N-Queens Problem Optimization Using Various Memetic Algorithms

[1]Er. Vishal Khanna, [2]Er. Abhishek Bhardwaj, [3]Er. Sarvesh Chopra
[1]Research Scholar, [2]Asstt. Prof, [3]Asstt. Prof
[1]CSE, [1]CT Group of Institute, Jalandhar, India

***Abstract*— NP hard problems like N-Queens problem are non-polynomial time problems. In this research study, we use the Simulated Annealing local search based MA (SALSMA), Genetic Algorithm(GA) and Hill-Climbing local search based MA (HCLSMA) to optimize N-Queens problem and make complexity analysis on the parameters viz. optimal solutions, time and convergence rate with respect to number of iterations. The MA is a hybrid algorithm, being a combination of the Genetic Algorithm (GA) and a local search algorithm. The performance of the MA is found to be superior to that of a solitary algorithm like GA. The MA solves the N-Queens in two stages. In the first stage, the randomly generated solutions are evolved till they become feasible (i.e., the hard constraints are satisfied) and in the second stage, these solutions are further evolved so as to minimize the violations of the soft constraints. In the final stage, the MA produces optimal solutions in which the hard as well as the soft constraints are completely satisfied.**

***Index Terms*— Queens, N-Queens, 8 Queens, Genetic Algorithm, Chromosome, Mutation, Selection, Crossover, Recombination, GA, MA, HCLSMA, SALSMA, Hill-Climbing, Simulated Annealing.**

I. Introduction

Problems with non-deterministic solutions that run in polynomial time are called NP-class problems. Because of their high complexity (e.g. O(2n) or O(n!)) they cannot be solved in a realistic timeframe using deterministic techniques. To solve these problems in a reasonable amount of time, heuristic methods must be used.
N-Queens problem which is to place the ‘n’ numbers of Queens on a chess board so that neighbor Queens cannot contradict each other vertically, horizontally and diagonally. Some of the recent AI techniques include the use of simulated annealing, hill-climbing, genetic algorithm, co-operative genetic algorithm, artificial immune system and different versions of evolutionary algorithms. The conventional GA does not yield satisfactory solutions. Therefore, we believe that a hybrid methodology involving an Evolutionary Algorithm that finds several feasible solutions and a Local Search exploiting the inherent knowledge of the problem to optimize the intermediate feasible solutions is an appropriate tool to tackle this highly complex problem. The hybridization of evolutionary algorithms (EAs) with other techniques can greatly improve the efficiency of search. EAs hybridized with local search techniques are named as Memetic Algorithms .A common approach is to apply the local search to the GA population after crossover and mutation, with the aim of exploiting the best search regions. An important aspect concerning MAs is the trade-off between the exploration abilities of the EA and the exploitation abilities of the local search technique.

In this study, we use a Memetic Algorithm (MA) to solve the complex n-Queens problem. The MA algorithm is a hybrid of the Genetic Algorithm (GA) and Local Search (LS). The GA follows a simple coding scheme and after the recombination operations, LS is applied using problem-specific knowledge. A number of random shift schedules are generated. Penalties are imposed for the violation of the hard as well as the soft constraints of the shift schedules. The MA solves the problem in two phases. In the first phase, it tries to resolve all the violations of the hard constraints. This leads to feasible solutions. In the second phase, MA works with the feasible solution and further evolves them eliminating, or at least minimizing the soft constraints. The result is optimal solutions satisfying the n-Queens problem.
.

II. N-Queens Problem (NQP)

The classic combinatorial problem is to place N-Queens on a chessboard so that no two attack each other. This problem can be generalized as placing ‘n’ non attacking queens on an N × N chessboard. Since each queen must be on a different row and column, we can assume that queen ‘i’ is placed in i^{th} column. All solutions to the NQP can therefore be represented as n-tuples (q1, q2, …, q^n) that are permutations of an n-tuple (1, 2, 3, …, n). Position of a number in the tuple represents queen's column position, while its value represents queen's row position (counting from the bottom) using this representation, the solution space where two of the constraints (row and column conflicts) are already satisfied should be searched in order to eliminate the diagonal conflicts. Complexity of this problem is O(n!).

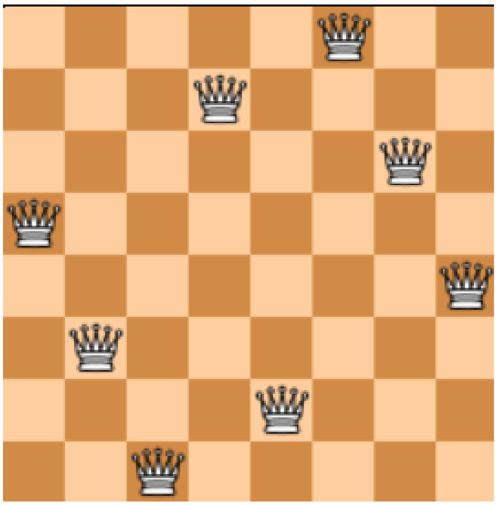

Figure 1 Chess board of size 8*8 with 8 queens. [2][4]

N-Queens dates back to the 19th century (studied by Gauss) Classical combinatorial problem, widely used as a benchmark because of its simple and regular structure Problem involves placing. Benchmark code versions include finding the first solution and finding all solutions. Input for this System: A positive integer ‘n’. Task for this system: place ‘n’ Queens on an n by n chessboard so that no two Queens attack each other (on same row, column, diagonal), or report that this is impossible. Solving particular problem for the different values of n=1, 2, 3, 4…n. [1]

III. Optimization

Optimization is the process of identifying the best solution among a set of alternatives. Single objective optimization employs a single criterion for identifying the best solution among a set of alternatives. [1]

1. Read the problem.
2. Reread the problem.
3. Draw a picture or graph if appropriate.
4. Identify the given information – What are the variables? – What are the constants? – Are there any constraints? – Label the graph or picture.
5. What quantity needs to be maximized or minimized?
6. Find an appropriate equation for what needs to be maximized or minimized, and reduce it to one variable.
7. Find the derivative and critical points for your equation.
8. Test your critical points and end points (where appropriate).
9. Reread the question and make sure you have answered what was asked.

IV. N-Queens optimization using Memetic Algorithm

The Memetic Algorithm (MA) we have used to solve the above QMC n-Queens consists of the Genetic Algorithm combined with a local search. The MA flowchart is shown in Fig. 1. It describes in detail each of the steps of the MA. Steps of Memetic Algorithm are given below:

1. **Random Generation:** In order to generate a power-law distribution $P(x)$ from a uniform distribution $P(y)$, write $P(x) = C x^n$ for $x \in [x_0, x_1]$. Then normalization gives

$$\int_{x_0}^{x_1} P(x)\, dx = C \frac{\left[x^{n+1}\right]_{x_0}^{x_1}}{n+1} = 1,$$

$$C = \frac{n+1}{x_1^{n+1} - x_0^{n+1}}.$$

Let Y be a uniformly distributed variate on $[0, 1]$. Then

$$D(x) = \int_{x_0}^{x} P(x')\,dx'$$

$$= C\int_{x_0}^{x} x'^{n}\,dx'$$

$$= \frac{C}{n+1}\left(x^{n+1} - x_0^{n+1}\right)$$

$$\equiv y,$$

and the variate given by

$$X = \left(\frac{n+1}{C}\,y + x_0^{n+1}\right)^{1/(n+1)}$$

$$= \left[\left(x_1^{n+1} - x_0^{n+1}\right)y + x_0^{n+1}\right]^{1/(n+1)}$$

is distributed as $P(x)$.

2. **Fitness &Selection**
3. **Crossover**
4. **Mutations**
5. **Local Search**

V. Applications of N-Queens

These are given below:

1. VLSI Testing.
2. Traffic control.
3. Deadlock Prevention.
4. Image Processing.
5. Motion Estimation.
6. Register Allocation. [6]

VI. Results

The below graphs show the results of applying the MA for the optimization of N-Queens problem. In this research paper the tables show that MA gives best results in comparison of GA in terms of space complexity as the number of iterations and number of queen's increases. In terms of optimal solutions HCLSMA and SALSMA gives best solutions than GA. In comparison between HCLSMA and SALSMA on optimal solutions HCLSMA provides best optimal solutions. Then convergence rate checked between these three techniques HCLSMA has best convergence rate than SALSMA and GA on different number of iterations and number of queens. Then time complexity calculated between three approaches HCLSMA has less time complexity than SALSMA and GA for different number of iterations and number of queens.

Table 1 Optimal solutions based on the number of Queens (n) and number of iterations=5.

No of Queens	GA	SALSMA	HCLSMA
8	5	5	5
16	2	3	5
24	0	1	5
32	0	1	5
40	0	0	5

Table 2 Optimal solutions based on the number of Queens (n) and number of iterations=8.

No of Queens	GA	SALSMA	HCLSMA
8	8	8	8
16	1	5	8
24	0	2	8
32	0	1	8
40	0	0	8

Table 3 Optimal solutions based on the number of Queens (n) and number of iterations=10.

No of Queens	GA	SALSMA	HCLSMA
8	10	10	10
16	2	5	10
24	0	3	10
32	0	1	10
40	0	0	10

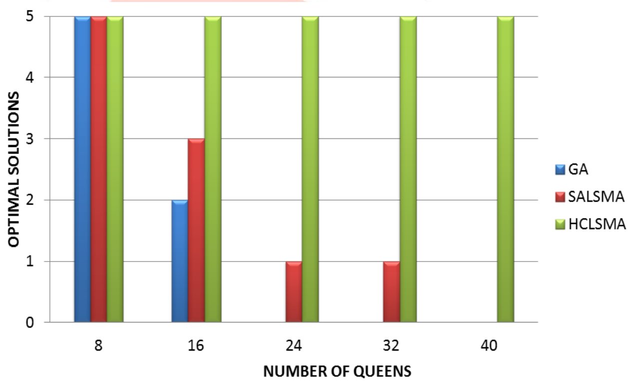

Figure 2 Optimal solutions based on the number of Queens (n) and number of iterations=5.

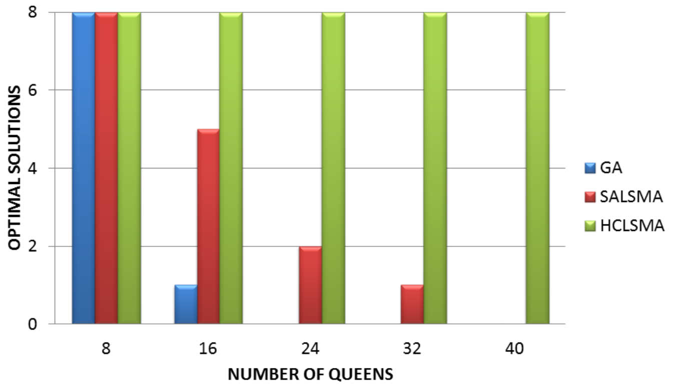

Figure 3 Optimal solutions based on the number of Queens (n) and number of iterations=8.

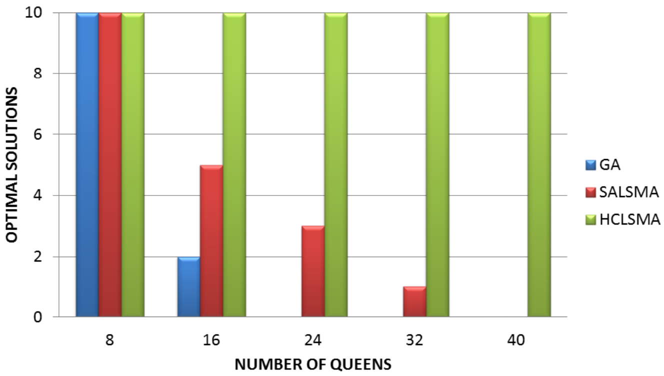

Figure 4 Optimal solutions based on the number of Queens (n) and number of iterations=10.

Table 4 Convergence rate based on the number of Queens (n) and number of iterations=5.

No of Queens	GA	SALSMA	HCLSMA
8	100	100	100
16	40	60	100
24	0	20	100
32	0	20	100
40	0	0	100

Table 5 Convergence rate based on the number of Queens (n) and number of iterations=8.

No of Queens	GA	HCLSMA	HCLSMA2
8	100	100	100
16	12.5	62.5	100
24	0	25	100
32	0	12.5	100
40	0	0	100

Table 6 Convergence rate based on the number of Queens (n) and number of iterations=10.

No of Queens	GA	SALSMA	HCLSMA
8	100	100	100
16	20	50	100
24	0	30	100
32	0	10	100
40	0	0	100

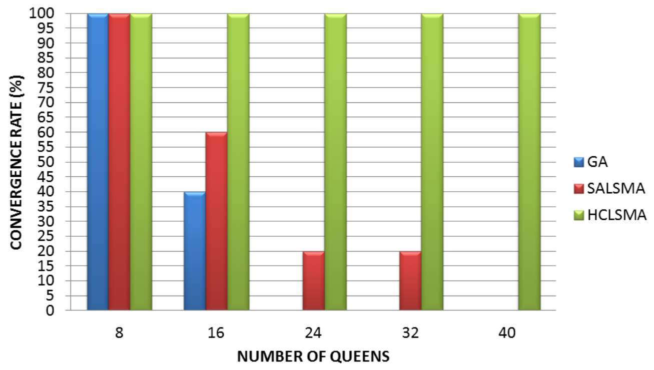

Figure 5 Convergence rate based on the number of Queens (n) and number of iterations=5.

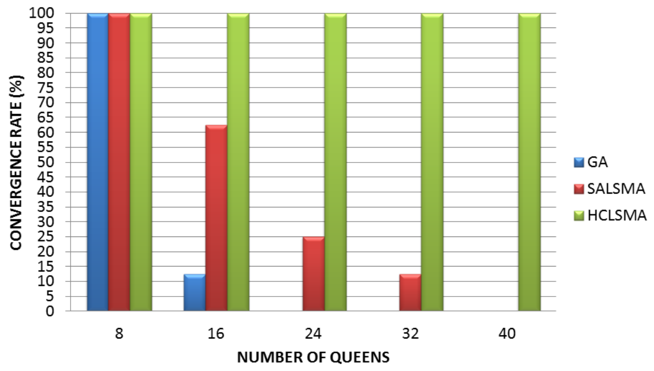

Figure 6 Convergence rate based on the number of Queens (n) and number of iterations=8.

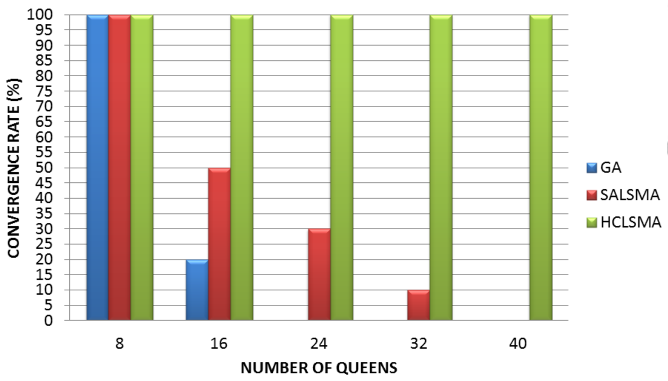

Figure 7 Convergence rate based on the number of Queens (n) and number of iterations=10.

Table 7 Time complexity based on the number of Queens (n) and number of iterations=5.

S.No.	No of Queens (n)	HCLSMA (ms)	SALSMA (ms)	GA (ms)
1	8	3420	3481	3606
2	16	46165	46201	46236
3	24	192104	192131	192157
4	32	187950	187987	188023
5	40	1577690	1577719	1577752

Table 8 Time complexity based on the number of Queens (n) and number of iterations=8.

S.No.	No of Queens (n)	HCLSMA (ms)	SALSMA (ms)	GA (ms)
1	8	3565	3595	3627
2	16	101821	101865	101903
3	24	160832	160867	160903
4	32	458919	458935	458951
5	40	1486878	1486913	1486949

Table 9 Time complexity based on the number of Queens (n) and number of iterations=10.

S.No.	No of Queens (n)	HCLSMA (ms)	SALSMA (ms)	GA (ms)
1	8	3607	3633	3660
2	16	121636	121674	121709
3	24	296828	296863	296900
4	32	756203	756238	756274
5	40	2550705	2550722	2550741

VII. Conclusion and Future-Work

The N-Queens Problem (NQP), like the well-0known Travelling Salesman Problem (TSP), is an NP-hard problem. Some studies show that the straight forward implementation of the Genetic Algorithm is incapable of obtaining a satisfactory solution. Therefore, we believe that a hybrid methodology involving an Evolutionary Algorithm that finds several feasible solutions and a Local Search exploiting the inherent knowledge of the problem to optimize the intermediate feasible solutions is an appropriate tool to tackle this highly complex problem. With this intuition, we have a Memetic Algorithm (GA + local search) to solve the NQP. The hybrid MA solves the NQP in two phases. In the first phase it tries to search for the Queens's shift patterns that do not violate the hard constraints. HCLSMA gives better results than GA for optimal solutions, time and convergence rate. Future work includes the parameters like conflict minimization and revolution rate for the comparison of HCLSMA, GA and SALSMA.

Reference

[1] Er. Vishal Khanna, Er. Sarvesh Chopra, Review on N-Queen Optimization Using Tuned Hybrid Technique, International Journal of Engineering Sciences & Research Technology (Thomson Reuters), pp.62-68, Vol.6, Issue.2, 2017.

[2] Yangming Zhou, Jin-Kao Hao, and B´eatrice Duval, Opposition-based Memetic Search for the Maximum Diversity Problem, IEEE Transactions on Evolutionary Computation, pp.1-15, 2017.

[3] Adam Santos, Reginaldo Santos, Moisés Silva, Eloi Figueiredo, Claudomiro Sales, and João C. W. A. Costa, A Global Expectation–Maximization Approach Based on Memetic Algorithm for Vibration-Based Structural Damage Detection, Ieee Transactions On Instrumentation And Measurement, pp.1-10, 2017

[4] Yifeng Zeng, Xuefeng Chen, Yew-Soon Ong, Jing Tang and Yanping Xiang, Structured Memetic Automation for Online Human-like Social Behavior Learning, Ieee Transactions On Evolutionary Computation, pp.1-14, 2017.

[5] A.H. Beg, Md Zahidul Islam, Advantages and Limitations of Genetic Algorithms for Clustering Records, 2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA), pp.2478-2483, 2016.

[6] Sarkan Guldal and Veronica Baugh, "N-Queens Solving Algorithm by Sets and Backtracking", IEEE Southeast Conference, pp.125-129, 2016.

[7] B Documentaries, "Full Solution of N-Queens Problem O Reilly", http://oreillynQueensproblem.blogspot.in, 3-Sept-2016.

[8] Tad Gonsalves and Kohei Kuwata, "Memetic Algorithm For The Nurse Scheduling Problem", International Journal of Artificial Intelligence and Application, pp. 43-52, 2015.

[9] Soham Mukherjee, Santanu Datta, Pramit Brata Chanda and Pratik Pathak, Comparative Study Of Different Algorithms To Solve N Queens Problem, International Journal in Foundations of Computer Science & Technology, Vol.5, Issue.2, pp.15-27, March 2015.

[10] Amarbir Singh and Sandeep Singh Dhillon, "A Comparative Study of Algorithms for N-Queens Problem", International Journal of Advance Foundation and Research in Science and Engineering , Vol.1, Special Issue, pp.1-4, 2015.

[11] Soham Mukherjee, Santanu Datta, Pramit Brata Chanda and Pratik Pathak, "Comparative Study of Different Algorithms To Solve N-Queens Problem", International Journal of Foundations of Computer Science and Technology, Vol.5, Issue.2, pp.15-27, 2015.

[12] Ahmed S. Farhan , Wadhan Z. Tareq and Fouad H. Awad, "Solving N-Queens Problem using Genetic Algorithm ", International Journal of Computer Applications, Vol.122, Issue.12, pp.11-14, 2015.

[13] Vikas Thada and Shivali Dhaka, "Performance Analysis of N-Queens Problem using Backtracking Algorithm Techniques ", International Journal of Computer Applications, Vol.102, Issue.7, pp. 26-29, 2014.

[14] Ellips Masehian, Hossein Akbaripour and Nasrin Mohabbati-Kalejahi,"Solving the n-Queens Problem Using a Tuned Hybrid Imperialist Competitive Algorithm ", The International Arab Journal of Information Technology, Vol.11, Issue.6, pp.550-559, 2014.

[15] Belal Al-Khateeb, Wadhah Z. Tareq, Solving 8-Queens Problem by Using Genetic Algorithms, Simulated Annealing and Randomization Method, 2013 Sixth International Conference on Developments in eSystems Engineering, pp.187, 2013.

[16] Vishal Kesri and Manoj Kumar Mishra, "A new approach to solve n-Queens problem based on series", International Journal of Engineering, Research and Applications, Vol.3, Issue.3, pp.1349-1349, 2013.

[17] Ram Gopal Sharma and Bright Keswani, "Implementation of N-Queens Puzzle using Meta-Heuristic Algorithm (Cuckoo Search)" International Journal of Latest Trends in Engineering and Technology, Vol. 2, Issue. 3, pp. 343-347, 2013.

[18] S.Pothumani, "Solving N-Queens Problem using Various Algorithms-A Survey", International Journal of Advance Research in Computer Science and Software Engineering, Vol. 3, Issue. 2, pp. 247-250, 2013.

[19] Kenekayoro Patrick, Comparison of simulated annealing and hill climbing in the course timetabling problem, African Journal of Mathematics and Computer Science Research, Vol. 5, Issue.11, pp.176-178, September 2012.

[20] Farhad Soleimanian, Bahareh Seyyedi and Golriz Feyziour, "A New Solution for N-Queens Problem using Blind Approaches: DFS and BFS Algorithms", International Journal of Computer Applications, Vol.53, Issue.1, pp.45-48, 2012.

[21] Vishal Kesri, Vaibhav Kesri and Prasant Ku. Pattnaik, "A Unique Solution for N-Queens Problem", International Journal of Computer Applications, Vol.43, Issue.12, pp.13-19, 2012.

[22] Baolei Gu," Research and Realization of N-Queens Problem Based on the Logic Language Prolog", Springer Computational Intelligence and Intelligent System, Vol.4, Issue.1, pp. 50-56, 2012.

[23] Aftab Ahmed, Attique Shah, Kamran Ali Sani and Abdul Hussain Shah Bukhari," International Journal of Advance Computer Science and Technology" Vol.1, Issue.2, pp. 57-63, 2012.

[24] Jun Zhang and Zili Zhang, "An Algebraic method for the n-Queens Problems based on Permutation Operation Group", International Journal of Computer Networks and Information Security, Vol.3, pp.19-25, 2011.

[25] Jordan Bell and Brett Stevens, "A Survey of Known results and research areas for n-Queens" Discrete Mathematics Science Direct, Vol.309, Issue.1, pp.1-31, 2008.

[26] H. Ahrabian, A. Mirzaei and A.Nowzari-Dalini," A DNA Sticker Algorithm for Solving N-Queens Problem", International Journal of Computer Science and Applications, Vol.5, Issue.3, pp.12-22, 2006.

[27] Chung-Neng Wang, Shin-Wei Yang, Chi-Min Liu and Tihao Chiang, "IEEE Transactions on Circuits and System for Video Technology", Vol.14, Issue.4, pp.429-440, 2004.

[28] Marko Bozikovic, Marin Golub and Leo Budin, "Solving N-Queens Problem Using Global Parallel Genetic Algorithm", European Conference Ljubljana Slovenia, pp.11-17, 2003.

[29] Roc Sosic and Jun Gu, "Fast Search Algorithms for the N-Queens Problem", IEEE Transactions on Systems, Man, and Cybernetics, Vol.21, Issue.6, pp.1572-1576, 1991.

[30] Rok Sosic and Jun Gu, “A Polynomial Time Algorithm for N-Queens Problem”, Special Interest Group on Artificial Intelligence, vol.1, Issue.3, pp.7-14, 1990.