

Beale, Ackley, Goldstein, Levi Test Fonksiyonları İle Ateş Böceği Algoritması ve Benzetimli Tavlama Algoritması Optimizasyonu

Burak AYTAŞ

1. Giriş

Optimizasyonun her yerde olduğunu söylemek abartılı değildir. Gerçek dünya uygulamalarında kaynaklar olarak, zaman ve para daima sınırlıdır. Değerli kaynakları en iyi şekilde kullanmak için çeşitli kısıtlamalar altında çözümler bulmak zorundayız. Matematiksel optimizasyon veya programlama, bunu benzer tasarım planlama problemlerinde kullanılmaktadır. Verilen kısıtlamalar altında, istenen faktörleri en üst düzeye çıkararak ve istenmeyen faktörleri en aza indirgeyerek en uygun maliyetli veya en yüksek erişilebilir performansa sahip bir alternatif bulmaktır.

Test fonksiyonları, optimizasyon algoritmalarının

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

özelliklerini değerlendirmek için kullanılmaktadır.

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

2. Ateş Böceği Algoritması

Firefly 2008 yılında Xin-She Yang tarafından geliştirilmiş sürü tabanlı Sezgisel Optimizasyon Algoritmasıdır. Ateş böceklerinin parlaklığa duyarlı sosyal davranışlarını ele alarak geliştirilmiştir. Algoritma ateş böceklerini cinsiyet olmadan ele alır. Yani tüm ateş böcekleri bir birlerine yönelebilirler. Daha parlak olan ateş böcekleri daha çekicidir. Daha az parlak olan ateş böcekleri çekici olan ateş böceklerine doğru yönelir. Parlaklık etkisi uzaklık arttıkça azalacağı için daha uzaktaki ateş böcekleri uzaktaki parlak ateş böceklerinden daha az etkilenir. Ayrıca bir ateş böceği eğer kendinden daha parlak bir ateş böceği bulamazsa rastgele hareket gerçekleştirir.

FireFly Algoritması akış diyagramı



Algoritmanın gereksinimlerinden bahsedelim

Test Fonksiyon : Algoritmada problem için bulduğumuz çözüm kümelerinin iyilik derecesini bulmamıza yarayan formül. Örn bu çalışma için Beale, Ackly ..

Popülasyon Boyutu: Popülasyondaki ateş böceği sayısının toplamıdır. Yani bir iterasyondaki çözüm kümesi sayısı.

Paremetre Aralığı: Parametrelerin alabileceği en büyük ve en küçük değerlerdir. Bu değer 4 fonksiyon içinde değişmektedir .

Baele(-4.5 , 4.5) , Ackley(-5.0 , 5.0) ,

Gold (-2.0 , 2.0) , Levi(-10.0 , 10.0)

α (alpha): Rastlantı değişkeni.

Υ (gama): Sabit emilim katsayısı.

Firefly Algorithm for Constrained Optimization

```
Input:
f(z), z = [z1, z2, ..., zn]T           {cost function}
S = [ak, bk], ∀k = 1, ..., n         {given constraints}
m, β0, γ, min ui, max ui             {algorithm's parameters}
Output:
ximin                               {obtained minimum location}

begin
  for i=1 to m do
    xi ← Generate.Initial.Solution ()
  end
  repeat
    imin ← arg mini f(xi)
    ximin ← arg minxi f(xi)
    for i=1 to m do
      for j=1 to m do
        if f(xj) < f(xi) then {move firefly i towards j}
          rj ← Calculate.Distance (xi, xj)
          β ← β0e-γrj {obtain attractiveness}
          ui ← Generate.Random.Vector (min ui, max ui)
          for k=1 to n do
            xi,k ← (1 - β)xi,k + βxj,k + ui,k
          end
        end
      end
    end
    uimin ← Generate.Random.Vector (min ui, max ui)
    for k=1 to n do
      ximin,k ← ximin,k + uimin,k {best firefly should move randomly}
    end
  until stop condition true
end
```

3. Benzetimli Tavlama

Benzetimli Tavlama Algoritması, yinelemeli rastgele bir arama tekniğidir ve çeşitli kombinyonel optimizasyon problemlerini çözmek için yaygın olarak kullanılmaktadır. Metropolis, Rosenbluth ve Teller tarafından metallerin tavlama işlemini simüle etme amacıyla önerilmiştir ve Kirkpatrick, Gelatt ve Veechi tarafından yinelemeli bir optimizasyon yöntemi olarak tanıtılmıştır. Benzetimli Tavlama algoritması başlangıç sıcaklığı, her sıcaklık değerinde üretilecek ,çözüm sayısı, sıcaklık azaltma fonksiyonu ve durdurma koşulu gibi parametrelere sahiptir.

```
Begin
  • Her bir çizelgeleme ölçütü için birer sıralama belirle ( S1, S2, S3 )
  • Belirlenen sıralamalar için amaç fonksiyon değerlerini hesapla
  • Bu amaç fonksiyon değerlerine göre başlangıç Pareto-optimal çözümü setini belirle ( P )
  • Başlangıç sıcaklığını seç ( T = 600 )
  • Sıcaklık azaltım oranını seç ( α = 0.9 )

  K=0
  repeat
    K=K+1  Aşama
    D=0
    repeat
      D=D+1  Arama yöntemi
      repeat
        i=0
        repeat
          i=i+1
          • Komşu arama mekanizmasına göre en iyi komşuyu seç ( S' )
          • Yeni çözüm için amaç fonksiyon değerlerini hesapla
          if Δ = ( fK(S') - fK(S) ) < 0 then yeni çözümü eskisiyle değiştir, P 'yi güncelle.
          else
            v ~ U(0,1) üret
            if e-Δ/(fK(S)-T) > v then çözümü eskisiyle değiştir.
          endif
        until ( i > 30 )
      until ( D > 2 )
    until ( K > 3 )
  end.
```

4. Yöntem

Çalışma Pycharm platformlarında Python programlama dili kullanılarak yapılmıştır. Ateş Böceği Algoritması belirlenen 4 adet Test Fonksiyonunu çalıştırarak sonuçlarını konsol a yazdırmaktadır.

- Problemin ve Popülasyon boyutu
- Test Fonksiyonu
- Fonksiyon değerlendirme sayısı
- Alfa parametresi
- Gamma parametresi
- Alt ve Üst sınır

Değerlerini girdi olarak alır.

Benzetimli Tavlama Algoritması ise kullanıcıdan optimizasyonda kullanılacak test fonksiyonunu belirlemesi istenmiştir. Seçilen Fonksiyona göre alt ve üst sınır programın içinde belirlenir ve optimizasyon sonucu grafiklerle ekrana yazdırılır.

5. Test Fonksiyonları Sonuçları

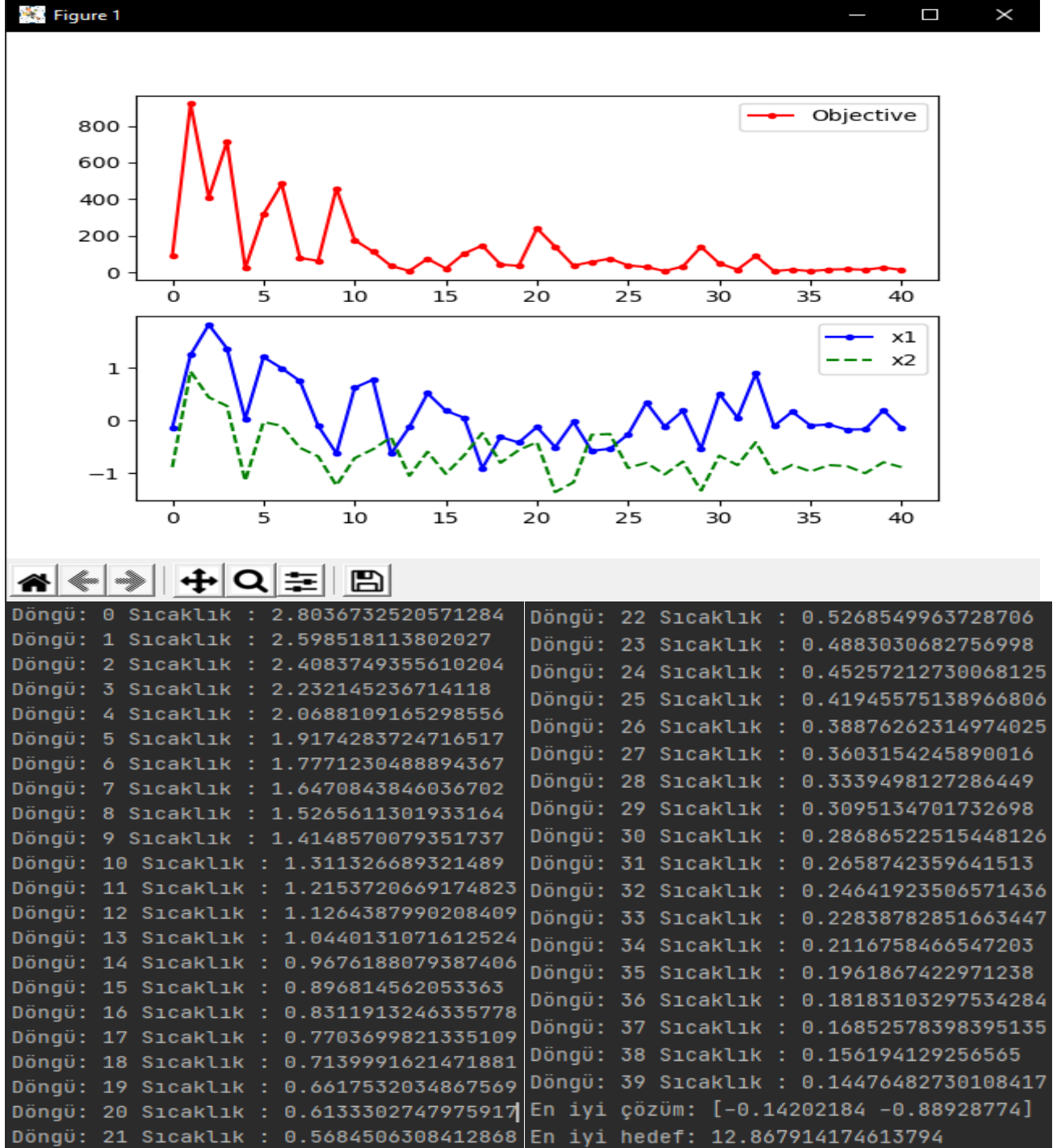
5.1 Goldstein Fonksiyonu(FireFly)

1857.0783863068123	11.865980551588935	11.263564963917853	5.441440125796273
926.0120865278562	35.97335155979985	6.932490063519483	3.2086772299135213
230.9989710609427	203.88888224343413	8.128638109143726	4.251396838420264
16728.653320370475	45.99794640502415	3.698836744320462	3.4107072442501307
36314.29874201377	159.25360831907236	3.289683881395659	5.9691029180148805
5052.25906104689	257.6201349632053	5.055385320237828	8.38874009633377
81879.01495882109	530.9557637040133	14.65635683725271	9.842020338826348
111.75599790443692	507.8532405240994	199.10214765815883	3.5596557007440954
135.40334739856638	766.7190113585199	31.284525104786752	3.0014746819539555
240503.82688222578	1918.130486598882	272.772382899537	3.0101941582384235
2552.1951675565856	992.5049244907755	34.09581180715458	3.022573804561911
3090.1867909198345	6194.014821021079	1171.6201983314868	3.0167208330060244
6427.969548443524	5611.419491583321	380.48399229907574	3.0212898662941385
2572.360156819465	4676.2847084001805	650.5780258715579	3.0334827636657717
230851.27898441025	2173.807357453116	3.026595648227985	3.0358463221683922
38777.87747421401	7867.954008199727	3.1877306386025532	3.0437259941254147
10226.002699435703	415.06277805826727	3.3099238625724228	3.107802909172666
824.5757078649026	3.7917506200864923	5.368199630075766	3.1319972359713457
19258.52124538188	4.330299892617635	3.26812324765263	3.1085905421020987

1-2-3-4

```
3.001727685307614
3.000239283914337
3.002275432535232
3.0104134966585296
3.008810677720671
3.015740622248377
Best Gold = 3.000000063228213
```

5.2 Goldstein Fonksiyonu(BTA)



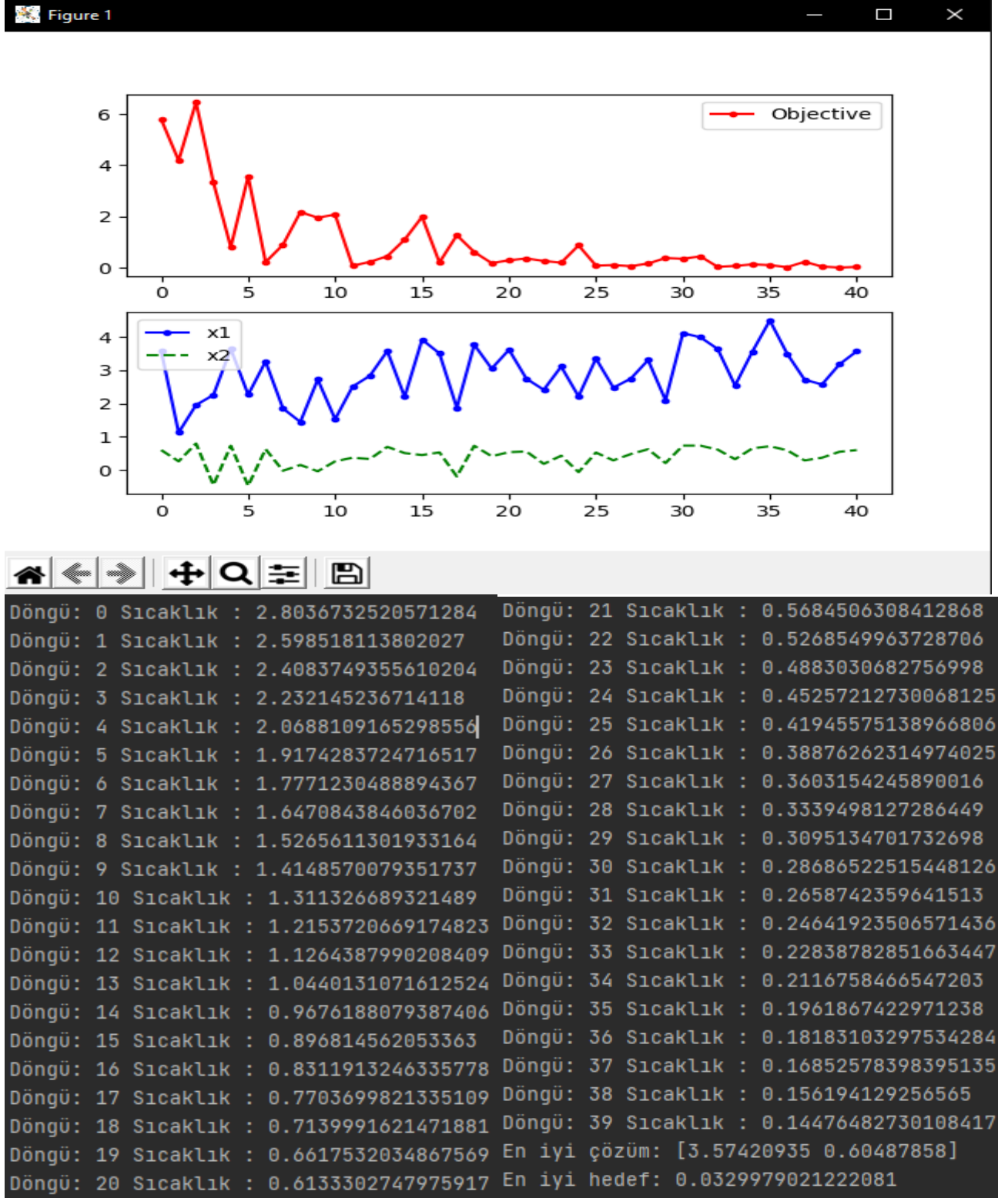
5.3 Beale Fonksiyonu (FireFly)

384.72859287962837	4.014786695193212	1.3722160090378095	1.0850164050590816
3.4381845035640186	55.110671715783866	1.0249716359748051	0.8660758748459477
81.38999100646615	47.904556520859074	24.950512818032635	0.8660647031977797
7.273386646716611	39.43532832888433	1.0703617436121358	0.8668873153173491
5268.77592522281	19.23943969429307	21.58373519910718	0.8660661542865176
16279.741700803508	36.627794124178656	4.136606608701253	0.8665743995281894
5193.457839494958	10.033292418888394	93.45793641200336	0.8677965262852978
41.323562781614704	16.606307397372845	20.54424752909072	0.8665903417949986
189.68056769573016	9.8136975963677	1.3897079803244015	0.8745856777305466
14519.596270079956	13.852045408635643	54.464129238562435	0.8823236548456456
914.778064013647	1.4165027654732592	45.3148311064799	0.8706341381465288
8855.46807249977	0.8907463480793703	35.26210241085461	0.8816321707226625
3025.4349003487214	1.1802642633601503	14.98765832190822	0.9647943063267581
716.6161369475966	1.6824729325378258	35.45197787640845	1.243511456750042
3.4381845035640186	5.412311818117245	0.8907463480793703	0.9412479322661519
4.5786945031204285	26.154813586791626	0.8684177300173322	1.7121993261838597
483.17009438044096	2.44016175058951	1.1590561284257594	1.055908988258957
33.077669039872326	21.685355954630495	0.9145387439651341	1.3491638020401064
74.45841234316391	1.5716998720264075	1.4123361474185314	1.342750585957714

1-2-3-4

0.9390257466158645
1.7096942654641976
1.0560925890333726
1.3469785557943406
1.3428867393342159
1.0685590403671794
Best Beale = 0.8660312081468228

5.4 Beale Fonksiyonu (BTA)



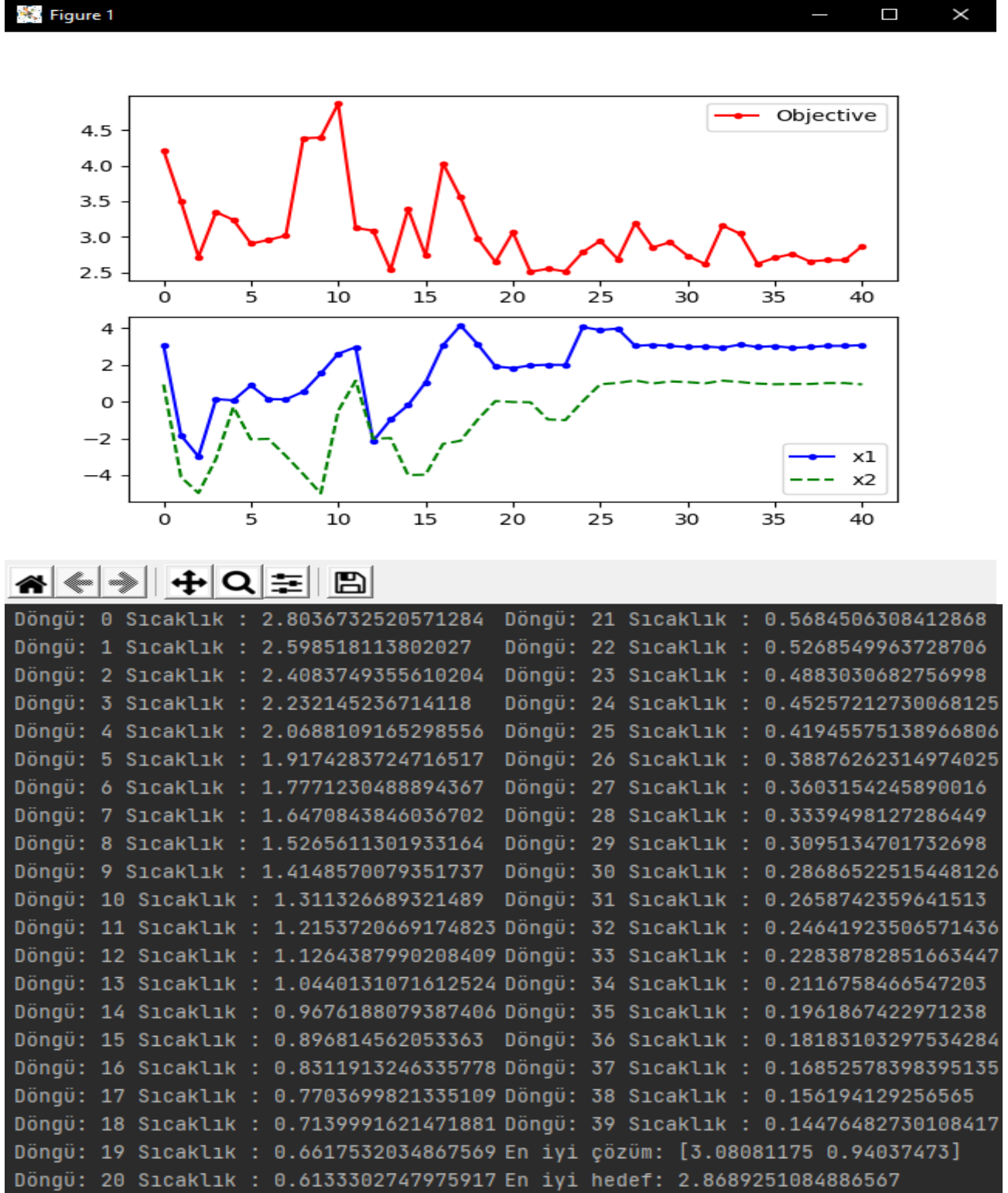
5.5 Ackley Fonksiyonu (FireFly)

4.622663580838061	4.2204206405655444	1.73744665136432	1.7213307329501721
4.413241575063982	4.12411075204775	1.7452140555810711	1.7215156314720472
4.449279498868392	4.325956929544435	1.7470575445256298	1.7218610341183478
4.445419358127902	3.8318647225671616	1.7703992543470815	1.7254348740549972
4.541456519920277	3.678150004632164	1.723485990843784	1.730084439726328
4.8055472194615305	2.9025776388873332	1.7939309297511916	1.7230002471486152
3.5152931050323817	4.384211135752585	1.7474883082901682	1.7453830304555442
4.9121776832563935	4.296155331672527	1.994381330497002	1.7328140250731803
4.749500222187855	2.778774748019663	2.616234101834642	1.7425091714716792
4.203027527762098	4.194629110081989	2.4218036609745957	1.7227993600055616
3.655226475988659	3.4340744285482323	3.2313857193154174	1.7846162504504628
4.120919955542394	3.6536202504694115	1.8270014889128743	1.747727847711782
4.817338798949578	4.128547371756836	3.7489242559138805	1.8083706827876305
3.203265325371329	3.682130280872107	2.820858417981934	1.8188117014656338
4.6068873901415275	4.003976483526108	2.5948218224437007	1.7297186303342356
4.5271857750759965	3.7142128975243582	4.275518507449215	1.8554925926279622
4.711100856389519	3.8432698630291267	4.250802518409428	1.805273722097291
4.310623724217722	2.1363418754810577	3.953458511742903	1.9788664138345124
2.724928024135331	3.6827804604385515	4.2931544517199995	2.458162047353554

1-2-3-4

1.7236976771876633
1.7260085945633672
1.7353899506026886
1.7222823171803916
1.741900696278579
Best Ackley = 1.7183815067427322

5.6 Ackley Fonksiyonu (BTA)



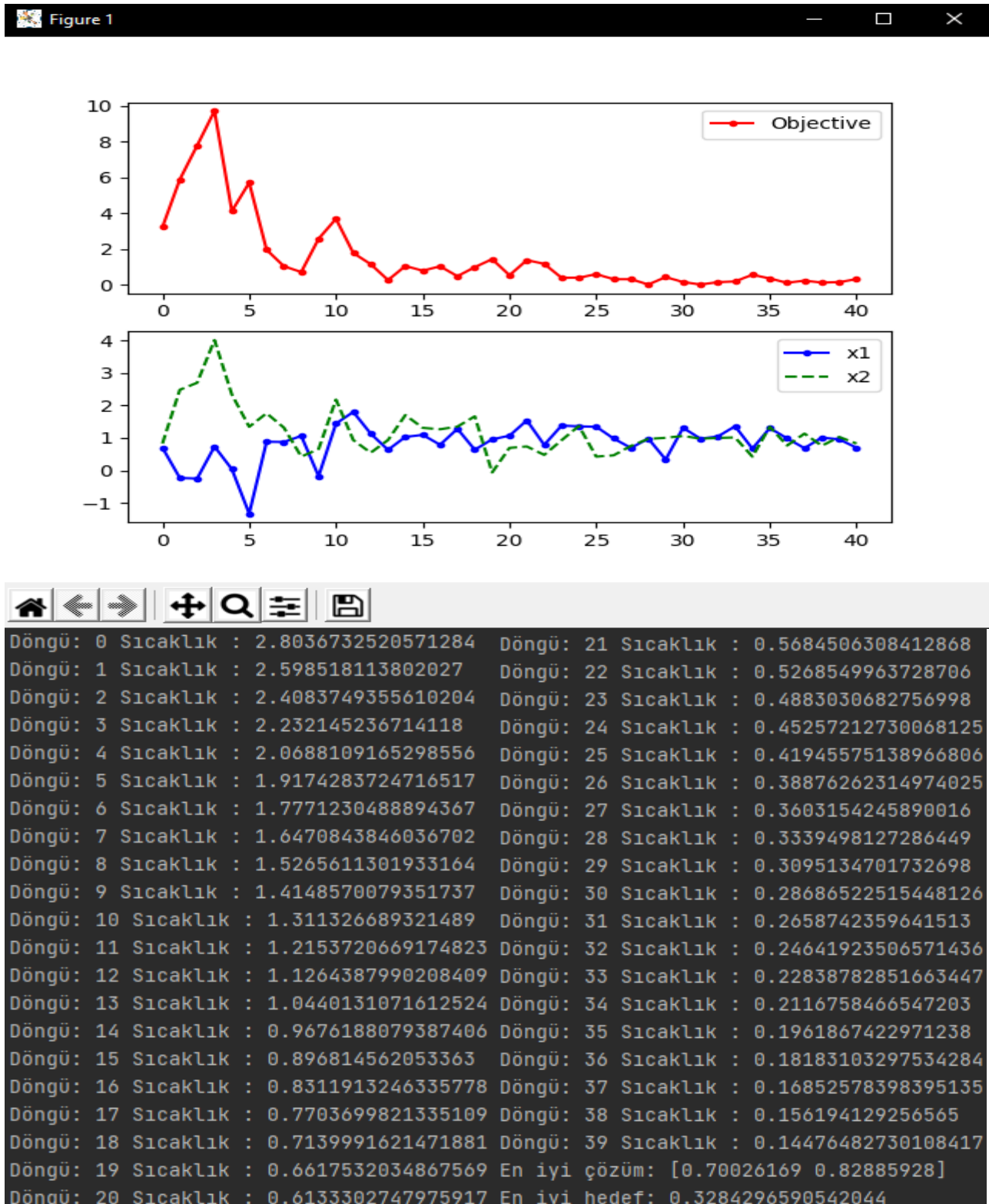
5.7 Levi Fonksiyonu (FireFly)

105.64628445184623	8.166351997645773	3.2411510448456062	0.03954352856352572
5.828633184461811	2.344208443715875	0.4532676725255234	0.00387206269151884
105.06164319517497	1.132818977868494	0.14499488915834763	0.02014686216937859
106.25718457742748	1.1928824695294744	2.0506328712228203	0.00638471956642674
72.65803289862471	5.44843686247251	2.7839744705476273	0.03498112454109943
88.80915067655522	9.372841759380618	5.767130699596646	0.1587077811103862
95.21558831413428	8.86460009318056	7.070775062535077	0.831129733140321
73.334152149104	15.258908076458436	13.721094651627368	0.01902076850234898
25.485803022358258	9.250175408857906	9.323436708666991	0.933472084933183
188.26534122784318	21.92418494201216	0.00586261435562538	1.0380724369629761
55.136829635244204	10.119925284500098	0.10342630577365526	0.2438848608651986
13.826071523495866	13.849930827591105	0.8190582437304187	1.4032224915116926
19.530764766623832	8.018550992810408	0.432581376201951	0.3376503187388695
142.36225224193961	14.687492820677114	0.8845716977946942	3.388886368616238
96.44379752167441	9.12013103060531	1.025446997513141	3.463799725440107
121.60893463882158	0.14411739976262208	0.4397849439458646	1.089910777184384
44.53427973154049	0.562216111449081	1.2941772794861734	3.248395286158792
58.7565331675174	0.8587234948730326	0.3359257276903246	0.00121282781335020
25.50914352776255	1.8191523187601994	3.7386794631359574	0.00114264549909011

1-2-3-4

0.0297908490636319
0.9139931383264746
1.0376849920553788
0.2843858562707493
1.379687331598516
Best Levi = 0.0010650976335289733

5.8 Levi Fonksiyonu (BTA)



6. Sonuç

Beale, Ackley, Goldstein ve Levi test fonksiyonları kullanılarak Ateş Böceği ve Benzetimli Tavlama optimizasyon algoritmaları uygun girdi değerleriyle çalıştırıldığında her iki algoritmanın da fonksiyonlar için minimize değerlere verdiği görülmektedir. Ancak Goldstein Fonksiyonu Benzetimli Tavlama algoritmasında optimize olamamıştır bunun sebebi olabilecek durumlar 1. Pycharm üzerinde yazılan python kodunda hata 2. Döngü sayısının az miktarda olması gibi durumlar olabilir. Bunun dışında her iki algoritma içinde başarılı bir şekilde optimizasyon işlemlerini gerçekleştiren algoritmalar olduğu söylenebilmektedir. İterasyon sayılarına bakıldığında Benzetimli Tavlama Algoritmasının Guguk Kuşu Algoritmasına göre çok az adım sayısında değerleri optimize edebildiği görülmektedir.

7.KAYNAKALAR

1. <https://bilalsaim.com/yapay-zeka/ates-bocegi-algoritmasi-fa-firefly-algorithm/#:~:text=2008%20Y%C4%B1l%C4%B1nda%20Xin-She%20Yang,sosyal%20davran%C4%B1%C5%9Flar%C4%B1n%C4%B1%20ele%20olarak%20geli%C5%9Ftirilmi%C5%9Ftir.&text=Daha%20az%20parlak%20olan%20ate%C5%9F%20b%C3%B6cekleri%20%C3%A7ekici%20olan%20ate%C5%9F%20b%C3%B6ceklerine%20do%C4%9Fru%20y%C3%B6nelir.>
2. <http://acikerisim.ktu.edu.tr/jspui/handle/123456789/306>
3. <https://www.sciencedirect.com/science/article/pii/S1319157818300673>