

The traveling salesman problem computer science essay

[Technology](#), [Computer](#)



Travelling sales man problem is one of the challenging problems in the real life and also most well studied combinatorial optimization problem. Many Researches from different fields like operational research, algorithms design and including artificial intelligence attract by it. This problem has been studied by different researches and come up with different solutions and this problem has been solved by using different algorithms like Blind search, Branch and Bound Search , Heuristic algorithm and Genetic algorithms etc. the problem was formulated as a mathematical problem in 1930 and later it is used as bench mark for many optimal solution.

1. 1 TRAVELING SALESMAN PROBLEM:

A Travelling salesman has a task of visiting N number of cities. He will start from a home location and want to visit each city just once and return back to the original location from where he starts. Travelling salesman route will be plan in such a way that in a given N number of cities cost of travelling from one city to any other city what is the minimum round trip route that visit each city once and then return to the starting place. The goal is to find the shortest tour that visit each city in a given cities exactly ones and then return to the starting city. The only solution to the travelling salesman problem is to calculate and compare the length of all possible ordered combinations.

1. 2 History of travelling salesman problem:

The travelling salesman problem was treated by a Irish mathematician sir William Rowan Hamilton and British mathematician Thomas Penyngton

kirkman in the early 1800s. Hamilton and kirkman work on game called Hamilton Icosian game that requires player to complete tours through 20 points using only the specified connections.

The general form of the travelling salesman problem studied by mathematician Karl menger during 1930s. He defines the problem using brute force algorithms and observed nearest neighbour heuristic non optimality. Soon after the name travelling salesman problem introduced by hassler Whitney at princeton university. In the 1940s the Travelling salesman problem was studied by statisticians Mahalanobiss, Jessen, Gosh, Marks. Among them P. C . Mahalanobiss took a sample survey of acreage under jute in Bengal discussed aspects of travelling salesman solutions through randomly chosen locations in the Euclidean plane. And this work is deal with survey of form lands one of the major cost to carrying out the survey was the transport ion of men equipment from one survey point to next. During the period between 1950s and 1960 the problem becomes more popular in scientific group in Europe and USA. a number of solutions designed by George B. Dantzing, Fulkerson and Johnson(1954) . in 1954 Heller published an 88 report which contains many basic solutions on the travelling salesman polytype. In 1957 L. L. Barachet published graphic solution of travelling salesman problem which describes an enumeration scheme for computing near optimal tours.

In 1964 R. L Karg and G. L. Thompson were applied heuristic algorithm for a 57 city problem. During the period R. M Karp and M. Held published an article about the travelling salesman and minimum spanning tree which

introduced one tree relaxation of the travelling salesman problem and using node weights to improve the bound given by optimal tree. In 1990s Bixby and Cook developed the programme for travelling salesman problem which are using recently.

2. 0 GRAPH:

2. 1 Travelling salesman problem modelled as graph:

Travelling salesman problem can be modelled as Graph where as the cities are the graph vertices, path is graph edges and path distance is edge distance. Our goal is to find the shortest tour that visits each city in a given graph exactly ones and then return to the starting city.

3. 0 Search Algorithms:

To solve travelling salesman problem we can use different types of algorithms like blind search, heuristic algorithms, uniform cost search etc

3. 1 Blind search:

Blind search algorithms are such algorithms which does not contain domain knowledge of the problem and it search blindly, hence it is called blind search algorithm. The only thing a blind search can do is it describes a non goal state from goal state. Assume that you are at city A and you want to reach at city D, if we draw a search tree level 1: city E, city B, city C. a blind search will have no idea which node should explore first, hence it explore each and every node blindly. In blind search we may not get information we

can use. We just be looking for an answer and we won't no until we found it. The blind search is also called as an informed search.

Blind search is divided into two searches Algorithm they are

1) Breadth first search and

2) Depth first search

3. 2 Breadth first search:

Breadth first search is a search which starts at one node and it expands the all the neighbour node then after completing those nodes, it expands remaining nodes unexpanded it process is continues up to reaching a goal.

The technique followed by breadth first search is FIFO (first in first out) queue, the difference between breadth first and depth first search is, the depth first uses stack i. e.

LIFO (last in first out). In this logic the items which are added is equal to the item which are deleted. This process continues up to reach a goal.

3. 3 Depth first search:

Depth first search is a general technique for traversing a graph. The technique which organise the to Do list was stack that is last in first out (LIFO). Depth first search start at one node and it explore as far as possible along each branch until a required goal node is found.

Then it takes backtracking and return to the next node which it hasn't finished exploring and it keep on repeat the same procedure until it reaches to its goal. If Depth first search goes down a infinite branch and if it does not find a goal state or if it does not find the solution may be a better solution at a higher level in tree. Therefore depth first search is neither complete nor optimal.

ALGORITHM:

INPUT : A connected graph G , a starting vertex 1

OUTPUT: An ordered spanning tree T of graph G with root vertex 1

Initialize tree as vertex 1

Initialize S as the set of proper edges incident on vertex 1

While s not equal to null

Let $e = \text{dfs next edge}(G, S)$.

Let w be the non tree end point of edge e .

Add edge e and vertex w to tree T .

Update Frontier(G, S).

Return Tree T .

4. 0 DEPTH FIRST SEARCH APPLY TO GRAPH STARTING AT NODE 1:

For the above tree diagram we are using Depth first search, from the graph the sales man starts at node a and visits each node and return to original node

Step1: from node 1 there is a three ways to travel i. e. node 2, node3 and node5 in level 2

Step2: by following depth first search condition LIFO it expands node 2 until the goal reach

So for node 2 there is a two ways to travel i. e. node5 and node4 there is only one way for sales man to reach node 3 so this is not minimal path to travel.

After expanding first branch sales man reaches node 3 from node 3 sales man reaches node4

With path length of 8 and from node 4 he reaches node2 with path length of 5 and from node 2 sales man reaches node5 with path length of 6 then length of whole sub tree is 27.

After expanding of second branch sales man expands node 5, from node 5 sales man travels node 2 with path length of 6 then from node2 sales man reaches node 4 with path length of 5

From node 4 sales man reaches node 3 path length of 8. then length of whole sub tree is 26

Here we follow the condition that sales man will not visit those not which have already visited. the left vertices in the graph are chosen before right, from above to roots the minimum spanning tree path is $1-5-2-4-3 = 26$. this is the best minimum root for sales man to visit each city started at city1.

4. 1 DEPTH FIRST SEARCH APPLYING TO THE GRAPH STARTING AT NODE 2:

For the above tree diagram we are applying depth first search on which sales man starting at node 2. From starting city we have three ways to travel i. e. node 1, node4 and node5. From node 1 sales man has again to two ways to travel i. e. node3 and node5. from node 3 it reaches node 4.

but here if we want to visit node5 we have to backtrack again it will not satisfy the condition visiting a city once so we didn't reach goal here sales man expands neighbour node 4, from node 4 sales man reaches node 3 with path length of 8, then sales man processed to node 1 with path length of 8, then sales man visits node 5 from node 1 with path length of 7.

The length of whole sub tree is $2-4-3-1-5 = 28$.

After expanding node4 travel sales man moves to node 5, from node 5 he reaches node1 path length of 7, from node 1 sales man reaches node 3 the path length of 8 and he proceeds to node 4 from node 3 with path length of 8, then whole length of sub tree is $2-5-1-3-4 = 29$.

Here sales man follow the condition that visiting each city, then the the best minimum cost route starting at node 2 is $2-4-3-1-5 = 28$.

5. 0 INFORMED SEARCH:

Informed search is an estimate availability of the distance (cost) from each node(city) to the goal. This estimation will help you to head into the correct direction.

The outline of inform search are heuristic search, best first search, greedy search and A* search

5. 1 Heuristic search:

Heuristic uses domain specific knowledge to estimate the quality or potential of partial solution . A heuristic search that find a good solution in less time comparing to blind search but not always find best solution. This search is very much useful for solving big problem which may may not solved by using other search and this search generate a possible solution which can be either route from the initial location or goal in the problem. For travelling salesman problem the nearest neighbour heuristic work well, but some time due to the arrangement of cites it will not find the shortest route.

5. 2 Best first search:

Best first search is a general approach of the tree search algorithm. In which it expand a nodes which is based upon a evaluation function. The estimation of cost is constructed as evaluation function then the node with minimum evaluation is expanded first. A best search is a combination of both breadth first search and depth search, a breath first is good because it will not go deeper and the depth first search is best because it can be found without

searching all the nodes, hence the best first search allows us to gain benefit both the approaches

5. 3 Greedy search:

Greedy search select the path that has lowest heuristic value or estimated distance to the goal. Greedy search is a example of best first strategy, and in some cases like depth breast search I may never find the solution and greedy search is not optimal solution take into costly paths . This can be happen in the last step or in the first step.

5. 4 A* search:

The most widely known form of best first search is called a* search or best first search is simplified as A*search. The evaluation of nodes is carried in A* search was combination of $g(n)$ the cost reached and $h(n)$ the cost get from the node to the goal

$$f(n) = g(n) + h(n)$$

Here $f(n)$ estimate cost of cheapest solution through node n , A* search is both complete and optimal which is identical to uniform cost search.

From the above equation:

$g(n)$ is the total length between starting place to current location.

$h(n)$ is estimated length from current location to goal. A heuristic function which estimate how much distance it takes to reach goal.

$f(n)$ is the sum of $g(n)$ and $h(n)$. then this will be a current estimated shortest path which founds until a A* algorithm is completed.

A* Algorithm:

A* Algorithm starts with initial nodes then it will take the best node on open such that $f(n) = g(n) + h(n)$ is minimum. If Best is a goal node quit then return to the path from the initial to best or remove the best from open and all among best node naming each node with its path from initial node.

6. 0 A* Algorithm applying for given graph:

from the above graph, it shows that the node 1 is the initial node where salesman starts and reaches the node 3 which is the current location. then sum of the distance between initial to current location is consider as $g(n)$ ie $1-5-2-4-3 = 26$. then the estimated length from current location to the goal is consider as $h(n)$ ie 8 and then total minimal distance of path is

$$A^* = 1-5-2-4-3-1 = 34$$

Then this is the minimal path in which salesman can travel starting at node 1.

7. 0 Comparison table for depth first search and A* algorithm

Algorithm

Domain knowledge

Memory space

Minimum route

Time complexity

efficient

Depth first search

It doesn't contain domain knowledge

more

It find after complete search

It takes more time

less

A* search

It contain domain knowledge

less

It has information about it

It takes less time

more

8. 0 CONCLUSION:

The Idea of Travel sales man problem has much application in different fields . To

Find best routes of travelling salesman we are used two algorithms they are Depth

First search and A* Algorithm. Hence we conclude that A* algorithm is more efficient then depth first search algorithm, the time complexity of depth first is more comparing to A* search and it takes less memory space comparing to depth first search. Depth first search algorithm is more suitable finding minimum tour for limited number of cities, because if we take 50 to 100 cites depth first search expands each node of a tree to reach to the goal which is time consuming and memory waste.