Volume 3, Issue 2, February 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Solving N Queen Problem Using Various Algorithms – A Survey

S. Pothumani
Department of CSE,
Bharath University,
India

***Abstract*— Combinatorial optimization problems form a set of problems which need a considerable effort and time to be solved. Their difficulty lies in the fact that there is no formula for solving them exactly. Every possibility has to be examined in order to find the best solution and the number of possibilities increases exponentially as the size of the problem increases. In this paper we discussed about the N queen problem which is one of the most popular NP- hard problem and various methods to solve n queen problem**

***Keywords*— NP-Hard Problems, N-Queen Problem, ACO, Genetic Algorithm, DNA Sticker Algorithm**

I. 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 defensive system (1986), searching ants population by simulating ants behavior in finding food (1991), and optimization particles swarm by following birds social behaviors (1995).

The n-Queens Problem

The *n*-queens problem, originally introduced in 1850 by Carl Gauss, may be stated as follows: find a placement of n queens on an $n \times n$ chessboard, such that no one queen can be taken by any other. While it has been well known that the solution to the n-queens problem is *n*, numerous solutions have been published since the original problem was proposed. Many of these solutions rely on providing a specific formula for placing queens or transposing smaller solutions sets to provide solutions for larger values of n (Bernhardsson, 1991 and Hoffman *et al*., 1969).Empirical observations of smaller-size problems show that the number of solutions increases exponentially with increasing *n.* Alternatively, search-based algorithms have been developed. For example, a backtracking search, will systematically generate all possible solution sets for a given $n \times n$ board. In practice, however, backtracking approaches provide a very limited class of solutions for large size boards because it is difficult for a backtracking search to find solutions that are significantly distinct in the solution space. Several authors have proposed other efficient search techniques to overcome this problem. These methods include search heuristic methods and local search and conflict minimization techniques. Recently, advances in research in the area of neural networks have led to several papers proposing solutions to the *n*-queens problem via neural networks. Specifically, the use of Hopfield networks has been applied to the n-queens problem by Mandziuk. The Hopfield neural network is a simple artificial network which is able to store certain patterns in a manner similar to the brain in that the full pattern for a given problem can be recovered if the network is presented with only partial information. The ability of neural networks to adapt and learn from information has applications in optimization problems beyond the *n*-queens problem.Finally, the problem has been stated as an integer programming similar to the assignment problem.

The n-QueensS Mathematical Program

Following Hoffman, *et al*. (1969), we define the chessboard as an $n \times n$ matrix of square elements for *n* 4; each square is identified as an ordered pair (i,j), where i and j are the row and column numbers of the square, respectively. Given each ordered pairs,
we can identify:

1. Row k ($k = 1,...,n$) consists of all ordered pairs (k,j) $j = 1,...,n$.
2. Row k ($k = 1,...,n$) consists of all ordered pairs (k,j) $j = 1,...,n$.
3. Minor diagonal k ($k = 2,...,2n$) consists of all ordered pairs (i,j) such that $i + j = k$.
4. Major diagonal k ($k = 1-n,...,n-1$) consists of all ordered pairs (i,j) such that $i - j = k$.

For a given chess board of size *n*, let $dij = 1$ if a queen occupies (i,j) and 0 otherwise $i, j = 1, ..., n$. Then a solution to the *n*-queens problem is obtained from the following mathematical program: The objective function has no practical use since it is know a priori that the value is *n* for *n* >= 4. In essence, this program uses the mathematical programming search routine to find a solution. Note that additional constraints can be added to force a queen onto a particular square (ordered pair). In this case, the objective function may not achieve a value of n. Recognizing that the decision variables are binary, the first constraint ensures that at most one queen will be placed in each row. The second constraint ensures that at most one queen will be placed in each column. We note that the first two constraints will hold with equality for *n* 4 since the solution is *n*. The third and fourth constraints force at most one queen on the diagonals.

Solution for n= 10

A solution for a chessboard of size *n = 10* obtained from the spreadsheet is shown in the following diagram:

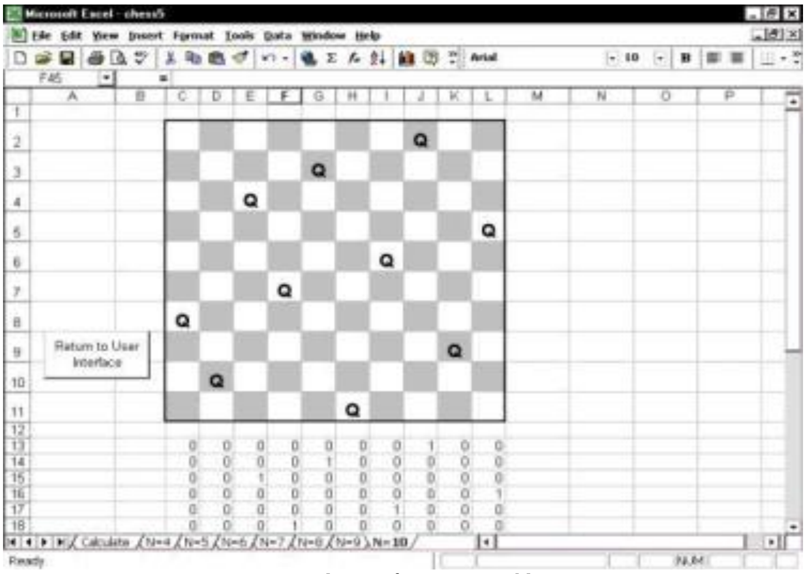

Fig 1. Solution of 10 queen problem

II LITERATURE REVIEW

A .PARTICLE SWARM OPTIMIZATION FOR N-QUEENS PROBLEM

In this paper , one of the most interesting toy problem of n queen problem is solved by using particle swarm optimization algorithm. In chess board, queen is the most powerful one which has various attacking options. In this paper, PSO algorithm provides optimal solution for tis problem. The computational approach is implemented over reasonable size of chessboard and the quality results revealed the positive impact and capability of adapted technique. The script was written on Intel® Core(TM) i3 CPU, M 350 @ 2.27 GHz, 2.0 GB RAM. The Python language version 2.6 was chosen to compile the program coding. It uses ring topology,20 neighborhood size and 1000 maximum evaluation. The paper examined the Particle Swarm Optimization (PSO) approach that set up the stunning results at the end, and also provided evidence to be very capable of producing feasible/optimal solutions over N-Queens Problem. The key objective of the study is to reveal the effectiveness and efficiency of PSO so it can be projected to other identical real-world problems

B. SWARM INTELLIGENCE FOR PERMUTATION OPTIMIZATION: A CASE STUDY OF N-QUEENS PROBLEM

This paper uses a modified Particle Swarm Optimizer which deals with permutation problems. Particles are defined as permutations of a group of unique values. Velocity updates are redefined based on the similarity of two particles. Particles change their permutations with a random rate defined by their velocities. A mutation factor is introduced to prevent the current pBest from becoming stuck at local minima. In this paper, the maximum velocity is set as range of permutation. Each parameter combination was run 100 times and the results represent the mean number of function evaluations. And also, it provides better results than genetic algorithm in performance. The comparison table is present in this paper. From the results, it can be seen that PSO successfully finds a solution of the n-queens problems in a short time

C. N-QUEEN PROBLEM USING DYNAMIC LOAD DISTRIBUTION

In general no of serial algorithms like brute force algorithms, genetic algorithms, heuristic algorithms are used to solve the n queen problem. But in this paper a parallel solution is provided using dynamic load balancing method. The objective is to find all solutions to the N-Queen problem on a distributed memory cluster using MPI. In this paper, the experiments are taken for board size 12 using 7 processors and board size 15 using 8 processors. The speed up is dependent both on the number of slots (keeping the board Size constant) and also its improvements over the serial algorithms. With respect to the serial program, the scaling is linear. Due to high volumes of data transfer, the scaling is not perfectly linear. Furthermore, the scaling is almost perfectly linear on the number of processors; at least when double the number of processors

D. COMPARISON OF HEURISTIC ALGORITHMS FOR THE N-QUEEN PROBLEM

In this paper the n queen problem which most interesting one is solved by using different heuristics algorithms like simulated annealing, tabu search and genetic algorithms. The methods are compared very efficiently. And also the test results are compared by performance.

E. EFFICIENCY OF PARALLEL GENETIC ALGORITHM FOR SOLVING N-QUEENS PROBLEM ON MULTICOMPUTER PLATFORM

In this paper the n queen problem is solved by using parallel genetic algorithm. This is based on the multicomputer platform. This paper deals with both flat (pure MPI) and hybrid (MPI+OpenMP) programming models. Analysis of scalability and performance profiling are discussed in respect to the workload and the size of the board

F. PRESENTING A NEW METHOD BASED ON COOPERATIVE PSO TO SOLVE PERMUTATION PROBLEMS: A CASE STUDY OF N-QUEEN PROBLEM

PSO is one of the optimal algorithms to provide good result to complex problems which has complex search space and nondeterministic answers. This paper deals with a new cooperative PSO method to solve the n queen problem. This method is more efficient than standard PSO. In this PSO, every particle is divided into sub particles and applied for n queen problem. A comparatively study is made to standard PSO and other Meta heuristic methods.

G. SOLUTION OF N-QUEEN PROBLEM USING ACO

Swarm intelligence algorithms are very efficient to handle a lot of NP-hard problems. In this paper ACO(Ant colony optimization) which solves a lot of NP-hard problems is applied to solve the n queen problem. First this ACO algorithm is implemented for 8 queens only. Then it is extended to solve n queen because of its easy implementation. This paper deals the problem , backround, complexity of n queen etc. It provides a comparatively study by using graphs.

H. N – QUEEN PROBLEM IN CELL: PROJECT REPORT

In this paper the author discussed about the n queen problem, size of the problem and possible heuristics. This paper compares the results of speed up of CBE-6SPUs and Intel core 2 Duo.It discussed the problem when we increase the no of queens

I. GENETIC ALGORITHMS FOR THE N-QUEENS PROBLEM

Various problem solving methods are used to solve the NP-Hard problems. At first the local search algorithms are used to solve these types of problems. These are effective depends upon the polynomial time. But in this paper the genetic algorithm is applied for this n queen problem. This paper presents a variety of methods to solve the problem and all are compared with one another. This paper proves that the genetic algorithm is easier than other local search algorithms.

J. A DYNAMIC PROGRAMMING SOLUTION TO THE N-QUEENS PROBLEM

This paper solves the n queen problem through dynamic programming. The authors describe a simple $O(f(n)8^n$ solution to this n queen problem. Here f(n) is a low order polynomial. The paper discuss about the combinatorial problems, algorithm designs and some search algorithms. This paper proves that this dynamic solution is better results than backtracking method

K. GENETIC ALGORITHM VERSUS PARTICLE SWARM OPTIMIZATION IN N-QUEEN PROBLEM

The n queen problem is solved by using different algorithms. In that algorithms, PSO(Particle Swarm Optimization) and GA(Genetic Algorithm) are very efficient to solve the n queen problem. This paper compares both the algorithms with its parameters like mutation, maximum population, rows, columns, no of iterations etc. This paper discusses which algorithm is best depending upon the performance. Although, both the algorithms are similar in performance, they are differs in search spaces.

L. SOLVING N-QUEEN PROBLEM USING GRAVITATIONAL SEARCH ALGORITHM

No of algorithms are available to solve the n queen problem. This paper discuss about the different heuristics methods. In that a new method named GSA(Gravitational Search Algorithm) is deeply discussed and solve the n queen problem. This paper tells about how GSA is efficiently solve this problem. And the experimental results are compares with Genetic algorithms results. This paper proves that the new GSA algorithm gives better results than Genetic algorithm.

M. A DNA STICKER ALGORITHM FOR SOLVING N-QUEEN PROBLEM

This paper solves the n queen problem through DNA sticker algorithm. This paper formulates N queen problem by DNA sequences. It provides all possible solutions by parallel computations. And also this paper proves that DNA algorithm can apply to all kind of NP hard problems master process from most tasks GPGA is not suitable for massive parallel processing, but it shows increase in performance for a small number of parallel-processing units. The experiment is done by using C programming and results are shown in this paper.

N. SOLVING N-QUEEN PROBLEM USING GLOBAL PARALLEL GENETIC ALGORITHM

This paper deals the n queen problem by using global parallel genetic algorithms. Genetic algorithms (GAs) are powerful heuristic methods, capable of efficiently searching large spaces of possible solutions. However, due to intense computations performed by GAs, some form of parallelization is desirable to increase performance.This algorithms improves the speed of genetic algorithms. This paper shows the results of genetic algorithms and this new algorithm. 3-way tournament selection enabled slaves to run simultaneous selections and crossovers, freeing master process from most tasks GPGA is not suitable for massive parallel processing, but it shows increase in performance for a small number of parallel-processing units. The experiment is done by using C programming and results are shown in this paper.

III. CONCLUSION

This paper deeply discusses about one of the NP hard problem n queen problem and the solving methods. There are lot of algorithms are used to solve this problem including swarm intelligence algorithms. All the algorithms provide a good result than the traditional methods like backtracking. The performance of algorithms are differ depend upon the size of the board and number of queens. From results GA, PSO, Tabu search, ACO are provide good results. This paper explains the solving methods to n queen problem. To make a comparison study of these algorithms with common parameters are very useful to better understand

REFERENCES

[1] Aftab Ahmed,Attique Shah, Kamran, Ali Sani and Abdul Hussain Shah Bukhari, Particle Swarm Optimization for N-Queen Problem, Journal of Advanced Computer Science and Technology, 1(2)(2012) 57-63 @ Science Publishing Corporation

[2] Xiaohui Hu'-' Russell C. Eberhart' Yuhui Shi', Swarm Intelligence for Permutation Optimization: A Case Study of n-Queens Problem,

[3] Ramnik Arora, N-Queen Problem Using Dynamic Load Distribution (Y5365) MTH698: Parallel Numerical Algorithms, http://home.iitk.ac.in/~ramnik/Mth698.html April 15, 2009

[4] Ivica Martinjak, Marin Golub ,Comparison of Heuristic Algorithms for the N-Queen Problem (Citations: 5), International Conference on Information Technology Interfaces - ITI , pp. 759-764, 2007

[5] Milena Lazarova , Efficiency of parallel genetic algorithm for solving N-queens problem on multicomputer Platform,EC'08 Proceedings of the 9th WSEAS International Conference on Evolutionary Computing, Pages 51-56

[6] Amooshahi, A. Islamic Azad Univ., Doroud, Iran Joudaki, M. Imani, M. Mazhari, N Presenting a new method based on cooperative PSO to solve permutation problems: A case study of n-queen problem, Electronics Computer Technology (IC EC T), 2011 3rd International Conference on 8-10 April 2011,Volume: 4, Page(s): 218 – 222

[7] Khan, S. Bilal, M,Sharif, M, Sajid, M, Baig, R, Solution of n-Queen problem using ACO, Multitopic Conference, 2009. INMIC 2009. IEEE 13th International Conference: 14-15 Dec. 2009, page(s): 1 – 5

[9] Josef Hynek, Genetic Algorithms for the N-Queens Problem

[10] Xiaohui Hu, Russell C. Eberhart, Yuhui Shi, Swarm intelligence for permutation optimization: a case study of n-queens problem,IEEE Conference, IEEE Swarm Intelligence Symposium - SIS , 2003

[11] Azhar W. Hammad, Dr. Ban N. Thannoon, GENETIC ALGORITHM VERSUS PARTICLE SWARM OPTIMIZATION IN N-QUEEN PROBLEM

[12] Homayun Motameni, Seyyed Hossein bozorgi, mohammad ali shaban nezhad, Golnaz Berenjian, Behnam Barzegar Solving N-Queen problem using Gravitational Search Algorithm,. Life Science Journal 2013;10(1)

