NECMETTİN ERBAKAN
ÜNİVERSİTESİ

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Bulanık Mantık Final Ödevi Raporu

Proje Başlığı
BOHACHEVSKY FUNCTIONS

Öğrenci Bilgileri	
Öğrenci No	19010011019
Öğrenci Ad Soyad	Ahmet Furkan DEMİR

Dr. Öğr. Üyesi
Ayşe Merve ACILAR

Ocak 2023
Konya

İçindekiler

1. MyFis Tasarımı:

Projemde **Bohachevsky Fonksiyonuna** ait bulanık mantık sistemini oluşturacağım. Öncelikle **Bohachevsky** fonksiyonunu tanıyalım.

Bohachevsky Fonksiyonu:

$$f_1(\mathbf{x}) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$
$$f_2(\mathbf{x}) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$$
$$f_3(\mathbf{x}) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$$

Şekil 1 - Bohachevsky Func.

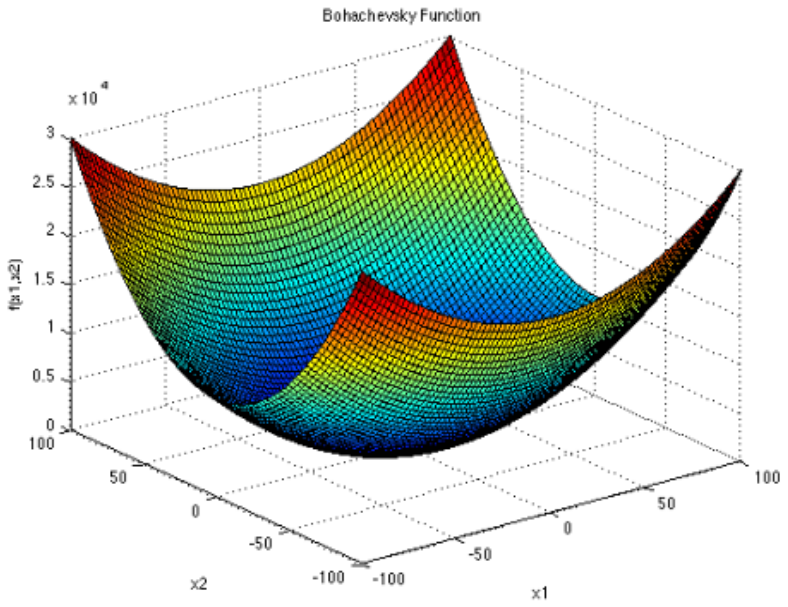
Değer aralığı:

Input Domain:

The functions are usually evaluated on the square $x_i \in [-100, 100]$, for all $i = 1, 2$.

Şekil 2 - Değer Aralığı

Bohachevsky Grafiği:



Şekil 3 - Bohachevsky Func. Surface

İlk işlem olarak veri seti oluşturuldu. Bu işlemi yapılırken matlab'in 'randi' fonksiyonu kullanıldı. Veri seti oluşturulurken fonksiyonun değer aralığını baz alarak -100 ile 100 değerleri arasında 125 adet random reel sayı üretildi ve matris biçiminde workspace'e çekildi. Daha sonra y değerlerini elde etmek için fonksiyonun sitesindeki matlab implementation da yararlanılarak yazmış olduğumuz fonksiyonumuza bu veri seti sokuldu ve y değerleri elde edildi. Daha sonra veri seti %60'ı eğitim (75), %40'ı test (50) için olacak şekilde ikiye bölündü.

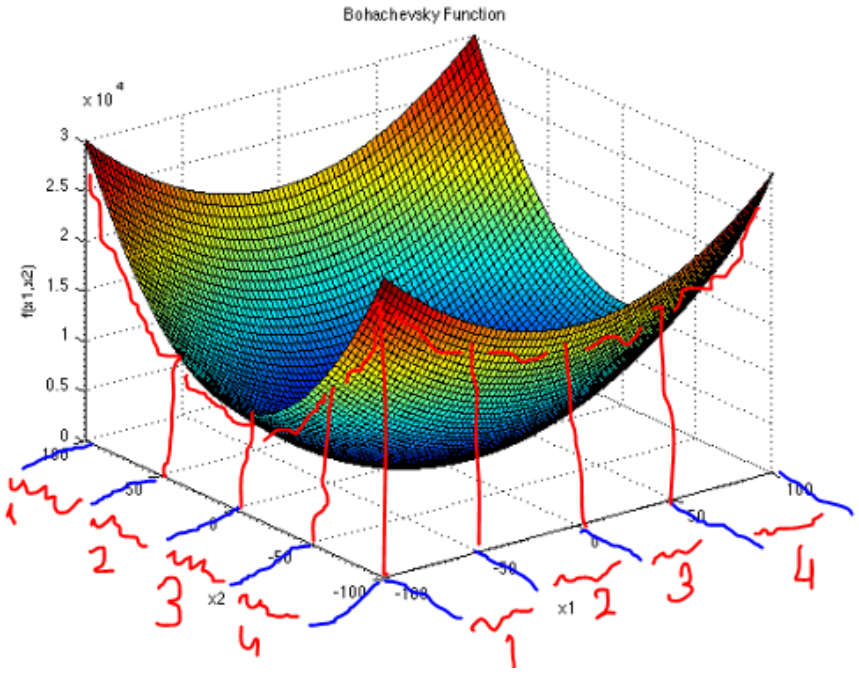
1	X1	X2	Y
2	-40	15	4800
3	40	37	3778
4	33	9	1217,6
5	8	-15	3264
6	40	29	3778
7	33	30	9539,6
8	-65	36	15475,6
9	-75	27	25625,6
10	100	89	18712
11	-66	-59	22028
12	-94	42	9124
13	12	-53	12002
14	77	-77	8241,6
15	34	22	8844
16	-62	-10	5196
17	-26	-8	804
18	-8	33	18882
19	97	54	18931,6
20	-69	-30	14843,6
21	71	33	6723,6
22	29	-17	2091,6
23	-25	69	8313,6
24	-62	67	4236
25	-14	-49	228
26	-4	23	11568

Şekil 4 - Eğitim Veri Setinden Görünüm

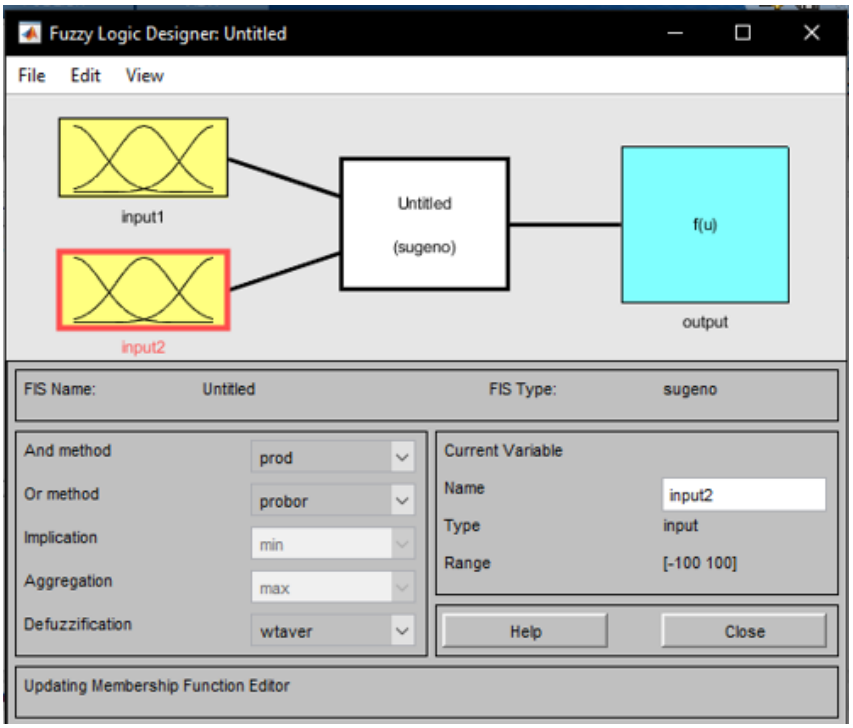
1	X1	X2	y_gerçek
2	-30	29	8342
3	-61	-24	8721,6
4	-50	63	3558
5	23	7	579,6
6	-5	-30	1825,6
7	-30	88	9612
8	66	76	4934
9	17	10	489,6
10	10	25	14212
11	84	17	10754
12	-43	-59	7257,6
13	52	-40	7906
14	51	-6	3753,6
15	-24	-54	968
16	14	69	14646
17	-85	-61	23425,6
18	-90	-55	8172
19	6	-66	6308
20	56	-55	18274
21	87	-13	18521,6
22	-74	-38	5868
23	14	85	268
24	-6	-14	19244
25	-98	-63	11782
26	-33	81	10337,6

Şekil 5 - Test Veri Setinden Görünüm

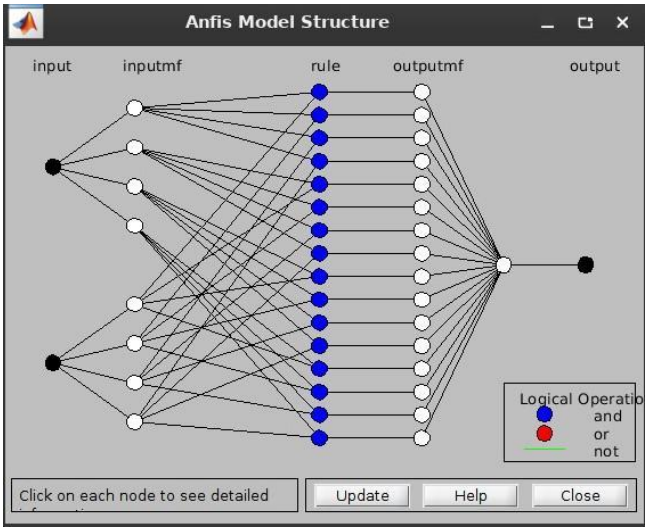
Sugeno modeli oluşturuldu. İki giriş ve bir çıkış ile oluşturulan sistemin her iki girişi de 4'er dilsel değere bölündü. Değer aralığı olarak -100 100 değerleri verildi. Her iki dilsel ifadenin adedinin belirlenmesinde grafiğin aralıkla izlediği şekiller dikkate alındı.



Şekil 6 - Dilsel Değerler Adedinin Belirlenmesi



Şekil 7 - Sistem, Girişler, Değer Aralıkları



Şekil 8 - Anfis Giriş, Kural ve Dilsel değerlerin şematik gösterimi

Oluşturulan sistem "BM_19_019MyFis.fis" şeklinde adlandırılarak dışarıya alındı ve projenin bir sonraki aşamasına geçildi.

2. Genetik Algoritma ile Bulanık Sistem Tasarımı:

Kromozom yapısı tasarlandı. Popülasyon uzunluğu olarak 5 değerinde kara kılındı.

Girişler: X ve Y																Çıkışlar					
X								Y								f1			f2		
A1		A2		A3		A4		B1		B2		B3		B4		Kural1			Kural2		
C_A1	Sigm	C_A2	Sg_A	C_A3	Sigm	C_A4	Sg_A	C_B1	Sigm	C_B2	Sg_B	C_B3	Sigm	C_B4	Sg_B	p1	q1	r1	p2	q2	r2
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

Şekil 9 - Kromozom Yapısı

f3			f4		
Kural3			Kural4		
p3	q3	r3	p4	q4	r4
23	24	25	26	27	28

f5			f6		
Kural5			Kural6		
p5	q5	r5	p6	q6	r6
29	30	31	32	33	34

f7			f8		
Kural7			Kural8		
p7	q7	r7	p8	q8	r8
35	36	37	38	39	40

f9			f10		
Kural9			Kural10		
p9	q9	r9	p10	q10	r10
41	42	43	44	45	46

f11			f12		
Kural11			Kural12		
p11	q11	r11	p12	q12	r12
47	48	49	50	51	52

f13			f14		
Kural13			Kural14		
p13	q13	r13	p14	q14	r14
53	54	55	56	57	58

f15			f16		
Kural15			Kural16		
p15	q15	r15	p16	q16	r16
59	60	61	62	63	64

Şekil 10 - Kromozom Yapısı (devamı)

“randi” hazır fonksiyonundan yararlanılarak -100 100 değer aralığında kromozomlar rastgele olarak oluşturuldu. Ekran görüntüsü olarak sığmadığı için ilk 5 genleriyle birlikte kromozomlar:

Kromozom1	10	-90	37	28	48
Kromozom2	-3	-86	-74	-34	-53
Kromozom3	78	-83	45	31	47
Kromozom4	60	60	-78	50	95
Kromozom5	47	89	-77	17	74

Şekil 11 - Kromozom Görünüm

Kromozomlar sırayla “eval” fonksiyonuna sokuldu ve alınan sonuçlar excel dosyasına yazıldı.

Eval(Kromozom1):	1,5200
Eval(Kromozom2):	2,07205
Eval(Kromozom3):	1,72699
Eval(Kromozom4):	1,6784
Eval(Kromozom5):	1,6959

Şekil 12 - Eval(KromozomX)

Kromozomların “eval” sonuçlarına 1 / Uygunluk, [1 / Uygunluk] / Toplam Uygunluk işlemleri sırasıyla uygulandı.

	1/Uygunluk	[1/Uygunluk]/ToplamUyg
1,5200	0,657894737	0,226469012
2,07205	0,482613837	0,16613156
1,72699	0,579042148	0,199325357
1,6784	0,595794880	0,205092198
1,6959	0,589664363	0,202981873
ToplamUyg:	2,905009965	1

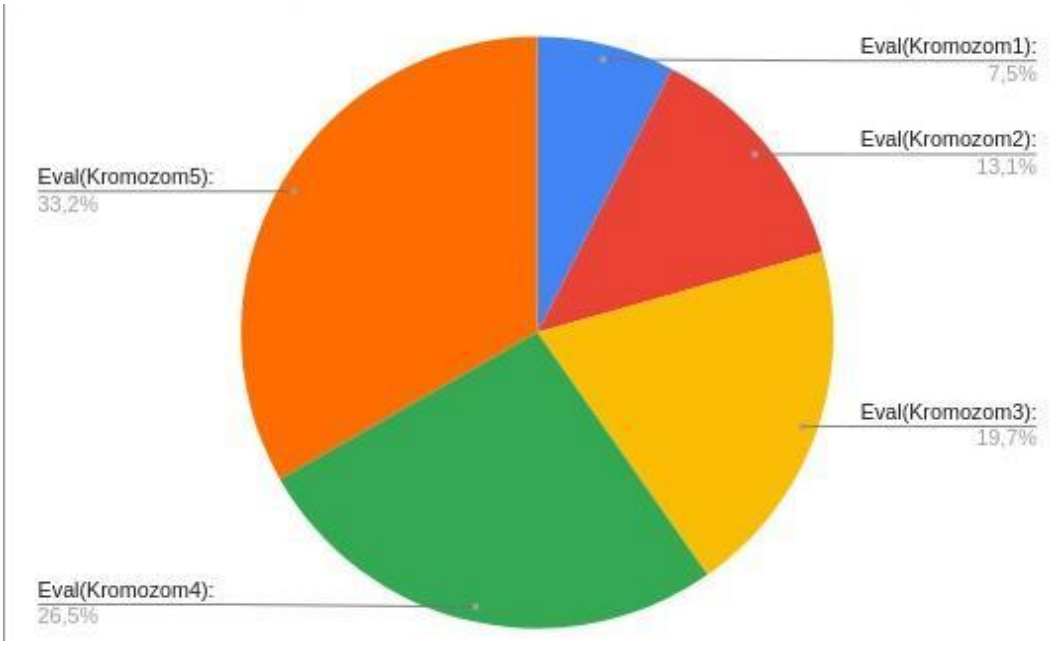
Şekil 13 - Matematiksel İşlemler

Elde edilen sonucun kümülatif toplamı alınarak kromozomların pasta grafiğindeki temsil alanları belirlendi.

[1/Uygunluk]/ToplamUyg	KumulatifToplam
0,226469012	0,226469012
0,16613156	0,392600572
0,199325357	0,591925929
0,205092198	0,797018127
0,202981873	1
1	

Şekil 14 - Kümülatif Toplam

Pasta Grafiği oluşturuldu.



Şekil 15 - Pasta Grafiği

Rastgele sayılar ürettirilerek pasta grafiği üzerinde "rulet" tekerleği işlemi gerçekleştirildi.

KumulatifToplam		RastgeleSayı		Seçilen
0,226469012		0,1129058826		Kromozom1
0,392600572		0,9398268500		Kromozom5
0,591925929		0,3325665290		Kromozom2
0,797018127		0,4785657824		Kromozom3
1		0,7596257942		Kromozom4

Şekil 16 - Rulet Tekerleği İşlemi

Yeni populasyon					
Kromozom1' (eski Kromozom1):	10	-90	37	28	48
Kromozom2' (eski Kromozom5):	47	89	-77	17	74
Kromozom3' (eski Kromozom2):	-3	-86	-74	-34	-53
Kromozom4' (eski Kromozom3):	78	-83	45	31	47
Kromozom5' (eski Kromozom4):	60	60	-78	50	95

Şekil 17 - Yeni Popülasyon

Çaprazlama işlemi için "pc"1.68 olarak belirlendi. Sonuç olarak 1. ve 4. kromozomlarımız çaprazlama için seçilmiş oldu.

Kromozom1'	Kromozom2'	Kromozom3'	Kromozom4'	Kromozom5'
1,5200	2,07205	1,72699	1,6784	1,6959

Şekil 18 - Çaprazlama / Seçilen kromozomlar

20 adet rastgele Beta değeri üretildi ve çaprazlama işlemi ilgili formülden yararlanılarak excel üzerinde gerçekleştirildi. Elde edilen değer tam sayıya çevrildi.

$$P_{yeni}=\beta P_{an}+(1-\beta)P_{bn}$$

β= 0 ve 1 arasında üretilen rasgele sayı

P_{an}= Anne kromozomun n. parametresi

P_{bn}= Baba kromozomun n. parametresi

Şekil 19 - Çaprazlama Formülü

β	0,3179	0,9741	0,4362	0,6411	0,4539	0,3932	0,8786	0,2125	0,7836
Kromozom1''	56	-90	41	29	47	-49	43	-5	9
Kromozom4''	31	-84	40	29	47	-61	-45	-45	44
Kromozom2' (eski Kromozom5):	47	89	-77	17	74	20	55	29	-41
Kromozom3' (eski Kromozom2):	-3	-86	-74	-34	-53	-27	-27	-22	-70
Kromozom5' (eski Kromozom4):	60	60	-78	50	95	37	-83	-54	-80

Şekil 20 - Çaprazlama sonrası ilk 9 gen

Mutasyon işlemine geçildi. "pm" değeri öncelikle 10 olarak seçildi ve excel kodlaması yapılarak ilgili genlerin otomatik olarak mutasyona uğraması sağlandı. Mutasyona uğrayan genlere ödevde belirtildiği üzre otomatik olarak 1-5 arasında rastgele değerler atandı. Mutasyona uğrayan genler renklendirildi.

Mutasyon1 pm=10					
Kromozom1''	76	5	39	28	47
Kromozom4''	11	2	42	30	47
Kromozom2' (eski Kromozom5	47	89	5	17	74
Kromozom3' (eski Kromozom2	2	2	5	4	1
Kromozom5' (eski Kromozom4	60	60	5	50	95

Şekil 21 - 1. Mutasyon İşlemi

İkinci mutasyon işlemi için "pm" değeri 20 olarak belirlendi ve ilk mutasyon işleminde yapılan işlemin tekrarı sağlandı yine mutasyona uğrayan genlerin renklendirilmesi yapılarak kolayca ayırt edilmesi sağlandı.

Mutasyon2 pm=20					
Kromozom1''	76	5	39	28	47
Kromozom4''	3	4	42	30	47
Kromozom2' (eski Kromozom5	47	89	2	1	74
Kromozom3' (eski Kromozom2	4	1	1	5	1
Kromozom5' (eski Kromozom4	60	60	5	50	95

Şekil 22- 2. Mutasyon İşlemi

3. Çaprazlama ve Mutasyon İşlemini Python ile Otomatik Hale Getirme:

2. adımda yapılan tüm işlemleri yani **Genetik Algoritma ile Bulanık Sistem Tasarımı** bölümünü Python programlama dili ile otomatik hale getirdim.

Bu aşamda Pandas, Numpy ve Random modülünü kullanarak kromozomları seçtim ardından çaprazlama işlemini tamamlayıp iki defa mutasyona uğratarak yeni popülasyonu elde ettim.

Bu aşama için yazdığım koda **BM_19_019CaprazlamaMutasyonKodu** klasörü içersinden erişebilirsiniz.

```
import numpy as np
import pandas as pd
import random
import os

# matlab dan gelen popilasyon
populasyon = pd.read_csv("popilasyon.csv", index_col=False)

# matlab de hesapladigimiz evalfis den gelen sonuclar
evall = {"Eval1":[115200],
        "Eval2":[2.07205],
        "Eval3":[1.72699],
        "Eval4":[1.6784],
        "Eval5":[1.6959]}

evalT = {"Eval1":[115200],
        "Eval2":[2.07205],
        "Eval3":[1.72699],
        "Eval4":[1.6784],
        "Eval5":[1.6959]}

# rastgele sayi, kromozom secmek icin kullancagiz
# exeldeki ile aynı olsun diye yani kontrol edebilesiniz diye ayni
rastgele sayilari sectim
rastgeleSayi = [0.11290588, 0.93982685, 0.33256652, 0.47856578,
0.75962579]

beta = np.array([0.8244, 0.0117, 0.4301, 0.1870, 0.6426, 0.9354,
0.1822, 0.8162, 0.6803, 0.0635, 0.0338, 0.1836, 0.1359, 0.6243,
0.4112, 0.9744, 0.9598, 0.7771, 0.4519, 0.3561, 0.5172, 0.1885,
0.7694, 0.3464, 0.8688, 0.5242, 0.9900, 0.7332, 0.5516, 0.0764,
```

```
0.2203, 0.1936, 0.6169, 0.2284, 0.2467, 0.1240, 0.3391, 0.9635,
0.1895, 0.0928, 0.4494, 0.1672, 0.5399, 0.7593, 0.9655, 0.0427,
0.9922, 0.5632, 0.6915, 0.5628, 0.2511, 0.4822, 0.2444, 0.5124,
0.0408, 0.6811, 0.0246, 0.3446, 0.8733, 0.9290, 0.8970, 0.7680,
0.7417, 0.2505])

# evalfisdeki degerleri bire boldum
def toBirBoluUygunluk(dictt):

    dictt["Eval1"][0] = 1/dictt["Eval1"][0]
    dictt["Eval2"][0] = 1/dictt["Eval2"][0]
    dictt["Eval3"][0] = 1/dictt["Eval3"][0]
    dictt["Eval4"][0] = 1/dictt["Eval4"][0]
    dictt["Eval5"][0] = 1/dictt["Eval5"][0]

    toplam = dictt["Eval1"][0] + dictt["Eval2"][0] +
dictt["Eval3"][0] + dictt["Eval4"][0] + dictt["Eval5"][0]

    return dictt, toplam

# (1/eval)/toplam
def uygunlukToplam(dictt, toplam):

    dictt["Eval1"][0] = dictt["Eval1"][0]/toplam
    dictt["Eval2"][0] = dictt["Eval2"][0]/toplam
    dictt["Eval3"][0] = dictt["Eval3"][0]/toplam
    dictt["Eval4"][0] = dictt["Eval4"][0]/toplam
    dictt["Eval5"][0] = dictt["Eval5"][0]/toplam

    return dictt

# kumeletif toplam
def kumeletifToplam(dictt):

    tempEval1 = dictt["Eval1"][0]
    tempEval2 = dictt["Eval1"][0] + dictt["Eval2"][0]
    tempEval3 = dictt["Eval1"][0] + dictt["Eval2"][0] +
dictt["Eval3"][0]
    tempEval4 = dictt["Eval1"][0] + dictt["Eval2"][0] +
dictt["Eval3"][0] + dictt["Eval4"][0]
    tempEval5 = dictt["Eval1"][0] + dictt["Eval2"][0] +
dictt["Eval3"][0] + dictt["Eval4"][0] + dictt["Eval5"][0]
```

```
    dictt["Eval1"][0] = tempEval1
    dictt["Eval2"][0] = tempEval2
    dictt["Eval3"][0] = tempEval3
    dictt["Eval4"][0] = tempEval4
    dictt["Eval5"][0] = tempEval5

    return dictt

# kromozom secme ve yerlerini degistirme
def secilen(kumeletifToplams, rastgeleSayi, populasyon):

    kumeletifToplams = [kumeletifToplams["Eval1"][0],
kumeletifToplams["Eval2"][0],
                            kumeletifToplams["Eval3"][0],
kumeletifToplams["Eval4"][0], kumeletifToplams["Eval5"][0]]

    secilenA = []

    for i in range(0,5):

       if rastgeleSayi[i]<kumeletifToplams[0]:

            secilenA.append("Kromozom1")

       elif rastgeleSayi[i]>kumeletifToplams[0] and
rastgeleSayi[i]<kumeletifToplams[1]:

            secilenA.append("Kromozom2")

       elif rastgeleSayi[i]>kumeletifToplams[1] and
rastgeleSayi[i]<kumeletifToplams[2]:

            secilenA.append("Kromozom3")


       elif rastgeleSayi[i]>kumeletifToplams[2] and
rastgeleSayi[i]<kumeletifToplams[3]:

            secilenA.append("Kromozom4")

       elif rastgeleSayi[i]>kumeletifToplams[3] and
```

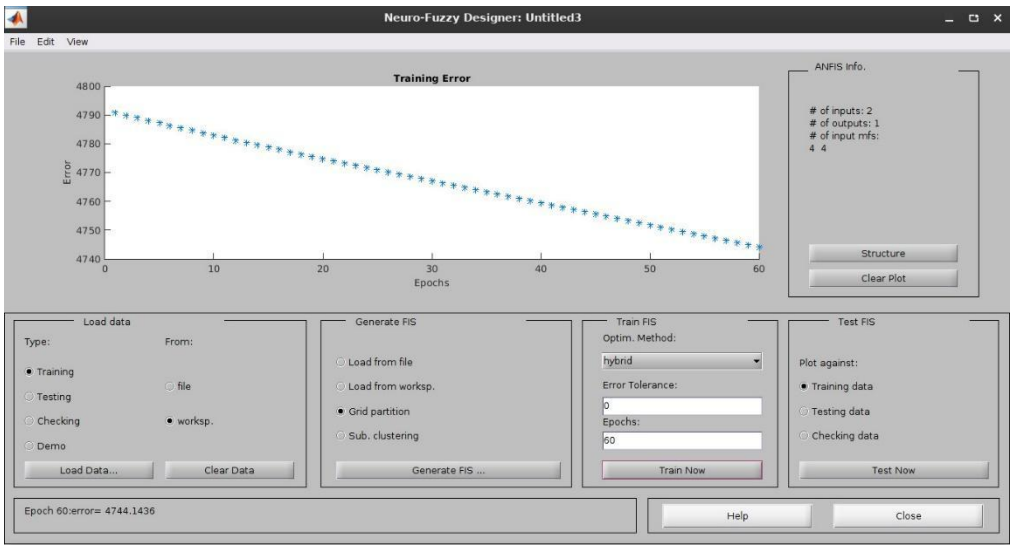
```
rastgeleSayi[i]<kumeletifToplams[4]:

            secilenA.append("Kromozom5")


    dictt =
{"Kromozom1":list(populasyon[secilenA[0]].to_dict().values()),
"Kromozom2":list(populasyon[secilenA[1]].to_dict().values()),

"Kromozom3":list(populasyon[secilenA[2]].to_dict().values()),
"Kromozom4":list(populasyon[secilenA[3]].to_dict().values()),
"Kromozom5":list(populasyon[secilenA[4]].to_dict().values())}


    newPopulasyon = pd.DataFrame.from_dict(dictt)
    return newPopulasyon

# caprazlanacak kromozomların secimi ve caprazlama islemi
def caprazla(newPopulasyon, evalT, deger, beta):

    capraz = ["Kromozom1"]

    if evalT["Eval1"][0] <= deger:

       capraz.append("Kromozom1")

    if evalT["Eval2"][0] < deger:

       capraz.append("Kromozom2")

    if evalT["Eval3"][0] < deger:

       capraz.append("Kromozom3")

    if evalT["Eval4"][0] < deger:

       capraz.append("Kromozom4")

    if evalT["Eval5"][0] < deger:

       capraz.append("Kromozom5")
```

```
    temp = newPopulasyon[capraz[0]] * beta +
newPopulasyon[capraz[1]] * (1-beta)
    temp1 = newPopulasyon[capraz[0]] * (1-beta) +
newPopulasyon[capraz[1]] * beta

    newPopulasyon[capraz[0]] = temp
    newPopulasyon[capraz[1]] = temp1

    return newPopulasyon

# mutasyon islemi
def mutasyon(caprazPop, pm):


    for i in caprazPop:

       counter = -1

       caprazPop[i] = pd.to_numeric(caprazPop[i], downcast="float")

       for j in caprazPop[i]:

            counter+=1

            if j < pm:

                 caprazPop[i][counter] = float(random.random())


    return caprazPop


birBoluUygunluk, toplam = toBirBoluUygunluk(evall)

uygunlukToplamS = uygunlukToplam(birBoluUygunluk, toplam)

kumeletifToplams = kumeletifToplam(uygunlukToplamS)

newPopulasyon = secilen(kumeletifToplams, rastgeleSayi,
populasyon)
```

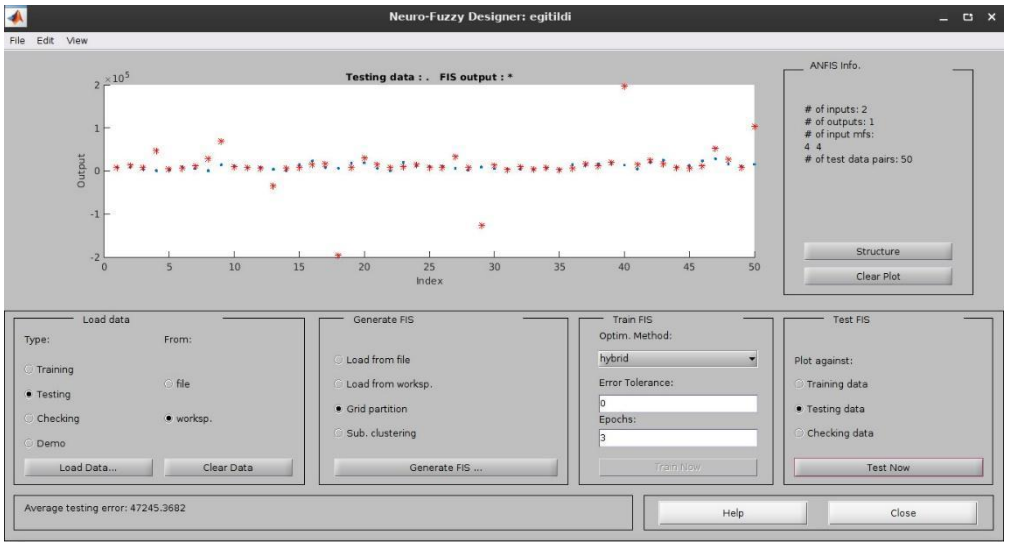
```
caprazPop = caprazla(newPopulasyon, evalT, 1.68, beta)

mutasyon1 = mutasyon(caprazPop, 10)

mutasyon2 = mutasyon(mutasyon1, 20)

mutasyon2.to_csv('newPopulasyon.csv', index=False)
```

4. Anfis Toolbox ile Eğitim:

Anfis Toolbox kullanılarak eğitim gerçekleştirildi.



Şekil 23 - Eğitim Kaybı

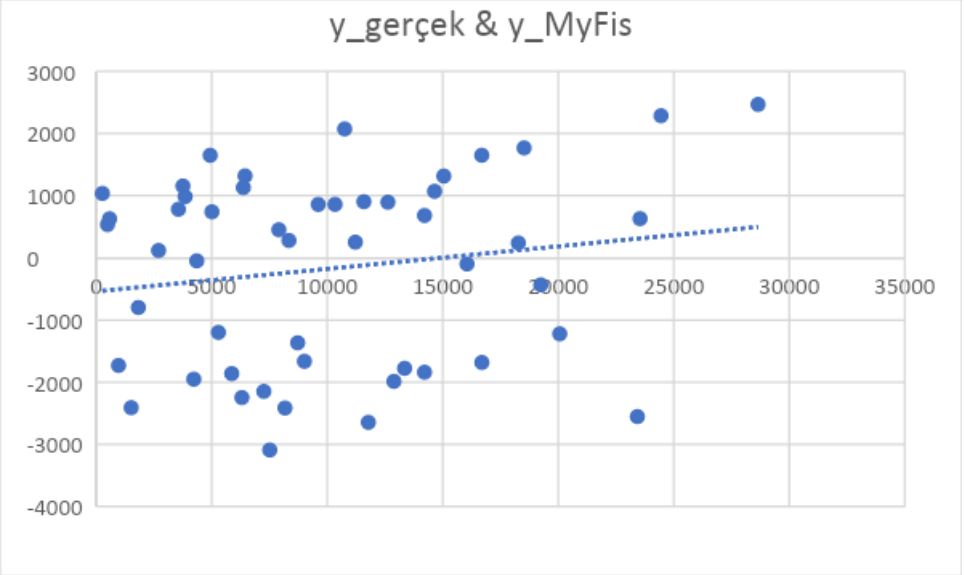


Şekil 24 - Anfis Toolbox üzerinde test sonuçları

5. Başarım Testi ve Sonuçlar:

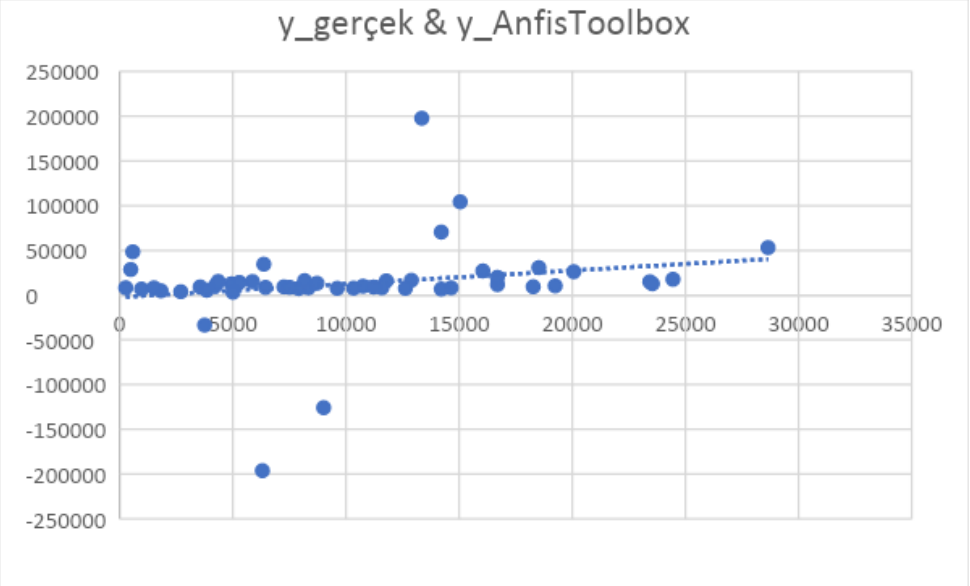
X1	X2	y_gerçek	y_MyFis	Y_AnfisToolbox
-30	29	8342	280,3819175	8174,351622
-61	-24	8721,6	-1365,92372	13248,94533
-50	63	3558	782,0275011	9142,453166
23	7	579,6	629,4628526	48370,38124
-5	-30	1825,6	798,1382657	4808,425423
-30	88	9612	859,7978924	7585,209228
66	76	4934	1649,666059	12642,6007
17	10	489,6	536,1893302	28628,45662
10	25	14212	682,2222913	70438,45016
84	17	10754	2073,993615	9962,977303
-43	-59	7257,6	2147,524459	9215,228308
52	-40	7906	452,8545397	7319,409733
51	-6	3753,6	1155,996695	-33461,20506
-24	-54	968	1729,980119	6804,275501
14	69	14646	1071,362246	8177,332437
-85	-61	23425,6	2552,319522	14952,82053
-90	-55	8172	2414,261654	16199,58955
6	-66	6308	-2246,49771	-196105,4138
56	-55	18274	239,9214747	9348,693725
87	-13	18521,6	1768,882161	30701,06266
-74	-38	5868	-1861,12859	15199,09314
14	85	268	1036,544934	8222,657839
-6	-14	19244	433,8515526	10376,32327
-98	-63	11782	2645,180432	15943,69693
-33	81	10337,6	860,4636517	7807,317564
-68	96	11586	904,2911417	8201,563309
59	-12	6369,6	1130,913139	34537,39829
-38	-78	1516	2408,714172	8123,961095
6	-49	9014	1663,712301	-125714,3699
-67	-18	5289,6	1198,393883	14210,76106
20	19	5008	740,8918742	3194,632214
-48	-48	4226	1950,711793	10339,24164
31	21	3849,6	986,3167692	5382,494453
38	42	6444	1318,93193	8619,56354
50	-56	2700	119,2179828	3828,557887
-10	-77	14212	1839,594348	6737,142967
-84	-41	12888	1986,157758	16574,40668
-54	-36	16694	1682,181864	11739,32578
83	-15	16689,6	1650,982835	19864,28372
-70	2	13350	1775,604806	197564,4362
65	-83	4353,6	47,95366409	15313,2539
8	-48	20064	1221,680178	26149,28386
100	61	24450	2286,173246	17661,89622
-85	-95	7513,6	3089,883592	8732,424035
-12	86	12626	895,4086637	7596,912733
-79	46	23539,6	631,0356864	12675,10333
93	-2	28649,6	2468,425871	53104,56412
-100	16	16050	97,50321816	27023,98468
55	-53	11217,6	252,5940653	8995,854661
64	-8	15048	1316,200459	104220,7673

Test veri setinin y değerleri ile My_Fis'in tahmini y değerlerinin saçılım grafiğinde gösterimi.



Şekil 25 - y_gerçek & y_MyFis Saçılım Grafiği

Test veri setinin y değerleri ile Anfis Toolbox'ın tahmini y değerlerinin saçılım grafiğinde gösterimi.



Şekil 26 - y_gerçek & y_AnfisToolBox Saçılım Grafiği

Kaynaklar:

https://www.sfu.ca/~ssurjano/optimization.html