

YAPAY ARI KOLONİSİ ALGORİTMASI İLE YAPILAN GELİŞTİRMELER VE SONUÇLARI

Derleme Makale

Burak KAYA¹
İbrahim EKE²

KAYA, B. ve EKE, İ., (2020), **Yapay Arı Kolonisi Algoritması İle Yapılan Geliştirmeler ve Sonuçları**, Verimlilik Dergisi, Yıl: 2020, Sayı: 1, T. C. Sanayi ve Teknoloji Bakanlığı Yayını.

ÖZET

Optimizasyon algoritmalarında genel olarak çoğu problem türünde iyi performans sağlayıp sağlayamadığının analiz edilmesi ve literatürdeki algoritmalarla kıyaslanarak davranışlarının incelenmesi gerekir. Bu nedenle optimizasyon türlerinden biri olan ve arıların yiyecek arama davranışlarını modelleyen Yapay Arı Kolonisi (ABC) algoritmasının ilk literatüre girişinden son zamanlardaki gelişim sürecine kadar Performans Analizi yapılmıştır.

Karaboğa tarafından 2005 yılında ortaya çıkarılan ABC'nin son yıllarda yapılan çalışmalar sonucunda yeni çözümleri bulma mekanizmasının çok iyi olduğu fakat yerel araştırma yapma mekanizmasının geliştirilebileceğini ortaya koymuştur. Algoritmanın klasik hali ve geliştirilen süreçler test problemlerinde belirli parametreler dikkate alınarak incelenmiş ve hangi iyileştirmenin standart ABC'ye göre daha iyi çözümler ürettiği gösterilmiştir.

Anahtar Kelimeler: *Optimizasyon, Yapay Arı Kolonisi, Test Problemleri.*

¹ **Burak KAYA**, Kırıkkale Sanayi ve Teknoloji İl Müdürlüğü, Mühendis. ORCID: 0000-0003-0146-0722

² **İbrahim EKE**, Dr. Öğretim Üyesi, Kırıkkale Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Öğretim Üyesi. ORCID: 0000-0003-4792-238X

* Makale Gönderim Tarihi: 06.05.2018 Kabul Tarihi: 01.08.2018

DEVELOPMENTS IN ARTIFICIAL BEE COLONY ALGORITHM AND THE RESULTS

ABSTRACT

In optimization algorithms, it is generally necessary to analyze whether it provides good performance in most problem types and to analyze its behavior by comparing with the algorithms in the literature. For this reason, performance analysis of Artificial Bee Colony (ABC) algorithm, which is one of the optimization types modeling the bee's food search behaviors, have been made.

The ABC, discovered by Karaboğa in 2005, is a good mechanism for finding new solutions. However, there is a need for improvements in ABC with respect to local research. The classical state of the algorithm and the developed processes have been examined by taking into account the specific parameters in the test problems and it has been shown that which improvement produces better solutions than standard ABC.

Keywords: *Optimization, Artificial Bee Colony, Test Problems.*

1. GİRİŞ

Günlük hayatta problemlerin çoğu kombinasyonel ve sürekli tipteki optimizasyon problemleridir. Optimizasyon genelde belli bir uzay aralığı olan optimizasyon problemine getirilen alternatif çözümlerin anlamlı şekilde karşılaştırılıp bölge içerisindeki en iyi olanı bulma sürecidir.

Optimizasyon kriterleri aşağıdaki süreçler dikkate alınarak belirlenir (Ara,1989).

- Tasarım Değişkenleri

Sistemi tanımlamak için en az sayıda değişkenle problemin en doğru bir şekilde ifade edilmesi gerekir.

- Amaç ve Uygunluk Fonksiyonu

Optimizasyon algoritmasının aday sonuçlarından hangisinin daha iyi olduğunun belirlenmesi için uygunluk kalite fonksiyonu kullanılır.

- Sınırlamalar

Bir sistemin tasarımı, tasarım değişkenlerine atanan değerlere alt ve üst sınırlar gibi bazı sınırlamalar belirleyerek en iyi tasarım sağlanır.

- Optimizasyon Yöntemleri

Temel olarak Klasik Teknikler ve Modern Sezgisel Algoritmalar (MSA) olarak ikiye ayrılır. Klasik teknikler kesin çözüm üretmekle birlikte; problemlere özgü olması, yani genel (esnek) olmaması, problemin matematiksel modellenmesi gibi birçok dezavantajı vardır. Klasik çözüm üreten metotların belirtilen dezavantajlarından dolayı optimum çözüm bulması zaman almakta ve gerçek hayattaki problemlere uygulanamamaktır (Dumitrescu, 2003). Bu nedenlerden dolayı bilim adamları genel amaçlı, esnek ve performansı yüksek yöntemler araştırmak için doğada var olan ve bu sistemlerde meydana gelen olaylara yönelmişlerdir. Bu konuda yapılan tespitlerle yaklaşık çözümler üreten MSA olan yapay ısıl işlem, tabu araştırma, genetik algoritma, karınca algoritması, yapay bağışıklık algoritması, diferansiyel gelişim algoritması ve ABC ortaya çıkmıştır.

- Gelişime Dayalı Algoritmalar

MSA'nın en temel gelişime dayalı olmasını sağlayan iki önemli etken vardır.

Bunlar, deęişim operatörleri popülasyonunun farklılığını sağlar ve seçim ise popülasyonun kalitesini artırır.

Seçim yöntemi rulet tekerlięi, sıralamaya dayalı, rastlantısal, örnekleme, turnuva yöntemi ya da dięer seçim şemalarından herhangi biri ile gerçekleşir (Eiben, 2003). Gelişime dayalı algoritmalarda en önemli bileşenler; gösterim, değerlendirme fonksiyonu, popülasyon, ebeveyn seçme mekanizması, deęişim operatörleri ve seçme mekanizması olarak dikkate alınmalıdır.

- Sezgisel Yöntemlerin Performanslarının Deęerlendirilmesi

Belirli bir problem üzerinde çalışırken şu sorulara cevap verilmelidir. Bulunan çözümler ne kadar kaliteli? En iyi çözüm bulma süresi ne kadar sürmektedir? Metotlar ne kadar kullanışlı? En iyi çözüm ile dięer çözüm arasındaki fark nedir? (Akay, 2009).

2. SÜRÜ ZEKASI

Sürü terimi ortak davranış sergileyen hayvan ve böcek topluluklarını ifade etmektedir. Araştırmacılar doğada var olan başka olayları da incelemeye ve bunların modellenmesini yapmaya çalıştıklarında sürüler halinde yaşayan canlıların toplu olarak gösterdikleri sürü zekası davranışı tespit etmişlerdir (Bonabeau, 1999).

Son yıllarda ise biyologlar ile bilgisayar uzmanları "Sürü Zekası" adı altında bu sürülerin davranışlarının nasıl modellenebileceęi ve aralarındaki iletişim mantıęı üzerine yaklaşımların, optimizasyon problemleri, robotbilim ve askeri uygulamalarda da başarılı olması nedeniyle bu konu üzerindeki çalışmaları yoğunlaştırmıştır (Korb, 2009).

Yiyecek arama davranışlarının modellenmesi ile oldukça başarılı algoritmalar ortaya çıkmıştır. Bunlardan ABC de Karaboęa tarafından arıların sürü olarak yiyecek arama davranışlarını model alarak geliştirilmiş bir optimizasyon algoritmasıdır.

İş bölümü yapabilme ve kendi kendine organize olabilme, sürü zekasının iki önemli özellięidir. Arılardaki sürü zekası şu şekilde açıklanabilir:

1- Pozitif Geri Besleme: Bir işin daha çok yapılması için gerekli koşullardır. Örneğin kaynağın nektar miktarı arttıkça bu kaynaęı geçen arı sayısı da artmaktadır.

2- Negatif Geri Besleme: Yiyecek tüketiminde ya da yarışma süreçlerinde

doyuma gitmeyi önlemek için negatif geri besleme mekanizmasına ihtiyaç duyulur. Örneğin, tükenen kaynak bırakılmaktadır.

3- Salınımlar: Keşifler veya hatalar gibi davranışlar yaratıcılık ve yenilikler açısından önem taşımaktadır. Örneğin, arılar yiyecek kaynağı bulabilmek için rastgele arama yapmaktadır.

4- Çoklu Etkileşimler: Sürüdeki bazı bireylerin diğer bireylerden gelen bilgi paylaşımını sağlar. Örneğin, arılar yiyecek kaynakları ile ilgili olarak dans alanında bilgi dağılımı gerçekleştirirler (Waibel, 2006).

İşlerini dinamik olarak dağıtabilen ve çevresel değişimlere karşı sürü zekalarıyla uyarlanabilir cevaplar verebilen bal arıları fotoğrafik hafızaya, uzay çağrı sensörlerine, navigasyon sistemine, sezgisel kavrama yeteneğine ve yeni yuva yeri seçerken sürü olarak karar verme özelliğine sahiptirler (Akay, 2009).

3. YİYECEK ARAMA DAVRANIŞLARI

Tereshko'nun reaktif difüzyon denklemlerine temel olarak önerdiği kolektif zekayı sağlayan minimal yiyecek arama modelinde temel 3 bileşen vardır: Yiyecek kaynakları, görevli işçi arılar ve görevsiz işçi arılardır. Minimal Model iki modda ele alınır. Bir yiyecek kaynağına yönelme ve kaynağı bırakma (Tereshko, 2000). Bileşenler şu şekilde ifade edilebilir:

1- Yiyecek Kaynakları: Arıların nektar, polen veya bal elde etmek için gittikleri yiyecek kaynağının değeri, türü, yuvaya yakınlığı, nektar konsantrasyonu veya nektarın elde edilmesinin kolaylığı gibi birçok faktöre bağlıdır.

2- Görevli İşçi Arılar: İşçi arılar, daha önceden keşfedilen kaynakların kovana getirilmesinden sorumludur. İşçi arılar ziyaret ettikleri kaynağın kalitesi ve yeriyle ilgili bilgiyi diğer arılara iletirler.

3- Görevsiz İşçi Arılar: Görevi belirsiz iki çeşit işçi arı bulunmaktadır: içsel bir duyguya veya bir dış faktöre bağlı olarak rastgele kaynak arayışında olan kaşif arılar ve kovanda izleyen ve görevli arılardan paylaşılan bilgiyi kullanarak yeni bir kaynağına yönelen gözcü arılar.

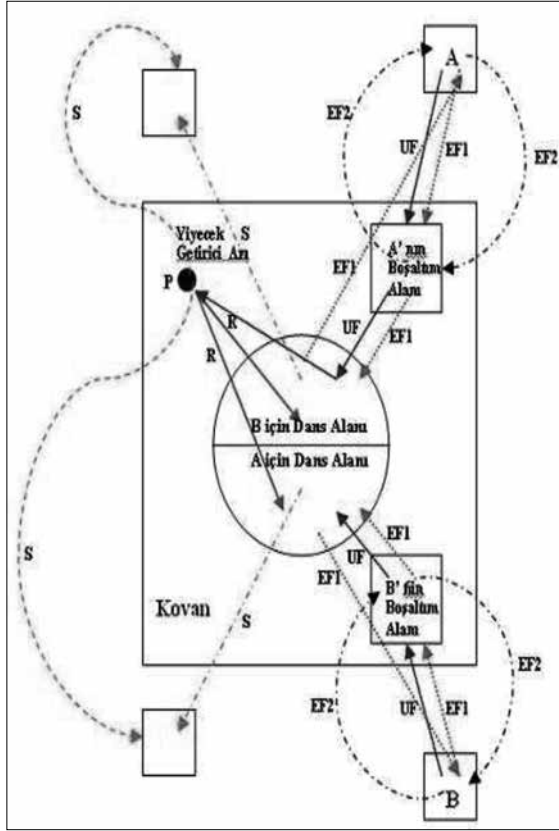
Yiyecek kaynağının kalitesi ve yeriyle ilgili bilgi paylaşımı dans alanında olmaktadır. Bir arı dans ederken diğer arı da ona antenleri ile dokunur ve kaynakla ilgili bilgi alırlar. Kovandan uzaklık, çiçek nektarının kıvamı, besinin durumu, kalitedeki göreceli farklılıklar, hava şartları ve günün hangi vaktinin olduğu dansı etkileyen diğer faktörlerdir.

Bileşik gözleri ile arılar, kendi rotaları ile güneş arasındaki açıyı hesaplayabilmektedirler. Arılar uzaklığa göre enerji tüketimini ve enerji tüketimlerini ayarlamaktadırlar.

Kaynağın kovana olan yakınlığına göre dairesel, kuyruk ve titreme dansı mevcuttur. Buradaki titreme dansı, nektar miktarı ve getirme yeteneği arasındaki dengeyi sağlamaktadır. Daire dansı, 50-100 metre uzaklıktaki kaynağın yön ve uzaklık bilgisi verilmeden sağlanan dans biçimidir. 100 metreden 10 kilometreye kadar geniş bir alan içerisinde 8 rakamına benzeyen ve yineleme sayısına göre uzaklık bilgisi sunan kuyruk dansı ile güneş ve yiyecek arasındaki açının 45° olduğunu anlamaktadırlar (Tereshko, 2005).

Yiyecek arayıcıların davranışlarının daha iyi anlaşılabilmesi için Şekil 1'de verilen modelin incelenmesi faydalı olacaktır. A ve B isminde iki keşfedilmiş kaynak olduğu varsayılıyorsa bu durumdaki bir arı için iki olası seçenek söz konusudur.

- Bu arı kaşif arı olabilir (Şekil 1'de S ile gösterilmektedir).
- Bu arı bir gözcü arı olabilir (Şekil 1'de R ile gösterilmektedir).
- Bu arı bir işçi arı olabilir. Nektarı aktardıktan sonra üç seçenek ortaya çıkar:
 - ✓ Gittiği kaynağa bırakarak bağımsız izleyici olabilir (Şekil 1'de UF ile gösterilmektedir).
 - ✓ Gittiği kaynağa dönmeden önce dans ile diğer arıları kaynağa yönlendirilir (Şekil 1'de EF1 ile gösterilmektedir).
 - ✓ Diğer arıları yönlendirmeden kaynağa gidebilir (Şekil 1'de EF2 ile gösterilmektedir) (Karaboğa, 2011).



Şekil 1. Yiyecek Arama Modeli

4. YAPAY ARI KOLONİSİ ALGORİTMASI

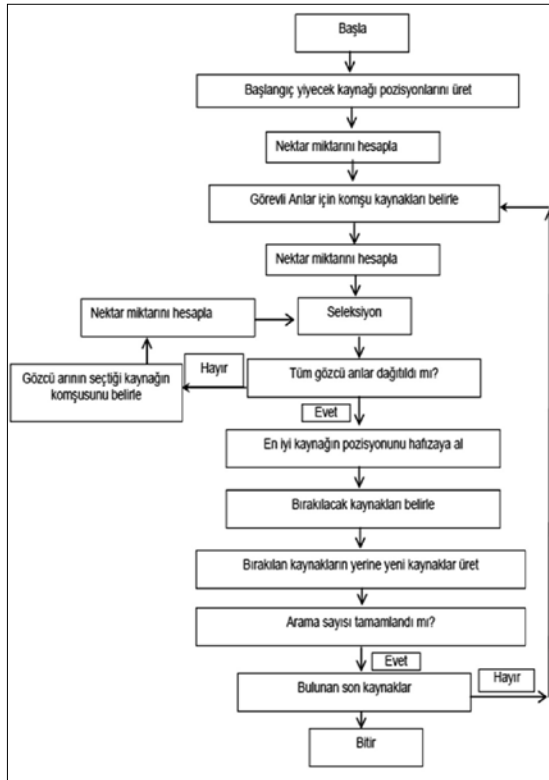
Karaboğa'nın ABC'sinin mantığında bazı kabuller yapılmaktadır. Buna göre, görevli arılar ile toplam yiyecek kaynağı sayısı birbirine eşittir, aynı zamanda işçi arıların ve gözcü arıların sayısı da eşittir. Ayrıca nektarı tükenmiş kaynağın görevli arısı modelde kaşif arı olarak görevini sürdürmektedir. Yiyecek kaynaklarının yerleri optimizasyona ait olasılık sonuçlarını ve kaynakların nektar miktarları ise sonuçların kalitesine karşılık gelmektedir. Bu kabullerden yola çıkıldığında ABC optimizasyon algoritması en fazla nektara sahip kaynağın yerini bulmaya çalışarak uzaydaki sonuçlardan problemin minimumu ya da maksimumunu veren sonucun en uygun değerini bulmaya çalışmaktadır (Karaboğa, 2009).

ABC temel adımları;

1. Yiyecek kaynağı alanlarının üretilmesi,
2. Tekrar,

3. İşçi arıların yiyecek kaynaklarına gönderilmesi,
4. Olasılık değerlerinin görevli arılardan gelen bilgi paylaşımına göre hesaplanması,
5. Gözcü arıların olasılık değerlerine göre yiyecek kaynağı alanları seçmeleri,
6. Kaynağı bırakma kriteri: Limit ve kaşif arı üretimi sayısı,
7. Kontrol=Maksimum çevrim.

Algoritma arama uzayındaki sonuçlara karşılık gelen yiyecek kaynakları parametrelerin alt ve üst sınırları arasındaki yerleri gelişigüzel üretmektedir. Oluşturulan her bir kaynağın geliştirilememe sayacı da aynı anda sıfırlanmaktadır. Bu süreçten sonra kaynağı bırakma kriteri sağlanıncaya kadar yiyecek kaynakları işçi arı, kaşif arı ve gözcü arı fazlarından geçirilerek en uygun değer bulunmaya çalışılmaktadır. Yiyecek kaynağının komşuluğunda yeni bir kaynak belirlenir ve bu kaynak daha iyi ise işçi arı veya gözcü arı bu kaynağı hafızasına almakta ve geliştirilememe sayacı sıfırlanmaktadır. Ters durumda ise geliştirilememe sayacı bir arttırılmaktadır (Akay, 2009).



Temel ABC'nin adımları aşağıdaki gibidir.

Adım 1: Eşitlik (1) ile x_{ij} $i=1, \dots, N, j=1..M$, çözümlerine başlangıç değerlerini ata ve geliştirilemeye sayaçlarını ($hata_j$) sıfırla. Uygunluk değerlerini hesapla. N , yiyecek kaynağı sayısı ve M , optimize edilecek parametre sayısıdır.

$$x_{ij} = x_j^{\min} + rand(0,1) * (x_j^{\max} - x_j^{\min}) \quad (1)$$

Adım 2: $i=1$ 'den N 'ye kadar Eşitlik (2)'yi kullanarak x_i çözümünün görevli arısı için yeni bir kaynak üret.

$$v_{ij} = x_{ij} + \phi_{ij} * (x_{ij} - x_{kj}) \quad (2)$$

Bu işlemde üretilen v_{ij} değerinin daha önceden belirtilmiş olan alt ve üst sınırları aşması durumunda Eşitlik (3) kullanarak j parametreye ait alt veya üst sınır değerlerine ötele.

$$v_{ij} = \begin{cases} x_j^{\min} , & v_{ij} < x_j^{\min} \\ v_{ij} , & x_j^{\min} \leq v_{ij} \leq x_j^{\max} \\ x_j^{\max} , & v_{ij} > x_j^{\max} \end{cases} \quad (3)$$

Bu kaynağın maliyet değerini $f(v_j)$ Eşitlik (4)'te yerine koyarak bu çözümün uygunluk değerini hesapla. v_i ve x_i arasında seçim işlemini uygula ve daha iyi olanı seç. x_i çözümü gelişmemiş ise $hata_i$ 'yi bir arttır. f_j v_i kaynağının maliyet değeridir.

$$uygunluk_i = \begin{cases} 1/(1 + f_i) , & f_i \geq 0 \\ 1 + abs(f_i) , & f_i < 0 \end{cases} \quad (4)$$

Adım 3: Gözcü arıların seçim işlemi yaparken kullanacakları uygunluk değerine dayalı olasılık değerlerini Eşitlik (5)'i kullanarak hesapla.

$$p_i = \frac{uygunluk}{SN} \quad (5)$$

$$\sum_{i=1}^{SN} uygunluk$$

Adım 4: Rulet tekerleğine göre seçim işleminde her bir kaynak için $[0, 1]$ aralığında üretilen p_i değeri gelişigüzel üretilen bir değerden büyükse gözcü arı için Eşitlik (2)'yi kullanarak yeni bir kaynak üret ve üretilen v_i ile x_i arasında seçim işlemi uygula, daha iyi olanı seç. x_i çözümü gelişmemişse $hata_i = hata_i + 1$, gelişmişse $hata_i = 0$ yap. Bu adımı tüm gözcü arılar yiyecek kaynağı bölgelerine dağılıncaya kadar tekrar et.

Adım 5: Kaynağın nektarının tükenip tükenmediğini kontrol et. Eğer tükenmişse Eşitlik (1)'i kullanarak gelişigüzel üretilen yeni bir çözümle değiştir.

Adım 6: En iyi çözümü hafızada tut.

Adım 7: Sonlandırma koşullarını kontrol et. Eğer koşullar sağlanmıyorsa Adım 2'den Adım 6'ya kadar tekrar et (Akay, 2009).

5. SİMÜLASYON ÇALIŞMASI

Literatür araştırmaları sonucu tespit edilen çalışmalar ve sonuçlar çizelge halinde sunulmuştur. Simulasyon çalışmalarında koloninin boyutunu 20, maximum döngüyü 2500, hata e^{-10} , limit 100, değişken sayısı 10, alt ve üst sınır 100 olarak seçilen parametreler ile 20 defa çalıştırılarak sonuçlar çizelgeler ve grafiksel olarak gözlemlenmiştir. Çizelge 1'de algoritmaların gelişim fazları ve mevcut modelleri karşılaştırılarak ele alınmıştır.

Çizelge 1. Algoritma Modelleri

Algoritma	Geliştirilen Fazlar
Algoritma 1 (Karaboğa, 2011)	İşçi Arı Fazı Klasik Hali: $x_{ij} = x_j^{\min} + \text{rand}(0,1)(x_j^{\max} - x_j^{\min})$ Q=(rand-0,5)*2 Gözcü Arı Fazı Klasik Hali: $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$ Q=(rand-0,5)*2
Algoritma 2 (Gao, 2013)	İşçi Arı Geliştirilen Fazı: $x_{ij} = x_j^{\min} + \text{rand}(0,1)(x_j^{\max} - x_j^{\min})$ Q=rand (0,1) Gözcü Arı Geliştirilen Fazı: $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$ Q=rand (-1,1)
Algoritma 3 (Jeya, 2010)	Olasılıksal Değer Klasik Hali: $p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{\text{SN}} \text{fitness}_i}$ Geliştirilen Hali: $P[i] = (0,9 \times (\text{fitness}[i]/\text{maxfit})) + 0,1$
Algoritma 4 (Pacurib, 2009)	Gözcü Arı Fazı Klasik Hali: $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$ Q=rand(0,1) Gözcü Arı Fazı Geliştirilen Hali: $V_{ij} = X_{ij} + \text{rand}[0,1] \times \text{abs}(X_{ij} - X_{kj})$ Q=rand(-1,1)
Algoritma 5 (Babayiğit, 2012)	Olasılıksal Değer Klasik Hali: $p_i = \frac{\text{fitness}_i}{\sum_{i=1}^{\text{SN}} \text{fitness}_i}$ Olasılıksal Değerin Geliştirilen Hali: $P_i = \exp(-1/1000 * f_i)$
Algoritma 6 (Lui, 2012)	Gözcü Arı Klasik Fazı: $v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$ Gözcü Arı Geliştirilen Fazı: $v_{ij} = x_{ij} - \Phi_{ij} * (x_{ij} - x_{kj})$

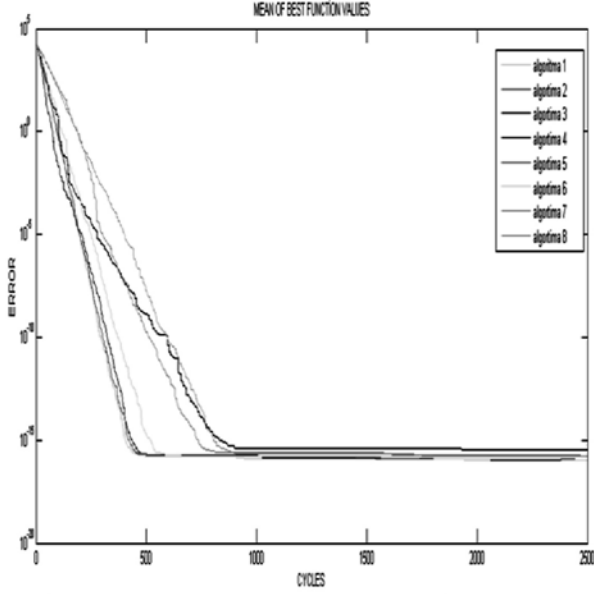
<p>Algoritma 7 (Sundareswaran, 2015)</p>	<p>Başlangıç Atama: $x_i = d_{\min} + \frac{(i-1)(d_{\max} - d_{\min})}{N_p - 1}$</p> <p>Gözcü Arı Fazı: $x_{i(k+1)} = x_{i(k)} + \frac{\phi * (d_{\max} - d_{\min})}{\frac{N_p}{2} - 1}$</p>
<p>Algoritma 8 (Zhang, 2013)</p>	<p>İşçi Arı Fazı: $v_{i,j} = \begin{cases} x_{i,j} + \phi(x_{i,j} - x_{r1,j}), & \text{if } j = j1 \\ x_{r2,j}, & \text{if } j = j2 \\ x_{i,j}, & \text{otherwise} \end{cases}$</p> <p>Gözcü Arı Fazı: $v_{i,j}^o = \begin{cases} x_{i,j} - \phi(x_{i,j} - x_{r1,j}), & \text{if } j = j1 \\ x_{i,j}, & \text{otherwise.} \end{cases}$</p>

Çizelge 2. Sphere, Schewefel ve Rosenbrock Fonksiyonları Uygulama Sonuçları

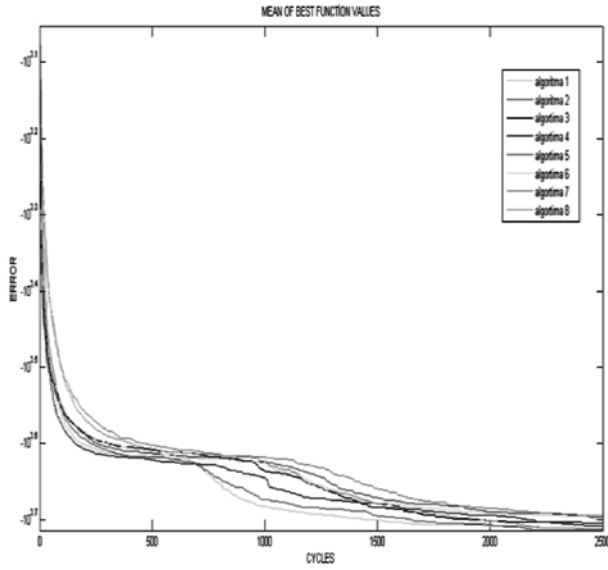
Sonuçlar 1			
Algoritma Adı	Sphere	Schewefel	Rosenbrock
Algoritma 1	1,13703e-016	-5168,52	2,95183
Algoritma 2	8,88178e-015	-5076,09	3,35714
Algoritma 3	2,31531e-016	-5065,13	3,80635
Algoritma 4	1,09219e-016	-5107,71	4,42979
Algoritma 5	1,09478e-016	-5175,57	2,80452
Algoritma 6	1,0262e-016	-5000,73	3,5518
Algoritma 7	1,77726e-016	-4944,58	3,14068
Algoritma 8	1,59306e-016	-5154,64	5,89876
Çalışma Aralığı	[-100,100]	[-500,500]	[-100,100]

Çizelge 3. Rastrigin, Griewank ve Ackley Fonksiyonları Uygulama Sonuçları

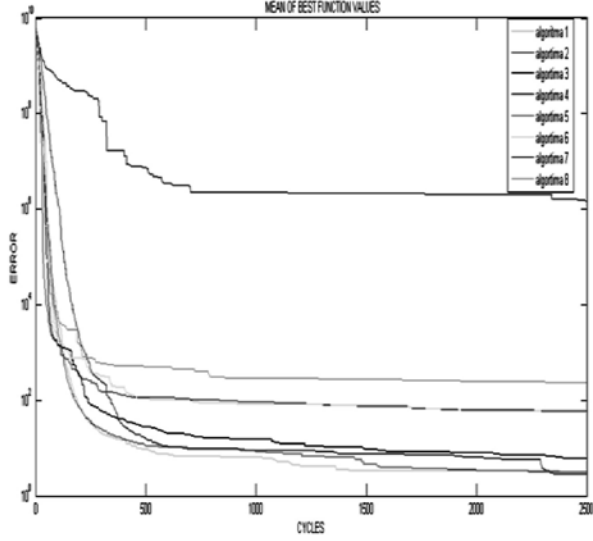
Sonuçlar 2			
Algoritma Adı	Rastrigin	Griewank	Ackley
Algoritma 1	1,22213e-013	0,00239463	1,03029e-014
Algoritma 2	1,93126e-012	0,0155535	2,91323e-014
Algoritma 3	1,64846e-013	0,0133844	5,70211e-014
Algoritma 4	2,34905e-012	0,0047115	1,10134e-014
Algoritma 5	2,98428e-014	0,00436597	1,04805e-014
Algoritma 6	4,41318e-012	0,0117487	1,1546e-014
Algoritma 7	1,65556e-013	0,0085004	2,30926e-014
Algoritma 8	4,76064e-014	0,0599055	2,91323e-014
Çalışma Aralığı	[-5,12,5,12]	[-600,600]	[-32,32]



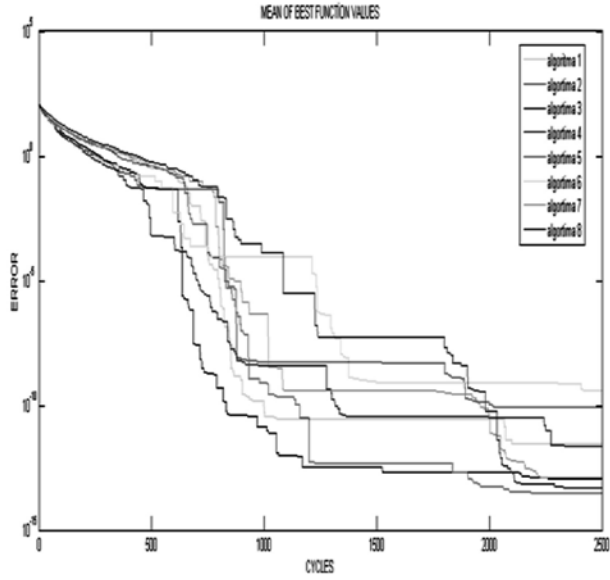
Şekil 3. Sphere Fonksiyonu Uygulama Grafik Sonuçları



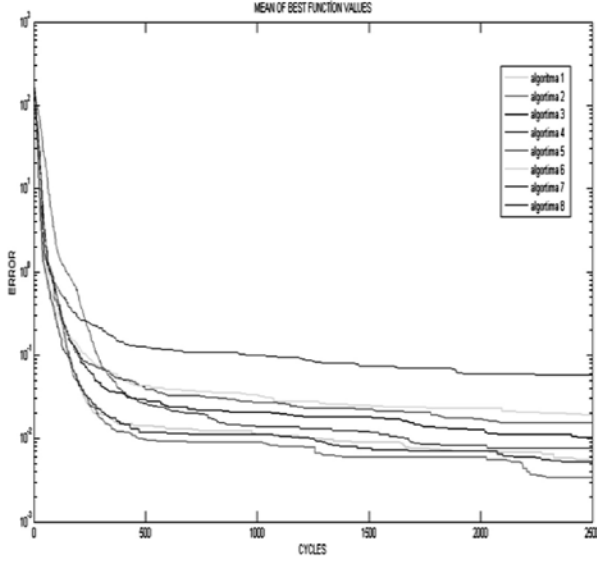
Şekil 4. Schwefel Fonksiyonu Uygulama Grafik Sonuçları



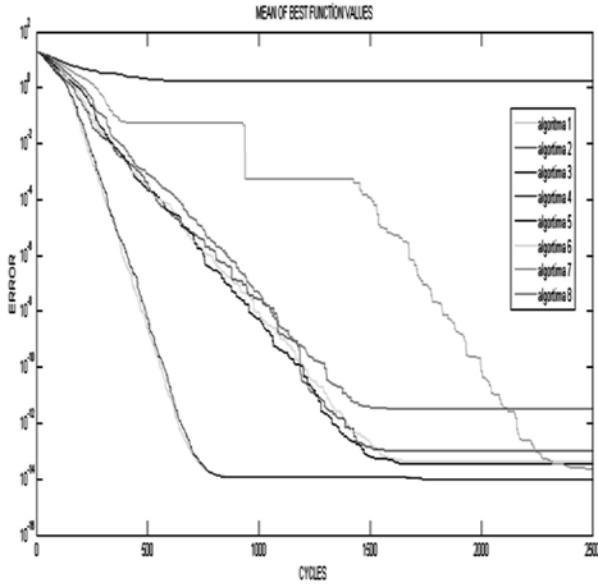
Őekil 5. Rosenbrock Fonksiyonu Uygulama Sonuları



Őekil 6. Rastrigin Fonksiyonu Uygulama Sonuları



Şekil 7. Griewank Fonksiyonu Uygulama Sonuçları



Şekil 8. Ackley Fonksiyonu Uygulama Sonuçları

6. SONUÇ

ABC birçok optimizasyon probleminde çözüm olanağı sağlamakta ve geliştirilen faz süreçleri ile en uygun sonuçları elde etmektedir. Bu çalışmada yapılan karşılaştırmalar dikkate alındığında her bir geliştirmenin farklı problemler üzerinde farklı sonuçlar elde edileceği görülmektedir.

Çizelge 1'de yer alan geliştirme fazlarından yola çıkarak elde edilen Çizelge 2 ve Çizelge 3 sonuçları incelendiğinde; Sphere Test Fonksiyonunda uygun çözüme gözcü arı fazındaki matematiksel değişiklik ile algoritma 6 (Lui, 2012)'de ulaşılmakta; Schwefel Test Fonksiyonunda uygun çözüme gözcü arı fazındaki matematiksel değişiklik ile algoritma 7 (Sundareswaran, 2015)'de ulaşılmakta; Rosenbrock Test Fonksiyonunda uygun çözüme olasılıksal değerde matematiksel değişiklik ile algoritma 5 (Babayiğit, 2012)'de ulaşılmakta; Rastrigin Test Fonksiyonunda uygun çözüme olasılıksal değerde matematiksel değişiklik ile algoritma 5 (Babayiğit, 2012)'te ulaşılmakta; Griewank Test Fonksiyonunda uygun çözüme klasik hali ile algoritma 1 (Karaboğa, 2011)'de ulaşılmakta; Ackley Test Fonksiyonunda uygun çözüme klasik hali ile algoritma 1 (Karaboğa, 2011)'de ulaşılmakta olduğu gözlemlenmiştir.

ABC konusunda araştırmacıların işçi arı fazı, gözcü arı fazı, başlangıç atama ve olasılıksal değer süreçlerinde uyguladıkları yeni matematik modelleri ve algoritmanın klasik hali 6 adet test fonksiyonunda incelenerek en uygun çözümler elde edilmeye çalışılmıştır.

Elde edilen veriler ışığında yapılan iyileştirmelerin her problemde istenilen sonucu veremeyeceği bu nedenle doğadaki karmaşık problemlerin çözümünde alternatif matematiksel modeller dikkate alınarak en uygun sonuca ulaşacak şekilde iyileştirmeler yapılması gerekmektedir.

ABC'nin diğer algoritmalara göre avantajlarını sıralayacak olursak;

1. Oldukça esnek ve basittir.
2. Gerçek yiyecek arayıcı arıların davranışlarına oldukça yakın şekilde benzetim eder.
3. Sürü zekasına dayalı bir algoritmadır.
4. Nümerik problemler için geliştirilmiştir ama ayrık problemler içinde kullanılabilir.
5. Oldukça az kontrol parametresine sahiptir.
6. Kaşif arılar tarafından gerçekleştirilen küresel ve görevli ile gözcü arılar tarafından gerçekleştirilen bölgesel araştırma kabiliyetine sahiptir ve iki araştırma da birlikte yürütülmektedir (Akay, 2009).

KAYNAKÇA

- AKAY, B., (2009), **Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi Ek-5 ABC Algoritması Kodları**, Doktora Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü 10-25.
- ARORA, J. S., (1989), **Introduction Optimum Design** , Mcgraw Hill 18.
- BABAYİĞİT, B., R. ÖZDEMİR, (2012), **A Modified Artificial Bee Colony Algorithm for Numerical Function Optimization**, Computers and Communications (ISCC), IEEE Symposium on 245-249.
- BONABEAU, E., M. DORIGO, G. THERAULAZ, (1999), **Swarm Intelligence: From Natural to Artificial Systems**, Oxford University Press, Ny,92.
- DUMITRESCU, I., STUTZLE, T., (2003), **Combinations of Local Search and Exact Algorithms, Applications of Evolutionary Computing**, LNCS, Volume 2611/2003, 57-68.
- EIBEN, A., SMITH, J., (2003), **Intorduction to Evolutionary Computing**, Springer 347-365.
- GAO, W., S. LIU and L. HUANG, (2013), **A Novel Artificial Bee Colony Algorithm Based on Modified Search Equation and Orthogonal Learning**, Ieee Transactions on Cybernetics, Vol. 43, No. 3, June,1011-1024.
- JEYA, D., V. MOHAN, M. KAMALAPRIYA, (2010), **Automated Software Test Optimisation Framework – An Artificial Bee Colony Optimisation-Based Approach**, IET Softw. , Vol. 4, Iss. 5, Pp. 334–348.
- KARABOĞA, D., (2011), **Yapay Zeka Optimizasyon Algoritmaları**, 202-221.
- KARABOĞA, D., AKAY B., (2009), **A Comparative Study of Artificial Bee Colony Algorithm**, Applied Mathematics and Computation 108–132.
- KORB, K., RANDALL, M., HENDTLASS, T., (2009), **Artificiallife: Borrowing from Biology**, Springer, 211-220.
- LIU, J., J. WANG, B. FENG, J. HUO, (2012), **Research on the Solving of Nonlinear Equation Group Based on Artificial Bee Colony Algorithm**, The 7th International Conference on Computer Science & Education, July 14-17, Melbourne, Australia, 75-79
- PACURIB, J., G. MAE, M. SENO, J. P. T. YUSIONG, (2009), **Solving Sudoku Puzzles Using Improved Artificial Bee Colony Algorithm**, Fourth International Conference on Innovative Computing Information and Control, 885-889.
- SUNDARESWARAN, K., P. SANKAR, P. NAYAK, S. SIMON, A. PALANI, (2015), **Enhanced Energy Output from a PV System Under Partial Shaded Conditions Through Artificial Bee Colony**, Ieee Transactions on Sustainable Energy, Vol. 6, No. 1, January, 198-209.
- TERESHKO, V., (2000), **Reaction–Diffusion Model of a Honeybee Colony’s Foraging Behaviour, in: Parallel Problem Solving from Nature PPSN VI, Lecture Notes in Computer Science**, Vol. 1917, Springer–Verlag, Berlin, Pp. 807–816.

- TERESHKO, V., A. LOENGAROV, (2005), **Collective Decision-Making in Honeybee Foraging Dynamics**, Computing and Information Systems Journal 9 (3).
- WAIBEL, M., (2006), **Divison of Labour and Colony Efficiency in Social Insects**, Proceedings of the Royal Society B., 273, 1815-23.
- ZHANG, X., S.YUEN, S. HO, W. FU, (2013), **An Improved Artificial Bee Colony Algorithm for Optimal Design of Electromagnetic Devices**, IEEE Transactions on Magnetics, Vol. 49, 4811-4816.