

# Küre Üzerinde 3 Boyutlu Gezgin Satıcı Problemi Çözümünde Parçacık Sürü Optimizasyonu Uygulaması

Hüseyin Eldem<sup>1</sup>, Erkan Ülker<sup>2</sup>

<sup>1</sup> Karamanoğlu Mehmetbey Üniversitesi, Bilgisayar Teknolojileri Bölümü, Karaman

<sup>2</sup> Selçuk Üniversitesi, Bilgisayar Mühendisliği Bölümü, Konya

[heldem@kmu.edu.tr](mailto:heldem@kmu.edu.tr) , [eulker@selcuk.edu.tr](mailto:eulker@selcuk.edu.tr)

**Özet:** Gezgin Satıcı Problemi (GSP) gerçek dünya problemi olup, verilen N düğüm (şehir) için, her düğüme bir kez uğramak şartıyla, tekrar başlangıç düğümüne geri dönen en kısa (en az maliyetli) rotayı bulma problemidir. Günümüzde problemi çözmek için bir çok optimizasyon algoritması kullanılmıştır. Bunların başında meta-sezgisel algoritmalar gelmektedir. Bu çalışmamızda bir küre yüzeyine rastgele yerleştirilen yedi farklı sayıdaki düğüm kümelerinden oluşan öklitsel TSP nin çözümü için meta-sezgisel yöntemlerden biri olan parçacık sürü optimizasyonu (PSO) yöntemi kullanılmıştır. Yöntemin performansı testlerle gösterilmiştir.

**Anahtar Sözcükler:** Öklitsel TSP, Parçacık Sürü Optimizasyonu, Meta-sezgisel, Küresel Geometri

## The Application of Particle Swarm Optimization In The Solution Of 3D Traveling Salesman Problem On A Sphere

**Abstract:** Traveling Salesman Problem (TSP) is a real-world problem involving the determination of the shortest route (the lowest cost) to travel N nodes (cities) such that all nodes will be visited once at least and the route will end at the start node. Nowadays, many optimization algorithms have been used to solve this problem. The primary of these are metaheuristic algorithms. In this study, one of the metaheuristic methods, Particle Swarm Optimization (PSO) method, was used to solve Euclidian TSP that is made out of seven different set of points, randomly located on a surface of a sphere. The performance of the method was demonstrated by different experiments.

**Keywords:** Euclidean TSP, Particle Swarm Optimization, Meta-heuristic, Spherical Geometry

## 1. Giriş

Gezgin satıcı Problemi (*TSP*) minimum maliyetle şehirleri seyahat ederek başlangıç noktasına geri dönmesi gereken bir satış personelinin çözmesi gereken bir problemidir. Problemde optimize edilecek maliyet, zaman, para ya da uzaklık değerlerinden birini temsil edebilir. Bu problem bilgisayar bilimlerinde verilerin modellenmesinde kullanılan ve graf teorisi kapsamında ele alınan Hamilton döngüsü olarak ta ifade edilebilir. Ayrık ve kombinyonel optimizasyon problemi alanında değerlendirilen *TSP*, kendi yapısına benzer bir çok popüler problemler alanında kapsamlı olarak çalışılmış bir problemidir. Simetrik *TSP* de  $i$  ve  $j$  şehirleri için iki şehir arasındaki uzaklık eşitken ( $d_{ij} = d_{ji}$ ) asimetric *TSP* de uzaklıkların eşitliği söz konusu değildir. Öklitsel *TSP*, maliyet fonksiyonunun üçgen eşitsizliğini sağladığı bazı  $m$  değerleri için şehirlerin  $R^m$  uzayında konumlandığı Simetrik *TSP* nin özel bir durumudur.  $\ell_m$  normunda ( $m=2$  ise öklitsel norm denir.)

$m \geq 1$  için  $(x_1, \dots, x_m), (y_1, \dots, y_m) \in R^m$  iken  $\ell_m$  normu  $(\sum_{i=1}^d |x_i - y_i|^m)^{1/m}$  olarak tanımlanır. İki boyutlu öklitsel *TSP* literatürde çalışılmış oldukça popüler bir konudur [1].

*TSP* nin çözümü için pek çok yöntem geliştirilmiştir. Bu yöntemler optimum sonuca ulaşma bakımından kesin çözüm yöntemleri ve sezgisel yöntemler olarak ikiye ayrılır. Aşamalı Geliştirme (Iterative Improvement), Dallan ve Sınırla (Branch-and-Bound), Dallan ve Kes (Branch-and-Cut) *TSP* için kesin çözüm yöntemleridir [2-3]. Belirli bir sürede en iyi çözüme yakın çözümler üreten sezgisel yöntemlerden tavlama benzetimi [4], genetik algoritmalar [5-7], tabu arama [8-9], yapay sinir ağları [10-12] ve karınca sistemi [13-18] tabanlı birçok algoritma tasarlanmıştır. Bazı araştırmacılar hibrid evrimsel algoritmaları ile daha iyi *TSP* sonuçları için çalışmalar yapmışlardır [18-21]. [22-24] de ise küre ve küboid gibi temel 3D geometrik şekiller üzerindeki

noktalar için *TSP* uygulamaları yapılmıştır. [22]' te *TSP* nin küboid geometrik şekil için, [23] te ise 3D şekillerden bir küre üzerinde *TSP* nin çözümü genetik algoritmalar yardımıyla yapılarak bir algoritma önerilmiştir.

Optimizasyon problemlerinin çözümünde kullanılan meta-sezgisel yöntemlerden birisi olan Parçacık Sürü Optimizasyonu (*PSO*), 1995'te Dr. Eberhart ve Dr. Kennedy tarafından geliştirilmiş popülasyon temelli sürü zekası kategorisinde değerlendirilen sezgisel bir optimizasyon tekniğidir. İnsanların konuşarak ya da bir başka şekilde bilgi paylaşımı yapmaları gibi, kuşlar ve balıklarında bilgi paylaşması sosyal bir zekâya işaret etmektedir. *PSO*, kuşların yön tayininde birbirlerinden faydalanmaları ve balık sürülerinin sosyal davranışlarından esinlenilerek geliştirilmiştir. *PSO*' da popülasyonu oluşturan bireyler parçacık olarak adlandırılır, her bir parçanın durum uzayında hareket ettiği varsayılır ve her parça potansiyel çözümü taşır. Her parça en iyi durumu hatırlayabilir ve parçacıklar kendi arasında bilgi alışverişinde bulunabilirler.

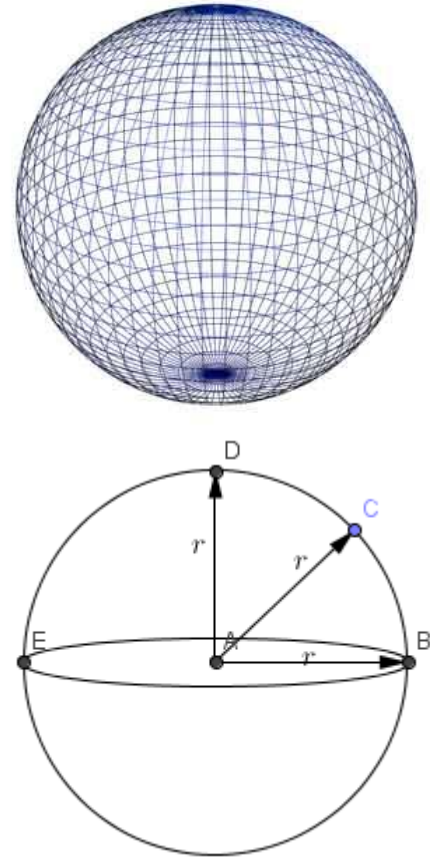
Bu çalışmada ise 3D şekillerden küre üzerindeki noktalar için *TSP*, *PSO* algoritması ile çözülmüştür. Yazarın bilgisine göre, 3D' de *TSP* yi *PSO* ile çözen bir çalışma yoktur. Mevcut *TSP* problemlerinde noktaların koordinatları ya da birbirine uzaklıkları bilinmektedir. Bu problem bütün noktaların bir küre üzerinde olması ve noktalar arası geçişin küre yüzeyi üzerinden yapılması yönünden mevcut *TSP* problemlerinden farklıdır. Küre üzerindeki nokta tanımlamaları ve noktaları arası uzaklıkların bulunması ardından problemin *PSO* ya uyarlanması bildirinin devamında açıklanmıştır. Bu çalışma ile küre şekline benzerliği ile ilk akla gelen dünya yüzeyi üzerinde hareket eden jet, uçak gibi araçların belirli nedenlerle yaptıkları uçuşların optimize edilecek her bir kriteri için bu yöntemden faydalanılabilir.

Bu çalışmada, *PSO* kullanılarak önerilen metodun performansı test edilmiş ve literatürde seçilen *GA* yöntemi ile kıyaslamaları sunulmuştur. Önce birim küre üzerindeki nokta tanımlamaları ve noktaları arası uzaklıkların bulunması açıklanmıştır. Sonra bildirinin devamında problemin *PSO* ya uyarlanması açıklanmıştır.

## 2. Küre Matematiği/Temeli/Temsili/Gösterimi

Küre, üç boyutlu bir cisim olup, uzayda sabit bir noktadan eşit uzaklıktaki noktaların oluşturduğu bir cisimdir. Küre merkezinden üç boyutta (x,y,z) eşit uzaklıktaki (yarıçap r) yayılmış her bir nokta kürenin yüzeyinde yer almaktadır. Başka bir deyişle orjin noktasından x-y koordinatlarında eşit uzaklıkta çizilen bir yayın z eksenine etrafında 360 derece döndürülmesi ile küre şekli elde edilir. Uzaydaki (x,y,z) koordinatları ile kürenin yarıçapı arasındaki ilişki şu şekilde formülize edilir: [25]

$$r = \sqrt{x^2 + y^2 + z^2} \quad (1)$$



Şekil 1. Küresel düzlem ve yarıçap

Şekil 1. deki kürenin merkezi A noktasının küre yüzeyindeki B, C, D ve E noktalarına olan uzaklık yarıçaptır (r). Yüzeydeki her bir noktanın x, y, z değerleri olup, bu değerler (1) denklemini her zaman sağlar.

Küre üzerinde bir problemin çözümü düşünüldüğünde akla gelen ilk örnek dünyanın şeklinin küreye olan benzerliğidir. Küre merkezinden geçen ve küre ile sınırlı daire büyük dairedir. Dünya üzerindeki büyük daireye dünyanın ekvatorudur. Büyük daire, kendisinin alt kesiti boyunca küre üzerindeki iki nokta arasındaki en kısa uzaklığın farkına varıldığında daha önemli hale gelir. Bu en kısa yola, jeodezik (iki nokta arasındaki en kısa yol) denir. Noktaları arasındaki mesafeyi en aza indirgeyen eğriler herhangi bir yüzey üzerindeki jeodezikler olarak adlandırılır. [23]

### 2.1. Küre Üzerindeki Noktaların Matematiksel Gösterimi

Öklidyen eğriler tek boyutlu nesnelere olup, üç-boyutlu eğri yolu boyunca tek bir u parametresi ile tarif edilebilir. Diğer bir deyişle, bu, u parametresi cinsinden üç kartezyen koordinatları ifade edebilir. Eğri üzerindeki

herhangi bir nokta, belirli bir kartezyen referans çerçevesine göre aşağıdaki vektör noktası fonksiyonu ile gösterilir. [26]

$$P(u) = (x(u), y(u), z(u)) \quad (2)$$

Genellikle koordinat denklemleri u parametresinin 0 ila 1 aralığında tanımlandığı şekilde kurulur. Örneğin koordinat orijin merkezi ile xy düzlemindeki bir daire aşağıdaki parametrik formda tanımlanır: [26]

$$\begin{aligned} x(u) &= r \cdot \cos(2\pi u) & y(u) &= r \cdot \sin(2\pi u) \\ z(u) &= 0, \quad 0 \leq u \leq 1 \end{aligned} \quad (3)$$

Daireleri ve dairesel yayları tanımlamak için başka parametrik formlarda mümkündür. Eğimli öklitsel yüzeyler iki boyutlu nesnelere olup, yüzey üzerindeki pozisyonları u ve v parametreleri ile tanımlanır. Yüzey üzerindeki bir koordinat konumu, kartezyen koordinat değerleri x, y, z için u ve v parametrelerinin fonksiyonu cinsinden parametrik vektör fonksiyonu ile temsil edilir: [26]

$$P(u) = (x(u, v), \quad y(u, v), \quad z(u, v)) \quad (4)$$

Kartezyen koordinat değerlerinin her biri, 0 ila 1 arasında değişebilen u ve v olan iki yüzey parametresinin fonksiyonudur. Koordinatlar, merkezi orijinde olan bir küresel yüzey için, r kürenin yarıçapı iken şu denklemlerle ifade edilir: [26]

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

u parametresi yüzey üzerindeki sabit enlem çizgilerini, v parametresi ise sabit boylam çizgilerini açıklamaktadır [26]. Örnek olarak, u ve v parametrelerinin farklı değerleri için x, y, z koordinatları (5) denklemine göre hesaplanarak Tablo 1 de verilmiştir. (küre yarıçapı r = 1 alınmıştır. r oransal artışında x, y, z değerleri de aynı oranda artırılmalıdır. )

**Tablo 1** u, v parametrelerinin farklı değerleri için küresel yüzeydeki koordinatlar

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

## 2.2. Küre Üzerindeki İki Nokta Arasındaki En Kısa Mesafenin Bulunması

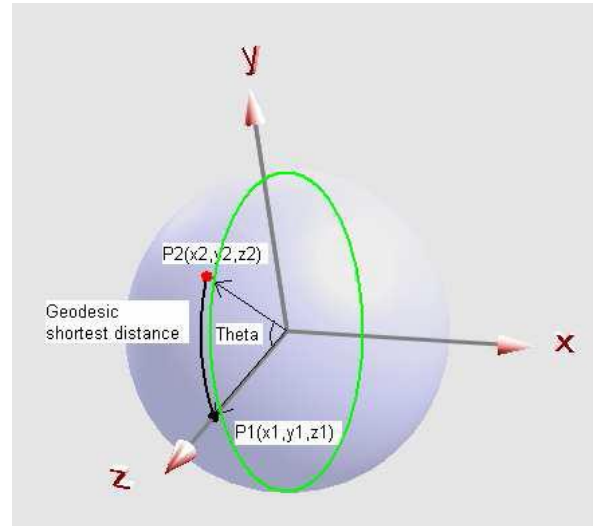
Küresel bir yüzey üzerindeki iki nokta (p1, p2) arasındaki en kısa mesafe büyük dairenin yayı boyuncadır. Yani,  $\vec{V1}$  ve  $\vec{V2}$  iki vektör arasındaki radyan cinsinden teta ( $\theta$ ) açı değeri kullanılabilir. İki vektörün skaler çarpımı: [23]

$$\vec{V1} \cdot \vec{V2} = |\vec{V1}| |\vec{V2}| \cos \theta \quad (\theta \text{ iki vektör yönünün arasındaki küçük olan açı})(6)$$

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

$\vec{V1}$  ve  $\vec{V2}$  vektörlerinin büyüklükleri birim kürenin yüzeyi üzerindeki noktaları için 1'dir. Böylece en kısa yol formülü şu şekildedir: [23]

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



**Şekil 2.** Geodesic: bir küresel yüzey üzerindeki iki nokta arasındaki en kısa mesafedir. [23]

Problem, Öklid TSP den farklıdır. Çünkü (p<sub>i</sub> ve p<sub>j</sub>) 2 noktası arasındaki en kısa mesafe, 3D Öklid TSP de 3D Öklid uzaklığı olarak hesaplanırken, problemimizde yay uzunluğu kullanılarak hesaplanır. Küre üzerindeki noktaların uzaklık matrisi simetrik TSP ile aynıdır. Uzaklık (p<sub>i</sub> : p<sub>j</sub>) = Uzaklık (p<sub>j</sub> : p<sub>i</sub>)

### 3. PSO Kullanarak Birim Küre Yüzey Üzerinde TSP Çözümü

Küre üzerinde uygulanacak *TSP*, normal *TSP* problemlerinden farklılık gösterir. Satış elemanı sadece küre yüzeyinde konumlanmış noktaları gezebilir. Bu problemdeki tek farklı kısıtlama noktaların kürenin içinde değil yüzeyinde olmasıdır.

Çözülmesi gereken problem, satış personelinin (parçacık) küre yüzeyinde koordinatları bilinen belirli sayıdaki nokta için ( $N$ ), standart *TSP* deki gibi bütün noktaları gezerek başlangıç noktasına geri dönüş işleminin minimum tur uzunluğu olarak ifade edilebilir.

Her bir nokta ikilisi arasındaki mesafeler (8) denklemine göre hesaplandıktan sonra problemin çözümü standart *TSP* ile aynı olmaktadır. Bu aşamadan sonra giriş bölümünde literatürde belirtilen *TSP* çözümü yöntemlerinin her biri ile problem çözümü test edilebilir. Bu bildiride *PSO* kullanılarak belirli sayıdaki nokta kümeleri için çözümler üretilmiştir.

#### PSO nun probleme uyarlanmış genel yapısı:

*For* her parçacık için

*Parçacığı başlangıç konumuna getir*

**End**

**Do**

*For* her parçacık için

*Uygunluk değerini (tur uzunluğunu) hesapla*

**Eğer** uygunluk değeri (tur uzunluğu) *pbest* ten

daha iyi ise,

*Şimdiki tur uzunluğunu yeni pbest olarak*

ayarla

**End**

*Tüm parçacıkların bulunduğu pbest değerlerinin en iyisini, tüm parçacıkların gbest'i olarak ayarla (en kısa tur uzunluğu)*

*For* her parçacık için

*Parçacıkların hızlarını hesapla*

*Parçacıkların konumlarını güncelle*

**End**

**While** maksimum iterasyon sayısına veya minimum hata koşulu sağlanana kadar devam et.

Bu genel yapıya göre, öncelikle *PSO* algoritmasının başlangıç bireyleri rastgele olarak oluşturulur. (8) denklemine göre noktaların her birinin bütün noktalara uzaklıklarının tutulduğu uzaklık matrisi oluşturulur. Uzaklık matrisi ile her bir bireyin uygunluk değerleri hesaplanır. Bireylerden minimum tur uzunluğuna sahip birey global en iyi olarak belirlenir. Bu iki en iyi değer bulunduktan sonra; parçacık, hızı ve konumu sırasıyla aşağıdaki (9) ve (10) denklemlerine göre güncellenir. Sonraki her bir iterasyonda her bir birey bir öncekinden daha kısa bir tur uzunluğu elde etmişse bireylerin en iyisi olarak bulunan yeni değerle değiştirilir.

Bireylerin yenilenmesinin ardından tekrar en iyi birey (*gbest*) belirlenir.

$$v_i^{k+1} = w.v_i^k + c_1.rand_1^k.(pbest_i^k - x_i^k) + c_2.rand_2^k.(gbest^k - x_i^k) \quad (9)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (10)$$

Burada *rand* (0,1) arasında üretilen rastgele bir değeri,  $i$  parçacık numarasını,  $k$  ise iterasyon sayısını gösterir.  $c_1$  ve  $c_2$  öğrenme faktörleridir. Bunlar parçacıkları *pbest* (personal best) ve *gbest* (global best) konumlarına doğru yönlendiren sabitlerdir.  $c_1$  parçacığın kendi tecrübelerine göre (kişisel hafıza),  $c_2$  ise sürüdeki diğer parçacıkların tecrübelerine göre (sosyal hafıza) hareketi yönlendirir. Düşük değerler seçilmesi parçacıkların hedef bölgeye doğru çekilmeden önce, bu bölgeden uzak yerlerde dolaşmalarına imkân verir. Ancak hedefe ulaşma süresi uzayabilir. Diğer yandan, yüksek değerler seçilmesi, hedefe ulaşmayı hızlandırırken, beklenmedik hareketlerin oluşmasına ve hedef bölgenin es geçilmesine sebep olabilir. Genellikle  $c_1=c_2=2$  olarak almanın iyi sonuçlar verdiği belirtilmiştir.  $w$  değeri ise her bir iterasyonda değeri düşürülerek bir önceki hız bilgisinin etkisini azaltmaya yarayan atalet ağırlığı (inertia weight) dir.

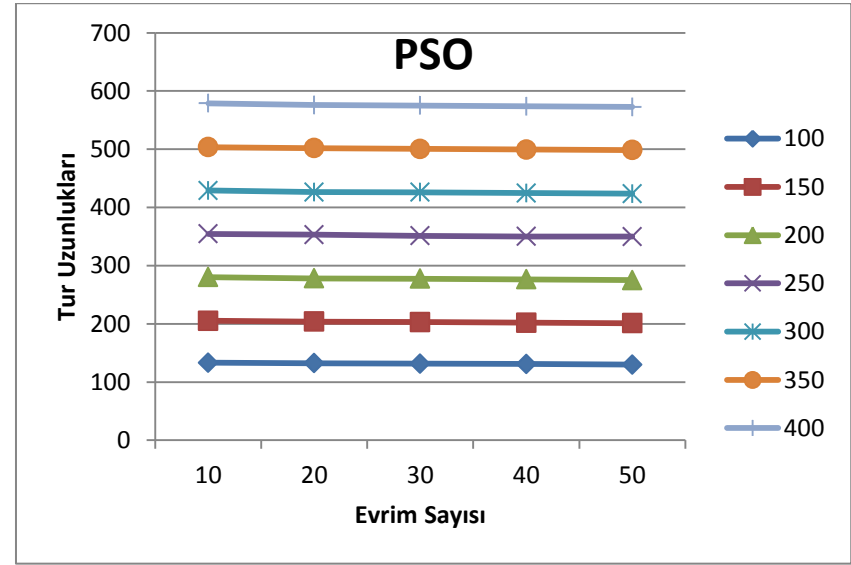
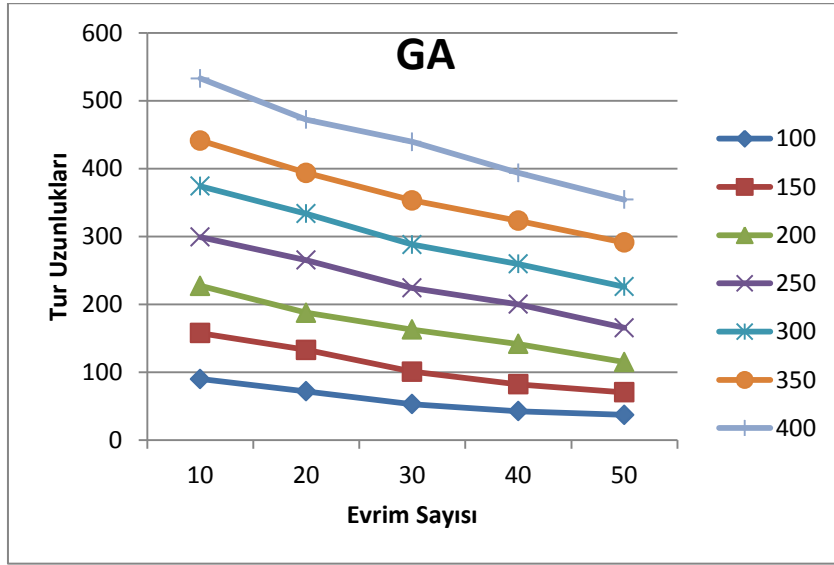
### 4. Deneysel Sonuçlar

Simülasyon sonuçları birim kürede  $N = 100, 150, 200, 250, 300, 350, 400$  nokta için elde edilmiştir. Simülasyonlar,  $N$ . nin her bir değeri için 100 kez tekrarlanmıştır. Her denemede rastgele bir nokta kümesi oluşturulmuştur. Birim küre üzerinde sonuçları genellemek için önceden tanımlanmış noktalar kümesinin kullanımı yerine bu yaklaşım tercih edilmiştir. Sonuçlar Matlab R2010a programlama dili kullanılarak gerçekleştirilmiştir. Önerilen yöntemin performansı Aybars tarafından önerilen *GA* sonuçları ile karşılaştırılmıştır.

Optimum tur uzunluğunu temsil eden sonuçlar, evrim sayısında (10, 20, 30, 40, 50 evrim) elde edilmiştir. Tüm deneyler için,  $c_1$  ve  $c_2$  öğrenme faktörleri 2 olarak alınmıştır. Değerler, bir birim kürenin yüzeyi üzerindeki yollar için elde edilmiştir. Belirtilen nokta sayıları için 100 iterasyon sonrası elde edilen ortalama tur uzunlukları, hem *GA* hem de önerilen *PSO* yaklaşımları için Tablo 2 de sunulmuştur. Şekil 3 te ise elde edilen sonuçlar diyagram halinde verilmiştir.

**Tablo 2.** Bir kürenin yüzeyindeki  $N = 100, 150, 200, 250, 300, 350, 400$  nokta için GA [23] ve PSO ile hesaplanan ortalama küresel TSP tur uzunlukları.

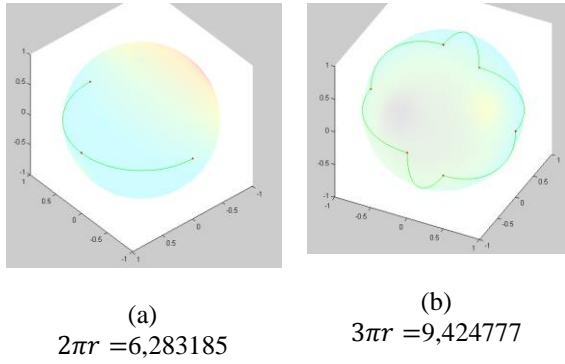
Evrimsel Sayısı	Nokta Sayıları													
	100		150		200		250		300		350		400	
	GA	PSO	GA	PSO	GA	PSO	GA	PSO	GA	PSO	GA	PSO	GA	PSO
10	90,1194	<b>133,1503</b>	157,6443	<b>205,3276</b>	227,2337	<b>280,1300</b>	299,1799	<b>354,6136</b>	374,1841	<b>428,9046</b>	441,3127	<b>503,5720</b>	532,6288	<b>578,7485</b>
20	72,0467	<b>132,2149</b>	132,9872	<b>203,9083</b>	187,7657	<b>278,0525</b>	265,2366	<b>352,9967</b>	333,5504	<b>426,2686</b>	393,6967	<b>501,8394</b>	472,2036	<b>575,7905</b>
30	53,0306	<b>131,6228</b>	100,9841	<b>202,8926</b>	162,8873	<b>277,5403</b>	224,2341	<b>351,1674</b>	288,0024	<b>425,9008</b>	353,1749	<b>500,4234</b>	439,496	<b>574,9534</b>
40	42,6922	<b>130,8192</b>	82,2588	<b>201,8222</b>	141,7211	<b>276,3966</b>	200,0556	<b>350,2219</b>	259,6622	<b>424,7062</b>	323,2888	<b>499,4872</b>	393,7457	<b>573,5802</b>
50	37,1942	<b>129,7220</b>	70,5737	<b>200,9317</b>	115,1155	<b>275,0707</b>	165,5674	<b>349,7115</b>	226,1183	<b>423,4063</b>	291,1789	<b>498,6680</b>	354,375	<b>572,6015</b>



**Şekil 3.** GA [23] ve PSO Çözümlerinde Birim Kürede Farklı Sayıdaki Nokta Sayıları İçin Ortalama Tur Uzunlukları

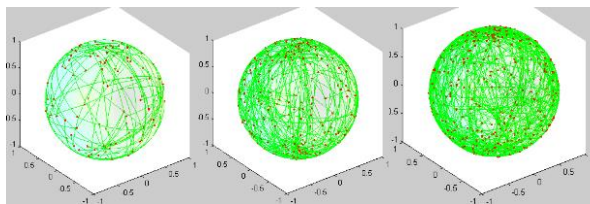
İki nokta ziyaret edilecek ise ve bunlar birim küre üzerinde karşıt noktaları ise Öklid *TSP* tur uzunluğu  $4*r=4$  dür. (başlangıç noktasına dönecek şekilde) Küresel *TSP* tur uzunluğu ise yaklaşık olarak  $2\pi r = 6,283185$  dir. (Şekil 3a).

Karşıt noktalar küre şekli üzerinde aralarında maksimum uzunluğun olduğu noktalardır. Belirtilen karşıt noktalardan geçen sonsuz sayıda büyük daire söz konusudur. Bu sonsuz sayıdaki büyük daire ise küre şeklini oluşturmaktadır. Bir küre üzerinde karşıt olmayan iki nokta arasında ise en kısa yolu izleyen sadece bir büyük daire vardır. Rastgele oluşan noktalar kümesinde karşıt nokta sayısı ne kadar fazla olursa minimum tur uzunluğu maliyeti orantılı şekilde artış gösterir.  $x, y, z$  koordinatlarının karşıt noktalarına ait nokta kümesinin tur uzunluğu gösterimi şekil 4b deki gibidir.

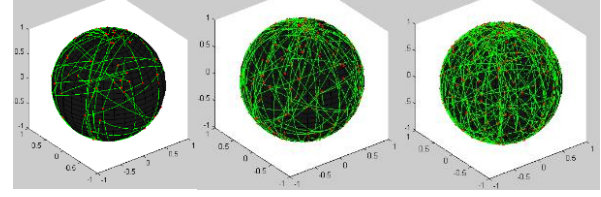


**Şekil 4.** Karşıt noktalar, Küresel *TSP* için birim küre üzerindeki en büyük mesafe

100, 250 ve 400 nokta için *SphereTSP* tarafından bulunan optimum rota Şekil 5 'te gösterilmiştir. Bütün noktalar ve güzergâh saydam modu ile aynı anda görülür. Şekil 6' da ise solid bir görüntü sunulmuştur.



**Şekil 5.** Küre Üzerinde Rastgele Yerleştirilmiş 100, 250 ve 400 Nokta İçin Elde Edilen Minimum Turların saydam görünümü



**Şekil 6.** Küre Üzerinde Rastgele Yerleştirilmiş 100, 250 ve 400 Nokta İçin Elde Edilen Minimum Turların katı görünümü

Algoritma tur sayısı ve evrim sayısı artırıldığında birim kürede bulunan optimum sonuçların daha iyileşebileceği öngörülmektedir. *PSO* parametrelerinin değerleri değiştirilerek daha iyi sonuçlar elde edilebilir. Ayrıca *PSO* nun yanı sıra *TSP* çözümü için elverişli yöntemler kullanılarak sonuçlar iyileştirilebilir.

## 5. Sonuç ve Öneriler

*TSP* nin küreye uyarlanması ve önerilen yöntem, özellikle dünya yüzeyi üzerindeki hareket planlamaları için önem arz etmektedir. Dünya yüzeyi üzerinden hareket eden uçak vb. araçların taşımacılık gibi belirli nedenlerle koordinatların seyahatinde maliyet-süre gibi problemlerini optimize etmelerinde bu yöntemden faydalanılabilir. Gerçek dünyadaki küresel her nesne üzerindeki parçacık davranışlarını anlamak bu çalışma ile faydalı olacaktır. Ayrıca kürenin yanı sıra diğer 3D şekilleri üzerindeki optimizasyon problemlerinde *PSO*, diğer meta-sezgisel yöntemler ve hibrid yaklaşımların kullanılması farklı çalışmalara esin kaynağı oluşturacaktır.

Bu çalışmanın önemli katkılarından birisi, yeryüzünde yaygın olarak karşılaşılan küre şekli üzerinde *TSP* nin ciddi bir uygulaması olması, diğeri ise 3 boyutlu Öklitsel *TSP* den farklı olarak küresel *TSP* nin çözümü için basit ve iyi sonuçlar üreten bir *PSO* algoritmasını önermektir. Küresel geometri öklitsel geometriden farklılık göstermektedir. Düzlemsel geometride minimum mesafe, noktalar arasındaki düz bir çizgi ile ifade edilirken, küresel geometride minimum mesafe büyük dairelere dayanır. Yani iki nokta, düz bir çizgi yerine küre yüzeyi üzerinden ziyaret edilir. Burada açısız mesafe söz konusudur.

Gelecekteki çalışmalarda *TSP* çözümünde kullanılan diğer yöntemler (örneğin Karınca Kolonisi Optimizasyonu-*ACO*), küresel *TSP* çözümünde test edilebilir. Ayrıca *PSO* ile *ACO* vb. yöntemlerin hibrit olarak kullanılması ile küresel *TSP* problemleri ele alınabilir.

## 6. Kaynaklar

- [1] S. Arora, Polynomial time approximation schemes for Euclidean TSP and other geometric problems, Proc. 37th Ann. Symp. Foundations of Computer Sci., IEEE Computer Soc., 2–11, 1996.
- [2] Dantzig G., Fulkerson R., Johnson S., “Solution of a Large-Scale Travelling Salesman Problem”, Journal of Operations Research Society, c.2, sf. 393-410, 1954.
- [3] Laporte G., “The Vehicle Routing Problem: An overview of exact and approximate algorithms”, European Journal of Operational Research, c. 59, sf. 345-358, 1992.
- [4] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671-680, 1983.
- [5] Y. Tsujimura, M. Gen, Entropy-based genetic algorithm for solving TSP, Knowledge-Based Intelligent Electronic Systems 2, 285-290, 1998.
- [6] Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading 1989.
- [7] Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor 1975.
- [8] Glover, F.: Tabu Search - Part I. ORSA Journal on Computing 1 (3), 190-206, 1989.
- [9] Glover, F.: Tabu Search - Part II. ORSA Journal on Computing 2 (1), 4-32, 1990.
- [10] Hopfield, J.J. and Tank, D.W.: Neural Computation of Decisions in Optimization Problems. Biological Cybernetics 52, 141-152, 1985.
- [11] Kohonen, T.: Self-Organizing Maps. Springer, Berlin 1995.
- [12] K. Shinozawa, T. Uchiyama and K. Shimohara, An approach for solving dynamic TSPs using neural networks, Neural Networks, IEEE International Joint Conference 3, 2450-2454, 1991.
- [13] Colomi, A., Dorigo, M. and Maniezzo, V.: Distributed Optimization by Ant Colonies. In Proceedings of the European Conference on Artificial Life (ECAL'91, Paris, France), eds F. Varela and P. Bourguine. Elsevier Publishing, Amsterdam, pp. 134-142, 1991.
- [14] Colomi, A., Dorigo, M. and Maniezzo, V.: An investigation of some properties of an ant algorithm. In Proceedings of the Second Conference on Parallel Problem Solving from Nature (PPSN II, Brussels, Belgium), eds R. Maenner and B. Manderick. North-Holland, Amsterdam, pp. 509- 520, 1992.
- [15] L.M. Gambardella and M. Dorigo, Solving Symmetric and Asymmetric TSPs by Ant Colonies, International Conference on Evolutionary Computation, 622-627, 1996.
- [16] Dorigo, M. ve Gambardella, L. M., “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”, IEEE Transactions On Evolutionary Computation, Vol. 1, No. 1, 53-66, (1997a).
- [17] Dorigo, M. ve Gambardella, L. M., “Ant Colonies for the Travelling Salesman Problem”, BioSystems 43, 73–81, (1997b).
- [18] Z. J. Lee, A hybrid algorithm applied to travelling salesman problem, Networking, Sensing and Control, IEEE International Conference 1, 237-242, 2004.
- [19] C. M. White and G. G. Yen, A hybrid evolutionary algorithm for traveling salesman problem, Congress on Evolutionary Computation (CEC2004) 2, 1473- 1478, 2004.
- [20] Y. Marinakis, A. Migdalas and P. M. Pardalos, A Hybrid Genetic-GRASP Algorithm Using Lagrangean Relaxation for the Traveling Salesman Problem, J. Comb. Optim. 10(4), 311-326, 2005.
- [21] Tsai, C., Tsai, C. ve Tseng, C., “A New Hybrid Heuristic Approach for Solving Large Traveling Salesman Problem”, Information Sciences , No: 166, pp. 67–81, 2004.
- [22] A. Uğur, Path Planning On A Cuboid Using Genetic Algorithms, Information Sciences 178, 3275-3287, 2008.
- [23] A. Uğur, S. Korukoğlu, A. Caliskan, M. Cinsdikici, A. Alp, Genetic Algorithm Based Solution For Tsp On A Sphere, Mathematical and Computational Applications, Vol. 14, No. 3, pp. 219-228, 2009.
- [24] S. Shoubao and C. Xibin, Jumping PSO with Expanding Neighborhood Search for TSP on a Cuboid, Chinese Journal of Electronics, Vol.22, No.1, Jan. 2013.
- [25] Wikipedia, [http://en.wikipedia.org/wiki/Spherical\\_coordinate\\_system](http://en.wikipedia.org/wiki/Spherical_coordinate_system), 2013
- [26] D. Hearn and M.P. Baker, Computer Graphics C Version, Second Edition, Prentice Hall, 1996..