(a) An example of a Tilepaint puzzle. (b) A solution of Figure 1a.

Figure 1. An example of Tilepaint Puzzle and its solution.

1. Introduction

Tilepaint (タイルペイント) is a logic pencil-and-paper-based puzzle invented by Toshihari Yamamoto in Japan and popularized by Nikoli, a publisher that specializes in logic puzzles.[1] According to Jimmy Goto—then a Nikoli manager—Tilepaint first appeared in issue 53 of Nikoli's quarterly Puzzle Communication magazine in 1995 and has been published regularly ever since [1]. This puzzle considers an $m \times n$ grid of cells where the cells are divided into some *tiles*. A tile (sometimes referred to as a *region*) is a collection of orthogonally connected cells separated by thick lines. Initially, all cells are left blank (uncolored). Each cell in every tile must be either colored or uncolored (i.e., we must color all cells in a tile or leave all of them uncolored). There are constraints indicated by numbers at the top and left of the grid, specifying the number of cells that must be colored in the corresponding row and column. The problem is to find any configuration that matches the number of colored cells according to the constraint described for each row and column (if any). An example of a Tilepaint puzzle instance and its respective solution are depicted in Figure 1.

Solving puzzles is valuable for enhancing mathematics, computational thinking, and problem-solving skills. Regularly solving puzzles stimulates various cognitive functions, including critical thinking, mathematical thinking, and creativity, which are fundamental to mathematical proficiency [2]. Some puzzles have notable connections to important computational and mathematical problems, sparking the scientific community's interest in exploring these puzzles [3–5]. Several one-player puzzles have been confirmed NP-complete, such as (in alphabetical order, the year in which the puzzle is confirmed NP-complete indicated inside the brackets): Five Cells (2022) [6], Juosan (2021) [7], Kurotto (2021) [7], Minesweeper (2000) [8], Moon-or-Sun (2022) [9], Nagareru (2022) [9], Nonogram (1996) [10], Nurimeizu (2022) [9], Path Puzzles (2020) [11], Sudoku (2003) [12], Suguru (2022) [13], Tatamibari (2020) [14], and Yin-Yang (2021) [15].

Tilepaint puzzles were recently confirmed NP-complete in 2022 by Iwamoto and Ide [6]. The NP-completeness of the puzzle means a polynomial time algorithm exists to verify whether a configuration satisfies the puzzle's rules. Moreover, this implies solving a general instance of Tilepaint puzzles currently requires an exponential time algorithm. Nevertheless, to the author's knowledge, there has never been any discussion of further research into the algorithms used to solve these puzzles. Various methods are proposed to solve NP-complete puzzles, including non-elementary techniques such as integer programming model [16] and SAT solver [17–19]. This paper discusses two elementary search-based techniques for solving Tilepaint puzzles: a complete search approach with a bitmasking technique and a prune-and-search approach with a backtracking method. Previous studies show that elementary search-based methods can be applied for solving NP-complete puzzles, such as the prune-and-search method for solving Yin-Yang puzzles [20] and exhaustive search technique for solving Tatamibari puzzles [21]. Moreover, this final project also discusses a SAT-based approach to solving a Tilepaint puzzle. This process includes transforming the rules of Tilepaint puzzles into Boolean satisfiability problems and using a SAT solver to find a solution to the puzzle. Additionally, this paper conducts experiments on all approaches, namely complete search, backtracking, and SAT solver, which are then compared and analyzed to determine their effectiveness and efficiency in solving Tilepaint puzzles. Some special tractable and intractable cases of the Tilepaint puzzle are also discussed in this paper. It is important to investigate tractable sub-problems of NP-completes problems [22].

Tilepaint puzzles are closely related to two-dimensional discrete tomography problems. In fact, Tilepaint puzzles can be extended from these problems by imposing the additional *tile rule*, namely, all cells within the same tile must have the same color. Other NP-complete puzzles related to two-dimensional discrete tomography problems are Nonogram [10] and Path Puzzles [11]. Nevertheless, it is well-known that the two-dimensional discrete tomography problem is solvable in polynomial time if the numerical constraints for all rows and columns in

[1] An example of a famous puzzle published by Nikoli is Sudoku.

the instance are known [23,24]. Modifying existing rules and introducing additional rules to the formerly tractable two-dimensional discrete tomography problem transforms this problem into interesting non-trivial NP-complete puzzles. In the case of the Nonogram puzzle, the NP-hardness arises due to multiple numerical constraints for each row and column, while for the Path Puzzles, the NP-hardness occurs due to the Hamiltonicity constraint.

This paper also discusses some tractable and intractable variants of the original Tilepaint puzzles. The study of originally NP-complete problems' variations that are tractable is important in theoretical computer science [22,25]. This paper shows that an $m \times n$ Tilepaint instance containing mn tiles of size 1×1 is solvable in polynomial time if all the numerical constraints are known. In contrast, this paper also shows that reducing the dimension of a Tilepaint puzzle to $m \times 1$ or $1 \times n$ does not necessarily make the puzzle becomes tractable.

The rest of the investigation is organized into the following sections. Section 2 covers the formal definition, data structure representation, and mathematical properties of Tilepaint puzzles. This final project explores some important mathematical observations of the Tilepaint puzzle; specifically discusses a condition where a Tilepaint puzzle does not have a solution. This condition can be used as a prune condition to optimize the Tilepaint solver. Section 3 discusses a polynomial time approach to verify whether a Tilepaint configuration satisfies the puzzle's rules. This paper shows that this verification algorithm takes $O(mn)$ time for an $m \times n$ Tilepaint configuration with any number of tiles. Section 4 discusses a complete search approach with a bitmasking technique to solve arbitrary Tilepaint puzzles of size $m \times n$ containing p tiles in $O(2^p \cdot p \cdot mn)$. The prune-and-search approach with a backtracking technique and pruning optimization for solving arbitrary Tilepaint puzzles of size $m \times n$ with p tiles in $O(2^p \cdot mn)$ is discussed in Section 5. Section 6 presents a declarative SAT-based approach for solving Tilepaint puzzles and its corresponding analysis. This final project discusses a tractable variant of Tilepaint puzzles, namely an $m \times n$ puzzle with mn tiles of size 1×1 in Section 7. Moreover, this paper also shows that a general Tilepaint puzzle of size $m \times 1$ or $1 \times n$ remains intractable in Section 8. Section 9 presents the experimental results showcasing the performance of both solver algorithms in solving Tilepaint puzzles of various sizes. Finally, the investigation is summarized and concluded in Section 10.