The Solutions of Travelling Salesman Problem using Ant Colony and Improved Particle Swarm Optimization Techniques
Pratibha Singh1 Dr. K.L. Bansal2
1Research Scholar 2Professor
1,2Department of Computer Science
1,2Himachal Pradesh University Shimla, India

Abstract—In this paper we consider two widely used Swarm Intelligence (SI) inspired heuristic approaches Ant Colony Optimization and Improved Particle Swarm Optimization to solve classical optimization problem called Travelling Salesman Problem (TSP) that cannot be solved conventionally because it is NP hard problem. If one tries to solve TSP using conventional approach it will take years to find optimal solution. Therefore, Heuristic algorithm is the feasible solution to such problem. Interest of researchers has been attracted by Ant Colony Optimization (ACO) and Improved Particle Swarm Optimization (PSO) algorithms because of their simple, effective and efficient nature in solving real world optimization problems. The comparative analyses based on Performance have been done by using ACO and Improved PSO respectively in solving TSP in this paper. The comparative results are shown and it is devised that Improved PSO is better approach to solve the traveling salesman problem.

Key words: Particle Swarm Optimization, Swarm Intelligence, Ant Colony Optimization

I. INTRODUCTION

A. Swarm Intelligence
Swarm intelligence (SI) is based on the collective behavior of decentralized and self-organized systems in computational intelligence [9]. Swarm Intelligence consists of a population which simulates the animals’ behavior in the real world. There are several swarm intelligence optimization algorithms, as bee colony, ant colony, particle swarm optimization algorithm, genetic algorithms, differential evolution, fish-warm algorithm, etc. SI is the property of a system where Coherent functional global patterns emerge because of the collective behaviors of agents interacting locally with their environment. The basis for unguided problem solving is provided by Social interactions. [1]

B. Solving the Travelling Salesman Problem (TSP)
One of the best known NP-hard problems is the Travelling Salesman Problem, which means that to solve it in polynomial time there is no exact algorithm. To obtain optimal solution the minimal expected time is exponential. For this reason, we use heuristics to help us to obtain an “optimal” solution. With more or less success many algorithms were applied to solve TSP. Travelling salesman problem is most studied optimization problem and it has been most favorable among the researchers [5]. Almost every new approach is first tested on TSP for solving Optimization problems. Various metaheuristic approaches like Genetic Algorithm (GA), Simulated Annealing (SA), Ant Colony Optimization (ACO), particle Swarm Optimization (PSO), Firefly Algorithm (FA) have been proposed by researchers to solve these kinds of problems [12]. In this paper we’ll discuss ACO and PSO approaches for solving TSP.

II. INVESTIGATED ALGORITHMS:
A. Ant Colony Optimization
The food foraging behavior of real ants is the inspiring source of ACO. Ants initially explore the area surrounding their nest when searching for food in a random manner. As soon as an ant finds a food source, they evaluate the quality and quantity of the food and bring some of it back to the nest. Ants deposit a chemical pheromone trail on the ground during their return trip. The quantity of pheromone deposited depends on the quantity as well as quality of the food; guide other ants to the food source. The main idea behind these interactions or communication through the environment is termed stigmergy. This feature makes swarm intelligence very attractive for optimization, network routing, robotics, etc. To the original ant algorithm a number of extensions are proposed. Main features of this model are the use of a constructive greedy heuristic; positive feedback and distributed computation. The Pheromone trails are updated in two ways in the Ant colony optimization algorithm which differentiates it from the classical ant system. Firstly, when ants construct a tour they change the amount of pheromone locally by a local updating rule on the visited edges. Secondly, A global updating rule is applied on the edges to modify the pheromone level that belong to the best ant tour found so far after all the ants have built their individual tours, the Behavior of ant’s colony is mimicked by the ACO algorithm while gathering food. Every ant starts from the nest and walks towards food till intersection is reached, where it has to decide which path to select. Choice is random in the beginning, but the majority of ants will be moving along the optimal path because of the colony’s collective intelligence after some time. Every ant, as he moves, chemical called pheromone is deposited thus marking the route taken. As time passes Pheromone trail evaporates. Hence, a shorter path will have more pheromone because it will get less time to evaporate before it is deposited again. Hence shorter routes will be selected with higher pheromone as every ant chooses paths that have more pheromone. [8]

B. Particle Swarm Optimization (PSO)
Small birds fly in coordination and show strong synchronization in the way they turn, flight initiation, and how they land without any clear leader. Certain rules are followed for their movement. To develop a particle swarm optimization (PSO) concept simulation of a bird swarm was used by Kennedy & Eberhart in 1995. Procedures of Searching by PSO can be described as follows:

1) Every particle evaluates the maximize function at each point it visits in space.
The Solutions of Travelling Salesman Problem using Ant Colony and Improved Particle Swarm Optimization Techniques

(IJSRD/Vol. 3/Issue 08/2015/137)

2) Every particle remembers the best value and its co-ordinates found so far (pbest).
3) The global best position (gbest) is known to every particle that is found by one of the flock fellow.
4) Using the pbest and gbest co-ordinates every particle calculates its new velocity and position.

The position of a particle is affected by the best position visited by the particle i.e. its own experience and the best particle’s position in its neighborhood which means the experience of neighboring particles. The best position in the neighborhood is denoted as the global best particle. Therefore, the resulting algorithm is referred as the gbest PSO at times when entire swarm is particle’s neighborhood. In general the algorithm is stated as the pbest PSO while smaller neighborhoods are used. Using fitness function that varies depending on the problem of optimization the performance of each particle is measured. Each Particle in the swarm is represented by the following features:
- Current position of the particle
- Current velocity of the particle

The particle swarm optimization is one of the evolutionary optimization techniques that conducts searches uses a population of particles. By moving through the problem space every particle has an updating position vector and updating velocity vector [4]. Boundary conditions are used in PSO in order to restrict particles to search within the solution space of interest during optimization procedure. Hence we used the constrained PSO that includes the information about the objective function and constraint violation. This information facilitated communications among swarm in the population space and assisted in selecting the leading particle. This algorithm is based on identifying and excluding the infeasible region of search space before and during the search i.e. it keeps check on the boundary conditions. PSO is started with a group of feasible solutions and to check whether new explored solutions satisfy all the constraints a feasibility function is used.

III. PSEUDO CODE FOR ACO AND IMPROVED PSO

A. Pseudo Code for ACO

Begin ACO procedure
Create pheromone trails and other parameters
while (criteria for termination not met)
{
    construct solutions
    update pheromone Trails
}
After process, results and output
End procedure of ACO

B. Pseudo Code for Improved PSO:

Begin procedure PSO
For (every particle)
Initialize particle
End of For loop
Do
For (every particle)
Compute fitness value
If the fitness value is more then the best fitness value (pBest) of past .
Then current fitness value is set as the new pBest.
For loop ends.
Select the particle with the gbest( best fitness value) .
For (every particle)
In the particle’s neighborhood find the particle with the best fitness.
Calculation of particle velocity according to the velocity equation
Apply the velocity constraints
According to the position equation; update particle position
Apply the position constraints
End For loop.
While (max iterations or min error criteria is not attained) end PSO procedure.

IV. SIMULATION RESULTS

The Simulation environment used for experimentation is MATLAB. Both ACO and PSO are implemented to solve conventional Travelling Salesman Problem to find out optimal path to generate minimum cost for the problem by visiting every city exactly once.

Figure 1, 2, 3 shows simulation results for ACO. It demonstrates following: a) coordinates of 10 cities b) Average tour distance Vs. Iterations c) Optimal path and Minimum cost (9,4,6,8,1,10,7,5,2,3 and 60.9622).

III. PSEUDO CODE FOR ACO AND IMPROVED PSO

A. Pseudo Code for ACO

Begin ACO procedure
Create pheromone trails and other parameters
while (criteria for termination not met)
{
    construct solutions
    update pheromone Trails
}
After process, results and output
End procedure of ACO

B. Pseudo Code for Improved PSO:

Begin procedure PSO
For (every particle)
Initialize particle
End of For loop
Do
For (every particle)
Compute fitness value
If the fitness value is more then the best fitness value (pBest) of past .
Then current fitness value is set as the new pBest.
For loop ends.
Select the particle with the gbest( best fitness value) .
For (every particle)
In the particle’s neighborhood find the particle with the best fitness.
Calculation of particle velocity according to the velocity equation
Apply the velocity constraints
According to the position equation; update particle position
Apply the position constraints
End For loop.
While (max iterations or min error criteria is not attained) end PSO procedure.

IV. SIMULATION RESULTS

The Simulation environment used for experimentation is MATLAB. Both ACO and PSO are implemented to solve conventional Travelling Salesman Problem to find out optimal path to generate minimum cost for the problem by visiting every city exactly once.

Figure 1, 2, 3 shows simulation results for ACO. It demonstrates following: a) coordinates of 10 cities b) Average tour distance Vs. Iterations c) Optimal path and Minimum cost (9,4,6,8,1,10,7,5,2,3 and 60.9622).

Fig. 1: Coordinates of cities

Fig. 2: Average of Tour Distance vs. Number of Iterations

Fig. 3: Optimal Path and Minimum Cost
Figure 4, 5, 6 shows simulation results for PSO. It demonstrates following: a) coordinates of 10 cities b) Standard deviation Vs. Iterations c) Optimal path and Minimum cost(7,10,3,8,6,2,9,4,1,5 and 58.5192)

VI. CONCLUSION AND FUTURE WORK

This paper presents a comparative assessment of most extensively used optimization algorithm techniques namely ACO and Improved PSO Optimization. The comparative results are shown and it is formulated that Improved PSO is better approach to solve the traveling salesman problem when it comes to generating optimal path to find the minimum cost. The ACO and PSO can be analyzed for future enhancement so that new research can be concentrated to generate better solution by reducing the limitations and improving the effectiveness of these algorithms. More possibilities for determining the best target through ACO can be developed and a plan to provide PSO with fitness sharing aiming to examine whether this helps in improving performance. Further work will be focused on: finding patterns in the evolved structures and hence this will help us design PSO algorithms that use larger swarms, developing better PSO algorithms for optimization and evolving PSO algorithm for other difficult problems. Another idea to be discovered is variable velocities in future researches. These techniques have immense potential and scope of applicability. For finding suitability of these techniques to certain applications; they need to be further explored. Also, there is a necessity to combine two or more techniques so that they nullify their respective limitations and complement each other. The idea that these kinds of techniques could be combined to give a better technique so as to give better results for more complex TSPs and other optimization problems has to be tested.

V. PERFORMANCE COMPARISON

Simulation results for both ACO and Improved PSO are generated by implementing both the algorithms in MATLAB Simulation tool. Figure 1, 2, 3 shows Simulation results for ACO whereas; Figure 4, 5, 6 shows Simulation results for Improved PSO. From Figure 1, 2, 3 and Figure 4, 5, 6 it can be seen that minimum cost for ACO is 60.9622 and minimum cost for PSO is 58.5192 therefore it can be said that Improved PSO is better than ACO because it gives optimal path to generate minimum cost in comparison to the other.

REFERENCE


