Comparison of Particle Swarm Optimization and Simulated Annealing Applied to Travelling Salesman Problem

M. Sumathi, U. Rahamathunnisa, A. Anitha, Druheen Das, Nallakaruppan. M. K

Abstract*: *The Travelling Salesman Problem (TSP) is a very old yet very popular and highly researched topic. This problem has numerous solution paths, but to find the path with the least distance, we would have to check the combination of all different paths but that would increase the time to achieve the result. So, in our quest to solve the problem, we try to optimize the algorithms to give the best output that is the shortest path in the least number of iterations. There are various techniques to solve the Travelling Salesman Problem of which we will be using the two most popular techniques that are Particle Swarm Optimization (PSO), and Simulated Annealing (SA). In this paper, we perform a comparative analysis of these two algorithms on the basis of a number of iterations required to reach the optimum path. We have implemented the algorithms in Matlab, and run these algorithms on multiple sets of inputs and various number of inputs in each set. We will be comparing on the basis of number of iterations required for reaching the optimum solution.

Index Terms*: *Travelling Salesman problem (TSP), Particle Swarm Optimization (PSO), Simulated Annealing (SA).

I. INTRODUCTION

The Travelling Salesman Problem (TSP) is a very old and very popular and highly researched topic with respect to combinatorial optimization algorithms. It is a problem in which we optimize the algorithm to find out the shortest distance in the least number of iterations. In this problem, there are multiple cities which have to be travelled by a salesman and must visit each city only once and return to the city where he started from. So, our job is to find out the order of cities in which he should visit such that the distance covered is the least compared to any other order thus reducing the total cost of the journey. It belongs to the class of NP-hard optimization problems [1]. The problem is very simple with a small number of cities, but with the increase in the number of cities that the salesman has to cover the computational complexity of finding the optimum path grows exponentially. Thus, if we want to find out the optimum path for 5 cities there would be 5! Combinations which equals to 120 routes, similarly if we were working to 50 cities there would be 50! Combinations which equals to 3.0414093201713378043612608166065e+64. Therefore, we can see that the computational complexity increases exponentially. TSP has numerous real-world applications such as integrated circuits [2], semiconductor industries [3], wireless communication [4], networking, transportation [5], vehicle routing [6] and logistics industry. There is no algorithm at present which is quite efficient enough for the Travelling Salesman Problem (TSP) and all other problems which belong to the same category. The need to quickly find a solution to these problems led to the creation of many algorithms that gave output through approximation. Some of the most popular algorithms for solving these problems are genetic algorithms (GA), simulated annealing (SA), ant colony optimization (ACO), particle swarm optimization (PSO) [7]. TSP being the main focus to solve these NP-hard problems. In this paper, we will be working with Simulated Annealing (SA) and Particle Swarm Optimization (PSO), we compare the performance of these two algorithms to find out which algorithm does it in lesser number of iterations.

II. LITERATURE SURVEY

Researchers from various fields such as biology, physics, mathematics, artificial intelligence, and many other fields are devoting themselves to solve the Travelling Salesman Problem (TSP) in much more efficient manner so as to develop much more efficient algorithms than the pre-existing ones. Some algorithms that have been developed are ant colony optimization (ACO) [8], neural networks [9], genetic algorithm (GA) [10], particle swarm optimization (PSO) [11], memetic computing [12], simulated annealing (SA) [13], etc. Travelling Salesman Problem (TSP) is one such problem which has a huge number and variety of real-world applications, vehicle routing, data traffic routing in networking, finding the sequence for drilling in printed circuit boards, DNA sequencing, pattern recognition in image processing, transportation and logistics industry being some of them. In computational intelligence, swarm intelligence is a heavily researched topic. Population being the attribute which simulates the behaviour of animals in the real world. There are numerous swarm intelligence algorithms that have been developed, ant colony optimization, fish-warm algorithm, bee colony algorithm being some of them.

Revised Manuscript Received on April 07, 2019.

M.Sumathi, Assistant Professor, K.Ramakrishnan College od Engineering, Trichy, India

U.Rahamathunnisa,Associate, Professor,Vellore Institute of Technology, Vellore, India.

A.Anitha,Associate, Professor, VelloreInstituteof Technology, Vellore, India

Druheen Das, MCA Student, Vellore Institute of Technology,Vellore, India

Nallakaruppan,M.K,, Assistant Professor, Vellore Institue of Technology, Vellore, India

Particle Swarm Optimization has become the most popular as it is the easiest to implement due to its simple concept and mainly because it has been applied in a variety of fields. Particle Swarm Optimization (PSO) was developed by its developers to solve the travelling salesman problem [14]. The authors proposed a new technique using a swap operator and swap sequence, they also redefined some operators on its basis. Hendtlass [15] in his paper proposes a method to improve diversity by including memory for the particles. The memory is used to store the list of all possible solutions (target points) for each node. This list of target points will act as alternatives to the current local optimal point. Two new parameters are added to the standard Particle Swarm Optimization which is memory list size and its probability.

In another research paper [16] researchers developed their own version of simulated annealing which they called hybrid simulated annealing. It deploys tabu search algorithm to solve TSP. The results from the experiments proved not only to improve in accuracy but also an improvement in efficiency of the customized algorithm. The research paper by Archetti [17] takes another step forward and uses integer linear programming on tabu search based algorithm, they concluded with the help of simulations that this algorithm is much more efficient and performed well in solving the vehicle routing problem.

III. Methods

A. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was developed by Kennedy and Eberhart and was introduced in 1995 [18] it is an optimizing technique which they developed after studying the behaviour of birds which move in flocks and fishes which stays in schools and work collectively towards the same goal at any given instance. Kennedy and his partner then mimicked this technique into their algorithm thus developing PSO. In Particle Swarm Optimization (PSO) there is a pool of solutions which is called the search space, and in this search space, each and every solution is called a particle. Now every particle has a fitness value and velocity. The fitness value is evaluated by the fitness function and also optimizes it, the velocity gives direction to the particle's flight. The particles move around in the search space which follows the current optimum particles. When the PSO algorithm starts, it is initialized with a pool of random particles (solutions) and then with each iteration the algorithm searches for a more optimum solution. With every iteration two attributes are updated in each and every particle, pbest (Personal Best) which is the best solution (fitness) which has been achieved by that particle so far, the next attribute that is updated is called the gbest (Global Best) it is the best solution achieved so far by all particles in the population collectively and this is tracked by the swarm optimizer. When a particle has the best solution amongst its immediate neighbours in its topology is called the lbest (Local Best). The particle then updates position and velocity after it has found the two best values.

Following are the two formulas used in PSO algorithm [18, 19]:

$$v_{id}^{t+1} = v_{id}^{t} + c1.rand(0,1).(p_{id}^{t} - x_{id}^{t}) + c2.rand(0,1).(p_{gd}^{t} - x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (2)$$

v_{id}^{t} velocity of the particle.

x_{id}^{t} position of the particle.

i index of the particle.

t iteration number.

$c1$ and $c2$ represents the speed.

p_i best position achieved so far by particle i.

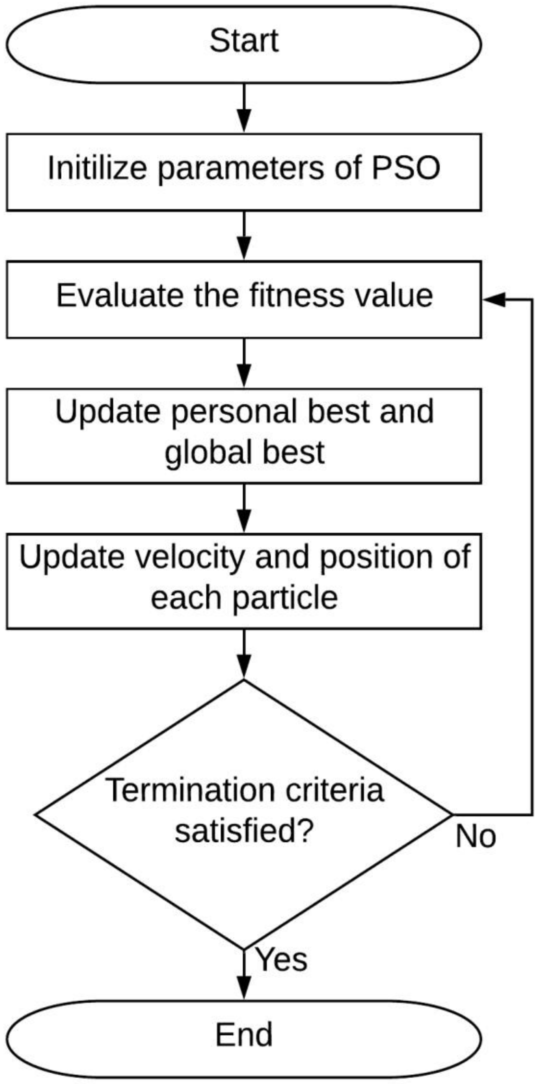

Figure 1. Flowchart for PSO algorithm

B. ***Simulated Annealing (SA)*** The Simulated Annealing (SA) belongs to the category of meta-heuristic optimization methods which was developed by taking inspiration from the process of re-crystallization of metals called annealing. Given enough time this algorithm finds the optimal solution to the problem. Based on the annealing process of cooling metals the algorithm mimics the process and replaces the current solution by a random nearby solution, thus gradually decreasing the number of solutions in the search pool.

This technique was first proposed by Kirkpatrick [20] in 1983, He proposed the idea which imitated the process of annealing of metals. The metal was heated to a very high temperature, due to the high temperature the atoms in the metal attain an unstable state in search of another state. Now when the cooling process starts the atoms of the metal cool down and will be in search of a state with lower energy than its original state. Now, this behaviour of state changing procedure is applied to real-world problems. The system will generate a new state and then compare it to the energy level of the new state, if the energy level of this new state is lower compared to the energy level of the current state, then the system will take this new state as its new state else it will reject.

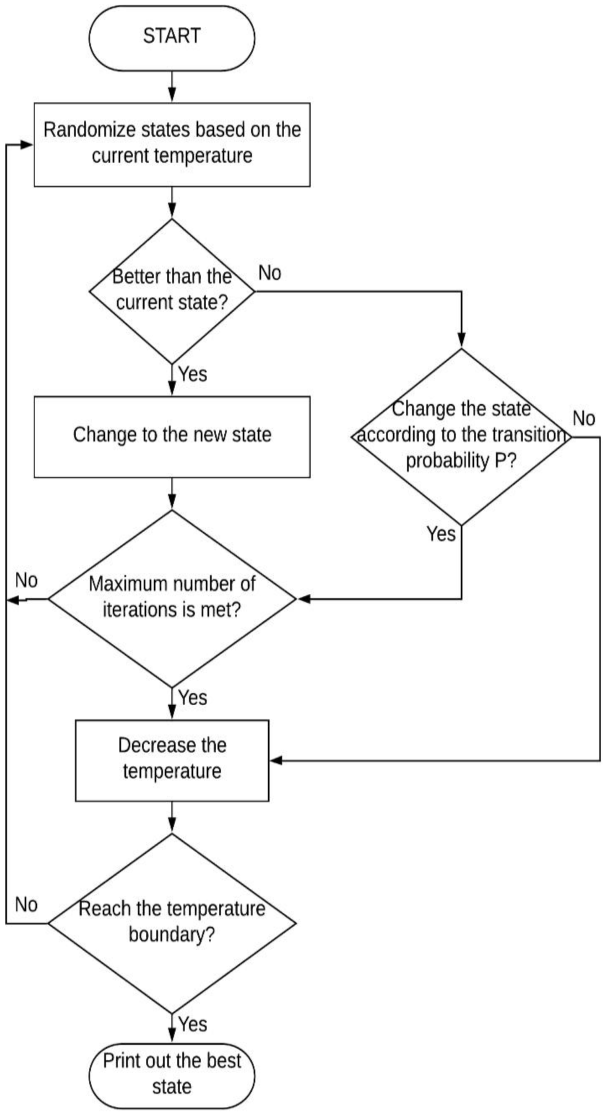

Figure 2. Flowchart for SA algorithm

IV. IMPLEMENTATION

The simulations were run on a Windows 10 64-bit laptop with i5-6200u processor @ 2.30GHz and having RAM of 12GB. We used Matlab and both the algorithms were implemented in Matlab.
In order to test the algorithm, we decided to run multiple experiments with varying complexity of dataset. The two algorithms were tested on three different datasets of varying complexity.

V. RESULTS

A. Dataset 1 (30 coordinates)

x=[72 63 19 68 75 83 58 36 76 7 44 55 94 85 95 1 31 34 100 40 91 96 43 78 21 64 25 46 86 17];

y=[24 64 75 34 21 81 89 32 71 94 1 35 7 61 82 70 9 19 98 86 68 87 4 72 10 92 42 49 91 54];

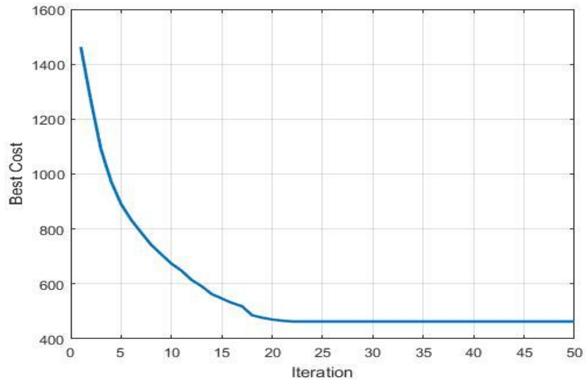

Figure 3. PSO on Dataset1

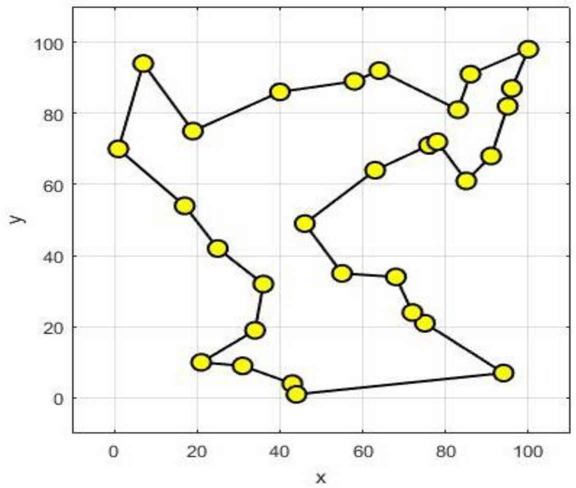

Figure 4. PSO on Dataset1

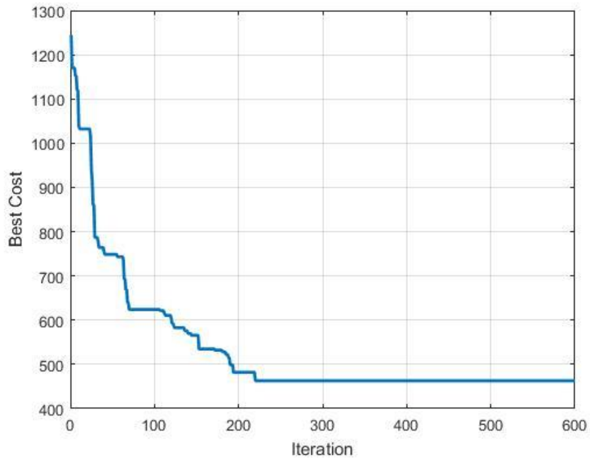

Figure 5. SA on Dataset1

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

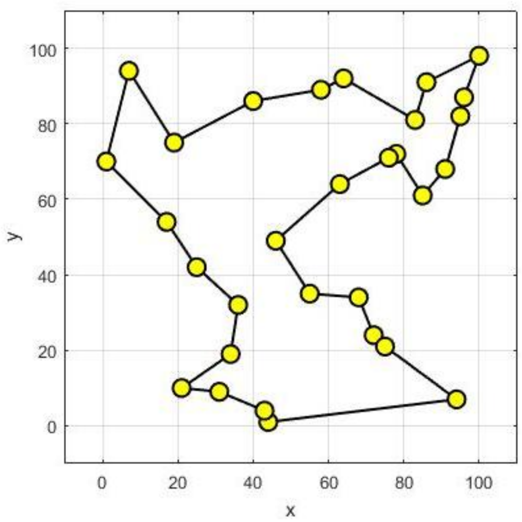

Figure 6. SA on Dataset1

B. Dataset 2 (50 coordinates)

x=[39 25 38 82 93 36 86 42 10 16 85 8 95 11 76 34 88 92 65 61 59 17 24 12 98 90 28 13 1 23 67 47 66 31 29 60 100 78 53 18 69 97 75 57 72 44 73 46 27 7];

y=[82 5 6 44 95 100 64 91 67 8 31 63 23 61 12 88 3 26 55 2 66 16 14 29 15 42 30 94 87 35 1 92 7 86 47 76 73 78 68 51 96 54 80 17 4 40 21 83 32 97];

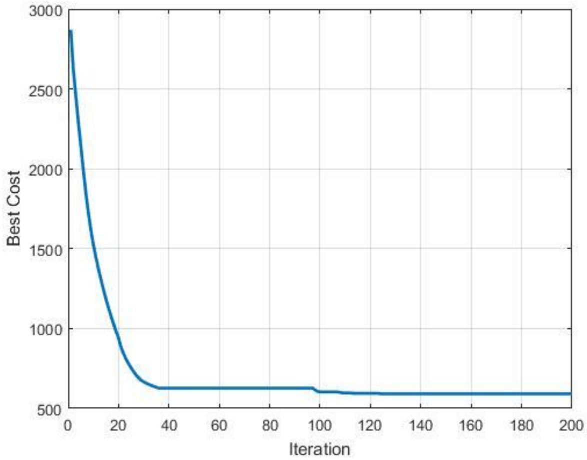

Figure 7. PSO on Dataset2

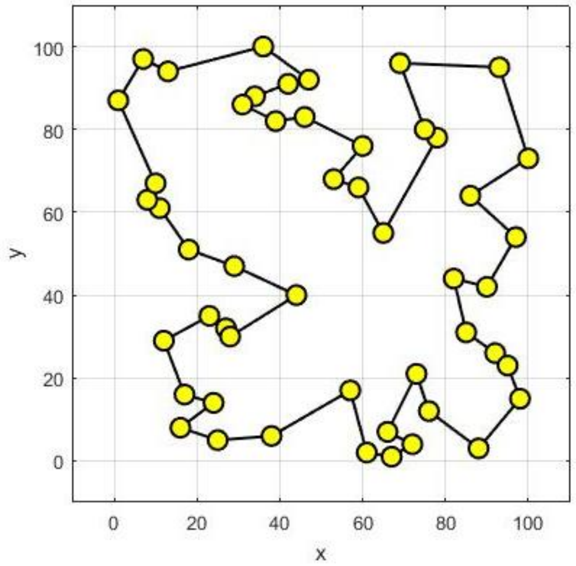

Figure 8. PSO on Dataset2

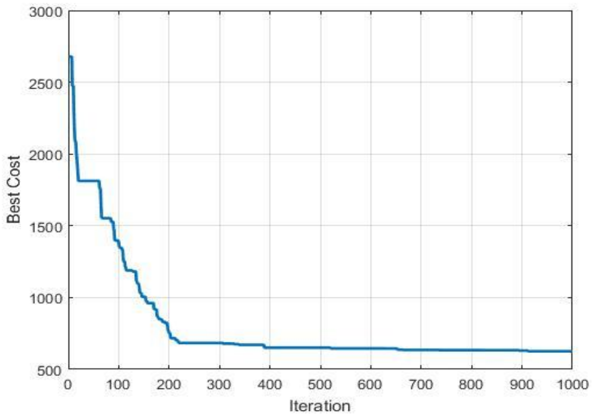

Figure 9. SA on Dataset2

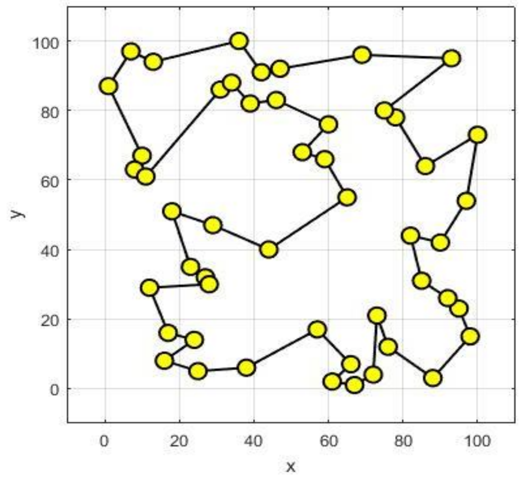

Figure 10. SA on Dataset2

C. Dataset 3 (70 coordinates)

x=[80 75 39 92 58 10 35 20 17 48 26 53 97 100 72 42 91 34 50 85 5 33 63 96 45 56 29 23 27 51 15 93 38 59 52 66 86 47 49 94 44 77 7 73 25 64 36 89 4 67 43 78 81 98 61 62 83 30 40 19 9 22 13 70 74 57 21 82 41 11];

y=[22 90 12 80 9 93 85 3 88 100 56 98 74 54 5 69 33 34 36 49 14 39 87 47 11 37 55 70 58 65 59 50 18 44 30 42 41 81 25 38 46 13 31 57 94 53 64 27 89 60 91 16 68 15 28 51 95 17 92 40 61 4 32 43 45 62 24 48 79 26];

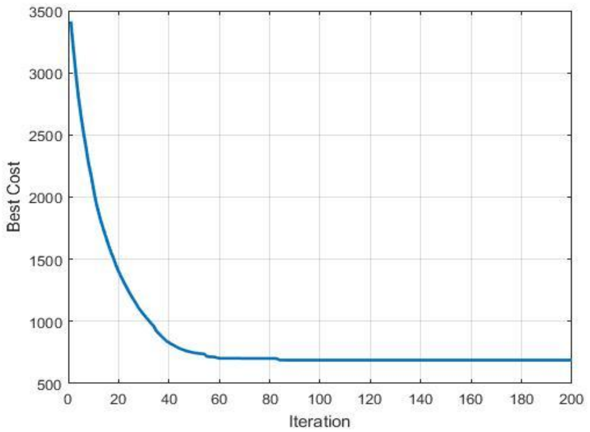

Figure 11. PSO on Dataset3

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

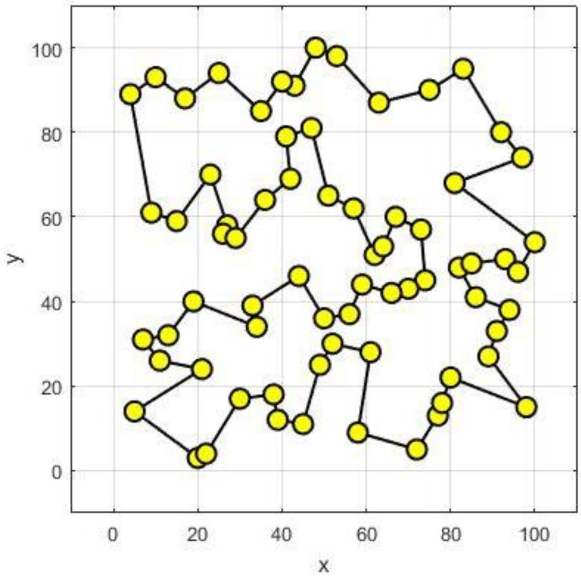

Figure 12. PSO on Dataset3

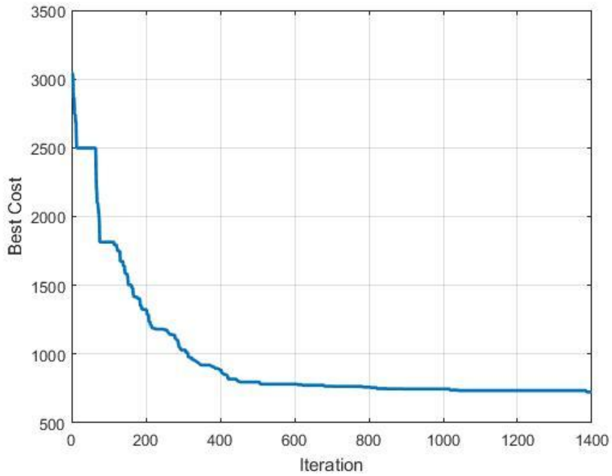

Figure 13. SA on Dataset3

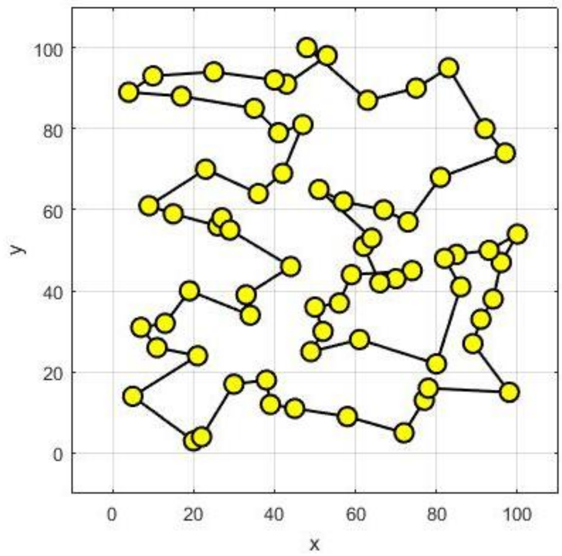

Figure 14. SA on Dataset3

	PSO			SA		
	Iteration	Time (SEC)	Best Cost	Iteration	Time (sec)	Best Cost
DATASET 1	22	3.66	462.9619	220	9.58	462.962
DATASET 2	163	12.23	591.4436	998	34.16	625.353
DATASET 3	86	58.12	688.1957	1385	58.02	724.685
DATASET 4	239	151.77	850.6034	2318	99.24	897.588

Table 1. Overview of Simulation Results

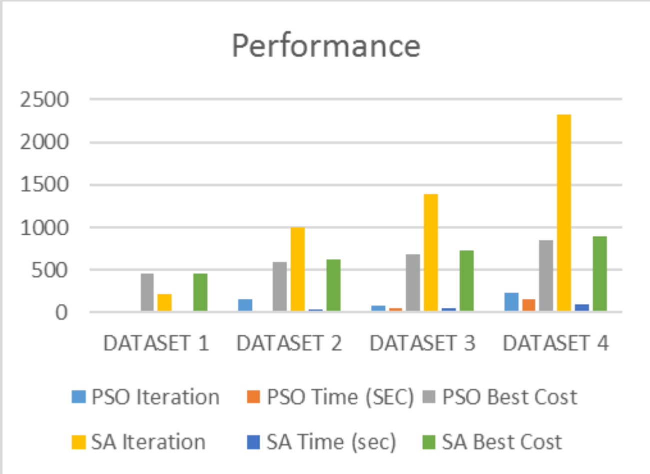

Figure 15. Graph representation of Simulation Results

VI. CONCLUSION

In this paper, we did a comparative analysis of two very popular optimization algorithms namely Particle Swarm Optimization (PSO) and Simulated Annealing (SA). The attributes we considered for comparing these algorithms are "Best Cost" and number of "Iterations" required. The algorithms were implemented in Matlab, and were benchmarked using three datasets. Dataset1 with 30 coordinates, Dataset2 with 50 coordinates, and Daatset3 with 70 coordinates. In all the three simulations, we saw that PSO outperformed SA by a great margin. SA took an exceptionally high number of iterations to achieve the same result that of PSO. However, we did not measure the execution time but it was evident at the time of simulation that each iteration of PSO took much longer in comparison to the time taken by an iteration of SA. Therefore, we conclude that both algorithms have its merits and demerits, PSO gives results with a lower number of iterations but SA gives results faster than PSO.

REFERENCES

1. S. Arora, Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems, Journal of the ACM (JACM) 45 (1998) 753–782.
2. M. Shokouhifar& A. Jalali, (2015) "An evolutionary-based methodology for symbolic simplification of analog circuits using genetic algorithm and simulated annealing", Expert Systems with Applications, Vol. 42, No. 3, pp. 1189-1201

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

3. C. A. Silvaa, M. C. Sousaa& T. A. Runkler, (2008) "Rescheduling and optimization of logistic processes using GA and ACO", Engineering Applications of Artificial Intelligence, Vol. 21, pp. 343-352
4. M. Shokouhifar& A. Jalali, (2015) "A new evolutionary based application specific routing protocol for clustered wireless sensor networks", AEU-International Journal of Electronics and Communications, Vol. 69, No. 1, pp. 432-441.
5. X. Wang & A. C. Regan, (2002) "Local truckload pickup and delivery with hard time window constraints", Transportation Research, Part B, pp. 97–112.
6. M. Shokouhifar, A. Jalali& H. Torfehnejad, (2015) "Optimal routing in traveling salesman problem using artificial bee colony and simulated annealing", In 1st National Road ITS Congress
7. J. Kennedy, R. Eberhart, Particle swarm optimization, in: Neural Networks, 1995.Proceedings., IEEE International Conference on, volume 4,IEEE, pp. 1942–1948.
8. Ellabib I, Calamai P, Basir O, "Exchange strategies for multiple ant colony system", Inf Sci 177(5), 007, p.1248–1264.
9. Masutti TA, de Castro LN, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem", Inf Sci 179(10), 2009, pp.1454–1468.
10. Hua Z, Huang F, "A variable-grouping based genetic algorithm for large-scale integer programming", Inf Sci 176(19), 2006, pp.2869–2885.
11. Onwubolu GC, Clerc M, "Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization", Int J Prod Res 42(3), 2004, pp.473–491.
12. Acampora G, Gaeta M, Loia V, "Combining multi agent paradigm and memetic computing for personalized and adaptive learning experiences", Comput Intell J 27(2), 2011, pp.141–165.
13. Lo CC, Hus CC, "Annealing framework with learning memory", IEEE Trans Syst Man Cybern Part A 28(5), 1998, pp.1–13.
14. Wang KP, Huang L, Zhou CG, Pang W, "Particle swarm optimization for traveling salesman problem", In: International conference on machine learning and cybernetics, 2003, pp.1583–1585.
15. Hendtlass, T, "Preserving diversity in particle swarm optimization, Developments in Applied Artificial Intelligence", Proceedings of the 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE 2003, Laughborough, UK, June 2003, In: Lecture Notes in Computer Science, Vol. 2718, 2003, pp.4104-4108.
16. Lin, Y.; Bian, Z.; Liu, X. Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing–tabu search algorithm to solve the symmetrical traveling salesman problem. Appl. Soft Comput. 2016, 49, 937–952.
17. Archetti, C.; Guastaroba, G.; Speranza, M.G. An ILP-refined tabu search for the directed profitable rural postman problem. Discret. Appl. Math. 2014, 163, 3–16.
18. Kennedy J, Eberhart R. Particle swarm optimization. IEEE International Conference on Neural Networks. 1995:1942–1948.
19. Shi Y, Eberhart R. A modified particle swarm optimizer. IEEE World Congress on Computational Intelligence. 1998: 69–73.
20. Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220(4598), 671–680.
21. Nallakaruppan, M.K., Kumaran, U.S. Quick fix for obstacles emerging in management recruitment measure using IOT-based candidate selection (2018) Service Oriented Computing and Applications, 12 (3-4), pp. 275-284.