Scratchwork: Sudoku

Let's try to solve Sudoku as an integer progrmaming problem. There are many acceptable answers.

3	7			9	4		5	1 2 6 7

We can solve this problem *twice*. Once with Linear Algebra, and again piping that linear algebra into programs like COIN-OR or possibly `numPy`, with Python or Julia language. The data structure common to all of these will be `[integer]` and `Array`. We'll solve a system of linea inequality with 100 equations and 800 unknowns, with values in $\{0, 1\}$.

Example How do we formalize the notion of square, where we know some of the values but not enough of them to put in a number?

1	2
6	7

Example How do we formalize that moment when because we know the value at one square, we can ignore all the othe rsquares it the row?

←	—	—	7	—	—	—	—	→

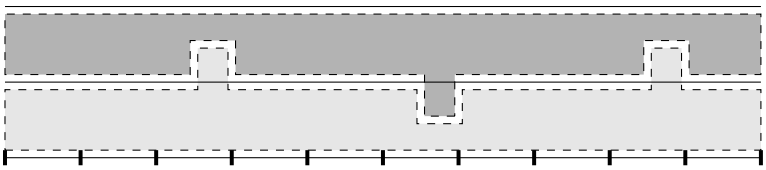
We could encode this sudoku as a $9 \times 9 \times 9$ array, x_{ijk} with $0 \leq i, j, k < 9$. If we set $x_{ijk} = 1$ we are saying there is the value k at the square (i, j). And $x_{ijk} = 0$ if the number is not there. Our constraints read:

- $x_{ijk} \in \{0, 1\}$
- each square contains a number $\sum_k x_{ijk} = 1$

- each row contains a number $\sum_i x_{ijk} = 1$
- each column contains a number $\sum_j x_{ijk} = 1$
- each 3×3 block contains a number $\sum_{(i,j) \in 3\times 3} x_{ijk} = 1$

We now have to tell our computer how to describe the same thing. `x[i][j][k]`

The integer programming textbook lumps all problems of this kind into the same format $Ax \leq b$ where A is some $m \times n$ matrix representing all linear constraints and b is a $n \times 1$ vector containing all the numbers. How do we list all of our Sudoku rules as "constraints" and linear inequalities?



References

[1] **Optimization Methods in Business Analytics** `http://web.mit.edu/15.053/www/`

[2] Michele Conforti, Gérard Cornuejols, Giacomo Zambelli. **Integer Programming** (GTM #271) Springer, 2014.