Fuzzy Inspired Hybrid Genetic Approach to Optimize Travelling Salesman Problem

Bindu
Student, JMIT Radaur
binduaahuja@gmail.com

Mrs. Pinki Tanwar
Asstt. Prof, CSE, JMIT Radaur
pinki.tanwar@gmail.com

Abstract

One of the category of algorithm Problems are basically exponential problems. These problems are basically exponential problems and take time to find the solution. In the present work we are optimising one of the common NP complete problem called Travelling Salesman Problem. In our work we have defined a genetic approach by combining fuzzy approach along with genetics. In this work we have implemented the modified DPX crossover to improve genetic approach. The work is implemented in MATLAB environment and obtained results shows the define approach has optimized the existing genetic algorithm results.

Keywords: Genetics, Travelling Salesman Problem, NP complete, Fuzzy approach, DPX crossover

1. Introduction

Travelling salesman problem is the most common used algorithmic concept used by most of the researchers working on optimizing the network communication. The Travelling salesman problem is easy to define but very hard to solve it. The problem is to find the shortest possible tour through a set of N vertices so that each vertex can visit exactly once. This problem is known to be NP-hard, and cannot be solved exactly in polynomial time. To solve this problem in the effective time is always a challenge for the researchers. We are also working in the same direction to find the optimal solution to the problem. The problem can have number of feasible solutions but the outcome that will gives the best result in terms of time and space will be represented as the optimal solution.

This means that a very large number of solution need to be tested in order to determine which solution is optimal[1].

In general terms or discussions in reference to TSP, cities are often called ‘nodes’. The line connecting two cities is called an ‘edge’. The distance between the cities are defined along with the edges. In any valid solutions, all nodes must be visited, but not all edges will be used. A solution can be described by listing the nodes visited in the order they are visited. Alternatively, a solution can be described as an unordered list of which edges are used[2].

There are a number of different variations of the problem. A TSP can be Euclidean or non-Euclidean. A Euclidean problem is one that could be drawn as a map on a Euclidean surface, such as a flat piece of paper. The problem shown as a drawing of the cities above could also be written as a chart showing the distance between each city, as below[2].

An algorithm for a Polynomial-time solvable problem might be too expensive in practice. A wide range of different algorithmic strategies exists to deal with such problems. These approaches can be roughly classified as follows:

1. Exact algorithms.
2. Approximate algorithms

Exact algorithms are guaranteed to find an optimal solution and prove its optimality for every finite size instance of combinatorial optimization problems (COPs) within an instance-dependent, finite run-time,
or prove that no feasible solution exists. If optimal solutions cannot be computed efficiently in practice, the only possibility is to trade the guarantee of optimality for efficiency. In other words, the guarantee of finding optimal solutions can be sacrificed for the sake of getting very good solutions in reasonably short time by using approximate algorithms. There are many different variations of the Travelling Salesman Problem.

Shortest Hamiltonian Path Problem: If in a graph, each edge has a weight and two nodes Vs and Vt are given and objective is to find the shortest Hamiltonian path from Vs to Vt. If an edge from Vt to Vs is added and give it weight M where M is large and positive, then optimal.

The Asymmetric Travelling Salesman Problem: When the cost of travelling from city i to city j is not the same as the cost from city j to city i then it is called Asymmetric Travelling Salesman Problem.

The Multisalesmen Problem: It is the same as the standard TSP except that there is more than one salesman. Problem is to decide where to send each salesman so that every city is visited exactly once and each salesman returns to the original city.

The Travelling Salesman Problem has many different real world applications, making it a very popular problem to solve. Some instances of the vehicle routing problem can be modeled as a Travelling Salesman Problem. Here the problem is to find which customers should be served by which vehicles and the minimum number of vehicles needed to serve each customer. There are different variations of this problem including finding the minimum time to serve all customers. Some of these problems can be modeled as the TSP[10].

The problem of computer wiring can also be modeled as a TSP, where number of pins represents several modules. Subsets of these pins are connected with wires in such a way that no pin has more than two wires attached to it and the length of the wire is minimized. The scheduling of jobs on a single machine given the time it takes for each job and the time it takes to prepare the machine for each job is also TSP. Objective is to minimize the total time to process each job[6]. A robot must perform many different operations to complete a process. In this application, as opposed to the scheduling of jobs on a machine, there are precedence constraints. This is an example of a problem that cannot be modeled by a TSP but methods used to solve the TSP may be adapted to solve this problem[3][4].

2. Problem Statement

The Travelling Salesman Problem is one of the most widely studied problems in computational mathematics. One of the reasons for this might very well be the ease of formulating and understanding the problem. This problem comes under the Artificial Intelligence and defines some sub problems like shortest path, Hamiltonian Cycle Problem etc. The TSP is the NP complete problem and our objective is to find the solution of the problem in a definite and optimized time. NP hard problems can be solved by number of exact algorithms with guaranteed optimal solution but the major drawbacks to take very large computational time. So for this, various approximate algorithms like GA has been developed to find near to optimal solution in very small computational time. A hybrid version of SGA has been proposed in which we are combining the Fuzzy Logic along with Genetic Algorithm. In DPX cross-over operator, all the edges which are not common in the other parents are then removed. The off-spring is then left with different sized “chunks” or city segments, which are actually the assignment sub-tours that are common in both the parents, these broken edges are then recombined without replacing any edge originally broken. The original DPX operator reconnects remaining edges using a greedy procedure. Due to the large search space in Travelling Salesman Problem (TSP), it is expected that random generation of initial solutions provides relatively weak results. For this, initial solution is obtained by application of some heuristics for finding near to optimal results in a very reasonable time. The fuzzy logic is basically applied on DPX Crossover. The present work is an effort towards the development of Hybrid Genetic Algorithm (HGA). The proposed system will give more optimized results than existing.

3. Research Methodology

We are providing the solution of above said problem using the genetic approach. The proposed system is fuzzy inspired Genetic approach to resolve the Travelling Salesman Problem. The Fuzzy logic is implied on Crossover Layer of Genetics. In this present work to perform the optimization DPX
(Distance Preserving Crossover) is implied. Instead of selecting the random values from the parent, a Fuzzy rule is defined here to select the optimal sequence. The proposed system is about to optimize the results driven from the Genetic algorithm in case of DPX Crossover.

3.1 Proposed Algorithm

The proposed Algorithm for Solving TSP i.e. Hybrid Genetic Algorithm is described as below:

1. Define the initial random population
2. Define the fitness rule to minimize the distance covered by visiting all cities
3. For i=1 to MaxIterations
   [Repeat steps 4 to 7]
4. Select two random parents from the population set that follow the fitness rule. Called parent1 and parent2
5. Perform the fuzzy inspired DPX Crossover on these two parents to generate the child node.
   ChildNode= FuzzyDPX(parent1,parent2)
6. Perform the Random Mutation algorithm.
7. Recombine the obtained value in the population set.
8. Return Optimized Sequence
9. Generate the graph of path sequence.

4. Result Analysis

   Genetic Algorithm computational analysis for both DPX and fuzzy inspired DPX has also been made for comparison. Results obtained from DPX and fuzzy inspired DPX for Traveling Salesman Problem in MATLAB environment are presented as below.

As we can see in Figure 1, the initial city network is shown. We have taken 100 number of cities and there is path between each pair of city with some specific length. The axis values here shows the maximum available area to the city.

Figure 2 Iteration wise Fitness Value of DPX

The work here is processed for 100 iterations as shown in Figure 2. In this figure the output is shown in tabular form with two columns. First column shows the iteration number and second column shows the
As we can see with the iterations the total distance covered is being reduced and is 36544.6 in case of DPX.

The work here is processed for 100 iterations as shown in Figure 3. In this figure the output is shown in tabular form with two columns. First column shows the iteration number and second column shows the total distance path. As we can see with the iterations the total distance covered is being reduced and is 32321 in case of fuzzy inspired DPX.

From Table 1 we conclude that Fuzzy Inspired DPX calculate distance 32321 whereas DPX calculate 36544.6 distance after 100 iterations. Thus Fuzzy Inspired DPX gives better result than DPX.

5. Conclusion

The present work deals with the optimization of travelling salesman problem which belongs to family of NP hard. NP hard problems are very difficult to solve as optimal solution in a reasonable time is not possible. The various exact methods such as branch and bound, Integer programming, Dynamic programming etc. can be used for exact solution but they consumes more time and hence not desirable. So, researchers are using various approximate algorithms for solving NP hard problems in a very reasonable time. Genetic algorithm, simulated annealing etc belongs to the main class of approximate algorithms which are based on the population and local search techniques inspired by nature. However main limitation of that algorithm is that they provide near to optimal solution in a very reasonable time but does not
guarantee optimality. Hence, different researchers are still working on such approximate algorithms to improve its optimality in a reasonable time for NP hard problems.

Simple genetic algorithm (SGA) have been used in literature for NP hard problems and SGA generate initial population randomly and drawbacks of the algorithm is that, the choice of the initialization procedure has an important influence on the quality of solution and a better initial solution might provide better results. Due to the large search space in TSP, it is expected that random generation of initial solutions provides relatively weak results. For this, initial solution is obtained by application of heuristics for finding near to optimal results in a very reasonable time. In this case, the special heuristic which is proposed for generations of initial chromosome and hybrid to SGA and named as HGA.

6. Future Enhancement

Implementation of Hybrid Function: A hybrid function is another minimization function that runs after the genetic algorithm termination. Any other Meta-heuristics may be hybrid after GA to improve the solution quality.

Designing optimal parameters for HGA: In this present work, implementation of fixed parameters such as stopping limit, crossover and mutation etc has been applied. The work can be extended for designing the optimal parameters through statistical approach.

7. References


