The Sudoku Game I (sudoku)

"The essence of mathematics is not to make simple things complicated, but to make complicated things simple."

Once on a cold winter day I was knocked down by a bad virus. Laying in bed I came across the number puzzle *Sudoku* while reading a magazine. Immediately I was addicted to this current Sudoku craze like many others and was knocked down a second time by the *Sudoku fever*. I went to my computer and made a search on "Sudoku". There are virtually hundreds of sites on Sudoku and looking at them, everybody must have been informed about Sudoku – except me. In many ways it resembles previous fads like Rubrik's Cube and "Instant Insanity". The challenging new puzzle game had taken the world by storm.

Problem: So, what is it? The puzzle is simple and can be explained in one sentence: "Fill in the grid (see, for example, Figure 1) so that every row, every column, and every 3×3 subblock contains the digits 1 to 9." That's all there is to it!

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

Figure 1: A Partially Filled Sudoku

It's challenging. It's addictive! Manual solving time is typically from 10 to 30 minutes, depending on skill and experience.

The puzzle seems to have first been published in New York in the late 1970s by the specialist puzzle publisher Dell in its magazine Math Puzzles and Logic Problems, under the title Number Place. The person who designed the puzzle and composed the first of its kind is not recorded, but it was probably Walter Mackey, one of Dell's puzzle constructors.

In 2004 and 2005, Sudoku experienced a surge of popularity in the United Kingdom. British newspapers have a long history of publishing crosswords and other puzzles, but before 2004, Sudoku was not a common feature on puzzle pages. The Times launched its regular Sudoku puzzle, introduced by Wayne Gould, on 12th November 2004.

A sudoku grid is a special case of a mathematical object called a *Latin square*. A Latin square consists of n sets of numbers from 1 to n arranged in a square pattern so that no row or column contains the same number twice.

The additional constraint of a standard nine-by-nine sudoku puzzle, that each three-by-three block also must contain each of the nine digits, reduces the enormous number of possible nine-by-nine Latin squares to a smaller but still-humungous number: 6,670,903,752,021,072,936,960, which is roughly the number of micrometers to the nearest star. This number is equivalent to $9! \cdot 72^2 \cdot 2^7 \cdot 27,704,267,971$, the last factor of which is prime.

See http://www.afjarvis.staff.shef.ac.uk/sudoku/
for a discussion on how Bertram Felgenhauer of Dresden, Germany, obtained this number, which represents how many unique, one-solution puzzles can be produced. Of course, some of the 9×9 grids can easily be transformed into others; by relabelling the numbers, by rotating or reflecting the grid, and by permuting certain rows and columns. Ed Russell and Frazer Jarvis have counted the number of "essentially different" sudoku grids as 5,472,730,538. The number of valid Sudoku solution grids for the 16×16 derivation is not known.

Depending on the number of clues and the size of the grid, sudoku puzzles can be extremely difficult to solve. Takayuki Yato and Takahiro Seta of the University of Tokyo have proved that solving $n \times n$ sudoku puzzles in general is NP-complete. Indeed, solving Sudoku puzzles can be expressed as a graph coloring problem. The aim of the puzzle in its standard form is to construct a proper 9-coloring of a particular graph, given a partial 9-coloring. The graph in question has 81 vertices, one vertex for each cell of the grid. The vertices can be labelled with the ordered pairs (x, y). where x and y are integers between 1 and 9. In this case, two distinct vertices labelled by (x, y) and (x', y') are joined by an edge if and only if

1. $x = x'$ or
2. $y = y'$ or
3. $\lceil x/3 \rceil = \lceil x'/3 \rceil$ and $\lceil y/3 \rceil = \lceil y'/3 \rceil$

The puzzle is then completed by assigning an integer between 1 and 9 to each vertex, in such a way that vertices that are joined by an edge do not have the same integer assigned to them.

(See http://en.wikipedia.org/wiki/Mathematics_of_Sudoku)

The maximum number of givens that can be provided while still not rendering the solution unique, regardless of variation, is four short of a full grid; if two instances of two numbers each are missing and the cells they are to occupy are the corners of an orthogonal rectangle, there are two ways the numbers can be added. The inverse of this problem – the fewest givens that render a solution unique – was an unsolved problem for several years, although the lowest number found for the standard variation without a symmetry constraint is 17, a number of which have been found by Japanese puzzle enthusiasts and 18 with the givens in rotationally symmetric cells.[1]

Gordon Royle (`(http://www.csse.uwa.edu.au/~gordon/sudokumin.php`) has a collection of 32930 distinct Sudoku configurations with 17 entries.

Of course, it is possible to set up starting grids with more than one solution and to set up partially filled grids with no solution, but such are not considered proper Sudoku puzzles, a unique solution is expected.

If an efficient solver is available, there is a simple method of automatic construction: randomly add a digit to the grid, and then look for a solution. If no solution is found, remove the digit and try another. Otherwise, look for a different solution. If there is no other solution, accept the current digit; otherwise, repeat this process. Another method is to start with a full grid (a solution) and remove repeatedly number, solve it each time and stop, before the removed number generates a grid that has more than one solution. (This method to generate Sudoku puzzle has also been implemented in LPL and is presented in model sudokuGen[2].)

Many freely downloadable efficient programs have been developed to solve Sudoku problems. Even Sudoku generator programs are available, that means, programs that generate partially filled grids that can be uniquely completed to a Sudoku grid.

Modeling Steps

To solve the puzzle, we use mixed integer programming (see [5] or [3], see also for a slightly different formulation: [2] and [1]). Dozens of enumeration algorithms implemented in various programming languages can be found in the Internet.

For a mathematical formulation, we define first the dimension S. Normally, the game is played with a dimension of $S = 3$. This gives a grid size of S^2 (S^2 rows times S^2 columns. The grid is partitioned into S^2 subblocks of size of $S \times S$ cells.

1. Let $i, j \in \{1 \dots S^2\}$ be the set of rows and columns, and let $k \in \{1 \dots S^2\}$ the set of numbers placed in the cells. These sets have the size S^2. Furthermore, we introduce the sets $h, g \in \{1 \dots S\}$ the subblock rows and columns. These sets have the size S.

2. The data is a partially filled grid, for which we use the notation $P_{i,j}$ (The data are read from a file).

3. The variable is the essential element of the formulation: we introduce a binary variable $x_{i,j,k}$ for every tripel (i, j, k).[3] This variable $x_{i,j,k}$ is 1, if the cell (i, j) contains the number k, otherwise it is 0. For example, $x_{2,3,5} = 1$ means that the cell $(2, 3)$ in the Sudoku table contains the number 5. The conditions for these variables can now be easily stated as follows (for all $i, j, k \in \{1 \dots S^2\}$ and $g, h \in \{1 \dots S\}$):

[1] In the meantime (January 2012), the problem has been solved. Gary McGuire, Bastian Tugemann and Gilles Civario have proven in 7 million hours of computer time that indeed there exists no unique sudoku issue from a 16 cells occupied state (see: http://arxiv.org/abs/1201.0749 (January 2012).

[2] `http://lpl.unifr.ch/lpl/Solver.jsp?name=/sudokuGen`

[3] An obvious and straightforward approach to introduce variables for this problem would be to define for each cell (i, j) a general integer variable $x_{i,j} \in [1 \dots S^2]$. The two reasons why we use $0 - 1$ variables are: (1) The constraints become easier to be formulated, (2) The model is much easier to solve although we get S^2 times more variables. nevertheless, in model sudokuInt[4] this integer formulation is also presented.

4. Each cell (i, j) contains a single integer k :

$$\sum_k x_{i,j,k} = 1$$

5. Each integer k appears only once in each row i :

$$\sum_j x_{i,j,k} = 1$$

6. Each integer k appears only once in each column j :

$$\sum_i x_{i,j,k} = 1$$

7. Each integer k appears only once in each subblock (h, g) :

$$\sum_{h1 \in h, g1 \in g} x_{(h-1)\cdot S+h1,(g-1)\cdot S+g1,k} = 1$$

   Explanation: Suppose we have the subblock $(3, 2)$ with $S = 3$ (the second block from the left and the third from the top). In the expression above, we sum all variables $x_{i,j,k}$ with $4 \leq i \leq 6$ and $7 \leq j \leq 9$. These are exactly all variables in the subblock $(3, 2)$. Then, each subblock must contain every k exactly once. This is what is imposed by the constraint.

8. Furthermore, we must fix the enforced cells by a fixed assignment to a single variable. In the Figure 1, for example, we have in cell $(2, 4)$ the number 9, already given. Hence, we must have $x_{2,4,9} = 1$. To assign all fixed cells, we read a table $P_{i,j}$ from a file and assign $x_{i,j,k}$ to 1, if the corresponding cell (i, j) in table $P_{i,j}$ has the number k. Hence:

$$x_{i,j,k} = 1, \qquad \text{for all } (i, j, k) \text{ such that } P_{i,j} = k$$

9. Since we are not looking for a particular solution, we can maximize anything or simply use the `solve` statement.

10. To translate the solution into a 2-dimensional table again, we use the parameter $Q_{i,j}$.

11. The next instructions of the LPL code are just for drawing the resulting grid. Several data sets for the sudoku puzzles can be found in the following file: puzzles/-sudoku.txt[5]. Another list with 24′620 puzzles with only 17 given entries are found here: puzzles/sudoku17.txt[6].

The complete model code in LPL for this model is as follows (see [4]):

[5] http://lpl.unifr.ch/lpl/lplmodels/puzzles/sudoku.txt
[6] http://lpl.unifr.ch/lpl/lplmodels/puzzles/sudoku17.txt

Listing 1: The Model

```
model sudoku "The Sudoku Game I";
  set i,j,k           "Rows/cols/numbers";
  set h,g             "Subblocks";
  parameter P{i,j}    "Given table";
  parameter S         "The dimension";
  parameter ID:=[1]   "The puzzle data ID";
  binary variable x{i,j,k} "k is in cell (i,j)?";
  constraint
    N{i,j}: sum{k} x = 1 "In each cell only one k";
    R{i,k}: sum{j} x = 1 "Every k in each row i";
    C{j,k}: sum{i} x = 1 "Every k in each column j";
    B{h,g,k}:
      sum{v in h,u in g} x[(h-1)*S+v,(g-1)*S+u,k] = 1;
    F{i,j,k|P[i,j]=k}: x = 1 "Fixed cells";
  solve;
  integer parameter Q{i,j} := argmax{k} x;

  Draw.Scale(25,25);
  {i} Draw.Line(i,0,i,S*#g,2);
  {j} Draw.Line(0,j,S*#h,j,2);
  {g,h} Draw.Rect(S*(g-1),S*(h-1),S+.02,S+.02,-1,0,3);
  {i,j|Q} Draw.Text(Q&'',j-.65,i-.35,12,if(P,5,3));
  model data "A data set read from sudoku.txt";
    Read('sudoku.txt,%'&ID&':Table', S);
    i := 1..S^2;
    h := 1..S;
    Read{i}('%'&ID&';1', {j}P);
  end
  model data1 "A data set read from sudoku17.txt";
    S:=3;
    string parameter li;
    Read('sudoku17.txt,%;'&(ID-1), li);
    i := 1..S^2;
    h := 1..S;
    P{i,j} := Strsub(li,(i-1)*9+j,1);
  end
end
```

Solution: A solution for the data in Figure 1 is shown in Figure 2. This MIP problem is easy to solve.

This MIP model formulation is also able to solve larger games, which are much more difficult otherwise. In Figure 3 we show a game of dimension $S = 5$. The solution is shown in Figure 4.

Question (Answer see)

1. Solve Sudoku Number 40 from file `sudoku17.txt`.

Answer (Question see)

1. The Sudoku data in file puzzles/sudoku17.txt[7] are compactly stored Each line defines the data $P_{i,j}$. The digit are lexicographically ordered. Modify $ID = 40$ and interchange the two submodel names `data` and `data1`. LPL defines a string `li` –

[7] `http://lpl.unifr.ch/lpl/lplmodels/puzzles/sudoku17.txt`

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

Figure 2: ThThe Solution of the Suduku

a line in the file. Before reading the line into `li`, the code "overreads" $39 (= ID - 1)$ lines. Every single digit from the string is then assigned to P.

References

[1] M. J. Chlond. http://archive.ite.pubs.informs.org/Vol5No2/Chlond/.

[2] M. J. Chlond. Classroom Exercises in IP Modeling: Su Doku and The Log Pile. *INFORMS Transactions on Education*, 5:2, 2005.

[3] Della F. Croce. Sudoku and OR. federico.dellacroce@polito.it, 2006. Working Paper.

[4] T. Hürlimann. Reference Manual for the LPL Modelling Language, most recent version. www.virtual-optima.com.

[5] Thorsten Koch. Rapid Mathematical Programming or How to Solve Sudoku Puzzles in a few Seconds. *ZIB-Report 05-51*, 2005.

					3	13	21	17			25			2			10	11	19			18	7	14
	16						10	7	20	13	4	19	1		9	12					21			5
					4	24	14			23		3	11	15	25	20		5	2	1	6			
			14	7		9			16	21	22						4	23		15	13	25		
	2	25	9	20	12			19	15			17	7									24		4
		9	23				18		5	11		12					6	2	22	13	20	19		
	18		6	1		7		13				14		25	23			12	21			3		22
	12						8	3			23	16				24		25	17	7				1
7	20		17		9	16	4			19		21		13	18		8		11	2		15	24	
						6			22	18	9							4		12		10	14	
6	23			15		18			25	17									13		1	22	20	7
		10					12	20	4		2	1	23					3		19		8	21	13
	3		1		11						19	20	13						4		15		23	
2	13	8		21		1					5	6	22		20	14	17					12		
20	11	24	7		22									9	12			1		6			10	2
	8	20		18		21							12	1	13			6						
	7	2		16	20		1		23	15		25		5			21	22	12		24		3	9
15				22	10	5		12				11	2			17	3						25	
9		5			8	3			19	14		22				4		18		10	11		12	
		6	3	24	7	14	22					13		20	1		9				8	5		
22		23									17	9			14	13			16	18	4	7	2	
		3	16	2		19	25						5	22	4			7		24	17			
			19	4	18	22		8	6	25	12	23		7			15	10	20					
18			21					23	2		13	10	3	14	6	11	1						5	
5	6	7			1	11	16			2			4			3	24	21	23					

Figure 3: A Sudoku of Dimension $S = 5$

23	4	15	24	6	3	13	21	17	1	12	25	5	9	2	8	16	10	11	19	20	22	18	7	14
3	16	18	22	17	6	25	10	7	20	13	4	19	1	8	9	12	14	24	15	23	21	2	11	5
21	10	13	12	19	4	24	14	22	8	23	18	3	11	15	25	20	7	5	2	1	6	16	9	17
8	5	1	14	7	2	9	11	18	16	21	22	24	20	10	17	6	4	23	3	15	13	25	19	12
11	2	25	9	20	12	23	5	19	15	6	14	17	7	16	21	1	22	13	18	3	10	24	8	4
16	25	9	23	8	14	10	18	1	5	11	7	12	24	4	3	15	6	2	22	13	20	19	17	21
10	18	11	6	1	15	7	2	13	24	5	20	14	17	25	23	9	19	12	21	8	16	3	4	22
4	12	14	2	5	19	20	8	3	21	22	23	16	15	6	10	24	13	25	17	7	9	11	18	1
7	20	22	17	3	9	16	4	25	12	19	1	21	10	13	18	5	8	14	11	2	23	15	24	6
24	19	21	15	13	17	6	23	11	22	18	9	2	8	3	16	7	20	4	1	12	5	10	14	25
6	23	19	5	15	16	18	3	24	25	17	10	4	14	12	2	21	11	8	13	9	1	22	20	7
17	22	10	18	9	5	15	12	20	4	7	2	1	23	11	24	25	16	3	6	19	14	8	21	13
25	3	16	1	12	11	2	6	10	14	8	19	20	13	21	7	22	18	9	4	5	15	17	23	24
2	13	8	4	21	23	1	19	9	7	24	5	6	22	18	20	14	17	15	10	11	25	12	16	3
20	11	24	7	14	22	8	17	21	13	3	16	15	25	9	12	19	23	1	5	6	18	4	10	2
14	8	20	11	18	25	21	15	16	9	4	3	7	12	1	13	10	5	6	24	17	2	23	22	19
19	7	2	10	16	20	17	1	4	23	15	6	25	18	5	11	8	21	22	12	14	24	13	3	9
15	1	4	13	22	10	5	24	12	18	9	8	11	2	23	19	17	3	20	14	21	7	6	25	16
9	21	5	25	23	8	3	13	6	19	14	24	22	16	17	15	4	2	18	7	10	11	1	12	20
12	17	6	3	24	7	14	22	2	11	10	21	13	19	20	1	23	9	16	25	4	8	5	15	18
22	15	23	8	11	21	12	20	5	3	1	17	9	6	24	14	13	25	19	16	18	4	7	2	10
1	14	3	16	2	13	19	25	15	10	20	11	8	5	22	4	18	12	7	9	24	17	21	6	23
13	24	17	19	4	18	22	9	8	6	25	12	23	21	7	5	2	15	10	20	16	3	14	1	11
18	9	12	21	25	24	4	7	23	2	16	13	10	3	14	6	11	1	17	8	22	19	20	5	15
5	6	7	20	10	1	11	16	14	17	2	15	18	4	19	22	3	24	21	23	25	12	9	13	8

Figure 4: The Solution of a Sudoku of Size $S = 5$