#G03 INTEGERS 9 (2009),621-627

ON THE COMPLEXITY OF N-PLAYER HACKENBUSH

Alessandro Cincotti
School of Information Science, Japan Advanced Institute of Science and Technology, Japan
cincotti@jaist.ac.jp

Received: 11/5/08, Accepted: 8/8/09, Published: 11/27/09

Abstract

Why are n-player games much more complex than two-player games? Is it much more difficult to cooperate or to compete? N-player Hackenbush is an n-player version of Blue-Red Hackenbush, a classic two-player combinatorial game played on graphs. Because of queer games, i.e., games where no player has a winning strategy, cooperation is a key-factor in n-player games and, as a consequence, n-player Hackenbush played on strings is $\mathcal{PSPACE}$-complete.

1. Introduction

Combinatorial game theory is a branch of mathematics devoted to studying the optimal strategy in perfect-information games where typically two players are involved. To extend this theory so as to allow more than two players is a challenging and fascinating problem for different reasons.

Typically, more than two parties are involved in real-world economical, social or political conflicts and a winning strategy is often the result of alliances. In two-player games there exist no coalitions because the two players are in conflict with each other, but in n-player games cooperation is a key-factor, because to determine the winning strategy of a coalition of players means to consider the worst scenario, i.e., assuming that all the other players are allied against that coalition.

The first theories of Li [9] and Straffin [13] concerning impartial three-player combinatorial games have made various restrictive assumptions about the rationality of one's opponents and the formation and behavior of coalitions. Loeb [10] introduces the notion of a stable winning coalition in a multi-player game as a new system of classification of games. Differently, Propp, in his work concerning three-player impartial games [12], seeks only to understand in what circumstances one player has a winning strategy against the combined forces of the other two.

Cincotti [3, 4] presents an extension of Conway's theory of partizan games [7, 8] to classify three-player partizan games [5]. Such a theory has been applied to three-player Hackenbush, that is to say a three-player version of Blue-Red Hackenbush. When Blue-Red Hackenbush is played on strings, cooperation is much more difficult than competition and, as a consequence, three-player Hackenbush played on strings is $\mathcal{NP}$-complete [6].

2. N-player Hackenbush

Blue-Red Hackenbush is a classic combinatorial game. Every instance of this game is represented by an undirected graph such that the following hold.

- Every edge is connected via a chain of edges to a certain line called the *ground*.
- Every edge is colored either blue or red.

Two players, called Left and Right, move alternately. Left moves by deleting any blue edge together with all the edges that are no longer connected to the ground and Right moves by deleting any red edge together with all the edges that are no longer connected to the ground. The first player unable to move because there are no edges of his/her color is the loser.

When Blue-Red Hackenbush is played on strings, it is easily solvable using Berlekamp's rule [1], but to determine the value of a Blue-Red Hackenbush position on a general graph is $\mathcal{NP}$-hard [2].

N-player Hackenbush is the multi-player version of Blue-Red Hackenbush. Every instance of n-player Hackenbush is represented by an undirected graph such that the following fold.

- Every edge is connected via a chain of edges to a certain line called the *ground*.
- Every edge is labeled by an integer $j \in \{1, 2, \ldots, n\}$.

The first player moves by deleting any edge labeled 1 together with all the edges that are no longer connected to the ground, the second player moves by deleting any edge labeled 2 together with all the edges that are no longer connected to the ground, and so on.

Players take turns making legal moves in cyclic fashion (1-st, 2-nd, $\ldots$, n-th, 1-st, 2-nd, $\ldots$). When one of the n players is unable to move, then that player leaves the game and the remaining $n-1$ players continue playing in the same mutual order as before. The remaining player is the winner.

We briefly recall the definition of *queer* game introduced by Propp [12]:

Definition 1. A position in a three-player combinatorial game is called *queer* if no player can force a win.

Such a definition is easily generalizable to n players:

Definition 2. A position in an n-player combinatorial game is called *queer* if no player can force a win.

In the game of n-player Hackenbush, it is not always possible to determine the winner because of queer games, as shown in Figure 1. In this case, no player has a winning strategy because if the first player removes the first string, then the third player has a winning strategy, but if the first player removes the second string, then the second player has a winning strategy.

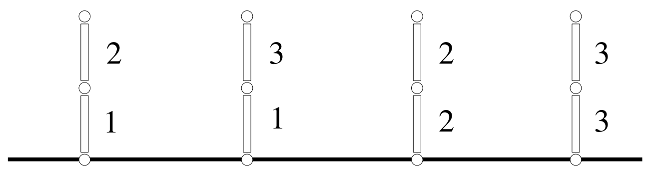

Figure 1: An example of a queer game.

When the game is queer, only cooperation between players can guarantee a winning strategy, i.e., one player of the coalition is always able to make the last move. As a consequence, to establish whether or not a coalition has a winning strategy is a crucial point.

3. N-player Hackenbush played on strings is $\mathcal{PSPACE}$-complete

In this section we show that the $\mathcal{PSPACE}$-complete problem of *Quantified Boolean Formulas* [11], QBF for short, can be reduced by a polynomial time reduction to n-player Hackenbush.

Let $\varphi \equiv \exists x_1 \forall x_2 \exists x_3 \ldots Q x_n \psi$ be an instance of QBF, where Q is $\exists$ for n odd and $\forall$ otherwise, and ψ is a quantifier-free Boolean formula in conjunctive normal form. We recall that QBF asks if there exists an assignment to the variables x_1, x_3, ..., $x_{2\lceil n/2 \rceil -1}$ such that the formula evaluates to true.

If n is the number of variables and k is the number of clauses in ψ, then the instance of n-player Hackenbush will have $n+k+2$ players and $2n+2$ strings, organized as follows:

- For each variable x_i, we add two new strings. Both strings have on the bottom one edge labeled i, with $1 \le i \le n$. On the top of the first string we add one edge for each clause that contains x_i and on the top of the second string we add one edge for each clause that contains $\overline{x_i}$. These edges are labeled j, with $n+1 \le j \le n+k$, and arranged in increasing order from top to bottom.
- One string contains two edges labeled $n+k+1$.
- One string contains $k+2$ edges. The edges on the bottom and on the top are labeled $n+k+2$, and the remaining k edges are labeled from bottom to top $n+k$, $n+k-1$, ..., $n+1$.

Let us suppose that the following hold.

- The first coalition is formed by $\lfloor n/2 \rfloor + 1$ players corresponding to the edges labeled 2, 4, …, $2\lfloor n/2 \rfloor$ and $n + k + 1$.
- The second coalition is formed by the remaining players.

An example is shown in Figure 2 where

$$\varphi \equiv \exists x_1 \forall x_2 \exists x_3 \forall x_4 (C_5 \wedge C_6 \wedge C_7)$$

and

$$\begin{aligned} C_5 &\equiv (x_1 \vee \overline{x_2} \vee x_3), \\ C_6 &\equiv (\overline{x_1} \vee x_2 \vee x_4), \\ C_7 &\equiv (\overline{x_1} \vee x_3 \vee \overline{x_4}). \end{aligned}$$

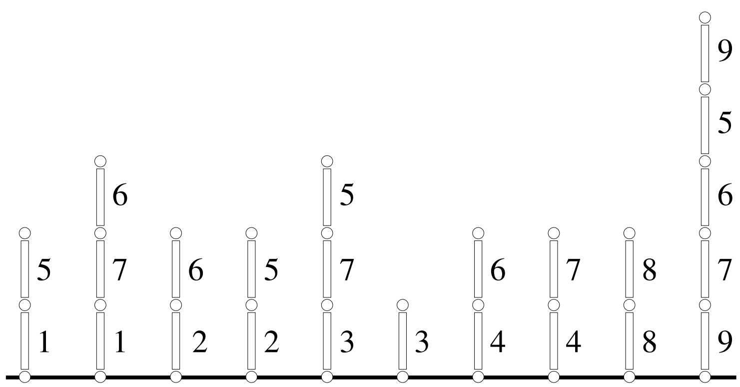

Figure 2: An example of Quantified Boolean Formula reduced to n-player Hackenbush.

The problem to determine the winning coalition is strictly connected to the problem of QBF, as shown in the following theorem.

Theorem 3. *Let G be a general instance of n-player Hackenbush played on strings. Then, to establish whether or not a given coalition has a winning strategy is a $\mathcal{PSPACE}$-complete problem.*

Proof. We show that it is possible to reduce every instance of QBF to a graph G representing an instance of n-player Hackenbush. Previously we have described how to construct the instance of n-player Hackenbush, therefore we just have to prove that QBF is satisfiable if and only if the second coalition has a winning strategy.

If QBF is satisfiable, then there exists an assignment of x_i such that ψ is true with $i \in \{1, 3, \ldots, 2\lceil n/2 \rceil - 1\}$. Each player plays twice. In the first round, if x_i is true, then the i-th player will remove the string with the edges corresponding to the clauses containing $\overline{x_i}$ else, if x_i is false, then the i-th player will remove the string with the edges corresponding to the clauses containing x_i. Every clause contains at least a true literal, therefore the i-th player with $i \in \{n+1, n+2, \ldots, n+k\}$ can always remove one edge from the string corresponding to that literal. In this way, the $(n+k+2)^{\text{th}}$ player can remove the edge on the top of the string containing $k+2$ edges. In the second round, players remove the remaining edges. Each player can remove only one possible edge when he/she has to play therefore, at the end of the game, the $(n+k+2)^{\text{th}}$ player will make the last move. As a result, the second coalition has a winning strategy.

Conversely, let us suppose that the second coalition has a winning strategy. We observe that the $(n+k+1)^{\text{th}}$ player is always able to make two moves, therefore even the $(n+k+2)^{\text{th}}$ player must be able to make two moves in order to assure a winning strategy for the second coalition. As a consequence, the i^{th} player with $i \in \{n+1, n+2, \ldots, n+k\}$ does not remove any edges in the string containing $k+2$ edges before the $(n+k+2)^{\text{th}}$ player makes his/her first move, i.e., every clause has at least one true literal and QBF is satisfiable.

Therefore, to establish whether or not a coalition has a winning strategy in n-player Hackenbush played on strings is $\mathcal{PSPACE}$-hard.

To show that the problem is in $\mathcal{PSPACE}$ we present a polynomial-space recursive algorithm to determine which coalition has a winning strategy. Let us introduce some useful notation:

- $G = (V, E)$ is the graph representing an instance of n-player Hackenbush played on strings;
- p_i is the i^{th} player;
- C_0 is the set of current players belonging to the first coalition;
- C_1 is the set of current players belonging to the second coalition;
- `coalition`(p_i) returns 0 if $p_i \in C_0$ and 1 if $p_i \in C_1$;
- `label`(e) returns the label of the edge e;
- `after`(p_i) returns the player which has to play after p_i;
- `remove`$(G$,$e)$ returns the graph obtained after that the edge e and all the edges no longer connected to the ground have been removed from G.

Algorithm 1 performs an exhaustive search until a winning strategy is found and its correctness can be easily proved by induction on the depth of the game tree.

Algorithm 1 is clearly in $\mathcal{PSPACE}$ because the number of nested recursive calls is at most $|E|$ and therefore the total space complexity is $O(|E|^2)$. □

input: A graph $G = (V, E)$, the two initial coalitions C_0 and C_1 and the player p_i that has to move;
output: 0 if the first coalition has a winning strategy and 1 if the second coalition has a winning strategy;

```
Algorithm Check(G, C_0, C_1, p_i)
j ← coalition(p_i);
if ∄e ∈ E : label(e) = i then                    // remove player p_i from game
    C_j ← C_j \ {p_i};
    if C_j = ∅ then                                  // C_j has no more players
        return 1 − j;
    else
        return Check(G,C_0,C_1,after(p_i));
    end
else
    forall e ∈ E : label(e) = i do    // check all the possible moves of p_i
        G' ← remove(G,e);
        if Check(G',C_0,C_1,after(p_i)) = j then        // C_j has a winning strategy
            return j;
        end
    end
    return 1 − j;
end
```

Algorithm 1: A polynomial-space algorithm for n-player Hackenbush.

Table 1 summarizes the results so far obtained about the relation between number of players and complexity in the domain of Hackenbush played on strings.

Table 1: Complexity of Hackenbush played on strings.

No. of players	Complexity
2	$O(n)$
3	$\mathcal{NP}$-complete
n	$\mathcal{PSPACE}$-complete

Acknowledgments. The author wishes to thank the anonymous referee for helpful suggestions which improved the presentation of the paper. Also, I would like to thank Mary Ann Mooradian for a careful reading of the manuscript.

References

[1] E. R. Berlekamp, The Hackenbush number system for compression of numerical data, *Information and Control*, **26** (1974) 134-140.

[2] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning way for your mathematical plays*, 2nd ed., A K Peters, 2001.

[3] A. Cincotti, Three-player partizan games, *Theoretical Computer Science* **332** (2005) 367-389.

[4] A. Cincotti, Counting the Number of Three-Player Partizan Cold Games, in *Computers and Games 2006*, pp. 181-189, edited by H. J. van den Herik, P. Ciancarini, and H. H. L. M. Donkers, LNCS Vol. 4630, Turin, Springer-Verlag, 2007.

[5] A. Cincotti, The Game of Cutblock, *INTEGERS: Electronic Journal of Combinatorial Number Theory*, **8** (2008) #G06.

[6] A. Cincotti, Three-player Hackenbush played on strings is $\mathcal{NP}$-complete, in *Proceedings of the International MultiConference of Engineers and Computer Scientists 2008*, pp. 226-230, edited by S. I. Ao, O. Castillo, C. Douglas, and D. D. Feng, J. Lee, Newswood Limited, Hong Kong, 2008.

[7] J. H. Conway, *On numbers and games*, 2nd ed., A K Peters, 2001.

[8] D. Knuth, *Surreal Numbers*, Addison-Wesley, 1974.

[9] S. Y. R. Li, N-person Nim and N-person Moore's games, *Internat. J. Game Theory* **7** (1978) 31-36.

[10] D. E. Loeb, Stable Winning Coalitions, in: R. J. Nowakowski (Ed.), *Games of No Chance*, MSRI Vol. 29, Cambridge University Press, 1996, pp. 451-471.

[11] C. H. Papadimitriou, *Computational Complexity*. Addison-Wesley, 1994.

[12] J. G. Propp, Three-player impartial games, *Theoretical Computer Science* **233** (2000) 263-278.

[13] P. D. Straffin Jr., Three-person winner-take-all games with Mc-Carthy's revenge rule, *College J. Math.* **16** (1985) 386-394.