International Journal of Computer Science and System Analysis
Vol. 5, No. 1, January-June 2011, pp. 23-29

ISSN 0973-7448

ANT COLONY OPTIMIZATION FOR SOLVING TRAVELING SALESMAN PROBLEM

[1]Dweepna Garg and [2]Saurabh Shah

Dept. of Computer Engineering, Babaria Institute of Technology, Varnama, Vadodara-India
E-mails: [1]dweeps1989@yahoo.co.in, [2]saurabh_ce2001@yahoo.com

Abstract: *An ant colony capable of solving the traveling salesman problem (TSP). TSP is NP-hard problem. Even though the problem itself is simple, when the number of city is large, the search space will become extremely large and it becomes very difficult to find the optimal solution in a short time. 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. Computer simulations demonstrate that the artificial ant colony is capable of generating good solutions to both symmetric and asymmetric instances of the TSP. The method is an example, like simulated annealing, neural networks, and evolutionary computation, of the successful use of a natural metaphor to design an optimization algorithm.*

Keywords: *Ant colony optimization; Ant System; Genetic Algorithm; traveling salesman problem; Quadratic assignment problem; Vehicle Routine Problem; Scheduling*

I. BACKGROUND

- Discrete optimization problem are difficult to solve.
- “Soft Computing Technique “ developed:

(A) Genetic Algorithm (GA)

GAs is search procedures based on the mechanics of natural selection and natural genetics.

(B) Ant Colony Optimization (ACO)

ACO is a meta heuristic inspired by the foraging behavior of ant colonies 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 behavior of real ant colonies, in particular, by their foraging behavior.

II. INTRODUCTION

Ant Colony Optimization (ACO) is a paradigm for designing meta heuristic algorithms for combinatorial optimization problems. A Meta heuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a meta heuristic is a general-purpose algorithmic framework that can be applied to different optimization problems with relatively few modifications. Swarm intelligence is the discipline that deals with natural and artificial systems composed of many individuals that coordinate using decentralized control and self-organization. In particular, the discipline focuses on the collective behaviors that result from the local interactions of the individuals with each other and with their environment.

III. NATURAL BEHAVIOUR OF ANTS

Real ants are capable of finding the shortest path from a food source to their nest. While walking ants deposit pheromone on the ground and follow pheromone previously deposited by other ants, the essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions.

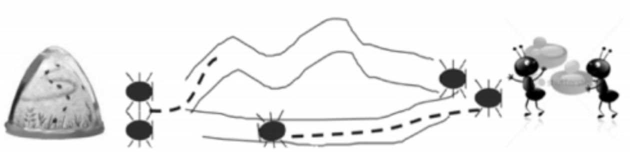


Figure 1

To find a shortest path, a moving ants lay some pheromone on the ground, so an ant encountering a previously trail can detect it and decide with high probability to follow it. As a result, the collective behavior that emerges is a form of a positive feedback loop where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path.

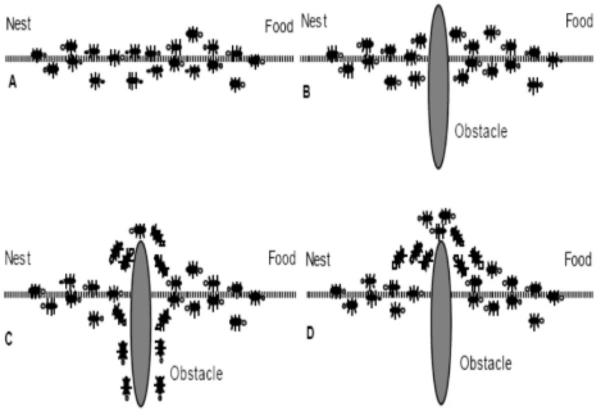


Figure 2

(A) Real ants follow a path between nest and food source.

(B) An obstacle appears on the path: Ants choose whether to turn left or right with equal probability.

(C) Pheromone is deposited more quickly on the shorter path.

(D) All ants have chosen the shorter path.

IV. APPLICATIONS FOR ACO

- TSP (Traveling Sales Person)
- QAP (Quadratic assignment problem)
- Scheduling
- VRP (Vehicle Routine Problem)
- Telecommunication Network
- Graph Coloring
- Water Distribution Network

TSP - Given a set of *n* cities, the Traveling Salesman Problem requires a salesman to find the shortest route between the given cities and return to the starting city, while keeping in mind that each city can be visited only once.

VRP- A direct extension of the TSP, the first problem AS was applied to, are Vehicle routing problems (VRPs). These are problems where a set of vehicles stationed at a depot has to serve a set of customers before returning to the depot, and the objective is to minimize the number of vehicles used and the total distance traveled by the vehicles.

QAP -The quadratic assignment problem (QAP) is the problem of assigning *n* facilities to *n* locations so that the assignment cost is minimized, where the cost is defined by a quadratic function. The QAP is considered one of the hardest CO problems, and can be solved to optimality only for small instances.

V. FLOWCHART FOR ACO

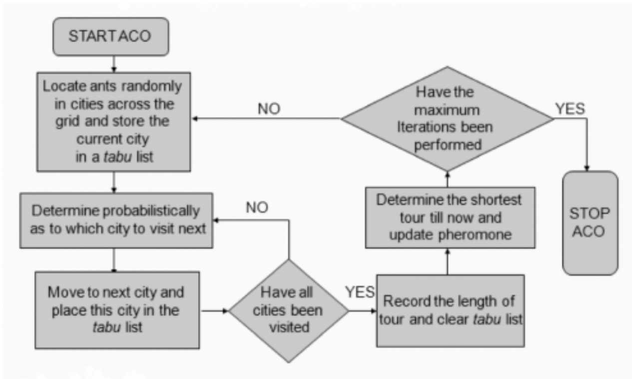


Figure 3

VI. TRAVELING SALESMAN PROBLEM

(A) History of TSP

Mathematical problems related to the traveling salesman problem were treated in the 1800s by the Irish mathematician Sir William Rowan Hamilton and by the British mathematician Thomas Penyngton Kirkman. The general form of the TSP appears to be have been first studied by mathematicians starting in the 1930s by Karl Menger in Vienna and Harvard. The problem was later promoted by Hassler Whitney and Merrill Flood at Princeton.

(B) Traveling Salesman Problem

The ACO algorithm, called Ant System (AS), has been applied to the Traveling Salesman Problem (TSP). Starting from Ant System, several improvements of the basic algorithm have been proposed. TSP is also known as a NP-hard problem. An artificial ant colony capable of solving the traveling salesman problem (TSP).

Ants of the artificial colony are able to generate successively shorter feasible tours by using information accumulated in the form of a pheromone trail deposited on the edges of the TSP graph. Computer simulations demonstrate that the artificial ant colony is capable of generating good solutions to both symmetric and asymmetric instances of the TSP. The method is an example, like simulated annealing, neural networks, and evolutionary computation, of the successful use of a natural metaphor to design an optimization algorithm.

TSP is to find the shortest trip of a salesman for a finite number of cities. A salesman is asked to start from a random city by visiting each city exactly once and then to return to the starting city 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. TSP is also called the Hamiltonian circuit, which is a closed tour visiting each city in G exactly once. Each edge (i, j) belongs to 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:

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

where (x_i, y_i) and $(x\,j, y\,j)$ are the coordinates of city *i* and of city *j*, respectively. The goal in TSP is to find a minimal length in the Hamiltonian circuit of the graph.

For symmetric TSPs, the distances between the cities are independent of the direction of traversing the arcs, that is, *dij* = *dji* for every pair of nodes. In the more general asymmetric TSP (ATSP) at least for one pair of nodes *i; j* we have $\bar{d}_{ij} \neq d_{ji}$.

In case of symmetric TSPs, we will use Euclidean TSP instances in which the cities are points in the Euclidean space and the inter-city distances are calculated using the Euclidean norm.

(C) With Metric Distances

In the metric*TSP*, also known as delta-TSPor Ä-TSP, the intercity distances satisfy the triangle inequality. This can be understood as “no shortcuts”, in the sense that the direct connection from A to B is never longer than the detour via C:

$$c_{ij} \leq c_{ik} + c_{kj}$$

These edge lengths define a metric on the set of vertices. When the cities are viewed as points in the plane, many natural distance functions are metrics.

1. In the Euclidian TSP the distance between two cities is the Euclidean distance between the corresponding points.
2. In the Rectilinear TSP the distance between two cities is the sum of the differences of their *x*- and *y*-coordinates. This metric is often called the Manhattan distance or city-block metric.
3. In the maximum metric, the distance between two points is the maximum of the differences of their *x*- and *y*-coordinates.

(D) Non-metric Distances

Distance measures that do not satisfy the triangle inequality arise in many routing problems. For example, one mode of transportation, such as travel by airplane, may be faster, even though it covers a longer distance.

In its definition, the TSP does not allow cities to be visited twice, but many applications do not need this constraint. In such cases, a symmetric, non-metric instance can be reduced to a metric one. This replaces the original graph with a complete graph in which the inter-city distance c_{ij} is replaced by the shortest path between *i* and*j* in the original graph.

For an asymmetric TSP, the length of an edge connecting two cities i and j depends on whether one goes from i to j or from j to i, and in general di j not equal to dji. Due to the simplicity of the TSP, it is easy to program an exhaustive search for the TSP. However, the complexities of the search space will rapidly grow with the city size. For n cities, it has (n-1)! solutions. For example, the 7-city problem has 360 solutions easily solved by using an exhaustive search method. When the heuristic approaches are applied to the TSP, they can be classified as tour constructive heuristic and tour improvement heuristic (also called local optimization heuristics). Construction heuristic is an algorithm that determines a tour according to some construction rules, but does not try to improve upon this tour. Tour construction heuristic randomly chooses a city as the starting point and then utilizes some heuristic rules to add new cities on the tour so as to build a feasible TSP solution.

In general, all the ACO algorithms for the TSP follow a specific algorithmic scheme. After the initialization of the pheromone trails and some parameters a main loop is repeated until a termination condition, which may be a certain number of solution constructions or a given CPU-time limit, is met. In the main loop, first, the ants construct feasible tours, then the generated tours are improved by applying local search, and finally the pheromone trails are updated. In fact, most of the best performing ACO algorithms for NP-hard combinatorial optimization problems follow this algorithmic scheme. It must be noted that the ACO meta-heuristic is more general than the algorithmic scheme given above. For example, the later does not capture the application of ACO algorithms to network routing problems.

```
procedure ACO algorithm for TSPs
Set parameters, initialize pheromone trails
while (termination condition not met) do
ConstructSolutions
ApplyLocalSearch % optional
UpdateTrails
end
end ACO algorithm for TSPs
```

Figure 4: Algorithmic Skeleton for ACO Algorithm Applied to the TSP

When Ant System (AS) was first introduced, it was applied to the TSP. Initially, three different versions of AS were proposed; these were called ant-density, ant-quantity, and ant-cycle. While in ant-density and ant quantity the ants updated the pheromone directly after a move from a city to an adjacent one, in ant-cycle the pheromone update was only done after all the ants had constructed the tours and the amount of pheromone deposited by each ant was set to be a function of the tour quality. Because ant-cycle performed much better than the other two variants, here we only present the ant-cycle algorithm, referring to it as Ant System in the following. In AS each of *m* (artificial) ants builds a solution (tour) of the TSP, as described before. In AS no local search is applied.

E. *Tour construction.* Initially, each ant is put on some randomly chosen city. At each construction step, ant *k* applies a probabilistic action choice rule. In particular, the probability with which ant *k*, currently at city *i*, chooses to go to city *j* at the *t*th iteration of the algorithm is:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}} \quad if \quad j \in \mathcal{N}_i^k$$

where $\eta_{ij} = 1/d_{ij}$ is an a priori available heuristic value α and β are two parameters which determine the relative influence of the pheromone trail and the heuristic information, and $\mathcal{N}_i^k$ is the feasible neighborhood of ant *k*, that is, the set of cities which ant *k* has not yet visited. The role of the parameters and is the following. If = 0, the closest cities are more likely to be selected. If *=0, on*ly pheromone amplification is at work. Search stagnation is defined in as the situation where all the ants follow the same path and construct the same solution.

F. *Pheromone update.* After all ants have constructed their tours, the pheromone trails are updated. This is done by first lowering the pheromone strength on all arcs by a constant factor and then allowing each ant to add pheromone on the arcs it has visited:

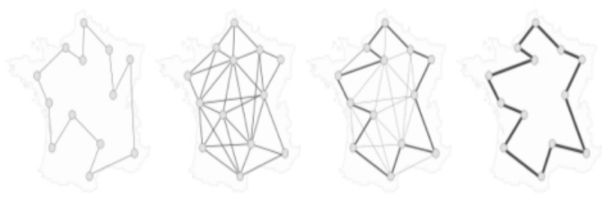
$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t)$$

where $0 < \rho \leq 1$ is the pheromone trail evaporation. The parameter ρ is used to avoid unlimited accumulation of the pheromone trails. If an arc is not chosen by the ants, its associated pheromone strength decreases exponentially. $\Delta\tau_{ij}^{k}(t)$ is the amount of pheromone ant *k* puts on the arcs it has visited; it is defined as follows:

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} 1/L^{k}(t) & if\ arc\,(i,j)\, is\, used\, by\, ant\, k \\ 0 & otherwise \end{cases}$$

Where $L^k(t)$ is the length of the *k*th ant's tour. The better the ant's tour is, the more pheromone is received by arcs belonging to the tour. In general, arcs which are used by many ants and which are contained in shorter tours will receive more pheromone and therefore are also more likely to be chosen in future iterations of the algorithm.

The idea is to give a strong additional reinforcement to the arcs belonging to the best tour found since the start of the algorithm; this tour is denoted as Tgb (global-best tour). Some limited results presented in suggest that the use of the elitist strategy with an appropriate number of elitist ants allows AS:



1 2 3 4

Figure 5

1. to find better tours, and
2. to find them earlier in the run.

VII. IMPROVEMENTS IN TSP

There are two main improvements for Travelling Salesman Problem. These improvements are as follows:

(A) Max-Min Ant System

MAX - MIN Ant System (MMAS) is a direct improvement over AS.

The solutions in MMAS are constructed in exactly the same way as in AS.

The main modifications introduced by MMAS with respect to AS are the following:

1. to exploit the best solutions found, after each iteration only one ant (like in ACS) is allowed to add pheromone.
2. to avoid search stagnation, the allowed range of the pheromone trail strengths is limited to the interval [τ_{min}, τ_{max}] that is, $\forall\tau_{ij}\ \overline{\tau}_{min} \le \tau_{ij} \le \tau_{max}$.
3. The pheromone trails are initialized to the upper trail limit, which causes a higher exploration at the start of the algorithm

(B) Update of Pheromone Trails

After all ants have constructed a solution, the pheromone trails are updated according to

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

Where $\Delta\tau_{ij}^{best} = 1/L^{best}$,

The ant which is allowed to add pheromone may be the iteration-best solution T^{ib}, or the global-best solution T^{gb}. Hence, if specific arcs are often used in the best solutions, they will receive a larger amount of pheromone.

In MMAS lower and upper limits on the possible pheromone strengths on any arc are imposed to avoid search stagnation. In particular, the imposed trail limits have the effect of indirectly limiting the probability pij of selecting a city *j* when an ant is in city i to an interval [pmin; pmax], with $0 < p_{min} \le p_{ij} \le p_{max} \le 1$. Only if an ant has one single possible choice for the next city, then pmin = pmax = 1. The lower trail limits used in MMAS are the more important ones, since the maximum possible trail strength on arcs is limited in the long run due to pheromone evaporation.

(C) Rank-Based Version of Ant System

Another improvement over Ant System is the rank-based version of Ant System (*A*Srank*)*. In ASrank, always the global-best tour is used to update the pheromone trails. Additionally, a number of the best ants of the current iteration are allowed to add pheromone. To this aim the ants are sorted by tour length (L1(t) · L2(t) · : : : · Lm(t)), and the quantity of pheromone an ant may deposit is weighted according to the rank *r* of the ant. Only the (w-1) best ants of each iteration are allowed to deposit pheromone. The global best solution, which gives the strongest feedback, is given weight *w*. The rth best ant of the current iteration contributes to pheromone updating with a weight given by max{0,w-r}. Thus the modified update rule is:

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

where $\Delta\tau_{ij}^{r}(t) = 1/L^{r}(t)$ and $\Delta\tau_{ii}^{gb}(t) = 1/L^{gb}$.

VIII. APPLICATIONS OF TSP

1. Lots of practical applications
2. Routing such as in trucking, delivery, UAVs
3. Manufacturing routing such as movement of parts along manufacturing floor or the amount of solder on circuit board
4. Network design such as determining the amount of cabling required
5. Two main types
   (a) Symmetric
   (b) Asymmetric
6. Genome Sequencing

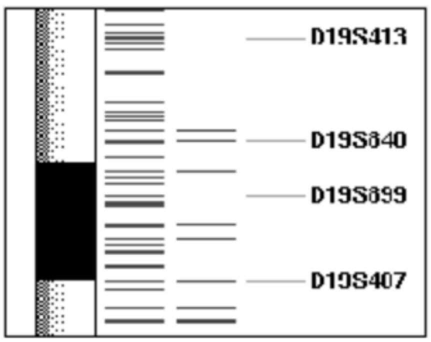
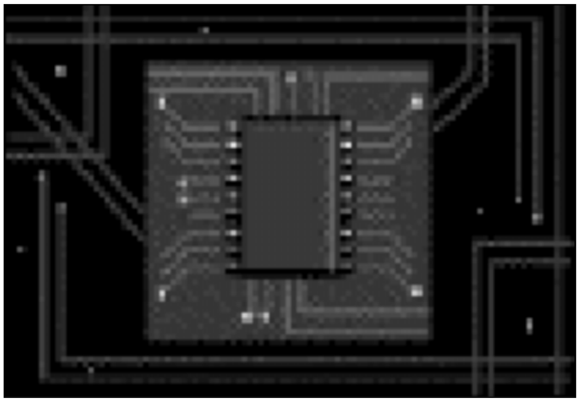


Figure 6

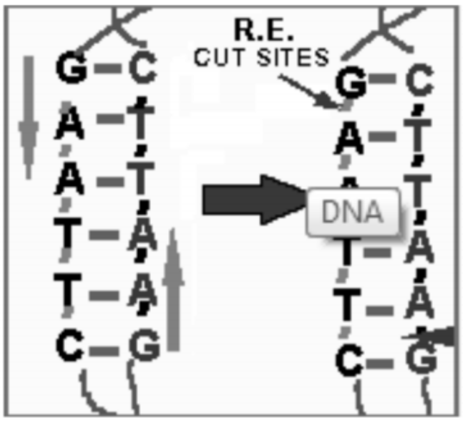
Researchers at the National Institute of Health have used Concorde's TSP solver to construct radiation hybrid maps as part of their ongoing work in genome sequencing. The TSP provides a way to integrate local maps into a single radiation hybrid map for a genome; the cities are the local maps and the cost of travel is a measure of the likelihood that one local map immediately follows another.

7. Scan Chain Optimization



A semi-conductor manufacturer has used Concorde's implementation of the Chained Lin-Kernighan heuristic in experiments to optimize scan chains in integrated circuits. Scan chains are routes included on a chip for testing purposes and it is useful to minimize their length for both timing and power reasons.

8. DNA Universal Strings



A group at AT&T used Concorde to compute DNA sequences in a genetic engineering research project. In the application, a collection of DNA strings, each of length *k*, were embedded in one universal string (that is, each of the target strings is contained as a substring in the universal string), with the goal of minimizing the length of the universal string. The cities of the TSP are the target strings, and the cost of travel is *k* minus the maximum overlap of the corresponding strings.

9. Touring Airports

Concorde is currently being incorporated into the Worldwide Airport Path Finder web site to find shortest routes through selections of airports in the world. The author of the site writes that users of the path-finding tools are equally split between real pilots and those using flight simulators.

10. A Tour Through MLB Ballparks

A baseball fan found the optimal route to visit all 30 Major League Baseball parks using Concorde's solver. The data for the problem can be found in the file mlb30.tsp (the data set is in TSPLIB format). The map above is a link to an interesting site devoted to current and past ballparks.

IX. CONCLUSION

The key to the application of ACS to a new problem is to identify an appropriate representation for the problem (to be represented as a graph searched by many artificial ants), and an appropriate heuristic that defines the distance between any two nodes of the graph. Then the probabilistic interaction among the artificial ants mediated by the pheromone trail deposited on the graph edges will generate good, and often optimal, problem solutions. Obviously, the proposed ACO algorithms for the TSP share many common features. Ant System can mainly be seen as a first study to demonstrate the viability of ACO algorithms to attack NP-hard combinatorial optimization problems, but its performance compared to other approaches is rather poor.

Ant Colony Optimization has been and continues to be a fruitful paradigm for designing effective combinatorial optimization solution algorithms. After more than ten years of studies, both its application effectiveness and its theoretical groundings have been demonstrated, making ACO one of the most successful paradigms in the metaheuristic area.

Acknowledgment

My sincere thank to Prof. Saurabh Shah, Head of Department, Computer Engineering, BIT for providing me the right ambience for carrying out this work. I express my immense pleasure and thankfulness to staff of the Department of Computer Engineering, BIT for their cooperation and support.

References

[1] M. Dorigo, Ant Colony Optimization Web page, *http://iridia.ulb.ac.be/mdorigo/ACO/ACO.html.*

[2] N. Ascheuer, Hamiltonian Path Problems in the On-line Optimization of Flexible Manufacturing Systems. Ph.D.Thesis, Technische Universität Berlin, Germany, 1995.

[3] M. E. Bergen, Canstraint-based Assembly Line Sequencing, Lecture Notes in ComputerScience, 2001.

[4] L. Bianchi, L. M. Gambardella, M. Dorigo. An Ant Colony Optimization Approach to the Probabilistic Traveling Salesman Problem. In Proceedings of PPSN-VII, Seventh International Conference on Parallel Problem Solving from Nature, Lecture Notes in Computer Science. Springer Verlag, Berlin, Germany, 2002.

[5] E. Bonabeau, M. Dorigo, G. Theraulaz, Nature, Volume Oliver, I., Smith, D. and Holland, J. R., 1987, A Study of Permutation Crossover Operators on the Travelling Salesman Problem, in: Proceedings of the Second International Conference on Genetic Algorithms, J.J. Grefenstette (ed.) (Lawrence Erlbaum, Hillsdale, New Jersey) pp. 224–230.

[6] Potvin, J. Y., (1993), The Traveling Salesman Problem: A Neural Network Perpective. *ORSA Journal of Computing*, 5 (4), 328–347.