

PAPER DETAILS

TITLE: The use of genetic algorithm and particle swarm algorithm in determining egg freshness

AUTHORS: Hasan Alp SAHIN, Hasan ONDER

PAGES: 81-88

ORIGINAL PDF URL: <https://dergipark.org.tr/tr/download/article-file/1053495>



Araştırma Makalesi (Research Article)

Cilt 3 - Sayı 3: 81-88 / Temmuz 2020
(Volume 3 - Issue 3: 81-88 / July 2020)

GENETİK ALGORİTMA VE PARÇACIK SÜRÜ ALGORİTMASININ YUMURTA TAZELİĞİNİN BELİRLENMESİNDE KULLANIMI

Hasan Alp ŞAHİN^{1*}, Hasan ÖNDER¹

¹Ondokuz Mayıs Üniversitesi Ziraat Fakültesi Zootekni Bölümü, 55139, Samsun, Türkiye

Gönderi: 04 Şubat 2020; **Kabul:** 10 Mart 2020; **Yayınlanma:** 01 Temmuz 2020
(**Received:** February 04, 2020; **Accepted:** March 10, 2020; **Published:** July 01, 2020)

Özet

Bu çalışmada yapay zeka uygulamaları grubunda yer alan genetik algoritma optimizasyonu (GAO) ve parçacık sürü algoritması optimizasyonu (PSO) etkinliklerinin belirlenmesi amaçlanmıştır. Deneme materyali olarak 50 adet yumurta 29 gün boyunca fotoğraflanmış ve elde edilen görüntüler veri olarak kullanılmıştır. Elde edilen bulgulara göre, PSO sınıflandırmasından elde edilen belirtme katsayısı 0,07 olarak ve GAO sınıflandırmasından elde edilen belirtme katsayısı 0,14 olarak elde edilmiştir. Yumurta tazeliğinin belirlenebilmesi amacıyla kullanılan GAO ve PSO algoritmalarından elde edilen sonuçlar her iki yöntemin de belirlenen amaç açısından yetersiz olduğunu göstermektedir. Elde edilen belirtme katsayıları oldukça düşük olup bu iki yöntemin yumurta tazeliğinin belirlenmesi amacıyla kullanılamayacağı anlaşılmıştır.

Anahtar kelimeler: Genetik algoritma, Parçacık sürü optimizasyonu, Yumurta, Tazelik

The Use of Genetic Algorithm and Particle Swarm Algorithm in Determining Egg Freshness

Abstract: In this study, it is aimed to determine the genetic algorithm optimization (GAO) and particle swarm algorithm optimization (PSO) activities in the artificial intelligence applications group. As experimental material, 50 eggs were photographed for 29 days and the images were used as data. According to the findings, the coefficient of determination obtained from the PSO classification was 0.07 and the coefficient of determination obtained from the GAO classification was 0.14. The results obtained from the GAO and PSO algorithms used to determine the freshness of the egg show that both methods are insufficient for the specified purpose. The coefficient of determination obtained were quite low and it was understood that these two methods could not be used to determine the freshness of eggs.

Keywords: Genetic algorithm, Particle swarm algorithm, Egg, Freshness

***Corresponding author:** Ondokuz Mayıs Üniversitesi Ziraat Fakültesi Zootekni Bölümü, 55139, Samsun, Türkiye

E mail: h.alpsahin@gmail.com (H.A. ŞAHİN)

Hasan Alp ŞAHİN  <https://orcid.org/0000-0002-7811-955X>

Hasan ÖNDER  <https://orcid.org/0000-0002-8404-8700>

Cite as: Şahin HA, Önder H. 2020. The use of genetic algorithm and particle swarm algorithm in determining egg freshness. BSJ Eng Sci, 3(3): 81-88.

1. Giriş

İnsan beyninin çalışma prensipleri göz önüne alınarak beynin işleyişi ve fonksiyonel yapısı konusunda yapılan birçok çalışma, bugün yapay zeka alanında yapılan çalışmaların temellerini oluşturmuştur (Chatterjee ve Laudato, 1995). İnsanoğlunun zekasının modellenmesi ile bir takım algoritmalar ile bilgisayarlara ve bilgisayarlar tarafından denetimi yapılan makinelere yapay zeka yardımıyla problemlerin çözülmesi asıl amaçtır (Baş, 2006).

Beynin bütün yapısının ve fonksiyonlarının doğru bir şekilde modellenmesi içinde yapay hücre ve ağ modelleri geliştirilmiştir. Yapay zekaya 1950'li yıllarda başlayan bu ilgi gitgide artış göstermiş ve günümüzde de gün geçtikçe aynı ilginin yenilenmesine ve artmasına neden olmuştur. Böylelikle günümüz bilgisayarlarının klasik algoritmik hesaplama yöntemlerinden farklı bir alanın doğmasını sağlamıştır (Çanakçı ve Hosoz, 2006). Yapay zeka teknolojisinin yaygın olarak kullanıldığı kullanım sahaları, genetik algoritma, uzman sistemler, yapay sinir ağları ve bulanık mantık olarak sıralamak mümkündür (Elmas, 2003). Genetik algoritma Darwin'in Genetik algoritma ise Darwin'in evrim teorisi üzerine kurulu bir yöntem olup, bu algoritmada bir veri kümesinin özel bir veriyi bulması istendiği durumlarda kullanılmaktadır. Bilinen klasik yöntemlerin yetersiz kaldığı karmaşık ve zor problemleri sanal olarak evrimden geçirerek optimum çözümü bulmayı amaçlayan bir algoritmadır.

Parçacık sürü ağları da genetik algoritmalarla benzer olarak rastgele çözümlerden oluşan popülasyonlar ile başlayarak potansiyel çözümleri güncelleyerek optimum sonuca ulaşmaya çalışır. Genetik algoritmalarla bulunan mutasyon ve çaprazlama gibi evrimsel operatörler parçacık sürü ağlarında bulunmazlar. Parçacık sürü optimizasyonunda parçacık olarak adlandırılan potansiyel çözüm noktaları, o anlık en iyi çözüm noktalarını izlerler.

Metasezgisel algoritmalar, çeşitli mühendislik alanlarında yer alan farklı problemlerin çözümünde günümüzde sıklıkla kullanılmaktadırlar. Bu yöntemlerin kullanılmasıyla büyük boyutlu problemlerde daha az hesaplama karmaşıklığı ile klasik yöntemlerle elde edilemeyecek kadar kısa sürede iyi çözümler ortaya çıkabilmektedir. Özellikle endüstri mühendisliği alanında araç rotalama problemi, gezgin satıcı problemi, atölye tipi ve akış tipi iş çizelgeleme problemi, zaman çizelgeleme, personel atama gibi çok çeşitli sorunlarda bu yöntemler başarıyla kullanılabilirler.

Sınıflandırma amacıyla kullanılan tekniklerden biri olan görüntü işleme tekniği, tarayıcı, kamera vb. araçlar tarafından bilgisayara aktarılan görüntülerin incelenmesini sağlan önemli tekniklerden biridir (Demirbaş ve Dursun, 2007). Görüntü işleme teknolojileri ile verimlilik artarken üretim maliyetleri de azalmaktadır (Chen ve ark., 2010).

Yumurta insan beslenmesinde çok önemli, uygun fiyatlı ve besleyici ürünlerden biridir. Taze yumurtaların

kalitesi zamanla azalır. Bununla birlikte, depolama günlerinin sayısı yumurta kalitesini etkileyen faktörlerden biridir (Abdel-Nour, 2011).

Yumurta pürüzsüz düz bir yüzeye kırıldığında, yumurta sarısı genellikle kalın albüminle çevrili merkezi bir konumdadır (Karoui ve ark., 2006; Robinson ve Monsey 1972; Wells ve Norris, 1987). Yumurta kalitesinin değerlendirilmesi için çeşitli yöntemler kullanılır. Bu yöntemler iki gruba ayrılır: yumurtalar kırılarak ve kırılmadan yöntemler. Haugh birimi (HU), yumurta kalitesini ölçmek için en yaygın kullanılan yumurta kırılarak elde edilen yöntemdir (Haugh, 1937). Bu yöntem, sağlam yumurtanın ağırlığı ile yumurtanın kırıldıktan sonra albümin yüksekliğinin ölçülmesiyle elde edilen değerler arasındaki ilişkiye dayanmaktadır. Yumurtanın yumurtlanmasından sonra zaman geçtikçe, su kaybına bağlı olarak bozulmamış yumurtanın ağırlığı ve albümin viskozitesi azalır (Abdel-Nour, 2011). Yumurtaların kırılması ile birlikte yumurtaların bireysel tazelikleri ölçülür. Ancak endüstriyel açıdan ve tüketici açısından yumurtaların kırılmadan tazeliklerinin belirlenmesi ekonomik açıdan önemlidir. Yumurtaların kırılmadan tazeliklerinin ölçümü için çeşitli çalışmalar yapılmıştır (Aboonajmi ve ark., 2014; Aboonajmi ve Najafabadi, 2014; Abdel-Nour, 2011; Karoui ve ark., 2008).

Bu çalışmanın amacı, buzdolabında 29 gün depolanan yumurtalarda, görüntü işleme yöntemleri ile parçacık sürü optimizasyonu ve genetik algoritmalar yöntemlerini kullanarak yumurta tazeliklerinin yumurtaya zarar verilmeden belirlenmesidir.

2. Materyal ve Metot

2.1. Materyal

Denemede Leghorn yumurtacı ırkı yetiştirilen ticari bir işletmeden elde edilen, beyaz kabuklu yumurtalar kullanılmıştır. Deneme materyali olarak 50 adet aynı gün yumurtlanan yumurtalar kullanılmıştır. Yumurta fotoğraflarını bilgisayara aktarmak amacıyla uçayak ile sabitlenmiş, Canon 550D fotoğraf makinesi kullanılmıştır. Çekilen fotoğraflar 18 MP boyutunda ve sabit iso değerinde kaydedilmiştir. Uygulanan optimizasyonlar MATLAB programı kullanılarak gerçekleştirilmiştir.

2.2. Fotoğrafların Elde Edilmesi

Yumurtalar buzdolabında saklanarak, sadece fotoğraf çekileceği zaman aynı anda dışarı çıkartılmış aynı anda buzdolabına konmuştur. Yumurtaların sivri kısımları aşağı gelecek şekilde düzeneğe yerleştirilmiştir. Sabit ışık miktarı ve tripot yardımıyla sabit kamera kullanılmıştır. Elde edilen fotoğraflarda yumurtaların arka planında bulunan siyah kısımlardaki kirli noktaları görüntü işleme programı yardımıyla temizlenmiştir (Şekil 1). Temizlenen fotoğraflar 2000×2000 piksel boyutunda kesilmiştir (Şekil 2).



Şekil 1. Temizlenmiş fotoğraf.



Şekil 2. Kesilmiş fotoğraf.

Matlab programı yardımıyla elde edilen görüntüler 2000×2000 pikselden 300×300 piksel boyutuna küçültülmüştür.

2.3. Ham Verilerin Oluşturulması

Programa tanımlanan her bir fotoğraf boyutu 300×300×3 boyutunda 270000 adet sayıdan oluşmaktadır. 300X300 (90000) en boy verisini oluştururken, her bir değer için RGB (kırmızı, yeşil, mavi) değeri verilerin ×3'lük kısmını oluşturmaktadır.

- Veriler program yardımıyla 90000×3 şeklinde yukarıdan aşağı yazdırılmıştır.
- Verilerin determinantları alınmıştır.
- Veriler ham halleri ile determinantları çarpılmıştır.
- Elde edilen veriler 3×3'lük matris şeklinde elde edilmiştir.

$$\begin{bmatrix} R^2 & RG & RB \\ RG & G^2 & GB \\ RB & GB & B^2 \end{bmatrix}$$

Elde edilen matrisin;

1. satır 1. sütununda kırmızı rengin kareler toplamı yer almaktadır.
2. satır 2. sütununda yeşil rengin kareler toplamı yer almaktadır.

3. satır 3. sütununda mavi rengin kareler toplamı yer almaktadır.

1. satır 2. sütununda, 2. satır 1. sütununda kırmızı ve yeşil renklerin çarpımlarının toplamı yer almaktadır.

1. satır 3. sütununda, 3. satır 1. sütununda kırmızı ve mavi renklerin çarpımlarının toplamı yer almaktadır.

2. satır 3. sütununda, 3. satır 2. satırında yeşil ve mavi renklerin çarpımlarının toplamı yer almaktadır.

Elde edilen 3×3'lük matris, tek sütuna çevrilmiş, 1×9 boyuna getirilmiştir.

Elde edilen her bir fotoğraf 9 rakamdan oluşturulmuştur. 50 adet yumurtadan 29 gün boyunca 1450 fotoğraf toplanmıştır. Giriş matrisi 9×1450 boyutunda (13050 adet sayı) oluşturulmuştur.

2.4. Hedef Matrisini Oluşturulması

Giriş verilerine karşılık tahmin edilmek istenen 29 gün hedef olarak belirlenmiştir. 1450 sütundan oluşan giriş matrisine karşılık hedef matrisi 1 satır 1450 sütundan oluşturulmuştur. Her bir fotoğrafı oluşturan 1×9'luk matrise bir gün değeri hedef olarak girilmiştir.

2.5. Genetik Algoritma Optimizasyonu (GAO)

Genetik algoritma, optimizasyon uygulanacak problemi, Darwin'in evrim teorisini göz önünde bulundurarak, sanal bir evrim sürecinden geçirmektedir. Basit genetik algortmada gerçekleştirilen bu işlemler, seçim (kopyalama), çaprazlama ve mutasyondur. Genetik algortmada kullanılan birtakım terimler aşağıda açıklanmaktadır (Dede, 2003).

Birey: Literatürdeki genel adının kromozom olduğu bilinmektedir. Problemin çözümü olabilecek tasarım değişkenlerinin belirli sıraya göre dizilmesinden oluşmaktadır.

Ebeveyn Birey: Genetik işlem görüp yeni birey (çocuk birey) oluşturabilen herhangi bir bireydir.

Nesil: Daha iyi ebeveyn bireyler elde etmek için belirli sayıda bireyin katılımıyla genetik algortmada oluşturulan herhangi bir toplumdur (Popülasyon).

Uygunluk: Tasarım başarımının ölçüsüdür.

Genetik İşlem: Ebeveyn bireyler arasında bilgi iletişimini sağlayan işlemlerdir.

Genetik algortmada, her bir çözüm nesildeki bir bireyle (kromozomla) gösterilmekte, bireyleri de sayı dizileri simgelemektedir.

2.5.1. Genetik algortmaların işleyişi

Bir genetik algortmada, ilk olarak tasarım değişkenlerinin diziler ile temsil edildiği muhtemel çözümlerden oluşan bir başlangıç nesli belirlenmektedir. Genetik algortmalar ayrıık tasarım değişkenlerini kullanabilmekte ve bu tasarım değişkenlerinin alabileceği değerler tasarıma başlamadan önce belirlenmektedir. Çözüm dizilerindeki tasarım değişkenlerinin kodlama işlemi de, çözümü oluşturan tasarım değişkeni değerinin değer kümesindeki sıra numarasının kodlanması şeklinde uygulanmaktadır. Bu kodlama işlemi, sıra numarasının ikili ya da üçlü sayı sisteminde kodlanması olabileceği gibi gerçek değerlerin kodlanması şeklinde de olabilmektedir. Bu yaklaşım içinde çözüm dizilerinin oluşturduğu kümenin nesle,

çözümlerin birer bireye ve çözüm dizilerini oluşturan rakamların ise genlere benzetilmesi mümkün olmaktadır. Başlangıç neslinin oluşturulmasının ardından evrimsel süreç başlamaktadır. Bu süreçte ilk olarak, nesildeki her çözümün uygunluk değeri belirlenmekte, belirlenen uygunluk değerine göre çaprazlama ve mutasyon işlemleri uygulanmaktadır. Bu işlemler, optimum çözüm bulununcaya kadar tekrarlanmaktadır.

2.5.2. Genetik işlemlerin olası çözümlere uygulanması

Genetik algoritmalar, çözüm topluluğuna adım adım genetik işlemler uygulayarak ve uygun topluluktan arama yoluyla yeni nesiller üreterek en iyi çözümlere ulaşılmasını sağlamaktadır. Basit genetik algoritmaların çoğu dört ana işlemten oluşmaktadır (Parlak, 2007).

Seçim

Seçim, topluluğu oluşturan her dizinin uygunluk değerlerine (amaç fonksiyonu değerleri) bağlı olarak kopyalanması işlemi olarak bilinmektedir. Bu işlem “doğal seçilimin” bilgisayar ortamındaki karşılığı olarak da görülebilmektedir. Topluluğu oluşturan her dizi, problemin olası çözümlerinden birini temsil etmekte ve seçim işlemi en uygun çözümlerin daha sonraki çözümleri (yeni toplulukları) daha yüksek olasılıkla etkilemesini amaçlamaktadır. Uygunluk değeri saptanan diziler, bir seçim mekanizması ile kopyalama işlemine uğratılmaktadır (Koç, 2002).

Kopyalama

Kopyalama işlemi uygunluk değeri kötü olan bireylerin uygunluk değeri iyi olanlarla değiştirilmesini sağlamaktadır. Bu işlemde nesildeki tüm bireylerin uygunluk değerleri hesaplanmakta ve önceden belirlenmiş bir değerden daha küçük uygunluğa sahip olan bireyler nesilden çıkarılmaktadır. Nesilde kalan iyi bireylerden uygunluk değeri yüksek olanlar çıkarılan birey sayısı kadar kopyalamak suretiyle nesildeki toplam birey sayısı tamamlanmaktadır. Eşleme havuzuna alınan bu bireyler birbirleriyle rasgele eşlenmekte ve çaprazlama işlemi bu eşler arasında gerçekleştirilmektedir. Tam bir eşleme olması için birey sayısının çift olması gerekmektedir (Öztürk, 2013).

Çaprazlama

Çaprazlama işlemi, seçim ve kopyalama işlemleriyle elde edilen dizilerden yeni diziler, başka bir deyişle yeni çözümler üretilmesi olarak tanımlanmaktadır. Bu işlem, öncelikle mevcut dizilerin rasgele eşleştirilmesi ve daha sonra eşleştirilen diziler arasında belirli kısımların karşılıklı değiştirilmesi şeklinde gerçekleştirilmektedir (Koç, 2002).

Çaprazlama işleminde nesildeki bireylerin kendisiyle eşlenmesine izin verilmemekte ancak birey kendilerinden üretilen kopyalarla eşlenebilmektedir. Çaprazlama genetik algoritmada yakınsama olasılığını artırmakta ve böylece topluma yeni bireyler kazandırmaktadır. Çaprazlama işlemiyle her iterasyonda yeni bireyler üretilerek genetik algoritmanın bu yeni bireyler üzerinde araştırma yapması sağlanmaktadır. Bu işlemi uygulamak üzere eşleme havuzunda eşleştirilen

bireyler çaprazlama havuzuna alınarak burada rasgele çaprazlama noktaları belirlenmektedir (Öztürk, 2013).

Mutasyon işlemleri

Mutasyon işlemi, genetik algoritmanın erken yakınsamasını önlerken, düşük uygunluk değerine sahip bireylerin problemde çözüm olarak belirlenmesini de önlemeye çalışmaktadır. Mutasyon işlemi uygunluk değeri yüksek bireylere uygulanarak algoritmanın erken yakınsamaması sağlanmaktadır. Bununla birlikte bu işlem, rasgele seçilen bireylere de uygulanabilmektedir. Bazı araştırmacılar mutasyon işleminin en iyi bireyi nesilden çıkarabileceği gerekçesiyle bu işlemin kullanılmamasını, kullanılacaksa da kullanılma oranının düşük tutulmasını önermektedirler (Öztürk, 2013). Ele alınan problemin yapısına göre aşağıdaki mutasyon türlerinden en uygun olanının seçilerek uygulanması gerekmektedir (Bolat, 2006).

- Ters Çevirme: Kromozomdan rasgele iki bölge seçilir ve iki ucu arasında ters çevrilir.
Kromozom A: 0 1 1 1 0 0 0 1 0 1
Mutasyon A: 0 1 0 0 1 1 0 1 0 1
- Ekleme: Rasgele bir parça seçilir ve rasgele bir yere yerleştirilir.
Kromozom A: 0 1 1 1 0 0 0 1 0 1
Mutasyon A: 0 1 0 1 0 0 0 1 0 1
- Yer Değişikliği: Rasgele bir alt dizi seçilir ve rasgele bir yere yerleştirilir.
Kromozom A: 0 1 1 1 0 0 0 1 0 1
Mutasyon A: 0 1 1 1 0 0 1 1 0 0
- Karşılıklı Değişim: Rasgele seçilen iki genin yerleri değiştirilir.
Kromozom A: 0 1 1 1 0 0 0 1 0 1
Mutasyon A: 0 0 1 1 0 1 0 1 0 1

2.6. Parçacık Sürü Algoritması Optimizasyonu (PSO)

Genetik

Kuş sürülerinin davranışlarından esinlenilerek geliştirilmiş, popülasyon tabanlı stokastik optimizasyon tekniğidir (Kennedy ve Eberhart, 1995). Bu algoritma doğrusal olmayan problemlerin çözümü için tasarlanmıştır. Çok parametrelili ve çok değişkenli optimizasyon problemlerine çözüm bulmak için kullanılmaktadır.

PSO, genetik algoritmalar gibi evrimsel hesaplama teknikleriyle birçok benzerlik gösterir. Genetik algoritmada olduğu gibi sistem rastgele çözümlere sahip bir popülasyon ile çalışmaya başlar ve belirli nesiller boyunca en iyi çözümü araştırır. Ancak PSO’da mutasyon ve çaprazlama gibi genetik operatörler yer almaz. Bunun yerine, ‘parçacık’ olarak adlandırılan, çeşitli parametre değerlerinden oluşan ve birer pozisyon ve hız vektörüne sahip olan olası çözümler yer almaktadır (Arumugam vd, 2007). Bu parçacıklar, o ana kadar elde edilen en iyi parçacıkları izleyerek problem uzayında dolaşırlar. Uygulama kolaylığı ve kabul edilebilir uygun çözümlere hızlı bir biçimde yakınsaması nedeniyle, PSO metodu çok popülerlik kazanmıştır (Ratnaweera ve Halgamuge, 2004).

PSO'nun klasik optimizasyon tekniklerinden en önemli farklılığı türev bilgisine ihtiyaç duymamasıdır. PSO'yu uygulamak, algoritmasında ayarlanması gereken parametre sayısının az olması sebebiyle oldukça basittir. PSO; fonksiyon optimizasyonu, bulanık sistem kontrolü, yapay sinir ağı eğitimi gibi birçok alanda başarıyla uygulanabilmektedir.

PSO algoritması kuş sürülerinin davranışlarının bir benzetimidir. Kuşların uzayda, yerini bilmedikleri yiyeceği aramaları, bir probleme çözüm aramaya benzetilir. Kuşlar yiyecek ararken yiyeceğe en yakın olan kuşu takip ederler. Parçacık olarak adlandırılan her bir tekil çözüm, arama uzayındaki bir kuştur. Parçacık hareket ettiğinde, kendi koordinatlarını bir fonksiyona gönderir ve böylece parçacığın kalite değeri ölçülmüş olur. Bu ölçüm kuşun yiyeceğe ne kadar uzaklıkta olduğunu değeridir. Bir parçacık, koordinatlarını, hızını, şimdiye kadar elde ettiği en iyi kalite değerini ve bu değeri elde ettiği koordinatları hatırlamalıdır. Çözüm uzayındaki her boyuttaki hızının ve yönünün her seferinde nasıl değişeceği, komşularının en iyi koordinatları ve kendi bireysel en iyi koordinatlarının birleşimi olacaktır.

PSO algoritması, tüm parçacıkların arama uzayına rastgele olarak yerleştirilmesiyle başlar ve her adımda parçacıkların pozisyonları güncellenir. Belirli bir hız ile hareket eden bu parçacıklar en iyi sonucu bulmak için arama işlemini sürdürürler (Zhang ve ark., 2007).

Klasik PSO modeli, reel değerli arama uzayında rastgele olarak üretilmiş ve n boyutlu uzayda hareket eden parçacık sürüsünden oluşmaktadır. PSO formülündeki (i,j) gösterimi, i . parçacığın j . boyutunu belirtmektedir ($j \in 1,2,...,n$). Çözümler arasında kıyaslama yapabilmek için bir kalite değeri (f) belirlenmiştir. Her bir parçacık X , pozisyon vektörü ve V , hız vektörüne sahiptir. Ayrıca, her bir parçacık küçük bir hafızaya sahiptir. Bu hafızada P_{eniyi} ile gösterilen ve o ana kadarki elde edilen kendi en iyi pozisyonu ile tüm parçacıklar tarafından elde edilen evrensel (global) en iyi pozisyon (g_{eniyi}) yer alır (Fan ve Chiu, 2007). Herhangi bir adımda i . parçacığa ait en iyi pozisyon vektörü (p_{best_i}) eşitlik (1)'de gösterilmiştir.

$$p_{best_i} = [P_{i1}, P_{i2}, \dots, P_{in}] \quad (1)$$

Tüm parçacıklar için g_{best} vektörü her iterasyonda tektir ve eşitlik (2)'de gösterilmiştir.

$$g_{best_i} = [P_1, P_2, \dots, P_n] \quad (2)$$

Sürüdeki i . parçacığa ait pozisyon ve hız vektörleri de sırasıyla eşitlik (3) ve (4)'te gösterilmiştir.

$$x_i = [X_{i1}, X_{i2}, \dots, X_{in}] \quad (3)$$

$$V_i = [V_{i1}, V_{i2}, \dots, V_{in}] \quad (4)$$

Her bir t adımında, parçacıklara ait yeni hız değeri olan $V(t+1)$ 'i elde etmek amacıyla, önceki hız değeri olan $V(t)$, (5) eşitliğine bağlı olarak güncellenir. (7) eşitliğindeki V_{max} , aşırı hızlanmaları önlemek amacıyla kullanıcı tarafından belirlenen ve parçacıkların sahip olabileceği maksimum hızı gösteren bir sabittir. (6) eşitliğinde gösterildiği şekilde, önceki pozisyon değerine yeni hız değerinin eklenmesi ile parçacık yeni pozisyonuna hareket etmektedir.

$$V_{i,j}(t+1) = V_{i,j}(t) + c_1 r_1 * (p_{best_{i,j}}(t) - X_{i,j}(t)) + c_2 r_2 * (g_{best_j}(t) - X_{i,j}(t)) \quad (5)$$

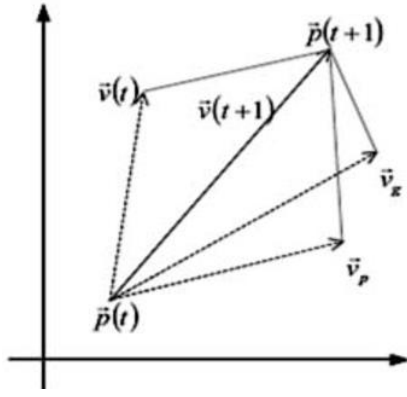
$$X_{i,j}(t+1) = X_{i,j}(t) + V_{i,j}(t+1) \quad (6)$$

$$\left\{ \begin{array}{ll} V_{i,j}(t+1) = V_{max}, & \text{if } V_{i,j}(t+1) > V_{max} \\ V_{i,j}(t+1) = -V_{max}, & \text{if } V_{i,j}(t+1) < -V_{max} \end{array} \right\} \quad (7)$$

Yukarıdaki formülde yer alan c_1 ve c_2 , hızlandırma katsayıları olarak adlandırılırlar. Bunlar genellikle $[0-2]$ aralığında belirlenen iki pozitif sabittir. c_1 ve c_2 katsayıları belirlemede sabit bir değer olmadığı gibi aynı problemlerin çözümünde farklı değerler kullanılabilir. Optimum sonuca ulaşabilmek üzere seçilen c katsayısının değerinin artırılması çözüm hızı artırırken hatayı artırmaktadır. Aynı zamanda c katsayısının daha düşük değerlerde seçimi optimum sonucu ulaşımı yavaşlattığı gibi hatayı da artırmaktadır. Problemlerin çözümünde hız ve sapmayı optimum düzeyde tutmak üzere farklı problemler için farklı katsayıların denenmesi gerekmektedir. Aynı formüldeki r_1 ($r_1 \sim U(0,1)$) ve r_2 ($r_2 \sim U(0,1)$) ise $[0-1]$ aralığında Uniform dağıtılmış rastgele üretilen sayılardır (Fan ve Chiu, 2007).

(5)'te ortaya konan formülü üç kısımda incelemek mümkündür. İlk kısım önceki hız bilgisini barındırmaktadır. İkinci kısım idrak kısmı olarak adlandırılır ve parçacığın kendi görüşünü ortaya koymaktadır. Üçüncü kısım ise sosyal kısım olarak adlandırılır ve parçacıklar arasındaki işbirliğini ortaya koymaktadır (Shi ve Eberhart, 1998). Bu formül ile parçacığa ait yeni hız değeri hesaplanmaktadır. Bu hesaplama için parçacığın önceki hız değeri, parçacığın o anki pozisyonunun hem kendi en iyi pozisyonuna hem de evrensel en iyi pozisyona olan uzaklığı bilgileri kullanılmaktadır. Yapılan bu hesaplamadan sonra parçacık (6) formülü ile yeni konumuna hareket etmektedir. Son olarak da çözülecek olan problem ile ilgili önceden belirlenmiş bir kalite fonksiyonuna bağlı olarak parçacığın performansı belirlenir.

PSO'nun temelinde yatan fikir, iyi çözümler hakkındaki bilginin sürünün tamamına yayılması ile sürüdeki tüm parçacıkların arama uzayındaki elverişli alanlara yönelme eğiliminde olmalarıdır. Şekil 3'de de gösterildiği gibi, parçacıkların hareketi $V(t)$, p_{best} ve g_{best} değerlerine bağlı olarak gerçekleşmektedir.



Şekil 3. PSO'da parçacık hareketi.

$P(t)$: Geçerli pozisyon

$P(t+1)$: Değiştirilmiş pozisyon

$V(t)$: Geçerli hız

$V(t+1)$: Değiştirilmiş hız

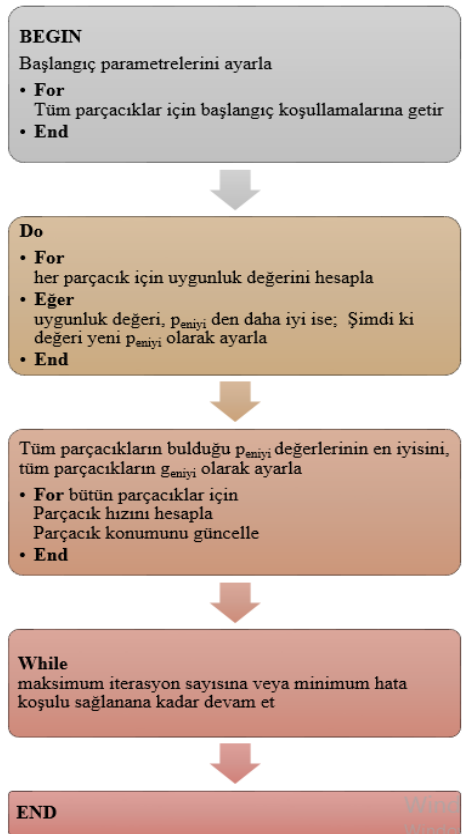
V_p : P_{best} 'in hızı

V_g : G_{best} 'in hızı.

2.6.1. Algoritma

PSO' da her parçacığın kendine özgü bir hızı olduğu için bu parçacığın hızı diğer parçacıklardan edindiği bilgilerle optimuma ulaşır. Her döngüde bu hız daha önceki en iyi sonuçlara göre tekrar hesaplanır. Her döngüde sürü daha iyi konuma gelir (Özsağlam, 2009).

PSO algoritması için gerekli olan prosedür Şekil 4'de verilmiştir (Tamer ve ark., 2006; Eldem, 2014).

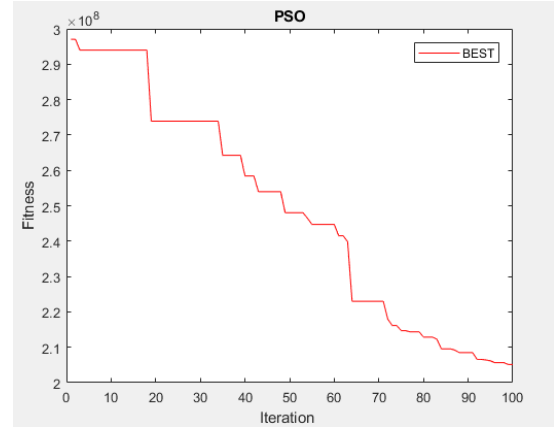


Şekil 4. PSO prosedürleri şeması.

3. Bulgular ve Tartışma

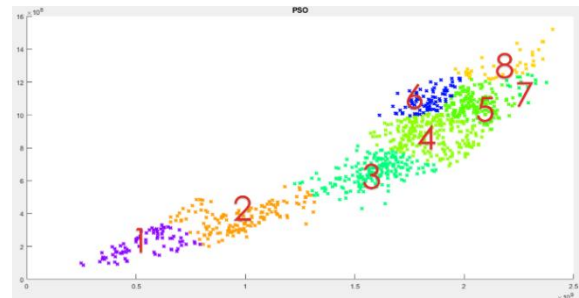
3.1. Parçacık Sürü Algoritması

PSO algoritmasında 29 sınıf sayısı 50 parçacık sayısı ve 100 maksimum iterasyon sayısı kullanılmıştır. Kullanılacak olan iterasyon sayısının belirlenmesi için farklı iterasyon sayıları kullanılmıştır. Yapılan ön denemeler sonucunda iterasyon sayısının 100'ün üzerinde olmasının sonuç üzerine etkisinin olmadığı belirlenmiştir (Şekil 5). Bu nedenle kullanılan iterasyon sayısının yeterli olduğu belirlenmiştir.



Şekil 5. İterasyon grafiği.

Parçacık sürü algoritması optimizasyonu uygulanan yumurtalarda, PSO'dan 29 sınıfa bölmesi beklenirken PSO verileri 8 sınıfa ayırabilmiştir (Şekil 6). PSO sınıflandırmasından elde edilen belirtme katsayısı 0,07 olarak elde edilmiştir.

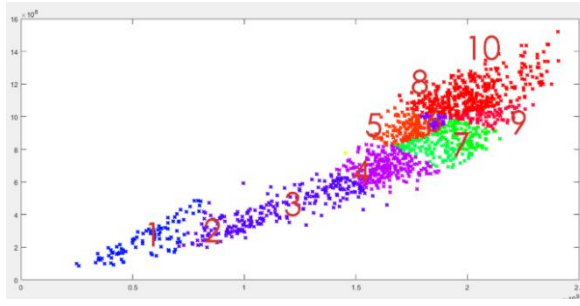


Şekil 6. PSO tahmin grafiği.

3.2. Genetik Algoritma

Yapılan çalışmada GA yöntemi için 50 iterasyon, 29 sınıf ve 10 popülasyon büyüklüğü kullanılmıştır. Maksimum 50 iterasyon sayısı yeterli olmuştur.

Uygulanan genetik algoritma optimizasyonu ile birlikte GA'dan beklenen 29 sınıf yerine GA'lar 10 sınıf oluşturabilmiştir (Şekil 7). GAO sınıflandırmasından elde edilen belirtme katsayısı 0,14 olarak elde edilmiştir.



Şekil 7. GAO tahmin grafiği.

4. Sonuç ve Öneriler

Yumurta tazeliğinin belirlenebilmesi amacıyla kullanılan genetik algoritma optimizasyonu ve parçacık sürü optimizasyonu algoritmalarından elde edilen sonuçlar her iki yöntemin de belirlenen amaç açısından yetersiz olduğunu göstermektedir. Elde edilen belirtme katsayıları oldukça düşük olup bu iki yöntemin amaç dahilinde kullanılamayacağını açıkça vurgulamaktadır.

Genetik algoritma optimizasyonun başarısızlığının nedeni büyük boyutlu verilerde rastgele seçimler konusundaki kararsızlıktan (Heris ve Oskoei, 2014) ya da GA optimizasyonunun tamamen şansa bağlı ve olasılıksal olmasından (Syahputra ve ark., 2016) kaynaklanabilir.

Parçacık sürü optimizasyonu algoritmasının başarısızlığının nedeni olarak yerel arama kabiliyeti ve küresel arama yeteneği arasında kontrol edebileceği denge konusunda taşınmış olduğu eylemsizlikten kaynaklanan büyük ölçekli kombinasyonel optimizasyon probleminin çözümünde kolayca yerel optimumluğa düşme eğilimi olabilir (Meng ve ark., 2017; Hemasian-Etefagh ve Safi-Esfahani, 2019).

Yumurta tazeliğinin belirlenebilmesi amacıyla diğer optimizasyon yöntemlerinin kullanılarak etkinliklerinin belirlenmesi, bu algoritmaların melezlenmesiyle oluşturulan yeni hibrit algoritmaların denemesi yararlı olabilir.

Çıkar İlişkisi

Yazarlar bu çalışmada hiçbir çıkar ilişkisi olmadığını beyan etmektedirler.

Teşekkür ve Bilgilendirme

Bu çalışma Hasan Alp ŞAHİN'in doktora tezinden üretilmiştir.

Kaynaklar

Abdel-Nour N, Ngadi M, Prasher S, Karimi Y. 2011. Prediction of egg freshness and albumen quality using visible/near infrared spectroscopy. *Food Bioprocess Technol*, 4: 731-736.

Aboonajmi M, Setarehdan SK, Akram A, Nishizu T, Kondo N. 2014. Prediction of poultry egg freshness using ultrasound. *Int J Food Prop*, 17(9): 1889-1899.

Aboonajmi M, Najafabadi TA. 2014. Prediction of poultry egg freshness using vis-nir spectroscopy with maximum likelihood method. *Int J Food Prop*, 17(10): 2166-2176.

Arumugam MS, Chandramohan A. 2007. A New and improved version of particle swarm optimization algorithm with

global-local best parameters. *Knowl Inf Syst*, DOI: 10.1007/s10115-007-0109-z.

Baş N. 2006. Yapay sinir ağları yaklaşımı ve bir uygulama. Mimar Sinan Güzel Sanatlar Üniversitesi. Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, İstanbul.

Bolat B. 2006. Asansör kontrol sistemlerinin genetik algoritma ile simülasyonu. Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, İstanbul.

Chatterjee S, Laudato M. 1995. Gender and performance in athletics. *Soc Biol*, 42: 397-412.

Chen X, Xun Y, Li W, Zhang J. 2010. Combining discriminant analysis and neural networks for corn variety identification. *Comp and Elect in Agri*, 71: 48-53.

Çanakçı M, Hosoz M. 2006. Energy and exergy analyses of a diesel engine fuelled with various biodiesels. *Energy Sour*, Part B: 379-394.

Dede T. 2003. Değer kodlaması kullanarak kafes sistemlerin genetik algoritma ile minimum ağırlıklı boyutlandırılması. Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Trabzon.

Demirbaş HY, Dursun İ. 2007. Buğday tanelerinin bazı fiziksel özelliklerinin görüntü işleme tekniğiyle belirlenmesi. *AÜ Tar Bil Derg*, 13(3): 176-185.

Eldem H. 2014. Karınca Koloni Optimizasyonu (KKO) ve Parçacık Sürü Optimizasyonu (PSO) Algoritmaları temelli bir hiyerarşik yaklaşım geliştirilmesi. Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya.

Elmas Ç. 2003. Fuzzy logic controllers. 1st ed, Seçkin Press, Ankara. pp. 35-40.

Fan SKS, Chiu YY. 2007. A decreasing inertia weight particle swarm optimizer. *Eng Opt*, 39(2): 203 - 228.

Haugh RR. 1937. The haugh unit for measuring egg quality. *US Egg Poultry Mag*, 43: 552-555.

HemAsian-Etefagh F, Safi-Esfahani F. 2019. Dynamic scheduling applying new population grouping of whales meta-heuristic in cloud computing. *J Supercomp*, 75(10): 6386-6450. DOI: 10.1007/s11227-019-02832-7.

Heris JEA, Oskoei M. 2014. Modified genetic algorithm for solving n-queens problem. *Iranian Conference on Intelligent Systems (ICIS)*, 4-6 Feb. 2014, DOI: 10.1109/IranianCIS.2014.6802550.

Karoui R, Kemps B, Bamelis F, De Katelaere B, Decuyper E, De Baerdemaeker J. 2006. Methods to evaluate egg freshness in research and industry: A review. *Europ Food Res Tech*, 222: 727-732.

Karoui R, Nicolai B, De Baerdemaeker J. 2008. Monitoring the egg freshness during storage under modified atmosphere by fluorescence spectroscopy. *Food and Biop Tech*, 1: 346-356.

Kennedy J, Eberhart R. 1995. Particle swarm optimization, IEEE International conference on neural networks. Perth, Australia. IEEE Service Center, 1942-1948, Piscataway, NJ.

Koç ML. 2002. Taş dolgu dalgakıranların yapay sinir ağları, bulanık mantık sistemleri ve genetik algoritma ile ön tasarımı ve güvenilirlik analizi. Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Ankara.

Meng XH, Lin YF, Qui D. 2017. Hybrid algorithm of adaptive inertia weight particle swarm and simulated annealing. *Int J Comp Tech*, 4(2): 105-110.

Özsağlam MY. 2009. Parçacık Sürü Optimizasyonu algoritmasının gezgin satıcı problemine uygulanması ve performansının incelenmesi. Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Konya.

Öztürk HT. 2013. Deprem bölgelerinde yapılacak betonarme sığ tünellerin yapay arı koloni algoritması ve genetik algoritmayla optimum tasarımı. Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Doktora Tezi, Trabzon.

Parlak M. 2007. Genetik algoritmaların hesapsal ve yapısal olarak incelenmesi. Ondokuz Mayıs Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Samsun.

Ratnaweera A, Halgamuge SK, Watson C. 2004. Self-Organizing hierarchical particle swarm optimizer with time-varying acceleration coefficient. *IEEE Trans Evol Comput*, 8(3): 240-255.

Robinson DS, Monsey JB. 1972. Changes in the composition of ovumucin during liquefaction of thick white. *J Sci of Food and Agri*, 23: 29-38.

- Shi Y, Eberhart RC. 1998. A modified particle swarm optimizer. In The IEEE International Conference of Evolutionary Computation, 69–73, Anchorage, Alaska
- Syahputra MF, Felicia V, Rahmat RF, Budiarto R. 2016. Scheduling diet for diabetes mellitus patients using genetic algorithm. J Phy, 801: 012033. DOI: 10.1088/1742-6596/801/1/012033.
- Tamer S, Karakuzu C. 2006. Parçacık sürüsü optimizasyon algoritmasının ve benzetim örnekleri. (Eleco'06) Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu ve Fuarı Bildirileri.
- Wells PC, Norris KH. 1987. Egg quality—current problem and recent advances. In B. M. Freeman (Ed.), Egg quality—current problems and recent advances. Abingdon: Carfax.
- Zhang JR, Zhang J, Lok TM, Lyu MR. 2007. A hybrid particle swarm optimization back-propagation algorithm for feedforward neural network training. App Math and Comp, 185: 1026-1037.