Application of an Improved Ant Colony Optimization on Generalized Traveling Salesman Problem

Krishna H. Hingrajiya[a]*, Anirudhdha M. Nayak[b]

[a]*Assistant professor, CE, Gandhinagar Institute of Technology, Gandhinagar, Gujarat 382721, India*
[b]*Assistant professor, IT, Gandhinagar Institute of Technology, Gandhinagar, Gujarat 382721, India*

Abstract

This paper presents ant colony system (ACS), a distributed algorithm that is applied to the generalized traveling salesman problem (GTSP). It gives a unique optimized implementation technique to lower the processing expenses engaged with routing of ants with inside the traditional ACO, and additionally the overall performance of the ACO via way of means of utilizing person range technique. Results display the velocity and meeting of the ACO may be advanced significantly, and the effects are pleasant in some event of GTSP. To attempt now no longer to stable in close by minima, an alternate interplay and a community searching through method are likewise added into this technique. Mathematical effects display that the proposed method can control the GTSP troubles truly well, and the created alternate interplay and close by hunt technique are powerful.

Keywords: Ant Colony Optimization, Generalized Traveling Salesman Problem (GTSP).

1. Introduction

The generalized traveling salesman problem (GTSP) is an augmentation of the notable traveling salesman problem. GTSP is a very important combinatorial optimization problem and is known to be NP hard. In the GTSP, the set of nodes is divided into clusters; the objective is to find a minimum-cost tour passing through one node from each cluster. Many applications of the GTSP exist in many fields. But researches still did not pay enough attention to GTSP specific local search and mostly use simple TSP heuristics with basic adaptations for GTSP [1]. Dorige presented the Ant Colony Optimization (ACO) in 1991[2], some strategies such as positive feedback and hidden parallel were proposed. By using positive feedback strategy, the ACO have the better result through parallel pheromone exchanging between ants. And by using hidden parallel strategy, jumping into the optimal solution can be prevented and the ACO is also very ancient. The researches and applications on ACO algorithm have made great progresses in the past years. Many scholars presented some efficient methods to solve these problems [3]. However, it still has some basic problems that have only been partially solved, such as searching time is too long and it may easily jump into local optimal solution. A big variety of publications had been dedicated to observe the TSP problem, together with some of its versions. The generalized TSP (GTSP) is a completely easy but sensible extension of TSP. In the GTSP problem, the set of nodes is the union of m clusters, which may also or may not be intersected. Every viable answer of GTSP, known as a g-excursion, is a closed path that consists of at the least one node from each cluster, and the objective is to discover an excursion with the minimum price. In a special case of GTSP, called E-GTSP, each cluster is visited exactly once [4]. The improvements of the ant colony algorithm are presented, provided the algorithm. Some simulation results are also provided.

The Ant Colony System

The Ant Colony System is evolved in step with the statement that actual ants are able to locating the shortest direction from a meals supply to the nest without the use of visible cues. To illustrate how the ‘‘actual” ant colony searches for the shortest direction; an instance from might be delivered for higher comprehension. In Fig. 1(a), assume A is the meals supply and E is the nest. The aim of the ants is to convey the meals again to their nest. Obviously, the shorter paths have gain as compared with the longer ones. Suppose that at time t = zero there are 30 ants at factor B (and 30 at factor D). And at this second there may be no pheromone path on any segments. So the ants randomly pick their direction with equal probability. Therefore, at the common 15 ants from every node will pass towards H and 15 towards C (Fig. 1(b)). At t = 1 the 30 new ants that come to B from A discover a path of depth, 15 at the direction that ends in H, laid through the 15 ants that went that manner from B, and a path of depth 30 at the direction to C, received because the sum of the path laid through the 15 ants that went that manner from B and through the 15 ants that reached B coming from D through C (Fig. 1(c)). The opportunity of selecting a course is consequently biased, in order

* Krishan H. Hingrajiya
E-mail address: krishna.hingrajiya@git.org.in

that the predicted quantity of ants going closer to C could be double of these going closer to H: 20 as opposed to 10, respectively. The identical is proper for the brand new 30 ants in D which come from E. This manner keeps till all the ants will in the end select the shortest course [5].

Given an n-town TSP with distances Dij, the synthetic ants are disbursed to those n towns randomly. Each ant will select the subsequent to go to consistent with the pheromone path remained at the paths simply as cited within side the above example. However, there are foremost variations among synthetic ants and actual ants: (1) the synthetic ants have ‘‘memory”; they could don't forget the towns they have got visited and consequently they could now no longer pick the ones towns again. (2) The synthetic ants aren't completely ‘‘blind”; they recognize the distances among towns and like to select the close by towns from their positions [6]. Therefore, the opportunity that town j is chosen via way of means of ant okay to be visited after town i may be written as follows:

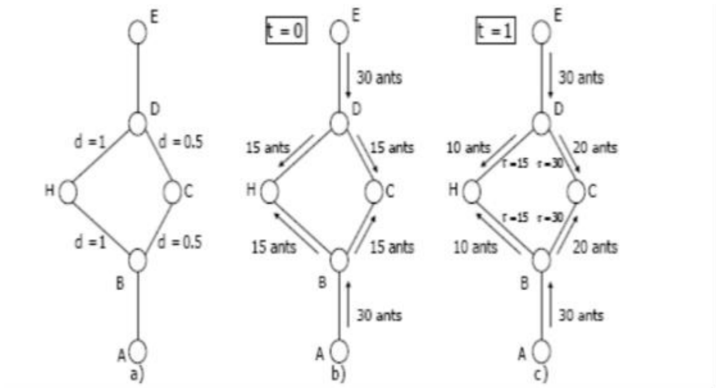

Fig. 1

Fig. 1. An instance with artificial ants. (a) The preliminary graph with distances. (b) At time t = zero there may be no path at the graph edges; therefore, ants pick whether or not to show proper or left with same chance. (c) At time t = 1 path is more potent on shorter edges, which might be therefore, within side the average, desired via way of means of ants.

$$p_{ij}^{k} = \begin{cases} \frac{[\tau_{ij}]^{\alpha}\cdot[\eta_{ij}]^{\beta}}{\sum_{s\in allowed_k}[\tau_{is}]^{\alpha}\cdot[\eta_{is}]^{\beta}} & j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In which Sij is the depth of pheromone path among towns i and j, a the parameter to modify the effect of Sij, Gij the visibility of metropolis j from metropolis i, that is constantly set as 1/Dij (Dij is the space among metropolis i and j), b the parameter to modify the effect of Gij and allowed k the set of towns which have now no longer been visited yet, respectively. At the beginning, l ants are located to the n towns randomly [7]. Then every ant comes to a decision the subsequent metropolis to be visited consistent with the chance Pij okay given via way of means of Eq. (1). After n iterations of this process, each ant completes excursion. Obviously, the ants with shorter excursions need to depart greater pheromone than people with longer excursions. Therefore, the path degrees are up to date as on a excursion every ant leaves pheromone amount given via way of means of Q/Lk, in which Q is a consistent and Lk the period of its excursion, respectively. On the alternative hand, the pheromone will evaporate because the time is going via way of means of. Then the updating rule of Sij can be written as follows:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \quad (2)$$

$$\Delta\tau_{ij} = \sum_{k=1}^{l} \Delta\tau_{ij}^{k} \quad (3)$$

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant } k \text{ travels on edge } (i,j) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Where t is the new release counter, q e [0, 1] the parameter to alter the discount of Sij, DSij the entire growth of path stage on edge (i, j) and DSij okay the growth of path stage on edge (i, j) due to ant okay, respectively. After the pheromone path updating process, the subsequent new release t + 1 will start.

ACO for the GTSP

The GTSP is a variation of the well-known TSP in which the set of nodes is divided into clusters; the objective is to find a minimum-cost tour passing through one node from each cluster. In GTSP, we are given n cities into m groups and we are required to find a minimum length tour that includes exactly one city from each group. Generally, we can mostly use simple TSP heuristics with basic adaptations for GTSP, but these conversions lead to the increasing of dimensions of the instance [8]. We use ACO to solve the GTSP without converting the instance. First, we introduce the GTSP with the mathematic model [13]: Let G= (V, E, W) be a completely weighted graph, in which V= {v1, v2... vn} (n≥3), E= {eij | vi, vj€ V}, and W {wij ≥0 and wij =0, ¥i, j N (n)} ij € N(n)} are vertex set, edge set and cost set, respectively. The vertex set V is partitioned into m possibly intersecting groups V1, V2…, Vm with |V j|≥1 and V=Um ,Vj. The unique Hamiltonian cycle is needed to by skip via all the groups, however now no longer all the vertices. At present there are forms of GTSP [15]: (1) the cycle passes precisely one vertex in every organization and (2) the cycle passes at least one vertex in every organization. In this paper, we add the parameter allowed k in the algorithm. The parameter allowed k denotes the city in the group has not been visited by ant t. By using allowed k, the algorithm avoids visiting the cities in the same group.

Allowedk = {x | x €V and c €G, ¥G€ tabuk } (5)

2. Ant colony optimization method for GTSP

Extended ACO technique for GTSP

In GTSP problem, the n towns are divided into m groups.
Each organization must be visited through precisely one town.
Initialize
For t=1 to new release quantity do
For k=1 to l do
Repeat until ant k has finished a excursion
Select the city j to be visited subsequent With possibility pij given by Eq. (1)
Calculate Lk
Update the trail levels according to Eqs. (2-4).
End

Algorithm:
1. Initialize:

 Set time:=0 {time is time counter}

 For each edge (i, j) set a preliminary sij = c for trail density

 and Dsij = 0.

2. Set s:=0 {s is tour step counter}

 For k:=1 to l do

 Place ant k on a town randomly. Place the town in visite dk.

 Place the organization of the town in tabuk.

3. Repeat until s 6 m

 Set s:=s + 1

 For k:=1 to l do

 Choose the following town to be visited in keeping with the probability pij k given by Eq. (1).

 Move the ant k to the selected city.

 Insert the selected city in visited k.

 Insert the group of selected city in tabuk.

4. For k:=1 to l do

Move the ant k from visitedk(n) to visitedk(1).

Compute the excursion period Lk traveled by ant k.

Update the shortest tour found.

For every edge (i, j) do

For k:=1 to l do

Update the pheromone trail density sij according to Eqs. (2)–(4).

time:=time + 1

5. If (time<TIME_MAX) then

6. Go to Step 2 and repeat till step 5.

Else print the shortest tour.

Stop

3. 2-OPT local search

nearby seek To accelerate the convergence, a nearby looking method called ‘‘2-OPT Local seek” is utilized in the proposed method.

Algorithm of the mutation process.

1. Randomly pick out t, let 0<t<m
2. Randomly select s, let 0<s<|XC(visited(t))|
3. For i:=1 to m-1 do
4. Insert the node N(visited(t), s) after the ith
5. node, compute duration i L
6. Find the shortest shortest L
7. If (shortest L< L origin)
8. Update the tour with the shortest one End

The function of the manner can be taken into consideration as to delete the crossover of visiting lines. The 2-OPT neighborhood seek basically eliminates edges from the excursion, and reconnects the 2 paths created. There is best one manner to reconnect the 2 paths in order that we nevertheless have a legitimate excursion (Fig. 4). We do that best if the brand new excursion may be shorter. It manner that a crossover factor is deleted within side the unique excursion. The pseudo-code of the 2-OPT manner is proven and the schematic diagram is proven respectively [9-10].

4. Numerical simulation

To confirm the validity of our proposed methods, those times may be received from TSPLIB library [13] and have been at the beginning generated for trying out preferred TSP algorithms. To take a look at GTSP algorithms, Fischetti et al. [12] furnished a partition set of rules to transform the times utilized in TSP to the ones which can be utilized in GTSP. Because the partition set of rules can generate the equal effects at distinctive experiments furnished that the records order are the equal, the partition set of rules may be used to generate take a look at records for distinctive algorithms. In the experiments, fundamental prolonged ACO method, ACO taken into consideration institution impact, ACO taken into consideration institution impact plus mutation procedure, ACO taken into consideration institution impact plus 2-OPT and ACO taken into consideration institution impact plus each mutation procedure and 2-OPT have been all executed 5 times [11].

Algorithm of the 2-OPT process.

1. Initialize

2. Select For i:=1 to i<m-3 do

3. Take For j:=i+2 to j<m do

4. If (di,i+1+dj,j+1>dij+di+1,j+1)

5. For k:=0 to k<(j-i)/2 do
6. Exchange (xj-k , xi+k+1).

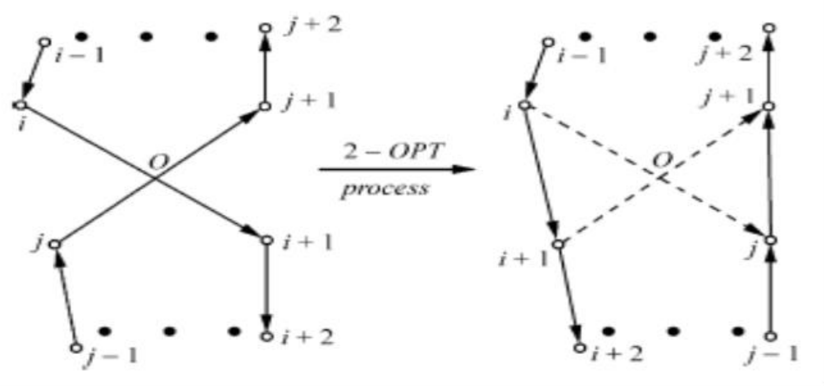

Fig. 2

5. Conclusion

Focused on the generalized travelling salesman problem, this paper extends the ant colony optimization technique from TSP to this field. To avoid locking into close by minima, a mutation technique is also delivered into this technique. Moreover, a close-by searching technique, namely, 2-OPT are seeking for is applied The paper affords a unique optimized enforcing technique that's designed to lessen the processing prices worried with routing of ants' with inside the ACO; second, the man or woman variant is delivered to the ACO, which allows the ants have one-of-a-kind course strategies. In this model, we alter the routing method of the ant who labored better; this is to beautify the effect of pheromones with inside the course of this ant. As a result, the first-rate of answers of ACO set of rules has additionally been enhanced.

References

1. Lien, Y., Ma, E., Wah, B.W.S.,1993. Transformation of the generalized traveling salesman problem into the standard traveling salesman problem, Information Science 74, p. 177–189.
2. Dorigo M, Gambardella LM.,1996. A study of some properties of ant-Q. In: Voigt H-M, Ebeling W, Rechenberg I, Schwefel H-S, eds. Proceedings of the PPSN 44th International Conference on Parallel Problem Solving from Nature. Berlin: Springer-Verlag, 656-665.
3. Gambardella LM, Dorigo M., 2000. An ant colony system hybridized with a new local search for the sequential ordering problem. INFORMS Journal on Computing, 12 (3), p. 237-255.
4. Parpinelli RS, Lopes HS, Freitas AA.,2002. Data Mining with an Ant Colony Optimization Algorithm. IEEE Trans. On Evolutionary Computation, 6(4), p.321-328.
5. Wu QH, Zhang JH, Xu XH.,1999. An ant colony algorithm with mutation features. Journal of Computer Research & Development, 36 (10), p.1240-1245.
6. Chen L, Shen J, Qin L., 2003. An adaptive ant colony algorithm based on equilibrium of distribution. Journal of Software, 14 (8), p. 1379-1387 (in Chinese).
7. Zhang Y., Pei Z.L., Yang J., Liang Y., 2007. An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Traveling Salesman Problem. BIBE, p. 693-698.
8. Colorni A, Dorigo M.,1996. Heuristics from nature for hard combinatorial optimization problems. International Trans Operational Research, 3 (1), p. 1-21.
9. Dorigo M, Maniezzo V, Colorni A.,1996. Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern, 26, p. 29–41.
10. Dorigo M, Caro GDi.,1999. The ant colony optimization meta-heuristic. In: New ideas in optimization. London: McGraw-Hill, p.11–32.
11. Reinelt G., 1991. TSPLIB – a traveling salesman problem library. ORSA J Compute , 3,p.376–84.

12. Fischetti M, Salazar JJ, Toth P.,1997. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. Oper Res. ,45, p. 378–94.

13. Snyder LV, Daskin MS.,2006. A random-key genetic algorithm for the generalized traveling salesman problem. Eur J Oper Res. , 174, 38–53.