Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.6, No.2, 2015

An Approach for Solving Multiple Travelling Salesman Problem using Ant Colony Optimization

Krishna H. Hingrajiya
University of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M. P.), India
Email: krishnahingrajiya@gmail.com

Ravindra Kumar Gupta   Gajendra Singh Chandel
University of Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M. P.)

Abstract
Ant Colony Optimization (ACO) is a heuristic algorithm which has been proven a successful technique and applied to a number of combinatorial optimization (CO) problems. The traveling salesman problem (TSP) is one of the most important combinatorial problems. Multiple traveling salesman problem (MTSP) is a typical computationally complex combinatorialOptimization problem, which is an extension of the famous traveling salesman problem (TSP). The paper proposed an approach to show how the ant colony optimization (ACO) can be applied to the MTSP with ability constraint. There are several reasons for the choice of the TSP as the problem to explain the working of ACO algorithms: it is an important NP-hard optimization problem that arises in several applications; it is a problem to which ACO algorithms are easily applied; it is easily understandable, so that the algorithm behavior is not obscured by too many technicalities; and it is a standard test bed for new algorithmic ideas as a good performance on the TSP is often taken as a proof of their usefulness.
Keywords— Ant colony optimization, Traveling salesman problem

I. INTRODUCTION

In recent years, many research works have been devoted to ant colony optimization (ACO) techniques in different areas. It is a relatively novel meta-heuristic technique and has been successfully used in many applications especially problems in combinatorial optimization. ACO algorithm models the behaviour of real ant colonies in establishing the shortest path between food sources and nests. Ants can communicate with one another through chemicals called pheromones in their immediate environment. The ants release pheromone on the ground while walking from their nest to food and then go back to the nest. The ants move according to the amount of pheromones, the richer the pheromone trail on a path is, the more likely it would be followed by other ants. So a shorter path has a higher amount of pheromone in probability, ants will tend to choose a shorter path. Through this mechanism, ants will eventually find the shortest path. Artificial ants imitate the behaviour of real ants, but can solve much more complicated problem than real ants can.

Consider Fig. 1A: Ants arrive at a decision point in which they have to decide whether to turn left or right. Since they have no clue about which is the best choice, they choose randomly. It can be expected that, on average, half of the ants decide to turn left and the other half to turn right. This happens both to ants moving from left to right (those whose name begins with an L) and to those moving from right to left (name begins with a R). Figs. 1B and 1C show what happens in the immediately following instants, supposing all ants walk at approximately the same speed. The number of dashed lines is roughly proportional to the amount of pheromone that the ants have deposited on the ground. Since the lower path is shorter than the upper one, more ants will visit it on average, and therefore pheromone accumulates faster. After a short transitory period the difference in the amount of pheromone on the two path is sufficiently large so as to influence the decision of new ants coming into the system (this is shown by Fig. 1D). From now on, new ants will prefer in probability to choose the lower path, since at the decision point they perceive a greater amount of pheromone on the lower path. This in turn increases, with a positive feedback effect, the number of ants choosing the lower, and shorter, path. Very soon all ants will be using the shorter path.

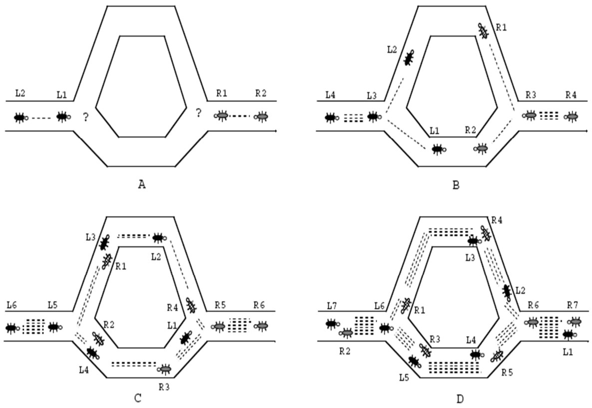

Fig.1 how real ant finds the shortest path

A) Ants arrive at a decision point. B) Some ants choose the upper path and some the lower path. The choice is random. C) Since ants move at approximately constant speed, the ants which choose the lower, shorter, path reach the opposite decision point faster than those which choose the upper, longer, path. D) Pheromone accumulates at a higher rate on the shorter path. The number of dashed lines is approximately proportional to the amount of pheromone deposited by ants.

The mutiple traveling problem (MTSP) is an extension of TSP. This problem relates to accommodating real world problems where there is a need to account for more than one saleman.The MTSP can be generalized to a wide variety of routing and scheduling problems, for example, the School Bus Routing Problem [6] and the Pickup and Delivery Problem [7].Therefore, finding a good optimal solution method for the MTSP is important and induces to improve the solution of any other complex routing problems. However, MTSP is a NP-complete problem for which optimal solutions can only be found for small size problems. It is known that classical optimization procedures are not adequate for this problem. Good heuristic techniques are necessary for solving MTSP due to its high computational complexity.

II. Travelling Salesman problem

Traveling salesman problem (TSP) is one of the well-known and extensively studied problems indiscrete or combinational optimization and asks for the shortest roundtrip of minimal total cost visiting each given city (node) exactly once. TSP is an NP-hard problem and it is so easy to describe and so difficult to solve. Graph theory defines the problem as finding the Hamiltonian cycle with the least weight for a given complete weighted graph. It is widespread in engineering applications and some industrial problems such as machine scheduling, cellular manufacturing and frequency assignment problems can be formulated as a TSP.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. *dij*can be defined in the Euclidean space and is given as follows:

$$d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2} \quad (1)$$

One direct solving method is to select the route which has minimum total cost for all possible permutations of N cities. The number of permutations can be very large for even 40cities. Every tour is represented in 2n different ways (for symmetrical TSP). Since there are n! possible ways to permute n numbers, the size of the search space is then |S|=n!/(2n)=(n-1)!/2.

III. ACO Background

3.1 The ANT System

Ant System was first introduced and applied to TSP by marcodorigo et al. Initially, each ant is placed on some randomly chosen city. An ant k currently at city I choose to move to city j by applying the following probabilistic transition rule:

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l\in J_k(i)}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} & if\ j\in J_k(i)\\ 0 & otherwise\end{cases} \quad (2)$$

where nij is the heuristic visibility of edge (i, j), generally it is a value of 1/dij, where dij is the distance between city i and city j. Jk (i) is a set of cities which remain to be visited when the ant is at city i. α and β are are two adjustable positive parameters that control the relative weights of the pheromone trail and of the heuristic visibility. If α=0, the closed vertex I more likely to be selected. This is responding to a classical stochastic greedy algorithm. If on the contrary β=0, only pheromone amplification is at work: This method will lead the system to a stagnation situation, i.e. a situation in which all the ants generate a sub-optimal tour. So the trade-off between edge length and pheromone intensity appears to benecessary.After each ant completes its tour, the pheromone amount on each path will be adjusted according to equation *(1-p)* is the pheromone decay parameter (*0<p<1*) where it represents the trail evaporation when the ant chooses a city and decides to move. *m* is the number of ants, *Lk* is the length of the tour performed by ant *k* and Q is an arbitrary constant.

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}(t) \quad (3)$$

In this equation,

$$\Delta\tau_{ij}(t)=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \quad (4)$$

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\dfrac{Q}{L_k}, & \text{if (i, j)} \in \text{tour done by ant k}\\ 0 & \text{otherwise}\end{cases} \quad (5)$$

3.2 The ACS Algorithm

The ACS is mainly different from the AS in these aspects: The decision rules of the ants are different; the global updating rules are different; and local updating rules which adjust the amount of the pheromone on various paths are newly added.

Step 1: Initiation. The amount of the pheromone on each side is initiated into a tiny constant value; allocate m ants randomly to n cities.

Step 2: In ACS, the so-called pseudorandom proportional rule is used: the probability for an ant to move from city i to city j depends on a random variable q uniformly distributed over [0, 1], and a predefined parameter q0.

$$j=\begin{cases}\arg\max_{u\in allowed_k(i)}\{[\tau_{iu}]^{\alpha}.[\eta_{iu}]^{\beta}\} & \text{if } q<q_0\\ J & \text{otherwise}\end{cases} \quad (6)$$

J is a random variable determined in accordance with equation (2). This strategy obviously increases the variety of any searching, thus avoiding any premature falling into the local optimal solution and getting bogged down.

Step 3: The local pheromone update is performed by all the ants after each construction step. Each ant applies it only to the chosen city,

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\rho.\tau_0 \quad (7)$$

Where 0< ρ<= 1 is a decay parameter, τ0=1/n.Lnn is the initial values of the pheromone trails, where n is the number of cities in the TSP and Lnn is the cost produced by the nearest neighbor heuristic. Equation (2) is mainly to avoid very strong pheromone paths to be chosen by other ants and to increase the explorative probability for other paths. Once the edge between city i and city j has been visited by all ants, the local updating rule makes pheromone level diminish on the edge. So, the effect of the local updating rule is to make an already edge less desirable for a following ant.

Step 4: Computing of the optimal path. After m ants have traveled through all the cities, compute the length of the optimal.

Step 5: Global updating of pheromone. After all the ants have traveled through all the cities, update only the amount of the pheromone on the optimal path with equation (8):

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \rho.\Delta\tau_{ij}(t) \quad (8)$$

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{1}{L_{gb}}, \text{ if (i, j)} \in \textit{global best tour} \\ 0 \quad \text{otherwise} \end{cases} \quad (9)$$

Where ρ is constant and Lgb is the length of global best tour.

Step 6: If the designated search number is not attained, then repeat the above steps.

IV Proposed Approach

4.1 The MTSP with ability constraint

The MTSP can be stated as follows: There are *m* salesmen who must visit a set of *n* cities, and each salesman is defined to start and end at the same depot. In this problem, each city must be visited exactly once by only one salesman and its objective is to find the minimum of total distances traveled by all the salesmen. An example is depicted in Fig.1, where *m*=3,*n*=7.Several authors[8,9] suggested transforming the MTSP with *m* salesmen and *n* cities into a TSP with *n*+*m*-1 cities by the introduction of m-1 artificial
depots (*n*+1,…,*n*+*m*-1).The transformation of the previous example is depicted in Fig.2

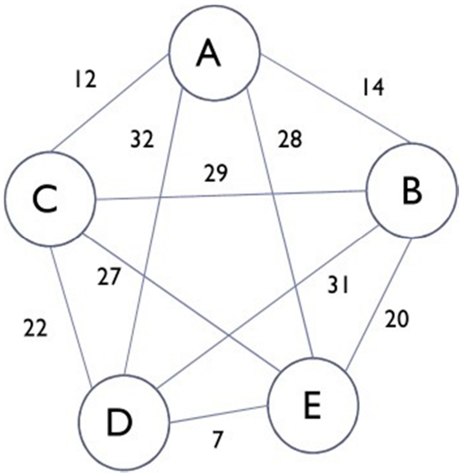

Fig 2. Example solution of MTSP

The general objective of the MTSP is to minimize the total distance which can be called minimum criterion; generally, there are *m*-1 cities always to select the nearest cities as their round trip. As a result, TSP which is made up of the left *n*-*m*+1cities is left. During the *m* salesmen, there are *m*-1 salesmen traveling only one city, and one salesman needs to travel the left *n*+*m*-1 cities. This is not up to the mustard. In practice, every salesman has the similar ability and the limit in ability.

4.2 Solution construction

Some authors used ACO to solute TSP, however, MTSP is different from TSP. In the paper, we improve on ACO according to the characteristic of the MTSP. One salesman should firstly travel an amount of cities, and the next salesman travels an amount of unvisited cities, in this way, all the salesmen travel the total cities. The traveled number of salesman is generated randomly in a certain range. Define the number of cities salesman *i* travel is *i tn, then*
(1,2,...,)

$$\{ 2 \leq tni \leq li$$

$$\sum_{i}^{m} tni = n-1 \quad \text{(Where i= 1, 2....m)} \qquad (10)$$

Where *m* is the number of salesmen, n is the number of cities, *i l* is the max number of cities salesman *i* can travel.

In the ACO algorithm, we assign a tour list to every ant. Every ant tours from the start city, and then select unvisited cities which is not start city or artificial cities. We suppose that the salesman *i* is visiting when ants start from the start city or artificial cities the *i*th time. While the salesman *i* is visiting, if the number of cities the ants have visited is equal to *tni*, then the ants select a unvisited artificial city, it means that the next salesman starts to visit. The cities which have been visited are recorded in the tour list ,until all the cities are visited by all the salesmen. In this way, a valid solution is constructed. In the algorithm, we used *k* ants, so there are *k* solutions constructed.

Route selection

Every ant selects the next city independently. Suppose that the probability of ant *k* moves from city *I* to city *j* is *Pk(t)*
ij ,uses the following probabilistic formula:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in J_k(i)} [\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} & if\ j \in J_k(i) \\ 0 & otherwise \end{cases} \qquad (11)$$

Where *A* denotes the cities that ant *k* has not visited, and *tabu (t)*denotes that cities that ant *k* has visited. Furthermore, the exponents α and β are positive parameters whose values determine the relation between pheromone information and heuristic information.

Conclusion

This paper presents an approach for solving multiple traveling salesman problem based on ant colony algorithm. ACO is a promising optimization technique for solving complex combinatorial optimization problems like the MTSP. The main contribution of this paper is a study of the avoidance of stagnation behavior and premature convergence by using distribution strategy of initial ants and dynamic heuristic parameter updating based on entropy. Then emergence of local search solution is provided. The experimental results and performance comparison showed that the proposed system reaches the better search performance over ACO algorithms do.

References

[1] X-song, B. LI, H. YANG, Improved Ant Colony Algorithm and Its Applications in TSP, Proceedings of Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), IEEE, 2006.
[2] Y. Zhang, Z-l.Pei, J-h.Yang, Y-c. Liang, An Improved Ant Colony Optimization Algorithm Based on Route Optimization and Its Applications in Traveling Salesman Problem, IEEE 2007. 1-4244-1509-8.
[3] M. Dorigo, M. Birattari, T. Stuzle, Ant Colony Optimization, Artificial Ants as a Computational Intelligence Technique, IEEE Computational Intelligence Magazine, November 2006.
[4] L. Li, S. Ju, Y. Zhang, Improved Ant Colony Optimization for the Traveling Salesman Problem, International Conference on Intelligent Computation Technology and Automation, IEEE, 2008.
[5] C. M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system:
Optimization by a colony of cooperating agents," *IEEE Transactionson System, Man, and Cybernetics*, Part B, vol.26, pp. 29-41, 1996.
[6] Angel RD,Caudle WL,Noonan R,Whinson A,Computer assisted school bus scheduling,*Management Science*,1972, 18.279-88.
[7] Christofides N,Eilon S,An algorithm for the vehicle dispatching problem,*Operations Research Quarterly*,1969, 20.309-318.
[8] Sveska JA,Huckfeldt VE,Computational experience with an *m*-salesman traveling salesman algorithm.*Management Science*,1973,19.790-799.
[9] Bellmore M,Hong S,Transformation of multisalesmen problem to the standard traveling salesman problem,*Jurnal of the Association Computer Machinery*,1974,21.500-504.