



Kablosuz Sensör Ağlarında Parçacık Sürü Optimizasyonu, Sürekli Zaman Karınca Koloni Algoritması ve Girdap Arama Algoritması Yöntemleri ile Konum Belirleme

Localization with Particle Swarm Optimization, Continuous Time Ant Colony Algorithm and Vortex Search Algorithm Methods in Wireless Sensor Networks

Feyza Nur Çakıcı¹ , Tolga Eren¹ 

¹ Kırıkkale Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Elektrik-Elektornik Mühendisliği Bölümü, 71450 Kırıkkale, TÜRKİYE

Başvuru/Received: 21/04/2021

Kabul / Accepted: 10/06/2021

Çevrimiçi Basım / Published Online: 18/06/2021

Son Versiyon/Final Version: 18/06/2021

Öz

Günümüzde kablosuz sensör ağlarının (KSA) kullanım alanları artması ile birlikte önemli bir konu haline gelmektedir. Doğada bulunan canlıların hareketlerinin izlenmesi, akıllı ev sistemleri, hava durumu tahminleri, askeri operasyonlar, sağlık alanları gibi birçok uygulamalarda sensör düğümleri kullanılmaktadır. KSA' ların kullanım alanları arttıkça sensörlerinin konumu doğru, hızlı ve güvenilir bir şekilde tespit edilmesi önemlidir. KSA' ların konumu belirlenmesi optimizasyon problemi olarak ele alınmıştır ve en az hata ile belirlenmesi için birçok farklı optimizasyon yöntemleri kullanılmıştır. Makalede Parçacık Sürü Optimizasyonu (Particle Swarm Optimization-PSO), Sürekli Zaman Karınca Koloni Algoritması (Ant Colony Optimization for Continuous Domains -ACO_R) ve Girdap Arama Algoritması (Vortex Search Algorithm-VSO) kullanılarak elde edilen sonuçlar karşılaştırılmıştır. Matlab ortamında test sistemine uygulanmış sonuçlar tablolar halinde ifade edilmiştir.

Anahtar Kelimeler

“PSO, ACO_R, VSO, Kablosuz sensör ağlar konum belirleme problemi”

Abstract

Today, with the increase in usage areas of wireless sensor networks (KSA), it is becoming an important issue. Sensor nodes are used in many applications such as monitoring the movements of living thing in nature, smart home systems, weather forecasts, military operations, health areas. As the usage areas of KSA' s increase, it is important to determine the location of the sensors accurately, quickly and reliably. Determining the location of KSA' s is considered as an optimization problem and many different optimization methods have been used to determine the with the least error. The article compares the results obtained using Particle Swarm Optimization (PSO), Ant Colony Optimization for Continuous Domains (ACO_R) and Vortex Search Algorithm (VSO). It is applied to the test system in Matlab environment and the results are expressed in tables.

Key Words

“PSO, ACO_R, VSO, Positioning problem in wireless sensor networks”

1. Giriş

Radyo Frekans (RF) ve Mikroelektronik Sistemlerin (MEMS) gelişmesiyle birlikte sensörlerin boyutları küçülmüş ve güç tüketimi azalmıştır.

Kablosuz sensör ağları (KSA) başlangıçta askeri alanlar için kullanılmış ve teknolojinin gelişmesiyle birlikte enerji, tarım, sağlık, çevre, ev ve ofis gibi birçok farklı alanda kullanılmaktadır. Ayrıca sensörlerin güvenilir olması, göreceli olarak daha az maliyetli olması, esnekliği, doğruluğu ve kurulum açısından kolay olması gibi avantajları sayesinde günümüzde çok tercih edilmektedir (Doğru, 2019). Kablosuz sensör ağlarında iletişim çift taraflı olduğu için veriyi alma, iletme ve yorumlama özellikleri bulunmaktadır.

KSA'da sensör düğümleri, izlenecek ve takip edilecek coğrafik alanlara rastgele dağıtılmakta ve elde edilen veriler üç aşamada gerçekleştirilmektedir. Birinci aşamada, sensör tarafından olaylar algılanmakta ve veriler her bir sensör tarafından ayrı ayrı işlenmektedir. İkinci aşamada, işlenen veriler komşu sensörlere iletilmektedir. Üçüncü ve son aşamada ise veriler baz istasyonuna gönderilmektedir.

KSA'da birden fazla düğüm bulunmaktadır. Bu düğümler, sensör (algılama) birimi, işlemci birimi, alıcı-gönderici (telsiz) birimi ve güç birimi olmak üzere dört ana bölümden meydana gelmektedir. KSA'da yer alan sensör düğümlerinin boyutları küçüktür ve daha az enerji kullanmaktadır. Sensör düğümleri, sensörler aracılığıyla veriyi toplayabilme, işleyebilme ve haberleşme yapabilme özelliklerine sahiptir. Sensör düğümlerinde, konumu bilinen ve konumu bilinmeyen düğümler bulunmaktadır. Konumu bilinen sensör düğümler "referans (çapa) düğüm", konumu bilinmeyen sensör düğümleri ise "hedef düğüm" olarak adlandırılmaktadır. Çapa düğümlerin sayısı hedef düğümlerin sayısına göre nispeten azdır. Çapa düğümlerin konumu GPS aracılığıyla veya manuel olarak belirlenebilmektedir. GPS tekniği hem maliyetlidir hem de enerji tüketimi fazladır. Ayrıca dağlar, bitki örtüsü veya binalar GPS uydularının görüntülenmesini engellediği için hedef düğümlerin konumun belirlenmesinde tercih edilmemektedir. Hareketli yapıya sahip sensörlerin ortama rastgele dağıtılması durumunda manuel olarak konum tespiti yapılamaz. Bu durumda hedef düğümlerin konumlarını belirlemek için GPS veya manuel yerine farklı yöntemler kullanılmaktadır. Konumu bilinmeyen hedef düğümlerin, çapa düğümleri ve hedef düğümleri arasındaki mesafe hesaplanarak konumu bulunmaktadır. Bu yöntem network localization (ağ konumu belirleme) olarak adlandırılmaktadır (Eren, et al., 2004).

KSA birçok alanda kullanıldığı için konum belirleme problemi önemli bir noktadır. Yapılan çalışmalarda KSA konum problemi bir optimizasyon problemi olarak yer almaktadır. Hedef düğümlerin konumunun belirlenmesinde farklı algoritmalar kullanılmıştır. Literatürde yer alan çalışmaların bir kısmına açıklamaları ile birlikte yer verilmektedir.

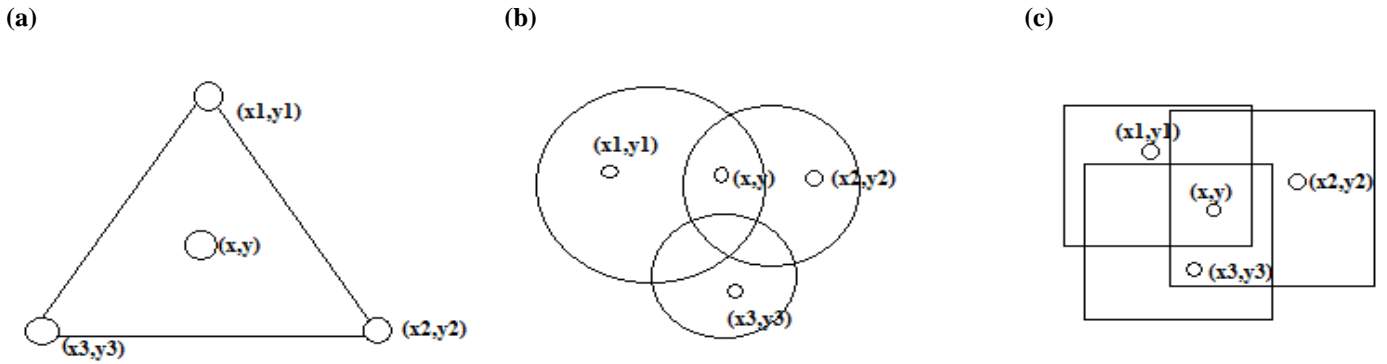
W. Chengqun, ve arkadaşları (2009) kablosuz sensör ağlarında konum belirleme sorunu için izomap algoritmasını ele almıştır. İzomap algoritması sensörler arasındaki farklılığı ölçmek için jeodezik mesafeyi kullanmıştır. Resul Doğan, (2018) yüksek lisans tezinde KSA ile konum belirlemede Sinyal Alım Gücü (RSS) değeri ile Yapay Sinir Ağ (YSA) yaklaşımını ele almıştır. YSA'nın RSS ile konum bulmada, Bounding Box'a göre daha iyi sonuçlar verdiği gözlenmiştir. Güngör Yıldırım, (2012) KSA konum belirlemede Bounding Box metodu kullanmıştır. H. Cui ve arkadaşları, (2018) büyük ölçekli kablosuz sensör ağlarında konum belirleme probleminde Niching Parçacık Sürü Optimizasyonu ve güvenilir çapa seçimini kullanmıştır. Bu çalışmada PSO ile daralma katsayısı (PSO with constriction coefficient-CPSO) kullanımının sağladığı avantaj sayesinde daha az iterasyon ile konum belirlenebilmiştir. Celal Öztürk, (2016) tarafından sunulan proje çalışmasında KSA ile konum belirlemede Yapay Arı Koloni Algoritması (Artificial Bee Colony Algorithm-ABC) kullanılmıştır. Projede PSO ile ABC algoritmaları karşılaştırılmıştır. ABC algoritması ile daha başarılı sonuçlar elde edilmiştir. A.S. Doğru ve arkadaşları (2019) tarafından yapılan çalışmada, KSA konum belirleme için PSO ve Yarasa Algoritmasının (Bat Algorithm-BA) simülasyon sonuçları karşılaştırılmış ve PSO ile daha iyi değerler elde edildiği gözlenmiştir. N. Bacanin ve arkadaşları Balina Optimizasyon Algoritması (Whale Optimization Algorithm-WOA) kullanarak KSA konum belirleme problemini iyileştirmeyi başarmıştır. S. Ivana ve arkadaşları (2018) ise KSA konum belirleme problemi için Fil Sürü Optimizasyon Algoritmasını (Elephant Herding Optimization Algorithm-EHO) kullanmıştır. EHO'nun sağlam ve verimli bir meta-sezgisel algoritma olduğu sonucuna varılmıştır. A. Kumar ve arkadaşları (2012) H-Best Parçacık Sürüsü Optimizasyon Algoritması (H-Best Particle Swarm Optimization Algorithm -HPSO) ve Biyoloji Tabanlı Optimizasyon (Biogeography-Based Optimization-BBO) Algoritmalarını kullanmıştır. KSA konum belirleme problemi için Genetik Algoritma ve Benzetimli Tavlam Algoritması (Simulated Annealing Algorithm-SAA) karşılaştırıldığında daha iyi sonuçlar elde edilmiştir. Son olarak S. Gayal ve M.S. Patterh, (2015) KSA konum belirleme sorunu için Çiçek Tozlaşma Algoritmasını (Flower Pollination Algorithm-FPA) kullanmış, çalışma PSO ile karşılaştırıldığında daha iyi konum belirleme doğruluğuna sahip olduğu gözlemlenmiştir.

Bu çalışmada KSA konum belirleme problemi için PSO, ACO_R ve VSO algoritmaları kullanılmıştır. PSO, ACO_R ve VSO algoritmalarında elde edilen sonuçlar karşılaştırılmıştır. Makalenin 2. bölümünde konum belirleme problemi ve hata fonksiyonu ele alınmıştır. 3. bölümde kullanılan optimizasyon algoritmaları hakkında bilgi verilmiştir. 4. Bölümde elde edilen sonuçlar şekil ve tablolar halinde gösterilmiş ve son olarak 5. bölümde ise yapılan çalışmaların sonuçları karşılaştırılmıştır.

2. KSA' larda Konum Belirleme Problemi ve Hata Fonksiyonu

KSA'lar birçok alanda uygulanabildiği için sensör düğümlerin konumlarının tespiti için önemli hale gelmiştir. KSA' da konumu bilinmeyen sensörlerin yerini tespit etmek için çeşitli metotlar kullanılmaktadır. GPS kullanmadan sensörlerin yeri tespit edilmektedir. Konum belirleme metotları, mesafeye bağımlı konum bulma ve mesafeden bağımsız konum bulma yöntemleri olmak üzere ikiye ayrılmaktadır. Mesafeye bağımlı ve mesafeden bağımsız konum bulma yöntemleri ise çapaya bağlı veya çapadan bağımsız olmak üzere iki farklı şekilde hesaplanmaktadır. Mesafeye bağımlı yöntemde sensör düğümleri arasındaki uzaklığı hesaplanmasıyla konum bulunmaktadır. Mesafe hesaplanmasında üçgen, daire ve dikdörtgen gibi geometrik yapılar kullanılmaktadır. Mesafeye bağımlı konum belirleme yönteminde elde edilen hata oranı daha azdır ve mesafe hesaplama iki süreçten meydana gelmektedir. Birinci süreç sensör düğümler ile çapa düğümleri arasındaki mesafenin hesaplanmasıdır.

İkinci süreç ise mesafe Time of Arrival (ToA), Time Difference of Arrival (TDoA), Gelen Sinyalin Gücüne Bağlı Mesafe Tahmini (RSSI), Angel of Arrival (AoA) ve İletim Aralığı yöntemleri kullanılarak hesaplanmaktadır. Mesafeden bağımsız konum belirleme yöntemi basit bir yapıdır ve daha az maliyete sahiptir ancak hata oranı yüksektir. Bu yöntemde konum Dv-Hop, Amorphus, Apıt ve Centroid yöntemleri kullanılarak bulunmaktadır. İletim aralığı diğer yöntemlerle kıyaslandığında daha basittir, ekstra bir donanım gerektirmediği için maliyeti daha azdır. İletim aralığı azaldıkça hata oranı artmaktadır. Hedef sensörlerin konumlarını belirlemek için hangi yöntemlerin kullanılması önemli bir noktadır.



Şekil 1.(a) Üçgen algoritmali mesafeye bağımlı konum belirleme; (b) Dairesel algoritmali mesafeye bağımlı konum belirleme; (c) Dikdörtgen algoritmali mesafeye bağımlı konum belirleme (Bekçibaşı & Tenruh, 2012)

Şekil 1'de mesafe hesaplanmasında kullanılan üçgen, daire ve dikdörtgen yapılar gösterilmiştir. Sensör düğümlerinin konumunu belirlemek için 2-boyutlu uzayda en az 3 tane, 3-boyutlu uzayda ise en az 4 tane konumunu bildiğimiz çapa (referans) düğümü gereklidir. Çapa düğümü ile hedef düğümlerin, komşularına olan mesafe hesaplanarak sensörlerin konumu belirlenmektedir.

KSA'larda konum belirleme probleminde M adet çapa düğümü ve N adet hedef düğümü olduğunu düşünürsek, çapa düğümlerin koordinatları (x_i, y_i) ve hedef düğümlerin koordinatları (x_j, y_j) şeklinde gösterilmektedir.

Gerçek uzaklık aşağıdaki denklem (1) kullanılarak hesaplanmaktadır.

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (1)$$

Bu denklemde (x, y) hedef düğümlerin, (x_i, y_i) ise çapa düğümlerin koordinatlarını ve d_i gerçek uzaklığı ifade etmektedir.

Tahmini uzaklık (d_{in}), gerçek uzaklığa (d_i) gürültü eklenerek bulunmaktadır. Tahmini uzaklık denklem (2) ile elde edilmektedir.

$$d_{in} = d_i + n_i \quad (2)$$

Denklem 2'de i . çapa düğümüne kadar elde edilen tahmini mesafenin denklemi gösterilmiştir.

Konum belirleme problemi makalede optimizasyon problemi olarak ele alınmıştır. Burada amaç, gerçek mesafe ile tahmini mesafe arasındaki farkı azaltmaktır. Denklem 3'te amaç fonksiyonu verilmiştir.

$$f(x, y) = \frac{1}{N} \sum_{i=1}^N (d_i - d_{in})^2 \quad (3)$$

Hata fonksiyonu (E) sezgisel algoritmalar kullanılarak minimize edilmektedir. Hata fonksiyonu, tahmin edilen düğümler ile gerçek düğümler arasında hesaplanmaktadır. Hata fonksiyonu pozitif bir değer almakta ve optimum değer hata fonksiyonunun sıfıra yaklaşması ile elde edilmektedir. Hata fonksiyonu denklem (4) ile hesaplanmaktadır.

$$E = \frac{1}{k} \sum_{i=1}^k \sqrt{(x_i - X_i)^2 + (y_i - Y_i)^2} \quad (4)$$

Denklemden k hedef düğüm sayısını, (x_i,y_i) gerçek düğümü ve (X_i,Y_i) tahmin edilen düğümü ifade etmektedir.

3. Konum Belirlemede Kullanılan Yöntemler

Kablosuz sensör ağlarında konum belirlemek için PSO, ACO_R ve VSO algoritmaları kullanılmış ve elde edilen sonuçlar karşılaştırılmıştır. PSO, ACO_R ve VSO algoritmaları hakkında detaylı bilgi aşağıda sunulan başlıklar altında verilmiştir.

3.1. Parçacık Sürü Optimizasyonu (PSO)

PSO algoritması Kennedy ve Eberhart (1995) tarafından balık ve kuşların yiyecek araması ve tehlikelerden kaçması sırasında sürü halinde hareket etmesinden esinlenerek geliştirilen bir algoritma türüdür. PSO algoritması sürü zekâsına dayanmaktadır. Kuş ya da balık popülasyonunda bulunan her bir bireye parçacık denmektedir. Parçacıkların meydana getirdiği topluluğa sürü adı verilmektedir. Yiyecek arama esnasında yiyeceğe en yakın kuş diğer kuşlar takip etmekte ve yiyeceğe en yakın kuş lider kuş olarak adlandırılmaktadır. Sürüde yer alan lider kuş değişebilmektedir. Parçacıklar arama uzayına rastgele dağıtılmaktadır. Parçacığın en iyi konumu yiyecek arama sırasında kendisinin önceki tecrübelerinden ve sürüde yer alan parçacıkların tecrübelerinden yararlanarak bulunmaktadır. Parçacıklar arasında sosyal veri paylaşımı bulunmaktadır.

PSO algoritması ilk önce doğrusal olmayan sürekli optimizasyon problemlerinin çözümünde kullanılırken yapılan modifikasyonlar ile daha karmaşık optimizasyon problemlerinin çözümünde kullanılmıştır. PSO algoritmasına yakınsama hızını iyileştirmek ve sürünün farklılaşmasını artırmak gibi modifikasyonlar uygulanmıştır.

Parçacıkların aşağıda ifade edilen denklem (5) ile başlangıç konumları yani ilk çözüm uzayı bulunmaktadır.

$$X_{i,d} = X_{min,d} + r_{i,d} \times (X_{max,d} - X_{min,d}) \quad (5)$$

Denklem (5) $X_{i,d}$, i tane parçacığın d-boyutlu uzayda başlangıç değerlerini, $X_{min,d}$ ve $X_{max,d}$ alt ve üst sınırları, $r_{i,d}$ (0,1) arasındaki rastgele sayıyı ifade etmektedir.

Her bir i. parçacığın d-boyutunda t zamandaki hızı aşağıda verilen Denklem (6) ile konumu ise Denklem (7) ile güncellenmektedir. Parçacığın hız vektörü sürünün önceki tecrübelerinden ve genel tecrübesinden faydalanmaktadır. Hız vektörü optimizasyon sürecini yönlendirmektedir.

$$V_{i,d}^{t+1} = (w^{t+1} \times V_{i,d}^t) + c_1 r_1 \times (Pbest_{i,d}^t - X_{i,d}^t) + c_2 r_2 \times (Gbest_{i,d}^t - X_{i,d}^t) \quad (6)$$

$$X_{i,d}^{t+1} = X_{i,d}^t + V_{i,d}^t \quad (7)$$

Denklem (6)'da yer alan c_1 ve c_2 öğrenme faktörleri olarak adlandırılmaktadır. Bir parçacığın kendi tecrübesini veya sürüdeki diğer parçacıkların tecrübesini ifade etmektedir. r_1 ve r_2 katsayıları her iterasyonda yenilenen (0,1) aralığında değer alan rastgele sayılardır. $Gbest$ sürüde bulunan en iyi parçacığın konum değerini belirtmektedir. Global en iyi konum olarak adlandırılmaktadır. $Pbest$, parçacığın belirlenen iterasyon sayısının sonunda elde edilen en iyi kişisel konum değerini belirtmektedir. Lokal en iyi konum olarak adlandırılmaktadır. Lokal en iyi konum Denklem (8) kullanılarak bulunmaktadır.

$$Pbest_{i,d}^{t+1} = \begin{cases} Pbest_{i,d}^{t+1} & , f(X_{i,d}^{t+1}) \geq f(Pbest_{i,d}^t) \\ X_{i,d}^{t+1} & , f(X_{i,d}^{t+1}) < f(Pbest_{i,d}^t) \end{cases} \quad (8)$$

Denklem (8)'de f amaç fonksiyonunu ifade etmektedir. Amaç fonksiyonu, optimizasyon probleminin optimuma ne kadar yakın olduğunu göstermektedir. PSO' da yapılan değişikliklerden biri de hız vektörünün sınırlanmasıdır. Hız vektörünün sınırlanması parçacık arama yeteneğini kontrol etmektedir. Hız vektörünün sınırlanması Denklem (9)'da gösterilmiştir.

$$V_{i,d}^{t+1} = \begin{cases} V_{i,d}^{t+1} & , V_{i,d}^{t+1} < V_{max} \\ V_{max} & , V_{i,d}^{t+1} \geq V_{max} \end{cases} \quad (9)$$

PSO' nun akış diyagramı Şekil 2' de gösterilmektedir.



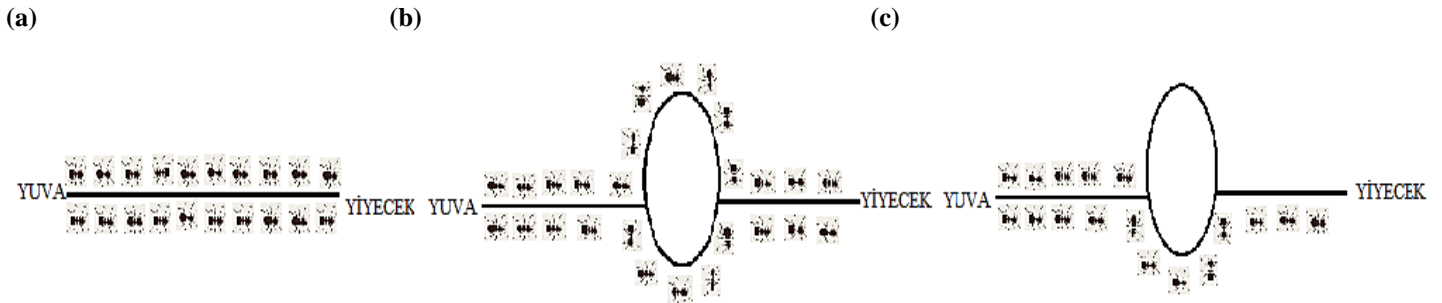
Şekil 2. PSO Akış Diyagramı

3.2. Karınca Koloni Algoritması (ACO)

Dorigo ve arkadaşları (1991) tarafından karıncaların yemek arama davranışından esinlenilerek oluşturulan siteme karınca sistemi, oluşturulan algoritmaya Karınca Koloni Algoritması (ACO) adı verilmiştir. ACO gerçek karıncaların yuvalarından yiyecek kaynağına doğru en kısa yolu bulmak için yaptıkları davranışlardan esinlenerek geliştirilmiştir. Karıncaların yuva ile yiyecek arasında en kısa yolu bulması ve diğer karıncalarla haberleşmesi için vücutları kimyasal bir madde salgılamaktadır. Salgıladıkları bu kimyasal maddeye feromon adı verilmektedir. Karıncalar geçtikleri yollarda belirli miktarda feromon maddesi bırakmaktadır. Feromon maddesinin miktarına göre gidecekleri yolu seçmektedirler.

Karıncalar feromon maddesinin daha yoğun olduğu yönü daha az olduğu yöne göre tercih etmektedir. Seçilen yolun kalitesi feromon yoğunluğu ile orantılıdır. Feromon izleri diğer karıncaların en kısa yolu bulmalarını sağlamaktadır. Karıncalar böylece optimum yönü seçebilmektedir. Feromon maddesi bir bilgiyi ifade etmekte ve karıncalar tarafından güncellenmektedir.

Karıncaların bir engelle karşılaştıklarında en kısa yolu nasıl buldukları Şekil 3' te gösterilmektedir.



Şekil 3.(a) Karıncaların Yiyeceğe Giderken İzlediği Yol; (a) Yuva İle Yiyecek Kaynağın Arasında Bir Engelle Konulması;

(c) Karıncaların En Kısa Yolu Bulması

Şekil 3(a)'da gösterildiği gibi karıncalar yiyecek kaynağına doğru gitmektedir. Gittikleri güzergah boyunca feromon maddesi bırakmaktadır. Feromon miktarı diğer karıncaların seçecekleri yolun belirlenmesinde önemli bir rol oynamaktadır. Şekil 3 (b)'de gösterildiği gibi karıncalar yiyecek kaynağına giderken bir engelle karşılaştığında feromonları takip edememekte ve gidilebilecek yolları eşit olasılıkla seçmektedir. Karıncaların hızları aynı olduğu varsayılırsa kısa yoldan geçen karınca sayısı uzun yoldan geçen karınca sayısından fazla olacaktır. Kısa yoldan geçen karınca sayısı daha fazla olacağından bırakılan feromon miktarı da daha fazla olacaktır. Şekil 3 (c)'de gösterildiği gibi karıncalar yiyecek kaynağına doğru en kısa yolu kullanmaktadır. Kısa yoldan geçen karıncaların sayısı arttıkça feromon miktarı da artacaktır. Bırakılan feromon miktarı yolun uzunluğu ile ters orantılıdır.

3.3. Sürekli Zaman Karınca Koloni Algoritması (ACO_R)

ACO algoritması ilk ayrık optimizasyon probleminin çözümü için kullanılmıştır ve daha sonra sürekli optimizasyon problemlerinin çözümü için geliştirilmiştir. Socha ve Dorigo (2008) tarafından ACO algoritması genişletilerek sürekli zaman karınca kolonisi algoritması (ACO_R) oluşturulmuştur. ACO_R algoritması sağlık alanı, akıllı sistemler, haberleşme sistemleri gibi birçok alanda kullanılmaktadır. Parametre değerlerinin uygun seçilmesi ile algoritmanın performansı önemli ölçüde artırılabilir. ACO_R

algoritması çözüm uzayından seçilen bir değişkenle Gaussian örnekleme kullanılarak yeni çözüm değişkeni oluşturulmasına bağlıdır (Socha ve Dorigo, 2008). ACO_R algoritmasında Gaussian örnekleme için değişken seçimi önemli bir konudur. Rulet Çarkı stratejisi uygulaması ile örnekleme için bir çözüm seçiminde aynı çözüm olmamasını ve daha geniş seçim sağlayarak ACO_R'nin performansının artırılmasını sağlamaktadır (Ojha, Abraham and Snasel, 2015).

Gaussian çekirdek fonksiyonu, Gaussian tabanlı birden fazla olasılık yoğunluk fonksiyonlarının ağırlıklı toplamından meydana gelmektedir. Gaussian çekirdek fonksiyonu Denklem (10)'da ifade edilmiştir.

$$G^i(x) = \sum_{l=1}^k w_l g_l^i(x) = \sum_{l=1}^k w_l \frac{1}{\sigma_l^i \sqrt{2\pi}} e^{-\frac{(x-\mu_l^i)^2}{2\sigma_l^{i2}}} \tag{10}$$

Burada i ($i=1, \dots, n$) problemin boyutunu, w her bir Gaussian fonksiyonunun ağırlık vektörünü, μ^i ortalama vektörü ve σ_l^i standart sapma vektörünü ifade etmektedir.

Tablo 1. Çözüm Tablosu (Socha and Dorigo, 2015)

s_1	s_1^1	...	s_1^i	...	s_1^n	$f(s_1)$	w_1
s_2	s_2^1	...	s_2^i	...	s_2^n	$f(s_2)$	w_2

s_l	s_l^1	...	s_l^i	...	s_l^n	$f(s_l)$	w_l

s_k	s_k^1	...	s_k^i	...	s_k^n	$f(s_k)$	w_k
	G^1		G^i		G^n		

Tablo 1’de yer alan s_l aday çözümlerini ve $f(s_l)$ ise amaç fonksiyonunu göstermektedir. Aday çözümlerin amaç fonksiyonların değerlerine göre $f(s_1) \leq f(s_2) \leq f(s_3) \leq \dots \leq f(s_l) \leq \dots \leq f(s_k)$ sıralanmıştır. Aday çözümlerinin, amaç fonksiyonlarının aldıkları değerleriyle ilişkili bir ağırlık oranı bulunmaktadır. Ağırlık oranları $w_1 \geq w_2 \geq w_3 \geq \dots \geq w_l \geq \dots \geq w_k$ şeklinde sıralanmaktadır. Gaussian çekirdek fonksiyonunu meydana getiren k adet Gaussian fonksiyonu bulunmaktadır.

μ^i , Gaussian çekirdek fonksiyonunun ortalama vektörünü ifade etmektedir ve Denklem (11) ile gösterilmiştir. (11)

$$\mu^i = \{\mu_1^i, \dots, \dots, \mu_k^i\} = \{s_1^i, \dots, \dots, s_k^i\}$$

Ağırlık vektörü w denklem (12) ile hesaplanmaktadır ve Gaussian fonksiyonun bir değeridir. Elde edilen çözümler değerlendirilerek amaç fonksiyonunun aldığı değerlere göre sıralanmaktadır.

$$w_l = \frac{1}{q_k \sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2 k^2}} \tag{12}$$

Burada q_k standart sapmayı belirtmektedir ve l argümanlı, 1.0 ortalamalıdır. Algoritmanın parametrelerinden biri olan q_k 'nin değeri küçük seçilirse en iyi çözümler elde edilebilmektedir.

Aşağıdaki Denklem (13)’te Gaussian fonksiyonların seçilebilme olasılığı formülü verilmiştir.

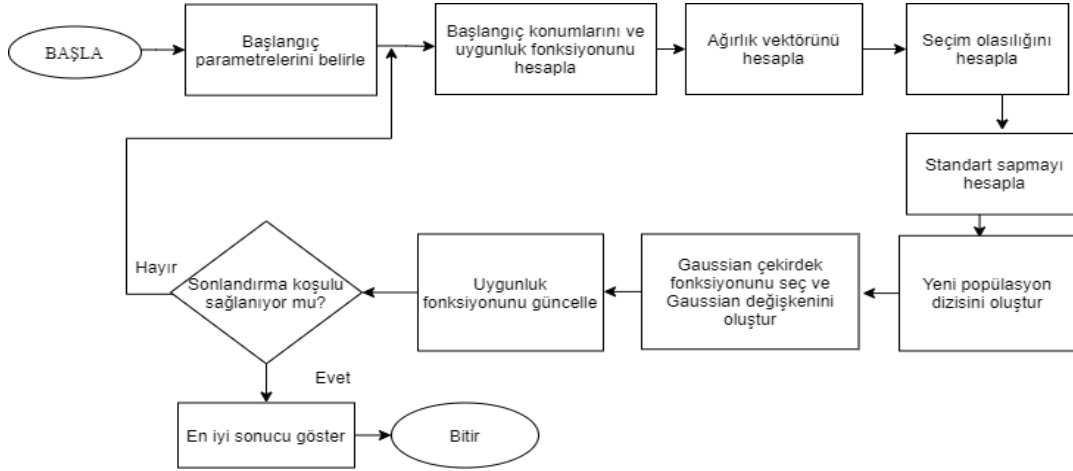
$$P_l = \frac{w_l}{\sum_{r=1}^k w_r} \tag{13}$$

Seçilebilme olasılığı hesaplandıktan sonra seçilen çözümün parametreleri Denklem (14) ile güncellenmektedir.

$$\sigma_l^i = \xi \sum_{e=1}^k \frac{|s_e^i - s_l^i|}{k-1} \tag{14}$$

Seçilen çözüm ile çözüm tablosunda bulunan diğer çözümler arasındaki ortalama mesafe Denklem (14)’te ifade edilen standart sapma σ_l^i ile hesaplanmaktadır. ξ parametresinin karınca koloni algoritmasında yer alan buharlaşma oranı ile aynı öneme sahiptir. Algoritmanın yakınsama hızı, ξ parametresinin değeri yükseldikçe azalmaktadır.

Sürekli zaman karınca koloni algoritmasının akış diyagramı Şekil 4'te gösterilmiştir.



Şekil 4. ACO_R Akış Diyagramı

3.4. Girdap Arama Algoritması (VSO)

Doğan ve Ölmez (2015) tarafından VSO önerilmiştir. VSO sürekli optimizasyon problemlerinin çözümü için kullanılan meta-sezgisel bir algoritmadır. Girdap modelinden esinlendiği için algoritmaya Girdap Arama Algoritması adı verilmiştir. VSO diğer optimizasyon algoritmalarına göre oldukça basittir ve tek çözüme dayalıdır.

İki boyutlu bir optimizasyon probleminin girdap modeli iç içe geçmiş birkaç daireden oluşmaktadır. Dairelerin yarıçapları azaltılarak oluşturulmaktadır. VSO'nun başlangıç aşamasında oluşturulan, girdabın en büyük çemberinin merkezi, Denklem (15) ile hesaplanmakta ve arama uzayının ortasında yer almaktadır. Başlangıç merkezi μ_0 denklem (15) ile bulunmaktadır.

$$\mu_0 = \frac{\text{üstlimit} + \text{altlimit}}{2} \quad (15)$$

Burada alt limit ve üst limit d boyutlu uzayda (karar değişkenlerinin sayısı) problemlerin sınırlarını ifade etmektedir. VSO başlangıç yarıçapı σ_0 denklem (16) ile bulunmaktadır.

$$\sigma_0 = \frac{\max(\text{üstlimit}) - \min(\text{altlimit})}{2} \quad (16)$$

a şekil parametresi olarak tanımlanmaktadır. Her iterasyonda Denklem (17) kullanılarak şekil parametresi hesaplanmaktadır.

$$a_t = a_0 - \frac{t}{\text{MaxItr}} \quad (17)$$

$a_0 = 1$ alınarak ilk iterasyonda arama alanının tam kapsamı sağlanmaktadır. Denklemdeki t iterasyon indeksini ve MaxItr ise maksimum iterasyon sayısını temsil etmektedir.

Yarıçapın değerini her iterasyonda azaltmak için ters tamamlanmamış gama fonksiyonu kullanılmaktadır. Çemberin yarıçapının değeri Denklem (18) ile hesaplanmaktadır.

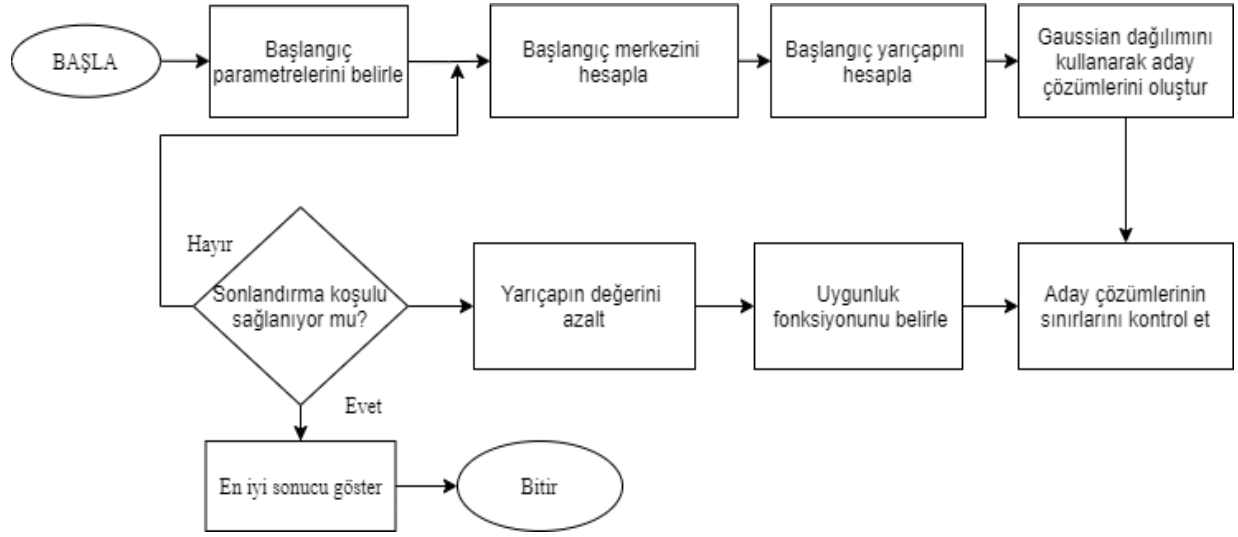
$$r_t = \sigma_0 \times (1/x) \times \Gamma(x, a_t) \quad (18)$$

Burada x değeri rastgele bir değer almakta ancak daha iyi performans elde etmek için 0.1 değeri alınmaktadır. Γ ters gamma fonksiyonunu ifade etmekte ve Matlab programında Γ yerine gammaincinv fonksiyonu kullanılmaktadır. VSO aday çözümler Gaussian dağılımı kullanılarak hesaplanmaktadır. Seçim yapılmadan önce n adet aday çözümler, arama sınırları içinde yer almalıdır. Sınırları aşan aday çözümler Denklem (19) kullanılarak sınırlara kaydırılmaktadır.

$$C(s)_k^i = \begin{cases} \text{rand.}(\text{üstlimit}^i - \text{altlimit}^i) + \text{altlimit}^i & , C(s)_k^i < \text{altlimit}^i \\ C(s)_k^i & , \text{altlimit}^i \leq C(s)_k^i \leq \text{üstlimit}^i \\ \text{rand.}(\text{üstlimit}^i - \text{altlimit}^i) + \text{altlimit}^i & , C(s)_k^i > \text{üstlimit}^i \end{cases} \quad (19)$$

$C(s)_k^i$, k. komşu çözümün i. boyutunu temsil etmektedir. rand uniform dağıtılmış rastgele bir sayıdır.

Girdap arama algoritmasının akış diyagramı Şekil 5' te gösterilmektedir.



Şekil 5. VSO Akış Diyagramı

4. Simülasyonlar

KSA'larda hedef sensörlerin konumlarını belirlemek için kullanılan optimizasyon algoritmalarının simülasyon sonuçları Şekil 6, 7 ve 8' te gösterilmektedir.

4.1. Parametreler ve Değerleri

PSO, ACO_R ve VSO algoritmalarında kullanılan parametre değerleri Tablo 2,3 ve 4' te gösterilmektedir.

Tablo 2. PSO Algoritması Parametreleri

PARAMETRELER	DEĞERLERİ
Problem Boyutu	2
Populasyon Sayısı	20
İterasyon Sayısı	100
Ağırlık Katsayısı (w)	1
Ağırlık Katsayısı Sönümlenme Oranı (wdamp)	0.99
Bilişsel Öğrenme Faktörü (c_1)	1.5
Sosyal Öğrenme Faktörü (c_2)	2.0
Alt Limit	0
Üst Limit	100

Tablo 3. ACO_R Algoritması Parametreleri

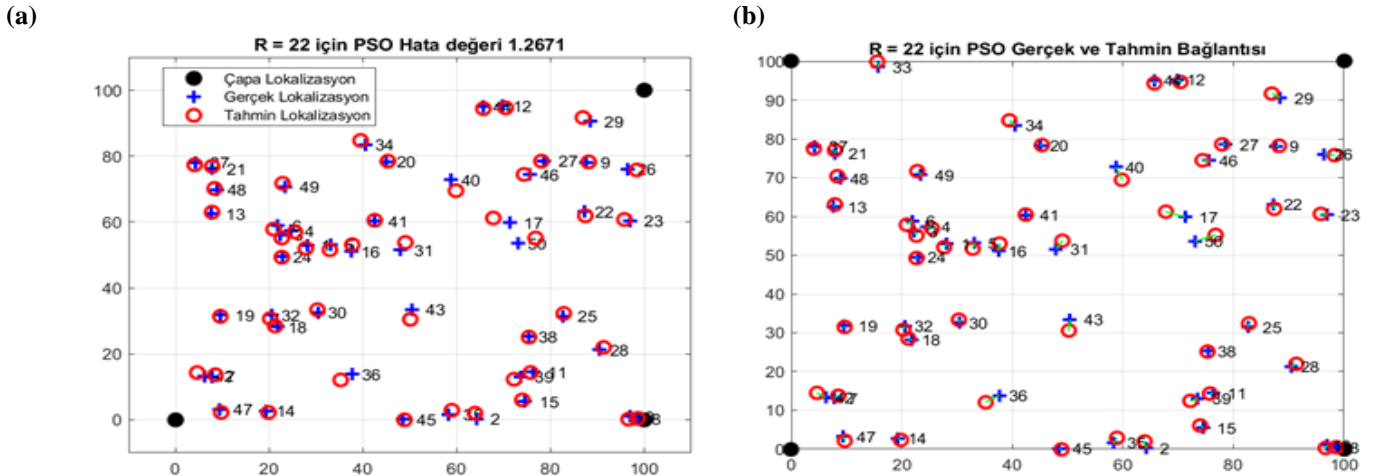
PARAMETRELER	DEĞERLERİ
Problemin Boyutu	2
Populasyon Sayısı	20
İterasyon Sayısı	100
Alt Limit	0
Üst Limit	100
Standart Sapma (q)	0.5
Sapma-Mesafe Oranı (ξ)	1

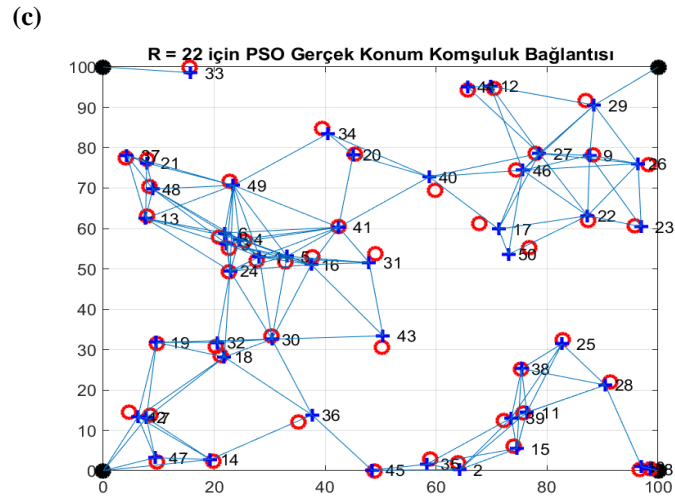
Tablo 4. VSO Algoritması Parametreleri

PARAMETRELER	DEĞERLERİ
Problemin Boyutu	2
Populasyon Sayısı	20
İterasyon Sayısı	100
Alt Limit	0
Üst Limit	100
x	0.1
Şekil parametresi (a_0)	1

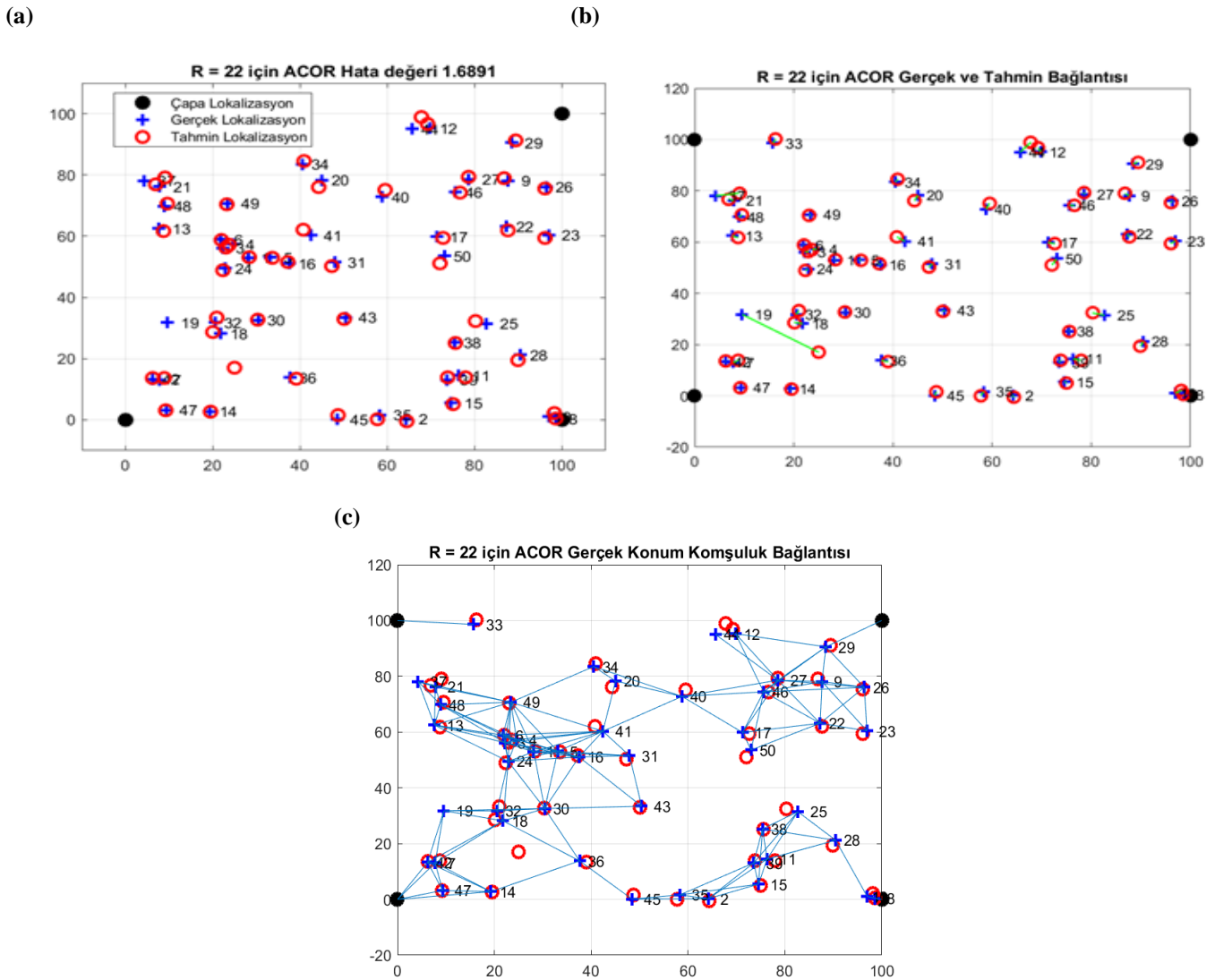
4.2. Bulgular

Matlab programında 100x100'lük bir alanda 4 adet referans düğüm (çapa) köşelere yerleştirilmiştir. Rastgele dağıtılan 50 adet düğümün konum tespiti için PSO, VSO ve sürekli dağılım için ACO_R kullanılmıştır. Her bir sensörün kapsama alanı yarıçapı R=22 birim alınarak PSO, VSO ve ACO_R algoritmaları için aynı düğüm koordinatları üzerinde çalıştırılmıştır. Tüm algoritmalar için simülasyon 1'e ait şekiller Şekil 6, 7, 8 ve 9'da gösterilmiştir. Şekil 6 ve 7'de PSO algoritmasına ait simülasyon sonuçları gösterilmiştir. Şekil 6 (a)'da yer alan mavi renkli + işaretler gerçek konumu, kırmızı renkli yuvarlak işaretler PSO algoritması ile elde edilen tahmini konumu ve köşelerde yer alan siyah renkli yuvarlak işaretler GPS ile yeri tespit edilen referans düğümü göstermektedir. Şekil 6 (b)'de gerçek konum ile tahmini konum arasındaki mesafe yeşil renkli çizgilerle gösterilmiştir. Şekil 7(c)'de ise R=22 için gerçek konum komşuluk bağlantısı gösterilmiştir. Şekil 8 (a), 8 (b) ve 8 (c)'de R=22 için VSO algoritmasının ve Şekil 9 (a), 9 (b) ve 9 (c)'de R=22 için ACO_R algoritması ile elde edilen simülasyon sonuçları gösterilmiştir. Simülasyonlarda R=22'de PSO, VSO ve ACO_R algoritmaları 100'er kere tekrarlanmıştır. Tablo 5'te tüm algoritmaların simülasyon sonuçları 20 defa tekrarlanarak elde edilen hata oranları ayrıntılı bir şekilde gösterilmiştir. Ayrıca Tablo 5'te 20 simülasyonun ortalama değeri verilmiştir. Tüm algoritmalar için simülasyon sonuçları 100 defa tekrarlanarak ortalama hata değerleri Tablo 6'da sunulmuştur.

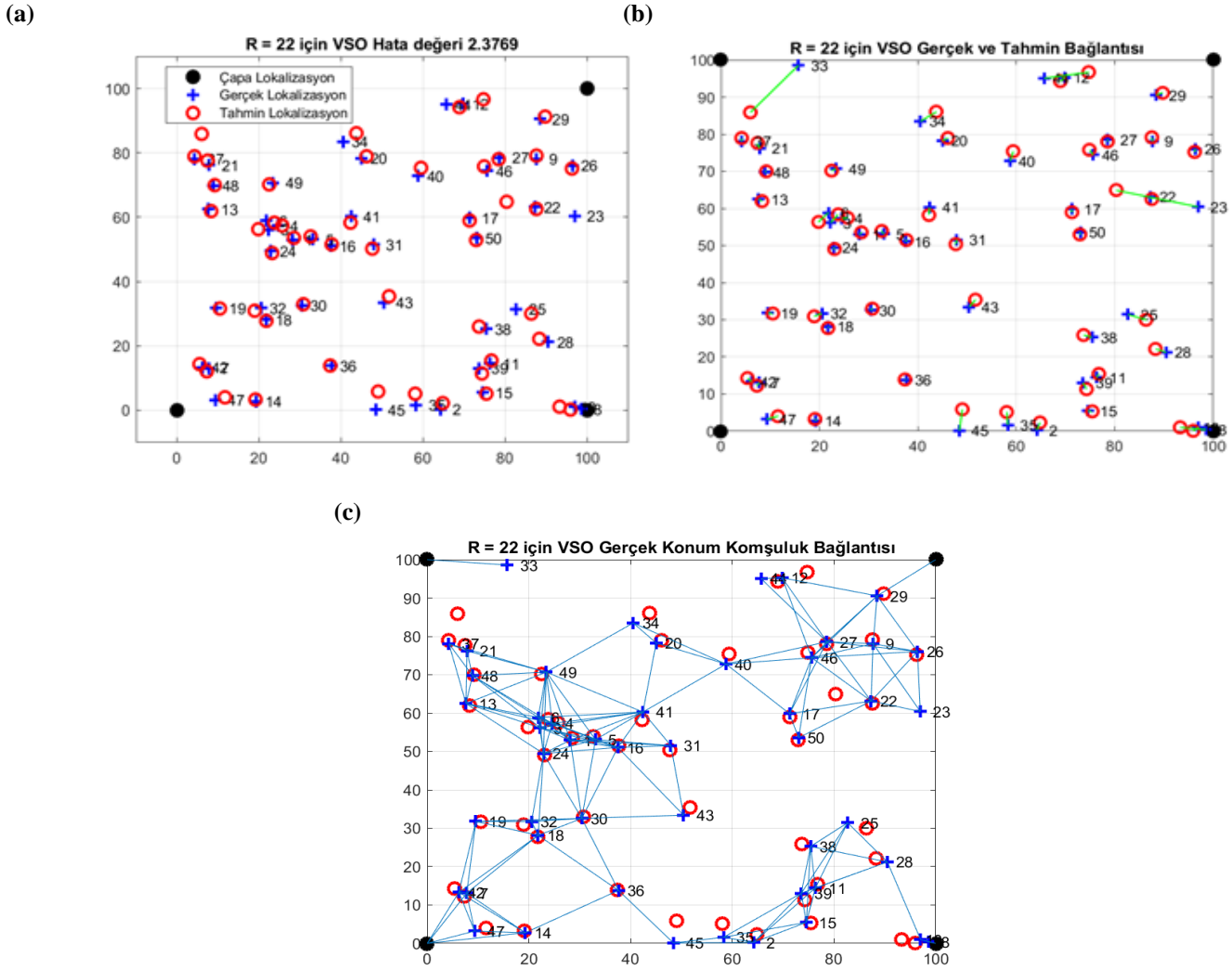
**Şekil 6.** (a) PSO için Hata Değeri; (b) PSO için Gerçek ve Tahmin Bağlantısı



Şekil 7. (c) PSO için Gerçek Konum Komşuluk Bağlantısı



Şekil 8. (a) ACO_R için Hata Değeri; (b) ACO_R için Gerçek ve Tahmin Bağlantısı; (c) ACO_R için Gerçek Konum Komşuluk Bağlantısı



Şekil 9. (a) VSO için Hata Değeri; (b) VSO için Gerçek ve Tahmin Bağlantısı; (c) VSO için Gerçek Konum Komşuluk Bağlantısı

Tablo 5. Algoritmaların Hata Değerleri
İLETİM MESAFESİ (R=22)

	PSO	ACO _R	VSO
Simülasyon 1	1.2671	1.6891	2.3769
Simülasyon 2	1.2214	2.0144	2.4070
Simülasyon 3	1.6462	1.5842	2.0642
Simülasyon 4	1.5783	2.0168	1.9336
Simülasyon 5	1.5061	2.0749	2.7797
Simülasyon 6	1.5547	1.9924	1.7697
Simülasyon 7	1.9272	1.7094	1.4003
Simülasyon 8	1.7964	2.0608	2.1830
Simülasyon 9	1.8548	1.9086	2.8971
Simülasyon 10	1.5013	1.3692	2.6733
Simülasyon 11	1.5216	1.7603	2.6872
Simülasyon 12	2.0558	2.1988	1.9177
Simülasyon 13	1.8275	2.1356	1.9222
Simülasyon 14	2.0547	1.9954	2.0141

Tablo 5 (Devam). Algoritmaların Hata Değerleri

İLETİM MESAFESİ (R=22)			
	PSO	ACO _R	VSO
Simülasyon 15	1.6993	1.9291	2.1023
Simülasyon 16	2.2072	2.2982	1.5084
Simülasyon 17	1.7768	2.2577	1.7496
Simülasyon 18	2.0178	1.9880	2.1831
Simülasyon 19	1.9201	1.6359	2.5364
Simülasyon 20	1.4411	2.2639	1.9704

Tablo 6. 100 adet Simülasyon için Algoritmaların Ortalama Hata Değerleri

İletim Mesafesi (R)	R=22
PSO	1.703336
ACO _R	1.875454
VSO	2.156924

5. Tartışma ve Sonuç

Çalışmada kablosuz sensör ağlarının hedef sensörlerin gerçek konumlarının belirlenmesinde PSO, ACO_R ve VSO algoritmaları kullanılmıştır. Algoritmalarda popülasyon boyutu 50 ve parametre değerleri birbirlerine yakın seçilmiştir. Aynı düğüm noktaları kullanılarak algoritmalar karşılaştırılmıştır. Elde edilen veriler grafiksel ve sayısal olarak şekil ve tablolar halinde sunulmuştur. Tablo 5'te PSO, ACO_R ve VSO algoritmalarının ortalama hata değeri simülasyonlar 20 kere tekrarlanarak hesaplanmıştır. PSO, ACO_R ve VSO göre daha az hata oranı elde edilmiştir. Tablo 6'da 100 adet simülasyon için algoritmaların ortalama hata değerlerine bakıldığında PSO'da hata oranının daha az olduğu tespit edilmiştir.

Literatürde birçok optimizasyon yöntemi kullanılarak en az hata ile hedef sensörlerin gerçek konumlarını tespit eden birden fazla çalışma bulunmaktadır. Bu makalede PSO algoritması kullanılarak en az hata tespit edilmiştir.

Referanslar

- Aspnes, J., Eren, T. and Goldenberg, D. (2006). "A Theory of Network Localization. IEEE Transactions on Mobile Computing", 5(12), pp. 1663–1678.
- Bacanin, N., Tuba, E., Zivkovic, M., Strumberger, I. and Tuba, M. (2019). "Whale Optimization Algorithm with Exploratory Move for Wireless Sensor Networks Localization", International Conference on Hybrid Intelligent Systems, pp. 328-338.
- Bekçibaşı, U., Tenruh, M. (2012). Kablosuz Algılayıcı Ağlarda Konum Saptama Teknikleri ve Mesafe Bağımlı Tekniklerde Dördüncü Çapa Yaklaşımı, Akademik Bilişim'12 - XIV. Akademik Bilişim Konferansı Bildirileri, 2012.
- Cui, H., Liang, Y., Zhou, C. and Cao, N. (2018) "Localization of Large-Scale Wireless Sensor Networks Using Niching Particle Swarm Optimization and Reliable Anchor Selection", Hindawi Wireless Communications and Mobile Computing, pp.1-18, <https://doi.org/10.1155/2018/2473875>.
- Dalkılıç, G. ve Türkmen, F. (2002) " Karınca kolonisi optimizasyonu", YPBS2002–Yüksek Performanslı Bilişim Sempozyumu, Kocaeli, Ekim.
- Dariane, A.B. and Moradi, A.M. (2009) "Reservoir Operating by Ant Colony Optimization for Continuous Domains (ACO_R) Case Study: Dez Reservoir", International Journal of Mathematical, Physical and Engineering Sciences", 3(2), pp.125-129.
- Doğan, B. ve Korürek, M. (2010) "EKG Vurularının Sürekli Zaman Karınca Koloni Optimizasyonu Yardımıyla Sınıflandırılması ECG Beat Classification Using Ant Colony Optimization for Continuous Domains", National Conference on Electrical, Electronics and Computer Engineering, Bursa, pp. 497-501.
- Doğan B. and Ölmez, T. (2015) "A new metaheuristic for numerical function optimization: Vortex Search algorithm, Information Sciences, 293, 125-145.

- Doğan, R. (2018) “KABLOSUZ SENSÖR AĞLARI İÇİN RSSI DEĞERİ İLE YAPAY SİNİR AĞI YAKLAŞIMLI KONUM BULMA”, Yüksek Lisans Tezi, Fırat Üniversitesi, Elazığ.
- Doğru, A.S., Temel, B. and 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.
- Dorigo M, Colorni A, Maniezzo V (1991) “Positive feedback as a search strategy”, Technical Report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy.
- Eberhart, R. and Kennedy, J. (1995) “A New Optimizer Using Particle Swarm Theory”, Sixth International Symposium On Micro Machine And Human Science, pp. 39-43.
- Eren, T. (2016). Graph Invariants for Unique Localizability in Cooperative Localization of Wireless Sensor Networks: Rigidity Index and Redundancy Index, Ad Hoc Networks, 44, pp. 32-45, 2016.
- Eren, T., (2017) “The effects of random geometric graph structure and clustering on localizability of sensor networks”, International Journal of Distributed Sensor Networks, 13(12), 1-15.
- 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 (2019).
- Eren, T. Goldenberg, D. and Whiteley, W. (2004) “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.
- Gangal, V. (2015) “ Kablosuz Algılayıcı Ağlarda Karınca Koloni Algoritmaları ile Enerji Etkin Rotalamanın İncelenmesi”, Yüksek Lisans Tezi, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü.
- Goyal, S. and Patterh, M.S. (2015), “Flower Pollination Algorithm based Localization of Wireless Sensor Network”, 2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS), doi:[10.1109/RAECS.2015.7453299](https://doi.org/10.1109/RAECS.2015.7453299).
- Güngüneş, R., Ateş, V., Lüy, M. ve Eke, İ. (2019) “Doğrusal Olmayan Yüklere Sürekli Zaman Karınca Kolonisi Algoritması İle Optimal Değerli Kondansatör Seçimi ve Güç Kontrolü”, Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, 11(3), 780-792, doi:[10.291374/umagd.636390](https://doi.org/10.291374/umagd.636390).
- Kannan, A.A., Mao, G. And Vucetic, B. (2006) “Simulated annealing based localization in wireless sensor network”, Journal of Computers, doi:[10.4304/jcp.1.2.15-22](https://doi.org/10.4304/jcp.1.2.15-22).
- Karaboğa, D. (2004). Yapay Zeka Optimizasyon Algoritmaları. Nobel Yayın Dağıtım, 2004
- Kennedy, J. And Eberhart, R. (1995) “Particle Swarm Optimization”, pp.1942-1948.
- Keskintürk, T. ve söyler, H. (2006) "GLOBAL KARINCA KOLONİSİ OPTİMİZASYONU", Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 21(4).
- Kızılkaplan, E., Eren, T. and Yalçınkaya, F. (2020) “Kablosuz Sensör Ağlarında Konum Belirlemede Sezgisel Algoritmaların Kuantum Davranışları ile Karşılaştırılması”, Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi, 12(2), 587-602, doi:[10.29137/umagd.746589](https://doi.org/10.29137/umagd.746589).
- Kumar, A., Khosla, A., Saini, J.S. ve Singh, S. (2012) “Meta-Heuristic Range Based Node Localization Algorithm for Wireless Sensor Networks”, International Conference on Localization and GNSS, doi:[10.1109/ICL-GNSS.2012.6253135](https://doi.org/10.1109/ICL-GNSS.2012.6253135).
- Ojha, V. K., Abraham, A. ve Snasel, V. (2017) "ACO for Continuous Function Optimization: A Performance Analysis", 14th International Conference on Intelligent Systems Design and Applications, (February 2015), pp. 145–150. doi:[10.1109/ISDA.2014.7066253](https://doi.org/10.1109/ISDA.2014.7066253).
- Özsağlam, M.Y. and Çunkaş, M. (2008) “Optimizasyon Problemlerinin Çözümü için Parçacık Sürü Optimizasyonu Algoritması”, Politeknik Dergisi, 11(4), pp.299-305.

- Öztürk, C. (2016) “Yapay Arı Koloni Algoritması ile Kablosuz Algılayıcı Ağlarda Lokalizasyon”, T.C. ERCİYES ÜNİVERSİTESİ BİLİMSEL ARAŞTIRMA PROJELERİ KOORDİNASYON BİRİMİ, Proje No:FBA-12-4029.
- Poli, R., Kennedy, J. And Blackwell, T. (2007) “Particle swarm optimization”, *Swarm Intell*, pp. 33-57, [doi:10.1007/s11721-007-0002-0](https://doi.org/10.1007/s11721-007-0002-0).
- Ramson, S.R.J. and Moni, D.J. (2017) “Applications of Wireless Sensor Networks-A Survey”, *Proceedings of IEEE International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology*, pp.325-329, [doi:10.1109/ICIEEIMT.2017.8116858](https://doi.org/10.1109/ICIEEIMT.2017.8116858)
- Rashid, H. and Turuk, A.K. (2013) “Localization of Wireless Sensor Networks Using a Single Anchor Node”, *Wireless Personal Communications*, 72(2), pp.975-986, [doi:10.1007/s11277-013-1050-y](https://doi.org/10.1007/s11277-013-1050-y).
- Reyes-Sierra, M. and Coello Coello, C.A. (2006) “Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art”, *International Journal of Computational Intelligence Research*, 2(3), pp.287-308, [doi:10.5019/j.ijcir.2006.68](https://doi.org/10.5019/j.ijcir.2006.68)
- Sağ, T. (2020) “Çok Merkezli Girdap Arama Algoritması”, *Düzce Üniversitesi Bilim Ve Teknoloji Dergisi*, 8, 1279-1294, [doi:10.29130/dubited.644881](https://doi.org/10.29130/dubited.644881).
- Shrivastava, A. and Bharti, P. (20115) “Localization Techniques for Wireless Sensor Networks”, *International Journal of Computer Applications*, 16(12), pp.13-18.
- Singh, S.P. and Sharma, S.C. (2015) “Range Free Localization Techniques in Wireless Sensor Networks: A Review”, 3rd *International Conference on Recent Trends in Computing*, *Procedia Computer Science*, pp.7-16, [doi:10.1016/j.procs.2015.07.357](https://doi.org/10.1016/j.procs.2015.07.357)
- Socha, K. and Blum, C. (2007) “An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training”, *Neural Comput& Applic*, 16, pp. 235-247, [doi:10.1007/s00521-007-0084-z](https://doi.org/10.1007/s00521-007-0084-z).
- Socha, K. and Dorigo, M. (2008) “Ant colony optimization for continuous domains”, *European Journal of Operational Research*, 185, pp. 1155-1173, [doi:10.1016/j.ejor.2006.06.046](https://doi.org/10.1016/j.ejor.2006.06.046).
- Strumberg, I., Beko, M., Tuba, M., Minovic, M. and Bacanin, N. (2018) “Elephant herding optimization algorithm for wireless sensor network localization problem”, *Technological Innovation for Resilient Systems*, pp.175-184, [doi:10.1007/978-3-319-785745_17](https://doi.org/10.1007/978-3-319-785745_17).
- Wang, C., Chen, J., Sun, Y. and Shen, X.S. (2009) “Wireless Sensor Networks Localization With Isomap”, *IEEE International Conference On Communications*, pp. 28-32, [doi:10.1109/ICC.2009.5199576](https://doi.org/10.1109/ICC.2009.5199576).
- Wilson, R.J. (1995) “Introduction to Graph Theory”, Fourth edition.
- Xiao, J. and Li, L.P. (2011) “A hybrid ant colony optimization for continuous domains”, *Expert Systems with Applications*, 38, 11072-11077.
- Yıldırım, G. (2012) “Kablosuz Sensör Ağı İle Konum Belirleme Ve İzleme, Yüksek Lisans Tezi, Fırat Üniversitesi, Elazığ.