An Integer Programming Model for the Sudoku Problem

Ch. Bhaskar Reddy, K. Krishna Reddy, D. Upendra

Abstract

Sudoku is the recent craze in logic puzzles. Players must fill in an \( n \times n \) matrix, which contains some given entries, so that each row, column, and \( m \times m \) submatrix contains each integer 1 through \( n \) exactly once. Two issues associated with these puzzles interest us mathematically: puzzle solution and puzzle creation. A Sudoku puzzle can be solved by creating a feasibility problem where the goal is to find at least one feasible solution to the puzzle. 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 challenges problems that use principles from optimization, combinatorics, linear algebra, and computer science are presented for students.

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 [6]. 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 determine the game’s level of difficulty. Figure 1 is an example of a \( 9 \times 9 \) Sudoku puzzle.

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. 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.
2.1 The Mathematical Model

A mathematical model is a description of a system using mathematical concepts and language. The process of developing a mathematical model is termed mathematical modeling. Mathematical models are used in the natural sciences (such as physics, biology, earth science, meteorology, and engineering disciplines (such as computer science, artificial intelligence), as well as in the social sciences (such as economics, psychology, sociology, political science). Physicists, engineers, statisticians, operations research analysts, and economists use mathematical models most extensively. A model may help explain a system and to study the effects of different components, and to make predictions about behaviour.

2.2 Elements of a mathematical model

Mathematical models can take many forms including dynamical systems, statistical models, differential equations, or game theoretical models. These and other types of models can overlap with a given model involving a variety of abstract structures. In general, mathematical models may include logical models. In many cases, the quality of a scientific field depends on how well the mathematical models developed on the theoretical side agree with results of repeatable experiments. Lack of agreement between theoretical mathematical models and experimental measurements often leads to important advances as better theories are developed.

- **Linear vs. nonlinear**: If all the operators in a mathematical model exhibit linearity, the resulting mathematical model is defined as linear. A model is considered to be nonlinear otherwise. The definition of linearity and nonlinearity is dependent on context and linear models may have nonlinear expressions in them. For example, in a statistical linear model, it is assumed that a relationship is linear in the parameters, but it may be nonlinear in the predicted variables. Similarly, a differential equation is said to be linear if it can be written with linear differential operators, but it can still have nonlinear expression in it. In a mathematical programming model, if the objective functions and constraints are represented entirely by linear equations, then the model is regarded as a linear model. If one or more of the objective functions or constraints are represented with a nonlinear equation, then the model is known as a nonlinear model. Nonlinearity, even in fairly simple systems, is often associated with phenomena such as chaos and irreversibility. Although there are exceptions, nonlinear systems and models tend to be more difficult to study than linear ones. A common approach to nonlinear problems is linearization, but this can be problematic if one is trying to study aspects such as irreversibility, which are strongly tied to nonlinearity.

- **Static vs. dynamic**: A dynamic model accounts for time-dependent changes in the state of the system, while a static (or steady-state) model calculates the system in equilibrium, and thus is time-invariant. Dynamic models typically are represented by differential equations.

- **Explicit vs. implicit**: If all of the input parameters of the overall model are known, and the output parameters can be calculated by a finite series of computations (known as linear programming, not to be confused with linearity as described above), the model is said to be explicit. But sometimes it is the output parameters which are known, and the corresponding inputs must be solved for by an iterative procedure, such as Newton’s method (if the model is linear) or Broyden’s method (if non-linear). For example, a jet engine's physical properties such as turbine and nozzle throat areas can be explicitly calculated given a design thermodynamic cycle (air and fuel flow rates, pressures, and temperatures) at a specific flight condition and power setting, but the engine’s operating cycles at other flight conditions and power settings cannot be explicitly calculated from the constant physical properties.

- **Discrete vs. continuous**: A discrete model treats objects as discrete, such as the particles in a molecular model or the states in a statistical model; while a continuous model represents the objects in a continuous manner, such as the velocity field of fluid in pipe flows, temperatures and stresses in a solid, and electric field that applies continuously over the entire model due to a point charge.

- **Deterministic vs. probabilistic**: A deterministic model is one in which every set of variable states is uniquely determined by parameters in the model and by sets of previous states of these variables; therefore, a deterministic model always performs the same way for a given set of initial conditions. Conversely, in a stochastic model—usually called a “statistical model”—randomness is present, and variable states are not described by unique values, but rather by probability distributions.

- **Deductive, inductive, or floating**: A deductive model is a logical structure based on a theory. An inductive model arises from empirical findings and generalization from them. The floating model rests on neither theory nor observation, but is merely the invocation of expected structure. Application of
mathematics in social sciences outside of economics has been criticized for unfounded models.\textsuperscript{[1]} Application of catastrophe theory in science has been characterized as a floating model

Significance in the natural sciences

Mathematical models are of great importance in the natural sciences, particularly in physics. Physical theories are almost invariably expressed using mathematical models.

Throughout history, more and more accurate mathematical models have been developed. Newton's laws accurately describe many everyday phenomena, but at certain limits relativity theory and quantum mechanics must be used; even these do not apply to all situations and need further refinement. It is possible to obtain the less accurate models in appropriate limits, for example relativistic mechanics reduces to Newtonian mechanics at speeds much less than the speed of light. Quantum mechanics reduces to classical physics when the quantum numbers are high. For example, the de Broglie wavelength of a tennis ball is insignificantly small, so classical physics is a good approximation to use in this case.

It is common to use idealized models in physics to simplify things. Massless ropes, point particles, ideal gases and the particle in a box are among the many simplified models used in physics. The laws of physics are represented with simple equations such as Newton's laws, Maxwell's equations and the Schrödinger equation. These laws are such as a basis for making mathematical models of real situations. Many real situations are very complex and thus modeled approximate on a computer, a model that is computationally feasible to compute is made from the basic laws or from approximate models made from the basic laws. For example, molecules can be modeled by molecular orbital models that are approximate solutions to the Schrödinger equation. In engineering, physics models are often made by mathematical methods such as finite element analysis.

Different mathematical models use different geometries that are not necessarily accurate descriptions of the geometry of the universe. Euclidean geometry is much used in classical physics, while special relativity and general relativity are examples of theories that use geometries which are not Euclidean.

A priori information

Mathematical modeling problems are often classified into black box or white box models, according to how much a priori information on the system is available. A black-box model is a system of which there is no a priori information available. A white-box model (also called glass box or clear box) is a system where all necessary information is available. Practically all systems are somewhere between the black-box and white-box models, so this concept is useful only as an intuitive guide for deciding which approach to take.

Usually it is preferable to use as much a priori information as possible to make the model more accurate. Therefore, the white-box models are usually considered easier, because if you have used the information correctly, then the model will behave correctly. Often the a priori information comes in forms of knowing the type of functions relating different variables. For example, if we make a model of how a medicine works in a human system, we know that usually the amount of medicine in the blood is an exponentially decaying function. But we are still left with several unknown parameters; how rapidly does the medicine amount decay, and what is the initial amount of medicine in blood? This example is therefore not a completely white-box model. These parameters have to be estimated through some means before one can use the model.

In black-box models one tries to estimate both the functional form of relations between variables and the numerical parameters in those functions. Using a priori information we could end up, for example, with a set of functions that probably could describe the system adequately. If there is no a priori information we would try to use functions as general as possible to cover all different models. An often used approach for black-box models are neural networks which usually do not make assumptions about incoming data. Alternatively the NARMAX (Nonlinear AutoRegressive Moving Average model with eXogenous inputs) algorithms which were developed as part of nonlinear system identification\textsuperscript{[3]} can be used to select the model terms, determine the model structure, and estimate the unknown parameters in the presence of correlated and nonlinear noise. The advantage of NARMAX models compared to neural networks is that NARMAX produces models that can be written down and related to the underlying process, whereas neural networks produce an approximation that is opaque.

Subjective information

Sometimes it is useful to incorporate subjective information into a mathematical model. This can be done based on intuition, experience, or expert opinion, or based on convenience of mathematical form. Bayesian statistics provides a theoretical framework for incorporating such subjectivity into a rigorous analysis: we specify a prior probability distribution (which can be subjective), and then update this distribution based on empirical data. An example of when such approach would be necessary is a situation in which an experimenter bends a coin slightly and tosses it once, recording whether it comes up heads, and is then given the task of
predicting the probability that the next flip comes up heads. After bending the coin, the true probability that the coin will come up heads is unknown; so the experimenter would need to make a decision (perhaps by looking at the shape of the coin) about what prior distribution to use. Incorporation of such subjective information might be important to get an accurate estimate of the probability.

The state $S_1$ represents that there has been an even number of 0s in the input so far, while $S_2$ signifies an odd number. A 1 in the input does not change the state of the automaton. When the input ends, the state will show whether the input contained an even number of 0s or not. If the input did contain an even number of 0s, $M$ will finish in state $S_1$, an accepting state, so the input string will be accepted.

The language recognized by $M$ is the regular language given by the regular expression $1^*(0 (1^*) 0 (1^*) *)^*$, where "*" is the Kleene star, e.g., $1^*$ denotes any non-negative number (possibly zero) of symbols "1".

Many everyday activities carried out without a thought are uses of mathematical models. A geographical map projection of a region of the earth onto a small, plane surface is a model which can be used for many purposes such as planning travel.

Another simple activity is predicting the position of a vehicle from its initial position, direction and speed of travel, using the equation that distance traveled is the product of time and speed. This is known as dead reckoning when used more formally. Mathematical modeling in this way does not necessarily require formal mathematics; animals have been shown to use dead reckoning.

Population Growth. A simple (though approximate) model of population growth is the Malthusian growth model. A slightly more realistic and largely used population growth model is the logistic function, and its extensions.

Individual-based cellular automata models of population growth

Model of a particle in a potential-field. In this model we consider a particle as being a point of mass which describes a trajectory in space which is modeled by a function giving its coordinates in space as a function of time. The potential field is given by a function and the trajectory, that is a function, is the solution of the differential equation:

that can be written also as:

Note this model assumes the particle is a point mass, which is certainly known to be false in many cases in which we use this model; for example, as a model of planetary motion.

Model of rational behavior for a consumer. In this model we assume a consumer faces a choice of $n$ commodities labeled 1, 2, ..., $n$ each with a market price $p_1, p_2, ..., p_n$. The consumer
is assumed to have a cardinal utility function $U$ (cardinal in the sense that it assigns numerical values to utilities), depending on the amounts of commodities $x_1, x_2, \ldots, x_n$ consumed. The model further assumes that the consumer has a budget $M$ which is used to purchase a vector $x_1, x_2, \ldots, x_n$ in such a way as to maximize $U(x_1, x_2, \ldots, x_n)$. The problem of rational behavior in this model then becomes an optimization problem, that is:

subject to:

This model has been used in general equilibrium theory, particularly to show existence and Pareto efficiency of economic equilibria. However, the fact that this particular formulation assigns numerical values to levels of satisfaction is the source of criticism (and even ridicule). However, it is not an essential ingredient of the theory and again this is an idealization.

3 Solving Variations on Sudoku Puzzles

3.1 Sudoku X

The “Sudoku X” puzzle is like the standard puzzle, with an extra requirement: the two long diagonals of the board must also contain each digit from 1 to 9 exactly once. Thus, any solution to the Sudoku X puzzle is also a solution to the standard Sudoku puzzle, but the converse is not the case. An example of a Sudoku X puzzle is given in Figure 3, where each long diagonal contains a dotted green line.

Figure 3: An example Sudoku X puzzle

The only additional requirements of a Sudoku X puzzle are that the two long diagonals have exactly one of each digit. This results in 18 additional constraints (two diagonals with nine digits each). To capture the requirement for the positive diagonal, we add the nine constraints $X 9 r=1 xrrk = 1, k = 1 : 9$. That is, each number on the positive diagonal appears exactly once. Similarly, for the anti-diagonal, the following set of nine constraints are added $X 9 r=1 xr(10−r)k = 1, k = 1 : 9$. As a result of these two additional constraints, every digit must appear exactly once on each diagonal.

3.2 Four Square Sudoku

The “Four Square” Sudoku is again a standard puzzle but with an extra requirement. There are four shaded $3 \times 3$ regions on the Sudoku board, and in addition to the requirements of the standard Sudoku, each shaded region must also contain each digit from 1 to 9 exactly once. A sample Four Squares puzzle is given in Figure 4.

4 Creating the Puzzles

How might Sudoku puzzles be created? Should our Sudoku puzzle have a unique solution? According to [7], the answer is yes. A simple search on your favorite search engine will also show that many web sites related to puzzle design agree. However, armed with the BILP from this paper, one can occasionally verify, for instance with the Matlab program sudoku.m, that not all Sudoku puzzles that are posted have unique solutions. One can stumble across such examples when the BILP’s computed solution is distinct from the solution provided by the puzzle creators.

4.1 Creating Puzzles by Brute Force

A first approach to creating Sudoku puzzles is to use brute force. One simple idea fills each element of the $9 \times 9$ matrix with a randomly chosen integer from 1 to 9, then checks to see if the resulting matrix satisfies the three Sudoku properties concerning rows, columns, and submatrices. This approach creates $9^{81} \approx 1.97 \times 10^{77}$ different matrices that require checking. Just how many of these would satisfy the Sudoku properties? In other words, how many feasible $9 \times 9$ Sudoku matrices are there (i.e., matrices satisfying constraints (1)-(4) of the BILP of Section 2.1)? First, note that a $9 \times 9$ Latin square consists of sets of the numbers 1 to 9 arranged in such a way that no row or column contains the same number twice. Therefore, every Sudoku puzzle is a special case of a $9 \times 9$ Latin square of which there are 5, 524, 751, 496, 156, 892, 842, 531, 225, 600 =
5.525 × 1027 [1]. How many of these Latin squares are Sudoku matrices? The answer to this question was provided by Felgenhauer and Jarvis in 2005. The number of Sudoku matrices for the standard 9 × 9 game was calculated to be 6, 670, 903, 752, 021, 072, 936, 960 ∼ 6.67 × 1021 [5]. This number is equal to 9! × 722 × 27, 704, 267, 971, the last factor being prime. The result was derived through logic and brute force computation. (Note, only about .00012% of 9 × 9 Latin squares are valid Sudoku puzzles.) Later Russell and Jarvis [11] showed that when symmetries were taken into account, there were, of course, many fewer solutions; 5,472,730,538 to be exact. Given that Sudoku matrices can be created by a brute force technique, assume we stop as soon as we find one. With a full Sudoku matrix in hand, we could then simply omit entries to create a puzzle. At which point, the question becomes how to do the omitting so that a proper Sudoku puzzle results. Specifically, we pose the following mathematical questions.

5 Conclusion

This paper examined the popular Sudoku puzzles from two angles: puzzle solution and puzzle creation. The first portion of the paper presented a binary integer programming formulation that solves any n × n Sudoku puzzle. A Matlab m-file, which executes a branch and bound solution method, is available for download. Further, such an approach was extended to variations on the traditional Sudoku puzzle. The second half of the paper presented theorems for creating new Sudoku puzzles. We discovered that, starting with one Sudoku puzzle, we can easily produce a daily calendar of Sudoku puzzles (enough for the entire next century!). By adding or removing givens, we can also vary the level of difficulty of the games. Answers to the exercises are provided below, and we hope students attempt and enjoy the challenge competitions.

References

5. Bertram Felgenhauer and Frazer Jarvis. There are 66709035202102936960 Sudoku grids. http://www.afjarvis.staff.shef.ac.uk/sudoku/