



Kablosuz Sensör Ağlarında Konum Belirlemede Parçacık Sürü Optimizasyonu, Yarasa Algoritması, Diferansiyel Gelişim Algoritması ve Ateşböceği Algoritması Yöntemlerinin Karşılaştırılması

Comparison of Particle Swarm Optimization, Bat Algorithm, Differential Evolution Algorithm and Firefly Algorithm Methods in Localization of Wireless Sensor Networks

Ahmet Sami Doğru , Tolga Eren 

Kırıkkale Üniversitesi, Elektrik Elektronik Mühendisliği Bölümü, Kırıkkale, TÜRKİYE

Başvuru/Received: 27/10/2020

Kabul / Accepted: 02/12/2020

Çevrimiçi Basım / Published Online: 31/12/2020

Son Versiyon/Final Version: 31/12/2020

Öz

Kablosuz sensör ağlarında konumlandırma birçok alanda her geçen gün önemli bir konu haline gelmektedir. Askeri uygulamalar, acil yardım, coğrafi takip gibi uygulamalarda sensörlerin yerini hata payı en düşük ve hızlı bir şekilde bulmak çok önemlidir. Kablosuz sensör ağlarında (KSA), konum tespiti çok yönlü bir optimizasyon sorunu olarak belirtilmektedir. KSA'ların kompleksliğine göre optimizasyon metotlarının verimi değişkenlik göstermektedir. Bu çalışmada, Parçacık Sürü Optimizasyonu (PSO), Yarasa Algoritması (YA), Diferansiyel Gelişim Algoritması (DGA) ve Ateşböceği Algoritması (AA) yöntemleri KSA'larda konum belirleme sorununun çözülmesi hedefi ile kıyaslanmıştır. Önerilen yöntemler, doğada bulunan canlıların karakteristiğinden ilham alınarak keşfedilmiş optimizasyon yöntemleridir. Bu çalışmada sezgisel optimizasyon yöntemlerinden olan PSO, YA, DGA ve AA Matlab ortamında bir örnek düğüm dağılımına uygulanmış olup sonuçlar gözlemlenmiştir.

Anahtar Kelimeler

"Kablosuz sensör ağları, Konum belirleme, Parçacık sürü optimizasyonu, Yarasa algoritması, Diferansiyel gelişim algoritması, Ateşböceği algoritması"

Abstract

Positioning in wireless sensor networks is becoming an important issue in many areas every day. It is very important to find the location of the sensors quickly and with low error rate in applications such as military applications, emergency assistance, geographic tracking. In wireless sensor networks (WSN), location detection is cited as a multidimensional optimization problem. The efficiency of optimization methods varies according to the complexity of WSN's. In this article, Particle Swarm Optimization (PSO), Bat Algorithm (BA), Differential Evolution Algorithm (DEA) and Firefly Algorithm (FA) methods were compared with the goal of solving the positioning problem in WSN. The proposed methods are optimization methods that are inspired by the characteristics of living beings in nature. In this study, heuristic optimization methods PSO, BA, DEA and FA were applied to a sample node distribution in Matlab environment and the results were observed.

Key Words

"Wireless sensor networks, Localization, Particle swarm optimization, Bat algorithm, Differential evolution algorithm, Firefly algorithm."

1. Giriş

Günümüzde KSA’larda bulunan sensörlerin konum bulma yöntemi etkin bir inceleme konusu olmuştur. KSA’lar ısı, ışık, gürültü, nem gibi çevredeki değişimleri izlemek için beraber hareket eden sensörlerden oluşan kablosuz ağ elemanıdır. KSA’larda, küçük ebatlı, daha az enerji tüketen sensör düğümleri bulunur. Bu düğümlerde, yaygın olarak sensör aracılığı ile bir ortam değişikliğini tespit edebilme, bilgiyi geliştirme ve diğer sensörlerle iletişim kurabilme özellikleri mevcuttur. KSA yapısı içinde konumu tespit edilmemiş birçok sensör bulunur. Bununla birlikte konumunu bilen az sayıda sensör de vardır; bunlara referans düğüm (anchor, çapa) denir. Referans düğümler konumlarını GPS yöntemi ile ya da farklı işlemlerle bulurlar. KSA’larda sensör adedi çok yüksek olduğundan konum tespiti için GPS ya da farklı metotlar kullanılamamaktadır. Bunun için konumu bilinmeyen sensörler, referans düğümleri ve diğer sensörler aracılığı ile konumunu tespit etmeye çalışırlar. Sensörlerin, referans düğümlerinin yanında, diğer normal sensörlerden destek alarak yerlerini bulmaları, iş birliğine dayalı konum belirlemede yeni bir paradigmatik metot 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 zamanlarda yapılan çalışmalarda sensör düğümlerinin konum tespiti yapılırken farklı algoritmalar kullanılmıştır. Bütün bu KSA konum bulma algoritmaları ilk olarak konumu bilinmeyen sensör düğümlerinin yerini bulurken ortak özellikler gösterir. Konum belirleme probleminde, çözüm aşamasında kullanılan optimizasyon algoritmaları arasında, sezgisel yöntemler de bulunmaktadır. Literatürde KSA’larda konum belirleme problemini çözerken optimizasyon yöntemlerini uygulayan çalışmalardan bazıları aşağıda verilmiştir.

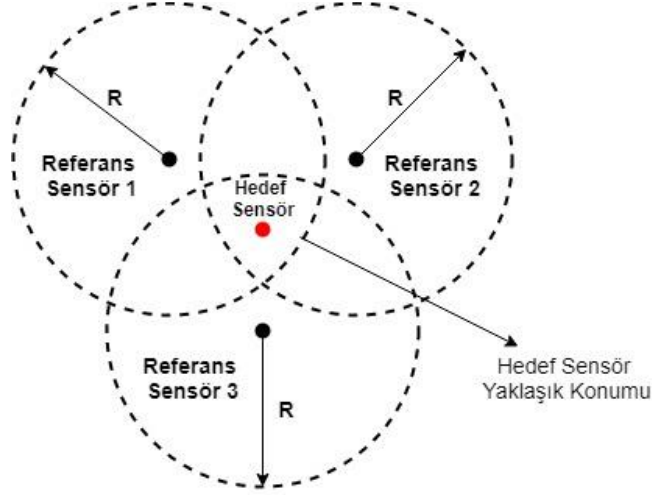
D. Li ve X. Wen, (2015) yaptıkları çalışmada KSA’larda konum belirleme problemini çözerken, hassasiyeti artırılmış iki aşamalı bir PSO algoritması kullandılar ve çıkan sonuçları klasik PSO ve HPSO metotlarıyla karşılaştırdılar. N. Bozkurt, (2015) yüksek lisans tezinde, KSA’larda konumlandırma için kullanılan mesafe ölçme tekniklerinden birisi olan Sinyal Alım Gücü (Received Signal Strength- RSS) yöntemini konum tahmininde kullandı. Q. Zhang vd., (2008) yaptıkları çalışmada KSA’larda konum belirleme problemini Genetik Algoritma (GA) kullanarak incelediler. N. Barak vd., (2016) konumlandırma için klasik PSO yönteminden daha iyi sonuçlara ulaşan farklı bir PSO algoritması kullandılar. A. Gopakumar ve L. Jacob, (2008) KSA’larda konum belirleme için klasik PSO metodunu kullandılar. P. Namin ve M. Tinati, (2011) KSA’larda konumlandırma problemini iki kademeli uzaklığa dayanan PSO algoritması ile çözdüler. J. Lv vd., (2012) KSA’larda konumu bilinmeyen sensörlerin yerini dağıtılmış PSO metodu ile incelediler ve önerilen yöntemin diğer algoritmalarından daha iyi sonuç verdiğini belirttiler. Kumar vd., (2012) KSA’larda konum belirleme için H-Best Particle Swarm Optimization (HPSO) ve Biogeography Based Optimization (BBO) yöntemlerini kullanarak sonuçları incelediler. Kulkarni ve Venayagamoorthy, (2010) sensörlerin konumunu belirlerken PSO ve Bacterial Foraging Algorithm (BFA) yöntemlerini karşılaştırdılar. Kulkarni vd., (2009) KSA’larda konumlandırma problemi için PSO ve BFA yöntemlerini incelediler. Shieh vd., (2016) GA, PSO, Grey Wolf Optimizer (GWO), Firefly Algorithm (FA) ve Brain Storm Optimization (BSO) yöntemlerini kullanarak konum belirleme problemini inceleyip sonuçları karşılaştırdılar. Kulkarni ve Venayagamoorthy, (2011) KSA’larda sensörleri optimum dağıtma, konum belirleme ve bilgi alma konularını PSO yöntemi ile incelediler. Özsağlam ve Çunkaş, (2008) PSO, GA ve DGA yöntemlerini bazı fonksiyonlar kullanarak karşılaştırma yaptılar. İnceleme sonucu PSO’nun diğer iki algoritmadan daha etkin sonuçlar verdiğini belirttiler. Doğru vd., (2019) KSA’larda konum belirleme problemini PSO ve YA kullanarak çözdüler. Çalışmada farklı iletim yarıçapları altında PSO’nun YA’dan daha iyi sonuçlar verdiğini belirttiler. Keskinürk, (2006) yaptığı çalışmada DGA’nın etkin sonuçlar üretebilen popülasyon temelli bir yöntem olduğunu belirtti. Ayrıca GA ile DGA’yı bir probleme uygulayarak sonuçları incelemiştir. Kurtulmuş, (2019) yüksek lisans tezinde DGA yöntemini en az risk ve en çok kar getiren bir portföye uygulayarak en iyi sonucu DGA’nın verdiğini belirtti. Demirdelen, (2018) kuru tip transformatörü DGA ile yeniden tasarlayarak etkin bir şekilde verimini artırdı ve uyguladığı yöntemin performans analizini anlattı. Gandomi vd., (2011) yaptıkları çalışmada 6 farklı yapısal probleme optimizasyon yöntemleri uygulayarak AA’nın GA, PSO ve DGA’dan daha iyi sonuçlar verdiğini incelediler. Kızılcıkaplan vd., (2020) Çiçek Tozlaşma Algoritması (FPA) ve Guguk Kuşu Arama Algoritmasını (CSA) kullanarak KSA’larda konum belirleme problemini çözmüşlerdir. Ayrıca algoritmaları kuantum yöntemi ile inceleyerek farklı sonuçlar almışlardır. Eren, (2019) KSA’larında toplanan bilgilerin mevcut konumları ile alındığında faydalı olduğunu belirtmiş olup düğümler arası işbirliğine dayalı konum bulmanın önemini anlatmıştır. Bayrakdar, (2020) KSA’larda gecikmeye duyarlı CSMA ortam erişim yönteminin performansını incelemiştir. Bayrakdar, (2019a) KSA’larda kullanılan TDMA erişim yönteminin karasal ağlarda daha ön planda olduğunu ortaya koymuştur. Bayrakdar, (2019b) yaptığı çalışmada yeraltı KSA’larda bulanık mantık yöntemini, toplayıcı istasyon karar işlemleri için önermiştir. Bu yöntem ile toplayıcı istasyon için en uygun seçimin yapıldığını ve enerji tüketiminin en az seviyede olduğunu belirtmiştir. Bayrakdar, (2019c) yaptığı çalışmada KSA’da bir bölgeyi en az sayıda düğüm ile kapsamak için farklı bir yöntem sunmuş olup başarı sağlamıştır. Bayrakdar, (2019d) yeraltı KSA’larda veri iletimi için sezme tabanlı CSMA erişim yöntemini kullanmış ve enerji tüketimini düşük seviyede tutarak performansı arttırdığını belirtmiştir. Bayrakdar, (2019e) yaptığı çalışmada yeraltı KSA’larda derinlik faktörünün sinyal iletiminde neden olduğu kayıpları incelemiş ve farklı sinyal gönderme tekniklerine dikkat çekmiştir. Bayrakdar, (2019f) sualtı KSA’larda ortam erişim yöntemi olarak Aloha tekniği seçmiş olup bu tekniğin düşük maliyetli ve daha etkin olduğunu belirtmiştir.

KSA’larda bazı uygulamalarda sensörler tarafından toplanan bir bilginin konumunun bilinmemesi o bilgiyi anlamsız kılar. Aldığımız bilginin konumu ile birlikte öğrenilmesi uygulama açısından çok önemlidir. Bu çalışmada toplanan kısıtlı bilgilerle sensör düğümlerinin gerçek konumu tahmin edilmeye çalışılmıştır. Konum tespiti yapılırken sezgisel algoritmalarından PSO, YA, DGA ve AA yöntemleri kullanılarak performansları karşılaştırılmıştır.

Bu çalışmanın ikinci bölümünde KSA 'larda konum belirleme işleminin nasıl hesaplandığı anlatılmıştır. Üçüncü bölümde sezgisel algoritmalar dört başlık altında incelenmiştir. Dördüncü bölümde algoritma parametreleri ve konum tahminindeki hata hesabı anlatılmıştır. Beşinci bölümde sezgisel algoritmaların probleme uygulanması sonucu edinilen bulgular incelenmiştir. Sonuç bölümünde öneriler sunulmuştur.

2. KSA'larda Konum Belirleme Problemi

KSA'larda konum belirleme problemi mevcut bir KSA'da konumunu bilmediğimiz sensörlerin yerinin tespit edilmesi işlemidir. KSA'larda sensörlerin yerini düşük hata ile tespit etme işlemi önemli bir husustur. Konum belirleme için farklı ölçüm teknikleri kullanılır. Uzaklığa dayalı ölçüm metodu kullanılan yöntemlerden biridir. Bu yöntemde yerini bulmak istediğimiz sensörün komşularına olan mesafesi hesaplanır ve en az 3 adet konumunu bildiğimiz sensör aracılığı ile bu sensörün yeri tespit edilebilir (Doğru vd., 2019).



Şekil 1. KSA'da Uzaklığa Dayalı Konum Belirleme

KSA'larda konum belirlenirken referans düğümler m adet ve hedef sensörler n adet kabul edilir. Referans düğümlerin koordinatları (x_i, y_i) olur.

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

Tahmin edilen hedef düğümlerin 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 hesaplanan mesafeye gerçek uzaklık (d_i) denir. Hedef düğümün koordinatları (x, y) , i . referans düğümün koordinatları (x_i, y_i) olduğunda eşitlik aşağıdaki gibidir.

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

Hesaplanan gerçek uzaklığa gürültü (n_i) eklenir ve yaklaşık uzaklık (d_{in}) hesaplanır. Hedef sensör ile i . referans düğüm arasındaki yaklaşık uzaklığın denklemi aşağıdaki gibidir.

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

KSA'larda konum tespiti bir optimizasyon problemidir. Hedef gerçek ile tahmini olarak hesaplanan uzaklığın arasındaki farkı yani hata oranını en aza indirmektir. Belirlenen hedef 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

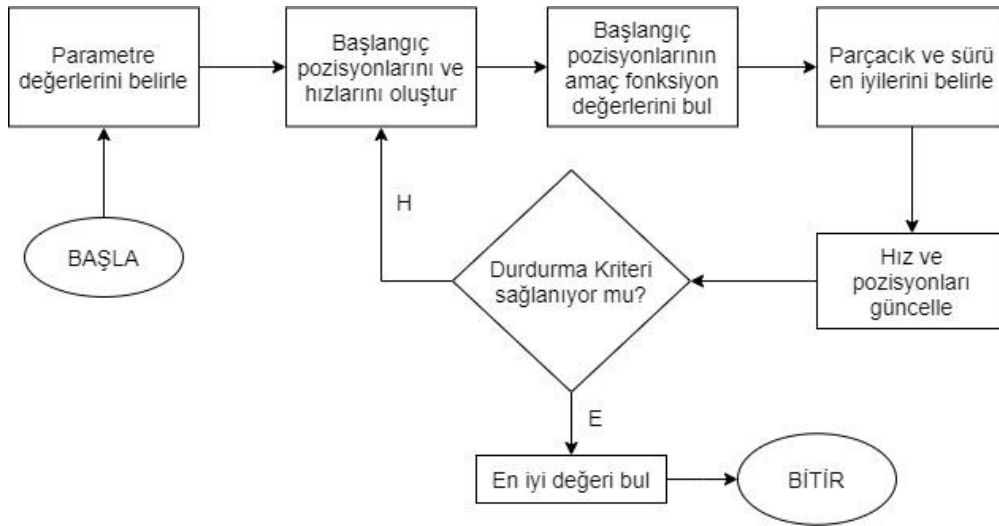
3.1. Parçacık Sürü Optimizasyonu (PSO)

Parçacık Sürü Optimizasyonu (PSO), Kennedy ve Eberhart (1995) tarafından keşfedilen balıklar ile kuşların sürü şeklindeki davranışından esinlenen bir optimizasyon yöntemidir. Genellikle sürü zekâsını ilke edinir. PSO 1995'te literatüre sunulduktan sonra üzerinde birçok iyileştirmeler yapılmıştır. Temel PSO 'da bulunan çoğu değişiklik PSO'nun yakınsama hızını artırmaya ve sürü davranışının farklılaşmasına yöneliktir (Karaboğa, 2004).

PSO algoritması amaç fonksiyonundaki değişimleri hesaplamak için matematiksel işlemlere ihtiyaç duymayan bir optimizasyon algoritmasıdır. Algoritmanın ilk keşfedilen bölümünde sadece lineer olmayan optimizasyon problemleri için çözüm bulunabiliyordu. Daha sonra metod geliştirilerek daha kompleks mühendislik problemlerini çözmek için uygulandı.

PSO 'nun ortaya çıkışı kuş ve balık sürülerinin hareketi ile olmuştur. Kuşların hareket ettikleri ortamda, konumunu bilmedikleri bir cisim gelişi güzel hareket ederek arayıp bulmaları, bir sorunu çözmeye benzer. Kuşlar bu cisim bulurken o cisme en yakın duran kuşa yakın olmak isterler. PSO 'nun temeli, sürüdeki kuşlar arasında gerçekleşen veri iletişimini iletlemektir. Bu algorithmada bireylere parçacık ismi verilir. Parçacıklardan oluşan gruba sürü denir (Kennedy ve Eberhart, 1995).

PSO 'da bulunan bir birey, önceki davranışlarından esinlenerek yerini sürünün en iyisine doğru belirler. PSO 'daki amaç, sürüdeki herkesin sürü en iyisine olan yaklaşma isteğidir. Her birey en iyiye yaklaşmaya çalışır. Bu yaklaşma olayı rastgele gelişir. Sürü içerisinde bulunan bir bireyin sonraki hareketinde gideceği yer genellikle önceki yerinden daha iyi olur. Sürüdeki hareketler istenen duruma gelinceye kadar devam eder. Şekil 2'de PSO 'nun akış diyagramı görülmektedir.



Şekil 2. PSO Akış Diyagramı

Algoritmanın, problem boyutu, popülasyon sayısı, atalet ağırlığı gibi belli başlı parametreleri vardır. Başlangıç iterasyonunda rastsal değerler alınarak süreç başlatılır. Alt ve üst sınır değerleri arasında rastgele değerler alınarak başlangıç popülasyonu oluşturulur ve amaç fonksiyon değerleri hesaplanır. Sonra karşılaştırmalar yapılarak sürü en iyisi bulunur. Sonraki adım hız vektörünün hesaplanmasıdır. Hız vektörü işlemi denklem 6 'daki gibidir. Hız vektörü hesaplandıktan sonra bazı bireylerde alt ve üst sınırlar aşılabılır. Aşan değerler alt ve üst sınır değerlerine ayarlanır (Çetin, 2011).

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

Denklemden yer alan r_1 ve r_2 , 0,1 aralığında dağılmış rastgele bir sayıyı, c_1 ve c_2 ise kişisel öğrenme ve global öğrenme değişken değerlerini ifade eder. Sonraki işlem bireyin önceki konumu ile hesaplanan hız vektörünün toplanarak yeni konumun bulunması işlemidir (Çetin, 2011). Pozisyon güncelleme denklemi eşitlik 7'deki gibidir.

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

Pozisyon güncelleme adımından sonra bazı bireylerde alt sınır ve üst sınır değerleri aşılabılır. Sınırları aşan değerler alt ve üst değerlere ayarlanır. Yeni oluşan konumların amaç fonksiyon değerleri hesaplanır. Sürü ve parçacık en iyileri yeniden bulunur. Bu işlemler durdurma kriterine kadar devam eder.

3.2. Yarasa Algoritması (YA)

Yarasa Algoritması (YA), Xin-She Yang tarafından 2010 yılında optimizasyon problemlerini çözmek amacıyla literatüre sunulmuştur. YA, yarasaların avlarını bulurken ve hareket ederken kullanmış oldukları yer belirleme yöntemini kullanır. Yarasa çıkardığı sesin nesnelere çarparak yankı yapması ve yankının geri dönüş zamanı ile konum belirler. Bu işleme ekolokasyon denir ve bir çeşit sonar olarak çalışır. YA yarasaların bu davranışından ilham alınarak geliştirilmiş bir sezgisel optimizasyon algoritmasıdır (Yang, 2010).

Ekolokasyon denilen iletişim ağı sayesinde yarasalar avlanır ve kendi aralarında iletişim kurarlar. Yarasa eko oluşturmak için yüksek ses dalgaları yayar ve yayılan bu dalgalar çeşitli seviyelerde geri döner. Yarasa bu ses seviyelerini algılayarak nesnenin konumunu belirler. YA 'nın yapısını geliştirmek ve yarasaların ekolokasyon özelliklerini gözlemlemek için bazı adımlar uygulanır (Ekinci, 2015):

- (a) Yarasalar avlarını veya tehlikeli engelleri tespit etmek için ekolokasyon özelliğini kullanır.
- (b) Yarasaların avını yakalaması için; v_i hızında, x_i pozisyonunda, sabit f_{min} frekansında, r değişkenli dalga boyunda ve L_0 ses şiddeti ile gelişigüzel hareket ederler. Hedef nesne ile aralarındaki uzaklığa göre yaydıkları sinyallerin frekanslarını ayarlarlar. 0 ile 1 arasında r sinyal yayılım oranlarını düzenlerler.
- (c) Yarasaların sinyal şiddeti, frekansı ve sinyal yayılım oranları değişiklik gösterir.
- (d) L_m^{iter} sinyal şiddeti, L_0 üst sınırdan L_{min} alt sınıra kadar değişebilir.

Optimizasyon boyunca her bireyin x_i pozisyonu ve v_i hızı, t zamanında x_i^t güncel pozisyon değerleri ve v_i^t güncel hızları aşağıda bulunan denklemlerden hesaplanır.

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

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

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

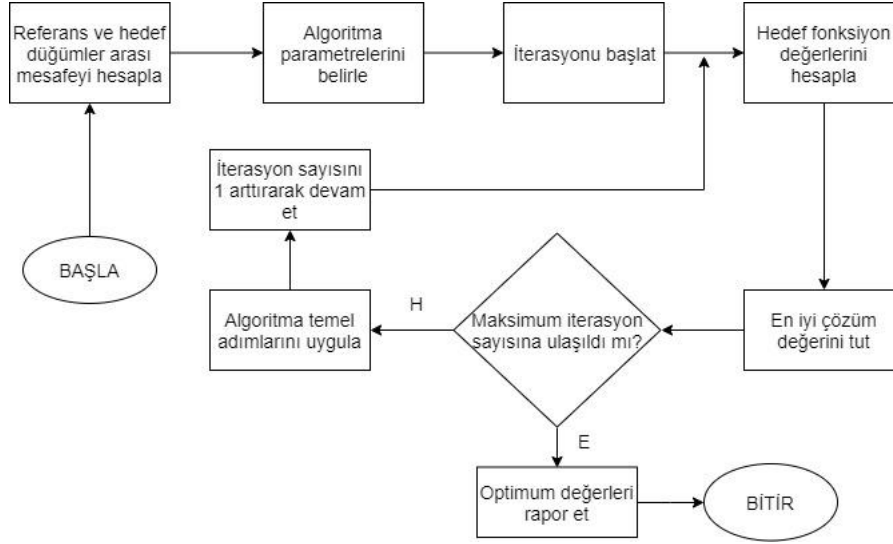
Denklem 8'de a , 0 ile 1 arasında rastgele alınmış bir vektörü, f_i i. yarasanın frekansı f_{min} ve f_{max} en küçük ve en büyük frekansları ve x^* popülasyondaki en iyi uygunluk değeridir. Hesaplanan sonuçlar içinden en iyi uygunluk değeri seçilir ve rastgele yapılan hesaplamalar sonunda yeni uygunluk değeri bulunur.

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

Denklem 11'de ε , 1 ile -1 arasında rastgele alınmış bir sayıyı ve L^t tüm bireylerin t zaman aralığındaki ortalama ses şiddetini ifade eder. Yarasaların ekolokasyon ile çıkardıkları sinyal şiddeti ve sinyal yayımlarının iterasyon sayısı ilerledikçe ve belirlenen hedefe yaklaşıldıkça tekrar hesaplanması gereklidir. Ses şiddeti (L), birey avına ulaştığında çoğunlukla azalır fakat sinyal yayılım oranı (r) yükselişe geçer.

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

Denklem 12'de β , 0 ile 1 arasında bir sabit sayı ve γ pozitif sabit değerdir. $t \rightarrow \infty$ iken ses şiddeti $L_i^t \rightarrow 0$ ve $r_i^t \rightarrow r_i^0$ olarak gerçekleşir. Şekil 3'te YA 'nın akış diyagramı görülmektedir.



Şekil 3. Yarasa Algoritması Akış Diyagramı

3.3. Diferansiyel Gelişim Algoritması (DGA)

Diferansiyel Gelişim Algoritması (DGA) optimizasyon problemleri için Price ve Storn tarafından 1996 yılında geliştirilmiş kullanımı kolay, tekniği sağlam bir optimizasyon algoritmasıdır. Bu algoritma doğrusal olmayan ve türevi alınamayan sürekli fonksiyonları minimize etmek amacıyla yeni bir yaklaşım olarak sunulmuştur. Diğer optimizasyon metodlarından daha hızlı ve sonuca daha kesin bir şekilde ulaştığı belirtilmiştir (Storn ve Price, 1997).

DGA daimi verilere sahip problemlerde etkili çözümler sunan çalışma mantığı ile Genetik Algoritma (GA) 'ya benzeyen bir optimizasyon tekniğidir. Popülasyon tabanlı çalışır. (Keskintürk, 2006)

DGA sayısal optimizasyon yöntemleri için geliştirilmiş bir algoritmadır. Bu algoritma öncekilerden farklı olup kolay olmayan etkin bir mutasyon yöntemi uygulamaktadır. Önceleri tanımlı olan, olasılık dağılım fonksiyonuna dayalı olarak çalışan GA gibi gelişim tabanlı algoritmaların aksine DGA gelişigüzel olarak seçilen amaç vektör çiftlerinin farklarına dayanan bir mutasyon yöntemi kullanır. DGA 'da uygulanan basit mutasyon yöntemi algoritmanın başarısını arttırmakta ve onu daha güçlü yapmaktadır (Karaboğa, 2004).

DGA, diğer sezgisel algoritmalarından farklı olarak yeni popülasyon oluştururken önce mutasyon operatörü ile uygulamaya başlar. Mutasyon ve çaprazlama operatörü ile yeni nesiller oluşturulur. Mutasyon sayısı aynı olasılık dağılım fonksiyonu ile belirlenmez aksine mevcut popülasyon bireyleri arasındaki farklılıklardan oluşan fark vektörü aracılığıyla bulunur (Kurtulmuş, 2019).

Algoritmanın, boyut yani gen sayısı, popülasyon sayısı, çaprazlama oranı, ölçekleme faktörü gibi parametreleri bulunmaktadır. Popülasyon sayısı ve gen sayısı uygulayıcı tarafından belirlenir. Başlangıç popülasyonu alt ve üst sınır değerleri arasında rastgele olarak belirlenir. Belirlenen popülasyondaki her bir değer için amaç fonksiyon karşılığı hesaplanır.

$$\forall i \leq NP \wedge \forall j \leq D : x_{j,i,G=0} = x_j^{(a)} + rand_j[0,1] \cdot (x_j^{(u)} - x_j^{(a)}) \quad (13)$$

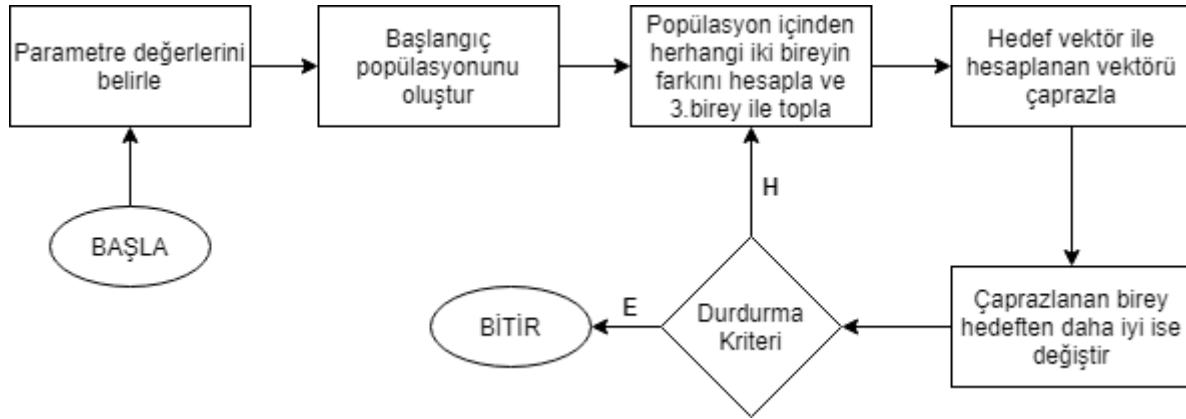
Denklem 13 'te yer alan NP popülasyon büyüklüğü, D gen sayısı, $x_{j,i,G}$ G jenerasyonunda i kromozomunun j geni, $x_j^{(u)}$ ve $x_j^{(a)}$ değişkenlere ait alt ve üst sınırları ifade eder. Daha sonra algoritma mutasyon adımı ile devam eder. Mutasyon işlemi için mutasyona uğrayacak kromozom hariç hepsi farklı rastgele üç adet kromozom seçilir. Kromozomlardan ikisinin farkı alınarak ölçekleme faktörü (F) ile çarpılır ve üçüncü kromozom ile toplanır (Keskintürk, 2006). Mutasyon işlemi denklem 14 'te gösterilmektedir.

$$\forall j \leq D : n_{j,i,G+1} = x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \quad (14)$$

Çaprazlama, mutasyon sonucu elde edilen kromozom ile mevcut kromozomdan birer gen alınarak yeni bir kromozom üretilmesi işlemidir. Algoritmada bu işlemi yapmak için CR (Çaprazlama Oranı) kullanılır. 0 ile 1 arasında rastgele üretilen sayı CR 'den küçük olursa gen mutasyona uğramış kromozomdan, aksi olursa mevcut kromozomdan alınır. Hedef belirlenen oranda genin mutasyona uğramış kromozomdan alınmasıdır (Keskintürk, 2006).

$$\forall j \leq D : n_{j,u,G+1} = \begin{cases} x_{j,n,G+1} & \text{eğer } rand[0,1] \leq RC \vee j = j_{rand} \\ x_{j,i,G} & \text{aksi durumda} \end{cases} \quad (15)$$

Daha sonra oluşturulan bu deneme kromozomunun amaç fonksiyon değeri hesaplanır. Bulunan değer mevcut kromozomdan daha iyi ise deneme kromozomu mevcut kromozomun yerine popülasyona atanır.



Şekil 4. Diferansiyel Gelişim Algoritması Akış Diyagramı

3.4. Ateşböceği Algoritması (AA)

Ateşböceği algoritması (AA), ateş böceklerinin karakteristik özelliklerinden esinlenerek 2008 yılında Xin-She Yang tarafından optimizasyon problemlerini çözmek amacıyla ortaya atılmıştır. Algoritma ateşböceğinin ışık verme özelliğine göre hareket etmesi esasına dayanır. Çekicilik parlaklıkla orantılı olup parlaklığı az olan ateşböceği parlaklığı daha fazla olana doğru hareket eder. Parlaklığı kendisinden fazla ateşböceği olmayan ortamda ateşböceği gelişigüzel hareket eder (Yang, 2008).

Son zamanlarda yapılan incelemelerde AA 'nın diğer algoritmalara göre daha etkin sonuçlar verdiği tespit edilmiştir. Algoritmanın çalışma prensibi herhangi bir çözüm kümesi kullanması ve denemeler boyunca ateşböcekleri arasındaki etkileşimi kullanarak sonuca gitmesidir. Tek boyutlu problemlerin yanı sıra çok boyutlu optimizasyon problemlerinde de iyi sonuçlar vererek uygulanabilmektedir (Demirdelen, 2018).

Ateşböceklerinin birbiri ile olan etkileşimi mesafeye bağlıdır. Mesafe arttıkça birbirleriyle olan etkileşim azalır. Bu etkileşimde görülen çeşitlilik ateşböceklerinin çok yönlü hareket etmesini sağlar. Ateşböceklerinin çok yönlü olması hareketliliğin artmasına neden olur ve arama uzayının daha etkin bir şekilde araştırılmasını sağlar. Algoritmada bulunan önemli sınırlamalar ile daha hassas ayarlamalar yapılabilir (Gandomi vd., 2011).

Algoritmanın popülasyon sayısı, boyutu, çekicilik katsayısı, ışık emme katsayısı gibi parametre değerleri belirlenir. Başlangıç popülasyonu alt ve üst limit değerleri arasında rastgele olarak belirlenir. Belirlenen popülasyondaki her bir değer için amaç fonksiyon değeri hesaplanır.

$$x_i = x_{min} + r(x_{max} - x_{min}) \quad (16)$$

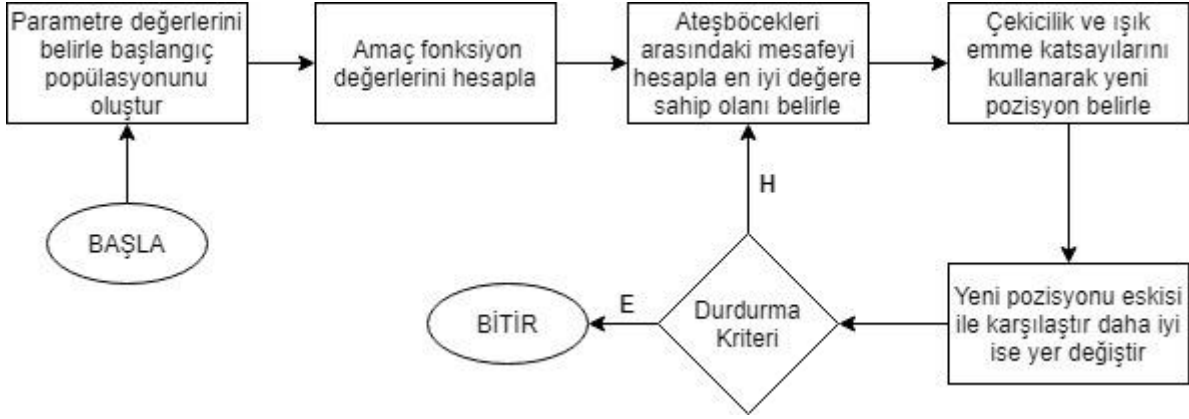
Denklem 16'da r, 0 ile 1 arasında rastgele seçilen bir sayıyı, x_{min} ve x_{max} alt ve üst sınır değerlerini ifade eder. Sıradaki işlemde ise i ve j noktalarındaki herhangi iki bireyin arasındaki mesafe denklem 17 kullanılarak hesaplanır.

$$r_{ij} = \|x_i - x_j\| \quad (17)$$

İki birey arası mesafe hesaplandıktan sonra ateşböceğinin yeni konumu denklem 18'e göre hesaplanır.

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + a_t \varepsilon_i^t \quad (18)$$

Denklemdaki β_0 çekicilik katsayısı, γ ışık emme katsayısı, x_j^t ve x_i^t i ve j noktalarında bulunan bireyleri, a_t 0 ile 1 arasında rassal bir sayıyı, ε_i^t alt üst sınırlara göre belirlenen bir değişkendir. x_i^{t+1} bireyin yeni konumu olup amaç fonksiyon değeri yeniden hesaplanır ve ilk konumuna göre daha iyi bir sonuca ulaşılır ise yeni konum olarak atanır.



Şekil 5. Ateşböceği Algoritması Akış Diyagramı

4. Simülasyon

Bu kısımda KSA'larda sensörlerin konumunun belirlenmesinde PSO, YA, DGO ve AA sezgisel optimizasyon yöntemleri kullanılmıştır. Kullanılan bu yöntemlerin performansları karşılaştırılmıştır.

4.1. Parametre Değerleri

Algoritmalarda kullanılan parametre değerleri Tablo 1, 2, 3 ve 4'te sunulmuştur. Algoritmalarda ortak olan parametreler 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)	0.8
Sinyal Emisyon Hızı (r)	0.25
İterasyon Sayısı	100

Tablo 3. Diferansiyel Gelişim Algoritması Parametreleri

Parametreler	Değerler
Boyut Sayısı	2
Popülasyon Sayısı	20
Ölçekleme Faktörü alt sınırı (F_{min})	0.2
Ölçekleme Faktörü üst sınırı (F_{max})	0.8
Çaprazlama Oranı (CR)	0.2
İterasyon Sayısı	100

Tablo 4. Ateşböceği Algoritması Parametreleri

Parametreler	Değerler
Boyut Sayısı	2
Popülasyon Sayısı	20
Çekicilik Katsayısı (β_0)	2.0
Işık Emme Katsayısı (γ)	1.0
İterasyon Sayısı	100

4.2. Hata Hesaplaması

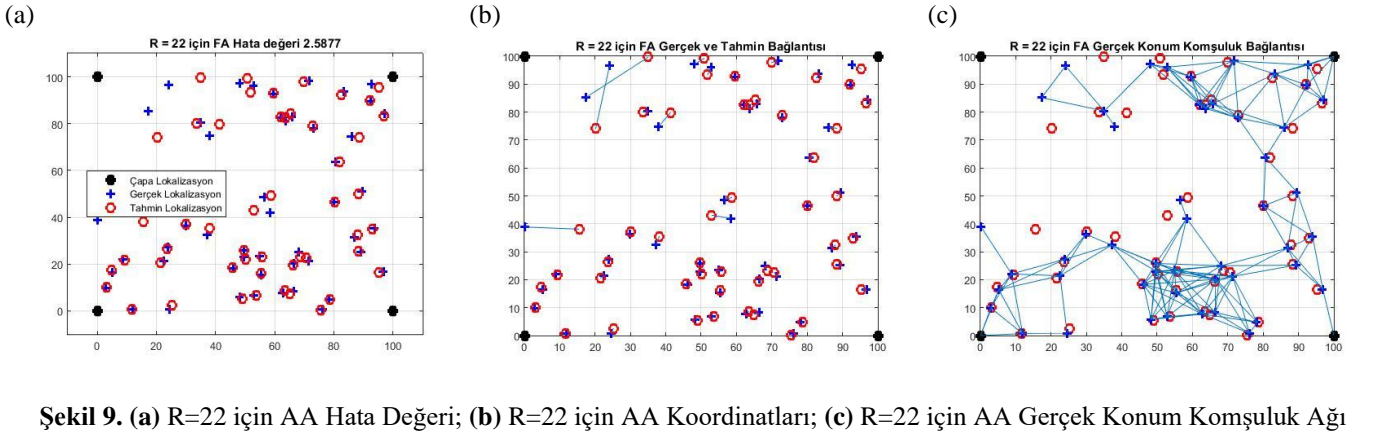
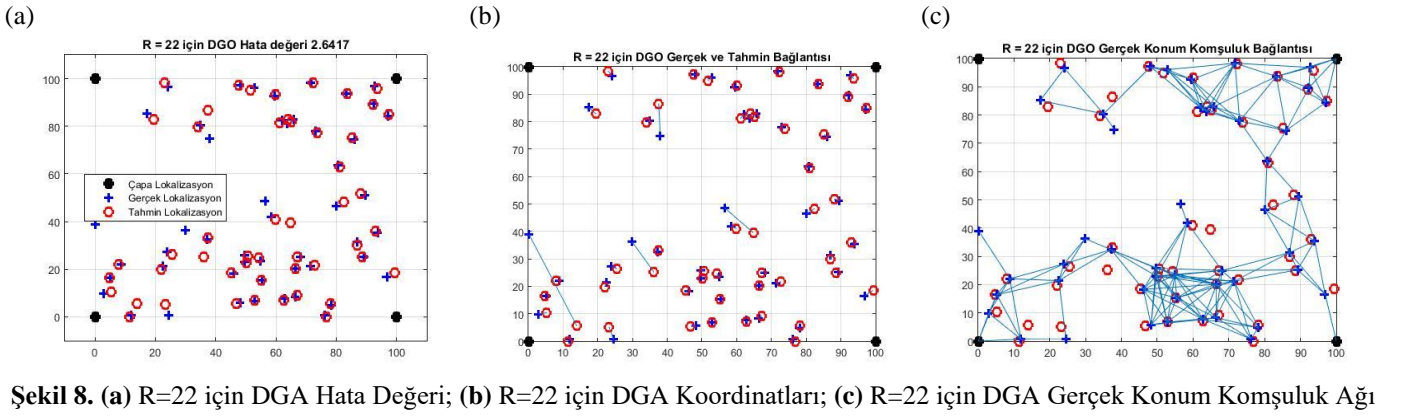
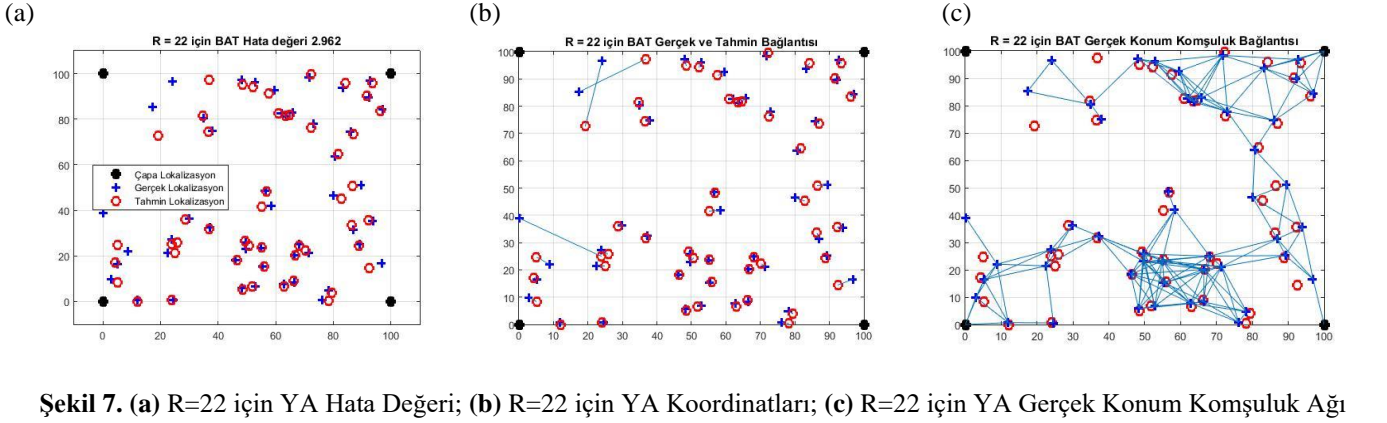
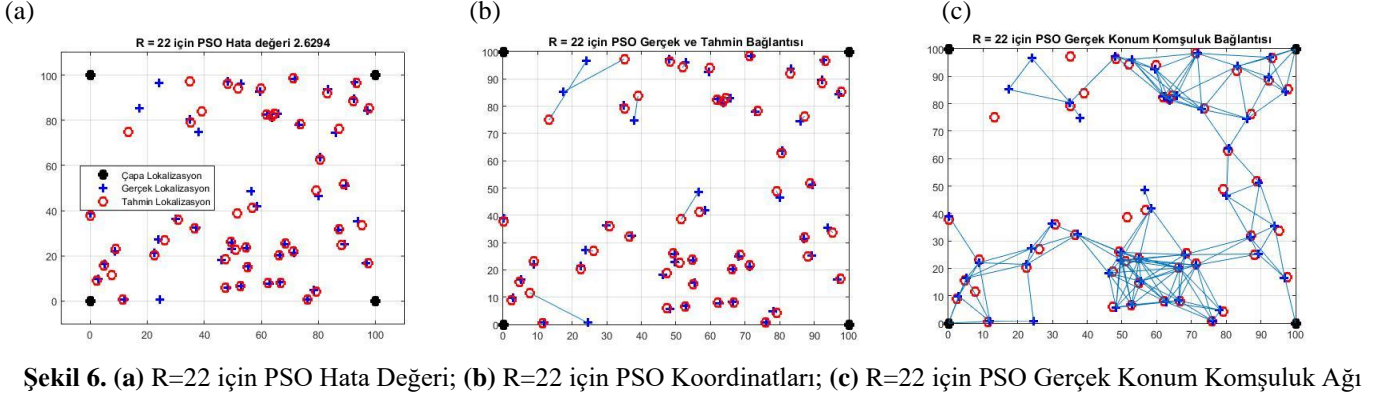
KSA’larda bir sensörün konumunu belirleyebilmek, o sensörün üç veya daha fazla referans sensörü ile iletim yarıçapı içerisinde haberleşebilmesi durumunda mümkün olabilir. Bu sebeple, daha çok sensörün yerini belirlemek için, sensörlerin iletim yarıçapı mesafesi büyük olmalıdır. Bu çalışmanın amacı yerini bilmediğimiz sensörlerin konumunu tespit etmek için optimizasyon algoritmaları kullanmaktır. Algoritmanın tahmin ettiği sensörler ile gerçek sensörlerin konumları arasındaki hatayı hesaplamak için eşitlik 13’teki hata fonksiyonu kullanılmıştır.

$$E = \frac{1}{N} \sum_{i=1}^N \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2} \quad (19)$$

(N) hedef sensör sayısını, (X_i, Y_i) algoritma tarafından tahmin edilen sensör koordinatlarını, (x_i, y_i) ise gerçek sensör koordinatlarını belirtmektedir.

5. Bulgular

MATLAB programında 100x100 birimlik alanda köşelere yerleştirilmiş 4 adet referans düğüm ile 50 adet düğüm noktasının yer tespiti PSO, YA, DGA ve AA algoritmaları ile yapılmıştır. Her bir sensör düğümünün kapsama alan yarıçapı $R = 22$ birim alınmıştır. Tüm algoritma aynı düğüm koordinatlarının tespiti için çalıştırılmıştır. Simülasyon 1’e ait grafikler Şekil 6,7,8,9’da gösterilmiştir. Şekil 6a’da PSO için $R=22$ birim alınarak yapılan simülasyon görülmektedir. Burada mavi artı işaretleri, düğümlerin gerçek konumlarını belirtmektedir. Kırmızı yuvarlak işaretler ise düğümlerin PSO tarafından tahmin edilen konumlarını göstermektedir. Siyah yuvarlaklar, referans düğümlerinin (çapa) yerlerini göstermektedir. Şekil 6b’de düğümlerin gerçek konumları ile tahmin edilen konumları arasındaki uzaklıklar, açık mavi çizgilerle gösterilmiştir. Şekil 6c’de ise, $R=22$ için gerçekleşen ağ çizgesi (network graph) görülmektedir. Şekil 7a, 7b, 7c’de Yarasa Algoritmasının, $R=22$ birimde verdiği sonuçlar; Şekil 8a, 8b, 8c’de Diferansiyel Gelişim Algoritmasının, $R=22$ birimde verdiği sonuçlar; Şekil 9a, 9b, 9c’de ise Ateşböceği Algoritmasının $R=22$ birimde verdiği sonuçlar gösterilmiştir. Simülasyon, $R=22$ birim olacak şekilde, PSO, YA, DGA ve AA için 50 kere tekrarlanmıştır. Hata oranlarının ayrıntılı tablosu on adet simülasyon için (Simülasyon 1’den, Simülasyon 10’a kadar) Tablo 5’te görülmektedir. Elli adet simülasyon için ortalama değerleri gösteren tablo ise Tablo 6’da sunulmuştur.



Tablo 5. Hata Oranları Tablosu (10 adet simülasyon)

	PSO	YA	DGA	AA
İletim Mesafesi (R)	R=22	R=22	R=22	R=22
Simülasyon 1	2,6294	2,962	2,6417	2,5877
Simülasyon 2	2,7868	3,3285	1,8901	3,2843
Simülasyon 3	3,0692	2,714	2,5328	2,2199
Simülasyon 4	2,2885	3,2323	2,9424	1,8733
Simülasyon 5	3,8974	2,8876	2,6629	3,1751
Simülasyon 6	2,6631	2,2618	2,7089	2,9967
Simülasyon 7	2,6077	3,303	3,2394	2,4364
Simülasyon 8	2,721	3,2369	3,3009	3,159
Simülasyon 9	2,3983	3,1865	2,6908	2,8278
Simülasyon 10	2,959	2,4231	3,6857	2,9558
Ortalama	2,80204	2,95357	2,82956	2,7516

Tablo 6. 50 adet simülasyon için ortalama hata oranları tablosu

İletim Mesafesi (R)	R=22
PSO	2,622256
YA	2,97842
DGA	2,634152
AA	2,595904

6. Tartışma ve Sonuç

Algoritmaların kendine özgü parametreleri birbirine yakın seçilmiş olup hepsinde ortak olan boyut, popülasyon ve iterasyon sayıları aynı alınmıştır. Algoritmaları ortalama en düşük hata oranına göre sıraladığımızda sırasıyla AA, PSO, DGA ve YA iyi sonuç verdiği görülmektedir. Tablo 5'te görülen ayrıntılı verilerde, bazı simülasyonlarda YA, DGA ve PSO'nun AA'dan daha iyi sonuç verdiği görülmüştür. Ancak, hem 10 simülasyon için, hem de 50 simülasyon için, ortalama olarak AA diğer üç algoritmadan daha iyi sonuç vermektedir. PSO, DGA ve AA'nın ortalama hata oranları birbirine yakın fakat YA'nın hata oranının daha yüksek olduğu gözlenmektedir.

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, YA, DGA ve AA kullanılmıştır. Hata oranlarını incelediğimizde Ateşböceği Algoritması ile alınan sonuçların daha iyi olduğu gözlemlenmektedir.

KSA'larda konum belirleme işlemi günümüzde birçok uygulamada kullanılmaktadır. Bu çalışmadan ilham alınarak farklı optimizasyon, yapay zeka ve analiz teknikleri kullanılarak daha performanslı yeni yöntemlerin keşfedilmesi yararlı olacaktır.

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, 2008 IET International Conference on Wireless, Mobile and Multimedia Networks, Beijing, 2008, pp. 227-230, doi: 10.1049/cp:20080185.
- 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.

- Bayrakdar, M.E. (2020). Kablosuz Algılayıcı Ağlar için Gecikme Duyarlı CSMA Ortam Erişim Tekniğinin Performans Değerlendirmesi, Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, 12(1): 227-235, 2020.
- Bayrakdar, M.E. (2019a). Karasal algılayıcı ağlarda gözlemlene için enerji etkin TDMA erişim tekniği, Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 21(2): 756-765, 2019.
- Bayrakdar, M.E. (2019b). Yeraltı Kablosuz Algılayıcı Ağlar için Bulanık Mantık Tabanlı Toplayıcı İstasyon Karar Yaklaşımı, Gümüşhane Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 9(4): 789-796, 2019.
- Bayrakdar, M.E. (2019c). Kablosuz Algılayıcı Ağlarda En Az Sayıda Düğüm Kullanımı için Maliyet Etkin Algılayıcı Düğüm Yerleştirme Yaklaşımı, Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi, 6, 59-73, 2019.
- Bayrakdar, M.E. (2019d). Yeraltı Algılayıcı Ağlarda Kayıpsız Veri İletimi için Sezme tabanlı Ortam Erişim Tekniğinin Başarım Analizi, Erzincan Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 12(2): 1028-1035, 2019.
- Bayrakdar, M.E. (2019e). Kablosuz Yeraltı Algılayıcı Ağlar için Düğüm İletişiminde Derinlik Faktörünün Analizi, Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, 27(2): 93-99, 2019.
- Bayrakdar, M.E. (2019f). Sualtı Kablosuz Algılayıcı Ağlarda Aloha tabanlı Maliyet Etkin Ortam Erişim Protokolü, Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, 19(1): 114-120, 2019.
- 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.
- Demirdelen, T. (2018). Kuru Tip Transformatör Optimizasyonuna Yeni Bir Yaklaşım: Ateş Böceği Algoritması, Çukurova Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 33(1): 87-96, 2018.
- Doğru A.S., Temel B., Eren T. (2019). Kablosuz Sensör Ağlarında Konum Belirlemede Parçacık Sürü Optimizasyonu ve Yarasa Algoritması Yöntemlerinin Karşılaştırılması, Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, 11(3): 793-801, 2019
- Ekinci 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., (2019). Conditions for Unique Localizability in Cooperative Localization of Wireless ad hoc and Sensor Networks. Cooperative Localization and Navigation: Theory, Research, and Practice: 31-48. Gao, C. (Ed.), Zhao, G. (Ed.), Fourati, H. (Ed.). Boca Raton: CRC Press.
- 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.
- Gandomi, A. H., Yang, X.S. & Alavi, A. H. (2011). Mixed variable structural optimization using Firefly Algorithm, Computer and Structres, pp. 2325-2336, 2011.
- 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.
- Keskintürk T. (2006). Diferansiyel Gelişim Algoritması, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 5(9):85-99, 2006.
- Kızılkaplan E., Eren T., Yalçınkaya F. (2020). Kablosuz Sensör Ağlarında Konum Belirlemede Sezgisel Algoritmaların Kuantum Davranışları ile Karşılaştırılması, International Journal of Engineering Research and Development, 12 (2), 587-602. DOI: 10.29137/umagd.746589.

- Kurtulmuş, C. (2019). Diferansiyel Gelişim Algoritması İle Kardinalite Kısıtlı Portföy Optimizasyonu, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstatistik, 2019.
- 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.
- Storn, R. & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization, 11: 341-359, 1997.
- 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 (NCSO 2010), 284, 65-74, 2010.
- Yang, X. S. (2008). Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.
- 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