Yazılım Mühendisliğine Özel Konular Dönem Projesi, 2022, 1–10

Izzet Emre Şen - 202802069
Güney Kaya - 202802042

MANİSA CELAL BAYAR ÜNİVERSİTESİ

YAZILIM MÜHENDISLIĞINE ÖZEL KONULAR

At Sürüsü Optimizasyon Algoritması: Yüksek boyutlu optimizasyon problemleri için doğadan ilham alan bir algoritma

Farid MiarNaeimi[1], Gholamreza Azizyan[1] and Mohsen Rashki[1]

[1]Civil Engineering & Architecture Engineering Department, University of Sistan and Baluchestan, Zahedan, Iran

Araştırma Özeti

Bu makale, yüksek boyutlu optimizasyon problemleri için atların sürü davranışından esinlenen yeni bir meta-sezgisel algoritma önermektedir. At Sürüsü Optimizasyon Algoritması (HOA) olarak adlandırılan bu yöntem, farklı yaşlardaki atların sosyal performanslarını altı önemli özelliği kullanarak taklit eder: otlama, hiyerarşi, sosyallik, taklit etme, savunma mekanizması ve gezinme. HOA algoritması, şimdiye kadar yapılan çalışmalarda değinilmeyen bu davranışlara dayalı olarak kurulmuştur. Algoritmadaki katsayıların en iyi değerlerini elde etmek amacıyla duyarlılık analizi de yapılmıştır. HOA, farklı yaşlardaki atların davranışlarına dayanan çok sayıdaki kontrol parametreleri sebebiyle, yüksek boyutlardaki karmaşık problemleri çözmede çok iyi bir performansa sahiptir. Önerilen algoritma, çekirge optimizasyon algoritması (GOA), sinüs kosinüs algoritması (SCA), çoklu evren iyileyicisi (MVO), güve-alev iyileyicisi (MFO), yusufçuk algoritması (DA) ve gri kurt iyileyicisi (GWO) dahil olmak üzere doğadan ilham alan popüler optimizasyon algoritmaları ile karşılaştırılmıştır. HOA algoritması doğruluk ve verimlilik açısından, en az işlemsel maliyet ve karmaşıklığa sahip olmasıyla yukarıda bahsedilen popüler optimizasyon algoritmalardan daha iyi performans sergiler.
Key words: Atların Yaşamı; Sürü Zekası; Metasezgisel; Yüksek Boyutlu; Küresel Optimizasyon

1. Giriş

Evrimsel Algoritmalar (EA'lar), Darwin'in evrim ilkelerini taklit eden buluşsal yöntemlerdir. Darwin "Türlerin Kökeni" adlı kitabında iki temel soru sormuştur.

- Bugünkü hayvanlar ve bitkiler tarih öncesi yaratıkların torunları mı?
- Bir tür başka bir türe dönüşür mü?

Bu teori, bu makalede açıklanamayacak kadar geniştir. Ancak, belgelendiği gibi, Darwin bu sorulara verilen olumlu yanıtların farkındaydı. EA'lar, bir çözüm popülasyonunun yinelemeli olarak geliştirildiği (iyileştirildiği) aynı Darwin ilkelerini takip eder.

Sürü Zekası (SI), organizmaların toplu davranışına atıfta bulunan, doğadaki bir başka ilginç zeka türüdür. SI yöntemleri son yıllarda en popüler araştırma konularından biri olmuştur. Çoğunlukla, çeşitli alanlarda karmaşık sorunları çözmek için birçok fırsat sağlayan üç özelliğe (adaptasyon, dağılım ve esneklik) sahip akıllı bir sistemin tasarımına odaklanırlar. Çok sayıda SI algoritması SI'ye dayalıdır ve hayvanların sosyal davranışlarından esinlenmiştir. Organizmalar ve SI arasında 'etkileşim' olarak adlandırılan güçlü bir bağlantı vardır: Canlıların toplu davranışları SI ile sonuçlanır ve SI bazı koşulları değiştirir. Doğadaki bu tür etkileşimli davranışların bazı örnekleri şunlardır:

- Termitler, büyük ve karmaşık yuvalar inşa eder ve bu, tek bir termitin kavrayışının ve yeteneğinin ötesindedir.
- Karınca kolonisindeki çeşitli görevler, merkezi bir yönetim olmaksızın otomatik olarak algılanır.
- Arıların sallanma dansı onları daha fazla yiyecek bulmaya yönlendirir.
- Kuşlar ve balıklar optimal mekansal desenlerde düzenlenmiştir.
- Bakteriler, molekülleri kullanarak çevresel değişiklikleri takip eder.

Hesaplamalı SI yöntemlerinin amacı, daha karmaşık davranışları tespit etmek için canlıların davranışlarını ve çevre ile yerel etkileşimlerini modellemektir.Bu yöntemler, karmaşık sürekli ve ayrık optimizasyon problemlerini çözmek için kullanılabilir.

Çeşitli problemlerin çözümünde meta-sezgisel algoritmaların kullanımı, bu tür algoritmaları anlamanın ve uygulamanın basitliği nedeniyle son yıllarda önemli ölçüde artmıştır.Metasezgisel algoritmaların temelini oluşturan doğal evrim fenomenlerinden bazıları:

birkaç nesil canlının evrimi, metallerde soğutma veya soğutma işlemi, bir kolonideki karıncaların yaşam döngüsü, kuşların göçü, insan savunma sistemi vb.

Optimizasyon problem uzaylarını aramak ve yeterince iyi yanıtları aramak için keşif ve sömürü olmak üzere iki temel özellikten yararlanan farklı optimizasyon problemlerini ele alan çeşitli meta-sezgisel algoritmalar vardır. Keşif, sonuçların doğruluğundan bağımsız olarak bir algoritmanın özgürce arama yapabilme yeteneğidir. Sömürü aynı zamanda bir algoritmanın önceki yineleme döngülerindeki başarılarına ilişkin performansını da ifade eder. Açıktır ki, arama yetenekleri arttığında bir algoritmanın davranışı çoğunlukla rastgele ve tahmin edilemez hale gelir. Bir algoritmadaki gelişmiş sömürü oranı, aksine daha dikkatli bir performansa yol açar. Hemen hemen tüm meta-sezgisel arama yöntemleri bazı ayarlanabilir parametrelere sahip olduğundan, bir algoritmadaki keşif ve yararlanma yetenekleri kontrol edilebilir.

Bu çalışmanın ana katkısı, atların davranışlarını taklit eden yeni bir optimizasyon algoritması sunmaktır. Önerilen algoritmanın basit ve karmaşık tek amaçlı yüksek boyutlu problemlerin çözümünde uygulanabilir olduğu gösterilmiştir. Bu özellik, yedi adet yüksek boyutlu örnek (500, 1000, 2000, 5000 ve 10000 boyut) çözülerek test edilir ve sonuçlar daha sonra mevcut en güçlü optimizasyon algoritmalarının çözümleriyle karşılaştırılır. Bu makalenin geri kalanı şu şekilde özetlenmiştir:

Derinlemesine bir literatür taraması Bölüm 2'de sunulmaktadır. Bölüm 3 HOA algoritmasını, Bölüm 4 ise karmaşıklığını açıklamaktadır. Hesaplanan sonuçlar ve varılan kanı sırasıyla Bölüm 5 ve 6'da verilmektedir.

2. Literatür Taraması

Meta-sezgisel optimizasyon algoritmaları üç ana kategoriye ayrılabilir: Evrimsel algoritmalar, fizik tabanlı algoritmalar ve SI algoritmaları.

İlk kategori doğadaki evrim fikrinden ilham alınmıştır. Bahsedildiği gibi bu algoritmalar Darwin'in teorilerine dayanmaktadır. Bu teori dinamik çevre şartlarına göre organizmanın hayatta kalma yeteneğini arttırmayı hedefleyen bir optimizasyon işlemidir. En iyi canlıların belirlenmesinde hayati rol oynayan faktörlerden biri de yaşanılan çevredir. Başka bir deyişle, belirli bir çevrede yaşayan birkaç organizma türü, birkaç nesil boyunca gelişir. Her canlı, belirli bir çevrede yaşama sorununu (adaptasyon) çözmek için doğadan gelen cevaplanmış bir soru olarak kabul edilir.

İkinci kategori, parçacıkların hareketinin manyetik alanlardaki fizik yasalarından, galaksilerin parçacıkları arasındaki yerçekimi kuvvetlerinden, elektron yükü aktarımından, kimyasal reaksiyonlardan vb. esinlendiği fizik tabanlı optimizasyon algoritmalarını içerir.

Meta-sezgisel optimize edicilerin üçüncü kategorisi, sürü tabanlı algoritmalardır (SI). Bu algoritmalar genellikle, tek bir parçacığın tespit edilemediği ve karşılaştırılamadığı parçacık sürüsünü kullanır. Parçacıklar, grup iletişimini kullanarak yanıtı bulur. Bir parçacık sürüsü, genellikle farklı faaliyetler sırasında birbirleriyle hareket eden ve iletişim kuran bir ajan grubu olabilir. SI'nın problem çözme davranışı genellikle organizmaların sosyal davranışlarının ve etkileşimlerinin incelenmesinden elde edilir.

EA algoritmaları (ve ayrıca EA'lı hibrit algoritmalar), Tablo 1'de belirtildiği gibi Darwin'in teorisinin yasalarına göre önerilmiştir. Ayrıca, bazı SI algoritmaları, yaşam kuralları, avcılık, savunma sistemleri, kanunlardan türetilmiştir. yerçekimi, gradyan tabanlı yöntemler [64,65] vb. Önerilen algoritmaların çokluğuna rağmen, algoritmaların hiçbiri tüm optimizasyon problemlerini çözemez. Bu, Bedava Öğle Yemeği Yok (NFL) teoremi ile mantıksal olarak kanıtlanmıştır. Bu, bir sürüdeki atların sosyal davranışlarını matematiksel olarak modellememiz ve yeni bir SI algoritması için bir girişimdi.

3. At Optimizasyon Algoritması (HOA)

Bu çalışma, atların yaşam ortamlarındaki davranış kalıplarına dayanmaktadır. Atların davranış kalıpları genellikle Otlatma (G), Hiyerarşi (H), Sosyallik (S), Taklit (I), Savunma mekanizması (D) ve Gezinmeyi (R) içerir. Bu algoritma, atların farklı yaşlarda belirtilen altı genel davranışından ilham almıştır. Atlara her iterasyonda uygulanan hareket Eş. (3.1).

$$X_m^{Iter,AGE} = \vec{V}_m^{Iter,AGE} + X_m^{(Iter-1),AGE}, AGE = \alpha, \beta, \gamma, \delta \quad (1)$$

burada, $X_m^{Iter,AGE}$ m'inci atın konumunu gösterir, yaş dikkate alınan atın yaş aralığını gösterir, Iter mevcut yinelemedir ve elma bu atın hız vektörünü gösterir. Atlar farklı yaşlarda farklı davranışlar sergilerler. Bir atın maksimum ömrü yaklaşık 25-30 yıldır [69]. Bu bağlamda δ 0-5 yaş aralığındaki atları, γ 5-10 yaş aralığındaki atları, β 10-15 yaş aralığındaki atları ve α 15 yıldan fazla yaşlı atları göstermektedir. Atların yaşını seçmek için her yinelemede kapsamlı bir yanıt matrisi yapılmalıdır. Bu bağlamda, matris en iyi yanıtlara göre sıralanabilir ve sonuç olarak, sıralanan matrisin üst kısmındaki atların ilk yüzde 10'luk kısmı α atlar olarak seçilir. Sonraki yüzde 20 β grubundadır. γ ve δ atları, kalan atların sırasıyla %30'unu ve %40'ını oluşturur. Atların altı davranışını simüle etmeye yönelik adımlar, hız vektörünü tespit etmek için matematiksel olarak uygulanır. Algoritmanın her çevriminde farklı yaşlarda atların hareket vektörü, yukarıdaki davranış kalıplarına göre Denklem (3.2) olarak yazılabilir. [66,67,69].

$$\vec{V}_m^{Iter,\alpha} = \vec{G}_m^{Iter,\alpha} + \vec{D}_m^{Iter,\alpha} \quad (2)$$

$$\vec{V}_m^{Iter,\beta} = \vec{G}_m^{Iter,\beta} + \vec{H}_m^{Iter,\beta} + \vec{S}_m^{Iter,\beta} + \vec{D}_m^{Iter,\beta} \quad (3)$$

$$\vec{V}_m^{Iter,\gamma} = \vec{G}_m^{Iter,\gamma} + \vec{H}_m^{Iter,\gamma} + \vec{S}_m^{Iter,\gamma} + \vec{D}_m^{Iter,\gamma} + \vec{D}_m^{Iter,\gamma} + \vec{R}_m^{Iter,\gamma} \quad (4)$$

$$\vec{V}_m^{Iter,\delta} = \vec{G}_m^{Iter,\delta} + \vec{I}_m^{Iter,\delta} + \vec{R}_m^{Iter,\delta} \quad (5)$$

3.1. Otlama

Atlar bitkilerle, otlarla, yemlerle vb. beslenen otlayan hayvanlardır. Meralarda günde 16 ila 20 saat otlanırlar ve dinlenme süreleri kısadır. Bu yavaş otlatma yöntemine sürekli yeme denir. Belki kısrakları taylarıyla otlakta otlarken görmüşsünüzdür [69].

HOA algoritması, Şekil 1'e göre her atın belirli alanlarda otladığı gibi, her atın etrafındaki otlatma alanını g katsayısıyla modeller. Atlar, yaşamları boyunca her yaşta otlanır. Otlatmanın matematiksel uygulaması Eş. (3.3) ve (3.4).

$$\vec{G}_m^{Iter,AGE} = g_{Iter}(\check{u} + P\check{l}) \left[X_m^{(Iter-1)}\right], AGE = \alpha, \beta, \gamma, \delta \quad (6)$$

$$g_m^{Iter,AGE} = g_m^{(Iter-1),AGE} X \omega_g \quad (7)$$

Burada, $\vec{G}_m^{Iter,AGE}$, i'nci atın hareket parametresidir ve ilgili atın otlatma eğilimini gösterir. Bu faktör, yineleme başına ω_g ile doğrusallığı azaltır. $\check{l}$ ve $\check{u}$ sırasıyla otlatma alanının alt ve üst sınırlarıdır ve P, 0 ile 1 arasında rasgele bir sayıdır. $\check{l}$ ve $\check{u}$'nun sırasıyla 0.95 ve 1.05'e eşit olması ve } katsayısının tüm yaş aralıkları için 1,5'a eşittir.

3.2. Hiyerarşi (H)

Atlar kendi başlarına özgür değildir [70]. Genellikle insanların üstlendiği bir lider figürünü izleyerek hayatlarını geçirirler. Yetişkin bir aygır veya bir kısrak da hiyerarşi kanununda ortaya çıkan vahşi at sürülerinde liderlikten sorumludur [70]. Bu durumda, HOA'daki h katsayısı, bir at sürüsünün en deneyimli ve en güçlü atı takip etme eğilimi olarak kabul edilir (Şekil 2). Çalışmalar, atların orta yaş β ve γ'da (5-15 yaş arası) hiyerarşi yasasını takip ettiğini göstermiştir [66,67]. Denklem (3.5) ve (3.6) olarak tanımlanabilir.

$$\vec{H}_m^{Iter,AGE} = h_m^{Iter,Age}\left[X_*^{(Iter-1)} - [X_m^{(Iter-1)}\right], AGE = \alpha, \beta and \gamma \quad (8)$$

$$h_m^{Iter,AGE} = h_m^{(Iter-1),AGE} X \omega_h \quad (9)$$

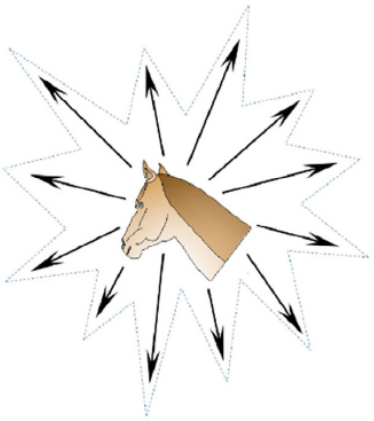


Fig. 1. Simulation of horse grazing in environment.

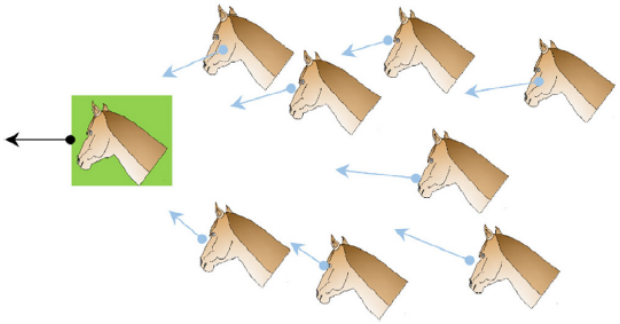


Fig. 2. Hierarchy of horses and simulation.

Burada, $\vec{H}_m^{Iter,AGE}$ en iyi atın konumunun hız parametresi üzerindeki etkilerini ve X_*^{Iter-1} en iyi atın konumunu gösterir

3.3. Sosyallik (S)

Atlar sosyal bir yaşam gerektirir ve bazen diğer hayvanlarla birlikte yaşarlar. Yırtıcı hayvanlar tarafından avlandıkları için sürü yaşamı, atların güvenliğini garanti altına almıştır. Çoğulculuk hayatta kalma şansını arttırır ve kaçmayı kolaylaştırır. Atların sosyal özelliklerinden dolayı birbirleriyle kavga ettiklerini ve sinirliliklerinin bir nedeninin atın tekilliği olduğunu sıklıkla görebilirsiniz. Bazı atlar, sığır ve koyun gibi diğer hayvanların yanında mutlu görünüyor; ancak nadiren yalnızlıktan hoşlanırlar [69].

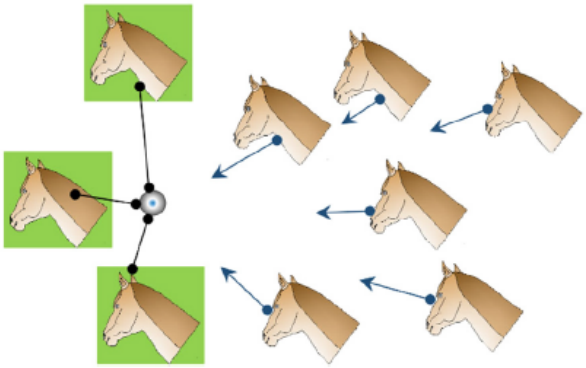


Fig. 4. Horses' imitation and simulation.

Bu davranış, diğer atların ortalama konumuna doğru bir hareket olarak kabul edilir ve Şekil 3'teki s faktörü ile gösterilir. Aşağıdaki denklemde ifade edildiği gibi, 5-15 yaşlarındaki atların sürüye ilgi duyduğu açıkça görülmektedir:

$$\vec{S}_m^{Iter,AGE} = s_m^{Iter,AGE}\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{(Iter-1)}\right) - X_m^{(Iter-1)}\right], \quad AGE = \beta, \gamma$$

$$s_m^{Iter,AGE} = s_m^{(Iter-1),AGE} \times \omega_s$$

Denklemdeki (3.7) $\vec{S}_m^{Iter,AGE}$, i'nci atın sosyal hareket vektörünü ve $s_m^{Iter,AGE}$, ilgili atın *Iter*'inci iterasyonda sürüye doğru yönelimini gösterir. $s_m^{Iter,AGE}$ her döngüde bir ω_s faktörü ile azalır. N ayrıca toplam at sayısını gösterir ve *AGE*, her atın yaş aralığıdır. Parametrelerin duyarlılık analizinde β ve γ atları için s katsayısı hesaplanır.

3.4. Taklit

Atlar birbirlerini taklit ederler ve uygun otlak yeri bulmak gibi birbirlerinin iyi ve kötü huylarını öğrenirler [70].

Atların taklit davranışı da mevcut algoritmada faktör i olarak kabul edilir. Denklemlerde (3.9) ve (3.10) tarif edildiği gibi genç atlar başkalarını taklit etmeye çalışırlar ve bu özellik tam olgunlukları boyunca kaybolmaz.

$$\vec{I}_m^{Iter,AGE} = i_m^{Iter,AGE}\left[\left(\frac{1}{pN}\sum_{j=1}^{pN} \hat{X}_j^{(Iter-1)}\right) - X^{(Iter-1)}\right], \quad AGE = \gamma$$

$$i_m^{Iter,AGE} = i_m^{(Iter-1),AGE} \times \omega_i$$

Denklem (3.9) ve (3.10)'da, $\vec{I}_m^{Iter,AGE}$, $\hat{X}$ konumlu en iyi atların ortalamasına doğru i'nci atın hareket vektörüdür. pN, en iyi konumlara sahip atların sayısını gösterir. p'nin atların %10'u olarak ayarlanması önerilmektedir. Ayrıca, ω_i, daha önce gösterildiği gibi i_{Iter} için döngü başına bir azaltma faktörüdür.

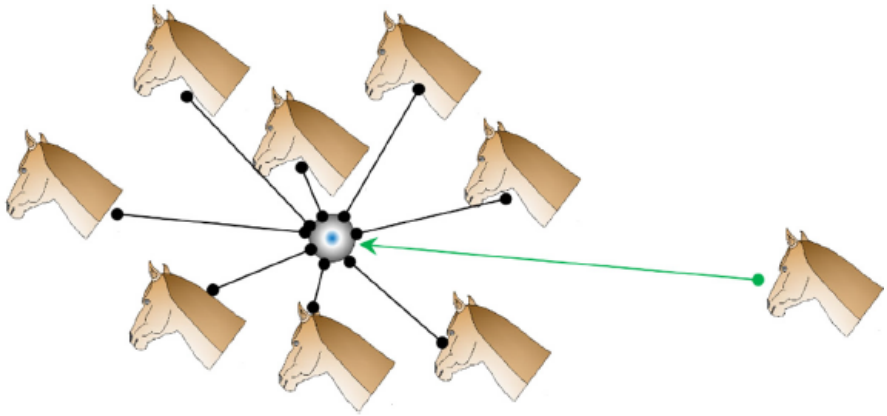


Fig. 3. General movement of horses, their sociability, and simulation.

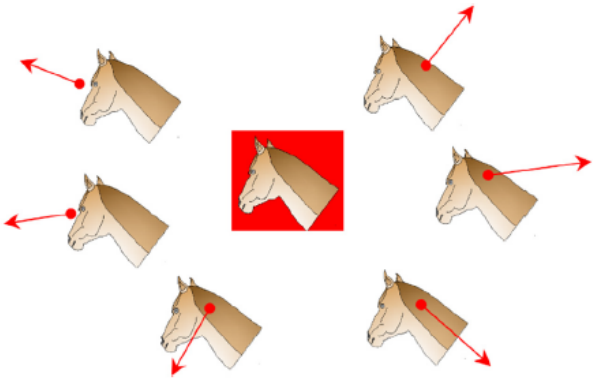


Fig. 5. Horses' defense mechanism and simulation.

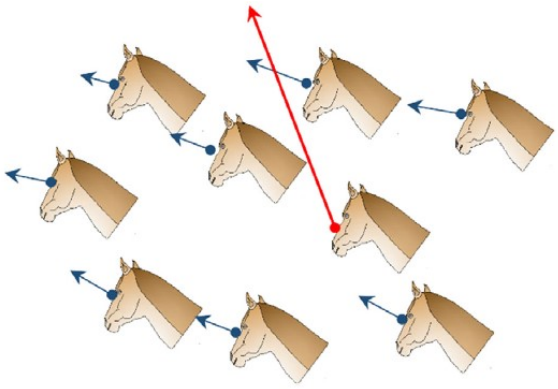


Fig. 6. Horses' roam and simulation.

3.5. Savunma Mekanizması (D)

Atların tepkisi, yırtıcı hayvanların kurbanı oldukları gerçeğinin bir yansımasıdır [66]. Savaş ya da kaç tepkisini göstererek kendilerini savunurlar. İlk tepkileri kaçmak olur. Ayrıca, tuzağa düşme durumunda geri tepirler. Atlar, içgüdüsel olarak kurtlar gibi düşmanların olduğu tehlikeli ortamlardan kaçınmak ve rakiplerini uzaklaştırmak için yiyecek ve su için savaşırlar [66,69].

HOA algoritmasındaki atların savunma sistemi, Şekil 5'e göre optimal olmaktan uzak uygunsuz tepkiler gösteren atlardan kaçarak çalışır. Savunma sistemleri d faktörü ile karakterize edilir. Atlar, yukarıda bahsedildiği gibi kaçmak veya düşmanlarına karşı savaşmak zorundadır. Böyle bir savunma mekanizması, mümkün olduğunda, genç veya olgun bir atın tüm yaşamı boyunca mevcuttur. Atların savunma mekanizması, atı uygun olmayan konumlardan uzak tutmak için Denklemlerde negatif bir katsayı ile sunulur.

$$\vec{D}_m^{\text{Iter},AGE} = -d_m^{\text{Iter},AGE}\left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\check{X}_j^{(\text{Iter}-1)}\right) - X^{(Iter-1)}\right]$$

$$AGE = \alpha, \beta \text{ and } \gamma$$

$$d_m^{\text{Iter},AGE} = d_m^{(Iter-1),AGE} \times \omega_d$$

Burada $\vec{D}_m^{Iter,AGE}$, $\hat{X}$ vektörü ile gösterilen en kötü konumlara sahip bazı atların ortalamasından i'nci atın kaçış vektörünü gösterir. qN ayrıca en kötü konumlara sahip atların sayısını da gösterir. q'nun toplam atların yüzde 20'sine eşit olduğu öne sürülüyor. ω_d, daha önce belirtildiği gibi d_Iter için döngü başına azaltma faktörünü gösterir.

3.6. Gezinme (G)

Atlar yiyecek bulmak için meradan meraya doğada dolaşırlar ve otlanırlar [66]. Bahsedilen özelliği muhafaza etmelerine rağmen çoğu at ahırlarda tutulur. Bir at otlatmak için aniden başka bir yere gidebilir. Atlar son derece meraklıdır ve yeni meralar keşfetmek ve mahallelerini tanımak için sık sık her yeri ziyaret ederler. Yan duvarlar atların birbirlerini görebilecekleri ve meraklarını uygun bir ahırda giderebilecekleri şekilde tasarlanmıştır [66].

Bu davranış rastgele bir hareket olarak simüle edilir ve r faktörü ile gösterilir. Atlarda gezinme hemen hemen genç yaşlarda görülür ve olgunluğa eriştikçe yavaş yavaş kaybolur. Bu süreç ayrıca Şekil 6, Denk. (3.13) ve (3.14) gösterilmiştir.

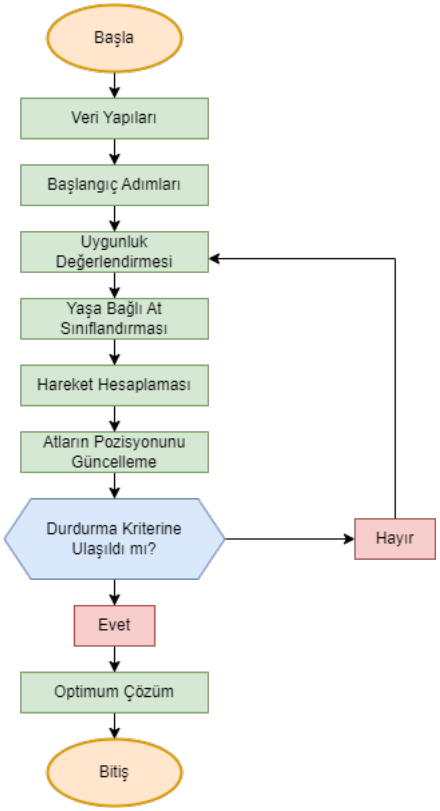
$$\vec{R}_m^{\text{Iter},AGE} = r_m^{\text{Iter},AGE}\mathcal{P}X^{(\text{Iter}-1)}), \quad AGE = \gamma, \delta$$

$$r_m^{\text{Iter},AGE} = r_m^{(\text{Iter}-1),\text{AGE}} \times \omega_r$$

Burada, $\vec{R}_m^{Iter,AGE}$, yerel arama ve yerel minimumlardan kaçış için i. atın rastgele hız vektörünü temsil eder ve ω_r, r_{Iter}, döngü başına $r_m^{Iter,AGE}$'nin azaltma faktörünü gösterir.

Algoritma Akışı

A. Temel Yapı: Sınır değişkenleri ve diğer parametreleri belirle.
B. Başlangıçta: Uygun bir alanda atların rastgele dağılımını sağlamak.
C. Uygunluğun kontrolü: Her atın maliyetinin, konumuna ve amaç fonksiyonuna göre hesaplanıp kontrol edilmesi.
D. Yaş belirlenmesi: α, β, γ ve (δ) atlarının yaşını belirle.
E. Hız denklemini uygula: Atların yaşına göre hızlarını hesapla.
F. Yeni pozisyonları belirle: Atların pozisyonlarını arama uzayında güncelle.
G. İterasyon: Durdurma kriterini sağlayana kadar C adımına geri dön.



HOA algoritmasının sözde kodu, Şekil 7(a) ve (b)'de sunulur. HOA'nın nasıl faydalı olabileceğini görmek için bazı açıklamalar aşağıdaki gibidir:

- α atları en iyi tepkileri alır ve aynı zamanda diğerleri için bir kılavuz görevi görür. En iyi yanıtı bulmak ve bir sömürü stratejisi oluşturmak için araştırmaya başlarken bir koç gibi hareket ederler. Bu, otlatma ve savunma mekanizması özelliklerinin uygulanması gerektiğinde gerçekleşir.
- β atları, α'ya özel bir dikkat göstererek en olası optimal konumları dikkatlice arar.
- Atların tüm doğal davranışları γ atları yaratmak için kullanılır. Hem sağlam hem de rastgele hareketlere sahiptirler ve hem keşif hem de kullanım aşamaları için faydalıdırlar.
- Genç atlar daha ahlaksız ve oyuncudur; bu nedenle, keşif aşaması için daha uygundurlar.

$0-5$ yaşındaki δ atlarının hızları:

$$\begin{aligned}\vec{V}_m^{\text{Iter},\delta} = &\left[g_m^{\text{Iter}-1),\delta}\,\omega_g(\breve{u}+\mathcal{I})\left[X_m^{(\text{Iter}-1)}\right]\right]\\ &+\left[i_m^{\text{Iter}-1,\delta}\,\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{\text{Iter}-1}\right)-X^{\text{Iter}-1}\right]\right]\\ &+\left[r_m^{\text{Iter}-1,\delta}\,\omega_r\mathcal{P}X^{\text{Iter}-1}\right]\end{aligned}$$

$5-10$ yaşındaki γ atlarının hızları:

$$\begin{aligned}\vec{V}_m^{\text{Iter},\gamma} = &\left[g_m^{\text{Iter}-1,\gamma}\,\omega_g(\breve{u}+\mathcal{P})\left[X_m^{\text{Iter}-1}\right]\right]\\ &+\left[h_m^{\text{Iter}-1,\gamma}\,\omega_h\left[X_*^{\text{Iter}-1}-X_m^{\text{Iter}-1}\right]\right]\\ &+\left[s_m^{\text{Iter}-1,\gamma}\,\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N}X_j^{\text{Iter}-1}\right)-X_m^{\text{Iter}-1}\right]\right]\\ &+\left[i_m^{\text{Iter}-1,\gamma}\,\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{\text{Iter}-1}\right)-X^{\text{Iter}-1}\right]\right]\end{aligned}$$

Diğer denklem aşağıdaki gibidir.

$$\begin{aligned}&-\left[d_m^{\text{Iter}-1,\gamma}\,\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\check{X}_j^{\text{Iter}-1}\right)-X^{\text{Iter}-1}\right]\right]\\ &+\left[r_m^{\text{Iter}-1,AGE}\,\omega_r\mathcal{P}X^{\text{Iter}-1}\right]\end{aligned}$$

$10-15$ yaşındaki β atlarının hızları:

$$\begin{aligned}\vec{V}_m^{\text{Iter},\beta} = &\left[g_m^{\text{Iter}-1,\beta}\,\omega_g(\breve{u}+\mathcal{P}\updownarrow)\left[X_m^{\text{Iter}-1}\right]\right]\\ &+\left[h_m^{\text{Iter}-1,\beta}\,\omega_h\left[X_*^{\text{Iter}-1}-X_m^{\text{Iter}-1}\right]\right]\\ &+\left[s_m^{\text{Iter}-1,\beta}\,\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N}X_j^{\text{Iter}-1}\right)-X_m^{\text{Iter}-1}\right]\right]\\ &-\left[d_m^{\text{Iter}-1,\beta}\,\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\check{X}_j^{\text{Iter}-1}\right)-X^{\text{Iter}-1}\right]\right]\end{aligned}$$

15 yaşından büyük α atlarının hızları:

$$\begin{aligned}\vec{V}_m^{\text{Iter},\alpha} = &\left[g_m^{\text{Iter}-1,\alpha}\,\omega_g(\breve{u}+\mathcal{P}\check{l})\left[X_m^{\text{Iter}-1}\right]\right]\\ &-\left[d_m^{\text{Iter}-1,\alpha}\,\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\check{X}_j^{\text{Iter}-1}\right)-X^{\text{Iter}-1}\right]\right]\end{aligned}$$

4. HOA Hesaplama Karmaşıklığı

Hesaplama karmaşıklığı, belirli bir yöntemin problem çözme süresini inceler. Başka bir deyişle, bir algoritmanın hesaplama karmaşıklığı, yöntemin hesaplama maliyetini tahmin etmede bir parametre olarak kullanılır. Meta-sezgisel algoritmaların hesaplama maliyeti ayrıca arama ajanlarının sayısına, problem boyutlarının sayısına ve maksimum yineleme sayısına dayalı olarak tahmin edilmektedir [71]. Atların hareketinde altı faktörü kullanmak, HOA algoritmasının keşif ve kullanım aşamaları arasında iyi bir denge sağlamasına neden olur ve problem çözmede hesaplama karmaşıklığını azaltır. Bu bağlamda HOA, küresel bir matriste sıralama mekanizmasını kullanarak yerel optimum yakalamadan kaçınmanın yanı sıra problem çözme hızını artırmak için uygun bir çözüm kullanır. Global matris, Denklemlerde açıklandığı gibi, konumlar (X) ve her bir konumun maliyeti (C (X)) yan yana getirilerek elde edilir. (4.1) ve (4.2).

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,d} \end{bmatrix}, C(X) = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

$$\text{Global Matrix} = \begin{bmatrix} X & C(X) \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} & c_1 \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} & c_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,d} & c_m \end{bmatrix}$$

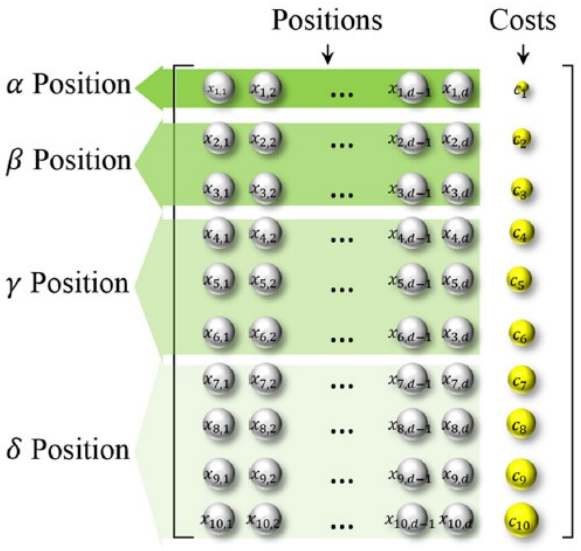


Fig. 8. Sorting process in the HOA algorithm.

X ve C (X) sırasıyla pozisyonları ve her bir pozisyonun maliyetini gösterir. d ve m aynı zamanda sırasıyla problem boyutlarının sayısı ve atların sayısıdır. sonraki adımda, global matris son sütuna (maliyetler) göre sıralanır. Atların yaşı bu aşamada ve Şekil 8'e göre uygulanır. Bu rakamın on at için küresel matrisi gösterdiğinden bahsetmek en kötüsüdür. Bu stratejinin en önemli başarısı, optimal nokta olma ihtimalinin yüksek olduğu durumlarda (α) yüksek doğruluk ve düşük hız ile istismar edilmesi ve optimal nokta olma ihtimalinin yüksek olduğu durumlarda ise düşük doğruluk ve yüksek hız ile keşfedilmesidir. optimum nokta düşüktür (δ ve γ atları civarında). HOA, belirtildiği gibi hızlı matris sıralama sistemini dikkate alır. Dolayısıyla, en iyi ve en kötü durumda sıralama adımındaki hesaplama maliyeti (CC ile gösterilir), sırasıyla CC(m × log (m)) ve CC(m2)'ye eşittir. Bu nedenle, HOA'nın genel hesaplama maliyeti de Denklem (4.3) ile elde edilir. [58].

$$\begin{aligned} CC(HOA) &= CC\left(\frac{\text{Iter}_{\max}}{2} \times [CC(\text{Sorting}) + CC(\text{position update})]\right) \\ &= CC\left(\frac{\text{Iter}_{\max}}{2} \times \left[m^2 + m \times d\right]\right) \\ &= CC\left(\frac{\text{Iter}_{\max} m^2}{2} + \frac{\text{Iter}_{\max} d}{2}\right) \end{aligned}$$

HOA algoritmasının hesaplama maliyetini azaltmadaki önemli faydalarından bazıları aşağıda listelenmiştir:

Algoritmanın ilk adımında ve atlar tarafından yapılan global arama sırasında, problem uzayının iyi bir yeniden yapılandırması sağlanır;

- α atlarının alanı arama yöntemi de uygulanır, böylece kullanım aşaması çok verimli bir şekilde yapılır;
- β atlarını α atlarını takip etmek (küresel en iyi konumların yaklaşık yüzde onu) α atları etrafında yerel optimumu bulmayı mümkün kılar. Ayrıca, diğer alanlar γ ve δ atları tarafından sürekli aranmaktadır.
- Atların altı hareket modeli, problem alanı için en iyi aramayı ve keşif ile kullanım aşamaları arasında iyi bir dengeyi sağlar.

Bu şekilde, önerilen algoritma mümkün olan en kısa sürede ve en düşük hesaplama maliyetiyle optimum yanıtı elde etme potansiyeline sahiptir. Bir sonraki bölümde, önerilen algoritmanın mevcut yöntemlere göre etkinliğini belirlemek için bazı kıyaslama matematiksel ve yapısal optimizasyon problemleri incelenmiştir.

5. Parametre Analizi

Birkaç parametre HOA'nın performansını etkiler. Bu parametrelerin etkinliğinin doğru bir şekilde anlaşılmasını sağlamak için, parametrelerdeki değişikliklere tepkinin duyarlılık analizi yapılır. g faktörünün değiştirilmesi, ortaya çıkan yanıtlar üzerinde önemli bir etkiye sahip değildir ve yalnızca her atın arama aralığını artırır veya azaltır. Duyarlılık analizinde aşağıdaki faktörler incelenir:

- β ve γ atları için hiyerarşi faktörü ($h\beta$ ve $h\gamma$);
- β ve γ atları için sosyallik faktörü ($s\beta$ ve $s\gamma$);
- γ atları için taklit faktörü ($i\gamma$);
- α, β ve γ atları için savunma faktörü ($d\alpha$, $d\beta$ ve $d\gamma$);
- δ ve γ atları için dolaşım faktörü ($r\delta$ ve $r\gamma$);

Katsayılar 0,1'den 1'e 0,1'lik bir adımla değiştirildi ve algoritma her durumda 500 boyutta küresel fonksiyon üzerinde çalıştırıldı. Şekil 9, her bir analizden elde edilen optimum cevabı göstermektedir ve her bir katsayı için en iyi değerler şekil üzerinde işaretlenmiştir.

Katsayılar için en iyi değerler şunları içerir:

- $h\beta$ ve $h\gamma$ sırasıyla 0,9 ve 0,5'e eşittir;
- $s\beta$ ve $s\gamma$ sırasıyla 0,2 ve 0,1'e eşittir;
- $i\gamma$ 0,3'e eşittir;
- $d\alpha$, $d\beta$ ve $d\gamma$ sırasıyla 0,5, 0,2 ve 0,1'e eşittir;
- $r\delta$ ve $r\gamma$ sırasıyla 0.1 ve 0.05'e eşittir.

Ama biz, her bir problemde faktörlerin optimum miktarını hesaplamak için bu yöntemi diğer optimizasyon algoritmalarıyla birleştiren, HOA'nın hiper sezgisel bir versiyonunu geliştirmeye karar verdik.

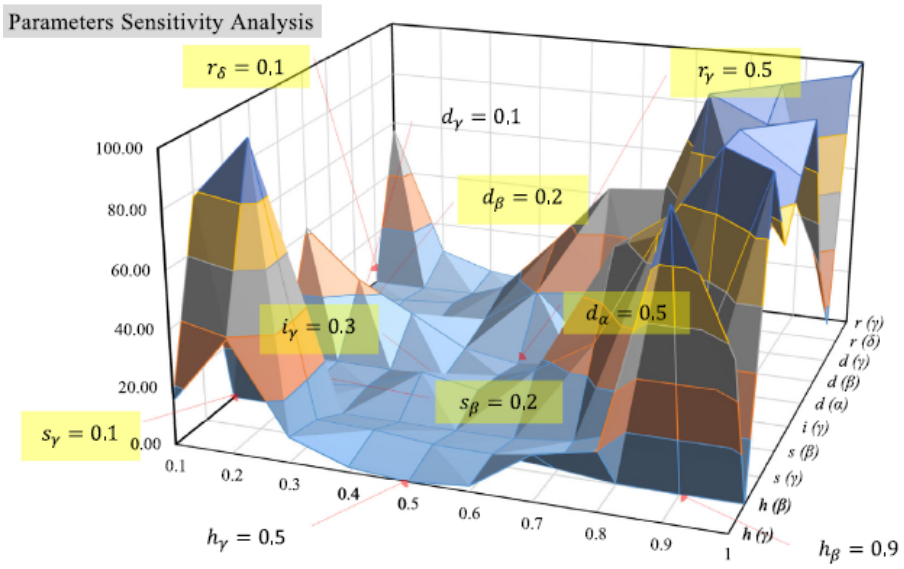


Figure 1. HOA parametre duyarlılık analizi.

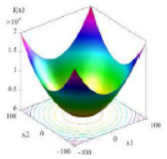
6. Sonuçlar ve Tartışma

HOA sonuçları bu bölümde sunulmaktadır. Bu algoritma, Grasshopper Optimizasyon Algoritması (GOA), Sinüs Kosinüs Algoritması (SCA), Multi-Verse Optimizer (MVO), güve-alev Optimizasyonu (MFO), Dragonfly Algoritması (DA) ve Gray Wolf Optimizer (GWO) ile karşılaştırılmıştır. Adil bir karşılaştırma yapabilmek için tüm algoritmalardaki at sayısı ve iterasyon sayısı sırasıyla 50 ve 1000 olarak ayarlanmıştır. Sonuçların önemini de göstermek için istatistiksel testler yapılır.

6.1. Uni-Modal benchmark işlemleri

Uni-Modal benchmark işlemleri, keşif aşaması boyunca algoritmanın gücünü öne çıkarmak için ideal araçlardır. bu bölümde, Sphere dahil olmak üzere Uni-Modal benchmark işlemi vardır. (F1)Küre, Döndürülmüş Hiper-Elipsoid (F2), Schwefel.1 (F3) ve Schwefel.2 (F4). Her fonksiyon için denklemler, fonksiyon şekli, sonuç tablosu, yakınsama diyagramları ve en iyi maliyetler sağlanır. Dim problem boyutlarının sayısını ifade eder (number of variables).

F1 (Sphere function) [72]

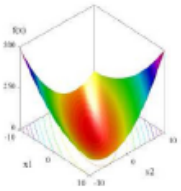


$$f(x) = \sum_{i=1}^{Dim} x_i^2$$
$$Range = [-100,100]$$
$$f_{min} = 0$$
$$x^* = [0,0,\dots,0]^{Dim} \quad (6.1)$$

Tablo 2, farklı boyutlarda Küre fonksiyonunu (F1) çözmek için, yukarıda bahsedilen algoritmaların tümü kapsamında en az maliyet, yanıtların standart sapması (SD), Wilcoxon sıralama toplamı testinin p değeri ve CPU çalışma süresi dahil olmak üzere sonuçları göstermektedir. Değerler, her boyut sayısı için 30 bağımsız algoritma çalışmasından elde edilir. Her test fonksiyonundaki en iyi sonuçlar seçilir ve diğer algoritmaların sonuçlarıyla ayrı ayrı karşılaştırılır. N/A (Uygulanamaz) en iyi algoritma ve her fonksiyon için yazılır çünkü en iyi algoritma kendi kendisiyle karşılaştırılamaz.

GOA, SCA, MVO ve DA algoritmaları yeterince iyi bir yanıt bulamadı. GWO ise kısa sürede, optimal yanıta yakın, yaklaşık olarak yeterince iyi yanıtlar verir. HOA'nın sonuçları, makul bir süre içinde bulunan mutlak optimum noktaya oldukça yakındır. HOA tarafından elde edilen en iyi maliyet ve SD, diğer tüm algoritmalar tarafından elde edilenlerden çok daha iyidir. Parametrik olmayan istatistiksel testin, yani Wilcoxon's rank sum testinin p değerleri, HOA için N/A'dır ve bu algoritmanın istatistiksel üstünlüğünü gösterir. Buna göre, HOA algoritması, no-free-lunch (NFL) teoremine göre, diğer algoritmalar tarafından verimli bir şekilde çözülemeyen sorunları çözme potansiyeline sahiptir.

F2 (Rotated Hyper-Ellipsoid function) [72]



$$f(x) = \sum_{i=1}^{Dim} \left(\sum_{j=1}^{i} x_j\right)^2$$
$$Range = [-100,100]$$
$$f_{min} = 0$$
$$x^* = [0,0,\dots,0]^{Dim}$$

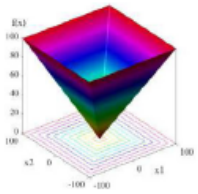
Tablo 3, daha önce belirtildiği gibi, beş farklı yüksek boyutlu uzayda F2 problemini çözmek için yedi algoritmanın yanıtlarını sağlar. HOA ile bu sorunu çözmek için gereken süre diğerlerine göre çok fazla olmamakla birlikte, HOA yanıtları en iyi maliyet ve SD açısından diğer algoritmalarla elde edilen yanıtlardan çok daha iyidir.

Table 2
Optimization results of F1 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	**1.84E−38**	349889	165259	313489	916271	225517	1.82E−14
	SD	2.85E−40	69212	72687	74992	50989	28738	7.66E−16
	P-Wlcxn	*N/A*	0.009773	0.005289	0.008345	0.00464	0.004098	0.002878
	Run Time (s)	49	479	8	17	9	290	8
1000	*Best Cost*	**5.35E−38**	705239	572255	672733	2432815	507688	5.36E−10
	SD	1.44E−40	95316	18146	17983	141492	63543	3.4E−11
	P-Wlcxn	*N/A*	0.00562	0.006541	0.005455	0.00405	0.000401	0.005264
	Run Time (s)	88	723	13	38	15	570	16
2000	*Best Cost*	**2.79E−37**	1780937	836374	1594276	5671196	1372209	9.70E−07
	SD	1.75E−38	230886	50946	180347	566349	77664	3.31E−08
	P-Wlcxn	*N/A*	0.008167	0.008768	0.004112	0.002392	0.008477	0.009365
	Run Time (s)	164	1862	24	82	29	755	49
5000	*Best Cost*	**1.94E−34**	4501328	1322409	4561398	1.5E+07	2.9E+07	1.15E−04
	SD	1.90E−35	327535	65737	371614	941069	1211897	6.31E−05
	P-Wlcxn	*N/A*	0.006501	0.009581	0.08234	0.004696	0.00924	0.001483
	Run Time (s)	400	2980	57	171	72	1621	80
10000	*Best Cost*	**5.47E−26**	11752861	2804620	1.1E+07	3.1E+07	7.1E+07	0.040605
	SD	7.26E−24	780214	92774	391081	665587	897899	5.22E−04
	P-Wlcxn	*N/A*	0.002851	0.009472	0.006134	0.003556	0.18954	0.000943
	Run Time (s)	795	6348	116	364	173	3912	154

F3, değişkenleri değiştirdikten sonra f(x)'in doğrusal olarak arttığı basit bir Schwefel işlevi sunar.

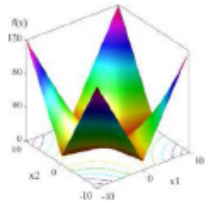
F3 (Schwefel.1 function) [72]



$$f(x) = \max_i\{|x_i|, 1 \le i \le Dim\}$$
$$Range = [-100,100]$$
$$f_{min} = 0$$
$$x^* = [0,0,\dots,0]^{Dim}$$

HOA'nın F3 sorununa verdiği yanıtlar yaklaşık 1E-20; bununla birlikte, Tablo 4'te bildirildiği gibi, GWO'dan alınan minimum değer, 500 boyutta 46'dır. Çoğu durumda, yanıtlar yaklaşık 100'dür.Bu durum yüksek boyutlarda bu basit sorunu çözmek için kullanılan bu algoritmaların zayıflığını gösterir.
Son Uni-Modal benchmark işlemi (F4) aynı zamanda toplama ve çarpma işlemlerine sahip bir Schwfel fonksiyonudur.

F4 (Schwefel.2 function) [72]



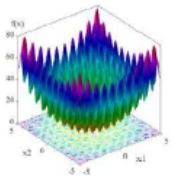
$$f(x) = \sum_{i=1}^{Dim}|x_i| + \prod_{i=1}^{Dim}|x_i|$$
$$Range = [-2.5,2.5]$$
$$f_{min} = 0$$
$$x^* = [0,0,\dots,0]^{Dim}$$

GWO, seçilen diğer algoritmalara kıyasla iyi bir performans sergiliyor. 500 boyutta F4'e yanıtı 1.57x10⁻9'dur, bu da mükemmel bir sonucu gösterir. İlginç bir şekilde, HOA tarafından sunulan optimum çözümler, GWO'nun yanıtlarından yaklaşık 1e+10 kat daha iyidir (500 boyut için 7e-20 ve 10000 boyut için 4.25x10⁻18)! (bkz. Tablo 5) GWO algoritmasının kabul edilebilir bir performansa sahip olduğu unutulmamalıdır; ancak, HOA en rekabetçi çözümleri içerir. Drop-Wave işlevi (F7)(yan paragraf), 2D versiyonunda bile çok modlu ve oldukça karmaşıktır. Tablo 8, HOA'nın eşsiz performansını göstermektedir. Bu karmaşık problem için, yüksek boyutlu uzaylarda, yanıtların standart sapması olmadan makul bir çalışma zamanında kesin çözümü bulmak, diğer algoritmalara kıyasla HOA'nın üstünlüğünü sunar. Öte yandan, diğer güçlü algoritmalar optimuma yakın bir çözüme ulaşmadı.

6.2. Multi-Modal benchmark işlemleri

Multi-Modal benchmark işlemleri, bir algoritmanın yararlanma kabiliyetine meydan okumak için etkili bir kriterdir. Bu bölümde Rastrigin (F5), Ackley (F6) ve Drop-Wave (F7) fonksiyonu olmak üzere üç farklı fonksiyon kullanılmaktadır. Belirtildiği gibi, bu işlevler HOA, GOA, SCA, MVO, MFO, DA ve GWO algoritmaları ile 500, 1000, 2000, 5000 ve 10000 boyutlarda optimize edilmiştir.

F5 (Rastrigin function) [72]



$$f(x) = \sum_{i=1}^{Dim}(x_i^2 - 10cos(2\pi x_i) + 10)$$
$$Range = [-100,100]$$
$$f_{min} = 0$$
$$x^* = [0,0,\dots,0]^{Dim}$$

Rastrigin fonksiyonu(F5) dışbükey olmayan bir işlevdir ve doğrusal olmayan çok modlu işlevin tipik bir örneğidir. Bu fonksiyonun minimum değerini bulmak, geniş arama uzayı ve çok sayıda yerel minimumu nedeniyle oldukça zor bir problemdir (bkz. Tablo 6). HOA'nın performansı bu durumda kesinlikle mükemmeldir ve sonuçların en iyi maliyeti ve SD'sinin sıfıra eşit olduğu bildirilir. Öte yandan, GWO dışındaki diğer algoritmalar kabul edilebilir bir yanıt vermemektedir. HOA tarafından tüm boyutlarda elde edilen küresel en iyi maliyet, her analizde sıfıra eşit olarak hesaplanır.

F6 (Ackley function) [72]

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{Dim}x_i^2}{n}}\right) - \exp\left(\frac{\sum_{i=1}^{Dim}\cos(2\pi x_i)}{n}\right) + 20 + e \quad (6.6)$$
$$Range = [-100,100]$$
$$f_{min} = 0$$
$$x^* = [0,0,\dots,0]^{Dim}$$

F7 (Drop-Waves function) [72]

$$f(x) = -\frac{1+\cos\left(12\sqrt{\sum_{i=1}^{Dim}x_i^2}\right)}{2+0.5\sum_{i=1}^{Dim}(x_i^2)}$$
$$Range = [-100,100]$$
$$f_{min} = -1$$
$$x^* = [0,0,\dots,0]^{Dim}$$

Ackley fonksiyonu (F6), optimizasyon algoritmalarını test etmek için yaygın olarak kullanılır. İşlev, optimizasyon algoritmalarının birçok yerel minimumundan birinde yakalanması için bir risk oluşturur. Bu problemin karmaşıklığı, problemin boyutlarının artmasıyla önemli ölçüde artmaktadır (bkz. Tablo 7). Diğer algoritmalardan en iyi maliyetlerin tümü yaklaşık 20'dir; ancak, HOA algoritması makul bir sürede oldukça düşük SD ile yeterince iyi yanıtlar (yaklaşık 1E-14) sağlar. HOA için her koşulda 'Yok' ifadesi, sonuçların sürdürülebilirliğini ve sağlamlığını da belirtir ve bu sonuçların tesadüfen elde edilmediğini doğrular.

Table 3
Optimization results of F2 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	**1.50E−37**	479301	4488844	891335	3719764	1.8E+07	36153.61
	SD	5.65E−39	30207	385713	50794	456604	655737	1754
	P-Wlcxn	*N/A*	0.07912	0.000544	0.008519	0.004663	0.008453	0.008229
	Run Time (s)	123	272	81	91	82	261	84
1000	*Best Cost*	**2.95E−37**	1176233	15722451	2.1E+07	1.3E+07	4.1E+07	724582
	SD	1.13E−38	69212	517414	7918964	1427702	2909442	9094
	P-Wlcxn	*N/A*	0.004589	0.006722	0.000574	0.005072	0.009946	0.003558
	Run Time (s)	272	584	190	173	119	672	314
2000	*Best Cost*	**3.87E−36**	2569258	6.2E+07	6.4E+07	8.0E+07	8.8E+07	3584493
	SD	4.45E−37	196434	5300869	2511438	3.5E+07	7520569	26178
	P-Wlcxn	*N/A*	0.000378	0.009433	0.000336	0.006853	0.004958	0.009203
	Run Time (s)	678	1709	537	362	203	1283	1122
5000	*Best Cost*	**1.48E−35**	6501678	7.18E+08	8.5E+07	6.4E+08	2.7E+08	2.1E+07
	SD	1.46E−36	344335	4.1E+07	103539	6.1E+07	4.7E+07	720118
	P-Wlcxn	*N/A*	0.002353	0.002236	0.00269	0.006248	0.13436	0.001843
	Run Time (s)	3127	4099	1049	805	524	2671	2417
10000	*Best Cost*	**2.77E−35**	1.8E+07	2.25E+09	9.1E+07	2.0E+09	5.8E+08	5.9E+07
	SD	3.47E−36	4003498	8.6E+07	8242734	1.7E+07	3.1E+07	6434784
	P-Wlcxn	*N/A*	0.054894	0.000629	0.004464	0.006261	0.001373	0.002169
	Run Time (s)	7453	9820	2183	1751	2329	5613	6719

Table 4
Optimization results of F3 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	**1.02E−20**	15.4288	98.6686	98.1294	97.7112	80.2307	46.6540
	SD	2.23E−22	1.60E−02	6.03E−02	5.63E−02	9.24E−02	8.99E−02	3.49E−02
	P-Wlcxn	*N/A*	0.007162	0.00671	0.008183	0.000906	0.001954	0.001781
	Run Time (s)	46	118	8	19	9	153	9
1000	*Best Cost*	**8.87E−21**	31.7921	99.1451	98.3313	99.6257	81.3990	53.7401
	SD	4.78E−23	5.40E−02	6.97E−02	8.75E−02	9.24E−02	8.07E−02	4.22E−02
	P-Wlcxn	*N/A*	0.003351	0.008855	0.000216	0.001244	0.001517	0.004454
	Run Time (s)	94	263	16	44	15	391	17
2000	*Best Cost*	**1.33E−20**	64.4268	99.8642	99.8959	99.8074	81.9507	86.1761
	SD	1.92E−23	5.70E−02	9.10E−02	9.15E−02	9.54E−02	8.77E−02	7.35E−02
	P-Wlcxn	*N/A*	0.005925	0.003162	0.00981	0.000795	0.004976	0.007199
	Run Time (s)	201	584	25	80	28	970	31
5000	*Best Cost*	**1.22E−20**	83.6499	99.9327	99.9148	99.8959	82.0782	89.2239
	SD	4.18E−22	8.42E−02	8.21E−02	8.84E−02	9.54E−02	5.67E−02	4.34E−02
	P-Wlcxn	*N/A*	0.00607	0.009674	0.12168	0.001275	0.004989	0.008739
	Run Time (s)	492	1179	57	171	73	2471	76
10000	*Best Cost*	**1.49E−20**	89.1371	99.9781	99.9301	99.9611	83.4455	91.1764
	SD	1.01E−21	7.80E−02	4.52E−02	4.04E−02	1.01E−02	4.21E−02	8.96E−02
	P-Wlcxn	*N/A*	0.006187	0.004122	0.009327	0.000367	0.007425	0.003827
	Run Time (s)	893	2346	134	339	217	4803	153

Table 5
Optimization results of F4 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	**7.00E−20**	381.5531	22.3979	563.4516	405.8499	3279.491	1.57E−09
	SD	8.75E−21	33.8411	2.4137	30.6118	57.0459	832.6754	5.20E−11
	P-Wlcxn	*N/A*	0.002496	0.006988	0.071736	0.001595	0.002301	0.008747
	Run Time (s)	51	172	8	18	9	196	8
1000	*Best Cost*	**1.49E−18**	822.3770	54.4122	952.7371	991.242	8431.233	2.73E−07
	SD	5.38E−20	36.9239	6.2500	57.3069	92.1812	303.5648	8.60E−08
	P-Wlcxn	*N/A*	0.003787	0.009429	0.009838	0.003838	0.008645	0.00146
	Run Time (s)	94	361	11	37	15	423	15
2000	*Best Cost*	**8.33E−16**	2351.962	137.7924	1982.484	2197.125	18143.76	6.24E−05
	SD	6.22E−18	145.2102	6.8321	94.1573	108.1125	1366.146	2.14E−06
	P-Wlcxn	*N/A*	0.000481	0.005189	0.007824	0.006866	0.006869	0.007767
	Run Time (s)	186	796	24	89	28	810	32
5000	*Best Cost*	**1.69E−13**	5798.4634	152.4714	4572.316	5914.408	41970.49	1.23E−03
	SD	6.19E−15	767.145	24.8406	983.9951	749.4983	9766.231	5.75E−05
	P-Wlcxn	*N/A*	0.004636	0.001007	0.001733	0.004541	0.006331	0.006963
	Run Time (s)	404	1508	71	183	71	1761	76
10000	*Best Cost*	**4.25E−18**	14733.8	178.8952	11232.44	12050.2	85046.17	0.047284
	SD	3.93E−19	512.9546	6.4995	681.9785	3065.61	1880.302	9.53E−03
	P-Wlcxn	*N/A*	0.008053	0.006661	0.000129	0.005119	0.34329	0.008311
	Run Time (s)	809	3671	126	417	148	4032	162

Table 6
Optimization results of F5 obtained by different algorithms in different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	**0.00**	421161.31	201021.5	955608.4	896710.2	51779.19	6.4812
	SD	0.00	56276.891	5343.534	9460.197	1273.110	2663.457	0.7462
	P-Wlcxn	*NAN*	0.008015	0.009429	0.001288	0.001028	0.004292	0.000622
	Run Time (s)	50	141	9	16	10	289	9
1000	*Best Cost*	**0.00**	1250767	451792	2374589	2382712	123814	69.87
	SD	0.00	91677.34	1975.73	89697.46	258401.7	13540.80	19.1935
	P-Wlcxn	*NAN*	0.095634	0.000801	0.007476	0.006585	0.002746	0.005747
	Run Time (s)	88	252	15	41	17	621	17
2000	*Best Cost*	**0.00**	3924005	663114.9	5003697	5659792	249312	55.8
	SD	0.00	96658.33	9737.795	685917.4	380769.0	98729.1	7.9057
	P-Wlcxn	*NAN*	0.001234	0.006138	0.009009	0.002525	0.008093	0.004507
	Run Time (s)	166	569	28	76	31	2493	33
5000	*Best Cost*	**0.00**	9220328	1433908	1.2E+07	1.5E+07	671962.3	82.9
	SD	0.00	670802.3	60759.83	252127.3	980875.1	25518.16	9.5847
	P-Wlcxn	*NAN*	0.004169	0.004585	0.00123	0.003528	0.008644	0.005891
	Run Time (s)	415	1247	56	166	71	6170	85
10000	*Best Cost*	**0.00**	3.0E+07	2953671	2.5E+07	3.1E+07	1570329	1118.547
	SD	0.00	1960644.7	90509.8	869366.7	1041173	87336.44	20.7947
	P-Wlcxn	*NAN*	0.006547	0.002096	0.16840	0.003805	0.00631	0.005187
	Run Time (s)	828	2364	139	314	176	11273	174

Table 8
Optimization results of F7 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	**−1.00**	−4.99E−05	−1.16E−05	−3.1E−04	−2.7E−06	−1.0E−03	−0.1893
	SD	0.00	1.19E−06	7.79E−06	8.56E−05	4.78E−07	2.75E−04	4.88E−02
	P-Wlcxn	*NAN*	0.001261	0.005092	0.007665	0.003257	0.009839	0.008352
	Run Time (s)	48	103	7	14	9	299	9
1000	*Best Cost*	**−1.00**	−7.34E−06	−7.22E−05	−5.6E−05	−1.2E−06	−1.0E−04	−0.1572
	SD	0.00	6.47E−07	2.99E−06	2.38E−06	7.71E−08	2.37E−06	8.10E−02
	P-Wlcxn	*NAN*	0.009319	0.00309	0.007763	0.002054	0.003583	0.004743
	Run Time (s)	96	242	21	33	15	487	15
2000	*Best Cost*	**−1.00**	−2.47E−07	−3.75E−06	−1.2E−07	−6.1E−07	−2.0E−06	−0.10768
	SD	0.00	9.30E−08	8.90E−07	4.53E−09	9.17E−08	5.35E−08	7.17E−03
	P-Wlcxn	*NAN*	0.007383	0.008912	0.007361	0.008819	0.009862	0.004367
	Run Time (s)	179	486	39	67	28	758	32
5000	*Best Cost*	**−1.00**	−8.36E−09	−8.72E−07	−4.8E−08	−2.4E−07	−2.0E−07	−0.05841
	SD	0.00	9.89E−11	6.36E−08	9.58E−09	4.58E−09	4.74E−08	2.99E−03
	P-Wlcxn	*NAN*	0.004609	0.001228	0.34806	0.005216	0.005451	0.002681
	Run Time (s)	434	961	75	156	71	1637	74
10000	*Best Cost*	**−1.00**	−3.72E−10	−5.22E−07	−1.3E−08	−1.2E−07	−1.7E−09	−2.2E−03
	SD	0.00	7.39E−12	8.59E−08	6.31E−09	3.71E−09	5.16E−11	7.25E−05
	P-Wlcxn	*NAN*	0.05167	0.003373	0.000746	0.00648	0.061525	0.006961
	Run Time (s)	854	451	159	332	181	3046	162

Table 7
Optimization results of F6 obtained by different algorithms at different dimensions.

Dim		HOA	GOA	SCA	MVO	MFO	DA	GWO
500	*Best Cost*	**4.00E−14**	20.4321	20.815	20.033	20.05759	20.053	21.4357
	SD	7.79E−15	3.26E−02	7.59E−02	9.72E−03	5.75E−02	2.95E−02	2.26E−02
	P-Wlcxn	*NAN*	0.00594	0.003758	0.006807	0.00139	0.007973	0.003961
	Run Time (s)	46	91	10	15	10	264	10
1000	*Best Cost*	**1.87E−14**	20.586	20.8542	20.052	20.07	20.072	21.4791
	SD	2.99E−16	8.40E−02	6.91E−02	6.95E−02	6.26E−02	4.83E−02	3.83E−02
	P-Wlcxn	*NAN*	0.00212	0.007515	0.07487	0.008049	0.0852	0.008502
	Run Time (s)	74	196	17	31	16	583	22
2000	*Best Cost*	**1.87E−14**	20.76156	20.9	20.06283	20.06	20.091	21.5
	SD	8.93E−17	9.07E−02	6.60E−02	2.11E−02	6.13E−02	5.11E−03	7.03E−02
	P-Wlcxn	*NAN*	0.009429	0.004587	0.000971	0.007349	0.002716	0.007783
	Run Time (s)	192	408	28	64	30	1245	35
5000	*Best Cost*	**1.90E−14**	20.9122	20.9218	20.084	20.09	20.164	21.5223
	SD	6.36E−16	5.80E−02	7.88E−02	8.73E−03	9.22E−02	5.43E−02	7.54E−02
	P-Wlcxn	*NAN*	0.000994	0.000847	0.006548	0.009492	0.002676	0.005626
	Run Time (s)	439	823	69	139	75	2389	89
10000	*Best Cost*	**7.99E−14**	20.9873	20.9679	20.097	20.101	20.221	21.5667
	SD	5.89E−15	5.15E−02	5.34E−02	1.69E−02	7.15E−03	8.31E−02	3.30E−02
	P-Wlcxn	*NAN*	0.008895	0.003993	0.09285	0.000792	0.002927	0.005983
	Run Time (s)	785	2342	132	311	160	5423	182

Table 10
Optimization results of further analysis.

Function		HOA	GOA	SCA	MVO	MFO	DA	GWO
F8(10,000 dimensions)	*Best Cost*	**21.8660**	90809.8	9814.90	86393.6	5644.81	67133.06	45.3480
	SD	1.97409	5674.91	724.889	6955.65	37.9459	6382.94	2.2564
	P-Wlcxn	*NAN*	2.80E−04	8.47E−02	7.41E−03	3.09E−02	8.67E−03	3.50E−04
	Run Time (s)	6037	8624	2650	1863	3410	6749	7308
F9(10,000 dimensions)	*Best Cost*	**4.65e−05**	235.748	377.156	892.019	9055.31	84.5329	42.8476
	SD	7.54E−07	14.7324	27.8552	71.8175	60.8722	8.03730	2.1319
	P-Wlcxn	*NAN*	9.98E−03	4.96E−02	5.85E−02	6.97E−02	9.41E−03	8.51E−04
	Run Time (s)	943	2746	275	405	301	5712	341
F10(10,000 dimensions)	*Best Cost*	**0.00**	19.4430	71.4311	1.53824	73.4271	22.1618	1.88E−01
	SD	1.77E−03	1.46823	5.30308	0.12784	1.14914	1.01347	1.16E−02
	P-Wlcxn	*NAN*	7.72E−04	8.50E−04	2.60E−02	1.43E−02	1.02E−02	7.47E−04
	Run Time (s)	857	2942	234	605	280	5316	196
F11(10,000 dimensions)	*Best Cost*	**4.39e−04**	426066.6	1358.47	261698.0	8354.09	615224.5	78.773
	SD	1.82E−02	39717.21	113.446	23433.05	486.641	35852.01	6.73450
	P-Wlcxn	*NAN*	6.34E−02	3.34E−03	5.72E−03	6.68E−02	5.38E−03	2.59E−04
	Run Time (s)	2419	6578	352	914	408	3492	565

6.3. İleri Analiz

Önerilen HOA yönteminin yüksek boyutlu problemlerle başa çıkmadaki etkinliğini kanıtlamak için Uni-Modal ve Multi-Modal fonksiyon daha 10.000 boyutta bu yöntemlerle incelenmiştir. Tablo 10'da verilen sonuçlara göre HOA ile elde edilen en iyi maliyet ve SD, diğer algoritmalardan daha iyiydi. Wilcoxon testinin bazı durumlarda P değeri 0,05'ten fazlaydı, SCA, MVO ve MFO dahil olmak üzere bazı algoritmaların çalışma süresinin HOA'dan daha kısa olmasına rağmen, tesadüfen elde edildiğini gösteriyor. Bölüm 6.1 ve 6.2'de sunulan sonuçlara göre, HOA, Uni-Modal ve Multi-Modal benchmark işlevleri için oldukça rekabetçi sonuçlar sağlar.
Bu algoritmanın olumlu performansı aşağıdaki nedenlerle açıklanabilir:

- Konumlarına ve uygunluklarına göre arama ajanlarının düzenli olarak sınıflandırılması;
- Her sınıflandırmada, parçacık durumunun iyileştirilmesi ile arama parçacıklarının sayısı azalır ve bu nedenle kümedeki liderlik süreci daha iyi olur;
- At sürüsünün altı doğal davranışını kullanarak performansıyla uyum;
- Sonuç olarak algoritmanın hesaplama maliyetini azaltan MATLAB'da oldukça hızlı sıralama kullanımı.

Buna göre, bu algoritma, yüksek boyutlu uzaylarda on bir test fonksiyonunu analiz etmede GOA, SCA, MVO, MFO, DA ve GWO'dan daha iyi performans gösterir. Atların hareketleri arasında uyumlu ilişkilerin kurulması, son derece karmaşık problemler için en iyi çözümü bulmak için HOA algoritmasının keşif ve kullanımda bazı avantajlara sahip olması da dahil olmak üzere, atların davranışlarından en yüksek ve en iyi şekilde yararlanılmasını sağlar.

7. Sonuç

Bu çalışma, farklı yaşlardaki atların genel davranışlarından ilham alan ve oldukça karmaşık optimizasyon problemlerini çözmek için kullanılan hızlı ve sağlam bir optimizasyon algoritması önermiştir. HOA algoritması, iyi bilinen yedi test fonksiyonu kullanılarak yüksek boyutlarda kıyaslanmış ve bu algoritmanın keşif ve kulanuş açısından oldukça verimli olduğu tespit edilmiştir. Sonuçlar (yani en iyi maliyet, SD, Wilcoxon testinin p değeri ve CPU çalışma zamanı), HOA'nın yüksek boyutlu uzaylarda çok sayıda bilinmeyen değişkene sahip bu zorlu problemlerle başa çıkma yeteneğini doğruladı. Bu sonuçlar aynı zamanda, mevcut çözümlere kıyasla 500, 1000, 2000, 5000 ve 10000 boyutlarında kıyaslama fonksiyonunun önemli bir gelişimini göstererek, son derece karmaşık ve zorlu problemlerle uğraşırken önerilen algoritmanın uygulanabilirliğini ima etti. HOA bileşenleri arasında oldukça uygun bir denge, önerilen algoritmanın olağanüstü yeteneğini haklı çıkarıyor. Bu denge, algoritma tarafından hesaplama karmaşıklığının verimliliğini arttırır. Yetişkin α atları, global optimum etrafında ultra yüksek hassasiyetle yerel bir aramaya başlar. β atları, onlara doğru hareket etme arzusuyla α atlarının etrafındaki diğer yakın durumları da ararlar; ancak, γ atları, α atlarına doğru hareket etmekle daha az ilgilenir. Yeni yerler aramak ve yeni olası küresel optimum yerleri bulmak için büyük bir istekleri vardır. Genç δ atları düzensizliklere alışkındır, çünkü belirli davranış özelliklerinden dolayı rastgele arama aşaması için oldukça uygun seçeneklerdir. Davranış katsayıları, otlatma, hiyerarşi, sosyallik, taklit, savunma mekanizması ve gezinme dahil olmak üzere atlarda altı önemli özelliği belirler. Bu parametrelerin dahil edilmesi, mümkün olan en kısa sürede optimum çözümü bulabilen tutarlı bir algoritmanın ortaya çıkmasına neden olur. HOA geliştirme süreci, gelecekteki çalışmalarda çok amaçlı optimizasyon problemlerini çözmek için ilerlemektedir.