SN Computer Science (2020) 1:95
https://doi.org/10.1007/s42979-020-0083-z

ORIGINAL RESEARCH

# Adaptive Dynamic Probabilistic Elitist Ant Colony Optimization in Traveling Salesman Problem

Amrita Chatterjee[1] · Eunjin Kim[1] · Hassan Reza[1]

Published online: 19 March 2020

**Abstract**
The ant colony optimization (ACO) is a population-based metaheuristic algorithm for the optimization problem, inspired by the foraging behavior of ants in the ant colony. One of its variants, the elitist ACO, further reinforces itself with the additional pheromone deposit to find the best path. Even though this usually leads to converging on the solution faster, it also has the drawback of getting stuck in local minima. In this paper, we describe a variation on elitist ACO where the pheromone contribution of the best path is further predicated by a probability factor. This probabilistic elitist ACO often produces a better solution for TSP, at the cost of a higher number of iterations. Some experimental results of this probabilistic elitist ACO are presented.

**Keywords** Ant system (AS) · Ant colony optimization (ACO) · Elitist ACO · Original ACO · Probabilistic elitist ACO

## Introduction

Ant colony optimization (ACO) [1] is a population-based metaheuristic algorithm [2, 3] for an optimization problem, inspired by the foraging behavior of ants [4] in ant colony [5]. ACO is expected to find the near-optimal solution like other nature-inspired metaheuristic algorithms or evolutionary algorithms such as particle swarm optimization algorithm (PSO), artificial bee colony algorithm, differential algorithm (DE). Thus, ACO is one of the leading candidates for producing a high-quality solution for traveling salesman problem (TSP) instances. The first version of ant algorithm, ant system (AS), was introduced by Dorigo et al. in 1992 [6] and further developed to ant colony system (ACS) by Dorigo and Gamberdella in 1996 [7].

Ant algorithms [8] are based on the behavior of ants in ant colonies. The ants are capable of finding the shortest route from their nest to a food source and coming back to the original place [9]. Initially they start their travel randomly at the same speed with no clue on how to find the shortest path. The shorter the path is, the more pheromones get accumulated over time; hence, more ants follow the route and form a shorter closed tour. Pheromone is the chemical substance excreted by the ants, and it works as a mechanism of stigmergy among ants [5]. It attracts other ants in search for food. The attractiveness of a given path depends on the intensity of pheromones that the ants have deposited during their travel [10]. The motion of ants is stochastic. Hence initially the ants choose any routes randomly, but they get to choose more preferable, i.e., shorter path for their food collection and return. Since the pheromone has the characteristics of evaporation, the paths which are less likely traversed by ants have less quantity of pheromones. As the longer path has also more evaporation of deposited pheromone, the quantity of pheromone on each edge of the path is indirectly proportional to the edge of the path. Therefore, the ants find the shortest path in their indirect communication via the amount of deposited pheromone on their path as the pheromone is a mechanism of the sign-based stigmergy among the ants.

This article is part of the topical collection “Advances in Computational Intelligence, Paradigms and Applications” guest edited by Young Lee and S. Meenakshi Sundaram.

✉ Amrita Chatterjee
amrita.chatterjee@und.edu

Eunjin Kim
eunjin.kim@und.edu

Hassan Reza
hassan.reza@und.edu

[1] School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks, ND 58202-9015, USA

ACO is used to solve combinatorial optimization problems such as vehicle routing, quadratic assignment, dynamic job scheduling, graph coloring and bin packing [11].

This paper tackles traveling salesman problem [12]. ACO was applied on TSP, and ants were used to determine the shortest path. In the original version of the ACO, there are many iterations in order to find the optimal solution [9]. On the other hand, in the elitist version of ACO, it converges to local minima sooner and does not have any probability of searching for more optimal path [9].

A hybrid version of the algorithm with probability 0.5 is being created in this paper. The probability was chosen as average of the original version of ACO and the elitist version of ACO. Two new versions of the algorithms were developed: probabilistic elitist ACO and dynamic probabilistic ACO. This hybrid approach does not converge to local minima quickly and leaves room for more optimal solution and still has less iterations as compared to the original ACO.

## Background

### Traveling Salesman Problem

Traveling salesman problem (TSP) is a combinatorial optimization problem to find the shortest path. In particular, TSP is an NP-hard problem where there exists no known algorithm to solve it in polynomial time; the optimal solution may be obtained in an exponential amount of time. In the map of a given number of cities, the salesman travels every city exactly once and has to return back to the original city making a whole closed tour. There are a number of practical applications of TSP such as planning bus lines, regular distribution of resources, finding of customer service route [12]. There are other application areas as well which are not related to traveling routes like chain diagram optimization, application in crystallography, industrial robot control, computer motherboard components layout, drilling holes for electric circuit [12].

### Ant System

The ant system is inspired by the behavior of the real ants where some artificial ants cooperate in order to find the solution virtual ants are used, which simulate the ant behavior and move from node to node to find the optimal solution. Ant system was the first ACO algorithm that was proposed [13]. The main characteristics are the change of pheromone values by all the ants those can complete the tour of the path. It is the first ACO algorithm which is first tested on TSP. It is represented by the following formula [13]:

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

where $\tau_{ij}$ is the pheromone deposit in the edge $(i, j)$, $\rho$ is the evaporation rate [14], and $\Delta\tau_{ij}^{k}$ is the pheromone update on the edge $(i, j)$ by each ant $k$ in the colony of $m$ ants.

### Ant Colony System

Ant Colony System is a distributed algorithm where ants play the role of cooperating agents in order to find the optimal solution. The ACS is built on top of Ant System in order to improve the efficiency when it is applied to both symmetric and asymmetric TSPs. The communication among the ants is made via the pheromone. Pheromones are deposited by the ants on the edges of the route they travel. In ACS the ants use a pseudorandom proportional rule. The probability of an ant to move to a particular city from the current city is determined by the parameter q0 and random variable q. ACS also introduces local pheromone update. With this the amount of pheromones decreases on the edges, and hence, the next ants can explore other routes which gives them a chance to find a better route than the previous ants.

### Ant Colony Optimization Algorithm for TSP

Ant system was first applied to TSP [15]. It utilizes a graph representation where each of the edges has cost measure, desirability measure. The pheromone is updated at every run time by the ants. For symmetric instances of the TSP, the desirability measure is $\tau(r, s) = \tau(s, r)$ and for asymmetric measure its $\tau(r, s) \neq \tau(s, r)$. Ants complete a closed tour by the probabilistic state transition rule. This is also known as a random-proportional rule. In this rule it helps the ants to take decision on which city to travel from a current city and then the next and so on to finish one full tour. The transition probability to choose the next city $s$ from $r$ is:

$$P_k(r, s) = \begin{cases} \frac{[\tau(r,s)]\cdot[\eta(r,s)]^{\beta}}{\sum_{u \in J_{k(r)}} [\tau(r,u)]\cdot[\eta(r,u)]^{\beta}}, & \text{if } s \in J_k(r) \\ 0, & \text{otherwise} \end{cases}$$

where $\tau$ $(r, s)$ is the pheromone amount in $(r, s)$ which implies a posteriori effectiveness of the move from $r$ to $s$, and $\eta(r, s)$ is the inverse of distance which implies a priori effectiveness of the move from $r$ to $s$ with its amplifying parameter $\beta$. $J_k(r)$ is the set of neighborhood nodes that are to be visited by ant $k$ positioned on city $r$. This formula of transition rule shows the probability of ant $k$ to travel from city $r$ to $s$ if s belongs to the set of cities otherwise '0.'

Global updating rule is applied once all the ants have completed their tour. Pheromones get updates eventually. Since the pheromones have the nature of evaporating; thus, the paths which do not get refreshed or not desirable by the ants will have a fewer pheromone. This rule was intended so that the total pheromone updated is equivalent to the remaining pheromone (after evaporation) added to the refreshed pheromone. This rule is applied at the end. Only globally best ant is allowed to construct the route (which deposited more pheromone). Thus, the intensity of pheromones is proportional to the shorter paths. The pheromone update is done by:

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \sum_{k=1}^{m}\Delta\tau_k(r,s)$$

where

$$\Delta\tau_k(r,s) = \begin{cases} \frac{1}{L_k}, & \text{if } (r,s) \in \text{ tour done by ant } k, \\ 0 & \text{otherwise.} \end{cases}$$

$L_k$ is the length of the tour performed by ant $k$, and $m$ is the number of ants. This pheromone updating rule is done in order to give higher reinforcement to better solutions. The pheromone updating formula is the combination of both the residue pheromone after it got evaporated and the refreshed pheromone. The pheromones which are located on the edges of the graph act as distributed long-term memory which is not present within the ants. The indirect communicated which is formed by this is termed as stigmergy [12].

Ant colony system introduced the three main rules: (1) state transition rule, (2) global updating rule and (3) local pheromone updating rule. A certain number of ants are initialized in a certain number of the city graph. Each of the ants start for their tour using both heuristic greedy search and by information obtained from the previous ants. The ants update the pheromone on the paths they are traversing by the local pheromone updating rule. Once all the tours are completed by the ants, the pheromones get updated globally while applying the global updating rule. Thus, the edges with more pheromones are visited by more ants. This generates the shortest path.

The ant chooses to go to the next city by applying the following rule:

$$s = \begin{cases} \arg\max_{u\in J_k(r)} \left\{[\tau(r,u)]\cdot\left[\eta^{\beta}(r,u)\right]\right\} \\ \quad \text{if } q \le q_0 \text{ (exploitation)} \\ S \text{ otherwise (biased exploration)} \end{cases}$$

where $J_k$ is the neighborhood of the city where an ant $k$ is currently located.

The state transition rule favors the tendency of the ants to move to the next city which contains more pheromone and thus the shortest path.

*Local Updating Rule* The edges of the graph get updated each time by the local update pheromone rule. An ant finishes each iteration, and the pheromone level is changed by the formula below.

$$\tau(r,s) \leftarrow (1-\rho)\cdot\tau(r,s) + \rho\cdot\Delta\tau(r,s)$$

*Global Updating Rule* The globally best ant which constructed the shortest route is allowed to deposit pheromone. This is performed at the end after all the ants have completed their tours. The following rule is applied, while the pheromone is updated:

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s) + \alpha\cdot\Delta\tau(r,s)$$

where $\alpha$ is the pheromone decay parameter. The pheromone update on the edge $(r, s)$, $\Delta\tau(r,s)$, is defined as

$$\Delta\tau(r,s) = \sum_{k}\Delta_{rs}^{k}$$

where

$$\Delta_{rs}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ uses edge } (r,s) \text{ in its tour.} \\ 0 & \text{otherwise.} \end{cases}$$

## Related Work

### Ant Colony Optimization for the Traveling Salesman Problem Based on Ants with Memory

Li et al. proposed a new model of ant colony optimization in order to solve the traveling salesman problem in [16]. Ants have the capability to remember the previous paths they have traversed, and thus, it will facilitate to find their best-so-far solution in less time. They have shown ants with memory can find better solution in less time. They introduced the idea of the memory of earlier solutions in order to make use of the previous best solution which was constructed by the ants who traversed the route before. Here they have proposed the algorithm having ants with memory named as Mant ACO. It was found that Mant algorithm found optimum value faster in the small-sized problems. Mant is a simple and interesting approach.

### Heuristics

In [12] exact algorithms or heuristics are termed in order to reach an optimal solution. Some of the exact algorithms mentioned are dynamic programming, explicit and implicit enumeration, branch and bound method. But there are some disadvantages to these methods. They work fine with a limited number of nodes (40–80). Thus, heuristics come to play when there is a large scale of problems. Heuristics solve specific types of problems. Heuristics comprise metaheuristics.

The advantage of metaheuristics is that they show only the way on how to apply the procedures in order to find solutions. Ant colony optimization belongs to the category of metaheuristics. Thus, ACO is used in order to solve problems like TSP.

In this strategy they have made two artificial paths. Ants were able to find shorter of the two paths. They set virtual ants in the scenario. These ants do not travel in a continuous manner, but they travel in jumps. After some time, they will jump to another graph node. Ants find the intensity of pheromones to be high on the edges, and hence, they travel through those paths.

### Solving the Traveling Salesman Problem Using the Ant Colony Optimization

In [12], Brezina and Čičková showed that the quality of the solution is directly proportional to the number of ants. The fewer the number of ants the more likely the path is going to be changed. In case more ants have involved, the pheromones are in large quality and thus, more ants get attracted to the path which eventually comes up to be optimal path. This ACO, when compared with the exact algorithms, proved that their method resulted in fewer iterations and thus less computation time.

### Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem

Dorigo and Gambardella showed that the ant colony optimization outperforms other nature-inspired algorithms like evolutionary computation, and simulated annealing in [9, 17]. In the final stage, one version of ant colony optimization which is augmented with local search procedure named ACS-3 [9] is compared to some best performing algorithms for symmetric and asymmetric traveling salesman problem. When a local optimization is added to ACS, then ACS-3 is obtained.

## Methodology

### Elitist ACO

Elitist is a variation on the original ACO algorithm where the best solution deposits additional pheromone on each edge that constitutes the current best solution during the global update rule application. Given by the equation:

$$\Delta\tau(r,s) = \sum_k \Delta_{rs}^k + \partial\tau(r,s)$$

where

$$\partial\tau(r,s) = \begin{cases} \frac{Q_{\text{val}}}{L_{\text{gb}}} & \text{if } (r,s) \in \text{global best tour,} \\ 0 & \text{otherwise.} \end{cases}$$

and $L_{\text{gb}}$ is the current global best tour and $Q_{\text{val}}$ is a parameter of the number of elite ants.

### Probabilistic Elitist ACO: Our Variation of the Elitist ACO

Our approach in this study is similar to the elitist ACO, but the additional pheromone contribution from the global best path is not deterministic, rather controlled by another probability factor, $p \in [0, 1]$:

$$\partial\tau(r,s) = \begin{cases} \frac{Q_{\text{val}}}{L_{\text{gb}}} & \text{if}(r,s) \in \text{global best tour with probability}, \boldsymbol{p}, \\ 0 & \text{otherwise.} \end{cases}$$

If $p < 0$, use the same value of $p$ from the previous iteration.

The elitist version of the algorithm has a tendency to get stuck in local minima due to the recurring pheromone contribution from the current best path starting from very first iteration. The probability factor prevents this saturation from happening, and in some sense, this variant can be thought of as a hybrid between the original and elitist ACO, thus benefitting from best path reinforcement of elitist ACO as well as new exploration potential in the original version.

Furthermore, we experiment with 2 different ways of choosing the value of the probability factor.

#### Static Probabilistic Elitist ACO (SPE ACO)

The 1st approach is to use a fixed static value of $p$. We choose $p = 0.5$ to ensure both the elitist and non-elitist update rule have an equal chance of being employed. This has the potential to prevent saturation at some local minima of the function to be optimized. The downside is that on problem instances where elitist update indeed would have led to quick convergence, however, this method still has an equal probability of choosing the non-elitist update rule. Intuitively we want non-elitist update rule to be more likely to be applied when the solution is far from convergence, while the elitist approach is more fruitful near convergence. Thus, we propose the adaptive dynamic probabilistic elitist approach to capture this behavior.

The elitist version of the algorithm has a tendency to get stuck in local minima due to the recurring pheromone contribution from the current best path starting from the very first iteration. The use of probability factor, $p$, may prevent it from an early convergence to the local minima by reducing the pheromone deposit. This approach with a probability factor can be considered as a hybrid between the original

ACO and elitist ACO, thus benefitting from best path reinforcement of elitist ACO as well as new exploration potential in the original ACO. In this paper, we studied two different ways to decide the proper value of the probability factor, $p$.

### Adaptive Dynamic Probabilistic Elitist ACO (ADPE ACO)

To capture the behavior of starting out with original update rule being most likely while increasing the probability of elitist updates as we approach convergence, we can define probability $p$ adaptive as a function of current best tour length of the current iteration as follows:

$$p = 1 - \frac{\text{the current best global tour length}}{\text{maximum global tour length}}$$

If $p < 0$ because the current best tour length is worse than maximum global tour length, use the same $p$ from the previous iteration.

Since the true maximum global tour length is unavailable at the beginning of the search, we use the tour length obtained in the first iteration of the algorithm itself as the maximum global length in the expectation of the tour length decreases as the search progresses through iteration. Note that our probability ensures that first few iterations will use the pheromone updates of the original approach, non-elitist (i.e., $p = 0$). As the iteration proceeds, however, the amount of pheromone update is more taken from the current global best tour which is improved from the previous iteration, and the probability to take an update from the elitist is increased. Thus, the update rule of the elitist ACO will be used with more likelihood. Hence, this dynamic probabilistic elitist approach trades the exploration and exploitation using the decreasing $p$; more exploration with the lower $p$ at the beginning of the search while more exploitation with the higher $p$ using the best tour as the search proceeds.

## Experiments

We generated our input data for the comparative study of the ACO algorithm variants in the following manner. We consider a 2-dimensional square of side 100 units. We pick random non-repeated integral ($x$, $y$) pairs to generate the list of cities. The Euclidean distance among the city points is the distance between cities for the traveling salesman problem. We generated datasets for 50 and 100 cities for our experiment, where each dataset contains 100 instances of the randomly generated TSP problem. We use number of cities multiplied by 100 as an upper bound of the TSP tour length to initialize the algorithm. Similarly, an arbitrary constant $c = 20$ multiplied by a number of cities is used as an upper bound on iterations.

We first present a case study on a 30-city [18], 10-instance small dataset to highlight some of the key findings. We then present results on the bigger datasets. For each instance of the dataset in the case study, the following results are examined:

1. Number of iterations to convergence
2. The tour length

Alongside original ACO and elitist ACO, we have incorporated the static value of probability ‘$p$’ to be 0.5 and the discussed adaptive dynamic probability for the pheromone update rule.

## Results

From the original approach, we have obtained the following results from 10 different random problem instances with corresponding required iterations to converge and tour length:

**Table 1** Comparisons on tour length for 30 cities and 100 ants

| TSP instance | Original ACO | Elitist ACO | Static Probabilistic Elitist ACO | Adaptive Dynamic Probabilistic Elitist ACO |
|---|---|---|---|---|
| 0 | 452.312 | 453.079 | 452.312 | 452.312 |
| 1 | 481.137 | 478.069 | 481.137 | 477.923 |
| 2 | 444.356 | 444.356 | 445.391 | 444.356 |
| 3 | 494.77 | 487.918 | 494.821 | 487.352 |
| 4 | 468.841 | 464.9 | 473.739 | 458.675 |
| 5 | 469.742 | 472.831 | 463.905 | 458.454 |
| 6 | 420.538 | 420.538 | 420.538 | 420.538 |
| 7 | 481.068 | 479.125 | 471.907 | 470.699 |
| 8 | 494.329 | 492.287 | 499.828 | 503.535 |
| 9 | 501.443 | 498.12 | 492.164 | 495.098 |

**Table 2** Comparisons of the number of iterations for 30 cities and 100 ants

| TSP Instance | Original ACO | Elitist ACO | Static Probabilistic elitist ACO | Adaptive Dynamic Probabilistic Elitist ACO |
|---|---|---|---|---|
| 0 | 390 | 330 | 510 | 90 |
| 1 | 240 | 480 | 240 | 480 |
| 2 | 180 | 210 | 240 | 330 |
| 3 | 240 | 180 | 150 | 210 |
| 4 | 90 | 60 | 390 | 480 |
| 5 | 150 | 120 | 90 | 60 |
| 6 | 120 | 60 | 90 | 90 |
| 7 | 270 | 150 | 330 | 120 |
| 8 | 150 | 90 | 510 | 150 |
| 9 | 510 | 180 | 480 | 240 |

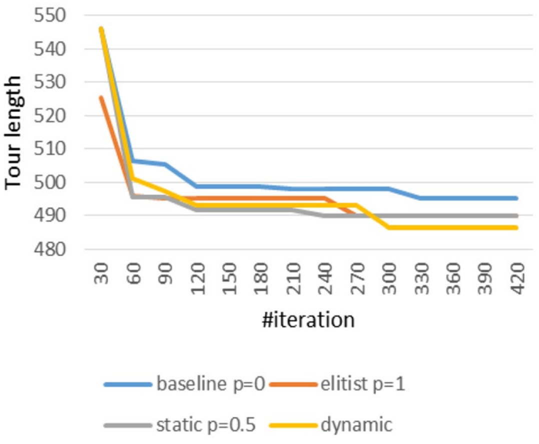

**Fig. 1** Comparison of convergence speed

Table 1 shows the solution tour length for each instance of problems for four ACO algorithms for 30 cities and 100 ants. The optimal or the best tour lengths are only recorded. For example, for the instance of '0,' the number of iterations is 390 and the best tour length is 452.312. The distance between the two nodes is measured by Euclidean distance with *x* and *y* coordinates.

For the ACO elitist approach we generated the results the same way as we did for the original ACO approach to generate the iterations and the tour lengths (optimal/shortest length).

Table 2 shows the number of iterations and for respective problem instances in each approach. The results show that in static probabilistic variant the number of iterations is greater compared to other approaches. However, the dynamic variant is able to achieve similar quality result in fewer iterations

Figure 1 shows convergence speed to the optimal tour in 4 approaches. Our approach of ADPE ACO has converged to a steady state faster than the other three approaches. But, in the end there is a change in the graph. It shows that our method has still gone further searching for an optimal path. Unlike the elitist ACO and SPE ACO, ADPE ACO did not remain at a certain steady state which may be considered as the plateau of local minima length, but it made further improvement after the plateau. Though a greater number of iterations were required for ADPE ACO compared to the other original ACO and elitist approach, ADPE ACO was more successful in finding the shorter length tour of the near-optimal solution.

**Table 3** Tour length in the instances of 50 cities and 100 ants (only 10 instances shown out of 100)

| TSP Instance | Original ACO | Elitist ACO | SPE-ACO | ADPE-ACO |
|---|---|---|---|---|
| 0 | 583.019 | 581.992 | 581.473 | 580.446 |
| 1 | 608.768 | 614.685 | 623.491 | 615.754 |
| 2 | 616.019 | 609.674 | 602.988 | 599.643 |
| 3 | 641.116 | 639.439 | 626.712 | 641.713 |
| 4 | 637.053 | 623.625 | 635.226 | 635.226 |
| 5 | 592.884 | 594.881 | 594.178 | 587.237 |
| 6 | 610.768 | 594.685 | 602.958 | 594.685 |
| 7 | 618.028 | 618.49 | 618.925 | 614.981 |
| 8 | 599.841 | 602.103 | 583.772 | 591.477 |
| 9 | 585.347 | 585.409 | 607.089 | 598.786 |

*Choice of Parameters* The parameters which are involved in this are $\alpha$, $\beta$, QVAL, $\tau$, $p$. $0 < \alpha<1$ is the pheromone decay parameter. $\beta$ is the parameter which determines the relative importance of pheromone versus distance. This parameter raises the weight of distance over pheromone. $\tau$ is the evaporation rate, the value of QVAL is set to 100, $p$ is the probability constraint that is used in our method, and we have set the value to be 0.5. The probability value is taken to be 0.5 so that there is equal chance between elitist and original behavior.

Table 3 and Table 4 show the average tour length of 50 cities with 100 ants and the mean of the tour length over 10 problem instances, respectively.

The same experiments were performed for the problem instances of 100 cities and 200 ants. Their results are summarized in Tables 5 and 6.

## Conclusion

In this paper we studied closely the three different algorithms the original ACO, elitist ACO and our approach of two probabilistic elitist ACOs: static probabilistic elitist ACO (SPE ACO) and adaptive dynamic probabilistic elitist ACO (ADPE ACO). Our approach of both probabilistic elitist ACO reasonably outperformed the original ACO and the elitist ACO in the points of the average shortest tour length and the convergence speed. In particular, ADPE ACO made further improvement to the global optimal tour length after it once reached to the steady state near to the

**Table 4** Average tour length of 50 cities 100 ants

| | Elitist-ACO | Original-ACO | SPE-ACO | ADPE-ACO |
|---|---|---|---|---|
| | (p=1) | (p=0) | (p=0.5) | |
| mean | 597.262 | 597.672 | 596.798 | 596.05 |
| stdev | 33.234 | 32.91 | 33.611 | 32.407 |
| # of best results | 32 | 21 | 30 | 39 |
| rank | 2 | 4 | 3 | 1 |

**Table 5** Tour length in the instances of 100 cities and 200 ants (only 10 instances shown out of 100)

| TSP Instance | Original ACO | Elitist ACO | Static Probabilistic Elitist ACO | Adaptive Dynamic Probabilistic Elitist ACO |
|---|---|---|---|---|
| 0 | 864.218 | 861.576 | 874.16 | 844.962 |
| 1 | 836.949 | 840.198 | 824.723 | 825.277 |
| 2 | 858.032 | 853.081 | 883.401 | 869.011 |
| 3 | 843.834 | 837.238 | 863.536 | 851.31 |
| 4 | 852.295 | 844.843 | 840.718 | 850.013 |
| 5 | 829.119 | 839.951 | 825.754 | 832.84 |
| 6 | 801.579 | 801.265 | 801.09 | 801.858 |
| 7 | 806.322 | 804.324 | 804.276 | 814.155 |
| 8 | 804.88 | 804.671 | 804.601 | 794.419 |
| 9 | 853.326 | 848.031 | 855.191 | 841.165 |

**Table 6** Average tour length of 100 cities 200 ants

| | Elitist-ACO | Original-ACO | SPE-ACO | ADPE-ACO |
|---|---|---|---|---|
| | (p=1) | (p=0) | (p=0.5) | |
| mean | 846.403 | 845.886 | 845.688 | 844.977 |
| stdev | 36.307 | 38.384 | 38.304 | 37.524 |
| # of best results | 21 | 26 | 30 | 26 |
| rank | 4 | 2 | 1 | 2 |

optimal tour length for a certain number of iterations, i.e., the plateau of the local minima.

## Future Work

Future work will involve doing the study on a larger sample size and doing appropriate statistical analysis. Parameters could be varied as well.

## References

1. Parpinelli RS, Lopes HS, Freitas AA. Data mining with an ant colony optimization algorithm. IEEE Trans Evol Comput. 2002;6(4):321–32.
2. Feller E, Rilling L, Morin C. Energy-aware ant colony based workload placement in clouds. In: Proceedings of the IEEE/ACM international conference on grid computing (GRID); 2011, p. 26–33.
3. Dorigo M, Stützle T. The ant colony optimization metaheuristic: algorithms, applications, and advances. In: Glover F, Kochenberger GA, editors. Handbook of metaheuristics. Boston: Springer; 2003. p. 250–85.
4. Zhang Z, Zhang X. A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation. In: Proceedings of the 2nd international conference on industrial mechatronics and automation; 2010, p. 240–243.
5. Blum C. Ant colony optimization: introduction and recent trends. Phys Life Rev. 2005;2(4):353–73.
6. Hajizadeh Y, Christie M, Demyanov V. Ant colony optimization for history matching and uncertainty quantification of reservoir models. J Pet Sci Eng. 2011;77(1):78–92.
7. Dorigo M, Di Caro G, Gambardella LM. Ant algorithms for discrete optimization. Artif Life. 1999;5(2):137–72.
8. Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In: The European conference on artificial life; 1991, p. 134–142.
9. Dorigo M, Gambardella LM. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput. 1997;1:53–6.
10. Maier HR, et al. Ant colony optimization for design of water distribution systems. J Water Resour Plan Manag. 2003;129(3):200–9.
11. Di Caro G, Dorigo M. Ant colony optimization and its application to adaptive routing in telecommunication networks; 2004.
12. Brezina I Jr, Čičková Z. Solving the travelling salesman problem using the ant colony optimization. Manag Inf Syst. 2011;16(4):010–4.
13. Dorigo M. Learning and natural algorithms. Ph.D. thesis, Politecnico di Milano, Milan, Italy, 1992.
14. Cordón O, de Viana IF, Herrera F. Analysis of the best-worst ant system and its variants on the TSP. Mathw Soft Comput. 2002;9(2/3):177–92.
15. Bullnheimer B, Hartl RF, Strauss C. A new rank based version of the ant system. A computational study. Cent Eur J Oper Res Econ. 1997;7:25–38.
16. Li B, Wang L, Song W. Ant colony optimization for the traveling salesman problem based on ants with memory. In: ICNC 2008; 2008, p. 496–501.
17. Lee Z, Lee CY, Su SF. An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem. Appl Soft Comput. 2002;2(1):39–47.
18. Durbin R, Willshaw D. An analogue approach to the travelling salesman problem using an elastic net method. Nature. 1987;326:689–91.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.