AO* ALGORITHM FOR SOLVING TRAVELLING SALESMAN PROBLEM

[1]Shraddha Ramteke, [2]Mrs. Deepty Dubey
[1]M.Tech Research Scholar, Computer Science & Engg. Dept.
Chattrapati Shivaji Institute of Technology,Durg, India
[2]Associate Professor, Computer Science & Engg.Dept.
Chattrapati Shivaji Institute of Technology,Durg, India
Email: [1]shraddharamteke7@gmail.com,[2]deepty.shukla@gmail.com

Abstract—**According to travelling salesman problem (TSP), an NP Hard problem, given a number of cities and the distances between each couple of cities, the aim is to find the smallest possible route that goes to each city exactly once and returns to the origin city i.e. find a least cost Hamiltonian cycle. It is definitely an NP-hard problem, important in operations investigation and theoretical computer scientific discipline. In the theory involving computational complexity, the decision version in the TSP where, given any length L, the task is to decide whether the graph offers any tour shorter than L belongs to the class of NP-complete issues. Thus, it is possible which the worst-case running time for virtually every algorithm for the TSP increases super-polynomial or perhaps exponentially with the quantity of cities. Heuristic search is definitely an AI search technique that employs heuristic to its moves. Heuristic is a rule of thumb that probably leads into a solution. Heuristics play a major role in search strategies because of exponential nature of the most extremely problems. Heuristics help to reduce the quantity of alternatives from an exponential number into a polynomial number. In AI, heuristic search incorporates a general meaning, and an increasingly specialized technical meaning. Within a general sense, the term heuristic is utilized for any advice which is often effective, but just isn't guaranteed to work always. The major aim of this research work is to use the concept of AO* for solving the TSP to get the optimal Hamiltonian route.**

Keywords—AO* alogrithm, Travelling Salesman Problem (TSP).

I. INTRODUCTION

[1]Travelling Salesman Problem is a combinational problem consisting of some cities and some edges connecting one city to other. TSP can be represented by a graph G (V, E), where V is the set of vertices (cities) and E is the set of edges (path) between each of the two vertices specific to the graph. TSP is to discover the shortest path in the graph G setting up a least cost Hamiltonian Cycle. If there exists a path between two cities i and j, then the distance between these cities can be computed as-.

$$d_{ij} = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2}$$

TSP deals with real time scenario for a traveling salesman, where the most important for him to follow the shortest and optimal path having the minimum cost so as to gain maximum profit and to deliver the goods lesser time as much as possible. Path optimization is one of the major problems while travelling from the source to destination city visiting each city only once.

Path can be calculated using many strategies but to decide the best possible strategy according to the number of cities is a difficult task.

TSP Heuristic: [2]Whenever the salesman is at town i this individual chooses as their next city i. e., the city j which is why the c (i, j) charge, is the bare minimum among all c(i, k) charges, where k will be the pointers of the city the salesman has not visited yet. In case more than one citygives the particular minimum cost, the city with the smaller k is going to be chosen. This greedy algorithm selects the lowest priced visit in each step and won't care whether this will lead to a correct solution or not.

Basic Approach for solving TSP:
1. Initialize cities for tour.
2. Start from root.
3. Select next city in tour with minimum cost.
4. Repeat step 3 until an optimum cost Hamiltonian cycle is formed.

Input

Network formed from *n* cities
Cost c(i,j) of traveling from one city to next city, w*here* *i* & *j* = 1,2,3 . . , *n*.
Start with initial city.

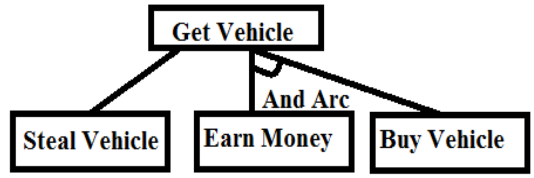
Output

A least cost Hamiltonian cycle.

II.PROPOSED METHODOLOGY

A. *Applying AO*for solving TSP*

When a problem can be split into a set of sub problems, where each sub problem can be solved separately and a combination of these will be a solution, AND-OR graphs or AND - OR trees are used for representing the solution. The decomposition of the problem or problem reduction generates AND arcs. One AND are may point to any number of successor nodes. All these must be solved so that the arc will rise to many arcs, indicating several possible solutions. Hence the graph is known as AND - OR instead of AND.



And Or Graph Example

An algorithm to find a solution in an AND - OR graph must handle AND area appropriately. A Star algorithm cannot work with AND - OR graphs correctly and efficiently.

*AO*Algorithm-*

Procedure Proposed AO* algorithm for TSP

Let G consists only to the node representing the initial state call this node INIT.
Calculate h' (INIT).
Until INIT is labeled SOLVED or h´(INIT) becomes greater than FUTILITY,
Repeat the following procedure.
Trace the marked arcs from INIT and select an unbounded node NODE.
Generate the successors of NODE .if there are no successors then assign FUTILITY as h' (NODE). This means that NODE is not solvable. If there are successors then for each one called SUCCESSOR, that is not also an ancestor of NODE do the following

(a) Add SUCCESSOR to graph G
(b) If successor is not a terminal node, mark it solved and assign zero to its h ' value.
(c) If successor is not a terminal node, compute its h' value.

Propagate the newly discovered information up the graph by doing the following. Let S be set of nodes that have been marked SOLVED. Initialize S to NODE. Until S is empty `repeat the following procedure;
(a) Select a node from S call if CURRENT and remove it from S.
(b) Compute h' of each of the arcs emerging from CURRENT, Assign minimum h' to CURRENT.
(c) Mark the minimum cost path as the best out of CURRENT.
(d) Mark CURRENT SOLVED if all of the nodes connected to it through the new marked are have been labeled SOLVED.
(e) If CURRENT has been marked SOLVED or its h ' has just changed, its new status

must propagate backwards up the graph, hence all the ancestors of CURRENT are added to S.

End Loop

End Loop

III. EXPECTED OUTCOME

In this paper, AO* search algorithm is modified and proposed for solving the travelling salesman problem. Also on applying AO* algorithm for TSP, we can obtain the optimal solution as the least cost Hamiltonian route. On various integer input range, the proposed AO* procedure can provide the solution for travelling salesman problem.

References

[1] http://en.wikipedia.org/wiki/Travelling_salesman_problem

[2] http://www.personal.kent.edu/~rmuhamma/

[3] Compgeometry/MyCG/CG-Applets/TSP/notspcli.htm