Experimental Comparison of GA and PSO methods for Solving Travelling Salesman Problem

Mona Borisagar[1] Mahesh Panchal[2]
[1]M.E. Scholar [2]Dy. Director
[1,2]Department of Computer Engineering
[1]KITRC, Kalol (N. G.), Gujarat, India [2]Gujarat Technological University, Chandkheda, Ahmedabad, Gujarat, India

Abstract— For solving hard and complex problems, nature has been main source of inspiration for many years. Nature Inspired heuristic algorithms is getting more popular among the researcher for solving real world NP hard problems like Travelling Salesman Problem(TSP) , Graph Colouring, Vehicle Routing etc. One such classical optimization problem is Travelling Salesman Problem, which is NP hard problem that cannot be solved conventionally particularly when no. of cities increase. It will take years to find optimal solution ,If one tries to solve TSP using conventional approach by considering all possible tours. So, Meta-Heuristic algorithm is the feasible solution to such problem. In this paper we consider three widely used nature inspired heuristic approaches Ant Colony Optimization, Genetic Algorithm and Particle Swarm Optimization to solve TSP. We also compare results of GA and PSO for Instance taken from TSPLIB.

***Key words*:** Travelling Salesman Problem, Ant Colony Optimization (ACO), Genetic Algorithm (GA),Particle Swarm Optimization (PSO).

I. INTRODUCTION

The TSP has held the interests of computer scientists and mathematicians because, even after about half a decade of research, the problem has not been completely solved. The TSP falls in a distinguished category of NP- hard problems[1]. The TSP is applied to solve many practical problems within our daily lives. Thus, TSP solution would be very beneficial. When we talk about a solution to the TSP, we are talking about a polynomial time solution to the general TSP.The TSP is defined as, given a complete graph, G, with a set of vertices, V, a set of edges, E, and a cost, cij, associated with each edge in E. The value cij is the cost incurred when traversing from vertex $i \in V$ to vertex $j \in V$. Given this information, a solution to the TSP must return the cheapest Hamiltonian cycle of G which is a cycle that visits each node in a graph exactly once. This is referred to as a tour in TSP terms.A lots of algorithms have been proposed to solve TSP.Some of them (based on dynamic programming or branch and bound -methods) provide the global optimum solution. Other algorithms are nature inspired heuristic ones(PSO,ACO,GA), which are much faster, but they do not guarantee the optimal solutions.

The rest of the paper is organized as follows, in section II, we present the PSO algorithm. In section III we have presented GA algorithm .In section IV, we have present ACO algorithm. In section V, we have compared these algorithm and In section VI, concludes the paper and discusses the future path of our work.

II. PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle swarm optimization (PSO) is a population based optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking.

A. *Pseudo Code of PSO[2]*

```
begin procedure PSO
For( each particle)
      Initialize particle
End For
Do
For (each particle)
      1.Calculate fitness value
      2.If the fitness value is better than the best fitness valu-
      -e (pBest) in history
      3.Set current value as the new pBest
End For
For (each particle)
      1.Find in the particle neighbourhood, the particle with
      the best fitness
      2.Calculate particle velocity according to the velocity
      equation                                  (2.1)
      3.Apply the velocity constriction
      4.Update particle position according to the position
      equation                 (2.2)
      Apply the position constriction
End For
While (maximum iterations or minimum error criteria is not
attained)
end PSO procedure
```

B. *Working of PSO*

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search space according to a few simple formulae. The movement of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. Formally, let $f : Rn \rightarrow R$ be the cost function which must be minimized. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the objective function value of the given candidate solution. The gradient of f is not known. The goal is to find a solution a for which $f(a) \leq f(b)$ for all b in the search-space, which would mean a is the global minimum. Maximization can be performed by considering the function $h = -f$ instead.

PSO was presented under the inspiration of bird flock immigration during the course of finding food and then be used in the optimization problems. In PSO, each optimization problem solution is taken as a bird in the searching space and it is called “particle”. Every particle has a fitness value which is determined by target functions and it has also a velocity which determines its destination and distance. All particles search in the solution space for their best positions and the positions of the best particles in the swarm. PSO is initially a group of random particles (random solutions), and then the optimum solutions are found by repeated searching. In every iteration, a particle will follow two bests to renew itself: the best position found for a particle called pbest ; the best position found for the whole swarm called gbest . All particles will determine following steps through the best experiences of individuals themselves and their companions. For particle id, its velocity and its position renewal formula are as follows:

V 'id =ωVid +η1 rand ()(Pidb − Xid) +η2rand ()(Pgdb − Xid) (2.1)

X 'id = Xid +V'id (2.2)

In here: ω is called inertia weight, it is a proportion factor that is concerned with former velocity,0 << ω < < 1, η1 and η2 are constants and are called accelerating factors, normally η1 =η2 = 2 , rand() are random numbers, Xid represents the position of particle id ; Vid represents the velocity of particle id ; Pidb , Pgdb represent separately the best position particle id has found and the position of the best particles in the whole swarm.

In formula(2.1), the first part represents the former velocity of the particle, it enables the particle to possess expanding

Tendency in the searching space and thus makes the algorithm be more capable in global searching; the second part is called cognition part, it represents the process of absorbing individual experience knowledge on the part of the particle; the third part is called social part, it represents the process of learning from the experiences of other particles on the part of certain particle, and it also shows the information sharing and social cooperation among particles. The flow of PSO can briefly describe as following: First, to initialize a group of particles, e.g. to give randomly

Each particle an initial position Xi and an initial velocity Vi , and then to calculate its fitness value f. In every iteration, evaluated a particle's fitness value by analyzing the velocity and positions of renewed particles in formula (2.1) and (2.2). When a particle finds a better position than previously, it will mark this coordinate into vector P1, the vector difference between P1 and the present position of the particle will randomly be added to next velocity vector, so that the following renewed particles will search around this point, it's also called in formula (2.1) cognition component. The weight difference of the present position of the particle swarm and the best position of the swarm Pgd will also be added to velocity vector for adjusting the next population velocity. This is also called in formula (2.1) social component. These two adjustments will enable particles to search around two bests[3].

The most obvious advantage of PSO is that the convergence speed of the swarm is very high, scholars like Clerc [4] has presented proof on its convergence.

C. *Advantages of PSO*

(1) PSO is efficient global optimizer for structural applications, with limited parameters[5].
(2) It is easy to implement, with very few parameters to finetune.
(3) It has very simple concept, easy to implement.
(4) PSO generates high quality solution with less calculation time and stable convergence[5].

D. *Limitations of PSO*

(1) Premature convergence is major limitation of PSO.
(2) Dependency on initial condition, parametric values, difficulty in identifying design parameters is problems to be solved in PSO[5].

III. GENETIC ALGORITHM

A Genetic Algorithm is one of the oldest and most successful optimization technique based on natural of Evolution. It was originally proposed by John Holland ain the 1960s at the University of Michigan, to study the process of evolution and adaption occurring in nature[6] .Genetic Algorithm is inspired by Charles Darwin's theory of evolution and natural selection. This uses survival of the fittest approach for selecting the best (fittest) solution from the available solutions.

A. *Pseudo Code of GA[7]*

```
begin procedure GA
generate populations and fitness function
evaluate population
while(termination criteria not meet)
     {
       while (best solution not meet)
            {
            crossover
            mutation
            evaluate
            }
     }
post-process results and output
end GA procedure
```

B. *Working of GA*

This approach uses the Chromosomes 'population as the starting point then each chromosome is tested against fitness using an appropriate fitness function. Then the best chromosomes are selected and they undergo process of crossover and mutation to create new set of chromosome.

GA uses three operations to maintain population that evolve from one generation to another. The first operation is "Selection" operation which is inspired by the principle of 'Survival of the Fittest'. The search begins from a randomly generated population that evolve over successive generations (iterations).A fitness function is used to evaluate the performance of the solutions. Each time two solutions are chosen as parent solutions by selection process based on fitness function. The second operation is the "Crossover" operation, which is inspired by mating in biological populations. The crossover operator inherits features of good surviving designs from the parent population into the future population, which will have better fitness value on average. The third operation is "Mutation" operation, which causes diversity in population characteristics. It causes local modifications to the new generation randomly. The new generation is identical to the parent except one or more changes made by mutation process. Repeat selection, crossover and mutation operations to produce more new solutions until the population size of the new generation is the same as that of the old one. The iteration then starts from the new population. Since better solutions have a larger probability to be selected for crossover and the new solutions produced carry the features of their parents, it is hoped that the new generation will be better than the old one. The procedure continues until the number of generations is reached to n or the solution quality cannot be easily improved.

C. *Advantages of GA*

1) It always gives solution and solution gets better with time. [8]
2) It supports multi-objective optimization. [8]
3) It is more useful and efficient when search space is large, complex and poorly known or no mathematical analysis is available. [9]
4) The GA is well suited to and has been extensively applied to solve complex design optimization problems because it can handle both discrete and continuous variables, and nonlinear objective functions without requiring gradient information.[10]

D. *Limitations of GA*

1) GA may converge towards local optima, When fitness function is not properly defined..[11]
2) Operations on dynamic sets is difficult.[11]
3) For constraint based optimization problem, GA is not appropriate choice. [11]

IV. ANT COLONY OPTIMIZATION

The ACO is the oldest and widely used approach proposed by Marco Dorigo in 1992 in his PhD thesis" Optimization, learning, and Natural Algorithms" [12]. The ACO is inspired by the food search behaviour of real ants and their ability in finding the optimum paths. It is a population-based general search technique for the solution of difficult combinatorial optimization problems.

A. *Working of ACO*

Key parameters in ACO are distances between two ants, pheromone update which includes pheromone deposit, and pheromone evaporation[13]. Initially ants start their search roams randomly. An ant selects the next node to be visited by probabilistic equation. When ant k is on node I, the probability of going to node j is given by equation as follows:

$$p_{ij}^{\,k} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l\in N_i^k}(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}} \quad \text{IF} \quad j \in N_i^{\,k} \qquad (4.1)$$

Where, N_i^k is the adjacent node of k which still not visited by ant i. α is the local pheromone coefficient that controls the amount of contribution pheromone plays in a components probability of selection and is commonly set to 0.1. β is the heuristic coefficient which controls the amount of contribution problem specific heuristic information plays in a components probability of selection and is commonly between 2 and 5, such as 2.5.η = 1/d which is the inverse od distance between I and j.The total number of ants (m) is commonly set low, such as 10. The arcs which is used by the most ants and which is the shortest, receives more pheromone and will be used by the ants in future[13].

Another main function is pheromone update which consists of pheromone deposit and pheromone evaporation. Pheromone values are updated each time an ant travels from one node to another. A first Pheromone value on each arc is decreased by constant factor which is known as pheromone evaporation. Then some amount of pheromone is added to each node which is being traversed by each ant, is known as pheromone deposit. Pheromone evaporation is given by equation follows:

$$\tau_{ij} \leftarrow (1-\rho)\,\tau_{ij} \qquad (4.2)$$

Where, ρ is the evaporation rate.
Each ant drops some amount of pheromone on each node which is known as pheromone deposit and given by equation follows:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \qquad (4.3)$$

Where,
m is the number of ants,
Δτikj is the amount of pheromone drop on k node. It is calculated as

$$\Delta\tau_{ij}^{\,k} = \begin{cases} \dfrac{1}{c^k} & \text{if arc (ij)} \in \text{T} \\ 0 & \text{otherwise} \end{cases} \qquad (4.4)$$

Where, Ck is the length of the tour by the Kth ant.

B. *Pseudo Code of ACO*

```
begin procedure ACO
generate pheromone trails and other parameters
while(termination criteria not meet)
{

construct solutions
update pheromone Trails
}
post-process results and output
end ACO procedure
```

C. *Advantages of ACO*

1) It has advantage of distributed computing.
2) It is robust and also easy to accommodate with other algorithms. [14]
3) ACO algorithms have advantage over simulated annealing and Genetic Algorithms approaches of similar problems (such as TSP) when the graph may change dynamically, the ant colony algorithms can be run continuously and adapt to changes in real time. [14]

D. *Limitations of ACO*

1) Though ant colony algorithms can solve some optimization problems successfully, we cannot prove its convergence. [14]
2) It is prone to falling in the local optimal solution. because the ACO updates the pheromone according to the current best path.[14]

V. EXPERIMENTS AND RESULTS

In Table-1, we have considered PSO, GA to solvestandard TSP instances downloaded from TSPLIB. TSP instance provides some cities with their coordinates. Table 1 lists TSP file name and execution time for PSO, GA. Results are taken from heuristic lab simulator with parameter MaxIterations=1000, Inertia=1, Swarm Size=10 . The comparison results demonstrate clearly the efficiency of PSO algorithm.

Sr. No.	TSBLIB Instance	PSO(Execution time in Second)	GA(Execution time in Second)
1	ulysses16	00.0187503	14.4499719
2	ulysses22	00.0187503	09.8086507
3	rd100	00.0725011	12.7124453
4	ch130	00.1137518	14.3559706
5	ts225	00.1262519	16.6102551
6	pr439	00.4227573	23.7963658
7	att532	00.6137595	16.1452480
8	rat783	00.6250096	26.3492545
9	rl11849	00.1187518	05:30.74373060

Table 1: Execution time Comparison of PSO and GA

VI. CONCLUSION AND FUTURE WORK

This paper presents a comparative view of most widely used optimization algorithm techniques namely ACO, GA and PSO Optimization. In sections II, III and IV we have outlined these three meta-heuristic approaches as they are described in the literature. ACO is the process used by ants to forage food source. They use pheromone trail deposition/evaporation technique to map their way. GA is an optimization technique, inspired by the law of biological reproduction and survival of fittest theory, where mutation and crossover operations used to find by the local optimal solution. PSO is presented under the inspiration of bird flock immigration during the course of finding food, where use pbest , gbest, velocity operation to find global optimal solution.

Section V, shows the experimental results obtained on standard TSP problems. Experiment results from Table-1 shows that PSO is better than GA in terms of Execution time.

By analyzing the testing results, we reach the conclusion that the PSO algorithm is efficient than the genetic algorithm in the optimization speed, for coping with the TSP. All these three techniques have immense potential and scope of application ranging from engineering to software engineering, and real world optimization problems. These techniques need to be further explored to find their suitability to certain applications. Also, there is a need to combine two or more techniques so that they complement each other and nullify their respective limitations.

REFERENCES

[1] Leonardo Zambito, " The Traveling Salesman Problem: A Comprehensive Survey" , project for CSE 4080, 2006.
[2] Website," PSO Algorithm Pseudo Code"http://tracer.uc3m.es/tws/pso/pseudocode.html.
[3] Xuesong Yan1, Can Zhang1 , Wenjing Luo1, Wei Li1, Wei Chen1 and Hanmin Liu2," Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm", IJCSI, Vol. 9, Issue 6, No 2, November 2012.
[4] M.Clerc and J.Kennedy, "The Particle Swarm: Explosion,Stability and Convergence in a Multi-Dimensional Complex Space", IEEE Trans. on Evolutionary Computation, Vol.6,2002, pp.58-73
[5] Shweta Singhal1, Shivangi Goyal2, Shubhra Goyal3 and Divya Bhatt4," A Comparative Study of a Class of Nature Inspired Algorithms", NCCFND-2011.
[6] Melanie Mitchell.,"An Introduction to Genetic Algorithms". MIT Press, Cambridge,MA, 1998.
[7] Sharad N. Kumbharana1, Prof. Gopal M. Pandey2," A Comparative Study of ACO, GA and SA forSolving Travelling Salesman Problem.
[8] Shwetasinghal, shivangigoyal, shubhragoyal and divyabhatt, "A comparative study of a class of Nature Inspired Algorithms", Proceedings of the 5th national conference :INDIACom, March 10-11, 2011.
[9] Binitha, S.S. Siva sathya, "A survey of bio inspired optimization algorithm", IJSCE, Vol-2, issue-2, May 2012.
[10] Rania Hassan,BabakCohanim, Olivier de Weck, "A comparison of PSO and GA", American Institute of Aeronautics and Astronautics,2004.
[11] Binitha, S.S. Siva sathya, "A survey of bio inspired optimization algorithm", IJSCE, Vol-2, issue-2, May 2012.
[12] M. Dorigo, L. Gambardella, "Ant colonies for the Traveling salesman problem." Biosystems 43, pp. 73-81, 1997.
[13] X. Yuneig, G. Junen, G.Bo, "An Ant Colony Optimization Algorithm based on the Nearest Neighbour Node Choosing Rules and the Crossover Operator", International Conference on Computer Science and Software Engineering, 2008.
[14] Peiyi Zhao, Peixin Zhao, Xin Zhang, "A New Ant Colony Optimization ", 2007.