Development of Algorithms for Ranking and Un-ranking Function of N-Queens Problem

---

[1]Khushboo Srivastava, [2]Surya Mohan, [3]Bhaneshwar Singh, [4]Ashwamedh Maurya, [5]Saurabh Srivastava

[1/5]Assistant Professor, [2]Senior Assistant, [3]Student, [4]Professor

[1/5/4]Ashoka Institute of technology & Management Pahariya Sarnath, [2]Banaras Hindu University, [3]Mahatma Gandhi Kashi Vidyapeeth, Varanasi

[1]khushboosrivastava5555@gmail.com, [2]suryamohan2926@gmail.com, [3]bhaneshwarsingh@gmail.com, [4]mashwamedh@gmail.com, [5]vijaysaurabhsrivastava024@gmail.com

https://doi.org/10.64882/ijrt.v14.iS1.1071

Abstract

The n-queens Problem, introduced in 1850 by Carl Gauss, the problem requires us to find the placement of N queens on an NXN chessboard such that no queen is attacking the other. This paper mainly composed of following problems: 1. To evaluate all the solutions for given number of queens. 2. Ranking Function: For given solution, to find rank or index of that solution in set of solution space. 3. Un-ranking Function: For given index find that solution in set of solutions.

We have covered these problems in this, we have conducted an extensive study on these problems and realised that no simple formula is there to solve these problems.

I. There are many algorithms to generate all the solutions of n-queens problem, but in this paper we have considered backtracking.

II. Ranking function: Ranking function states that to find index of solution in set of all solution.

III. Un-ranking function: un-ranking function states that to find particular solution for particular index.

I will discuss definition of n-queens problem, its relationship with CSP and history, various method of solving n-queens problem, normal method to find ranking and un-ranking function, at last we will discuss new concept of ranking and un-ranking function respectively.

Keywords- N-queens Problem, Ranking Function-ranking Function, Constraint Satisfaction Problem, Euler Project 24.

Introduction

The n-queens problem is a generalized form of 8-queens problem, proposed by the chess player Max Bezel. In 8- queen problem, 8 queens are required to be placed on an 8x8 chess board in such a way that no queen attacks any other queen. A queen can move in horizontal (in the same row), vertical (in the same column) and diagonal direction. Also, an n-queens problem must follow the following rules:

1. There is at most one queen in each column.
2. There is at most one queen in each row.

3. There is at most one queen in each diagonal.

The 8-queens problem is computationally very expensive since the total numbers of possible arrangements of queen are $64! /(56! \times 8!) \sim 4.4 \times 10^9$ and the total number of possible solutions are 92.

Literature review

N-Queens and CSP

We can represent the n-queens problems as a constraint satisfaction problem. A Constraint Satisfaction Problem consists of 3 components: 1. A set of variables. 2. A set of values for each of the variables. 3. A set of constraints between various collections of variables.

In n-queens

No queen can attack any other queen.

Given any two queens Q_i and Q_j they cannot attack each other.

Now we translate each of these individual conditions into a separate constraint.

Q_i cannot attack Q_j $(i \neq j)$

- Q_i is a queen to be placed in column i, Q_j is a queen to be placed in column j.
- The value of Q_i and Q_j are the rows the queens are to be placed in.

Queens can attack each other

- Vertically, if they are in the same column, this is impossible as Q_i and Q_j are placed in different columns.
- Horizontally, if they are in the same row, we need the constraint $Q_i \neq Q_j$.
- Along a diagonal, they cannot be the same number of columns apart as they are rows apart, we need the constraint $|i-j| \neq |Q_i-Q_j|$

A solution to the n-queens problem will be any assignment of values to the variables $Q_1,\ldots,Q_n$ that satisfies all of the constraints

A Brief History

In 1848 chess player Max Bezel proposed the 8-queens problem and since then many mathematicians including Gauss, have worked on this problem and its generalized form i.e. n-queens problem. The first solution to the 8-queens problem was found out by Franz Nauck in 1850. Nauck later extended this problem to n-queens (placing n queens on an n×n chessboard such that no queen attacks any other queen using the standard moves). In 1874 S. Gunther proposed a method of finding solutions by using determinants and J.W.L. Glaisher refined this approach. Alternatively, search-based algorithms have been developed. For example, a backtracking search, an algorithm generates all possible solution of a given n×n board (Bitner and Reingold, 1975 and Purdom and Brown, 1983).

Application of N-Queens Problem

The n-queens problem is really a puzzle but, surprisingly, there are some practical applications such as parallel memory storage schemes, VLSI testing, traffic control, and deadlock prevention.

Various Methods of Solving N-Queens Problem

Currently various methods are available to solve it as A Novel Method for Solving N-Queens Problem, [6], Solution of N-Queens Problem Using Genetic Algorithms, Minimal Conflict Algorithms, Solving N-Queen Problem by Backtracking

Methodology

Process for Generating Solutions

Input- Number of queens.

Output- All possible solutions.

Take input as no of queens;

If no of queens is either 0 or 1 or 3

Return no solutions;

Else

Solutions exist;

If Solution Exist

Take first row

Check each column,

If any column of this row is safe

Mark it as solution and go to next row

Continue this process till next safe place,

If there is no safe place in this row

Come back to previous row and check next column of this row

Continue this process until no of marks position is equal to number of queens and print this solution,

This is the first solution, for next solution continue this process from position of next queen.

Process for Finding Solution for Given Index

Store all possible solution in a character array;

Take input index of solution

Go to proper place of array

Print N×N elements of character array

This is required solution

Example**:** Find 6^{th} solution of 5 queens problem.

First store all possible solution of 5 queens problems in array b[ ]

1	3	5	2	4	1	4	2	5	3	2	4	1	3	5	2	5	3	1	4	3	1	4	2	5

3	5	2	4	1	4	1	3	5	2	4	2	5	3	1	5	2	4	1	3	5	3	1	4	2

Now count 6^{th} block of b[ ] which is (3,5,2,4,1) is required solution.

Process for Finding Index for Given Solution

Take input solution in a character array a[ ];

Compare N character of already stored solution and input solution;
If N×N elements are all equal
This is required result
If any one of the N×N elements do not match this means this is not a required solution
Go to next N×N elements comparison.
Example**:** Find rank of (3,5,2,4,1) in set of 5 queens solutions.
First, we will take array a[5]

3	5	2	4	1

Now we will store all solution of 5 queens problem in array b[50];

1	3	5	2	4	1	4	2	5	3	2	4	1	3	5	2	5	3	1	4	3	1	4	2	5

us

3	5	2	4	1	4	1	3	5	2	4	2	5	3	1	5	2	4	1	3	5	3	1	4	2

re

Let a map $f:S \rightarrow N$ defined by

$f(n,S_i) = T-t+s$

were

n = Number of queens. S_i =any solution of N queens problem. T=Rank of S_i in the set of lexicographic permutation of 1,2,3......n. t=number of smaller permutation than S_i. s=Number of solutions in smaller permutation.

There are two rules to check for solution:

Rule 1:

If difference between any two consecutive numbers is 1 then that permutation cannot be solution otherwise may be solution.

Example**-** check (2, 1, 4, 3) can be solution or not. Here difference between first number 2 and second number 1 and absolute difference |2-1|=1, this shows that (2,1,4,3) cannot be a solution.

Rule 2

If difference between element and difference between corresponding indexes is same then that permutation is also not solution otherwise it is solution.

For above example

Possible solution is......3 1 5 2 4
Index is.......................1 2 3 4 5
Difference is .... |3-5|=2
|1-3|=2

Hence (3,1,5,2,4) cannot be solution.

Take another example check (3,1,4,2) is solution or not

Solution is 3 1 4 2

Index is 1 2 3 4

Differences:

$|3-1| \neq |1-2|$ $|3-4| \neq |1-3|$ $|3-2| \neq |1-4|$

$|1-4| \neq |2-3|$ $|1-2| \neq |2-4|$

$|4-2| \neq |3-4|$

From the above observation we can see those differences between numbers and their corresponding indices is not equal, this shows that this is one solution.

Rank of Permutation in Lexicographic Order

There is n! Permutation of (1,2,3.........n), if we generate this permutation in lexicographic order by Euler Project 24, then we can assign a number to each of them.

For example, 24 permutations of (1,2,3,4) in which (1,2,3,4) ranked as 1, (1,2,4,3) ranked as 2, (1,3,2,4) ranked as 3 and so on, at last (4,3,2,1) ranked as 24.

Now our aim is to find rank of any permutation. Suppose we have to find rank of (3,1,4,2).

3 1 4 2

Now we find total number of smaller numbers for each digit here for 3→2, 1→0, 4→1, 2→0

2 0 1 0

3! 2! 1! 0!

Sum= 2X3! + 0X2! + 1X1! + 0X0!

= 12+0+1+0 = 13

Rank = 13+1 = 14

i.e. (3,1,4,2) will come at 14^{th} position.

Un-Ranking Function

Means if rank of solution is given, then we have to find that particular solution. For example, 8 queens problem has total 92 solutions, suppose you want to find 10^{th} solution, then a function which find 10^{th} solution among those 92 solutions is called un-ranking function. Suppose n is number of queens, r is rank for which solution is to be find then a function f

f :S→N

Where S is the set of number of all solution, 1,2 .... ,92 in case of 8 queens.

N is set of natural number from 1 to factorial of n. n is fixed.

Function f is defined by

f(n,r)=i

Means that r^{th} solution is i^{th} permutation of (1,2,....n). Where i is defined as

$$i = \begin{cases} \text{Checked,} & \text{if count} = r \\ \text{Not exist,} & \text{checked} > n! \end{cases}$$

Where count = count permutation which is solution.

Checked = all checked permutation until count=r;

Result & Discussion

Number of queens	Total number of solutions	Number of queens	Total number of solutions
1	1	14	365596
2	0	15	2279184
3	0	16	14772512
4	2	17	95815104
5	10	18	666090624
6	4	19	4968057848
7	40	20	39029188884
8	92	21	314666222712
9	352	22	7016442691008
10	724	23	24233937684440
11	2680	24	227514171973736
12	14200	25	2207893435808350
13	73712		

When we implement the ranking and un-ranking function in real environment, we find desired result as for 8 queens problem, for solution (7,2,6,3,1,4,8,5) rank is 79. In similar way for 10 queens problem, 100th solution is (2,8,10,4,1,5,9,6,3,7) & for 15 queens problem, 100th solution is (1,3,5,12,9,11,14,7,15,13,2,8,6,4,10).

As increasing number of queens, the storage of computer overflows is main constraints for finding solutions. It may overcome as the computational power increases day by day.

Conclusion

N-queens problem is very interesting program among computer programmer as well as mathematician; there are several methods to solve this problem like Novel method, Genetic Algorithms, Ant Colony Optimization, backtracking etc. But there is no method for ranking and un-ranking function; in this project my work is to find a way to find ranking and un-ranking function. Since to develop these functions I need all solutions of n-queens problem in lexicographic order, so I have used Backtracking method to solve n-queens. To generate permutation of (1,2,3,4......n) I have used a new technique which is dependent on project Euler 24 which is very efficient to find i^{th} permutation directly among all n! Permutations. Also, I have found two rules by applying which we can easily find that i^{th} permutation is solution of n-queens problem or not. In old method it is used very simple concept in which all solutions of n-queens problem are stored in an array and we can reach at desired place by counting all block, one block means one solution, but in new method I have used lexicographic order of permutation which is easy to understand and easy to implement. Finally, I have written two functions ranking and un-ranking function respectively.

References

1. 3rd N QUEENS ETSI PlugtestsTM CONTEST Counting the number of solutions Single and Distributed Program. http://nguyendangbinh.org/Proceedings/IPCV08/Papers/GEM4444.pdf

2. Hynek, J. Genetic Algorithms for the N-Queens Problem. Available at:
3. International Journal of Advanced Research in Computer Science and Software Engineering- A Novel Method for Solving N-Queens Problem by Avani Gupta, S.Ravi Rohith and Satya Pramodh Mazumdar SCSE, VIT University, India.
4. International Journal of Advanced Research in Computer Science and Software Engineering Solving N Queen Problem Using Various Algorithms – A Survey S. Pothumani Department of CSE, Bharath University,
5. International Journal of Computer Applications (0975 – 8887) Volume 43– No.12, a Unique Solution for N queen Problem April 2012
6. Lexicographic generation of ordered tree by S. ZAKS, Department of Computer Science University of Illinois Urbana IL 61801 U.S.A.
7. Modified Genetic Algorithm for Solving n-Queens Problem -Jalaleddin Aghazadeh heris Faculty of Mathematics and Computer Science Allameh Tabataba University Tehran, Iran j.aghazadeh@st.atu.ac.ir Mohammadreza Asgari Oskoei Faculty of Mathematics and Computer Science Allameh Tabataba University Tehran, Iran oskoei@atu.ac.ir.
8. Solution of n-Queen Problem Using ACO-Salabat Khan, Mohsin Bilal, M. Sharif, Malik Sajid, Rauf BaigNational University Of Computer and Emerging Science Islamabad, Pakistan Email: salabat.khan@nu.edu.pk Telephone: (+92)51-4532308
9. Vishal Kesri, Prasant Ku. Pattnaik School of Computer Engineering KIIT University, India Vaibhav Kesri Department of Electrical Engineering NIT Kurukshetra, India.