Solving N-Queen Problem by Genetic Algorithm using Novel Mutation Operator

View the article online for updates and enhancements.

Solving N-Queen Problem by Genetic Algorithm using Novel Mutation Operator

[1]Mr. Sachin Sharma, [2]Vinod Jain
[1,2]Assistant Professor,
GLA University, Mathura-281406 (U.P.). India
Email id: [1] sachin.sharma@gla.ac.in, [2]vinod.jain@gla.ac.in

Abstract

N-Queen is a well-known NP-Hard problem. In N-Queen problem, the challenge is to place n queens on n*n chess board such that no two queens can attack each other. The problem can-not be solved using traditional algorithms. Genetic algorithm is a well-known optimization technique. The problem has to be represented in genetic form to solve it using genetic algorithm. In this paper a new mutation operator is proposed to solve N-Queen problem. The proposed algorithm is applied on some instances of N-Queen problem and it is observed that the proposed performs better as compared to other existing algorithms.

Keywords: N-Queen problem, Genetic Algorithm, Mutation Operator

1. Introduction

The N-Queen problem came into the existence after the generalization of 8-Queen Problem. This problem was introduced by a German chess player; Max Bazzel in the year of 1848. The 8-Queen problem objective was to place the 8 queens in an 8 X 8 chessboard in such a manner that neither of the queen out of eight, could attack each other. Figure 1 is showing a 4-Queen problem.
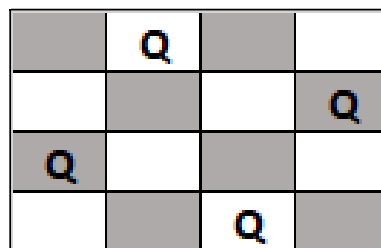

Fig 1. Four Queen Problem

As we aware in a chessboard game, a queen can attack in row, column, diagonal and anti-diagonal directions. The N- queen problem can be represented as a generalized form of 8- Queen Problem.

The N-Queen problem comes in the category of NP- Hard (Non-deterministic polynomial time hard). It is basically a permutation problem. The N-queen problem is a classical way to demonstrate many algorithms like backtracking, Divide & conquer and neural network etc.

The first person who claimed to solve 8-queen problem was Gauss (1850). After some time, it was confirmed that only 72 solutions found by Gauss out of 92 solutions. F.Nau (1850),was the first person who found all 92 solutions. However, many authors proposed various methods & algorithms that claim to solve N-queen problem but most of them correctness are not examined. Just because of hardness and complexity of N-queen problem, researchers used the techniques like heuristic algorithm such as Genetic Algorithm to solve the N-queen Problem. John Holland was the person who firstly

introduced the concept of Genetic Algorithm in 1975.The Genetic algorithm can be described as a cluster of computational models having the ability to employ the processes which are found in natural biological evolution. This algorithm plays a very important role to identify the best choice for practical applications like scheduling, Optimizations, computer aided designing etc. The genetic algorithm need not required any kind of supervisor or observer. An adaptive approach is to be used by Genetic algorithm to search and solve optimization problem. This algorithm is capable to search large non-linear solution space which would be difficult by the knowledge of an expert.

Figure 2 is showing a flowchart of genetic algorithm. The genetic algorithm starts form initial population generation and then perform selection, reproduction and mutation operators to solve optimization problems.
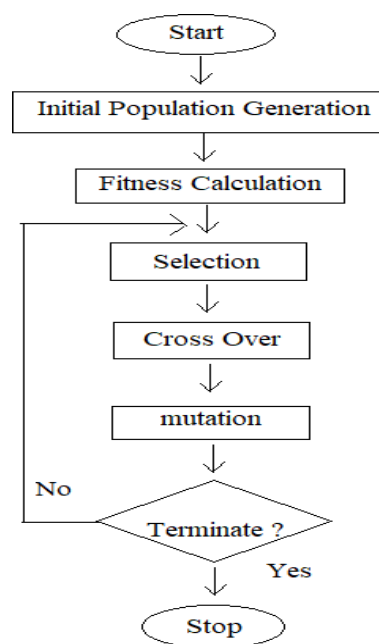

Fig 2: Flowchart of Genetic Algorithm

The next section illustrates the current state of art research done in this area.

2. Literature Survey

Many researchers proposed various algorithms to solve N- queen problem: in this section we will summarize some literature work done to solve this problem. In paper [1], the author proposed a backtracking algorithm with hybrid approach and claim that it reduces many numbers of trial and error attempt due to the hybrid approach. In paper [2], the authors introduced a hybrid particle swarm optimization algorithm. Tis algorithm is capable of refining the initial swarm more accurately. It is also demonstrated that this technique is much better than randomly produced particles. In paper [3], the genetic algorithm is used to solve N-queen problem with the board size n=200. In experiment results it is to be found that the average time to reach the solution is1023.84 seconds. In paper [4], the parallel algorithm is used to solve N-queen problem which is coded on the basis of previously written MPI version. The key concern to get best result will be how the problem is divided into sub problems so that no processing unit remain idle. In paper [5], the authors proposed swapping algorithm and meta heuristic solution for optimized N-queen problem. This algorithm is capable to find the solution from any random starting

position only by performing swaps between queens in various rows. In paper [6], the authors proposed modified genetic algorithm. They have introduced new techniques to add in genetic algorithm like "Greedy initializing" and "best break point" that improve the efficiency in solving N-queen problem. In paper [7], the Authors proposed a new method which is co-operative particle swarm optimization to solve permutation problem. They demonstrate the use of this technique in the form case study on the basis of N-queen problem.

In paper [8], the author introduced a derivative free method of optimization to solve N-queen problem. The N- queen problem is further categorized into two forms. One is single optimization problem and other is multi objective optimization problem and then two methods are proposed to solve both categories. In paper [9][10][11][14][15], different techniques to solve N-Queen problem to provide better solutions of it.

After going through this literature, it is observed that there is need to improve the performance of genetic algorithm to solve N-Queen problem. The next section illustrates the proposed work.

3. Proposed Work

In this work genetic algorithm is used to solve N-Queen problem. The genetic algorithm uses its operators such as initial population generation, fitness calculation, selection, reproduction and mutation. In this work a novel mutation operator is proposed to improve the performance of genetic algorithm. The mutation operator, mutates the chromosome of the population randomly. In this work, the selection of two queens for mutation is done using a greedy approach. A queen is selected which is having highest number of attacks. That queen is shifted to any other location. The new location is selected randomly. This approach is very useful and most of the time it reduces total number of attacks in the chromosome. Figure 4 is showing a sample of 8-Queen problem.

Fig 3. A sample of 8-Queen problem

The chromosome of the figure 3 is shown in figure 4. Every number is representing the row position of the queen in the respected column.

1	2	6	5	2	5	4	5

Fig 4. A chromosome of the 8-Queen problem

Proposed Mutation Operator – While applying mutation operation of the chromosome, we first find the queen which is having highest number of attacks with other queens. In the above chromosome, the queen in the last column is attacking on four other queens. So, using this proposed mutation operator, the queen in the column is shifted to some other

row. Let row 8 is selected a new position of the queen. The modified chromosome is shown in figure 5.

1	2	6	5	2	5	4	8

Fig 5. Mutated chromosome using proposed algorithm

The genetic algorithm with proposed mutation operator is as follows.

Algorithm: New genetic algorithm with novel mutation operator.

> **Input:** A chromosome of 8-Queen problem
> **Output:** Mutated chromosome of 8-Queen problem
> **Step 1:** Generate Initial Population
> **Step 2:** Calculate fitness of every chromosome
>
> **Step 3**: Perform Selection
> **Step 4**: Perform reproduction
> **Step 5**: Select a chromosome for mutation
>   5.1 Find number of attacks for every queen
>   5.2 Select a queen attacking on maximum queens
>   5.3   Select a new position for this queen
>   5.4   Shift the queen to new Selected position
> **Step 6:** If stopping criteria reached then stop otherwise go to step 3

The proposed algorithm is applied on some instances of N- Queen problem. The next section illustrates the results of it.

4. Result and Analysis

The proposed algorithm is implemented using JAVA programming language using JDK1.8 and NetBeans 8.0.2 software's. The table 1 is showing the results of the experiments.

Table 1 RESULTS OF THE PROPOSED GENETIC ALGORITHM

Algorithm/Queens	8	16	30	40	50
	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)
SRPSO	-	-	6.59	23.73	40.12
PERPSO	-	-	10.32	34.30	53.25
GA			17.29	35.66	54.43
Proposed GA	0.037	0.201	0.234	0.847	1.438

From the Table 1 it is clear that the proposed algorithm performs better as compared to other existing algorithms. The next section illustrates the conclusion and future scope.

5. Conclusion and Future Scope

In this paper N-Queen problem is solved using genetic algorithm. The mutation operator of genetic algorithm is improved using a novel approach. While performing the mutation, a queen attacking on maximum number of other queens is selected and its position is shifted. This approach improves the performance of the genetic algorithm. The work is having some limitations and future directions. In future, the performance of the algorithm can be checked on large number of queens. The other operators of the GA can also be modified and improved.

REFERENCES

[1] A. Alhassan, "Build and Conquer: Solving N Queens Problem using Iterative Compression," 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, Sudan, 2019, pp.   1-5.doi: 10.1109/ICCCEEE46830.2019.9070976

[2] A. Amooshahi, M. Joudaki, M. Imani and N. Mazhari, "Presenting a new method based on cooperative PSO to solve permutation problems: A case study of n-queen problem," 2011 3rd International Conference on Electronics Computer Technology, Kanyakumari, 2011, pp. 218-222. doi: 10.1109/ICECTECH.2011.5941890

[3] A. Khademzadeh, M. A. Sharbaf and A. Bayati, "An Optimized MPI- based Approach for Solving the N-Queens Problem," 2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Victoria, BC, 2012, pp. 119-124.doi: 10.1109/3PGCIC.2012.8

[4] A. M. Turky and M. S. Ahmad, "Using genetic algorithm for solving N- Queens problem," 2010 International Symposium on Information Technology, Kuala Lumpur, 2010, pp. 745-747.doi: 10.1109/ITSIM.2010.5561604

[5] J. E. AghazadehHeris and M. A. Oskoei, "Modified genetic algorithm for solving n-queens problem," 2014 Iranian Conference on Intelligent Systems   (ICIS), Bam, 2014,   pp.   1-5.doi: 10.1109/IranianCIS.2014.6802550

[6] J. E. A. Heris and M. A. Oskoei, "Modified genetic algorithm for solving n-queens problem," in Intelligent Systems (ICIS), 2014 Iranian

[7] M. A. Ayub, K. A. Kalpoma, H. T. Proma, S. M. Kabir and R. I. H. Chowdhury, "Exhaustive study of essential constraint satisfaction problem techniques based on N-Queens problem," 2017 20th International Conference of Computer and Information Technology (ICCIT), Dhaka, 2017, pp. 1-6.doi: 10.1109/ICCITECHN.2017.8281850

[8] M. Kovac, "N-queens problem by efficient non-backtracking algorithm using local search, heuristics and Tabu search elements," The 10th International Conference on Digital Technologies 2014, Zilina, 2014, pp. 159-163.doi: 10.1109/DT.2014.6868708

[9] N. Vaughan, "Swapping algorithm and meta-heuristic solutions for combinatorial optimization n-queens problem," 2015 Science and Information Conference (SAI), London, 2015, pp. 102-104.doi: 10.1109/SAI.2015.7237132

[10] R. Kumar Rout, Shubham, J. Kumar Das and P. Pal Choudhury, "Solution of 16-Queens Problem Using 3-variable Affine Boolean Functions (2^{n+1})," 2018 8th

International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, 2018, pp. 853-855.doi: 10.1109/CONFLUENCE.2018.8442597

[11] S. Güldal, V. Baugh and S. Allehaibi, "N-Queens solving algorithm by sets and backtracking," SoutheastCon 2016, Norfolk, VA, 2016, pp. 1- 8.doi: 10.1109/SECON.2016.7506688

[12] Y. Azuma, H. Sakagami and K. Kise, "An Efficient Parallel Hardware Scheme for Solving the N-Queens Problem," 2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems- on-Chip (MCSoC), Hanoi, 2018, pp. 16-22.doi: 10.1109/MCSoC2018.2018.00015

[13] Y. Wang, H. Lin and L. Yang, "Swarm Refinement PSO for Solving N- queens Problem," 2012 Third International Conference on Innovations in Bio-Inspired Computing and Applications, Kaohsiung, 2012, pp. 29- 33.doi: 10.1109/IBICA.2012.43

[14] Y. Sasaki, M. Fukui and T. Hirashima, "Development of iOS Software n-Queens Problem for Education and Its Application for Promotion of Computational Thinking," 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 2019, pp. 563-565. doi: 10.1109/GCCE46687.2019.9015331

[15] F. Jesus de Souza and F. Luis de Mello, "N-Queens Problem Resolution Using the Quantum Computing Model," in IEEE Latin America Transactions, vol. 15, no. 3, pp. 534-540, March 2017. doi: 10.1109/TLA.2017.7867605