Journal of Game Theory 2019, 8(2): 25-31
DOI: 10.5923/j.jgt.20190802.01

Algorithm for Multilple Player Last Nim for Any Alliance System

Essam El_Seidy*, Hassan El Kady, Dieaa I. Nassr

Department of Mathematics, Faculty of Science, Ain Shams University, Egypt

Abstract The Last Nim game is an impartial combinatorial game studied only in the case of the standard alliance matrix. In this paper, we consider the Last Nim game with N linearly ordered nonempty piles containing counters, and n players for any alliance matrix. For this, we give an algorithm to get the winner, the tree of all possibilities of the game, and the best strategy for the winner. For the practical result, we implement this algorithm using C++ language and give some examples.

Keywords Game Theory, Combinatorial game, Nim Game and MLNim(N,n)

1. Introduction

Combinatorial game theory is an important branch of game theory and a pure strategy game with no chance. It is played by two or more players who play alternately, they can move to a finite number of positions. All players have a complete knowledge of the game, such as Hackenbuch, Chess, Tic Tac Toe and Nim. W.A.Liu and J.Yang [1] introduced the multi-player last Nim when the standard alliance matrix is adopted.

E.El-Seidy, S.E.S.Hussein and A.T.Alabdala [2], that is, the combinatorial game is with perfect information. The determination of the winner in the combinatorial game is based on normal play or misere play. In the normal game, the player who makes the last legal move wins, while in the misere play, the player making the last legal move makes the next player the winner. There are two main types of combinatorial play, impartial play and partizan play. In the impartial game, all players have the same option and the same position. e.g., Nim and Hachenbuch . In the game of partizan, each player has a different position that can move and all players play according to the same rules. e.g., Chess and Tic Tac Toe. The Nim game is an impartial game played alternately by two players. The game is a set of piles where these piles are ordered linearly and in each pile a number of counters. The player step is taken by removing any number of piles from any pile. Nim game is introduced by Bouton [3], Albert MH and Nowakowski [4], Flammenkamp et al. [5], Holshouser et. al. [6], Liu and Zhao [7]. There are many types for the Nim games, We conclude these results as follows:

- *Circular Nim*, this game was introduced by Matthieu Dufour and Silvia [8] is an impartial combinatorial game consisting of several piles of counters placed in a circle. The player chooses m consecutive piles. The player removes at least one counter from one or more piles of the m consecutive piles. The player who makes the last move wins.
- *The Classical Game of Fibonacci Nim*, this game was introduced by Michael J. Whinihan [9], there is a pile of n stones (counters) and players make moves by removing stones under certain constraints. A player must remove at least one stone from the pile and if a player removes m stones, the next player should remove at most 2m stones. The first player cannot remove all the stones.
- *Nim on Graph* by M. Fukuyama [10] is an impartial game played by two players. It contains an undirected graph in which each edge is assigned to a positive integer. To start the game, place an indicator at a vertex called the starting position. Both players take turns and the move can be summarized as follows: Choose an edge incident with the vertex containing the indicator piece. Decrease this edge to any strictly smaller non-negative integer. Move the piece to the adjacent vertex along this edge. If each edge incident with the piece has zero value, then the player to play is the loser and the game is over.
- *Small Nim game*, this game was introduced by W.A.Liu and J.J. Zhou [11], in this game there are linearly ordered piles containing counters where a player makes a move by removing any number of counters in a pile of the smallest number of counters. The player who cannot make a move loses the game.

* Corresponding author:
esam elsedy@hotmail.com (Essam El_Seidy)
Published online at http://journal.sapub.org/jgt

- _ End Nim game, this game was introduced by Albert MH and Nowakowski RJ [12], there are two players called Left and Right and a vector (string) of nonempty piles containing counters. Both players take turns by removing one or more counters, the player on the left removing counters from the leftmost nonempty pile and the player on the right removing counters from the rightmost nonempty pile, unless do not stay a pile. The player who can not make a move loses.
- *Last Nim game*, this game was introduced by Friedman E. [13], there are linearly ordered piles containing counters where a player takes his turn by removing any number of counters from the last pile. In the normal play, the player who makes the last move wins. While in misere play, the last player who cannot make a move wins.

In recent years, games of two players have been the subject of a comprehensive study.

These games are normal to be generalized to three players or even to $n \geq 3$ players to be called multiplayer games. In addition, multiplayer games are studied in different forms. For example, multi-player without alliance [14-18] multi-player with two alliances [19,20] and multi-players with a system of alliances [22,23].

In this article, we give Algorithms 3.1, 3.2 and 3.3 to determine the game value function of the multi-player Last Nim game for any alliance matrix. In addition, the _rst move can be made by any player. i.e., it is not necessary for P0 to perform the first move. As a result, the game value function and the winner player are modified according to the player who makes the first move. Therefore, we denote the player who make the first move in our implementation by P_f .

The paper is organized as follows. In Sect 2, we give some basic concepts of the multi-player last Nim game. New Algorithms to compute the game value function of the Last multi-player last Nim game for any alliance matrix are presented in Sect 3. In Sect 4, we give the result of our implementation. Section 5 includes the conclusion.

2. Multi-player Last Nim with Alliance System

The following definitions are to summarize the last Name with alliance alliance system [1, 22]

Definition 2.1 (Multiple players Last Nim). Given linearly ordered N piles $(x_1, x_2, \dots, x_N)$ where each pile has counters. There are n players who take turn by removing any number of counters from the last nonempty pile (containing at least one counter). If each pile has zero value, then the player to play is the winner and the game is over. We denote this game by $MLNim(N, n)$.

Definition 2.2 (Krawec [22, 23]).

(i) Assume that the players are given by $P_0, P_1, \dots, P_n$. The player P_0 makes the first move followed by P_1 and so on. The Player P_0 has to play again after P_{n-1} has played. i.e., $P_{i+1\ (mod\ n)}$ is the next player after P_i .

(ii) For a game G with n players, the game value $g(G, i)$ is a non-negative integer less than n that specifes the winner in relation to the player pi. , that is, if $g(G, i) = j$ then the player $P_{i+j\ (mod\ n)}$ has a winning strategy.

(iii) An endgame, denoted by _ is a game which has no legal move is available.

(iv) The options of G, denoted by opt(G) is the set of all cases the current player can move to it.

Definition 2.3.

(i) For a game G with n players, an alliance matrix is an $n \times n$ matrix which takes the form:

$$\begin{pmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,n-1} \\ A_{1,0} & A_{1,1} & & A_{1,n-1} \\ \vdots & & \ddots & \vdots \\ A_{n-1,0} & A_{n-1,1} & \cdots & A_{n-1,n-1} \end{pmatrix}$$

where $A_{i,j} \in \{0, 1, \dots, n-1\}$ The row $A_{i,0}, A_{i,1}, \dots, A_{i,n-1}$ gives the linear order of priority of the desired winner relative to the player P_i where the player P_i prefers the player $P_{i+A_{i,j}\ (mod\ n)}$ more than the player $P_{i+A_{i,j}\ (mod\ n)}$ if $j < j^{'}$.

(ii) The Standard Alliance Matrix, denoted by SAM in which $A_{i,j} = j$, i.e., the player P_i prefers player $P_{i+j\ (mod\ n)}$ more than the player $P_{i+j^{'}\ (mod\ n)}$ if $j < j^{'}$ This matrix takes the following form:

$$\begin{pmatrix} 0 & 1 & \cdots & n-1 \\ 0 & 1 & & n-1 \\ \vdots & & \ddots & \vdots \\ 0 & 1 & \cdots & n-1 \end{pmatrix}$$

Example 2.4.

If $n = 3$ and
P_0 prefers P_1 over P_0 over P_2.
P_1 prefers P_0 over P_2 over P_1.
P_2 prefers P_0 over P_1 over P_2.

Definition 2.5 (The Game Value Function [22]). The game value function is given by $g : C_G \times \mathbb{Z}_n \to \mathbb{Z}_n$ (where C_G denotes the set of all impartial combinatorial games, $\mathbb{Z}_n = \{0,1,2, \dots, n-1\}$)

$$g(G, i) = \begin{cases} 0, & G = \emptyset, \\ A_{i,l}, & other\ wise. \end{cases}$$

where:

$$l = \min\{i \in \mathbb{Z}_n \mid g(G^{'}, i+1) + 1 = A_{i,j}\ with\ G^{'} \in Opt(G) \}$$

Wen An Liu and Juan Yang (2017) [1] generalized the game value function for any game position for $MLNim(N, n)$ as a vector $\boldsymbol{p} = (x_1, x_2, \dots, x_N)$ that can move in two ways. to it, which are $\boldsymbol{p_0} = (x_1, x_2, \dots, x_{N-1})$ and $\boldsymbol{p_m} = \{(x_1, x_2, \dots, x_{N-1}, m) \mid 1 \leq m \leq x_N - 1\}$ therefore the game value function will take the following form:

$$g(G,i) = \begin{cases} 0, & \boldsymbol{p} = \emptyset, \\ \min\{g(\boldsymbol{p_t}) + 1 \mid 1 \le m \le x_N - 1\}, & other\ wise. \end{cases}$$

3. Algorithm

In this section, we present algorithms that can be used to compute the game value function of the Last Nim multi-player game for any alliance matrix and the first move can be performed by any player. We describe the formate of game state in every move and then present Algorithms 3.1 and 3.2 to build the tree of all possible moves (cases) of the game recursively. After that, we use this tree as input for Algorithm 3.3 to compute the game value function. i.e., find the winner. Let the state of the game be described by the player who has the order to move and counters remaining in each pile during the game. We write the state as $(i, [b_1, b_2, \dots, b_N])$, where i is the player number (P_i is the player to move), N is the number of piles and $b_1, b_2, \dots, b_N$ is a list of piles' counter.

Algorithm 3.1.

Input: n is the number of players, a list $[b1, b2, \dots, bN]$ of piles' counter. i.e. b_i is the counter in the pile $i, 1 \le i \le N$

Output: tr is the tree of all possible cases of the game.

```
Begin
    $state = (0, [b_1, b_2, \dots, b_N])$
    set state to be the root of tr
    call Algorithm 3.2 for the root state $= (0, [b_1, b2, \dots, b_N])$
    return tr
End
```

In the following algorithm, we have the pile bi will be emptied before the pile b_{i-1}, $i = 2, \dots, N - 1$.

Algorithm 3.2.

Input: n is the number of players, the current state = (k; [b1; b2; : : : ; bN]); where Pk is the current player and $[b_1, b_2, \dots, b_N]$ is a list of piles' counter.

Output: recursively return the tree of all possible cases.

```
Begin
    if game over then
        return
    end if
    $k' = (k + 1)\ (mod\ n)$ . ▷ $k'$ is the next player
    $j$ is the last nonempty pile.
    for $i = 1$ to $b_j$ do
        append child $stat\,e_i = (k', [b_1, \dots, b_j - i, \dots, b_N])$ for the current state
        recursive call Algorithm 3.2 for the child $stat\,e_i = (k', [b_1, \dots, b_j - i, \dots, b_N])$.
    end for
End
```

We give Algorithm 3.3 to find the winner.

Algorithm 3.3.

Input: tr is the tree of all possible moves (cases) of the game, n is the number of players,

M is the alliance matrix, root is the root of tr.

Output: the winner of tr.

```
Begin
    let $m$ be the number of children of root
    if $m = 0$ then
        return the winner is root
    end if
    define $c_1, c_2, \dots, c_m$ to be the children of root
    for $k = 1$ to $m$ do. ▷ looping through children
        let trk be the subtree with root $c_k$
        recursively call Algorithm 3.3 for the subtree $tr_k$
    end for
    let rootId be the player in root
    for $i = 0$ to $n - 1$ do. ▷ looping through players according priority
        for $k = 1$ to $m$ do. ▷ looping through winners of subtrees
            let $w_k$ be the winner of $tr_k$
            if $(M[rootId][i] + rootId)\ (mod\ n) = w_k$ then
                return the winner state is $w_k$
            end if
        end for
    end for
End
```

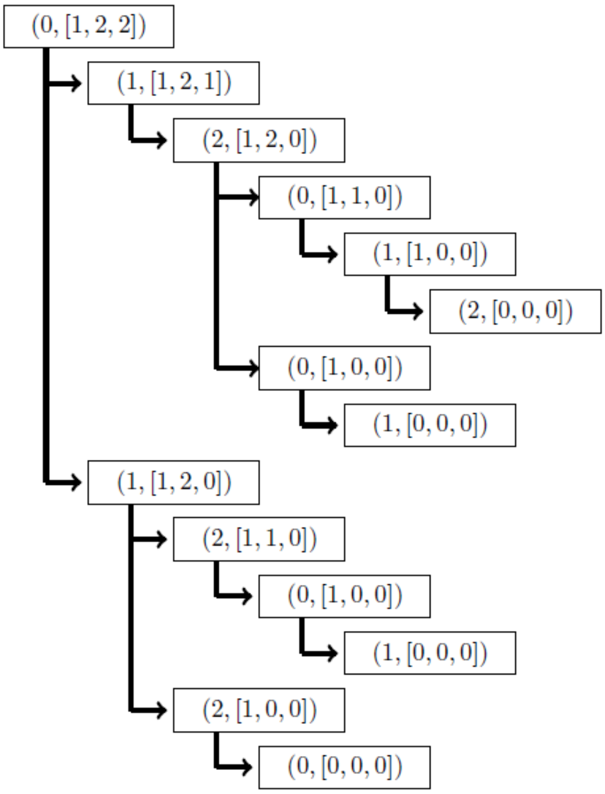

Figure 1. Tree of All Possible Cases

Example: Let the number of players be 3, the number of piles equal to 3, the list of counters be 2,2,1, and the alliance matrix be

$$\begin{pmatrix} 0 & 2 & 1 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix}$$

In the case where 0 is the _rst player, we have the tree of all possible cases that is constructed, recursively, by Algorithms 3.1 and 3.2 as in Figure 1.

The winner is the player 2; where the game steps are:

$$(0,[1,2,2]),(1,[1,2,1]),(2;\ [1,2,0]),(0,[1,1,0]),$$
$$(1,[1,0,0]),(2,[0,0,0])$$

4. Implementation

We give a simple implementation of Algorithms 3.1, 3.2 and 3.3. The implementation was written in the C++ language. Our platform was Pentium IV 3.2 GHz with windows operating system, we present some results of our implementation for multiple players last Nim game with different cases of N (of piles) and n (of players). In addition, we have examples of last Nim game with different alliance matrices and we indicate that any change in the alliance matrix leads to a change in the game value function and the winner. The result can be summarized as the following:

1. In table 1, we use our implementation to find the game value of multiple players last Nim with different alliance matrices in the case $n > N + 1$: Accordingly, we confirm that the output of our implementation agree with the result of W.A.Liu and J.Yand [1] in the case of the standard alliance matrix.

2. In table 2, we consider the case n = N +1; then we compute the game value function and the winner according to various alliance matrices. The outputs confirm the resultof W.A.Liu and J.Yang [1]. In this case, the result of W.A.Liu and J.Yang [1] gives the game value function equal to the number of piles if the last pile contains one counter and if the last pile contains several counters, it equals to zero., i.e.

$$g(\boldsymbol{p}) = \begin{cases} N, & if\ x_N = 1, \\ 0, & if\ x_N > 1. \end{cases}$$

3. Wen An Liu and Juan Yang [1] studied the game value in the case of $N \geq n$ and defined the variable d the difference between the number of piles and the number of players, i.e. $d = N - n$: In this case, we denote the game by $MLNim(n + d, n)$ and give examples for various alliance matrices.

(a) In table 3, we give the output of the implementation for $MLNim(n+d,n)$; $d \geq 0$ and $n \geq d+3$: The output confirms the result of W.A.Liu and J.Yang [1], in this case, the game value is given by:

$$g(\boldsymbol{p}) = \begin{cases} d, & if\ x_{n-1} = 1, \\ d+1 & if\ x_{n-1} > 1. \end{cases}$$

(b) In table 4, we give the output of the implementation for $MLNim(n + d, n)$ and $n = d + 2 \geq 3$: The output confirms the result of W.A.Liu and J.Yang [1], where in this case, the game value is given by:

$$g(\boldsymbol{p}) = \begin{cases} d, & if\ x_{n-1} = 1, \\ d+1, & if\ x_{n-1} > 1\ and\ x_{n+d} = 1, \\ d+2, & if\ x_{n-1} > 1\ and\ x_{n+d} > 1. \end{cases}$$

(c) In table 5, we give the output of the implementation for $MLNim(n + d, n)$ and $n = d + 1 \geq 3$. The output confirms the result of W.A.Liu and J.Yang [1], where in this case, the game value is given by:

$$g(\boldsymbol{p}) = \begin{cases} d, & if\ x_{n-1} = 1\ and\ x_{n+d} = 1, \\ d+1, & if\ (x_{n-1} = 1\ and\ x_{n+d} > 1)\ or\ (x_{n-1} > 1\ and\ x_{n+d-1} = 1), \\ d+2, & if\ x_{n-1} > 1\ and\ x_{n+d-1} > 1. \end{cases}$$

4. Wen An Liu and Juan Yang [1] give the game value function for $MLNim(6,3)$ $(N = 6, n = 3)$ where the alliance matrix is the standard matrix by:

$$g(p) = \begin{cases} 2, & if\ x_2 > 1\ and\ x_4 > 1\ and\ x_6 = 1, \\ 0, & if\ (x_2 > 1 and\ x_4 > 1 and\ x_6 > 1)\ or\ (x_2 = 1 and\ x_5 = 1), \\ 1, & if\ (x_2 = 1\ and\ x_5 > 1)\ or\ (x_2 > 1 and\ x_4 = 1). \end{cases}$$

Table 1. Last Nim when $n > N + 1$

no. row	n	N	Alliance Matrix	G	P_f	$g(G,i)$	winner
1	4	2	SAM	(4 3)	P_2	2	P_2
2	4	2	$\begin{pmatrix} 2 & 1 & 3 & 0 \\ 3 & 1 & 2 & 0 \\ 0 & 3 & 2 & 1 \\ 1 & 0 & 3 & 2 \end{pmatrix}$	(4 3)	P_0	0	P_0
3	4	2	$\begin{pmatrix} 2 & 1 & 3 & 0 \\ 3 & 1 & 2 & 0 \\ 0 & 3 & 2 & 1 \\ 1 & 0 & 3 & 2 \end{pmatrix}$	(4 3)	P_2	0	P_2
4	6	4	$\begin{pmatrix} 5 & 2 & 4 & 3 & 0 & 1 \\ 5 & 4 & 3 & 1 & 2 & 0 \\ 3 & 5 & 4 & 0 & 2 & 1 \\ 3 & 4 & 5 & 0 & 1 & 2 \\ 0 & 1 & 5 & 2 & 4 & 3 \\ 0 & 1 & 5 & 2 & 4 & 3 \end{pmatrix}$	(1 2 3 2)	P_0	0	P_0
5	6	4	$\begin{pmatrix} 5 & 2 & 4 & 3 & 0 & 1 \\ 5 & 4 & 3 & 1 & 2 & 0 \\ 3 & 5 & 4 & 0 & 2 & 1 \\ 3 & 4 & 5 & 0 & 1 & 2 \\ 0 & 1 & 5 & 2 & 4 & 3 \\ 0 & 1 & 5 & 2 & 4 & 3 \end{pmatrix}$	(1 2 3 2)	P_3	1	P_4

Table 2. Last Nim when $n = N + 1$

no. row	n	N	Alliance Matrix	G	P_f	$g(G,i)$	winner
1	3	2	SAM	(2 4)	P_0	0	P_0
2	3	2	$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 1 \\ 2 & 1 & 0 \end{pmatrix}$	(2 4)	P_0	1	P_1
3	3	2	$\begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 1 \\ 2 & 1 & 0 \end{pmatrix}$	(2 4)	P_2	2	P_1
4	5	4	$\begin{pmatrix} 4 & 3 & 2 & 1 & 0 \\ 3 & 4 & 2 & 0 & 1 \\ 3 & 1 & 0 & 2 & 4 \\ 0 & 3 & 2 & 1 & 4 \\ 4 & 0 & 2 & 1 & 3 \end{pmatrix}$	(2 1 2 3)	P_0	3	P_3
5	5	4	$\begin{pmatrix} 4 & 3 & 2 & 1 & 0 \\ 3 & 4 & 2 & 0 & 1 \\ 3 & 1 & 0 & 2 & 4 \\ 0 & 3 & 2 & 1 & 4 \\ 4 & 0 & 2 & 1 & 3 \end{pmatrix}$	(2 1 2 3)	P_3	0	P_3

Table 3. Last Nim when $N = n + d, n \geq d + 3, and\ d \geq 0$

no. row	n	d	Alliance Matrix	G	P_f	$g(G,i)$	winner
1	3	0	SAM	(1 2 3)	P_1	0	P_1
2	3	0	$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{pmatrix}$	(1 2 3)	P_0	2	P_2
3	3	0	$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 2 & 1 & 0 \end{pmatrix}$	(1 2 3)	P_1	1	P_2
4	4	1	$\begin{pmatrix} 3 & 0 & 1 & 2 \\ 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 \\ 1 & 0 & 3 & 2 \end{pmatrix}$	(2 2 3 1 2)	P_0	0	P_0
5	4	1	$\begin{pmatrix} 3 & 0 & 1 & 2 \\ 0 & 2 & 1 & 3 \\ 3 & 2 & 0 & 1 \\ 1 & 0 & 3 & 2 \end{pmatrix}$	(2 2 3 1 2)	P_3	1	P_0

Table 4. Last Nim when $N = n + d;\ n = d + 2\ \_\ 3\ and\ d \geq 1$

no. row	n	d	Alliance Matrix	G	P_f	$g(G,i)$	winner
1	3	1	SAM	(3 4 2 3)	P_0	0	P_0
2	3	1	$\begin{pmatrix} 2 & 1 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix}$	(3 4 2 3)	P_0	0	P_0
3	3	1	$\begin{pmatrix} 2 & 1 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 2 \end{pmatrix}$	(3 4 2 3)	P_1	2	P_0
4	4	2	$\begin{pmatrix} 0 & 1 & 3 & 2 \\ 1 & 0 & 3 & 2 \\ 0 & 1 & 3 & 2 \\ 3 & 2 & 0 & 1 \end{pmatrix}$	(3 2 1 1 2 3)	P_0	1	P_1
5	4	2	$\begin{pmatrix} 0 & 1 & 3 & 2 \\ 1 & 0 & 3 & 2 \\ 0 & 1 & 3 & 2 \\ 3 & 2 & 0 & 1 \end{pmatrix}$	(3 2 1 1 2 3)	P_1	1	P_2

Table 5. Last Nim when $N = n + d;\ n = d + 1 \geq 3\ and\ d \geq 2$

no. row	n	d	Alliance Matrix	G	P_f	$g(G,i)$	winner
1	3	2	SAM	(2 2 3 4 2)	P_0	1	P_1
2	3	2	$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 0 & 2 \end{pmatrix}$	(2 2 3 4 2)	P_0	0	P_0
3	3	2	$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 0 & 2 \end{pmatrix}$	(2 2 3 4 2)	P_2	1	P_0
4	4	3	$\begin{pmatrix} 0 & 3 & 2 & 1 \\ 2 & 1 & 0 & 3 \\ 2 & 1 & 0 & 3 \\ 3 & 2 & 1 & 0 \end{pmatrix}$	(3 2 2 1 2 2 1)	P_0	3	P_3
5	4	3	$\begin{pmatrix} 0 & 3 & 2 & 1 \\ 2 & 1 & 0 & 3 \\ 2 & 1 & 0 & 3 \\ 3 & 2 & 1 & 0 \end{pmatrix}$	(3 2 2 1 2 2 1)	P_2	2	P_0

Table 6. Last Nim when $N = 6\ and\ n = 3$

no. row	n	N	Alliance Matrix	G	P_f	$g(G,i)$	winner
1	3	6	SAM	(1 2 3 2 1 2)	P_0	0	P_0
2	3	6	$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 2 & 0 \end{pmatrix}$	(1 2 3 2 1 2)	P_0	0	P_0
3	3	6	$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 1 & 2 & 0 \end{pmatrix}$	(1 2 3 2 1 2)	P_2	1	P_0
4	3	6	$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix}$	(1 2 3 2 1 2)	P_0	0	P_0
5	3	6	$\begin{pmatrix} 0 & 2 & 1 \\ 2 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix}$	(1 2 3 2 1 2)	P_2	2	P_0

5. Conclusions

In this paper, we proposed an algorithm for the latest Nim multi-player game with any alliance system. In [1], W.A.Liu and J.Yang have studied the game value function in the case of the standard alliance matrix with the number of piles N and the number of players n : To the best of our knowledge, no results were taken into account with an alliance matrix different from the standard one. Therefore, the algorithm proposed in this article is the first attempt to calculate the game value function of the Last Nim multiplayer game for any alliance matrix. We implemented this algorithm in C ++ language. The entries for our implementation are the initial state of the Nim game, the alliance matrix, and the player making the first move. The outputs are the winning player, the tree of all possible moves of the game and the winning strategy to which the winning player must play. We also performed the implementation in the case of the standard alliance matrix in order to compare our result with those of W.A.Liu and J.Yang and we saw that the result is identical in the different cases of N and n.

REFERENCES

[1] W.A. Liu, J. Yang (2017) Multi-player Last Nim with Passes. Int J Game Theory.

[2] E. El-Seidy, S.E.S. Hussein and A.T. Alabdala (2016) Models of Combinatorial Games and Some Applications: A Survey. Journal of Game Theory, 5(2): 27-41.

[3] C.L. Bouton CL (1901) Nim, a game with a complete mathematical theory. Ann Math 3:3539.

[4] M.H. Albert, R.J. Nowakowski RJ (2004) Nim restrictions. Int Electron J Combin Number Theory 4:Article G01.

[5] A. Flammenkamp ,A. Holshouser, H.Reiter(2003) Dynamic one-pile blocking Nim.Electr JCombin 10:Article N4.

[6] A. Holshouser, H. Reiter, J. Rudzinski (2003) Dynamic one-pile Nim. Fibonacci Q 41(3):253-262.

[7] W.A. Liu, X. Zhao (2016). Nim with one or two dynamic restrictions. Discr Appl Math 198:4864.

[8] M. Dufour and S. Heubach (2013), Circular Nim Games,The electronic journal of combinatorics 20(2).

[9] M.J. Whinihan (1963), Fibonacci nim. Fibonacci Quart, 1(4):913.

[10] M. Fukuyama(2003), A Nim game played on graphs, Theoret. Comput. Sci..

[11] W.A. Liu and J.J. Zhou (2017) Multi-player Small Nim with Passes.Discrete appliedmathematics.

[12] M.H. Albert, R.J. Nowakowski (2001) The game of End-Nim. Electr J Combin 8:Article R1.

[13] E. Friedman (2000) Variants of Nim. Math Magic Game Arch. Accessed Nov 2000.

[14] S.Y.R. Li (1978) N-person Nim and n-person Moores Games. Int J Game Theory 7:3136.

[15] P.D. Straffin (1985) Three-person winner-take-all games with Mc-Carthys revenge rule. Coll JMath 16:386 394.

[16] D.E. Loeb (1996) Stable winning coalitions. In: R.J.Nowakowski (ed) Proc MSRI workshop on combinatorial games, games of no chance 29:451471. Cambridge University Press, Cambridge.

[17] J. Propp (1996) Three-player impartial games. Theor Comput Sci 233:263278.

[18] A. Cincotti (2010) N-player partizan games. Theor Comput Sci 411:32243234.

[19] A.R. Kelly (2006) One-pile misre Nim for three or more players. Int J Math Math Sci 2006:18.

[20] A.R. Kelly (2011) Analysis of one pile misre Nim for two alliances. Rock Mt J Math 41(6):18951906.

[21] X. Zhao, W.A. Liu (2016) One pile misre bounded Nim with two alliances. Discr Appl Math 214:1633.

[22] W.O. Krawec (2012) Analyzing n-player impartial games. Int J Game Theory 41:345367.

[23] W.O. Krawec(2015) n-Player impartial combinatorial games with random players. Theo rComput Sci 569:1 12.

[24] A.S. Fraenkel , M. Lorberbom (1991) Nimho_ games. J Combin Theory Ser A 58:1{25.

[25] R.K. Guy, R.J. Nowakowski (2009) Unsolved problems in combinatorial games. Int Games of no chance 3, Math. Sci. Res. Inst. Publ, vol 56. Cambridge Univ Press, Cambridge, pp 475500.

[26] R.E. Morrison, E.J. Friedman, A.S.Landsberg (2012) Combinatorial games with a pass: a dynamical systems approach. arXiv:1204.3222.

[27] R.M. Low, W.H. Chan (2015) An atlas of N- and P-positions in Nim with a pass. Int Electron J Combin Number Theory 15:Article G02.

[28] W.A. Liu, J.W. Duan, Misre Nim with multi-player, Discrete Appl. Math. 219 (2017) 4050.

[29] W.A. Liu, H. Li, General restriction of (s, t)-Wytho..s game, Electron. J. Combin. 21. (2) (2014) P2.44.

[30] W.A. Liu, H. Li, B. Li, A restricted version of Wytho..s game, Electron. J. Combin. 18 (1) (2011) P207.

[31] W.A. Liu, M.Y. Wang, Multi-player subtraction games, Theoret. Comput. Sci. 659 (2017) 1435.

[32] W.A. Liu, X. Zhao, Adjoining to (s, t)-Wythoffs game its P-positions as moves, Discrete Appl. Math. 179 (2014) 2843.

[33] W.A. Liu, M. Zhao, General restrictions of Wytho..-like games, Theoret. Comput. Sci 602 (2015) 8088.

[34] W.A. Liu, X. Zhao, Nim with one or two dynamic restrictions, Discrete Appl. Math. 198 (2016) 4864.