Performance Analysis of N-Queen Problem using Backtracking and Genetic Algorithm Techniques

VikasThada Asst.Prof(CSE),ASET Amity University Gurgaon, India Shivali Dhaka Asst.Prof(CSE),ASET Amity University Gurgaon, India

ABSTRACT

In this paper the research work has done comparative analysis of one of the famous NP hard problem: NQueen using traditional Backtraking and Genetic Algorithm(GA). The research work has implemented the solution of the NQueen problem using backtracking and using GA. Both the methods of solving NQueen problem are entirely different. The first one is general method and takes time in days, months and years as N increases. In this paper the time taken by the two methods for a given value of N are compared. The work has restricted values of N upto 50 only as beyond this it is extremely difficult to get the solution of the problem using backtracking method. Both the implementation have been carried out in MATLAB using Pentium Core 2 Duo T6600 Core 2.2 GHz processor on Windows 8 with 4GB RAM. Because of the random nature of GA instead of time taken in obtaining all possible solutions for a given value of N, the time taken in obtaining say count number of solutions was first determined using GA, then time taken in same number of solutions that is count using backtracking was noted. Results are then presented in tabular manner.

General Terms

Backtracking, collisions, NP, problem

Keywords

NQueen, genetic, algorithm, solutions

1. INTRODUCTION

In this research paper solution of NQueen problem is presented using Genetic Algorithm and is compared with backtracking method. It's a problem that belongs to the NP class of problems. There are number of problems for which a deterministic algorithm exists with polynomial running time. These problems belong to class P. Other problems where no deterministic algorithm exists in polynomial time belong to class NP. These class of problems are very complex and difficult to solve because of their high time complexity like O(2n) or O(!n). Solutions of these problems cannot be obtained in real time. To solve these problems some approximations or heurisitics methods must be used[1]. GA is approximation based evolutionary method that can be used for approximate solution of these types of problems. Same the reseach work has employed. As compare to backtracking method the GA based method takes very less time and returns results in an efficient manner.

2. GENETIC ALGORITHM

GAs are search algorithms that follow the concept of natural selection and genetics [2]. GA are powerful and very efficient search and optimization techniques motivated by the natural selection theory of Darwin [3].

Genetic Algorithms [4] are based on the principle of heredity and evolution which claims "in each generation the stronger individual survives and the weaker dies". Therefore, each new generation would contain stronger (fitter) individuals in contrast to its ancestors. The process of GA's is iteration based of constantpopulation size of candidate solutions. In eachgeneration/iteration each chromosome's fitness in the current population is evaluated and new population evolves. Chromosomes with higher fitness values goes through reproduction phase in which selection, crossover and mutation operators are applied to get new population. Chromosomes with lower fitness values are discarded. Again this generated new population is evaluated and selection, crossover, mutation operators are applied. This process continues until we get an optimal solution for the given problem

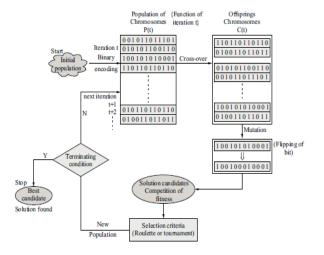


Fig 1: Basic operation of Genetic Algorithm [7]

2.1 Fitness Evaluation

Fitness function is a function that is responsible for evaluating some value to indicate among number of solutions which one is optimum. It can also be considered as a measure of performance or fitness to show how fit is the candidate solution. Fitness function for this research work will be discussed in the next section.

2.2 Selection

Once the fitness evaluation process is done next step is to perform selection operation. Process of selection operation is based on the principle of "survival of the fittest". Higher fitness valued chromosomes goes through reproduction. Lower fitness valued chromosomes are discarded. There are number of ways to implement this operator, but all relies on the concept that candidates with good fitness values are to be preferred over poor fitness values. The idea is to give

preference to better individuals. This selection operation does the replication of candidate chromosomes with good fitness values and eliminating those with poor fitness values [4].

Individual	Fitness		Individual	Fitness
A	8		A	8
В	7	$\qquad \qquad \qquad \bigcirc$	В	7
С	3		A	8
D	5		D	5

Fig 2: Selection Operator on a Population of 4 Individuals[5]

The research work uses roulette wheel selection method, tournament selection method, stochastic uniform, uniform and remainder method as selection operator.

2.3 Crossover[2,6]

In the crossover operation mating of two chromosomes is performed that gives birth to two new offspring. This operation of crossover always happens with one parameter that is known as probability of crossover (ProC). When ProC is say 0.8 it means only 80% of the total population goes for crossover operation. Rests 20% chromosomes remain abstain from this operation and has no effect of crossover. Motive behind performing crossover operation is to explore new solutionsand exploit use of old solutions. GA forms an optimum solution by mating two fit chromosomes together. Chromosomes with higher fitness will always have good selectionprobability then others with lower fitness values, thus a good solution moves from one generation to next generation

One point crossover (crossover point 7)

Doc2 0 0 1 0 0 0 1 0 1 0 1 0 0 1	0 0 0 0 0 0 0 0	1

After crossover

Doc1	1	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1
Doc2	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0

Fig 3: Single Point Crossover Explained

2.4 Mutation [2,6]

Mutation involves changing one bit of a chromosome from 0 to 1 or viceversa. This is performed under the constraint parameter called probability of mutation (ProM). For example if ProM is 0.10 then 10% genes of total chromosomes will go for mutation. The concept of mutation is based on this natural theory that varying breeds are possible only by varying gene values. After this operation fitness quality of new chromosomes may be high or low then old ones. In case new chromosomes are poor then old ones they are removed during selection process. The motive behind mutation is regaining the lost and discovering varying breeds. For example: randomly mutate chromosome at position 5



After mutation (bit number 5 is mutated i.e. bit is negated)

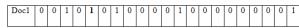


Fig4: Mutation Operation Explained

3. N-OUEEN PROBLEM

The N-Queen problem is to place N queens on chessboard of size N by N such that no queen can attack any other queen. One queen is vulnerable to attack another queen if they share same row, column or diagonal. There are number of ways these N queens can be placed and each way is one solution. The problem is so difficult because of the fact that there are $\binom{n2}{n}$ i.e n^2 Cn possible solutions of the problem but considering the fact the each queen is placed on a distinct row and column solutions are only !n[8] . Each solution can be represented as a tuple of length N as (Q_1,Q_2,\ldots,Q_N) . Index positions of the tuple represents row positions and values represent column position. Reverse can also be chosen. Thus solutions are basically a permutation of the tuple(1,2,3,...N). For example consider a tuple for 8 Queen: (4 3 5 7 8 3 2 1) here Q1 will be placed in row=1,col=4, Q2 in row=2,col=3 and so on.

As each queen will be placed on distinct row and column we do not have to worry about collision of queens on row and columns. We simply have to concern about whether any collision is possible on diagonal positions. For a board of size N by N there are N*(N-1)/2 pairs of non attacking queens . For examples for N=8 , number of pairs of nonattacking queens are 28 [9]. This formula will be used later on.

For finding number of collisions between a pair of attacking queens there is a simple method by [8].

Consider a tuple as (Q_1, Q_2, \dots, Q_N) . Two queens say Qi and Qj will be on attacking position diagonally if:

$$|\mathbf{i}-\mathbf{Q}\mathbf{i}| = |\mathbf{j}-\mathbf{Q}\mathbf{i}| \tag{1}$$

Or

$$\mid i+Qi\mid = |j+Qj| \tag{2}$$

From 1 or 2

$$|Oi-Oj| = |i-j| \tag{3}$$

This concept is used in writing the fitness function in GA and same is used to find proper position of the queen Qj so that Qi and Qj does not attack diagonally in solution using backtracking method. Lets take an example to understand this.

Consider a tuple of N=8. (4,3,6,7,2,8,1,5). Take i=3,j=4 i.e 3^{rd} and 4^{th} queen, thus Q3=6 and Q4=7. Using equation (3)

|6-7|=|3-4|

|1| = |1|

	1	2	3	4	5	6	7	8
1							Q	
2					Q			

3		Q				
4	Q					
5						Q
6			Q			
7				Q		
8					Q	

Fig 5: Explaining Concept of Collisions in 8-Queen Problem

Its clearly visible from the figure that there are different pairs of queens on attacking positions diagonally like i=5,j=8 and i=1,j=2.

4. EXPERIMENTATION

In this section work carried out in implementing the solution of N queen problem using backtracking and GA is discussed. Its quite easy to implement the solution using backtracking method so we emphasis on implementation using GA mainly on crossover and fitness function.

4.1 Chromosome Representation

The representation of the chromosome will be a tuple from 1 to N and in the population every chromosome will be a permutation from 1 to N. As discussed earlier the indexes of the tuple elements will be column numbers and their values will represent rows.

4.2 Crossover Function[1,10]

As elements in each tuple have to be distinct, it is very much possible that after croosoverduplicacy of elements may be there in new offsprings. That's why general crossover functions like 1-point, 2-point or random point crossover methods does not work. Consider two tuples:

P1=(8 3 6 7 5 1 2 4) P2=(6 1 5 3 7 2 8

Assuming crossover point chosen is 3, new offsprings will be:

C1=(8 3 6 3 7 2 8 4)

C2=(6 1 5 7 5 1 2 4)

Clearly the new chromosomes are not valid.

Thus we have to rely on new crossover techniques where chromosomes are some permutation of some numbers between 1 and N. Table given below shows some of the permutation respected crossover techniques.

Table 1: Few Permuation Respected Crossover Techniques

Partially Mapped Crossover (PMX)	Goldberg and Lingle (1985) [14]
Order - Crossover (OX1)	Davis (1985) [12]
Order Based Crossover (OX2)	Syswerda (1991) [18]
Position Based Crossover (POS)	Syswerda (1991) [18]

Heuristic Crossover (HX)	Grefenstette (1987) [16]
Edge Recombination Crossover (ER)	Whitley et al. (1989) [19]
Sorted Match Crossover (SMX)	Brady (1985) [11]
Maximal Preservative Crossover (MPX)	M'uhlenbein et al. (1988) [16]
Voting Recombination Crossover (VR)	M"uhlenbein (1989) [17]
Alternating - Position Crossover (AP)	Larranaga et al. (1996) [15]

In the research work PMX technique have been used. In this technique after the crossover has been done all elements where there is a conflict are adjusted by some swapping operation[1]. Consider the two tuples

2	5	1	3	8	4	7	6
8	4	7	2	6	1	3	5

Using 2-point crossover new tuples are:

2	5	1	2	6	1	7	6
8	4	7	3	8	4	3	5

The conflicts are removed as:

- 1. 2 appear in first tuple at position 1 and 4, 3 at position 7 is swapped with 2 at position 1.
- 1 at position 3 and at position 6 conflict. 1 at position 3 is swapped with 4 at position 2 in second tuple.
- 3. Similarly 6 in first tuple is swapped with 8 in second tuple.

The new tuples will be:

3	5	4	2	6	1	7	8
6	1	7	3	8	4	2	5

4.3 Mutation Function

For mutation two random points have been generated between 1 and N and values at these positions have been swapped.

4.4 Fitness Function

As discussed earlier total numbers of pairs of non attacking queens for a board of size N by N is: N*(N-1)/2. This value is used as maximum fitness for board of size N. For each chromosome collisions are calculated and subtracted from this value. For exact solution collisions should be zero and fitness should be N*(N-1)/2.

4.5 Selection Function

Populations were duplicated by 50% of top fitness chromosomes.

4.6 Results

For a population of size=1000 ,generations=100 and taking values of N=4 to 50 using both methods execution time was calculated. It's not possible to actually find out all solutions using backtracking method without super computing power so first GA implementation was run and an average of 10 values was chosen. Time and number of solutions thus obtained were noted. Same time was also noted using backtracking for same number of solutions. Results are mentioned in the table 1. The time is measures in seconds.

5. ANALYSIS AND CONCLUSION

It is clearly visible from the table 1 that for initial values of N (from 4 to 25) for fixed number of solutions backtracking techniques results in less execution time compare to GA technique. This is considering the fact all solutions are not possible using GA due to its random nature and it gives only approximate solutions. Further we made the provision in the backtracking technique that as soon as fixed number of solutions are obtained it has to stop and display execution time. Comparing results for N=30 and N=35 we can see that GA performs well for higher values of N. This can also be manifested by the fact that we almost waited for 3 days to get a solution for N=40,45,50 but couldn't get at least one solution using backtracking method and same was obtained using GA in only 15 seconds. Thus GA performs much better for higher values of N in comparison to backtracking method.

Table 2: Execution Time Analysis of N Queen using Backtracing& Genetic Algorithm Method

N	Backtracking	GA	No. of Solutions
4	0.311	7.9086	2
5	0.428	8.3754	10
6	0.460	8.3317	4
7	0.773	9.5949	28
8	0.492	9.3168	35
9	0.429	9.2099	21
10	0.510	8.8798	10
11	0.476	8.7881	8
12	0.503	8.9483	8
13	0.443	9.0594	9
14	0.482	9.2338	11
15	0.474	9.1055	4
16	0.469	9.2217	7
17	0.532	9.448	4
18	1.302	9.6218	7
19	0.513	9.4781	5
20	1.124	9.6125	4
21	0.528	9.8035	4
22	6.425	9.9599	8
23	0.780	10.1447	3
24	2.993	10.1882	4
25	1.338	10.6158	10
30	225.306	11.2727	5
35	1041.889	11.8459	2
40	??	13.1276	4
45	??	14.2622	2
50	??	15.5400	3

6. REFERENCES

- [1] Kelly D. Crawford, "Solving n-Queen problem using genetic algorithms" SAC '92 Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing,pp:1039-1047,1994
- [2] B.Klabbankoh, O.Pinngern. "applied genetic algorithms in information retrieval" *Proceeding of IEEE*, pp.702-711, Nov 2004

- [3] S.S.Satya and P.Simon, "Review on Applicability of Genetic Algorithm to Web Search," *International Journal of Computer Theory and Engineering*, vol. 1, no. 4, pp. 450-455, 2009.
- [4] Shokouhi, M.; Chubak, P.; Raeesy, Z "Enhancing focused crawling with genetic algorithms" Vol. 4-6, pp.503-508,2005.
- [5] M.A.Kauser, M. Nasar, S.K.Singh, "A Detailed Study on Information Retrieval using Genetic Algorithm", *Journal of Industrial and Intelligent Information* vol. 1, no. 3, pp.122-127 Sep 2013.
- [6] J.R. Koza, "Survey Of Genetic Algorithms And Genetic Programming", Proceedings of the Wescon,pp.589-595,1995
- [7] V.Thada, V.Jaglan, "Use of Genetic Algorithm in Web Information Retrieval", International Journal of Emerging Technologies in Computational and Applied Sciences, vol.7,no.3,pp.278-281, Feb,2014
- [8]E. Horowitz ,S. Sahni,S.Rajasekaran, "Fundamentals of Computer Algorithms", University Press India. 2007.
- [9] Planning and Search, Available at: www.cs.nott.ac.uk/~nza/G52PAS/lecture4.pdf
- [10] G. Üçoluk, "Genetic Algorithm Solution of the TSP Avoiding Special Crossover and Mutation", Intelligent Automation & Soft Computing, vol. 8, no.3, 2002
- [11] Brady, R.M. "Optimization Strategies Gleaned from Biological Evolution." Nature 317, 1985, pp. 804.
- [12] Davis, L. "Applying Adaptive Algorithms to Epistatic Domains." Proceedings of theInternational Joint Conference on Artificial Intelligence, 1985, pp. 162-164.
- [13] Grefenstette, J. "Incorporating Problem Specific Knowledge into Genetic Algorithms." Genetic Algorithms and Simulated Annealing, edited by Davis L., Morgan Kaufmann, Los Altos, CA, pp. 42-60, 1987.
- [14] Goldberg, D.E., and R. Lingle. "Alleles, Loci, and the Traveling Salesman Problem." Proceedings of the First International Conference on Genetic Algorithms and Their Application, edited by Grefenstette J., Lawrence Erlbaum Associates, Hillsdale, NJ,1985, pp. 154-159.
- [15] Larranaga, P., C.M.H. Kuijpers, M. Poza, and R.H. Murga. "Decomposing BayesianNetworks: Triangulation of the Moral Graph with Genetic Algorithms." Statistics andComputing, 1996.
- [16] M"uhlenbein, H., M. Gorges-Schleuter, and O. Kramer. "Evolution Algorithms in Combinatorial Optimization." Parallel Computing, 7, 1988, pp. 65-85.
- [17] M"uhlenbein, H. "Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization." Proceedings on the Third International Conference on Genetic Algorithms, edited by Schaffer J., Morgan Kaufmann Publishers, Los Altos, CA, 1989 pp. 416-421.
- [18] Syswerda, G. "Schedule Optimization Using Genetic Algorithms." Handbook of Genetic Algorithms, edited by Davis L., Van Nostrand Reinhold, New York, 1991,pp. 332-349.
- [19] Whitley, D., T. Starkweather, and D. Shaner. "The Traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination." Handbook of Genetic Algorithms, edited by Davis L., Van Nostrand Reinhold, New York, 1991, pp. 350-372