Performance Analysis of Shortest Path Routing Problem using Heuristic Algorithms

R. Somasundara Manikandan [1]
[1] Department of Computer Science,
Raja Doraisingam Govt. Arts College,
Sivaganga, Tamilnadu, India.

***Abstract:-* In this paper, we discuss an algorithm that dynamically plans a tourism travel planning by giving importance to two types of conditions they are a) hard constraint b) Soft constraint. Since Tourism is one of the essential thing in all of our life. Starting and ending hours of PTVs (Place to Visit), time duration and tour allocated, cost comes under the first condition, while the satisfaction factors of the PTVs and travelling distance in the tour are belongs to second condition. We use the factor that comes under the second condition to calculate the solution of the algorithm. Here we use Tabu search algorithm to find the optimal solution. Tabu search algorithm uses insert, delete, swap operator. Delete operator is used whenever the algorithm tends to enter in an endless cycle. In MATLAB programming software, the algorithm is tested with 30 instances of PTVs of the city of Tamilnadu. Various parameters of the algorithm are used to test its performance. The final output is obtained and compared in respect to the optimal solution.**

Keywords: Place to visit, optimization, planning, Swap, Insert, Delete, iteration

1. INTRODUCTION

Tourism in our life is considered as an essential. Tourism is a Part and parcel of our life. Tourists that visit one city or region during a tour within a particular time, find it tedious job to visit all places that exist in that particular area. Thus, they have to select some places that they consider as more interesting and worthy for them. There are so many interesting places to visit during their travel, for the available time, is usually a complex task. In such situations, it would be helpful for the tourist to have a system that which would enable him to correctly plan the touristic trip. Usually, the constraints are various locations of places and their starting and ending hours, individual ranking of each place to visit for the tourist, duration of the tour, etc. In order to produce a best route that satisfy all these conditions that tends to find an optimized route needs to be introduced.

The main aim of this paper is working with various shortest path algorithms that find optimal solutions between source and destination node, with in some seconds. The goal is to find a absolute permanent solution in less than some seconds. In this paper, we use tabu search heuristic method for finding the proper route between the source and destinations. In the Tabu search algorithm, it temporarily stores the preused information in the buffer and finally it takes the information from the buffer to find the optimal solution.

2. LITERATURE REVIEW

The Review of literature shows the steps of various algorithms that are handled for tour planning.

1. Orienteering Problem (OP) [1], where a number of *n* locations are given, each of them having a score *s*. The goal is to have a single tour trip that includes as many points as possible, so the satisfaction factor of the tour is maximized.
2. The Team Orienteering Problem (TOP) is an extended form of OP, which generalizes the problem for multiple tours [2].
3. Further, the Team Orienteering Problem with Time Windows (TOPTW) is an advanced version of TOP, where each location is associated with a time window [3]. to determine *m* routes, each limited by T_{max}, that maximizes the total collected score. In fact, the TOPTW is a simplified version of the Tourist Trip Design Problem (TTDP) [4].

There are many algorithms that are discussed in the literature. These algorithms could be also applied for the purpose of tour planning. A tabu search heuristic that effectively solves the TOP is presented in [5]. Roughly, this tabu finds the solution by iteratively storing the results in memory buffer and reuse it fort further iteration, optimized solution and future evaluation. It uses a number of input parameters, which are used to obtain optimal good solution of the algorithm.

Guided Local Search (GLS) method can solve the TOP problem uses consecutive iterations of the algorithm is discussed in [6]

Variable Neighborhood Search (VNS) method [8], is presented in [9]. searches for a better solution, by changing the procedure of neighborhood creation is discussed in [6] to produce solutions of high quality.

Iterated Local Search (ILS) [11] is presented, this algorithm is able of finding good solutions when applied for the data sets known in the literature.

3. HEURISTIC ALGORITHMS

There are different types of heuristic algorithms they are

- Ant Colony Optimization
- Simulated Annealing
- Genetic Algorithm
- Tabu Search Algorithm

Ant Colony Optimization

The name came after the behavior of ant in the food paths[13]. Ant Colony Optimization meta heuristic search (ACO) is a method for obtaining optimization results.[7] Ant colony is NOT to design an accurate model of nature [8] since it is based on stochastic process. Ant Colony uses Pheromone as a communication medium [14]. In short it makes optimization but not the best solution.

Simulated Annealing

Simulated annealing (SA) is an optimization technique that traverse the search space. It generates neighboring solution of the current solution. A superior and inferior neighbor is find using temperature parameters. Accepting worse solutions is a fundamental property of metaheuristics because it allows for a more extensive search for the optimal solution.

Simulated Annealing search is a probabilistic method used for obtaining the overall minimum of a function that may possess several local minima is focused in [9]. The Simulated Annealing procedure randomly generates a large number of possible solutions, keeping both good and bad solutions [10]. It only finds the good solution but not the best.

Genetic algorithm

The genetic algorithm is a heuristic algorithm that follows Darwin Theory. In real world Genetics that have the chromosome, mutation, population as the parameters. Similarly the heuristic Genetic algorithm that also has the parameters such as crossover, mutation, population, and chromosome.

A typical genetic algorithm requires:

1. A genetic representation and
2. a fitness function

It is responsible for creating its population and evolving it until a termination condition is reached. Genetic Algorithms (GAs) are search algorithms based on the mechanics of the natural selection process. Every GA follows the comment “Survival of the fittest" concept. That is the result is based on the optimization

Genetic Algorithm heuristic search [11] to find the near-optimal solutions. [12]. The Genetic Algorithm transforms a population of individual objects, each with an associated fitness value and survival of the fittest and naturally occurring genetic operation such as a cross over (recombination) and mutation. Each individual in the population represents a possible solution to a given problem. [13] Repeated fitness function evaluation for complex problems are often the most Finding the optimal solution to complex high dimensional problems often requires very expensive fitness function evaluations.

Tabu Search

Tabu search (TS) is similar to simulated annealing that finds the solution by testing the mutation of individual results. While simulated annealing generates only one mutated solution, tabu search generates many mutated solutions and moves to the solution with the lowest fitness generated. It contains memory buffer to handle datas and they reuse it for finding best solutions. Hence Tabu search algorithm is considered as best algorithm when compared with others.

3. MATHEMATICAL FORMULATION OF TABU-SEARCH

The algorithm yields the optimized solution for tours, where a number of conditions are considered for planning and optimization of the tour. The goal is to plan a several day trip that will serve the tourist to visit a number of touristic sites/PTVs (Place To Visits). This is the level of Orienteering Problem with Time Windows (OPTW).
A set of n locations is given, where each of them *(*i=1,…, n) is associated with a satisfaction value S, a fee f_i, a duration t_i an starting (O_i) and ending (C_i) hour. Normally a package that contains several nights and days. Each tour is restricted to upto the limit of time T_{max} and it starts at a source point and ends at destination point. In general, the opening / closing time and tour duration are variable for each day. The time t_{ij} needed to travel from source i to j, and vice versa, is known for all locations.
The maximum time is represented by $T_{max.}$ The maximum budget B_{max}, which does not exceeds the limit. The aim of the algorithm is to find a trip with m tours that includes as many available PTVs as possible, and also satisfy the time limit and cost factor. That is to obtain lower cost within short travel time. Both the budget and time duration comes under hard conditions. Satisfaction factor and travel time are taken as soft conditions which make the output as optimized result.
The soft conditions such as satisfaction factor and travel time are inversely proportional to each of them For example, if there are more places to visit in the tour, the satisfaction factor will be higher, while the travel time will be also higher. On the other hand, if we want to decrease the travel time there is need to have less number of nodes in the trip, which results in minimized satisfaction factor, Again there problem arises to increase the satisfaction factor. Hence the system of equation that finds the optimal value of both satisfaction factor and travel time constraints is needed.
Based upon the above condition the tabu search problem is mathematically expressed as

$$Max\{\sum_{i=1}^{n}(S_i * x_i)\} \qquad (1)$$

$$\sum_{i=1}^{n}(B_i * x_i) \leq Bmax \qquad (2)$$

where:
X_i = { 1, if point ***i*** is visited during the trip
0, if point ***i*** is not visited during the trip}
n – No of available PTVs for visit
S_i – Satisfaction factor of point ***i***
B_i – Entry fee of point ***i***

$$Min\left\{\left[\sum_{i=1}^{n}\left(\sum_{\substack{j=1\\ j\neq i}}^{n} t_{ij} * y_{ij}\right)\right] + \sum_{i=1}^{n}(t_{si}\, u_i + t_{ie} v_i)\right\} \qquad (3)$$

y_{ij} = { 1, if a visit to ***i*** is followed by a visit to j
0, if a visit to ***i*** is not followed by a visit to j}
U_i = { 1, if visit ***i*** is first visit in the tour
0, if visit ***i*** is not first visit in the tour}
V_i = { 1, if visit ***i*** is last visit in the tour
0, if visit ***i*** is not last visit in the tour}
t_{ij} – travel time from point ***i*** to ***j***
t_i – travel time from start point to point ***i***
t_e – travel time from point ***e*** to end point

$$\sum_{i=1}^{m} z_{ij} \leq 1 \quad (j=1, \ldots, n) \quad (4)$$

Where:
Z_{ij} = { 1, if visit ***i*** in trip is point **j**
0, if visit ***i*** in trip is not point **j** }
m – Number of POIs visited during the entire trip

$$o_i \leq v_i \;\&\; v_i + t_i \leq c_{i,} \;\; i = 1, \ldots, m \quad (5)$$

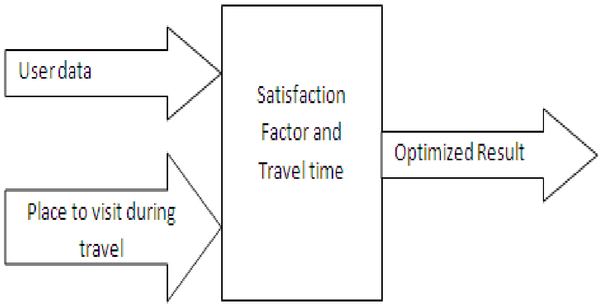
a) Equation (1) mention the maximal satisfaction factor
b) Equation (2) mention the cost is lower than or equal to the allocated budget.
c) Equation (3) mention the minimal travel time between source and destination
d) Equation (4) mention the particular point is visited one time
e) Equation (5) mention the trip feasible

4. DESCRIPTION OF THE ALGORITHM

Route planning is done based on the user's preference that describes the trip. It is described below

- Data that describe the tour, such as: start/end date of tour, cost estimation, lodging, and number of tours to be taken during the trip, coefficient of weight of satisfaction factor and travel time, tourist preferences for categories and types of places to visit
- Data that describe the PTVs, such as: name of the place, visiting time, location, fee, working hours, type and category of place to visit
- Data about travel distances between source and destination including places to visit that is available.

Tabu Search Routing algorithm


User data
Place to visit during travel
Satisfaction Factor and Travel time
Optimized Result

This tabu search algorithm consists of different phases. During the first phase It gather the information from user and their interest. During the Second phase they check whether it satisfies the satisfaction factor and the travel time and finally in the last phase it brings out the optimized result to the user.

The process of tour route planning creates a step by step itinerary process that consists of prior output during the trip period. The optimization is done by utilizing the tabu search heuristic. In our case, the tabu search heuristic uses two operators such as swap and insert for finding the search space. The Delete operator is used wherever to reduce the local optimum. The iteration is continued as far as the optimum is reached. Since tabu includes memory with previous results which can omit local optimum. The pseudo code of the tabu search heuristic is given below.

```
Step 1 Generate initial solution x.
Step 2 Initialize the Tabu List.
Step 3 While set of candidate solutions S” is not complete.
Step 3.1 Generate candidate solution x” from current solution x
Step 3.2 Add x” to S” only if x” is not tabu or if at least one Aspiration Criterion is satisfied.
Step 4 Select the best candidate solution x* in S” .
Step 5 If fitness(x*) > fitness(x) then x = x*.
Step 6 reenter the value of tabulist and aspiration criteria
Step 7 If terminates exit, otherwise go to Step 3.
```

Determining the neighbor

A neighbor would be correct if it fulfills the hard constraints:

- All visits in the trip are scheduled when respective PTVs are open,
- The trip budget (cost) is not exceeded the limit and
- The length of each tour in the trip

The pseudo code for determining candidate feasibility is given in the following:

Determine correctness of neighbor

```
begin
    correctness=false;

    if new visit is open in scheduled time do

        if neighbor cost is under budget do

            if neighbor is viable in
                time do
                correctness=
                true;

            end
        end

    end
end
return legality;
```

Determine time of neighbor(changed tour)

```
begin
    viability=false;

    if length of changed tour is not greater than
    original tour length do

        vilabilty=true;

    end
end

Return viabilty;
```

Evaluation function

Evaluation is done by considering two soft constraints, namely *the total trip satisfaction factor* and *total trip travel time*. The goal is to find an optimal best trip that satisfies the condition such as the total satisfaction factor as higher as possible, while the travel time remains as low as mush as possible. In order to realize this, we have used an evaluation/fitness function that consists of two components:

Evaluation function = w1*[satisfaction $factor_{norm}$] + w2 * [travel $time_{norm}$]

Parameters w_1 and w_2 represent the weight coefficients

$$Satisfaction\ factor_{norm} = 100 * \frac{total\ satisfaction\ factor}{Maximum\ satisfaction\ factor}$$

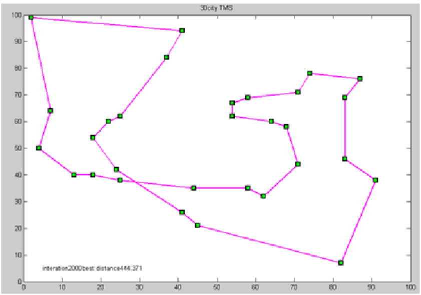
The goal is to find an optimal best trip that satisfies the condition such as the total satisfaction factor as higher as possible, while the travel time remains as low as possible. The travel time norm which will minimize the travel time, by using

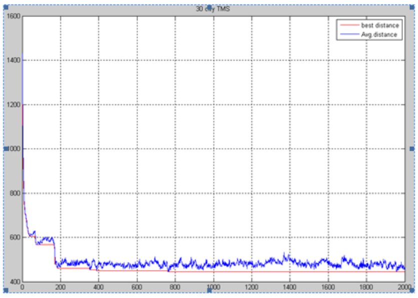
$$travel\ time_{norm} = 100 * \left(1 - \frac{travel\ time}{TDWB}\right)$$

5. EXPERIMENTAL RESULTS

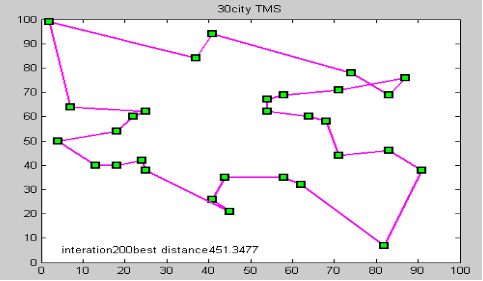
The algorithm is tested using 30 instances of PTVs of the city. Various algorithms are applied and their results are discussed.

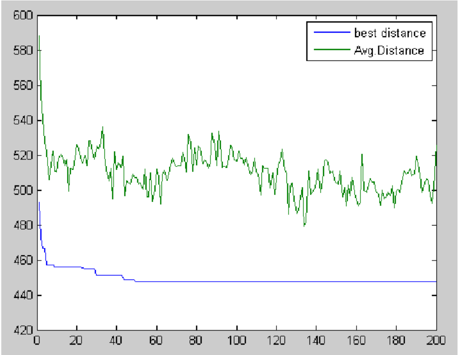
Graph shows the result of shortest path using Tabu tabu search heuristic method



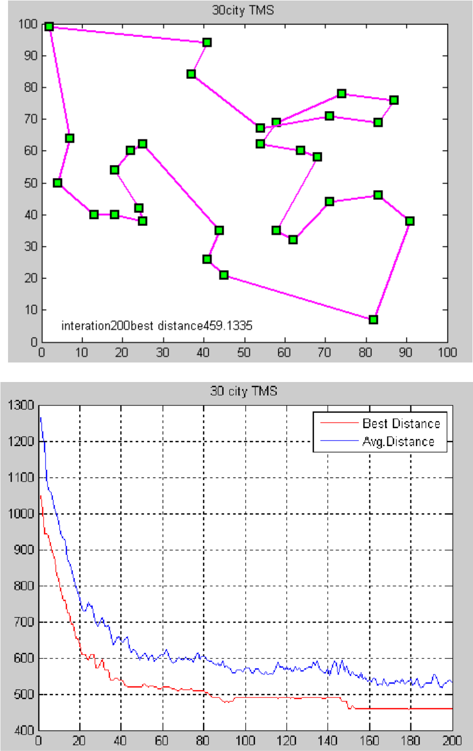


Graph shows the result of shortest path using Ant Colony Optimisation method

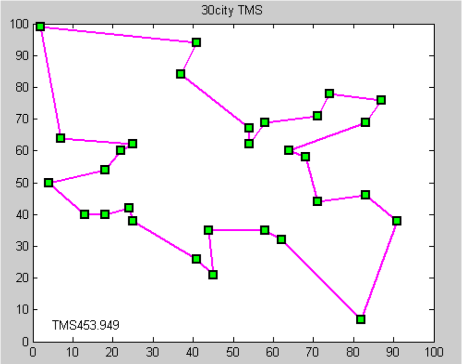


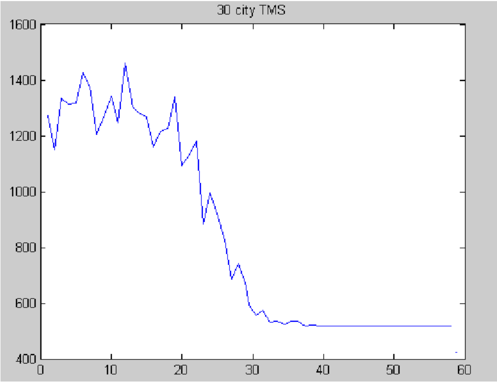


Graph shows the result of shortest path using Genetic Algorithm



Graph shows the result of shortest path using Simulated Annealing method



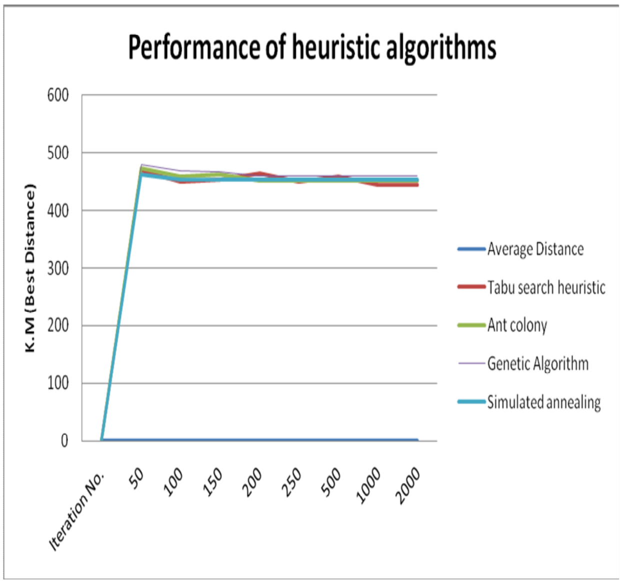


The table gives the comparison of various algorithms and give the results for 30 cities with various iterations

	Best Distance	Tabu search heuristic	Ant colony	Genetic Algorithm	Simulated annealing
Iteration No.	**In K.M**	**30 cities**	**30 cities**	**30 cities**	**30 cities**
50	500 K.M	469.2722	472.6521	478.3942	462.9527
100	500 K.M	449.5213	458.2596	467.2894	**453.949**
150	500 K.M	452.7241	462.3581	466.2714	**453.949**
200	500 K.M	464.1323	**451.3477**	**459.1335**	**453.949**
250	500 K.M	450.3285	**451.3477**	**459.1335**	**453.949**
500	500 K.M	458.8963	**451.3477**	**459.1335**	**453.949**
1000	500 K.M	445.3216	**451.3477**	**459.1335**	**453.949**
2000	500 K.M	**444.3710**	**451.3477**	**459.1335**	**453.949**

The table shows the iterations of various algorithms. The tabu search that gives the optimized result in higher iterations. But the other Algorithms that gives the solution in fewer iterations but they are not optimized one. Hence Tabu search is the best .

The below figure gives the idea of optimized result obtained from various algorithm and represented by a graph



6. DISCUSSIONS

As per our implementation there are several algorithms that are used to find the optimum solution. To satisfy our condition, the Tabu- search algorithm which optimize the solution among all the three algorithms. Even though in tabu search the iteration No is increased in order to refine the best solution. All the three algorithms that obtain the result in very close iteration but their result is not optimize when compare to tabu search. Hence tabu search meta heuristic approach gives the best optimized result.

7. CONCLUSIONS

This paper completely describes the tabu search method in a very performance manner while compare with the other heuristic algorithms. This comparative paper gives the optimized result for the users who use the tabu-search algorithm in a right manner. This paper satisfies the user constraints as well as satisfaction factor and the cost.

REFERENCES

[1] B.L. Golden, L. Levy, and R. Vohra, "The orienteering problem", Naval Research, Logistics, 1987. 34:307-318

[2] Chao, I.-M., B. L. Golden, E. A. Wasil, " the team orienteering problem", Eur. J. Oper. Res., 1996b, 88(3) 464–474.

[3] Kantor, M. G., M. B. Rosenwein, "The orienteering problem with time windows", . Oper. Res. Soc., 1992, 43(6) 629– 635.

[4] P. Vansteenwegen, D. Van Oudheusden, " The mobile tourist guide: an OR opportunity", OR Insights, 2007; 20(3):21-7.

[5] Tang, H., Miller-Hooks, "A Taboo search heuristic for the team orienteering problem", Comput. Oper. Res. 2005, 32, 1379 – 1407

[6] [6] Voudouris, C., Tsang, " Guided local search and its application to the travelling salesman Problem", Eur. J. Oper. Res., 1999, 113, 469-499

[7] G. di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications Network", Journal of Artificial Intelligence Research (JAIR), 1998, Vol. 9, Pag. 317-365,

[8] T.A. Feo and M.G.C. Resende, "Greedy randomized adaptive search procedures", Journal of Global Optimization, 1995,

[9] Anthony Ralston and Edwin D. Reill, "Encyclopedia of Computer Science", Chapman & Hall, 1993. Second Edition

[10] Adewole A.P., Otubamowo K.,Egunjobi , " Comparative Study of Simulated Annealing and Genetic Algorithm for Solving the Travelling Salesman Problem", International Journal of Applied Information Systems (IJAIS), 2012, ISSN : 2249-0868: Volume 4– No.4,.

[11] Yagvalkya Sharma, Subhash Chandra Saini, Manisha Bhandhari, "Comparison of Dijkstra's Shortest Path Algorithm with Genetic Algorithm for Static and Dynamic Routing Network", International Journal of Electronics and Computer Science Engineering, ISSN-2277-1956/V1N2-416-425.

[12] Hanif Khan, Nasiruddhin Khan, Syed Inayatullah And Shaikh Tajuddin Nizami,Solving TSP Problem by using Genetic Algorithm Fozia", International Journal of Basic & Applied Sciences IJBAS, Vol: 9 No: 10

[13] M.Dorigo, V.Maniezzo, A.Colorni, "Ant system: optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man, and Cybernetics, 1996,.Part B, Vol.26, Issue1, pp.29-41,

[14] M. Dorigo and G. Di Caro, "The Ant Colony Optimization meta-heuristic", In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization,. 1999, Mc Graw Hill London, UK, pages 11–32,109--133.