

# An Integer Programming Model for the Sudoku Problem

Andrew C. Bartlett<sup>(1)</sup>, Timothy P. Chartier<sup>(2)</sup>,  
Amy N. Langville<sup>(1)</sup> and Timothy D. Rankin<sup>(2)</sup>

<sup>(1)</sup>Department of Mathematics, College of Charleston, Charleston, SC, USA.

<sup>(2)</sup>Department of Mathematics, Davidson College, Davidson, NC, USA.

## 1 Introduction

Sudoku is a logic-based puzzle that first appeared in the U.S. under the title "Number Place" in 1979 in the magazine *Dell Pencil Puzzles & Word Games* (Garns, 1979). The game was designed by Howard Garns, an architect who, upon retirement, turned to puzzle creation. In the 1980s, the game grew in popularity in Japan and was renamed by publisher Nikoli to "suji wa dokushin ni kagiru," which translates as the "the digits must remain single." This was eventually shortened to "sudoku" or "single number."

By 1997 an entrepreneur named Wayne Gould saw the financial potential available in the game. Gould spent six years refining his computer program so that it could quickly generate puzzles of varying levels of difficulty. In November 2004, Gould convinced *The Times* of London to print a puzzle. From there the popularity of the puzzle spread until now they commonly appear in a wide variety of newspapers and magazines. Interestingly, Gould does not charge newspapers for his puzzles, but they must include the Web address <http://www.sudoku.com> where his Sudoku program can be downloaded (for a free trial version and a fee for permanent use).

Sudoku most commonly appears in its  $9 \times 9$  matrix form. The rules are simple: fill in the matrix so that every row, column, and  $3 \times 3$  submatrix contains the digits 1 through 9 exactly once. Each puzzle appears with a certain number of *givens*. The number and location of these initial entries determine the game's level of difficulty. Figure 1 below is an example of a  $9 \times 9$  Sudoku puzzle. If you wish to solve the Sudoku puzzle, you may enter elements into the tableau. Initially, elements that you enter into the tableau are considered tentative values and appear in a colored box in the tableau. When an element of the tableau is selected and a carriage return entered, the contents of that element are entered into the tableau and no longer considered tentative. As such, any entries contained in colored elements are not considered part of the tableau. By clicking the "violate constraints?" button, the Java application will indicate if your entries (ignoring tentative values) have violated any of Sudoku rules.

**Figure 1.** An example Sudoku puzzle

This puzzle idea can accommodate games of other sizes. Of course, a  $4 \times 4$  puzzle would be easier and a  $16 \times 16$  puzzle harder. In general, any  $n \times n$  game can be created, where  $n = m^2$  and  $m$  is any positive integer. There are numerous other variants of the game; see (Riley, 2007), Brainfreeze Puzzles, and (Pegg, 2005).

Sudoku puzzles elicit the following two interesting mathematical questions:

1. How can these puzzles be solved mathematically?
2. What mathematical techniques can be used to create these puzzles?

In the following sections, we explore these questions. We present a binary integer linear program to solve this feasibility problem. Further, such an approach is extended to variations on the traditional Sudoku puzzle.

In addition, we speculate as to how Sudoku puzzles are created, and provide several theorems for generating many new puzzles from one given original puzzle. Exercises and challenge problems that use principles from optimization, combinatorics, linear algebra, and computer science are presented for students. Answers to the exercises are contained at the conclusion of the article.

---

## Table Of Contents

1. Introduction
  2. Question 1: Solving the Puzzle
  3. Solving Variations on Sudoku Puzzles
  4. Question 2: Creating the Puzzles
  5. Conclusion
  6. References
  7. Answers
- 

Next page > > Question 1: Solving the Puzzle

Pages: | **1** | 2 | 3 | 4 | 5 | 6 | 7 |