Solving N Queen Problem using Genetic Algorithm

Article *in* International Journal of Computer Applications · July 2015

DOI: 10.5120/21750-5005

3 authors, including:

Fouad H Awad
University of Information Technology and Communications
9 PUBLICATIONS **20** CITATIONS

SEE PROFILE

Solving N Queen Problem using Genetic Algorithm

Ahmed S. Farhan
Department of Computer Science
Al Maaref University College
Erbil, Iraq

Wadhah Z. Tareq
Department of Computer Science
Al Maaref University College
Istanbul, Turkey

Fouad H. Awad
Department of Computer Science
Al Maaref University College
Erbil, Iraq

ABSTRACT
This paper, explain solution to find the 92 solution of n-Queen problem based on GA (Genetic Algorithm). The n-Queen problem become a Widespread platform for the AI researcher for implement their intelligence algorithms and try them. The Genetic algorithm used to solve the problem and each chromosome is be a solution for the problem and depending on the steps of the GA, The 92 solution, all possible solution for 8 Queen problem is founded. The represent of each chromosome have been by using one dimension array with size equal 8 contain only the queens which represent a one solution and the empty location are aborted to reduce the searching time.

General Terms
Computer Science - Search Algorithm

Keywords
N-Queen Problem, 8-Queen Problem, Heuristic Techniques, Genetic Algorithm (GA), Swarm Intelligence (SI).

1. INTRODUCTION
Researchers and scientists offered various Heuristic algorithms for optimization by modeling from physical and biological processes in nature, which often operate collectively. Heuristic algorithms against classic algorithms operate randomly and search along with the space. The other difference between them is that Heuristic algorithms don't use space gradient information. These kinds of methods just use fitness function for guiding the search, but because of having intelligence as type of collective intelligence, are able to find solution. Examples of these algorithms includes inherited algorithms that have inspired by Genetics and evolution science (1975), simulated annealing by modeling from thermodynamics observations (1983), immunity algorithm by simulating human defense system (1986), searching ants population by simulating ants behavior in finding food (1991), and optimization particles swarm by following birds social behaviors (1995).

2. RELATED WORK
Previously, lots of work is done on this problem. K. D. Crawford in [2], applied Genetic Algorithm and have discussed two ways to solve n-Queen problem. Ivica et al. provided a comparison of different heuristic techniques in [1]. The techniques include Simulated Annealing, Tabu Search and Genetic Algorithm.

We found no solution to the problem based on Ant Colony Optimization. So, we can say that this is first ever application of ACO to the n-Queen problem. In order to apply ACO, we first organized the search space. We then discussed a few modifications in the calculation of some parameters for ACO. We have added a few constraints in basic ACO as it cannot be applied directly to the n-Queen problem. A detail discussion on all these is provided in the subsequent sections.

3. N-QUEEN PROBLEM
For each an n x n chessboard we know how much the maximum number of queen can be placed, so there is no one attack the another, is equal to n. One of the classical combinatorial problem is the eight queens problem which is a method of putting the eight queens on a board with 8 x 8 size such as chessboard and each one unable to capture any another queen. The 8 queens problem was generalized as putting n queen in a way that non-attacking between the queens. The number, of the ways to putting n of queen on the board without any attacking for the first eight n different ways, are 1, 0, 0, 2, 10, 4, 40, 92 [4].

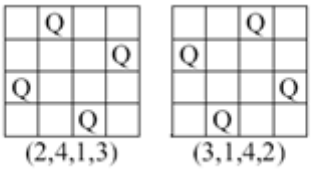

Fig 1: N-tuple notation exemplar

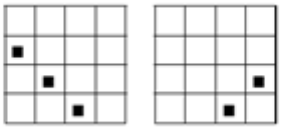

Fig 2: Third "left" and second "right" diagonal

So poor, weak, algorithm for find the solution for the n-queen problem, which depending on pleasing a one single queen in row, and that placements reduce the conflict in the rows. because each queen must be in a one row and different from another row and column, we will use solution representation as n-tuples (q1 ,q2 ,..., qn) that are permutation of n-tuple (1,2,..., n). Using this representation, guaranteeing no rook attacks. This leads to reduce the rate of complexity of problem since to solve part of the conflict. Figure 1 explain 4-tuples for the 4-queens problem (all (two) solutions in the 4-queens problem are shown). Since the n-tuple representation eliminates row and column conflicts, the problem have only in the diagonal attacks between queens. Accordingly, the fitness function 3 should count diagonal attacks. The 2n-1 "left" and 2n-1 "right" diagonals have to be checked (Figure 2), but there cannot be a conflict on the first and last diagonal (such diagonals consist of only one field) so that algorithm only will check the 2n-3 "left" and 2n-3 "right" diagonals in which the conflict will founding[3]. Fitness function will return

value equal to zero if and only if there is no conflicts and this will be correct solution. A queen that occupies i-th column and qi-th row is located on the i+qi-1 left and n-i+qi right diagonal. i-th and j-th queens share a diagonal if:

$$i - qi = j - qi \quad (1)$$

or

$$i - qi = j - qi \quad (2)$$

Equation (1) represents "left diagonal" and on the other hand the vice versa. And this leads to O(n) complexity of the fitness function.So that each new solution will be differs from the previous solution only in two positions, After that calculating the new value of the fitness function can be by observing the 8 diagonals that directly found the change depending in the number of queens.

4. GENETIC ALGORITHMS

One of the search and optimization procedures, which based on 3 main principles: first selection, then crossover and the mutation, the Genetic Algorithm. In Genetic algorithm the solutions are represent as a chromosome in an individuals that are estimated using function known as the fitness function which represent the correctness of each solution in the individual [5]. The genetic algorithm depending on a Basic structure of which shown in the following list:

A random population of individuals (potential solutions) is created. All individuals are evaluated using a fitness function.

Certain number of individuals that will survive into next generation is selected using selection operator. Selection is somewhat biased, favoring "better" individuals.

Selected individuals act as parents that are combined using crossover operator to create children.

A mutation operator is applied on new individuals. It randomly changes few individuals (mutation probability is usually low).

Children are also evaluated. Together with parents they form the next generation.

The first step is randomly and always the fitness is computing after it. Steps 2.-5. are repeated number times depending on a given times of iterations which have been set in the start, solution improvement rate falls below some threshold, or some other stop condition has been satisfied[6]. One modification of this basic structure is a 3-way tournament selection used here. Instead of selecting individuals from one generation to the next, selection and crossover are performed continuously. First, the three individuals usually selected completely depending on random. Then, the two individuals which have the high fitness value and near to stop condition are combined using crossover to produce a new offspring from two old random solution and will replace the worst individual. There is no clear distinction between generations. Individual representation and fitness function for n-Queen problem were presented in the previous chapter. It is also important to design suitable crossover method and mutation operation that will operate on n-tuple representation.

Mutation operator which was use very simple: its depending on randomly, for a given tuple, select two positions and change the numbers between themselves. This lead to creates a new individual, similar to the original one, and of the tuple is preserved. An example is given in figure 3:

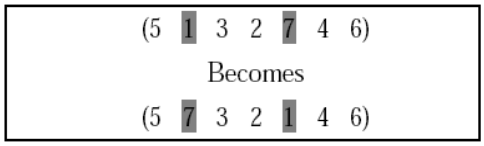

Fig 3 : Mutation Operator

There are several of different possibilities for a crossover operator. One of this version is equivalent to the mutation operator: swapping two random positions in a tuple. Obvious drawback of this operator is that it does not combine genetic material of parents[7].

Another crossover operator is PMX1 crossover. It is similar to much to the two-point binary crossover . First step is random selection of two positions within chromosomes and exchange of genetic material:

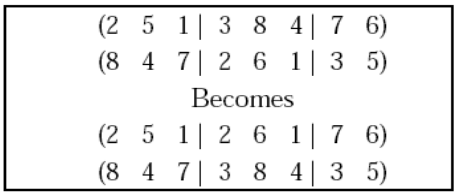

Fig 4: PMX crossover – first step

In most cases, this will result in invalid tuples, since numbers in a tuple must be unique. Second step in PMX crossover eliminates duplicates. In the example above, number 2 occurs at positions 1 and 4 in the first offspring. The 2 at position 4 is newer (from the crossover), so the 2 at position 1 is changed into 3 that was at position 4 before the crossover[8].

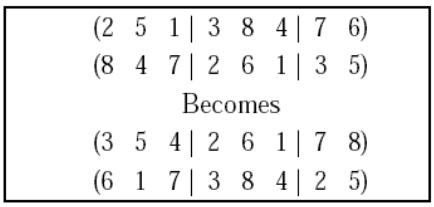

Fig 5: PMX crossover – second step

The third operator is designed for 3-way tournament selection: parents are compared, and equivalent positions are copied to the offspring. Other positions in the offspring tuple are filled in randomly, but care is taken to preserve tuple validity. If parents are equivalent, one of them is replaced by a randomly created tuple to avoid chromosome duplication.

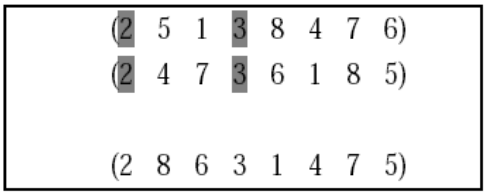

Fig 6: 3-way tournament crossover

5. N-QUEEN WITH GENETIC

The GA is implemented for solving 8-queen problem in order to find all 92 correct solutions. The algorithm starts by initial solution.

Each chromosome is represented by one dimension array with size equal to 8 and this the idea which reduce the space search.

Each position refer to row in 8*8 board and the value containing in the position is represent the Colum number for example:

2	5	3	7	6	4	1	0

Where number 2 refer to the Colum 2 and the array position which equal 0 refer to the raw so the queen location is [0,2].

Algorithm: Genetic Algorithm:

Input: Initial random solutions.

Output: All possible solutions for eight queens problem.

Step1
Generate 92 random solutions.

This was done by initializing 92 chromo-some (with length of 8), This is the initial population for GA.

Step2
Evaluate the fitness of each chromosome (solution).

Step3
Rank the chromosomes dep-ending on their fitness's values

Step4
apply the crossover and mutation on each pair of chromosomes to generate new solutions.

Step5
Repeat the steps 2,3,4 until rich to the stop condition.

The fitness is calculated be founding the numbers of crossover between the queens and the stop condition.

6. RESULTS AND DISCUSSION

Tables 1 show the results of applying the genetic algorithm to find the 92 solutions for the 8-Queen, with the number of iterations that are required to achieve each solution.

Table 1. The 92 Solution And Number of Iterations

z	Solution
1017	3 5 7 1 6 0 2 4
2894	4 6 0 2 7 5 3 1
2722	2 6 1 7 4 0 3 5
313	5 1 6 0 2 4 7 3
2417	3 6 0 7 4 1 5 2
455	6 3 1 4 7 0 2 5
3056	3 1 6 4 0 7 5 2
1046	1 3 6 0 4 2 5 7
2097	3 6 4 1 5 0 2 7
4199	1 5 7 2 0 3 6 4
861	3 7 0 2 5 1 6 4
8724	3 6 2 7 1 4 0 5
406	3 1 6 2 5 7 0 4
6325	0 6 4 7 1 3 5 2
1454	5 3 1 7 4 6 0 2
1263	6 0 2 7 5 3 1 4
129	2 5 1 6 0 3 7 4
4676	3 6 4 2 0 5 7 1
2845	4 7 3 0 2 5 1 6
2046	4 2 0 5 7 1 3 6
373	2 0 6 4 7 1 3 5
3437	2 5 7 1 3 0 6 4
172	3 1 7 4 6 0 2 5
755	4 1 3 6 2 7 5 0
400	3 0 4 7 1 6 2 5
13	4 2 7 3 6 0 5 1
6056	3 1 6 2 5 7 4 0
3199	3 1 7 5 0 2 4 6
641	2 6 1 7 5 3 0 4
4053	1 6 4 7 0 3 5 2
740	5 3 6 0 7 1 4 2
11526	3 7 4 2 0 6 1 5
859	1 3 5 7 2 0 6 4
2738	5 7 1 3 0 6 4 2
5181	3 5 0 4 1 7 2 6
1341	4 1 3 5 7 2 0 6
3811	2 5 1 6 4 0 7 3
4333	2 4 7 3 0 6 1 5
4702	1 6 2 5 7 4 0 3
4082	4 0 7 3 1 6 2 5
3524	4 6 3 0 2 7 5 1
720	2 5 3 1 7 4 6 0
3829	2 5 7 0 4 6 1 3
12279	6 1 5 2 0 3 7 4
167	4 1 7 0 3 6 2 5
8402	4 2 0 6 1 7 5 3
909	6 2 0 5 7 4 1 3
3685	5 3 6 0 2 4 1 7
1548	6 4 2 0 5 7 1 3
128	1 5 0 6 3 7 2 4
2924	5 2 0 7 4 1 3 6
1399	0 5 7 2 6 3 1 4
3210	0 6 3 5 7 1 4 2
5188	4 6 1 3 7 0 2 5
1596	5 2 0 7 3 1 6 4
1812	6 3 1 7 5 0 2 4
4994	2 4 1 7 5 3 6 0
8069	4 1 5 0 6 3 7 2
4700	3 5 7 2 0 6 4 1
7784	4 6 0 3 1 7 5 2
7016	5 0 4 1 7 2 6 3
9710	5 2 6 1 3 7 0 4
1821	4 6 1 5 2 0 7 3
419	5 1 6 0 3 7 4 2
4958	7 2 0 5 1 4 6 3
66	1 7 5 0 2 4 6 3
1692	7 3 0 2 5 1 6 4
603	1 4 6 3 0 7 5 2
9827	6 2 7 1 4 0 5 3
1782	2 7 3 6 0 5 1 4
26584	5 2 0 6 4 7 1 3
7314	6 1 3 0 7 4 2 5
10854	0 4 7 5 2 6 1 3
3072	4 0 3 5 7 1 6 2
3289	2 5 3 0 7 4 6 1
23834	3 0 4 7 5 2 6 1
13955	2 5 7 0 3 6 4 1
8247	7 1 4 2 0 6 3 5
2751	2 5 1 4 7 0 6 3
2975	5 2 4 7 0 3 1 6
24804	2 4 6 0 3 1 7 5
37184	5 3 0 4 7 1 6 2
56132	4 7 3 0 6 1 5 2
1944	5 2 4 6 0 3 1 7
59389	4 0 7 5 2 6 1 3

169369	7 1 3 0 6 4 2 5
8087	4 6 1 5 2 0 3 7
4817	3 1 4 7 5 0 2 6
2499	1 4 6 0 2 7 5 3
3355	3 7 0 4 6 1 5 2
8448	5 2 6 3 0 7 1 4
1502	2 4 1 7 0 6 3 5

The result is good special in number of solution and for future can applying another meta heuristics algorithm and comparing the result with this research.

From table 1 there is a solution founded by repeat the algorithm 13 times and this is the faster solution founded between the 92 solution.

Another solution founded after repeat the solution 169369 and this difference related to the randomization in initial solution and the random in the solution steps.

7. CONCLUSION

This paper found the total aver able solution 92 by applying the GA. Each chromosome represent by one dimension array with size equal to 8. This representation reduce the empty cells which required more time comparing with our representation. The one array hold the 8 queens for one solution and depending on it the GA applying and the fitness calculated.

8. REFERENCES

[1] I. Martinjak and M. Golub, "Com-parison of Heuristic Algorithms for the N-Queen Problem", Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces, June 25, 2007.

[2] K. D. Crawford, "Solving the N-Queens Problem Using GA", In Proceedings ACM/SIGAPP Symposium on Applied Computing, Kansas City, 1992, pages 1039-1047.

[3] Božiković, Marko, G. "paralleling genetic algorithm", Faculty of Electrical Engineering and Computing, Zagreb, 22.05.2006.

[4] Sloane, Neil J. A., Number of ways of placing n non attacking queens on n x n board, The On-Line Encyclopedia of Integer Sequences id:A000170, http://www.research.att.com/~njas/sequence,s/A000170, (30.01.2007.)

[5] David E. Goldberg, Genetic algori-thms in search, optimization and machine learning, Addison-Wesley Publishing Company Inc., Reading, MA, 1989.

[6] Kelly D. Crawford, "Solving n Queen problem using genetic algorithms", Tulsa University.

[7] Eric Cantú-Paz, "A summary of research on parallel genetic algorithms", Computer Science Department and The Illinois Genetic Algorithms (IlliGAL), University of Illinois at Urbana-Champaign, cantupaz@uiuc.edu

[8] Eric Cantú-Paz, "A survey of parallel genetic algorithms, Computer Science Department and The Illinois Genetic Algorithms Laboratory", University of Illinois at Urbana-Champaign, cantupaz@illigal.ge.uiuc.edu.