






## Kablosuz Sensör Ağlarında Konum Belirlemede Parçacık Sürü Optimizasyonu ve Yarasa Algoritması Yöntemlerinin Karşılaştırılması

### Comparison of Particle Swarm Optimization and Bat Algorithm Methods in Localization of Wireless Sensor Networks

Ahmet Sami Doğru , Burak Temel , Tolga Eren 

Kırıkkale Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, Kırıkkale, TÜRKİYE

Başvuru/Received: 28/11/2019

Kabul/Accepted: 20/12/2019

Son Versiyon/Final Version: 31/12/2019

#### Öz

Kablosuz sensör ağlarında konum belirleme birçok uygulamada her geçen gün önemli bir konu haline gelmektedir. Arama kurtarma, doğal afet, coğrafi izleme ve takip gibi uygulamalarda sensörlerin konumunu doğru ve hızlı bir şekilde tespit etmek çok önemlidir. Kablosuz sensör ağlarında (KSA), konum belirleme çok boyutlu bir optimizasyon problemi olarak ifade edilmektedir. KSA'ların karmaşıklığına göre optimizasyon yöntemlerinin performansları da değişmektedir. Bu çalışmada, Parçacık Sürü Optimizasyonu (PSO) ve Yarasa Algoritması (YA) metotları KSA'larda konum belirleme probleminin çözülmesi amacı ile karşılaştırılmıştır. Önerilen yöntemler, doğada yaşayan varlıkların hareketinden esinlenerek geliştirilmiş optimizasyon yöntemleridir. Bu çalışmada sezgisel optimizasyon metotlarından olan PSO ve YA, Matlab ortamında test sistemine uygulanmış olup sonuçlar değerlendirilmiştir.

#### Anahtar Kelimeler

"Parçacık sürü optimizasyonu, Yarasa algoritması, Kablosuz sensör ağlarında konum belirleme"

#### Abstract

Positioning in wireless sensor networks is becoming an important issue in many applications. It is very important to accurately and quickly determine the position of sensors in applications such as search and rescue, natural disaster, geographical monitoring and tracking. In wireless sensor networks (WSN), positioning is expressed as a multidimensional optimization problem. According to the complexity of WSN's, the performance of optimization methods also changes. In this article, Particle Swarm Optimization (PSO) and Bat Algorithm (BA) methods have been compared in solving the problem of positioning in WSN's. The proposed methods are optimization methods inspired by the movement of living beings in nature. In this study, one of the heuristic optimization methods, PSO and BA, was applied to the test system in Matlab environment and the results were evaluated.

#### Key Words

"Particle swarm optimization, Bat algorithm, Localization in wireless sensor networks"

## 1. Giriş

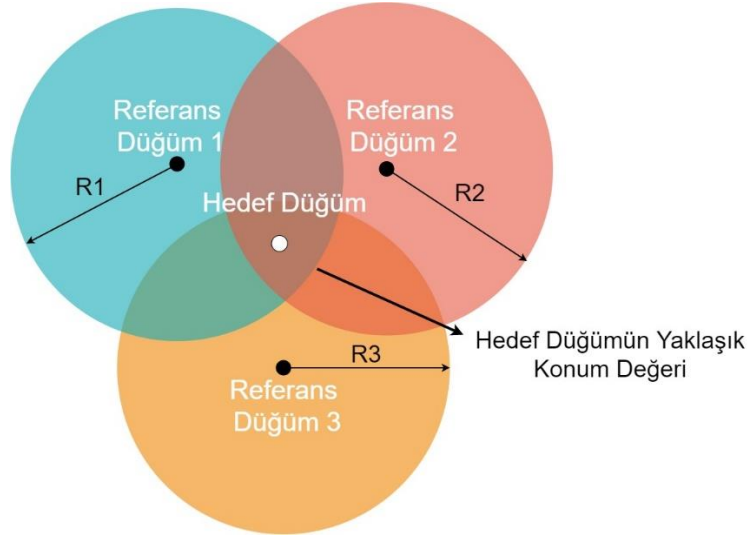
Günümüzde KSA’larda bulunan sensörlerin konum belirleme işlemi aktif bir araştırma konusu olmuştur. KSA’lar titreşim, hareket, sıcaklık, ses ve kirlilik gibi ortamdaki değişiklikleri takip etmek için birlikte hareket eden sensörlerden oluşan kablosuz ağ yapılarıdır. KSA’larda, küçük boyutlu, düşük enerji kullanan sensör düğümleri mevcuttur. Bu düğümlerde, genel olarak sensör vasıtası ile algı yapabilme, veriyi işleyebilme ve diğer sensörlerle haberleşme yapabilme özellikleri bulunmaktadır. KSA içerisinde konumu bilinmeyen birçok sensör yer alır. Bunun yanında konumunu bilen az sayıda sensör de bulunur; bunlara referans düğüm (anchor, çapa) denir. Referans düğümler konumlarını GPS aracılığı ile ya da farklı yöntemlerle bilmektedir. KSA’larda sensör sayısı çok fazla olduğundan konum belirlemek için GPS ya da farklı yöntemler kullanılamamaktadır. Bunun için yeri bilinmeyen sensörler, referans düğümleri ve diğer sensörler aracılığı ile konumunu bulmaya çalışırlar. Normal sensörlerin, referans düğümlerinin yanında, diğer normal sensörlerle de yardımlaşarak konumlarını bulmaları, iş birliğine dayalı konum belirlemede yeni bir paradigmatik yöntem olarak ortaya çıkmıştır (Akyildiz vd., 2002; Wymeersch vd., 2009; Patwari vd., 2005; Eren, 2017; Eren, 2016; Eren, 2011; Eren vd., 2004; Aspnes vd., 2006).

Son yıllarda yapılan çalışmalarda sensör düğümlerinin konumunu belirlemede farklı algoritmalar önerilmiştir. Bütün bu KSA konum belirleme algoritmaları başlangıçta bilinmeyen konumlara sahip sensör düğümlerinin yerini tespit ederken ortak özellikler gösterir. Konum belirleme probleminde, çözüm aşamasında kullanılan optimizasyon algoritmaları arasında, sezgisel yöntemler de bulunmaktadır. Literatürde KSA’larında konum belirleme problemini optimizasyon yöntemleri kullanarak inceleyen çalışmalardan bazıları aşağıda verilmiştir.

D. Li ve X. Wen, (2015) yaptıkları çalışmada KSA’larda konum belirleme problemini çözmek, verimliliği ve hassasiyeti artırmak için geliştirilmiş iki fazlı bir PSO algoritması önerdiler ve çıkan sonuçları klasik PSO ve HPSO yöntemleriyle karşılaştırdılar. N. Bozkurt, (2015) yüksek lisans tezinde, KSA’larda konumlandırma problemini inceledi ve konumlandırma için kullanılan uzaklık ölçme tekniklerinden birisi olan Sinyal Alım Gücü (Received Signal Strength- RSS) yöntemini uzaklık tahmininde kullandı. Q. Zhang ve arkadaşları, (2008) yaptıkları çalışmada Genetik Algoritma (GA) kullanarak KSA’larda konum belirleme problemini incelediler. N. Barak ve arkadaşları, (2016) KSA’larda konum belirleme problemi için değiştirilmiş bir PSO algoritması metodu kullanarak geleneksel PSO tekniğinden daha iyi yakınsama sonuçlarına ulaştılar. A. Gopakumar ve L. Jacob, (2008) KSA’larda konum belirleme problemini geleneksel PSO yöntemiyle çözdüler. KSA’larda konum belirleme problemini P. Namin ve M. Tinati, (2011) iki aşamalı olarak mesafeye dayalı PSO algoritması ile incelediler. J. Lv ve arkadaşları, (2012) KSA’larda daha doğru bir arama alanında mümkün olduğunca çok sayıda bilinmeyen sensörün yerini dağıtılmış PSO tabanlı yöntem ile bulmaya çalıştılar. Önerilen yöntemin önceki önerilen algoritmalarından daha iyi performans gösterdiğini belirttiler. Kumar ve arkadaşları, (2012) yaptıkları çalışmada KSA’larda konum belirleme problemi için H-Best Particle Swarm Optimization (HPSO) ve Biogeography Based Optimization (BBO) algoritmalarını kullanıp sonuçları karşılaştırdılar. Kulkarni ve Venayagamoorthy, (2010) KSA’larda sensör yerleştirme ve sensörlerin konumunu belirleme probleminde, PSO ve Bacterial Foraging Algorithm (BFA) metodlarını kullandılar. Kulkarni ve arkadaşları, (2009) KSA’larda konum belirleme probleminde PSO ve BFA yöntemlerini kullanıp sonuçları karşılaştırdılar. Shieh ve arkadaşları, (2016) KSA’larda konum belirleme problemini GA, PSO, Grey Wolf Optimizer (GWO), Firefly Algorithm (FA) ve Brain Storm Optimization (BSO) metodlarıyla çözüp sonuçları incelediler. Kulkarni ve Venayagamoorthy, (2011) KSA’larda optimal dağıtım, düğüm yeri belirleme, kümeleme ve veri toplama konularındaki problemleri PSO metodlarıyla açıkladılar. Özsağlam ve Çunkaş, (2008) yaptıkları çalışmada PSO ile GA ve DGO performanslarını, bazı test fonksiyonlarını kullanarak karşılaştırdılar. Çalışma sonucunda, PSO ’nun her iki algoritmaya göre yakınsama oranı ve performans açısından daha iyi sonuçlar ürettiğini belirttiler.

## 2. KSA’larda Konum Belirleme Problemi

KSA’larda konum belirleme problemi bir KSA’da bulunan sensörlerin yerinin tespit edilmesini ifade eder. KSA’larda sensörlerin konumunu doğru belirleme işlemi çok önemlidir. Konumlandırma için sistemler çeşitli ölçüm bilgileri kullanır. Kullanılan yöntemlerden biri de mesafeye dayalı hesaplamadır. Mesafeye dayalı yöntemde yerini bulmak istediğimiz sensörün komşularına olan uzaklığını tespit ederek en az 3 adet yerini bildiğimiz sensör yardımıyla bu sensörün yerini tespit edebiliriz.



Şekil 1. KSA'da Mesafeye Dayalı Konum Belirleme

KSA'larda konum belirleme probleminde  $m$  adet referans düğüm ile  $n$  adet hedef sensörlerin olduğunu varsayarsak, referans düğümlerin koordinatları  $(x_i, y_i)$  olur.

$$x_i = [x_1, \dots, x_m], \quad y_i = [y_1, \dots, y_m] \quad (1)$$

Hedef düğümlerin tahmini konum koordinatları  $(x_j, y_j)$  olur.

$$x_j = [x_1, \dots, x_n], \quad y_j = [y_1, \dots, y_n] \quad (2)$$

Referans düğümleri ile hedef düğümler arasındaki ölçülen mesafeye gerçek uzaklık ( $d_i$ ) denir. Hedef düğümün koordinatları  $(x, y)$ ,  $i$ . referans düğümünün koordinatları  $(x_i, y_i)$  olduğunda denklemi aşağıdaki gibidir.

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (3)$$

Ölçülen gerçek uzaklığa gürültü ( $n_i$ ) eklenerek tahmini uzaklık ( $d_{in}$ ) bulunur. Bir sensörden  $i$ . referans düğüme kadar elde edilen tahmini mesafenin denklemi aşağıdaki gibidir.

$$d_{in} = d_i + n_i \quad (4)$$

KSA'larda konum belirleme problemi bir optimizasyon problemi olarak kabul edilir. Amaç ölçülen gerçek mesafe ile tahmini olarak hesaplanan mesafe arasındaki farkı minimize etmektir. Tanımlanan amaç fonksiyonu aşağıdaki gibidir (Aloor ve Lillykutty, 2008).

$$f(x, y) = \frac{1}{m} \sum_{i=1}^m (\sqrt{(x - x_i)^2 + (y - y_i)^2} - d_{in})^2 \quad (5)$$

### 3. Kullanılan Optimizasyon Algoritmaları

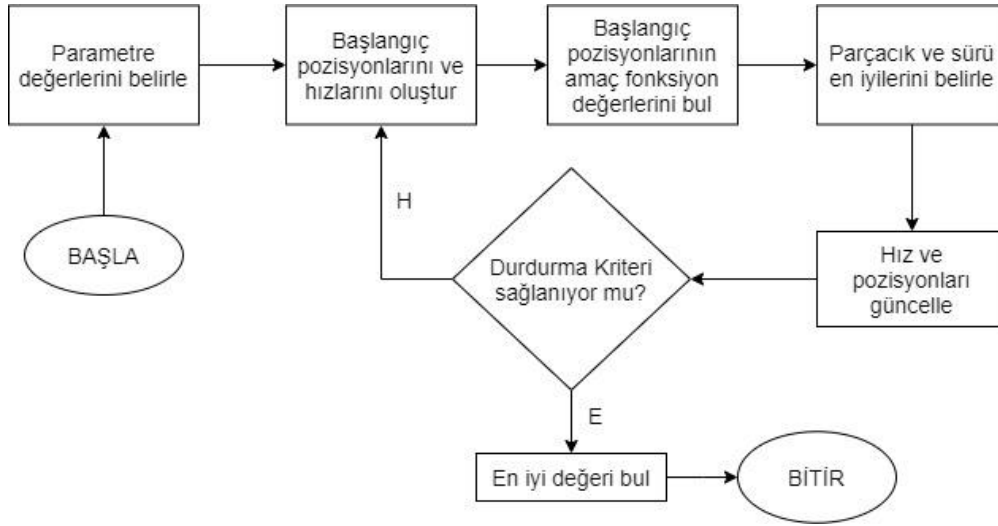
#### 3.1. Parçacık Sürü Optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (PSO), balıklar ile kuşların toplu halde hareketinden ilham alan Kennedy ve Eberhart (1995) tarafından keşfedilen bir optimizasyon yöntemidir. Genel olarak sürü zekâsına dayanır. PSO 1995'te tanıtılmasından sonra üzerinde çeşitli geliştirmeler yapılmış ve uygulamalar gerçekleştirilmiştir. Temel PSO üzerindeki çoğu modifikasyon PSO'nun yakınsama hızını geliştirmeye ve sürünün farklılaşmasının artırılmasına yöneliktir (Karaboğa, 2004).

PSO algoritması hedef fonksiyonunun değişimini hesaplamak için cebirsel işlemlere gerek duymayan bir global optimizasyon algoritmasıdır. Yöntemin ilk bulunan kısmında sadece doğrusal olmayan sürekli optimizasyon problemleri çözülebiliyordu. Daha sonra yöntem ilerletilerek daha karmaşık mühendislik problemlerinin en optimum çözümlerinde kullanılmaya başlandı.

PSO, balık ve kuş sürülerinin hareketlerinden ortaya çıkmıştır. Kuşların hareket uzayında, yerini bilmedikleri bir nesneyi rastgele gelişen hareketler sayesinde arayıp bulmaları, bir problemin çözümüne benzetilir. Kuşlar nesneyi ararken o nesneye en yakın olan kuşu takip ederler. PSO'nun temel olarak dayandığı nokta, gruptaki bireyler arasında gerçekleşen veri paylaşımını geliştirmektir. Bu yöntemde bireylere parçacık adı verilir. Bu parçacıklardan oluşan grup da sürü olarak adlandırılır (Kennedy ve Eberhart, 1995).

PSO 'da her bir birey kendi konumunu, bir önceki deneyiminden faydalanarak sürüdeki en iyi konuma doğru ayarlar. PSO, genel olarak sürüde bulunan bireylerin konumunun, sürünün en iyi konumunda bulunan bireye yaklaşması esasına dayanır. Bu yaklaşma durumu rastgele gelişen bir olaydır. Çoğunlukla sürü içerisinde bulunan bireyler bir sonraki hareketlerinde bir önceki konumdan daha iyi bir konuma ulaşırlar ve bu durum hedeflenen sonuca varıncaya kadar devam eder. Şekil 2'de PSO'nun akış diyagramı görülmektedir.



Şekil 2. PSO Akış Diyagramı

Algoritmanın, problem boyutu, popülasyon sayısı, atalet ağırlık değeri gibi parametreleri bulunmaktadır ve bunlar başlangıçtaki iterasyonda rastgele değerler olarak sürece başlanır. Bu işlem konum ve hız vektörünün belirtilen sınırlar içerisinde değer almasıdır. Rastgele değerler olarak oluşan konum vektör değerlerinin hedef fonksiyon değerleri hesaplanır. Daha sonra bu hedef fonksiyon değerleri karşılaştırılarak parçacık en iyisi ile sürü en iyisi değerleri bulunur. Sonraki aşama ise hız vektörünün güncellenmesidir. Hız vektörünün güncellenmesi denklem 6'da gösterilmektedir. Hız güncelleme denklemi uygulandıktan sonra bazı parçacıklarda alt sınır ve üst sınır değerlerinin aşıldığı durumlar meydana gelebilir. Dolayısıyla sınırları aşan hız değerleri alt sınır ve üst sınır değerlerine çekilir (Çetin, 2011).

$$V_i^{k+1} = (w^{k+1} * V_{ij}^k) + c_1 * r_1 (p_{ij}^k - x_{ij}^k) + c_2 * r_2 (g_{ij} - x_{ij}^k) \quad (6)$$

Denkleminde yer alan  $r_1$  ve  $r_2$ , 0,1 aralığında homojen dağıtılmış rastgele değerleri,  $c_1$  ve  $c_2$  ise kişisel öğrenme ve global öğrenme değişken değerleridir. Bir sonraki adım ise yeni konum vektörü elde edilen hız vektör ifadesi ile bir önceki konum vektörünün toplanarak güncellenmesi işlemidir (Çetin, 2011). Pozisyon güncellenmesi denklem 7'de gösterilmektedir.

$$x_{ij}^{k+1} = x_{ij}^k + v_{ij}^{k+1} \quad (7)$$

Pozisyon güncelleme denklemi uygulandıktan sonra bazı parçacıkların pozisyon değerleri alt sınır ve üst sınır değerlerini aşabilir. Dolayısıyla sınırları aşan değerler sınır değerlerine çekilir. Yeniden oluşturulan pozisyon vektör değerleri ile amaç fonksiyonu değerleri, sürü en iyisi ve parçacık en iyileri yeniden hesaplanır. Bu işlemler iterasyon sayısı bitene kadar devam eder.

### 3.2. Yarasa Algoritması (YA)

Global optimizasyon problemlerinin çözümünde kullanılacak YA, Xin-She Yang tarafından 2010 yılında bilim dünyasına sunulmuş bir algoritmadır. YA sesle konum belirleme (echolocation) esasına dayanır. Doğal hayatta çeşitli yarasa türleri avlarını tespit etme, engellerden kaçınma ve karanlıkta tüneme amacıyla kullandıkları yarıklarını bulmak için bir çeşit sonar olan sesle konum belirleme yöntemini kullanmaktadır. YA bu doğal sürecin çeşitli optimizasyon problemlerini çözmek amacıyla taklit edildiği bir doğal metasezgisel optimizasyon algoritmasıdır (Yang, 2010).

Sürüdeki her yarasa avlarının yerini belirlemek ve kendi aralarında haberleşmek için ekolokasyon adı verilen bir çeşit iletişim ağı kullanırlar. Yarasa ekolokasyonu eko meydana getirmek için bir dizi yüksek sesli ultrasonik sinyallerin oluşturulduğu bir algılama sistemidir. Bu sinyaller yarasalara belirli bir avın yerini tespit etmesini sağlayan çeşitli ses düzeyleri ile geri gelir. Yarasa algoritmasının mimarisini büyütmek ve yarasaların ekolokasyon niteliklerini değerlendirmek için bazı ilkeler uygulanmaktadır (Ekinci, 2015):

(a) Yarasalar avlarını veya kendisi için tehlike oluşturacak engelleri algılamak için ekolokasyon özelliğinden yararlanır.

(b) Yarasa ların avını bulması için;  $v_i$  hızında,  $x_i$  pozisyonunda, sabit  $f_{\min}$  frekansında,  $r$  değişkenli dalga boyunda ve  $L_0$  ses şiddeti ile gelişigüzel uçarlar. Hedefleri ile olan mesafeye göre oluşturdukları dalgaların frekanslarını ayarlarlar ve 0 ile 1 arasında  $r$  sinyal yayılım oranlarını düzenlerler.

(c) Her yarasanın frekansı, ses şiddeti ve sinyal yayılım oranı farklılık gösterebilir.

(d)  $L_m^{iter}$  ses şiddeti,  $L_0$  büyük değerden  $L_{\min}$  en küçük sabit değere kadar farklı olabilir.

Optimizasyon süresince her yarasanın  $x_i$  pozisyonu ve  $v_i$  hızı tanımlanabilir ve güncellenebilir.  $t$  zaman aralığında  $x_i^t$  güncel pozisyon değerleri ve  $v_i^t$  güncel hızları aşağıda bulunan eşitliklerden hesaplanır.

$$f_i = f_{\min} + (f_{\max} - f_{\min})a \quad (8)$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^p)f_i \quad (9)$$

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

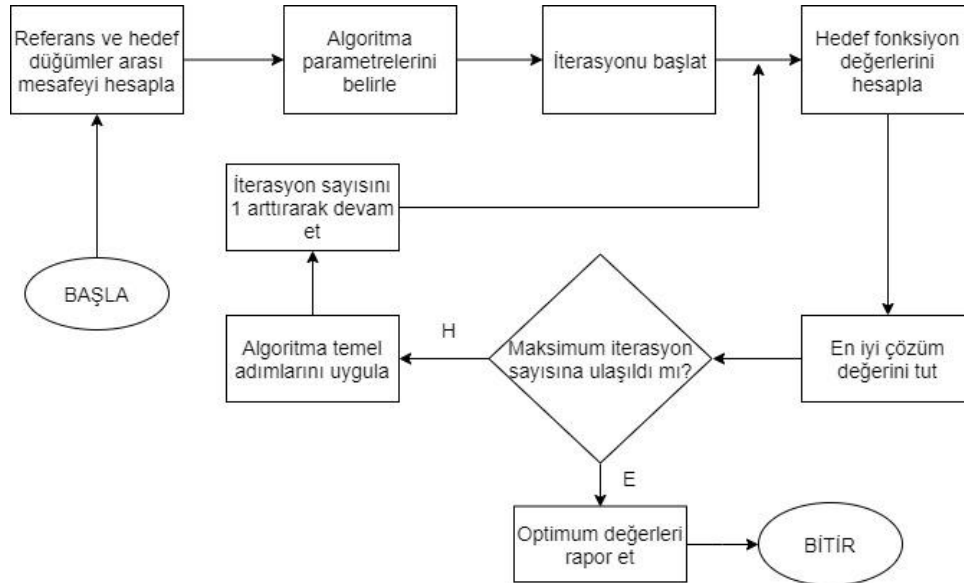
Eşitlik 8'de  $\alpha$ , 0 ile 1 arasında alınan rastsal bir vektörü,  $f_i$  yarasanın frekans değerini  $f_{\min}$  ve  $f_{\max}$  sırasıyla en küçük ve en büyük frekans değerlerini ve  $x^*$  sürüde bulunan en iyi çözüm değerini ifade eder. Hesaplanan değerler içinden en iyi çözüm değeri alındıktan sonra, yerel olarak rastgele yapılan işlemler sonucunda yeni bir çözüm değeri bulunur.

$$x_{new} = x_{old} + \varepsilon L^t \quad (11)$$

Eşitlik 11'de  $\varepsilon$ , 1 ile -1 arasında rastsal dağılmış bir değeri ve  $L^t$   $t$  zaman aralığındaki tüm yarasa ların ortalama ses şiddetini ifade etmektedir. İterasyon ilerledikçe ve istenen amaç doğrultusunda yaklaşıldıkça yarasa ların ekolojasyon ile çıkardıkları sesin şiddeti ve sinyal yayılım oranlarının güncellenmesi gerekir. Yarasa avını tespit ettiğinde çoğunlukla ses şiddeti ( $L$ ) düşmesine rağmen sinyal yayılım oranı ( $r$ ) artışa geçmektedir.

$$A_i^{t+1} = \beta A_i^t, r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (12)$$

Eşitlik 12'de  $\beta$ , 0 ile 1 arasında bir sabit ve  $\gamma$  pozitif sabit değerdir.  $t \rightarrow \infty$  iken ses şiddeti  $L_i^t \rightarrow 0$  ve  $r_i^t \rightarrow r_i^0$  olarak gerçekleşmektedir. Şekil 3'te YA'nın akış diyagramı görülmektedir.



Şekil 3. Yarasa Algoritması Akış Diyagramı

#### 4. Simülasyon

Bu kısımda KSA’larda hedef sensörlerin lokalizasyon probleminin çözümünde PSO ve YA sezgisel optimizasyon yöntemleri kullanılmıştır. Kullanılan bu yöntemlerin performansları karşılaştırılmıştır.

##### 4.1. Parametre Değerleri

Algoritmalarda kullanılan parametre değerleri Tablo 1 ve 2’de sunulmuştur. Algoritmaların ortak değişkenleri için aynı değerler kullanılmıştır.

**Tablo 1.** PSO Algoritması Parametreleri

Parametreler	Değerler
Boyut Sayısı	2
Popülasyon Sayısı	20
Kişisel Öğrenme Katsayısı (c1)	1.5
Global Öğrenme Katsayısı (c2)	2
Atalet Ağırlık Katsayısı Oranı (wdamp)	0.99
İterasyon Sayısı	100

**Tablo 2.** Yarasa Algoritması Parametreleri

Parametreler	Değerler
Boyut Sayısı	2
Popülasyon Sayısı	20
Maksimum Frekans ( $f_{max}$ )	2
Ortalama Ses şiddeti ( $L$ )	0.25
Sinyal Emisyon Hızı ( $r$ )	0.8
İterasyon Sayısı	100

##### 4.2. Hata Hesaplaması

KSA’larda bir sensör düğümünün konumunu belirlemek için bu düğümün üç veya daha fazla komşu referans düğüm içermesi durumunda konumu tespit edilebilir. Bu nedenle, daha fazla hedef düğümün yerini belirlemek için, her bir sensör düğümünün iletim yarıçapı yeterince büyük olmalıdır. Bu simülasyonun amacı hedef düğümlerin konumunu optimizasyon algoritmaları kullanarak tespit etmektir. Tahmin edilen düğüm konumlarının gerçek düğüm konumları ile arasındaki hatayı hesaplamak için denklem 13’teki hata fonksiyonu kullanılmıştır.

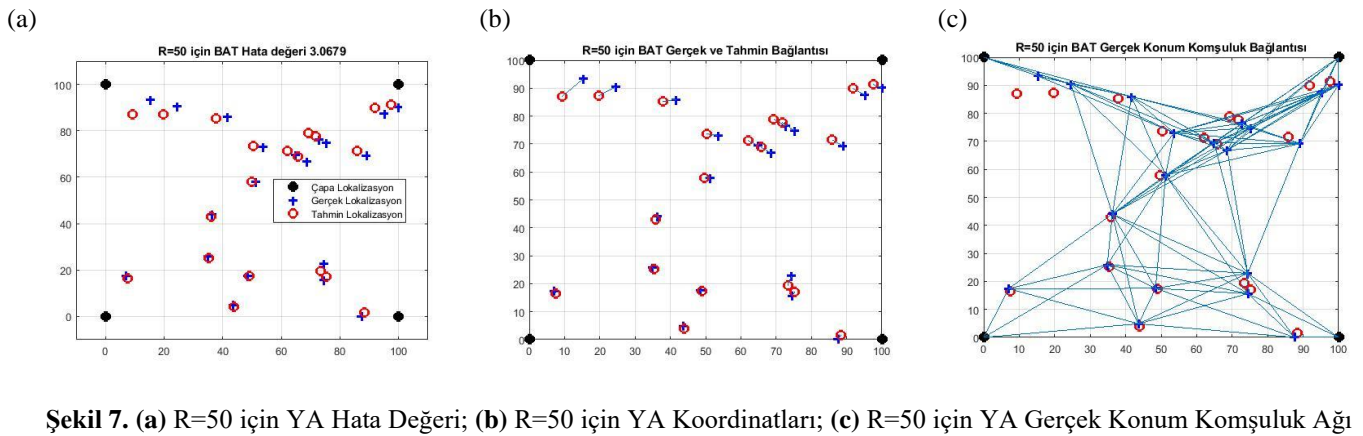
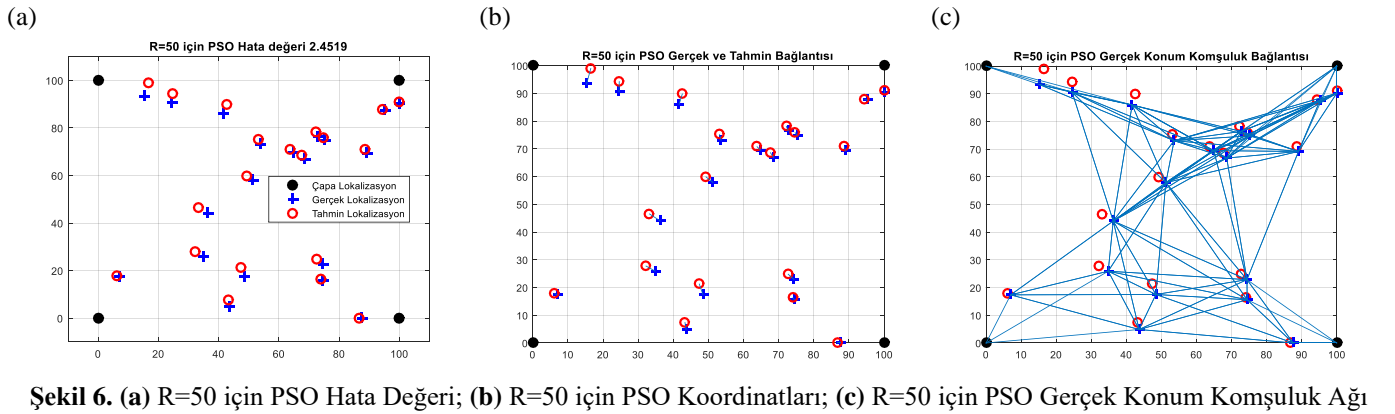
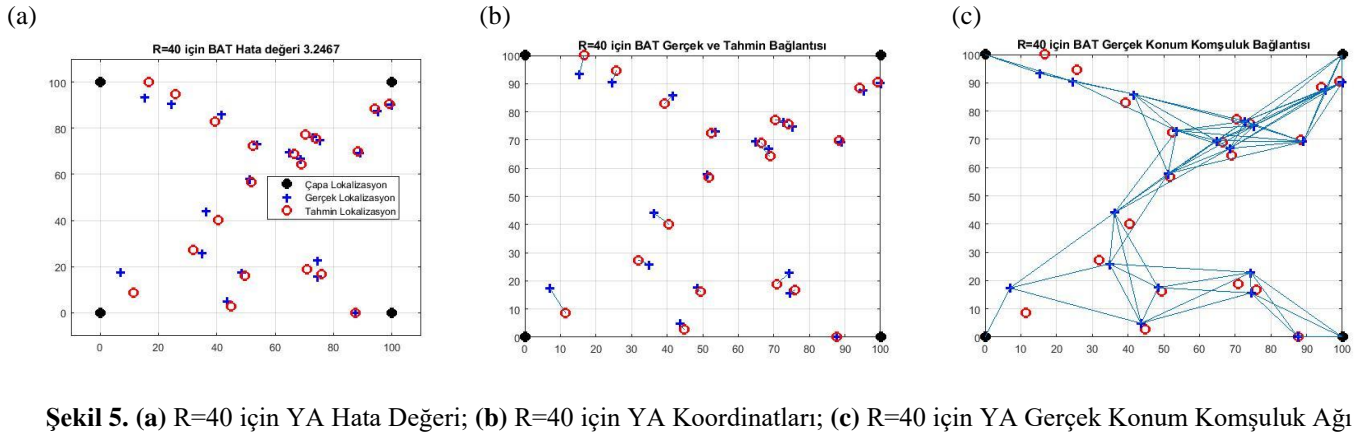
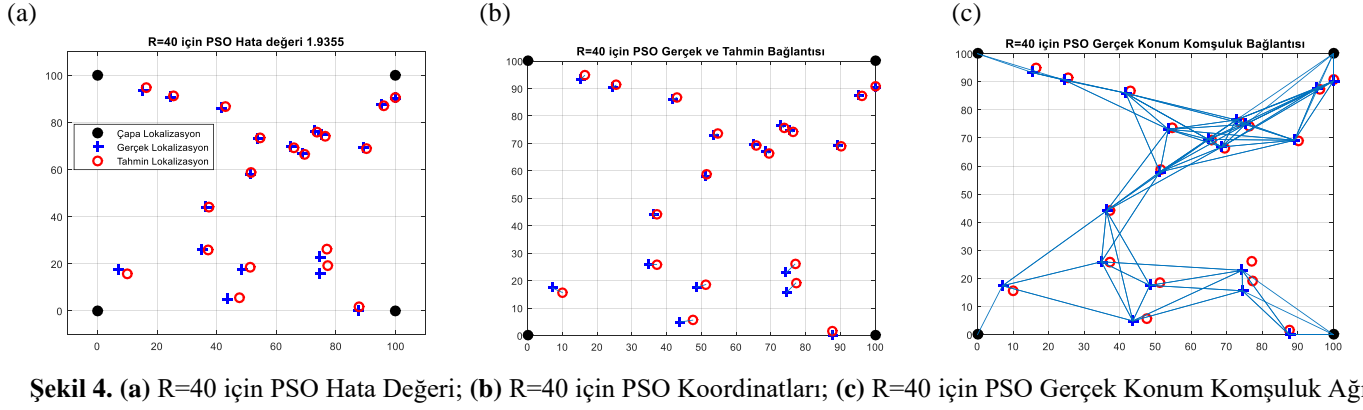
$$E = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2} \quad (13)$$

(N) hedef düğüm sayısını, ( $X_i, Y_i$ ) tahmin edilen hedef düğüm koordinatlarını, ( $x_i, y_i$ ) ise gerçek hedef düğüm koordinatlarını ifade eder.

#### 5. Bulgular

MATLAB programında 100x100 lük bir alanda köşelere yerleştirilmiş 4 adet referans düğüm ile 20 adet düğüm noktasının yer tespiti PSO ve YA algoritmaları ile yapılmıştır. Her bir sensör düğümünün kapsama alan yarıçapı R = 40 birim ve R=50 birim alınarak iki farklı senaryo uygulanmıştır. Her iki algoritma aynı koordinatların tespiti için çalıştırılmıştır. Simülasyon 1’e ait grafikler Şekil 4,5,6,7’de gösterilmiştir. Şekil 4a’da PSO için R=40 birim alınarak yapılan simülasyon görülmektedir. Burada mavi artı işaretleri, düğümlerin gerçek yerlerini göstermektedir. Kırmızı yuvarlak işaretler ise düğümlerin PSO tarafından tahmin edilen konumlarını göstermektedir. Siyah yuvarlaklar, referans düğümlerinin (çapa) yerlerini göstermektedir. Şekil 4b’de düğümlerin gerçek konumları ile tahmin edilen konumları arasındaki uzaklıklar, açık mavi çizgilerle gösterilmiştir. Şekil 4c’de ise, R=40 için gerçekleşen ağ çizgesi (network graph) görülmektedir. Şekil 5a, 5b, 5c’de Yarasa Algoritmasının (BAT), R=40 birimde verdiği sonuçlar gösterilmiştir. Şekil 6a, 6b, 6c’de PSO Algoritmasının R=50 birimde verdiği sonuçlar; Şekil 7a, 7b, 7c’de ise Yarasa Algoritmasının R=50 birimde verdiği sonuçlar gösterilmiştir. Simülasyon, R=40 birim ve R=50 birim olacak şekilde, hem PSO

hem de YA için 50 kere tekrarlanmıştır. Hata oranlarının ayrıntılı tablosu on adet simülasyon için (Simülasyon 1'den, Simülasyon 10'a kadar) Tablo 3'de görülmektedir. Elli adet simülasyon için ortalama değerleri gösteren tablo ise Tablo 4'te sunulmuştur.



**Tablo 3.** Hata Oranları Tablosu (10 adet simülasyon)

İletim Mesafesi (R)	PSO		YA	
	R=40	R=50	R=40	R=50
Simülasyon 1	2.4519	1.9355	3.0679	3.2467
Simülasyon 2	6.0936	3.5143	33.0676	13.4585
Simülasyon 3	24.3887	13.7524	18.3394	3.9278
Simülasyon 4	10.059	6.0075	7.6632	5.161
Simülasyon 5	11.3471	2.8167	13.5062	4.0474
Simülasyon 6	13.0386	2.101	14.8112	5.6601
Simülasyon 7	8.4032	12.4097	7.9247	2.3033
Simülasyon 8	18.7929	6.4541	5.4007	8.6512
Simülasyon 9	14.9099	6.6066	11.2055	7.6509
Simülasyon 10	7.1678	3.8517	24.048	10.267
<b>Ortalama</b>	<b>11.66527</b>	<b>5.94495</b>	<b>13.90344</b>	<b>6.43739</b>

**Tablo 4.** 50 adet simülasyon için ortalama hata oranları tablosu

İletim Mesafesi (R)	R=40	R=50
PSO	10,028766	4,4588004
YA	11,901	4,701206

## 6. Tartışma ve Sonuç

Tablo 3 ve Tablo 4'te görüldüğü gibi, R arttıkça, PSO ve YA için hata oranları düşmektedir. Bu beklenen bir sonuçtur, çünkü R'nin artması ağ üzerindeki bağlantı sayısını artırmaktadır. Bu da düğümlerin daha fazla veri ile konumlarını belirlemelerini sağlamaktadır. PSO ve YA'yı karşılaştırdığımızda, PSO'nun YA'dan ortalama olarak daha iyi sonuç verdiği görülmektedir. Tablo 3'te görülen ayrıntılı verilerde, bazı simülasyonlarda YA, PSO'dan daha iyi sonuç vermiştir. Ancak, hem 10 simülasyon için, hem de 50 simülasyon için, ortalama olarak PSO, YA'dan daha iyi sonuç vermektedir. Bu fark, R=40'ta daha bariz olarak görülmektedir. R=40'ta bağlantılı sayılarının az olmasından dolayı veri sayısının da az olacağı dikkate alındığında, PSO'nun YA'dan az verili durumlarda daha belirgin şekilde daha iyi sonuç verdiğini söyleyebiliriz.

Sonuç olarak, bu makalede üstünde çalıştığımız konum belirleme probleminin temel amacı, doğadan esinlenen optimizasyon algoritmalarının yardımıyla hedef sensörlerin gerçek konumunu en az hata ile tespit etmektir. Bu kapsamda sezgisel optimizasyon yöntemleri olan PSO ve YA kullanılmıştır. Hata oranlarını incelediğimizde PSO algoritması ile alınan sonuçların YA'ya göre daha iyi olduğu gözlemlenmektedir.

## Referanslar

- Akyildiz IF, Su W, Sankarasubramaniam Y et al. Wireless sensor networks: a survey. *Computer Networks* 2002; 38(4): 393–422.
- Aspnes J, Eren T, Goldenberg D et al. A theory of network localization. *IEEE Transactions on Mobile Computing* 2006; 5(12): 1663–1678.
- Aloor, G. & Lillykutty, J. (2008). Localization in Wireless Sensor Networks using Particle Swarm Optimization, no. 525, pp. 424–, 2008.
- Barak, N., Gaba, N. & Aggarwal, S. (2016). Localization of sensor nodes using modified particle swarm optimization in wireless sensor networks, 2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016, pp. 2608–2613, 2016.
- Bozkurt, N. (2015). Sinyal Alım Gücü Ağırlıklı Ortalamaya Dayalı Ağ Konumlandırması İçin Kablosuz Ağlarda Referans Düğümlerinin Yerleştirilmesi, Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği, 2015.



- Çetin, E. (2011). Rüzgar Enerjisi Dahil Olan Güç Sistemlerinde Ekonomik Yük Dağıtım Probleminin Çözümü, Erciyes Üniversitesi Fen Bilimleri Enstitüsü, 2011.
- Ekinçi S. (2015). Power system stabilizer design for multi-machine power system using bat search algorithm, *Sigma Mühendislik ve Fen Bilimleri Dergisi*, 33(4): 627-637, 2015.
- Eren T, The effects of random geometric graph structure and clustering on localizability of sensor networks, *International Journal of Distributed Sensor Networks*, vol. 13, no.12, pp. 1-14, 2017.
- Eren T, Graph invariants for unique localizability in cooperative localization of wireless sensor networks: rigidity index and redundancy index, *Ad Hoc Networks*, vol. 44, pp. 32-45, 2016.
- Eren T, “Cooperative localization in wireless ad hoc and sensor networks using hybrid distance and bearing (angle of arrival) measurements” *EURASIP Journal on Wireless Communications and Networking* 2011, 2011:72.
- Eren T, Goldenberg DK, Whiteley W et al. Rigidity, computation, and randomization in network localization. In *Proceedings of the 2004 International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004)*. Hong Kong, pp. 2673–2684.
- Karaboğa, D. (2004). *Yapay Zeka Optimizasyon Algoritmaları*. Nobel Yayın Dağıtım, 2004.
- Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization, *Nat. Comput. Ser.*, pp. 97–102, 1995.
- Kulkarni, R. V., Venayagamoorthy, G. K. & Cheng, M. X. (2009). Bio-inspired node localization in wireless sensor networks, *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, no. October, pp. 205–210, 2009.
- Kulkarni, R. V. & Venayagamoorthy, G. K. (2010). Bio-inspired algorithms for autonomous deployment and localization of sensor nodes, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 40, no. 6, pp. 663–675, 2010.
- Kulkarni, R. V. & Venayagamoorthy, G. K. (2011). Particle swarm optimization in wireless-sensor networks: A brief survey, *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 41, no. 2, pp. 262–267, 2011.
- Kumar, A., Khosla, A, Saini, J. S. & Singh, S. (2012). Computational intelligence based algorithm for node localization in Wireless Sensor Networks, *IS'2012 - 2012 6th IEEE Int. Conf. Intell. Syst. Proc.*, pp. 431–438, 2012.
- Li, D., & Wen, X. b. (2015). An Improved PSO Algorithm for Distributed Localization in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*. Vol 2015. doi:10.1155/2015/970272
- Lv, J., Cui, H. & Yang, M. (2012). Distribute localization for wireless sensor networks using particle swarm optimization, *ICSESS 2012 - Proc. 2012 IEEE 3rd Int. Conf. Softw. Eng. Serv. Sci.*, pp. 355–358, 2012.
- Namin, P. H. & Tinati, M. A. (2011). Node localization using Particle Swarm Optimization, *Proc. 2011 7th Int. Conf. Intell. Sensors, Sens. Networks Inf. Process. ISSNIP 2011*, pp. 288–293, 2011.
- Özsağlam, Y. & Çunkaş, M. (2008). Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması, *Politek. Derg.*, vol. 11, no. 4, pp. 299–305, 2008.
- Patwari N, Ash JN, Kyperountas S et al. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine* July 2005; 22(4): 54–69.
- Shieh, C. S., Sai, V. O., Lin, Y. C., Lee, T. F., Nguyen, T. T. & Le, Q. D. (2016). Improved node localization for WSN using heuristic optimization approaches, *Proc. - 2016 Int. Conf. Netw. Netw. Appl. NaNA 2016*, no. 4, pp. 95–98, 2016.
- Wymeersch H, Lien J and Win M. Cooperative localization in wireless networks. *Proceedings of the IEEE* 2009; 97(2):427–450.
- Yang, X. S. (2010). A New Metaheuristic Bat-Inspired Algorithm, *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 284, 65-74, 2010.
- Zhang, Q., Wang, J., Jin, C., Ye, J., Changlin, M., & Zhang, W. (2008). Genetic algorithm based wireless sensor network localization, *Proc. - 4th Int. Conf. Nat. Comput. ICNC 2008*, vol. 1, no. 2007, pp. 608–613, 2008. doi: 10.1109/ICNC.2008.206