Solving Travelling Salesman Problem by Genetic and Exact Algorithms

A. BHUVNESH PRATAP SINGH (17BCB0057)
B. VAIBHAV HEMANT BHANDARI (17BCB0102)

1. ABSTRACT

Ant Colony Optimization algorithms (ACO) are meta-heuristic algorithms inspired from the cooperative behaviour of real ants that could be used to achieve complex computations and have been proven to be very efficient to many different discrete optimization problems. Simulated Annealing is also a meta-heuristic algorithm to approximate global optimization in a large search space. It is often used when the search space is discrete to very efficient results. Additionally, both the algorithms will apply the 2-opt method to optimize the solution.One such problem is the Travelling Salesman Problem (TSP) which belongs to the class of NP-hard problems, which means that there is no exact algorithm to solve it in polynomial time.
The importance of this problem appears in many application areas such as telecommunications, electronics, logistics, transportation, astronomy, industry and scheduling problem. Many algorithms had been proposed to solve TSP each with its own merits and demerits.

***Index Terms*—Travelling Salesman Problem, Ant Colony Algorithm, Simulated Annealing Algorithm, 2-Opt optimization**

2. INTRODUCTION

The traveling salesman problem (abbreviated TSP) presents the task of finding the most efficient route (or tour) through a set of given locations. Each location should be passed through only once, and the route should loop so that it ends at the starting location. This paper will focus on the symmetric TSP, in which the distance between any two locations is the same in both directions.

The TSP was introduced in 1832 in a German business manual written for the "traveling salesman." The manual explains that a traveling salesman should take an efficient tour through cities in Germany in an attempt to save time and visit as many cities as possible.The manual did not use any mathematical techniques to solve the problem, but suggested tours throughout 45 German cities that could potentially save the most time.
It is thought that mathematical study of the TSP was likely begun in 1930 by a mathematician named K. Menger. One of Menger's conclusions was that the technique of simply always choosing the next closest city to determine the best tour (Nearest Neighbor Algorithm) is not a consistently effective solution to the problem. The TSP gained notable mathematical interest from then on and research on how to improve solutions to the problem is still done today.
Theoretically, one could use what is referred to as "brute force search" and consider each possible tour through the given set of cities to determine which tour is the shortest. This is easy enough for small sets, but it doesn't take very many cities in a given set for that kind of computation to become extremely expensive.
Consider the small set of points {A,B,C,D,E}. The number of possible tours beginning and ending with point A and traversing every other point exactly once is equal to the number of permutations of the set $\{B,C,D,E\}$, which is 4!=24. Furthermore, as far as the symmetric TSP is concerned, the tour A, B, C, D, E, A is equivalent to the tour A, E, D, C, B, A since both take the same distance and one is easily deduced from the other.

A) Ant Colony Algorithm

Ant Algorithms are a recently developed, population-based approach which has been successfully applied to several np-hard combinatorial optimization problems. As the name suggests, ant algorithms have been inspired by the behaviour of real ant colonies, in particular, by their foraging behaviour. One of the main ideas of ant algorithms is the indirect communication of a colony of agents, called (artificial) ants, based on pheromone trails (pheromones are also used by real ants for

communication).

The (artificial) pheromone trails are a kind of distributed numeric information which is modified by the ants to reflect their experience while solving a particular problem. Recently, the ant colony optimization (ACO) metaheuristic has been proposed which provides a unifying framework for most applications of ant algorithms [15, 16] to combinatorial optimization problems. In particular, all the ant algorithms applied to the tsp fit perfectly into the ACO meta-heuristic and, therefore, we will call these algorithms also ACO algorithms.

The first ACO algorithm, called ant system (as) has been applied to the travelling salesman problem (tsp). Starting from ant system, several improvements of the basic algorithm have been proposed. Typically, these improved algorithms have been tested again on the tsp. All these improved versions of as have in common a stronger exploitation of the best solutions found to direct the ants' search process; they mainly differ in some aspects of the search control. Additionally, the best performing ACO algorithms for the tsp improve the solutions generated by the ants using local search algorithms.

In this paper, a modified ant colony system for solving tsp using 2-opt/3-opt and dynamic updating of heuristic parameter is developed. This algorithm is used to produce near-optimal solutions to the tsp.

B) Simulated Annealing Algorithm

Simulated Annealing is a recently developed probabilistic approach for finding the global optimum of a given function. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing may be preferable to alternatives.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. Both are attributes of the material that depend 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 an approximation of a global minimum for a function with a large number of variables to the statistical mechanics of equilibration (annealing) of the mathematically equivalent artificial multiatomic system.

This notion of slow cooling implemented in the simulated annealing algorithm is interpreted as a slow decrease in the probability of accepting worse solutions as the solution space is explored. Accepting worse solutions is a fundamental property of metaheuristics because it allows for a more extensive search for the global optimal solution. In general, the simulated annealing algorithms work as follows. At each time step, the algorithm randomly selects a solution close to the current one, measures its quality, and then decides to move to it or to stay with the current solution based on either one of two probabilities between which it chooses on the basis of the fact that the new solution is better or worse than the current one. During the search, the temperature is progressively decreased from an initial positive value to zero and affects the two probabilities: at each step, the probability of moving to a better new solution is either kept to 1 or is changed towards a positive value; instead, the probability of moving to a worse new solution is progressively changed towards zero.

3. Literature Survey

A) Ant Colony Algorithm

Various researches have been made using the ant colony algorithm and also many variations in the basic ant system is made specific to a application. In **t**his paper (*Oshin,et,al.(2016)*) they uses different parameters to provide analytical study of variants of Ant Colony Optimization for scheduling sequential jobs in grid systems. Based on the literature analysis, one can summarize that ACO is the most convincing technique for scheduling problems. However, incapacitation of ACO to fix up a systematized start-up and poor scattering capability cast down its efficiency.

To overpower these constraints researchers have proposed different hybridizations of ACO that manages to sustain more effective results than standalone ACO. And also In other paper (*Ivan Brezina Jr.Zuzana Čičková, et.al(2011)*) they uses this algorithm for Travelling salesman problem (TSP) consists of finding the shortest route in complete weighted graph *G* with *n* nodes and *n(n-1)* edges, so that the start node and the end node are identical and all other nodes in this tour are visited exactly once. They are shortest path of costumer servicing route planning bus lines. The higher number of ants in population causes the higher accumulation of pheromone on edges, and thus an individual keeps the path with higher concentration of pheromone with a high probability. It is useable for solving problems occurring in practical applications In this paper (*Amin Ahmadi,*

et.al(2012)) use of ant Colony Optimization (ACO) that is one of the meta-heuristic methods that constructs solutions of hard combinatorial optimization problem is for VRP. In the ant colony based algorithms to VRP, initial pheromone trails is calculated based on the best known route distances found for the particular problem. In this study it is calculated based on the feasible solution found. The visibility of an arc is calculated as a function of distance between two customers, customers' distance to the depot and the time window associated with the customer to whom the ant is considered to move.

B) Simulated Annealing Algorithm

Simulated annealing was a method introduced in 1983 by Scott Kirkpatrick, Daniel Gelatt and Mario Vecchi.

To understand the motivation for applying simulated annealing to the TSP, it is useful to first look at another method of optimization known as greedy search. Greedy search is a simple heuristic that breaks down large problems (like the TSP) into smaller yes-or-no choices that are always determined by whatever is immediately most beneficial.

Consider a landscape with a large number of hills and valleys and suppose that there is an objective to locate the highest point on this landscape in a timely manner.

One way to do this would be to arbitrarily pick a spot to start within the cluster of hills, and then only take steps in directions that result in a highest gain in altitude. When there are no more upward steps available, the process is completed and the current hill is declared the highest. This is known as a hill climbing; a type of greedy search. The tallest hill in this scenario is analogous to the global optimum solution tour of a particular TSP, and each location on the landscape corresponds to a specific tour. Taking a step is equivalent to slightly modifying the tour. Unfortunately, one of the major drawbacks to this technique is that it is very likely to have only led to a local maximum, and, returning to the landscape analogy, there is no indication that the height of the hill chosen is even close to the height of the tallest hill. One solution would be to just repeat this method many times, but doing so is costly in regards to time.

A better solution is to occasionally take steps to a lower altitude, thus allowing the search for a higher hill in areas that wouldn't have been accessible otherwise. This is one of the key concepts of simulated annealing. With simulated annealing, a random tour is chosen and its permutation is slightly modified by switching the order of as few as two points to obtain a new tour. If this new tour's distance is shorter than the original tour, it becomes the new tour of interest. Else, if this new tour has a greater distance than the original tour, there exists some probability that this new tour is adopted anyways. At each step along the way, this probability decreases to 0 until a final solution is settled upon.

Simulated annealing is named after a heat treatment process used on metals. During the annealing process, a metal is heated enough to allow its molecular structure to be altered. The temperature is then steadily lowered, which subsequently lowers the energy of the atoms and restricts their rearrangement until the metal's structure finally becomes set. This process minimizes the number of defects in the structure of the material.

Similarly, simulated annealing slowly and increasingly restricts the freedom of the solution search until it only approves moves toward superior tours. The probability of accepting an inferior tour is defined by an acceptance probability function. This function is dependent on both the change in tour distance, as well as a time-dependent decreasing parameter appropriately referred to as the Temperature. The way that the Temperature changes is referred to as the Cooling Schedule.

4. PROBLEM FORMULATION

Travelling Salesman Problem is a well-known, popular and extensively studied problem in the field of combinatorial optimization. Its statement is deceptively simple, but yet it remains one of the most challenging problems in operational research. It also an optimization problem of finding a shortest closed tour that visits all the given cities. It is known as a classical NP-complete problem, which has extremely large search spaces and is very difficult to solve.

The definition of a TSP is: given *N* cities, if a salesman starting from his home city is to visit each city exactly once and then return home, find the order of a tour such that the total distances (cost) travelled is minimum. Cost can be distance, time, money, energy, etc. Graph theory defines the problem as finding the Hamiltonian cycle with the least weight for a
given complete weighted graph.

A complete weighted graph $G=(N, E)$ can be used to represent a TSP, where N is the set of n cities and E is the set of edges (paths) fully connecting all cities. Each $edge\ (i,j) \in E$ is assigned a cost dij, which is the distance

between cities *i* and *j*.

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

5. PROPOSED METHODOLOGY

A) Ant Colony Algorithm

ACS differs in three main aspects from ant system. First, ACS uses a more aggressive action choice rule than AS. Second, the pheromone is added only to arcs belonging to the global-best solution. Third, each time an ant uses an arc (i; j) to move from city i to city j it removes some pheromone from the arc. In the following we present these modifications in more detail.

Tour construction.
In ACS ants choose the next city using the pseudorandom-proportional action choice rule: When located at city i, ant k moves, with probability q0, to city l with probability q0 the best possible move as indicated by the learned pheromone trails and the heuristic information is made (exploitation of learned knowledge). With probability (1 - q0) an ant performs a biased exploration of the arcs according to Equation

Global Pheromone trail update.
In ACS only, the global best ant is allowed to add pheromone after each iteration. Thus, The update according to Equation

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \rho\cdot\Delta\tau_{ij}^{gb}(t),$$

Local Search
Local Search starts from some initial assignment and repeatedly tries to improve the current assignment by local changes. If in the neighbourhood of the current tour T a better tour T_0 is found, it replaces the current tour and the local search is continued from T_0.
For the local search we use 2-opt, enhanced by the speed-up techniques

Algorithm:

Set parameters, initialize pheromone trails
Calculate the maximum entropy
Loop
Each ant is positioned on a random node
For k=1 to m do
Repeat
Select node j to be visited next (the next node must not be visited by the ant) until ant k has completed a tour
End for
A global updating rule is applied (global best ant will update)
Compute entropy value of current pheromone trails
Update the heuristic parameter
Until end_condition
Local search is applied to improve tour (opt-2)
End

B) Simulated Annealing Algorithm

The Basic Iteration
At each step, the simulated annealing heuristic considers some neighbouring states' of the current state s, and probabilistically decides between moving the system to state s' or staying in state s. These probabilities ultimately lead the system to move to states of lower energy. This step is repeated until the system reaches a state that is good enough for the application, or until a given computation budget has been exhausted.

The Neighbors
Optimization of a solution involves evaluating the neighbours of a state of the problem, which are new states produced through conservatively altering a given state. In the travelling salesman problem, each state is typically defined as a permutation of the cities to be visited, and its neighbours are the set of permutations produced by reversing the order of any two successive cities. The well-defined way in which the states are altered to produce neighbouring states is called a "move", and different moves give different sets of neighbouring states. These moves usually result in minimal alterations of the last state, in an attempt to progressively improve the solution through iteratively improving its parts.

The Probability Function

$P(\delta, T_k) = e^{-(\frac{\delta}{T_k})}$ for $\delta > 0$ and for $T_k > 0$

$P(\delta, T_k) = 1$ for $\delta > 0$ and for $T_k > 0$

Where

$\mathbf{T_k}$ = The temperature at the k^{th}instance of accepting the new solution state.
s = A particular tour through the given set of cities.
s' = State obtained by randomly switching the order of two cities.
c = Determines the total cost(in this case, distance) of a state.
$\delta =$ The relative change in cost c between s and s'.
$\beta =$ The rate at which the temperature is lowered each time a new solution is found(Cooling Constant)

Note that for $\delta>0$, for any given T, P is greater for smaller values of δ. In other words, a state s' that is only slightly more costly than s is more likely to be accepted than a state s' that is much more costly than s. Also, as T decreases, P also decreases.

Stopping Conditions
The stopping conditions are quite important in simulated annealing. If the algorithm is stopped too soon, the approximation won't be as close to the global optimum, and if it isn't stopped soon enough, wasted time and calculations are spent with little to no benefit. For the stopping conditions in this function, there is a maximum iteration which serves as a stopping point if the optimum result is not found.

Search
Searching in simulated annealing starts form a initial assignment got from a basic Greedy's algorithm. It then improves upon that by iterating through the whole global range and finding a better solution. This range keeps reducing with time.
We then improve on this using 2-opt.

Pseudocode

- Choose a random state s and define T and β.
- Create a new state s' by randomly swapping two cities in s.
- Compute $\boldsymbol{\delta = \frac{C(s')-C(s)}{C(s)}}$
    a) If $\boldsymbol{\delta <= 0}$, **then** $\mathbf{s = s'}$
    b) If $\boldsymbol{\delta > 0}$, then assign s= s' with probability $P(\delta, T_k)$
        i) Compute $T_{k+1} = T_k$
- Repeat steps 2 and 3 keeping track of the best solution until stopping conditions are met.

C) 2-opt Optimization Algorithm

The pairwise exchange or 2-opttechnique involves iteratively removing two edges and replacing these with two different edges that reconnect the fragments created by edge removal into a new and shorter tour.

Algorithm

```
2optSwap (route, i, k)
{
1. take route [0] to route[i-1] and add them in order to new_route
2. take route[i] to route[k] and add them in reverse order to new_route
3. take route[k+1] to end and add them in order to new_route
return new_route;
}


Repeat until no improvement is made
{
        start_again:
        best_distance=
        calculateTotalDistance(existing_route)
        for (i = 1; i < number of nodes eligible to be swapped - 1; i++)
        {
        for (k = i + 1; k < number of nodes eligible to be swapped; k++)
         {
        new_route = 2optSwap(existing_route, i, k)
        new_distance=
        calculateTotalDistance(new_route)
        If (new_distance < best_distance)
        {
        existing_route = new_route
        goto start_again
        }
        }
        }
}
```

6. RESULTS AND DISCUSSION:

Ant colony Optimization

TSP PROBLEM	CITIES	OPTIMAL TOUR	ANT COLONY (best)	ANT COLONY + 2-OPT (best)
WESTERN SAHARA WI29	29	27603	27809 (1.4274)	27603 (1.471)
DIJIBOUTI DJ38	38	6656	6673.4 (1.8992)	6664.11 (1.94)
QATAR QA194	194	9352	10581.08 (25.2345)	9927.07 (71.14)
URUGUAY UY734	734	79114	96291.43 (45.65)	

Simulated Annealing Algorithm

TSP PROBLEM	CITIES	OPTIMAL TOUR	SIMULATED ANNEALING (best)	SIMULATED ANNEALING + 2-OPT (best)
WESTERN SAHARA WI29	29	27603	27748.71 (0.24 sec)	27620 (0.91 sec)
DIJIBOUTI DJ38	38	6656	6670.19 (0.29 sec)	6662.85 (1.07 sec)
QATAR QA194	194	9352	10942.47 (0.98 sec)	9981.54 (68 sec)
URUGUAY UY734	734	79114	99214.77 (2.51 sec)	

7. CONCLUSION AND FUTUREWORK

From this we can see that the answers got by Ant Colony Algorithm and the Simulated Annealing Algorithm are very accurate for small number of cities and their accuracy decreases as the number of cities increases. Both the answers, however, are improved by the addition of the 2-opt optimization algorithm.

We can also conclude that the most efficient algorithm for small cities is the Simulated Annealing Algorithm, but as the number of cities increases, Ant Colony swiftly becomes the dominant algorithm.

In the future, we will aim to optimize the algorithm even more and try to add extra dimensions and extra features like City Preference and Road Blocking.

8. REFERENCES:

1. **ACO Algorithms for the Travelling Salesman Problem,** Thomas STUTZLEand Marco DORIGO
2. **Solving Travelling Salesman Problem by Using Improved Ant Colony Optimization Algorithm**, Zar Chi Su Su Hlaing and May Aye Khine, Member, IACSIT
3. **Simulated Annealing,** Per Brinch Hansen, SURFACE
4. **Optimization by Simulated Annealing,** S. Kirk Patrick, C. D. Gelatt Jr, M P Vecchi, SCIENCE
5. **Serial and Parallel Simulated Annealing And Tabu Search Algorithmsfor the Travelling Salesman Problem,** Miroslav Malek, Mohan Guruswamy, Mihir Pandya.
6. **Genetic Algorithms for the Travelling Salesman Problem,** John Grefenstette, Rajeev Gopal, Brian Rosmaita, Dirk Van Gucht