PSPACE-completeness of the Weighted Poset Game

Hiro Ito Satoshi Takata
School of Informatics, Kyoto University, Kyoto 606-8501, Japan

Abstract Poset game, which includes some famous games, e.g., Nim and Chomp as sub-games, is an important two-player impartial combinatorial game. The rule of the game is as follows: For a given poset (partial ordered set), each player intern chooses an element and the selected element and it’s descendants (elements succeeding it) are all removed from the poset. A player who choose the last element is the winner. On the complexity of poset game, although it is clearly in PSPACE, it have not known whether it is in P or NP-hard. Recently a weighted poset game, which is a generalization of poset game, have been presented and it was found that some sub-games of it can be solved in polynomial-time. The complexity of this game is also open. This paper shows that weighted poset game is PSPACE-complete even if the weights are restricted in \{1, -1\}, the dag, which represents the poset, is bipartite, and the length of each path in the dag is at most two.

Keywords Poset game, computational complexity, winning strategy, PSPACE-complete, dag

1 Introduction

Poset game [1] is an impartial two-player combinatorial game played on a poset (= partial ordered set) \((S, \preceq)\). Each player in turn takes an element of \(S\) and the element chosen and it’s descendants (elements succeeding it) are all removed from the poset. A player who select the last element is a winner. Poset game is a general game and it includes some well-known games as sub-games, e.g., Nim and Chomp (see [2] for details).

If a poset is fixed, clearly one of the players has a winning strategy. For a given poset, which player has a winning strategy can be solved in polynomial space, since the maximum move is at most the number of the elements. Thus poset game is in PSPACE. However whether it is in P or NP-hard is still open. Resolving it seems an important and challenging open problem.

The authors presented a more general game, named Weighted Poset Game [2], which is defined as follows. We give an integer weight \(w(x)\) for each element \(x \in S\) of a poset \((S, \preceq)\), and each player \(i \in \{1, 2\}\) has his/her own non-negative number \(L_i\) of lives. When a player \(i\) deletes a set \(S' \subseteq S\) of elements, the sum of weights of these elements \(\sum_{x \in S'} w(x)\) is decreased from \(L_i\), i.e., \(L_i\) is replaced with \(L_i - \sum_{x \in S'} w(x)\). A player whose lives become negative first is a loser. An instance of a weighted poset game is represented by \((S, \preceq, w, L_1, L_2)\). An element having a positive weight is called a poison element, and an element having a negative weight is called a medicine element.
It is easy to see that this game is a generalization of poset game, since if a poset game \((S, \preceq)\) is given, by adding an extra element \(s \notin S\) preceding every elements in \(S\), and giving weights and lives as \(w(s) = 1\), \(w(x) = 0\) for \(x \in S\), and \(L_1 = L_2 = 0\), the poset game is represented by the weighted form.

**WEIGHTED POSET GAME** is a problem for determining which player has a winning strategy for a given weighted poset game \((S, \preceq, w, L_1, L_2)\).

Some properties on winning strategies of this game are given in [2], e.g., it gives a polynomial time algorithm for calculating a winning strategy for the case if the partial order is a total order. On the other hand, however, from a viewpoint of lower bounds, nothing without trivial results have been known. Although it is clearly in PSPACE from the same reason of poset game, we don’t know whether or not it is PSPACE-complete so far. This paper solves this problem.

Before explaining our theorem, we explain a *dag representation* of a poset. A poset can be represented by a *dag* (= directed acyclic graph). Let \(D = (V, A)\) be a given dag, where \(V\) is a vertex set and \(A\) is an arc set, representing a poset \((S, \preceq)\). An arc \((x, y)\) is in \(A\) means that element \(x\) precedes element \(y\), i.e., \(x \preceq y\). In this paper an arc which can be obtained by using the transitive law (we call it a *redundant arc*) may be omitted for making the representation simple, e.g., if arcs \((x, y)\) and \((y, z)\) are both in \(A\), then from the transitive law arc \((x, z)\) should be in \(A\) also, and hence \((x, z)\) is redundant and it can be omitted in the representation. If \(D\) includes no redundant arc, then it is called the *canonical dag* of the poset.

This paper proves the following theorem.

**Theorem 1.1.**

**WEIGHTED POSET GAME** \((S, \preceq, w, L_1, L_2)\) is PSPACE-complete even if the following conditions are simultaneously hold, where \(D\) is the canonical dag of the poset:

1. \(w(x) \in \{1, -1\}\) for all \(x \in S\), and \(L_1 = L_2 = 0\),
2. \(D\) is bipartite, and
3. the length of each path in \(D\) is at most two.

## 2 Proof

### 2.1 Formula game

In this section we show a proof of Theorem 1.1. As mentioned before weighted poset game is clearly in PSPACE, and to show PSPACE-hardness is enough to prove it. We use the following **FORMULA GAME**, which is known to be PSPACE-complete [3].

This problem based on the following game:

1. Play on a given boolean formula \(\phi\) in CNF. Let \(x_1, x_2, \ldots, x_k\) be the variables used in \(\phi\), where \(k\) is a positive even number.
2. Two players, called \(P_T\) and \(P_F\), take turns selecting the value of the variables \(x_1, x_2, \ldots, x_k\). \(P_T\) (resp., \(P_F\)) selects values with odd (resp., even) quantifiers. The order of play is the same as that of the quantifiers at the beginning of the formula, i.e., first \(P_T\) selects a value of \(x_1\), next \(P_F\) selects a value of \(x_2\), next \(P_T\) selects a value of \(x_3\), \ldots and so on.
3. \( P_T \) wins if \( \phi \) becomes true by using the selected values at the end of play (after fixing all variables), and \( P_F \) wins otherwise.

**Example:**
\[
\phi = (x_1 + x_2)(x_1 + x_3)(x_1 + x_4)
\]

If the game goes
1. \( P_T \) sets \( x_1 = 0 \),
2. \( P_F \) sets \( x_2 = 0 \),
3. \( P_T \) sets \( x_3 = 1 \),
4. \( P_F \) sets \( x_4 = 0 \),

then \( \phi = 0 \), which means \( P_T \) wins the game.

**FORMULA GAME** is a problem to determine which player has a winning strategy on this game for a given \( \phi \).

### 2.2 Reduction

We give a polynomial-time reduction from **FORMULA GAME** to **WEIGHTED POSET GAME**.

For an instance \( \phi \) of **FORMULA GAME**, let \( x_1, \ldots, x_n \) be the variables and \( c_1, \ldots, c_m \) be the clauses of \( \phi \). By reducing \( \phi \) we construct an instance \((D = (V,A), w, 0, 0)\) in the canonical dag representation of **WEIGHTED POSET GAME**.

\[
V = \bigcup_{i=1}^{n} V_i \cup \bigcup_{j=1}^{m} W_j \cup U,
\]

where \( V_i \) \((i = 1, \ldots, n)\) corresponds to \( x_i \), \( W_j \) \((j = 1, \ldots, m)\) corresponds to \( c_j \), and \( U \) is a set of extra vertices. For each \( i \in \{1, \ldots, n\} \), \( V_i \) consists of the following 7 vertices (see Fig. 1 (a)):
\[
V_i = \{ v_{i}^j, v_{j}^f, v_{M,1}^i, v_{M,2}^i, v_{M,3}^i, v_{P,1}^i, v_{P,2}^i \},
\]

where \( w(v_{i}^j) = w(v_{j}^f) = w(v_{P,1}^i) = w(v_{P,2}^i) = 1 \), \( w(v_{M,1}^i) = w(v_{M,2}^i) = w(v_{M,3}^i) = -1 \), and \( \{v_{i}^j, v_{M,1}^i, v_{M,2}^i, v_{M,3}^i, v_{P,1}^i, v_{P,2}^i\} \in A, \forall k \in \{1, 2, 3\}, \forall h \in \{1, 2\} \).
Figure 2: An instance of WEIGHTED POSET GAME reduced from $\phi = (x_1 + x_2 + x_3)(x_1 + x_3 + x_4)(x_2 + x_3 + x_4)$, where a vertex of filled (resp., blank) circle is a poison (resp., medicine) vertex and it's weight is 1 (resp., -1).

For each $j \in \{1, \ldots, m\}$, $W_j$ consists of the following 4 vertices (see Fig. 1 (b)):

$$W_j = \{w_{M,1}^j, w_{M,2}^j, w_{P,1}^j, w_{P,2}^j\},$$

where $w(w_{M,1}^j) = w(w_{M,2}^j) = -1$, $w(w_{P,1}^j) = w(w_{P,2}^j) = 1$, and $(w_{M,k}^j, w_{P,h}^j) \in A$, $\forall k, h \in \{1, 2\}$.

For representing that literal $x_i$ is included in clause $c_j$, add the following arcs: $(v_i^j, w_{M,1}^j)$, $(v_i^j, w_{M,2}^j) \in A$. For representing that literal $\overline{x}_i$ is included in clause $c_j$, add the following arcs: $(v_i^j, w_{M,1}^j)$, $(v_i^j, w_{M,2}^j) \in A$.

To force the order of selecting the value of variables, cascade $V_1, \ldots, V_n$ by the following arcs: $(v_{M,1}^{i+1}, v_{P,1}^i) \in A$, $\forall i \in \{1, \ldots, n-1\}$.

Finally, let $U = \{u_M, u_P\}$, where $w(u_M) = -1$, $w(u_P) = 1$, and $(u_M, u_P), (u_P, w_{P,1}^j) \in A$, $\forall j \in \{1, \ldots, m\}$.

See Fig. 2 for an example of the reduction.
2.3 Correctness of the reduction

Now we show the equivalence of the two instances:

Lemma 2.1. \( P_T \) has a winning strategy in \( \phi \) if and only if player 1 has a winning strategy in \( (D = (V,A), w, 0, 0) \).

Proof: In \( (D = (V,A), w, L_1, L_2) \), the first choice (of player 1) is restricted in \( v_i^t \) or \( v_i^f \), since if the player chooses another vertex, the sum of the weights of the vertex and it’s descendants is greater than zero, and he/she immediately loses (note \( L_1 = L_2 = 0 \)). If \( v_i^t \) or \( v_i^f \) is chosen, the next player (2) can choose \( v_j^t \) or \( v_j^f \), since a poison vertex \( v_{P,1} \), which is a descendant of them, have been deleted. In the same argument, \( v_j^t \) or \( v_j^f \), \( v_j^t \) or \( v_j^f \), . . . are chosen in this order. Vertices in \( W_j \) are deleted if and only if a vertex \( v_i^t \) with \( (v_i^t, w_{M,1}^i), (v_i^t, w_{M,2}^i) \in A \) or a vertex \( v_j^t \) with \( (v_j^t, w_{M,1}^j), (v_j^t, w_{M,2}^j) \in A \) is deleted. Note that \( (v_i^t, w_{M,1}^i), (v_i^t, w_{M,2}^i) \in A \) means \( x_i \in c_j \), i.e., \( c_j \) is satisfied by letting \( x_i = 1 \); and \( (v_j^t, w_{M,1}^j), (v_j^t, w_{M,2}^j) \in A \) means \( \exists \tilde{x} \in c_j \), i.e., \( c_j \) is satisfied by letting \( x_i = 0 \).

After player chooses \( v_i^t \) or \( v_i^f \), if all \( W_1, . . . , W_m \) have been deleted, player 1 can choose \( u_M \) and wins (since the remaining vertices for player 2 are all poison vertices), otherwise player 1 loses (since every selection for him/her has a positive weight).

Hence if \( P_T \) has a winning strategy of \( \phi \), player 1 can simulate the strategy by choosing \( v_i^t \) if \( x_i = 1 \) or \( v_i^f \) if \( x_i = 0 \) and wins. Contrary, if if \( P_T \) has a winning strategy of \( \phi \), player 2 can simulate the strategy by choosing \( v_i^t \) if \( x_i = 1 \) or \( v_i^f \) if \( x_i = 0 \) and wins. \( \square \)

Proof of Theorem 1.1: Obvious from the reduction shown in 2.2 and Lemma 2.1. \( \square \)

3 Conclusion

This paper shows that deciding a winner of weighted poset game is PSPACE-complete even under strong restrictions. The most important open problem is how about (unweighted) poset game? Although it is clearly in PSPACE, we have not known whether it is in P or NP-hard.

Acknowledgements

We would like to give a sincerely gratitude to Professor Gisaku Nakamura for his invaluable comments and helps.

References