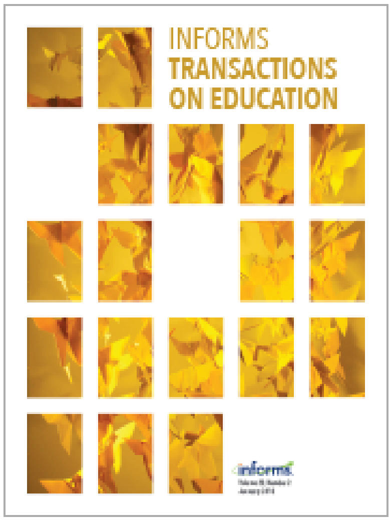

INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Puzzle—Integer Linear Programming: Spreadsheet Solver Excellence Without Excel

Evan Barlow

INFORMS TRANSACTIONS ON EDUCATION
Vol. 24, No. 2, January 2024, pp. 196–199
ISSN 1532-0545 (online)
https://pubsonline.informs.org/journal/ited

Puzzle

Integer Linear Programming: Spreadsheet Solver Excellence Without Excel

Evan Barlow[a]

[a] Goddard School of Business and Economics, Weber State University, Ogden, Utah 84414
Contact: evanbarlow@weber.edu, https://orcid.org/0000-0001-9793-9193 (EB)

Received: November 8, 2022
Revised: July 6, 2023; August 10, 2023
Accepted: August 14, 2023
Published Online in Articles in Advance: September 11, 2023

https://doi.org/10.1287/ited.2022.0068

Abstract. Because of the limitations on problem size with the built-in optimization solver in Microsoft Excel, other software applications are required to illustrate the power of integer linear programming in spreadsheets. The built-in linear optimization solver in LibreOffice Calc allows much larger problems. In this paper, we build a spreadsheet in LibreOffice Calc to solve sudoku problems and a simple generalization that further illustrates the power of integer linear programming in spreadsheets. The generalization we consider, n-doku, also introduces an interesting puzzle that includes, as a special case, a union of multiple sudoku puzzles. Besides further demonstrating the power of integer linear programming in spreadsheets, we use the n-doku solutions to develop optimization intuition and to, more generally, foster critical thinking. We use this spreadsheet in the classroom to transition from continuous to categorical decisions, emphasizing the idea of one-hot encoding. We also illustrate multiple optima and add preferences to sudoku puzzles as a way introduce multiobjective optimization.

Supplemental Material: The online supplement is available at https://doi.org/10.1287/ited.2022.0068.

Keywords: integer linear programming • spreadsheet • sudoku

1. Introduction

With the built-in limits on the size of optimization problems of Microsoft Excel's Solver add-in, it is difficult to show the full power and current limits of integer linear programming in spreadsheets. Furthermore, as of the time of publication, Microsoft Excel's built-in Solver is buggy on Mac computers when binary and/or integer constraints are added to the optimization program. Because of these limitations in Excel, we demonstrate the solution of integer linear programs using LibreOffice Calc's built-in CoinMP linear solver. We illustrate linear problems with binary constraints with the puzzle sudoku. A generalization of sudoku, which we call n-doku, is also presented.

2. Literature Review

Games and puzzles have been shown to improve engagement and learning in classroom environments (Whitton and Moseley 2012). Sudoku is a type of Latin square puzzle (Chlond 2013) that has been used to aid instruction on several different topics. Previous publications on the use of sudoku puzzles to enhance learning in optimization courses include Chlond (2005), where the author encourages solution using Mosel; Rasmussen and Weiss (2007), where the authors focus on the use of the AllDifferent constraint in Excel's solver; and Weiss and Rasmussen (2007), where the authors focus on the integration of Excel's built-in solver with Visual Basic for Applications.

In this paper, attention is focused on the use of the sudoku puzzle and a simple generalization to illustrate basic elements of the model formulation and the puzzle's solution using a solver built into the LibreOffice Calc spreadsheet program.

3. Model

In this section, we first present the model of the sudoku puzzle and then generalize the model to our version of n-doku.

In the classic version of sudoku, a 9×9 grid is divided into nine disjoint 3×3 cells. We let i refer to the row and j refer to the column of a location in the grid. We let the set of 3×3 cells be denoted $\mathcal{C}$ with a single element denoted by c. The grid itself is denoted Γ. There are nine unique objects (typically the numbers one through nine). We let the set of objects to be assigned to locations of the grid be $\mathcal{S}$ with an individual object denoted s. If object s is in the grid element at row i and column j, we say that $\Gamma_{i,j} = s$. We let $x_{i,j,s}$ refer to the number of objects s located in the grid element at the ith row and the jth column. For the classic version of sudoku, $x_{i,j,s}$ is a binary decision variable. At the beginning of each puzzle, some grid locations have the object specified. We denote the starting grid by Γ^0 and the number of objects of type s in the grid element (i,j) by $x^0_{i,j,s}$.

We now present the constraints that a solution of the sudoku puzzle must satisfy.

- Each grid element must contain exactly one object:

$$\sum_{s\in\mathcal{S}} x_{i,j,s} = 1 \quad \forall i \in \{1\ldots 9\}, j \in \{1\ldots 9\}. \tag{1}$$

- Each object must appear once (and only once) in each row of the grid:

$$\sum_{j=1}^{9} x_{i,j,s} = 1 \quad \forall i \in \{1\ldots 9\}, s \in \mathcal{S}. \tag{2}$$

- Each object must appear once (and only once) in each in each column of the grid:

$$\sum_{i=1}^{9} x_{i,j,s} = 1 \quad \forall j \in \{1\ldots 9\}, s \in \mathcal{S}. \tag{3}$$

- Each object must appear once (and only once) in each 3×3 cell of the grid:

$$\sum_{(i,j)\in c} x_{i,j,s} = 1 \quad \forall c \in \mathcal{C}, s \in \mathcal{S}. \tag{4}$$

- The quantity of each object in each grid location must be at least as large as the quantity of that object in that grid location in the starting grid:

$$x_{i,j,s} \ge x^0_{i,j,s} \quad \forall i \in \{1\ldots 9\}, j \in \{1\ldots 9\}, s \in \mathcal{S}. \tag{5}$$

- Each decision variable is binary:

$$x_{i,j,s} \in \{0,1\} \quad \forall i \in \{1\ldots 9\}, j \in \{1\ldots 9\}, s \in \mathcal{S}. \tag{6}$$

We also consider a generalization of sudoku that we term n-doku, in which the right-hand side of the constraints in Equations (1)–(4) is n instead of unity. Furthermore, the constraint in Equation (6) is modified so that each x is no longer binary but instead, belongs to the set $\{0,1,\ldots,n\}$. A special case of this generalization to n-doku is the union of n sudoku puzzles.

4. Spreadsheet

In the spreadsheet, the following ranges are needed: a 1×1 range (a single cell) representing n (equal to one for a sudoku puzzle), a 9×9 range for the input sudoku (or n-doku) puzzle in its original grid form, an 81×9 range for the decision variables, an 81×1 range for the sum of each row of the decision variable range, an 81×9 range representing the "initial puzzle constraint" (i.e., Equation (5)), a 9×9 range representing the "row constraint" (i.e., Equation (2)), a 9×9 range representing the "column constraint" (i.e., Equation (3)), a 9×9 range representing the "subgrid constraint" (i.e., Equation (4)), and a 9×9 range representing the solution in grid form. For the generalization to n-doku, an additional 81×9 range was used to help translate the decision cells to grid form for the solution.

For each of the ranges with nine columns, each column represents an $s \in \mathcal{S}$. For each of the ranges with 81 rows, each row represents an element of the puzzle grid (i.e., an (i,j) pair).

Finally, it is worth noting that in our implementation of the spreadsheet, the 9×9 range with the starting grid (i.e., Γ^0) must be set to a data type of text. This allows for our use of the "LEN" spreadsheet formula to calculate $x^0_{i,j,s}$ (i.e., the number of each $s \in \mathcal{S}$ in each starting grid location).

Our implementation of the LibreOffice Calc spreadsheet accompanies this publication as the online supplement.

5. Classroom Execution

In this section of the paper, we outline our sudoku and n-doku classroom experience and illustrate how it advances students toward the following learning outcomes:

- gain practice and insights regarding model building with spreadsheets,
- experience the power and ease of integer linear programming in spreadsheets,
- gain exposure to optimization of categorical decisions in spreadsheet solvers using a "counting" approach (e.g., one-hot encoding for a categorical decision with mutually exclusive options), and
- experience iterative objective definition and refinement of multiple optima.

5.1. Introduction to Sudoku Puzzles

To introduce sudoku and its rules, we show an easy puzzle to walk through and illustrate the decisions and constraints. Then, we highlight a single grid cell whose value is unknown but can easily be determined by application of the constraints. The students are asked to work independently or with another student to identify the correct value of the highlighted grid cell in less than a minute. Next, to further solidify the constraints and to introduce the idea of multiple possible puzzle solutions, students are given a blank grid and are told to generate a solution to the puzzle in less than two minutes.

At this point in the lesson, we typically ask several questions to tie the puzzle to the learning objectives. Some example questions are given.

- Is there any mathematical significance to the numbers one through nine? Could they be replaced by any nine distinct objects? What are some real-world decisions that you encounter that are categorical in nature instead of numerical? Note that some students identify decisions that are discrete (e.g., integer) rather than categorical.

- What is the objective function for sudoku puzzles? Is there anything that makes one complete feasible grid better than another complete feasible grid? What are some real-world contexts in which any feasible solution is, for all intents and purposes, as good as any other feasible solution?
- What are some puzzle-irrelevant objectives functions you could add to make some feasible solutions better than others? Note that some examples would be to minimize the numerical value of the top left grid cell, to minimize the sum of values along the grid diagonals, and to minimize the sum of values in the top left half of the grid. Most student suggestions (like those given here) add a numerical relevance to the decisions so that they are no longer purely categorical in nature. Some student suggestions will be nonlinear, rendering problems unsolvable in LibreOffice Calc's built-in solver.
- What can you do if adding a single objective function still results in multiple optima? How can we apply a second objective function while making sure the decisions are still optimal for the first objective function? Can you think of real-world examples in which objective functions have priority in this way?

5.2. Mathematical Model Building

There are many opportunities for learning while building the mathematical model.

5.2.1. Learning from Student Approaches. If students are given a chance to first try to formulate the model before instructor guidance or intervention, different students (or groups of students) typically try different approaches. During implementation of this class exercise, some students build a combinatorial optimization problem. Most other students try to develop an algorithm that mimics the logical reasoning most people leverage when trying to solve a sudoku puzzle: sequentially identifying grid cells for which the existing partial solution fully identifies the correct grid cell value through the enforcement of constraints. Although both of these approaches are more intuitive for most people, programmatically solving the puzzles using either of these approaches presents great difficulty. If no students develop a integer linear programming problem, a good way to guide them toward it is to challenge them to formulate the problem in a way that there are nine decisions associated with each grid cell. It should be highlighted that this kind of problem reformulation is an essential problem-solving skill. It also aligns with the important critical thinking skill of identifying benefits and drawbacks of multiple approaches.

During the formulation of the problem as a integer linear programming problem, emphasis may be placed on the idea of one-hot encoding as a common way of modeling categorical decisions. Parallels to common machine learning tasks that use one-hot encoding to model categorical variables can help reinforce learning at this stage.

5.2.2. Feasible Vs. Optimal. Formulation of the mathematical model presents an opportunity to reinforce the idea that the original sudoku puzzle problem has no objective function; any set of decisions that satisfy all constraints is no better than any other feasible set of decisions. If puzzle-irrelevant objective functions are identified by students, the objective functions can be built into the mathematical model. Students should be able to identify whether each objective function maintains linearity of the model. Students should also be challenged to consider whether the objective functions developed give any numerical meaning to the categorical decisions.

5.2.3. Multiobjective Optimization. If students suggest more than one puzzle-irrelevant objective function to the mathematical formulation, this is a good opportunity to illustrate how multiobjective optimization can be used in the case that a "primary" objective function fails to yield a unique solution.

5.2.4. Generalization to N-Doku. If time allows, the mathematical formulation presented for "n-doku" can be illustrated. This generalization includes the solution of multiple independent sudoku puzzles as a possibility. In the case that the mathematical formulation of an n-doku puzzle is the union of multiple independent sudoku puzzles, multiple solutions may exist even if each independent sudoku puzzle only offers a unique solution.

5.3. Model Solution

Solving the model using LibreOffice Calc's built-in solver is possible because the model size limits are much more forgiving than Excel's built-in solver. Because Calc's solver often requires up to a minute on a desktop or laptop computer, this problem is very fitting for illustrating current practical limits of linear solvers with integer constraints. These practical limits are much looser than the model size constraints in Excel's built-in solver. Furthermore, because Excel's current solver add-in on Mac computers is riddled with bugs for integer optimization, Calc's linear solver is more reliable for these problems.

5.3.1. Feasible Vs. Optimal. If students are given the opportunity to suggest puzzle-irrelevant objective functions, the instructor can show that for many published sudoku puzzles, the objective function is irrelevant because only one feasible solution exists. However, if starting values are removed from the initial puzzle, multiple solutions become possible, and the objective functions carry meaning.

Table 1. Student Feedback

Statement	Average response
Exploring a spreadsheet solution to sudoku puzzles was a valuable usage of class time	4.4
It was beneficial to see the current practical limits of integer linear optimization in spreadsheets	4.7
Because of the sudoku spreadsheet in-class exercise, I better understand categorical variables and how to represent them numerically	4.3
The in-class sudoku puzzle activity helped me understand the difference between a feasible and optimal set of decisions	4.2
Because of the sudoku spreadsheet in-class activity, I better understand how to deal with multiple objective functions in a decision problem	4.4
I appreciated learning about LibreOffice Calc as an alternative spreadsheet application	4.6

5.3.2. N-Doku Solutions and Uniqueness. As an exercise with the spreadsheet, the instructor could show the solution of three different puzzles; the first two would be any two independent sudoku puzzles (preferably with unique solutions), and the third would be the n-doku solution of the union of the two sudoku puzzles. Although it is possible that the solver's solution of the n-doku puzzle is equal to the union of the two separate sudoku puzzles, it is very unlikely because each starting puzzle's starting values lose association with the individual puzzle and instead, become associated with the puzzle of the union. Because many different sudoku puzzles can be combined to give the same n-doku union, the generated solution of the n-doku puzzle could be the union of any two such puzzles.

6. Student Feedback

Students in an undergraduate business class were asked to provide feedback regarding their in-class experience with the sudoku puzzle solver spreadsheet. They were asked to respond to the statements in Table 1 on a scale from one to five (five equals strongly agree and one equals strongly disagree). As seen in the student feedback, students in this undergraduate business class appreciated understanding the current practical limits of integer linear optimization in spreadsheets to a very high degree. In addition, students greatly appreciated learning about LibreOffice Calc as a spreadsheet alternative to Microsoft Excel. Although the student response ratings on the other learning objectives were lower, students still positively rated the exercise and its ability to aid in learning.

7. Conclusion

In this article, we present a spreadsheet for solving both sudoku puzzles and a generalization we call "n-doku" puzzles. In building and solving the spreadsheet to solve these puzzles, several important problem-solving and optimization lessons are illustrated. First of all, we use the LibreOffice Calc built-in solver to illustrate alternative spreadsheet applications and to demonstrate the current practical limitations of linear integer optimization solvers. Students also learn how to represent categorical variables numerically with one-hot encoding. The sudoku puzzles help illustrate the differences between feasible and optimal decisions. Students also learn about multiobjective optimization through the use of puzzle-irrelevant objective functions. Overall, students reported an enjoyable use of class time to learn valuable lessons.

References

Chlond MJ (2005) Classroom exercises in IP modeling: Su doku and the log pile. *INFORMS Trans. Ed.* 5(2):77–79.

Chlond MJ (2013) Puzzle—Latin square puzzles. *INFORMS Trans. Ed.* 13(2):126–128.

Rasmussen RA, Weiss HJ (2007) Advanced lessons on the craft of optimization modeling based on modeling sudoku in Excel. *INFORMS Trans. Ed.* 7(3):228–237.

Weiss HJ, Rasmussen RA (2007) Lessons from modeling sudoku in Excel. *INFORMS Trans. Ed.* 7(2):178–184.

Whitton N, Moseley A (2012) *Using Games to Enhance Learning and Teaching* (Taylor & Francis, New York).