Analysis of a Game Played On a Graph Based on Experiments With Computers

Shuji Jimbo
Guest Researcher, Graduate School of Environmental, Life, Natural Science and Technology, Okayama University

E-mail address: `sjimbo@m.ieice.org`

Abstract

This research aims to determine win/loss outcomes from specific starting positions in Yasutomi's game, played on a graph called the pyramid of height 13. Due to the game's vast search space, a basic minimax approach is infeasible. The researchers initially hoped to reduce complexity by detecting special winning forms but abandoned this after finding them rare. They now focus on minimizing redundant searches by identifying isomorphic graphs and simplifying the search graph using cut-vertex decomposition. The terminology and game structure are based on prior studies by Kobayashi.

KEYWORDS. computer-based proof, minimax search, depth-first search, game played on a graph.

1 Introduction

The goal of this research is to determine the win/loss outcome for a specific starting position in a game played on a graph. We report on our current approach to achieving this goal. The game is referred to as Yasutomi's game, and the graph used is called the pyramid of height 13, as described in [2]. The game was originally given a Japanese name meaning "Challenge game to Euler" [5]. In this report, we adopt the terminology from [2].

It seems clear that a simple minimax search for Yasutomi's game on the pyramid of height 13 cannot be completed in a typical computing environment due to the enormous search space.

We initially expected to frequently encounter special winning forms that could be easily identified during the search. We considered a strategy to significantly reduce the search space by applying a winning form determination to the graph being searched. However, this strategy was abandoned after a simple experiment revealed that special winning forms rarely appeared during a random game.

Currently, we are exploring the possibility of completing the desired search by eliminating re-searching of isomorphic graphs and by reducing the search space through simplifying the search graph based on graph decomposition by cut-vertices.

Section 2 will describe the games considered in this research. In Section 3, we will discuss our inability to predict the frequency of positions that can be easily identified as a guaranteed win for the second player. Section 4 will present the concept of a computer experiment designed to optimistically estimate the size of the search space in a minimax search, thereby demonstrating that the game favors the first player. Section 5 will discuss the concluding remarks.

2 Preliminaries

In *Yasutomi's game*, a planar graph P_n, referred to as a *pyramid*, serves as the game board. Here, n is a positive integer representing the height of the pyramid. Pyramids P_5 and P_{13} are depicted in Fig. 1.

To begin the game, a piece is placed at the top vertex of the pyramid that is represented by the black filled circle in Fig. 1. Two players then take turns moving the piece along the graph's edges, removing each edge as it is traversed.

The vertex where the piece currently resides is called the *current vertex*. The current vertex in the initial position is termed the *starting vertex*. A position in Yasutomi's game is represented by a pair (H, w), consisting of the current subgraph of the pyramid H and the current vertex w.

The player who cannot make a move loses the game. Yasutomi's game is a two-person, zero-sum, finite, deterministic game with perfect information. Since the number of edges on the graph decreases by one with each turn, it is clear that draws are not possible in Yasutomi's game.

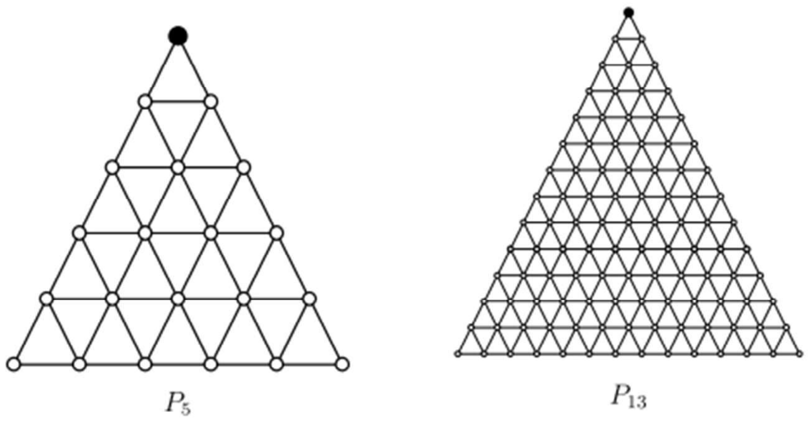

Figure 1: Pyramids P_5 and P_{13}

3 Frequency of appearance of forts

It is known that Yasutomi's game is a guaranteed win for the second player if the graph has a bipartite subgraph satisfying certain conditions [5, 2]. This subgraph is referred to as a *fort* in [2]. The definition of a fort is as follows:

A fort of a game position (G, v) is a bipartite subgraph of G defined by a subset S of $V(G)$, where S satisfies the following conditions:

- The current vertex v belongs to S ($v \in S$).
- Any two vertices x, y in S are not adjacent ($\forall x, y \in S$, $xy \notin E(G)$).
- Any vertex not in S is adjacent to an even number of vertices in S ($\forall x \notin S$, $|\{s \in S \mid xs \in E(G)\}| \in 2\mathbb{Z}$).

Let $A(S)$ denote the set of all vertices adjacent to a vertex in S. Let $G[X]$ denote the vertex-induced subgraph of G by the set X of vertices. Then, the bipartite subgraph $G[S \cup A(S)] - E(G[A(S)])$ is a fort. Note that, in [2], the set of vertices S itself is defined as a fort. In this report, consistent with the definition in [5], a fort refers to a bipartite subgraph rather than a set of vertices.

The following proposition is well-established [5, 2]. More specifically, an explicit construction of the fort is provided in [5] for any pyramid P_n that meets the proposition's condition. Consequently, in most cases, Yasutomi's game played on pyramid P_n with its top vertex as the starting vertex results in a guaranteed win for the second player.

Proposition 1 *A fort exists in the initial position (P_n, v_0) of Yasutomi's game, where v_0 is the top vertex of P_n, if $n + 3 \geq 4$ is not a power of 2.*

Determining whether a position in Yasutomi's game has a fort is relatively straightforward using a SAT solver or an integer programming solver, such as SCIP [1]. Initially, we anticipated that positions with a fort would frequently appear during the search, leading to a significant reduction in the search space. However, a simple experiment proved this assumption incorrect.

For instance, when the game progressed along the trails shown in Fig. 2, no positions containing a fort were found.

Our small-scale experiments indicated that positions with a fort only arose when the game progressed along a corridor-like trail, as illustrated in Fig. 3.

4 Measures to reduce the search space in minimax search

We conjecture that Yasutomi's game played on a pyramid P_{13}, with its top vertex as the initial position, is a guaranteed win for the first player. In this section, we propose an experimental method to assess the feasibility of proving this conjecture via a minimax search.

The following techniques are employed to reduce the search space in the minimax search for this proof:

1. During the minimax search, the first player's move is always unique, and chosen randomly using a uniform random number generator, say /dev/urandom in Unix-like operating systems. Thus, our experimental minimax search effectively acts as a depth-first search to find a guaranteed win for the first player, branching only at the second player's nodes.

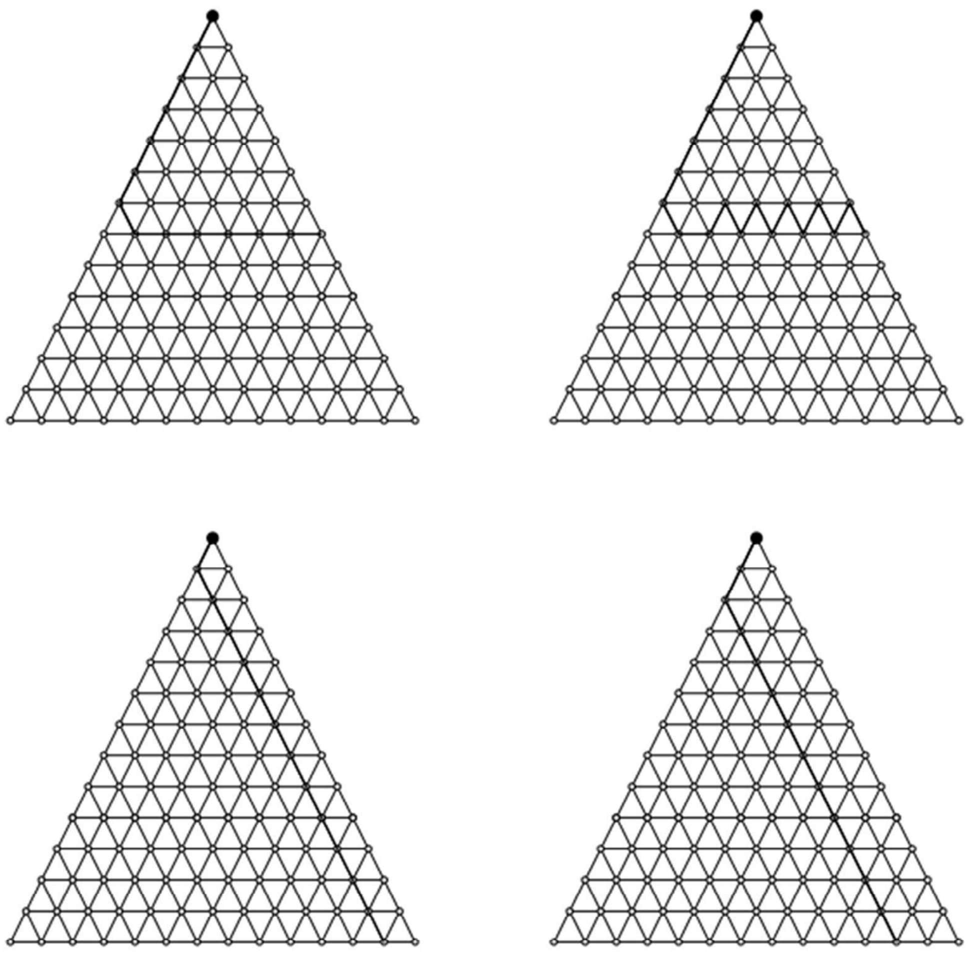

Figure 2: Game progressions in which no forts arise

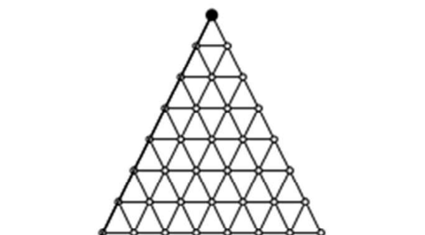

Figure 3: A game progression in which forts arise

2. To prevent re-searching isomorphic graphs, newly generated graphs are *canonically labeled* and stored in a hash table. The hash table is based on Zobrist hashing [6]. We are considering using the *nauty and Traces library* [3] for canonical labeling.

3. When a new graph arises, a depth-first search is performed, restricted to the region reachable from the current vertex, to identify cut-vertices. Cut-vertices can be easily found in a single depth-first search [4]. If the graph contains a cut-vertex, any connected components formed by its removal that do not include the current vertex are deleted. While this process is somewhat heuristic, we consider it suitable for an optimistic estimation of the search space.

4. Only hash values and win/loss states are recorded in the hash table, not the entire graph. This approach minimizes the main memory occupied per graph and enables a greater number of entries compared to storing complete graphs. Although this method carries the risk of hash value collisions and potential search errors, we believe the search's reliability can be arbitrarily increased. This is achieved by repeating the same search multiple times. In each trial, we compute a "search hash value" for the entire search operation and use Zobrist hashing for game positions based on different random numbers for each trial. If the same search hash value is obtained across all trials, the estimated probability of a hash value collision occurring decreases as the number of trials increases.

Let G be the graph representing the initial game position. If H is a connected component formed by deleting a cut-vertex w from G, $H + w$ denotes the subgraph of G induced by the vertices in H and w. The win/loss states for the starting position $(H + w, w)$ to be stored in the hash table are categorized as follows:

- The first player wins, and the vertex where the second player is located on the last turn is not w.
- The first player wins, and the vertex where the second player is located on the last turn is w.
- The second player wins, and the vertex where the first player is located on the last turn is not w.
- The second player wins, and the vertex where the first player is located on the last turn is w.

Additionally, each hash table element must be capable of storing the following two types of information:

- Nothing is registered in the element of that hash table.
- The element has a position registered in the hash table, but its win/loss state has not yet been determined.

Consequently, each hash table element is represented as a 64-bit unsigned integer. Of these, 61 bits are assigned to the hash value, and the remaining 3 bits are used to represent auxiliary information, such as the win/loss state.

5 Concluding remarks

Our immediate next step to the goal is to conduct sufficient experiments incorporating the techniques described in the previous section. If these experiments show the estimated search space is sufficiently small, the next phase will involve performing an actual minimax search.

A challenging aspect is determining how to prioritize moves during the first player's turn. One approach is to select moves that minimize the number of possible moves for the second player in the subsequent turn.

We plan to report the results once these proposed experiments are fully implemented.

Acknowledgment

This work was supported by the *Research Institute for Mathematical Sciences*, an International Joint Usage/Research Center at Kyoto University.

This research utilizes *SCIP*, free software from The Zuse Institute Berlin, licensed under the Apache 2.0. Specifically, we used the software's integer programming capabilities to detect forts.

We extend our deepest gratitude to Professor Yuji Kobayashi of the Mathematics and Games Laboratory for providing us with a document proving that Yasutomi's game, with a pyramid of height 5 as an initial position, is a guaranteed win for the first player without the aid of a computer.

References

[1] Suresh Bolusani, Mathieu Besançon, Ksenia Bestuzheva, Antonia Chmiela, João Dionísio, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Mohammed Ghannam, Ambros Gleixner, Christoph Graczyk, Katrin Halbig, Ivo Hedtke, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Dominik Kamp, Thorsten Koch, Kevin Kofler, Jurgen Lentz, Julian Manns, Gioni Mexi, Erik Mühmer, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Mark Turner, Stefan Vigerske, Dieter Weninger, and Lixing Xu. The SCIP Optimization Suite 9.0. Technical report, Optimization Online, February 2024.

[2] Yuji Kobayashi. Games played on a graph. personal communication, September 2020.

[3] Brendan D. McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94–112, 2014.

[4] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[5] Shin-ichi Yasutomi and Izumi Nakashima. The game on euler graph. *Memoirs of Suzuka College of Technology*, 25(1):57–60, 1992.

[6] Albert L Zobrist. A new hashing method with application for game playing. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 1970.