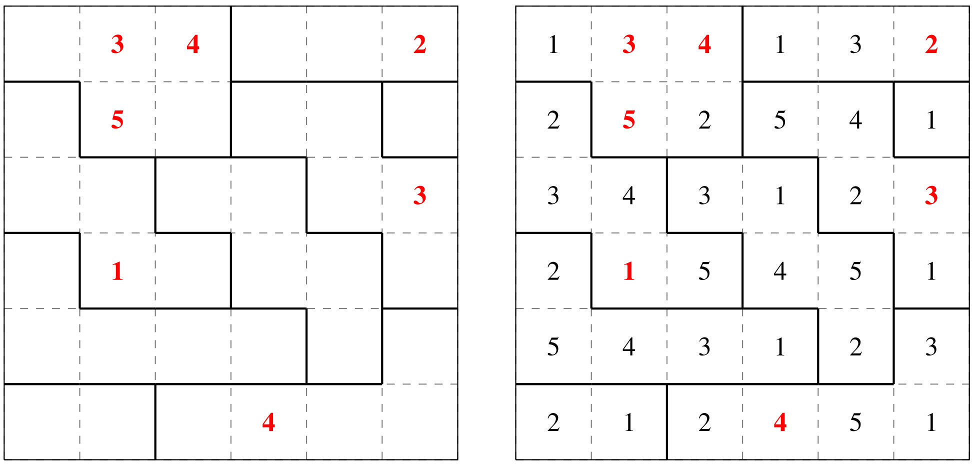
1 Introduction

Suguru (also known as Nanbaburokku) is a single-player pencil-and-paper puzzle invented by Naoki Inaba, a prominent Japanese puzzle designer. It first appeared in 2001 [1] and was recently proven NP-complete by Robert et al. in 2022 [2]. Like the famous Sudoku, the player must fill the empty cells in a rectangular grid, satisfying some puzzle rules. This puzzle is played on an $m \times n$ grid partitioned into regions. A region is a collection of orthogonally connected cells. The goal is to fill all cells with numbers such that:

1. a number in a cell must be between 1 and the size of the region it belongs to, where the size of a region is defined as the number of cells in it;
2. no two cells in a region can contain the same number;
3. no two adjacent cells, either orthogonally or diagonally, can contain the same number.

For a visual representation of a Suguru puzzle along with its solution, see Fig. 1.



(a) Suguru puzzle of size 6×6. (b) A solution to the Suguru puzzle in Fig. 1a.

Figure 1. An example of a Suguru puzzle (Fig. 1a) and its solution (Fig. 1b). The bold red-colored numbers in the cells indicate that those cells are initially filled cells.

Puzzles have long been regarded as captivating mental challenges that have entertained and engaged individuals throughout history. They provide leisure and diversion opportunities and stimulate cognitive skills such as critical thinking and problem-solving [3]. Moreover, theoretical aspects of puzzles have garnered substantial interest from the scientific community in the last twenty years owing to their profound links with crucial problems in mathematics and the theory of computation, resulting in extensive investigations into their mathematical and computational aspects (see [4–6] for extensive investigations). Furthermore, a variety of pencil-and-paper-based puzzles have been confirmed NP-complete, including but not limited to (in chronological order): Nonogram (1996) [7], Sudoku (2003) [8], Nurikabe (2004) [9], Heyawake (2007) [10], Hashiwokakero (2009) [11], Kurodoko (2012) [12], Shikaku and Ripple Effect (2013) [13], Yosenabe (2014) [14], Fillmat (2015) [15], Dosun-Fuwari (2018) [16], Tatamibari (2020) [17], Kurotto and Juosan (2020) [18], Yin-Yang (2021) [19], Tilepaint (2022) [20], and Suguru (2022) [2].

The NP-completeness of Suguru puzzles implies that there is a polynomial-time verification procedure for checking whether an arbitrary configuration is a solution to a Suguru instance. However, solving a Suguru puzzle remains an exponential-time task because no known polynomial-time algorithm exists for any NP-complete problem. Moreover, formal algorithmic investigation for solving Suguru puzzles has been relatively limited as it has only recently proven NP-complete. Investigations on elementary algorithmic methods such as the exhaustive search and prune-and-search—which utilizes a similar approach to the methods used in this paper—have been carried out on puzzles such as Yin-Yang [21] and Tatamibari [22]. More advanced techniques are also available for solving NP-complete puzzles, such as SAT solvers [23, 24] and the deep learning method [25].

This paper discusses an elementary approach, the backtracking method, enhanced with pruning optimizations. This approach demonstrates its ability to solve any Suguru puzzle, with the caveat that the solving time increases in factorial factor in terms of the puzzle size and the number of hints. Furthermore, this final project explores an alternative approach for solving Suguru puzzles using the SAT-based approach. In addition to this, this paper delves into the exploration of a tractable variant of the Suguru puzzle. Investigating such variants of NP-complete problems holds significant importance in the field of computational complexity theory [26].

The remainder of this paper is structured as follows. Section 2 introduces some definitions and notations regarding Suguru puzzles' data structure and mathematical representation, as well as some relevant theoretical results. In Section 3, an algorithm is presented that verifies a solution to a Suguru puzzle of size $m \times n$ in $O(mn)$ time. Section 4 discusses the proposed backtracking algorithm—which incorporates pruning optimizations—for solving arbitrary $m \times n$ Suguru puzzles. Specifically, it is proven that the backtracking algorithm can solve an arbitrary $m \times n$ Suguru instance with R regions and H hint cells in $O(R \cdot (mn - H + 2)!)$ time. In Section 5, this paper examines SAT encodings for Suguru puzzles. Section 6 investigates a tractable variant of the puzzle. It is shown that any $m \times n$ Suguru instances where $m = 1$ or $n = 1$ are solvable in linear time. Nevertheless, this paper argues that finding all solutions to a tractable Suguru puzzle may require a non-polynomial number of computational steps. The experimental results to evaluate the proposed algorithm's practical performance are discussed in Section 7. Lastly, the important results of this paper are summarized and concluded in Section 8.