ∃-player (∀-player) for odd (even, respectively) values of $i$. For $i = 1$ only (iii) is done. (For this reason we consider below the case of $i > 1$, because the proof for $i = 1$ is similar.)

So, assume that the ball is at $e_{i-1}$. By the induction hypothesis, the white stones on the right-hand side of $e_{i-1}$ are no longer on the board. Moreover, both for odd and even values of $i$, the vertical lines of stones next to 10 and 17 (20 and 21) of $G_p(x_{i-1})$ lead to the $X$'s ($\overline{X}$'s, respectively) board line. We have that $X$ must place a white stone, because he cannot make a jump. Clearly, he cannot put a stone at 10 of $G_P(x_{i-1})$. If he does not occupy one of the fields 4–11 of $G_P(x_{i-1})$ then in the next turn $\overline{X}$ puts a white stone at the point 10 of $G_P(x_{i-1})$ and it is easy to see that $\overline{X}$ wins the game. Observe, that if $\overline{X}$ is able to move the ball to the point 17 then he is one jump away from his opponent's goal line. By a simple case analysis we obtain that if $X$ places a stone in one of the points 4, 5, 6, 7, 9, 11 of $G_P(x_{i-1})$ then $\overline{X}$ can reach 17 and win the game. Thus, $X$ places a white stone at the field 8 of $G_P(x_{i-1})$, i.e. (i) occurs. Then, $\overline{X}$ can either: (1) make one of the moves $e_{i-1} \to 18$ or $e_{i-1} \to 18 \to 20$, but it is easy to see that in both cases his opponent wins in the next turn, or (2) put a white stone, and if he chooses a field different from 18 then, similarly as in (1), he loses, because either 20 or 21 of $G_P(x_{i-1})$ is unoccupied. So, $X$ can proceed by $e_{i-1} \to 18 \to 20$ or $e_{i-1} \to 18 \to 21$, respectively, and reach his opponent's goal line in the next jump. This proves that (ii) must happen.

The fact that $X$ must reach the point $e_i$ by jumping over the lines of stones corresponding to (2) or (3) follows from the observation that otherwise he loses the game. In particular, if $X$ places a white stone somewhere on the board instead of making some jumps, then one of the points $a_i$ or $c_i$ of $G_P(x_i)$ is still unoccupied. Then, $\overline{X}$ reaches $a_i$ immediately or by $e_{i-1} \to c_i \to d_i \to b_i \to a_i$, respectively, and his next jump places the ball at the $X$'s goal line. So, $X$ is forced to make a sequence of jumps. By similar arguments, $X$ proceeds by $e_{i-1} \to a_i \to c_i \to d_i \to e_i$ or by $e_{i-1} \to a_i \to b_i \to d_i \to e_i$, for otherwise $\overline{X}$ can reach $a_i$ and place the ball on the $X$'s goal line. □

The conversion of the formula component $G(F)$ to the configuration of stones is shown in Figs. 5 and 6. In particular, Fig. 5 depicts the board representation of the edges $(x_{j-1}, x_j)$, $(x_j, y_j)$ and $(y_j, z_j)$ while Fig. 6($a$) (respectively Fig. 6($b$)) gives the configuration of stones coding the situation when $w_{j,t}$ is pointing at an edge $(b_l, d_l)$ ($(a_l, c_l)$, respectively), where $j \in \{1, \ldots, m\}, t \in \{1, 2\}$ and $l \in \{1, \ldots, n\}$. For $t = 3$ the construction is analogous, except that the horizontal line of stones to the right of $w_{j,3}$ is missing. See Fig. 7 for an example where $G_P(x_2)$ is given together with $w_{1,2}$ pointing at $(b_2, d_2)$ and $w_{2,3}$, $w_{3,2}$ both pointing at $(a_2, c_2)$. The configuration of stones corresponding to a vertex $z_j$, denoted by $G_P(z_j)$ is identical to the one in Fig. 4($a$), but it is rotated with the angle of 180 degrees, and we use the symbol $z_j$ to refer to the point denoted by $e_i$ in the case of $G_P(x_i)$'s. The configuration of stones corresponding to $G(F)$ is denoted by $G_P(F)$.

It remains to mention that the number of white stones in a line does not change the analysis of the game. In particular the distance between $b_l$ and $d_l$ can be arbitrary large and in our reduction it depends on the number of vertices $w_{j,t}$ pointing at $(b_l, d_l)$ or $(a_l, c_l)$. We require that there is a distance of at least one 'field' between each pair of vertical lines appearing on the board. In this way if one stone, say from a field $z$, in a line between $x$ and $y$ has been removed from the board during the

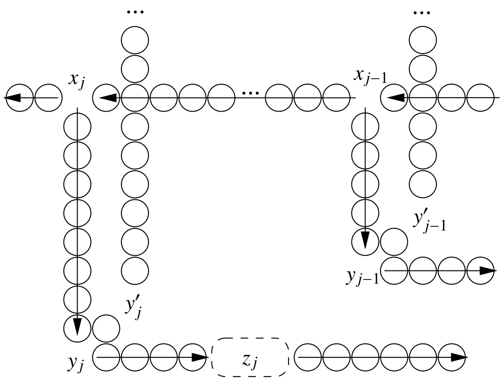

**Fig. 5.** The configuration representing the paths in (4).

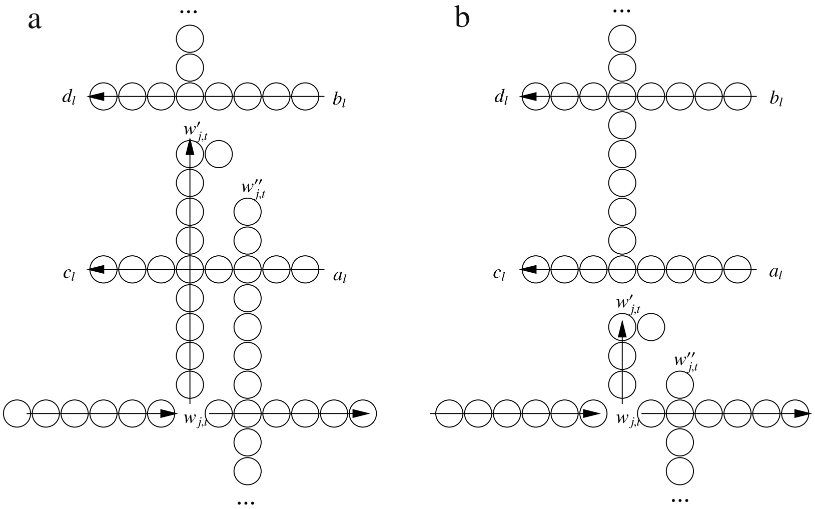

**Fig. 6.** (*a*) $w_{j,t}$ and pointing at $(b_l, d_l)$ and (*b*) $w_{j,t}$ and pointing at $(a_l, c_l)$.

game then a jump $x \to y$, where $y$ may be on the goal line, is replaced by two jumps $x \to z \to y$. Thanks to the distance between the vertical lines, no other jump from $z$ is possible, and a player either ends his move at $z$ or continues to $y$. If a player ends his move at $z$ then such a situation does not differ from the case when the move would end at $y$ and our analysis covers that.

**Proof of Fact 2.** From Fact 1 it follows that when the ball is at point $e_{n-1}$ then the ∀-player is the current player. Furthermore, similarly as in the proof of Fact 1 one can argue that the ball follows the path on the board corresponding to (2) and (3), where $i = n$, and reaches $e_n$. The ∀-player cannot stop here, because the ∃-player plays

$$e_n \to x_1 \to y_1 \to y'_1 \tag{5}$$

and then places the ball at the ∀-player's goal line.

By a simple induction on $i$ one can prove that when the ∀-player reaches the point $x_i$ then he is forced to continue by either jumping to $x_{i+1}$ or to $y_i$. So, there exists $j \in \{1, \dots, m\}$ such that the ∀-player reaches $y_j$. He cannot end his sequence of jumps at $y_j$ or $y'_j$, because, as before, he would immediately lose. By the arguments used in the proof of Fact 1, the moves (i) and (ii) in $G_P(z_j)$ must occur and the ∃-player begins his move when the ball is at $z_j$. □

**Proof of Fact 3.** By Fact 2, the ∀-player reaches $z_j$. (Note that $G_P(z_j)$ is a rotation of a variable component, and for this reason we do not repeat the analysis of the game in $G_P(z_j)$ here, but we just make a list of moves performed by the players.) The ∃-player places a stone at 8 of $G_P(z_j)$, the ∀-player places a stone at 18 of $G_P(z_j)$. Then, the ∃-player is forced to move the ball, because otherwise the ∀-player follows a path

$$z_j \to w_{j,1} \to \cdots \to w_{j,t} \to w'_{j,t} \to w''_{j,t}$$

(see Fig. 6), where $t \in \{1, 2, 3\}$ can be chosen in such a way that none of $w_{j,t}$, $w'_{j,t}$, $w''_{j,t}$ is occupied by a white stone placed by the ∃-player in his last move. So, the ∃-player has three paths to follow:

$$z_j \to w_{j,1} \to \cdots \to w_{j,t} \to w'_{j,t}, \quad t = 1, 2, 3. \tag{6}$$

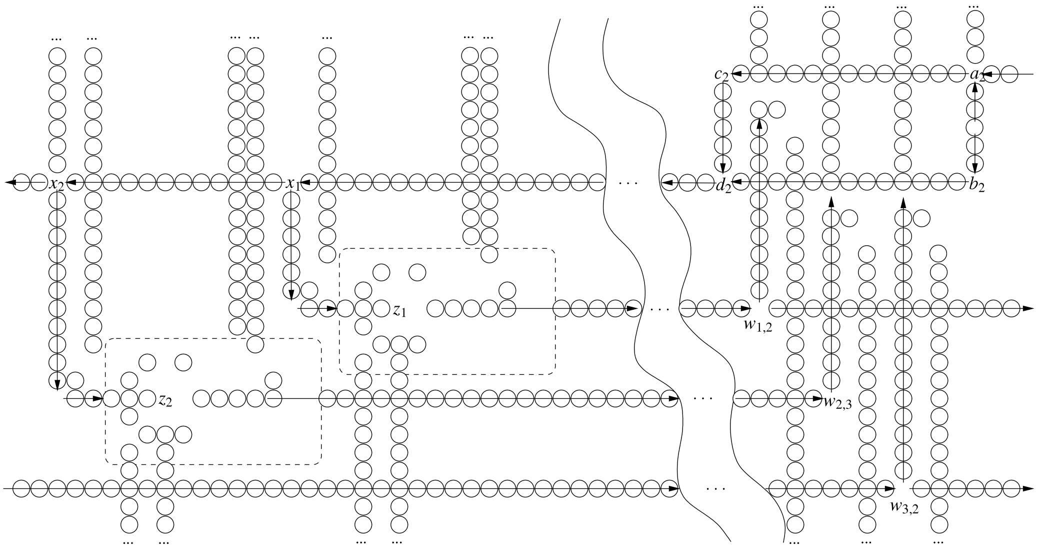

**Fig. 7.** Some parts of the graph in Fig. 2.

If he ends his move at $w'_{j,t}$ (or earlier) then the ∀-player reaches $w''_{j,t}$ and can jump to the ∃-player's board line. There are two cases to consider: (i) $w_{j,t}$ points at an edge $(b_l, d_l)$ in $G$; (ii) $w_{j,t}$ points at an edge $(a_l, c_l)$ in $G$. Both cases are analogous, so we consider only (i). From the construction of the board it follows that the only continuation of moves in (6) by the ∃-player in direction different from $w''_{j,t}$ is possible when no jump from $b_l$ to $d_l$ has been made during the game, and in this case the ∃-player makes a jump from $w'_{j,t}$ to the ∀-player's board line. So the ∃-player wins if and only if no jump from $b_l$ to $d_l$ has been made. This is equivalent to the case when the edge $(b_l, d_l)$ still belongs to the corresponding graph $G$, i.e. $w_{j,t} \in R^{-1}(E(G))$ when $z_j$ is the active vertex. It remains to mention that the line of stones between $a_l$ and $c_l$ has been removed from the board, but the result is that a player makes two jumps instead of one in order to move the ball from $w_{j,t}$ to $w'_{j,t}$ or from $w'_{j,t}$ to the board line. □

Fig. 7 presents some parts of the board obtained on the basis of the graph $G$ corresponding to the formula in (1) and shown in Fig. 2.

**Theorem 2.** *The game of Phutball is* PSPACE*-hard.*

**Proof.** The theorem follows from Facts 1–3, the proof of Theorem 1 and an observation that the size of the board is polynomial in $n + m$. □

## 4. Summary

There are several natural questions one may ask about the complexity of a game. One of them is: given an arbitrary state of the game, is it possible for the current player to win in the next move? Such a problem has been considered in [11] where it has been shown that it is NP-complete for Phutball. According to the discussion in [11] the games of Checkers and Phutball have many similarities. However, it turns out that we can give a positive answer to the above question in the case of Checkers in polynomial time [11,15].

Another question to ask about the complexity of a game is the one considered in this paper. In the case of Checkers, Fraenkel et al. have shown that the game is PSPACE-hard. Their result has been strengthened by the paper of Robson: the game is EXPTIME-complete [23]. In this paper we developed the first result concerning the complexity of Phutball and an open question remains whether it belongs to PSPACE or is as hard as EXPTIME?

## Acknowledgement

The first author is partially supported by the Foundation for Polish Science (FNP).

## References

[1] E.R. Berlekamp, J.H. Conway, R.K. Guy, Winning Ways for your Mathematical Plays, Vol. I, II, Academic Press, London, 1982, 2nd edition: vol. 1 (2001), vol. 2 (2003), vol. 3 (2003), vol. 4 (2004), AK Peters, Natick, MA.
[2] E.R. Berlekamp, J.H. Conway, R.K. Guy, Philosopher's football, in: Winning Ways for Your Mathematical Plays, 2nd edition, A K Peters Ltd, 2003, pp. 752–755.
[3] T.C. Biedl, E.D. Demaine, M.L. Demaine, R. Fleischer, L. Jacobsen, J.I. Munro, The complexity of Clickomania, in: R.J. Nowakowski (Ed.), More Games of No Chance, Cambridge University Press, 2002, pp. 389–404.

[4] T. Cazenave, Iterative Widening, in: Proceedings of IJCAI-01, vol. 1, 2001, pp. 523–528.
[5] T. Cazenave, Admissible moves in two-player games, in: Proceedings of SARA, in: Lecture Notes in Comp. Sci., vol. 2371, 2002, pp. 52–63.
[6] T. Cazenave, A generalized threats search algorithm, in: In Proceedings of the Third International Conference on Computers and Games, in: Lecture Notes in Comp. Sci., vol. 2883, 2003, pp. 75–87.
[7] T. Cazenave, Gradual abstract proof search, ICGA Journal 25 (2002) 3–15.
[8] J.H. Conway, On numbers and games, Academic Press, London, 1976.
[9] J. Culberson, Sokoban is PSPACE-complete, in: Proceedings of the International Conference on Fun with Algorithms, 1998, pp. 65–76.
[10] E.D. Demaine, Playing games with algorithms: algorithmic combinatorial game theory, in: Proc. 26th Symposium on Mathematical Foundations in Comp. Sci., in: Lecture Notes in Comp. Sci., vol. 2136, 2001, pp. 18–32.
[11] E.D. Demaine, M.L. Demaine, D. Eppstein, Phutball endgames are hard, in: R.J. Nowakowski (Ed.), More Games of No Chance, Cambridge University Press, 2002, pp. 351–360.
[12] E.D. Demaine, M.L. Demaine, R. Fleischer, R.A. Hearn, T. von Oertzen, The complexity of Dyson Telescopes, in: M.H. Albert, R.J. Nowakowski (Eds.), Games of No Chance 3, in: MSRI Publications, vol. 56, Cambridge University Press, 2009, pp. 271–285.
[13] S. Even, R.E. Tarjan, A combinatorial problem which is complete in polynomial space, Journal of the Association for Computing Machinery 23 (1976) 710–719.
[14] G.W. Flake, E.B. Baum, Rush Hour is PSPACE-complete, or Why you should generously tip parking lot attendants, Theoretical Computer Science 270 (2002) 895–911.
[15] A.S. Fraenkel, M.R. Garey, D.S. Johnson, T. Schaefer, Y. Yesha, The complexity of Checkers on an $N \times N$ board, in: Proc. 19th Annual Symposium on Foundations of Comp. Sci., 1978, pp. 55–64.
[16] T. Furtak, M. Kiyomi, T. Uno, M. Buro, Generalized Amazons is PSPACE-Complete, in: L.P. Kaelbling, A. Saffiotti (Eds.), IJCAI, 2005, pp. 132–137.
[17] J.P. Grossman, R.J. Nowakowski, One-dimensional Phutball, in: R.J. Nowakowski (Ed.), More Games of No Chance, in: MSRI Publications, vol. 42, Cambridge University Press, 2002, pp. 361–367.
[18] R.A. Hearn, Games, puzzles, and computation, Ph.D. Thesis, Massachusetts Institute of Technology, 2006.
[19] S. Iwata, T. Kasai, The Othello game on $n \times n$ board in PSPACE-complete, Theoretical Computer Science 123 (1994) 329–340.
[20] R. Kaye, Minesweeper is NP-complete, Mathematical Intelligencer 22 (2000) 9–15.
[21] D. Lichtenstein, M. Sipser, Go is polynomial-space hard, Journal of the Association for Computing Machinery 27 (1980) 393–401.
[22] J.M. Robson, The complexity of Go, in: Proceedings of the IFIP 9th World Computer Congress on Information Processing, 1983, p. 413–417.
[23] J.M. Robson, $N$ by $N$ Checkers is EXPTIME complete, SIAM Journal on Computing 13 (1984) 252–267.
[24] T.J. Schaefer, On the complexity of some two-person perfect-information games, Journal of Computer and System Sciences 16 (1978) 185–225.
[25] A.N. Siegel, Loopy games and computation, Ph.D. Thesis, University of California at Berkeley, 2005.
[26] L.J. Stockmeyer, A.R. Meyer, Word problems requiring exponential time, in: Proceedings 5th ACM Symposium on the Theory of Computing, ACM Press, 1973, pp. 1–9.
[27] R. Uehara, S. Iwata, Generalized Hi-Q is NP-complete, Transactions of the IEICE 73 (1990) 270–273.
[28] D. Wolfe, Go endgames are hard, in: R.J. Nowakowski (Ed.), More Games of No Chance, in: MSRI Publications, vol. 42, Cambridge University Press, 2002, pp. 125–136.