Mathematical Modeling of N-Queen Problem

Farzana Kousar[a,b,*], Farah Jabeen[c]

[a] *University of Engineering and Tehnology, Taxila, Pakistan*
[b] *COMSATS University Islamabad, Wah Campus, Pakistan*
[c] *COMSATS University Islamabad, Abbottabad Campus, Pakistan*

Abstract

N-Queen Problem is a traditional encounter of N Queens on $N \times N$ board where, position of every Queen should be such that they can not to be threatened by each other. It is extremely famed among scientists and researchers because of its numerous applications in current research areas of Artificial Intelligence and Optimizations. Here, we proposed a new mathematical model to solve $N-$Queen problem for any value of $N \geq 4$. To make the problem computationally efficient, the presented model reduces the number of diagonal computations from $(2N - 1)$ to $(2N - 3)$, while finding solution of N Queens. Simulation are carried out to find all possible solutions for given $N \geq 4$.

Keywords: N-Queen problem, Modelling, Diagonals, Constraints, Decision variable

1. Introduction

The N-Queen problem is very well famous problem in the field of Computer Science, Engineering and Mathematics. The N-Queen problem is a general form of N-Queen chess puzzle since 1848. It was first introduced by Max Bezzel and first solved by Carl Gauss in 1850 [1]. This problem has been research focus in the recent past for the researchers as it is employing the real world applications in optimization and Artificial Intelligence[2], VLSI testing, parallel memory storage schemes, deadlock prevention, traffic control[3], image

*Corresponding author
Email address: farzana.k7513@gmail.com (Farzana Kousar)

processing[4] and motion estimation[5]. N-Queen puzzle is considered a classical example in the field of computation for studying recursion, depth-first search and backtracking[6]. This problem may have N number of Queens with no defined formula for possible number of solution for that N. The 8-Queens puzzle has 92 distinct solutions[7],out of these 92 solution, some are fundamental while others are reflection of each other. The problem is about placing N-Queens on a board of dimension $N \times N$ such that no Queen can be attacked by any other Queen.The 8-Queen problem is shown in Figure 1. According to the rule set followed by chess, Queen is bound to move straight in a same row or column and can move diagonally. To avoid attacking a Queen by any other, there should be only one Queen in each row, column and diagonals. By following the rules one can end up with multiple solutions where some solutions are distinct, and some are unique. Section 2 discussed the different components needed to model the problem mathematically. Section 3 describe simulation results of N-Queen problem for $N \geq 4$. Section 4 concludes the research work.

2. Problem Statement and Modeling

For a given dimension $N \times N$, there are N-Queen to be placed. The placement of theses Queen should be such that no Queen can appear more than once in any same row, same column, or in a same diagonal. Here $N \times N$ board is represented by a matrix with N rows and N columns. Generally, while solving N-Queen problem, number of diagonals considered are $(2N-1)$. In our model we have reduce the number of diagonals from $(2N-1)$ to $(2N-3)$ needed to model the problem. We omit the corner square elements which were considered as diagonals in previous models [2]. Omitting computations of Corner square diagonals does not affect in possible placement of N-Queens, as they are already a part of major right and left diagonals, so number of diagonal, either left or right becomes $(2N-3)$ each as shown by Figure 2. However by omitting these corner diagonals reduce the diagonal computations during the implementation of model. This may lead to reduce computational complexity. To elaborate

N-Queen problem modeling, we used example set of 4×4 board which can be further generalized for any value of N.
To model N-Queen problem, we need to define inputs, $N \times N$ board with N Queen. Our model will define function which will best describe the objective of the problem that is to place N-Queens on $N \times N$ board while meeting the constraint (no Queen should be attacked by any other Queen). Variable of a function is called decision variable whose value is determined by mathematical model.

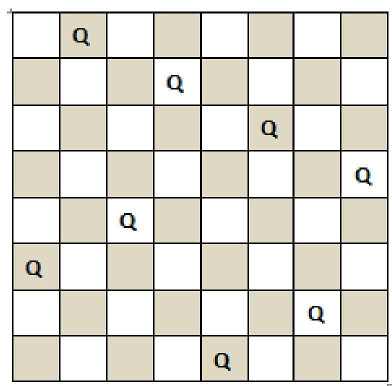

Figure 1: 8-Queen chess board

2.1. Inputs and Outputs

The $N \times N$ board can be represented by adjacency/ incident matrix of $N \times N$ dimensions where N is the number of rows and columns of the matrix. It also represents the number of Queens. Output is the solution of N-Queen , where Queens are placed such that No Queen can be overlooked by any other Queen in same row, column and diagonal.
number of left diagonals (LD)= number of Right Diagonals (RD)= $(2N-3)$.

2.2. Decision Variable

A decision variable is a quantity which need to be best determined to solve the problem. These are unknowns of a mathematical model of a problem, which decide that what value a variable should have, in order to find best possible solution of the problem. In our case, each square element of a $N \times N$ board represents decision variable for modeling of N-Queen problem. Decision variable is represented and defined as:

$$Y_{ij} = \begin{cases} 1, & \text{if Queen is placed} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Here Y_{ij} represents the entry of the $N \times N$ matrix at i^{th} row and j^{th} column. The Mathematical model determine that to which Y_{ij}, a Queen should be placed so that it can't be threatened by any other Queen.

2.3. Objective Function

The objective function is to maximize the number of non attacking Queens over the $N \times N$ board, satisfying all the constraints as:

$$f = max(\sum_{i=1}^{N}\sum_{j=1}^{N} Y_{ij}) \leq N \tag{2}$$

2.4. Constraints

Following are the constraints which are imposed on objective function represented by 'C' with constraint number.

C1: Only one Queen can be placed in any row to protect it from attacking by other Queen. This can be translated as:

$$\begin{aligned} Y_{11} + Y_{12} + Y_{13} + \cdots + Y_{1N} &= 1 \\ Y_{21} + Y_{22} + Y_{23} + \cdots + Y_{2N} &= 1 \\ &\vdots \\ Y_{N1} + Y_{N2} + Y_{N3} + \cdots + Y_{NN} &= 1 \end{aligned} \tag{3}$$

which can be written cumulatively as:

$$\sum_{i=1}^{N} Y_{ij} = 1 \quad \forall\ j = \{1, 2, \cdots, N\} \tag{4}$$

C2: Only one Queen can be placed in each column to protect it by attack of any other Queen as:

$$\begin{aligned} Y_{11} + Y_{21} + Y_{31} + \cdots + Y_{N1} &= 1 \\ Y_{12} + Y_{22} + Y_{32} + \cdots + Y_{N2} &= 1 \\ &\vdots \\ Y_{1N} + Y_{2N} + Y_{3N} + \cdots + Y_{NN} &= 1 \end{aligned} \tag{5}$$

which can be written cumulatively as:

$$\sum_{j=1}^{N} Y_{ij} = 1 \quad \forall\ i = \{1, 2, \cdots, N\} \tag{6}$$

C3: No two Queens in same diagonal: There should be only one Queen in each

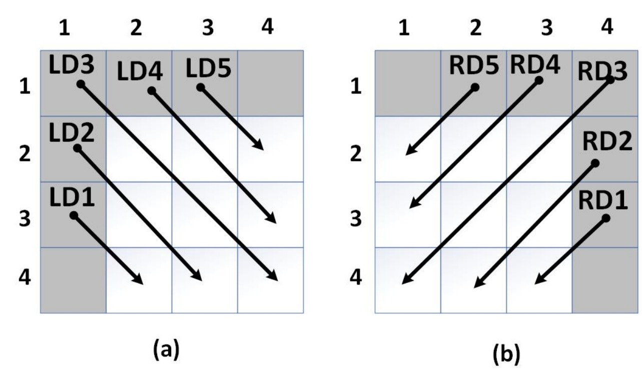

Figure 2: 4×4 *board* : right & left diagonals

diagonal to avoid attacking by any other Queen. For $N \times N$-Queen problem there are $(2N-3)$ diagonal (right and left each). Left diagonals are obtained when we move in left most column and topmost row of shaded right half T as shown by Figure-2(a). Tracing the right half T, moving from $(N-1)^{th}$ left bottom square through $(N-1)^{th}$ right top square. First element of each left diagonals belongs to either the first column or the first row. Similarly, for right diagonals we will move in left half shaded-T from $(N-1)^{th}$ right bottom square to $(N-1)^{th}$ left top square as shown in Figure-2(b).

Left Diagonals:. Following are $(2N-3)$ left diagonals:

$$\begin{aligned}
&\textbf{LD1:} Y_{31} + Y_{42} = 1 \\
&\textbf{LD2:} Y_{21} + Y_{32} + Y_{43} = 1 \\
&\textbf{LD3:} Y_{11} + Y_{22} + Y_{33} + Y_{44} = 1 \\
&\textbf{LD4:} Y_{12} + Y_{23} + Y_{34} = 1 \\
&\textbf{LD5:} Y_{13} + Y_{24} = 1
\end{aligned} \tag{7}$$

To model the left diagonal constraints, these diagonals are split into two parts as shown in Figure 3.

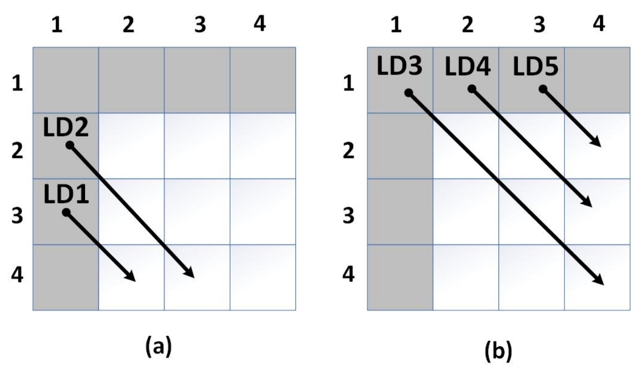

Figure 3: 4×4 *board* : split left diagonals

The element of diagonal LD1 and LD2 as shown in Figure 3(a), can be achieved by starting from i^{th}th row & 1^{st} column and decreasing row number & increasing column number while moving from left to right as shown by arrowhead. Linear equation of LD1 & LD2 is as:

$$Y_{i1} + \sum_{m=1}^{N-i} Y_{i+m,1+m} \leq 1 \quad \forall \;\; i = \{N-1, \cdots, 2\} \tag{8}$$

The diagonals LD3, LD4 and LD5 as shown in Figure 3(b), are obtained by starting from row 1^{st} & j^{th} column and then increasing column number & Row number while moving from left to right as:

$$Y_{1j} + \sum_{m=1}^{N-1} Y_{1+m,j+m} \;\leq 1 \quad \forall \;\; j = \{1, \cdots, N-1\} \tag{9}$$

Right diagonals:. Following are the $(2N-3)$ right diagonals.

$$\begin{aligned}
&\textbf{RD1:} Y_{34} + Y_{43} = 1 \\
&\textbf{RD2:} Y_{24} + Y_{33} + Y_{42} = 1 \\
&\textbf{RD3:} Y_{14} + Y_{23} + Y_{32} + Y_{41} = 1 \\
&\textbf{RD4:} Y_{13} + Y_{22} + Y_{31} = 1 \\
&\textbf{RD5:} Y_{12} + Y_{21} = 1
\end{aligned} \tag{10}$$

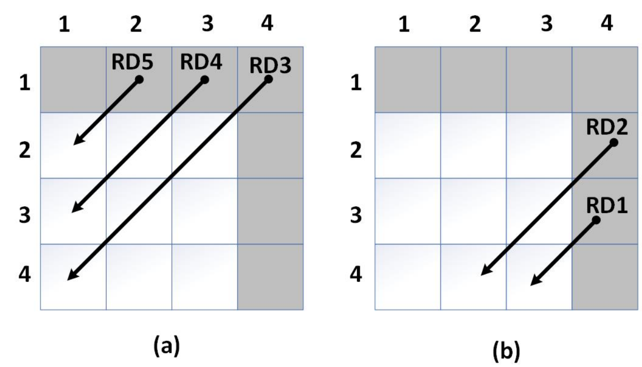

Figure 4: 4×4 *board* : split Right diagonals

To mathematical formulate right diagonal constraints, we split these diagonals in two parts as shown in Figure 4. The elements of RD1 & RD2 as shown in Figure 4(a), are obtain by starting from i^{th} row & N^{th} column and then decreasing row number & decreasing column number, while moving from right to left bottom as:

$$Y_{iN} + \sum_{m=1}^{N-i} Y_{i+m,N-m} \leq 1 \quad \forall \quad i = \{N-1, \cdots, 2\} \tag{11}$$

The diagonals RD3, RD4 & RD5 as shown in Figure 4(b), can be achieved by starting from 1^{st} row and j^{th} column and then increasing row number and decreasing column number while moving from right top to left bottom as follows:

$$Y_{1j} + \sum_{m=1}^{j-1} Y_{1+m,j-m} \leq 1 \quad \forall \quad j = \{N, \cdots, 2\} \tag{12}$$

2.5. N-Queen Mathematical Model

The mathematical model of N-Queen problem summed up as follows:

$$\textbf{max } f = (\sum_{i=1}^{N}\sum_{j=1}^{N} Y_{ij}) \leq N$$

s.t

C1:

$$\sum_{i=1}^{N} Y_{ij} = 1 \quad \forall\ j = \{1, 2, \cdots, N\}$$

C2:

$$\sum_{j=1}^{N} Y_{ij} = 1 \quad \forall\ i = \{1, 2, \cdots, N\}$$

C3:

$$\begin{aligned}
Y_{i1} + \sum_{m=1}^{N-i} Y_{i+m,1+m} \leq 1 \quad \forall \;\; i = \{N-1, \cdots, 2\} \\
Y_{1j} + \sum_{m=1}^{N-1} Y_{1+m,j+m} \leq 1 \quad \forall \;\; j = \{1, \cdots, N-1\} \\
Y_{iN} + \sum_{m=1}^{N-i} Y_{i+m,N-m} \leq 1 \quad \forall \;\; i = \{N-1, \cdots, 2\} \\
Y_{1j} + \sum_{m=1}^{j-1} Y_{1+m,j-m} \leq 1 \;\; \forall\ j = \{N, N-1, \cdots, 2\}
\end{aligned}$$

where

$$Y_{ij} = \{0, 1\} \quad \forall \quad i \;\; \& \;\; j$$

The above presented model works for $N \times N$ dimensional board for $N \geq 4$

3. Simulation & Results

The N-Queen Problem is simulated for $N \geq 4$. The simulation gives all possible solutions and their count after running some iterations. Figure 5 shows all possible solutions obtained for $N = 5$. Different solution matrices are generated in each iteration and at the end, the program counts similar matrices generated in different iterations to give count of all solutions. After some iterations the largest number is achieved, after which, no greater than that number is generated. As value of N is increasing, it takes more iterations and more time to find all possible solution count. The program's results are tested for some values of N by comparing the already known results[8].

$$\begin{matrix} 0&0&0&0&1 \\ 0&1&0&0&0 \\ 0&0&0&1&0 \\ 1&0&0&0&0 \\ 0&0&1&0&0 \end{matrix} \qquad \begin{matrix} 0&1&0&0&0 \\ 0&0&0&0&1 \\ 0&0&1&0&0 \\ 1&0&0&0&0 \\ 0&0&0&1&0 \end{matrix} \qquad \begin{matrix} 0&0&1&0&0 \\ 0&0&0&0&1 \\ 0&1&0&0&0 \\ 0&0&0&1&0 \\ 1&0&0&0&0 \end{matrix} \qquad \begin{matrix} 0&1&0&0&0 \\ 0&0&0&1&0 \\ 1&0&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&0&1 \end{matrix} \qquad \begin{matrix} 0&0&0&0&1 \\ 0&0&1&0&0 \\ 1&0&0&0&0 \\ 0&0&0&1&0 \\ 0&1&0&0&0 \end{matrix}$$

$$\begin{matrix} 0&0&0&1&0 \\ 0&1&0&0&0 \\ 0&0&0&0&1 \\ 0&0&1&0&0 \\ 1&0&0&0&0 \end{matrix} \qquad \begin{matrix} 0&0&1&0&0 \\ 1&0&0&0&0 \\ 0&0&0&1&0 \\ 0&1&0&0&0 \\ 0&0&0&0&1 \end{matrix} \qquad \begin{matrix} 1&0&0&0&0 \\ 0&0&0&1&0 \\ 0&1&0&0&0 \\ 0&0&0&0&1 \\ 0&0&1&0&0 \end{matrix} \qquad \begin{matrix} 1&0&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&0&1 \\ 0&1&0&0&0 \\ 0&0&0&1&0 \end{matrix} \qquad \begin{matrix} 0&0&0&1&0 \\ 1&0&0&0&0 \\ 0&0&1&0&0 \\ 0&0&0&0&1 \\ 0&1&0&0&0 \end{matrix}$$

Figure 5: 5×5 *board* : All possible solution

4. Conclusion and future work

This work presents new mathematical model of $N-$Queen problem. The presented model reduces the number of diagonal computations from $(2N - 1)$ to $(2N - 3)$ while placing N-Queen to avoid attacking by any other Queen. In this model, we omit the corner square diagonals as compared to other models[2]. Omitting computations of Corner square diagonals cause no affect in possible placement of N-Queens, as they are already a part of major right and left Diagonals. The future work is to compare the computational complexity with other models already presented in the literature.

References

[1] F. Becker, Forgetten books.

[2] C. L. C. Letavec, J. Ruggiero, J. Ruggiero, The n-queens problem, INFORMS Transactions on Education 2 (2002) 64–103. doi:10.1287/ited.2.3.101.

[3] M. M. T. C. Erbas, Z. Aliyazicioglu, Linear congruence equations for the solutions of the n-queens problem, Information Processing Letters 41 (6) (1992) 301–306. doi:10.1016/0020-0190(92)90156-P.

[4] J. H. G. Goos, J. Leeuwen, Lecture notes in computer science, Lecture Notes in Computer Science 1716 (1999) 437–445. doi:10.1108/ir.1999.04926fae.00.

[5] C.-M. L. Chung-Neng Wang, Shin-Wei Yang, T. Chiang, A hierarchical decimation lattice based on n-queen with an application for motion estimation, IEEE Signal Processing Letters 10 (2003) 228–231. doi:10.1109/LSP.2003.814403.

[6] C. Clay, A new solution to the n ¡= 8 queens problem, ACM SIGPLAN Notices 21 (1986) 28–30. doi:10.1145/382278.382391.

[7] S. Jain, Mathematical approach for n-queens problem with isomorphism, 2017.

[8] J. Bell, B. Stevens, A survey of known results and research areas for n-queens, Discrete Mathematics 309 (1) (2009) 1–31.