IP Modeling of Chessboard Placements and Related Puzzles

Martin J. Chlond and Cath M.Toase
Lancashire Business School
University of Central Lancashire
Preston, PR1 2HE, UK

mchlond@uclan.ac.uk
cmtoase1@uclan.ac.uk

Abstract

A number of chessboard placement and closely related puzzles are examined and formulated as Integer Programs. These puzzles require the participant to place pieces on a board according to certain constraints and may be generalized to boards containing any number of squares, and pieces displaying properties unfamiliar in the game of chess.

1. Introduction

The literature on recreational mathematics contains many references to puzzles that require the participant to place pieces on a chessboard according to specific constraints (Dudeney, 1917;Kraitchik, 1942; Schuh, 1943). These may be generalized to boards containing any number of squares and to pieces displaying properties not permitted in the game of chess. By formulating the problem situations as Integer Programs, the solutions to puzzles of this type may be methodically deduced. A few examples of these chessboard placement problems, and the IP models to solve them, will be presented in this paper. The aim is to encourage the use of such puzzles in an educational context to improve and develop modeling skills.

The applicability of a known technique from the field of OR/MS to the seemingly unrelated context of puzzles for leisure has educational implications. If OR/MS techniques can play a role in games for fun, then, conversely, games may have a role to play in teaching OR/MS.

'Making learning fun' as a means of motivating students to learn is an established teaching approach, well documented in the literature on school and higher education (Gage & Berliner, 1992; Walkin, 1990; M.Rawson, 1999). The value of game playing, specifically, to encourage active rather than passive learning has also been expounded (D.Adams, 1973; Cowan, 1998).

Students of business in universities come from a range of educational backgrounds. For some, mathematics may be a subject that they have not recently studied and, of which, they have less than fond memories. Integer Programming techniques, requiring an understanding of mathematical logic and the ability to use mathematical notations, may well be anxiety provoking to such students. The usefulness of integer programming in making operational decisions is not likely to be sufficiently motivating in such cases. As teaching tools, the beauty of puzzles and games is in their capacity for mass appeal and in that they do not, on the face of it, have anything to do with the more serious and contextually dry worlds of business or mathematics. As Martin Gardner (1975) puts it:

> *"No student is motivated to learn advanced group theory, for example, by telling him that he will find it beautiful and stimulating, or even useful, if he becomes a particle physicist. Surely the best way to wake up a student is to present him with an intriguing mathematical game, puzzle, magic trick, paradox, limerick or any score of things that dull teachers tend to avoid because they seem frivolous."*

In Britain, the Department for Education's Numeracy Initiative in schools seeks to echo this attitude in its guidelines for teaching mathematics in British schools. The sense of satisfaction at solving a puzzle, the buzz of discovery, the joy of 'winning' are all recognized as intrinsically motivating forces (DFE, 1999; Anderson et al, 1997).

The growth of international students among the UK universities means that it is increasingly important to find methods and techniques to encourage learning that have cross cultural application and appeal (Glauco de Vita, 1999). The puzzles we discuss in this paper, being based upon or closely related to the universally popular game of chess, fulfil this requirement.

2. A Pot-Pourri of Placement Puzzles

A number of chess piece placement problems are now described together with a few problems that are similar enough in structure to warrant discussion in this context.

2.1 Standard Chess Piece Placements

The Non-attacking Queens Problem

What is the maximum number of queens that can be placed on a chessboard such that no queen is attacked by another.

The Queen Domination Problem

What is the minimum number of queens that can be placed on a chessboard so that each square contains a queen or is attacked by one.

Non-dominating Queens

A puzzle referred to as 'the non-dominating queens problem' has been considered at some length in the literature. That is, place ***n*** queens on a board of order ***n*** in such a way as to maximise the number of squares not under attack. The formulation of this is straightforward but as ***n*** increases the solutions provide a challenge to current I.P. software. Mario Velucchi summarises the progress made and gives optimal solutions for boards up to and including order 16 and best known solutions for boards up to and including order 30.

Bishops, Rooks and Knights

The 'Non-attacking' and 'Domination' problems above have also been considered in the literature for Bishops, Knights and Rooks.

The most popular of these is perhaps the knight domination problem and is usually presented in one of two forms. That is, where occupied squares must be under attack and where occupied squares must not be under attack.

Foulds and Johnson (1984) formulate non-attacking problems for rooks, bishops, queens, knights and kings and prove that, in each case, the constraint matrix is totally unimodular.

Rook placement puzzles are less well represented in the literature as the solutions are easily achieved by casual inspection. For example, the minimum number of rooks that may be placed on the board such that every square is attacked is eight – there must be one on every rank or one on every file. The maximum number of rooks that may be placed such than none are attacked by any other is also eight - there may be one on each square of a main diagonal.

Nevertheless, the simplicity of the rook's move will provide us with a useful starting point in our analysis and there are lessons to be learned that will be of benefit in the consideration of the less accommodating pieces.

2.2 The Crowded Board

"You are given a chessboard together with 8 queens, 8 rooks, 14 bishops, and 21 knights. The puzzle is to arrange the 51 pieces on the chessboard so that no queen shall attack another queen, no rook attack another rook, no bishop attack another bishop, and no knight attack another knight. No notice is to be taken of the intervention of pieces of another type from that under consideration - that is, two queens will be considered to attack one another although there may be, say, a rook, a bishop, and a knight between them. It is not difficult to dispose of each type of piece seperately; the difficulty comes in when you have to find room for all the arrangements on the board simultaneously." (Dudeney, 1917)

2.3 Detective Chess

This type of puzzle was originated by Jaime Poniachek and explored by Martin Gardner in Isaac

Asimov's Science Fiction Magazine, reprinted in Gardner's Puzzles from other Worlds.

A number of chessboard squares are marked and these squares are to be occupied by a given set of pieces (pawns are not included). The objective is to deduce which pieces must occupy each of the marked squares. Clues are given in the form of numbered squares and these represent the number of attacks on the respective square. A sample puzzle is given below.

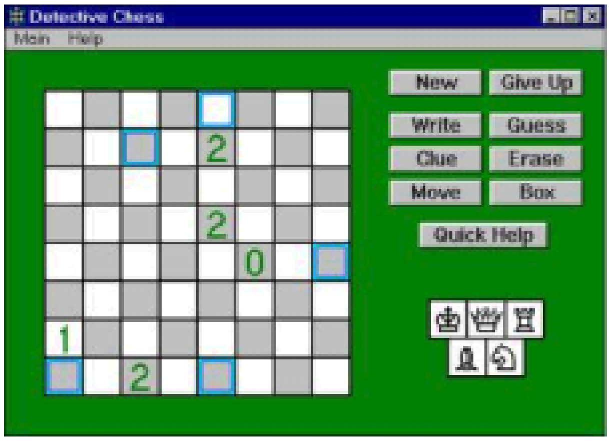

Figure 1: Detective Chess

The puzzle has been programmed by Gerry Quinn (http://indigo.ie/~gerryq/Det/Det.htm). His implementation has the facility to generate random puzzles on a range of board sizes and with a variety of pieces for placement including several borrowed from 'Fairy' chess. Each puzzle generated has a unique solution.

2.4 The Gentle Art of Stamp Licking

"If you have a card divided into sixteen spaces (4 x 4), and are provided with plenty of stamps of the values 1d., 2d., 3d., 4d., and 5d., what is the greatest value that you can stick on the card if the Chancellor of the Exchequer forbids that you place any stamp in a straight line (that is, horizontally, vertically, or diagonally) with another stamp of similar value? Of course, only one stamp can be affixed in a space." (Dudeney, 1917)

2.5 5-by-5

(http://www.chlond.demon.co.uk/Five.html)

Each of the squares in a 5-by-5 grid can be in one of two states, lit or unlit. If the player clicks on a square then that square and each orthogonal neighbor will toggle between the two states. Each mouse click constitutes one move and the objective of the puzzle is to light all 25 squares in the least number of moves. The starting configuration has all 25 squares unlit.

2.6 Lights on

(http://www.chlond.demon.co.uk/Lights.html)

This is a puzzle similar to 5-by-5 but is played on a 4-by-4 grid. The squares affected by a mouse click are the current square and all those squares within a single rook's move. Again the objective is to light up all squares. The starting configuration is generated with a random selection of squares already lit.

2.7 Equal Vision

A watchman looks in all directions (horizontal, vertical and diagonal). On the board below, each watchman has five vacant cells under his gaze. A watchman can see beyond another watchman. What is the maximum number of watchmen that can be placed so that each sees six empty cells? What if each watchman must see seven empty cells? (Poniachek, 1998)

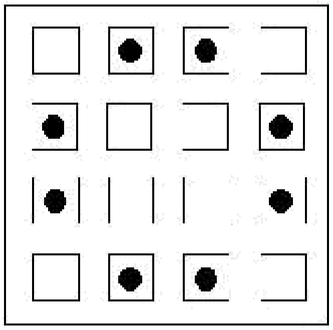

Figure 2: Equal Vision

2.8 The Abbott's Window

Dudeney's description of the Abbott's Window is quaint though unnecessarily convoluted. The puzzle in effect consists of an 8-by-8 grid where individual squares must be covered in such a way that all horizontal, vertical and diagonal lines contain an even number of uncovered squares. An added condition is that each corner square must not be covered.

3. Tools of the Trade

A collection of useful modeling structures is now presented. These include subscript and variable definitions, constraint generation expressions for the various pieces and a brief discussion on the modeling of logical conditions.

We will adopt the notation ***i = 1..s*** for row subscripts and ***j = 1..s*** for column subscripts, where ***s*** = size of board, i.e. for standard chessboard, ***s=8***.

Binary variables, $x_{i,,j}$, are defined for each square, ***{i,j}***, such that if square ***{i,j}*** is occupied then $x_{i,,j}=1$, otherwise $x_{i,,j}=0$.

$$x_{i,j} \in \{0,1\}; \forall i, j$$

An expression is required to compute the number of attacks on a specified square from pieces elsewhere on the board. If $n_{i,,j}$ = number of attacks on square ***{i,j}*** then:

$$n_{i,j} \in Z_+; \forall i, j$$

3.1 Rook attacks

$$\sum_{\substack{m=1, m\neq i,\\ 1\leq m-i+j\leq s}}^{s} x_{m,m-i+j} + \sum_{\substack{m=1, m\neq i,\\ 1\leq i+j-m\leq s}}^{s} x_{m,i+j-m} = n_{i,j}; \forall i, j$$

3.2 Bishop attacks

$$\sum_{\substack{m=1, m\neq i,\\ 1\leq m-i+j\leq s}}^{s} x_{m,m-i+j} + \sum_{\substack{m=1, m\neq i,\\ 1\leq i+j-m\leq s}}^{s} x_{m,i+j-m} = n_{i,j}; \forall i, j$$

3.3 Queen attacks

The computation of queen attacks on each square is, quite simply, a concatenation of the equivalent expressions for bishop and rook.

3.4 Knight attacks

The knight attacks are perhaps the most awkward to model in that the moves defy easy definition. Even Dudeney was exasperated by the knight's shenanigans and his contempt for the piece is made clear by his remark, "The knight is the irresponsible low comedian of the chessboard". He goes on to quote an unnamed American writer who states, "He is a very uncertain, sneaking, and demoralizing rascal". There seems to be no elegant way to capture the bizarre antics of this particular 'rascal'.

Notwithstanding the difficulties of deriving a single succinct statement to summarize the knight attacks on each square, the following compromise is presented for consideration and improvement.

Consider a 12-by-12 ***(S=s+4)*** board where the outer two ranks and files consist of 'dummy' squares. The actual chessboard consists of the central 64 squares. Then

$$X_{i-2,j-1} + X_{i-1,j-2} + X_{i+1,j-2} + X_{i+2,j-1} + X_{i+2,j+1} + X_{i+1,j+2} + X_{i-1,j+2} + X_{i-2,j+1} = n_{i,j}$$

$$i = 3,\dots, s+2, j=3,\dots, s+2$$

computes the number of knight attacks on each real square.

In addition

$$\sum_{\substack{i=1\\(i<3\\ or\\ i>s+2)}}^{S} \sum_{\substack{j=1\\(j<3\\ or\\ j>s+2)}}^{S} x_{i,j} = 0$$

ensures that no knights are placed on dummy squares.

3.5 A general constraint

An alternative approach is to construct a general constraint that is appropriate for all the pieces. If ***A{i,j}*** is equal to the set of squares that if occupied by the piece under consideration would be able to attack square ***{i,j}*** then the statement to define $n_{i,j}$ would be the same for each piece.

This seems to provide a more elegant formulation but is perhaps a little less easy to follow. Also, in order to model this using I.P. software it would be necessary to construct statements to define ***A{i,j}*** for each different piece separately.

3.6 Non-attacking constraints

Sections 3.1 to 3.4 contain statements that are of general use in the modeling of placement problems. However, non-attacking problems in particular may be formulated more efficiently by considering ranks, files and diagonals rather than individual squares. For example, if each rank and file contains no more than one rook, then no two rooks are attacking each other. The model for 'The Crowded Board' problem (section 4.4) illustrates this in more detail.

3.7 Logical constraints

Many of the placement problems require squares to be attacked or not attacked according to some specified condition, for example, maximize the number of pieces on the board such that all unoccupied squares must be under attack and all occupied squares not under attack. Williams (1999) demonstrates the use of binary variables to model logical conditions. For our purposes, it is useful to define binary variables for each square, ai,,j, such that if square {i,j} is attacked by one or more pieces then ai,,j=1, otherwise ai,,j=0. This may be achieved as follows:

$$n_{i,j} \ge a_{i,j}; \forall i,j$$
$$n_{i,j} \le M a_{i,j}; \forall i,j$$
$$a_{i,j} \in \{0,1\}; \forall i,j$$

Where ***M*** is a suitable upper bound on all $n_{i,j}$. Note that a tighter formulation is possible by defining Mi,j as upper bounds for each $n_{i,j}$.

The first statement forces $a_{i,j}$ to 0 if $n_{i,j}=0$ and the second statement forces $a_{i,j}$ to 1 if $n_{i,j} > 0$. It is usually unnecessary to include both of these statements to ensure a valid solution (see below).

These binary variables may in turn be used to ensure that the conditions of the problem under consideration are fulfilled. Taking the above example where unoccupied squares are attacked and occupied squares are not attacked, the constraints are written as follows:

$$x_{i,j} + a_{i,j} = 1; \forall i,j$$

This statement ensures that every square is either attacked or occupied, but not both. The problem here is to maximize the number of pieces placed which in effect will minimize the number of squares under attack. It is only necessary to force the ai,,j variables to 1 if the square is under attack, the converse will be implicit in the objective function.

3.8 Objective functions

In many problems, the aim is to find the maximum or minimum number of pieces that may be placed on the board without violation of some condition. In such cases, the objective function is:

$$Max \, / \, Min \sum_{i=1}^{s} \sum_{j=1}^{s} x_{i,j}$$

In other problems, the aim is to place a given set of pieces according to specified conditions. In these cases, any suitable objective function may be chosen, so that a feasible solution is obtained.

4. A Menagerie of Models

We are now in a position to formulate the puzzles described in Section 2.

4.1 Bishop placements

$$Max \, / \, Min \sum_{i=1}^{s} \sum_{j=1}^{s} x_{i,j}$$

$$\sum_{\substack{m=1, m \neq i, \\ 1 \leq m-i+j \leq s}}^{s} x_{m,m-i+j} + \sum_{\substack{m=1, m \neq i, \\ 1 \leq i+j-m \leq s}}^{s} x_{m,i+j-m} \geq a_{i,j}; \forall i, j$$

$$\sum_{\substack{m=1, m \neq i, \\ 1 \leq m-i+j \leq s}}^{s} x_{m,m-i+j} + \sum_{\substack{m=1, m \neq i, \\ 1 \leq i+j-m \leq s}}^{s} x_{m,i+j-m} \leq M a_{i,j}; \forall i, j$$

4.2 Queen placements

The two common queen puzzles, that is, queen domination and maximum queens, are formulated as for the bishop problems except that the bishop attack constraints are replaced with those of the queen.

4.3 Knight domination

The two knight domination puzzles described in Section 2.3 are modeled by combining the structure developed in Section 3.4 to compute the number of knight attacks on each real square with the logical constraints presented in Section 3.6.

In addition

$$a_{i,j} = 1; i = 3..10, j = 3..10$$

ensures all real squares are attacked.

Alternatively

$$x_{i,j} + a_{i,j} = 1; i = 3..10, j = 3..10$$

ensures only unoccupied real squares are attacked.

4.4 The Crowded Board

The objective of this puzzle is merely to place all the pieces subject to the constraints, hence, any suitable objective function will suffice.

The subscript ***k*** may be used to represent each piece type to be placed (1 = queen, 2 = bishop, 3 = knight, 4 = rook) and decision variables defined as follows:

$\boldsymbol{x_{i,j,k} = 1}$ if piece type ***k*** is placed on square ***{i,j}, 0*** otherwise

The requirement to place a number of knights on the board necessitates the use of the artifice introduced in Section 3.4, that is, we need to consider a 12x12 board where the outer two ranks and files consist of 'dummy' squares.and a set of constraints is included to ensure that no pieces are placed on dummy squares (see Section 3.4).

Constraints are required to ensure that the correct number of each piece type are placed on the board and that each square contains, at most, a single piece.

The statements required to model the condition that no two queens attack each other are listed as follows.

Firstly, it is necessary to ensure that each rank and file contains, at most, one queen.

$$\sum_{j=3}^{10} x_{i,j,1} \le 1; \boldsymbol{i} = 3..10$$

$$\sum_{i=3}^{10} x_{i,j,1} \le 1; \boldsymbol{j} = 3..10$$

Secondly, each diagonal must contain, at most, one queen.

$$\sum_{k=1}^{i} x_{k,i-k+1,1} \le 1; \boldsymbol{i} = 2..11$$

$$\sum_{k=j}^{12} x_{k,12-k+j,1} \le 1; \boldsymbol{j} = 1..11$$

$$\sum_{k=1}^{13-j} x_{k,j+k-1,1} \le 1; \boldsymbol{j} = 1..11$$

$$\sum_{k=i}^{12} x_{k,k-i+1,1} \le 1; \boldsymbol{i} = 2..11$$

Similar statements are required to ensure that each rank and file contains, at most, one rook, and each diagonal contains, at most one bishop.

An extension of the structure presented in Section 3.4 ensures that each knight is not under threat by any other knight.

4.5 Detective Chess

The Detective Chess puzzles, in their original form, are easily modeled as integer programs. Consider the following example.

Marked squares are ***{1,1}, {1,5}, {4,3}, {4,6}*** and ***{8,7}*** and will be referred to as squares 1 to 5 respectively. Squares ***{3,6}, {4,7}*** and ***{5,5}*** are numbered ***2, 1*** and ***0*** respectively and we will refer to these as target squares. The pieces to be placed are a single King, Queen, Rook, Bishop and Knight and will be referred to as pieces ***1*** to ***5*** respectively.

Let $\boldsymbol{x_{i,j} = 1}$ if square i is occupied by piece ***j, 0*** otherwise.

The equations

$$\sum_{j=1}^{5} x_{i,j} = 1; \forall i$$

and

$$\sum_{i=1}^{5} x_{i,j} = 1; \forall j$$

are assignment constraints; they ensure that each square is occupied by exactly one piece and that each piece is placed in exactly one square.

The equations

$X_{2,5} + X_{4,1} + X_{4,2} + X_{4,3} = \mathbf{2}$

$X_{4,1} + X_{4,2} + X_{4,3} + X_{5,2} + X_{5,3} = \mathbf{1}$

$X_{1,2} + X_{1,4} + X_{2,2} + X_{2,3} + X_{3,5} + X_{4,1} + X_{4,2} + X_{4,,4} = \mathbf{0}$

ensure that the target squares are attacked by the required number of pieces.

Any linear objective function will suffice, as the aim of the problems is merely to fulfil the constraints.

Quinn's implementation provides for further exploration of the idea. In particular, the 'All Numbers' game is an interesting development. In this mode, the numbers of attacks are given for all squares but no clues are forthcoming as to the location of the pieces to be placed. This presents the trainee modeler with a more strenuous exercise and offers the opportunity to bring into play the full range of tools developed above.

4.6 The Gentle Art of Stamp Licking

Define variables as follows:

$\boldsymbol{s = 4}$ - size of grid

$\boldsymbol{k = 1..5}$ - values of stamps

$\boldsymbol{x_{i,j,k} = 1}$ if stamp of value k is placed on square $\boldsymbol{\{i,j\}}$, $\boldsymbol{0}$ otherwise

$\boldsymbol{a_{i,j,k}}$ – dummy variables

Maximise the total value of stamps placed:

$$Max \sum_{i=1}^{5} \sum_{j=1}^{5} \sum_{k=1}^{5} k x_{i,j,k}$$

Each square contains, at most, one stamp:

$$\sum_{k=1}^{5} x_{i,j,k} \le 1; \forall i, j$$

$a_{i,j,k}$ ***= 1*** if square ***{i,j}*** is in line with stamp of value ***k:***

$$\sum_{\substack{m=1, m\neq i,\\ 1\leq m-i+j\leq s}}^{s} x_{m,m-i+j,k} + \sum_{\substack{m=1, m\neq i,\\ 1\leq i+j-m\leq s}}^{s} x_{m,i+j-m,k} + \sum_{\substack{m=1\\ m\neq i}}^{s} x_{m,j,k} + \sum_{\substack{m=1\\ m\neq j}}^{s} x_{i,m,k} \leq 99a_{i,j,k}; \forall i,j,k$$

If a stamp of value k is placed on square ***{i,j}*** then it is not in line with any stamp of the same value:

$$x_{i,j,k} + a_{i,j,k} \leq 1; \forall i,j,k$$

4.7 5-by-5

Consider a placement puzzle involving a hypothetical piece which may make rook-like moves but with a maximum of one square per move. Each occupied square must be attacked by an odd number of pieces and each unoccupied square must be attacked by an even number of pieces. This is analogous to the five-by-five puzzle. A mechanism to achieve these conditions is:

$$\sum_{\substack{m=i-1\\ 1\leq m\leq s\\ m\neq i}}^{i+1} x_{m,j} + \sum_{\substack{m=j-1\\ 1\leq m\leq s}}^{j+1} x_{i,m} = 1 + 2d_{i,j}; \forall i,j$$

Where $\mathbf{d}_{i,,j}$ is a dummy integer variable for each square.

The objective is to minimize the number of placements.

4.8 Lights on

This is essentially a rook placement problem.

$$\sum_{m=1}^{s} x_{m,j} + \sum_{\substack{m=1\\ m\neq i}}^{s} x_{i,m} = 2d_{i,j} + r_{i,j}; \forall i,j$$

where di,,j is defined as for the five-by-five puzzle and $\boldsymbol{r}_{i,,j}$ ***= 1*** if square ***{i,j}*** is unlit in the initial configuration, ***0*** otherwise.

4.9 Equal Vision

A moment's reflection reveals the 'watchmen' problem to be the inverse of a queen placement problem on a four-by-four board. If we imagine a queen on each empty square and an empty square where each watchman is currently placed then each empty square is attacked by five queens. The problem is then to maximize the number of queens placed such that six and then seven queens attack each empty square. The empty squares in the solution to the revised puzzle will be the squares where the watchmen are to be placed in the original puzzle.

To solve the problem directly the binary decision variables, $\boldsymbol{x}_{i,,j}$, are defined for each square, ***{i,j}***, such that if square ***{i,j}*** is occupied by a watchman then $\boldsymbol{x}_{i,,j}$ ***= 0***, otherwise $\boldsymbol{x}_{i,,j}$ ***= 1.***

In order to maximize the number of occupied squares, the objective function will be minimized as follows:

$$Min \sum_{i=1}^{s} \sum_{j=1}^{s} x_{i,j}$$

The constraints necessary to ensure compliance with the conditions of the puzzle are:

$$\sum_{\substack{m=1, m \neq i, \\ 1 \leq m-i+j \leq s}}^{s} x_{m,m-i+j} + \sum_{\substack{m=1, m \neq i, \\ 1 \leq i+j-m \leq s}}^{s} x_{m,i+j-m} + \sum_{\substack{m=1 \\ m \neq i}}^{s} x_{m,j} + \sum_{\substack{m=1 \\ m \neq j}}^{s} x_{i,m} = n_{i,j}; \forall i,j$$

$$n_{i,j} \geq V - Mx_{i,j}; \forall i,j$$
$$n_{i,j} \leq V + Mx_{i,j}; \forall i,j$$

where ***V*** = number of vacant cells visible to watchmen, i.e. either six or seven.

4.10 The Abbott's Window

Constraints may be generated for each square using modifications of the bishop and rook moves. This renders the formulation exercise trivial but produces an unwieldy model involving many redundant constraints. A more efficient approach is to consider the grid on a line by line basis rather than by individual squares, similar to the approach taken for "The Crowded Board".

5. Conclusion

The IP formulations developed to solve these puzzles may be relatively straightforward to mathematically confident students but can be somewhat daunting to those who are less so. Such students need to be motivated to invest the necessary effort to become comfortable with these types of structure and notation. We feel that the puzzles may help to captivate the students' interest and hence generate this motivation. These skills, once acquired, may be transferable into the less whimsical world of OR.

Note: Complete models of the above problems and others using the XPRESS-MP modeling language may be found at Integer Programming in Recreational Mathematics.

References:

Adams, D., (1973), *Simulation Games. An Approach to Learning*, Wadsworth Publishing Co.

Anderson, D., S. Brown & P. Race, (1997), *Inspiring the inexperienced. 500 tips for further and continuing education lecturers,* Kogan Page

Cowan (1998), *On becoming an innovative university teacher,* SRHE & O.U. Press

De Vita, G., (1999), "Cross cultural work groups. A business and management perspective." *New Academic*, Vol. 8, 3. SEDA

Dudeney, H.E., (1917), *Amusements in Mathematics*, Thomas Nelson and Sons.

Foulds, L.R., & D.G. Johnson, (1984), "An Application of Graph Theory and Integer Programming:Chessboard Non-attacking Problems," *Mathematical Magazine*, Vol. 57, no. 2, pp. 95-104

Gage & Berliner, (1992), *Educational Psychology,* 5th ed., Houghton Mifflin Co.

Gardner, M., (1975), *Mathematical Carnival,* Penguin

Gardner, M., (1986), *Puzzles from Other Worlds*, Oxford University Press.

Kraitchik, M., (1942), *Mathematical Recreations,* W.W. Norton and Company, Inc.

Poniachik, J. & L., (1998), *Hard-to-Solve Brainteasers,* Sterling.

Quinn, G., (1996), http://indigo.ie/~gerryq/index.htm

Rawson, M., (1999), "Learning to learn: purely a skill?," *New Academic,* Vol. 8, 3. SEDA

Schuh, F., (1943), *Wonderlijke Problemen; Leerzam Tijdverdrijf Door Puzzle en Speel*, W.J. Thieme & Cie, Zutphen.

UK Department for Education (1999), The National Numeracy Strategy

Velucchi, M., (1995), "Non-Dominating Queens Problem," http://anduin.eldar.org/~problemi/papers.html

Walkin, L., (1990), *Teaching and learning in further and adult education,* Stanley Thomas Ltd.

Williams, H.P., (1999), *Model Building in Mathematical Programming,* 4th ed., Wiley