# Sudoku: Strategy Versus Structure*

J. Scott Provan

## 1 Introduction

Sudoku puzzles, and their variants, have become extremely popular in the last decade, and can now be found daily in most major U.S. newspapers. In addition to the countless books of Sudoku puzzles, there are many guides to Sudoku strategy and logic. (Some good references are the books [1, 3], and the web pages [5, 6]. The reader is also directed to these for explanations of some of the terms mentioned throughout this discussion.) The purpose of this paper is to relate a common class of strategies, used to solve the vast majority of Sudoku puzzles, to the formulation of Sudoku puzzles as *assignment problems* and as *linear programs*. In particular, we give a simple characterization of this class, using a well-known graph theorem, and show further how the ability of this set of strategies to solve a Sudoku puzzle also implies that the solution can be represented as the unique nonnegative solution to a system of linear equations. These results provide excellent applications of principles commonly presented in introductory classes in finite mathematics and combinatorial optimization, and point as well to some interesting open research problems in the area.

## 2 Sudoku Puzzles, Solution Formats, and Linear Systems

A general Sudoku puzzle is defined by

- a set $S$ of $n$ *grid squares*,

- an *index set* $I = \{1, \ldots, m\}$,
- a collection $\mathcal{B}$ of *blocks*, each block $B \in \mathcal{B}$ consisting of a set of exactly $m$ squares in $S$, and
- an *initial assignment* $A = \{(p_i, k_i) : \ i = 1, \ldots, r\}$, with square $p_i \in S$ assigned index $k_i \in I$, $i = 1, \ldots, r$.

The goal is to assign indices from $I$ to each of the remaining squares of $S$ in such a way that each block $B \in \mathcal{B}$ has a complete set $I$ of indices assigned to it. A requirement for all valid Sudoku puzzles is that there is *exactly one* solution that is consistent with the initial assignment.

A *standard* Sudoku puzzle has grid squares comprising a $9 \times 9$ grid, with 27 blocks represented by the nine rows, nine columns, and nine $3 \times 3$ subsquares of that grid. An example, along with its unique solution, is given in Table 1.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 4 | | | | | | 6 | 8 |
| 7 | | | | | 5 | 3 | | |
| | | 9 | | 2 | | | | |
| 3 | | | 5 | | | | | 7 |
| | | 1 | 2 | 6 | 4 | 9 | | |
| 2 | | | | | 7 | | | 6 |
| | | | | 5 | | 7 | | |
| | | 6 | 3 | | | | | 1 |
| | 4 | 8 | | | | | 3 | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 5 | 4 | 2 | 7 | 9 | 3 | 1 | 6 | 8 |
| 7 | 1 | 8 | 6 | 4 | 5 | 3 | 9 | 2 |
| 6 | 3 | 9 | 8 | 2 | 1 | 5 | 7 | 4 |
| 3 | 6 | 4 | 5 | 8 | 9 | 2 | 1 | 7 |
| 8 | 7 | 1 | 2 | 6 | 4 | 9 | 5 | 3 |
| 2 | 9 | 5 | 1 | 3 | 7 | 8 | 4 | 6 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 9 | 5 | 6 | 3 | 7 | 8 | 4 | 2 | 1 |
| 4 | 8 | 7 | 9 | 1 | 2 | 6 | 3 | 5 |

Table 1: A standard Sudoku puzzle and its solution.

Other popular Sudoku puzzles such as *Jigsaw Sudoku*, *Sudoku X*, *Windodoku*, and *Asterisk* can also be described in the above form.

The solution to a Sudoku puzzle can be represented as the unique 0-1 solution for a specific set of linear equations. In particular, define the $mn$ variables $x_{pk}$ to have $x_{pk} = 1$ if the index $k$ is assigned to square $p$ of the grid and $x_{pk} = 0$ otherwise. In order to solve a given Sudoku puzzle, the variables $x_{pk}$ must satisfy the following set of constraints:

- Every square contains exactly one index:

$$\sum_{k \in I} x_{pk} = 1,\ p \in S, \tag{1}$$

- Every block has exactly one of each index:

$$\sum_{p \in B} x_{pk} = 1,\ B \in \mathcal{B},\ k \in I, \tag{2}$$

- Each initial assignment is honored:

$$x_{p_i k_i} = 1,\ i = 1, \ldots, r. \tag{3}$$

A 0-1 solution to equations (1)–(3) will be a solution to the associated Sudoku puzzle. It is useful to note here that equations (1) and (2) comprise a set of *assignment* constraints, where 1-1 assignments are made in each block. The fact that assignments are made over multiple overlapping blocks makes the problem difficult. It is known that solving general-sized Sudoku puzzles is NP-hard, even for square grids with blocks consisting of the sets of rows and columns (Latin Squares) [2], or for $p^2 \times p^2$ grids with blocks consisting of rows, columns, and the $p^2$ partitioned $p \times p$ subsquares [7], Section 3.2. Finding 0-1 solutions to equations (1)–(3) for the standard $9 \times 9$ version, however, is quite easy using any reasonable integer program solver.

Solving Sudoku puzzles by hand is generally done through *elimination strategies* that keep track of what indices are available to be placed in each square of the grid, and updating these by eliminating indices that cannot be allowed in a square based on some line of reasoning. Specifically, we define a *candidate set* associated with each square $p$, denoted $C_p$, to be the set of indices that have not been eliminated from consideration for that square. Initially, $C_{p_i} = \{k_i\}$ for assigned squares $(p_i, k_i)$, and $C_p = I$ for unassigned squares. In this context equation (3) can be replaced by

$$x_{pk} = 0,\ p \in S,\ k \notin C_p. \tag{3$'$}$$

When the candidate set for any square has only one index in it, then that square can be assigned this index. The Sudoku is solved when only one index remains in every one of the candidate sets.

Two questions are of interest here.

- How easy is it to solve Sudoku puzzles, that is, how likely is it that one can solve a Sudoku puzzle by employing a specified set $\mathcal{R}$ of rules for eliminating elements from candidate sets? A rule set $\mathcal{R}$ can become quite complex, and could include extensive chain reasoning, in order that it be powerful enough to solve the harder Sudoku puzzles. At present there is no known set of rules, short of using trial and error, that is guaranteed to solve all standard $9 \times 9$ Sudoku puzzles.

- How close is the linear system given by (1)–(3) above to solving a Sudoku puzzle? One popular heuristic for finding 0-1 solutions to a set of linear equations is to *relax* the problem to that of finding *nonnegative* solutions to these equations. Finding nonnegative solutions to a system of equations is considerably easier than finding 0-1 solutions, and if the relaxed solution has all 0-1 values, then it will in fact be a solution to the Sudoku puzzle. It would be interesting to know under what circumstances this does occur, since this would make the solution for larger puzzles much faster.

It turns out that these two questions are related in an interesting way. We investigate a particular set $\mathcal{R}$ of rules, called the *one-block strategies*, that are almost universally used among Sudoku enthusiasts. We give a simple characterization for this set of rules, and show that the success of the one-block strategies in solving a Sudoku puzzle also means that the relaxation above is guaranteed to give a solution to the associated Sudoku puzzle.

# 3 One-Block Strategies and Relaxations

To answer the first question above, we investigate one of the simplest classes of elimination strategies:

> One-block strategy: A strategy that eliminates a particular assignment based on the constraints (1), (2), and (3′) as they apply to a *single* block $B \in \mathcal{B}$.

That is, a one-block strategy involves looking at the relationship between candidate indices in just one block, ignoring how they interact with other blocks. Note that a one-block strategy is not restricted to be used only on a *particular* block, but can be applied successively to different blocks, so long as each application considers just one block, along with the current candidate sets.

A set of one-block elimination strategies that is in virtually every intermediate player's arsenal can be described fairly succinctly.

> PIGEON-HOLE RULE: Let $M \subset I$ be a subset of indices, and let $D$ be a subset of squares, all contained in a single block $B$, such that (a) $|M| = |D|$ and (b) $C_p \subseteq M$ for every square $p \in D$. Then the elements of $M$ can be removed from $C_p$ for each $p \in B \setminus D$.

In other words, if there is any subset of $k$ squares in a single block whose candidate sets together contain only $k$ different indices, then these indices can appear nowhere else in that block. This rule, or tandem uses of it, includes most of the basic Sudoku strategies such as "squeezing," "crosshatching," "lone-number spotting," "naked/hidden pairs/triples/quads," etc., and is fairly easy to implement. Using this set of rules alone seems to solve about 90% of all Sudoku puzzles, and for our example it reduces the number of candidate indices considerably, as shown in Table 3. (We leave to the interested reader the task of applying the appropriate rules to obtain the table numbers.) Although the pigeon-hole rules were able to determine the solution numbers for 63 out of the possible 81 squares in this example, the rules are not quite powerful enough to solve the entire puzzle.

It turns out that the pigeon-hole rules account for *all* of the one-block solution strategies.

**Theorem 1** *Any elimination that can be made using a one-block strategy can also be inferred using one of the pigeon-hole rules.*

*Proof:* Consider a one-block strategy applied to block $B \in \mathcal{B}$ at a particular stage of the solution to the puzzle, with current candidate sets $C_p,\ p \in S$. Then the one-block strategy must use only the following subsets of constraints (1), (2), and (3′) to imply that a particular assignment $x_{p_0 k_0} = 1$ cannot hold:

$$\sum_{k \in I} x_{pk} = 1,\ p \in B, \tag{4}$$

$$\sum_{p \in B} x_{pk} = 1,\ k \in I, \tag{5}$$

$$x_{p,k} = 0,\ p \in B,\ k \notin C_p. \tag{6}$$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 5 | 4 | 2 3 | 7 | 9 | 1 3 | 1 2 5 | 6 | 8 |
| 7 | 1 2 | 8 | 6 | 4 | 5 | 3 | 1 2 9 | 2 9 |
| 6 | 1 3 5 | 9 | 8 | 2 | 1 3 | 1 5 | 7 | 4 |
| 3 | 6 | 4 | 5 | 8 | 9 | 1 2 | 1 2 | 7 |
| 8 | 7 | 1 | 2 | 6 | 4 | 9 | 5 | 3 |
| 2 | 9 | 5 | 1 | 3 | 7 | 8 | 4 | 6 |
| 1 9 | 1 2 3 | 2 3 | 4 | 5 | 6 | 7 | 8 | 2 9 |
| 5 9 | 2 5 | 6 | 3 | 7 | 8 | 4 | 2 9 | 1 |
| 4 | 8 | 7 | 9 | 1 | 2 | 6 | 3 | 5 |

Table 2: The candidate sets $C_p$ after applying all possible pigeon-hole rules to the puzzle given in Table 1.

In particular, the assignment $(p_0, k_0)$ is eliminated if, when we set $x_{p_0 k_0} = 1$, we cannot find a 0-1 solution to equations (4)–(6).

Now equations (4)–(6) can be interpreted as requiring an assignment of the indices in $I$ to the squares in block $B$, using only assignments allowed by the sets $C_p$, $p \in B$, or equivalently, finding a *perfect matching* on the bipartite subgraph $G$ of $B \times I$ that contains only the edges $(p, k)$, $k \in C_p$. Assignment $(p_0, k_0)$ is eliminated by demonstrating that no perfect matching exists that contains the edge $(p_0, k_0)$.

Forcing the edge $(p_0, k_0)$ into the matching is equivalent to removing the vertices $p_0$ and $k_0$ from $G$, along with their adjacent edges. The assignment $(p_0, k_0)$ is then eliminated if and only if the resulting graph $G'$ admits no perfect matching. By Hall's Theorem [4], Section 6.3.1, if $G'$ has no perfect matching then there exist subsets $X \subseteq I \setminus \{k_0\}$ and $D \subseteq B \setminus \{p_0\}$, with $|X| < |D|$, such that every edge of $G'$ that is adjacent to a vertex in $D$ is also adjacent to a vertex in $X$. But this in turn means that the set $M = X \cup \{k_0\}$ has the property that $|M| \leq |D|$ and every edge of the original graph $G$ that is adjacent to a vertex in $D$ is also adjacent to a vertex in $M$, that is, $C_p \subseteq M$ for every $p \in D$. Further, since the original graph $G$ does admit a perfect matching, then $|M| = |D|$, and it follows that the pigeon-hole rule, using sets $D$ and $M$ as defined above, eliminates the assignment $(p_0, k_0)$. ■

To answer the second question posed in Section 1, we investigate under what circumstances the relaxation of the linear system defined by (1)–(3) to nonnegative variables is guaranteed to produce a 0-1 solution, and hence to solve the Sudoku puzzle. It turns out that the success of the pigeon-hole rules in solving a Sudoku puzzle is also a guarantee that the relaxation solves it as well.

**Theorem 2** *If a Sudoku puzzle can be solved completely using pigeon-hole rules, then there exists a* unique *nonnegative solution to equations (1)–(3), which in turn is a solution to the puzzle itself.*

*Proof:* Let $x^*$ be the unique 0-1 solution for equations (1)–(3), and suppose there is a second nonnegative solution $\hat{x}$ for these equations. Begin solving the puzzle using pigeon-hole rules until the first point at which an index $k_0$ is eliminated from one of the candidate sets $C_{p_0}$ for which $\hat{x}_{p_0 k_0} \neq 0$. This must always happen, since the pigeon-hole rules eventually eliminate every candidate pair $(p, k)$ for which $x^*_{pk} = 0$, and there must be at least one of these for which $\hat{x}_{pk} \neq x^*_{pk}$. Now by Theorem 1, this elimination is forced by the one-block equations (4)–(6), and further, equation (6) does not yet include the pair $(p_0, k_0)$. Consider the linear program

$$\max z = x_{p_0 k_0} : x \geq 0 \text{ and equations (4)–(6) hold.} \tag{7}$$

Then $\hat{x}$ is feasible for (7), since again by the choice of $(p_0, k_0)$, the other assignments for equation (6) are already satisfied by $\hat{x}$. Further, since the constraints of (7) are assignment constraints, then (7) will always have a 0-1 solution (see [4], Section 6.3.1). Therefore the optimal objective function value $z_*$ for (7) must likewise be 0 or 1. But the fact that $\hat{x}_{p_0 k_0} > 0$ implies that $z_* \neq 0$, and the fact that $k_0$ was eliminated from $C_{p_0}$ at this point using a one-block strategy implies that $z_* \neq 1$. This is a contradiction, and therefore there cannot be a second nonnegative solution to equations (1)–(3). ■

# 4 Extensions and Further Questions

One further question one might consider is whether Theorem 2 can be extended to rule sets that are more sophisticated than the one-block strategies. A logical extension would be to *two-block strategies*, that is, strategies that involve simultaneously considering two blocks in order to eliminate a candidate. Strategies such as "intersection removal," "cross-constraints," and "pointing pairs" are examples of two-block strategies. It turns out that our example can be completely solved by a single application of a two-block strategy (though not of any of the types above), along with the one-block strategies. Specifically, start with the candidate sets in Table 3 obtained after applying the one-block strategies, and consider the two intersecting blocks consisting of the top left $3 \times 3$ square and the second column. We can eliminate the index 1 from the square (1,1), since if we assign 1 to this square, we must assign 2 to square (2,2) and 5 to square (3,2) in order to satisfy the block constraints for the $3 \times 3$ square. But then square (8,2) cannot be assigned either of the indices 2 or 5 without violating the block constraints for column 2. Eliminating index 1 from square (1,1), and continuing to apply the pigeon-hole rules, we proceed to obtain the complete solution shown in Table 1.

One could now ask whether expanding our strategy set by adding the two-block strategies moves us out of the realm of problems where Theorem 2 continues to hold. The example above shows that this indeed happens. A nonnegative but fractional solution to equations (1)–(3) is given in Table 3, where a singleton $k$ in a square $p$ represents the assignment $x_{pk} = 1$, and a doubleton $k_1$ and $k_2$ in square $p$ represents the assignment $x_{pk_1} = x_{pk_2} = 1/2$. Thus this example cannot be solved uniquely by relaxing equations (1)–(3), even though it can be solved by using only one- and two-block strategies.

We end the paper by leaving the reader with several interesting open questions that are suggested by the above results:

1. Is there a good description of all two-block elimination strategies analogous to the pigeon-hole rules for one-block strategies?

2. What is the minimum number $k$ for which the $k$-block strategies (elimination strategies that consider $k$ blocks simultaneously) solve all standard Sudoku puzzles? Obviously 27-block strategies will work, and one can find examples where 2-block strategies are not strong enough. We

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 5 | 4 | 2 3 | 7 | 9 | 1 3 | 2 5 | 6 | 8 |
| 7 | 1 2 | 8 | 6 | 4 | 5 | 3 | 1 9 | 2 9 |
| 6 | 3 5 | 9 | 8 | 2 | 1 3 | 1 5 | 7 | 4 |
| 3 | 6 | 4 | 5 | 8 | 9 | 1 2 | 1 2 | 7 |
| 8 | 7 | 1 | 2 | 6 | 4 | 9 | 5 | 3 |
| 2 | 9 | 5 | 1 | 3 | 7 | 8 | 4 | 6 |
| 1 9 | 1 3 | 2 3 | 4 | 5 | 6 | 7 | 8 | 2 9 |
| 5 9 | 2 5 | 6 | 3 | 7 | 8 | 4 | 2 9 | 1 |
| 4 | 8 | 7 | 9 | 1 | 2 | 6 | 3 | 5 |

Table 3: A fractional LP solution for the puzzle given in Table 1.

suspect, though, that the number is considerably closer to 2 than to 27.

3. Is there a set of elimination rules whose success in solving a Sudoku puzzle characterizes all Sudoku instances where equations (1)–(3) admit a unique nonnegative solution?

4. What is the simplest set of elimination rules that is guaranteed to solve all standard Sudoku puzzles?

# References


[1] D. Berthier, *The Hidden Logic of Sudoku*, 2nd ed., Lulu, Morrisville, NC, 2007.

[2] C. Colbourn, The complexity of completing partial Latin squares, *Discrete Appl. Math.* **8** (1984), 25–30.

[3] P. Gordon and F. Longo, *Mensa Guide to Solving Sudoku*, Sterling, New York, 2006.

[4] S. M. Ross, *Topics in Finite and Discrete Mathematics*, Cambridge University Press, Cambridge, 2000.

[5] Andrew Stuart's webpage, available at `http://www.scanraid.com`.

[6] Wikipedia entry on Sudoku, available at `http://wikipedia.org`.

[7] T. Yato, *Complexity and Completeness of Finding Another Solution and Its Application to Puzzles*, Master's thesis, University of Tokyo, 2003, available at `http://www-imai.is.s.u-tokyo.ac.jp/~yato/data2/MasterThesis.pdf`.

J. Scott Provan received his B.A. from Carnegie-Mellon University and his Ph.D. from Cornell University. He currently teaches at the University of North Carolina.
*Department of Statistics and Operations Research, University of North Carolina, Chapel Hill, NC 27599-3260*
*Scott_Provan@UNC.edu*