Sudoku and OR

F. Della Croce
Dipartimento di Automatica ed Informatica, Politecnico di Torino
E-mail address: federico.dellacroce@polito.it

Keywords: Combinatorial Optimization, Integer Programming

1 Introduction

Following the description of Wikipedia the free encyclopedia [5], Sudoku can be described as a logic-based placement puzzle. The aim of the classic version of the puzzle is to enter a numerical digit from 1 through 9 in each cell of a $9 \times 9$ grid made up of $3 \times 3$ subgrids, starting with various digits given in some cells (the givens); each row, column, and region must contain only one instance of each numeral. Further only proper puzzles are considered, that is puzzles having a unique solution. Below are reported a classic Sudoku proper puzzle and the corresponding solution.

The Sudoku puzzle was invented in Indianapolis in 1979 but reached widespread international popularity just in 2005 after being launched at the end of 2004 by one of the leading british newspapers, "The Times". A nice scientific survey of the Sudoku phenomenon was presented in [1]. Several variants have been proposed of the Sudoku puzzle. We cite among others SudokuX, where as additional constraint it is required that also the main diagonals must contain only one instance of each numeral. Alphabetical variations
have also emerged (the so-called wordoku): there is no functional difference in the puzzle unless the letters spell something. Some variants include one or two words reading somewhere in the grid once solved, where this/these word/s are described as in crosswords: determining the word in advance can be viewed as a solving aid. Below are reported a Wordoku puzzle using letters (A - B - C - I - K - O - S - T - U) and the corresponding solution.

The current president of EURO

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>O</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>I</td>
<td></td>
<td>C</td>
<td></td>
<td>T</td>
<td>U</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>I</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>B</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td>K</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B</td>
<td></td>
<td></td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S</td>
<td></td>
<td>C</td>
<td>U</td>
<td></td>
<td>I</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>K</td>
<td></td>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2 Solving proper Sudoku puzzles by means of ILP modeling

Every proper Sudoku puzzle can be tackled by means of various AI & OR techniques. For instance, on one hand the puzzle can be solved by means of constraint programming techniques (see [4]) such as the all-different operator, on the other hand it can be easily formulated as an ILP model. Here we focus on this latter approach. Given an initial grid where some elements \((i, j)\) have already been filled by digits, let \(RQ - h\) \((h = 1, \ldots, 9)\) be the \(h\)-th block of the grid. An ILP formulation \((P1)\) of the SUDOKU puzzle can be expressed as follows:

Variables:

\(x_{i,j,k} = 1\) if element \((i, j)\) has value \(k\) \((1 \leq k \leq 9, \text{ k integer})\) in the puzzle solution, else \(x_{i,j,k} = 0\).

Objective function:

there is no objective function as the purpose is just to search for a feasible solution.
Constraints:
\[ \sum_{k=1}^{9} x_{i,j,k} = 1 \quad \forall i, j \]
(each element of the grid contains one of the digits 1-9)
\[ \sum_{j=1}^{9} x_{i,j,k} = 1 \quad \forall i, k \]
(each row of the grid contains the digits 1 – 9 exactly once)
\[ \sum_{i=1}^{9} x_{i,j,k} = 1 \quad \forall j, k \]
(each column of the grid contains the digits 1 – 9 exactly once)
\[ \sum_{i,j: (i,j) \in RQ \setminus h} \sum_{k=1}^{9} x_{i,j,k} = 1 \]
\[ \forall h = 1, \ldots, 9 \]
(each block of the grid contains the digits 1 – 9 exactly once).
\[ x_{i,j,k} = 1 \quad \forall \text{ element of the entry grid } (i,j) \text{ with value } k. \]

3 Building proper Sudoku puzzles by means of ILP modeling

To build a proper Sudoku puzzle, we need to derive an initial grid such that there exists a feasible Sudoku solution compatible to that grid and this solution is unique. Below (see [2]) is indicated how to check whether a solution (given an initial grid) is or not unique. Let denote by \( SOL(i,j) \) the value of element \((i,j)\) of the grid in the feasible solution. Consider solving the following ILP model \((P2)\):

Objective function:
\[ \min Z = \sum_{i,j,k:SOL(i,j)=k} x_{i,j,k} \]
(we minimize the sum of all variables \( x_{i,j,k} \) corresponding to elements \((i,j)\) having value \( k \) in the feasible solution).

Constraints (the same constraints considered in model \( P_1 \)):
\[ \sum_{k=1}^{9} x_{i,j,k} = 1 \quad \forall i, j \]
\[ \sum_{j=1}^{9} x_{i,j,k} = 1 \quad \forall i, k \]
\[ \sum_{i=1}^{9} x_{i,j,k} = 1 \quad \forall j, k \]
\[ \sum_{i,j: (i,j) \in RQ \setminus h} \sum_{k=1}^{9} x_{i,j,k} = 1 \quad \forall h = 1, \ldots, 9 \]
\[ x_{i,j,k} = 1 \quad \forall \text{ element of the entry grid } (i, j) \text{ with value } k. \]

The solution of model \( P_2 \) clearly provides a feasible solution also to model \( P_1 \) and therefore to the SUDOKU puzzle. Further, the objective function of model \( P_2 \) minimizes the sum of the elements in the grid having the same value obtained in the solution of model \( P_1 \). Hence, as each grid is composed by 81 elements, if the objective function value of model \( P_2 \) is equal to 81 (\( Z = 81 \)), then the considered SUDOKU puzzle has unique solution, else the solution to the SUDOKU puzzle is not unique.

A straightforward approach for building a proper Sudoku puzzle is provided by the following procedure (notice that the sequence indicated below for emptying the elements is not compulsory, but it is sufficient to handle all elements under any sequence).

Procedure "Build_initial_grid":
INPUT: final solution.
OUTPUT: initial grid.
{
    for \( i=1 \) to 9
        for \( j=1 \) to 9
            fill element \((i,j)\) with Sol\((i,j)\);
            empty element \((i,j)\);
            Solve model \( P_2 \);
            if the objective function value of model \( P_2 \) is < 81 (solution not unique),
                then
                    fill element \((i,j)\) with Sol\((i,j)\);
            end if
        end for
    end for
}

This approach guarantees that the corresponding Sudoku puzzle is also irreducible, namely no element\((i,j)\) can be emptied, or else the solution becomes not unique.
4 A challenging problem for the OR community

From an OR point of view, solving/building a proper sudoku puzzle is actually quite a trivial task. However a strongly challenging combinatorial problem related to the Sudoku puzzle is concerned with the minimum number of givens for proper puzzles (notice that the inverse problem, that is the maximum number of givens that can be provided while still not rendering the solution unique has a trivial solution, namely four short of a full grid). The best available solution value for this problem is 17 (see [3] for a collection of distinct Sudoku proper initial grids with 17 givens) and it is conjectured that no 16-givens initial grids exist, evidence for which stems from extensive randomised searching. On the other hand, a trivial lower bound = 8 can be determined for this problem: indeed if two numerals $k, l$ are absent from the initial grid, then there exist at least two different solutions (it is sufficient to assign value $k$ to all elements having value $l$ and vice versa). However, to the author’s knowledge, at the current state of the art no better lower bound is available: this induces indeed quite an impressive gap and room for extensive research!

References