Engineering Science and Technology, an International Journal 20 (2017) 1242–1248

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

The application of ant colony optimization in the solution of 3D traveling salesman problem on a sphere

Hüseyin Eldem [a,*], Erkan Ülker [b]

[a] Karamanoğlu Mehmetbey University, Computer Technologies Department, Karaman, Turkey
[b] Selçuk University, Computer Engineering Department, Campus, Konya, Turkey

ARTICLE INFO

Article history:
Received 7 July 2017
Revised 18 August 2017
Accepted 23 August 2017
Available online 14 September 2017

Keywords:
Ant colony optimization
Metaheuristic
Spherical geometry
Max-Min Ant System
nonEuclidean TSP

ABSTRACT

Traveling Salesman Problem (TSP) is a problem in combinatorial optimization that should be solved by a salesperson who has to travel all cities at the minimum cost (minimum route) and return to the starting city (node). Todays, to resolve the minimum cost of this problem, many optimization algorithms have been used. The major ones are these metaheuristic algorithms. In this study, one of the metaheuristic methods, Ant Colony Optimization (*ACO*) method (Max-Min Ant System – *MMAS*), was used to solve the *Non-Euclidean* TSP, which consisted of sets of different count points coincidentally located on the surface of a sphere. In this study seven point sets were used which have different point count. The performance of the *MMAS* method solving Non-Euclidean TSP problem was demonstrated by different experiments. Also, the results produced by *ACO* are compared with Discrete Cuckoo Search Algorithm (*DCS*) and Genetic Algorithm (*GA*) that are in the literature. The experiments for *TSP* on a sphere, show that *ACO's* average results were better than the *GA's* average results and also best results of *ACO* successful than the *DCS*.

1. Introduction

Traveling salesman problem (*TSP*) is the problem of a salesman who needs to visit all the cities in the schedule and return to the starting point by spending less. One of the parameters such as path, time, cost and path in *TSP*, can be optimized. *TSP* is called also as a Hamiltonian path problem that is used in computer science for data modeling. The *TSP's*, evaluated in discrete and combinatorial problems has been comprehensively studied in the field of similar graph theory problems. *TSP* is considered in two categories as symmetric and asymmetric. In the symmetric *TSP*, always, the distance between x. city and y. city is equal, i.e., $d_{xy} = d_{yx}$. In the asymmetric *TSP*, the distance matrix between cities may not be the equal for all cities.

In order to solve *TSP*, many methods have been developed. They are divided into two groups as heuristics and exact methods in terms of obtaining the optimal results. Branch-and-bound, branch-and-cut and iterative improvement are the exact solution methods for *TSP* [4,23]. Various heuristic algorithms, based on Simulated Annealing [21,12], Genetic Algorithms (*GA*) [35,16,18,38,26]), Tabu Search [14,15,25], Artificial Neural Networks [19,22,30,28] and Ant Colony Systems [2,3,5,13,7,8,33,32,1] have been developed which make the closest possible solutions to the best solutions at a reasonable time. In the meantime, to solve *TSP*, 2-opt, 3-opt and 4-opt local search algorithms were also used [20]. Some researchers to make optimum results of *TSP*, have studied hybrid evolution algorithms [24,39,27,34,25]. Some *TSP* applications were executed on the basic 3D geometric figures like spheres and cuboids [36,37,31,29,10,9,11]. An algorithm was proposed by making the solution of *TSP* with *GA* on a cuboid [36] and a sphere [37]. In [31], the particle swarm optimization algorithm (*PSO*) was proposed by making the solution of *TSP* on cuboid. An algorithm was proposed by making the solution of *spherical TSP* with Cuckoo Search algorithms on a sphere [29]. And also, algorithms were proposed by making the solution of *sphericalTSP* and *cuboidTSP* with ACO and PSO on a sphere and cuboid [9].

One of the metaheuristic algorithms, Ant colony optimization (*ACO*), used to solve discrete optimization problems, was proposed by Marco Dorigo in 1992 as a PhD thesis [5]. *ACO* is a metaheuristic computational algorithm technique. *ACO* was used to solve graph problems by investigating possible paths on the graphs. *ACO* is inspired by the behavior of ants that provides to find shortest distance between their nest and food resource by means of pheromone.

* Corresponding author.
E-mail addresses: heldem@kmu.edu.tr (H. Eldem), eulker@selcuk.edu.tr (E. Ülker).

Peer review under responsibility of Karabuk University.

http://dx.doi.org/10.1016/j.jestch.2017.08.005
2215-0986/© 2017 Karabuk University. Publishing services by Elsevier B.V.

Ants choose shortest way while searching food resources rapidly in progress of time. Various *TSP* applications have been successfully solved with *ACO* techniques.

MAX-MIN Ant System *(MMAS)* that is an improvement over the Ant System *(AS)* proposed by Stützle and Hoos [32]. *MMAS* differs from AS at pheromone update. In AS, when complete the tours, each of ants updates their pheromone trials. But in *MMAS* just the best ant updates the pheromone trials and pheromone level is bounded between minimum-maximum limit.

In this study, *TSP* was solved for the points on a sphere by *ACO* algorithm (*MMAS*). To our knowledge, so far there is no study solving *TSP* by this technique in 3D. For the available *TSP*s, the coordinates of the points and the distances between them are known. Since all the points are present on a sphere and passage from one point to the other is carried out from the sphere surface, this problem is different from the existing *TSP*s. The study covers the definition of points on a sphere, finding the distances between the points and adaptation of the problem to the *ACO*.

2. The basic of a sphere

A sphere is a 3D object made up of points that are at the same distance from a given point in space. Every point (with coordinates of *x, y, z*) distributed at an equal distance *r* from the center is located on the sphere surface. In other words, a sphere is obtained by turning of an arc, drawn at a same distance from the origin with coordinates of *x-y*, around the z-axis. The relation between the *x, y, z* coordinates and the radius of a sphere is formulated by the Eq. (1):

$$r = \sqrt{x^2 + y^2 + z^2} \tag{1}$$

The radius of a sphere, *r* is the distance from the center (point A) to the points on the sphere (B, C, D and E) and shown in Fig. 1. Every point on the sphere has coordinates of *x, y, z* and these values always satisfy the Eq. (1).

When a problem is considered on a sphere, the first example that comes to mind is the geometric similarity of the Earth to a sphere. The circle passing throughout the sphere center and bounded by a sphere is big circle called equator of the Earth. This circle becomes important when the minimum distance between two points, i.e., geodesic, on a sphere along the lower cross-section is considered. The curves are called as geodesics on any surface of sphere that minimize the distances between their points [37].

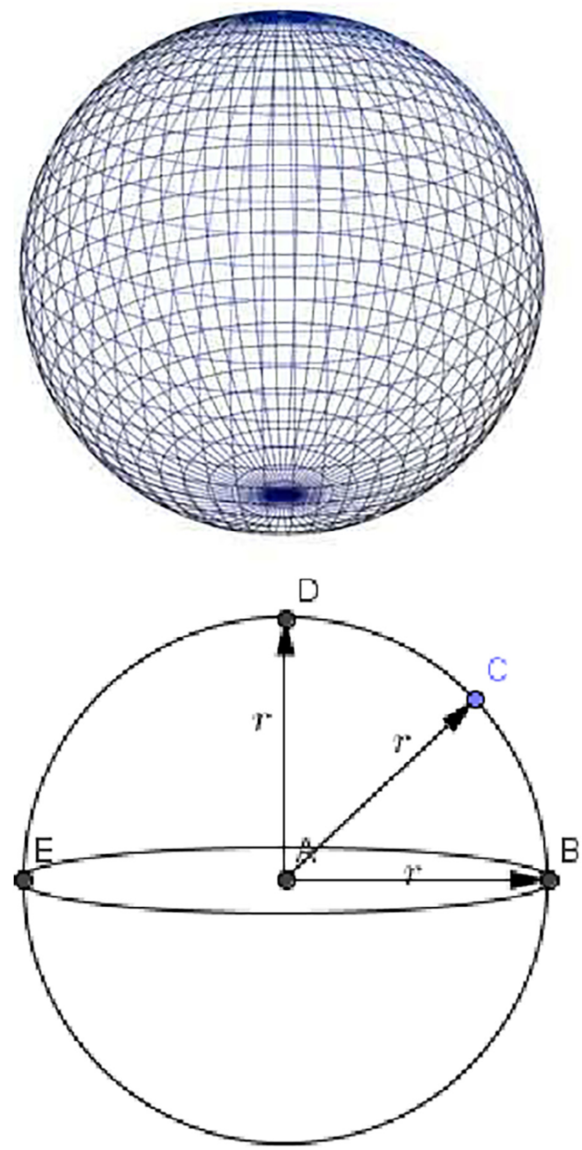

Fig. 1. Spherical surface and the radius.

2.1. Mathematical notation of points on a sphere

Euclidean curves have a single dimension. These curves can be defined by a single parameter called *u* along a 3D curve. In other words, in terms of parameter *u* points out the Cartesian coordinates. Any point on a curve can be represented by a point vector function according to the given reference Cartesian coordinates [17]:

$$P(u) = (x(u), y(u), z(u)) \tag{2}$$

Generally, coordinate equations can be set up in a way that where parameter *u* is described between *0* and *1*. As an example, a circle on the xy-plane centered at the origin is defined in a parametric form given below [17]:

$$x(u) = r.\cos(2\pi u) \quad y(u) = r.\sin(2\pi u) \quad z(u) = 0, \quad 0 \leqslant u \leqslant 1 \tag{3}$$

Circles and circular curves can also be defined in other parametric forms. Sloping Euclidean surfaces are two-dimensional varieties described by parameters *u* and *v*. A coordinate position on a surface can be represented by a parameterized vector function with *u* and *v* parameters for the coordinate values of *x, y* and *z* [17].

$$P(u) = (x(u, v), y(u, v), z(u, v)) \tag{4}$$

Each of Cartesian coordinate values is a function of surface parameters *u* and *v*, which change between *0* and *1*. The coordinates of a spherical surface centered at the origin with a radius *r* can be defined by the Eq. (5) [17]:

$$\begin{aligned} x(u, v) &= r.\cos(2\pi u).\sin(\pi v) \\ y(u, v) &= r.\sin(2\pi u).\sin(\pi v) \\ z(u, v) &= r.\cos(\pi v) \end{aligned} \tag{5}$$

where, the parameters *u* and *v* define the constant lines of latitude and constant lines of longitudes on the surface, respectively [17]. To illustrate, *x, y, z* coordinates for the different values of parameters *u* and *v* were calculated according to the Eq. (5) and are given in Table 1. Note that *r* was taken as *1* and upon the increase in value of *r*, the values of *x, y, z* should also be increased in the same manner.

2.2. Finding the shortest distance between all pairs of points on the surface of unit sphere

On a spherical surface, minimum distance between two points (P_1, P_2) is along the arc of a great circle (Fig. 2). So, in radians, the

Table 1
Coordinates on a spherical surface for different values of parameters u and v.

u	v	x	y	z
0	**0**	0	0	1
0	**0.5**	1	0	6.123233e−17
0	**1**	1.224646e−16	0	−1
0.5	**0**	0	0	1
0.5	**0.5**	−1	1.224646e−16	6.123233e−17
0.5	**1**	−1.224646e−16	1.499759e−32	−1
1	**0**	0	0	1
1	**0.5**	1	−2.449293e−16	6.123233e−17
1	**1**	1.224646e−16	−2.999519e−32	−1

value of angle *theta* (θ) can be used between two vectors $\vec{V1}$ and $\vec{V2}$. The scalar product of two vectors is [37]:

$$\vec{V1} \cdot \vec{V2} = |\vec{V1}||\vec{V2}| cos\theta \tag{6}$$

where θ is a small angle between the direction of two vectors.

$$\vec{V1} \cdot \vec{V2} = P_{1X}P_{2X} + P_{1Y}P_{2Y} + P_{1Z}P_{2Z} \tag{7}$$

The size of vectors $\vec{V1}$ and $\vec{V2}$ are 1 for points on the surface of unit sphere. So shortest distance is [37]:

$$\theta = \arccos(\vec{V1} \cdot \vec{V2}) \tag{8}$$

The problem is *nonEuclidean TSP* and different from *Euclidean TSP*. Because on the *Euclidean TSP*, the shortest distance between two points (P_i, P_j) is calculated by using Euclidean distance which is a straight line) instead of arc length [37]. The points distance matrix on a sphere is the same with symmetric *TSP* (d($P_{i:}$ P_j) = d ($P_{j:}$ P_i)).

3. On the unit sphere, solution of TSP by using ACO (MMAS)

The three-dimensional TSP to be applied to the surface of the sphere differs from the normal two-dimensional TSP. In two-dimensions, the ant moves only in one plane. But in a three-dimensional TSP, salesperson (agent, ant, etc.) could only travel between two points through the surface (not through inside of sphere). In this study, the TSP points are on the surface of the sphere.

Similarly to standard *TSP*, the problem to be solved can be described as the detection of the shortest tour distance for an robot to travel all points (*N* points in total with known coordinates and stored distance matrix) located on a surface of a sphere and return to the original point. In this study, it is aimed to solve the described problem by the *MMAS* method that improved pheromone update according to *ACO*.

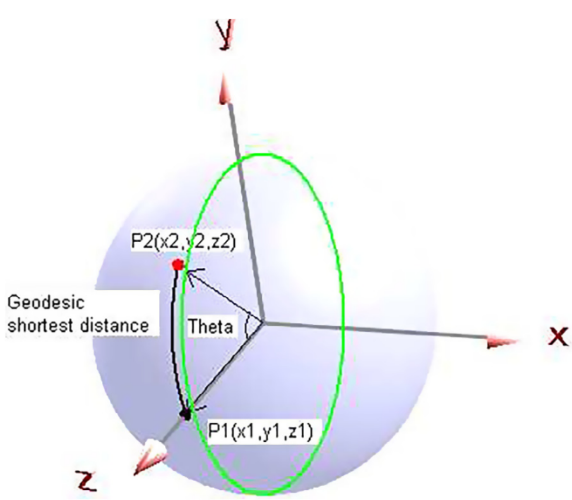

Fig. 2. Geodesic: shortest distance between two points on a spherical surface [37].

According to Eq. (8), the solution of the problem is equal to standard *TSP*, after calculation of the distances between each pair of points. After this step, the solution of the problem can be examined by each method to solve the *TSP* described in the literature survey of the introduction section. In this article, solutions for a specific number of randomly generated points were obtained for each iteration by using *MMAS*.

General structure of the *MMAS*:
set initial pheromone level for all edges;
place ants to random cities on the problem;
for each iteration do:
  According to the probability function, move each ant to next city
    for each ant with a complete tour do:
    if ant's tour length is best of tours
      calculate the pheromone on each edge of best ant's tour
        if new pheromone level > τ_{max}
          set pheromone level to τ_{max}
        else if pheromone level < τ_{min}
          set pheromone level to τ_{min}
      apply pheromone update;
      if (iteration best tour is shorter than the global solution)
        update global solution to iteration best
    end
    until all ants have completed its solution
end

According to this general structure, first, the initial values of the parameters of the *ACO* algorithm are adjusted. Thus, the initial pheromone data of each corner between points are adjusted and written into the pheromone matrix. The distance matrix, where the distance of each point to all other points is given by Eq. (8), is obtained. Initially, in *ACO* algorithm, each agent ant, helping for solution, is located on nodes (cities) randomly. In each iteration, ants select the next city that they will travel according to Eq. (9).

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)^{\alpha}.[\eta_{ij}]^{\beta}]}{\sum[\tau_{ik}(t)]^{\alpha}.[\eta_{ik}]^{\beta}} & \text{if} j \in \text{allowed}_k \end{cases} \tag{9}$$

At a time, t; $\tau_{ij}(t)$ is the density of the mark left at the arc. η_{ij}, i.e., visibility, is multiplicative reciprocal of d_{ij}. d_{ij}, is the distance between i and j points on a sphere. α and β are two parameters control the importance of the pheromone according to its visibility. After completing the tours of the each ants, i.e., and finding the solution of each ants, the evaporation of the pheromone and the updating density process are performed respectively. With evaporation, it

Table 2
Calculated average Spherical *TSP* tour lengths with *GA* [37] and *ACO (MMAS)* for $N = 100, 150, 200, 250, 300, 350$ and 400 points on the surface of unit sphere.

Evolution number	Number of points													
	100		150		200		250		300		350		400	
	GA	ACO	GA	ACO	GA	ACO	GA	ACO	GA	ACO	GA	ACO	GA	ACO
10	90.1194	**24.810**	157.6443	**29.861**	227.2337	**37.717**	299.1799	**42.686**	374.1841	**46.633**	441.3127	**50.870**	532.6288	**54.706**
20	72.0467	**24.745**	132.9872	**29.796**	187.7657	**37.564**	265.2366	**42.262**	333.5504	**46.588**	393.6967	**50.508**	472.2036	**54.325**
30	53.0306	**24.676**	100.9841	**29.762**	162.8873	**37.420**	224.2341	**42.150**	288.0024	**46.450**	353.1749	**50.456**	439.496	**54.260**
40	42.6922	**24.647**	82.2588	**29.623**	141.7211	**37.461**	200.0556	**42.024**	259.6622	**46.432**	323.2888	**50.350**	393.7457	**54.037**
50	37.1942	**24.401**	70.5737	**29.615**	115.1155	**37.657**	165.5674	**42.155**	226.1183	**46.371**	291.1789	**50.234**	354.375	**54.136**

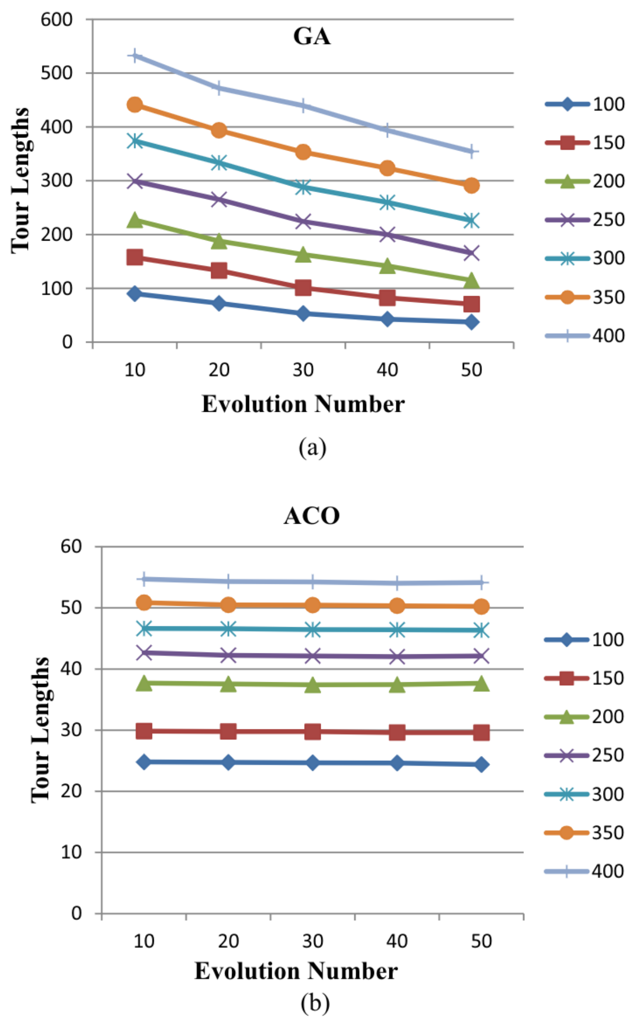

Fig. 3. Average tour lengths for different amount points on the surface of unit sphere founded by *GA* [37] and *ACO* solutions.

may be possible to forget the wrong solutions and give opportunity to the new tours by preventing pheromone accumulation. In the meantime, the best solution provided by ants is a solution that will provide global best displacement and is the best end result when iterations are completed.

4. Experimental results

On this experiment, *MMAS* is tested for $N = 100, 150, 200, 250, 300, 350$, and 400 points for the unit sphere. For each value of N, *MMAS* algorithm for *TSP* on unit sphere were repeated *100* times. Instead of using a predefined set of points to generalize the results in a unit sphere, a new set of random points was created for each trial. Like Dorigo et al.'s paper, the default value of α and β parameters was 1 and 5 respectively. And also pheromone trail evaporation parameter ρ was set to 0,5 [6]. In this paper, in case study #1, results of our method *ACO (MMAS)* was compared with the *GA* method given in [37]. In case study #2, results obtained using *ACO (MMAS)* algorithm was compared with the results of novel Discrete Cuckoo Search Algorithm (*DCS*) [29] having better performance than *GA* based method. The results were obtained by using Matlab R2010a software.

Uğur et al. [37] calculated the results of *TSP* on a sphere with *GA* and for different *GA* generation sizes (10, 20, 30, 40, and 50 generation) and for $N = 100, 150, 200, 250, 300, 350$, and 400 points. For each generation, Uğur et al. [37] fixed *GA* population size as *100*. The mutations of individuals in a population observed in every generation can be called evolution. The total evolution equals the population size multiplied by the number of generation.

For the *TSP's* solutions in the literature, generally, the number of ants should be equal to the number of cities for optimum results. In this study, *spherical TSP* solution was applied considering this equality. To make a fair comparison with the *GA* [37] results in the literature and to achieve an equal number of evolution for *MMAS*, the number of tours for each generation is determined by using Eq. (10):

$$Number\,of\,Tours_{MMAS} = \frac{The\,Size\,of\,Generation_{GA} \times The\,Size\,of\,Population_{GA}}{Number\,of\,Ants_{MMAS}} \tag{10}$$

For the optimum length of the tours, the results were obtained for different evolution numbers, i.e., 10, 20, 30, 40 and 50 evolutions. For all simulations, constants are $\alpha = 1$, $\beta = 5$ and $\rho = 0.50$ (the coefficient of evaporation). And also the number of ants were

Table 3
Calculated average *Spherical TSP* tour computation times with *ACO (MMAS)* for $N = 100, 150, 200, 250, 300, 350$, and 400 points on the surface of unit sphere.

Evolution Number	Number of Points						
	100	150	200	250	300	350	400
	Time (sec)	Time (sec)	Time (sec)	Time (sec)	Time (sec)	Time (sec)	Time (sec)
10	5,79	9,45	12,68	16,81	25,47	27,78	38,20
20	18,52	30,01	39,43	50,93	52,67	55,78	64,21
30	27,43	42,45	58,44	75,70	78.73	83,65	102,73
40	24,62	43,48	60,16	76,27	87,78	108,93	127,01
50	28,07	45,75	63,12	83,90	108,64	139,08	167,39

Table 4
Calculated best Spherical *TSP* tour lengths with *DCS* [29] and *ACO (MMAS)* for *N* = 100, 150, 200, 250, 300, 350 and 400 points on the surface of unit sphere.

Evolution Number	Number of Points													
	100		150		200		250		300		350		400	
	DCS	ACO	DCS	ACO	DCS	ACO	DCS	ACO	DCS	ACO	DCS	ACO	DCS	ACO
10	25,5412	**22,7628**	31,1714	**28,8980**	36,5940	**34,5537**	41,1404	**40,4088**	45,6111	**43,9647**	45,6336	**44,1669**	52,2950	**51,7597**
20	25,4120	**22,6715**	31,1952	**28,7744**	36,4697	**34,4814**	40,9506	**39,1512**	45,3540	**43,7251**	45,3357	**44,1524**	51,9555	**50,7783**
30	25,3846	**22,3405**	30,9975	**28,6522**	36,4538	**34,4579**	40,8280	**39,0770**	45,2773	**43,7411**	45,2284	**44,1019**	51,8304	**50,6808**
40	25,3409	**22,3147**	30,9768	**28,0550**	36,3856	**34,2200**	40,7622	**38,9580**	45,2367	**42,7122**	45,1079	**44,0692**	51,7429	**50,6607**
50	25,3341	**22,2944**	30,9735	**27,9894**	36,3792	**34,1734**	40,7557	**38,7441**	45,2005	**42,3198**	45,0918	**43,9462**	51,6725	**49,5383**

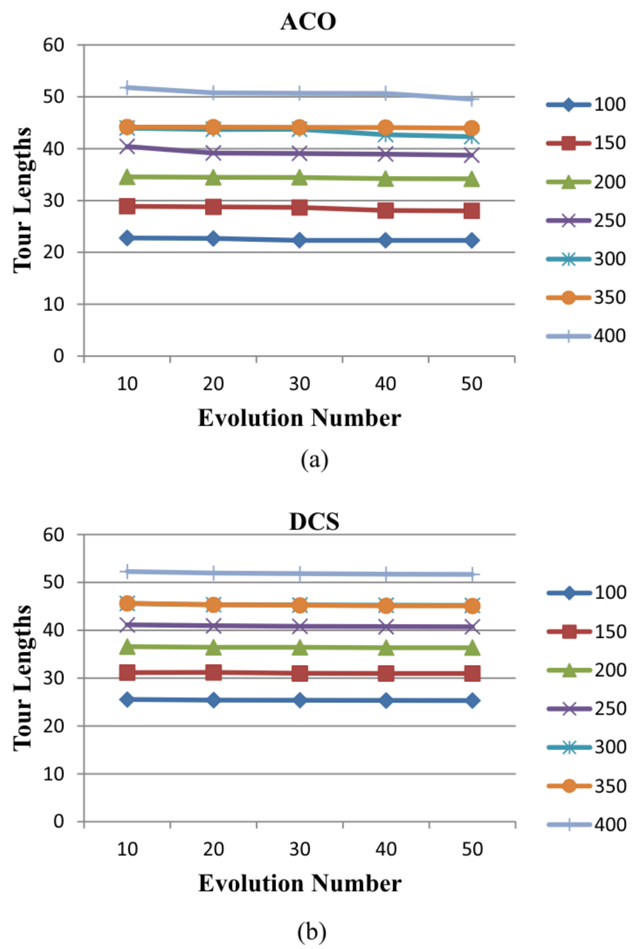

Fig. 4. Best tour lengths for different amount points on the surface of unit sphere founded by *DCS* [29] and *ACO* solutions.

equal to that of points (cities) number for all experiments. The calculated average tour distances by proposed *ACO (MMAS)* approach were compared with *GA*'s average tour distances as shown in Table 2 and Fig. 3. Meanwhile, average calculation times are shown in Table 3. The values were obtained for ways at a unit surface of the sphere.

If Table 2 is examined, when evolution number for 150 points is taken as 50, the tour length obtained with ACO is 29.615 and with GA is 70.5737. For example, when evolution number is 50 for 250 and 400 points, GA's tour lengths are 165.5674 and 354.375, ACO's tour lengths are 42.155 and 54.136 respectively. When *GA* and *MMAS* results were compared, it is observed that the results of the *MMAS* algorithm were much more successful than *GA* for the spherical *TSP*, considering the other columns in Table 2.

In literature [29], Discrete Cuckoo Search Algorithm (*DCS*) is also tested with the same generation size and number of points as like in Uğur et al. [37] and results are given. But, in Uğur et al. [29], only best results of *DCS* are compared with *GA*. In this paper, results that indicated in Table 2 found by the proposed *ACO* are average results of iterations. In case study #2, the calculated best tour distances by *ACO* were compared with the *DCS*'s best tour distances as shown in Table 4. In Table 4, when evolution number for 300 points is taken as 40, the tour length obtained with ACO is 42.7122 and with DCS is 45.2367. In all evaluation numbers for 100 points ACO results better than DCS. For each other set of points (150, 200, 250, 300, 350 and 400), the ACO method is more successful than the DCS method for each evolution number. When Fig. 4 and Table 4 are examined, it is seen that the tour lengths are within the range of [22, 52] for *ACO (MMAS)* and [25, 53] for *DCS*. When the results of *DCS* and those of ACO were compared, see another columns of Table 4, it was observed that the results of ACO algorithm were successful than *DCS* for *sphericalTSP*.

The optimum route determined by *SphereTSP* for *N* = 100, 250 and 400 points is shown in Fig. 5 and Fig. 6. For Fig. 5 and Fig. 6 all points and route are viewed simultaneously with a transparent mode and solid view, respectively.

5. Conclusion

Spherical geometry shows differences from Euclidean geometry. In planar geometry, the shortest distance is given by a straight line, while in spherical geometry is formed by large circles. That is, distance between two points is traveled through a curved line on the surface of the sphere in place of a straight line. Where, the angular distance is in consideration. The contribution of this study that it is suggests a *ACO (MMAS)* algorithm giving reliable results for the solution of spherical *TSP*.

ACO (MMAS) has been successful in the spherical *TSP*, which can be used effectively with optimal results in the existing planar *TSP* solution. When the results of the proposed method and those of spherical *TSP* application through *GA* given by Uğur et al. [37] are compared, it can be seen that the *ACO* is more successful than *GA* for spherical *TSP*. When the best results for the spherical *TSP* of the proposed *ACO* method and *DCS* method given by Ouyang et al. [29] are compared, *ACO* is more successful than *DCS* in all best results of spherical TSP.

In the future studies, as a suggestion, other heuristic methods used in the *TSP* solution, such as Particle Swarm Optimization (*PSO*), can be tested for spherical *TSP* solution. Meanwhile, spherical *TSP* problems can be studied by hybrid utilization of heuristic methods. It can be predicted that with the increasing algorithm evolution and iteration, the optimum results obtained for unit sphere can be improved further. By changing the value of *ACO* constants, much better optimum results can be obtained.

The application of *TSP* for spherical conditions and the proposed method are important for the planning of the motions on the surface of the world. For vehicles traveling to different points on the surface of the world due to variety of reasons such as transportation, this method can be used to optimize the cost-time problem.

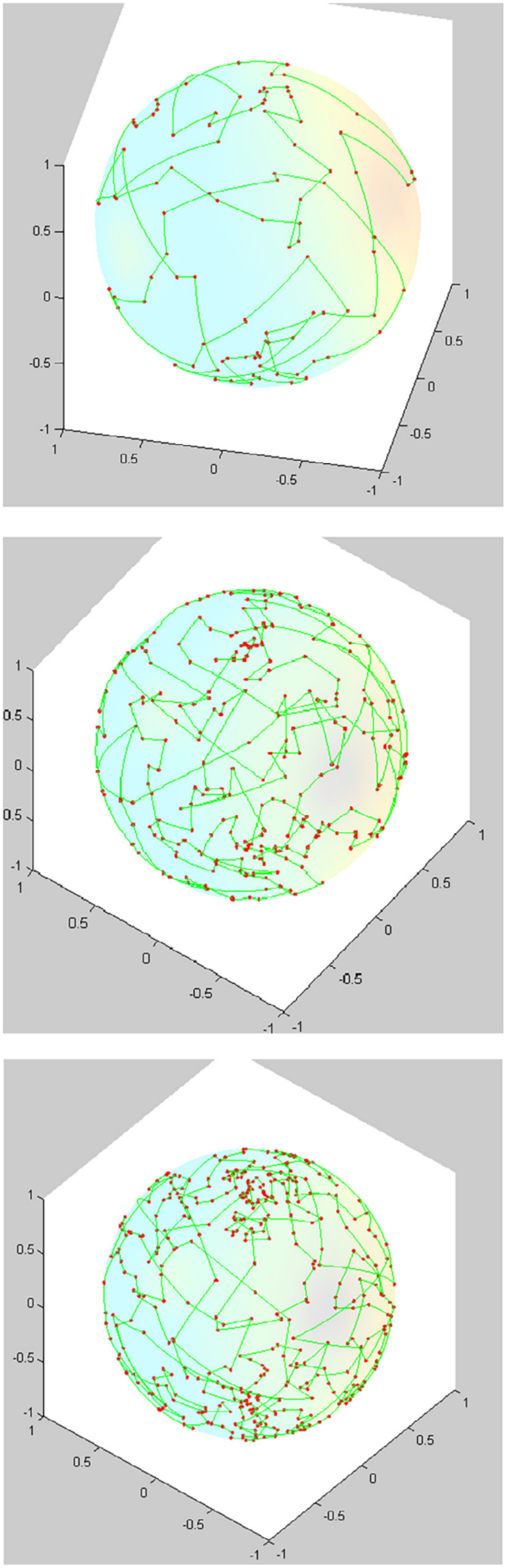

Fig. 5. The transparent view of the shortest routes obtained for 100, 250 and 400 points randomly placed on the sphere.

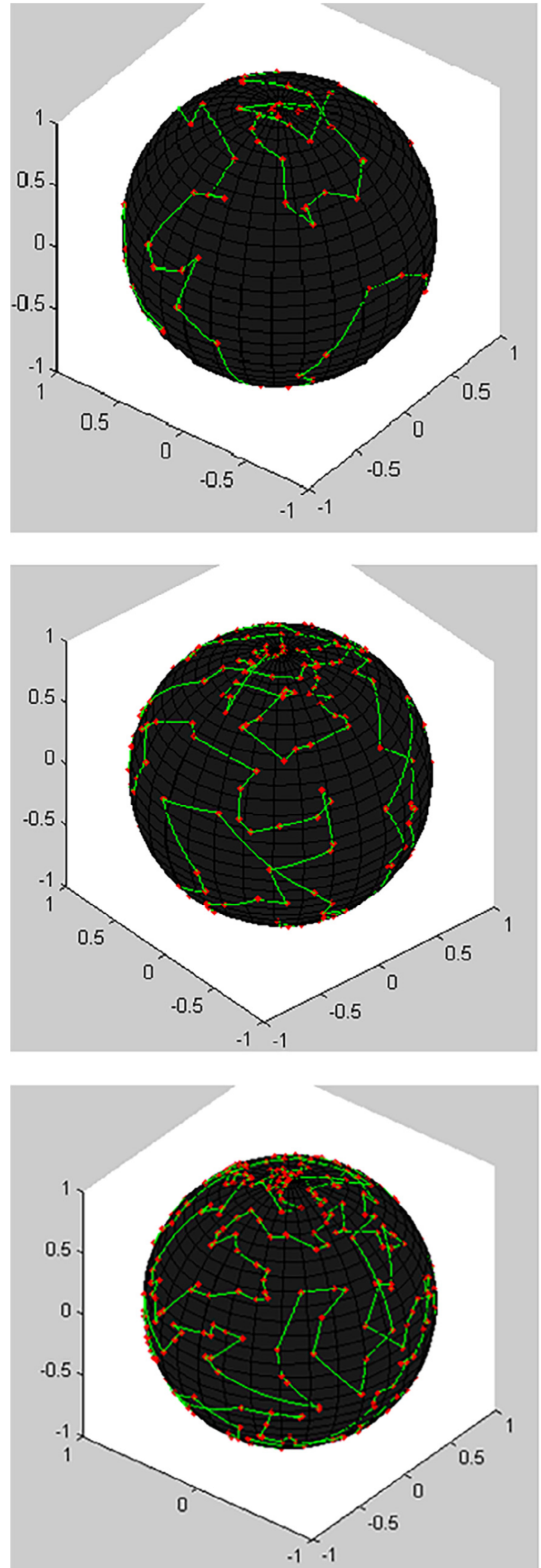

Fig. 6. The solid view of the shortest routes obtained for 100, 250 and 400 points randomly placed on the sphere.

Proposed method is beneficial to understand the behavior of the ant (agent) present on each spherical object in the real world. Meanwhile the use of *ACO*, metaheuristic methods and hybrid approaches for optimization problems of 3D shapes including sphere could be the source of inspiration for different studies.

References

[1] S. Chen, C. Chien, Parallelized genetic ant colony systems for solving the traveling salesman problem, Expert Syst. Appl. 38 (2011) 3873–3883.
[2] A. Colorni, M. Dorigo, V. Maniezzo, Distributed Optimization by Ant Colonies, Elsevier Publishing, Amsterdam, 1991, pp. 134–142.
[3] A. Colorni, M. Dorigo, V. Maniezzo, An investigation of some properties of an ant algorithm, North-Holland, Amsterdam, 1992, pp. 509–520.
[4] G. Dantzig, R. Fulkerson, S. Johnson, Solution of a Large-Scale Traveling Salesman Problem, J. Oper. Res. Soc. 2 (1954) 393–410.
[5] M. Dorigo, Optimization, Learning and Natural Algorithms, Politecnico di Milano, Italy, 1992. Ph.D. thesis.
[6] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern.-Part B Cybern. 26–1 (1996) 29–41.
[7] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Trans. Evolut. Comput. 1 (1) (1997) 53–66.
[8] M. Dorigo, L.M. Gambardella, Ant colonies for the travelling salesman problem, BioSystems 43 (1997) 73–81.
[9] H. Eldem, E. Ülker, The Application of Particle Swarm Optimization In The Solution Of 3D Traveling Salesman Problem On A Sphere. Akademik Bilişim'14 – XVI. Akademik Bilişim Konferansı Bildirileri, 2014, pp. 461–469 (in Turkish Language).
[10] H. Eldem, E. Ülker, Application of ant colony optimization for the solution of 3 dimensional cuboid structures, J. Comput. Commun. 2 (2014) 99–107.
[11] H. Eldem, E. Ülker, Optimizing The Tour On 3D Cuboid Structures With Particle Swarm Optimization Method. 3rd International Symposium on Innovative Technologies in Engineering and Science. Universidad Politecnica de Valencia, 2014, pp. 1607–1617.
[12] A.E. Ezugwu, A.O. Adewumi, M.E. Frîncu, Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem, Expert Syst. Appl. 77 (2017) 189–210.
[13] L.M. Gambardella, M. Dorigo, Solving Symmetric and Asymmetric TSPs by Ant Colonies. Proceedings of the Int. Conf. on Evolutionary Computation. Nagoya, Japan, 1996, pp. 622–627.
[14] F. Glover, Tabu search – Part I, ORSA J. Comput. 1 (3) (1989) 190–206.
[15] F. Glover, Tabu search – Part II, ORSA J. Comput. 2 (1) (1990) 4–32.
[16] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989. Reading.
[17] D. Hearn, M.P. Baker, Computer Graphics C Version, second ed., Prentice Hall, 1996.
[18] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
[19] J.J. Hopfield, D.W. Tank, Neural computation of decisions in optimization problems, Biol. Cybern. 52 (1985) 141–152.
[20] D.S. Johnson, L.A. McGeoch, The traveling salesman problem: A case study in local optimization, in: E.H.L. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, John Wiley & Sons, New York, 1997. 215–310.
[21] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[22] T. Kohonen, Self-Organizing Maps, Springer, Berlin, 1995.
[23] G. Laporte, The vehicle routing problem: an overview of exact and approximate algorithms, Eur. J. Oper. Res. 59 (1992) 345–358.
[24] Z.J. Lee, A hybrid algorithm applied to travelling salesman problem, Networking, Sensing and Control, Proceedings of the IEEE International Conference, 2004, pp. 237–242.
[25] Y. Lin, Z. Bian, X. Liu, Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing – tabu search algorithm to solve the symmetrical traveling salesman problem, Appl. Soft Comput. 49 (2016) 937–952.
[26] S. Maity, A. Roy, M. Maiti, An imprecise multi-objective genetic algorithm for uncertain constrained multi-objective solid travelling salesman problem, Expert Syst. Appl. 46 (2016) 196–223.
[27] Y. Marinakis, A. Migdalas, P.M. Pardalos, A hybrid genetic-GRASP algorithm using lagrangean relaxation for the traveling salesman problem, J. Comb. Optim. 10 (4) (2005) 311–326.
[28] T.A.S. Masutti, L.N. Castro, Neuro-immune approach to solve routing problems, Neurocomputing 72 (2009) 2189–2197.
[29] X. Ouyang, Y. Zhou, Q. Luo, H. Chen, A novel discrete cuckoo search algorithm for spherical traveling salesman problem, Appl. Math. Inf. Sci. 7 (2) (2013) 777–784.
[30] K. Shinozawa, T. Uchiyama, K. Shimohara, An approach for solving dynamic TSPs using neural networks Neural Networks, Proceedings of the IEEE International Joint Conference. 3 (1991) 2450–2454.
[31] S. Shoubao, C. Xibin, Jumping PSO with expanding neighborhood search for TSP on a cuboid, Chin. J. Electron. 22 (1) (2013) 202–208.
[32] T. Stützle, H. Hoos, MAX-MIN ant system, Future Gener. Comput. Syst. 16 (8) (2000) 889–914.
[33] T. Stützle, H. Hoos, Improvements on the ant system: introducing the MAX-MIN ant system, Artif. Neural Networks Genet. Algorithms (1998) 245–249.
[34] C. Tsai, C. Tseng, A new hybrid heuristic approach for solving large traveling salesman problem, Inf. Sci. 166 (2004) 67–81.
[35] Y. Tsujimura, M. Gen, Entropy-based genetic algorithm for solving TSP, Knowledge-Based Intell. Electron. Syst. 2 (1998) 285–290.
[36] A. Uğur, Path planning on a cuboid using genetic algorithms, Inf. Sci. 178 (2008) 3275–3287.
[37] A. Uğur, S. Korukoğlu, A. Calıskan, M. Cinsdikici, A. Alp, Genetic algorithm based solution for Tsp on a sphere, Math. Comput. Appl. 14 (3) (2009) 219–228.
[38] Y. Wang, The hybrid genetic algorithm with two local optimization strategies for traveling salesman problem, Comput. Ind. Eng. 70 (2014) 124–133.
[39] C.M. White, G.G. Yen, A hybrid evolutionary algorithm for traveling salesman problem, Congress on Evolutionary Computation (CEC2004). 2, 2004, 1473–1478.