Nevşehir Üniversitesi Fen Bilimleri Enstitü Dergisi 1 (2012) 36-50

Güncel Sürü Zekâsı Optimizasyon Algoritmaları

Sinem AKYOL ve Bilal ALATAŞ

Tunceli Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Tunceli-TÜRKİYE

Özet

Optimizasyon, bir problemde belirli koşullar altında mümkün olan alternatifler içinden en iyisini seçme işlemidir. Optimizasyon problemleri için birçok algoritma önerilmiştir. Sezgisel algoritmalar, büyük boyutlu optimizasyon problemleri için, kabul edilebilir sürede optimuma yakın çözümler verebilen algoritmalardır. Genel amaçlı sezgisel optimizasyon algoritmaları, biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olmak üzere altı farklı grupta değerlendirilmektedir. Sürü zekâsı tabanlı optimizasyon algoritmaları kuş, balık, kedi ve arı gibi canlı sürülerinin hareketlerinin incelenmesiyle geliştirilmiştir. Bu çalışmada, sürü zekâsı optimizasyon algoritmaları (Ateşböceği Algoritması, Ateşböceği Sürü Optimizasyonu, Karınca Koloni Optimizasyonu, Parçacık Sürü Optimizasyonu, Yapay Balık Sürüsü Algoritması, Bakteriyel Besin Arama Optimizasyon Algoritması, Kurt Koloni Algoritması) tanıtılmış ve bu optimizasyonlardan kedi sürüsü optimizasyonu ile yapay arı koloni algoritması ayrıntılı olarak incelenmiştir.

Anahtar Kelimeler: *Sürü tabanlı optimizasyon, Kedi sürüsü optimizasyonu, Yapay arı koloni algoritması*

The Current Swarm Intelligence Optimization Algorithms

Abstract

Optimization is the process of finding the best solution of a problem. There are many optimization algorithms proposed for optimization problems. The heuristic algorithm can give solutions close to optimum in an acceptable period of time for large-scaled optimization problems. The metaheuristic optimization algorithms are evaluated in six different groups which are biology-based, physics-based, swarm-based, social-based, music-based, and chemistry-based. The swarm-based optimization algorithms have been developed by observing the behaviors of creatures e.g. birds, fishes, cats, bees etc. In this study, swarm-based optimization algorithms (Firefly Algorithm, Glowworm Swarm Optimization, Ant Colony Optimization, Particle Swarm Optimization, Artificial Fish-Swarm Algorithm, Bacterial Foraging Optimization Algorithm, Wolf Colony Algorithm) are described and cat swarm optimization and artificial bee colony algortihms are studied in detail.

Keywords: *Swarm-based optimization, Cat swarm optimization, Artificial bee colony algorithm*

1.Giriş

Optimizasyon, en iyileme anlamına gelmektedir. Bir problem için, verilen şartlar altında tüm çözümler arasından en iyi çözümü elde etme işidir. Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilmektedir (Murty, 2003).

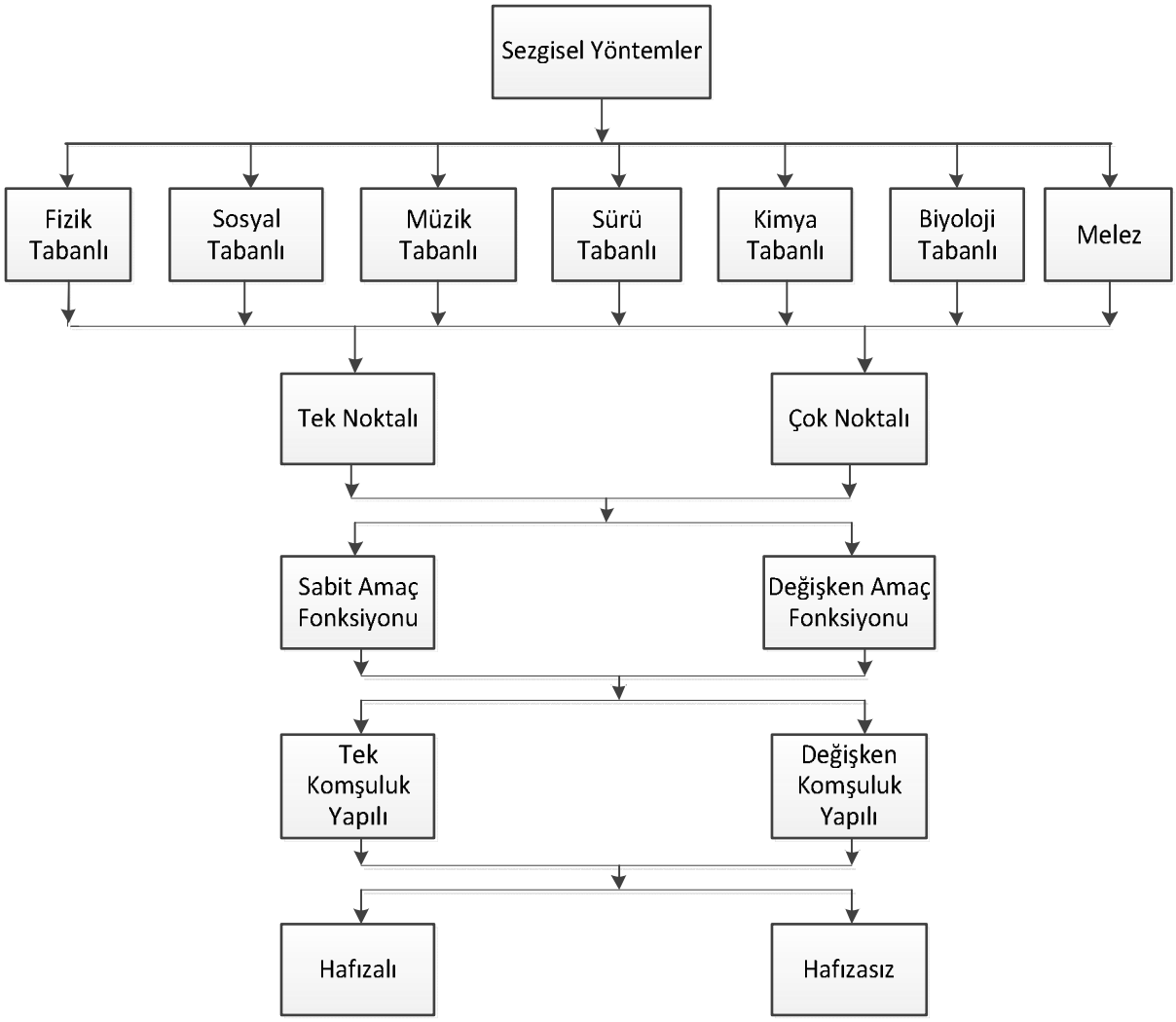
Bazen tek başlarına hiçbir iş yapamayan varlıklar, toplu hareket ettiklerinde çok zekice davranışlar sergileyebilmektedir. Bir topluluğa ait bireyler, en iyi bireyin davranışından ya da diğer bireylerin davranışlarından ve kendi deneyimlerinden yararlanarak yorum yapmakta ve bu bilgileri ileride karşılaşacakları problemlerin çözümleri için bir araç olarak kullanmaktadırlar. Örneğin, bir canlı sürüsünü oluşturan bireylerden birisi bir tehlike sezdiğinde bu tehlikeye karşı tepki verir ve bu tepki sürü içinde ilerleyip tüm bireylerin tehlikeye karşı ortak bir davranış sergilemesini sağlar. Canlıların sürü içerisindeki bu hareketleri gözlemlenerek sürü zekâsı tabanlı optimizasyon algoritmaları geliştirilmiştir.

Bu çalışmada sürü optimizasyon algoritmaları anlatılmış ve bu algoritmalardan Kedi Sürüsü Optimizasyonu (KSO) ve Yapay Arı Koloni Algoritması (YAKA) ayrıntılı olarak incelenmiştir. İkinci bölümde sezgisel optimizasyon ve sezgisel optimizasyonun grupları hakkında bilgi verilmiştir. Üçüncü bölümde sürü zekâsı optimizasyon algoritmalarından bahsedilmiştir. Dördüncü bölümde KSO hakkında bilgi verilmiştir. Beşinci bölümde Yapay Arı Sistemleri ile ilgili yapılan çalışmalar bir tablo halinde sunulmuştur ve YAKA ayrıntılı olarak incelenmiştir. Son olarak sonuçlar altıncı bölümde yazılarak çalışma sonlandırılmıştır.

2.Sezgisel Optimizasyon

Sezgisel algoritmalar, herhangi bir amacı gerçekleştirmek veya hedefe varmak için doğal fenomenlerden esinlenen algoritmalardır. Bu algoritmaların, çözüm uzayında optimum çözüme yakınsaması ispat edilememektedir. Yani sezgisel algoritmalar yakınsama özelliğine sahip olmaktadır, ama kesin çözümü garanti edememektedir ve bu kesin çözümün yakınlarında bir çözüm garanti edebilmektedir. Anlaşılırlık yönünden sezgisel algoritmaların karar verici açısından çok daha basit olabilmesinden, optimizasyon problemlerinin kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olmasından ve öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanabilirliğinden sezgisel algoritmalara ihtiyaç duyulmaktadır (Karaboğa, 2011).

Genel amaçlı sezgisel yöntemler; biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olmak üzere altı farklı grupta değerlendirilmektedir. Ayrıca bunların birleşimi olan melez yöntemler de vardır. Bahsedilen bu yöntemler Şekil 1'de sunulmaktadır. Genetik algoritma (GA), diferansiyel gelişim algoritması, karınca koloni algoritmaları, yapay sinir ağları, arı koloni algoritmaları ve yapay bağışıklık sistemleri biyolojik tabanlı; emperyalist yarışmacı algoritma, parlamenter optimizasyon algoritması ve tabu arama sosyal tabanlı; yapay kimyasal reaksiyon algoritması kimya tabanlı; armoni arama algoritması müzik tabanlı; ısıl işlem, büyük patlama büyük sıçrama, yerçekimsel arama algoritması, merkez kuvvet optimizasyonu, zeki su damlacıkları algoritması ve elektromanyetizma algoritması fizik tabanlı ve Parçacık Sürü Optimizasyonu (PSO), KSO sürü tabanlı algoritma ve modellerdir. Kültürel algoritma da hem biyoloji hem de sosyal tabanlı algoritma olarak sınıflandırılabilir (Alataş, 2007).



Şekil 1. Sezgisel Yöntemler

3.Sürü Zekâsı Optimizasyon Algoritmaları

Sürü, birbirleriyle etkilesen dağınık yapılı bireyler yığını anlamında kullanılır. Bireyler insan veya karınca olarak ifade edilebilir. Sürülerde *N* adet temsilci bir amaca yönelik davranışı gerçekleştirmek ve hedefe ulaşmak için birlikte çalışmaktadır. Kolaylıkla gözlenebilen bu "kollektif zekâ" temsilciler arasında sık tekrarlanan davranışlardan doğmaktadır. Temsilciler faaliyetlerini idare etmek için basit bireysel kurallar kullanmakta ve grubun kalan kısmıyla etkileşim yolu ile sürü amaçlarına ulaşmaktadır. Grup faaliyetlerinin toplamından bir çeşit kendini örgütleme doğmaktadır.

Aşağıda şimdiye kadar farklı araştırmacıların önerdiği sürü zekâsı optimizasyon algoritmaları alt başlıklar halinde açıklanmıştır.

3.1. Ateş Böceği Algoritması

Ateş böceği algoritması, Dr. Xin-She Yang (Cambridge Üniversitesi-2007) tarafından geliştirilen ve tropikal iklim bölgelerindeki ateşböceklerinin sosyal davranışlarını baz alan bir metasezgisel optimizasyon algoritmasıdır (Yang, 2009). Bu algoritma diğer sürü zekâsı tabanlı algoritmalarla bir çok benzerliği bulunmasına rağmen kavram ve uygulamada daha basittir. Bir ateş böceğinin ışıklarını yakıp söndürmesinin birincil amacı, diğer ateş böceklerini çekmek için bir sinyal sistemi olarak hareket etmektir. Yanıp sönen ışıkların üretimindeki karmaşık biyokimyasal sürecin detayları ve gerçek amacı bilim dünyasında hâlâ bir tartışma konusu olmasına rağmen, birçok araştırmacı yanıp sönen ışıkların ateşböceğine, arkadaşlarını bulmada, olası avlarını çekmede ve avcılarından kendilerini korumada yardımcı olduğuna inanmaktadır.

Ateşböceği algoritmasında, verimli optimal çözümler elde etmek için, verilen bir optimizasyon probleminin amaç fonksiyonu, ateşböceği sürüsüne parlak ve daha çekici yerlere gitmede yardım eden yanıp sönen ışık ya da ışık şiddeti ile ilişkili olmaktadır. Bütün ateş böcekleri tek cins olarak kabul edilmektedir ve birbirilerini çekmeleri bu algoritmanın temelini oluşturmaktadır. Bir ateş böceği ne kadar parlak olursa diğer ateş böcekleri için o kadar çekici hale gelmektedir. Kendisinden daha parlak bir ateşböceği gördüğünde ona doğru gidecektir (Apostolopoulos, Vlachos, 2011; Yang, 2010).

3.2. Ateş Böceği Sürü Optimizasyonu

Ateş böceği sürü optimizasyonu, K. N. Krishnanand ve D. Ghose tarafından 2005 yılında geliştirilmiştir (Krishnanand, Ghose, 2005). Çok modelli fonksiyonları optimize etmek için önerilen sürü zekâsı tabanlı bir algoritmadır. Bu yöntemi kullanmanın temel amacı tüm yerel maksimumları yakalamayı sağlamaktır. Çok modelli fonksiyon optimizasyon problemlerinde, ateş böceği sürü optimizasyonu ve önceki yaklaşımlar arasındaki en önemli fark, birden çok zirveyi verimli bir şekilde yerleştiren sürüdeki bireylerin kullandığı dinamik karar alanıdır. Sürüdeki her bir birey komşularını seçmek için karar alanını kullanmaktadır ve komşularından aldığı sinyal gücüyle hareketlerini belirlemektedir. Bu biraz, bir ateş böceğinin eşlerini veya avlarını çekmek için kullandığı ışık yakıp söndürmeye neden olan lusiferine (bir enzim ile birleşerek ışın üreten bir madde) benzer olmaktadır. Daha parlak ışık daha çekici olmaktadır.

Bu algoritmanın ateş böceği algoritmasından (firefly algorithm) farkı "komşuların yeterlilik sayısı" sınırı olmaması ve mesafeye dayalı herhangi bir algı sınırı olmamasıdır (Krishnanand, Ghose, 2009).

3.3. Karınca Koloni Optimizasyonu

Gerçek karınca koloni davranışlarının matematiksel modelleri üzerine dayalı bir algoritmadır. İlk çalışma Dorigo ve arkadaşları tarafından yapılmıştır (Dorigo, Maniezzo, Colorni, 1991). Yaptıkları çalışmada kendi sistemlerine “karınca sistemi”, elde ettikleri algoritmaya ise “karınca algoritması” adını vermişlerdir. Karınca çevre şartlarına göre besin kaynağı ile evi arasında gidebileceği yolları belirlemektedir. Belirlenen yollardan birinden ilk geçen karınca yola feremon adında bir koku bırakmaktadır. Eğer yol kısa ise bu koku daha yoğun olmaktadır ve diğer karıncalar da aynı şekilde yolda devam etmektedirler. İki yolun kesiştiği noktada karınca hangi yola gideceğini belirlemektedir. Hangi yolu seçeceğine ilk önce koku miktarının yoğunluğuna göre ikinci olarak ise gelişigüzel bir ölçüte göre karar vermektedir. Bu gelişigüzel seçimin nedeni ise bütün karıncaların aynı yolda gitmesini engelleyerek yeni ve daha kısa yolları keşfetmektir (Karaboğa, 2011).

3.4. Parçacık Sürü Optimizasyonu

Sezgisel yöntemlerden biri olan PSO tekniği ilk olarak kuş ve balık sürülerinin hareketlerinden esinlenerek doğrusal olmayan nümerik problemlere optimal sonuçlar bulmak için 1995–1996 yıllarında sosyolog-psikolog James Kennedy ve elektrik mühendisi Russel Eberhart tarafından ortaya atılmıştır (Kennedy, Eberhart, 1995). PSO popülasyon tabanlı olasılıksal bir optimizasyon yöntemi olup çok parametreli ve çok değişkenli optimizasyon problemlerine çözümler üretmek için kullanılmaktadır (Alataş, 2007).

Parçacık sürü kavramı basitleştirilmiş sosyal sistemin bir simülasyonu olarak ortaya çıkmıştır. Başlangıçtaki amaç, kuş ya da balık sürü koreografisinin grafiksel olarak simülasyonlarını yapmaktı. Ancak grafiksel simülasyondan sonra, parçacık sürü modelinin bir optimizasyon yöntemi olarak kullanılabileceği keşfedilmiştir.

Kuş toplulukları gerçek yiyecek kaynağını bilmemelerine rağmen, yiyecek kaynağından ne kadar uzakta olduklarını öğrenmeye çalışırlar. Öğrenmek için izlenen yöntem yiyecek kaynağına en yakın olan kuşu izlemektir. PSO'da her bir kuş parçacık olarak, kuş topluluğu da sürü olarak temsil edilir. Parçacık hareket ettiğinde, kendi koordinatlarının uygunluk değeri yani yiyeceğe ne kadar uzaklıkta olduğu hesaplanır. Bir parçacık, koordinatlarını, hızını yani çözüm uzayındaki her boyutta ne kadar hızla ilerlediği bilgisini, şimdiye kadar elde ettiği en iyi uygunluk değerini ve bu değeri elde ettiği koordinatları hatırlamalıdır. Çözüm uzayında her boyuttaki hızının ve yönünün her seferinde nasıl değişeceği, komşularının en iyi koordinatları ve kendi kişisel en iyi koordinatlarının birleşiminden elde edilecektir (Alataş, 2007).

3.5. Yapay Balık Sürüsü Algoritması

Yapay balık sürüsü optimizasyonu, yiyecek aramada balık sürüsünün sosyal davranışlarının benzetimi tabanlı bir zeki optimizasyon algoritmasıdır. Doğada balık, yiyeceğini besin değeri yüksek alanları tek tek arayarak ya da diğer balıkları izleyerek bulabilmektedir. Çok balıklı bölgenin besin değeri genellikle daha yüksek olmaktadır. Bu optimizasyonun temel fikri, küresel optimuma ulaşmak için balık bireyinin yerel aramasıyla toplanma ve izleme gibi balık davranışlarını taklit etmektir. Bir yapay balığın yaşadığı ortam başlıca çözüm uzayıdır ve diğer yapay balıkların konumudur. Bir sonraki davranışı mevcut durumuna ve yerel çevresel durumuna bağlı olmaktadır. Bir yapay balık kendi faaliyetleri ve arkadaşlarının faaliyetleri yolu ile çevresini etkileyecektir (Jiang, Yuan, Cheng, 2009).

3.6. Bakteriyel Besin Arama Optimizasyon Algoritması

Bakteriyel Besin Arama Algoritması, E. coli bakterisinin beslenme davranışından esinlenerek karmaşık mühendislik problemlerini çözmek için geliştirilmiş bir hesaplama tekniğidir. Bakteriler, karmaşık yaşam formlarındaki diğer canlılara göre çok daha basit yapıdadırlar. Sınırlı algı ve hareket kabiliyetlerini kullanarak optimum düzeyde enerji harcayıp beslenme faaliyetlerini gerçekleştirmeleri gerekmektedir. Diğer yaşam formlarına nazaran modellenebilmeleri daha kolaydır. Bu türden canlılardan biri olan E. coli bakterisi, yapısı ve çalışma sekli en iyi anlaşılan mikroorganizmalardan birisidir. E. coli bakterisi besin maddesine ulaştığında diğer bakterileri uyarıcı etkiye sahip kimyasal bir madde salgılamaktadır. Bu madde, diğer E. coli bakterilerinin besini bulan bakterinin bulunduğu yere doğru hareket etmesini sağlamaktadır. Eğer gıda yoğunluğu çok fazla ise bakteriler kenetlenerek grup halinde hareket edebilmektedirler (Başbuğ, 2008).

3.7. Kurt Kolonisi Algoritması

Bu algoritma, kurt kolonisinin sıkı bir organize sisteme sahip olmasından esinlenilerek geliştirilmiştir. Kurtlar görevleri diğerleriyle bölüşmektedirler ve avlandıkları zaman tutarlı adımlar atmaktadırlar. Az miktarda yapay kurt aktif olduğu av aralığında aramaya atanmaktadır. Arama kurtları avı keşfettiği zaman, avın konumunu diğer kurtlara ulumayla bildirmektedir. Diğer yapay kurtlar ava yaklaşmakta ve avı kuşatmaktadırlar. Kurt kolonisinin atanma kuralı, yiyeceğin ilk olarak güçlü kurda atanması ve daha sonra zayıf olana atanmasıdır. Kurt koloni algoritması bu davranışların taklit edilmesiyle geliştirilmiştir (Liu, Yan, Liu, Wu, 2011).

4. Kedi Sürüsü Optimizasyonu

KSO, kedigillerin hareketlerinin incelenmesiyle ortaya çıkarılmıştır. Kedilerin davranışlarına benzetim yapılarak iki alt model oluşturulmuştur. Bu optimizasyondaki matematiksel modeller kedilerin hareketlerinin çözümlenmesiyle meydana gelmiştir (Chu, Tsai, Pan, 2006). KSO kullanılarak yapılan bazı çalışmalar şunlardır: Santaso ve arkadaşları kümeleme problemi için KSO'yu kullanmışlardır (Santosa, Ningrum, 2009). Wang ve arkadaşları en az önemli bitin yerine en iyisini getirmek için KSO stratejisi kullanmışlardır (Wang, Chang, Li, 2010). Hwang ve arkadaşları müşterilere göre en uygun sözleşme kapasitesi problemini çözmek için KSO ve PSO'yu birlikte kullanmışlardır (Hwang, Chen, Pan, Huang, 2009). Destek vektör makinesi için özellik seçimi ve parametre optimizasyonu probleminde KSO, Lin ve Chen tarafından önerilmiştir (Lin, Chien, 2009). Kalaiselvan ve arkadaşları filigran performansının arttırılması amacıyla KSO uygulamışlardır (Kalaiselvan, Lavanya, Natrajan, 2011). Wang ve Wu e-öğrenmede duygu tanıma problemi için KSO'yu destek vektör makinesiyle birlikte kullanmışlardır (Wang, Wu, 2011). KSO algoritmasının paralel versiyonu da Tsai ve arkadaşları tarafından önerilmiştir (Tsai, Pan, Chen, Liao, Hao, 2008).

4.1.Kedigillerin Hareketlerini İnceleme

Biyolojik sınıflandırmaya göre, yaklaşık 30 tane farklı örneğin aslan, leopar, kaplan, kedi vb. kedi cinsinden yaratık uzayı vardır. Çoğunun farklı yaşam alanı olmasına rağmen, kedigiller benzer davranış modellerini sergilemektedirler. Kedilerin avlanma becerisi, kediler için kalıtsal değildir, alıştırmalar aracılığıyla kazanılmaktadır. Bu avlanma becerisi ile, yaban kedileri yiyeceklerini temin etmeyi sağlamaktadırlar ve türlerinin hayatta kalması garanti altına alınmaktadır. Ayrıca evcil kedilerde benzer doğal avlanma becerisi ve hareketli nesnelere güçlü bir merak sergilemektedirler. Bütün kedilerin, bu güçlü merakı paylaşmasına rağmen, zamanlarının çoğunu hareketsiz (durağan) geçirmektedirler. Kediler çok yüksek seviyede atikliğe sahiptirler. Bu atiklik dinlenme zamanlarında bile kendilerini bırakmamakta büyük geniş gözler sürekli olarak etrafını gözetlemektedir. Kediler, çok zeki ve bilinçli (planlı) yaratıklar oldukları halde tembel gibi görünmektedirler. Kedilerin dokuz canlı olduğu söylenerek, kedilerin güçlü canlılığına gönderme yapılmaktadır. Ev içinde olan kedi sık sık alçak frekansta ses çıkarmaktadır. Kediler hoşnut oldukları, tehlikede veya hasta oldukları zaman mırlarlar. Mırlamanın alçak frekansının, hücre onarımına yardım ettiğine inanılmaktadır. Bu kedilerin canlılığı için bir neden olabilir (Chu, Tsai, 2007).

4.2.Önerilen Algoritmalar

Önerilen KSO için kedilerin başlıca iki tane davranışsal özelliği modellenmiştir. Bunlar "arama modu" ve "izleme modu" olarak isimlendirilmiştir. Bu iki modun birleşimi KSO'ya daha iyi bir performans için izin vermektedir (Chu, Tsai, 2007).

4.2.1.Çözüm Kümesinin Sunumu

KSO'da önerilen algoritmada, optimizasyon problemini çözmek için kedileri ve kedilerin davranışlarının modeli kullanılmaktadır, örneğin çözüm kümesini tasvir etmek için kediler kullanılmaktadır.

KSO'da ilk önce iterasyonda kaç kedinin kullanılacağına karar verilmektedir, daha sonra problemi çözmek için kediler KSO'nun içine uygulanmaktadır. Bütün kediler *M* boyuttan oluşan kendi pozisyonlarına, her bir boyut için hızlara, kedinin uyumunu uygunluk fonksiyonuna yansıtan bir uygunluk değerine ve kedinin izleme modunda mı yoksa arama modunda mı olduğunu belirlemek için bir bayrağa (flag'e) sahiptir. Final çözüm kedilerden bir tanesinin en iyi pozisyonu olacaktır. KSO en iyi çözümü iterasyon sonuna ulaşıncaya kadar saklayacaktır (Chu, Tsai, 2007).

4.2.2.Dinlenme ve Tetikte Olma - Arama Modu

Bu alt mod dinlenmede fakat tetikte olmanın bir periyodu boyunca kedinin modellenmesi için kullanılmaktadır (sonraki hareketi için çevresine bakınma). Arama modu aşağıda belirtildiği gibi dört gerekli faktöre sahiptir: arama hafızası havuzu (*AHH*), seçilen boyutun arama aralığı (*SBA*), değişen boyutların sayısı (*DBS*) ve kendi pozisyonunu değerlendirme (*KPD*). *AHH* her bir kedinin arama hafızasının boyutunu tanımlamak için kullanılır, bazı noktaları kediye göre sıralayarak belirtir. Daha sonra anlatılacak kurallara göre kedi, hafıza havuzundan bir nokta ayıracaktır. *SBA* seçilen boyutlar için mutasyon oranını bildirir. Arama modu süresince, eğer bir boyut mutasyon için seçilmişse, yeni ve eski değerler arasındaki farklılık aralık dışında olmamalıdır, aralık *SBA* tarafından tanımlanır. *DBS* boyutlardan kaç tanesinin değişime uğrayacağını ifade eder. Bütün bu faktörler arama modunda önemli rol oynar. *KPD* bilinen bir ikili değerdir, ve kedilerin bulunduğu noktanın hareket için aday noktalardan biri olup olmayacağını bildirir. *KPD*, *AHH* değerini etkilemez (Chu, Tsai, 2007).

Arama modunun adımları:

Adım 1: *j=AHH* olduğunda $kedi_k$'nin bulunduğu pozisyonda *j* tane kopya yap. Eğer *KPD* değeri doğruysa, *j=(AHH-1)* olur, sonra mevcut pozisyonu, adaylardan biri olarak tut.

Adım 2: *DBS*' ye göre, her bir kopya için mevcut değerin *SBA* yüzdesini gelişigüzel olarak artır veya azalt ve eskisiyle yerini değiştir.

Adım 3: Bütün aday noktaların uygunluk değerini hesapla.

Adım 4: Eğer bütün uygunluk değerleri (*UD*) tam olarak aynı değilse, eşitliğe göre her bir aday noktanın seçilen olasılığını Eşitlik (1)'e göre hesapla, aksi takdirde her bir aday noktanın seçilen olasılıklarının tümüne 1 ata.

Adım 5: Aday noktalardan taşınmak için noktayı gelişigüzel çıkart (ayır), ve $kedi_k$' nin pozisyonuyla değiştir.

$$P_i = \frac{|UD_i - UD_b|}{|UD_{maks} - UD_{min}|}, \; 0 < i < j \; olduğunda \qquad (1)$$

Eğer uygunluk fonksiyonunun amacı minimum çözümü bulmaksa, $UD_b=UD_{maks}$, aksi takdirde $UD_b=UD_{min}$ (Chu, Tsai, 2007; Santosa, Ningrum, 2009; Wang, Chang, Li, 2010).

4.2.3.Hareket- İzleme Modu

İzleme modu, kedinin hedefi izlemedeki durumunu modellemek için bir alt modeldir. Bir kere kedi izleme moduna gittiğinde, her bir boyutu için kendi hızlarına göre hareket etmektedir. İzleme modundaki çalışma aşağıdaki gibi açıklanabilir (Chu, Tsai, 2007).

Adım 1: Bütün boyutlar için hızları ($v_{k,d}$) Eşitlik (2)'yi kullanarak güncelle.

Adım 2: Hızların, maksimum hızın aralığında olduğunu kontrol et. Yeni hız aralığının üzerinde olduğu takdirde limite eşitliği at (*limit=sınır*).

Adım 3: *kedi*$_k$'nin pozisyonunu Eşitlik (3)'ü kullanarak güncelle.

$$v_{k,d} = v_{k,d} + r_1 * c_1 (x_{best,d} - x_{k,d}), \qquad d = 1,2,\dots,M \tag{2}$$

$x_{best,d}$, en iyi uygunluk değerine sahip kedinin pozisyonu; $x_{k,d}$, *kedi*$_k$'nin pozisyonu, c_1 bir sabit ve r_1 gelişigüzel bir değerdir (Chu, Tsai, 2007; Santosa, Ningrum, 2009; Wang, Chang, Li, 2010).

$$x_{k,d} = x_{k,d} + v_{k,d} \tag{3}$$

4.3.Kedi Sürüsü Optimizasyonunun Temel Tanımlaması

KSO'nun, arama modu ve izleme modu adında iki alt modu vardır. Bu iki modu algoritma şeklinde birleştirmek için, arama moduyla izleme modunu birleştirmeyi gerektiren bir karışım oranı (*KO*) tanımlanmaktadır. Kediler dinlenme zamanında hareket etmeye karar verdiyseler, hareket çok dikkatli ve yavaşça yapılmaktadır. Bu hareket, arama moduna yansıtılmaktadır. İzleme modu kedi tarafından bir hedefin takip edilmesini modellemektedir. Kediler, enerji kaynaklarını fazla kullanmalarına yol açan objeleri takip etmeye çok az zaman harcamaktadırlar. Kedilerin zamanlarının çoğunu dinlenmeye ve gözetlemeye (mesela zamanlarının çoğu arama modunda geçmektedir) harcadığını garantilemek için *KO*'ya çok küçük bir değer atanmaktadır. KSO'nun süreci aşağıda anlatılmaktadır (Chu, Tsai, 2007; Santosa, Ningrum, 2009; Wang, Chang, Li, 2010).

Adım 1: Süreçte *N* tane kedi yarat.

Adım 2: *M* boyutlu çözüm uzayına gelişigüzel kediler serpiştir ve her kedinin hızına maksimum hız aralığında olan gelişigüzel değerler ver. Sonra kedilerin numaralarını gelişigüzel ayır ve onları *KO*'ya göre izleme modunun içine ata ve diğerlerini arama modunun içine ata.

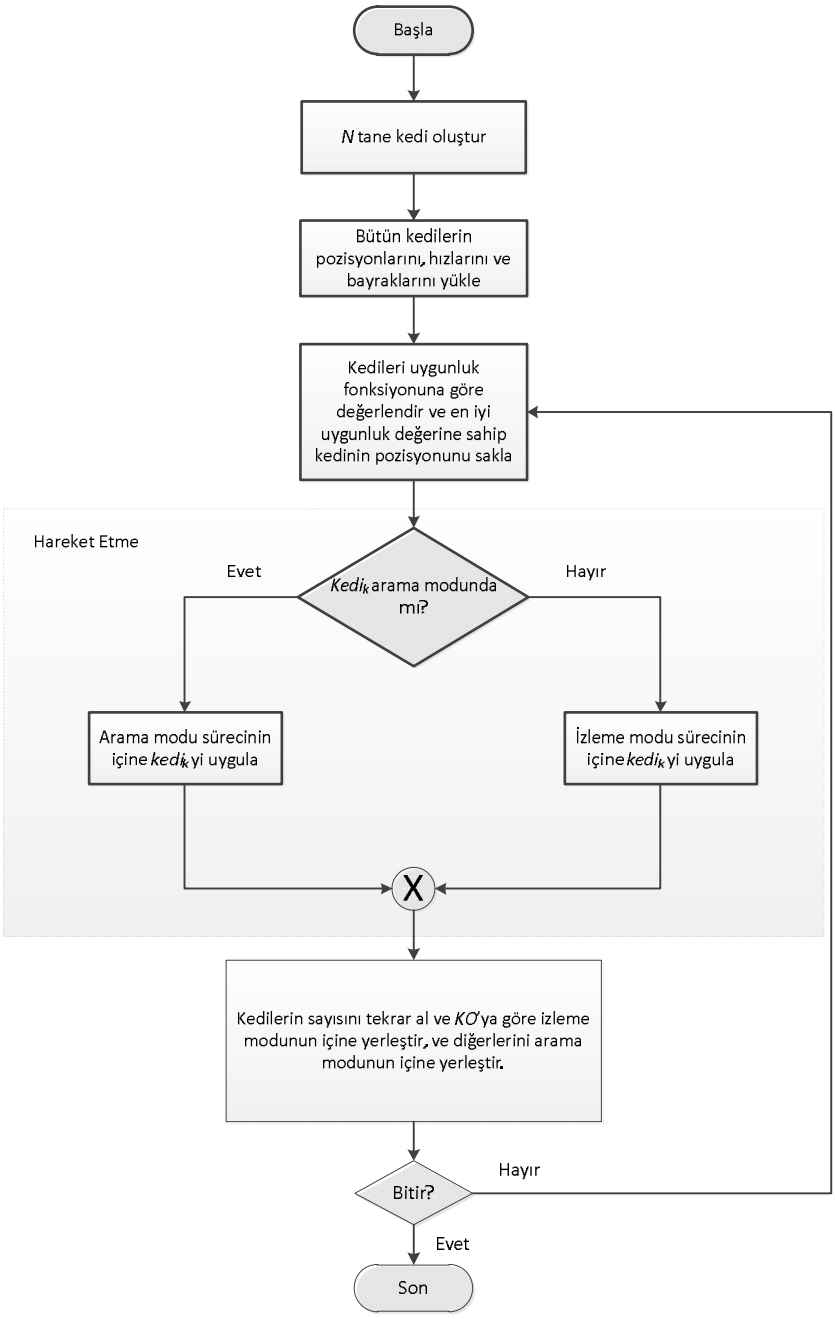
Adım 3: Amaç kriterini yansıtan uygunluk fonksiyonunu kedilerin pozisyona göre belirle ve en iyi kediyi hafızada sakla. Şimdiye kadarki en iyi çözümü yansıtmasından dolayı sadece en iyi kedinin pozisyonu saklanır.

Adım 4: Bayraklarına (flag'lerine) göre kedileri hareket ettir, eğer *kedi*$_k$ arama modundaysa, kediyi arama modu sürecine uygula, aksi takdirde izleme modu sürecine uygula.

Adım 5: Kedilerin numarasını yeniden ayır ve *KO*'ya göre izleme modunun içine ata, sonra diğer kedileri arama modunun içine ata.

Adım 6: Sonlandırma (bitirme) koşullarını kontrol et, eğer doyuma ulaşmışsa programı sonlandır, aksi durumda Adım 3'ten Adım 5'e kadar tekrar et.

KSO sürecinin diyagramı Şekil 2'de sunulmaktadır.



Şekil 2. KSO'nun Süreç Diyagramı

5.Yapay Arı Koloni Algoritması

Arıların yiyecek arama davranışı, öğrenme, bilgi paylaşımı ve ezberleme özellikleri, son zamanlarda sürü zekâsında en ilginç araştırma alanlarından biri olmuştur. Bal arıları üzerinde çalışmalar son yıllarda literatürde bir artış trendi içindedir. Ayrıntılı literatür taraması için "A survey: algorithms simulating bee swarm intelligence" (Karaboga, Akay, 2009) makalesi incelenebilir.

Doğal bir arı kolonisinde arılar arasında yapılacak işlere göre bir görev paylaşımı vardır. Arılar bu iş paylaşımını merkezi bir birim olmadan, kendi kendilerine gerçekleştirmektedirler. Bu iş paylaşımı ve kendi kendine organize olabilme sürü zekâsının iki önemli özelliğidir. Tereshko'nun reaktif difüzyon denklemine dayalı olarak ortaklaşa zekânın ortaya çıkmasını sağlayan minimal yiyecek arama modelinde üç temel bileşen vardır. Bunlar yiyecek kaynakları, görevli işçi arılar ve görevsiz işçi arılardır. Ayrıca bu minimal model bir yiyecek kaynağına yönelme ve yiyecek kaynağını bırakma olmak üzere iki modda çalışmaktadır (Tereshko, 2000).

Arılar bal, polen veya nektar bulmak için yiyecek kaynaklarına gitmektedirler. Yiyecek kaynağının değeri, yuvaya yakınlığı, çeşidi, nektar yoğunluğu, nektarın çıkarılmasının kolaylığı gibi birçok etkene bağlıdır. Ayrıca basit olması açısından sadece yiyecek kaynağının zenginliği gibi tek bir özellik de ele alınabilir. Görevli işçi arılar, nektarın, önceden keşfedilmiş olan belli kaynaklardan kovana getirilmesinden sorumludurlar ve gittikleri kaynağın kalitesi ve yeriyle ilgili bilgileri kovandaki diğer arılarla paylaşmaktadırlar. Görevsiz işçi arılar ise nektarı toplanabilecek yeni yiyecek kaynaklarını aramaktadırlar. İçsel bir dürtüye veya dış bir etmene bağlı olarak gelişigüzel kaynak arayışında olan kaşif arılar ve kovanda bekleyip görevli arıları izleyerek bu arılar tarafından paylaşılan bilgiyi kullanıp yeni bir kaynağa yönelen gözcü arılar olmak üzere görevi belirsiz iki tür arı vardır. Kaşif arıların sayısının kovandaki diğer arılara oranı %5-10 arasındadır (Karaboğa, 2011).

Ortaklaşa bilginin oluşumundaki en önemli etmen arılar arasındaki bilgi paylaşımıdır. Yiyecek kaynağının yeri ve kalitesi hakkındaki bilgi paylaşımı kovandaki dans alanında olmaktadır. Dans eden arıya diğer arılar antenleri aracılığıyla dokunarak kaynağın tadı ve kokusu hakkında da bilgi alırlar. Arıların yiyecek arama davranışı modellenerek geliştirilen en güncel algoritma YAKA'dır. Bu algoritmada temel alınan modelde basit olması nedeniyle bazı kabuller yapılmaktadır. Buna göre: (kaynak sayısı = görevli arı sayısı = işçi arıların sayısı) olarak belirlenmektedir ve nektarı tükenmiş kaynağın görevli arısı kaşif arıya dönüşmektedir. Yiyecek kaynaklarındaki nektar miktarı, kaynaklarla ilgili çözümlerin uygunluğuna ve bu kaynakların yerleri ise optimizasyon problemine ait olası çözümlere karşılık gelmektedir. YAKA en fazla nektara sahip yiyecek kaynağının yerini bulmaya çalışarak uzaydaki çözümlerden problemin minimumunu veya maksimumunu veren çözümü bulmaya çalışmaktadır (Karaboğa, 2011).

YAKA kullanılarak yapılan bazı çalışmalar şunlardır: Hedayatzadeh, Hasanizadeh, Akbari ve Ziarati çok amaçlı problemlerin optimizasyonunda YAKA'yı kullanmışlardır (Hedayatzadeh, Hasanizadeh, Akbari, Ziarati, 2010). Banharnsakun, Achalakul ve Sirinaovakul, büyük boyutlu problemler için YAKA'nın dağıtık bir sürümünü önermişlerdir (Banharnsakun, Achalakul, Sirinaovakul, 2010). Alam, Kabir ve Islam sürekli fonksiyon optimizasyonu için YAKA'yı kullanmışlardır (Alam, Kabir, Islam, 2010). Karaboğa ve Görkemli gezgin satıcı probleminde YAKA'yı uygulamışlardır (Karaboğa, Görkemli, 2011). Guo, Cheng ve Liang sayısal optimizasyon algoritmasında YAKA'yı kullanmışlardır (Guo, Cheng, Liang, 2011). Ning ve Zhang bulanık sinir ağına dayalı bir konuşma tanıma sisteminde YAKA'yı uygulamışlardır (Ning, Zhang, 2011). Jadhav, Patel, Sharma ve Roy YAKA'yı rüzgar enerjisi geçişimli kombine emisyon görev dağıtımı problemi için kullanmışlardır (Jadhav, Patel, Sharma, Roy, 2011). Çelik, Karaboğa ve Köylü kural tabanlı sınıflandırma modelinin optimizasyonu için YAKA'yı önermişlerdir (Çelik, Karaboğa, Köylü, 2011).

5.1. Önerilen Algoritma

YAKA'da arama uzayı, yiyecek kaynaklarını içeren kovan çevresidir. Algoritma ilk olarak arama uzayındaki çözümlere karşılık gelen gelişigüzel yiyecek kaynakları yerleri üretmektedir ve bu yerler her bir parametrenin alt ve üst sınırları arasında gelişigüzel değer üretilerek oluşturulmaktadır. Aynı zamanda oluşturulan her bir kaynağın geliştirilememe sayaçları da sıfırlanmaktadır. Bu aşamadan sonra durdurma kriteri sağlanıncaya kadar yiyecek kaynakları görevli arı, kaşif arı ve gözcü arı süreçlerinden geçirilerek daha iyisi bulunmaya çalışılmaktadır. Her bir yiyecek kaynağının bir görevli arısı olmaktadır, bu şekilde yiyecek kaynağı sayısı ile görevli arı sayısı eşitlenmektedir. Yiyecek kaynağının komşuluğunda yeni bir yiyecek kaynağı belirlenmektedir. Eğer bu kaynak daha iyi ise işçi arı bu kaynağı hafızasına almaktadır ve geliştirilememe sayacı sıfırlanmaktadır. Aksi durumda ise geliştirilememe sayacı bir arttırılmaktadır (Karaboğa, 2011; Banharnsakun, Achalakul, Sirinaovakul, 2010).

Gözcü arılar danslardan öğrendikleri bilgilerle nektar miktarlarına göre bir kaynak seçmektedir. Burada nektar miktarı uygunluk değerine karşılık gelmektedir ve yiyecek kaynağının bir gözcü arı tarafından seçilme olasılığı rulet tekerleği yöntemi kullanılarak hesaplanmaktadır. Bir kaynağın uygunluk değeri arttıkça, bu kaynak bölgesini seçecek gözcü arı sayısı da artmaktadır. Gözcü arılar daha iyi kaynağı bulabilmek için bir seçim yapmaktadırlar. Eski kaynak ile yeni kaynak arasında karşılaştırma yapmaktadırlar. Eğer eski kaynak daha iyi ise bu çözüm saklanmaktadır ve geliştirilememe sayacı bir arttırılmaktadır. Aksi halde yeni çözüm saklanmaktadır ve geliştirilememe sayacı sıfırlanmaktadır. Bütün gözcü arılar yiyecek kaynaklarına dağılıncaya kadar bu işlemler devam etmektedir (Karaboğa, 2011; Ning, Zhang, 2011).

Geliştirilememe sayacı kontrol edilerek, bir kaynağın tükenip tükenmediğine bakılmaktadır. Eğer yiyecek kaynağı tükenmişse kaynağın görevli arısının bu kaynağı bırakıp yeni kaynak araması gerekmektedir. Bu şekilde görevli arı kaşif arı olmaktadır ve bu arı için gelişigüzel çözüm arama süreci başlamaktadır. Temel YAKA'nın adımları aşağıdaki gibidir

Adım 1: Eşitlik (4) ile x_{ij}, $i=1,...,N$, $j=1..M$, çözümlerine başlangıç değerlerini ata ve geliştirilememe sayaçlarını ($hata_i$) sıfırla. Uygunluk değerlerini hesapla. N yiyecek kaynağı sayısı ve M optimize edilecek parametre sayısıdır.

$$x_{ij} = x_j^{min} + rand(0,1)\left(x_j^{maks} - x_j^{min}\right) \tag{4}$$

Adım 2: $i=1$ den N' ye kadar Eşitlik (5)'i kullanarak x_i çözümünün görevli arısı için yeni bir kaynak üret.

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{5}$$

Bu işlemde üretilen v_{ij} değerinin daha önceden belirtilmiş olan alt ve üst sınırları aşması durumunda Eşitlik (6)'yı kullanarak j. parametreye ait alt veya üst sınır değerlerine ötele.

$$v_{ij} = \begin{cases} x_j^{min}, & v_{ij} < x_j^{min} \\ v_{ij}, & x_j^{min} \le v_{ij} \le x_j^{maks} \\ x_j^{maks}, & v_{ij} > x_j^{maks} \end{cases} \tag{6}$$

Bu kaynağın maliyet değerini $f(v_i)$ Eşitlik (7)'de yerine koyarak bu çözümün uygunluk değerini hesapla. v_i ve x_i arasında seçim işlemini uygula ve daha iyi olanı seç. x_i çözümü gelişememiş ise $hata_i$ bir arttır.

$$uygunluk_i = \begin{cases} 1/(1+f_i)\ , & f_i \geq 0 \\ 1+abs(f_i)\ , & f_i < 0 \end{cases} \quad (7)$$

$f_i, \overrightarrow{v_i}$ kaynağının maliyet değeridir.

Adım 3 : Gözcü arıların seçim işlemi yaparken kullanacakları uygunluk değerine dayalı olasılık değerlerini Eşitlik (8)'i kullanarak hesapla.

$$P_i = \frac{uygunluk_i}{\sum_{j=1}^{N} uygunluk_j} \quad (8)$$

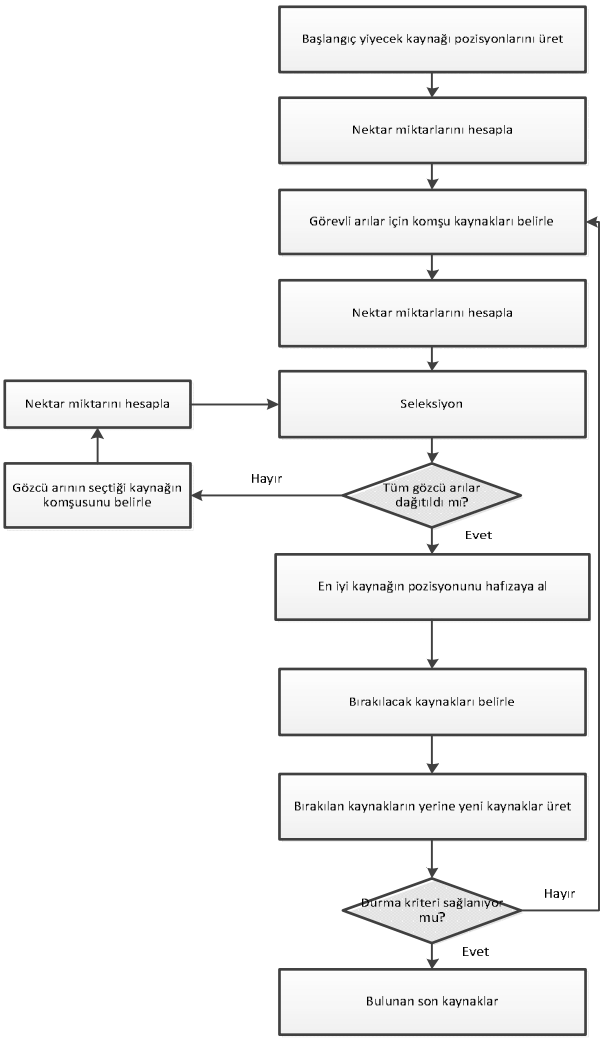
Adım 4 : Rulet tekerleğine göre seçim işleminde her bir kaynak için [0, 1] aralığında üretilen p_i değeri gelişigüzel üretilen bir değerden büyükse gözcü arı için Eşitlik (5)'i kullanarak yeni bir kaynak üret ve üretilen v_i ile x_i arasında seçim işlemi uygula, daha iyi olanı seç. x_i çözümü gelişememişse $hata_i = hata_i + 1$, 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 (4)'ü 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.

YAKA'nın temel adımları Şekil 3'te sunulmaktadır.



Şekil 3. YAKA'nın Akış Diyagramı

6.Sonuç

Optimizasyon problemleri için birçok algoritma önerilmiştir. Sezgisel algoritmalar, büyük boyutlu optimizasyon problemleri için, kabul edilebilir sürede optimuma yakın çözümler verebilen algoritmalardır. Bu çalışmada sezgisel optimizasyon algoritmalarından sürü zekâsı tabanlı olanlar incelenmiş ve bu algoritmalardan KSO ile YAKA ayrıntılı şekilde açıklanmıştır.

Doğadaki birçok süreç, olay ya da model incelenerek değişik sürü zekâsı tabanlı yöntemler geliştirilip farklı problemler için etkili çözüm bulmak amacıyla kullanılabilir. Sezgisel algoritmaların her biri tüm problemler için en iyi sonucu vermeyebilir. Bir problem için iyi çözüm veren bir yöntem, başka bir problem için kötü olabilir. Bu yüzden algoritmaların iyi çözüm verdiği problem tipleri belirlenmelidir. Sonraki çalışmalarda, bu yöntemlerin sürekli değerli veriler içeren veritabanlarında bilgi keşfi amacıyla kullanılması planlanmaktadır.

7.Teşekkür

Bu proje Tunceli Üniversitesi Bilimsel Araştırma Projeleri Birimi (TÜNİBAP) tarafından YLTUB011-15 No'lu proje kapsamında desteklenmektedir.

8. Kaynaklar

1. Murty, K. G., Optimization Models For Decision Making, vol. 1, Internet Edition, Chapter 1: Models for Decision Making, 1-18, 2003.

2. Karaboğa, D., Yapay Zekâ Optimizasyon Algoritmaları, Nobel Yayın Dağıtım, 2011.

3. Alataş, B., Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları Geliştirme, Doktora Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, 2007.

4. X. S. Yang, Firefly Algorithms Formultimodal Optimization, Proceedings of the Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computing Sciences, vol. 5792, 178-178, Springer, Sapporo, Japan, 2009.

5. X. S. Yang, Firefly Algorithm, Levy Flights and Global Optimization, Research and Development in Intelligent Systems, XXVI, 209-218, Springer, London, UK, 2010.

6. Apostolopoulos, T., Vlachos, A., Application of the Firefly Algorithm for Solving the Economic Emissions Load Dispatch Problem, Hindawi Publishing Corporation International Conference Journal of Combinatorics, vol. 2011, doi:10.1155/2011/523806, 2011.

7. Krishnanand, K.N., Ghose, D. Detection of Multiple Source Locations Using a Glowworm Metaphor with Applications to Collective Robotics, IEEE Swarm Intelligence Symposium, 84-91, 2005.

8. Krishnanand, K.N., Ghose, D., Glowworm Swarm Optimisation: a New Method for Optimisin Multi-Modal Functions, International Journal of Computational Intelligence Studies, vol. 1, no. 1., 2009.

9. Dorigo M., Maniezzo, V., Colorni, A., The Ant System: An Autocatalytic Optimizing Process. Tech. Rep. No. 91-016, Dipartimento di Elettronica, Politecnico di Milano, Italy, 1991.

10. Kennedy, J., Eberhart, R. C, Particle Swarm Optimization. IEEE International Conference on Neural Networks, vol. IV, 1942-1948, Piscataway, NJ, 1995

11. Jiang, M., Yuan, D., Cheng, Y., Improved Artificial Fish Swarm Algorithm, Fifth International Conference on Natural Computation, 281-285, 2009.

12. Başbuğ, S., Bakteriyel Besin Arama Algoritması ile Lineer Anten Dizilerinin Diyagram Sıfırlaması, Yüksek Lisans Tezi, Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, 2008.

13. Liu, C., Yan, X., Liu, C., Wu, H., The Wolf Colony Algorithm and Its Application, Chinese Journal of Electronics, vol. 20, no. 2, 2011.

14. Chu, S. C., P. W. Tsai, J. S. Pan, Cat Swarm Optimization, 9th Pacific Rim International Conference on Artificial Intelligence, LNAI, vol. 4099, 854-858, 2006.

15. Chu, S. C., Tsai, P. W., Computational Intelligence Based on The Behavior of Cats, International Journal of Innovative Computing, Information and Control, vol. 3, 163-173, 2007.

16. Santosa, B.; Ningrum, M.K., Cat Swarm Optimization for Clustering, International Conference of Soft Computing and Pattern Recognition, 54-59, 2009.

17. Wang, Z. H., Chang, C. C., Li, M. C., Optimizing Least-Significant-Bit Substitution Using Cat Swarm Optimization Strategy, Information Sciences, 10.1016/j.ins.2010.07.011, 2010.

18. Hwang, J. C., Chen, J. C., Pan, J. S., Huang, Y.C., CSO and PSO to Solve Optimal Contract Capacity for High Tension Customers, International Conference on Power Electronics and Drive Systems, 246-251, 2009.

19. Tsai, P. W., Pan, J. S., Chen, S.M., Liao, B.Y., Hao, S.P., Parallel Cat Swarm Optimization, International Conference on Machine Learning and Cybernetics, vol. 6, 3328-3333, 2008.

20. Lin, K. C., Chien, H. Y., CSO-Based Feature Selection and Parameter Optimization for Support Vector Machine, Joint Conferences on Pervasive Computing, 783-788, 3-5, 2009.

21. Kalaiselvan, G., Lavanya, A., Natrajan, V., Enhancing the Performance of Watermarking Based on Cat Swarm Optimization Method, International Conference on Recent Trends in Information Technology, 1081-1086, 2011.

22. Wang, W., Wu, J., Emotion Recognition Based on CSO&SVM in E-Learning, Seventh International Conference on Natural Computation, vol. 1, 566-570, 2011.

23. Karaboga, D., Akay, B., A Survey: Algorithms Simulating Bee Swarm Intelligence, Artificial Intelligence Review, vol. 31 no., 1-4, 61-85, 2009.

24. Tereshko, V., Reaction-Diffusion Model of a Honey Bee Colony's Foraging Behaviour, 6th International Conference on Parallel Problem Solving from Nature, 807-816, 3-540-41056-2, Springer-Verlag, London, UK, 2000.

25. Hedayatzadeh, R., Hasanizadeh, B., Akbari, R., Ziarati, K., A Multi-Objective Artificial Bee Colony for Optimizing Multi-Objective Problems, 3rd International Conference on Advanced Computer Theory and Engineering, vol. 5, V5-277-V5-281, 2010.

26. Banharnsakun, A., Achalakul, T., Sirinaovakul, B., Artificial Bee Colony Algorithm on Distributed Environments, Second World Congress on Nature and Biologically Inspired Computing, 13-18, 2010.

27. Alam, M. S., Ul Kabir, M. W., Islam, M. M., Self-Adaptation of Mutation Step Size in Artificial Bee Colony Algorithm for Continuous Function Optimization, 13th International Conference on Computer and Information Technology, 69-74, 2010.

28. Karaboga, D., Gorkemli, B., A Combinatorial Artificial Bee Colony Algorithm for Traveling Salesman Problem, International Symposium on Innovations in Intelligent Systems and Applications, 50-53, 2011.

29. Guo, P., Cheng, W., Liang, J., Global Artificial Bee Colony Search Algorithm for Numerical Function Optimization, Seventh International Conference on Natural Computation, vol. 3, 1280-1283, 2011.

30. Ning, A., Zhang, X., A Speech Recognition System Based on Fuzzy Neural Network Trained by Artificial Bee Colony Algorithm, International Conference on Electronics, Communications and Control, 2488-2491, 9-11 2011.

31. Jadhav, H. T., Patel, J., Sharma, U., Roy, R., An Elitist Artificial Bee Colony Algorithm for Combined Economic Emission Dispatch Incorporating Wind Power, 2nd International Conference on Computer and Communication Technology, 640-645, 2011.

32. Çelik, M., Karaboğa, D., Köylü, F., Artificial Bee Colony Data Miner (ABC-Miner), Innovations in Intelligent Systems and Applications (INISTA), 96-100, 2011.