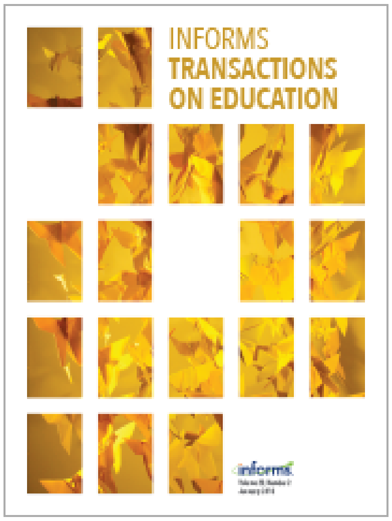


INFORMS Transactions on Education

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

Puzzle—Chess Avoidance Puzzles

Martin J. Chlond

Vol. 15, No. 3, May 2015, pp. 254–256
ISSN 1532-0545 (online)

INFORMS Transactions on Education

http://dx.doi.org/10.1287/ited.2015.0141

Puzzle

Chess Avoidance Puzzles

Martin J. Chlond
Lancashire Business School, University of Central Lancashire, Preston PR1 2HE, United Kingdom, mchlond@uclan.ac.uk

A type of chess based puzzle devised by Erich Friedman is described and modeled using integer programming. In these puzzles the solver is required to place a given set of chess pieces on to an irregular grid such that no occupied squares are attacked by another piece.

A MathProg model is included together with an Excel spreadsheet with the model embedded The purpose of this is to encourage instructors to investigate the possibility of using the Excel add-in SolverStudio as a teaching tool.

Keywords: integer programming; puzzle
History: Received March 2015; accepted March 2015.

A type of chess based puzzle devised by Erich Friedman requires the solver to place a given set of chess pieces, not including pawns, onto an irregular grid such that no occupied squares are attacked by another piece. A number of challenging examples can be found at Friedman's website at http://www2.stetson.edu/~efriedma/chessavoid/ and the reader is encouraged to attempt a couple of these before proceeding.

A typical grid is in Figure 1 and it may be required, for example, to place four bishops and two knights on the grid such that no two pieces attack each other. The solution is in the appendix.

To model these puzzles as integer programs we need to define objects as follows.

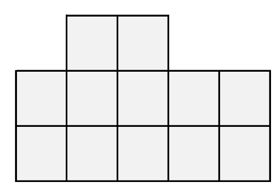
Sets
$R = 1 \ldots r$, $C = 1 \ldots c$, $P = 1 \ldots p$ where r, c, and p are the number of rows, columns, and piece types respectively.

Decision Variables
$x_{i,j,k} = 1$ if cell $\{i, j\}$ contains a piece of type k, 0 otherwise, $\forall\, i \in R,\ j \in C,\ k \in P$.

Auxiliary Variables
$a_{i,j} = 1$ if cell $\{i, j\}$ is occupied by some chess piece,
$qa_{i,j} = 1$ if cell $\{i, j\}$ is occupied by a queen, and
$ba_{i,j} = 1$ if cell $\{i, j\}$ is occupied by a bishop,

in each case, 0 otherwise, $\forall\, i \in R,\ j \in C$.

Figure 1 Typical Grid for Chess Avoidance Puzzle



Parameters
N, where n_k is the number of pieces of type k with $k = 1, 2, 3, 4, 5$ representing queen, bishop, knight, rook and king respectively.
DAS, parameters table, which contains coordinates of disallowed cells,
NM, which contains details (explained below) to assist in the modeling of knight moves. For display purposes N is used to represent a knight as K represents a king.

Constraints
Compute whether or not each cell is occupied:

$$\sum_{k \in P} x_{i,j,k} = a_{i,j} \quad \forall\, i \in R,\ j \in C.$$

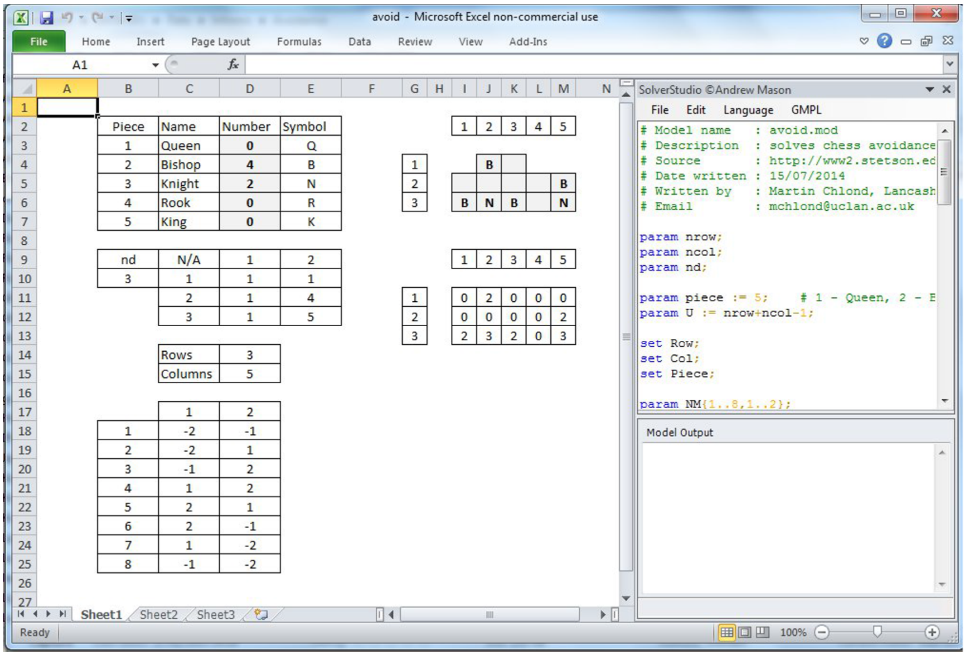
Correct number of each piece used:

$$\sum_{i \in R} \sum_{j \in C} x_{i,j,k} = N_k \quad \forall\, k \in P.$$

No more than one piece per square:

$$a_{i,j} \le 1 \quad \forall\, i \in R,\ j \in C.$$

Figure 2 **SolverStudio in Action**



No occupied square is attacked by a queen. First ensure that auxiliary variables are set to one for all squares attacked by at least one queen:

$$\sum_{\substack{p\in R\\ p\neq i}} x_{p,j,1} + \sum_{\substack{q\in C\\ q\neq j}} x_{i,q,1} + \sum_{\substack{m\in R\\ m\neq i\\ m-i+j\geq 1\\ m-i+j\leq c}} x_{m,m-i+j,1}$$

$$+ \sum_{\substack{m\in R\\ m\neq i\\ i+j-m\geq 1\\ i+j-m\leq c}} x_{m,i+j-m+j,1} \leq 99qa_{i,j} \quad \forall\, i\in R,\ j\in C.$$

(Note that the value 99 used in the right-hand-side constraint may be any value greater than or equal to the number of cells in the grid.)

Then make sure that a cell cannot be both occupied and attacked by a queen:

$$qa_{i,j} + a_{i,j} \leq 1 \quad \forall\, i\in R,\ j\in C.$$

A similar two-stage procedure ensures that no occupied square is attacked by a bishop:

$$\sum_{\substack{m\in R\\ m\neq i\\ m-i+j\geq 1\\ m-i+j\leq c}} x_{m,m-i+j,2} + \sum_{\substack{m\in R\\ m\neq i\\ i+j-m\geq 1\\ i+j-m\leq c}} x_{m,i+j-m+j,2} \leq 99ba_{i,j} \quad \forall\, i\in R,\ j\in C,$$

$$ba_{i,j} + a_{i,j} \leq 1 \quad \forall\, i\in R,\ j\in C.$$

No occupied square is attacked by a rook:

$$\sum_{\substack{p\in R\\ p\neq i}} x_{p,j,4} + \sum_{\substack{q\in C\\ q\neq j}} x_{i,q,4} + 99a_{i,j} \leq 99 \quad \forall\, i\in R,\ j\in C.$$

No occupied square is attacked by a king:

$$\sum_{\substack{k=i-1\\ k\geq 1\\ k\leq r}}^{i+1} \sum_{\substack{m=j-1\\ m\geq 1\\ m\leq c}}^{j+1} x_{k,m,5} - 2x_{i,j,5} + 99a_{i,j} \leq 99 \quad \forall\, i\in R,\ j\in C.$$

No occupied square is attacked by a knight:

$$\sum_{\substack{k=1\\ 1\leq i+NM_{k,1}\leq r\\ 1\leq j+NM_{k,2}\leq c}}^{8} x_{i+NM_{k,1},\,j+NM_{k,2},\,3} + 99a_{i,j} \leq 99 \quad \forall\, i\in R,\ j\in C.$$

Ensure disallowed cells are unoccupied:

$$a[DAS_{h,1}, DAS_{h,2}] = 0 \quad \forall\, h\in 1..nd.$$

A MathProg implementation of this model (avoid.mod) and a spreadsheet with the model embedded (avoid.xlsx) are provided with this article (available as supplemental material at http://dx.doi.org/10.1287/ited.2015.0141). To access and use the model within the spreadsheet it is necessary to install the Excel add-in SolverStudio. This is available at http://solverstudio.org and some useful background and description to the add-in is in Mason (2013). SolverStudio is a very useful teaching tool. It combines the ease of use and familiarity of a spreadsheet with the more powerful features offered by mathematical programming languages. To encourage the reader to investigate this further we present and briefly describe the image in Figure 2.

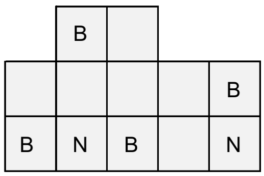
The right hand panel contains the working Mathprog model with a couple of modifications to read input data from and output results to the spreadsheet area in the left hand panel.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/ited.2015.0141.

Appendix

Figure A.1 Solution to Puzzle in Figure 1



References

Mason AJ (2013) SolverStudio: A new tool for better optimisation and simulation modelling in Excel. *INFORMS Trans. Ed.* 14(1):45–52.