Int. J. Pure Appl. Sci. 7(1):100-112 (2021)

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

Kalite Test Fonksiyonları Kullanılarak Güncel Metasezgisel Optimizasyon Algoritmalarının Karşılaştırılması

Soner Kızıloluk[1]*, Ümit Can[2]

[1] Malatya Turgut Özal Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Bilgisayar Mühendisliği Bölümü, Malatya, Türkiye
[2] Munzur Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Tunceli, Türkiye
**soner.kiziloluk@ozal.edu.tr, ucan@munzur.edu.tr*
Makale gönderme tarihi: 07.01.2021, Makale kabul tarihi: 26.02.2021

Öz

Doğadaki canlıların sürü davranışlarından, bitkilerden, insana özgü olgulardan, fizik, matematik, biyoloji ve kimya gibi bilimsel alanlardaki olaylardan esinlenen onlarca metasezgisel optimizasyon yöntemi mevcuttur. Bu yöntemler belirli problemlerde başarılı olmakla birlikte bütün problemlerde başarılı olamamaktadır. Bundan dolayı araştırmacılar tarafından her geçen gün yeni metasezgisel yöntemler önerilmektedir. Bu çalışmada ilk defa güncel Yapay Deniz Anası Optimizasyonu, Etçil Bitki Optimizasyonu, Giza Piramitleri İnşaatı Optimizasyonu, Gradyan Tabanlı Optimizasyon, Öğrenci Psikolojisine Dayalı Optimizasyon ve Tunik Sürüsü Optimizasyonu olmak üzere altı güncel metasezgisel optimizasyon algoritması 10 adet matematiksel kalite test foksiyonunda 10, 30 ve 50 boyut değerleri baz alınarak ayrıntılı bir şekilde karşılaştırılmıştır. Elde edilen sonuçlara göre 10 kalite testinden 7'sinde en iyi sonuçları Öğrenci Psikolojisine Dayalı Optimizasyon vermiştir. Gradyan Tabanlı Optimizasyon'un ise 4 kalite testinde en iyi sonuçları verdiği görülmüştür. En kötü performansı ise Etçil Bitki Optimizasyonu ve Tunik Sürüsü Optimizasyonu göstermiştir. Süre bakımından karşılaştırmak üzere algoritmalar 50 boyutlu test fonksiyonlarında 1000 iterasyonda çalıştırılmış ve elde edilen ortalama çalışma süreleri incelendiğinde, Yapay Deniz Anası Optimizasyonu ve Tunik Sürüsü Optimizasyonu'nun en hızlı çalışan algoritmalar olduğu görülmektedir. Etçil Bitki Optimizasyonu ve Öğrenci Psikolojisine Dayalı Optimizasyon ise en yavaş çalışan algoritmalar olmuştur.

Anahtar kelimeler: Global optimizasyon, kalite test fonksiyonları, metasezgisel algoritmalar

Comparison of Current Metaheuristic Optimization Algorithms by Using Benchmark Functions

Abstract

There are dozens of metaheuristic optimization methods inspired by the swarm behaviors of creatures in nature, plants, human-specific phenomena, events in scientific fields such as physics, mathematics, biology and chemistry. These methods are successful in certain problems but not in all problems. Therefore, new metaheuristic methods are suggested by researchers frequently. In this study, for the first time, six up-to-date metaheuristic optimization algorithms, namely Jellyfish Search Optimizer, Carnivorous Plant Algorithm, Giza Pyramids Construction Algorithm, Gradient Based Optimizer, Student Based Psychology Optimization and Tunicate Swarm Optimization, were compared by using 10 mathematical benchmark functions with 10, 30 and 50 dimensions. According to the results obtained, Student Based Psychology Optimization gave the best results in 7 out of 10 benchmark functions. It was observed that Gradient Based Optimizer gave the best results in 4 benchmark functions. Carnivorous Plant Algorithm and Tunicate Swarm Optimization showed the worst performance. In order to compare in terms of time, algorithms were run at 1000 iterations in 50-dimensional benchmark functions. When the average run times obtained are examined, it is seen that Jellyfish Search Optimizer and Tunicate Swarm Optimization are the fastest running algorithms. Carnivorous Plant Algorithm and Student Based Psychology Optimization were the slowest running algorithms.

Keywords: Benchmark functions, global optimization, metaheuristic algorithms

Int. J. Pure Appl. Sci. 7(1):100-112 (2021)

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

GİRİŞ

Çeşitli bilim ve mühendislik alanlarındaki birçok gerçek dünya uygulaması optimizasyon problemlerine dönüştürülebilir. Çözülmeye çalışılan bu problemler doğrusal olmayan, çok modlu ve oldukça karmaşık problemlerdir. Bir problemin çözüm uzayının tümünün değerlendirilemeyeceği ve sonsuz büyüklükte olduğu durumlarda optimum çözüme en yakın sonucun makul bir zaman diliminde bulunabilmesi için metasezgisel optimizasyon algoritmaları kullanılmaktadır. Metasezgisel optimizasyon algoritmaları oldukça popüler yöntemlerdir ve bunun dört önemli nedeni vardır; (a) Basit konseptlere dayanmaktadırlar ve uygulanmaları kolaydır; (b) amaç fonksiyonu sınırlayıcıların ve kullanılan değişkenlerin tipine bağlı değildir; (c) yerel minimum noktasını baypas edebilirler; ve (d) çeşitli alanlardaki değişik problemleri çözmek için kullanılabilirler (Altunbey ve Alataş, 2015; Mirjalili ve Lewis, 2016; Gao ve Silva, 2018; Chou ve Truong, 2021). Metasezgisel optimizasyon algoritmaları esinlendikleri alanlara göre evrimsel tabanlı, fizik tabanlı, kimya tabanlı, sürü tabanlı, insana dayalı ve bitki tabanlı olmak üzere temel olarak altı sınıfa ayrılabilir.

Evrimsel yöntemler içindeki en popüler olanları Genetik Algoritma (Holland, 1992) ve Diferansiyel Gelişim Algoritmasıdır (Storn ve Price, 1997). Fizik kurallarından veya kanunlarından esinlenilerek geliştirilen fizik tabanlı metasezgisel optimizasyon algoritmalarına Elektromanyetizma Benzeri Algoritma (Birbil ve Fang, 2003), Yerçekimsel Arama Algoritması (Rashedi ve ark., 2009), Parçacık Çarpışma Algoritması (Sacco ve ark., 2005) ve Yapay Fizik Algoritması (Xie ve ark., 2009) örnek olarak verilebilir. Kimya tabanlı yöntemlere ise Yapay Kimyasal Reaksiyon Optimizasyon (Alatas, 2012) algoritması örnek verilebilir. Doğada hayvanların sosyal davranışlarını taklit ederek ve onların sürü zekâsını kullanarak geliştirilen yöntemler sürü tabanlı algoritmalar olarak adlandırılırlar. Bunlar güve, arı, kuş, kedi, kurt ve balina gibi hayvanların davranışlarından esinlenen algoritmalardır. Parçacık Sürü Optimizasyonu (Kennedy ve Eberhart, 1995), Yapay Arı Kolonisi (Karaboga ve Akay, 2009) ve Yarasa Algoritması (Yang ve Gandomi, 2012) gibi algoritmalar sürü tabanlı algoritmalara örnek olarak verilebilir. Spor, müzik, eğitim-öğretim ve yönetim biçimi gibi çeşitli insana dayalı kaynaklardan esinlenerek geliştirilen birçok algoritma bulunmaktadır. Lig Şampiyonası Algoritması (Kashan, 2014) ve Altın Top Optimizasyonu (Osaba ve ark., 2014) spordan esinlenen algoritmalara örnek olarak verilebilir. Harmoni Arama (Lee ve Geem., 2005) ve Melodi Arama (Ashrafi ve Dariane, 2011) algoritmaları müzikten esinlenen yöntemlerdir. Parlamenter sistemden esinlenen Parlamenter Optimizasyon Algoritması (Borji ve Hamidi, 2009) parlamenter sistemin işleyişini simule etmiştir. Bir öğretmenin sınıftaki öğrencilerin öğrenme üzerindeki etkisinden esinlenilerek Öğretme-öğrenmeye dayalı optimizasyon algoritması (Rao ve ark., 2012) geliştirilmiştir. Tüm bu kategorilerin dışında araştırmacılar tarafından bitki davranışlarından esinlenen bitki tabanlı algoritmalar karmaşık problemlerin çözümü için önerilmiştir. Kök kütlesi optimizasyon algoritması (Qi ve ark., 2013; Can ve Alataş, 2015) ve Çiçek Tozlaşması Algoritması (Yang, 2012) bu tür tekniklere örnektir.

Araştırmacılar tarafından yeni birçok optimizasyon algoritması geliştirilmesine rağmen hala bu zorlu problemlerin çözümünde tatmin edici sonuçlar ortaya koymada yeterli olamamaktadırlar (Ahmadianfar ve ark., 2020). Optimizasyon problemlerine daha başarılı çözümler sunmak için araştırmacılar tarafından yeni sezgisel yöntemler önerilmektedir. Yapay Deniz Anası Optimizasyonu (YDAO) (Chou ve Truong, 2021), Etçil Bitki Optimizasyonu (EBO) (Ong ve ark., 2021), Giza Piramitleri İnşaatı Optimizasyonu (GPİO) (Harifi ve ark., 2020), Gradyan Tabanlı Optimizasyon (GTO) (Ahmadianfar ve ark., 2020), Öğrenci Psikolojisine Dayalı Optimizasyon (ÖPDO) (Das ve ark., 2020) ve Tunik Sürüsü Optimizasyonu (TSO) (Kaur ve ark., 2020) 2020 yılı içerisinde önerilmiş güncel metasezgisel yöntemlerdir. Bu çalışmada son bir yıl içerisinde önerilen güncel altı adet metasezgisel algoritma incelenmiş ve bu algoritmalar ilk defa on adet sıkça kullanılan kalite test fonksiyonu kullanılarak karşılaştırılmıştır. Elde edilen sonuçlar detaylı bir şekilde verilmiştir.

Giriş bölümünde metasezgisel optimizasyon yöntemleri anlatılarak literatürdeki mevcut çalışmalar hakkında bilgi verilmiştir. Güncel Metasezgisel Optimizasyon Algoritmaları başlığında ise çalışmada kullanılan güncel metasezgisel algoritmaların esinlenme kaynakları verilerek bu algoritmaların kaba kodları gösterilmiştir. Deneysel Çalışma ve Sonuçlar bölümünde algoritmalar on adet matematiksel kalite test fonksiyonu kullanılarak

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

karşılaştırılmış ve sonuçlar ayrıntılı bir şekilde verilmiştir. Sonuç bölümünde ise deneysel çalışmalar sonucunda elde edilen sonuçlar değerlendirilmiştir

GÜNCEL METASEZGİSEL OPTİMİZASYON ALGORİTMALARI

Bilimsel araştırmalar sürdükçe insanların keşifleri artmakta bu da yeni metasezgisel algoritmalar geliştirilmesi için farklı esin kaynaklarının ortaya çıkmasını sağlamaktadır. Bu bölümde, deneysel çalışmada karşılaştırmaları yapılan 6 farklı güncel metasezgisel optimizasyon algoritmasının temel esin kaynakları ve algoritma adımlarını içeren kaba kodları alt başlıklarda verilmiştir.

Yapay Denizanası Arama Optimizasyonu

Denizanaları çeşitli boyutlarda ve çeşitli renklerde var olan ve farklı sıcaklıklardaki sularda farklı derinliklerde yaşayabilen canlılardır. Denizanalarının beslenme yöntemleri birbirinden farklıdır. Bazı denizanaları yiyecekleri ağızlarına götürmek için dokunaçlarını kullanırken, bazıları ise akıntıların getirdikleri ile beslenmek için filtreleme yöntemini kullanırlar. Bazı denizanaları avlarını aktif olarak avlar ve dokunaçlarıyla sokarak onları hareketsiz hale getirirler. Yapay denizanası arama optimizasyon (YDAO) algoritması denizanalarının okyanuslardaki yiyecek arama davranışından esinlenilmiş ve matematiksel olarak bir denizanası topluluğu modellenmiştir. Bu modeli temel alan algoritma üç kural üzerine kurulmuştur (Chou ve Truong, 2021).

a) Bir denizanası ya okyanus akıntısını takip eder ya da sürü içinde hareket eder ve bir "zaman kontrol mekanizması" bu hareketler arasındaki geçişi belirler.

b) Denizanası yiyecek aramak için okyanusta hareket eder. Mevcut yiyecek miktarının daha fazla olduğu yerlere daha çok çekilir.

c) Bulunan yiyeceğin miktarına konum ve konumun uygunluk fonksiyonu karar verir.

YDAO algoritmasının temel adımları Algoritma 1' deki kaba kodda verilmiştir.

ALGORİTMA 1
START
Arama uzayı sınırlarını, popülasyon sayısı (N_{pop}) ve maksimum iterasyon (*Max it*) sayısını ayarla.
Başlangıç popülasyonunu oluştur.
Popülasyondaki herbir denizanasının uygunluk değerini hesapla.
Uygunluğu en iyi (En çok yiyeceğin bulunduğu) denizanasını X^* olarak tanımla.
İterasyonu $t=1$ olarak ayarla.
DO
FOR1 $i=1:N_{pop}$
Zaman kontrolü $c(t)$'yi hesapla.
IF1 $c(t)\geq 0.5$: (Denizanası okyanus akıntısını takip eder)
(1) Okyanus akıntısına karar ver.
(2) Denizanasının yeni konumuna karar ver.
ELSE (Denizanası sürü içinde hareket eder)
IF2 $rand(0,1)>(1-c(t))$
(1) Denizanasının yeni konumuna karar ver.
ELSE
(2) Denizanasının yönüne karar ver.
(3) Denizanasının yeni konumuna karar ver.
END IF2
END IF1
Sınır şartlarını kontrol et ve yeni noktadaki yiyecek miktarını bul
Mevcut denizanasının konumunu ve en çok yiyeceğe sahip denizanasının (X^*) konumunu güncelle.
END FOR1
$t=t+1$.
WHILE $t>Max\ it$
En iyi deniz anasını problemin çözümü olarak kabul et.
STOP

Etçil Bitki Optimizasyonu

Bitkilerin birçoğu hayvanlar için besin kaynağı konumunda iken etçil bitkiler için bu durum tam tersidir. Etçil bitkiler zor şartlarda hayatta kalmanın yanı sıra sinek, kelebek, kertenkele ve fare gibi hayvanları avlar (Ong ve ark., 2021). Bu noktada Etçil Bitki Optimizasyon (EBO) algoritması etçil bitkilerin zorlu şartlarda hayatta kalmak için avlanma davranışlarından ve üremek için tozlaşma adaptasyonlarından esinlenilerek önerilmiştir. Optimizasyon için etçil bitkilerin çekicilik, tuzağa düşürme, sindirme ve üreme stratejileri matematiksel olarak modellenmiştir. Algoritma 2' de EBO algoritmasının kaba kodu gösterilmektedir.

Int. J. Pure Appl. Sci. 7(1):100-112 (2021)

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

ALGORİTMA 2
START
grup_iter, *cekim_orani*, *buyume_orani*, *ureme_orani*, *nEBitki* ve *nAv* değişkenlerini tanımla.
d boyutlu ve *n* bireyli rastgele başlangıç popülasyonu oluştur.
Uygunluk değerlerine göre bireyleri sırala.
En iyi bireyi *g** olarak tanımla.
WHILE (Bitim şartı sağlanıncaya kadar)
En iyi bireyleri etçil bitkiler olarak sınıflandır (*nEBitki*).
Geriye kalan bireyleri av olarak sınıflandır (*nAv*).
Etçil bitkileri ve avları gruplandır
FOR1 *i=0: nEBitki*
FOR2 *j=1: grup_iter*
IF1 *cekim_orani* > rastgele üretilen sayı
Yeni bir etçil bitki üret.
ELSE
Yeni bir av üret.
END IF1
END FOR2
END FOR1
FOR3 *i=1: nEBitki*
Yeni bir etçil bitki üret.
END FOR3
Her bir yeni etçil bitkinin ve yeni avların uygunluğunu hesapla.
Eski ve yeni üretilmiş etçil bitkileri ve avları birleştir.
Bireyleri uygunluk değerlerine göre sırala ve en iyi n adet bireyi sonraki nesle aktar.
g'* ı güncelle.
END WHILE
En iyi bireyi (*g**) problemin çözümü olarak kabul et.
STOP

Giza Piramitleri İnşaatı Optimizasyonu

Giza Piramitleri eski Mısır'da inşa edilmiş üç büyük piramitten oluşan bir yapı bloğudur. Arkeologlara göre birbirinden farklı büyüklükteki bu piramitlerin yapımı zamana yayıldığı için yapım yöntemleri birbirinden farklıdır. Bu inşaatların yapımında en önemli konulardan biri her biri inşaat alanında farklı sorumluluklara sahip hamallar, köleler, masonlar, metal işçileri ve marangozların nasıl yönetileceği meselesi idi. Bunlar firavunun ajanı tarafından yönlendirilirlerdi. Ayrıca kullanılan inşaat malzemelerinin kısıtlılığı, inşaat süresi sorunu ve kullanılan taş bloklar nedeniyle piramitlerin yapımı optimize edilmiştir. Giza Piramitleri İnşaatı Optimizasyon (GPİO) algoritması bu piramitlerin yapıldığı dönemdeki metotları, teknolojileri ve stratejileri gözlemleyip bunlardan esinlenen bir algoritma olarak ortaya çıkmıştır (Harifi ve ark., 2020). Algoritma 3' te GPİO algoritmasının kaba kodu verilmiştir.

ALGORİTMA 3
START
Rastgele üretilmiş *n* adet taş bloklar veya işçiler ile başlangıç popülasyonunu oluştur.
Taş blok ve işçilerin uygunluk değerini hesapla.
En iyi işçiyi firavunun ajanı olarak belirle.
FOR1 *It=1:MaxIt*
FOR2 *i=1:n*
Taş blok yer değiştirme miktarını hesapla.
İşçi hareketinin miktarını hesapla.
Yeni pozisyonu tahmin et.
İşçileri ikame etme olasılığını araştır.
Yeni pozisyon ve uygunluk değerlerini hesapla.
IF1 *yeni uygunluk değeri<Firavunun ajanı uygunluk değeri*
Yeni uygunluk değerini firavunun ajanı uygunluk değeri olarak ayarla.
END IF1
END FOR2
Sonraki iterasyon için çözümleri sırala.
END FOR1
STOP

Gradyan Tabanlı Optimizasyon

Gradyan Tabanlı Optimizasyon (GTO) algoritması Newton'un gradyan tabanlı metodundan esinlenilerek önerilmiştir (Ahmadianfar ve ark., 2020). Newton'un metodu denklemleri sayısal olarak çözen güçlü bir metottur ve bu yöntem, Taylor serisinin ilk terimlerini kullanan bir kök bulma algoritmasıdır. Bu metottan esinlenen GTO algoritmasının işleyişi sırasında iki önemli operatör kullanılır. Bunlar gradyan arama kuralı (GAK) ve yerel kaçış operatörü (YKO) olarak adlandırılırlar. GAK, arama uzayında keşif eğilimini arttırmak ve yakınsama oranını hızlandırmak için gradyan tabanlı yöntemi kullanır. YKO ise GTO algoritmasının yerel minimumdan kaçınması için kullandığı bir operatör işlevi görmektedir (Ahmadianfar ve ark., 2020).

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

Algoritma 4' te GTO algoritmasının kaba kodu gösterilmektedir.

ALGORİTMA 4
START
pr,*ε* ve *M* parametrelerinin değerlerini ata.
Başlangıç popülasyonunu oluştur.
Popülasyondaki her elemanın uygunluk değerini hesapla.
En iyi ve en kötü elemanı belirle.
WHILE *m<MaxIt*
FOR1 *n =1:Eleman sayısı*
FOR2 *i=1:Boyut sayısı*
[1, N] aralığında *r1≠r2≠r3≠r4≠n* değerlerini rastgele seç.
Elemanın yeni pozisyonunu hesapla ($X^m_{n,i}$).
END FOR2
//Yerel kaçış operatörü (YKO)
IF *random<pr*
X^m_{YKO}' nun pozisyonunu hesapla.
$X^{m+1}_{n,i} = X^m_{YKO}$.
END IF
En iyi ve en kötü elemanı güncelle.
END FOR1
m=m+1.
END WHILE
STOP

Öğrenci Psikolojisine Dayalı Optimizasyon

Öğrenci Psikolojisine Dayalı Optimizasyon (ÖPDO) algoritması Hindistan'ın Batı Bengal bölgesindeki öğrencilerin psikolojileri üzerine yapılan çalışmalar sonucu elde edilen veriler üzerine ortaya çıkmıştır (Das ve ark., 2020). ÖPDO algoritması, bir sınıfın en iyisi olmaya çabalayan bir öğrencinin psikolojisine dayanarak önerilmiştir. Öğrenciler bir sınavda başarılı notlar almak ister fakat bu öğrencilerin verimliliğine ve derse olan ilgilerine bağlıdır. Bu psikolojiye bağlı olarak öğrenciler bir sınıf içerisinde en başarılı öğrenci olmak için çabalarlar (Das ve ark., 2020). Algoritma 5' te ÖPDO'nun kaba kodu verilmiştir.

ALGORTİMA 5
START
Popülasyonu ve yakınsama kriterini oluştur.
Sınıfın başlangıç performansını hesapla.
WHILE *P <= MaxIt*
FOR1 *m=1:önerilen konu sayısı*
// Öğrencilerin katagorilerinin kontrolü
IF1 *öğrenci=en iyi*
En iyi öğrencinin performansını güncelle.
ELSE IF2 *öğrenci=iyi*
//her öğrenci için, öğrencinin yalnızca en iyi öğrenciyi takip edip etmediğinin kontrolü.
IF3 *Evet*
Mevcut öğrencinin performansını en iyi öğrencinin performansı doğrultusunda güncelleştir.
ELSE
Mevcut öğrencinin performansını en iyi öğrencinin performansı ve tüm öğrencilerin performanslarının ortalaması doğrultusunda güncelleştir.
END IF3
END ELSE IF2
ELSE IF4 *öğrenci=ortalama*
Mevcut öğrencinin performansını tüm öğrencilerin performanslarının ortalaması doğrultusunda güncelleştir.
END ELSE IF4
ELSE
Mevcut öğrencinin performansını gelişigüzel değerler doğrultusunda güncelleştir.
END IF1
Sınırları kontrol et.
Sınıfın yeni performansını hesapla.
IF5 *yeni performans> Eski performans*
Eski performansı yeni performans ile değiştir
ELSE
Eski performansı koru
END IF5
m=m+1
END FOR1
P=P+1
END WHILE
STOP

Tunik Sürüsü Optimizasyonu

Tunik Sürüsü Optimizasyonu (TSO) bir deniz canlısı olan tuniklerin yön bulma ve beslenme davranışlarından esinlenen sürü tabanlı bir algoritmadır (Kaur ve ark., 2020). Tuniklerin denizdeki besin kaynaklarının yerini bulma yetenekleri vardır. Burada besin kaynaklarını bulmak için tunikler jet itme hareketi ve sürü zekâsı olmak üzere iki davranış sergiler. Optimizasyon için jet itme hareketi matematiksel olarak modellenmiştir. Bunun için arama ajanlarının birbirleri ile çatışmalarını engellemek, en iyi arama ajanına doğru hareket etmek

Int. J. Pure Appl. Sci. 7(1):100-112 (2021)

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

ve en iyi ajana yakın kalmak gibi üç şart sağlanmalıdır. Sürü davranışı da diğer bireylerin en iyi çözüme göre kendi konumlarını güncellemeleri ile gerçekleşir (Kaur ve ark., 2020). Algoritma 6' da TSO algoritmasının kaba kodu verilmiştir.

ALGORİTMA 6
START
Başlangış popülasyonunu rastgele tunikler ile oluştur.
Başlangıç parametre değerlerini belirle.
Her bir tunikin uygunluk değerini hesapla.
Uygunluk değeri en iyi olan tuniki $BEST_t$ olarak belirle.
WHILE (Bitim şartı sağlanıncaya kadar)
Tuniklerin jet itme değerini ve sürü davranışlarını hesapla.
Her tunikin konumunu güncelle.
Tuniklerin arama uzayı sınırlarının dışına çıkıp çıkmadığını kontrol et.
IF1 tunik arama uzayı sınırları dışındamı
Tunikin değerlerini sınırlar içinde kalacak şekilde güncelle.
END IF1
Konumu güncellenen tuniklerin uygunluk değerlerini hesapla.
Eğer bir önceki iterasyondaki en iyi uygunluk değerine sahip tunikten ($BEST_t$) daha iyi uygunluk değerine sahip bir tunik mevcut ise, o tuniki popülasyonun en iyi tuniki ($BEST_t$) olarak güncelle.
END WHILE
Uygunluk değeri en iyi tuniki problemin çözümü olarak kabul et.
STOP

DENEYSEL ÇALIŞMA VE SONUÇLAR

Bu bölümde güncel metasezgisel optimizasyon algoritmalarının karşılaştırılması için kullanılan kalite test fonksiyonları anlatılmış ve bu fonksiyonlar kullanılarak elde edilmiş sonuçlar ayrıntılı bir şekilde verilmiştir.

Kullanılan Kalite Test Fonksiyonları

Sezgisel optimizasyon algoritmalarının performanslarını değerlendirebilmek amacıyla literatürde bir çok matematiksel kalite test fonksiyonu mevcuttur. Bu fonksiyonlar gerçek hayattaki mühendislik problemlerinin zorluk ve karmaşıklığına sahiptirler. Bu fonksiyonlar optimizasyon algoritmalarını yakınsama, hassasiyet, sağlamlık ve genel performans açısından değerlendirmek ve karşılaştırmak için sıklıkla kullanılır. Bu kıyaslama fonksiyonlarının doğal işlevleri, karmaşıklıkları ve diğer özellikleri tanımlarından elde edilebilir ve bu fonksiyonlarının zorluk seviyeleri, boyut ve aralık parametreleri değiştirilerek ayarlanabilir (Alatas ve ark., 2009; Kızıloluk ve Özer, 2016).

Tablo 1. *Kullanılan kalite test fonksiyonları*

Fonksiyon	Denklem	Tür	Sınır	Optimum
Sphere (F1)	$\sum_{i=1}^{d} x_i^2$	TM	±100	0
Rosenbrock (F2)	$\sum_{i=1}^{d-1}[100(x_i^2 - x_{i+1})^2 + (1 - x_i^2)]$	TM	±30	0
Step (F3)	$\sum_{i=1}^{d}(x_i + 0.5)^2$	TM	±100	0
Rastrigin (F4)	$10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)]$	ÇM	±5.12	0
Ackley (F5)	$20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)$	ÇM	±32	0
Griewank (F6)	$\sum_{i=1}^{d}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	ÇM	±600	0
Levy (F7)	$\sin^2(\pi y_1) + \sum_{i=1}^{d-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2, y_i = 1 + \frac{x_i - 1}{4}$	ÇM	±10	0
Alpine (F8)	$\sum_{i=1}^{d}\lvert x_i \sin(x_i) + 0.1\, x_i\rvert$	ÇM	±10	0
Quintic (F9)	$\sum_{i=1}^{d}\lvert x_i^5 - 3x_i^4 + 4x_i^3 + 2x_i^2 - 10x_i - 4\rvert$	ÇM	±10	0
Trigonometric (F10)	$\sum_{i=1}^{d} 8\sin^2[7(x_i - 0.9)^2] + 6\sin^2[14(x_i - 0.9)^2] + (x_i - 0.9)^2$	ÇM	±500	0

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

Bu çalışmada 10 adet matematiksel test fonksiyonu kullanılmıştır. Kullanılan kalite fonksiyonlarının denklem, tür, sınır değerleri ve optimum değer bilgileri Tablo 1' de verilmiştir (Jamil ve Yang, 2013). Tablodaki denklemlerde *x* değerleri değişkenleri *d* değeri ise problem boyutunu temsil etmektedir. Bu fonksiyonlardan sphere, rosenbrock ve step Tek Modlu (TM) yani sadece tek bir global optimuma sahip ve geri kalan fonksiyonlar ise Çok Modlu (ÇM) yani tek bir global optimuma, birden çok yerel optimuma sahip.

Deneysel Sonuçlar

Tüm deneysel çalışmalar MATLAB 2020b platformu üzerinde gerçekleştirilmiştir. Çalışmada karşılaştırılan tüm sezgisel algoritmalar için başlangıç popülasyon sayıları 30 ve maksimum iterasyon sayısı 1000 olarak alınmıştır. Fonksiyonların boyut değerleri 10, 30 ve 50 alınarak algoritmalar 3 farklı zorlukta test edilmiştir. Her test fonksiyonu için algoritmaların performansları, 30 bağımsız çalışmada elde edilen sonuçlara göre değerlendirilmiştir. Elde edilen karşılaştırmalı test sonuçları 10 boyutlu fonksiyonlar için Tablo 2' de, 30 boyutlu fonksiyonlar için Tablo 3' te ve 50 boyutlu fonksiyonlar için Tablo 4' te verilmiştir. Tablolarda 30 bağımsız çalıştırmada elde edilen ortalama, en iyi ve standart sapma değerleri verilmiştir. Ortalama ve en iyi değerler algoritmaların global optimuma yakınsama performansını temsil eder. Standart sapma ise algoritmanın kararlı çalışıp çalışmadığını gösterir.

Tablo 2. *10 boyutlu test fonksiyonlarında elde edilen sonuçlar*

		YDAO	EBO	GPİO	GTO	ÖPDO	TSO
F1	Ort.	2.63E-84	3.50E-76	8.02E-25	6.09E-261	**1.45E-289**	7.31E-84
	En iyi	4.43E-94	6.80E-84	1.31E-29	9.87E-281	**2.60E-294**	2.45E-90
	Std.	9.86E-84	1.68E-75	3.12E-24	**0**	**0**	2.01E-83
F2	Ort.	0.06227	2.73034	7.19726	**0.00185**	0.07724	13.49050
	En iyi	6.14E-08	0.03693	6.54867	**5.10E-09**	0.00120	5.11381
	Std.	0.17952	3.29056	0.26851	**0.00695**	0.09705	19.49559
F3	Ort.	3.91E-27	1.31E-32	0.34769	2.05E-32	**0**	1.16185
	En iyi	3.76E-31	**0**	0.15776	**0**	**0**	0.50588
	Std.	8.33E-27	3.80E-32	0.10437	2.91E-32	**0**	0.33068
F4	Ort.	2.89E-09	10.01591	**0**	**0**	**0**	26.04894
	En iyi	**0**	1.98992	**0**	**0**	**0**	10.95320
	Std.	1.55E-08	4.63792	**0**	**0**	**0**	9.05345
F5	Ort.	1.48E-15	0.17039	1.64E-13	**8.88E-16**	6.57E-15	1.05424
	En iyi	**8.88E-16**	4.44E-15	**8.88E-16**	**8.88E-16**	4.44E-15	4.44E-15
	Std.	1.32E-15	0.44129	2.30E-13	**0**	1.74E-15	1.57406
F6	Ort.	**0**	0.04963	**0**	**0**	**0**	0.40022
	En iyi	**0**	**0**	**0**	**0**	**0**	**0**
	Std.	**0**	0.03303	**0**	**0**	**0**	0.25573
F7	Ort.	5.71E-27	0.11120	0.60254	3.16E-30	**1.50E-32**	0.83622
	En iyi	2.10E-30	**1.50E-32**	0.45402	**1.50E-32**	**1.50E-32**	0.18068
	Std.	1.95E-26	0.34592	0.07917	8.44E-30	**0**	1.49648
F8	Ort.	1.25E-08	4.53E-16	4.83E-15	**2.21E-131**	1.04E-29	2.54171
	En iyi	6.52E-45	9.50E-111	3.46E-21	**3.83E-141**	7.30E-71	0.17516
	Std.	3.30E-08	5.88E-16	8.36E-15	**1.18E-130**	3.12E-29	1.61732
F9	Ort.	1.19E-11	1.18E-15	14.40151	1.03E-11	**0**	16.24432
	En iyi	**0**	**0**	10.65436	**0**	**0**	3.80510
	Std.	4.45E-11	5.07E-15	1.83854	5.56E-11	**0**	5.29426
F10	Ort.	7.29E-25	1.15176	13.65040	4.29291	**0**	26.26650
	En iyi	7.04E-30	**0**	4.40142	0.44864	**0**	2.28676
	Std.	3.62E-24	1.28337	3.03316	3.13423	**0**	10.97233

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

Tablo 2' deki 10 boyutlu test fonksiyonlarından elde edilen sonuçlar incelendiğinde F1, F3, F7, F9 ve F10 fonksiyonlarında en iyi ortalama değerleri ÖPDO' nun verdiği görülmektedir. F2, F5 ve F8' de ise GTO en iyi performansı göstermiştir. F4' de GPİO, GTO ve ÖPDO, F6' da ise YDAO, GPİO, GTO ve ÖPDO en iyi sonuçları vermiştir. En kötü performansı ise TSO ve EBO göstermiştir. 10 boyutlu test fonksiyonlarında toplam 7 fonksiyonda en iyi değerleri veren ÖPDO ilk sırada yer alırken onu 5 fonksiyonda en iyi değerleri vererek GTO takip etmiştir.

Tablo 3' teki 30 boyutlu test fonksiyonlarından elde edilen sonuçlar incelendiğinde toplam 7 fonksiyonda en iyi ortalama ve standart sapma değerlerini veren ÖPDO yine en iyi performansı göstermiştir. GTO ise toplam 4 fonksiyonda en iyi sonuçları vermiş ve en başarılı ikinci algoritma olmuştur. Bu iki algoritmayı YDAO ve GTO toplam ikişer fonksiyonda en iyi değerleri vererek takip etmiştir. En kötü performansı ise EBO ve TSO göstermiştir.

Tablo 4' teki 50 boyutlu test fonksiyonlarından elde edilen sonuçlar incelendiğinde yine ÖPDO toplam 7 fonksiyonda en iyi değerleri vererek en başarılı algoritma olmuştur. GTO ise toplam 4 fonksiyonda, YDAO ve GPİO ise toplam 2 fonksiyonda en iyi sonuçları vermiştir. 50 boyutlu test fonksiyonlarında da yine en kötü başarıyı EBO ve TSO göstermektedir.

Tablo 3. *30 boyutlu test fonksiyonlarında elde edilen sonuçlar*

		YDAO	EBO	GPİO	GTO	ÖPDO	TSO
F1	Ort.	1.18E-46	1.66E-15	4.60E-22	2.11E-241	**5.44E-273**	1.66E-47
	En iyi	2.40E-75	1.35E-17	3.74E-26	2.68E-259	**4.10E-276**	1.16E-50
	Std.	6.34E-46	5.94E-15	1.10E-21	**0**	**0**	2.89E-47
F2	Ort.	**0.10381**	59.63437	27.43257	20.81292	1.69222	28.37646
	En iyi	**4.58E-05**	4.01466	26.85205	17.40447	0.00169	26.17606
	Std.	0.43619	59.80774	**0.20578**	1.61170	4.74616	0.70659
F3	Ort.	9.48E-09	1.88E-15	4.46893	2.92E-09	**0**	3.75670
	En iyi	1.71E-12	1.32E-17	4.00592	4.18E-12	**0**	2.30009
	Std.	2.66E-08	3.84E-15	0.19747	7.99E-09	**0**	0.70124
F4	Ort.	0.00392	49.91371	**0**	**0**	**0**	163.36447
	En iyi	2.27E-11	29.84875	**0**	**0**	**0**	92.62398
	Std.	0.01712	13.57688	**0**	**0**	**0**	36.22485
F5	Ort.	3.49E-15	0.86503	3.27E-12	**8.88E-16**	3.42E-14	1.37351
	En iyi	**8.88E-16**	1.45E-09	1.87E-14	**8.88E-16**	2.22E-14	7.99E-15
	Std.	1.57E-15	1.05580	3.58E-12	**0**	4.72E-15	1.48571
F6	Ort.	**0**	0.01066	**0**	**0**	**0**	0.01105
	En iyi	**0**	**0**	**0**	**0**	**0**	**0**
	Std.	**0**	0.01249	**0**	**0**	**0**	0.01232
F7	Ort.	1.24E-09	4.17400	2.30824	1.38E-08	**1.50E-32**	50.28789
	En iyi	2.73E-12	0.63338	2.10798	1.49E-12	**1.50E-32**	17.30331
	Std.	1.74E-09	2.01820	0.08005	4.92E-08	**0**	19.57959
F8	Ort.	0.00034	1.87E-11	1.43E-13	**1.17E-125**	3.78E-08	26.87974
	En iyi	2.76E-06	1.37E-12	5.70E-16	**3.51E-133**	1.40E-26	17.69641
	Std.	0.00077	2.37E-11	1.36E-13	**4.91E-125**	2.04E-07	5.55862
F9	Ort.	0.00743	8.72E-10	81.66148	0.03955	**8.53E-11**	78.42991
	En iyi	2.18E-05	1.95E-10	74.91313	5.16E-05	**0**	60.14089
	Std.	0.02756	5.53E-10	4.06835	0.11534	**1.69E-10**	10.14987
F10	Ort.	2.05E-08	33.13881	137.04358	24.19155	**0**	155.73646
	En iyi	1.01E-11	8.07758	108.92262	5.38433	**0**	63.76087
	Std.	4.73E-08	17.99387	8.33325	10.50401	**0**	33.66602

Int. J. Pure Appl. Sci. 7(1):100-112 (2021)

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

Tablo 4. *50 boyutlu test fonksiyonlarında elde edilen sonuçlar*

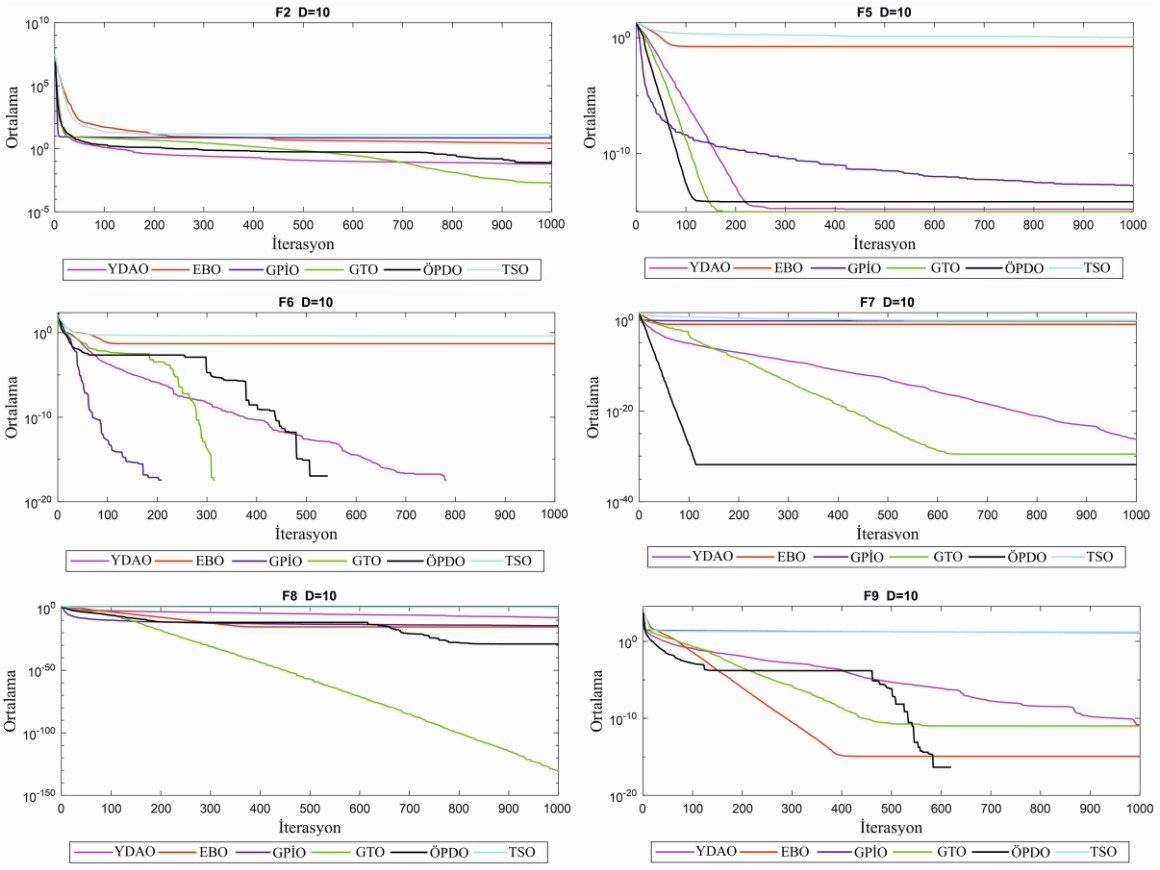
		YDAO	EBO	GPİO	GTO	ÖPDO	TSO
F1	Ort.	1.21E-50	0.00193	2.27E-21	5.70E-241	**3.52E-267**	2.18E-36
	En iyi	5.55E-66	9.26E-07	5.10E-26	5.79E-257	**4.96E-270**	3.02E-39
	Std.	6.54E-50	0.00915	5.22E-21	**0**	**0**	4.23E-36
F2	Ort.	**0.04310**	123.45188	47.58939	42.19588	3.77961	48.39646
	En iyi	0.00076	27.19353	47.03578	39.09889	**0.00040**	46.14923
	Std.	**0.08878**	63.91115	0.40007	2.06084	13.57655	0.60663
F3	Ort.	4.55E-07	0.00011	9.16572	3.48E-06	**0**	6.39201
	En iyi	8.69E-10	4.03E-07	8.55662	3.04E-07	**0**	4.64615
	Std.	9.46E-07	0.00026	0.20253	5.06E-06	**0**	0.88287
F4	Ort.	0.08959	107.62119	**0**	**0**	**0**	383.93784
	En iyi	1.46E-07	69.64706	**0**	**0**	**0**	283.56746
	Std.	0.28393	19.71767	**0**	**0**	**0**	52.43582
F5	Ort.	3.97E-15	2.67599	6.41E-12	**8.88E-16**	6.50E-14	1.08088
	En iyi	**8.88E-16**	0.01144	3.95E-13	**8.88E-16**	5.77E-14	2.22E-14
	Std.	1.21E-15	1.04502	5.96E-12	**0**	4.25E-15	1.43268
F6	Ort.	**0**	0.02702	**0**	**0**	**0**	0.00578
	En iyi	**0**	3.11E-06	**0**	**0**	**0**	**0**
	Std.	**0**	0.06199	**0**	**0**	**0**	0.00800
F7	Ort.	3.28E-07	8.87983	4.11957	7.33E-06	**1.50E-32**	126.87876
	En iyi	3.19E-09	1.99635	3.99599	8.51E-07	**1.50E-32**	44.98305
	Std.	6.14E-07	4.30005	0.07264	1.07E-05	**0**	34.38624
F8	Ort.	0.00242	7.52E-05	4.39E-13	**1.32E-122**	1.18E-06	56.68548
	En iyi	1.75E-05	3.33E-06	1.59E-14	**3.29E-131**	1.90E-23	41.54560
	Std.	0.00249	0.00028	4.72E-13	**4.30E-122**	5.03E-06	9.35117
F9	Ort.	0.32004	0.00920	156.44662	0.62138	**8.38E-05**	152.84064
	En iyi	0.00200	0.00058	146.46686	0.04119	**0**	116.88283
	Std.	0.87080	0.02111	4.08889	0.56817	**0.00033**	18.65439
F10	Ort.	1.86E-06	250.20085	284.58709	53.09751	**0**	324.08870
	En iyi	1.83E-08	51.63057	270.91279	26.47955	**0**	256.90893
	Std.	6.05E-06	376.04332	6.07368	19.73190	**0**	56.81305

Şekil 1, Şekil 2 ve Şekil 3' te sırasıyla 10, 30 ve 50 boyutlu F2, F5, F6, F7, F8 ve F9 fonksiyonlarında elde edilen yakınsama grafikleri görülmektedir. Şekil 1 incelendiğinde F2 için tüm algoritmalar benzer yakınsama göstermektedir. F5' te YDAO, GTO ve ÖPDO yaklaşık 200 iterasyon içerisinde optimum noktaya yakınsadığı fakat geri kalan iterasyonlarda ise yatay seyrettiği görülmektedir. F6' da GPİO, GTO, ÖPDO ve YDAO sırası ile yaklaşık 200, 300, 550 ve 800. iterasyonlarda optimum noktaya ulaşmaktadır. F7' de ÖPDO çok hızlı bir biçimde yaklaşık 100 iterasyonda optimum noktaya ulaştığı görülmektedir. F8' de GTO her iterasyonda sürekli optimum noktaya yakınsarken geri kalan algoritmalar çok daha yavaş bir yakınsama göstermektedir. F9' da ise sadece ÖPDO' nun optimum noktaya ulaşabildiği görülmektedir. Şekil 2 incelendiğinde F2 için tüm algoritmalar ve F8 için GTO hariç diğer algoritmalar benzer yakınsama göstermektedir. F5' te YDAO, GTO ve ÖPDO yaklaşık 300 iterasyon içerisinde hızlı bir biçimde optimum noktaya yakınsadığı fakat geri kalan iterasyonlarda ise yatay seyrettiği görülmektedir. F6' da GTO, YDAO ve ÖPDO ise Şekil 1' deki 10 boyutlu fonksiyona göre 30 boyutlu fonksiyonda daha hızlı bir şekilde yaklaşık 150 iterasyon içinde optimum noktaya yakınsadığı görülmektedir. F7' de yine ÖPDO çok hızlı bir biçimde yaklaşık 130 iterasyonda optimum noktaya yakınsadığı görülmektedir. F9' da EBO her iterasyonda optimum noktaya doğru yakınsarken diğer algoritmalar genelde yatay seyretmiştir. Fakat ÖPDO yaklaşık ilk 850 iterasyonda yavaş bir biçimde optimum noktaya yakınsarken geri kalan iterasyonlarda çok hızlı bir biçimde optimum noktaya yakınsadığı görülmektedir. Şekil 3 incelendiğinde tüm algoritmalar genel olarak Şekil 2' deki 30

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

boyutlu test fonksiyonlarına benzer bir yakınsama göstermişlerdir. Fakat ÖPDO 30 boyutlu F9 fonksiyonunda ilk 850 iterasyonda çok yavaş bir biçimde optimum noktaya yakınsarken geri kalan iterasyonlarda çok hızlı bir biçimde optimum noktaya yakınsamaktaydı. Fakat 50 boyutlu F9 fonksiyonunda her iterasyon yavaş fakat sürekli olarak optimum noktaya yakınsadığı görülmektedir. Tablo 5' te ise algoritmaların 50 boyutlu test fonksiyonlarında 1000 iterasyondaki 30 bağımsız çalıştırmada elde edilen ortalama çalışma süreleri saniye cinsinden verilmiştir. Sonuçlar incelendiğinde YDAO ve TSO' nun süre bakımından en hızlı çalışan algoritmalar olduğu görülmektedir. EBO ve ÖPDO ise en yavaş çalışan algoritmalar olmuştur. Özellikle F7 ve F9 fonksiyonlarında ÖPDO' nun çalışma süresi çok daha uzun sürmektedir.



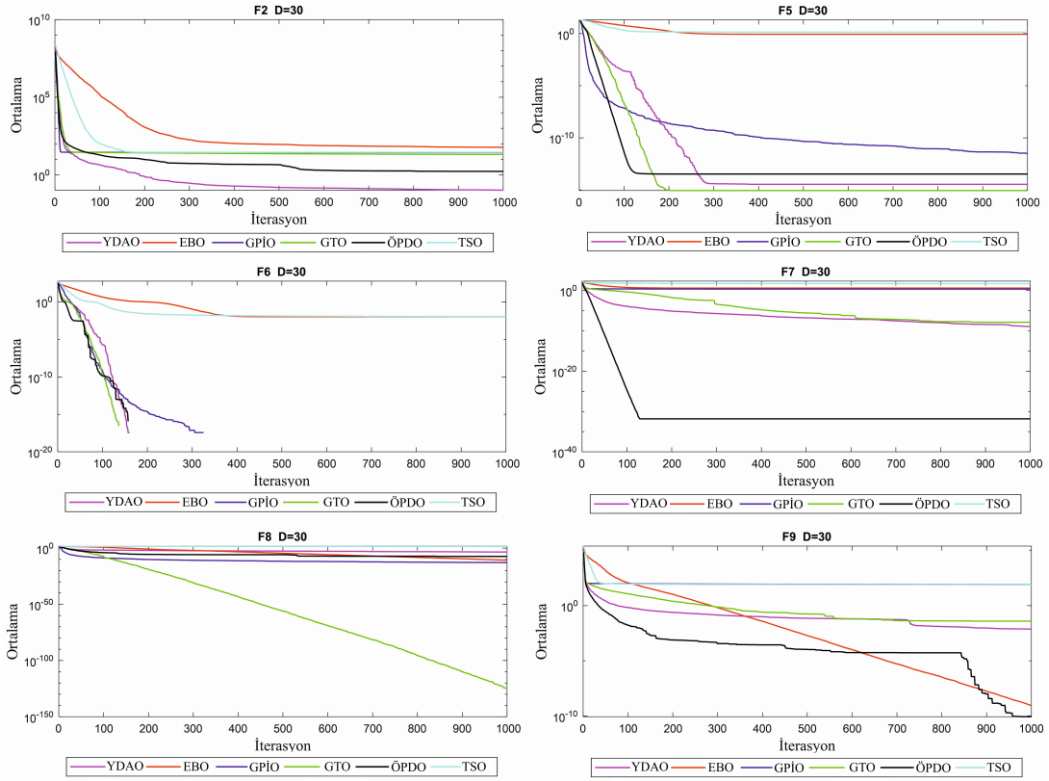
Şekil 1. 10 boyutlu test fonksiyonlarında elde edilen yakınsama grafikleri

Tablo 5. *50 boyutlu test fonksiyonlarında 1000 iterasyonda elde edilen ortalama çalışma süreleri (sn.)*

Fonksiyon	YDAO	EBO	GPİO	GTO	ÖPDO	TSO
F1	0.6006	10.4534	3.6321	2.2497	5.2366	0.7263
F2	0.6591	9.8305	3.6458	1.9316	8.6151	0.8024
F3	0.5626	9.9163	3.3574	1.8068	7.3788	0.7242
F4	0.6317	10.8736	3.7927	1.7773	7.2985	0.7958
F5	0.5997	10.4969	3.3960	1.8451	6.6720	0.7580
F6	0.7381	10.4719	3.4796	1.8589	10.8462	0.8353
F7	1.9460	11.0680	4.6248	3.0221	52.6764	1.8325
F8	0.6420	9.9106	3.3586	1.8649	6.0859	0.8460
F9	1.9099	11.3575	4.9544	3.1950	64.0522	1.9996
F10	0.6496	10.1604	3.5544	1.9391	7.9229	0.8512

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869



Şekil 2. 30 boyutlu test fonksiyonlarında elde edilen yakınsama grafikleri

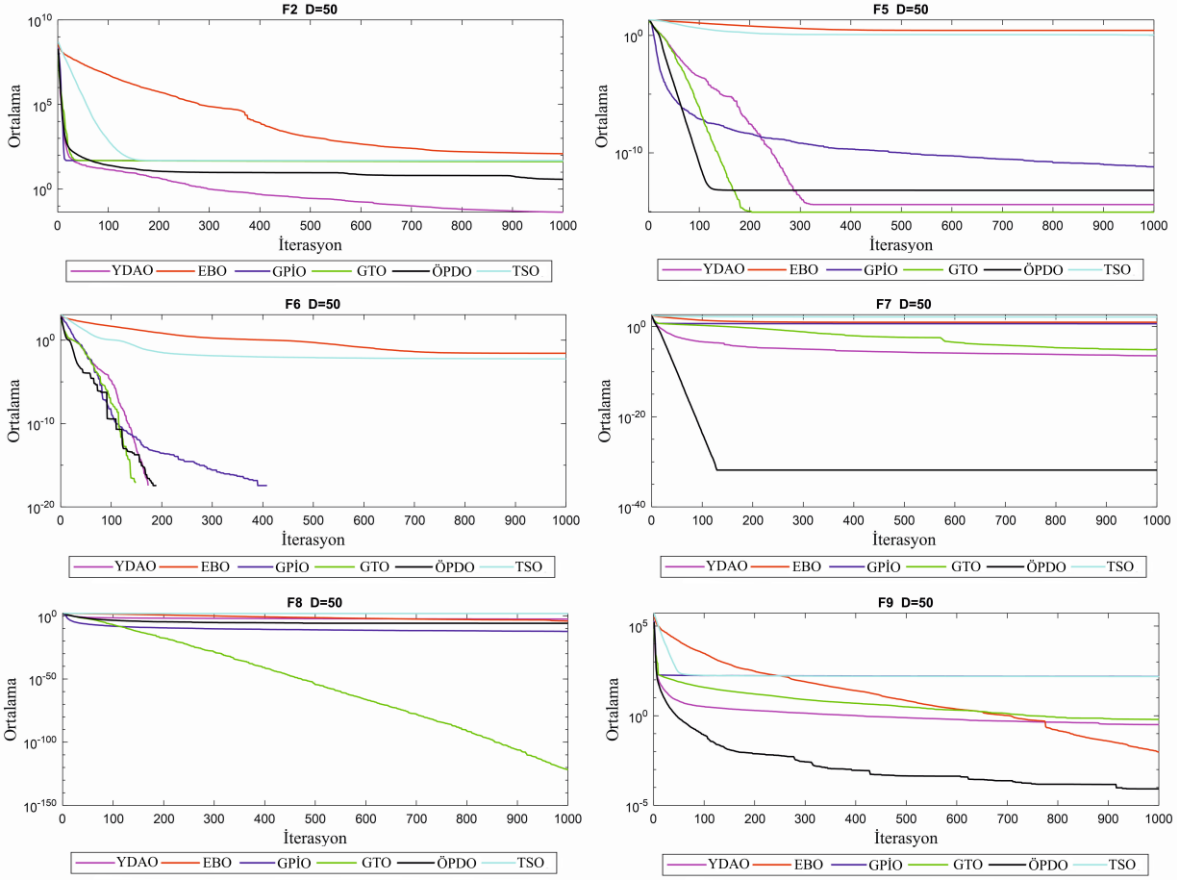
SONUÇ

Metasezgisel algoritmalar birçok optimizasyon problemi için kabul edilebilir sürelerde optimuma yakın çözümler veren, klasik algoritmalar gibi ilgilenilen problemler üzerinde değişiklik yapmadan o problemlere kolaylıkla uyarlanabilen, farklı tipteki karar değişkenleri ve sınırlayıcıların olması durumunda bile çözüm stratejileri sunabilen algoritmalardır. Bu avantajlarından dolayı son birkaç on yıldır araştırmacılar birçok metasezgisel algoritma önermiştir ve halen yeni algoritmalarda literatüre kazandırılmaya devam etmektedir. Bu çalışmada en yeni metasezgisel algoritmalardan olan YDAO, EBO, GPİO, GTO, ÖPDO ve TSO' nun performansları 10, 30 ve 50 boyutlu 10 farklı matematiksel kalite test fonksiyonu kullanılarak ayrıntılı bir biçimde test edilmiştir. Test sonuçlarına göre, bu 10 matematiksel fonksiyondan 7 tanesinde en iyi çözümleri veren ÖPDO' nun en başarılı algoritma olduğu görülmüştür. GTO ise 4 fonksiyonda en iyi sonuçları verdiği görülmüştür. YDAO ve GPİO ise 2'şer fonksiyon en iyi çözümleri vermiştir. En kötü performansı ise EBO ve TSO göstermiştir. ÖPDO her ne kadar en iyi çözümleri üretmişse de çalışma süresi diğer algoritmalara göre daha uzun sürmektedir. Bu durum büyük optimizasyon problemlerinin çözümünde bir dezavantaj olabilir.

Bu çalışma, araştırmacıların en güncel metasezgisel optimizasyon algoritmalarından bazılarına kolaylıkla bir arada ulaşması açısından hem de araştırmacılara kendi çalışmalarında kullanabilecekleri metasezgisel algoritmaların performansları hakkında bir ön fikir vererek uygun algoritmayı seçmesine yardımcı olması açısından önemlidir.

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869



Şekil 3. 50 boyutlu test fonksiyonlarında elde edilen yakınsama grafikleri

ÇIKAR ÇATIŞMASI BEYANI

Yazarlar bu makale ile ilgili herhangi bir çıkar çatışması bildirmemektedir.

ARAŞTIRMA VE YAYIN ETİĞİ BEYANI

Yazarlar bu çalışmanın araştırma ve yayın etiğine uygun olduğunu beyan eder.

KAYNAKLAR

Ahmadianfar, I., Bozorg-Haddad, O. ve Chu, X. (2020). Gradient-based optimizer: A new Metaheuristic optimization algorithm. Information Sciences, 540, 131-159.

Alatas, B. (2012). A novel chemistry based metaheuristic optimization method for mining of classification rules. Expert Systems with Applications, 39(12), 11080-11088.

Alatas, B., Akin, E. ve Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. Chaos, Solitons & Fractals, 40(4), 1715-1734.

Altunbey, F. ve Alataş, B. (2015). Sosyal ağ analizi için sosyal tabanlı yapay zekâ optimizasyon algoritmalarının incelenmesi. International Journal of Pure and Applied Sciences, 1(1), 33-52.

Ashrafi, S. M. ve Dariane, A. B. (2011). A novel and effective algorithm for numerical optimization: melody search (MS). In 2011 11th international conference on hybrid intelligent systems (HIS) (pp. 109-114). IEEE.

Birbil, Ş. İ. ve Fang, S. C. (2003). An electromagnetism-like mechanism for global optimization. Journal of Global Optimization, 25(3), 263-282.

Borji, A. ve Hamidi, M. (2009). A new approach to global optimization motivated by parliamentary political competitions. International Journal of Innovative Computing, Information and Control, 5(6), 1643-1653.

Can, Ü. ve Alataş, B. (2015). Bitki zekâsında yeni bir alan: kök kütlesi optimizasyonu. Türk Doğa ve Fen Dergisi, 8.

Chou, J. S. ve Truong, D. N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. Applied Mathematics and Computation, 389, 125535.

Das, B., Mukherjee, V. ve Das, D. (2020). Student psychology based optimization algorithm: A new population based optimization algorithm for solving

IJPAS
ijpas@munzur.edu.tr
ISSN: 2149-0910

Research article/Araştırma makalesi
DOI: 10.29132/ijpas.855869

optimization problems. Advances in Engineering Software, 146, 102804.

Gao, S. ve De Silva, C. W. (2018). Estimation distribution algorithms on constrained optimization problems. Applied Mathematics and Computation, 339, 323-345.

Harifi, S., Mohammadzadeh, J., Khalilian, M. ve Ebrahimnejad, S. (2020). Giza Pyramids Construction: an ancient-inspired metaheuristic algorithm for optimization. Evolutionary Intelligence, 1-19.

Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.

Jamil, M. ve Yang, X. S. (2013). A literature survey of benchmark functions for global optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation, 4(2), 150-194.

Karaboga, D. ve Akay, B. (2009). A comparative study of artificial bee colony algorithm. Applied Mathematics And Computation, 214(1), 108-132.

Kashan, A. H. (2014). League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships. Applied Soft Computing, 16, 171-200.

Kaur, S., Awasthi, L. K., Sangal, A. L. ve Dhiman, G. (2020). Tunicate swarm algorithm: a new bio-inspired based metaheuristic paradigm for global optimization. Engineering Applications of Artificial Intelligence, 90, 103541.

Kennedy, J. ve Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks (Vol. 4, pp. 1942-1948). IEEE.

Kızıloluk, S. ve Özer, A. B. (2016). Melez elektromanyetizma benzeri-parçacık sürü optimizasyon algoritması. Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi, 7(3), 515-526.

Lee, K. S. ve Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer Methods in Applied Mechanics and Engineering, 194(36-38), 3902-3933.

Mirjalili, S. ve Lewis, A. (2016). The whale optimization algorithm. Advances in Engineering Software, 95, 51-67.

Ong, K. M., Ong, P. ve Sia, C. K. (2021). A carnivorous plant algorithm for solving global optimization problems. Applied Soft Computing, 98, 106833.

Osaba, E., Diaz, F. ve Onieva, E. (2014). Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts.Applied Intelligence, 41(1), 145-166.

Qi, X., Zhu, Y., Chen, H., Zhang, D. ve Niu, B. (2013). An idea based on plant root growth for numerical optimization. In International Conference on Intelligent Computing (pp. 571-578). Springer, Berlin, Heidelberg.

Rao, R. V., Savsani, V. J. ve Vakharia, D. P. (2012). Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Information Sciences, 183(1), 1-15.

Rashedi, E., Nezamabadi-Pour, H. ve Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information Sciences, 179(13), 2232-2248.

Sacco, W. F. ve Oliveira, C. R. D. (2005). A New Stochastic Optimization Algorithm based on a Particle Collision Metaheuristic. 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janerio, Brazil.

Storn, R. ve Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341-359.

Xie, L., Zeng, J. ve Cui, Z. (2009). General framework of artificial physics optimization algorithm. In 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 1321-1326). IEEE.

Yang, X. S. ve Gandomi, A. H. (2012). Bat algorithm: a novel approach for global engineering optimization. Engineering Computations, 29(5), 464-483.

Yang, X. S. (2012). Flower pollination algorithm for global optimization. In International conference on unconventional computing and natural computation (pp. 240-249). Springer, Berlin, Heid