

Kraitchik's Queen Placement

Tony Hürlimann
tony.huerlimann@unifr.ch

22. Januar 2007

Kraitchik's Queen Placement (**chess4**)

How many Queens are needed, and in what position, so that every unoccupied square of the chessboard could be attacked directly by a Queen? (see [?] and [?]). **Problem**

Modeling Steps

We introduce three names for the rows/columns set: **h**, **i**, and **j**.

1. We introduce a binary variable **x** for every (i, j) -position within the chessboard, saying that $x[i, j]=1$ if a Queen is placed at (i, j) .
2. If a Queen is placed at position (i, j) , then all positions on the row i and the column j as well as the diagonals crossing at (i, j) are under attack by this Queen.
3. It is enough to require for any (i, j) -position, that at least one of the position under attack is occupied by a Queen.
4. At least means that the **OR**-connector can be used to collect all positions under attack.
5. That's exactly the constraint.
6. We want to have the minimal number of Queens.

The complete model code in LPL for this model is as follows:

```

MODEL KQUEENS "Kraitchik's Queen Placement";
SET h ALIAS i,j := /1:8/;
BINARY VARIABLE x{i,j} "Square (i,j) occupied?";
CONSTRAINT ATT{i,j}:
    OR{h|h<>i and h-i+j>=1 and h-i+j<=#h} x[h,h-i+j]
    OR OR{h|h<>i and i+j-h>=1 and i+j-h<=#h} x[h,i+j-h]
    OR OR{h|h<>i} x[h,j] OR OR{h|h<>j} x[i,h];
--C: SUM{j} x[4,j]=5;
MINIMIZE obj: SUM{i,j} x "Minimise number of Queens";
WRITE 'The minimal number of Queens is: %3d\n
      The chessboard is\n      %2s \n      %s %2s \n':
      obj, COL{j} j, ROW{i} (i,COL{j} IF(x,'*', '-'));
Draw.Ratio(50,50);
Draw.Rect(.95,.95,#i+1.05,#j+1.05,0);
FOR{i,j} DO Draw.Rect(i,j,i+1,j+1,if((i+j)%2,0,1)); END
FOR{i,j|x} DO Draw.Ellipse(j+.2,i+.2,j+.8,i+.8,6); END
END

```

At least 5 Queens must be placed to have all the positions attacked by one. **Solution**
 Their positions can be as follows (A star * means a Queen is placed, a dash
 - means no Queen is placed on this position):

```

      1 2 3 4 5 6 7 8
1  - - - - - - - -
2  - - - - - - - -
3  - - - - * - - -
4  - - - * - - - -
5  - - - * - * - -
6  - - - * - - - -
7  - - - - - - - -
8  - - - - - - - -

```

A graphical output is given by Figure ??.

Question 1 (*Answer to 1*)

1. *Can the Queens be placed in such a way that there is at most 1 Queen per row as well as at most 1 Queen per column?*
2. *Is it possible to have a solution with 5 Queens when two Queens are placed in row 1?*

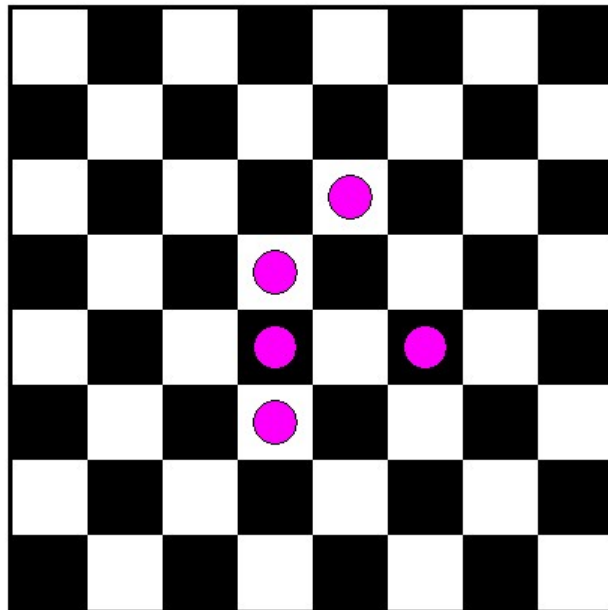


Abbildung 1: Solution of chess4

3. *Can we have all 5 Queens in one single row?*
4. *Could these 5 Queens be in another row than found in the previous answer?*

Answer 1 (*Question of 1*)

1. *Add the two constraints:*

$$A\{i\}: \text{SUM}\{j\} x \leq 1;$$

$$B\{j\}: \text{SUM}\{i\} x \leq 1;$$

The solution is still 5 Queens. And their placement is as follows:

	1	2	3	4	5	6	7	8
1	-	-	-	-	*	-	-	-
2	-	-	-	-	-	-	-	-
3	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	*
5	-	-	-	-	-	-	-	-

```

6 * - - - - -
7 - * - - - - -
8 - - * - - - -

```

2. Yes! Add the constraint:

```
C: SUM{j} x[1,j]=2;
```

This could be repeated for each row (not just for the first one). For each row it exists a solution. However, we need 6 Queens if we would liked to place 3 Queens in row 1 or 8. Verify!

3. Yes! a solution is in row 4: on columns (1,4,5,6,7), for example. We can generate this solution by imposing the additional constraint:

```
C: SUM{i,j|i>4} x=0; -- no Queen in one half
```

This constraint does not impose exactly what we want! The constraint forces all Queens to be in one half of the chessboard. Chances are high that they must be placed in row 4 to attack all cells in the other half. And we are lucky! But we are not sure in advance.

A proper formulation would be to impose 5 Queens on each row in turn. Solving it for the first 4 rows shows us that we could place 5 Queens in row 4.

4. No, except we accept at least 7 Queens. Again we could find this solution by imposing the constraint:

```
C: SUM{i,j|i>3} x=0; -- all Queens in the first 3 rows.
```