

Çok Amaçlı Metasezgisel Optimizasyon Algoritmalarının Performans Karşılaştırması

Eyüp ERÖZ, Erkan TANYILDIZI

Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği Bölümü, Elazığ
erozeyup@gmail.com

ÖZET

Optimizasyon, karşılaşılan kısıtlı veya kısıtsız problemlerin amaçları doğrultusunda uygun çözümler üretme işlemidir. Optimizasyon işlemini gerçekleştirmek için çeşitli optimizasyon algoritmaları geliştirilmiştir. Tek amaçlı optimizasyon algoritmaları, günlük hayatta karşılaştığımız birden fazla amacı bulunan problemler için yetersiz kaldığından dolayı çok amaçlı optimizasyon algoritmaları geliştirilmiştir. Geliştirilen algoritmalarda, problemler üzerinden en uygun çözüm kümesini bulmak için çeşitli yöntemler kullanılmıştır. Bu yöntemlerden en etkili, yaygın olarak kullanılmakta olan pareto optimal yöntemidir. Pareto optimal yönteminde çok amaçlı optimizasyonun ulaştığı çözümlerden oluşan pareto optimal kümesi, problemlerin tek bir noktadaki çözümlerini değil belli aralıklardaki tüm en iyi çözümlerini içermektedir. Bu çalışmada, literatürde bulunan metasezgisel çok amaçlı optimizasyon algoritmalarından, Çok Amaçlı Karınca Aslanı Optimizasyon Algoritması ile Çok Amaçlı Yusufçuk Algoritması'nın, güncel kıyaslama fonksiyonları ve mühendislik problemleri üzerindeki başarımını görmek için performans karşılaştırılması yapılmıştır.

Anahtar Kelimeler: Çok amaçlı optimizasyon, Metasezgisel, Pareto Optimal, Kıyaslama fonksiyonları

ABSTRACT

Optimization is the process of producing appropriate solutions for the purposes of unconstrained or constrained problems. Various optimization algorithms have been developed to realize the optimization process. As single-objective optimization algorithms are inadequate for problems with more than one purpose in daily life, multiobjective optimization algorithms have been developed. In the developed algorithms, various methods have been used to find the most suitable solution set. The most effective of these methods is the pareto optimal method which is widely used. The pareto optimal set of solutions achieved by multi-objective optimization in the Pareto optimal method includes all the best solutions at certain intervals, not the solutions of the problems at a single point. In this study, performance comparison of Multi-Objective Ant Lion Optimization Algorithm and Multi-Objective Dragonfly Algorithm on current comparative functions and engineering problems were compared.

Keywords: Multi-Objective Optimization, Metaheuristic, Pareto Optimal, Benchmark Functions

I. GİRİŞ

Dünya var olduğundan beri insanlar pek çok problem ile mücadele etmek zorunda kalmıştır. Problemler zaman geçtikçe daha fazla karmaşık ve çözülmesi zor bir hale gelmiştir. Problemlerin karmaşıklığı ve zorluğundan dolayı klasik(deterministik veya stokastik) yöntemler yetersiz kalmaktadır. Bu nedenle uzmanların modelleme, hesaplama ve algoritma geliştirme adına yaptıkları çalışmaların önemi giderek artmaktadır[1].

Klasik yöntemlerin dışında problemleri çözmek için geliştirilen yöntemlerden biri optimizasyon işlemidir. Optimizasyon bir problem üzerinde belirli amaçlar doğrultusunda en uygun çözümler üretme işlemidir. Optimizasyon, gerçek hayatta sıkça karşılaşılan araç rotalama, üretim, yatırım ve tesis planlama, iş ve proje çizelgeleme gibi birçok alanda, mühendislik problemlerinde ve kıyaslama fonksiyonlarına uygun çözümler bulmada önemli rol almaktadır.

Optimizasyon işlemini gerçekleştirebilmek için pek çok optimizasyon algoritması geliştirilmiştir. Geliştirilen algoritmaların çoğu amaç fonksiyonu doğrultusunda

matematiksel model oluşturmaya ihtiyaç duymaktadır. Karmaşık sistemlerde matematiksel modeli oluşturmak zor ve maliyetlidir. Matematiksel model oluşturmaya ihtiyaç duyan algoritmalar, karmaşık problemlere etkili çözüm üretmekte yetersiz kalmaktadır. Aynı durum, tamsayı ya da ayrık karar değişkenlerinin kullanıldığı çoğu doğrusal optimizasyon modelleri için de geçerlidir. Bu nedenle metasezgisel yöntemlere başvurulmuştur.

Metasezgisel yöntemler, deterministik yöntemlerin makul bir sürede çözemediği problemlere yaklaşık bir çözüm bulan yöntemlerdir. Metasezgisel yöntemler problemten bağımsız olarak arama uzayını aktif ve verimli bir şekilde incelemeyi amaçlamaktadır. Bu yöntemler, her zaman global en iyi çözümü bulmayı garanti etmezler ancak oldukça kullanışlıdır. Çünkü büyük ölçekli ve karmaşık problemlere etkin çözümler üretebilmektedirler [2,3].

Genel olarak metasezgisel yöntemler biyoloji, fizik, kimya, matematik, müzik, sosyal ve sürü tabanlı olmak üzere yedi farklı grupta değerlendirilmektedir. Bu grupların birleşmesiyle oluşan melez yöntemler de bulunmaktadır. Bu algoritmalar, doğadan ve farklı bilim dallarından ilham alan

etoloji, biyoloji, fiziksel davranışları taklit ederek sorunları çözmektedir[4].

Günlük hayatta karşılaşılan problemlerin birden fazla amacı olmaktadır. Amaçların çatışması, karmaşıklığı veya büyüklüğü nedeniyle tek amaçlı optimizasyon algoritmaları etkili sonuçlar üretememektedir. Bu nedenle çok amaçlı optimizasyon algoritmaları geliştirilmiştir. Çok Amaçlı Yusufçuk Algoritması(MODA)[5], Çok Amaçlı Karınca Aslanı Optimizasyonu (MOALO) [6], Çok Amaçlı Diferansiyel Evrim (MODE)[7], Çok Amaçlı Bozkurt Optimizer (MOGWO)[8], Çok Amaçlı Parçacık Sürü Optimizasyonu(MOPSO)[9] ve Baskın Olmayan Sıralama Genetik Algoritması II(NSGA)[10] gibi algoritmalar, güncel çok amaçlı optimizasyon algoritmalarına örnek olarak verilebilir[11].

Bu çalışmada geliştirilen MOALO ve MODA metasezgisel optimizasyon algoritmalarının, kıyaslama test fonksiyonları ve optimizasyon problemleri üzerindeki etkisi incelenmektedir. İkinci bölümde çok amaçlı optimizasyon ve pareto optimal kavramları, üçüncü bölümde performans karşılaştırılması yapılan algoritmalar, dördüncü Bölüm 'de algoritmaların performans değerleri tablolar halinde verilmiştir ve son bölümde ise genel sonuç yer almaktadır.

II. ÇOK AMAÇLI OPTİMİZASYON

Çok amaçlı optimizasyon, birden fazla amacın sistematik bir şekilde eş zamanlı olarak optimize edilmesi işlemidir. İnsanlar zaman geçtikçe gerçek hayatta daha fazla zorluklarla karşılaşır. Amaçların birbiriyle çatışması, karmaşıklığı ve çözüm uzayının genişliği çözümü daha da zor hale getirmektedir. Bu nedenle çok amaçlı optimizasyon algoritmaları ön plana çıkarılmıştır. Günlük hayatta üretim, finans, mühendislik gibi birçok alanda çok amaçlı problemler ile karşılaşırız. Bu problemlerin birden fazla amacı olduğu için tek bir optimum çözüm yerine bir çözüm kümesinden bahsedebiliriz. Bu durumda literatürde yer alan tek amaçlı optimizasyon algoritmaları yetersiz kaldığı için, çok amaçlı optimizasyon algoritmaları geliştirilmektedir. Geliştirilen algoritmalar problemlerin optimum değerine yaklaşık çözümler ürettiğinden farklı çok amaçlı optimizasyon algoritmalarına ihtiyaç duyulmaktadır. Geliştirilen algoritmaların verimliliği ölçmek için literatürdeki algoritmalar ile kıyaslama yapılmaktadır. Yapılan kıyaslamalar sonucunda algoritmaların kullanılabilirliği ve problemler üzerindeki başarımları hakkında çıkarımlarda bulunulmaktadır[12].

A. Pareto Optimal

Çok amaçlı optimizasyon problemlerinin çözümü için birden fazla yöntem bulunmaktadır. Bu yöntemler çözüm uzayının yeteri kadar taranabilmesi ve etkin sonuçlar üretilmesi için geliştirilmiş ve algoritmalara adapte edilmiştir. Çok amaçlı optimizasyon problemlerinde en etkin çözüm kümesini sağlayan yöntem pareto optimal

yöntemidir. Çok amaçlı problemin en uygun çözüm kümesini elde etmek için, amaçlar arasında en iyi dengeleri temsil eden çözümlerin kümesi olan Pareto optimal setinin bulunması gerekir. Çok amaçlı optimizasyon problemlerinin çözümü için kullanılan pareto yönteminde dört ana tanım aşağıdaki gibidir:

Tanım 1 (Pareto Baskınlık): Belirtildiği gibi iki vektörün olduğunu varsayalım; $\vec{x} = (x_1, x_2, \dots, x_k)$ ve $\vec{y} = (y_1, y_2, \dots, y_k)$ $\vec{x}$ vektörü $\vec{y}$ vektörüne baskın geliyorsa Denklem 2.1' deki gibi gösterilir[13].

$$\forall i \in \{1, 2, \dots, k\}: f_i(\vec{x}) \leq f_i(\vec{y}) \wedge \exists i \in \{1, 2, \dots, k\}: f_i(\vec{x}) < f_i(\vec{y}) \quad (2.1)$$

Tanım 2 (Pareto Optimal): Bir çözümde $\vec{x} \in X$ ise pareto optimaldir ve Denklem 2.2'deki gibi gösterilir[13].

$$\{\vec{y} \in X \mid \vec{y} < \vec{x}\} \quad (2.2)$$

Tanım 3 (Pareto Çözüm Kümesi): Tüm pareto optimal çözümlere pareto optimal kümesi denir. Denklem 2.3' deki gibi ifade edilmektedir[13].

$$P_s = \{\vec{x}, \vec{y} \in X \mid \vec{y} < \vec{x}\} \quad (2.3)$$

Tanım 4 (Pareto Optimal Front): Pareto çözüm kümesi için nesnel fonksiyonların değerini içeren bir küme Denklem 2.4' deki gibi ifade edilmektedir[13].

$$P_f = \{f(\vec{x}) \mid \vec{x} \in P_s\} \quad (2.4)$$

Çok amaçlı bir problemi çözmek için, amaçlar arasındaki en iyi dengeyi temsil eden çözümler kümesi olan Pareto optimal setini bulmak gerekmektedir[13].

III. PERFORMANS KARŞILAŞTIRILMASI YAPILAN OPTİMİZASYON ALGORİTMALARI

Çok amaçlı optimizasyon algoritmaları sürekli gelişim göstermektedir. Gerek mevcut tek amaçlı optimizasyon algoritmalarının dönüştürülmesi gerekse yeni bulunan bir metasezgisel yöntem üzerine pek çok çok amaçlı optimizasyon algoritması geliştirilmektedir. Geliştirilen algoritmaların performans kriterlerini ölçmek, hangi algoritmanın hangi problemler üzerinde etkili sonuç üretebileceğini görmek üzere literatürde yer alan MODA ve MOALO algoritmaları ele alınacaktır.

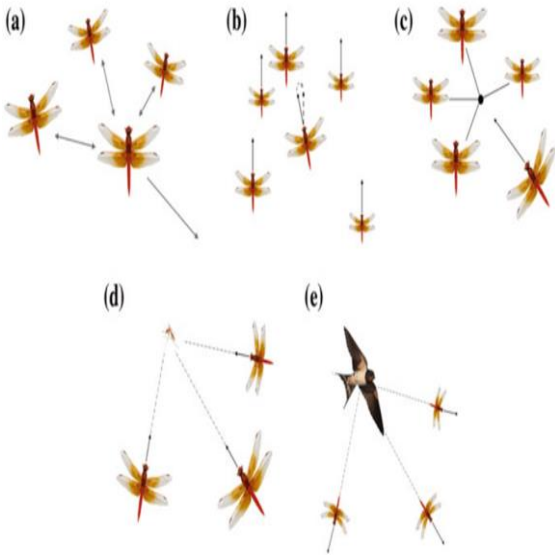
A. MODA

DA, metasezgisel optimizasyon algoritmalarına başka bir örnek olarak verilebilir. Yusufçuklar, doğada neredeyse tüm diğer küçük böcekleri avlayan küçük yırtıcı hayvanlardır. Yusufçukların ilginç özelliği, onların eşsiz ve nadir kaynaştırma davranışlarıdır. Yusufçuklar sadece iki amaç için sürülür: avlanma ve göç. Birincisi statik (besleme) sürüsü olarak adlandırılır ve ikincisi dinamik (göçmen) sürüsü olarak adlandırılır. Statik sürüde, yusufçuklar küçük gruplar yaparlar ve kelebek ve sivrisinekler gibi diğer uçan avları avlamak için küçük bir alan üzerinde ileri geri uçarlar. Yerel hareketler ve uçan yoldaki ani değişimler, statik bir sürünün temel özellikleridir. Bununla birlikte, dinamik

sürüler halinde, çok sayıda yusufoçuk böceğinin uzun mesafelerde bir yönde göç yapmak için sürüyü yapar. DA algoritmasının ana ilham kaynağı, statik ve dinamik kaynaştırma davranışlarından kaynaklanır. Bu iki kaynaştırma davranışı, meta-sezgiselliği kullanarak oluşturulan iki ana optimizasyon evresine çok benzemektedir: keşif ve sömürü. Yusufoçuklar, keşif evresinin ana hedefi olan alt sürülerini yaratır ve statik bir sürünün içinde farklı alanlarda uçarlar. Ancak, statik sürüde, yusufoçuklar büyük sürüler halinde ve sömürü evresinde uygun olan tek yön boyunca uçarlar[5]. Yusufoçuk sürülerinin davranışlarının modellenmesinde 3 temel ilke vardır:

- Ayırma, mahalledeki bireylerin statik çarpışmadan kaçınmasını ifade eder.
- Hizalama, bireylerin çevredeki diğer bireylerin hız uyusmasını gösteren hizalama
- Birlik, bireylerin mahallenin kütlelerinin merkezine doğru eğilimini ifade eder.

Herhangi bir sürünün temel amacı hayatta kalmaktır. Bu nedenle her türlü sürü için ortak iki davranış vardır: yiyecek bulma ve düşmandan kaçınma. Bu iki davranış göz önünde bulundurulduğunda, Şekil 3.1' de gösterildiği gibi sürülerdeki bireylerin pozisyon güncellemesinde beş ana faktör vardır.



Şekil 3.1: Yusufoçukların davranışlarının modellenmesi ((a) Ayırma, (b) hizalanma, (c) birleşme, (d) yemek için cazibe, (e) düşmandan kaçış)

Yusufoçuk algoritmasının bu davranışlarının her biri aşağıdaki gibi matematiksel olarak modellenmiştir: Ayrılma davranışı Denklem 3.1' de görüldüğü gibi hesaplanmaktadır.

$$S_i = - \sum_{j=1}^N X - X_j \quad (3.1)$$

Burada, X mevcut bireyin konumunu, X_j komşu bireyin j . pozisyonunu ve N komşu bireylerin sayısını göstermektedir.

Hizalama davranışı Denklem 3.2'deki gibi hesaplanmaktadır.

$$A_i = \frac{\sum_{j=1}^N V_j}{N} \quad (3.2)$$

Burada, V_j komşu bireyin j . hızını göstermektedir.

Birleşme davranışı Denklem 3.3'deki gibi hesaplanmaktadır:

$$C_i = \frac{\sum_{j=1}^N X_j}{N} - N \quad (3.3)$$

Burada, X mevcut bireyin konumu, N mahallelerin sayısı ve X_j komşu bireyin j . pozisyonunu göstermektedir.

Bir besin kaynağına doğru olan çekim Denklem 3.4'deki gibi hesaplanmaktadır.

$$F_i = X^+ - X \quad (3.4)$$

Burada, X mevcut bireyin konumu ve X^+ gıda kaynağının konumunu gösterir.

Dikkat dağıtıcı bir düşman Denklem 3.5'deki gibi hesaplanmaktadır.

$$E_i = X^+ - X \quad (3.5)$$

Burada, X mevcut bireyin konumu ve X^+ düşmanın konumunu göstermektedir.

Yusufoçukların davranışlarının bu beş matematiksel modelin kombinasyonu olduğu varsayılmaktadır. Yapay yusufoçukların pozisyonlarını yeni arama alanına göre güncellemek ve hareketlerini simüle etmek için iki vektör dikkate alınmaktadır: adım (ΔX) ve pozisyon (X). Adım vektörü, yusufoçukların hareketinin yönünü gösterir ve Denklem 3.6'deki gibi hesaplanmaktadır.

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + wX_t \quad (3.6)$$

Burada, s ayırma ağırlık katsayısıdır, S_i, i . bireyin ayrılmasını belirtmektedir. a hizalama ağırlık katsayısıdır, $A_i i$. bireyin hizalanmasıdır. c , birleşme ağırlığını gösterir, $C_i i$. bireyin birleşmesidir. f besin faktörü, $F_i i$. bireyinin besin kaynağıdır. e düşman faktördür, E_i, i . düşmandır. w eylemsizlik ağırlığıdır ve t yineleme sayacını ifade etmektedir.

Adım vektörü hesaplandıktan sonra, pozisyon vektörleri, denklem 3.7' deki gibi hesaplanmaktadır. Yusufoçukların pozisyonu, Denklem 3.8 ve Denklem 3.9 kullanılarak güncellenmektedir.

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (3.7)$$

$$X_{t+1} = X_t + Levy(d) \times X_t \quad (3.8)$$

$$Levy(x) = 0.01 \times \frac{r_1 \times \sigma}{|r_2|^{\frac{1}{2}}} \quad (3.9)$$

Bahsedilen yusuřuk algoritması çok amaçlı optimizasyon problemleri üzerinde yetersiz kaldığı için, pareto optimal yöntemi ile birleştirilerek çok amaçlı optimizasyon algoritması olan MODA geliştirilmiştir. MODA' nın sözde kodu Tablo 3.1' deki gibidir.

Tablo 3.1: MODA nın sözde kodu

```

Yusuřuk popülasyon giriş deęerleri
Xi (i = 1, 2, ..., n)
Adım vektörleri başlat ΔXi (i = 1, 2, ..., n)
Maksimum hiper kürelerin sayısını tanımla
Arşiv boyutunu tanımla
while durdurma kriteri
    Tüm yusuřukların nesnel deęerlerini hesapla
    Baskın olmayan çözümleri bul
    Arşivi, baskın olmayan çözümlere göre güncelle
if arşiv dolu ise
    Mevcut arşiv üyelerinden birini çıkarmak için arşiv
    bakım mekanizmasını çalıştırın.
    Arşive yeni çözümü ekleyin
end if Arşivdeki yeni eklenen çözümlerden herhangi biri
    hiper alanların dışındaysa
    Yeni çözümlerin tümünü kapsayacak şekilde tüm hiper
    küreleri güncelle ve yeniden konumlandır
end if
    Arşivden bir yiyecek kaynağı seçin:
         $X^+ = SelectFood(arşiv)$ 
    Arşivden bir düşman seçin:
         $X^- = SelectEnemy(arşiv)$ 
    Denklemi kullanarak adım vektörlerini güncelle.
    Denklemi kullanarak konum vektörlerini
    güncelle.
    Deęişkenlerin sınırlarına göre yeni pozisyonları
    kontrol edin ve düzeltin
end while

```

B. MOALO

ALO, karınca aslanlarının avlanma mekanizmalarını ve karıncalarla etkileşimini inceleyerek geliştirilmiş bir metasezgisel algoritmadır. Popülasyon tabanlı dięer algoritmalara benzer şekilde rasgele oluşturulmuş bir dizi çözümleri kullanarak, optimizasyon problemlerine optimum çözümler bulmaya çalışmaktadır. ALO' da iki çeşit popülasyon bulunmaktadır: karıncalar ve karınca aslanları. ALO' nun genel olarak adımları şöyledir[13]:

- Karınca popülasyonu rastgele deęerlerle başlatılır ve ALO' daki temel arama ajanları oluşturulur.

- Her bir karıncaya ait uygunluk deęeri, her iterasyonda hedef fonksiyon kullanılarak hesaplanmaktadır.
- Karıncalar, karınca aslanlarının etrafındaki rasgele yürüyerek arama alanı üzerinde hareket ederler.
- Karınca aslanlarının popülasyonu asla deęerlendirmeye alınmaz. Karınca aslanlarının konumu, ilk iterasyondaki karıncaların bulunduğu yer olduęu varsayılır ve dięer karıncalar daha iyi konumda olursa yer deęiştirilir.
- Her bir karıncaya atanmış bir karınca aslanı vardır ve iterasyon boyunca uygun görülen yerlerde konumları güncellenir.
- Mesafeden bağımsız olarak karıncaların hareketini inceleyen bir elit karınca aslanı vardır.
- Herhangi bir karınca aslanı elit karınca aslanından daha iyi olursa, elit ile deęiştirilir.
- b'den g'ye kadar olan adımlar, durdurma kriteri yerine getirilinceye kadar tekrar tekrar uygulanır.
- Elit karınca aslanının konumu ve uygunluk deęeri, global optimum için en iyi tahmin olarak geri döndürülür.

Karıncaların asıl görevi arama alanını keşfetmektir. Arama alanında rasgele yürüyüşlerle hareket etmektedirler. Karınca aslanları, karıncalar tarafından elde edilen en iyi pozisyonu korur ve karınca arama için gelecek vaat eden yerlere doğru yönlendirilir. ALO, optimizasyon problemlerini çözmek için, karıncaların rastgele dolaşmasını, bir karınca aslanı çukuruna tutunmayı, bir çukur inşa etmeyi, karınca aslanlarına doğru kaymayı, çukuru avlamayı ve çukuru yeniden inşa etmeyi ve elit karınca aslanı seçimini taklit eder. Belirtilen adımların her biri için kurulan matematiksel model sonraki paragraflarda sunulmuştur.

ALO algoritmasında karıncaların rastgele yürüyüşü için formül Denklem 3.10' da verilmiştir:

$$X(t) = [0, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \dots, cumsum(2r(t_n) - 1)] \quad (3.10)$$

Burada, *cumsum* kümülatif toplamı hesaplar, *n* maksimum yineleme sayısıdır, *t* rastgele yürüyüş basamağını gösterir

Rastgele yürüyüşün arama alanı içerisinde kalmasını sağlamak ve karıncaların aşırı taşmasını önlemek için, rastgele yürüyüşler aşağıdaki Denklem 3.11 kullanılarak normalleştirilir:

$$X_i^t = \frac{(x_i^t - a_i) \times (d_i^t - c_i^t)}{(b_i - a_i)} + c_i^t \quad (3.11)$$

Burada, c_i^t *t*. iterasyonda minimum *i*. deęişkeni, d_i^t *t*. iterasyonda maksimum *i*. deęişkenini belirtirken, a_i *i*. deęişkenin rastgele yürüme minimumunu ve b_i , *i*. deęişkenin rastgele yürüme maksimumudur.

ALO, karıncaların rasgele yürüyüşlerle karınca aslanının çukuruna sıkışmasını simule etmektedir ve bu işlem Denklem 3.12 ve Denklem 3.13' te verilmiştir.

$$c_i^t = Antlion_j^t + c^t \quad (3.12)$$

$$d_i^t = Antlion_j^t + d^t \quad (3.13)$$

Burada, c^t , t .iterasyonda tüm değişkenlerin minimumunu ve d^t .iterasyondaki tüm değişkenlerin maksimumunu içeren vektörü gösterir. c_i^t , i .karınca için tüm değişkenlerin minimumunu, d_i^t , i .karınca için tüm değişkenlerinin maksimumunu gösterir. $Antlion_j^t$, seçilen j .karınca aslanlarının t .iterasyondaki konumunu gösterir.

Büyük karınca aslanları hayatta kalma sürelerini arttırmak için daha büyük çukurlar oluşturur. Bu işlemin simülasyonu için, ALO uygunluk değerlerine göre karınca aslanlarını seçen rulet tekerleği operatörünü kullanır. Rulet çarkı, daha fazla karınca çekmek için çabalayan karınca aslanlarına yardımcı olur.

Kayan karıncaları karınca aslanlarına benzetmek için Denklem 3.14 ve Denklem 3.15' daki formüller kullanılarak rastgele yürüyüşlerin sınırları uyarlamalı olarak azaltılmalıdır:

$$c^t = \frac{c^t}{I} \quad (3.14)$$

$$d^t = \frac{d^t}{I} \quad (3.15)$$

I bir oran, c^t .iterasyondaki tüm değişkenlerin minimumudur, d^t , t .iterasyondaki tüm değişkenlerin maksimumunu içeren vektörü belirtir.

Yukarıdaki denklemlerde $I = 1 + 10^w \frac{t}{T}$, burada t geçerli iterasyon, T , maksimum iterasyon sayısıdır ve w , geçerli yinelemeye göre tanımlanmaktadır. ($t > 0.1T$ ise $w = 2$, $t > 0.5T$ ise $w = 3$, $t > 0.75T$ ise $w = 4$, $t > 0.9T$ ise $w = 5$ ve $t > 0.95T$ ise $w = 6$).

I denklemindeki w parametresi, doğruluk seviyesine göre ayarlanabilir.

ALO'da son adım, karıncaları yakalamak ve çukuru yeniden inşa etmektir. Denklem 3.16' daki gibi simule edilir:

$$Antlion_j^t = Ant_i^t \text{ if } f(Ant_i^t) < (Antlion_j^t) \quad (3.16)$$

Burada, t mevcut iterasyon sayısını gösterirken, $Antlion_j^t$, seçilen j .karınca aslanlarının t .iterasyondaki konumunu gösterir, Ant_i^t i .karıncanın t .iterasyondaki konumunu gösterir.

ALO' daki son operatör, optimizasyon sırasında oluşan en uygun karınca aslanının saklandığı elitliktir. Elit, tüm karıncaları etkileyebilecek tek karınca aslanıdır. Bu, karınca aslanlarına rastgele yürüyüşlerin, seçilen bir karınca

aslanına veya elit karınca aslanına doğru çekildiği anlamına gelmektedir. İki durumu de dikkate almak için formül Denklem 3.17' de verilmiştir.

$$Ant_i^t = \frac{R_A^t + R_E^t}{2} \quad (3.17)$$

Buradaki Ant_i^t i .karıncanın t .iterasyondaki konumunu gösterir, R_A^t , t .iterasyonda rulet tekerleği tarafından seçilen karınca aslanının etrafındaki rastgele yürüyüş ve R_E^t , t .iterasyonda elit karınca aslanının etrafındaki rastgele yürüyüştür.

Belirtilen matematiksel yöntemlerle oluşturulan ALO algoritması çok amaçlı problemler için geliştirilerek MOALO algoritması literatüre kazandırılmıştır[14]. Pareto yöntemi kullanılarak optimal çözüm üretmeye çalışan MOALO' da arşiv yöntemi kullanılarak bulunan optimum çözümler depolanmıştır. MOALO algoritmasının yakınsaması, ALO algoritmasından miras kalmıştır. Eğer arşivden bir çözüm seçilirse, ALO algoritması kalitesini artırabilir. Ancak arşiv için bir sınır bulunmalı ve çözümler dağıtımı iyileştirecek şekilde arşivden seçilmelidir. Bu yaklaşımda, her bir çözümün çevresi önceden tanımlanmış bir yarıçap dikkate alınarak incelenir. Çevredeki çözümlerin sayısı sayılır ve dağılımın ölçüsü olarak kabul edilir. Çözümlerin arşivdeki dağılımını iyileştirmek iki aşamada gerçekleştirilmektedir. İlk olarak, karınca aslanları en az nüfuslu popülasyona sahip olan çözümlerden seçilmiştir. Bu bağlamda, Denklem 3.18, arşivde bir çözüm seçme olasılığını hesaplamak için kullanılır:

$$P_i = \frac{c}{N_i} \quad (3.18)$$

Buradaki c , bir sabittir ve $c > 1, N_i$, i .çözüm kümesindeki çözümlerin sayısıdır.

İkincisi, arşiv dolduğunda, en kalabalık çözüm kümesi, yeni çözümleri yerleştirmek için arşivden çıkarılır. Arşivden çıkarma olasılığını hesaplamak için Denklem 3.19 kullanılmaktadır.

$$P_i = \frac{N_i}{c} \quad (3.19)$$

Buradaki c , bir sabittir ve $c > 1, N_i$, i .çözüm kümesindeki çözümlerin sayısıdır. ALO' nun çok amaçlı sorunları çözmesini zorunlu kılmak için, çok amaçlı nesnelerin niteliği nedeniyle Denklem 3.16 ile Denklem 3.20 değiştirilmelidir.

$$Antlion_j^t = Ant_i^t \text{ if } f(Ant_i^t) < f(Antlion_j^t) \quad (3.20)$$

Burada, t mevcut iterasyon sayısını gösterirken, $Antlion_j^t$, seçilen j .karınca aslanlarının t .iterasyondaki konumunu gösterir, Ant_i^t i .karıncanın t .iterasyondaki konumunu gösterir.

Çok amaçlı problemlerin çözümü için gerekli başka değişim de Denklem 3.17 üzerinde yapılmaktadır. Arşivden baskın olmayan bir çözüm seçmek için bir rulet çarkı ve

Denklem 3.17 kullanılmaktadır. MOALO' daki operatörlerin geri kalanı ALO' dakilerle aynıdır. Sonuçta, MOALO algoritmasının sözde kodları Tablo 3.2'de gösterilmektedir.

Tablo 3.2: MOALO nun sözde kodu[14]

```

While durdurma kriteri
For her karınca
    Arşivden rastgele bir karınca aslanı seç
    Rulet tekerleğini kullanarak elitleri seç
    Denklem (3.14,3.15) kullanarak c ve d
    güncelle
    Rastgele bir yürüyüş oluştur ve Denklem
    (3.10,3.11) kullanarak normalleştir.
    Denklem 3.17 ile karınca konumunu güncelle
end for
    Tüm karıncaların değerlerini hesapla
    Arşivi güncelle
    if arşiv dolu ise
        Yeni çözümler sunmak için Rulet tekerleği ve
        Denklem 3.19 kullanarak arşivden bazı
        çözümleri sil
    end
    return arşiv

```

IV. İSTATİSTİKSEL SONUÇ

Yapılan çalışmada 11 kısıtlı ve 9 kısıtsız olmak üzere toplam 20 adet çok amaçlı optimizasyon problemi üzerinden karşılaştırma işlemi gerçekleştirilmiştir. Karşılaşma yaparken her bir algoritma 20 adet test fonksiyonu için ayrı ayrı çalıştırılmıştır. Problemlerin her biri bağımsız olarak 30defa çalıştırılmış ve her çalışma 100 iterasyonda gerçekleştirilmiştir. Performans özelliklerini karşılaştırırken literatürde bulunan; General Distance (GD), Reverse Generational Distance (RGD), Spacing (S), Delta, Maximum Spread (MS) metrik ölçütleri kullanılmıştır.

Algoritmaların test edilmesi için Osyczka(OSY), Kursawa(KUR) ve Tanaka(TNK) kıyaslama fonksiyonları kullanılmıştır.

OSY:

Osyczka ve Kundu, Denklem 4.1, 4.2 ile verilen fonksiyonu önermiştir[15]. Burada 4 tanesi doğrusal olmak üzere 6 adet kısıtlama vardır. Kısıtlamaların ve fonksiyonun analizi Şekil 4.1' deki pareto optimal cephesini ortaya çıkarmaktadır[16].

$$\text{Min } f_1(x) = - [25 (x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2] \quad (4.1)$$

$$\text{Min } f_2(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 \quad (4.2)$$

Kısıtlar;

$$C_1(x) = x_1 + x_2 - 2 \geq 0, \quad (4.3)$$

$$C_2(x) = 6 - x_1 - x_2 \geq 0, \quad (4.4)$$

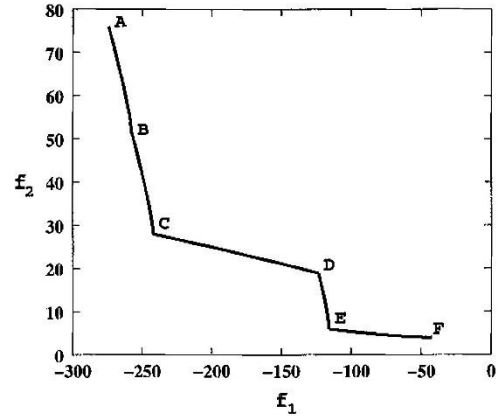
$$C_3(x) = 2 - x_2 + x_1 \geq 0, \quad (4.5)$$

$$C_4(x) = 2 - x_1 + 3x_2 \geq 0, \quad (4.6)$$

$$C_5(x) = 4 - (x_3 - 3)^2 - x_4 \geq 0, \quad (4.7)$$

$$C_6(x) = (x_5 - 3)^2 + x_6 - 4 \geq 0, \quad (4.8)$$

$$0 \leq x_1, x_2, x_6 \leq 10, 1 \leq x_3, x_5 \leq 5, 0 \leq x_4 \leq 6.$$



Şekil 4.1: OSY için kısıtlı pareto optimal cephe [16]

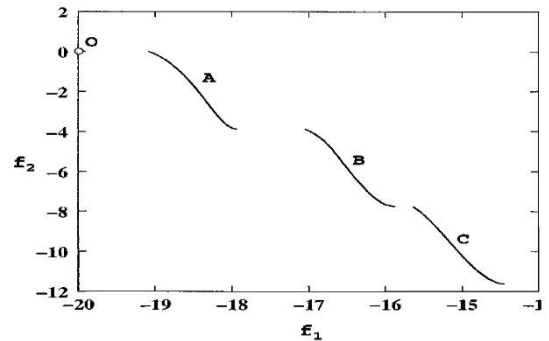
KUR:

1990 yılında Kursawa tarafından kullanılmıştır[17].Formülü Denklem 4.9 ve 4.10 da gösterilmiştir. KUR fonksiyonunun pareto optimal cephesi Şekil 5.2'de gösterilmektedir[16].

$$\text{Min } f_1(x) = \sum_{i=1}^2 \left[-10 \exp \left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right] \quad (4.9)$$

$$\text{Min } f_2(x) = \sum_{i=1}^3 [|x_i|^{0.8} + 5 \sin(x_i^3)] \quad (4.10)$$

$$-5 \leq x_i \leq 5, \quad i = 1, 2, 3.$$



Şekil 4.2: KUR için pareto optimal cephe [16]

3 ayrı Pareto-optimal bölge vardır. Gerçek Pareto-optimal çözümlere karşılık gelen karar değişkeni değerlerinin bu fonksiyon için bilinmesi zordur. Çözüm O, $i=1,2,3$ için $x_i^* = 0$ olan pareto optimum çözümdür. A bölgesindeki pareto optimal çözümler $x_1^* = x_2^* = 0$, B ve C bölgesindeki gibidir $x_1^* = x_3^*$ olan çözümlerdir.

TNK:

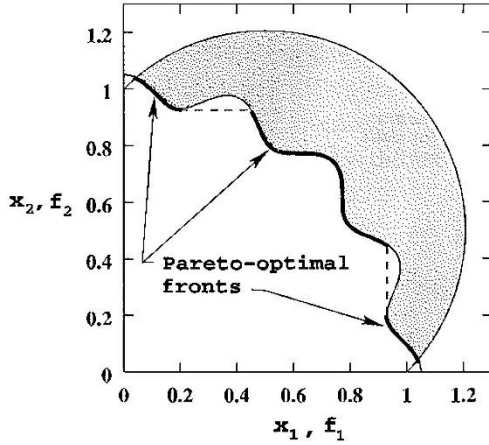
1995 yılında önerilen çok amaçlı bir problem olan Tanaka (TNK), süreksiz bir Pareto optimal cephesine sahip olup iki kısıt ve iki tasarım değişkeni vardır[18]. TNK, problemine ait amaç fonksiyonu ve kısıtlar Denklem 4.11-4.14' te verilmiştir. TNK' ya, ait pareto optimal çözüm kümesi ise Şekil4.3 de verilmiştir.

$$\text{Min } f_1(x) = x_1 \quad (4.11)$$

$$\text{Min } f_2(x) = x_2 \quad (4.12)$$

$$g_1(x) = -x_1^2 - x_2^2 + 1 + 0.1 \cos\left(16 \arctan\left(\frac{x_1}{x_2}\right)\right) \quad (4.13)$$

$$g_2(x) = 0.5 - (x_1 - 0.5)^2 - (x_2 - 0.5) \quad (4.14)$$



Şekil 4.3: TNK için pareto optimal cephe

MOALO ve MODA çok amaçlı optimizasyon algoritmaları 20 adet çok amaçlı optimizasyon problemi üzerinden karşılaştırılmıştır. Performans ölçüm metriklerinin her bir problem üzerinde optimal değerleri Tablo 4.1' de verilmiştir.

Tablo 4.1: Performans ölçütlerinin kullanılan problemler için optimal değerleri

FONKSİYON		OPTİMAL				
		GD	RGD	DELTA	MS	S
F1	BNH2	0	0	0,4507	8,2968	1,0698
F2	CNEX	0	0	0,9488	8,0221	0,0248
F3	CTP0	0	0	0,8573	1,4084	0,0061
F4	CTP8	0	0	0,9488	8,0221	0,0248
F5	CTP9	0	0	0,5268	307,6930	25,3548
F6	FON	0	0	0,3619	1,3881	0,0010
F7	KUR	0	0	0,7571	12,8846	0,0317
F8	MOC2	0	0	1,2886	242,0837	11,6193
F9	MOC4	0	0	0,8573	1,4084	0,0061
F10	OSY	0	0	1,2886	242,0837	11,6193
F11	SRN	0	0	0,5268	307,6930	25,3548
F12	TNK	0	0	0,8573	1,4084	0,0061
F13	VIE	0	0	0,8686	8,4503	0,1948
F14	ZDT1	0	0	0,2776	1,4142	0,0001
F15	ZDT2	0	0	0,2293	1,4142	0,0005
F16	ZDT3	0	0	0,9671	1,9673	0,0012
F17	ZDT4	0	0	0,2770	1,4142	0,0002
F18	ZDT6	0	0	0,1692	0,1692	0,0003
F19	POL	0	0	1,0885	29,6018	0,0233
F20	BEAM	0	0	0,4542	2,6232	0,0028

Test sonucunda elde edilen veriler, MOALO için Tablo 4.2 ve MODA için Tablo 4.3' de verilmiş ve başarılı bulunan değerler koyu renkle gösterilmiştir. MOALO çok amaçlı optimizasyon algoritması için, GD performans ölçüm metriği göz önüne alındığında F2-F4, F9, F12 ve F20 kısıtlı, F6, F13-F16, F18 ve F19 kısıtsız kıyaslama fonksiyonlarında optimum değere yakın sonuçlar elde edilmiştir. RGD, performans ölçüm metriği göz önüne alındığı zaman F2-F5, F9, F11, F12 ve F20 kısıtlı, F6, F13-F16, F18 ve F19 kısıtsız kıyaslama fonksiyonlarında optimuma yakın, F15 kısıtsız kıyaslama fonksiyonunda optimum değer üretilmiştir. DELTA performans ölçüm metriği göz önüne alındığında, F2-F5, F9, F11, F12 ve F20 kısıtlı ve F6, F13, F16, F18 ve F19 kısıtsız test fonksiyonlarında optimuma yakın değer üretilmiştir. MS ve S performans ölçüm metriği göz önüne alındığında neredeyse tüm problemlerde optimuma yakın değere ulaşılmıştır.

MODA çok amaçlı optimizasyon algoritması için, GD performans ölçüm metriği göz önüne alındığında F2-F4, F9, F12 ve F20 kısıtlı, F6, F13 ve F19 kısıtsız test fonksiyonlarında optimum değere yakın sonuçlar elde edilmiştir. RGD, performans ölçüm metriği göz önüne alındığı zaman F2-F4, F9, F12 ve F20 kısıtlı, F6, F13, F15 ve F19 kısıtsız test fonksiyonlarında optimuma yakın değer bulunmuştur. DELTA performans ölçüm metriği göz önüne alındığında, F2-F4, F9, F10, F12 ve F20 kısıtlı ve F6, F13, F16 ve F19 kısıtsız test fonksiyonlarında optimuma yakın değer üretilmiştir. MS ve S performans ölçüm metriği göz önüne alındığında tüm problemlerde optimum değere ulaşılmıştır.

Çok amaçlı problemlerin çözümünde kullanılan algoritmaların iyi sonuç üretmesinin yanı sıra ne kadar sürede elde ettiği de önemlidir. Tablo 4.2 ve Tablo 4.3' deki ortalama süre (T(s)) sütununda, daha kısa sürede sonuca varıldığı fonksiyonlar koyu renkle gösterilmiştir. Zaman açısından karşılaştırdığımızda MOALO F1-F4, F9, F12 ve F20 kısıtlı, F6-F7, F14, F17 ve F19 kısıtsız test fonksiyonlarını daha kısa sürede çalıştırmıştır. MODA ise F5, F10 ve F11 kısıtlı test fonksiyonlarını, F13, F15, F16 ve F18 kısıtsız test fonksiyonlarını daha kısa sürede çalıştırmıştır.

Tablo 4.2: MOALO' nun test sonuçları (EN İYİ: iterasyonlar sonucunda belirlenen değerlerin minimumu, ORT: iterasyonlar sonucunda belirlenen değerlerin ortalaması, EN KÖTÜ: iterasyonlar sonucunda belirlenen değerlerin maksimumu, STD: iterasyonlar sonucunda belirlenen değerlerin standart sapması, T(s): saniye cinsinden ortalama çalışma süresi)

	GD				RGD				DELTA				MS				S				T(s)
	EN İYİ	ORT	EN KÖTÜ	STD	EN İYİ	ORT	EN KÖTÜ	STD	ENİYİ	ORT	EN KÖTÜ	STD	EN İYİ	ORT	EN KÖTÜ	STD	EN İYİ	ORT	EN KÖTÜ	STD	
F1	8,6499	8,6721	8,6815	0,0070	7,8160	8,3028	8,8780	0,2849	0,8219	0,8281	0,8311	0,0024	8,2968	8,2968	8,2968	0,0000	1,0698	1,0698	1,0698	0,0000	21,3792
F2	0,0753	0,1585	0,3828	0,0711	0,0018	0,0064	0,0118	0,0027	0,9493	0,9532	0,9598	0,0030	8,0221	8,0221	8,0221	0,0000	0,0248	0,0248	0,0248	0,0000	9,0025
F3	0,0110	0,0399	0,0749	0,0180	0,0048	0,0083	0,0248	0,0042	0,8578	0,8705	0,8808	0,0064	1,4084	1,4084	1,4084	0,0000	0,0061	0,0061	0,0061	0,0000	4,2711
F4	0,0813	0,1712	0,4532	0,0915	0,0016	0,0053	0,0082	0,0016	0,9493	0,9537	0,9611	0,0034	8,0221	8,0221	8,0221	0,0000	0,0248	0,0248	0,0248	0,0000	9,3933
F5	4,8864	8,0177	20,4503	2,9068	0,0599	0,5279	4,1713	0,8148	0,5500	0,5923	0,6381	0,0181	307,6930	307,6930	307,6930	0,0000	0,0656	0,0656	0,0656	0,0000	27,7968
F6	0,0291	0,0536	0,1603	0,0293	0,0009	0,0014	0,0030	0,0005	0,3970	0,4343	0,5237	0,0288	1,3881	1,3881	1,3881	0,0000	0,0010	0,0010	0,0010	0,0000	13,5647
F7	11,2574	14,9065	15,6080	0,9517	15,0604	15,7229	16,3015	0,2862	0,9881	0,9894	0,9900	0,0004	1,2885	1,2885	1,2885	0,0000	0,0032	0,0032	0,0032	0,0000	7,1647
F8	23,7825	47,1469	99,3943	19,9920	5,4596	24,8199	68,3276	13,9619	1,1803	1,2168	1,2407	0,0153	242,0837	242,0837	242,0837	0,0000	4,2582	4,2582	4,2582	0,0000	6,5613
F9	0,0166	0,0412	0,1138	0,0199	0,0027	0,0092	0,0312	0,0053	0,8610	0,8692	0,8809	0,0053	1,4084	1,4084	1,4084	0,0000	0,0061	0,0061	0,0061	0,0000	3,0747
F10	21,0404	42,9831	97,3997	16,5418	3,0519	21,2435	55,0311	11,5320	1,1859	1,2211	1,2456	0,0147	242,0837	242,0837	242,0837	0,0000	4,2582	4,2582	4,2582	0,0000	6,9244
F11	5,1649	7,2183	10,8213	1,3578	0,0264	0,6371	2,5044	0,6960	0,5468	0,5783	0,6184	0,0193	307,6930	307,6930	307,6930	0,0000	0,0656	0,0656	0,0656	0,0000	27,8779
F12	0,0166	0,0412	0,1138	0,0199	0,0027	0,0092	0,0312	0,0053	0,8610	0,8692	0,8809	0,0053	1,4084	1,4084	1,4084	0,0000	0,0061	0,0061	0,0061	0,0000	3,8980
F13	0,1249	0,2369	0,4319	0,0851	0,0011	0,0068	0,0235	0,0051	0,8687	0,8691	0,8702	0,0003	8,4503	8,4503	8,4503	0,0000	0,1945	0,1945	0,1945	0,0000	120,1182
F14	0,3038	0,4266	0,6288	0,0903	0,0003	0,0059	0,0408	0,0095	0,5080	0,5498	0,5893	0,0221	1,4142	1,4142	1,4142	0,0000	0,0001	0,0001	0,0001	0,0000	18,0007
F15	0,7098	0,7296	0,7309	0,0048	0,0000	0,0000	0,0004	0,0001	0,6032	0,6060	0,6062	0,0007	1,4142	1,4142	1,4142	0,0000	0,0005	0,0005	0,0005	0,0000	11,0605
F16	0,1222	0,2953	0,4694	0,0861	0,0004	0,0121	0,0345	0,0114	0,9722	0,9759	0,9783	0,0014	1,9673	1,9673	1,9673	0,0000	0,0010	0,0010	0,0010	0,0000	16,4227
F17	10,5512	31,4732	64,5637	13,4532	9,9189	35,8849	70,0586	13,9943	0,9523	0,9829	0,9929	0,0097	1,4142	1,4142	1,4142	0,0000	0,0002	0,0002	0,0002	0,0000	6,7164
F18	0,0189	0,2852	2,6162	0,6416	0,0002	0,6791	3,2909	0,9389	0,1729	0,7186	0,8996	0,1602	1,1687	1,1687	1,1687	0,0000	0,0003	0,0003	0,0003	0,0000	9,8200
F19	0,2896	0,7215	1,2356	0,2945	0,0135	0,0434	0,1988	0,0435	1,0748	1,0819	1,0869	0,0034	29,6018	29,6018	29,6018	0,0000	0,0233	0,0233	0,0233	0,0000	9,3564
F20	0,0427	0,0944	0,3104	0,0554	0,0008	0,0015	0,0029	0,0005	0,4627	0,5118	0,6157	0,0399	2,6232	2,6232	2,6232	0,0000	0,0028	0,0028	0,0028	0,0000	13,5258

Tablo 4.3: MODA' nın test sonuçları (EN İYİ: iterasyonlar sonucunda belirlenen değerlerin minimumu, ORT: iterasyonlar sonucunda belirlenen değerlerin ortalaması, EN KÖTÜ: iterasyonlar sonucunda belirlenen değerlerin maksimumu, STD: iterasyonlar sonucunda belirlenen değerlerin standart sapması, T(s):saniye cinsinden ortalama çalışma süresi)

	GD				RGD				DELTA				MS				S				T(s)
	EN İYİ	ORT	EN KÖTÜ	STD	EN İYİ	ORT	EN KÖTÜ	STD	EN İYİ	ORT	EN KÖTÜ	STD	EN İYİ	ORT	EN KÖTÜ	STD	EN İYİ	ORT	EN KÖTÜ	STD	
F1	8,6631	8,6697	8,6751	0,0034	7,8882	8,2688	8,7792	0,2004	0,8239	0,8294	0,8306	0,0013	8,2968	8,2968	8,2968	0,0000	1,0698	1,0698	1,0698	0,0000	34,8928
F2	0,1852	0,2983	0,7460	0,1434	0,0292	0,2241	0,5573	0,1003	0,9508	0,9580	0,9923	0,0116	8,0221	8,0221	8,0221	0,0000	0,0248	0,0248	0,0248	0,0000	11,0731
F3	0,0397	0,0770	0,1264	0,0255	0,0142	0,0365	0,0723	0,0157	0,8616	0,8731	0,8864	0,0073	1,4084	1,4084	1,4084	0,0000	0,0061	0,0061	0,0061	0,0000	16,2973
F4	0,1832	0,2702	0,5356	0,1001	0,1339	0,2306	0,6207	0,0894	0,9506	0,9581	0,9929	0,0103	8,0221	8,0221	8,0221	0,0000	0,0248	0,0248	0,0248	0,0000	10,5108
F5	13,4348	23,3372	75,6211	11,7993	67,3802	236,9578	401,2469	88,0679	0,8338	0,8604	0,8708	0,0085	307,6930	307,6930	307,6930	0,0000	0,0656	0,0656	0,0656	0,0000	25,4848
F6	0,0226	0,0283	0,0362	0,0031	0,0023	0,0055	0,0148	0,0030	0,3637	0,3818	0,4092	0,0099	1,3881	1,3881	1,3881	0,0000	0,0010	0,0010	0,0010	0,0000	23,4202
F7	13,4045	15,2360	15,5850	0,4104	15,2758	15,7576	16,0844	0,1928	0,9892	0,9898	0,9900	0,0002	1,2885	1,2885	1,2885	0,0000	0,0032	0,0032	0,0032	0,0000	23,5906
F8	27,8839	27,8839	62,3500	12,3130	639,0762	880,5214	1098,2782	117,9588	1,0446	1,0457	1,0470	0,0007	242,0837	242,0837	242,0837	0,0000	4,2582	4,2582	4,2582	0,0000	7,3872
F9	0,0372	0,0678	0,1175	0,0236	0,0125	0,0324	0,0677	0,0152	0,8645	0,8729	0,8842	0,0054	1,4084	1,4084	1,4084	0,0000	0,0061	0,0061	0,0061	0,0000	19,5579
F10	14,2273	27,8686	48,7335	7,2412	5,6858	18,4799	44,0008	11,7171	1,1981	1,2273	1,2457	0,0101	242,0837	242,0837	242,0837	0,0000	4,2582	4,2582	4,2582	0,0000	4,2582
F11	13,5848	20,6747	32,5857	5,1879	114,5369	220,5872	383,2920	77,0378	0,8500	0,8615	0,8671	0,0038	307,6930	307,6930	307,6930	0,0000	0,0656	0,0656	0,0656	0,0000	25,6312
F12	0,0360	0,0674	0,1181	0,0212	0,0120	0,0358	0,0653	0,0145	0,8633	0,8729	0,8852	0,0055	1,4084	1,4084	1,4084	0,0000	0,0061	0,0061	0,0061	0,0000	19,0031
F13*	0,1311	0,3041	0,5617	0,1071	0,0031	0,0123	0,0304	0,0063	0,8687	0,8689	0,8695	0,0002	8,4503	8,4503	8,4503	0,0000	0,1945	0,1945	0,1945	0,0000	143,6259
F14	1,2900	1,6329	2,0466	0,1764	1,1344	1,6255	2,1210	0,2464	0,7955	0,8233	0,8521	0,0129	1,4142	1,4142	1,4142	0,0000	0,0001	0,0001	0,0001	0,0000	39,0979
F15	1,2502	2,6381	3,2304	0,4691	0,7340	2,3967	2,9089	0,5191	0,7297	0,8427	0,8664	0,0296	1,4142	1,4142	1,4142	0,0000	0,0005	0,0005	0,0005	0,0000	10,7591
F16	1,0410	1,5122	1,9215	0,2046	1,2783	1,8088	2,3803	0,2668	0,9884	0,9899	0,9908	0,0006	1,9673	1,9673	1,9673	0,0000	0,0010	0,0010	0,0010	0,0000	12,2106
F17	6,3277	18,6782	42,3614	9,3447	5,6311	20,0257	50,7460	12,4949	0,9227	0,9653	0,9894	0,0201	1,4142	1,4142	1,4142	0,0000	0,0002	0,0002	0,0002	0,0000	7,3574
F18	2,9423	5,3501	6,4904	0,8250	3,6662	5,4077	6,4233	0,6455	0,9060	0,9278	0,9387	0,0070	1,1687	1,1687	1,1687	0,0000	0,0003	0,0003	0,0003	0,0000	8,2496
F19	0,4338	2,3239	15,0394	4,2578	0,0142	0,0367	0,0803	0,0153	1,0531	1,0794	1,0876	0,0092	29,6018	29,6018	29,6018	0,0000	0,0233	0,0233	0,0233	0,0000	24,3147
F20	0,0251	0,0486	0,0917	0,0146	0,0008	0,0014	0,0021	0,0003	0,4589	0,4761	0,5202	0,0139	2,6232	2,6232	2,6232	0,0000	0,0028	0,0028	0,0028	0,0000	20,9612

V. SONUÇ

Bu çalışmada literatürde yer alan MOALO ve MODA çok amaçlı optimizasyon algoritmalarının başarımını ölçmek için kısıtlı ve kısıtsız çok amaçlı test fonksiyonları kullanılarak performans karşılaştırılması yapılmıştır. Karşılaştırma sonucunda, MOALO algoritması, GD ölçütüne göre 13 (6 kısıtlı, 7 kısıtsız), RGD ölçütüne göre 15 (8 kısıtlı, 7 kısıtsız), Delta ölçütüne göre 13 (8 kısıtlı, 5 kısıtsız) test fonksiyonunda başarılı sonuç üretmiştir. MODA algoritması ise, GD ölçütüne göre 9 (6 kısıtlı, 3 kısıtsız), RGD ölçütüne göre 10 (6 kısıtlı, 4 kısıtsız), Delta ölçütüne göre 11 (7 kısıtlı, 4 kısıtsız) test fonksiyonunda başarılı sonuç üretmiştir. MS ve S performans ölçütlerinde ise her iki algoritma da tüm test fonksiyonlarında optimuma yakın değer üretmiştir. Çalışma süresi açısından bakıldığında MOALO algoritması 20 test fonksiyonun 13'ünde daha kısa sürede sonuca ulaşmıştır. Hem ölçütler hem de çalışma süresi göz önüne alındığında, seçilen çok amaçlı test fonksiyonları için MOALO algoritmasının daha başarılı olduğu görülmektedir.

Çok amaçlı problemlere optimum çözümler üretmek oldukça zor olduğu için optimuma en yakın olan değeri üretmek önemli bir başarıdır. Literatürde bulunan algoritmalar başarılı sonuçlar üretse de, her problemde aynı başarı sağlayamamaktadır. Bu nedenle eldeki algoritmalar geliştirilmesine veya yeni metasezgisel yöntemlere ihtiyaç duyulmaktadır.

VI. KAYNAKLAR

- [1] K. Murty, "Optimization Models For Decision Making", (http://www.personal.umich.edu/~murty/books/opti_mode1/junior-0.pdf), 2003.
- [2] E. Talbi, "Metaheuristic: Design to Implementation, 2nd Edition", New Jersey: Wiley, 2009.
- [3] Alataş. B., "Kaotik Haritalı Parçacık Sürü Optimizasyonu Algoritmaları Geliştirme" 2007.
- [4] Akyol S, Alataş. B, "Güncel Sürü Zekâsı Optimizasyon Algoritmaları," Nevşehir Üniversitesi Fen Bilimleri Enstitüsü Dergisi 1, pp. 36-50, 2012.
- [5] S. Mirjalili, "Dragonfly Algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems" Neural Comput & Applic 27, pp. 1053-1073, 2016.
- [6] S. M. P. J. S. S., "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems," Appl Intell, DOI 10.1007/s10489-016-0825-8, 2016.
- [7] K. P. Rainer S., "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," Journal of Global Optimization 11, pp. 341-359, 1997.
- [8] S. S. S. M. M. L. S. C. Seyedali M., "Multi-objective grey Wolf optimizer: A novel Algorithm for multi-criterion Optimization," Expert Systems With Applications 47, pp. 106-119, 2016.
- [9] G. P. M. L. C.A.C. Coello, "Handling multiple objectives with particle swarm Optimization," IEEE Transactions on Evolutionary Computation Volume: 8, Issue: 3, 2004.
- [10] K. D. N. Sirinivas, "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms," Journal of Evolutionary Computation, Vol. 2, No. 3, pp. 221-248, 1994.
- [11] A. B., "Kaotik Haritalı Parçacık Sürü Optimizasyonu Algoritmaları Geliştirme, Doktora Tezi", Fırat Üniversitesi Fen Bilimleri Enstitüsü, 2007.
- [12] T. Cigal, «Global Optimizasyonda Yeni Yaklaşımlar, Yüksek Lisans Semineri,» 2016.
- [13] M. S., "The ant lion optimizer", Adv Eng Softw, pp. 80-98, 2015.
- [14] P. J. S. S. Mirjalili, "Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems," Springer Science+Business Media, 10.1007/s10489-016-0825-8, 2016.
- [15] A. a. K. S. Osyczka, "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm," Structural Optimization, pp. 94-99, 1995.
- [16] D. K., Multi-Objective "Optimization using Evolutionary Algorithms", England: Wiley, 2001.
- [17] F. Kursawe, "A variant of evolution strategies for vector optimization," In Parallel Problem Solving from Nature I, pp. 193-197, 1990.
- [18] M. Tanaka, "GA-based decision support system for multi-criteria optimization. In Proceedings of the International Conference on Systems," Man and Cybernetics, Volume 2, pp. 1556-1561, 1995.
- [19] Eröz E., "Çok Amaçlı Optimizasyon Algoritmaları," Yüksek Lisans Semineri, Elazığ, 2018.