Eurasian Journal of Analytical Chemistry
ISSN: 1306-3057 OPEN ACCESS 2018 13 (SP): 292-299

A Survey on Approaches to Solve Travelling Salesman Problem

S. Prabakaran, T. Senthil Kumar, J. Ramana, K. Chandan Reddy

Received 05 November 2018 ▪ Revised: 23 November 2018 ▪ Accepted: 02 December 2018

Abstract: Travelling Salesman Problem (TSP) is widely used in traffic management and the transportation industry where the route optimization would speed up services and increase efficiency. In this paper, we survey the most popular approaches including dynamic and meta-heuristic algorithms to solve the Travelling Salesman Problem. Our survey clearly indicates that meta-heuristic algorithms like Ant Colony Optimisation (ACO) take additional time to arrive at the solution but returns the best optimal route when compared with others. Hence, we take our research forward in improving the ACO technique to support environmental changes like weather, traffic congestions and delays in real time and return the optimal route.

INTRODUCTION

The Traveling Salesman Problem (TSP) is one of the well-known problems for finding the optimal path. The problem is for finding the shortest closed route among n cities, with input of complete distance matrix among all the cities. A most common application of TSP is the movement of people, equipment and vehicles from one city to another and aiming to minimize the total traveling cost. The Travelling salesman problem was first proposed by mathematicians, Carl Menger and Hustler Wietni in 1930. The problem is that a travelling salesman wants to visit many cities with the goal to find the shortest path. In the 1950's and 1960's, the problem gained popularity in scientific circles of Europe and the U.S. Dantzig et al. ,made a contribution by expressing the problem as an integer linear program and then developed the cutting plane method for its solution. TSP is one of the NP-hard problem and as a reason the complexity order of these problems is exponential leading to an execution time which is not acceptable. In solving these kind of problems, obtaining certain solutions might require more time than lifetime of the system. Dynamic programming solutions are one of the best techniques to solve these problems in terms of execution time and convert time order of the problem into polynomial form. But dynamic programming suffer with the issues like memory consumption where in large problems, system cannot meet dynamic programming requirements. To overcome the drawbacks of dynamic programming and considering the size and complexity of optimization problems like travelling salesman and real world problems, a lot of researchers got attracted towards meta-heuristic algorithms and local investigation that inspire from social intelligence of creatures [2]. Meta-heuristic algorithms are the ones that try to obtain logical results in an acceptable time by consuming minimum memory. In this paper we compare and contrast the popular approaches to Travelling salesman problem and propose a new hybrid approach for efficient and better route optimization.

PROPOSED ALGORITHMS FOR SOLVING TRAVELLING SALESMAN PROBLEM

A* Algorithm

The accelerated version of the Dijkstra's algorithm, the A* algorithm [1] was proposed by Peter Hart, Nils Nilsson and Bertram Raphael in 1968 to find the best optimal route by using heuristics to guide its search. Nils first suggested a path-finding algorithm, called A1, a faster version of the then best known formal approach, Dijkstra's algorithm, for finding shortest paths in graphs. Then Bertram Raphael suggested significant improvements over this algorithm, naming the revised version A2. Finally Peter E.

S. Prabakaran, Professor, Dept of CSE, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India
T. Senthil Kumar, Assistant Professor, Dept of CSE, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India.
J. Ramana, PG Scholar, Dept of CSE, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India.
K. Chandan Reddy, PG Scholar, Dept of CSE, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India.

Hart introduced an argument that established A2, with only minor changes, to be the best possible algorithm for finding shortest paths. So, they named the new algorithm in Kleene star syntax to be the A* algorithm that starts with A and includes all possible versions. The algorithm had only two sets OPEN and CLOSED where OPEN set was the only one node initially i.e., the starting node and the CLOSED set contained nodes that were already been examined. A neighbour that is in OPEN is scheduled to be looked at and finally the path with lowest or minimum cost is chosen as the optimal path to reach the destination.

Particle Swarm Optimisation

Particle Swarm Optimization (PSO), inspired by social behavior of bird flocking or fish schooling, is a population based stochastic optimization technique proposed by Kennedy and Eberhart[2]. PSO is a population based search algorithm. In a D-dimensional search space, each particle was treated as a point and has two attribute values: position and velotown. The position of the ith particle is represented as xi=(x1, x2, x3,…,xD). The velotown of the ith particle is represented as vi=(v1,v2, v3, vD). PSO was initialized with a group of random particles and then search for optimum solution is done by updating velotown and position of each particle. In each iteration, each particle updates itself continuously by following two extreme values, the best position found by the particle by far (pbest) and the best position in the swarm at that time (gbest). After finding the above extremes, the scheme for updating the position and velotown of each particle is shown using the formula:

$$v^{k+1} = w_k v^k + c_1 r_1 (pbest^k - x^k) + c_2 r_2 (gbest^k - x^k)$$
$$x^{k+1} = x^k + v^{k+1}$$

where v is velotown of ith particle in kth iteration, x is position of ith particle in kth iteration and r1,r2 are random numbers between 0 and 1 and C1,C2 are learning factors which is equal to 2 & w_k is the inertia weight factor which is between 0.1 to 0.9.

```
Procedure Particle Swarm
Begin
        For each particle
                initialise particle
        For each particle
                calculate cost
                if cost is better than pbest
                Set current value as new pbest
        Choose particle with best cost as gbest
        Update particle position
  While maximum iterations or optimum result
End
```

Dijkstra's algorithm

Dijkstra's algorithm is used in finding the shortest paths between nodes in a graph. In 1995, this was enhanced to find the shortest path from a location to the destination. H.Shimizu, M.Kobayashi and Y.Yonezawa [3] presented a traffic control system by using mean travel time to find the optimal route. The optimal route guidance algorithm [1] was derived from the Dijkstra's algorithm, weighted by the mean travel time of each link. The mean travel time was calculated based on several parameters like traffic congestion lengths, traffic signal controls, moving directions of vehicles etc., In this guidance system, the optimal routes including the shortest mean travel time route from one's current position to the destination are outputted. The Optimal route guidance system was stimulated at twelve signalised intersections in Fukuyama town, Japan and compared with actual measurement values in traffic system and was found to be efficient.

```
Procedure (Enhanced Dijkstra)
Let v1 be the origin vertex,
  and initialise W and ShortDist[u] as
  W := {v1}
  ShortDist[v1] :=0
  FOR each u in V - {v1}
    ShortDist[u] := T[v1,u]
  until W includes all vertices in V
  WHILE W <> V
    MinDist := INFINITE
    FOR each v in V - W
```

```
    IF ShortDist[v] < MinDist
      MinDist = ShortDist[v]
      w := v
    END {if}
  END {for}
  W := W U {w}
  FOR each u in V - W
    ShortDist[u]:= Min(ShorDist[u],ShortDist[w]+ T[w,u])
END {while}
```

Ant Colony Optimisation (ACO)

Ant colony optimization algorithm [4] is a kind of meta-heuristic parallel optimization algorithm that uses some artificial ants, which are similar to the real ants in nature, to find a minimum cost path for a given problem. Artificial ants lay some pheromone on the paths which they pass and as time goes on, the pheromone dropped will evaporate. Artificial ants choose their path with respect to the probabilities that depend on pheromone trails which have been previously laid by the artificial ants. In this way, artificial ants can easily choose the paths with high concentration pheromone and lay more pheromone on the paths which they choose. So, the paths with high concentration pheromone will attract more ants to choose them. After a number of iterations , the best route covering all nodes is obtained for the Travelling Salesman problem.

Procedure:
Ant-Solver(maxcycle,nAnts,α, β,ρ);
repeat
for k in 1…nAnts do
A_k=Construct(t,α, β,(X,D,C));
end for
Update hermonesTrails(t,ρ,{A1,….,A_{nAnts}})
until conf(A_i)=0 for $\forall_i \in$ {1,…,nAnts}
or maxcycle reached
end procedure

Fuzzy Logic Model (FML)

Y. Murat, and N. Uludag proposed a fuzzy logic model (FLM) [5] in 2008 , to find the best route in urban transportation. They performed a survey on decision of routes for two directions from Kampus to clanr and vice versa. Rates of reasons of the decision and route choice for peak hours traffic conditions are collected from the public.There were also some questions about maximum and minimum travel time of different transportation modes serving on the routes. With the help of these, best routes were selected and noted as actual root. Then fuzzy logic and logistic regression models were structured to take four input parameters : travel time, traffic safety, congestion and environmental effects and finally output the route utility. In first stage , the input parameters from survey are collected and converted to fuzzy logic using membership functions and at second stage, the rule base was used for inference and in third stages the fuzzy output were converted to crisp numbers. Finally deffuzzification method in the fourth stage was applied. After processing, the probabilities of routes were determined by multinomial logit formula. The FLM results were compared to LRM data and it was found that average accuracy of FLM for all routes from Kampus to clanr was 91.75%.

Artificial bee colony

Artificial bee colony algorithm is based on the intelligent foraging behaviour of honey bee swarm. Li-Pei Wongi, Malcolm Yoke Hean Lowii and Chin Soon Chong proposed a model [6] in which the bees are allowed to explore and search for the paths. Before making a move to a town ,bee will randomly observe the dance performed by other bess. In this dance the set of moves which are performed by other bees, this set of moves are named as "preferred path" and denoted as θ, which contains a tour that has been explored by bee. A heuristic transition rule is used to help the bees to decide its next town. This rule consists of two factors: arc fitness and heuristic distance. The arc fitness is calculated for all possible paths to towns that can be visited by a bee from a particular town at a particular time. The preferred path which is a part of the arc is assigned a higher fitness value. By doing this, a bee tends to choose the next visiting town based on the preferred path. Hence, a bee tends to choose the next visiting town which is nearest to its current town with the help of heuristic distance influence. The above given procedure is continued till the destination town is arrived.

```
Initialize_Population( )
   while stop criteria are not fulfilled do
         while(bees!=built paths) do
         Observe_Dance( )
         Forage_ByTransRule( )
           Perform_Dance ( )
         end while
     end while
end procedure BCO
```

Genetic Algorithm

Genetic algorithm [7] is a population based algorithm that follows the idea of biological evolution and natural selection where the fittest individuals survive. A genetic algorithm can be divided into several sub-parts that are used in this algorithm: representation, fitness function evaluation, initialization, selection, recombination (crossover and mutation), termination. GA Operators: The following operators based on natural selection

- Reproduction — Selects good strings population and forms a mating pool.
- Crossover — New strings are created by exchanging information among strings of the mating pool.
- Mutation — does a local search around the current solution to create a better string and also used to manintain diversity in the population.

The initial population is created, where each individual is expressed via defined representation. Then the fitness function is evaluated for the initial population and the subset of the population (so-called parents) is selected that will be used in recombination operators to generate offspring. The crossover operator is applied to parents to create new offspring and the mutation operator is applied with a certain probability, then the fitness function is evaluated and the individuals with the worst fitness value are removed. This process is continued until the individual with best fitness value is found.

Procedure:

P=initialize($population_{size}$, $problem_{size}$)

evaluate(population)

Best=getSolution(population)

while(!stop())

parents=select(population,$population_{size}$)

children=null

for($parent_1$,$parent_2$∈ parents)

$child_1$,$child_2$=

crossover($parent_1$,$parent_2$,$parent_{crossover}$)

children=mutate($child_1$,$p_{mutation}$)

children=mutate($child_2$,$p_{mutation}$)

End

evaluate (children)

Best=getSolution(children)

population=replace(population,children)

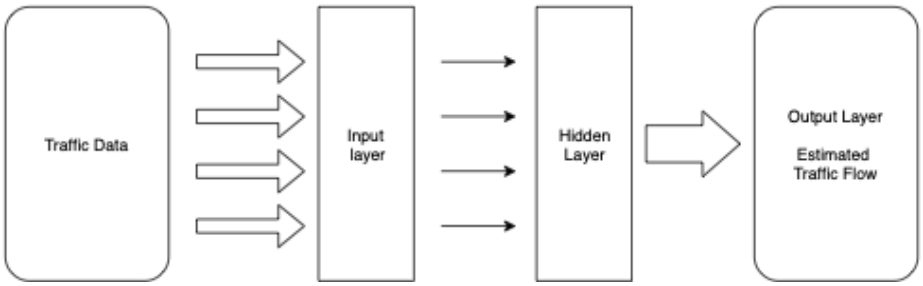
End

Return(Best)

Artificial neural networks (ANN)

Artificial neural networks are computing systems inspired by the biological neural networks that constitute animal brains. The neural network itself isn't an algorithm, but a framework for many different machine learning algorithms to work together and process complex data inputs. It is based on a collection

of connected units or nodes called artificial neurons, which model the neurons in a biological brain. Each connection, like the synapses in a biological brain, they can transmit a signal from one artificial neuron to another. An artificial neuron that receives a signal can process it and then signal additional artificial neurons connected to it. “Salehinejad and Talebi” applied ANN in with fuzzy logic and ant colony algorithm [8] in multi parameter route selection system for finding optimal route. The proposed system is executed locally for every single vehicle.



It finds directions with minimum costs based on the importance rates of user desired parameters. ANNs are used for traffic estimation of coming minutes. The employed ANN consist of one hidden layer with m hidden neurons, n inputs, and one output .

Statistical traffic data of last n years is the input and the ANN structure is trained for traffic prediction. This system can evade upcoming congestions and the system is aware of current vehicle location by using GPS. Therefore, if a congestion happens on the suggested direction, the system immediately recommends nearest direction to the user parameters and current direction based on the new conditions to evade upcoming congestion. The system also has the capability of considering the previously selected directions.

Hill Climbing

Hill climbing algotithm is a optimization technique which belongs to the family of local search, proposed by Y. Bykov and S. Petrovic [9]. It is an iterative algorithm that starts with an initial or arbitrary solution to a problem, then attempts to find a better solution by making an incremental change to the current solution. If the change produces a better solution, another incremental change is made to the current new solution, and so on until no further incremental changes shows any improvements. In this algorithm, data structure of the current node only store state and value of the objective function. Hill climbing looks at current neighbors. Hill climbing is called Greedy local search because it always selects a good neighbor without thinking about the destination. Hill climbing usually converges fast because it can improve bad state [1]. To solve TSP using hill climbing method it will give local optima which are different from best possible results, however this method can be used to obtain an acceptable local optima fast. Most important issue in hill climbing is to select the objective function which is considered for the travelling salesman through the passed route. It is obvious that shorter lengths are closer to the final result. To avoid getting stuck in local optima, one could use repeated local search or more complex schemes like iterated local search or reactive search optimization and tabu search or simulated annealing. In this algorithm, memory is consumed for the current solution and the neighboring solution so size of each solution will be equal to number of cities in the problem. Therefore it requires 2n memory where n is the number of cities.

```
HillClimb(path: points sequence)
Begin
        Compute initial path length
        Loop
          choose points pair U,V when
          swap U,V = shortest length
          if improvement=NULL
                return Path
          else swap U,V && Decrement length
        End Loop
End
```

Simulated Annealing

Simulated annealing is a probabilistic technique for finding the best solution or global optimum of a given problem of function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem. Simulated annealing is a algorithm which was inspired from the technique involving heating and controlled cooling of a material to increase the size of its crystals. The attributes of the material depends on its thermodynamic free energy. Heating and cooling the material affects both the temperature and the thermodynamic free energy. The simulation of annealing can be used to find a global minimum for a function with a large number of variables. Sh. Zhan, J. Lin, Z. Zhang, and Y. Zhong [10] proposed the idea of minimizing temperature to optimize TSP and to implement this algorithm, first, a solution is created randomly. Next a neighbor of the current solution is required on which the local search is used. Using the probability which is found from the above algorithm, neighbouring solution is considered as the current best solution, even if the objective function is not improved. Thus, chance of being trapped in local optima is decreased. After a certain time, probability of accepting a solution which does not improve the current best solution becomes lower. This process will be continued till stopping condition is reached. In this algorithm, a function called g(T) is used. This function is represented as follows:

$$g(T) = (T_0 - T_{F)} / \max_{iterations}$$

Similar to hill climbing this algorithm also consumes memory for the current solution and the neighbouring solution. So, the size of solution will be equal to twice the number of cities in the problem.

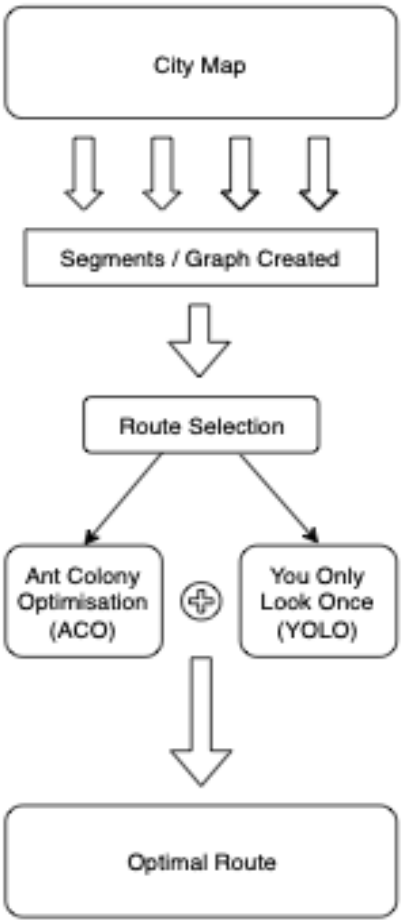
```
Procedure
Begin SA
  t=0, initialise T
  select current point Vc && evaluate Vc
   Repeat
     repeat
          select Vn from Vc neighbourhood
          if cost(Vc)<cost(Vn)then
                    Vc=Vn
          else if
           random(0,1)<e((cost(Vn)-cost(Vc))/T)
                              Vc=Vn
    until (termination)
          T = g(T)
  until(Halt)
End
```

PROPOSED MODEL

Ant colony optimization algorithm is a meta-heuristic parallel optimization algorithm that uses some artificial ants to find the best route. Pheromones are laid on the paths as the ants pass by and other ants can easily choose the paths with high concentration pheromone. So, Ant colony algorithm is applied on the map in order to find the optimal route.

In addition to make the algorithm adaptive to changing factors like traffic, weather etc., we also provide the real time traffic intensity data along with the pheromone concentration. So, ants can choose the optimal path based on these two parameters. We collect the real time traffic data using the You Only Look Once (YOLO) algorithm which is a state-of-art real time object detection system from images captured in real time from CCTVs installed in traffic signals and junctions.

This proposed model can be used by traffic department and transportation industry to find the best optimal route to reach a destination and efficiently control the traffic flow as the model predicts in real time.



CONCLUSION

In this paper we surveyed 10 approaches to solve the Travelling Salesman Problem. From all the above algorithms we learnt that Ant Colony Optimisation algorithm performed better due to its depth of understanding the paths and finding new ways to reach the best solution. However ACO suffered from adapting to changes happening to paths in which may affect the optimal solution. Hence, we propose a new hybrid model that can adapt to environmental, traffic congestions in real time and output the best optimal route. Our Proposed model uses Enhanced Ant colony meta-heuristic algorithm and traffic density determination using YOLO algorithm as an additional parameter to ants, along with phermones to make the system dynamic and adaptive in nature to real time data.

REFERENCES

[1] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, *4*(2), 100-107.

[2] Kennedy, J., & Eberhart, R. (1995). PSO optimization. In *Proc. IEEE Int. Conf. Neural Networks* (Vol. 4, pp. 1941-1948). IEEE Service Center, Piscataway, NJ.

[3] Kobayashi, M. A., Shimizu, H., & Yonezawa, Y. (1997). Dynamic route search algorithms of a traffic network. *Proceedings of the 36th SICE Annual Conference. International Session Papers*, 1211-1216.

[4] Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, *1*(1), 53-66.

[5] Murat, Y. S., & Uludag, N. (2008). Route choice modelling in urban transportation networks using fuzzy logic and logistic regression methods. *Journal of Scientific & Industrial Research*, 2008, 67, 19-27.

[6] Wong, L. P., Low, M. Y. H., & Chong, C. S. (2008). A bee colony optimization algorithm for traveling salesman problem. I*Second Asia International Conference on Modelling & Simulation (AMS)*, 818-823.

[7] Nallusamy, R., Duraiswamy, K., Dhanalaksmi, R., & Parthiban, P. (2010). Optimization of multiple vehicle routing problems using approximation algorithms. *International Journal of Engineering Science and Technology, 1* (3).

[8] Salehinejad, H., & Talebi, S. (2010). Dynamic fuzzy logic-ant colony system-based route selection system. *Applied Computational Intelligence and Soft Computing*, *2010*.

[9] Bykov, Y., & Petrovic, S. (2016). A step counting hill climbing algorithm applied to university examination timetabling. *Journal of Scheduling*, *19*(4), 479-492.

[10] Zhan, S. H., Lin, J., Zhang, Z. J., & Zhong, Y. W. (2016). List-based simulated annealing algorithm for traveling salesman problem. *Computational intelligence and neuroscience*, *2016*.