Meta Heuristic Optimization of TSP Using Genetic Algorithm

Richa Singh *, Suraiya Parveen, Aparna
Department Of Computer Science and Engineering, Jamia Hamdard, Faridabad, India

Article Info

Article history:
Received 2 January 2014
Received in revised form
29 January 2014
Accepted 20 February 2014
Available online 15 March 2014

Keywords

Heuristic,
Combinatorial problem,
Genetic Algorithm,
Travelling Salesman Problem

Abstract

Initiation of this paper demarcates the limits of artificial intelligence, as it calls artificial intelligence a science for its extensibility and genetic algorithm to provide an affable decision to solve a popular routing problem named as Travelling Salesman Problem. This study will help more in moving to a world where a computer will be able to program based on natural selection and evolution.
The travelling salesperson (or, salesman) problem (TSP) is an important combinatorial optimization problem. A combinatorial optimization problem helps to find an optimal object from a finite set of objects which does not follows an exhaustive search but the set of feasible solution is discrete.
TSP aims to find the shortest path that travels through every city in a provided set of cities exactly once and travels back to the initial city using genetic algorithm. TSP is complex as it is a NP-complete problem. This literature proposes a heuristic approach for solving TSP: To find shortest path that travels through every city in a provided set of cities exactly once and travels back to the initial city. Proposed solution will provide a faster solution but not necessarily an optimal solution.

1. Introduction

AI deals with the circumstances surrounding computers; therefore it is related to computer science [1]. It is also bounded to cognitive science [1]. This paper concerns about tasks performed by AI will require intelligence thinking.AI can be viewed as an automated tool for answering questions regarding the behaviour of human intelligence. [2]

Genetic Algorithms came into existence in 1975 by Holland. [3] These algorithms represent the problem search space of a problem as a set of individual elements. These individuals can be Character strings (or matrices). [4]

The purpose of using a genetic algorithm is to find the individual from the search space with the best "genetic material". [4] An evaluation function helps to measure the quality of an individual.
Genetic algorithm usually proceeds as:

1. Choose the initial search space
2. Determine the quality of the population
3. Iteration parents are chosen from the population. These parents will help in future for evolution either by mutation or crossover. [4]

   Mutation can be used where local optima are to be avoided and crossover can be used to improve the quality of the search space.

Corresponding Author,
E-mail address: itrichasingh@gmail.com
All rights reserved: http://www.ijari.org

2. Elements of GA

Methods called "GAs" must have: Chromosomes search space, fitness, crossover to evolve into a new generation, and random mutation. [3]

The gene in a genetic algorithm initial population takes the form strings of bit. Each specific location in the gene has two possible states: 0 and 1. [4] Each genetic code for a gene can be considered as a point in the provided search space of candidate solutions.

The fitness of a gene depends on how well that gene solves a given problem at hand. [5]

3. Example of Fitness Functions

A recurrent application of Genetic Algorithms is function optimization. The objective is to achieve a set of parameter values which will maximize complex multi parameter function. [5]

4. Evolutionary Computation in genetic algorithm

A lot of computational problems require a computer program to have a learning algorithm that performs well in a refining environment. [5] This is similar to the problems in robot control where a robot has to perform a task in a refining environment with computer interfaces that must be extensible to the eccentricity of different users. [4]

Some problems would like to have innovative computer programs so as to construct an original solution; such algorithm helps in producing a solution for a computational task or may even provide a scientific discovery. [5]

Finally, there exist a set of computational problems which have complex solutions that are highly complex to program. Example: Problem of developing artificial intelligence. Initially AI practitioners thought that it would be simple to encode the rules that would create intelligent program. This early optimism gave expert system as results. Currently many AI researchers believe that encoding the "rules" containing intelligence by hand in top down manner is very complex for scientists. Instead they think that the best way to develop artificial intelligence is by using a "bottom−up" approach where simple rules are written manually for once, and these rules interact with the help of provided applications to produce complex behaviours. [5] An example of such evolution is Connectionism (i.e., the study of computer programs inspired by neural systems). In connectionism the rules are typically simple "neural" threshold, activation spreading, and strengthening or weakening of connections; the hoped−for emergent behaviour is sophisticated pattern recognition and learning. [6]

5. Evaluation of Travelling Salesman Problem

In this section, we briefly summarize some aspects of the TSP which are important for the implementation of the TSP package described in this proposal. [4] It has to be noted that in the origin of the TSP is the term distance to be used. [9] Distance can be used as a substitute of dissimilarity or cost and triangle equality measures are obeyed unless any restriction is specifically made. [4] There exist explicit difference between TSP and ATSP. For the symmetric case (normally referred to as just TSP), for all distances in X the equality $x_{ij} = x_{ji}$ holds, i.e., it does not matter if we travel from i to j or the other way round, the distance is the same. In the asymmetric case (called ATSP), the distances are not equal for all pairs of cities. [6] Problems of this kind arise when we do not deal with spatial distances between cities, e.g., with the cost or time associated with travelling between locations, where the price for the plane ticket between two cities may be different depending on which way we go.

TSP is a complex problem ;A set of given *p* cities is to be travelled by a salesman, each city must be visited only once, starting from one, considered as source and coming back to it.

The cost of the conveyance among the cities (which ever combination possible) is being provided. The journey as desired must minimize the cost of transportation.

Assume the cities to be labelled as numbers from 1 to *p*, and let city 1 to be the source city from where the salesman initiates journey and will conclude.

Let's say that c(m,n) is the visiting cost from m to n. There can be c(m,n)<>c(n,m).[7]

Evidently all the viable solutions are *(p-1)!*.A traditional search suggests calculating the cost of each viable solution and then determining the one with minimum cost to be declared as the desirable solution. This tedious task will need at least *(p-1)!* Steps. [7]If for example there were 15 cities the steps required are *(p-1)! = (15-1)! =14!* steps. Even if a *msec is required for each step* would need centuries for calculations. Such exhausting examination of all viable solutions is rather not likely.

A heuristic algorithm is used here as there is no other algorithm available to find the best solution. [9] The algorithm says whenever the salesman is in town *m* chooses the next city *n*, for which the *c(m ,n)* cost, is the minimum among all *c(m, k)* costs, where k is the pointers to cities which are still to be visited.[12] A case may arise where two cities will have the minimum cost. In such condition the city with minimum k is chosen [12] it is a greedy algorithm that selects the cheapest visit in every iteration apart from the care if that will lead to a correct or incorrect result.

Travelling Salesman Problem - AlgorithmInput:
Cities: p
Array of costs c(m,n)
Where m,n=1,..p (We begin from city number 1)
Output:
City's vector and the total cost.

TC=0
c=0
Veach=0
x=1 (*x=visited city)
(* computation of round and cost)

for round=1 to p-1 do
Choose of pointer n with
min=cost(x,n)=min{cost(x,k);v_{each}(k)=0 and k=1,..,p}
c=c+min
x=n
TC (round) =n
end round-loop
TC (p) =1
c=c+cost(e,1)

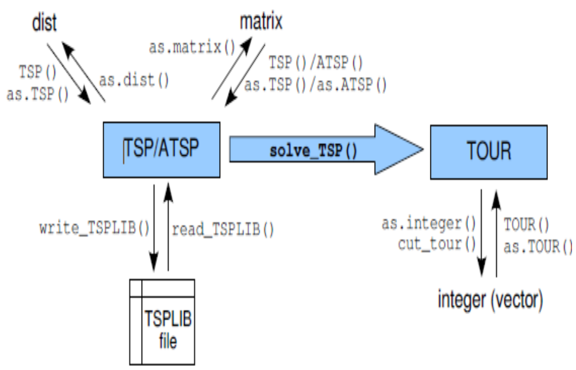

Fig: 1. An overview of TSP classes. [7]

By using full backtracking, the optimal solution can always be found, but the running time would be immense. [10] Therefore, only limited backtracking is allowed in the procedure, which helps to find better local optima or even the optimal solution.

6. Result and Conclusion: GA a solution for TSP

The paper may help to find the optimal solution for the TSP using genetic algorithm. This provides the desirable solution which will start from the source city and will travel back to it in the end. Formally, the TSP can be stated as follows. The distances between n cities are stored in a distance matrix D with elements d_{ij} where i; j = 1:: : n and the diagonal elements d_{ii} are zero. A tour can be represented by a cyclic permutation π of {1, 2, .., n}where π (i) represents the city that follows city i on the tour.[9] The travelling salesperson problem is then the optimization problem to find a permutation π that minimizes the length of the tour denoted by

$$\sum_{i=1}^{n} d_{i\pi(i)}$$

For this minimization task, the tour length of (n - 1)! permutation vectors have to be compared. This results in a problem which is very hard to solve and in fact known to be NP- complete (Johnson and Papadimitriou, 1985b) [9].Solving TSPs is an important part of applications in many areas including vehicle routing, computer wiring, machine sequencing and scheduling, frequency assignment in communication networks (Lenstra and Kan, 1975; Punnen, 2002)[7]. Applications in statistical data analysis include ordering and clustering objects. For example, data analysis applications in psychology ranging from profile smoothing to finding an order in developmental data are presented by Hubert and Baker (1978) [8].

An exact solution for 15,112 German towns from TSPLIB was found in 2001 using the cutting-plane method proposed by George Dantzig, Ray Fulkerson, and Selmer Johnson in 1954, based on linear programming [8]. The computations were performed on a network of 110 processors located at Rice University and Princeton University. [10] The total computation time was equivalent to 22.6 years on a single 500 MHz Alpha processor.[11] In May 2004, the travelling salesman problem of visiting all 24,978 towns in Sweden was solved: a tour of length approximately 72,500 kilometers was found and it was proven that no shorter tour exists.[11]

References

[1] R. A. Brooks, Intelligence without representation, Artif. Intell, 47 (19’31) 139-159.
[2] D. A. Pomerleau, J. Gowdy C. E. Thorpe, Combining artificial networks and symbolic processing for autonomous robot guidance, J. Eng. Appl. Artif. Intell. 4 (1991) 961-967
[3] D. Waterman, Generalization learning techniques for automating the learning of heuristics, Artif.Intell. 1 (1970) 120-170.
[4] M. Lalena, “Traveling Salesman Problem Using Genetic Algorithms, http://www.lalena.com/ai/tsp/, 2003
[5] S. R. Shubhra, B. Sanghamitra, K. Pal. Sankar, “New Operators of Genetic Algorithms for Traveling Salesman Problem”, IEEE, 0-7695-2128-2/04, 2004
[6] B. P. Buckles, P. E. Petry, R. I. Kuester, “Schema Survival Rates and Heuristic Search in Genetic Algorithms”, IEEE, 1990, PP 86-91
[7] M. Gendreau, A. Hertz, G. Laporte, “A tabu search heuristic for the vehicle routing problem”, Management Science 40(1984) 1276–1290

[8] Kuk-Hyun Han Genetic quantum algorithm and its application to combinatorial optimization problem 0-7803-637 5-2

[9] S. R. Shubhra, B. Sanghamitra, K. Pal. Sankar, "New Operators of Genetic Algorithms for Traveling Salesman Problem", IEEE, 0-7695-2128-2/04, 2004

[10] Naef Taher Al Rahedi, Jalal Atoum, "Solving the Traveling Salesman Problem Using New Operators in Genetic Algorithms", American Journal of Applied Sciences 6 (8): 1586-1590, 2009

[11] Buthainah Fahran Al-Dulaimi, Hamza A. Ali, "Enhanced Traveling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA)", World Academy of Science, Engineering and Technology 38 2008

[12] Michael Hahsler, Kurt Hornik "from Introduction to TSP: Infrastructure for the Traveling Salesperson Problem", publisged in March 24, 2009

.