

Beale, Ackley, Goldstein, Levi Test Fonksiyonları İle Guguk Kuşu Algoritması ve Benzetimli Tavlama Algoritması Optimizasyonu

Mehmetcan DALMAZGİL

Bursa Teknik Üniversitesi, 18360859001@ogrenci.btu.edu.tr

Özet

Matematiksel test fonksiyonları optimizasyon algoritmalarını test etmek için kullanılan matematiksel formüllerdir[1]. Bu fonksiyonlar, optimizasyon algoritmalarını hem hız hem de bulunan çözüm performansı açısından karşılaştırmamıza olanak tanır. Bu çalışmada Ackley, Beale, Goldstein, Levi test fonksiyonları kullanılarak Guguk Kuşu Optimizasyon Algoritması(GOA) ile Benzetimli Tavlama Algoritması (BT) optimizasyon işlemi gerçekleştirilmiştir. Çalışma Pycharm ve Microsoft Visual Studio Code geliştirme ortamları kullanılarak Python programlama diliyle gerçekleştirilmiş olup iki optimizasyon algoritması karşılaştırılmıştır. Sonuçlar incelenerek Benzetimli Tavlama Algoritmasının Guguk Kuşu Algoritmasına göre daha az iterasyonla optimize hale ulaştığı gözlemlenmiştir.

1. Giriş

Optimizasyon, verilen amaç veya amaçlar için belirli kısıtlamaların sağlanarak en uygun çözümün elde edilme sürecidir. Optimizasyon problemlerinin çözümünde klasik yöntemler olarak adlandırılan matematiksel yöntemler önceleri çok yaygın olarak kullanılmaktaydı. Bu tür yöntemlerin esnek olmaması ve matematiksel fonksiyonlarla tanımlama gereksinimi gibi dezavantajları, son zamanlarda, bilim adamlarında genel amaçlı ve performansı yüksek yöntemler geliştirme çabalarını artırmış ve doğadaki olaylardan esinlenmeye başlamışlardır. Tabiattaki olaylar temel alınarak geliştirilen optimizasyon algoritmaları sezgisel yöntemler olarak

adlandırılmaktadır[2]. Bu sezgisel algoritmalarından biri de çalışma içerisinde kullanılan Guguk Kuşu Optimizasyon Algoritması(GOA)'dır. Benzetimli Tavlama Algoritması ise en iyi çözümün en kısa zamanda üretimini hedefleyen, eniyileme problemi için tasarlanmış olasılıksal yaklaşımlı bir optimizasyon algoritmasıdır[3].

Test fonksiyonları, optimizasyon algoritmalarının

- Yakınsama oranı,
- Hassaslık,
- Sağlamlık,
- Genel performans,

özelliklerini değerlendirmek için kullanılmaktadır[4].

Bu çalışmada Ackley, Beale, Goldstein, Levi test fonksiyonları kullanılarak Guguk Kuşu Algoritması ile Benzetimli Tavlama Algoritmasının optimizasyon işlemi gerçekleştirilmiş olup, iki algoritma içinde test fonksiyonu sonuçları görselleştirilmiştir.

2. Guguk Kuşu Algoritması

Guguk kuşlarının yaşam tarzından esinlenerek Ramin Rajabioun (2011) tarafından geliştirilen yeni bir sezgisel algoritma olan guguk kuşu optimizasyon algoritmasının temelini kuşların yerleşimi ve üremesi almaktadır. Guguk kuşları bir kuş yuvasının gerçek sahibi kuş uzaklaştığında gözetlediği yerden gelir ve yumurtasını

ev sahibi kuşun yumurtaları arasına bırakır. Bu olay sadece 10sn sürmektedir. Yuva sahibi kuşun yumurtalarından birisini de gagasıyla alarak yuvadan uzaklaştır. Daha sonra yavru guguk kuşu, yuva sahibi kuştan daha büyük hale gelmesine rağmen üvey anne tarafından beslenmeye devam edilir. Üvey anne tarafından fark edilmeyen yumurtalar, kuluçka dönemi sırasında üvey kardeşlerinden önce yumurtadan çıkarlar. İlk 4 gün içerisinde gözleri henüz açılmamış olmasına rağmen çeşitli hareketlerle üvey kardeşlerini yuvadan dışarı atmaya başlar[5].

2.1. Popülasyonu başlatma

Bir optimizasyon problemini çözmek amacıyla problem değişkenlerini bir dizide tutmak gerekir ve bu dizi GA ve PSO terminolojilerinde sırasıyla “Kromozom” ve “Parçacık Konumu” olarak adlandırılmaktadır. GOA ’da bu dizi, kuşların yetiştiği doğal yaşam ortamı olarak ifade edilen “ortam” kelimesiyle tanımlanmaktadır [6].

2.2.Yumurtlama süreci

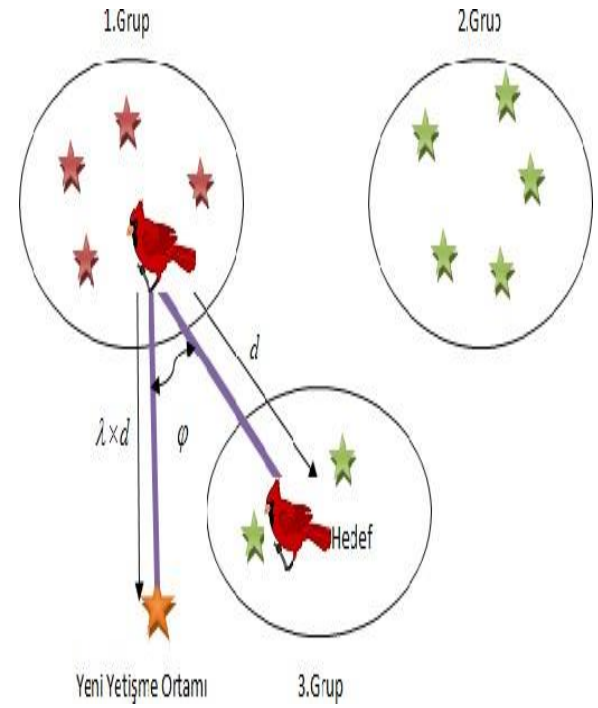
Yumurtalar, yumurtlama yarıçapı dikkate alınarak yumurta merkezleri etrafında rastgele yerleştirilir. Yumurtalar yerleştirildikten sonra ev sahibi kuş tarafından fark edilen yumurtalar yuvadan atılır. O anki yumurtalar arasında istenmeyen %p (genelde %10) kadarı silinir. Algoritmada tanımlanan parametresi ortamda yaşamını sürdüren guguk kuşu sayısını kontrol eder. Bu sebeple her bir iterasyonda en iyi maliyet değerlerine sahip sayısı kadar guguk kuşu hayatta kalacaktır [7].

2.3.Guguk kuşlarının göç etmesi

Guguk kuşlarının farklı alanlarda grup oluşturduktan sonra en iyi maliyete sahip olan topluluk diğer guguk kuşlarının göç etmesi için hedef nokta olarak seçilmektedir. Guguk kuşu gruplarını belirlemek amacıyla K-Ortalama Kümeleme Algoritması (KKA) kullanılmaktadır[9]. Guguk kuşlarını kümelemede, Rajabioun(2011) tarafından gerçekleştirilen benzetimler sonucunda k değerinin 3-5 arasında bir değer seçilir.

Her bir guguk kuşu çözümü, hedef ortama olan uzaklığının λ katı kadar hedefe yaklaşır ve radyan kadar bir sapma eğilimi gösterir.

Guguk kuşlarının yaşam alanları içerisinde gerçekleşen göç işlemi Şekil 1’de gösterilmektedir.



Şekil 1 Guguk kuşlarının yaşam alanları[8].

Guguk kuşu algoritmasının sözde kodu:

1. İlk başlangıç ortamı oluşturulur
2. Her kuşa rastgele sayıda yumurtalar tahsis edilir.
3. Her kuş için maksimum yumurtlama yarıçapı belirlenir.
4. Belirlenen yumurtlama yarıçapı alanı içine yumurta bırakılır.
5. Ev sahibi kuşlar tarafından farkedilen yumurtalar yok edilir.
6. Cıvcıvlerin yumurtadan çıkmasına ve büyümesine izin verilir.
7. Büyüyen her kuşun yaşam alanı değerlendirilir.
8. Alan içinde yaşayabilecek kuş sayısı limitlendirilir ve istenmeyen alandakiler yok edilir.
9. En iyi kuş grubu tespit edilir, hedef yaşam alanı seçilir.
10. Yeni guguk kuşu nüfusunun hedef yaşam alanına göç etmesi sağlanır.
11. İstenilen şart sağlanmış ise optimizasyon durdurulur, sağlanmamış ise 2.adımdan devam edilir.

Guguk Kuşu Algoritması'na ait akış diyagramı Şekil 2'dedir.

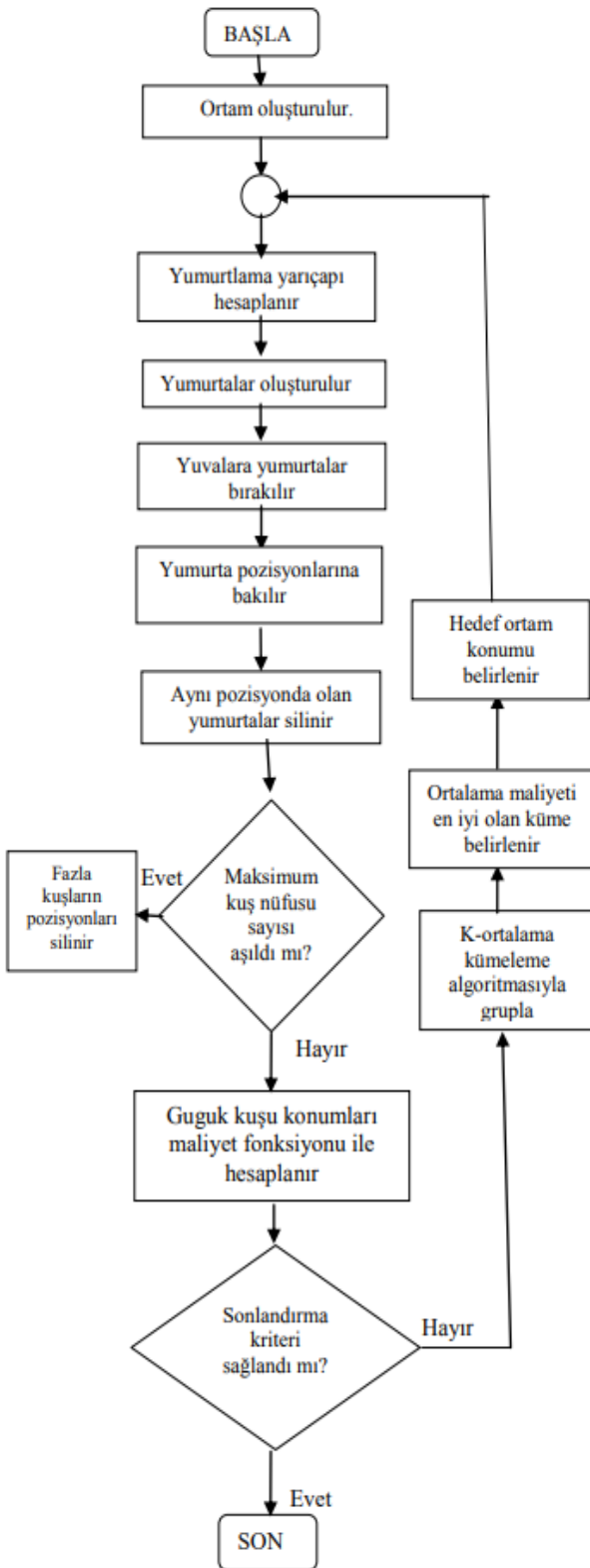
3. Benzetimli Tavlama

Benzetimli Tavlama Algoritması, yinelemeli rastgele bir arama tekniğidir ve çeşitli kombinasyonel optimizasyon problemlerini çözmek için yaygın olarak kullanılmaktadır. SA ilk olarak Metropolis, Rosenbluth ve Teller (1953) tarafından metallerin tavlama işlemini simüle etme amacıyla önerilmiştir ve Kirkpatrick, Gelatt ve Veechi (1983) tarafından yinelemeli bir optimizasyon yöntemi olarak tanıtılmıştır. Benzetimli Tavlama Algoritması önceden belirlenen bir başlangıç sıcaklığı ile başlar (T_0). Bir başlangıç çözümü oluşturulur (S_0) ve bu aynı zamanda geçerli çözüm (SC) ve ilk en iyi çözüm olur (SB). Başlangıç çözümünün amaç fonksiyonu değeri (E_0), geçerli çözüm (EC) ve en iyi çözümün amaç fonksiyonu değeri olur (EB). Geçerli çözümünden (SC) bir değişiklik ile komşu çözüm (SN) oluşturulur ve amaç fonksiyon değerleri hesaplanır (ΔE). Eğer komşu çözümün amaç fonksiyonu değeri (EN), EC 'den daha iyiyse, yeni geçerli çözüm olarak kabul edilir. Ancak EN EC 'den daha kötüyse, $\exp(-\Delta E/T)$ (T geçerli sıcaklık) olasılığı ile SN yeni SC olarak kabul edilir, diğer durumda SN reddedilir. T parametresi bir soğutma fonksiyonu ($T = \alpha \times T$) ile yavaşça azaltılır ve T_{min} ulaşılabacak son sıcaklıktır. TB algoritmasının kötü çözümleri kabul ederek yerel optimum noktalardan kurtulma kapasitesi vardır[10]

Benzetimli Tavlama Sözde Kodu[11]:

1. Giriş parametreleri: $T_0, T_{min}, \alpha, T = T_0$.
2. Bir S_0 üret ve E_0 'ı hesapla.
3. $SC = S_0, SB = S_0, EC = E_0, EB = E_0$.
4. SC 'den bir SN üret ve EN 'i hesapla
5. $\Delta E = EN - EC$.
6. Eğer $\Delta E \leq 0$ komşu çözümü geçerli çözüm olarak Kabul et, $SC = SN, EC = EN$ ve Adım 9'a git
7. Eğer $\Delta E > 0$ rnd isminde $[0, 1]$ arasında bir değişken üret.
8. Eğer $\exp(-\Delta E/T) > rnd$ ise komşu çözümü geçerli çözüm olarak Kabul et, $SC = SN, EC = EN$; diğer durumda, SC ve E değişmeden kalır ve Adım 10'a git.
9. Eğer $EC < EB$ ise $SB = SC; EB = EC$; diğer durumda, SB ve EB değişmeden kalır.
10. $T = T \times \alpha$
11. Eğer $T \geq T_{min}$ ise Adım 4'e git; diğer durumda, SB ve EB 'yi raporla ve bitir.

Benzetimli Tavlama Algoritması'na ait akış diyagramı Şekil 3'dedir.



Şekil 2. Guguk Kuşu Optimizasyon Algoritması akış diyagramı[9]

4. Yöntem

Çalışma Microsoft Visual Studio Code ve Pycharm platformlarında Python programlama dili kullanılarak yapılmıştır. Guguk Kuşu Algoritması için kullanıcıdan optimizasyonda kullanılacak test fonksiyonunu, algoritma içinde üretilicek random sayıların minimum ve maximum sınır değerlerini ve iterasyon sayısını belirlemesi istenmiştir. Benzetimli Tavlama Algoritması içinde kullanıcıdan optimizasyonda kullanılacak test fonksiyonunu belirlemesi istenmiştir.. Sonuç olarak iki algoritmada, optimizasyon sonucunda kullanılan test fonksiyonlarına göre optimum çözüm değerleri(x,y) ve bu çözüm değerlerine göre fonksiyon sonucunu(f(x,y)) çıktı olarak ekrana vermektedir. Aynı zamanda her iterasyonda optimizasyon işlemi gerçekleşirken çözüm değerleri ile fonksiyon sonuçları kaydedilerek bu değerlerin değişimleri kodun sonunda grafiksel olarak gösterilmiştir..

Guguk Kuşu Algoritmasının girdi olarak aldığı değişkenler şunlardır;

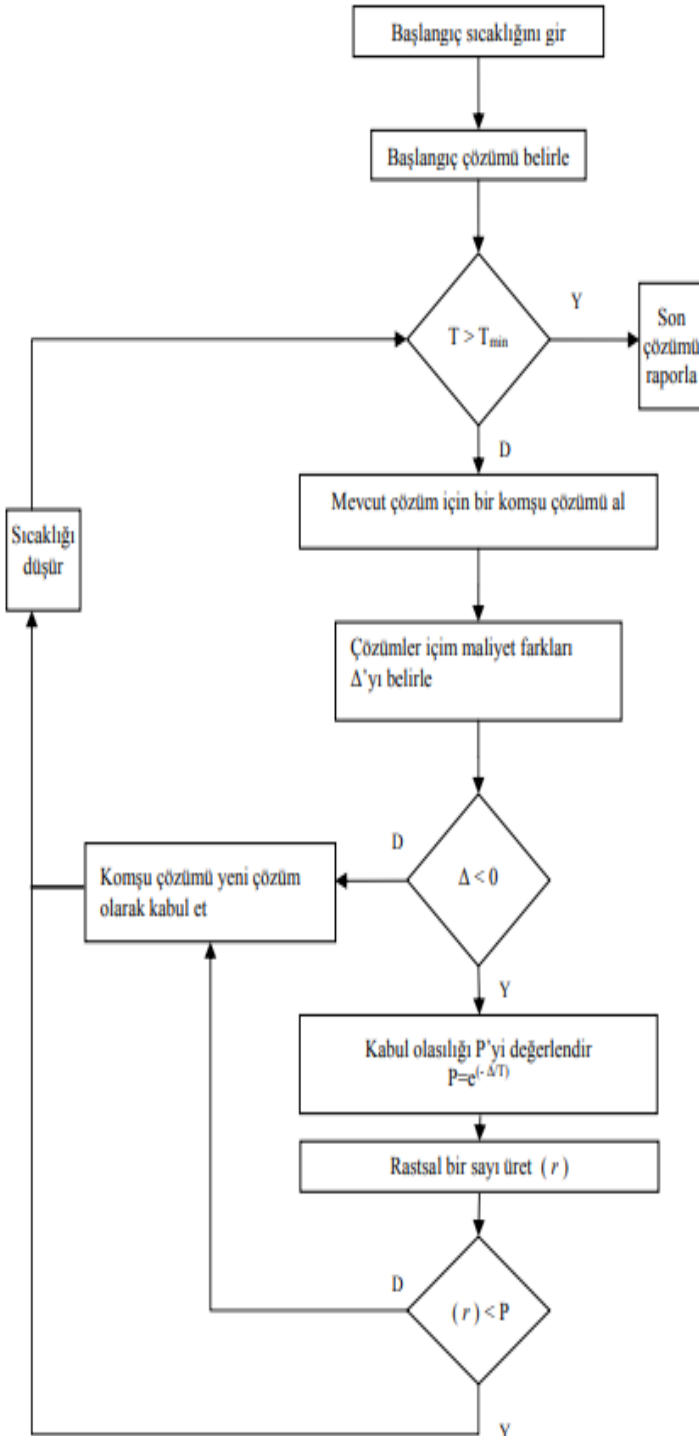
- n : Farklı yuvaların/çözümlerin sayısı
- dimension : Çözümlerin kaç boyutlu vektörler olduğu
- pa : Guguk kuşu yumurtalarının keşfedilme oranı
- min_deger : Bir çözümdeki/yuvadaki bir bileşenin alabileceği minimum değer
- max_deger : Bir çözümdeki/yuvadaki bir bileşenin alabileceği maximum değer
- iteration : İterasyon sayısı

GOA'da min_deger, max_deger ve iteration değişkenlerinin değerleri kullanıcı tarafından belirlenirken; n, dimension ve pa değişkenlerinin değerleri algoritmada default olarak verilmektedir.

Bu değerlerin default olarak verilmesinin nedeni algoritmanın bu değişken değerlerinde daha optimize(istenilen şekilde) çalışmasıdır. Bu değerler kod içerisinde değiştirilebilmektedir.

Benzetimli Tavlama Algoritmasının girdi olarak aldığı değişkenlerde şunlardır:

- t : Başlangıç sıcaklığı
- m : Döngü başına deneme sayısı
- alpha : Daha kötü çözümü kabul etme olasılığı
- min_deger : Minimum lokasyon değeri
- max_deger : Maximum lokasyon değeri



Şekil 3. Benzetimli Tavlama Algoritması akış diyagramı[12]

BT’de kullanıcı sadece optimizasyon için kullanılacak test fonksiyonunu belirlemektedir. Diğer değerler default olarak kod içerisinde verilmiştir.

4.1. Algoritma Test Fonksiyonu Sonuçları

Her iki algortmada sırayla Beale, Ackley, Goldstein ve Levi test fonksiyonları için çalıştırılmıştır. Bu test fonksiyonlarının ulaşması beklenen çözüm değerleri(x,y) ve çözüm değerlerine bağlı olarak beklenen sonuç değerleri şu şekildedir;

- Beale : $f_{min}(3,0.5) = 0$
- Ackley : $f_{min}(0,0) = 0$
- Goldstein : $f_{min}(0,-1) = 3$
- Levi : $f_{min}(1,1) = 0$

Algoritmaların çalıştırılmasıyla elde edilen sonuçlar Tablo 1 ve Tablo 2’de verilmiştir.

Tablo 1: GOA’nın test fonksiyonları için uygulanmasıyla elde edilen sonuçlar

Guguk Kuşu Optimizasyon Algoritması			
Test Fonk.	Sonuç(x)	Sonuç(y)	Sonuç(Fonk)
Beale	3.125	0.522	0.003
Ackley	-0.021	-0.016	0.019
Goldstein	-0.010	-0.994	3.050
Levi	1.003	0.893	0.016

Not : GOA’da tüm optimizasyon işlemleri 100000 adımda için gerçekleştirilmiştir.

Tablo 2: BT’nin test fonksiyonları için uygulanmasıyla elde edilen sonuçlar

Benzetimli Tavlama Algoritması			
Test Fonk.	Sonuç(x)	Sonuç(y)	Sonuç(Fonk)
Beale	3.081	0.519	0.001
Ackley	0.023	0.004	0.017
Goldstein	-0.032	-1.013	3.247
Levi	0.982	0.745	0.158

Not : BT’de tüm optimizasyon işlemleri 40 adım için gerçekleştirilmiştir..

4.2. Algoritma Girdileri ve Grafikselle Sonuçlar

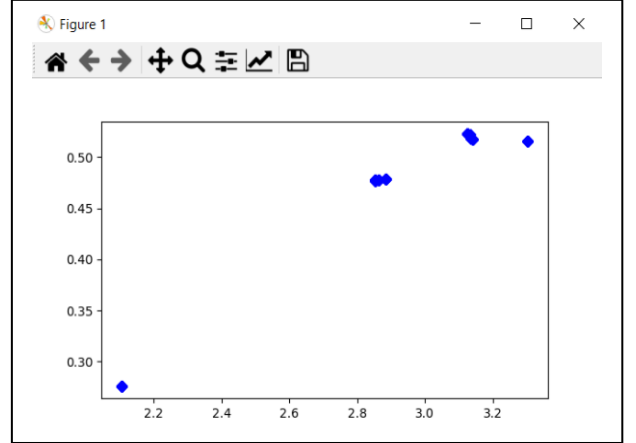
Her iterasyonda optimizasyon işlemi gerçekleşirken çözüm değerleri ile fonksiyon sonuçları kaydedilerek bu değerlerin değişimleri kodun sonunda grafikselle olarak verilmektedir. İki algoritma içinde her bir test fonksiyonu için algortmalara verilen girdi değerleri ile elde edilen sonuç grafikleri aşağıdadır.

4.2.1. Guguk Kuşu Algoritması Girdiler ve Grafikselle Sonuçlar

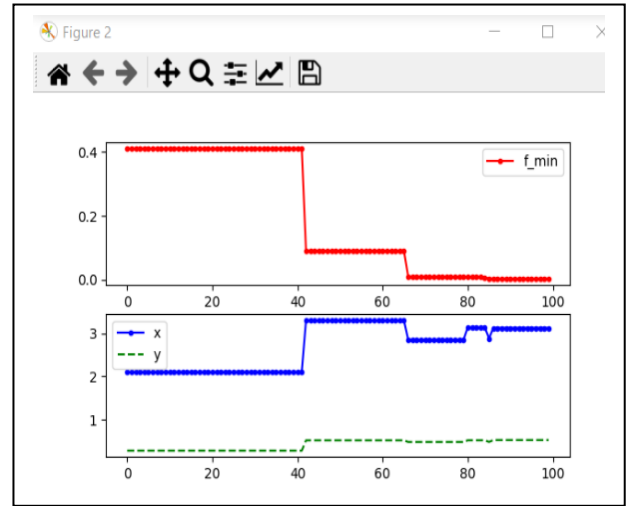
Beale fonksiyonu verilen girdiler :

- Minimum Değer : -4.5
- Maksimum Değer : 4.5
- İterasyon Sayısı : 100000

Beale fonksiyonu grafikselle sonuçlar :



Şekil 4.1. Beale Fonksiyonu için oluşan x ve y değerleri (GOA)

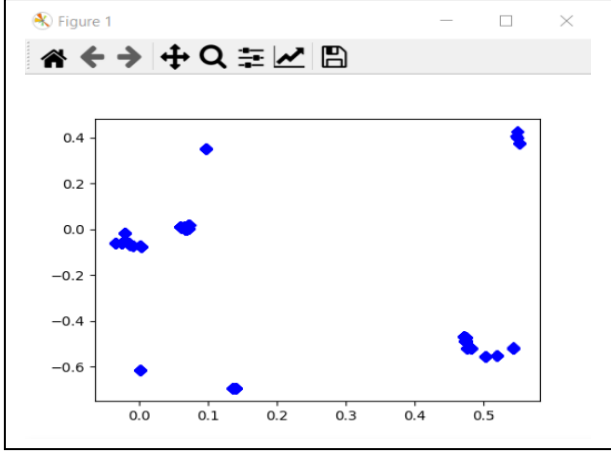


Şekil 4.2. Beale Fonksiyonu çözüm ve fonksiyon değerleri (GOA)

Ackley fonksiyonu verilen girdiler :

- Minimum Değer : -5
- Maksimum Değer : 5
- İterasyon Sayısı : 100000

Ackley fonksiyonu grafiksel sonuçlar :

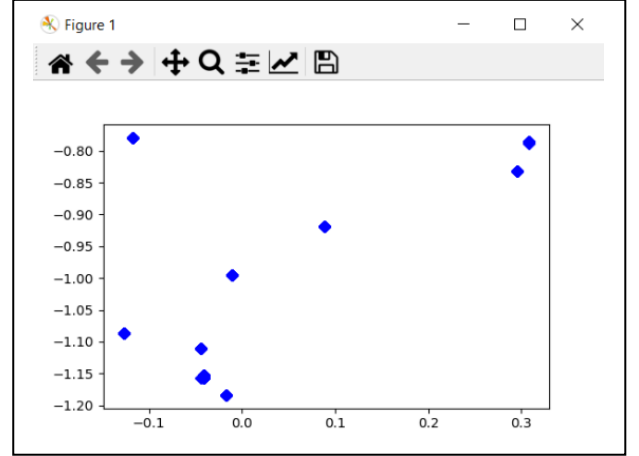


Şekil 5.1. Ackley Fonksiyonu için oluşan x ve y değerleri (GOA)

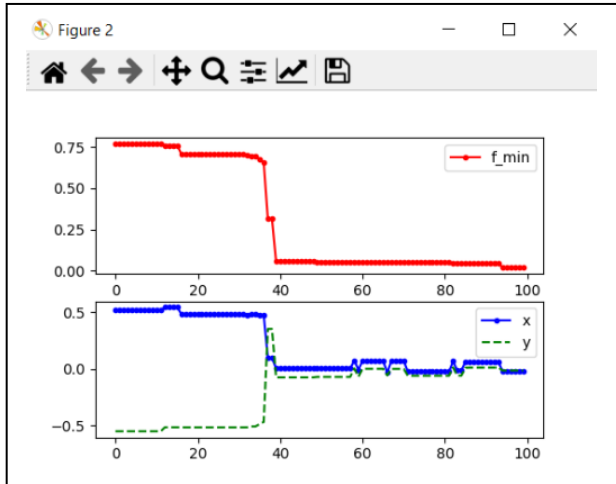
Goldstein fonksiyonu verilen girdiler :

- Minimum Değer : -2
- Maksimum Değer : 2
- İterasyon Sayısı : 100000

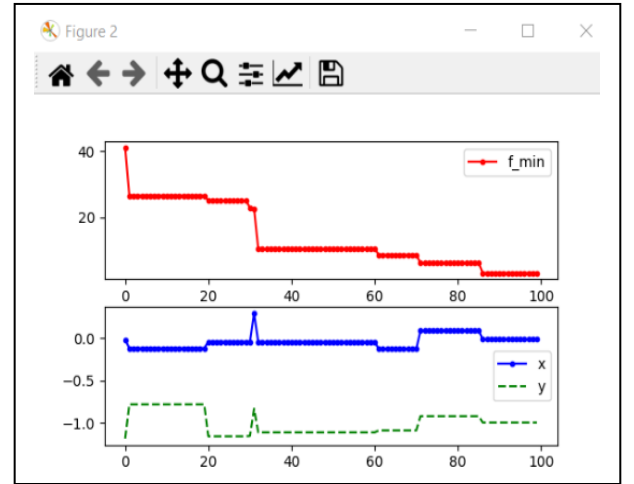
Goldstein fonksiyonu grafiksel sonuçlar :



Şekil 6.1. Goldstein Fonksiyonu için oluşan x ve y değerleri (GOA)



Şekil 5.2. Ackley Fonksiyonu çözüm ve fonksiyon değerleri (GOA)

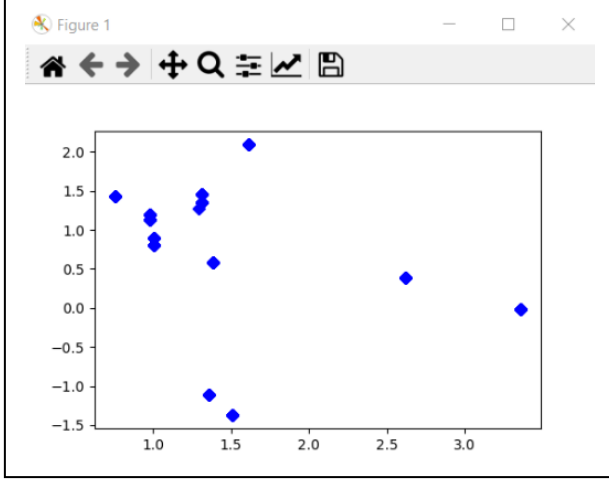


Şekil 6.2. Goldstein Fonksiyonu çözüm ve fonksiyon değerleri (GOA)

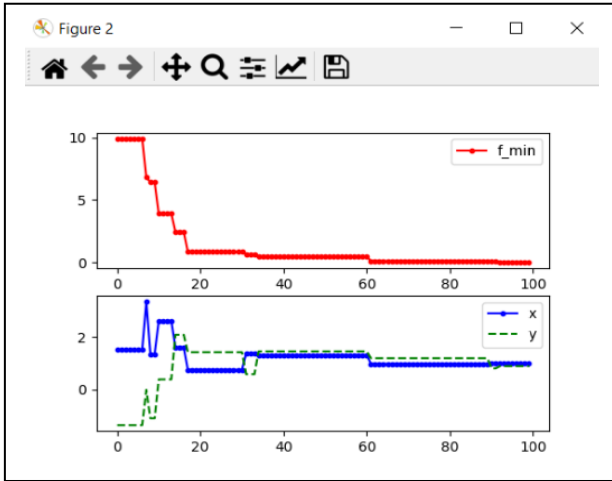
Levi fonksiyonu verilen girdiler :

- Minimum Değer : -10
- Maksimum Değer : 10
- İterasyon Sayısı : 100000

Levi fonksiyonu grafiksel sonuçlar :



Şekil 7.1. Levi Fonksiyonu için oluşan x ve y değerleri (GOA)



Şekil 7.2. Levi Fonksiyonu çözüm ve fonksiyon değerleri (GOA)

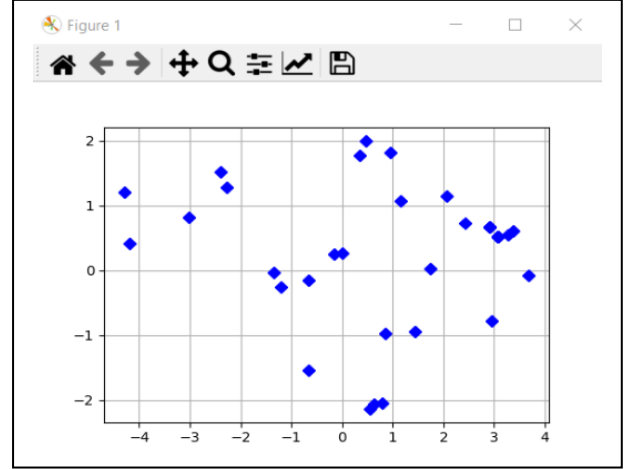
4.2.2. Benzetimli Tavlama Algoritması Girdiler ve Grafiksel Sonuçlar

BT'de aşağıda verilen girdi değerleri test fonksiyonu seçildiğinde default olarak gelmektedir.

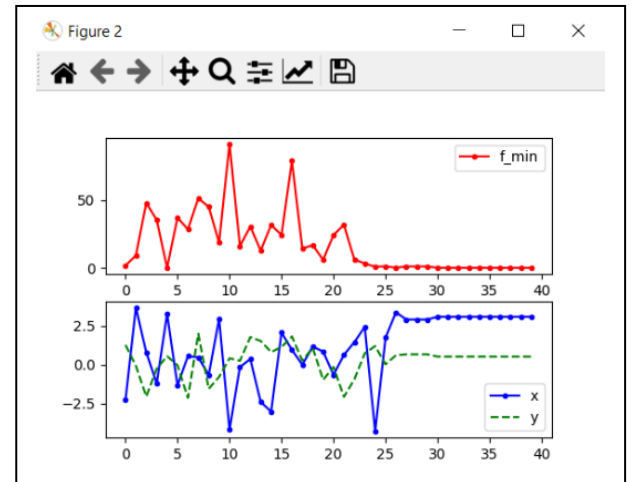
Beale fonksiyonu verilen girdiler :

- Minimum Değer : -4.5
- Maksimum Değer : 4.5
- İterasyon Sayısı : 40

Beale fonksiyonu grafiksel sonuçlar :



Şekil 8.1. Beale Fonksiyonu için oluşan x ve y değerleri (BT)

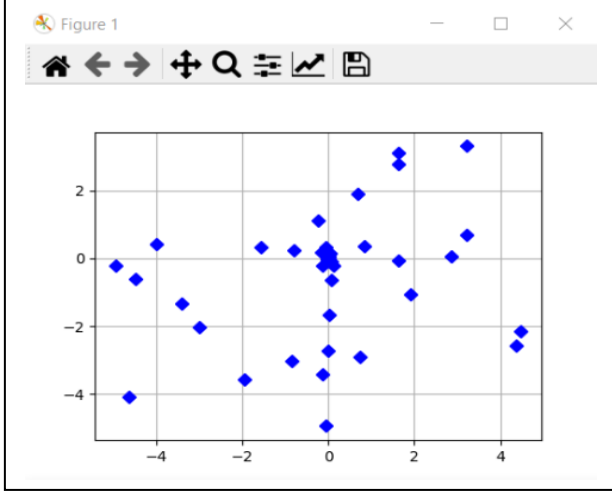


Şekil 8.2. Beale Fonksiyonu çözüm ve fonksiyon değerleri (BT)

Ackley fonksiyonu verilen girdiler :

- Minimum Değer : -5
- Maksimum Değer : 5
- İterasyon Sayısı : 40

Ackley fonksiyonu grafiksel sonuçlar :

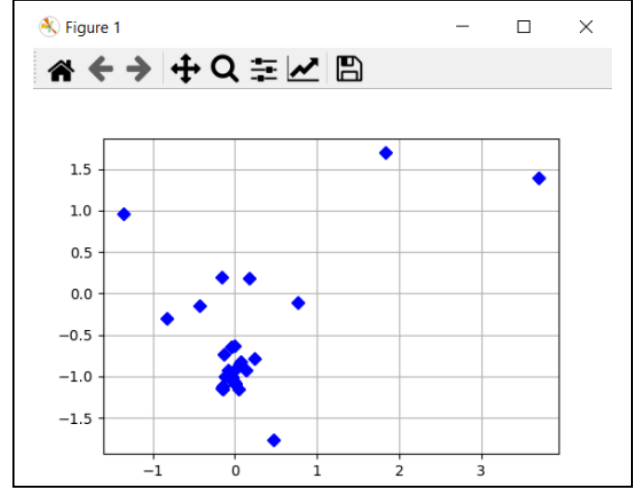


Şekil 9.1. Ackley Fonksiyonu için oluşan x ve y değerleri (BT)

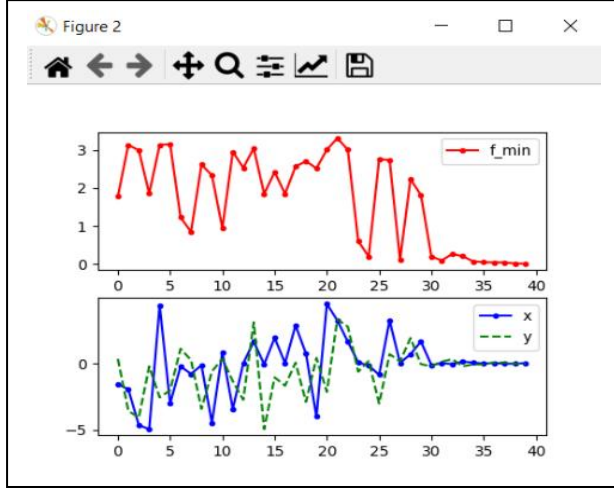
Goldstein fonksiyonu verilen girdiler :

- Minimum Değer : -2
- Maksimum Değer : 2
- İterasyon Sayısı : 40

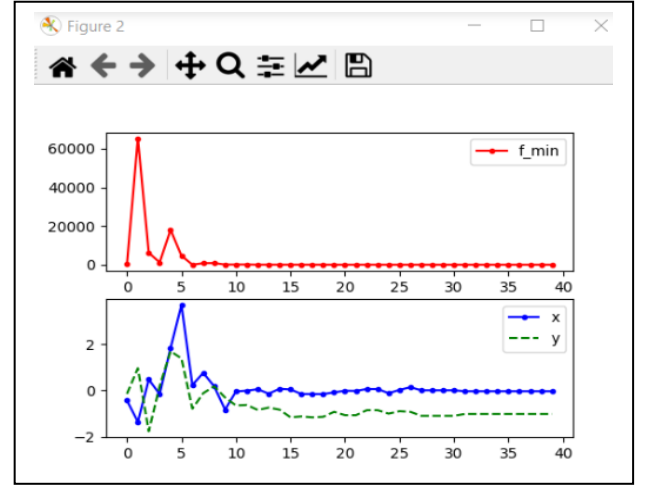
Goldstein fonksiyonu grafiksel sonuçlar :



Şekil 10.1. Goldstein Fonksiyonu için oluşan x ve y değerleri (BT)



Şekil 9.2. Ackley Fonksiyonu çözüm ve fonksiyon değerleri (BT)

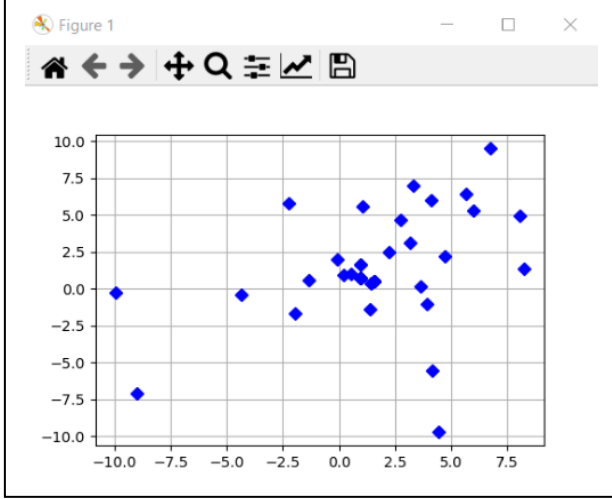


Şekil 10.2. Goldstein Fonksiyonu çözüm ve fonksiyon değerleri (BT)

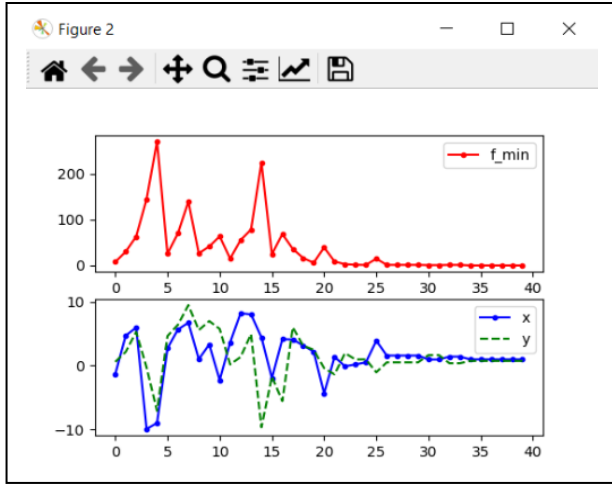
Levi fonksiyonu verilen girdiler :

- Minimum Değer : -10
- Maksimum Değer : 10
- İterasyon Sayısı : 40

Levi fonksiyonu grafiksel sonuçlar :



Şekil 11.1. Levi Fonksiyonu için oluşan x ve y değerleri (BT)



Şekil 11.2. Levi Fonksiyonu çözüm ve fonksiyon değerleri (BT)

4. Sonuç

Beale, Ackley, Goldstein ve Levi test fonksiyonları kullanılarak Guguk Kuşu ve Benzetimli Tavlama optimizasyon algoritmaları uygun girdi değerleriyle çalıştırıldığında her iki algoritmanın fonksiyonlar için minimize değerlere (fmin) çok yakın değerler üretebildiği görülmektedir. Buradan her iki algoritma içinde başarılı bir şekilde optimizasyon işlemlerini gerçekleştiren algoritmalar olduğu söylenebilmektedir. İterasyon sayılarına bakıldığında Benzetimli Tavlama Algoritması'nın Guguk Kuşu Algoritmasına göre çok az adım sayısında değerleri optimize edebildiği görülmektedir. Buradan BT'nin GOA'ya göre daha hızlı çalışan bir optimizasyon algortması olduğu anlaşılmaktadır.

5. Kaynaklar

- [1] <https://bilalsaim.com/yapay-zeka/optimizasyon-icin-test-fonksiyonlari/>
- [2] <https://dergipark.org.tr/tr/download/article-file/384717>
- [3] https://tr.wikipedia.org/wiki/Benzetimli%C5%9F_tavlama
- [4] https://en.wikipedia.org/wiki/Test_functions_for_optimization
- [5] http://www.turkcebilgi.com/ansiklopedi/guguk_kusu
- [6] <http://iranarze.ir/wp-content/uploads/2017/10/7962-English-IranArze.pdf>
- [7] https://www.academia.edu/22969260/Modified_cuckoo_optimization_algorithm_MCOA_to_solve_graph_coloring_problem
- [8] <http://iranarze.ir/wp-content/uploads/2017/10/7962-English-IranArze.pdf>
- [9] <https://www.ahmetcevahircinar.com.tr/2016/12/24/guguk-kusu-algoritmasinin-gezgin-satici-problemine-uygulanmasi-ve-simulasyonu/>
- [10] https://www.researchgate.net/publication/334128669_Supermarket_Yerlesim_Problemi_Icin_Tavlama_Benzetimi_Algoritmasi_Yaklasimi
- [11] https://www.researchgate.net/publication/334128669_Supermarket_Yerlesim_Problemi_Icin_Tavlama_Benzetimi_Algoritmasi_Yaklasimi
- [12] <http://www.ytusigmadergisi.com/pdfs/91.pdf>