

Sosyal Ağ Analizi İçin Sosyal Tabanlı Yapay Zekâ Optimizasyon Algoritmalarının İncelenmesi

Feyza Altunbey, Bilal Alataş*

Yazılım Mühendisliği Bölümü, Fırat Üniversitesi, 23119, Elazığ

*Yazışmalardan sorumlu yazar: E-mail: balatas@firat.edu.tr

Özet

Bu makalede, genel amaçlı arama ve optimizasyon tekniklerinden sosyal tabanlı güncel yapay zekâ optimizasyon algoritmaları, ilk kez bir bütün olarak tanıtılmış ve sosyal ağ analizinde bağlantı tahmini, topluluk keşfi, duygu analizi, metin özetleme vb. alanlarda etkili olarak dolaylı ya da direkt çözüm algoritması olarak kullanılabilirliği belirtilmiştir.

Anahtar Kelimeler: Optimizasyon, Sosyal ağ analizi, Yapay zekâ

Review of Social-Based Artificial Intelligence Optimization Algorithms for Social Network Analysis

Abstract

In this paper, current social-based artificial intelligence optimization techniques that are general-purposed search and optimization techniques have been introduced for the first time as a whole and it has been shown that these techniques can be efficiently used as direct or indirect search algorithm within social network analysis such as link prediction, community discovery, sentimental analysis, text summarization, and etc.

Keywords: Optimization, Social network analysis, Artificial intelligence

GİRİŞ

Bireyler arasında kişisel ya da profesyonel ilişkilerin oluşturduğu ağ olarak adlandırılabilir olan sosyal ağlar, günlük yaşam tarzının önemli bir parçası haline gelmiş ve gittikçe önem kazanmaya başlamıştır. Kullanıcılar, sosyal ağlar üzerinden yeni iletişim ortamlarının sunduğu hemen hemen tüm özelliklerden yararlanmakta, sürekli bilgi paylaşımı sağlamaktadırlar. Bilgi değişiminin bu kadar yaygın hale gelmesi her geçen gün yeni sosyal ağların doğmasına yol açmaktadır.

Sosyal ağlar eğitim, akademik, sağlık ve daha birçok alanda insanlar tarafından aktif bir şekilde kullanılmaktadır. Sosyal ağ kavramının bu kadar çok kullanılması sosyal ağları

oluşturan yapıların incelenip analiz edilmesi konusunun beraberinde getirmiştir. Ağları oluşturan düğümlerin ya da elemanların ve bunlar arasındaki ilişki ya da etkileşimlerin incelenmesi sosyal ağ analizi konusunu oluşturmaktadır. İncelenen sosyal ağın analiz edilmesi, sözü edilen ağ yapısını kullanan bireyler için kullanışlı bilgilerin çıkarılmasını sağlamaktadır.

Sosyal ağ analizi konusunda son yıllarda en çok ilgilenilen konu bağlantı tahminidir. Bağlantı tahmini, analiz edilen ağda gelecekte olması muhtemel bağlantıları veya var olan gizli bağlantıları tespit etme işlemidir. Verilen sosyal ağda bulunan bağlantılar incelenip belirli ölçütlere göre değerlendirilerek aralarında bağlantı

Derleme/Review

olmayan herhangi iki düğüm arasında oluşabilecek bağlantılar tahmin edilir. Bağlantı tahmini moleküler biyoloji, kriminal keşifler ve tavsiye edici sistemler gibi birçok farklı alanda kullanılmaktadır (Catherine ve ark., 2014).

Sosyal ağ analizi ile ilgili diğer güncel konu ise ağlardaki topluluklar ve toplulukların keşfedilmesidir. Bir ağdaki topluluklar, gruplar arasındaki bağlantıların seyrek ve grup içindeki bağlantıların ise sık olduğu düğüm grupları olarak tanımlanır. Bir başka tanım ile topluluk, sıklıkla iletişim halinde olan bireylerin birlikteliğidir. Bu nedenle topluluklar çoğunlukla ortak özellikler paylaşan ve etkileşim içinde benzer roller oynayan düğümlerin gruplarıdır (Coscia ve ark., 2011). Ağ yapıları içerisindeki topluluklar bize bireylerin ortak ilgi alanları, çalışma konuları, eğilimleri, benzerlikleri vb. hakkında somut bir fikir sunmaktadır. Gerçek ağlarda ağ yapısı homojen değildir. Belirli bir alanda yoğunlaşan, kümeleşen ve topluluk olarak adlandırdığımız yapılar muhtemelen aynı özelliği paylaşan ve/veya benzer rolü bulunan düğüm kümeleridir (Fortunato, 2010).

Bir diğer önemli sosyal ağ analizi konusu ise fikir madenciliği ya da duygu sınıflandırma olarak ta bilinen yazılı halde verilen bir ifadenin (belge, yorum, e-posta vb.) yansıttığı duygunun bilgisayar yardımıyla otomatik olarak tespit edilmesi işi olan duygu analizidir. Tespit edilmesi hedeflenen duygu, yazarın ruh halini, konu hakkındaki düşüncesini, yapmak istediği vurguyu ya da oluşturmak istediği etkiyi içerebilmektedir (Ucan, 2014).

Sosyal ağlarda metin özetleme, metin sınıflandırma vb. konular da aktif olarak bilim adamları tarafından çalışılmaktadır. Bu makalede de genel

amaçlı arama ve optimizasyon algoritması olan ve sosyal ağ analizi konularına direkt ya da dolaylı çözüm algoritması olarak kullanılabilir sosyal tabanlı güncel yapay zekâ optimizasyon algoritmaları tanıtılmıştır.

YAPAY ZEKÂ OPTİMİZASYON ALGORİTMALARI

Belirli sınırlamaları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilir. Optimizasyon ise en iyileme anlamına gelmektedir. Bir problem için, verilen şartlar altında tüm çözümler arasından en iyi çözümü elde etme işidir. Yani sezgisel algoritmalar yakınsama özelliğine sahiptir, fakat kesin çözümü garanti edememektedir ve bu kesin çözümün yakınlarında bir çözüm vermektedir.

Sezgisel algoritmalara ihtiyaç duyulmasının nedeni:

- Optimizasyon problemi kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olabilir.
- Anlaşılabilirlik açısından sezgisel algoritmalar karar verici açısından çok daha basit olabilir.
- Sezgisel algoritmalar, öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir.
- Matematik formülleriyle yapılan tanımlamalarda genellikle gerçek dünya problemlerinin en zor tarafları (hangi amaçlar ve hangi sınırlamalar kullanılmalı, hangi alternatifler test edilmeli, problem verisi nasıl toplanmalı) ihmal edilir. Model parametrelerini belirleme aşamasında kullanılan verinin hatalı olması, sezgisel yaklaşımın üretebileceği alt en uygun

Derleme/Review

çözümünden daha büyük hatalara sebep olabilir (Karaboga, 2011).

Yapay zekâ optimizasyon algoritmaları ise günlük hayatta sıklıkla kullanılan sezgisel optimizasyon algoritmaları üzerinde çalışan bir karar mekanizmasıdır. Yapay zekâ optimizasyon algoritmaları basit bir yaklaşımı, arama ya da optimizasyon problemlerinin çözüm tekniği olarak kullanan algoritmalar ve son yıllarda gittikçe güçlenmekte ve her geçen gün biraz daha popüler olmaktadır. Bunların nedeni şu şekilde özetlenebilir:

- Eş zamanlı olarak, farklı tipte karar değişkenleri, amaç fonksiyonları ve sınırlayıcıların olması durumunda probleme uygulanabilecek genel çözüm stratejileri sunmaktadırlar. Çözüm stratejileri amaç fonksiyonu ve sınırlayıcıların tipine ve problemi modellemede kullanılan değişkenlerin tipine bağlı değildir.
- Çözüm uzayı tipine, karar değişken sayısına ve sınırlayıcı sayısına bağlı değildir.
- Sistemin modeli ve amaç fonksiyonu için kurulması zor olan ve bazen de kurulup ta çözüm zamanı maliyeti çok yüksek olduğundan kullanılmayan çok iyi tanımlanmış matematiksel modellere ihtiyaç duymamaktadır.
- Hesaplama güçleri iyidir, yani aşırı derecede hesaplama zamanına ihtiyaç duymazlar.
- Dönüşümleri ve uyarlanmaları kolaydır.
- Büyük ölçekli kombinasyonel ve doğrusal olmayan problemlerde etkili sonuçlar vermektedir.
- Klasik algoritmalarındaki gibi verilen bir probleme bir çözüm algoritması uyarlamada geçerliliğinin onaylanması zor olabilen bazı varsayımları gerektirmemektedir.
- Klasik algoritmalarındaki gibi ilgilenilen problem üzerinde değişiklik

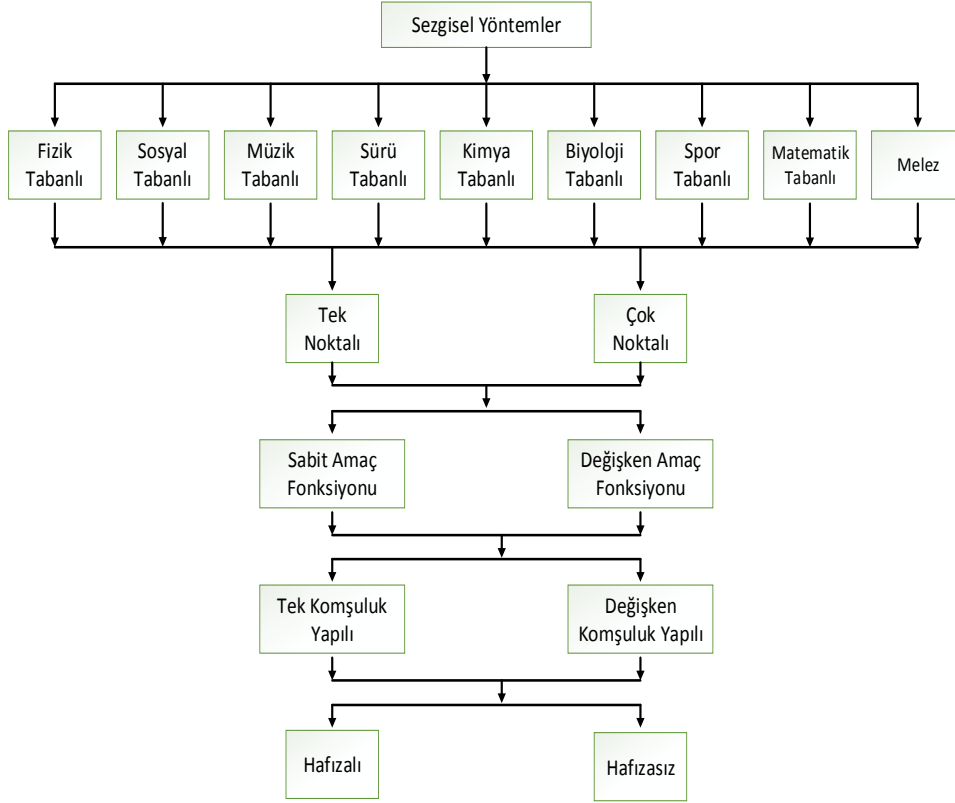
gerektirmemektedir. Farklı türdeki problemleri çözmek için kendilerini uyarlarlar.

Bu avantajlarından dolayı yapay zekâ optimizasyon algoritmaları; yönetim bilimi, mühendislik, bilgisayar gibi birçok farklı alanda yoğun olarak kullanılmakta ve yeni versiyonları önerilmektedir.

Genel amaçlı yapay zekâ optimizasyon algoritmaları Şekil 1’de gösterildiği gibi, biyolojik tabanlı (evrimsel algoritmalar, karınca koloni algoritması, arı koloni algoritması, yapay bağışıklık algoritmaları, ateşböceği algoritması, enzim algoritması, fidan gelişim algoritması, istilacı yabancı ot optimizasyonu, maymun arama algoritması, bakteriyel yem arama algoritması), fizik tabanlı (çok noktalı ısıl işlem algoritması, elektromanyetizma algoritması, parçacık çarpışma algoritması, büyük patlama - büyük çökme algoritması), sürü tabanlı (parçacık sürü optimizasyonu, karınca koloni optimizasyonu, arı koloni optimizasyonu), sosyal tabanlı (çok noktalı tabu araştırma algoritması, emperyalist yarışmacı algoritma, parlamenter optimizasyon algoritması), müzik tabanlı (armoni arama), spor tabanlı (lig şampiyonası algoritması), kimya tabanlı yöntemler (yapay kimyasal reaksiyon optimizasyon algoritması), matematik tabanlı (matheuristic ve baz algoritması) olmak üzere sekiz farklı grupta değerlendirilmektedir. Ayrıca bunların birleşimi olan melez yöntemler de bulunmaktadır.

Her ne kadar literatüre kazandırılmış çok başarılı algoritmalar ve teknikler geliştirilmiş olsa da; bilimsel alanda sürekli iyileşme ve daima daha iyiyi arama felsefesi altında yeni tekniklerin tasarlanması,

Derleme/Review



Şekil 1. Yapay zekâ optimizasyon algoritmaları

geliştirilmesi ve uygulanması önemli bir görevdir. Ayrıca tüm problemler için en iyi sonucu veren algoritma henüz tasarlanmadığından sürekli, yeni yapay zekâ optimizasyon algoritmaları önerilmekte ya da var olanların daha etkili çalışması için önerilen sunulmaktadır. Son yıllarda bu bilinçle araştırmacılar, yeni yapay zekâ optimizasyon algoritmalarını başarılı bir şekilde literatüre kazandırmışlar ve başarılı uygulamalar gerçekleştirmişlerdir.

SOSYAL TABANLI YAPAY ZEKÂ OPTİMİZASYON ALGORİTMALARI

Literatürde yeni önerilen birçok sosyal tabanlı sezgisel optimizasyon algoritması bulunmaktadır. Bunlardan en bilineni ve uygulaması en çok yapılanı tabu arama algoritmasıdır. Yakın zamanda önerilenler ise Şekil 2’de listelenmiştir. Alt başlıklarda güncel olan bu sosyal tabanlı yapay zekâ algoritmaları tanıtılacaktır.



Şekil 2. Sosyal tabanlı yapay zekâ optimizasyon algoritmaları

Derleme/Review

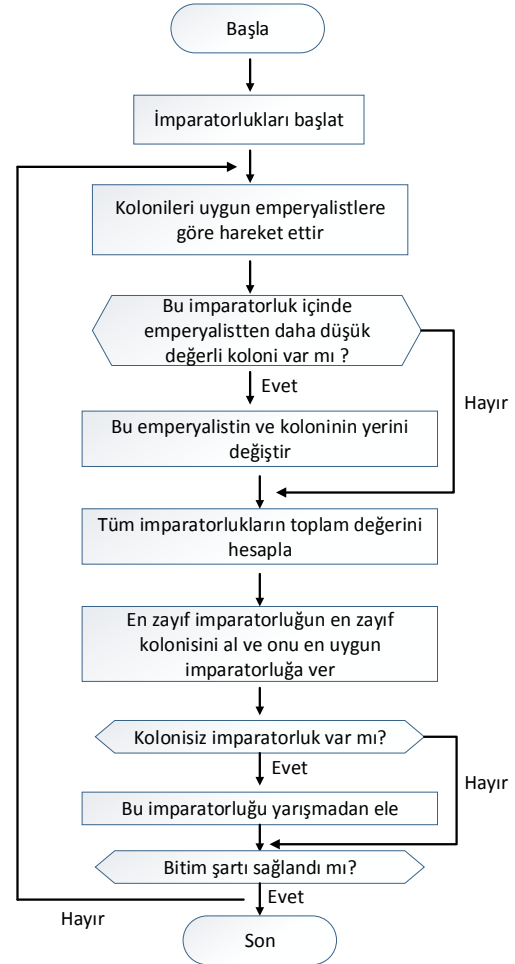
Emperyalist Yarışmacı Algoritma

Benzer evrimsel algoritmalar gibi Emperyalist Yarışmacı Algoritma (EYA) da bir başlangıç popülasyonunun oluşturulması ile algoritmaya başlar. Başlangıç popülasyonundaki en iyi ülkelerin birkaçı emperyalist olmak için seçilir ve kalan bireyler emperyalistlerin kolonisi olur. Belirlenen kolonilerin hepsi emperyalist devletler arasında dağıtılır. Kolonilerin emperyalist devletler arasında dağıtılmasından sonra, koloniler uygun emperyalistlere doğru yönelmeye başlar. İmparatorlukların güçleri emperyalistin ve emperyaliste verilen kolonilerinin gücüne bağlıdır. Emperyalistler arasında başlayan yarış ile algoritma süreci devam eder. Gücünü artıramayan ya da başarılı olamayan emperyalist yarıştan elenir. Yarış sırasında güçlü imparatorluklar var olan güçlerini artırırken zayıf imparatorlukların gücü azalır ve imparatorluk yıkılmaya doğru gider. Yarış tek bir imparatorluk kalana kadar devam eder ve algoritma sonucunda diğer ülkeler kalan imparatorluğun bir kolonisi haline gelir. Yarış sonunda oluşan ideal dünyada koloniler ve emperyalistler aynı konuma ve güce sahip olacaktır (Atashpaz-Gargari ve Lucas, 2007). Şekil 3'te algoritmanın akış şeması görülmektedir.

Öğretme Öğrenme Tabanlı Optimizasyon Algoritması

Son zamanlarda geliştirilen metasezgisel optimizasyon algoritmalarından bir diğeri ise Öğretme Öğrenme Tabanlı Optimizasyon (ÖÖTO) Algoritmasıdır (Rao ve Patel, 2007). ÖÖTO, bir sınıfta öğretmenin öğrenciler üzerindeki etkisine göre çalışan bir algoritmadır. Algoritma bir sınıftaki öğretmen ve öğrencilerin, öğretme ve öğrenme yeteneklerini

tanımlar. Öğretmen ve öğrenci bu algoritmanın iki önemli bileşenidir.



Şekil 3. EYA'nın akış şeması

Algoritmada öğrencilerin olduğu grup popülasyon olarak kabul edilir ve öğrencilere sunulan farklı konular ise optimizasyon probleminin farklı tasarım değişkenleri olarak kabul edilir. Bir öğrencinin sonucu optimizasyon probleminin uygunluk değerine benzerdir. Tüm popülasyon içinde en iyi çözüm öğretmen olarak kabul edilir. Tasarım değişkeni olarak kullanılan terimler verilen optimizasyon probleminin uygunluk fonksiyonunda yer alan parametre olarak gösterilir ve en iyi çözüm uygunluk fonksiyonunun en iyi değeridir. ÖÖTO algoritmasının

Derleme/Review

çalışma süreci Öğretme Süreci ve Öğrenme Süreci olmak üzere iki durumdan oluşmaktadır.

Öğretme Süreci'nde öğretmen genellikle öğrencilerle bilgilerini paylaşan kişi olarak kabul edilir ve bu süreçte çok önemlidir. Bir öğretmenin kalitesi sonucunu öğrencilerin üzerinde gösterir. İyi öğretmene sahip öğrencilerin notlarında ve durumlarında gelişmeler olduğu gözlenmektedir. Bu yüzden Öğretme Süreci öğretmen ve öğrenci arasındaki ilişkilere bağlıdır. Öğrenme Süreci'nde ise ana faktör öğrencilerdir.

Algoritmanın adımlarının daha iyi anlaşılması açısından ÖÖTO algoritması için Şekil 4'te verilen akış şeması oluşturulmuştur.

Birçok optimizasyon yöntemi, algoritmanın performansını etkileyen algoritma parametreleri gerektirir. Genetik algoritma çaprazlama olasılığı, mutasyon oranı ve seçim yöntemi gerektirir; parçacık sürü optimizasyonu öğrenme faktörleri, ağırlık değişimini ve hızın maksimum değerini gerektirir; yapay arı kolonisi bir takım işçi arı, seyirci arı ve limit değerini gerektirir. Armoni arama, uyum bellek dikkat oranını, açıklık ayar oranını ve doğaçlama sayısını gerektirir. Kurbağa algoritması bir takım mem topluluğu ve her mem için iterasyon gerektirir. Karınca koloni optimizasyonu üs parametreleri, feromon buharlaşma oranı ve ödül faktörü gerektirir. Diğer optimizasyon teknikleri aksine ÖÖTO uygulama ayarlarının yapılması için herhangi bir algoritma parametresi gerektirmez, böylece ÖÖTO gerçekleştirimi daha kolaydır.

ÖÖTO yapay arı kolonisi ve kurbağa algoritması gibi popülasyonu bölmez. Seçimi, çaprazlamayı ve mutasyon aşamalarını kullanan genetik algoritma gibi ve izci, seyirci ve işçi

arılar aşamalarını kullanan yapay arı kolonisi gibi ÖÖTO iki farklı aşama kullanır. 'Öğretmen Aşaması' ve 'Öğrenci Aşaması'. ÖÖTO çözümü güncellemek için ortalama değeri kullanır. ÖÖTO, yapay arı kolonisi gibi iyi çözümü kabul etmek için açgözlülüğü uygular. ÖÖTO yönteminin gücü algoritmanın çalışması için herhangi bir parametre ayarı gerektirmemesidir.

Sosyal Duygusal Optimizasyon Algoritması

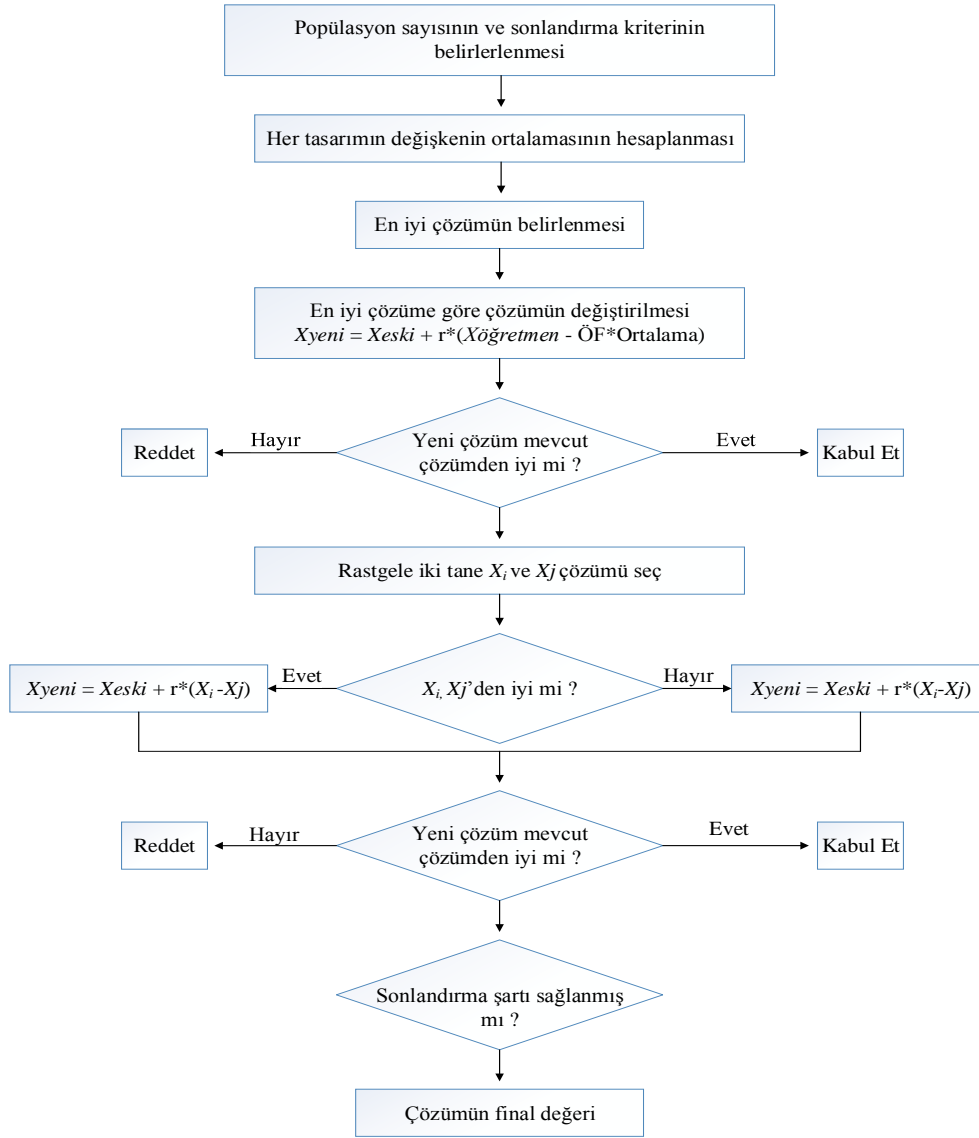
Sosyal Duygusal Optimizasyon Algoritması (SDOA), insan davranışlarının benzerini yapan yeni bir sosyal tabanlı optimizasyon tekniğidir (Cui ve ark., 2010). Sosyal kelimesi insan topluluğu ile ilişkilidir. Toplumda yaşayan insanlar sosyal statülerini artırmaya çalışmaktadır.

SDOA'da her bir birey sanal bir kişiyi temsil eder. Her bir adımda, bireyler davranışlarını ilişkili duygusal indekse göre belirler. Duygusal indeks düşük, orta ve yüksek olmak üzere üçe ayrılır. Duygusal indekse göre, bir davranış seçilir ve daha sonra seçilen davranışa göre durum değeri, seçilen davranışın doğru olup olmadığına bağlı olarak toplumdan geri dönüşüm alınır. Eğer bu seçim sosyal durum değerini artırırorsa, bireyin duygusal indeksi artar. Aksi durumda sosyal durum değerini azaltmak için duygusal indeks azalır (Devi ve Singla, 2012).

SDOA'nın işlem adımları Tablo 1'de verilmiştir. Akış diyagramı ise Şekil 5'tedir.

Beyin Fırtınası Optimizasyonu

Beyin fırtınası yaratıcı düşünmeyi kolaylaştırmak gibi geniş olarak kabul edilen organizasyonlarda yaratıcılığı artırmak için yaygın olarak kullanılan bir araçtır. Beyin fırtınası ilk



Şekil 4. ÖÖTO algoritmasının akış şeması

olarak 1939'da Osborn tarafından reklam firmasında geliştirilmiştir. 1957 sonlarında ise bu problem çözme metodunu Applied Imagination isimli kitabında sistemleştirmiştir (Osborn, 1957; Zhan ve ark., 2012).

Bundan sonra beyin fırtınası, tüm dünyada hem akademik hem sanayi alanında büyük ilgi uyandırmıştır. Beyin fırtınası süreci, bir problemin çözümü için büyük fikirler üretmek amacıyla işbirliği içinde bulunacak, etkileşim kuracak farklı etnik kökene

sahip insanları bir araya getirmektir. Beyin fırtınası süreci, grup üyeleri arasındaki sosyal engelleri azaltmak, fikir üretimine teşvik etmek ve grubun genel yaratıcılığını artırmak için Osborn'un dört kuralına uyar.

Bu dört kural şu şekildedir:

1. Kural: Ne kadar fazla fikir, o kadar daha iyi.
2. Kural: Herhangi bir kural için eleştiriyi engelleme.
3. Kural: Sıra dışı fikirleri hoş karşılama.

Derleme/Review

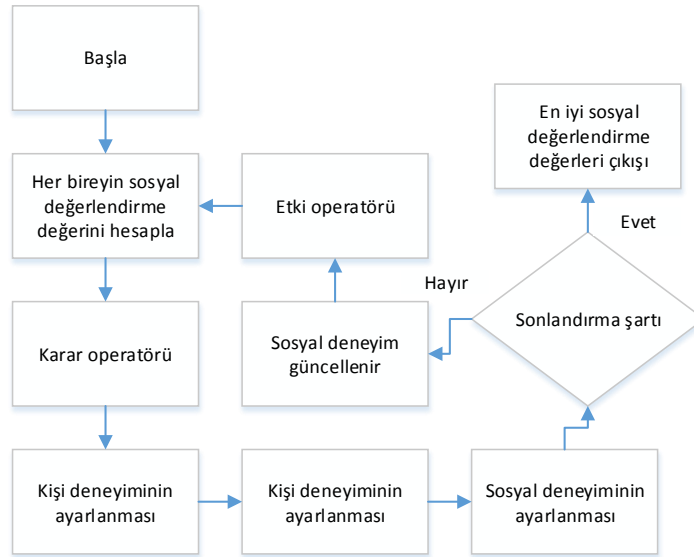
4. Kural: Fikirleri birleştir ve geliştir.

Tablo 1. SDOA'nın işlem adımları

<ol style="list-style-type: none"> 1. Tüm bireyler sırasıyla oluşturulur ve bireylerin başlangıç konumları problem uzayına rastgele dağıtılır. 2. Amaç fonksiyonuna göre her bir bireyin uygunluk değeri hesaplanır. 3. j. birey için, duygusal indeksine göre davranış hareketleri belirlenir. 4. Tüm popülasyon için konum güncellenir. 5. Duygusal indeks belirlenir. 6. Sonlandırma şartı sağlanmışsa en iyi sonuç çözüm kabul edilir, eğer şart sağlanmamışsa 2. adıma tekrar dönlür.

Beyin Fırtınası Optimizasyonu (BFOA) sezgisel olarak, insanların fikir üretme sürecine dayalı olan ve hayvanların işbirlikçi davranışlarından esinlenerek ortaya konulan bir optimizasyon algoritması olarak tasarlanmıştır. Beyin fırtınasından esinlenerek geliştirilen BFOA işlem adımları Tablo 2'de verilmiştir.

BFOA'da popülasyon büyüklüğü genellikle bütün iterasyonlar için kolaylık açısından sürekli sabittir.



Şekil 5. SDOA'nın akış diyagramı

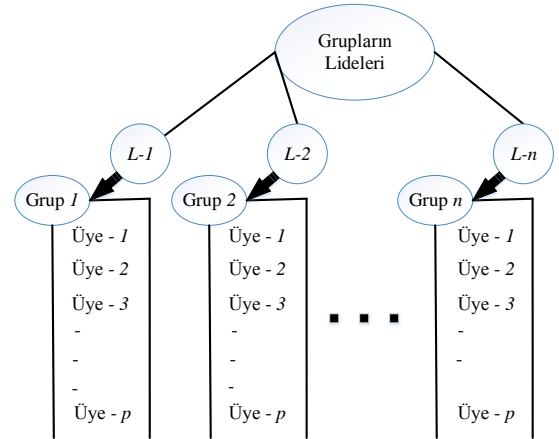
Grup Liderleri Optimizasyon Algoritması

Grup Liderleri Optimizasyon Algoritması (GLOA), sosyal gruplarda liderlerin etkilerinden esinlenerek geliştirilen evrimsel bir algoritmadır. Problem uzayı farklı gruplara bölünür ve her bir grubun kendi lideri oluşturulur. Her bir grubun üyelerinin benzer karakterlerde olmalarına gerek yoktur, rastgele oluşturulabilirler. Her

grubun en iyisi lider olarak seçilir. Her grubun üyeleri her bir iterasyonda liderlerine benzemeye çalışırlar. Bu şekilde, algoritma lider ve grup üyeleri arasında çözüm alanı oluşturur. Bazı adımlardan sonra grup üyelerinin liderlere benzediği görülmüştür. Grup içindeki çeşitliliği artırmak için üyelerden biri rastgele seçilir ve bazı değişkenleri diğer gruptaki üyelerin değişkenleri ile değiştirilir. Buna ek

Derleme/Review

olarak, bir çaprazlama operatörü grubun yerel minimuma ulaşması için yardım eder ve çözüm uzayı çeşitliliği artırmak için tekrar aranabilir (Pooranian ve ark., 2013). Algoritmanın sözde kodu Tablo 3'te verilmiştir. P adet üyeden oluşan n adet grubun oluşturulduğu ve uygunluk değerlerine göre grup liderlerinin belirlendiği algoritma ilk üç adımı Şekil 6'da verilmiştir. Adım 4'te yapılanlar Tablo 4'te, Adım 5'te yapılanlar ise Tablo 5'te gösterilmiştir. Burada r_3 ve r_2 'nin 0.5'ten küçük ve r_1 'in 0.5'ten büyük seçilmesinin algoritmayı daha stabil yaptığı gösterilmiş ve bu parametrelerin etkisinin minimize ettiği belirtilmiştir.



Şekil 6. GLOA'nın başlangıç adımı

Tablo 2. BFOA'nın işlem adımları

1. n potansiyel çözüm (birey) oluşturulur.
2. n tane birey m kümeye bölünür.
3. n birey değerlendirilir.
4. Her bir kümedeki bireyler sıralanır ve en iyi birey kümenin merkezi olarak belirlenir.
5. 0 ile 1 arasında rastgele a değeri üretilir.
 - a) Üretilen a değeri önceden belirlenen $P5a$ değerinden küçükse
 - i. Rastgele bir a küme merkezi seç.
 - ii. Seçilen küme merkezi ile değiştirmek için rastgele bir birey üret.
6. Yeni bireyler üret.
 - a) 0 ile 1 arasında rastgele a değeri üretilir.
 - b) Üretilen değer $P6b$ değerinden küçük ise,
 - i. $P6i$ olasılıkla rastgele a kümesi seç.
 - ii. 0 ile 1 arasında rastgele bir a değeri üret.
 - iii. Değer önceden belirlenen $P6biii$ değerinden küçükse,
 - 1) Küme merkezi seç ve yeni bireyler üretmek için rastgele değeri ekle.
 - iv. Aksi durumda kümeden rastgele bir birey seç ve yeni bireyler elde etmek için bu bireye rastgele üretilen değeri ekle.
 - c) Aksi durumda yeni bireyler üretmek için rastgele iki küme seç.
 - i. Rastgele bir değer üret.
 - ii. Üretilen değer önceden belirlenen $P6c$ olasılığından küçükse, iki küme merkezi seç ve birleştir ve yeni bireyler üretmek için rastgele üretilen değeri ekle.
 - iii. Aksi durumda, seçilen her bir kümeden birleştirmek için rastgele iki birey seçilir ve yeni bireyler üretmek için üretilen değer eklenir.
7. Eğer n yeni birey üretilirse 8. adıma git, aksi durumda 6. Adıma git.
8. Önceden belirlenen maksimum iterasyon sayısına ulaşılmışsa sonlandır, aksi durumda 2. adıma git.

Tablo 3. GLOA'nın sözde kodu

1. Her grup için p sayıda popülasyon üret
2. Tüm grup üyelerinin uygunluklarını hesapla
3. Her grubun liderini belirle
4. Mutasyon ve rekombinasyon uygula
5. Diğer gruplardan parametre transferi.
6. Sonlandırma şartı sağlanıncaya kadar 3-5 arası adımları tekrar et.

Hiyerarşik Sosyal Algoritma

Hiyerarşik Sosyal Algoritma (HSA), insan örgütleri ya da biyolojik sistemlerin çeşitliliğinde gözlenen sosyal davranışlardan ilham almıştır. Bu metasezgisel yaklaşım sınırsız kaynaklara sahip DFG zamanlama (Fernandez, 2003a) ve kritik devre hesaplama (Fernandez, 2003b) gibi birkaç probleme başarı ile uygulanmıştır. HSA'nın temel fikri uygun çözümler kümesinin eşzamanlı optimizasyonudur. Toplumun her bir grubu uygulanabilir bir çözüm içerir ve bu gruplar başlangıçta çözüm alanlarının ayrı parçalarını üretmek için rastgele dağıtılmıştır. Gelişim stratejilerini kullanarak her grup kendi amaç fonksiyonunu artırmak ya da komşularıyla yarışmak için çalışır. Bu durumda; ilgili sosyal rekabet, işbirliği ve toplumsal gelişim yolu ile daha iyi bir çözüm elde edilir. Bu toplumsal gelişimde daha düşük kaliteli gruplar yok olma eğilimindedir. Böylelikle nesnel çözüm optimize edilir. İşlem süreci bulunan en iyi çözümü içeren tek bir grup ile biter (Duarte ve ark., 2006).

Bir toplumun durumu hiyerarşik politika grafiği tarafından modellenir. Bu hiyerarşik politika grafiği aynı zamanda her bir ayrık grup kümesi $\Pi = \{g_1, g_2, \dots, g_g\}$ tarafından birleştirilmiş toplum bölümünü belirtir, burada her bir birey veya düğüm bir gruba atanır. Her grup $g_i \in \Pi$, uygun toplum tarafından kısıtlı bir dizi bireyler ve aktif ilişkiler

tarafından oluşturulur. Tüm grupların bireyleri bütün toplumun bireylerini kapsamaktadır. Hiyerarşik politika grafiğinin özellikleri probleme bağımlıdır. Başlangıç toplum bölümü, grupların keyfi sayısını belirler ve gruplara bireyleri atar. Bir toplum bölümü, tek grup ya da tekeli bölüm olarak ayrılabilir, bu toplum bölümünde sadece bir grup ve içinde birden fazla grup bulunan çoklu-grup veya rekabet bölümü vardır.

Tablo 4. GLOA'nın dördüncü adımının içeriği

```

for i=1 to n do {
    for j=1 to p do {
        yeniij= r1* uyeij + r2*Li +
        r3*random
        if uygunluk (yeniij),
        uygunluk (uyeij)'den daha iyi
        then
            uyeij = yeniij
        end if
    } end for
} end for

```

Bir toplumun her bireyinin kendine özgü amaç fonksiyonu f_1 vardır. Her grup, o grubun tüm bireyler tarafından paylaşılan grup amaç fonksiyonuna f_2 sahiptir. Ayrıca her grup g_i iki ayrık bölüme ayrılmıştır: Merkez ve çevre. Merkez, ilgili grup amaç fonksiyonunun f_2 değerini belirler ve çevre ise tutulan grubun yerel arama bölgesini tanımlar. Burada dikkate alınan HS algoritmaları, algoritmanın işletme aşamasına bağlı nesnel (f_1 veya f_2) işlevlerinden birini optimize etmeye çalışır. Kazanan strateji aşamasında her bir grup g_i , bağımsız grup amaç fonksiyonunu f_2 geliştirmeyi amaçlamaktadır. Kaybeden strateji aşamasında, her birey bireysel amaç fonksiyonunu f_1 geliştirmek için çalışır, orijinal grup bağlılığı kaybolur ve ilgili

Derleme/Review

bireysel amaç fonksiyonunu optimize etmek için grafik bölümü modifiye edilir.

Tablo 5. GLOA'nın beşinci adımının içeriği

```

for  $i = 1$  to  $n$ 
   $t = \text{random} * (\text{parametre} / 2 + 1)$ ;
  for  $j = 1$  to  $t$ 
     $x = \text{random} * n$ ;
     $k = \text{random} * p$ ;
     $pr = \text{random} * \text{parametre}$ ;
     $\text{yeni}_{ik} = \text{eleman}_{ik}$  ;
     $\text{yeni}_{ik}(pr) = \text{eleman}_{ik}(pr)$  ;
    if uygunluk( $\text{yeni}_{ik}$ ) uygunluk( $\text{eleman}_{ik}$ )
      )'den daha iyi
       $\text{eleman}_{ik} = \text{yeni}_{ik}$ ;
    end if
  end for
end for

```

Algoritma rastgele seçilmiş uygulanabilir çözümler kümesinden başlar. Başka bir deyişle buna karşılık gelen grup yapısını tanımlayan bir başlangıç bölmesini elde eder. Buna ek olarak, her bir grup için, bir rastgele başlangıç değeri elde edilir. Gruplar, yerel arama stratejilerine dayalı bir dizi rekabet aşamasıyla ardışık olarak dönüştürülür. Her bir grup için, iki ana strateji vardır: Kazanan stratejisi ve kaybeden stratejisi.

Kazanan stratejisini uygulayan 'kazanan gruplar' adlı gruplar, daha yüksek grup amaç fonksiyonuna (f_2) sahip olan gruptur. Geriye kalan gruplar kaybeden stratejisini uygular ve 'kaybeden gruplar' olarak adlandırılır. Kazanan ve kaybeden stratejileri hiyerarşik sosyal algoritmanın temel arama araçlarıdır. Yarışma aşamaları arasında isteğe bağlı özerk aşamalar süresince, tüm gruplar kazanan gruplar gibi davranırlar. Bu isteğe bağlı özerk

fazlar arama prosedürü yeteneğini geliştirirler.

Kazanan strateji, bir yerel arama prosedürü olarak kabul edilebilir ki, her bir grup içinde bulunan çözüm kalitesi, her bir gruba ait bireylerin ve bu bireyler arasındaki ilişkilerin bağımsız olarak çalışması ile geliştirilmektedir.

Kaybeden stratejisi, gruplar arasında bireylerin değişimini sağlamak için yönlendirilmiştir. Bu şekilde, düşük kaliteli (düşük grup amaç fonksiyonu f_2) gruplar yok olma eğilimindedir, çünkü onların bireyleri bu gruplardan başka gruplara taşınarak azami birey promosyonuna izin verir (en yüksek bireysel amaç fonksiyon f_1)

Daha düşük grup amaç fonksiyonuna sahip olan kaybeden grup bireyleri, kendi birey amaç fonksiyonlarını geliştirmek için kaybeden stratejisi süresince grup değiştirirler. Bu şekilde, kaybeden gruplar ilgili evrim sürecinde yok olma eğilimindedir. En yüksek grup amaç fonksiyonuna sahip olan, kazanan grup bireyleri, grup amaç fonksiyonunu (f_2) geliştirmek amacıyla, merkezden çevreye veya ters yönde hareket edebilir.

Bu algoritmanın ana arama prosedürleri şunlardır: Kazanan Stratejisi ve Kaybeden Stratejisi. Belirli bir g_i grubu için, merkez ve çevre arasındaki düğüm değişimi, grup amaç fonksiyonunu optimize etmeyi sağlar. Kaybeden Stratejisi bireyleri başka bir gruba taşımak için izin verir. Kazanan strateji bir grubun f_2 fonksiyon değerini geliştirmek için yönlendirilmiştir. Daha yüksek fonksiyon değerli f_2 (daha düşük kalite) gruplara ait bireyler, bireysel amaç fonksiyonunu f_1 artırmak için gruplarını değiştirebilirler. İki düğüm arasında tek bir yinelemeye paralel ve eş zamanlı olarak düğüm değişimine izin verilir. Tek kısıtlama şudur: bir v

Derleme/Review

düğümü konum değiştiriyorsa ona komşu düğümlerin hiçbiri aynı tekrarda konumunu değiştiremez.

Bu kısıtlama, ilgili prosedürde düğüm döngüsünü önler (Bir döngüde sadece bir düğüm taşınır). Verilen bir g_i grubu için, eğer ilgili grup g_i nin amaç f_2 fonksiyonunda bir gelişme varsa merkez ve çevre arasındaki düğüm değişimi başarılıdır. Her düğüm kendi bireysel amaç fonksiyonundaki en iyi gelişimi veren grubu arar.

İnsan Grup Formasyon Algoritması

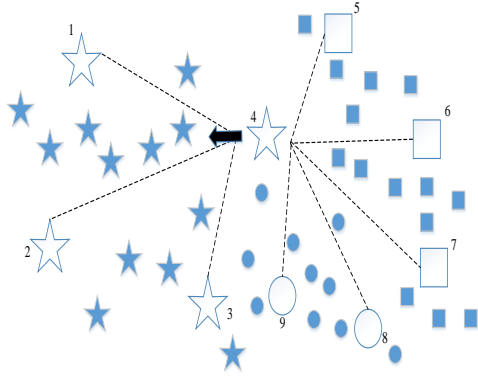
İnsan Grup Formasyon Algoritması (İGFA), kendi gruplarıyla olabildiğince birleşmeye çalışan grup içi üyeler ile, aynı zamanda grup dışı üyelerle sosyal korumaya çalışan grup dışı üyelerin davranışlarından esinlenerek geliştirilmiş sosyal tabanlı güncel bir yapay zekâ optimizasyon algoritmasıdır. Sosyologlar, insan sosyal kategorisini tanımlamak için grup içi ve grup dışı durumunu tanımlamışlardır. Grup içi üyeler, gruba ait olduğu grup tarafından kabul edilmiş bireylerdir. İnsanlar bir grubun üyesi olarak tanımlandığında, kendi gruplarını benimser ve grup üyelerini diğer gruplardan farklı olarak algırlar. Kendi gruplarını diğer gruplara üstün kabul ederler. Bu nedenle grup üyeleri gruptan uzaktayken bile mümkün olduğunca gruplarını birleştirmeye çalışırlar (Thammano ve Moolwong, 2010).

Sosyologlar insanların toplumsal sınıflandırılmasını açıklamak için, grup, grup içi ve grup dışı kavramlarını kullanmaktadırlar. Bir grubun üyesi o grubu sahiplenmiş ve o grup tarafından da sahiplenilmiş kişidir. Topluma gruplandırarak inceleme karmaşıklığı azaltsa da ön yargı, ırkçılık ve taraflılık gibi istenmeyen tarafları da vardır.

Grup bir insan topluluğudur. Ancak, her rastgele toplanmış insan topluluğu grup sayılamaz. Bir topluluğun grup sayılabilmesi için hiç olmazsa ortak bir amaç için bir araya gelmeleri ve etkileşim içinde olmaları gerekir. Bir grubun üyeler arası ilişkiler düzeni vardır. Grup üyeleri; hem ortaklaşa olarak bu grubun varlığından hem de kendilerinin o grubun üyesi olduğundan haberdirlirler.

Bir insan kendini bir grubun üyesi olarak tanımladığında bu kendisini ve grubunu diğer insanlardan farklı görüyor manasına gelir. Ayrıca kendi grubunun diğerlerinden üstün olduğuna da inanır. Bu da grup üyelerinin diğer grupların üyelerine karşı kendi gruplarını desteklemelerini sağlar. Bir birinden uzak olsalar da üyeler grubu bir arada tutmaya çalışır, aynı zamanda grubun içinden olmayanlara karşı tavrı olarak veya düşmanca tutumlar sergileyerek aradaki mesafeyi korumaya çalışır. Ancak grup üyeleri kendi aralarında, bir birlerine sıkıca tutunmak yerine kendi bölgelerinde kalmak eğilimindedirler. Şekil 7’de bahsedilen kavramların uygulamaya dönüştürülme şeklini göstermektedir. Bu örnekte, veri seti 35 üyeden oluşmaktadır. 3 sınıf mevcuttur; “Yıldız”, “Çember” ve “Kare”. Yıldız grubunun 12, çember grubunun 10 ve kare grubunun 13 üyesi vardır. Küçük yıldızlar, çemberler ve kareler eş üyeleri, büyük olanlar ise küme merkezlerini ifade etmektedirler.

Derleme/Review



Şekil 7. İGFA kavramının örneği

Örnekleme yoluyla yıldız grubuna ait 4. merkezin durumuna bakıldığında; insan grup oluşma mantığına göre bu merkezin kendi grup üyelerine doğru (1., 2. ve 3. küme merkezlerine doğru) hareket etmesi ve diğer küme merkezlerinden (5., 6., 7., 8. ve 9. merkezlerden) uzaklaşması gerekmektedir. Bu hareketin hızı 3 özelliğe bağlıdır:

- 4. merkezin diğer merkezlere uzaklığı
- 4. merkezin bir arama uzayındaki hareket kabiliyeti. (Eğer bu merkez durumu iyileştirmeyecek bir hareket yaparsa bu faktörün değeri önceden belirlenmiş bir miktarda azalacaktır)
- Her grup merkezinin bölgesel sınırları (Bu faktör aynı sınıfın kümelerinin bir birine çok yakın olmasının önüne geçer)

Önerilen algoritma önceki bölümde anlatılan “İnsan Grup Oluşturma” teorisine göre geliştirilmiş bir sınıflama algoritmasıdır. Bu algoritmanın detayları aşağıda verilmiştir:

(A) Her sınıftan eğitim için kullanılan veri kümesinden keyfi olarak bir giriş örüntüsü seçilir ve bunlar başlangıç için küme merkezleri olarak tanımlanır. Başlangıçta, hedeflenen çıktı sınıfı kadar sayıda, toplam Q kadar küme vardır.

(B) Modelin doğruluğu aşağıda verildiği gibi hesaplanır:

$$\text{Doğruluk} = \frac{\sum_{i=1}^P A_i}{P} \quad (1)$$

$$A_i = \begin{cases} 1, & \text{eğer } J \in Y_i \\ 0, & \text{aksi halde} \end{cases} \quad (2)$$

$$J = \underset{j}{\operatorname{argmin}} (d_j(X_i)) \quad (3)$$

$$d_j(X_i) = \|X_i - Z_j\| \quad (4)$$

Argmin: Geometrik medyan bir uzayda bulunan aralıklı set halindeki örneklem noktaları arasında, bu noktalar arasındaki uzaklıkların toplamını en küçük (minimum) yapan bir nokta olarak tanımlanır.

P : Eğitim veri kümesindeki toplam örüntü sayısı

J : X_i giriş örüntüsüne en yakın referans örüntüsü olan kümenin indeks numarasıdır.

Y_i : i . giriş örneğinin beklenen çıktısı

z_j : j . kümenin merkezi

$d_j(X_i)$: X_i giriş örüntüsünün j . küme merkezi ile arasındaki uzaklık

(C) İnsan Grupları Oluşumu teoremine göre grup üyeleri kendi grupları ile birleşmek; üye olmayanlarla ise sosyal mesafeyi mümkün olduğunca korumak isterler. Her kümenin merkez değeri olan (z_j) aşağıdaki formül ile bulunur:

$$Z_{jk}^{\text{yeni}} = Z_{jk}^{\text{eski}} + \Delta Z_{jk} \quad (5)$$

$$\Delta Z_{jk} = \frac{\sum_{m \in q} \eta_{jm} \beta_j \delta_{jm} (Z_{mk} - Z_{jk}) - \sum_{n \in q} \eta_{jn} \beta_j \delta_{jn} (Z_{nk} - Z_{jk})}{k} \quad (6)$$

$$k = 1, 2, 3, \dots, K$$

Derleme/Review

K: Giriş örüntüsündeki özelliklerin sayısı

q: j. kümenin üyesi olduğu sınıf

$$\eta_{jm} = e^{-\left(\frac{z_{jk}-z_{mk}}{\sigma}\right)^2} \text{ ve } \eta_{jn} = e^{-\left(\frac{z_{jk}-z_{nk}}{\sigma}\right)^2}$$

η_{jm} ve η_{jn} : 0 ile 1 arasında değerler alabilir. Bunlar m. ve n. kümelerin j. kümeye olan etkisini belirler. m. ve n. kümeler j. kümeye yakın olursa yüksek değerler alacaktır. Ancak eğer m. ve n. kümeler j. kümeden uzak ise η_{jm} ve η_{jn} değerleri düşük olacaktır. Ayrıca m. ve n. kümeler η_{jm} ve η_{jn} 'nin en düşük değerleri olan j. kümeden gelmektedirler.

β_j : j. kümenin arama uzayında, kendi kabiliyeti nispetindeki hareket hızıdır.

δ_{jm} : j. kümenin iki etkene göre değişen hızıdır; birincisi j. küme ile m. küme arasındaki mesafe, ikincisi ise kümelerin bölgesel sınırları (T). Bu parametre aynı sınıfın farklı kümelerinin bir birine çok yakın olmasını engeller. Eğer j. ve m. kümeler arasındaki mesafe T'den az olursa δ_{jm} önceden belirlenmiş bir miktar kadar azalacaktır.

Tüm merkezler güncellendikten sonra doğruluk 1-4 denklemlerinde gösterildiği gibi yeniden hesaplanır. Eğer doğruluk yüksek çıkarsa bu değeri yeni merkez değeri olarak kaydedilir ve sonraki merkez güncellenir. Eğer daha düşük çıkarsa yeni merkez değeri atılır ve önceki merkeze geri dönlür. Ancak, eğer doğruluk değişmiyor ise yeni merkez değeri kaydedilir ve β_j 'nin değeri önceden belirlenen miktar kadar düşürülür. Doğruluk değişmeden kaldığı sürece β_j 'nin değeri sürekli azaltılır. Bu yapılırken, hareket etmesi iyi sonuçlar doğurmayacaksa bile, algoritma küme merkezinin arama uzayı etrafında

hareket etmesine izin verir, ancak bu çok küçük hareketlerle olur.

(D) Bu adım küme azaltma adımındır. Aşağıdaki denklemi sağlayan kümeler silinmektedir.

$$-\frac{2}{2 \log_2 \left(\frac{n_j}{p}\right)} \left(\frac{n_j^q}{n_j}\right) \left(\frac{\sum_{X_i^j \in q} \|X_i^j - z_j\|}{n_j}\right) < \rho \quad (7)$$

n_j : j. kümedeki giriş örüntüsü sayısı

n_j^q : j. kümedeki, beklenen çıktıları (Y) q

olan giriş örüntüsü sayısı

q: j. kümenin ait olduğu sınıf

P: Eğitim veri kümesindeki toplam örüntü sayısı

X_i^j : j. kümedeki i. giriş örüntüsü

ρ : Sınır parametresi (dikkat parametresi)

(E) Modelin doğruluğu 1-4 denklemlerinde gösterildiği gibi yeniden hesaplanır.

(F) Kalan her bir küme için, C adımında güncellenen yeni merkez ile eski merkez arasındaki fark 0.0001'den az ise $(\|Z_{jk}^{yeni} - Z_{jk}^{eski}\| < 0.0001)$ -0,1 ile +0,1 arasında rasgele K küçük sayıları üretilir ve bunlar kümenin merkez değerine atanır. Bu adımın amacı algoritmanın erken yerel optimum değerlere takılı kalmasını engellemektir.

$$\begin{bmatrix} Z_{j1}^{yeni} \\ Z_{j2}^{yeni} \\ \vdots \\ Z_{jk}^{yeni} \end{bmatrix} = \begin{bmatrix} Z_{j1}^{eski} \\ Z_{j2}^{eski} \\ \vdots \\ Z_{jk}^{eski} \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_k \end{bmatrix} \quad (8)$$

Eğer yukarıda verilen işlem doğruluğu daha yüksek yapıyorsa bu yeni merkez değeri kaydedilir ve sonraki küme merkezi ile devam edilir. Ancak eğer doğruluk değerinde bir iyileşme olmuyorsa yeni merkez değeri iptal edilir ve bu adım doğruluk

Derleme/Review

iyileşene veya önceden belirlenmiş iterasyon (tekrar) sayısı aşılanaya kadar tekrar edilir.

(G) Önceden belirlenmiş iterasyon sayısı aşılmış veya sonlandırma şartı sağlanmış ise döngü durdurulur. Eğer değilse aşağıdaki durumlar uygulanır:

(G.1) Eğer modelin doğruluğu önceki iterasyondan daha iyi ise; halen hatalı olan her hedef çıktı sınıfının eğitim veri kümesinden gelişigüzel bir giriş örneği seçilir ve B adımına dönülür.

(G.2) Eğer doğruluk iyileşmiyorsa her hedef çıktı sınıfının eğitim veri kümesinden gelişigüzel giriş örüntüsü ve D adımında silinen kümelerin sayısına eşit olan sayı seçilir. Sonra da B adımına gidilir.

Sosyal Tabanlı Algoritma

Sosyal Tabanlı Algoritma (STA), Evrimsel Algoritma ve EYA tabanlı sosyo-politik bir sürecin birleştirilmesiyle önerilmiş yeni bir algoritmadır. İnsanlar farklı tipteki topluluklarda yaşarlar: Monarşi, Cumhuriyet, Otokrasi ve Çok uluslu.

Her topluluktaki liderlik stili de farklıdır. Bu yaklaşım topluluk gelişme karakteristiğinde birkaç kişiyi yakalamaya çalışır (Ramezani ve Lotfi, 2013). Tablo 6, STA'nın işlem adımlarını göstermektedir.

Toplum ve Medeniyet Algoritması

Toplum ve Medeniyet Algoritması (TMA) Ray ve Liew tarafından önerilen bir algoritmadır. TMA eşitlik ya da eşitsizlik kısıtları bir minimizasyon problemi kapsamında tarif edilir:

$x=[x_1 \ x_2 \ \dots \ x_n]$, n tasarım değişkenlerinin vektörüdür. Eşitlik sınırı $h_j(x) \leq \delta$ ya da $h_j(x) \geq -\delta$ (δ , pozitif küçük bir sayı) gibi eşitsizliğin bir kümesine dönüştürülmüştür Bu nedenle, r eşitlik sınırı $2r$ eşitsizliğine yol açacaktır ve

problem için eşitsizliklerin toplam sayısı s ile gösterilir. $s = q + 2r$ ile ifade edilir. Her bir birey için c , $c=[c_1 \ c_2 \ \dots \ c_n]$ tarafından verilen kısıt doyum vektörünü ifade eder.

$$c_i = \begin{cases} 0, & i. \text{ kısıt sağlanmışsa} \\ & i = 1, 2, \dots, s \\ -g_i(x), & i. \text{ kısıt ihlal edilmişse} \\ & i = 1, 2, \dots, q \\ h_i(x) - \delta, & i. \text{ kısıt ihlal edilmişse} \\ & i = q + 1, q + 2, \dots, q + r \\ \delta - h_i(x), & i. \text{ kısıt ihlal edilmişse} \\ & i = q + r + 1, q + r + 2, \dots, s \end{cases} \quad (9)$$

Yukarıdaki c_i 's için, $c_i = 0$ i . kısıtın sağlandığını gösterir, fakat $c_i > 0$ kısıtın ihlalini gösterir. G çözümünün bir kümesi için Kısıt matrisi (10)'daki gibi ifade edilir.

$$\text{KISIT} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1s} \\ c_{21} & c_{22} & \dots & c_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_{G1} & c_{G2} & \dots & c_{Gs} \end{bmatrix} \quad (10)$$

C_{ij} , $i = 1, \dots, G$, $j = 1, \dots, s$ i bireyi için j kısıtının sağlanmasını ifade eder. Bu G bireyleri arasında, baskın olmayan tüm bireyler kısıt matrislerine göre bir derece atanır.

Seviye 1 bireyler r kümesinden kaldırılır ve baskın olmayan bireylerin yeni kümesi iki derece atanır. İşlem adımları tüm bireyler bir dereceye konulana kadar devam eder. Kısıt matrisindeki derecenin 1 olması, baskın olmayan bireyleri gösterir (Ray ve Liew, 2003). Tablo 7, TMA'nın sözde kodunu göstermektedir.

Derleme/Review

Tablo 6. STA'nın işlem adımları

<ol style="list-style-type: none">1. Parametrelerin yüklenmesi2.<ol style="list-style-type: none">2.1. Optimizasyon probleminin tanımlanması,2.2. Rastgele bireylerin oluşturulması,2.3. Bazı güçlü insanların rastgele lider olarak seçilmesi,2.4. Kalan bireylerin rastgele farklı bölgelere konumlandırılması,2.5. Emperyalist maliyet fonksiyonu $T.P_{ci}$ ile imparatorlukların başlatılması,2.6. Çok güçlü liderlerin imparatorluk olarak seçilmesi,3. On döngü $N_d=N_d+1$4. $i= 1, 2, \dots, N$<ol style="list-style-type: none">4.1. Seçim4.2. Çaprazlama4.3. Mutasyon4.4. Yerine koyma5. $i=1, 2, \dots, N$<ol style="list-style-type: none">5.1. İnsan asimilasyon politikası: Her bir grubun liderlerinin kendi imparatorluklarına taşınması, $x \sim U(0, \text{katsayı}_{i\text{ç}} \text{ asimilasyon} \times d)$ d: lider ve emperyalist arasındaki uzaklık5.2. İnsanların devrimi5.3. Ülkelerin asimilasyon politikası: Her bir grubun liderlerinin kendi imparatorluklarına taşınması ve her bir ülkenin insanların aynı liderlerin gibi taşınması, $x \sim U(0, \text{katsayı}_{dış} \text{ asimilasyon} \times d)$ d: lider ve emperyalist arasındaki uzaklık5.4. Ülkelerin devrimi5.5. Konumun değiştirilmesi5.6. Emperyalist yarış; zayıf imparatorluktan zayıf ülkenin seçilmesi ve ona sahip olma ihtimali en yüksek imparatorluğa verilmesi5.7. Eleme yapılması; güçsüz ilke ve imparatorluğun elenmesi6. Sonlandırma şartının kontrol edilmesi, sonlandırma şartı sağlanana kadar 3-6 arasındaki adımların tekrar edilmesi.

Kültürel Algoritma

Kültürel algoritma (KA) Robert G. Reynolds tarafından geliştirilmiştir. Genetik kavramlar ve doğal seleksiyon mekanizması odaklı metafor tamamlayıcı olarak evrimsel algoritmalar tarafından kabul edilir. Kültürel algoritma, kültürel gelişimleri

modellemek için sosyoloji ve arkeoloji alanında önerilen bazı teorilere dayanmaktadır. Bu gibi teoriler, kültürel gelişimi iki seviyede gerçekleşen bir kalıtım süreci olarak göstermektedir: Mikro gelişim seviyesi ve makro gelişim seviyesi.

Tablo 7. TMA'nın sözde kodu

<ol style="list-style-type: none"> 1. $t \leftarrow 0$ 2. Medeniyetleri gösteren N birey üret: Parametrik alanda eşit $Civ(t) = \{I_1, I_2, \dots, I_N\}$ 3. Her bir bireyi değerlendir: Amaç ve kısıtları hesapla: $i = 1, 2, \dots, N$ bireyler ve $j = 1, \dots, s$ kısıtlar için $f(I_i)$ ve $C_j(I_i)$ gibi. 4. Topluları inşa et: Kümelerin toplulukları gösterdiği $K(t)$ oluştur. $i \neq j$ için $U_{i=1}^{K(t)} Soc_i(t) = Civ(t)$ ve $Soc_i(t) \cap Soc_j(t) = \emptyset$ $Soc_1(t), \dots, Soc_{K(t)}(t)$, 5. Liderlerin belirlenmesi: Her toplumun liderleri $Soc_{L_i}(t) \subset Soc_i(t)$, $i = 1, 2, \dots, K(t)$ 6. Taşı: Her $I_j \in Soc_i(t)$ ve $I_j \notin Soc_{L_i}(t)$ için yeni konumuna taşı. a) $Civ(t+1) \leftarrow \emptyset$ b) Toplumun en yakın lideri tarafından yönlendirilen $I_j \rightarrow I_j^1$ taşı 7. Liderlerin tanımlanması: Medeniyetin liderleri: $Civ_{L_i}(t) \subset U_{i=1}^{K(t)} Soc_{L_i}(t)$ 8. Medeniyet liderleri dışındaki tüm liderleri yeni bir konuma taşı. a) Her $I_j \in Soc_{L_i}(t)$ ve $I_j \notin Civ_{L_i}(t)$ için En yakın medeniyet liderleri tarafından $I_j \rightarrow I_j^1$ taşı. $Civ(t+1) \leftarrow Civ(t+1) \cup \{I_j^1\}$ 9. $Civ(t+1) \leftarrow Civ_{L_i}(t) \cup Civ(t+1)$ 10. $t \leftarrow t+1$ 11. Eğer $t < T_{max}$ 3. adıma geri dön, değilse bitir.

KA sosyal popülasyon ve inanç uzayı olmak üzere iki uzay üzerinde çalışır. Birincisi, onlar bireylerin bir küme olarak kabul edildiği bir evrimsel hesaplama tekniği gibi popülasyon uzayı üzerinde işlem görürler. Her bir birey onların uygunluk değerlerini hesaplamaya izin veren birbirinden bağımsız özellik kümesine sahiptir. Zamanla, bu bireyler, popülasyon operatörlerinin kümesinin uygulanmasından sonra torunları ile yer değiştirirler (Coello ve ark., 2003).

İkinci uzay ise inanç uzayıdır. Bu uzayda, bilgi evrimsel süreç boyunca saklanan bireylerden elde edilir. Uzayda bulunan bilgiye herhangi bir birey tarafından ulaşılabilir.

Bu nedenle bilgi davranışları değişmek için kullanılabilir.

KA çatısı Şekil 8'de verilmiştir. Etki fonksiyonunda yapılanlar ise Şekil 9'da açıklanmıştır.

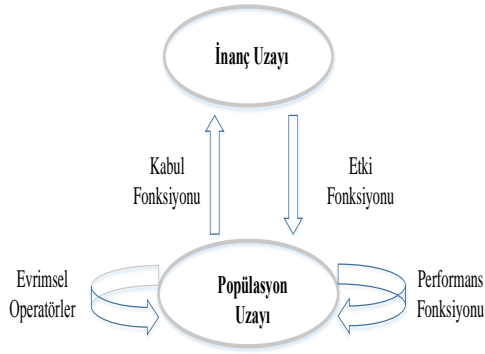
İki uzayı birleştirmek için, bir iletişim protokolü kurulmuştur. Protokol iki uzay arasında değiştirilecek bilgi tipi ile ilgili kuralları belirler. Örneğin;

inanç uzayını güncellemek için bireylerin seçilen bir kümesinin bireysel deneyimleri dâhil edilir. Bireylerin seçilen grubu kabul fonksiyonu ile elde edilir. Kabul fonksiyonu popülasyonunun tamamına uygulanır. Diğer yandan, popülasyonu değiştiren operatör ve seçim operatörü etki fonksiyonu ile değiştirilir. Kültürel algoritmanın sözde kodu Tablo 8'de verilmiştir.

Parlamente Optimizasyon Algoritması

Parlamentarizm olarak da bilinen parlamente sistem, yasa yapan ve düzenleyen bir hükümet sistemidir (Borji ve Gamidi, 2009). Parlamento üyeleri genel seçimlerde halk tarafından seçilirler. Parlamente Optimizasyon Algoritması (POA), gerçek dünyadaki parlamente sistemden etkilenecek geliştirilen bir algoritmadır.

Derleme/Review



Şekil 8. Kültürel algoritma çatısı

Tablo 8. KA'nın sözde kodu

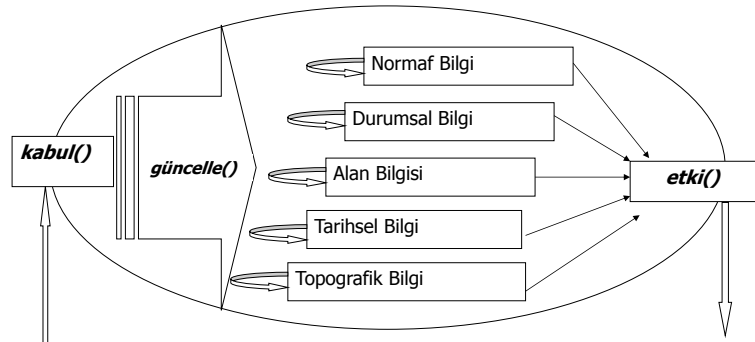
1.	Begin
2.	$t = 0;$
3.	Popülasyonu üret $POP(t);$
4.	İnanç uzayını üret $İNANC(t);$
5.	repeat
	i. $POP(t)$ Popülasyonunu değerlendir;
	ii. Ayarla $(İNANC(t, Kabul Et(POP(t)));$
	iii. Ayarla($İNANC(t);$
	iv. Vasyasyon($POP(t-1)$ 'den $POP(t);$
6.	until sonlandırma şartı sağlandı
7.	End

POA, optimizasyon adımları başlangıç birey popülasyonunun oluşturulması ile başlar. Bu bireyler parlamentonun üyeleri olarak kabul edilir. Bir sonraki adımda, popülasyon politik gruplara (partilere) bölünür ve en yüksek uygunluğa sahip sabit sayıda üye

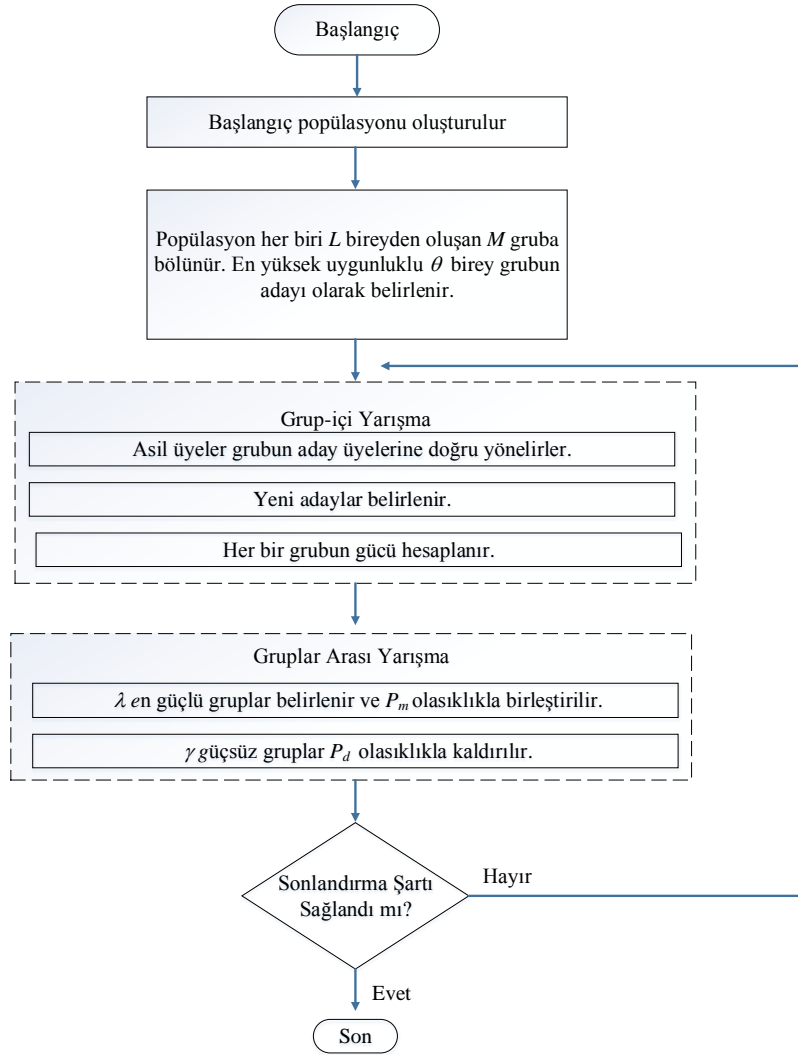
grubun adayı olarak kabul edilir. Bu adımdan sonra, grup içi yarışma başlar. Grup içi yarışma adımı asil üyeler kendilerine uygun aday üyelere doğru yönelirler. Bu durum asil üye adaylarının vektörlerinin ağırlıklı ortalaması olarak modellenmiştir. Grup içi yarışma adımının sonunda en yüksek uygunluğa sahip birkaç aday her bir grubun final adayı olarak belirlenir. Bu adaylar bir sonraki adımda diğer grupların adayları ile yarışır. Grubun toplam gücünün belirlenmesi bakımından grubun asil ve adayları önemlidir. Gruplar arası yarışma grup içi yarışma adımının sonunda başlar. Parlamento içerisindeki politik gruplar kendi adaylarını güçlendirmek için diğer gruplarla yarışır. Güçlü gruplar bazen kazanma şanslarını artırmak için birleşirler ve tek bir grup olurlar (Altunbey ve Alatas, 2015).

Algoritma sonunda, bir grup yarışı kazanır ve onun en iyi elemanı optimizasyon probleminin çözümü olarak nitelendirilir. İki sonlandırma durumu mevcuttur: Maksimum sayıda iterasyona ulaşıldığında veya bazı başarılı iterasyon sonucunda uygunluk değerinde dikkate değer iyileşme gözlenmezse algoritma sonlandırılır.

POA'nın akış diyagramı Şekil 10'da verilmiştir.



Şekil 9. Bilgi kaynaklarını etki fonksiyonuna entegre etme



Şekil 10. POA'nın akış diyagramı

SONUÇ

Yapay zekâ optimizasyon algoritmaları her ne kadar genel amaçlı arama ve optimizasyon algoritmaları olsalar ve geniş şekilde farklı problemlere etkili çözümler sunsalar da, özellikle sosyal tabanlı olanlar esinlendikleri kaynak itibariyle sosyal ağ analizinde de etkili şekilde kullanılabilir görülmektedir. Bu makalede de çok güncel olan ve ileride popüler hale gelmeye başlaması çok muhtemel sosyal tabanlı yapay zekâ

algoritmaları tanıtılıp, sosyal ağ analizindeki olası kullanım alanları hakkında bilgiler sunulmuştur. Bu kullanım alanlarına örnek olarak bağlantı tahmini, topluluk keşfi, duygu analizi, metin özetleme, metin sınıflandırma verilebilir.

İlerleyen çalışmalarda, bu yapay zekâ algoritmalarının sosyal ağ analizindeki ilk çalışmalarının yapıları sonuçlarının bilim dünyasına sunulması hedeflenmektedir.

Derleme/Review

KAYNAKLAR

- Altunbey, F., Alatas, B.,** 2015. Overlapping community detection in social networks using parliamentary optimization algorithm. *International Journal of Computer Networks and Applications*, 2(1): 12-19.
- Atashpaz-Gargari, E., Lucas, C.,** 2007. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *IEEE Congress on Evolutionary Computation*, CEC 2007, 4661-4667.
- Borji, A., Gamidi, M.,** 2009. A new approach to global optimization motivated by parliamentary political competitions. *Int. Journal of Innovative Computing, Information and Control*, 5(6): 1643-1653.
- Bliss, C.A., Frank, M.R., Danforth, C.M., Dodds, P.S.,** 2014. An evolutionary algorithm approach to link prediction in dynamic social networks. *Journal of Computational Science*, 5(5): 750-764
- Coello, C., Carlos, A., Becerra, R. L.,** 2003. Evolutionary multiobjective optimization using a cultural algorithm. *In Swarm Intelligence Symposium*, 2003. SIS'03. Proceedings of the 2003 IEEE, pp: 6-13.
- Coscia, M., Giannotti, F., Pedreschi, D.,** 2011. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining*, 4(5): 512-546.
- Cui, Z. H., Shi, Z. Z., Zeng, J. C.,** 2010. Using social emotional optimization algorithm to direct orbits of chaotic systems. *1st International Conference on Swarm, Evolution and Memetic Computation (SEMCCO2010)*, pp. 389-395.
- Devi, S., Singla, S.,** 2012. Comparative analysis of modified social emotional optimization algorithm & particle swarm optimization techniques. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, pp. 82-84.
- Duarte, A., Sánchez, A., Fernández, F., Montemayor, A.S.,** 2006. Improving image segmentation quality through effective region merging using a hierarchical social metaheuristic. *Pattern Recognition Letters*, 27: 1239-1251.
- Fernandez, F., Duarte, A., Sanchez, A.,** 2003a. A software pipelining method based on hierarchical social algorithms. *Proceedings of MISTA Conference*, 1: 382-385.
- Fernandez, F., Duarte, A., Sanchez, A.,** 2003b. Hierarchical social algorithms: A new metaheuristic for solving discrete bilevel optimization problems. Technical Report ESCET/URF - DTF/UPM.
- Fortunato, S.,** 2010. Community detection in graphs. *Physics Reports* 486(3): 75-174.
- Karaboğa, D.,** 2011. Yapay zekâ optimizasyon algoritmaları, Nobel Yayın Dağıtım, Ankara.
- Osborn, A.F.,** 1957. Applied imagination, New York: Scribner.
- Pooranian, Z., Shojafar, M., Abawajy, J. H., Singhal, M.,** 2013. GLOA: A new job scheduling algorithm for grid computing. *International Journal of Interactive Multimedia and Artificial Intelligence*, 2(1): 59-64.
- Rao, R.V., Patel, V.,** 2012. An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Scientia Iranica*, pp. 710-720.
- Ramezani, F., Lotfi, S.,** 2013. Social-based algorithm (SBA). *Applied Soft Computing*, 13(5): 2837-2856.
- Ray, T., Liew, K. M.,** 2003. Society and civilization: an optimization algorithm based on the simulation of social behavior. *Evolutionary Computation, IEEE Transactions on*, 7(4): 386-396.
- Thammano, A., Moolwong, J.,** 2010. A new computational intelligence technique based on human group formation. *Expert Systems with Applications*, 37: 1628-1634.
- Uçan, A.,** 2014. Otomatik duygu sözlüğü çevirimi ve duygu analizinde kullanımı, *Yüksek Lisans Tezi*, Hacettepe Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.
- Zhan, Z., Zhang, J., Shi, Y., Liu, H.,** 2012. A modified brain storm optimization. *WCCI 2012 IEEE World Congress on Computational Intelligence*, pp. 1-8.