A Particle Swarm Optimization and Fuzzy Based Algorithm for Solving Classical Travelling Sales Person Problem

Azmath Mubeen¹, D.Hemalatha², D.Rama Krishna Reddy³

¹Asst.Professor, Dept. of Computer Science, Osmania University College for Women, Koti, Hyderabad, Telangana, India
²Asst. Professor, Dept. of Computer Science, Osmania University College for Women, Koti, Hyderabad, Telangana, India
³Asst. Professor in Computer Science, Dept. of Mathematics, UCS, Osmania University, Hyderabad, Telangana, India

Abstract— Travelling Sales Person Problem is one of the classical combinatorial optimization problems that belong to the NP-complete class. It is the problem of finding the optimized path for a given set of cities. The path is drawn in such a way that the salesperson has to visit each city exactly once. This paper provides an efficient method for solving the classical Travelling Sales Person Problem by using Particle Swarm Optimization (PSO) based on fuzzy logic. A particle is represented using a particle encoding/decoding scheme for the Travelling Sales Person Problem (TSPP). The searching ability of the PSO is expanded here by hybridizing the PSO with fuzzy logic. The local bests will not be the point of meeting for the particles and the global targeted goals can be searched in a shorter span of time if the PSO is correctly adjusted for the particles with the help of fuzzy logic rules. Numeric values for random weights have been taken to illustrate the efficiency of the system proposed for solving the Travelling Sales Person Problem.

Keywords: Travelling Sales Person Problem, Particle Swarm Optimization, Fuzzy Logic, evolutionary algorithm, Velocity Clamping, Construction Factor Method(CFM).

1 Introduction

The Classical Travelling Sales Person Problem has been solved by various methods and techniques to find an optimized solution. This paper present an approach which combines the Particle Swarm Optimization with Fuzzy logic rules. Particle Swarm Optimization(PSO) has been used to solve Optimization problems since it was proposed by Kennedy and Eberhart in 1995. The algorithm of PSO has the behavior of animal societies that don’t have any leader in their group or swarm such as bird flocking or fish schooling. If the group of animals does not have leaders will find the food by random and follow tone of the members of the group that has the closest position of the food source and find the better solution. Animals which have better solution will inform it to its flocks and others will move accordingly. This process happens repeatedly until the best condition or the food source is discovered. Travelling Sales Person Problem is the most basic computational problems for finding the optimized route in a network. This paper provides a novel approach to find the optimized solution for the single source and finding the shortest path by applying the fuzzy rules to the Particle Swarm optimization technique. The Travelling Sales Person Problem (TSPP) is one of the important basic computational problems in graph theory, and of greatest importance in communication networks. This TSP problem is concerned with exploring the shortest path from a particular origin to a specific target in a specified network however minimalizing the cost and perhaps taking particular limits into consideration. This problem has many varied applications, such as route scheduling in robotic systems [1], vehicle routing in transportation systems [2], sequence alignment in molecular biology [3], and traffic routing in communication networks [4], has made this significant computational problem the focus of interest in the scientific and research communities. The performance of a computer network is mostly motivated by routing, particularly in multi-hop networks, such as the Internet and mobile Ad-hoc Networks. An appropriate routing algorithm must be able to find an optimal path for communication within a specified period of time to fulfill the Quality of Service (QoS) [5, 6, 7]. Different distinguished known deterministic algorithms, such as Dijkstra [8] and Bellman-ford [9] are usually used to solve the
Travelling Sales Person Problem. Nevertheless, these classic algorithms experience some severe limitations, one of which is that they may not be used for networks with negative weights of edges. For example, in some communications networks, the weights can characterize the transmission line capacity, and the negative weights depict the links with gain rather than loss. Another problem of these algorithms is the point that they need complicated calculations for simultaneous communications involving rapidly changing network topologies such as the earlier-mentioned wireless ad-hoc networks [10]. Therefore, there is an evident requirement for more competent optimization algorithm for the Travelling Sales Person problem. In recent times, there has been a huge interest in the Particle Swarm Optimization (PSO) due to its huge capability as an evolutionary algorithm, which is built on the regular social activities of flocks of birds and schools of fish [11]. In this paper, a modified version of the PSO, based on the use of fuzzy logic, is proposed for computation of the single source Travelling Sales Person Problem, which can be of great use in improving the routing in multi-hop communication networks. However, the PSO itself is not flawless. It can fall into the local optimum trap and converges slowly. By combining the PSO with fuzzy logic [12], these problems can be solved.

2 Background

A number of scientists have created simulations of various interpretations of the movement of organisms in a bird flock or a fish school. Particle Swarm Optimization works on the swarm of candidate solution called Particle, each having a velocity that is update recurrently and added to the particle's current position to move it to a new position. Two different entities can hold identical manners and theories without hitting together, but two birds can occupy the same position in space without bumping into each other. Bird and Fish adjust their physical movement to avoid predators, optimize environmental parameters such as temperature, seek food and mates etc. The swarm of particles were initialized with a population of candidate solutions through d-dimensional problem space to search the new solutions.

3 Particle Swarm Optimization

The Particle Swarm Optimization algorithm is grounded on specific social behaviors noted in flocks of birds, schools of fish, etc., from which specific features of intelligence emerge. After its development by Kennedy and Eberhart [13] in 1995, this evolutionary model has been genuinely studied and developed in the past era. The standard PSO model comprises of a swarm of particles, moving interactively across the practical problem space to discover new solutions. Every particle has a position characterized by a position vector; where i is the index of the particle, and a velocity represented by a velocity vector. Every particle retains its particular best position so far in the vector pbest and the best position vector among the swarm is stored in a vector gbest. The exploration to find the optimal position (solution) advances as the particles' velocities and locations are updated. In each iteration, the fitness of each particle's position is calculated using a pre-defined strength function and the speed of each particle is updated using the gbest and pbest which were previously defined.

\[ V_{id} = W V_{id} + c_1 r_1 (p_{best} - X_{id}) + c_2 r_2 (g_{best} - X_{id}) \]

\[ i = 1,2,3, \ldots \ldots \ldots N \]

\[ d = 1,2,3, \ldots \ldots \ldots D \]

Where \( c_1 \) and \( c_2 \) are two learning factors that control the effects of pbest and gbest on the way the particles travel along the exploration space. In many of the researches done on the PSO, \( c_1 \) and \( c_2 \) are given the value of 2. Nevertheless, mainly the particles far from the global best reach velocities with large values; hence have enormous position updates, and may leave the limits of the search space as a result. Therefore, the speed of the particles must be controlled. Velocity clamping, which could be used in (1), gives a particle in a dimension the velocity of \( V_{id} \) if the right side of (1) for that particle goes beyond the maximum value in that dimension. It must be observed that various other improvements have been made into this algorithm. Manrice [14] proposed the use of a constriction factor \( \chi \) to prevent the velocity from increasing out of bounds so that there would be no need for clamping. In the Constriction Factor Method (CFM), (1) is modified as follows:

\[ V_{id} = \chi \left[ V_{id} + c_1 r_1 (p_{best} - X_{id}) + c_2 r_2 (g_{best} - X_{id}) \right] \]

\[ \chi = 2 \left( 1 - \phi - \sqrt{\phi^2 - 4\phi} \right)^{-1} \text{ if } \phi = c_1 + c_2 > 4 \]

Where \( w \) is the inertia weight which generally drops linearly in the interval [0,1]. \( c_1 \) and \( c_2 \) are positive constants, called acceleration coefficients, \( N \) is the total number of particles in the swarm, \( D \) is the dimension of the search space or in other words,
number of the factors of the function which are optimized, and \( r_1 \) and \( r_2 \) are two independently generated random numbers in the interval \([0,1]\). In (1), \( w \) is the inertia weight, which as stated earlier decreases in the interval \([0,1]\). \( w \) is one of the elements that regulates the velocity of the particles and hence their position updates. The larger the \( w \), the more globally the particles search the space; and the smaller the \( w \), the more locally the particles search the space. Thus, by reducing the \( w \) as the iterations move on, the global exploration transforms into a local search slowly.

### 3.1 PSO Algorithm

PSO Algorithm is a population based set of potential solutions evolves to approach a convenient solution for a problem. It is based on three factors:

1. The knowledge of the environment (its fitness value).
2. The individual’s previous history (its memory).
3. The previous history of the states of individual’s neighborhood.

In PSO algorithm, each individual is called a “particle.” Particles have memory and are subjected to a movement in a multi-dimensional space that represents the belief space. Each particle’s movement is the composition of an initial random velocity and to randomly waited influences that tends to return to the particles best previous position and also socially tends to move towards the neighborhood’s best previous position. There are two kinds of basic PSO algorithms that is continuous and binary. This algorithm uses a real-valued multi-dimensional space such as belief space and evolves the position of each particle in that space using the following equations:

\[
v(t+1) = (w \times v(t)) + (c_1 \times r_1 \times (p(t) - x(t)) + (c_2 \times r_2 \times (g(t) - x(t))
\]

\[
x(t+1) = x(t) + v(t+1)
\]

where:

- \( v(t+1) \): Component in the dimension \( d \) of the \( i \)th particle velocity in the iteration \( t \).
- \( x(t+1) \): Component in the dimension \( d \) of the \( i \)th particle position in the iteration \( t \).

- \( C_1, C_2 \): Constant weight factors.
- \( P(t) \): Best position achieved so long by the particle \( i \).
- \( G(t) \): Best position found by the neighbors by the particle \( i \).
- \( W \): Inertia weight.

Algorithm is initialized with the particles at random positions and then it explores the search space to find better solutions. In every iteration, each particle adjust it’s velocity to follow two best solutions.

1. The first is the cognitive part where the particles follow its own best solution found so far which is called as P-best (particles best value).
2. The other best value is the current best solution of the swarm that is the best solution of any particle of the swarm, which is called as G best (global best value).

### 3.2 The Optimal Set Of PSO Parameters

The PSO algorithm is well-selected parameter that can set good performance. In our TSP problem, we use \( n \) number of swarms, the number of particles in each swarm and inertia weight according to Eberhart and Shi [3], the acceleration coefficients \( c_1, c_2 \) represents the stochastic acceleration that force each particle towards G-best and P-best position. The fitness \( f \) can be calculated as the quality measures. Each particle has the position represented by \( I \) that is the index of the particle and a velocity represented by velocity vector \( i \). After every iteration, the best position vector among the swarm is stored in a vector. The update of the velocity from the previous velocity to the new velocity is been determined.

How it works:

The birds are the solutions which are termed as particles.
1. Each particle has a fitness value and our aim is to find out the fitness value as evaluated by the fitness function.
2. Now, each particle will have its own velocity and position which is calculated by the velocity and position function respectively.
3. Now, initially PSO is initialized with the group of particles whose parameters are altered during each iteration.
4. In each iteration, every particle updates its fitness value and it’s personal best value.
5. Meanwhile, during the process of each iteration the PSO reviews g-best (i.e., the best p-best value obtained by any particle to that point).

4 Modified PSO And Fuzzy-Based Method For Travelling Sales Person Problem

This section proposes a modified PSO algorithm with Fuzzy-based rules. Two main components of the proposed method are particle representation and fuzzy inference system, which are discussed in details.

4.1 Particle Representation

One of the highly significant issues in solving the Travelling Sales Person Problem and the ones similar to it, is how to encode a path in a network graph into a particle (or a chromosome). The way in which this encoding is prepared wholly influences the efficiency of the search process. In the method proposed in this paper, the position vector of a particle in the PSO is denoted by a priority vector, which comprises some guiding information about the nodes that represent the path in the graph. This technique of encoding, which was first used by Gen et al. [16] in a GA-based method, involves the significances of several nodes in the network. These priorities are primarily assigned randomly. The path is created as a sequence of nodes starting with the source node and ending at the destination node. According to the nature of the Travelling Sales Person Problem, as a path is being constructed, there are usually several nodes available for consideration, at each step of the path construction. In this approach, the node with the highest priority is chosen and the process continues until the destination node is reached. Fig. 1 illustrates a typical 20-node random network [17], on which the Travelling Sales Person Problem solving methods can be applied. The described encoding scheme is depicted in Fig. 2. p1, p2,... are the priorities of the nodes 1,2,... respectively. Fig. 2(b) shows a simple example of the encoding method explained above for the graph in Fig. 1. The path creation begins from node 1, and from the node adjacency relations, the node with the highest priority (node 4) is chosen as the next node in the path. Then, out of all the possible non-visited nodes that can be visited from node 4, the node with the highest priority (node 9) is chosen. The method is repeated until a complete path (1, 4, 9, 15, 14, 20) is obtained.

Figure 1. A typical 20-node random network. node numbers are encircled. The weights of the connecting edges are also shown adjacent to the corresponding edges [17].

4.2. Fuzzy Inference System

Fuzzy inference is the process of making the mapping from a given input to an output using fuzzy logic. The process of fuzzy inference involves membership functions, logical operations, and If-Then rules. There are two types of fuzzy inference systems, Mamdani-type [18] and Sugeno-type [19]. These two types of inference systems differ slightly in the way outputs are established. Mamdani’s fuzzy inference method, which is used in this paper, is the most regularly seen fuzzy methodology. Mamdani’s approach was amongst the first control systems built using fuzzy set theory. Sinking into the trap of local optimum and slow merging are of the most important limitations of the PSO. There have been many
schemes proposed to solve the first problem, all of which comprise detecting the local optimum and preventing it. In [20], to prevent from obtaining the local optimum, when the velocity of the particle is lower than a specific level, but the fitness is not appropriate, a function is used to give a jolt to the particle and increase its velocity. In [21, 22], a non-linear function for reducing the inertia weight is used to rise the velocity of a particle when the inertia weight is small, but the fitness is undesirable. All these methods prevent the particles to converge to a local optimum and some even speed up the convergence.

In this paper, a fuzzy-based method proposed by Noroozi and Meybodi [12] is used to overcome the above-mentioned shortcomings of the PSO. In this method, a variable called the CBPE (Current Best Performance Evaluation), which indicates the fitness level of a particle at the moment, is used. CBPEmin is the best fitness attained so far and CBPEmax is the worst fitness attained so far. In (5), a normalized value NCBPE in the interval [0, 1] is obtained using the three mentioned-above variables:

\[
NCBPE = \frac{CBPE - CBPE_{\text{min}}}{CBPE_{\text{max}} - CBPE_{\text{min}}} \quad (5)
\]

In this method, a fuzzy function is defined with the parameters d1, d2 and NCBPE as its input and w as its output (Fig. 3).

\[
d1 = |p_{\text{best}} - x|, \quad d2 = |g_{\text{best}} - x| \quad (6)
\]

In (6), d1 and d2 represent the distance between the current position of the particle and its local best, and the global best, respectively. The lingual values “low”, “medium”, and “high” are used to describe the parameters. d1 and d2 are determined with respect to hithe size of the search space; therefore, these three parameters are the basis for the fuzzy system to decide the value of w, which is in the interval [0,1]. Choosing the correct fuzzy rules has a direct influence on the obtained results. A number of the rules used in this system are illustrated in Table 1. It should be noted that a large number of rules in the system can not affect the result significantly, but the quality of the chosen rules is what produces accurate results.

![Figure 3. Fuzzy inference system for solving shortest path problem.](image)

**Table 1. Few of Fuzzy Inference System Rules.**

<table>
<thead>
<tr>
<th>Rules</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LOW</td>
<td>LOW</td>
</tr>
<tr>
<td>2</td>
<td>LOW</td>
<td>NOT LOW</td>
</tr>
<tr>
<td>3</td>
<td>LOW</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>4</td>
<td>HIGH</td>
<td>HIGH</td>
</tr>
</tbody>
</table>

- Rule 1: if d1, d2, and NCBPE are low, the particle is near to the optimal best and the fitness is acceptable; therefore w is given a low value so that the search continues around the global optimum.
- Rule 2: if d1 and d2 are low, but NCBPE is not low, it means that the particle is close to the optimum, but the fitness is not acceptable (local optimum); therefore w is given a high value to increase the particle’s velocity and change its position.
- Rule 3: if d1 is low, d2 is not low, and NCBPE is medium, it means that the particle is close to the local optimum but not close to the global optimum; hence w is given a high value to increase the particle’s velocity and change its position.
- Rule 4: if both d1 and d2 are not low; the particle’s velocity must increase; therefore w is given a high value. Conferring to these fuzzy inference rules, the inertia weight w is modified.

## 5 Experimental Results

Several networks with many numbers of nodes are used to assess the functioning of the proposed approach. Networks with many numbers of nodes are created to investigate the quality of solution and the convergence speed of the proposed method. The highly apparent advantage of PSO is that the convergence speed of the swarm is very high. The weight variance of the present position of the particle swarm and the best position of the swarm Pbest will also be added to velocity vector for adjusting the next population velocity. These two alterations will enable particles to search around two bests. These Networks are arbitrarily generated with the maximum number of nodes and edges are given random values between (0, 100). To have a improved assessment, the proposed algorithm is run 100 times for each network. The other PSO parameters are chosen as: w reduces linearly from 0.9 to 0.2; c1 and c2 are chosen to be 2. The performance of the algorithm is assessed by success rate which is defined as the number of...
times the shortest path is found over the number of runs. The hit rate for different swarm sizes between 10 and 40 is found to contrast the PSO with the proposed method which is illustrated in Fig. 4. In Fig. 5, the Average Best So Far is shown in various iterations of both algorithms. As it can be seen evidently, the proposed approach presents more precise results. Although both algorithms seem to be similar in the beginning iterations, the proposed method merges to a improved solution as the time passes. A unique tour for each of the test is found. Here The testing process using randomly chosen cities is more objective. The use of the fuzzy logic along with PSO helps in finding random city sets which leads to find an optimized solution. All statistics are generated after 100 runs on each city set. When number of iterations is taken as 100, the average results show considerable difference. The tours of the cities with 100 iterations obtained leads to finding optimized results by generating a tour in relatively short time. The experiments show that PSO with fuzzy logic finds better solutions for instances with up to 100 cities. Both average and best results are better than other algorithms. For city sets with 50 or less, PSO with fuzzy rules finds optimum routes in every execution. Corresponding to Fig. 4 and 5, it is easy to see that by using a fuzzy inertia weight the performance of PSO can be enhanced and have similar or improved results than that of PSO with a linearly dropping inertia weight. Figure 5. Hit rate vs. Swarm Size for a network of 30 nodes

Particle swarm optimization is a simple algorithm that seems to be effective for optimizing a wide range of operations. Significantly, it lies between genetic algorithm and evolutionary programming. It is highly dependent on stochastic processes with the variation towards the t-best and g-best by particles swarm optimizer to crossover operation utilized by Travelling Sales Person Problem. Travelling Sales Person Problem provides an optimized solution for networks to find the optimized route. A hybrid PSO–fuzzy search algorithm for solving the single source optimized path for TSP problem is presented in this paper. The method takes advantage of an efficient encoding mechanism in the PSO so as to include the parameters of the path graph in the representation itself. Additionally, in order to enhance the search efficiency, the inertia weight whose right values can prevent the search from falling in the trap of local optima, is determined using fuzzy rules. The results illustrate that the variation in values improves the performance of the algorithm significantly by achieving a success rate of 0.99.

7 References