IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

Analysis of GA Performance on Its Various Parameters for Solving Travelling Salesman NP-Hard Problem

R.K. Singh[1*], V.K. Panchal[2], B.K. Singh[3]

[1*]Department of Computer Science, Al-Falah University, Faridabad, Haryana, India
[2]Department of Computer Science, Al- Falah University, Faridabad, Haryana, India
[3]Department of Computer Science, Satyug Darshan Institute of Engineering & Technology, Faridabad, Haryana, India

[*]Corresponding Author: rajeshsingh22@gmail.com, Tel.: +91-9654040845

Available online at: www.ijcseonline.org

Accepted: 23/Jan/2019, Published: 31/Jan/2019

***Abstract*—**Genetic Algorithm (GA) is a well-known heuristic algorithm inspired by theory of adaptation. GA is applicable to solve many problems of science and engineering. GA performs its operations such as selection, reproduction and mutation to solve a problem. The genetic parameters such as population size, cross over rate and mutation rate control the performance and effectiveness of GA to solve a problem. In this paper, GA is applied to solve Traveling Salesman Problem (TSP). TSP problem is an optimization problem and it is a member of the set NP-Hard problems. In this paper the performance of GA on its parameters is analyzed to solve TSP problem.

Keywords—*NP-Complete, Genetic Algorithm, Genetic Parameters*

I. INTRODUCTION

Travelling Salesman Problem is a linear optimization problem. In TSP problem a salesman has to visit n number of cities. The constraint is that the salesman has to visit each and every city exactly once and the distance travelled should be minimum. The salesman has to start and stop the journey at the same city. This problem is represented by a fully connected weighted graph having n number of vertices. The distance between all the cities is stored in the form of a matrix having rows and columns equal to number of cities n. As the number of cities increases in a TSP instance the complexity of the algorithm to solve TSP problem also increases. The complexity of algorithm used to solve TSP is factorial of n. So it is very difficult to solve TSP problem using normal algorithms. Genetic Algorithm GA is an optimization technique which is inspired by theory of adaptation. The concept of genetic algorithm is taken form medical sciences. In nature, individuals in a population of same kind of species survive by following 'survival the fittest' rule. While applying GA to solve a problem, a population of individuals is generated. The individuals of the population are called chromosomes. The quality of the solution, represented by a chromosome is called fitness of the chromosome. In case of TSP problem, the distance travelled by the salesman is used to calculate the fitness value. Larger is the distance travelled, lower is the fitness. Then other genetic parameters such as cross over and mutation are applied on the TSP problem. Figure 1 is showing a flowchart of the genetic algorithm.

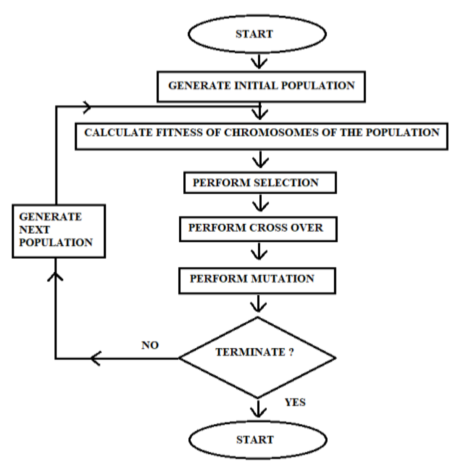

Figure 1. Flowchart of genetic algorithm.

In this paper, the performance of GA to solve TSP on different GA parameters is analysed. The organization of paper is as follows. The next section illustrates the recent research in this area. Next section describes the proposed work which is followed by analysis of results and conclusion.

II. Related Work

Kusum Gupta and Sonal Sharma applied genetic algorithm for solving TSP problem [1]. A new variation of cross over operator was applied and its performance was verified. Basima Hani Hasan et. al [2] analyze the performance of GA on solving NP problems. The performance of GA on mutation operator was analysed. M. A. H. Akhand et.al. applied PSO algorithm on TSP problem [3]. Hala ElAarag and Sam Romano applied animation of Traveling Salesman Problem. Shigang Cui, Shaolong Han [5] Qiuying Bai et.al. proposed an improved hybrid GA to solve TSP [6]. The improved GA produce better solutions of TSP problem in less time. Mud-Armeen Munlin and Mana Anantathanavit [7] apply K-mean algorithm and PSO algorithm to solve TSP. The results of hybrid GA was found better, Muhao Chen et.al. applied Virtual Instrument Technology with GA to solve TSP problem [8]. [9] Jiri Stastný et.al. applied graph based algorithm to solve TSP. The graphs based algorithm solve TSP instances in reasonable time. [10] MUDASSIR KHALIL et.al. proposed a new algorithm to solve TSP problem. The new algorithm solved TSP efficiently. [11] Hao Qian et.al. proposed Discrete Fruit Fly Optimization Algorithm to solve TSP. In recent years a lot of research work has been done to solve TSP problem by various methods [12] [13] [14]. M. Bandyopadhyay and S. Chattopadhyay applied genetic algorithm over different PID tuning methods for controlling Servo Systems [15]. The paper concluded that GA method of tuning better as compared to other conventional methods. Rajeev Ranjan and P.J. Pawar applied real coded genetic algorithm for assembly line balancing [16]. Genetic algorithm provide better solutions for assembly line balancing in manufacturing process as compared to other traditional methods. But there is a need to improve the performance of GA by applying right combination of GA parameters such as population size, cross over rate and mutation rate. In this paper the performance of GA on various parameters is analysed.

III. Methodology

In this work, the genetic algorithm is applied to solve Traveling Salesman Problem. The genetic algorithm to solve Traveling Salesman Problem is illustrated in Algorithm-1

Algorithm-1 Genetic Algorithm to Solve TSP

1. Encode TSP problem into genetic form
2. Generate initial population of chromosomes using permutation encoding
3. For I =1 to IterationCount Do
4. Calculate fitness of every chromosome of the population
5. Perform selection to select chromosomes for cross over
6. Perform one-point cross over
7. Add new children in the population
8. Perform mutation
9. Generate next population
10. End Do
11. Return best chromosome of the population as solution of TSP

The Algorithm -1 is implemented and executed for different values of three genetic algorithm parameters namely population-count, cross-over-rate and mutation-rate. The population count is kept from 20 chromosomes to 200 chromosomes in the steps of 20. The cross over rate is kept 10%, 20% ----- 100%. The mutation rate is kept 0.01, 0.02, 0.03 ---- 0.10. The given genetic algorithm is executed for 100 iterations and results are evaluated. Next section illustrated the results of genetic algorithm for different GA parameters.

IV. Results and Discussion

The Algorithm-1 is implemented for different value of population size and keeping other GA parameters as fixed. The population size is varied from 20 chromosomes to 100 chromosomes. Cross over rate is kept 40%. Mutation rate is kept 2%(0.02). One-point cross over is used. Random selection operator is applied in selection operation.

Table 2 is showing results of GA with parameters shown in Table-1. From the results of Table-2 it is clear that GA performs better for higher values of population size. The path length of 597 is achieved for population size 160. However, for high values of population size, the algorithm takes more time.

Table 1. GA parameters for different values of population count

GA Parameter	Value
Encoding Scheme	Permutation Encoding
Population Size	Varied from 20 – 100 in steps of 20
Iteration Count	100
Selection Technique	Random Selection
Cross Over Rate	40%
Cross Over Type	One point cross over
Mutation Rate	2%

The Algorithm-1 is implemented for different value of cross over rate and keeping other GA parameters as fixed. The cross over rate is varied from 10% to 100%. Population size is kept 100. Mutation rate is kept 2%(0.02). One-point cross over is used. Random selection operator is applied in selection operation. Table 3 is showing various GA parameters.

Table 4 is showing results of GA with parameters shown in Table-3. From the results of Table-4 it is clear that GA performs better for higher values of cross over rate. The path length of 597 is achieved for cross over rate 80% and 100% which is smallest as compared to other results. However, for high values of cross over rate, the algorithm takes more than 18000 ms to execute.

The Algorithm-1 is implemented for different value of mutation rate and keeping other GA parameters as fixed. The mutation rate is varied from 1% to 10%. Population size is kept 100. Cross over rate is kept 40%. One-point cross over is used. Random selection operator is applied in selection operation. Table 5 is showing various GA parameters

Table 6 is showing results of GA with parameters shown in Table-5. From the results of Table-6 it is clear that GA performs better for higher values of mutation rate. The path length of 562 is achieved for mutation rate 10% which is smallest as compared to other results. However, for high values of mutation rate, the algorithm takes more than 7900 ms to execute.

Table 2. Results of Algorithm -1 implementation for GA parameters shown in Table-1.

Sr. No.	Iterations	Population Size	Cross Over rate	Mutation Rate	Best Chromosome	Time (ms)
1	100	20	40%	2%	803	2707
2	100	40	40%	2%	757	3253
3	100	60	40%	2%	735	4759
4	100	80	40%	2%	725	6308
5	100	100	40%	2%	672	7943
6	100	120	40%	2%	709	9178
7	100	140	40%	2%	721	10477
8	100	160	40%	2%	597	12036
9	100	180	40%	2%	743	13306
10	100	200	40%	2%	711	14788

. Table 3. GA parameters for different values of cross over rate

GA Parameter	Value
Encoding Scheme	Permutation Encoding
Population Size	100
Iteration Count	100
Selection Technique	Random Selection
Cross Over Rate	Varied from 10% to 100% in steps of 10%.
Cross Over Type	One point cross over
Mutation Rate	2%

Table 4. Results of Algorithm -1 implementation for GA parameters shown in Table-3.

Sr. No.	Iterations	Population Size	Cross Over rate	Mutation Rate	Best Chromosome	Time (ms)
1	100	100	10%	2%	761	3860
2	100	100	20%	2%	703	4321
3	100	100	30%	2%	654	6722
4	100	100	40%	2%	649	9963
5	100	100	50%	2%	646	11535
6	100	100	60%	2%	656	13065
7	100	100	70%	2%	711	14862
8	100	100	80%	2%	597	18528
9	100	100	90%	2%	611	18672
10	100	100	100%	2%	597	18022

Table-5. GA parameters for different values of mutation rate

GA Parameter	Value
Encoding Scheme	Permutation Encoding
Population Size	100
Iteration Count	100
Selection Technique	Random Selection
Cross Over Rate	40%
Cross Over Type	One point cross over
Mutation Rate	Varied from 1% to 10% in steps of 1%.

Table 6. Results of Algorithm -1 implementation for GA parameters shown in Table-5.

Sr. No.	Iterations	Population Size	Cross Over rate	Mutation Rate	Best Chromosome	Time (ms)
1	100	100	40%	1%	814	9181
2	100	100	40%	2%	653	7976
3	100	100	40%	3%	717	8039
4	100	100	40%	4%	739	7921
5	100	100	40%	5%	711	7717
6	100	100	40%	6%	731	7883
7	100	100	40%	7%	604	7870
8	100	100	40%	8%	628	8030
9	100	100	40%	9%	595	7931
10	100	100	40%	10%	562	7997

From the results shown in Table 2 it is concluded that a result of path length 597 is achieved at high values of population size (160 chromosomes). But algorithm takes 12036 ms to execute. From the results shown in Table 4 it is concluded that a result of path length 597 is achieved at high values of cross over rate (80%). But algorithm takes 18528 ms to execute. From the results shown in Table 6 it is concluded that a result of path length 562 is achieved at high values of mutation rate (40%) and algorithm takes 7997 ms to execute.

V. CONCLUSION AND FUTURE SCOPE

In this paper genetic algorithm is applied on Traveling Salesmen Problem. The GA is applied on standard TSP instance Eil51 having 51 cities. The GA is applied with different values of GA parameters. Different values of population size, cross over rate and mutation rate are used and results are analysed. A path length of 597 is achieved in 12036 ms at 160 population size. A path length of 597 is achieved in 18528 ms at 80% cross over rate and a path length of 562 is achieved in 7997 ms at 10% cross over rate. Thus it is concluded that the performance of Ga is better at high values of mutation rate because it produces better results (of path length 562) in less time (7997 ms).

REFERENCES

[1] S. Sharma and K. Gupta, "*Solving the traveling salesmen problem through genetic algorithm with new variation order crossover," 2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*, Udaipur, 2011, pp. 274-276.

[2] B. H. Hasan and M. S. Mustafa, "*Comparative Study of Mutation Operators on the Behavior of Genetic Algorithms Applied to Non-deterministic Polynomial (NP) Problems," 2011 Second International Conference on Intelligent Systems, Modelling and Simulation*, Kuala Lumpur, 2011, pp. 7-12.

[3] M. A. H. Akhand, S. Akter, S. Sazzadur Rahman and M. M. Hafizur Rahman, "*Particle Swarm Optimization with partial search to solve Traveling Salesman Problem," 2012 International Conference on Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, 2012, pp. 118-121.

[4] H. ElAarag and S. Romano, "*Animation of the Traveling Salesman Problem," 2013 Proceedings of IEEE Southeastcon*, Jacksonville, FL, 2013, pp. 1-6.

[5] S. Cui and S. Han, "*Ant Colony Algorithm and Its Application in Solving the Traveling Salesman Problem," 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control*, Shenyang, 2013, pp. 1200-1203.

[6] Q. Bai, G. Li and Q. Sun, "*An improved hybrid algorithm for traveling salesman problem," 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI)*, Shenyang, 2015, pp. 806-809.

[7] M. Munlin and M. Anantathanavit, "*Hybrid K-means and Particle Swarm Optimization for symmetric Traveling Salesman Problem*," *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, Auckland, 2015, pp. 671-676.

[8] Muhao Chen, Chen Gong, Xiaolong Li and Zongxin Yu, "*Research on solving Traveling Salesman Problem based on virtual instrument technology and genetic-annealing algorithms," 2015 Chinese Automation Congress (CAC)*, Wuhan, 2015, pp. 1825-1827.

[9] J. Stastný, V. Skorpil and L. Cizek, "*Traveling Salesman Problem optimization by means of graph-based algorithm," 2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, Vienna, 2016, pp. 207-210.

[10] M. Khalil, J. Li, Y. Wang and A. Khan, "*Algorithm to solve travel salesman problem efficently," 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, 2016, pp. 123-126.

[11] Q. Hao, L. Fang and S. Tao, "*A Discrete Fruit Fly Optimization Algorithm for Traveling Salesman Problem," 2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, Wuhan, 2017, pp. 254-257.

[12] I. B. K. Widiartha, S. E. Anjarwani and F. Bimantoro, "*Traveling salesman problem using multi-element genetic algorithm*," *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, Lombok, 2017, pp. 1-4.

[13] A. H. M. Alaidi and A. Mahmood, "*Distributed hybrid method to solve multiple traveling salesman problems*," *2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA)*, Wasit, 2018, pp. 74-78.

[14] Z. Pan, Y. Chen, W. Cheng and D. Guo, "*Improved fruit fly optimization algorithm for traveling salesman problem*," *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, Nanjing, 2018, pp. 466-470.

[15] M. Bandyopadhyay, S. Chattopadhyay , A. Das, "*Emphasis on Genetic Algorithm (GA) Over Different PID Tuning Methods of Controlling Servo System Using MATLAB*", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.3, pp.8-13, 2013.

[16] Rajeev Ranjan , P.J. Pawar, "*Assembly Line Balancing Using Real Coded Genetic Algorithm*", International Journal of Scientific Research in Computer Science and Engineering, Vol.2, Issue.4, pp.1-5, 2014.

Authors Profile

Mr Rajesh Singh completed his M.Tech from YMCA University of science and technology Faridabad, Haryana, India. He is a research scholar of Al-Falah University Faridabad, Haryana, India. He has published twelve papers in various conferences and journals till date. His research area includes Algorithms, Compiler Design and Genetic Algorithm.

Prof. (Dr.) Bhupesh Kumar Singh received B.E. in (Computer Science & Engineering) from Government College of Engineering. Tirunelveli, Anna University, India in the year of 1999, and Ph. D in Sketch Recognition from BIT Mesra, Ranchi in the year 2013. Currently working as Professor & Director at Satyug Darshan Institute of Engineering & Technology, Faridabad, India. His research interest includes theoretical computer science, image processing and pattern recognition.

Dr. V.K. Panchal is a retired associate director at Defense Terrain & Research Lab, Defense and Research Development Organization (DRDO), New Delhi, India. He is an Associate Member of IEEE (Computer Society) and Life Member of Indian Society of Remote Sensing. He has done Ph.D in Artificial Intelligence. He has chaired sessions & delivered invited talks at many national & international conferences. His research interests are in the synthesis of terrain understanding model based on incomplete information set using bio-inspired intelligence and remote sensing.

