A new approach to solve n-queens problem based on series

Vishal Kesri, Manoj Kumar Mishra
School of Computer Engineering KIIT University, India

Abstract
The N-queens problem is a popular classic puzzle where numbers of queen were to be placed on an n x n matrix such that no queen can attack any other queen. The Branching Factor grows in a roughly linear way, which is an important consideration for the researchers. However, many researchers have cited the issues with help of artificial intelligence search patterns say DFS, BFS and backtracking algorithms. The proposed algorithm is able to compute one unique solution in polynomial time when chess board size is greater than 7. This algorithm is based on 8 different series. For each series a different approach is taken to place the queen on a given chess board.

General Terms: N-Queen, Knight move, NP-hard.
Keywords: Analyzer, Move, Rule, $Part_z$.

I. INTRODUCTION

The n-queens problem is proposed for the first time in 1850 by Carl Gauss. It is to determine a placement of n queens on an n × n chessboard, such that no two queens can attack each other [3]. This problem falls in a special class of problems well known as NP hard, whose solution cannot be found out in polynomial time. Let's consider the 8-queen problem, which is computationally very expensive since the total number of possible arrangements queen is 64! / (56! x 8!) ~ 4.4 x 10^9 and the total number of possible solutions are 92 [1]. But there exists only 12 unique solutions. There are some solutions that are to be the same and can be obtained one from the other by taking rotations or symmetry. The n-queen problem follows the same rules as in 8-queen problem with N queens and an n x n chessboard. Current knowledge of total number of solutions of n-queens problem can be viewed from the Table 1.

A. Applications

The N-queen problem can be applied in many different areas, like parallel memory storage schemes, VLSI testing, traffic control and deadlock prevention. It is also applicable to find solutions of those problems which require permutations like travelling salesman problem [1].

N:	Unique Solution	Distinct Solution
1	1	1
2	0	0
3	0	0
4	1	2
5	2	10
6	1	4
7	6	40
8	12	92
9	46	352
10	92	724
11	341	2680
12	1787	14200
13	9233	73712
14	45752	365596

Table 1: The number of total and unique solutions for the N-Queens problem

II. N-QUEENS PROBLEM DEFINATION

The n-queens Problem of size n, where need to place n queen into n × n chessboard, so that no two queen can attack each other that is no two of them are on the same row, column, or diagonal. Therefore the rules to obtain the solution of n-queens problem are follows [5]:

1. Only one queen can placed in each row
2. Only one queen can placed in each column
3. At any time maximum one queen can be in the diagonal.

Suppose two queen are placed at position (i, j) and (k, l). Then they are on the same diagonal only if

$$i - j = k - l \quad or \quad i + j = k + l$$

A. Notation

Here some notations are defined, which are used in the propose algorithm .These notations are defined in table 2.

Symbols	Definition
S_1	Positive Integer Series-A_1
S_2	Positive Integer Series-A_2
S_3	Positive Integer Series-A_3
S_4	Positive Integer Series-A_4

S_5	Positive Integer Series-A_5
S_6	Positive Integer Series-A_6
S_7	Positive Integer Series-A_7
S_8	Positive Integer Series-A_8
R_1	Rule-1, used to get the solution.
R_2	Rule-2, used to get the solution.
MV_1	It is first move function, which defines the pattern of placing the queens.
MV_2	It is second move function, which defines the pattern of placing the queens.
MV_3	It is third move function, which defines the pattern of placing the queens.
'n'	Given number of queen (Input Size)
'C_{ij}'	Cell position of the chess board.

Table-2: List of Symbols

III. SERIES

The Proposed algorithm is a combination of two rules and eight distinct series. These series are given below:

A. *Series*- A_1

The Series- A_1 is used to get the solution such that:

$$f(x) = 6x \quad | x > 1 \qquad (1)$$
$$S_1 = \{12, 18, 24, 30, \dots \dots \dots, \infty\}$$

B. *Series*- A_2

The Series- A_2 is used to get the solution such that:

$$f(x) = 6x + 1 \quad | x > 1 \qquad (2)$$
$$S_2 = \{13, 19, 25, 31, \dots \dots \dots, \infty\}$$

C. *Series*- A_3

The Series- A_3 is used to get the solution such that:

$$f(x) = 6x + 4 \quad | x \geq 1 \qquad (3)$$
$$S_3 = \{10, 16, 22, 28, \dots \dots \dots, \infty\}$$

D. *Series*- A_4

The Series- A_4 is used to get the solution such that:

$$f(x) = 6x + 5 \quad | x \geq 1 \qquad (4)$$
$$S_4 = \{11, 17, 23, 29, \dots \dots \dots, \infty\}$$

E. *Series*- A_5

The Series- A_6 is used to get the solution such that:

$$f(x) = 12x + 8 \quad | x \geq 0 \qquad (5)$$
$$S_5 = \{8, 20, 32, 44, \dots \dots \dots, \infty\}$$

F. *Series*- A_6

The Series- A_6 is used to get the solution such that:

$$f(x) = 12x + 9 \quad | x \geq 0 \qquad (6)$$
$$S_6 = \{9, 21, 33, 45, \dots \dots \dots, \infty\}$$

G. *Series*- A_7

The Series- A_7 is used to get the solution such that:

$$f(x) = 12x + 14 \quad | x \geq 0 \qquad (7)$$
$$S_7 = \{14, 26, 38, 40, \dots \dots \dots, \infty\}$$

H. *Series*- A_8

The Series- A_8 is used to get the solution such that:

$$f(x) = 12x + 15 \quad | x \geq 0 \qquad (8)$$
$$S_8 = \{15, 27, 39, 51, \dots \dots \dots, \infty\}$$

IV. RULES

There are two Rules, which is used with a particular series to achieve a solution for a given chess board. Some series do not need any of the rules, such as Series-A.

Analyzer: The core of these rules is an analyzer function. The function of the analyzer is described as below.

Step-1: initialize the cell 'C_{ij}' of chess board to zero, except where, 'C_{ij}' already acquired by a queen.

0	Q	0	0	0
0	0	0	0	Q
0	0	0	0	0
0	0	0	0	0
0	0	Q	0	0

Figure-1: Put Zero in remaining 'C_{ij}'

Step-2: In the beginning take any queen and add '1' to the value of its attacking cells. Then this step is repeated for each queen as shown in figure 2.

1	Q	3	2	2
2	2	3	2	Q
2	3	3	3	2
1	2	2	2	3
1	2	Q	1	2

Figure-2: The value of each 'C_{ij}' after step 2

The 'C_{ij}' values shown in the figure 2 indicates that the position C_{ij} is attacked by that many queens e.g. 'C_{23}' has an integer value 3, that means this cell is attacked by 3 different queens e.g. by C_{12} ,C_{25} ,C_{53}.

First Rule: First rule is applied to find the solution lies in the series S_2 from the solution of S_1. Algorithm of First Rule is described below:

Algorithm for "Rule"

Step-1: If the solution falls in any one of the series (S_1, S_3)

Step-2: Increase the dimension of the chess board by one, that is row +1 and column +1

Step-4: Call Analyzer function with this updated dimensions

Step-5: Place a queen, where "C_{ij}= =0"

Algorithm-1: Describe Steps of first Rule

Second Rule: Second rule is applied to find the solution lies in the series S_6 from the solution of S_5. The conditions for the second rule is "remove the queen from C_{ij} If C_{in} = = C_{1j} = = 1 and $|i\text{-}1| \neq |n\text{-}j|$. And place queens at the positions C_{1j} and C_{in}. Algorithm of second Rule is described below:

Algorithm of Second Rule

Step-1: If the solution falls in any one of the series (S_5)

Step-2: Increase the dimension of the chess board by one, that is row +1 and column +1

Step-3: Call Analyzer function with this updated dimensions

Step-4: Remove Queen from C_{ij}

Step-5: Call Analyzer function with this updated dimensions
Step-6: Place queen, where "C_{ij}= =0"

Algorithm-2: Describe Steps of second Rule

Table 3 show which series those rules apply:

	S_1	**S_2**	**S_3**	**S_4**	**S_5**	**S_6**	**S_7**	**S_8**
R_1	✕	✓	✕	✓	✕	✕	✕	✕
R_2	✕	✕	✕	✕	✕	✓	✕	✕

Table-3

V. Move Function

MV function shows us the pattern of placing queen on chess board. There is three Move Functions defined namely MV_1, MV_2 and MV_3. The move function along with a particular series defines the placement of queens on the chess board. The algorithm for placing queen according to pattern is described below:

Move-1: This function increases the row value of the chess board by 3 and reduces the column value by 1. Algorithm-3 for MV_1 is given below.

```
Algorithm for "MV1"

 /* The Place of first queen depend on series */
/* i = row value */

Cij← Place First Queen
for←a=1 to a ≤ n
      i = i+3;
      j = j-1;
      Cij← Place Queen
      a = i;
```

Algorithm-3: Algorithm for MV_1 function

Move-2: This function increases the row value of the chess board by 2 and reduces the column value by 1. Algorithm-4 for MV_2 is given below.

```
Algorithm for "MV2"

 /* The Place of first queen depend on series */
/* i = row value */

Cij← Place First Queen
for←a=1 to a ≤ n
      i = i+2;
      j = j-1;
      Cij← Place Queen
      a = i;
```

Algorithm-4: Algorithm for MV_2 function

Move-3: This function decreases the row value of the chess board by 2 and increase column value by 3 or decrease column value by 1. Algorithm-5 for MV_3 is given below.

```
Algorithm for "MV3"

/* The Place of first queen depend on series */
/* i = row value, j = Column */
count = 0;
Cij← Place First Queen
for←a=n to a ≥ 1
      i = i-2;
      if (count % 2 =0 )
            j = j+3;
      else
            j = j-1;
      Cij← Place Queen
      a = i;
```

Algorithm-5: Algorithm for MV_3 function

Table-4 shows, which move function is applied on what series.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
MV_1	✓	✗	✓	✗	✗	✗	✗	✗
MV_2	✗	✗	✗	✗	✓	✓	✓	✓
MV_3	✗	✗	✗	✗	✓	✓	✓	✓

Table-4

Algorithm for n-queens

The proposed algorithm finds a unique solution of n-queens problem, which is described below:

Algorithm to solve n-queens Problem

Step1: $n \leftarrow Number\ of\ queen\ need\ to\ place$

Step2: If $(n \in S_1)$
{
Call "MV_1", where i = 1 && j = n/3
Call "MV_1", where i = 2 && j = (n/3) *2
Call "MV_1", where i = 3 && j = n
}
Step3: If $(n \in S_2)$
{
Get the solution of "n-1" chess board
/* n-1 chess board $\in S_1$ */
Apply 'R_1'
}
Step4: If $(n \in S_3)$
{
Call "MV_1", where i = 2 && j = (n-1)/3
Call "MV_1", where i = 1 && j = (n-1/3) *2+1
Call "MV_1", where i = 3 && j = n
}
Step5: If $(n \in S_4)$
{
Get the solution of "n-1" chess board
/* n-1 chess board $\in S_3$ */
Apply 'R_1'
}
Step6: If $(n \in S_5)$
{
Call "MV_2", where i = 1 && j = n/2
Call "MV_3", where i = n && j = (n/2) + 2
}
Step7: If $(n \in S_6)$
{
Call "MV_2", where i = 2 && j = (n-1)/2
Call "MV_3", where i = n && j = (n-1)/2 +2
Apply 'R_2'
}
Step8: If $(n \in S_7)$
{
Call "MV_2", where i = 1 && j = (n/2) + 1
Call "MV_3", where i = n && j = (n/2) + 3
Place Queen At C_{ij}, Where i = 2 && j = 1
}
Step9: If $(n \in S_8)$
{
Call "MV_2", where i = 1 && j = (n-1) /2+2
Call "MV_3", where i = n && j = (n-1) /2) + 4
Place Queen At C_{ij}, Where i = 1 && j = 1
Place Queen At C_{ij}, Where i = 3 && j = 2
}

Algorithm-6: Algorithm for Calculate Solution of N-Queen

VI. DISADVANTAGE

The proposed algorithm provides only one unique solution when chess board size is greater than 7 and the algorithm is not able to provide all the solutions. By Rotation of chess board we can get few more solutions but those solutions are not distinct.

VII. CONCLUSION AND FUTURE WORK

This Algorithm provides only one unique solution for n > 7. This method can be enhanced in future to find all other solutions.

REFERENCES

[1] V. Kesri, Va. Kesri and P. Ku. Pattnaik. "An Unique Solution for N queen Problem". International Journal of Computer Applications.Vol:43 Issue:12 pp: 1-6, 2012.

[2] Gutiérrez-Naranjo, Miguel A., Martínez – del - Amor Miguel A. , Pérez-Hurtado Ignacio, and Pérez-Jiménez Mario J. February 02, 2009. "Solving the N-Queens Puzzle with P Systems", 7th Brainstorming Week on Membrane Computing, volume 1, pp199-210.

[3] Letavec, C. and Ruggiero, J. 2002. "The n-queen problem", Informs Transaction on Education, volume 2 number 3, May 2002.

[4] Uninformed Search Analysis of the N-queens Problem, Ben Sauerwine, September 20, 2003. Available at: http://www.andrew.cmu.edu/user/bsauerwi/research/P1.pdf

[5] Russell, S. and Norvig, P. "Artificial Intelligence A Modern Approach", Second Edition, pp 139-140.

[6] Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, "Fundamental of computer Algorithm (pp 339)".

[7] S.Ahmadi V. Kesri, and Va. Kesri. "Clustering in backtracking for solution of N-queen Problem". Third International Conference on Contemporary Issues in Computer and Information Sciences CICIS 2012