

Sürü zekâsında yeni bir yaklaşım:

Kuş sürüsü algoritması

Elif VAROL¹, Bilal ALATAŞ^{2*}

¹ Kırklareli Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Kırklareli

² Firat Üniversitesi, Mühendislik Fakültesi, Yazılım Mühendisliği Bölümü, Elazığ

Makale Gönderme Tarihi: 02.07.2016

Makale Kabul Tarihi: 05.10.2016

Öz

Matematiksel programlama olarak da bilinen optimizasyon, bir amaç (değerlendirme) fonksiyonuna göre bir problemde belirli aralıktaki sayısal değerlerin en uygununu seçen işlemler topluluğudur. Optimizasyon problemleri için birçok algoritma önerilmiştir. Bu algoritmaların çoğu sistemin modeli ve amaç fonksiyonu için matematiksel modellere ihtiyaç duymaktadır. Sürü zekâsına dayalı algoritmalar, büyük boyutlu optimizasyon problemleri için, kabul edilebilir sürede optimum ya da optimuma yakın çözümler verebilen algoritmalarlardır. Matematiksel modelin çıkarılmadığı durumlarda kabul edilebilir sürede sonuç elde edebilmek amacıyla genel amaçlı sezgisel optimizasyon algoritmaları kullanılır. Genel amaçlı sezgisel optimizasyon algoritmaları, biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı, kimya tabanlı, spor tabanlı ve matematik tabanlı olmak üzere sekiz farklı grupta değerlendirilmektedir. Sürü zekâsı tabanlı optimizasyon algoritmaları kuş, balık, kedi ve arı gibi canlı sürülerinin hareketlerinin incelenmesiyle geliştirilmiştir.

Bu çalışmada, sürü zekâsı optimizasyon algoritmalarının en güncellerinden biri olan kuş sürüsü optimizasyon algoritması ayrıntılı olarak incelenmiştir. Bu algoritmanın performansı, farklı boyutlardaki tek modlu ve çok modlu kalite testi fonksiyonları kullanılarak test edilmiştir. Yapılan deneylerde, optimuma yakınsama eğilimi ve elde edilen sonuç değerleri, performans ölçütü olarak kullanılmıştır. İnceleme sonuçları karşılaştırmalı tablolar aracılığıyla sunulmuş ve yorumlanmıştır. Bu algoritma ile hem tek modlu hem de çok modlu kalite testi fonksiyonlarında diğer sürü zekâsı algoritmalarından çok daha iyi sonuçlar elde edildiği için, algoritmanın ileride birçok problemde etkili olarak kullanılacağı beklenmektedir.

Anahtar Kelimeler: Sürü Zekâsı; Optimizasyon; Kuş Sürüsü Algoritması; Kalite Testi Fonksiyonları

Giriş

En temel tanımı ile optimizasyon eldeki az miktardaki verileri en optimum biçimde kullanmaktır (Bunday, 1984). Kısaca optimizasyon matematiksel olarak bir fonksiyonun minimize veya maksimize edilmesi işlemidir (Kahaner vd., 1989). Optimizasyonun bir başka tanımı ise “en iyi amaç kriterinin en iyi değerini veren kısıtlardaki değişkenlerin değerini bulmaktır” (Himmelblau, 1989). Özetle optimizasyonun tanımını en iyi sonuçları içeren işlemler topluluğu şeklinde yapabiliriz (Rao, 1978). Optimizasyon problemlerinde ilk yapılması gereken, amaç fonksiyonunu seçmektir. Amaç fonksiyonu olası çözümlerin kalitesini ve performansını belirlemektedir. Kâr, zaman gibi herhangi bir özelliği ya da özellikler grubunu yansıtan sayısal bir ifade, amaç fonksiyonun sonucu olabilir. Hedef, değişkenlerin en uygun değerlerini bulmaktır. Amaç fonksiyonları problemin çeşidine göre bir maksimizasyon ya da minimizasyon fonksiyonu olabilmektedir. Probleme göre amaç fonksiyonlarının hesaplanması sırasında bazı kısıtlara dikkat edilmesi gerekmektedir. Uygunluk fonksiyonlarında ise çözüm uzayındaki bütün değişkenler değil, bu kısıtlara uyan değişkenler kullanılmaktadır.

Bir optimizasyon probleminde değişkenlerin, kısıtların ve amaç fonksiyonlarının tanımlanması modelleme olarak adlandırılmaktadır. Bütün optimizasyon problemlerinin çözümü için kullanılacak genel bir optimizasyon algoritması yoktur. Probleme dayalı olarak kullanılacak çeşitli optimizasyon algoritmaları vardır. Probleme göre bu algoritmanın seçimi kullanıcı tarafından yapılmaktadır ve problemi çözme konusunda önemli bir etkiye sahiptir.

Optimizasyon algoritmaları genellikle tekrarlamalıdır. Rastgele ya da belirli bir değer ile başlayan iterasyonlar program boyunca en iyi sonucu bulmayı hedeflemektedir. Bir iterasyondan, diğer bir iterasyona geçiş bütün optimizasyon algoritmalarında farklılık göstermektedir. Algoritmaların bazıları bir önceki

iterasyondaki bulduğu sonucu direk olarak kullanırken, diğerleri her bir iterasyonda bağımsız çözümler elde etmektedir. Etkin bir optimizasyon algoritmasında olması gereken temel özellikler şunlardır (Rao, 1978):

- Geniş bir uygulama alanına sahip olmalıdırlar.
- Çok fazla hesaplama zamanı ve hafızası gerektirmemeliler.
- Verilerdeki hatalardan veya matematiksel yuvarlamalardan çok etkilenmeden detaylı çözümler bulabilmelidirler.

Metasezgisel son zamanlarda etkili olarak birçok kompleks arama ve optimizasyon problemlerinde çözüm yöntemi olarak kullanılmaktadır. Bunların nedeni şu şekilde özetlenebilir (Altunbey ve Alataş, 2015):

- Eş zamanlı olarak, farklı tipte karar değişkenleri, amaç fonksiyonu ve kısıtlayıcı fonksiyonları olması durumunda probleme uygulanabilecek genel çözüm stratejileri sunmaktadırlar.
- Çözüm stratejileri amaç fonksiyonu ve kısıtlayıcı fonksiyonların tipine ve problemi modellemede kullanılan değişkenlerin tipine bağlı değildir.
- Çözüm uzayı tipine, karar değişken sayısına ve kısıtlayıcı fonksiyon sayısına bağlı değildir.
- Sistemin modeli ve amaç fonksiyonu için kurulması zor olan ve bazen de kurulup da çözüm zamanı maliyeti çok yüksek olduğundan kullanılmayan çok iyi tanımlanmış matematiksel modellere ihtiyaç duymamaktadır.
- Hesaplama güçleri iktidar yani aşırı derecede hesaplama zamanına ihtiyaç duymazlar.
- Dönüşümleri ve adaptasyonları kolaydır.
- Klasik algoritmalarındaki gibi verilen bir probleme bir çözüm algoritması uyarlamada geçerliliğinin onaylanması zor olabilen bazı varsayımları gerektirmemektedir.
- Klasik algoritmalarındaki gibi ilgilenilen problem üzerinde değişiklik gerektirmemektedir. Farklı türdeki problemleri çözmek için kendilerini adapte ederler.

Bu avantajlarından dolayı metasezgisel algoritmalar; yönetim bilimi, mühendislik, bilgisayar gibi birçok farklı alanda yoğun olarak kullanılmakta ve yeni versiyonları önerilmektedir. Son zamanlarda, canlıların sürü halindeki davranışlarından esinlenerek geliştirilen sürü zekâsı tabanlı optimizasyon algoritmaları da matematiksel model oluşturulamayan veya oluşturulsa bile çözüm zamanı çok uzun zaman alan birçok kompleks problemin çözümünde popüler hale gelmeye başlamıştır.

Bu çalışmada sürü zekâsı optimizasyon algoritmalarının en güncellerinden birisi olan Kuş Sürüsü Algoritması (KSA) incelenmiştir. Ayrıca KSA'nın performansı, yine popüler sürü zekâsı ve metasezgisel algoritmalarından parçacık sürü optimizasyonu, yapay balık sürü optimizasyonu, kırlangıç sürü optimizasyonu ve karga arama algoritması ile tek ve çok modlu kalite testi fonksiyonlarında ilk kez karşılaştırılmalı olarak sunulmuştur.

Materyal ve Yöntem

Sürü zekâsı merkezi olmayan kontrol, doğal ya da yapay kendi kendini organize eden sistemlerin gösterdiği toplu davranış biçimidir (Bonabeau vd., 1999). Özellikle bireylerin birbirleriyle ve çevreleriyle yerel etkileşimlerinden kaynaklanan kolektif davranışlar üzerine odaklanan disiplindir. Hiçbir kontrol mekanizması olmadan sürüdeki bireyler basit yapı ve özellikte olup basit iletişim yolları ile az maliyet harcararak, esnek ve sağlam bir yapı oluşturup zekice hareket etmeleri ve kendi karşılaştıkları problemlere çözüm getirmeleri biyologların ilgisini çektiği gibi bilgisayar bilimcilerin ve mühendislerin de ilgisini çekmektedir. Çünkü iletişim, taşımacılık, endüstriyel üretim ve robot-bilimi gibi birçok alanda çözümler getirmeye çalışan bilgisayar ve mühendislik bilimleri bu alanların birçoğunda çözümleri oldukça zor veya zaman ya da maliyet olarak masraflı olan problemlerle karşı karşıya kalmaktadır. Sürülerin karşılaştıkları ve çözümledikleri problemlerin birçoğu bu problemlerle paralellik göstermektedir. Bu da sürü zekâsı kavramının geliştirilmesinde önemli

bir rol oynamıştır. Aşağıda şimdiye kadar farklı araştırmacıların önerdiği ve bu makalede performans karşılaştırması için seçilen sürü zekâsı optimizasyon algoritmaları alt başlıklar halinde açıklanmıştır.

Parçacık Sürü Optimizasyon Algoritması

Parçacık Sürü Optimizasyonu (PSO), ilk olarak Kennedy ve Eberhart tarafından 1995 yılında geliştirilmiş bir optimizasyon yöntemidir (Kennedy ve Eberhart, 1995). PSO balık, böcek, kuş vb. hayvan sürülerinin davranışından esinlenerek geliştirilmiştir. Yöntem, genel olarak söz konusu sürünün, besin kaynağı ararken ortaya koyduğu davranış üzerine kuruludur. Tek başlarına bir şey yapamayan bu tür hayvanlar sürü halinde zeki davranışlar sergileyebilmektedir. PSO;

- Her bir bireyin kendi hafızalarında yer etmiş olan iyi konuma gitme eğilimi olarak tanımlanabilecek bilişsel davranış biçimi,
- Her bir bireyin iyi konumlarda bulunan diğer bireyleri takip etme eğilimi olarak tanımlanabilecek sosyal davranış biçimi,
- Her bir bireyin rastgele arama yapma eğilimini tanımlayabilecek keşifsel davranış biçimleri arasında bir denge olduğu varsayımına dayanmaktadır (Eberhart ve Kennedy 1995; Karaboğa, 2011). Sürünün bu davranış formüle edilerek optimizasyon problemlerinin çözümünde kullanılmaktadır. Şekil 1'de PSO'nun akış diyagramı görülmektedir.

Algoritma temel olarak aşağıdaki basamaklardan oluşur.

Adım 1: Rastgele üretilen başlangıç pozisyonları ve hızları ile başlangıç sürüsü oluşturulur.

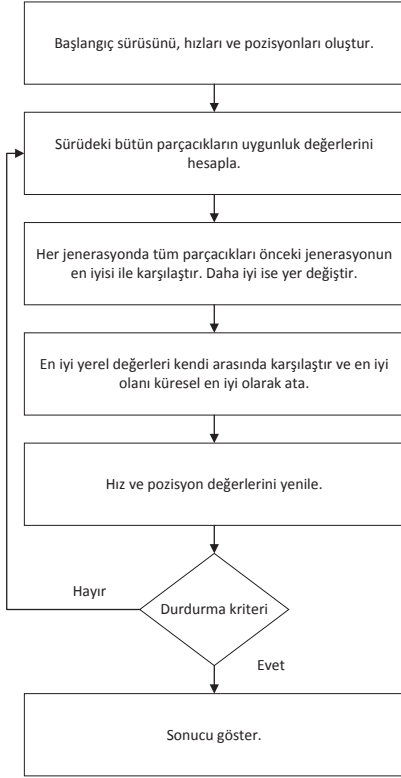
Adım 2: Sürüdeki parçacıkların uygunluk değerleri hesaplanır.

Adım 3: Her bir parçacık için mevcut nesilden yerel en iyi bulunur. Sürü içerisinde en iyilerin sayısı parçacık sayısı kadardır.

Adım 4: Mevcut nesildeki yerel eniyiler içerisinde global en iyi (gbest) seçilir.

Adım 5: Pozisyon ve hızlar güncellenir.

Adım 6: Durdurma kriteri sağlanıncaya kadar Adım 2, 3, 4 ve 5 tekrar edilir.



Şekil 1. PSO'nun akış diyagramı

PSO algoritması, her bir parçacığın problem için muhtemel bir çözümü temsil ettiği parçacıkların bir sürüsü (popülasyonu) ile uğraşır. Evrimsel hesaplama paradigmalarına benzer olarak, sürü popülasyona karşılık gelmekte ve parçacıklar bireylere benzemektedir. Basit bir ifadeyle, parçacıklar çok boyutlu araştırma uzayı boyunca akış içerisindeyler ve her bir parçacığın pozisyonu kendi ve komşularının tecrübelerine göre ayarlanmaktadır.

PSO genel olarak iki ana amaca yönelik olarak özelleştirilmiştir. Birincisi global en iyiyi bulma (Global En iyi PSO ya da kısaca Gbest-PSO), ikincisi lokal en iyiyi bulma (Lokal En iyi PSO ya da Lbest-PSO). Gbest-PSO için her bir parçacığın komşuluğu sürünün bütününe kapsamaktadır. Gbest-PSO'nun sahip olduğu

sosyal ağ, yıldız topolojisini temsil etmektedir. Yıldız komşuluk topolojisinde parçacık hız değişiminin sosyal bileşeni, sürüdeki tüm parçacıklardan elde edilen bilgiyi temel almaktadır.

Yapay Balık Sürüsü Optimizasyon Algoritması

Yeni bir evrimsel hesaplama tekniği olan Yapay Balık Sürüsü Algoritması (YBSA) ilk kez 2002'de önerilmiştir (Liv vd., 2002). YBSA'nın fikri sürü teorisine ve basitleştirilmiş balık sürüsünün doğal sosyal davranışlarının simülasyonuna dayanmaktadır.

YBSA genetik algoritmada kullanılan mutasyon süreçleri ve çaprazlamaya sahip değildir. Bu yüzden daha kolay bir şekilde yürütülmektedir. Sistem başlangıçta rastgele üretilen bir dizi potansiyel çözümlerle başlatılmakta ve daha sonra tekrarlı bir şekilde optimum bir çözüm elde etmek için arama yapmaktadır (Zhang vd., 2000).

Gerçek balığın hayali bir varlığı olan Yapay Balık (YB), problemin açıklanmasında, analizinin yapılmasında ve hayvanlar için ekoloji kavramının fark edilebilmesinde rol oynamaktadır. Kuyruk ve yüzgeçlerin kontrolü ile uyarıcı bir tepki verilmektedir. Bundan sonraki davranış onun çevresinin durumuna ve bugünkü durumuna bağlıdır (şu anda soru çözümlerinin kalitesi ve diğer eşlik eden arkadaşların durumları da dâhil). Bu çevre, onun kendi aktivitelerini ve diğer eşlik eden arkadaşlarının aktivitelerini de etkilemektedir (Zhang vd., 2007; Mehdi vd., 2012).

YB'nin davranışlarını içeren temel fonksiyonlar; YB_Av, YB_Sürü, YB_Takip, YB_Hareket, YB_Atama ve YB_Değerlendirme'dir.

Balık genellikle birçok yiyecek ile bir yerde kalır. Bu yüzden YBSA'nın temel fikri olan global optimumu bulmak için bu karakteristiklere dayalı balıkların davranışlarını taklit etmektedirler. YB'nin maksimum amaç değeri elde etmek için temel davranışları aşağıdaki gibidir (Bao vd., 2016; Quan vd., 2016):

YB_Av: Bu gıda eğilimi temel biyolojik bir davranıştır. Genellikle balık görme ve hissetme duyusu ile hareket belirlemek için, su gıda yoğunlaşmasını algılar ve sonra eğilimi seçer.

YB_Sürü: Sürü davranışı bir balığın sürünün merkezine doğru giden ve aşırı kalabalıktan kaçan bir davranıştır. Güncel durumdaki bir YB'ye eşlik eden arkadaşının numarasını ve onun bugünkü *komşuluk()*'ta merkezi konumunu arar. Eğer balık kolonisinin merkezindeyse yeterli yiyecek vardır ve çok kalabalık değildir anlamına gelmektedir. Balık koloninin varlığını garanti etmek ve tehlikelerden kaçmak için bir tür yaşam alışkanlığı olan hareketli süreçte doğal bir biçimde gruplar halinde toplanacaktır.

YB_Hareket: Balık suda rastgele yüzer. Aslında daha büyük aralıklarda yiyecek ya da eşlik edecek arkadaş arar.

YB_Atlama: Objektif değerlerdeki farklılık bazı iterasyonlarda daha küçük olmakta ve böylece yerel ekstremum noktasına düşmektedir. Parametrelerdeki rastgele değişiklik dışarı sızan mevcut değişiklik durumunu bildirmektedir.

Tablo 1. YBSA'nın sözde kodları (Mehdi vd., 2012)

<pre>Prosedür Yapay Balık Sürüsü Algoritması YB başlatma() While sonuçtan memnun Switch (YB_yiyecekkararlılık()) Case değer1 YB_Takip(); Case değer2 YB_Sürü(); Default YB_Arama(); End switch YB_Hareket(); Get_sonuç(); End while End Yapay Balık Sürüsü Algoritması</pre>
--

YBSA sürecinde arama davranışı YB'nin temelini oluşturmaktadır. Davranış seçimi algoritmanın kararlılığını ve yüksek verimliliğini garanti eder. Davranış seçim yoluyla YBSA yüksek verimlilik ile optimizasyon stratejisi oluşturabilmektedir.

YBSA'nın sözde kodları Tablo 1'de gösterilmiştir.

Kırlangıç Sürüsü Optimizasyon Algoritması

Kırlangıç Sürüsü Optimizasyonu (KSO) sürü zekâsına dayalı yeni bir algoritmadır. Bu optimizasyon yöntemi kırlangıç sürüsünün hareketlerini ve davranışlarını inceleyen yeni bir optimizasyon yöntemidir. Bu yöntemde parçacıkların 3 tipi vardır. Bunlar; kâşif parçacıklar, amaçsız parçacıklar ve lider parçacıklardır. Her parçacığın kişisel bir özelliği vardır ama bunların ortak özelliği, uçan bir merkezi kolonisinin olmasıdır. Her parçacık sürekli olarak adaptif bir yarıçap ile çevreyi araştırarak akıllı bir davranış sergilemektedir. Komşu parçacıkların durumları genel lider ve lokal lider olarak kabul edilmekte ve daha sonra bir hamle yapılmaktadır (Mehdi vd., 2012).

KSO algoritmasıyla yüksek verimlilik kanıtlanmıştır. Bu yüksek verimlilik; düz alanlarda hızlı hareketi (bu alanlarda gıda bulunmadığı için türev sıfıra eşittir), elde kalmış lokal ekstremum noktaları, yüksek yakınsama hızını ve parçacıkların farklı gruplar içindeki akıllı katılımını sağlamaktadır (Mehdi vd., 2012).

Bu parçacıklar birbirlerine paralel olarak hareket eder ve her zaman etkileşim halindedirler. Kolonideki her parçacık (her koloni alt kolonilerden oluşmaktadır) bir şeyler yaparak ve deneyerek daha iyi bir duruma doğru kolonileri yönlendirmeden sorumludur.

Kırlangıç hareketlerinin yüksek hız ve yüksek dinamikte gerçekleşmesi nedeniyle alt koloniler arasındaki gerçek sınırlar asla işaretlenemez. Kırlangıçların diyagramı ve sayıları alt koloni büyüklüklerine göre değişir.

Bu parçacıkların tümü sürekli birbirleriyle etkileşim halindedir ve her parçacık bu üç rolden birini üstlenir. Arama süresi sırasında bu parçacıklar sık sık rollerini değiştirebilir ama öncelikli hedef optimum noktayı bulmaktır (Mehdi vd., 2012).

Karga Arama Algoritması

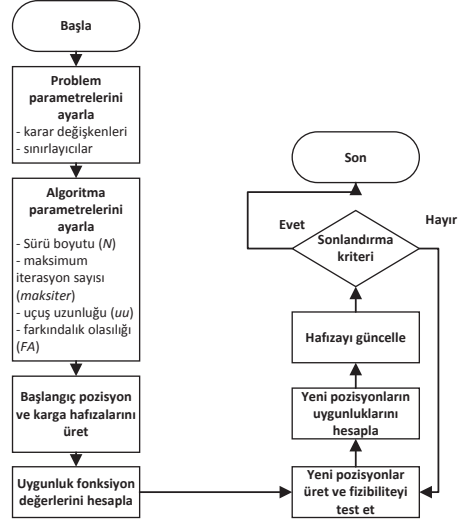
Kargalar en zeki hayvanların başında gelmektedir. Yüzleri hatırlayabilmekte, araç kullanabilmekte, diğer kargalarla iletişime geçebilmekte, mevsimlerde yiyecek bulup saklayabilmektedir. Daha iyi yiyecek kaynağını bulabilmek için birbirlerini takip ederler. Karga Arama Algoritması (KAA) da kargaların bir sürü şeklindeki yaşamları, saklandıkları yerleri hatırlayabilme kabiliyetleri, yiyecek bulmadaki birbirlerini takip etmeleri ve gizlendikleri yerleri koruma davranışları göz önüne alınarak Askarzadeh tarafından 2016'da önerilmiştir (Askarzadeh, 2016). Bu açıklamalara bağlı olarak kargaların davranışı ve popülasyon tabanlı metasezgisel algoritma arasındaki benzerlik Tablo 2'de gösterilmiştir. Algoritmanın akış diyagramı ise Şekil 2'de gösterilmiştir (Askarzadeh, 2016; Akyol ve Alatas, 2016).

Tablo 2. Kargalar ve metasezgisel optimizasyon arasındaki benzerlik

Kargalar	Metasezgisel Optimizasyon
Kargalar	Arama ajanları
Çevre	Arama Uzayı
Çevrenin her pozisyonu	Uygun çözümler
Yiyecek kaynağının kalitesi	Amaç fonksiyonu
Çevrenin en iyi yiyecek kaynağı	Global çözüm

Kuş Sürüsü Algoritması

Kuş Sürüsü Algoritması (KSA) kuş sürülerindeki sosyal davranış ve sosyal etkileşim temelli sürü zekâsı optimizasyon algoritmalarının en yenilerindedir. Kuşların çoğunlukla üç davranış biçimi vardır. Bunlar beslenme davranış biçimi, uyanıklık davranışı ve uçma davranışdır. Kuşlar yiyecek arayabilir ve hayatta kalma şansını artırabilmek için sosyal etkileşimler aracılığıyla avcılardan kaçabilmektedirler. Böyle sosyal davranışları modelleyerek, sosyal etkileşimler ve ilişkili sürü zekâsı, beş temel kural ile dört arama stratejisi KSA'da formüle edilmiştir (Meng vd., 2016).



Şekil 2. KAA'nın akış diyagramı

İspinoz gibi birçok kuş türü sürü halinde yaşamaktadır. Toplu bir şekilde sürüde uçabilmekte, yiyecek arayışında bulunabilmekte ve konaklayabilmektedirler (Anderson, 2006). Böyle davranışlar ayrılık, hizalama, uyum gibi basit kurallardan kaynaklanan acil davranışlar olarak kabul edilmektedir. En basit sosyal etkileşim yoluyla sürü davranışlarında kompleks hareketler ve etkileşimler geliştirilebilmektedir. Sürüde yiyecek arayan kuşlar kendi hisleriyle yola çıkarak daha fazla bilgi toplayabilmekte ve hayatta kalma şansına sahip olup, iyi ve verimli besin arayışında bulunabilmektedirler. Eğer bir kuş farklı yiyecek yolları bulursa diğerleri onlardan beslenir (Kennedy vd., 2001). Besin ararken kuşlar genellikle avlanma tehdidine karşı bir araya toplanmaktadırlar (Krause ve Ruxton, 2002). Sık sık kafalarını kaldırarak çevrelerini taramaktadırlar. Böyle davranışlar uyanıklık davranışı olarak (Anderson, 2006) yorumlanmakta ve tespit eden avcılar için elverişli olabilmektedir (Lima ve Dill, 1990). Çalışmalar kuşların rastgele bir şekilde yiyecek arama ve uyanık kalma arasındaki seçimi göstermektedir (Bednekoff ve Lima, 1998). Yırtıcı bir hayvan tespit ettiklerinde kuşlar

genellikle alarm çağrıları vermektedirler (Pulliam, 1973). Bu yüzden bütün grup aceleyle hareket etmektedir. Sürüdeki kuşların yalnız olan kuştan potansiyel bir tehdidi daha iyi hissetme sansına sahip olduğu mantıklı bir şekilde sonuçlanmıştır. Grup boyutundaki bir artış ile bireysel uyanıklıkta azalma birçok kuştan oldukça yaygındır (Ekman, 1987). Başka bir deyişle, bir kuş yırtıcı bir hayvan tarafından artan saldırıya uğrama riskinden etkilenmeksizin grup büyüklüğü arttıkça besin aramada daha fazla zaman harcayabilmektedirler (Beauchamp, 1998; Roberts, 2003).

Bir grubun çevresindeki kuşların avcılar tarafından saldırıya uğrama şansı, merkezdekilerden daha fazladır. Çalışmalar sürünün merkezindeki besin arayışında bulunan hayvanların avcılar tarafından saldırıya uğradıklarında kendilerini korumak için komsularına hareket edebilir fikrini önermiştir (Pulliam, 1973). Her kuş, sürünün merkezine doğru hareket etmeye çalışabilmektedir. Çünkü onu algılayabilmektedirler. Fakat bu hareketin, kuş sürüleri arasında oluşan rekabetten etkilenme olasılığı vardır (Beauchamp, 2003). Bu yüzden kuşlar sürünün merkezine doğru, doğrudan hareket etmeyebilirler.

Kuşlar açıkçası besin aramak için ya da avcılardan kaçmak için diğer yönlere uçabilmektedir. Yeni bir yere vardıldıktan sonra yeniden yiyecek aramak istemektedirler. Burada sürüden beslenen kuşlarda üretme ve bedavacılığın var olduğu gözlenmektedir. Üreticiler aktif bir şekilde yiyecek arayışında bulunurken bedavacılar sadece üreticiler tarafından bulunan yiyeceklerden beslenirler (Barnard ve Sibly, 1981; Giraldeau ve Caraco, 2000). Bireyler genellikle alternatif davranış stratejileri kullanabilmekte ve üretim ve bedavacılık yapma arasında seçim yapabilmektedirler (Liker ve Barta, 2002; Johnson vd., 2001; Barta ve Giraldeau, 2000). Çalışmalar yüksek rezervli biri üretici olurken küçük rezervli kuşların sık sık bedavacı olabileceğini, düşünmektedir (Barta ve Giraldeau, 2000).

Sürüden beslenen kuşlar için sosyal davranışlar sosyal bir hayvanın hayatta kalabilmesi için bir avantaj sağlamaktadır (Siroto, 2006). Her kuş optimum hayatta kalma ve iyi yem arama verimliliğini elde etmek için sosyal etkileşimlerden faydalanabilmektedir.

Yukarıda belirtilen sosyal davranışlar aşağıdaki gibi bazı idealize edilmiş kurallarla basitleştirilmiştir (Meng vd., 2016).

Kural 1: Her kuş uyanıklık davranışı ve besin arama davranışı arasında geçiş yapabilir. Kuş ister yemlesin ister uyanık kalsın, bu bir stokastik karar olarak modellenmiştir.

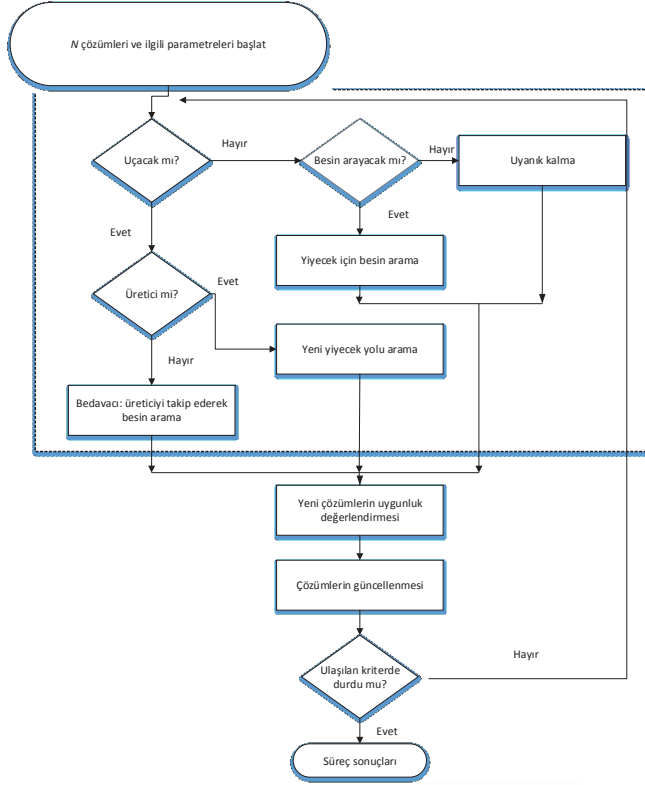
Kural 2: Besin ararken her kuş deneyimlerini hemen kaydetmeli ve onun önceki en iyi deneyimini ve sürülerin önceki gıda yönü hakkında en iyi deneyimini güncellemelidir. Bu deneyim aynı zamanda yiyecek arayışı için de kullanabilmektedir. Sosyal bilgiler bütün sürüler arasında anında paylaşılmaktadır.

Kural 3: Uyanık tutulduğunda her kuş sürünün merkezine doğru hareket etmeye çalışacaktır. Bu davranış sürü arasındaki rekabet ile uyarılan müdahaleden etkilenebilmektedir. Daha yüksek rezervli kuşların sürünün merkezine daha düşük rezervli olanlara göre daha yakın olması daha muhtemeldir.

Kural 4: Kuşlar periyodik bir şekilde başka bir yöne uçmaktadır. Başka yöne uçtuklarında kuşlar sık sık üretim ve bedavacılık arasında geçiş yapabilmektedirler. En düşük rezervli kuş bedavacı olabilirken en yüksek rezervli kuş bir üretici olabilmektedir. En yüksek ve en düşük rezervli kuşlar arasındaki diğer rezervli kuşlar rastgele üretici ve bedavacı olmayı seçmektedirler.

Kural 5: Üreticiler aktif bir şekilde yiyecek ararken, bedavacılar rastgele bir üreticiyi takip ederek yiyecek aramaktadırlar.

Verilen bu kurallar sözlü olarak formüle edildiğinde kesin matematiksel model ile yeni metasezgisel algoritma geliştirilmiştir. KSA'nın temel akış diyagramı Şekil 3'te gösterilmiştir.



Şekil 3. KSA'nın akış diyagramı (Meng vd., 2016)

Tüm N sanal kuşlar, onların pozisyonları tarafından tasvir edilen t zaman adımında $x_i^t (i \in [1, \dots, N])$ d boyutlu uzayda uçmakta ve besin için yiyecek aramaktadır.

a) Besin Arama Davranışı

Her kuş kendi tecrübe ve sürülerin deneyimlerine göre yemek aramaktadır. Kural 2 denklem (1)'deki gibi ifade edilmektedir.

$$x_{ij}^{t+1} = x_{ij}^t + (p_{ij} - x_{ij}^t) \times c \times rand(0, 1) + (g_j - x_{ij}^t) \times s \times rand(0, 1) \quad (1)$$

Burada $j \in [1, \dots, D]$ $rand(0, 1)$ 0 ile 1 arasındaki bağımsız eşit dağıtılmış sayıları temsil etmektedir. d boyut, i kuş sayısı, t iterasyon, c ve s sırasıyla bilişsel ve sosyal hızlandırılmış

katsayılar olarak adlandırılan iki pozitif sayıdır. x_{ij}^t , i . kuşun j . boyutunun t . anındaki pozisyonunu ifade etmektedir. $p_{i,j}$ i . kuşun en eski konumu ve g_j sürü tarafından paylaşılan en eski pozisyonudur.

Kolaylık olması açısından Kural 1 stokastik bir karar olarak formüle edilmiştir. Eğer 0 ile 1 arasında düzgün bir rastgele sayı $PC[0, 1]$ sabit değerinden daha küçükse, kuş yiyecek için besin arayışında olur. Aksi takdirde kuşun uyanıklığı devam eder.

b) Uyanıklık Davranışı

Kural 3 göz önüne alındığında, kuşlar sürünün merkezine hareket etmeye çalışmaktadır ve kaçınılmaz bir şekilde birbirleriyle rekabet halinde olmaktadır. Bu yüzden her kuşun

doğrudan sürünün merkezine doğru hareket etme eğilimi yoktur. Böyle hareketler denklem (2-4)'teki gibi ifade edilmektedir (Meng vd., 2016):

$$x_{ij}^{t+1} = x_{ij}^t + A_1(\text{mean}_j - x_{ij}^t) \times \text{rand}(0, 1) + A_2(p_{kj} - x_{ij}^t) \times \text{rand}(-1, 1) \quad (2)$$

$$A_1 = a_1 \times \exp\left(-\frac{pFit_i}{\text{sumFit} + \epsilon} \times N\right) \quad (3)$$

$$A_2 = a_2 \times \exp\left(-\frac{pFit_i - pFit_k}{|pFit_k - pFit_i| + \epsilon}\right) \frac{N \times pFit_k}{\text{sumFit} + \epsilon} \quad (4)$$

$k(k \neq i)$ 1 ve N arasında rastgele seçilen pozitif bir tamsayıdır. a_1 ve a_2 [0, 2] arasında sabit iki pozitif sayıdır. $pFit_i$ i . kuşun en iyi uygunluk değerini göstermektedir ve sumFit sürünün en iyi uygunluk değerinin toplamını temsil etmektedir. 0 bölünme hatasını önlemek için kullanılan ϵ , bilgisayardaki en küçük sabittir. mean_j bütün sürünün j . elemanının ortalama konumunu ifade etmektedir.

Bir kuş sürünün merkezine doğru hareket ettiğinde kaçınılmaz bir şekilde birbirleriyle yarışacaklardır. Kolaylık olması açısından sürünün ortalama uygunluk değeri bir kuş sürünün merkezine doğru hareket ettiğinde çevresi tarafından uyarılan dolaylı etki olarak kabul edilmektedir. Her kuşun sürünün merkezinde durmak istediği gerçeği göz önüne alındığında A_1 ve $\text{rand}(0, 1)$ 'in çarpımı 1'den fazla olmamalıdır. Burada bir kuş sürünün merkezine hareket ettiğinde A_2 özel bir girişim tarafından uyarılan doğrudan etkiyi simüle etmek için kullanılır. Rastgele k . bir kuşun en iyi uygunluk değeri ($k \neq i$) $A_2 > a_2$ olur. Yani i . kuşun k . kuştan daha büyük bir girişim ortaya çıkarabildiği anlamına gelir. Bazı rastgelelik ve öngörülmezlik olmasına rağmen k . kuşun i . kuşa göre sürünün merkezine doğru daha fazla hareket etmesi muhtemeldir. Aksi belirtilmedikçe bu çalışma minimal optimizasyon problemlerini çözer (Meng vd., 2016).

c) Uçma Davranışı

Kuşlar bir başka bölgeye avlanma tehdidine karşılık olarak, besin arama ya da diğer

herhangi bir sebepten uçabilirler. Yeni bir bölgeye vardıklarında yine yiyecek için besin aramaktadırlar. Üretici olarak davranan bazı kuşlar yiyecek yollarını arayabilmekte diğerleri üreticiler tarafından bulunan yiyecek yolundan beslenmeye çalışmaktadırlar. Kural 4'e göre üreticiler ve bedavacılar sürüden ayrılabilirler. Üreticilerin ve bedavacıların davranışları sırasıyla denklem (5, 6)'daki gibi matematiksel olarak ifade edilmiştir (Meng vd., 2016):

Tablo 3. KSA'nın sözde kodları

KSA
<p>Girdi: N: Popülasyon tarafından içerilen bireylerin (kuş) sayısı M: İterasyonun maksimum sayısı FQ: Kuşların uçma davranışlarının sıklığı P: Yiyecek için besin bulma olasılığı C, s, a_1, a_2, FL: 5 sabit parametre $t = 0$: popülasyonu başlatmak ve ilgili parametreleri tanımlamak N bireylerinin uygunluk değeri değerlendirme ve en iyi çözüm bulma While($t < M$) If($t \% FQ \neq 0$) For $i = 1 : N$ If $\text{rand}(0, 1) < P$ Kuşlar yiyecek için besin arar (Denklem 1) Else Kuşlar uyanık kalır (Denklem 2) End If End For Else Sürüyü iki parça haline bölün: Üreticiler ve bedavacılar For $i = 1 : N$ If i yapıcı ise Üretici (Denklem 5) Else Bedavacı (Denklem 6) End If End For End If yeni çözümleri değerlendirme If yeni çözümler önceki çözümlerden daha iyiyse bunları güncelleştir Mevcut en iyi çözümü bul $t = t + 1$; End While Çıkış : Popülasyondaki en iyi amaç fonksiyonlu birey</p>

$$x_{ij}^{t+1} = x_{ij}^t + \text{rand}(0, 1) \times x_{ij}^t \quad (5)$$

$$x_{ij}^{t+1} = x_{ij}^t + (x_{kj}^t - x_{ij}^t) \times FL \times \text{rand}(0, 1) \quad (6)$$

$\text{rand}(0, 1)$ standart sapması 1 ve ortalaması 0 olan rastgele sayı üreten Gauss'u göstermekte, $k \in [1, 2, 3, \dots, N]$, $k \neq i$. $FL(FL \in [0, 2])$ bedavacıların yiyecek aramak için üreticileri takip ettiği anlamına gelmektedir.

d) KSA süreci

Söz konusu açıklamalara dayanarak, KSA Tablo 3'te gösterilen sözde kod ile özetlenmiştir.

Kalite Testi Fonksiyonları

Kalite testi fonksiyonları optimizasyon algoritmalarının performanslarının karşılaştırılmasında sıklıkla kullanılmaktadır. Literatürde birçok kalite testi fonksiyonu tanımlanmıştır. Bu fonksiyonlar yerel optimum sayılarına ve sıklıklarına göre sınıflandırılmaktadırlar. KSA'nın performansını test etmek için bazı kriterler göz önünde bulundurularak PSO, YBSO, KSO ve KAA ile karşılaştırmalar yapılmıştır. Bu fonksiyonların her biri özel koşullarda optimizasyon algoritmalarını test etmektedir. Bu şekilde optimizasyon algoritmasının zayıf noktaları ortaya çıkmaktadır. Bu çalışmada da, yoğun olarak test amaçlı kullanılan dört temel fonksiyon seçilmiştir. Bu fonksiyonlar Tablo 4'te gösterilmiştir.

Tablo 4. Kalite testi fonksiyonları

Fonksiyon	Denklem	Tanımlı olduğu aralık	Boyut
f_1	Sphere $\sum_{i=1}^D x_i^2$	[-5.12, 5.12]	30
f_2	Rosenbrock $\sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-50, 50]	30
f_3	Griewank $\frac{1}{4000} \sum_{i=1}^D (x_i - 100)^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}} - 100\right) + 1$	[-600, 600]	30
f_4	Rastrigin $\sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12, 5.12]	30

Uygulama ve Başarımlar

KSA'da parametrelerin değerleri $FQ=10$, $c_1=1.5$, $c_2=1.5$, $a_1=1$, $a_2=1$, boyut=30 olarak seçilmiştir. KSA ile birlikte standart YBSA, standart PSO ve standart KSO algoritmaları da MATLAB'da simüle edilmiştir ve karşılaştırma yapmak için 4 tane fonksiyon seçilmiştir. Bu simülasyonun amacı 4 metodun yakınsama performanslarını karşılaştırmaktır. Bu amaçla, aynı sayıda fonksiyon değerlendirme yapacak şekilde, tüm algoritmalar için bitim şartı 1000 iterasyon olarak belirlenmiştir. Bu bitim şartı sonrası algoritmaların performansı, karşılaştırmalı olarak ilerleyen paragraflarda sunulmuştur.

[-5.12, 5.12] arasında tanımlı ve 30 boyutlu Sphere fonksiyonunda farklı parçacık sayısı ve 1000 iterasyonla dört optimizasyon algoritmasının performans sonuçlarının

değerlendirilmesi Tablo 5'te gösterilmiştir. Deneylerde 1000 iterasyonlu farklı parçacık sayıları ile PSO, YBSA, standart KSO, KAA ve KSA tek modlu Sphere fonksiyonu ile karşılaştırılmış olup KSA ile diğer dört algoritmaya göre optimum noktayı yakalama performansı açısından çok daha iyi sonuçlar elde edilmiştir.

Tablo 5. Sphere fonksiyonunda PSO, YBSA, KSO, KAA ve KSA performanslarının karşılaştırılması

POPULASYON SAYISI	PSO _{gbest}	YBSA _{gbest}	KSO _{gbest}	KAA	KSA _{GREST}
100	1.9240E-082	0	0	3.0184E-06	0
50	7.5243E-078	0	0	6.2655E-05	0
30	3.4892E-064	2.2378E-091	0	2.9015E-04	0
10	4.9240E-041	6.4182E-087	8.0016E-095	0.0022	0
2	4.5519E-028	7.0584E-041	7.4218E-012	0.2212	2.2912E-124

[-50, 50] arasında tanımlı 30 boyutlu Rosenbrock fonksiyonunda farklı parçacık sayısı ve 1000 iterasyon kullanan dört optimizasyon algoritması arasındaki performans sonuçlarının değerlendirilmesi Tablo 6'da gösterilmiştir. Yapılan deneylerin sonuçlarına göre; KSA algoritması ile PSO, YBSA ve KSO algoritmalarından daha kötü, ancak KAA'dan daha iyi sonuçların elde edildiği görülmektedir.

[-600, 600] arasında tanımlı ve 30 boyutlu Griewank fonksiyonunda farklı parçacık sayısı ve 1000 iterasyonlu dört optimizasyon algoritmasının sonuçları Tablo 7'de gösterilmiştir. Bu deney sonuçlarına göre KSA algoritması çok modlu Griewank fonksiyonunda; standart PSO, standart YBSA, standart KSO ve KAA'ya göre performans açısından çok iyi sonuçlar elde etmiştir.

Tablo 6. Rosenbrock fonksiyonunda PSO, YBSA, KSO, KAA ve KSA performanslarının karşılaştırılması

POPULASYON SAYISI	PSO _{gbest}	YBSA _{gbest}	KSO	KAA	KSA
100	5.0158	8.4573E-001	4.1217E-004	2.5790E+08	28.9610
50	11.2104	1.3811	9.2483E-003	1.8596E+08	28.9617
30	18.0025	3.4502	2.1172E-003	1.3635E+08	28.9769
10	26.5891	4.8604	1.0043E-001	6.9406E+06	28.9306
2	38.6271	13.8004	4.8151	6.2448E+04	28.9994

Tablo 7. Griewank fonksiyonunda PSO, YBSA, KSO, KAA ve KSA performanslarının karşılaştırılması

POPULASYON SAYISI	PSO _{gbest}	YBSA _{gbest}	KSO	KAA	KSA
100	2.4192E-003	2.0008E-009	7.5301E-012	0.0405	0
50	7.7481E-002	1.7812E-007	1.4704E-011	0.0642	0
30	3.4521E-002	3.2304E-006	3.3501E-009	0.1913	0
10	1.0081E-002	1.4102E-005	4.8516E-008	1.0382	0
2	2.1105E-001	4.0529E-004	1.0063E-002	1.8185	0

[-5.12, 5.12] aralığında tanımlı 30 boyutlu Rastrigin fonksiyonunda farklı parçacık sayısı ve 1000 iterasyonlu dört optimizasyon algoritması arasındaki performans sonuçlarının değerlendirilmesi Tablo 8’de gösterilmiştir. Bu tabloya göre KSA; PSO, YBSA, KSO ve KAA’ya göre çok modlu Rastrigin fonksiyonunda çok daha iyi sonuçlar elde etmiştir.

Tablo 8. Rastrigin fonksiyonunda PSO, YBSA, KSO, KAA ve KSA performans karşılaştırılması

POPULASYON SAYISI	PSO _{gbest}	YBSA _{gbest}	KSO	KAA	KSA
100	1.08254E-004	7.3841E-009	3.3557E-015	25.8705	0
50	2.3961E-002	1.0014E-006	7.2471E-013	16.9198	0
30	8.73	4.5106E-005	8.4519E-012	24.9272	0
10	10.26	9.0045E-005	1.0041E-009	56.9561	0
2	14.73	4.78	2.12	1.22	0

Sonuçlar ve Tartışma

Bu çalışmada sürü zekâsı optimizasyon algoritmalarının en güncellerinden biri olan kuş sürüsü optimizasyon algoritması ayrıntılı olarak incelenmiştir. Bu algoritmanın performansı, farklı boyutlardaki tek modlu çok modlu kalite testi fonksiyonları kullanılarak, yine güncel ve popüler sürü zekâsı algoritmalarından parçacık sürü optimizasyonu, yapay balık sürü optimizasyonu, kırlangıç sürü optimizasyonu ve karga arama algoritması ile karşılaştırılmıştır. Yapılan deneylerde, optimuma yakınsama eğilimi ve elde edilen sonuç değerleri, performans ölçütü olarak kullanılmıştır. İnceleme sonuçları karşılaştırmalı tablolar aracılığıyla sunulmuş ve yorumlanmıştır.

İnceleme için kalite testi fonksiyonları arasından en güncel olan dört fonksiyon seçilmiştir. Tek

modlu fonksiyon olan Sphere fonksiyonunda KSA karşılaştırma için seçilen diğer dört sürü zekâsı algoritmalarından çok daha iyi sonuç elde etmiştir. Çok modlu fonksiyonlardan Griewank ve Rastrigin fonksiyonlarında KSA diğer dört algoritmadan çok daha iyi sonuç elde etmiştir. Sadece Rosenbrock fonksiyonunda istenilen başarı elde edilememiştir.

Genel olarak en güncel sürü zekâsı algoritmalarından biri olan KSA ile parametre ayarlaması yapılmadan hem tek modlu hem de çok modlu kalite testi fonksiyonlarında diğer sürü zekâsı algoritmalarından daha iyi sonuçlar elde edildiği için, algoritmanın ileride birçok kompleks problemde etkili olarak kullanılacağı beklenmektedir. KSA’nın performansının artırılması amacıyla; optimize edilmiş parametreler, dağıtık ve paralel versiyonlar, çok amaçlı çalışmalar, melez sistemler de geliştirilebilir.

Kaynaklar

- Bunday B., (1984). Basic Optimization Methods, London: Edward Arnold Ltd.
- Kahaner D., Moler C., Nash S., (1989). Numerical Methods and Software. Englewood Cliffs: Prentice Hall.
- Himmelblau D.E., (1989). Optimization of Chemical Processes, NY: McGrawHill, Inc.
- Rao S., (1978). Optimization Theory and Applications, 2nd. Edition Halsted Inc.
- Altunbey F., Alataş B., (2015). Sosyal Ağ Analizi için Sosyal Tabanlı Yapay Zekâ Optimizasyon, IJPAS, 1(1), 33-52.
- Bonabeau E., Dorigo M. ve Theraulaz G., (1999). Swarm Intelligence: From Natural to Artificial System, New York: Oxford University Press.
- Kennedy J., Eberhart R., (1995). Particle Swarm Optimization, IEEE International Conference on Neural Networks, 1942-1948.
- Eberhart R.C., Kennedy J., (1995). A New Optimizer Using Particle Swarm Theory, Proceedings of the sixth international symposium on micro machine and human science. Vol. 1, 39-43.
- Karaboğa D., (2011). Yapay Zekâ Optimizasyon Algoritmaları, Nobel Yayın Dağıtım.
- Li X.L., Shao Z.J., Qian J.X., (2002). An Optimizing Method Based on Autonomous Animals: Fish-

- Swarm Algorithm, *Syst Eng Theory Pract* 22(11):32–38.
- Zhang M., Shao C., Li F., Gan Y., Sun J., (2000) Evolving Neural Network Classifiers And Feature Subset Using Artificial fish Swarm, In: Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, June 25–28.
- Zhang M., Shao C., Li F., Gan Y., Sun J., (2007). Spread Spectrum Code Estimation by Artificial Fish Swarm Algorithm, IEEE International Symposium on Intelligent Signal Processing (WISP).
- Mehdi N., Ghodrati S., Mehdi S. ve Adel N.T., (2012). Artificial Fish Swarm Algorithm: A Survey of the State-of-the-Art, Hybridization, Combinatorial and Indicative Applications, Springer Science.
- Bao L., Chu J., Liang Q., (2016). Artificial Fish Swarm Algorithm Based on Improved Topological Structure, *Computer Science and Application*, 6(3), 137-142.
- Duan Q., Mao M., Duan P., Hu B., (2016). An Improved Artificial Fish Swarm Algorithm Optimized by Particle Swarm Optimization Algorithm with Extended Memory, *Kybernetes*, 45(2): 210-222.
- Mehdi N., Ghodrati S. ve Mehdi S., (2012). Swallow Swarm Optimization Algorithm: A New Method to Optimization, *Neural Computing and Applications*.
- Meng X.B., Gao X.Z., Lu L., Liu Y., Zhang H., (2016). A New Bio-Inspired Optimisation Algorithm: Bird Swarm Algorithm, *Journal of Experimental & Theoretical Artificial Intelligence*, 28(4).
- Anderson T.R., (2006). *Biology of the Ubiquitous House Sparrow: From Genes to Populations*, Oxford University Press.
- Kennedy J., Eberhart R.C., Shi Y., (2001). *Swarm Intelligence*, Burlington: MA: Morgan Kaufmann.
- Krause J., Ruxton G.D., (2002). *Living in Groups*, Oxford University Press.
- Lima S.L., Dill L.M., (1990). Behavioral Decisions Made Under the Risk of Predation: A Review and Prospectus, *Canadian Journal of Zoology*, 68, 619–640.
- Bednekoff P.A., Lima S.L., (1998). Randomness, Chaos and Confusion in the Study of Antipredator Vigilance, *Trends in Ecology and Evolution*, 13, 284–287.
- Pulliam H.R., (1973). On the Advantages of Flocking, *Journal of Theoretical Biology*, 38, 419–422.
- Ekman J., (1987). Exposure and Time Use in Willow Tit Flocks: The cost of subordination, *Animal Behavior*, 35, 445–452.
- Beauchamp G., (1998). The Effect of Group Size on Mean Food Intake Rate in Birds, *Biological Reviews*, 73, 449–472.
- Roberts G., (2003). The Group-Size Effect in Non-feeding Animals, *Behavioral Processes*, 63, 127–128.
- Beauchamp G., (2003). Group-size Effects on Vigilance: A search for mechanisms, *Behavioral Processes*, 63, 111–121.
- Barnard C.J., Sibly R.M., (1981). Producers and Scroungers: A General Model and Its Application to Captive Flocks of House Sparrows, *Animal Behavior*, 29, 543–550.
- Giraldeau L.A., Caraco T., (2000). *Social Foraging Theory*, Princeton: Princeton University Press.
- Liker A., Barta Z., (2002). The Effects of Dominance on Social Foraging Tactic Use in House Sparrows, *Behavior*, 139, 1061–1076.
- Johnson C.A., Giraldeau L.A., Grant J.W., (2001). The Effect of Handling Time on Interference Among House Sparrows Foraging at Different Seed Densities, *Behavior*, 138, 597–614.
- Barta Z., Giraldeau L.A., (2000). Daily Patterns of Optimal Producer and Scrounger Use Under Predation Hazard: A State-Dependent Dynamic Game Analysis, *American Naturalist*, 155, 570–582.
- Sirot E., (2006). Social Information, Antipredatory Vigilance and Fight in Bird Flocks, *Animal Behavior*, 72, 373–382.
- Askarzadeh A., (2016). A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, *Computers & Structures*, 169, 1-12.
- Akyol S., Alatas B., (2016). Efficiency Evaluation of Crow Search Algorithm in Benchmark Functions for Optimization, *ICENS 2016*.

A new approach to swarm intelligence: Bird Swarm Algorithm

Extended abstract

Optimization known as also mathematical programming, is a collection of processes that selects the most appropriate values of decision variables according to a goal (evaluation) function. Many algorithms have been proposed for optimization problems. Most of these algorithms need mathematical models for model of system and objective function.

Metaheuristics algorithms are the algorithms that utilize a simple approach as a solution technique of search and optimization problems and are recently getting strong and becoming more popular due to their advantages. They are population based techniques and begin to search the solution with multiple points. They have good reputation and wide acceptability as being powerful tools for many different fields such as management science, engineering, computer, etc. and new versions of these algorithms have been proposed. General purpose heuristic optimization algorithms are evaluated in eight different groups including biology-based, physics-based, swarm-based, social-based, music-based, chemical-based, sports based, and mathematics based. Due to the philosophy of continually searching the best and absence of the most efficient metaheuristic method for all types of problems, novel algorithms or new variants of current algorithms are being proposed.

Swarm optimization algorithms are relatively newer subfield of computational intelligence and recently getting strong and becoming more popular. They have been developed by observing the movements of live swarms such as bird, fish, cat and bee. They are inspired from intelligent behaviors resulting from the local interactions of swarm agents between each other and environment. They are adaptable and general purposed solution methods that can be applied to the high-scale combinatorial and non-linear search and optimization problems in case of concurrent different decision variables, objective functions, and constraints and they do not depend on the solution space type, the number of decision variables, and the number of constraint functions. Furthermore, they do not require very well defined mathematical models that are hard to derive. Their computation power is also good and they do not require excessive computation time. Their

transformations and adaptations are easy. Due to these advantages, these algorithms are densely being used in many different fields.

In this work, bird swarm algorithm that is one of the most current swarm intelligence optimization algorithm was studied in detail. In this study, the performance of this new algorithm is tested within unimodal and multi modal benchmark functions with different dimensions. In these investigations, tendency of converging to optimum and the obtained result values are used as a measure of performance. Experimental results have been presented and interpreted through comparative tables. It is expected that this algorithm will be efficiently used in many different types of complex problems due to high performance of the algorithm in both unimodal and multi modal functions.

Keywords: Swarm intelligence based optimization algorithms, Bird swarm algorithm, Benchmark functions.