

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DİFERANSİYEL(FARKSAL) GELİŞİM ALGORİTMASI
KULLANILARAK KISIT YÖNETİM METOTLARININ
PERFORMANSLARININ KARŞILAŞTIRILMASI**

**Tezi Hazırlayan
Serdar Kasım KARATAŞ**

**Tezi Yöneten
Yrd. Doç. Dr. Mustafa DANACI
Doç. Dr. M. Fatih TAŞGETİREN**

Yüksek Lisans Tezi

**Temmuz 2011
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DİFERANSİYEL(FARKSAL) GELİŞİM ALGORİTMASI
KULLANILARAK KISIT YÖNETİM METOTLARININ
PERFORMANSLARININ KARŞILAŞTIRILMASI
(Yüksek Lisans Tezi)**

**Tezi Hazırlayan
Serdar Kasım KARATAŞ**

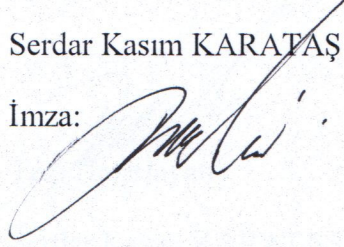
**Tezi Yöneten
Yrd. Doç. Dr. Mustafa DANACI
Doç. Dr. M. Fatih TAŞGETİREN**

**Temmuz 2011
KAYSERİ**

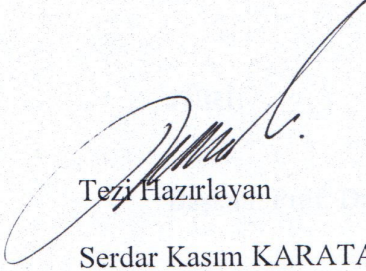
Bu alıřmadaki tm bilgilerin, akademik ve etik kurallara uygun bir řekilde elde edildiđini beyan ederim. Aynı zamanda bu kural ve davranıřların gerektirdiđi gibi, bu alıřmanın znd olmayan tm materyal ve sonuları tam olarak aktardıđım ve referans gsterdiđimi belirtirim.

Serdar Kasım KARATAŐ

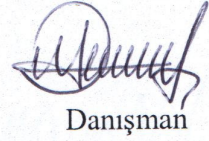
İmza:



“Diferansiyel(Farksal) Gelişim Algoritması Kullanılarak Kısıt Yönetim Metotlarının Performanslarının Karşılaştırılması” adlı yüksek lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.



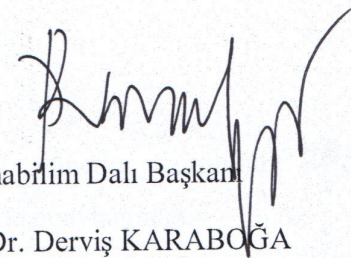
Tezi Hazırlayan
Serdar Kasım KARATAŞ



Danışman

Yrd. Doç. Dr. Mustafa DANACI

Doç. Dr. M. Fatih TAŞGETİREN 7



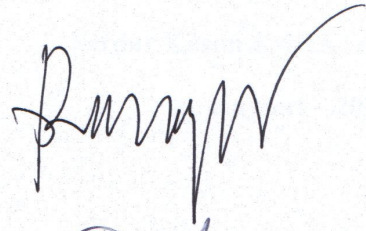
Anabilim Dalı Başkanı
Prof. Dr. Derviş KARABOĞA

Yrd. Doç. Dr. Mustafa DANACI danışmanlığında **Serdar Kasım KARATAŞ** tarafından hazırlanan “**Diferansiyel (Farksal) Gelişim Algoritması Kullanılarak Kısıt Yönetim Metotlarının Performanslarının Karşılaştırılması**” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği** Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

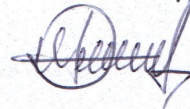
08/08/2011

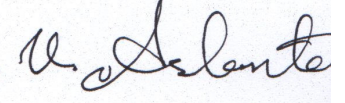
JÜRİ:

Başkan: Prof. Dr. Derviş KARABOĞA



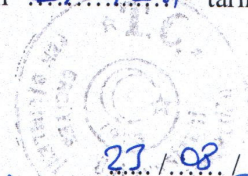
Üye : Yrd. Doç. Dr. Mustafa DANACI



Üye : Doç. Dr. Veynel ASLANTAŞ 

ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulunun 23/08/2011 tarih ve 2011/29-07 sayılı kararı ile onaylanmıştır.



23/08/2011


Prof. Dr. Necmettin MARAŞLI

Enstitü Müdürü

TEŞEKKÜR

Tezin hazırlanmasındaki bütün aşamalarda bilgi ve tecrübesi ile desteğini hiç esirgemeyen tez danışmanı hocalarım Yrd. Doç. Dr. Mustafa DANACI' ya ve Doç. Dr. M. Fatih TAŞGETİREN' e, hayatım boyunca hep yanımda olmuş, bugünlere gelmemde en büyük emeği bulunan aileme, tez hazırlama aşamasında doğrudan ya da dolaylı katkıda bulunan herkese teşekkürü bir borç bilirim.

Serdar Kasım KARATAŞ

Kayseri – 2011

**DİFERANSİYEL(FARKSAL) GELİŞİM ALGORİTMASI KULLANILARAK
KISIT YÖNETİMİ METOTLARININ SONUÇLARININ VE
PERFORMANSLARININ KARŞILAŞTIRILMASI**

Serdar Kasım KARATAŞ

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Temmuz 2011

Tez Danışmanı: Yrd. Doç. Dr. Mustafa DANACI

Doç. Dr. M. Fatih TAŞGETİREN

ÖZET

Gerçek hayattaki optimizasyon problemlerinin çoğu araştırma uzayının biçimini değiştiren fiziksel, zamansal, geometrik vb. farklı açılardan kısıtlamalara sahiptir. Doğrusal olmayan problemlerin çözümüne yönelik olarak geliştirilmiş birçok teknik söz konusudur. Özellikle değişken sayısına ve veri tiplerine bağlı olarak problemlerin zorluk dereceleri de artabilmektedir. Bu tip problemlerin deterministik yöntemlerle çözümü problemin özelliğine bağlı olarak hem modelleme de hem de çözüm sürecinde zorluklar içerebilmektedir. Bunların üstesinden gelebilmek için son 20 yıl boyunca çok fazla çeşitte sezgisel algoritmalar geliştirilmiş ve kısıtlı optimizasyon problemlerinin çözümünde kullanılmıştır. Sezgisel yöntemlerin kesin sonucu vermediği ancak makul sürede kesin çözüme yakın bir çözüm elde etmeyi sağladığı görülebilmektedir.

Gelişimsel algoritmalar ve diğer birçok sezgisel algoritmalar, optimizasyon için kullanıldığında, doğal olarak kısıtlamasız araştırma teknikleri gibi kullanılmıştır. Bu yüzden uygunluk fonksiyonlarının içerisine kısıtlamaları da dâhil eden penaltı fonksiyonları gibi ilave bir mekanizmaya ihtiyaç duyulmaktadır. Bu ihtiyaca cevap veren popülasyon temelli sezgisel bir optimizasyon tekniğinin Diferansiyel(farksal) Gelişim Algoritması (DGA) olduğu görülebilmektedir. DGA kullanırken araştırmacılar sınırlamaların üstesinden gelmek için self-adaptive penalty, epsilon constraint handling, stochastic ranking gibi birtakım farklı yaklaşımlar önermektedir.

Bu tez çalışmasında DGA ile birlikte kısıt yönetimi (constraint handling) metotlarından olan Near Feasibility Threshold ve Epsilon Constraint metotları, 18 kısıtlı test problemleri (Benchmark Problems) üzerinde uygulanmıştır. Bu uygulamalar neticesinde ortaya çıkan performanslara göre parametrelerin başarılı bir şekilde optimize edildiği ve Epsilon Constraint metodunun Near Feasibility Threshold metoduna olan bariz üstünlüğü gözlemlenmiştir.

Anahtar Kelimeler: Diferansiyel Gelişim Algoritması, Uyarlamalı Ceza Fonksiyonları, Constraint Handling, Yakın Uygunluk Eşiği, Epsilon Kısıtlama Metodu.

COMPARISON OF CONSTRAINT HANDLING METHODS BY USING DIFFERENTIAL EVOLUTION ALGORITHMS

Serdar Kasım KARATAŞ

Erciyes University, Graduate School of Natural and Applied Sciences

M.Sc. Thesis, July 2011

Thesis Supervisor: Assist. Prof. Dr. Mustafa DANACI

Assoc. Prof. M. Fatih TAŞGETİREN

ABSTRACT

Most real world optimization problems have constraints of different types (e.g., physical, time, geometric, etc.) which modify the shape of the search space. There are several techniques developed for solving nonlinear optimization problems. These problems become more difficult related to the number of variables and types of parameters. Solution of these problems with deterministic methods may include difficulties in both modeling and solving depending on the type of the problem. During the last couple of decades, a wide variety of heuristics have been designed in order to overcome these difficulties and applied to solve constrained optimization problems. As can be noticed that Heuristics can provide an approximate solution near exact solution in admissible time.

Evolutionary algorithms and most other heuristics, when used for optimization, naturally operate as unconstrained search techniques. Therefore, they require an additional mechanism as penalty functions to incorporate constraints into their fitness function. Differential Evolution Algorithm, an efficient population based heuristic optimization technique, is one of the solutions to this requirement. With Differential Evolution Algorithm, researchers have also proposed a number of other approaches to handle constraints such as the self-adaptive penalty, epsilon constraint handling and stochastic ranking.

In this thesis, two constraint handling methods, namely, Near Feasibility Threshold and Epsilon Constraint Methods are employed in the differential evolution algorithm to solve 18 benchmark problems used in the literature. Experimental results showed that Epsilon Constraint method was superior to Near Feasibility Threshold Method.

Keywords: Differential Evolution Algorithm, Adaptive Penalty Functions, Constraint Handling, Near Feasibility Threshold, Epsilon Constraint Methods.

İÇİNDEKİLER

BİLİMSEL ETİĞE UYGUNLUK	ii
YÖNERGEYE UYGUNLUK	iii
KABUL VE ONAY	iv
TEŞEKKÜR.....	v
ÖZET.....	vi
ABSTRACT	viii
İÇİNDEKİLER	x
TABLolar LİSTESİ.....	xiii
ŞEKİLLER LİSTESİ	xv
KISALTMALAR ve SİMGELER DİZİNİ.....	xvii
GİRİŞ	1
1. BÖLÜM	
EVRİMSEL ALGORİTMALAR	
1.1. Evrimsel Algoritmalar Giriş	4
1.2. Diferansiyel Gelişim Algoritması	5
1.2.1. Problemler ve Parametreler.....	8
1.2.2. Kodlama ve Başlangıç Popülasyonu.....	10
1.2.3. Mutasyon.....	10
1.2.4. Çaprazlama.....	11
1.2.5. Uygunluk Fonksiyonu.....	11
1.2.6. Seçim.....	12
1.2.7. Algoritmanın Durdurulması.....	12

2. BÖLÜM

KISIT YÖNETİMİ (CONSTRAINT HANDLING) METOTLARI

2.1. Kısıt Yönetimi Metotlarına Giriş	14
2.2. Kısıtlara Giriş	15
2.3. Ceza Fonksiyonlarının Tanıtımı	17
2.4. Statik Ceza Fonksiyonları	19
2.5. Dinamik Ceza Fonksiyonları	21
2.6. Uyarlanabilir Ceza Fonksiyonları	21
2.6.1. Yakın Uygunluk Eşik Değeri Metodu(Near-Feasibility Threshold).....	22
2.6.2. Epsilon Kısıtlama Metodu(Epsilon Constraint Method).....	24

3. BÖLÜM

BULGULAR

3.1. Deney Tasarım İlkeleri.....	29
3.2. Deney Tasarımın Tarihçesi	30
3.3. Deney Tasarımın Kullanım Alanları	30
3.4. Deney Tasarımının Adımları.....	31
3.4.1. Problemin Tanımlanması	31
3.4.2. Faktörlerin, Seviyelerin ve Sıranın Seçimi	31
3.4.3. Tepki Değişkeninin Seçimi	31
3.4.4. Deney Tasarımının Seçimi.....	32
3.4.5. Deneyi Uygulama	32
3.4.6. Verilerin İstatistiksel Analizi	32
3.4.7. Sonuçlar	32
3.5. Faktöriyel Denemeler.....	32
3.5.1. Tam Faktöriyel Deneyler	32

3.6. Kısıt Yönetimi Parametrelerinin Deneş Tasarımı İle Belirlenmesi.....	33
4. BÖLÜM	
TARTIŞMA, SONUÇLAR ve ÖNERİLER	61
EKLER.....	71
KAYNAKLAR	74
ÖZGEÇMİŞ	78

TABLOLAR LİSTESİ

Tablo 3.1. Yakın uygunluk eşik değeri metodu için olası faktör seviyeleri ve değerleri .	34
Tablo 3.2. Epsilon constraint metodu için olası faktör seviyeleri ve değerleri.....	34
Tablo 3.3. Yakın uygunluk eşik değeri metodunun 20.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri	35
Tablo 3.4. Yakın uygunluk eşik değeri metodunun 100.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri	37
Tablo 3.5. Yakın uygunluk eşik değeri metodunun 200.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri	39
Tablo 3.6. Yakın uygunluk eşik değeri metodunun 20.000 fonksiyon hesaplaması için en uygun parametre seti	44
Tablo 3.7. Yakın uygunluk eşik değeri metodunun 100.000 fonksiyon hesaplaması için en uygun parametre seti	45
Tablo 3.8. Yakın uygunluk eşik değeri metodunun 200.000 fonksiyon hesaplaması için en uygun parametre seti	46
Tablo 3.9. Epsilon constraint metodunun 20.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri	46
Tablo 3.10. Epsilon constraint metodunun 100.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri	48
Tablo 3.11. Epsilon constraint metodunun 200.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri	50
Tablo 3.12. Epsilon constraint metodunun 20.000 fonksiyon hesaplaması için en uygun parametre seti	55
Tablo 3.13. Epsilon constraint metodunun 100.000 fonksiyon hesaplaması için en uygun parametre seti	56
Tablo 3.14. Epsilon constraint metodunun 200.000 fonksiyon hesaplaması için en uygun parametre seti	57
Tablo 3.15. 20.000 fonksiyon hesaplaması için nft-ec performansı	62

Tablo 3.16. 100.000 fonksiyon hesaplaması için nft-ec performansı	64
Tablo 3.17. 200.000 fonksiyon hesaplaması için nft-ec performansı	66

ŞEKİLLER LİSTESİ

Şekil 1.1. Yeni vektörün oluşturulması işlemi.....	7
Şekil 1.2. DGA akış şeması.....	13
Şekil 2.1. NFT metodu örnekleme [32].....	24
Şekil 2.2. EC metodu ile farklı çözümler üretilmesi [31]	26
Şekil 3.1. Yakın uygunluk eşik değeri metodunun 20.000 fonksiyon hesaplaması için ana etkiler grafiği.....	41
Şekil 3.2. Yakın uygunluk eşik değeri metodunun 100.000 fonksiyon hesaplaması için ana etkiler grafiği.....	42
Şekil 3.3. Yakın uygunluk eşik değeri metodunun 200.000 fonksiyon hesaplaması için ana etkiler grafiği.....	42
Şekil 3.4. Yakın uygunluk eşik değeri metodunun 20.000 fonksiyon hesaplaması için etkileşim grafiği.....	43
Şekil 3.5. Yakın uygunluk eşik değeri metodunun 100.000 fonksiyon hesaplaması için etkileşim grafiği.....	44
Şekil 3.6. Yakın uygunluk eşik değeri metodunun 200.000 fonksiyon hesaplaması için etkileşim grafiği.....	45
Şekil 3.7. Epsilon constraint metodunun 20.000 fonksiyon hesaplaması için ana etkiler grafiği.....	52
Şekil 3.8. Epsilon constraint metodunun 100.000 fonksiyon hesaplaması için ana etkiler grafiği	53
Şekil 3.9. Epsilon constraint metodunun 200.000 fonksiyon hesaplaması için ana etkiler grafiği	54
Şekil 3.10. Epsilon constraint metodunun 20.000 fonksiyon hesaplaması için etkileşim grafiği.....	55
Şekil 3.11. Epsilon constraint metodunun 100.000 fonksiyon hesaplaması için etkileşim grafiği.....	56
Şekil 4.12. Epsilon constraint metodunun 200.000 fonksiyon hesaplaması için etkileşim grafiği.....	57

Şekil 4.13. 20.000 fonksiyon hesaplamasına göre nft–ec ortalama uygunluk	58
Şekil 4.14. 100.000 fonksiyon hesaplamasına göre nft–ec ortalama uygunluk dağılım durumu.....	59
Şekil 4.15. 200.000 fonksiyon hesaplamasına göre nft–ec ortalama uygunluk dağılım durumu.....	60

KISALTMALAR ve SİMGELER DİZİNİ

DGA	:	Diferansiyel Gelişim Algoritması (DEA: Differential Evolution Algorithm)
GA	:	Genetik Algoritma (GA: Genetic Algorithm)
NFL Theorem	:	Kolay Olmayan Teorem(No free lunch theorem)
EC	:	Epsilon Kısıtlama (ϵ -Constraint)
NFT	:	Yakın Uygunluk Eşik Değeri(Near Feasibility Threshold)
CR	:	Çaprazlama Oranı
MR	:	Mutasyon Oranı
F	:	Ölçekleme Faktörü
NP	:	Popülasyon Büyüklüğü
D	:	Parametre Sayısı (değişken sayısı(gen sayısı) (1, 2, 3, .., j))
G	:	Nesil (1, 2, 3, .., Gmax)
jrand	:	yeni kromozomun üretilmesinde kullanılacak rasgele seçilmiş kromozomlar
$x_j^{(u)}$:	Değişkenin Üst Sınır Değeri
$x_j^{(l)}$:	Değişkenin Alt Sınır Değeri
$x_{j,i,G}$:	G jenerasyonunda, i . vektörün j parametresi
$n_{j,i,G+1}$:	Mutasyon ve çaprazlamaya tabi tutulmuş ara vektör
$u_{j,i,G+1}$:	$x_{j,i,G}$ den bir sonraki jenerasyon için üretilen vektör
$r_{1,2,3}$:	Yeni vektörün hesaplanmasında kullanılacak rastgele seçilmiş vektörler $r_{1,2,3} \in \{1,2,3,\dots, NP\}$ $r_1 \neq r_2 \neq r_3 \neq i$

GİRİŞ

Bilimdeki ve mühendislikteki çoğu optimizasyon problemleri sınırlamalar içermektedir. Sınırlamaların varlığı uygun bölgeyi azaltır ve araştırma işlemini karmaşıklaştırır. Evrimsel Algoritmalar daima sınırlamasız araştırmalarda icra edilir. Evrimsel Algoritmalar, sınırlamalı optimizasyon problemlerini çözerken, sınırlamaların üstesinden gelmek için ek mekanizmalara ihtiyaç duyar [1]. Son 30 yıl süresince, evrimsel algoritmalarda kullanılan sınırlamaların üstesinden gelen birtakım teknikler geliştirilmiştir. Kolay olmayan teoreme (no free lunch theorem-NFL Theorem) göre, her bir problem üzerinde bütün diğer tekniklerden üstün olan tek bir kısıt yönetimi (constraint handling) tekniği mümkün değildir. Başka bir deyişle uygun araştırma uzayı ile bütün araştırma uzayı arasındaki orana, problemin multimodal durumuna, seçilen evrimsel algoritmaya ve araştırma işleminin küresel arama/lokal çalıştırma aşamaları gibi birtakım faktörlere bağlıdır. Bu yüzden araştırma işleminin farklı aşamaları sırasında farklı kısıt yönetimi metotları daha verimli olabilir [2]. Ancak uygun bir kısıt yönetimi tekniğini seçmek ve ilişkili parametrelerin en uygun değerlerini belirlemek için pek çok deneme yanılma çalışmalarına ihtiyaç duyar. Bu yaklaşım eğer amaç fonksiyon hesaplama açısından çok uzun zaman alıyor ise ya da gerçek zamanlı çözüm ihtiyacı varsa, gerçekdışı sayısal gereksinimlerden dolayı sıkıntı oluşturacaktır [3].

Sınırlamalı optimizasyon problemlerini çözerken, tüm sınırlamalara cevap veren çözüm adayları uygun bireyleri oluştururken sınırlamaların herhangi biri konusunda yetersiz kalanlar uygun olmayan bireylerdir. Sınırlamalı optimizasyonda temel sorunlardan biri; araştırma işleminin başından sonuna kadar uygun olmayan bireylerle nasıl uğraşılacağıdır. Üstesinden gelmenin bir yolu uygun olmayan bireyleri tamamıyla dikkate almamak ve araştırma işlemine sadece uygun bireylerle devam etmektir. Bu yaklaşım, Evrimsel Algoritmalar gibi olasılıksal araştırma metotları için başarısız olabilir ve uygun olmayan bireylerde sunulan potansiyel bilgi ziyan edilmiş olabilir. Eğer araştırma uzayı kesikli ise, Evrimsel Algoritma local minimaya takılabilir. Bu

yüzden uygun olmayan bireylerdeki arařtırmayı daha iyi sonuçlara götürebilecek bilginin elde edilmesi için farklı teknikler geliştirilmiřtir. Michalewicz ve Schoenauer [4] Evrimsel Algoritmalar bünyesinde sınırlamaların üstesinden gelmek için 4 kategori řeklinde metotları gruplamıřlardır: Çözümlerin geçerliliđini korumak [5], penalty(ceza) fonksiyonları, uygun ve uygun olmayan çözümler arasında ayırım yapmak ve hibrid metotlar. Bir sınırlamalı optimizasyon problemi aynı zamanda multiobjective (çok amaçlı) problem olarak da formüle edilebilir fakat baskın olmayan sıralamadan dolayı sayısal olarak yoğunluk içerir [6].

Bu çalıřmada, verilmiř bir problemin uygun olan çözümlerinin sayılarının arttırılması ve optimal çözüme en yakın çözümlerin elde edilmesini sađlamak amacıyla DGA ile NTF ve EC kısıt yönetimi teknikleri bir arada kullanılmıř ve farklı parametre setlerinin problem üzerindeki performansları karřılařtırılarak mümkün olan en uygun parametrelerin elde edilebileceđi bir yapı önerilmiřtir.

DGA' nın sahip olduđu güçlü ve avantajlı yönleriyle kısıt yönetimi Metotları birleřtirilerek, sınırlamalara sahip test problemleri (Benchmark) üzerinde çalıřtırılması, performanslarının karřılařtırılması ve bu sınırlama sıkıntısının üstesinden gelinmesinin sađlanması. Bu sayede de gerçek hayatta karřılařılan problemlerin çözümünde ve optimize edilmesinde faydalı olması amaçlanmıřtır.

Optimizasyon, hemen her alanda kullanılan bir kavram olup, kazancı maksimum ve maliyeti minimum yapmayı amaçlamaktadır [7]. Doğrusal olmayan optimizasyon problemlerinin çözümünde sezgisellerin kullanımı oldukça yaygındır. Bunun başlıca nedenleri arasında modellenmesindeki kolaylık ve hızlı bir řekilde sonuç vermesidir [8]. Çalıřmada, sahip olduđu güçlü ve avantajlı yönleriyle sezgisel optimizasyon tekniklerinden biri olan DGA' nın içerisine kısıt yönetimi metotları yerleřtirilerek uygunluk oranı mümkün olduđu kadar arttırılmaya ve böylece performanslar karřılařtırılarak optimal sonuçlar elde edilmeye çalıřılmıřtır. Kullanılan kısıt yönetimi metotları:

- Near Feasibility Threshold (NFT)
- Epsilon Constraint Methods (ϵ -Constraint)

Bu tez çalışmasında simülasyon sonuçları, Diferansiyel Gelişim Algoritmasının kodlanması, kısıt yönetimi metotlarının Diferansiyel Gelişim Algoritmasının içerisine yerleştirilmesi ve parametrelerin optimize edilmesi için Microsoft Visual C++ 6.0 programı kullanılmıştır. Ayrıca deney tasarımı aşamasında farklı parametre setlerinin oluşturulması için MINITAB 14 programı kullanılmıştır.

Tez çalışmasının 1. Bölümünde Evrimsel Algoritmalara kısaca değinilip Diferansiyel Gelişim Algoritması detaylı bir şekilde sunulmuştur. Bölüm 2’de Diferansiyel Gelişim Algoritması ile birlikte kullanılan kısıt yönetimi teknikleri verilmiştir. Bölüm 3’te çözüm yöntemlerinin parametrelerini belirlemek için kullanılan deney tasarımı yöntemlerinden bahsedilmiştir. 4. bölümde ise Deneysel Sonuçlar tablolar üzerinde sunularak sonuçların değerlendirilmesi ve karşılaştırmalı tartışmalar verilmiştir. Test problemleri olarak Ek A da sunulan 18 adet test problemi (Benchmark Problems) kullanılmıştır.

1. BÖLÜM

EVİRİMSEL ALGORİTMALAR

1.1. Evrimsel Algoritmalara Giriş

Bilinen eniyileme tekniklerinin fayda sağlayamadığı durumlarda evrim algoritmaları geliştirilmiştir. Evrim algoritmaları aşağıdaki 5 maddeye gereksinim duymaktadır:

- a. Çözümün genetik temsili,
- b. Başlangıç popülasyonunun oluşturulması
- c. Uygunluk fonksiyonunu belirleme
- d. Hesaplama işlemlerini (operatörleri) belirleme
- e. Kontrol parametrelerini belirleme (popülasyon büyüklüğü, operatör uygulama oranları, ölçekleme faktörü gibi)

Evrimsel algoritmalar; genetik algoritmalar, evrimsel programlama, evrimsel stratejiler ve diferansiyel gelişim algoritması gibi algoritmaları ihtiva eder.

Evrimsel hesaplama teknikleri birbirlerine benzemekle birlikte problemlere uygulanması açısından birbirlerinden farklılıklar da gösterebilmektedir. Örneğin çözümlerin temsili farklı şekilde yapılabilmekte ve buna bağlı olarak farklı operatörler kullanılmaktadır.

Örneğin evrim algoritmalarının temel adımları şu şekildedir:

Başlangıç durumunu ayarla

Başlangıç durumunu değerlendir

Durdurma kriteri sağlanıncaya kadar aşağıdaki adımları tekrarla

Mevcut popülasyona seçme işlemi uygulayarak bir sonraki neslin popülasyonunu oluştur.

Yeni popülasyonu değiştir

Yeni popülasyonu değerlendir [9].

1.2. Diferansiyel Gelişim Algoritması

Doğrusal olmayan optimizasyon problemlerinin çözümünde deterministik ve sezgisel yöntemler kullanılmaktadır. Deterministik yöntemlerle modelleme yapmak çok fazla matematiksel altyapı gerektirdiğinden ve problem boyutları arttığında yetersiz kaldığından modellenmesindeki kolaylık ve hızlı bir şekilde sonuç verme gibi üstünlüklerinden dolayı sezgisel yöntemlerin önemi her geçen gün artmaktadır [8].

Sürekli uzayda küresel eniyileme içeren optimizasyon problemleri bilimsel toplumda oldukça yaygındır. Genellikle amaç sistem parametrelerini belirleyerek belirli özellikleri eniyilemektir. Sistem parametreleri vektör olarak ifade edilmektedir. Bir eniyileme problemine standart yaklaşım amaç fonksiyonunu tasarlamaktır ve bu genellikle en küçükleme şeklinde olabilir. En küçükleme istenen maliyet, zaman veya mesafe olabilir. Amaç fonksiyonu doğrusal olmayan ya da türevlenemez (non-differentiable) olduğu durumda doğrudan arama yöntemleri tercih edilir ki bunlar genetik algoritma ve evrimsel stratejiler şeklinde olabilir. Genellikle en küçükleme tekniklerinden, bazı şartları yerine getirmesi beklenir. Bu şartlar;

- a. Doğrusal olmayan, türevlenemez (non-differentiable) ve çoklu maliyet fonksiyonlarını çözebilme,
- b. Hesaplama yoğun maliyet fonksiyonlarını çözmek için paralelleştirebilme, (örneğin bir maliyet fonksiyonunun sadece bir kez bile değerlendirilmesi çok fazla zaman alabilir. Değerlendirilmeler aynı özelliğe sahip farklı bilgisayarlarda yapılabilecek şekilde paralelleştirme yapılabilmesi)
- c. Kullanım kolaylığı,
- d. İyi yakınsama özelliği.

şeklinde sıralanabilir. Bu şartları yerine getirebilmek için geliştirilen başarılı optimizasyon algoritmalarından birisi diferansiyel gelişim algoritmasıdır [10].

Diferansiyel Gelişim Algoritmasının adımları:

Parametreleri başlatmak

Hedef popülasyonu başlatmak

Hedef popülasyonu değerlendirmek

Yerel arama uygulamak (seçimlik)

Do

{ Mutant popülasyon elde etmek

Deneme popülasyon elde etmek

Deneme popülasyon bireylerinin turunu oluşturmak

Deneme popülasyonu değerlendirmek

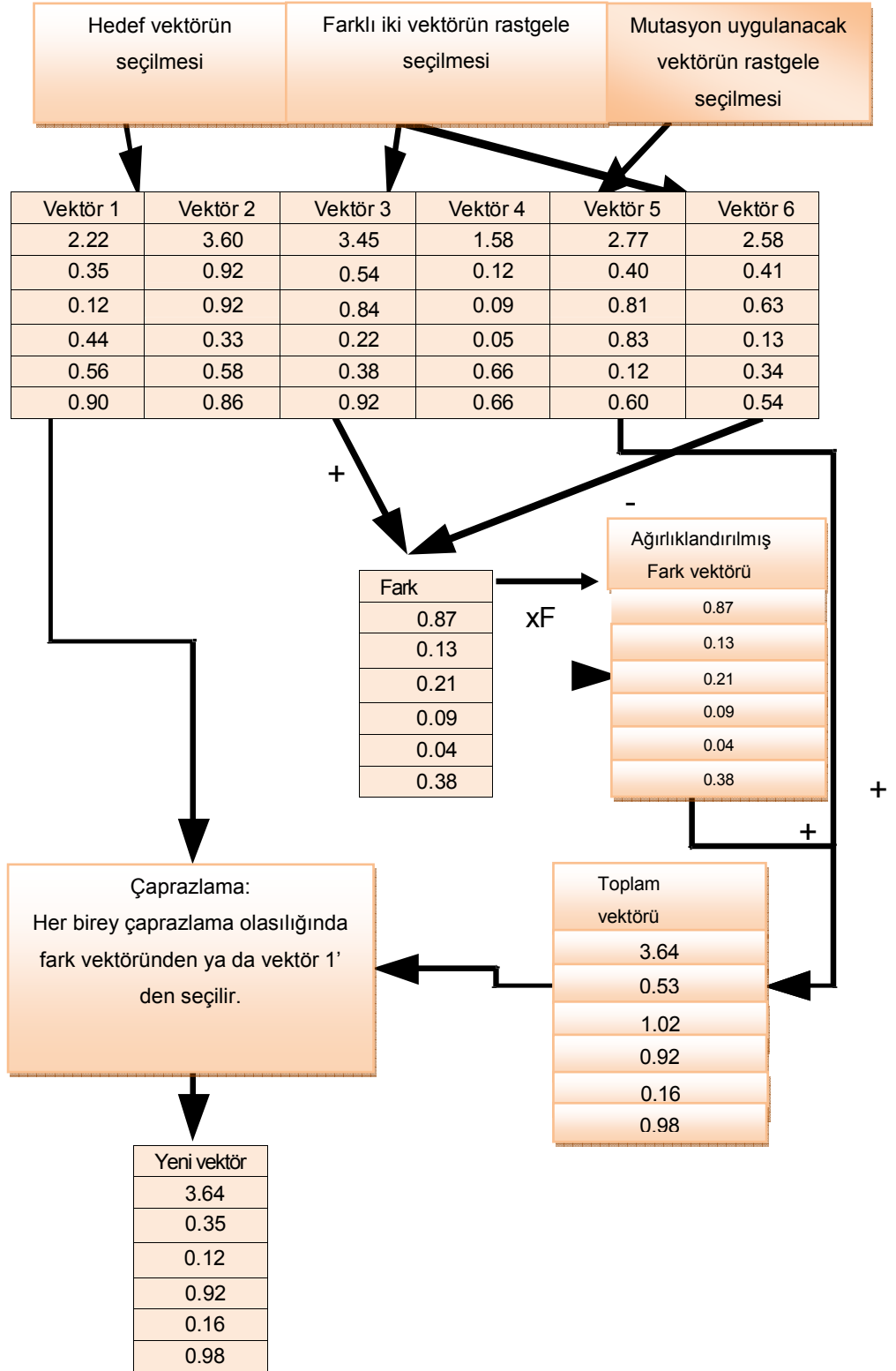
Hedef popülasyon ile deneme popülasyon arasından seçim yapmak

Hedef popülasyona yerel arama uygulamak (seçimlik)

}

While (Durdurma kriteri sağlanuncaya kadar işlemleri gerçekleştirmek) [11].

Diferansiyel Gelişim Algoritması (DGA), Price ve Storn tarafından geliştirilmiş, sürekli (continuous time) verilerin söz konusu olduğu problemlerde etkin sonuç verebilen, temel olarak Genetik Algoritmaya (GA) bağlı sezgisel bir optimizasyon tekniğidir [12]. Popülasyon tabanlı olması sebebiyle aynı anda birçok noktada araştırma yapabilmektedir. Çalışması süresince operatörler yardımıyla problemin çözümü için daha iyi noktalar araştırılmaktadır. İkili (binary) GA'dan farklı olarak değişkenler kendi değerleriyle temsil edilmektedir. Price ve Storn gerçek değerlerin kullanıldığı problemlerin çözüm performansını artırmaya çalışmışlardır. GA'daki çaprazlama, mutasyon ve seçim operatörleri DGA' da da kullanılmaktadır. GA'dan farklı olarak her operatör tüm popülasyona sırayla uygulanmamakta, vektörler tek tek ele alınarak rastgele seçilen diğer üç vektörle yeni bir birey elde etmede kullanılmaktadır. Yeni vektörün oluşturulma adımları Şekil 1.1.'de verilmektedir [13].



Şekil 1.1. Yeni vektörün oluşturulması işlemi

Bu işlemler sırasında mutasyon ve çaprazlama operatörleri kullanılmaktadır. Mevcut vektörle yeni vektörün uygunluklarının (fitness) karşılaştırılmasıyla uygunluğu daha iyi olan vektör yeni birey olarak bir sonraki popülasyona aktarılmaktadır. Böylelikle seçim operatörü de kullanılmış olmaktadır. Üretilen çözümün uygunluk değeri, uygunluk fonksiyonu tarafından üretilen çözüme verilen uygunluk değeridir.

DGA' nın diğer sezgisellere önemli bir üstünlüğü kolayca kodlanabilmesidir. Diğer algoritmalar için binlerce satırdan oluşan kodlar söz konusu iken DGA için yaklaşık 20 satırlık kod yeterli olmaktadır [13, 14]. DGA' nın basit yapısı, kullanım kolaylığı, hızı ve dinçliği en önemli avantajları arasında yer almaktadır.

DGA' nın temel adımları Evrimsel Algoritmalarla farkını göstermek adına aşağıdaki gibi verilebilir:

Başlangıç popülasyonunun oluşturulması

Değerlendirme

Repeat

Mutasyon

Rekombinasyon

Değerlendirme

Seleksiyon

Until (*durdurma kriteri sağlanıncaya kadar*) [15]

DGA, diğer Evrimsel Algoritmalarından mutasyon ve rekombinasyon aşamalarında farklılık göstermektedir. DGA popülasyonu karıştırmak için, çözüm vektörleri arasındaki ağırlıklandırılmış farkları kullanmaktadır [15].

1.2.1. Problemler ve Parametreler

NP : Popülasyon büyüklüğü $NP \geq 4$ (1, 2, 3, ..., i)

D : Değişken sayısı (1, 2, 3, ..., i)

CR : Çaprazlama oranı [0.1,1.0]

G : Jenerasyon (1, 2, 3, ..., G_{max})

F : Ölçekleme faktörü

$x_{j,i,G}$: G jenerasyonunda, i . vektörün j parametresi

$n_{j,i,G+1}$: Mutasyon ve çaprazlamaya tabi tutulmuş ara vektör

$u_{j,i,G+1}$: $x_{j,i,G}$ den bir sonraki jenerasyon için üretilen vektör

$r_{1,2,3}$: Yeni vektörün hesaplanmasında kullanılacak rastgele seçilmiş vektörler $r_{1,2,3} \in$

$\{1,2,3,\dots,NP\}$ $r_1 \neq r_2 \neq r_3 \neq i$

$x_j^{(l)}$, $x_j^{(u)}$: Değişkenlere ait alt ve üst sınır değerleri.

Optimizasyon problemleri genel olarak aşağıdaki gibi ifade edilebilir [16].

$f(X)$ 'i

$g_k(X) \leq 0$ ve

$x_j^{(l)} \leq x_j \leq x_j^{(u)}$ $j=1,\dots,n$

kısıtları ile minimize et.

Burada $f(X)$ uygunluk fonksiyonunu, $g_k(X)$ kısıtlar setini ve $X = \{x_1, x_2, \dots, x_n\}$ gerçek değerli değişkenler setini ifade etmektedir. $x_j^{(l)}$ ve $x_j^{(u)}$ sırasıyla değişkenlere ait alt ve üst sınır değerleridir.

Amaç tüm kısıtları sağlayan en iyi çözümü araştırmaktır. Burada $f(X)$ fonksiyonu DGA' da uygunluk fonksiyonu olarak belirlenmekte ve her bir vektör için bir çözümü temsil etmektedir. DGA da GA gibi kısıtlarla çalışmamaktadır. Kısıtların amaç fonksiyonu içerisinde ele alınması gerekmektedir. Bunun için ceza yöntemi kullanılmaktadır. Kısıtlardan uzaklaşmalar meydana geldiğinde uygunluk değerine ceza değeri eklenerek bu noktanın iyi bir çözüm olmadığı DGA' nda dikkate

alınmalıdır. Ayrıca problem parametrelerinin belirlenen alanın dışına çıkmasına izin verilmemelidir. Zira buna izin verilirse DGA sonsuza gidebilecektir. Bu sorun için ya parametreler sınır değerlere çekilmeli veya sınır dışında değer almış parametrelerin yeni değerleri rastgele olarak sınır değerleri içinde yeniden belirlenmelidir. Problemin amaç fonksiyonu, değişkenler ve kısıtlar belirlendikten sonra aşağıdaki adımlar izlenerek DGA uygulanır.

1.2.2. Kodlama ve Başlangıç Popülasyonu

Probleme ait değişken sayısı her bir vektöre ait parametre (boyut (D)) sayısını belirlemektedir. NP ise kullanıcı tarafından belirlenen vektör sayısıdır. Vektör sayısı üçten büyük olmalıdır. Çünkü DGA' da yeni vektörlerin üretilmesi için mevcut vektör dışında üç adet vektör gerekmektedir. Başlangıçta NP adet D boyutlu vektörden meydana gelen başlangıç popülasyonu (P_0) aşağıdaki gibi üretilir [17].

$$\forall_i \leq NP \forall_j \leq D : x_{j,i,G=0} = x_j^{(l)} + \text{rand}_j [0,1]. (x_j^{(u)} - x_j^{(l)}) \quad (1)$$

1.2.3. Mutasyon

Mutasyon, mevcut vektörün bir kısım parametreleri üzerinde, rastgele belirlenmiş miktarlarda değişiklikler yapmaktır. Bu değişiklikler sayesinde vektörünün temsil ettiği çözüm noktası, artık çözüm uzayında farklı bir noktayı temsil etmektedir. Mutasyonun hedefine ulaşabilmesi için, doğru yönde doğru miktarda hareketi sağlayacak değişikliklerin belirlenmesi gerekmektedir.

DGA' da, mutasyon işlemine tabi tutulacak olan vektör dışında birbirlerinden farklı olan üç vektör seçilir. Seçilen vektörlerden ilk ikisinin farkı alınır. Daha sonra bu fark vektörü F parametresiyle çarpılır. F parametresi genellikle 0-2 arasında değerler almaktadır. Elde edilen ağırlıklandırılmış fark vektörü ile seçilen üçüncü vektör ile toplanır. Böylece, mutasyon sonucu çaprazlamada kullanılacak olan vektör elde edilmiş olur ($n_{j,i,G+1}$). Bu işlem Denklem 2 de matematiksel olarak ifade edilmiştir. Literatürde kullanılan farklı birçok mutasyon operatörü bulunmaktadır [14].

$$\forall_j \leq D : n_{j,i,G+1} = x_{j,r3,G} + F. (x_{j,r1,G} - x_{j,r2,G}) \quad (2)$$

1.2.4. Çaprazlama

Çaprazlama yapılırken, mutasyon sonucu elde edilen fark vektörü ve $x_{i,G}$ vektörü kullanılarak yeni jenerasyona aday, deneme vektörü ($u_{i,G+1}$) üretilir. Deneme vektörüne ait her bir parametre, CR olasılıkla fark vektöründen $1-CR$ olasılıkla mevcut vektörden seçilir. Düzenli çaprazlamada her bir parametre ayrı olarak değerlendirilip eşit olasılıkla iki ebeveyn vektöründen birinden seçilmektedir.

DGA' da ise eşit olasılık yerine CR olasılığı söz konusudur. 0 ile 1 arasında üretilen rasgele sayı CR ' den küçükse gen, $n_{j,i,G+1}$ ' den, aksi takdirde mevcut vektörden seçilir. Denklem 3'te bu ifade yer almaktadır. Amaç belirlenen oranda parametrenin yeni fark vektöründen alınmasıdır. Buradaki $j=j_{rand}$ koşulu, en az bir tane parametrenin üretilen yeni vektörden alınmasını garanti etmek amacıyla konulmuştur. Rastgele seçilen j_{rand} noktasındaki parametre CR ' ye bakılmaksızın $n_{j,i,G+1}$ ' den seçilir [13].

$$\forall_j \leq D : x_{j,u,G+1} = \begin{cases} X_{j,n,G+1} & \text{eğer } rand [0,1] \leq CR \vee j = j_{rand} \\ x_{j,i,G} & \text{aksi durumda} \end{cases}$$

(3)

1.2.5. Uygunluk Fonksiyonu

Yukarıdaki adımlardan sonra elde edilen yeni vektörün veya mevcut vektörün yeni jenerasyona geçmesi söz konusudur. Bu noktada elde edilen yeni vektörün uygunluk değeri hesaplanır. Mevcut vektörün uygunluk değeri zaten bilinmektedir. Seçim

operatörü ile yeni jenerasyon eldeki vektörlerden üretilecektir.

1.2.6. Seçim

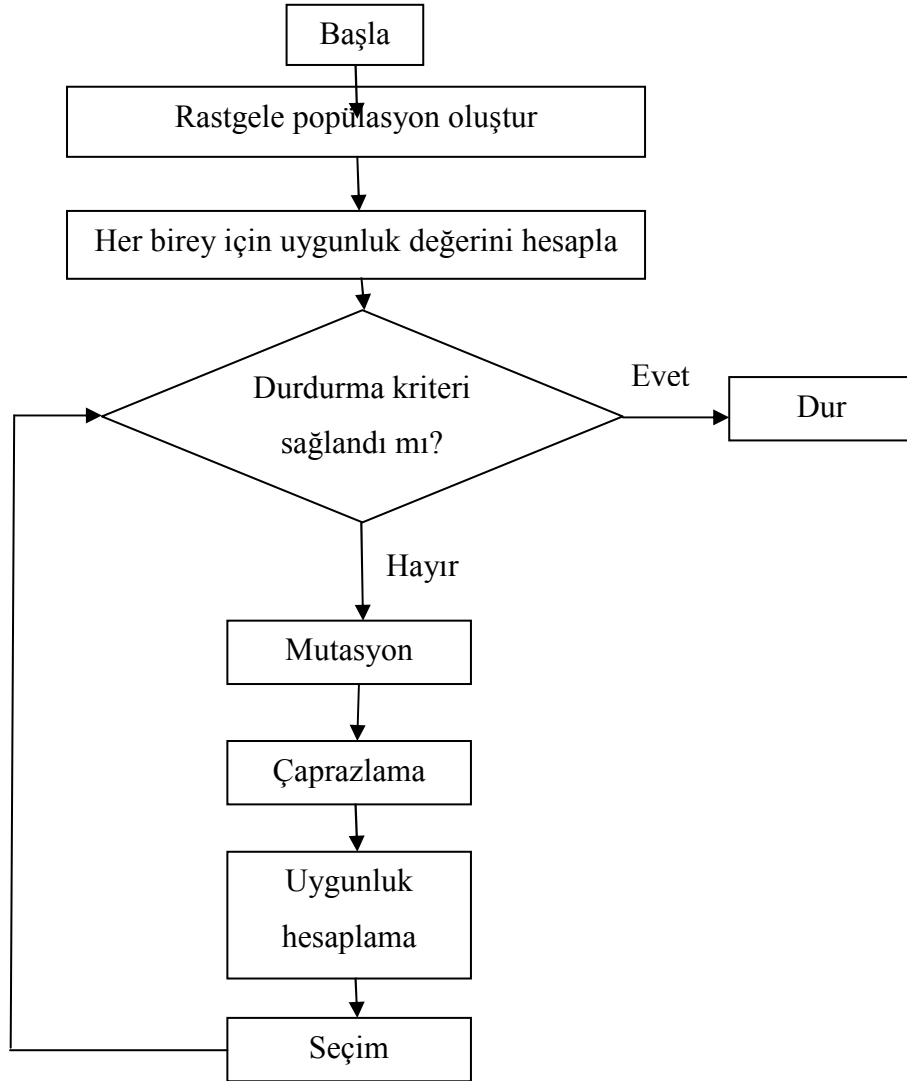
Seçim operatörü ile mevcut jenerasyon ve üretilen yeni vektörler değerlendirilerek yeni jenerasyon oluşturulur. Vektörlerin yeni jenerasyonda yer alma olasılıkları uygunluklarına bağlıdır. DGA' da karşılaştırma birebir yapıldığından seçim için karmaşık prosedürü olan seçim operatörlerine ihtiyaç duyulmamaktadır. Karşılaştırılan vektörlerden uygunluğu yüksek olan vektör yeni jenerasyonun bireyi olarak atanmaktadır [18].

$$\forall j \leq NP : x_{i,G+1} = \begin{cases} X_{u,G+1} & \text{eğer } f(x_{u,G+1}) \leq f(x_{i,G}) \\ x_{i,G} & \text{diğer durumlarda} \end{cases}$$

(4)

1.2.7. Algoritmanın Durdurulması

Algoritma yukarıda verilen adımlarda ilerleyerek optimizasyon işlemini gerçekleştirecektir. Bu noktada dikkat edilmesi gereken husus durdurma kistasının belirlenmesidir. Durdurma kistası olarak belirli bir iterasyon sayısı verilebilir. Bu durumda algoritma istenen iterasyona ulaştığında çalışmasını durduracaktır veya çalışma esnasında elde edilen en iyi uygunluk değeri ile en kötü uygunluk değerinin birbirine çok yaklaşması durumunda da çalışma sonlandırılabilir. Eğer algoritmanın ulaşması istenen uygunluk değeri önceden biliniyorsa bu değer verilerek de algoritmanın çalışması durdurulabilir. Şekil 1.2.'de DGA akış şeması verilmiştir.



Şekil 1.2. DGA akış şeması

2. BÖLÜM

KISIT YÖNETİMİ (CONSTRAINT HANDLING) METOTLARI

2.1. Kısıt Yönetimi Metotlarına Giriş

Çoğu uygulanabilir optimizasyon problemleri kısıtlamalara sahiptir. N parametrelili optimize edilecek olan bir kısıtlamalı optimizasyon problemi genellikle nonliner programlama problemi şeklinde aşağıdaki gibi yazılır [19]:

$$\text{Min : } f(X) \quad , X=(x_1,x_2,\dots,x_n) \text{ ve } X \in S$$

$$\text{s.t : } g_i(X) \leq 0, \quad i = 1,\dots,p$$

$$h_j(X) = 0, \quad j = p+1,\dots,m$$

S : Tüm araştırma uzayı.

p : Eşitsizlik kısıtlamalarının sayısı.

m-p : Eşitlik kısıtlamalarının sayısı.

Burada f'nin sürekli olmasına gerek yoktur ancak sınırlandırılmış olmalıdır.

Küresel optimum çözümde $g_i(X) = 0$ 'ı tatmin eden eşitsizlik kısıtlamaları aktif kısıtlamalar olarak isimlendirilirler. Tüm eşitlik kısıtlamaları aktif kısıtlamalardır.

Eşitlik kısıtlamaları eşitsizlik formuna dönüştürülebilir ve diğer eşitsizlik kısıtlamaları ile aşağıdaki şekilde birleştirilebilir :

$$G_i(X) = \begin{cases} \text{Max} \{ g_i(X), 0 \} & , \quad i = 1, \dots, p \\ \text{Max} \{ |h_i(X)| - \delta, 0 \} & , \quad j = p+1, \dots, m \end{cases}$$

δ : Eşitlik kısıtlamaları için tolerans parametresidir.

Bir kısıtlamalı problemde, küresel optimumda uygun olan bireylerin bulunmasının araştırma işlemi ebeveyn popülasyonu ve onun yavru popülasyonunun kombine edilmesi ile ortaya konan uygun olan çözümlerin sayısına bağlı olarak aşağıdaki şekilde 3 evreye ayrılabilir:

Evre 1 : Uygun çözüm yok.

Evre 2 : En azından bir tane uygun çözüm.

Evre 3: Yavru ve ebeveyn popülasyonunun karması gelecek nesil ebeveyn popülasyonunun boyutundan daha fazla uygun çözüme sahiptir.

Farklı kısıt yönetimi teknikleri bu 3 evrede farklı şekilde işlemektedir [20].

2.2. Kısıtlara Giriş

Pek çok optimizasyon problemi kısıtlamalara sahiptir. Bir evrimsel algoritmanın en son elde edilen çözüm ya da çözümler kümesi ister istemez uygun olmalıdır, yani, tüm kısıtlamalara cevap vermelidir. Kısıtlamalar, (a) sayı (b) ölçü (c) kritiklik ve (d) zorluk tan oluşan bir sınıf olarak nitelendirilebilir. İlk durum kısıtlamaların sayısıdır. Bazen birden çok amaçlı problemler kısıtlamalar şeklinde davranan amaçların bazıları ile

yeniden düzenlenebilir. Kısıtlamaları memnun etmedeki zorluk (genellikle doğrusallıktan daha fazla) kısıtlamaların sayısı ile artacaktır. Kısıtlamaların ikinci bir durumu onların ölçüsüdür. Bu ölçüyü kullanarak uygunluktan uzaktaki bir kısıtlamanın ihlaline değer biçilebilir. Üçüncü etken, tam memnuniyet bakımından kısıtlamanın kritikliğidir. Bir kısıtlama genel olarak zor bir şekilde formüle edilir. Yani, küçük ihlaller eğer çözüm başka bakımlardan diğer çözümlere üstün ise final çözümü için dikkate alınabilir. Evrimsel Algoritmalar özellikle araştırmanın her bir noktasındaki çözümlerin popülasyonu geri döndürülürken basit kısıtlamaların üstesinden gelebilir. Bu, kullanıcıya en iyi olan bir çözüm üzerinde basit bir kısıtlama ihlali yapan (teknik olarak uygun olmayan çözüm) teknik olarak uygun olan bir çözüm seçmeye olanak sağlar. Sınırlamaların 4.durumu kısıtlamaların memnun edilmesindeki zorluk olarak nitelendirilebilir.

Bu zorluk örnek uzayın(S) boyutuna nazaran uygun bölgenin(F) boyutu ile nitelenebilir. Zorluk deneyden önce bilinemeyebilir fakat iki yolla ölçülebilir. Birinci yol, kısıtlama ihlali yapan bir çözümün kısıtlama ihlali yapmayan bir çözümle değiştirilmesinin ne kadar sade ve basit olduğudur. İkinci yol, araştırma esnasında kısıtlamayı ihlal etme olasılığıdır. Örneğin, bir kısıtlama sık sık ihlal edilebilir fakat çözüm kolay bir şekilde uygun yapılabilir. Tam tersine, bir kısıtlama ihlalini ortadan kaldırmak çok zor olabilir fakat araştırmada nadiren meydana gelebilir [21].

İlk zamanlarda tasarlanmış olan evrimsel metotlar mevcut çözümlerin yeniden birleştirilmesi ve/veya karıştırılması ile yeni çözümler üretmekteydi. Belli şartlarda, bir kodlama, bir çoğalma (örneğin çaprazlama) operatörü ve bir mutasyon operatörü planlanmış olabilir, öyle ki uygun mevcut çözümler(ebeveynler) daima uygun yeni çözümler(çocuklar) üreteceklerdir. İkili tam sayı kullanımı bunu örnekler: 2 k-bit tam sayı arasındaki bit değişimi daima bir k-bit tam sayı üretir. Eğer k-bit tamsayı ve uygun çözümler arasında bire bir örtüşme sağlanırsa her yeni üretilen kodlama orijinal problemde uygun bir çözüme karşılık gelecektir [21].

Uygunluğu korumak için kodlama ve operatörlerin nasıl tanımlanabileceği açık değildir. Bunu sağlamak için önerilen bir yaklaşım; düzeltme operatörlerinin eklenmesiyle evrim operatörlerinin karmaşıklığını arttırmaktır. Böylece uygun kodlamaların üretilmesi

garanti edilir. Gezgin Satıcı Problemi (TSP) gibi problemler için, dengeleyici çakışma çözüm operatörleri(düzeltilme operatörleri) uygulanabilmektedir. Böylece sonuçta oluşan çocuk kodlamanın kendisi geçerli bir permütasyon olmaktadır. Ebeveyn çözüm yapılarının çoğunu ya da tamamını sürdüren uygun olmayan bir sonucun uygun bir çözüme yerini bırakması için kolayca modifiye edilebildiğinde düzeltme mekanizmaları etkili ve tesirli olabilmektedir [21].

Ne yazık ki birçok optimizasyon problemi onu uygun yapmak amacıyla uygun olmayan bir çözümün kolay olmayacak şekilde düzeltilmesini gerektiren kısıtlamalar ihtiva eder. Düzeltme operatörleri aşırı derecede hesaplama karmaşasına sahip olabilir ki bu, çocuklarda taşınan ebeveyn çözümlerin üstün nitelikli durumlarının dahi göz ardı edilmesine sebebiyet vererek evrimin temel gücünü bozguna uğratabilir. Bu durumu dikkate alan bir metot, evrim esnasında korunan sadece uygun çözümlere izin vermektir. Kısıtlamaları ihlal eden çözümler, direk atılarak dikkate alınmaması verimli olabilmektedir [21].

2.3. Ceza Fonksiyonlarının Tanıtımı

Ceza fonksiyonları, kısıtlamalı optimizasyon üzerinde on yıllardan beri literatürün bir parçası olmuştur. Ceza fonksiyonlarının 3 seviyesi mevcuttur: Uygun olmayan çözümlerin dikkate alınmadığı Bariyer(duvar) metotları; Uygunluk sınırı yakınında bir ceza uygulanan Kısmi Ceza Fonksiyonları; Uygun olmayan bölgenin başından sonuna kadar uygulanan Küresel Ceza Fonksiyonları. Kombinasyonel optimizasyon konularında popüler Lagrangian gevşeme metodu, problemin en zor kısıtlamalarını geçici olarak gevşetmek, uygun bölgeden çok uzakta dolaşmaktan kaçınmak için modifiye edilmiş bir amaç fonksiyon kullanmaktır. Genel olarak, bir ceza fonksiyonu yaklaşımı aşağıda belirtildiği biçimdedir. Verilmiş bir optimizasyon problemi [21],

$$\min f(x)$$

$$\text{s.t } x \in A, x \in B$$

x ; karar deęişkenleri vektörü, “ $x \in A$ ” göreceli olarak memnun etmesi kolay kısıtlamalar, “ $x \in B$ ” göreceli olarak memnun etmesi zor olan kısıtlamalardır. Problem şu şekilde yeniden düzenlenebilir:

$$\min f(x) + p(d(x,B))$$

$$\text{s.t } x \in A$$

$d(x,B)$: B bölgesinden x vektör çözümünün uzaklığını tanımlayan ölçü fonksiyonudur.

$p(\cdot)$: Monoton bir şekilde azalmayan ceza fonksiyonu şöyle ki $p(0)=0$ dır.

Eđer harici ceza fonksiyonu, $p(\cdot)$, B'nin yeterince dışında çabucak büyürse (P)' nin optimum çözümü aynı zamanda (R) için optimum çözüm olacaktır. Ayrıca, (R)' nin herhangi bir optimum çözümü (P) için optimum bir üst sınır sağlayacaktır ve bu sınır genel olarak A üzerinde $f(x)$ in basit bir şekilde optimize edilmesiyle elde edilenden daha sıkı olacaktır [21].

Pratikte, “ $x \in B$ ” kısıtlamalar eşitsizlik ve eşitlik kısıtlamaları olarak aşağıda verilen formdaki gibi ifade edilir:

$$g_i(X) \leq 0, \quad i= 1, \dots, q$$

$$h_j(X) = 0, \quad j= p+1, \dots, m$$

q : Eşitsizlik kısıtlamalarının sayısı.

$m-q$: Eşitlik kısıtlamalarının sayısı.

Göz ardı edilmiş kısıtlamalar için etkili ve verimli temsili bir ceza fonksiyonu bulmak zor olabilir. Verilen bir problem örneğinde ceza fonksiyonunu ayarlamak için çaba gereklidir aksi durumda araştırma esnasında onu defalarca hesaplamak nihai çözüm kalitesindeki herhangi bir kazanımı boşa çıkarabilir. Optimum çözüm çoęunlukla uygun bölgenin sınırında kalabilir, bu durumda optimum çözümün yapısına çok benzeyen

çözümlerin birçoğu uygun olmamış sayılacaktır. Bu nedenle, sadece uygun çözümler için kısıtlayıcı araştırma yapmak ya da optimuma doğru popülasyonu hareket ettirecek şemaları bulmak için çok sert cezalar dayatmak çözümleri zor hale getirecektir. Diğer taraftan, eğer ceza yeterince sert olmazsa çok geniş bir bölge araştırılmış olacak ve araştırma zamanının çoğunu uygun bölgelerden uzaktaki bölgeleri araştırmak için kullanmış olacaktır [21].

2.4. Statik Ceza Fonksiyonları

Uygun olmayan çözümleri cezalandırmanın basit bir metodu herhangi bir şekilde uygunluğu ihlal eden bu çözümlere sabit bir ceza uygulamaktır. Cezalı amaç fonksiyonu, cezasız amaç fonksiyonuna bir ceza eklenmiş durumu olacaktır (bir minimizasyon problemi için düşünüldüğünde). Basit ceza fonksiyonunun bir çeşidi, birçok kısıtlamaların olduğu ve ihlal edilmiş kısıtlamaların sayısının temel alındığı bir ölçü eklemektir. m kısıtlamalı bir problem için ceza fonksiyonu aşağıdaki şekilde tanımlanabilir [21]:

$$f_p(x) = f(x) + \sum_{i=1}^m C_i \delta_i$$

$$\delta_i = \begin{cases} 1, & i \text{ kısıtlaması ihlal edildiyse} \\ 0, & i \text{ kısıtlaması memnun edildiyse} \end{cases}$$

$f_p(x)$: Cezalı amaç fonksiyonu.

$f(x)$: Cezalandırılmamış amaç fonksiyonu.

C_i : i . Kısıtlamanın ihlali için uygulanan bir sabit

Bu ceza fonksiyonu sadece ihlal edilmiş kısıtlamaların sayısını temel alır ve genellikle uygun bölgeden olan bazı uzaklık ölçülerini temel alan bir yaklaşımdır.

Daha alışılmış ve etkili olanı uygunluğa olan uzaklığa göre ceza vermektir. Yalnızca ihlale uğramış kısıtlamaların sayısı tarafından belirlenebilecek olan uzaklık göz önünde tutularak bu, sabit ceza fonksiyonlarında kabaca yapılır. Daha gelişmiş ve daha etkili ceza, her bir kısıtlama için bir uzaklık ölçüsü içerir ve uygunluğa olan uzaklık ile birlikte daha da sert olan ceza eklenir [21].

Bir minimizasyon problemi için genel formül aşağıdaki şekildedir:

$$f_p(x) = f(x) + \sum_{i=1}^m C_i \delta_i^K$$

$$d_i = \begin{cases} \delta_i g_i(x) , & i = 1, \dots, q \\ |h_i(x)| , & i = q+1, \dots, m \end{cases}$$

d_i : Çözüm x' e uygulanan i kısıtlamasının uzaklık ölçüsü.

K : Kullanıcı tanımlı üs kuvveti (sıklıkla 1 ya da 2 kullanılır.)

C_i 'nin seçimi çok daha zordur. Çözümü düzeltmek için maksimum ya da beklenen maliyet temel alınarak seçilir. Çoğu problem için, bununla birlikte, C_i ' i belirlemek bu mantığı kullanarak mümkün değildir. Bunun yerine, bir kısıtlamayı memnun etmenin zorluğu ve bir kısıtlama ihlalinin ciddiliği ya da deneyerek belirlenmiş olma durumları da göz önüne alınarak birden çok kısıtlamanın uzaklık ölçülerinin göreceli ölçülmesine dayanacak şekilde tahmin edilmiş olmalıdır [21].

2.5. Dinamik Ceza Fonksiyonları

Statik ceza fonksiyonlarının başlıca eksikliği C_i katsayısı için kriter tanımlama konusunda kullanıcının yetersizliğidir. Aynı zamanda, uygun olmayan bölgenin araştırılmasına izin verilmesi ile henüz hala final çözümün uygun olmasını gerektiren çelişkili amaçlarla karışmaktadır. Bu zorlukları hafifleten uzaklık temelli ceza fonksiyonlarının bir çeşidi genel olarak araştırma ilerlerken verilen bir uzaklık için cezanın şiddetini arttıran dinamik bir durum katmaktır. Bu yaklaşımda, neticede final çözümü uygun bölgeye hareket ettirmek için maruz bırakılan ceza devamlı olarak artarken, araştırmanın ilk safhalarında büyük ölçüde uygun olmayan çözümlere izin verilen bir özelliğe sahiptir. Araştırmanın uzunluğuna dayanarak dinamik bir durum katan uzaklık temelli ceza metodunun genel bir formu, t , bir minimizasyon problemi için aşağıdaki şekildedir [21]:

$$f_p(x,t) = f(x) + \sum_{i=1}^m s_i(t) d_i^K$$

$s_i(t)$: t ile monoton bir şekilde azalmayan bir kriter.

t için ölçüler araştırılan çözümlerin sayısını ya da jenerasyon sayısını içerir. Bu dinamik ceza formülasyonlarının ortak kaygısı, evrimin sonunda uygun çözümlerle sonuçlanıp sonuçlanmayacağıdır. Eğer $s_i(t)$ çok yumuşaksa, sonunda, uygun olmayan çözümlerle neticelenebilir ve eğer $s_i(t)$ çok sertse, araştırma optimal olmayan çözümlere yakınsayabilir. Bu nedenden dolayı, bu ceza fonksiyonları tipik olarak daha iyi icra edilmesi için özel bir problem ayarına ihtiyaç duyar. $s_i(t)$ 'nin belirgin bir örneği aşağıdaki gibidir [21]:

$$s_i(t) = (C_i t)^\alpha \quad \alpha = 1 \text{ ya da } 2 \text{ ye eşit olan bir sabittir.}$$

2.6. Uyarlanabilir Ceza Fonksiyonları

Ceza fonksiyonunun içerisine araştırmanın uzunluğu ile birlikte uzaklığı katmak genel olarak etkili olurken bu cezalar araştırmanın diğer safhalarında dikkate alınmaz. Bu

bakımdan, araştırmanın sürmekte olan başarısı için (ya da noksanlık nedeniyle) onlar uyarlamalı değildir ve cazibesiz bölgelerden uzaklaşmak ya da çekici bölgeler başta olmak üzere halihazırda gözlenenleri baz alarak araştırmaya kılavuzluk edemez. Bazı araştırmacılar, en iyi uygunluk ya da uygunsuzluğu temel alacak şekilde yeniden düzenleyerek ceza fonksiyonları önermişlerdir. Onların ceza fonksiyonları aşağıda gösterildiği gibi evrim esnasında dayatılan cezanın artışı ya da azalışından herhangi birine izin verir. Ceza fonksiyonunu çarpanını uyarlamalı olarak güncellemek için ve birbirini izleyen N_f jenerasyonunun mesafeleri üzerinde en iyi çözümün uygunluğunun değerlendirilmesi için iki sabitin seçimini içerir, β_1 ve β_2 ($\beta_1 > \beta_2 > 1$). Araştırma ilerlerken her bir N_f jenerasyonunda ceza fonksiyon çarpanı bu süre esnasında en iyi çözümün uygun olup olmadığı temel alınarak güncellenir. Belirli bir biçimde ceza fonksiyonu aşağıdaki şekildedir [21]:

$$\lambda_{k+1} = \begin{cases} \lambda_k \beta_1, & \text{Eğer önceki } N_f \text{ jenerasyonları uygun olmayan en iyi çözüme sahipse} \\ \lambda_k / \beta_2, & \text{Eğer önceki } N_f \text{ jenerasyonları uygun en iyi çözüme sahipse.} \\ \lambda_k, & \text{diğer durumlarda.} \end{cases}$$

Yukarıda temel olarak ele alınan ceza fonksiyon yaklaşımlarının her birisinin sahip olduğu avantajlı ve dezavantajlı yönler göz önünde bulundurularak her açıdan daha başarılı ceza fonksiyonları oluşturulmuştur. Ceza katsayısı için iyi bir değer tanımlama görevi açısından kullanıcıyı rahatlatmayı hedefleyen ve kısıtlamalı problemlerde sıklıkla kullanılan yaklaşımlardan ikisi NFT ve EC metotlarıdır.

2.6.1. Yakın Uygunluk Eşik Değeri Metodu(Near-Feasibility Threshold)

NFT, uygun bölgenin ötesinde “umut veren ” bir bölgeye tekabül eden bir yakın uygunluk eşik değeri kavramıdır. Bazı araştırmacılar, ceza fonksiyonlarında araştırmanın uzunluğunu ve kısıtlama güçlüğünün geri bildirimini ikisini birden

kullanılmaktadırlar. Bu ceza fonksiyonları her bir kısıtlama için yakın uygunluk eşik değerinin(NFT) tahminini içerir. Kavramsal olarak, NFT uygun bölgeden olan eşik değeri uzaklığıdır ki kullanıcı araştırmannın “çözümüne yaklaşıldığını” göz önüne alır. Ceza fonksiyonları, evrimsel algoritmayı uygun bölgenin içerisinde ve uygun bölgenin yakın uygunluk eşik değeri civarında araştırmak için cesaretlendirir ve bu eşik değerin ötesindeki araştırmalarda ise caydırır. Ayrıca, şimdiye kadar bulunan en iyi uygun değer ve en iyi uygun olmayan değer arasındaki aralığı dikkate almak için ceza fonksiyonlarına uyarlanabilir bir terim eklenmiştir [21, 22].

$$f_p(x,t) = f(x) + (F_{feas}(t) - F_{all}(t)) \sum_{i=1}^m \left(\frac{d_i}{NFT_i} \right)^\alpha$$

$F_{all}(t)$: Şimdiye dek bulunan en iyi çözümün cezasız değerini gösterir.

$F_{feas}(t)$: Şimdiye dek bulunan en iyi uygun çözümün değerini gösterir.

α : Önceden tanımlanmış bir sertlik parametresi.

$F_{all}(t)$ ve $F_{feas}(t)$ terimleri birtakım amaçlara hizmet eder. İlk olarak, araştırmannın sonuçlarına dayanan cezanın uyarlamalı olarak ölçeklenmesini sağlar. İkinci olarak ise, özgün bir araştırma ve kısıtlamaya belirli ceza sağlamak için NFT_i terimi ile birleşir. NFT' nin genel formu aşağıdaki şekildedir [21]:

$$NFT = \frac{NFT_0}{1 + \Lambda}$$

NFT_0 : NFT için bir üst sınır.

Λ : Araştırma geçmişine dayanan NFT yi ayarlamak için kullanılan dinamik bir araştırma parametresi.

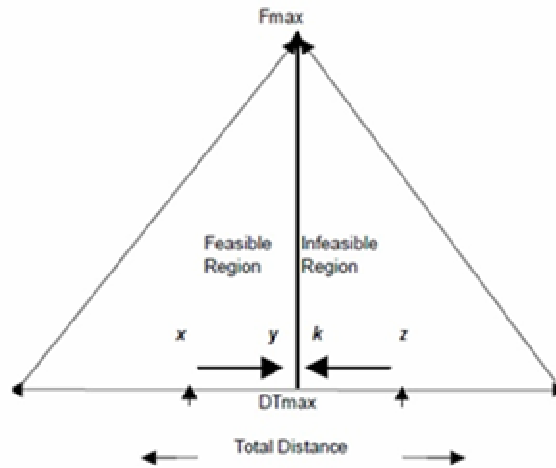
En kolay durumda, Λ , 0' a ayarlanabilir ve statik bir NFT sonuçlanır. Λ , aynı zamanda araştırmannın bir fonksiyonu olarak da tanımlanmış olabilir. Örneğin, jenerasyon sayısının(t) bir fonksiyonu;

$$\Lambda = f(t) = \lambda_t \quad NFT = \frac{NFT_0}{1 + \lambda * Ng}$$

N_g : Jenerasyon sayısını temsil eder.

λ 'nın pozitif bir değeri monoton bir şekilde azalan NFT ile sonuçlanır (ve bu yüzden daha büyük bir ceza) ve araştırma ilerlerken daha büyük λ daha da hızlı azalan NFT ile sonuçlanır.

Eğer NFT deneyden önce iyi tanımlanmamış ise, araştırmayı uygun bölgeye tekrarlanarak sevk edecek şekilde başlangıçta pozitif sabit λ ' lı büyük bir değere ayarlanabilir [21, 22].



Şekil 2.1. NFT metodu örnekleme [32]

2.6.2. Epsilon Kısıtlama Metodu(Epsilon Constraint Method)

Epsilon kısıtlama metodu çok amaçlı bir optimizasyon problemini birtakım kısıtlamalı tekil amaç problemlerine dönüştüren matematiksel programlama tekniğidir. Bu metot, çoğu evrimsel algoritmanın yaptığı gibi tek bir çalışmada bir takım baskın olmayan çözümler üretmediğinden dolayı evrimsel hesaplamada çok sık kullanılmamıştır. Dahası, bu metotta birçok tekil amaç optimizasyonlar icra edildiğinden dolayı çok amaçlı “kolay” problemleri çözerken göreceli olarak pahalı bulunur. Epsilon kısıtlama

metodunun arkasındaki temel düşünce, gelecek jenerasyon için seçim yaparken daha küçük toplam kısıtlama ihlali çözümler, $\varepsilon(k)$ uygun çözümler olarak işlem görür. Çok amaçlı problemi birçok tek amaçlı kısıtlamalı problemlere dönüştürmek için aşağıdaki prosedürü kullanır (tüm amaç fonksiyonlarının minimize edilmesi göz önüne alınacak olursa) [23, 26]:

$$\begin{aligned} & \text{Min } f_k(x) \\ & \text{s.t } f_i(x) \leq \varepsilon_i \quad i=1,2,\dots,m \quad i \neq k \\ & x \in X, k \in \{1,2,\dots,m\} \end{aligned}$$

X : Eşitlik ya da eşitsizlik kısıtlamaları ile tanımlanabilen uygun bölge.

ε : Üst sınır vektörü ($\varepsilon_1, \varepsilon_2, \dots, \varepsilon_m$) her bir amacın sahip olabileceği maksimum değeri belirtir.

Epsilon kısıtlama metodu, ε parametresinin kullanılmasıyla kısıtlamaların gevşetilmesini kontrol eder. Bir kısıtlamalı optimizasyon problemini çözerken aktif kısıtlamaların var olması işi meşakkatli hale getirir. Eşitlik kısıtlamalı problemlerde yüksek kalitede çözümler elde etmek için ε parametresinin uygun şekilde kontrol edilmesi temel esastır. ε düzeyi, jenerasyon sayacı k , kontrol jenerasyonu T_c ' ye ulaşıncaya kadar güncellenir. Jenerasyon sayacı T_c sınırını aştıktan sonra kısıtlama ihlali olmayan çözümleri elde etmek için ε düzeyi "0" olarak ayarlanır [22].

$$\varepsilon(0) = V(X_\theta)$$

$$\varepsilon(k) = \begin{cases} \varepsilon(0) \left(1 - \frac{k}{T_c}\right)^{cp}, & 0 < k < T_c \\ 0, & k \geq T_c \end{cases}$$

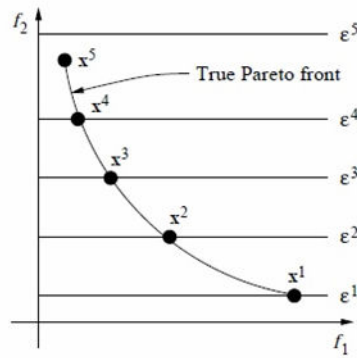
X_θ : θ . üst birey ve $\theta = (0.05 * NP)$.

Tavsiye edilen parametre dağılımı:

$$T_c \in [0.1T_{\max}, 0.8T_{\max}] \text{ ve } c_p \in [2,10]$$

Bu metodun amacı, kabul edilebilir epsilon seviyesi ile diğer amaçları kısıtlama olarak hesaba katarken bir seferde bir amaç fonksiyonunu minimize etmektir. Her bir amacı sadece bir kez optimize eder. Yani diğer tüm amaç fonksiyonları ek kısıtlamalar vasıtası ile üstten sınırlandırılırken amaç fonksiyonlarının biri küçültülür. Geleneksel epsilon kısıtlama metodu, kısıtlama sınırını önceden tanımlanmış sabit bir δ ile tekrarlanarak arttırmaktadır [23, 24].

Birtakım optimum sonuçlar elde edilmesi ve her bir yeni vektör için yeni bir optimizasyon süreci yapmak amacı ile her bir amaç için üst sınır vektörünün değişime uğraması gerekmektedir. Üst sınırın farklı değerlerini kullanarak farklı noktaların üretimi Şekil 2.2. de gösterilmiştir [25, 26].



Şekil 2.2. EC metodu ile farklı çözümler üretilmesi [31]

Herhangi bir non-liner çok amaçlı optimizasyon problemi için, bir epsilon kısıtlama probleminin çözümü, güçsüz bir Pareto optimum çözümü ürün olarak verir. Doğru bir pareto optimum çözümü eğer çözüm eşsizse(mükemmelse) ya da çözümü rapor etmeden önce optimizasyon tüm amaçlar için yapıldıysa elde edilebilir. Bununla birlikte, çözümlerin üretilme hızını geliştirmek için pareto optimum takımının bir benzerini elde etmek amacıyla nokta başına yalnızca bir optimizasyon uygulanabilir [26].

3. BÖLÜM

BULGULAR

Bir sistem veya sistemin bir kısmına ait ayrıntıları incelemek için yapılan faaliyetlere deney denilmektedir. Deney terimi istatistik biliminde ise yeni gerçekler meydana getirmek veya var olanı doğrulamak ya da çürütmek için kullanılan bir iş olarak tanımlanmaktadır. Deneyler kesin ve karşılaştırmalı deneyler ve tek ve çok etmenli deneyler olmak üzere iki farklı şekilde gerçekleştirilebilmektedir.

Kesin ve karşılaştırmalı deneyler, bilinmeyeni ortaya çıkarmak aynı zamanda bilineni de doğrulamak ya da çürütmek için yapılmaktadır.

Tek ve çok etmenli deneyler ise bir veya daha fazla etmenin durum ya da işlem üzerindeki etkisini belirlemek için yapılan deney türüdür ve mevcut duruma özel olarak tasarlanır.

Deney tasarımı ise, bir işlemdeki girdilerde sistematik değişiklikler yapılarak girdilerin sonuç üzerindeki etkilerini gözlemlemek için kullanılan bir tekniktir.

Deney tasarımının amaçları kısaca aşağıdaki şekilde sıralanabilir.

- Sonuç üzerinde en etkin girdi faktörlerini belirlemek.
- Performans karakteristiğini hedef değere en yakın sonuca ulaştıracak girdi faktörlerinin seviyelerini belirlemek.
- Kontrol edilemeyen girdi faktörlerinin çıktı üzerindeki etkisinin en az olacağı kontrol edilebilen girdi faktörleri kombinasyonunu oluşturmak.

Deney tasarımı, süreci etkileyen her türlü etken “faktör” olarak adlandırılır. Faktörlerin çeşitli kategorileri, dereceleri ve yoğunlukları ise “seviye” olarak adlandırılır. Bir deney tasarımı sırasında birden fazla faktör ve seviye grupları bulunabilir [27]. Denemenin planlama ve yürütülme şekli denemenin stratejisini belirlemektedir. 4 tür deneme stratejisi vardır, bunlar: en iyi tahmin, bir defada bir faktör, faktöriyel tasarım ve kısmi faktöriyel tasarımdır.

En iyi tahmin yaklaşımında sonuç üzerinde önemli etkisi olduğu düşünülen faktörlerin rassal bir bileşimi ele alınır ve test edilir. Ancak bu yöntemin dezavantajları vardır. Bunlardan birincisi, önemli etkisi olduğu düşünülen faktörlerin bileşimi iyi sonuç vermeyebilir. Bu durumda araştırmacı yeni bir bileşim oluşturmak zorunda kalır. Bu durum en iyi bileşimi elde etme garantisi olmadan devam eder. Bu da fazla zaman kaybına sebep olabilir ya da ikinci bir sakınca araştırmacı oluşturduğu bileşimin iyi sonuç verdiğini görünce test etmeyi bırakır. Bu durumda da daha iyi sonuçlardan vazgeçilmiş olunur.

Bir defada bir faktör yaklaşımında her bir faktör başlangıç seviyesinde, diğer faktörlerin sabit tutularak onların üzerine dağıtılması ile test edilir. Her faktör ile sabit tutulan faktörlerin değiştirilmesi ile sonucun nasıl etkilendiği gözlemlenir. Yaklaşımın dezavantajı her bir faktör tek tek incelendiğinden faktörlerin birbiri ile etkileşiminin gözlemlenemiyor olmasıdır. Etkileşim bir faktörün, diğer faktöre ait farklı seviyelerde aynı etkiyi yaratmamasıdır. Etkileşimi olan faktörlerin bulunduğu testlerde bu yaklaşımın kullanılması araştırmacıyı doğru sonuca ulaştırılamaz.

Üçüncü bir yaklaşım olan faktöriyel deneme, faktörlerin tek tek değil hep birlikte ele alınması ve birbirleri ile etkileşiminin de ele alınması sebebi ile daha güvenilir bir yaklaşımdır. Faktör sayısı arttıkça deneme sayısı da artmaktadır bu da test süresinin uzaması şeklinde bir dezavantaj oluşturmaktadır. Zaten faktör seviyelerinin bütün bileşimlerini kullanmak gereksiz olabilmektedir. Böyle durumlarda ise kısmi faktöriyel deneme yaklaşımı kullanılmaktadır. Kısmi faktöriyel deneme yaklaşımında faktörler içerisinde altkümelere ayrılarak işlem görür. Bu şekilde zamandan tasarruf edilir [28].

3.1. Deney Tasarım İlkeleri

Arařtırmacıyı doęru sonuca gtrecek deneylerin ncelikle iyi tasarlanması gereklidir. Deneylerin iyi tasarlanması iin Fisher tarafından geliřtirilen tekrarlar, rassallık ve bloklama ilkelerine baęlı kalınmalıdır [29].

Tekrarlar

Tekrarlar deneyin birden fazla kere yapılması demektir. Aynı faktrn tekrar tekrar llmesi deęil esas denemenin tekrarlanması anlamında kullanılmaktadır. Tekrarlar, deęiřkenlięe sebep olan faktrlerin ortaya ıkardığı hataların lm iin kullanılmaktadır. Tekrarlar olmadan deęiřkenlięe sebep olan unsurlar belirlenememektedir.

Tekrarların iki nemli zellięi vardır. Birincisi, deneysel hataları giderme imkanı vermesidir, ikincisi ise eęer rnek kmenin ortalaması herhangi bir faktrn etkisinin tahmin edilmesinde kullanılıyor ise sonucun daha kesin olmasını saęlamasıdır.

Rassallık

Rassallık, deneysel materyallerin herhangi bir kurala uygun olmaması, tesadfi olması demektir. Deneysel verilerin toplanmasının da rassallık ilkesine baęlı olarak yapılması sonucun daha gereęi yansımasının en nemli gerekesidir. Deney sonularını analiz etmede de rassallık kullanılacak yntemler iin gereklidir. Rassallık nyargıyı giderici bir etkidir.

Bloklama

Bloklama, deneyin hassasiyetini artıran bir ilkedir. Deneme yapılacak alanın homojen olacak řekilde veya belirli farklı zelliklere gre blmlere ayrılması řeklinde yapılır.

3.2. Deney Tasarımın Tarihçesi

Deney tasarımı ilk olarak 1920'lerde ünlü İngiliz istatistikçi Sir A. Ronald Fisher tarafından geliştirilmiştir. Fisher, yöntemi tarım alanında ürünlerin verimliliğini artırmak amacı ile birtakım çalışmalar yaparak geliştirmiştir. Fisher ürün verimliliğini artırmak için verimliliği aynı olacak şekilde ekim yapılacak toprağı parsellere ayırmıştır. Ayırdığı parsellere ürünleri ve gübre çeşitlerini rassal olarak dağıtarak ekim yapılmasını sağlamıştır. Fisher'ın yaptığı bu çalışma deney tasarımının temellerini oluşturmaktadır. Deney tasarımı 1950'li yıllara kadar kimya ve ilaç endüstrisinde de kullanılmıştır. 1950'li yıllardan sonra deney tasarımının, toplam kalite yöntemi ile ilgili çalışmalarda başarı göstermesi yöntemin daha yaygınlaşmasına sebep olmuştur. 1970'li yıllarda yaşanan sanayi devriminden sonra yöntem üretim sektöründe de yaygın olarak kullanılmaya başlanmıştır.

Motorola firmasının kalite grubundan K. Bhote'nin "World Class Quality" adlı kitabında kalite geliştirme de üç yöntemin çok önemli olduğunu belirtmiştir. Bunlar, geleneksel muayene, istatistiksel işlem kontrol ve deney tasarım yöntemleridir. Japonya'da kalite kavramının gelişmesinde en büyük katkıyı deney tasarımı sağlamıştır. Amerika'da ise 1980'lerden sonra deney tasarımı yöntemi gelişmeye başlamıştır [29].

3.3. Deney Tasarımın Kullanım Alanları

Deney tasarımı yöntemi pek çok alanda kullanılabilir. Özellikle ürün ya da işlem tasarımındaki problemleri ve üretim aşamasındaki problemleri çözmeye kullanılmaktadır. Örneğin,

- Tasarım bileşenlerinin değerlendirilmesi ve karşılaştırılmasında,
- Malzeme seçiminde alternatiflerin değerlendirilmesinde
- Tasarım parametrelerinin seçiminde
- Ürün performansını belirleyici parametrelerin seçilmesinde deney tasarımı yöntemi etkin şekilde kullanılabilir [29].

3.4. Deney Tasarımının Adımları

- a. Problemin tanımlanması
- b. Faktörlerin, seviyelerin ve sıranın seçimi
- c. Tepki değişkeninin seçimi
- d. Deney tasarımının seçimi
- e. Deneyi uygulama
- f. Verilerin istatistiksel analizi
- g. Sonuçlar [29].

3.4.1. Problemin Tanımlanması

Problemin tanımlanması kolay gibi görünse de aslında tasarımın en önemli adımlarından birisidir. Tasarımdan en iyi sonucu alabilmek için problemi iyi belirlemek gerekmektedir ve görünürde birden fazla problem olsa da temel problemi saptamak gerekmektedir. Bunun için konu ile ilgili veriler sağlıklı şekilde tüm birimlerden toplanmalı ve takım yaklaşımı ile problemin saptanması için beyin fırtınası yapılmalıdır.

3.4.2. Faktörlerin, Seviyelerin ve Sıranın Seçimi

Tasarımın ilk aşaması olan problem belirlemeden sonra tasarım sonucunu etkileyen faktörler ve bu faktörlere ait seviyeler belirlenmelidir. Faktörler ve seviyeler belirlenirken daha önce yapılmış çalışmalardaki saptamalardan faydalanılmalıdır.

3.4.3. Tepki Değişkeninin Seçimi

Belirlenen problem doğrultusunda, faktör ve seviyeleri kullanılarak yapılan tasarım alternatifleri denenir ve bu denemelerin sonucunda doğru tasarımın belirlenmesinde kullanılacak performans kriteri saptanır. Bu kriter ölçülen karakteristiğin ortalaması veya standart sapması ya da her ikisi birden olabilmektedir.

3.4.4. Deney Tasarımının Seçimi

İlk üç aşama gerçekleştirildikten sonra tasarımcı denemeyi uygulayacağı tasarımı seçmektedir.

3.4.5. Deneyi Uygulama

Asıl verilerin elde edildiği aşamadır.

3.4.6. Verilerin İstatistiksel Analizi

Bu adımda elde edilen asıl veriler istatistiksel yöntemler ile analiz edilmektedir. Bu yöntemler grafiksel yöntem ya da varyans analizi yöntemi olabilmektedir.

3.4.7. Sonuçlar

Yapılan istatistiksel analiz sonucunda artık kesin ve net şekilde konuşulabilir.

3.5. Faktöriyel Denemeler

Araştırmalarda çok kez sonuca etkisi olan faktörlerin etki ölçüleri inceleme konusu olabilmektedir. Bu faktörlerin tek başlarına etkileri ve birbirleri ile aynı anda sonuç üzerindeki yarattıkları etki belirlenmeye çalışılmıştır. Faktörlerin tek başlarına etkileri yeterli olmamaktadır çünkü faktörlerin başka faktörler ile bir araya geldiği zaman sonuç üzerindeki etkileri tek başlarına yarattıkları etkiden farklı olabilmektedir. Bu şekilde sonuç üzerindeki önemli etki belirlenebilmektedir ve bu faktör üzerindeki değişiklikler ile en iyi sonuç elde edilebilmektedir [28].

3.5.1. Tam Faktöriyel Deneyler

Tek Faktörlü Deneyler: Her bir faktörün sonuç üzerindeki etkisinin tek tek ölçüldüğü deney türüdür. Seçilen faktörün sonuç üzerinde etkisi olmadığı belirlendiği takdirde yeni bir faktör denenmektedir. Bu işlem beklenen sonuca ulaşıncaya kadar sürdürülmektedir. Bu deneme türünde faktörlerin birbiri ile etkileşimi ölçülememektedir. Yani faktörlerin farklı faktörler ile farklı şekilde yan yana gelerek sonuç üzerindeki etkileri bu deneme türü ile gözlemlenememektedir. Ayrıca faktör

sayısı çok olduđu zaman her denemeyi yapmak zaman kaybına ve dolayısı ile fazla maliyete sebep olabilmektedir bu nedenle faktör sayısının fazla olduđu durumlarda ve faktörlerin etkileşiminin ölçülmesi istendiđi durumlarda tam faktöriyel deney türünün kullanılması deney tasarımcısına avantaj sağlayabilmektedir.

Tam faktöriyel deneyler, her bir faktörün her bir seviyesi için eşit sayıda deneme ile faktörlerin diđer faktörlerden bağımsız olarak ürün performansına etkisini ölçmede kullanılan deney türüdür. Tam faktöriyel deney stratejisi faktörlerin birleştirilmiş halini inceler ve hem tek tek her faktörün sonuç üzerindeki etkisini hem de faktörlerin birlikte sonuç üzerindeki etkisini belirlemede kullanılmaktadır. Kritik faktörleri saptamaktadır. Tam faktöriyel deneyler ürün performansına etki edecek faktör sayısının 5 ten küçük olduđu durumlarda kullanılabilir çünkü daha fazla faktör olduđu zaman deneme sayısı üstel olarak artmaktadır. Bu da deneme sayısının çok fazla olması anlamına gelmektedir. Deneme sayısı arttıkça sonuca ulaşmada zaman problemi ortaya çıkabilmektedir. Faktör sayısının 5' ten fazla olduđu durumlarda kısmi faktöriyel deneyler kullanılmaktadır.

Tam faktöriyel deneylerde, faktörlerin tek başlarına ve birlikte sonuca olan etkilerini belirleyebilmek için gerekli deneme sayısı a^k formülü ile hesaplanmaktadır.

a: faktörün seviye sayısı

k: incelenen faktör sayısı

Örneđin, 3 faktörlü 2 seviyeli bir deneyin $2^3=8$ tane denemesi yapılmaktadır.

Faktörlerin seviye sayısı belirlenirken 2 yöntem kullanılabilir. Tecrübe ile ürün performansına etkisi olası seviyeler seçilerek deney yapılabilir ya da seviyelerin oluşturduđu bir kümeden rassal olarak seçilip deney yapılabilir [30].

3.6. Kısıt Yönetimi Parametrelerinin Deney Tasarımı İle Belirlenmesi

Bu bölümde çalışmanın sonucunu etkileyecek olan faktörleri ve seviyelerini belirlemek için deney tasarımı çalışması yapılmıştır. Tez çalışmasında Diferansiyel Gelişim

Algoritması ve problemlerdeki kısıtlamaların üstesinden gelmek için Yakın Uygunluk Eşik Değeri Metodu ile Epsilon Constraint Metodu kullanılmaktadır. Öncelikle algoritmalarda kullanılacak, sonucu belirleyici faktörlerin neler olduğuna karar verilmiştir. Bunlar, Diferansiyel Gelişim Algoritması ile birlikte kullanılacak olan Yakın Uygunluk Eşik Değeri Metodu için popülasyon büyüklüğü, mutasyon oranı ve çaprazlama oranı iken Epsilon Constraint Metodu için popülasyon büyüklüğü, mutasyon oranı ve çaprazlama oranına ek olarak bu metodun parametreleri olan kontrol jenerasyonu, Teta ve cp (üssü) dahil edilmiştir. Faktörler kararlaştırıldıktan sonra seviyeler belirlenmiştir. Seviye olarak, literatürdeki konu ile ilgili çalışmalarda genellikle iki seviye kullanıldığı gözlemlenmesine rağmen hassasiyeti arttırmak adına Yakın Uygunluk Eşik Değeri Metodunda her bir faktör için 1-2-3-4 olmak üzere dört ölçü kullanılmıştır. Epsilon Constraint Metodunda ise her bir faktör için yüksek ve düşük olmak üzere iki ölçü kullanılmıştır. Faktörler ve seviyeleri bu şekilde belirlendikten sonra tam faktöriyel deney yapılması kararlaştırılmıştır.

DGA ile birlikte kullanılacak olan NFT ve EC Metotları için faktör seviyeleri tablo 3.1. ve 3.2. deki gibi belirlenmiştir.

Tablo 3.1. Yakın uygunluk eşik değeri metodu için olası faktör seviyeleri ve değerleri

Seviye	$A(CR)$	$B(MR)$	$C(NP)$
1	0,2	0,2	30
2	0,4	0,4	50
3	0,6	0,6	70
4	0,9	0,9	100

Tablo 3.2. Epsilon constraint metodu için olası faktör seviyeleri ve değerleri

Seviye	$A(Tc)$	$B(Teta)$	$C(cp)$	$D(CR)$	$E(MR)$	$F(NP)$
Düşük (-1)	0,1	0,1	2	0,2	0,2	30
Yüksek(1)	0,2	0,2	5	0,9	0,9	100

NFT metodu için 20.000-100.000 ve 200.000 fonksiyon değerlendirme sayıları için $4^3 = 64$ ve $64*3 = 192$ deneme yapılmıştır. (bkz. Tablo 3.3, Tablo 3.4, Tablo 3.5) Aynı şekilde EC metodu için $2^6 = 64$ ve $64*3 = 192$ deneme yapılmıştır (bkz. Tablo 3.9, Tablo 3.10, Tablo 3.11). Denemeler EkA da belirtilen ve çeşitli zorluklardan ve farklı

yapılardan oluşan 18 adet Benchmark test problemi üzerinde yapılmıştır. Denemeler sonucunda elde edilen ortalama uygunluk değerleri alınarak uygun deney tasarımı belirlenmiştir.

Tablo 3.3. Yakın uygunluk eşik değeri metodunun 20.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri

DENEME	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
1	3	3	3	0,29
2	1	3	3	0,3
3	2	1	3	0,32
4	2	1	1	0,32
5	2	4	4	0,27
6	4	3	4	0,28
7	4	3	2	0,32
8	4	4	3	0,25
9	1	4	1	0,32
10	3	2	4	0,29
11	2	2	4	0,29
12	2	2	3	0,3
13	3	3	2	0,31
14	2	3	1	0,34
15	4	1	2	0,3
16	3	4	3	0,27
17	3	1	1	0,32
18	4	3	1	0,37
19	2	2	2	0,33
20	2	3	2	0,31
21	3	2	1	0,36
22	1	4	4	0,28
23	1	3	4	0,32
24	4	1	4	0,32
25	2	2	1	0,35
26	4	2	2	0,34
27	4	3	3	0,28
28	3	2	2	0,34
29	2	1	4	0,3
30	1	2	1	0,33
31	1	3	2	0,32
32	3	1	3	0,33

Tablo 3.3. Devamı

DENEME	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
33	4	1	1	0,48
34	2	1	2	0,34
35	1	2	3	0,31
36	3	2	3	0,33
37	2	4	2	0,31
38	1	2	2	0,33
39	2	3	4	0,31
40	3	1	2	0,31
41	4	2	1	0,32
42	1	2	4	0,31
43	3	4	2	0,28
44	4	4	4	0,26
45	1	4	2	0,3
46	4	4	2	0,26
47	2	4	1	0,3
48	4	4	1	0,28
49	3	3	4	0,28
50	3	4	1	0,29
51	3	4	4	0,25
52	4	2	4	0,35
53	2	3	3	0,3
54	3	1	4	0,31
55	1	1	2	0,32
56	2	4	3	0,29
57	1	1	3	0,32
58	1	1	4	0,32
59	1	4	3	0,3
60	1	1	1	0,35
61	4	1	3	0,31
62	4	2	3	0,35
63	3	3	1	0,34
64	1	3	1	0,32

Yakın Uygunluk Eşik Değeri Metodunun 20.000 fonksiyon hesaplaması için yapılan denemede ortalama uygunluk değerinin **Deneme 33'** te en uygun değeri verdiği gözlemlenmektedir (bkz. Tablo 3.3), yani çaprazlama oranının yüksek, mutasyon oranının düşük ve popülasyon büyüklüğünün düşük olarak belirlenmesinin uygun olduğu saptanmaktadır.

Tablo 3.4. Yakın uygunluk eşik değeri metodunun 100.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri

DENEME	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
1	3	3	3	0,42
2	1	3	3	0,41
3	2	1	3	0,42
4	2	1	1	0,40
5	2	4	4	0,36
6	4	3	4	0,56
7	4	3	2	0,48
8	4	4	3	0,34
9	1	4	1	0,43
10	3	2	4	0,43
11	2	2	4	0,40
12	2	2	3	0,43
13	3	3	2	0,43
14	2	3	1	0,41
15	4	1	2	0,46
16	3	4	3	0,38
17	3	1	1	0,35
18	4	3	1	0,47
19	2	2	2	0,44
20	2	3	2	0,43
21	3	2	1	0,47
22	1	4	4	0,39
23	1	3	4	0,42
24	4	1	4	0,28
25	2	2	1	0,45
26	4	2	2	0,44
27	4	3	3	0,53
28	3	2	2	0,47
29	2	1	4	0,40
30	1	2	1	0,44
31	1	3	2	0,40
32	3	1	3	0,40

Tablo 3.4. Devamı

DENEME	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
33	4	1	1	0,58
34	2	1	2	0,42
35	1	2	3	0,41
36	3	2	3	0,44
37	2	4	2	0,39
38	1	2	2	0,43
39	2	3	4	0,39
40	3	1	2	0,39
41	4	2	1	0,37
42	1	2	4	0,39
43	3	4	2	0,38
44	4	4	4	0,32
45	1	4	2	0,41
46	4	4	2	0,38
47	2	4	1	0,42
48	4	4	1	0,51
49	3	3	4	0,41
50	3	4	1	0,41
51	3	4	4	0,34
52	4	2	4	0,48
53	2	3	3	0,41
54	3	1	4	0,41
55	1	1	2	0,45
56	2	4	3	0,36
57	1	1	3	0,45
58	1	1	4	0,41
59	1	4	3	0,41
60	1	1	1	0,46
61	4	1	3	0,30
62	4	2	3	0,43
63	3	3	1	0,46
64	1	3	1	0,42

Yakın Uygunluk Eşik Değeri Metodunun 100.000 fonksiyon hesaplaması için yapılan denemede ortalama uygunluk değerinin yine **Deneme 33'** te en uygun değeri verdiği ve uygunluk oranı ortalamasının arttığı gözlemlenmektedir (bkz. Tablo 3.4), yani çaprazlama oranının yüksek, mutasyon oranının düşük ve popülasyon büyüklüğünün düşük olarak belirlenmesinin uygun olduğu saptanmaktadır.

Tablo 3.5. Yakın uygunluk eşik değeri metodunun 200.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri

DENEME	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
1	3	3	3	0,48
2	1	3	3	0,45
3	2	1	3	0,49
4	2	1	1	0,46
5	2	4	4	0,42
6	4	3	4	0,49
7	4	3	2	0,48
8	4	4	3	0,52
9	1	4	1	0,49
10	3	2	4	0,50
11	2	2	4	0,44
12	2	2	3	0,49
13	3	3	2	0,52
14	2	3	1	0,47
15	4	1	2	0,50
16	3	4	3	0,46
17	3	1	1	0,37
18	4	3	1	0,49
19	2	2	2	0,50
20	2	3	2	0,49
21	3	2	1	0,52
22	1	4	4	0,45
23	1	3	4	0,45
24	4	1	4	0,33
25	2	2	1	0,50
26	4	2	2	0,47
27	4	3	3	0,45
28	3	2	2	0,52
29	2	1	4	0,48
30	1	2	1	0,50
31	1	3	2	0,47
32	3	1	3	0,44

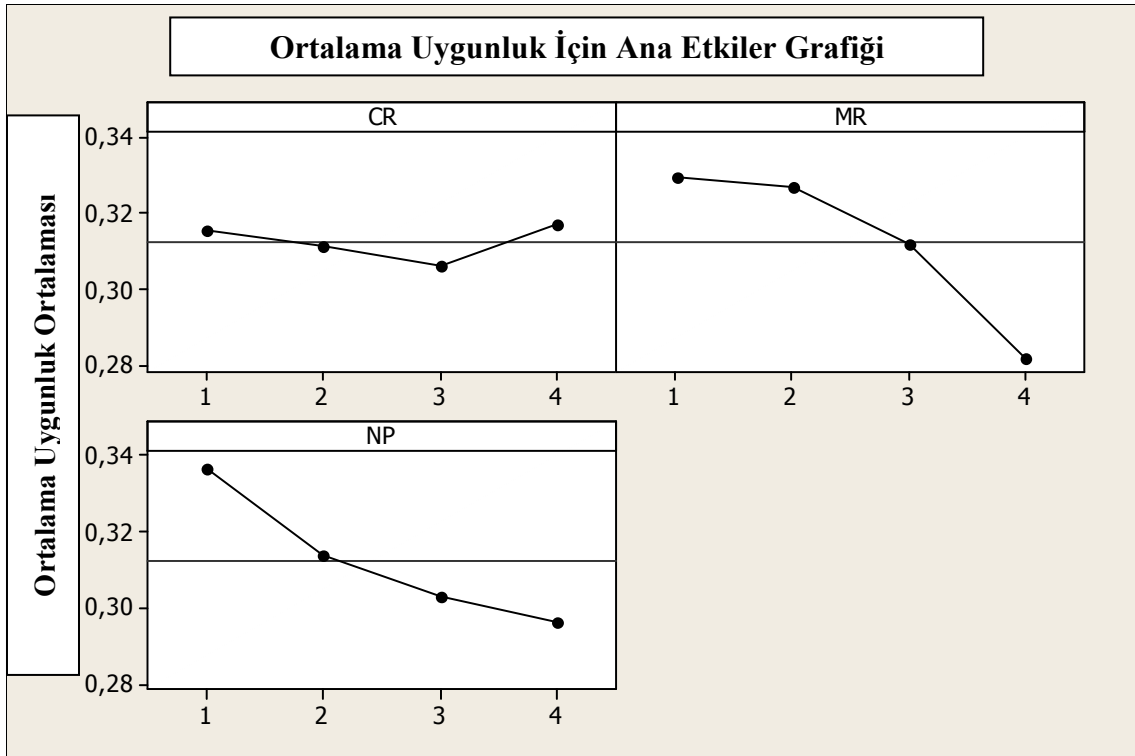
Tablo 3.5. Devamı

DENEME	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
33	4	1	1	0,59
34	2	1	2	0,47
35	1	2	3	0,47
36	3	2	3	0,54
37	2	4	2	0,43
38	1	2	2	0,49
39	2	3	4	0,44
40	3	1	2	0,45
41	4	2	1	0,37
42	1	2	4	0,45
43	3	4	2	0,46
44	4	4	4	0,37
45	1	4	2	0,45
46	4	4	2	0,54
47	2	4	1	0,48
48	4	4	1	0,47
49	3	3	4	0,46
50	3	4	1	0,48
51	3	4	4	0,44
52	4	2	4	0,53
53	2	3	3	0,46
54	3	1	4	0,47
55	1	1	2	0,52
56	2	4	3	0,44
57	1	1	3	0,50
58	1	1	4	0,47
59	1	4	3	0,46
60	1	1	1	0,51
61	4	1	3	0,39
62	4	2	3	0,50
63	3	3	1	0,51
64	1	3	1	0,47

Yakın Uygunluk Eşik Değeri Metodunun 200.000 fonksiyon hesaplaması için yapılan denemede ortalama uygunluk değerinin yine **Deneme 33'** te en uygun değeri verdiği ve uygunluk oranı ortalamasının arttığı gözlemlenmektedir (bkz. Tablo 3.5), yani çaprazlama oranının yüksek, mutasyon oranının düşük ve popülasyon büyüklüğünün düşük olarak belirlenmesinin uygun olduğu saptanmaktadır.

Elde edilen sonuçlara göre, ana etkiler (main effects plot) grafiđi ve faktörler arasında etkileşim olup olmadığını kontrol etmek için etkileşim grafikleri minitab 14 programı ile çizilmiştir ve onların da sonuçları değerlendirilmiştir.

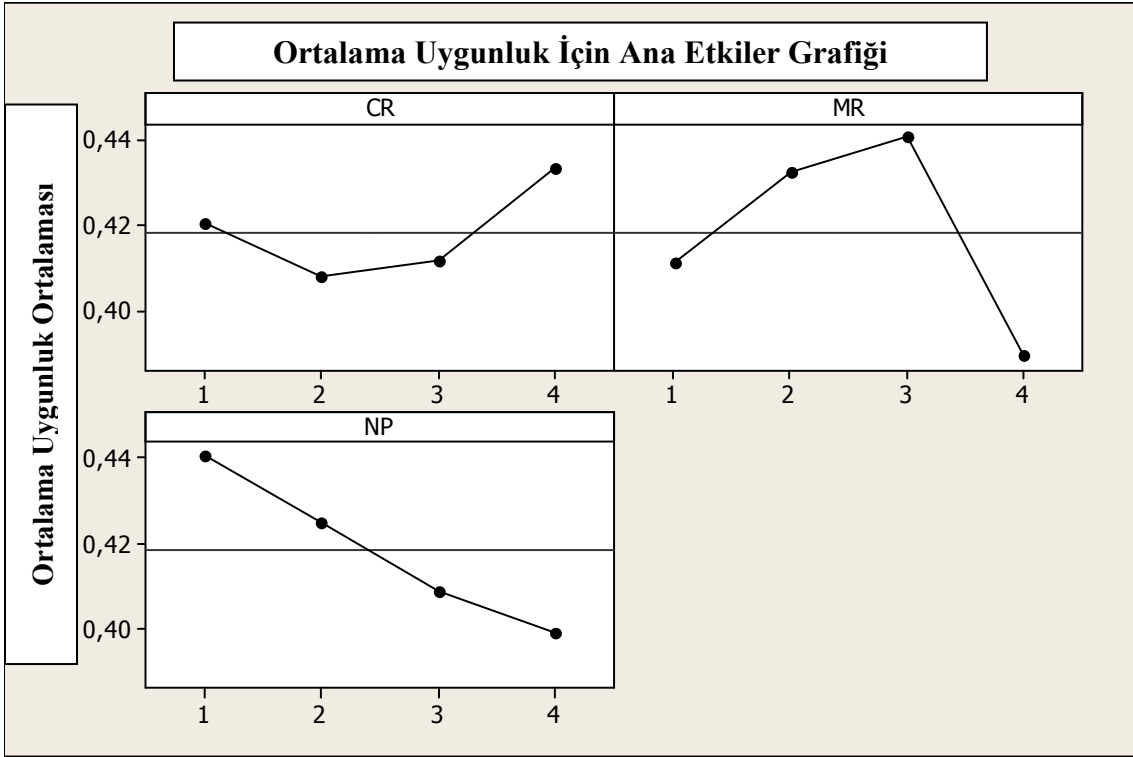
Yakın Uygunluk Eşik Deđeri Metodu için yapılan deney tasarımı incelenecek olursa;



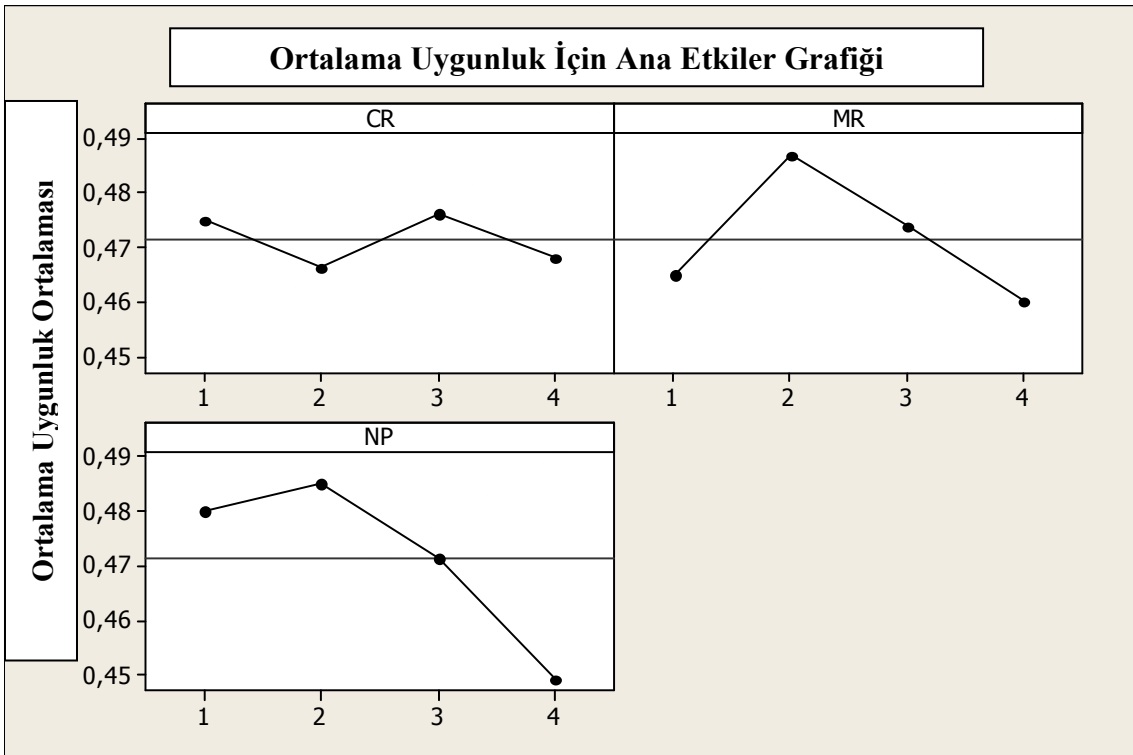
Şekil 3.1. Yakın uygunluk eşik değeri metodunun 20.000 fonksiyon hesaplaması için ana etkiler grafiđi

Popülasyon büyüklüğü faktörünün düşük, mutasyon oranı faktörünün düşük, çaprazlama faktörünün ise yüksek seçilmesi gerektiđi ana etkiler grafiđinden de okunabilmektedir.

Şekil 3.2.'ye bakıldığında fonksiyon hesaplama sayısı arttıkça durumun genel olarak deđişmediđi ancak belli deđer noktalarında ilgili parametrelerin en uygun performansı sergilediđi de gözlemlenmektedir. Yine Popülasyon büyüklüğü faktörünün düşük, mutasyon oranı faktörünün düşük ya da orta deđerde, çaprazlama faktörünün ise yüksek seçilmesi gerektiđi ana etkiler grafiđinden de okunabilmektedir.



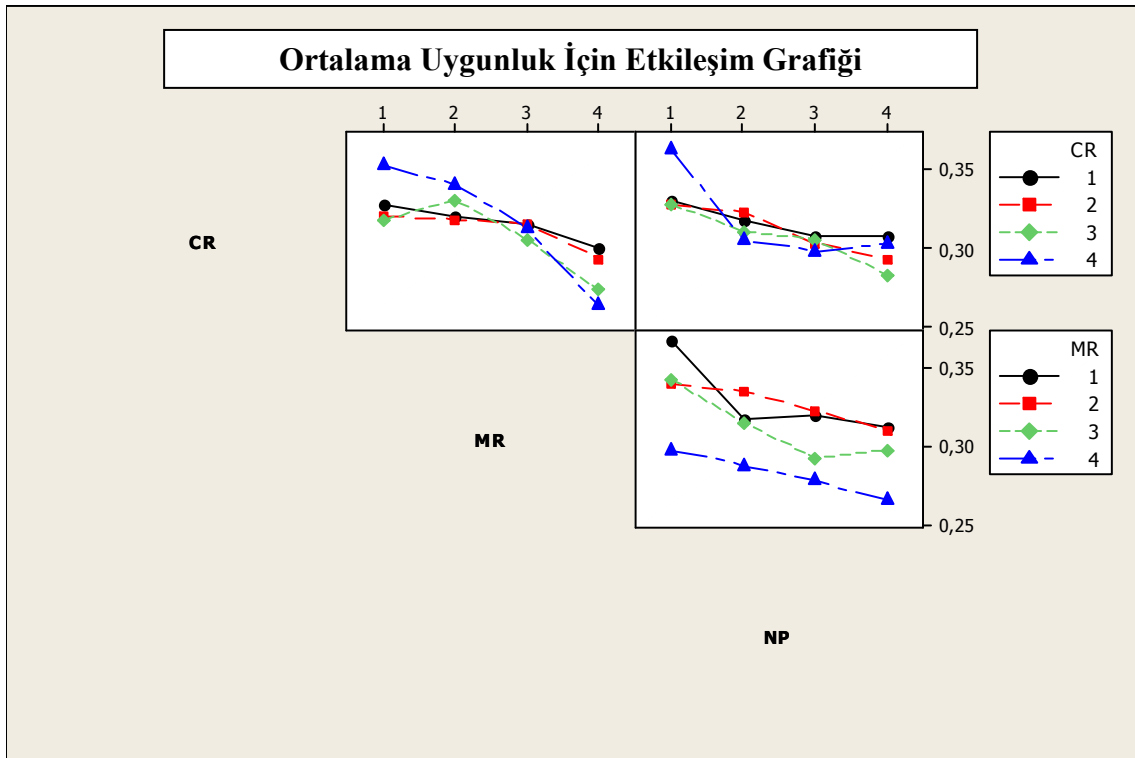
Şekil 3.2. Yakın uygunluk eşik değeri metodunun 100.000 fonksiyon hesaplaması için ana etkiler grafiği



Şekil 3.3. Yakın uygunluk eşik değeri metodunun 200.000 fonksiyon hesaplaması için ana etkiler grafiği

Şekil 3.3.'e bakıldığında fonksiyon hesaplama sayısı arttıkça uygunluk ortalamalarının yükseldiği ve durumun genel olarak değişmediği ancak belli değer noktalarında ilgili parametrelerin en uygun performansı sergilediği de gözlemlenmektedir. Yine Popülasyon büyüklüğü faktörünün düşük, mutasyon oranı faktörünün düşük ya da düşüğe yakın, çaprazlama faktörünün ise yükseğe yakın seçilmesi gerektiği ana etkiler grafiğinden de okunabilmektedir.

Faktörler arasında etkileşim olup olmadığını belirlemek için 20.000-100.000 ve 200.000 fonksiyon hesaplamalarında NFT için faktörlerin etkileşim grafikleri çizilmiştir (şekil 3.4 - şekil 3.6).



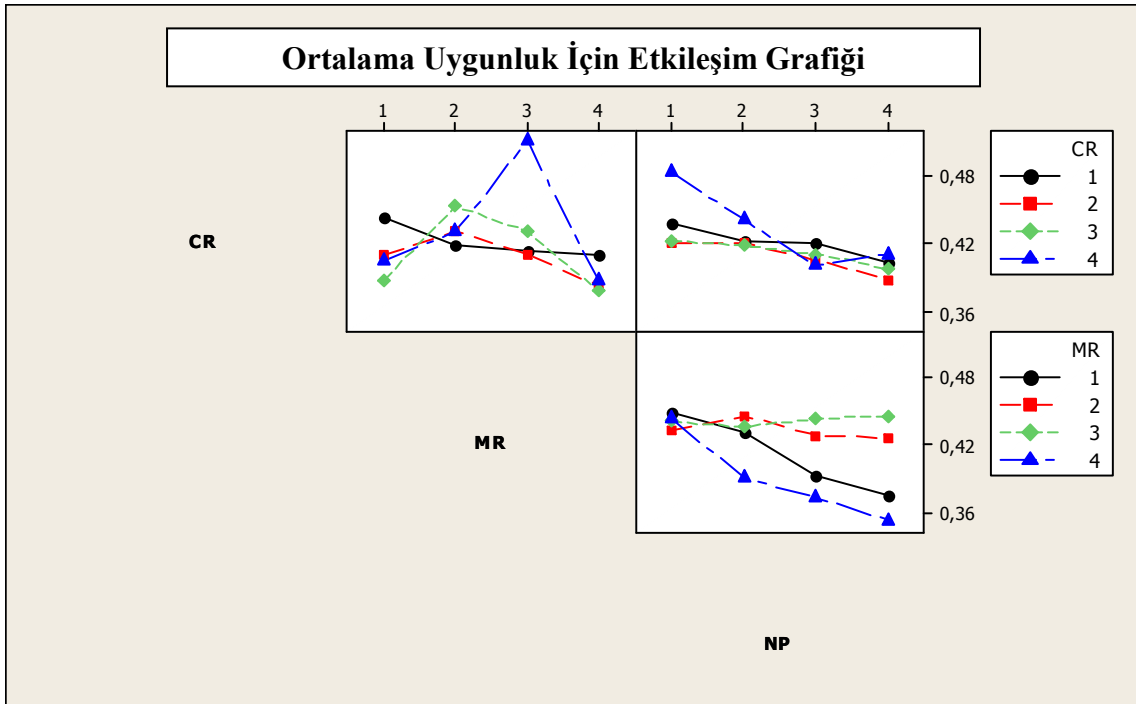
Şekil 3.4. Yakın uygunluk eşik değeri metodunun 20.000 fonksiyon hesaplaması için etkileşim grafiği

Etkileşim grafiğine bakılarak faktörler arasında etkileşimin fazla olduğunun gözlemlenebilmesi ile birlikte CR-MR etkileşiminde MR nin değerinin artmasıyla birlikte CR nin farklı değerlerine rağmen ortalama uygunluk oranının azaldığı; MR-NP etkileşiminde NP nin değerinin artmasıyla birlikte MR nin farklı değerlerine rağmen ortalama uygunluk oranının azaldığı; CR-NP etkileşiminde NP nin değerinin

artmasıyla birlikte CR nin farklı değerlerine rağmen yine ortalama uygunluk oranının azaldığı gözlemlenebilmektedir. Bu durumda ana etkiler grafiğine bakılarak uygun deney tasarımı, Tablo 3.6 da gösterildiği gibi Deneme 33 te kullanılan parametreler şeklinde belirlenebilmektedir.

Tablo 3.6. Yakın uygunluk eşik değeri metodunun 20.000 fonksiyon hesaplaması için en uygun parametre seti

CR Seviyesi	MR Seviyesi	NP Seviyesi
4	1	1



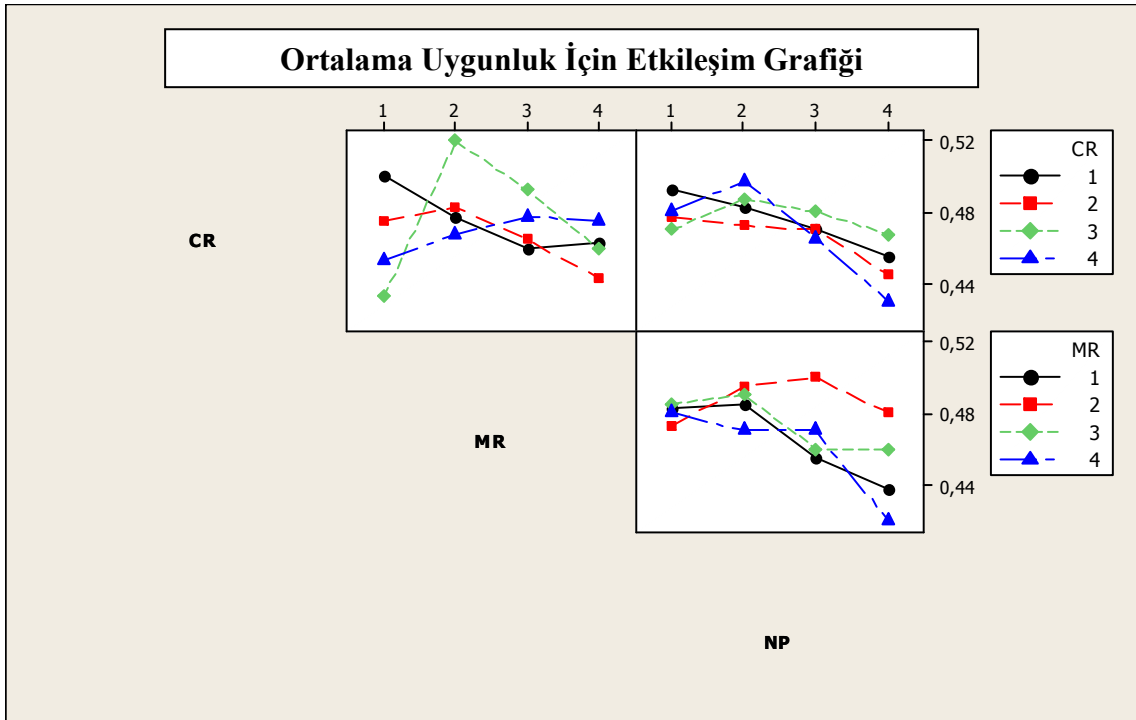
Şekil 3.5. Yakın uygunluk eşik değeri metodunun 100.000 fonksiyon hesaplaması için etkileşim grafiği

Etkileşim grafiğine bakılarak faktörler arasında etkileşimin fazla olduğunun gözlemlenebilmesi ile birlikte CR-MR etkileşiminde MR nin değerinin ortalama değerine doğru artmasıyla birlikte CR nin farklı değerlerinde en yüksek ortalama uygunluk oranına ulaşılabildiği; MR-NP etkileşiminde NP nin değerinin artmasıyla birlikte MR nin farklı değerlerine rağmen ortalama uygunluk oranının azaldığı; CR-NP etkileşiminde NP nin değerinin artmasıyla birlikte CR nin farklı değerlerine

rağmen yine ortalama uygunluk oranının azaldığı gözlemlenebilmektedir. Bu durumda ana etkiler grafiğine bakarak uygun deney tasarımı, Tablo 3.7 de gösterildiği gibi Deneme 33 te kullanılan parametreler şeklinde belirlenebilmektedir.

Tablo 3.7. Yakın uygunluk eşik değeri metodunun 100.000 fonksiyon hesaplaması için en uygun parametre seti

CR Seviyesi	MR Seviyesi	NP Seviyesi
4	1	1



Şekil 3.6. Yakın uygunluk eşik değeri metodunun 200.000 fonksiyon hesaplaması için etkileşim grafiği

Etkileşim grafiğine bakılarak faktörler arasında etkileşimin fazla olduğunun gözlemlenebilmesi ile birlikte CR-MR etkileşiminde MR nin değerinin ortalama değerine doğru artmasıyla birlikte CR nin farklı değerlerinde en yüksek ortalama uygunluk oranına ulaşılabildiği ancak MR nin daha da artmasının bu oranı düşürdüğü ; MR-NP etkileşiminde NP nin değerinin ortalama değerine doğru artmasıyla birlikte MR nin farklı değerlerinde en yüksek ortalama uygunluk oranına ulaşılabildiği ancak NP nin daha da artmasının bu oranı düşürdüğü; CR-NP etkileşiminde NP nin

değerinin artmasıyla birlikte CR nin farklı değerlerine rağmen yine ortalama uygunluk oranının azaldığı gözlemlenebilmektedir. Bu durumda ana etkiler grafiğine bakarak uygun deney tasarımı, Tablo 3.8 de gösterildiği gibi Deneme33 te kullanılan parametreler şeklinde belirlenebilmektedir.

Tablo 3.8. Yakın uygunluk eşik değeri metodunun 200.000 fonksiyon hesaplaması için en uygun parametre seti

CR Seviyesi	MR Seviyesi	NP Seviyesi
4	1	1

Tablo 3.9. Epsilon constraint metodunun 20.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri

DNM	Tc SVY	Teta SVY	cp SVY	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
1	1	1	1	1	1	1	0,36
2	1	-1	-1	1	-1	-1	0,52
3	1	1	-1	-1	-1	-1	0,54
4	1	-1	-1	1	-1	1	0,35
5	-1	1	-1	1	1	1	0,37
6	1	-1	-1	-1	1	-1	0,52
7	-1	-1	1	-1	1	1	0,38
8	1	1	1	1	-1	-1	0,51
9	-1	-1	1	1	-1	1	0,38
10	-1	1	-1	1	-1	-1	0,50
11	-1	-1	-1	1	1	-1	0,51
12	-1	-1	1	1	1	1	0,38
13	-1	-1	1	-1	-1	1	0,38
14	-1	-1	1	1	1	-1	0,46
15	-1	-1	-1	1	-1	1	0,37
16	1	-1	1	1	1	1	0,38
17	-1	1	-1	1	1	-1	0,50
18	-1	1	-1	-1	1	1	0,37
19	1	1	1	-1	-1	-1	0,51
20	1	1	-1	1	-1	1	0,35
21	-1	-1	1	-1	-1	-1	0,46
22	1	1	-1	-1	-1	1	0,35
23	1	-1	1	-1	-1	1	0,38
24	-1	1	-1	-1	-1	-1	0,50
25	-1	-1	1	-1	1	-1	0,46

Tablo 3.9. Devamı

DNM	Tc SVY	Teta SVY	cp SVY	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
26	1	-1	-1	1	1	-1	0,52
27	1	-1	-1	1	1	1	0,35
28	1	1	-1	1	1	1	0,35
29	-1	1	1	-1	1	-1	0,46
30	1	-1	1	1	-1	1	0,38
31	-1	1	1	1	1	-1	0,46
32	1	1	1	1	1	-1	0,51
33	1	1	1	-1	1	-1	0,51
34	-1	1	-1	-1	1	-1	0,50
35	-1	-1	-1	-1	1	-1	0,51
36	-1	-1	-1	-1	-1	1	0,37
37	1	-1	1	-1	-1	-1	0,48
38	1	1	-1	1	1	-1	0,54
39	1	1	1	-1	-1	1	0,36
40	-1	-1	-1	-1	-1	-1	0,51
41	1	1	1	1	-1	1	0,36
42	-1	1	1	-1	1	1	0,36
43	-1	1	1	1	-1	1	0,36
44	1	-1	1	1	1	-1	0,48
45	1	1	-1	-1	1	1	0,35
46	1	1	1	-1	1	1	0,36
47	1	1	-1	1	-1	-1	0,54
48	-1	-1	1	1	-1	-1	0,46
49	1	-1	1	1	-1	-1	0,48
50	1	-1	-1	-1	-1	-1	0,52
51	-1	1	1	1	-1	-1	0,46
52	1	1	-1	-1	1	-1	0,54
53	-1	1	1	-1	-1	1	0,36
54	-1	-1	-1	-1	1	1	0,37
55	-1	1	1	-1	-1	-1	0,46
56	-1	1	-1	-1	-1	1	0,37
57	-1	1	1	1	1	1	0,36
58	-1	-1	-1	1	-1	-1	0,51
59	-1	1	-1	1	-1	1	0,37
60	1	-1	1	-1	1	-1	0,48
61	-1	-1	-1	1	1	1	0,37
62	1	-1	-1	-1	1	1	0,35
63	1	-1	1	-1	1	1	0,38
64	1	-1	-1	-1	-1	1	0,35

Epsilon Constraint Metodunun 20.000 fonksiyon hesaplaması için yapılan denemede ortalama uygunluk değerinin Deneme 3, Deneme 38, Deneme 47 ve Deneme 52 de aynı en uygun değeri verdiği gözlemlenmektedir (bkz. Tablo 3.9), yani Tc nin büyük, Tetanın büyük, cp nin küçük ve popülasyon büyüklüğünün düşük olarak belirlenmesinin uygun olduğu saptanmaktadır. Ayrıca çaprazlama ve mutasyon oranının düşük ya da yüksek seçilmesinin hiçbir etki etmediği de gözlemlenmektedir. (DNM : Deneme Sayısı ve SVY: Seviye)

Tablo 3.10. Epsilon constraint metodunun 100.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri

DNM	Tc SVY	Teta SVY	cp SVY	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
1	1	1	1	1	1	1	0,66
2	1	-1	-1	1	-1	-1	0,86
3	1	1	-1	-1	-1	-1	0,87
4	1	-1	-1	1	-1	1	0,68
5	-1	1	-1	1	1	1	0,67
6	1	-1	-1	-1	1	-1	0,86
7	-1	-1	1	-1	1	1	0,60
8	1	1	1	1	-1	-1	0,87
9	-1	-1	1	1	-1	1	0,63
10	-1	1	-1	1	-1	-1	0,88
11	-1	-1	-1	1	1	-1	0,87
12	-1	-1	1	1	1	1	0,63
13	-1	-1	1	-1	-1	1	0,63
14	-1	-1	1	1	1	-1	0,86
15	-1	-1	-1	1	-1	1	0,65
16	1	-1	1	1	1	1	0,66
17	-1	1	-1	1	1	-1	0,88
18	-1	1	-1	-1	1	1	0,67
19	1	1	1	-1	-1	-1	0,87
20	1	1	-1	1	-1	1	0,74
21	-1	-1	1	-1	-1	-1	0,86
22	1	1	-1	-1	-1	1	0,74
23	1	-1	1	-1	-1	1	0,66
24	-1	1	-1	-1	-1	-1	0,88
25	-1	-1	1	-1	1	-1	0,86
26	1	-1	-1	1	1	-1	0,86
27	1	-1	-1	1	1	1	0,68
28	1	1	-1	1	1	1	0,74

Tablo 3.10. Devamı

DNM	Tc SVY	Teta SVY	cp SVY	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
29	-1	1	1	-1	1	-1	0,87
30	1	-1	1	1	-1	1	0,66
31	-1	1	1	1	1	-1	0,87
32	1	1	1	1	1	-1	0,87
33	1	1	1	-1	1	-1	0,87
34	-1	1	-1	-1	1	-1	0,88
35	-1	-1	-1	-1	1	-1	0,87
36	-1	-1	-1	-1	-1	1	0,65
37	1	-1	1	-1	-1	-1	0,84
38	1	1	-1	1	1	-1	0,87
39	1	1	1	-1	-1	1	0,66
40	-1	-1	-1	-1	-1	-1	0,87
41	1	1	1	1	-1	1	0,66
42	-1	1	1	-1	1	1	0,64
43	-1	1	1	1	-1	1	0,64
44	1	-1	1	1	1	-1	0,84
45	1	1	-1	-1	1	1	0,74
46	1	1	1	-1	1	1	0,66
47	1	1	-1	1	-1	-1	0,87
48	-1	-1	1	1	-1	-1	0,86
49	1	-1	1	1	-1	-1	0,84
50	1	-1	-1	-1	-1	-1	0,86
51	-1	1	1	1	-1	-1	0,87
52	1	1	-1	-1	1	-1	0,87
53	-1	1	1	-1	-1	1	0,64
54	-1	-1	-1	-1	1	1	0,65
55	-1	1	1	-1	-1	-1	0,87
56	-1	1	-1	-1	-1	1	0,67
57	-1	1	1	1	1	1	0,64
58	-1	-1	-1	1	-1	-1	0,87
59	-1	1	-1	1	-1	1	0,67
60	1	-1	1	-1	1	-1	0,84
61	-1	-1	-1	1	1	1	0,65
62	1	-1	-1	-1	1	1	0,68
63	1	-1	1	-1	1	1	0,66
64	1	-1	-1	-1	-1	1	0,68

Epsilon Constraint Metodunun 100.000 fonksiyon hesaplaması için yapılan denemede ortalama uygunluk değerinin Deneme 10, Deneme 17, Deneme 24 ve Deneme 34 de aynı en uygun değeri verdiği gözlemlenmektedir (bkz. Tablo 3.10), yani Tc nin küçük, Tetanın büyük, cp nin küçük ve popülasyon büyüklüğünün düşük olarak belirlenmesinin uygun olduğu saptanmaktadır. Ayrıca çaprazlama ve mutasyon oranının düşük ya da yüksek seçilmesinin hiçbir etki etmediği de gözlemlenmektedir. (DNM : Deneme Sayısı ve SVY: Seviye)

Tablo 3.11. Epsilon constraint metodunun 200.000 fonksiyon hesaplaması için belirlenen faktörlerin tasarım seçenekleri

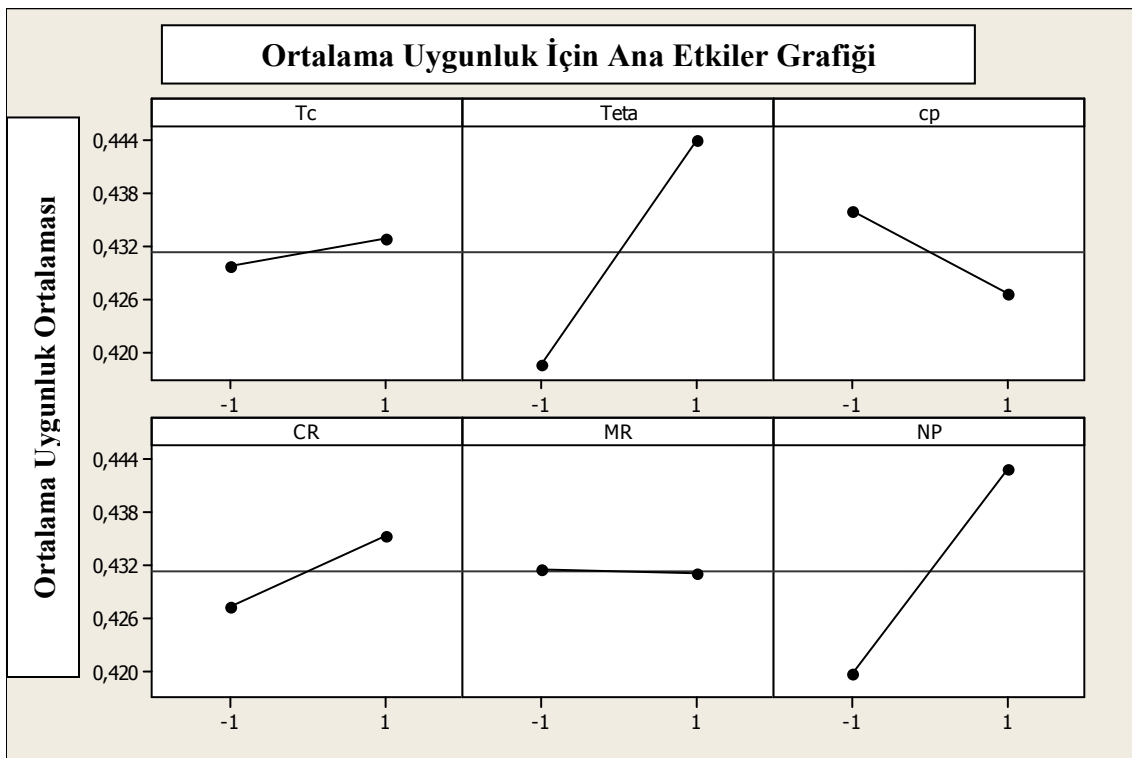
DNM	Tc SVY	Teta SVY	cp SVY	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
1	1	1	1	1	1	1	0,83
2	1	-1	-1	1	-1	-1	0,87
3	1	1	-1	-1	-1	-1	0,87
4	1	-1	-1	1	-1	1	0,84
5	-1	1	-1	1	1	1	0,83
6	1	-1	-1	-1	1	-1	0,87
7	-1	-1	1	-1	1	1	0,77
8	1	1	1	1	-1	-1	0,86
9	-1	-1	1	1	-1	1	0,77
10	-1	1	-1	1	-1	-1	0,88
11	-1	-1	-1	1	1	-1	0,88
12	-1	-1	1	1	1	1	0,77
13	-1	-1	1	-1	-1	1	0,77
14	-1	-1	1	1	1	-1	0,87
15	-1	-1	-1	1	-1	1	0,77
16	1	-1	1	1	1	1	0,76
17	-1	1	-1	1	1	-1	0,88
18	-1	1	-1	-1	1	1	0,83
19	1	1	1	-1	-1	-1	0,86
20	1	1	-1	1	-1	1	0,88
21	-1	-1	1	-1	-1	-1	0,87
22	1	1	-1	-1	-1	1	0,88
23	1	-1	1	-1	-1	1	0,76
24	-1	1	-1	-1	-1	-1	0,88
25	-1	-1	1	-1	1	-1	0,87
26	1	-1	-1	1	1	-1	0,87
27	1	-1	-1	1	1	1	0,84
28	1	1	-1	1	1	1	0,88

Tablo 3.11. Devamı

DNM	Tc SVY	Teta SVY	cp SVY	ÇAPRAZLAMA ORANI SEVİYESİ	MUTASYON ORANI SEVİYESİ	POPÜLASYON BÜYÜKLÜĞÜ SEVİYESİ	ORTALAMA UYGUNLUK DEĞERİ
29	-1	1	1	-1	1	-1	0,87
30	1	-1	1	1	-1	1	0,76
31	-1	1	1	1	1	-1	0,87
32	1	1	1	1	1	-1	0,86
33	1	1	1	-1	1	-1	0,86
34	-1	1	-1	-1	1	-1	0,88
35	-1	-1	-1	-1	1	-1	0,88
36	-1	-1	-1	-1	-1	1	0,77
37	1	-1	1	-1	-1	-1	0,86
38	1	1	-1	1	1	-1	0,87
39	1	1	1	-1	-1	1	0,83
40	-1	-1	-1	-1	-1	-1	0,88
41	1	1	1	1	-1	1	0,83
42	-1	1	1	-1	1	1	0,77
43	-1	1	1	1	-1	1	0,77
44	1	-1	1	1	1	-1	0,86
45	1	1	-1	-1	1	1	0,88
46	1	1	1	-1	1	1	0,83
47	1	1	-1	1	-1	-1	0,87
48	-1	-1	1	1	-1	-1	0,87
49	1	-1	1	1	-1	-1	0,86
50	1	-1	-1	-1	-1	-1	0,87
51	-1	1	1	1	-1	-1	0,87
52	1	1	-1	-1	1	-1	0,87
53	-1	1	1	-1	-1	1	0,77
54	-1	-1	-1	-1	1	1	0,77
55	-1	1	1	-1	-1	-1	0,87
56	-1	1	-1	-1	-1	1	0,83
57	-1	1	1	1	1	1	0,77
58	-1	-1	-1	1	-1	-1	0,88
59	-1	1	-1	1	-1	1	0,83
60	1	-1	1	-1	1	-1	0,86
61	-1	-1	-1	1	1	1	0,77
62	1	-1	-1	-1	1	1	0,84
63	1	-1	1	-1	1	1	0,76
64	1	-1	-1	-1	-1	1	0,84

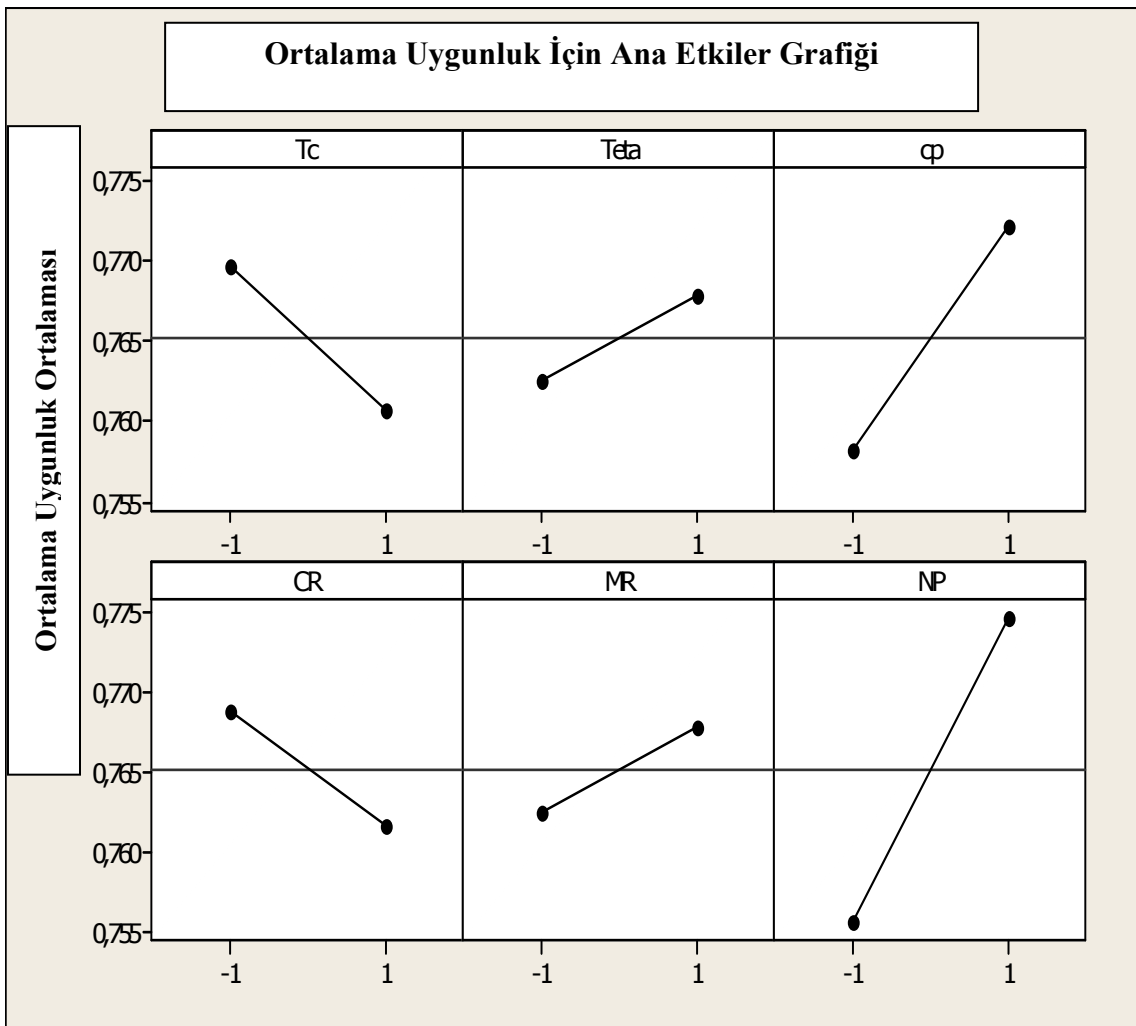
Epsilon Constraint Metodunun 200.000 fonksiyon hesaplaması için yapılan denemede ortalama uygunluk değerinin Deneme 20, Deneme 22, Deneme 28 ve Deneme 45' te alta ve üste yuvarlama durumları da göz önüne alındığında aynı en uygun değeri verdiği gözlemlenmektedir (bkz. Tablo 3.11), yani Tc, Teta, popülasyon büyüklüğü, çaprazlama ve mutasyon oranının düşük ya da yüksek seçilmesinin hiçbir etki etmediği de gözlemlenirken cp nin düşük olarak belirlenmesinin uygun olduğu saptanmaktadır. (DNM : Deneme Sayısı ve SVY: Seviye)

Epsilon Constraint Metodu için yapılan deney tasarımı inceleyecek olursak;



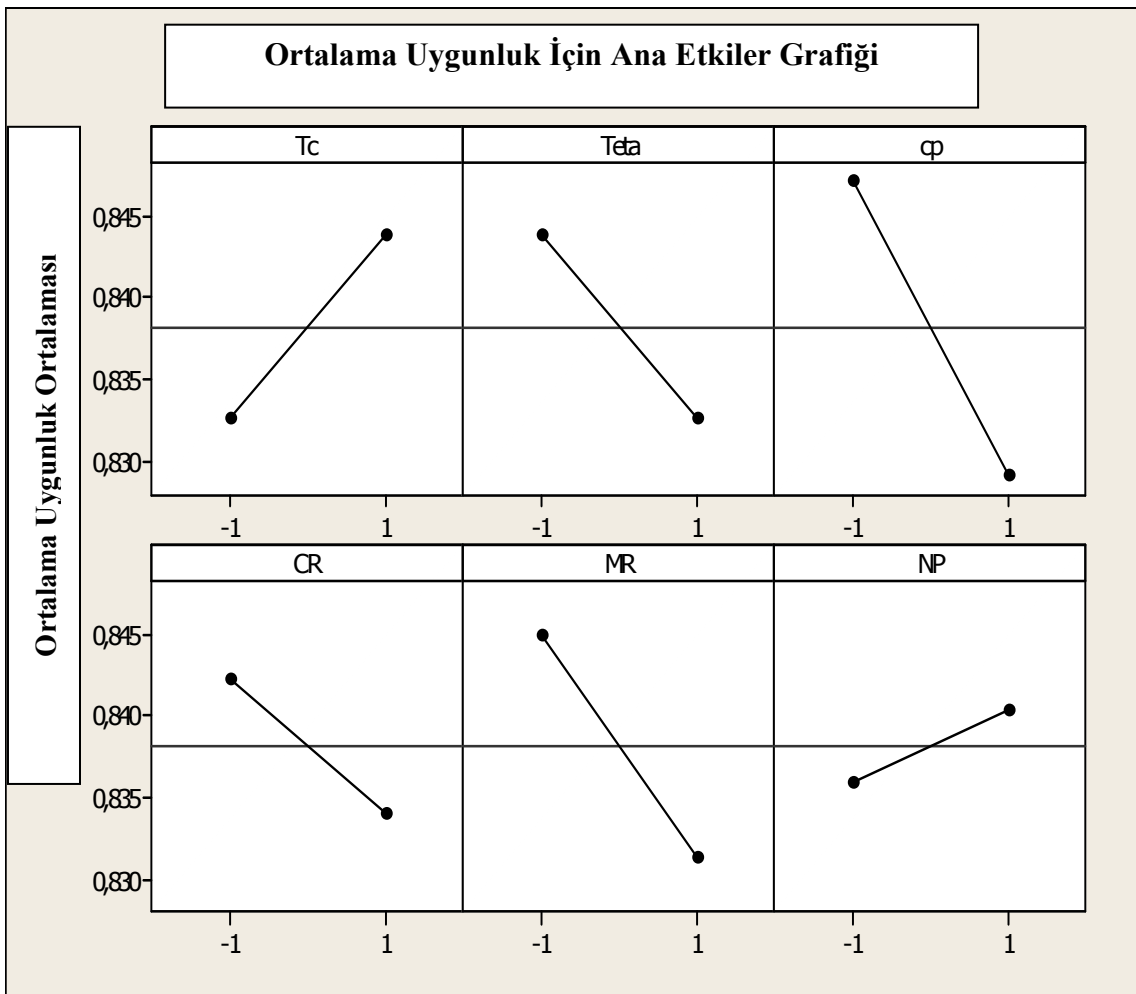
Şekil 3.7. Epsilon constraint metodunun 20.000 fonksiyon hesaplaması için ana etkiler grafiği

Tc faktörünün yüksek, Teta faktörünün yüksek, cp faktörünün düşük, Popülasyon büyüklüğü faktörünün yüksek, çaprazlama faktörünün yüksek seçilmesi gerektiği ve mutasyon oranı faktörünün ise yüksek ya da düşük seçilmesinin pek fazla etki etmediği ana etkiler grafiğinden de okunabilmektedir.



Şekil 3.8. Epsilon constraint metodunun 100.000 fonksiyon hesaplaması için ana etkiler grafiği

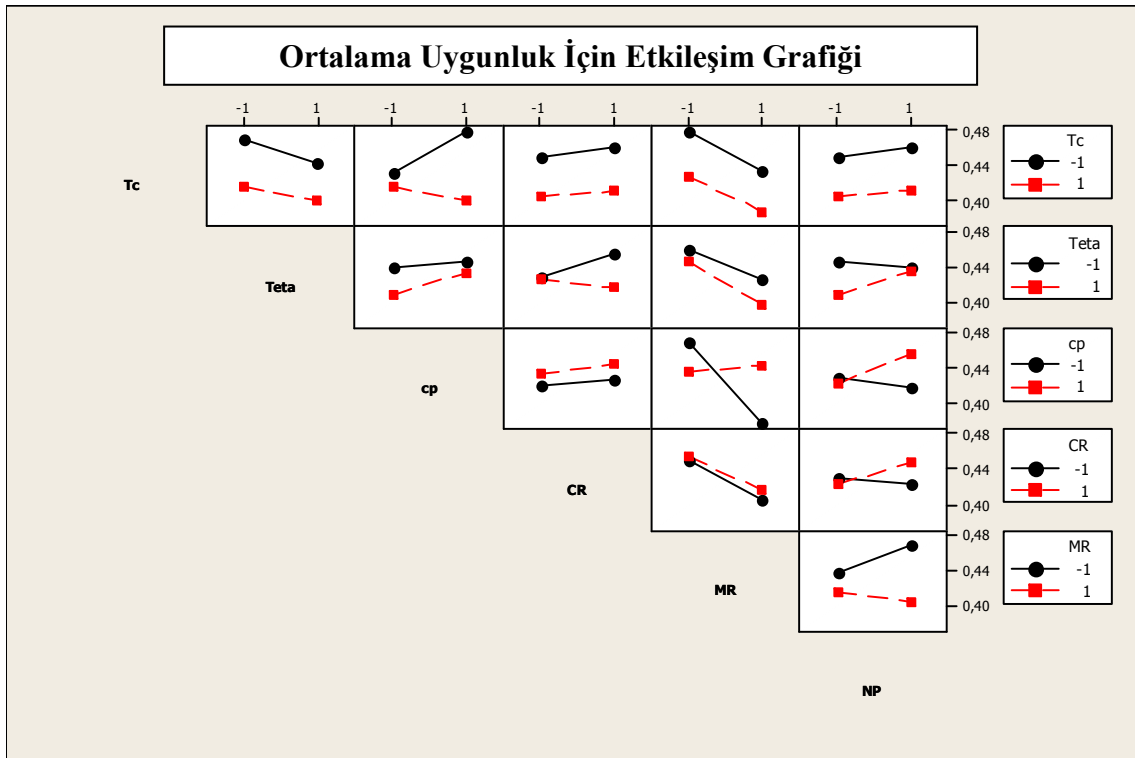
Fonksiyon hesaplama sayısı arttıkça Tc faktörünün düşük, Teta faktörünün yüksek, ϕ faktörünün yüksek, Popülasyon büyüklüğü faktörünün yüksek, çaprazlama faktörünün düşük seçilmesi gerektiği ve mutasyon oranı faktörünün ise 100.000 fonksiyon hesaplamasında etkisinin daha da arttığı ve yüksek seçilmesi gerektiği ana etkiler grafiğinden de okunabilmektedir.



Şekil 3.9. Epsilon constraint metodunun 200.000 fonksiyon hesaplaması için ana etkiler grafiği

Fonksiyon hesaplama sayısı arttıkça Tc faktörünün yüksek, Teta faktörünün düşük, φ faktörünün düşük, Popülasyon büyüklüğü faktörünün yüksek, çaprazlama faktörünün düşük seçilmesi gerektiği ve mutasyon oranı faktörünün ise 200.000 fonksiyon hesaplamasında etkisinin daha da fazla arttığı ve düşük seçilmesi gerektiği ana etkiler grafiğinden de okunmaktadır.

Faktörler arasında etkileşim olup olmadığını belirlemek için 20.000-100.000 ve 200.000 fonksiyon hesaplamasında EC için faktörlerin etkileşim grafiği çizilmiştir. (şekil 3.10 - şekil 3.12).

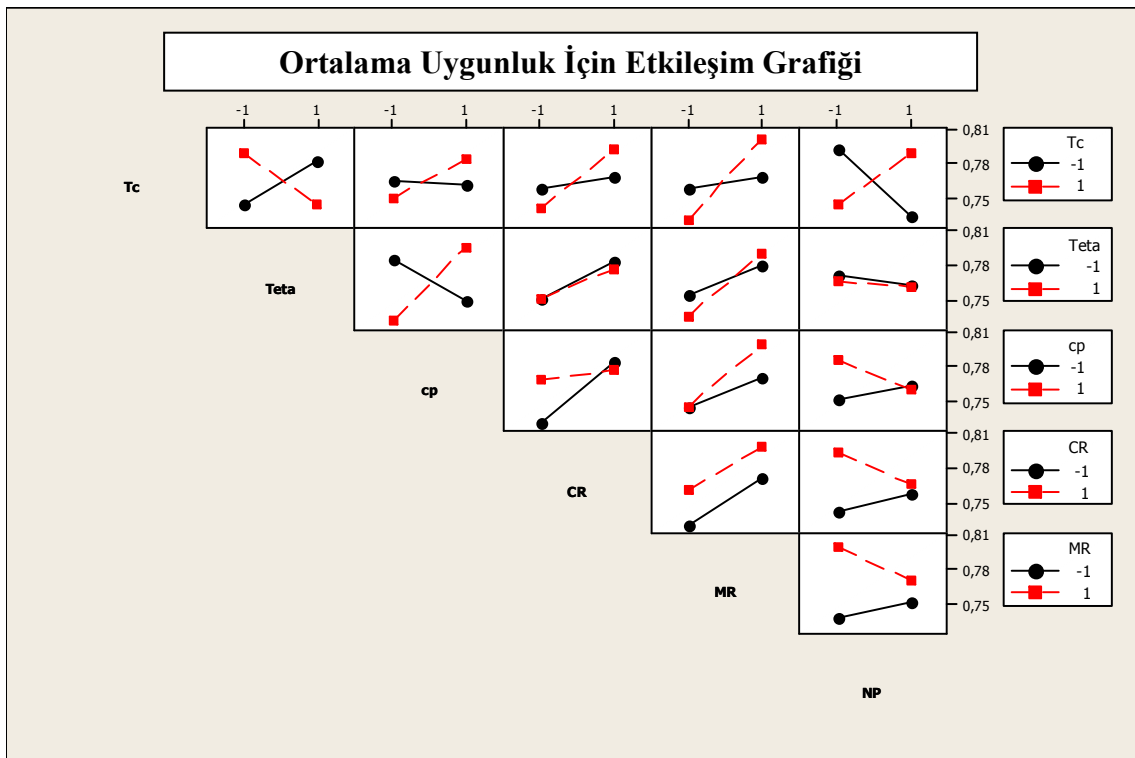


Şekil 3.10. Epsilon constraint metodunun 20.000 fonksiyon hesaplaması için etkileşim grafiği

Etkileşim grafiğine bakılarak faktörler arasında etkileşimin az olduğunun gözlemlenebilmesi ile birlikte Popülasyon Sayısı NP nin diğer faktörlerle etkileşiminin çok olduğu bunu yanında Teta-CR, cp-MR ve Tc-cp etkileşimleri de gözlemlenebilmektedir. Bu durumda ana etkiler grafiğine bakarak uygun deney tasarımı, Tablo 3.12 de gösterildiği gibi Deneme3, Deneme38, Deneme47 ve Deneme52 de kullanılan parametreler şeklinde belirlenebilmektedir.

Tablo 3.12. Epsilon constraint metodunun 20.000 fonksiyon hesaplaması için en uygun parametre seti

Tc Seviyesi	Teta Seviyesi	cp Seviyesi	CR Seviyesi	MR Seviyesi	NP Seviyesi
1	1	-1	-1	-1	-1
1	1	-1	1	1	-1
1	1	-1	1	-1	-1
1	1	-1	-1	1	-1

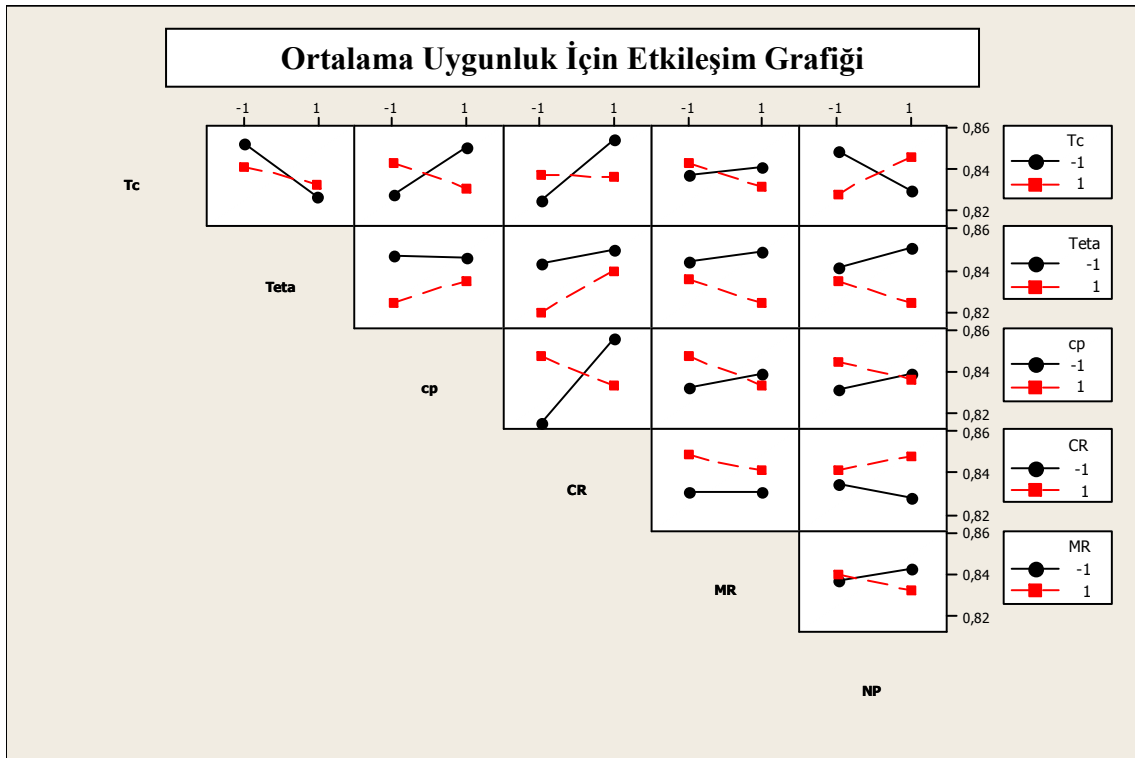


Şekil 3.11. Epsilon constraint metodunun 100.000 fonksiyon hesaplaması için etkileşim grafiği

Etkileşim grafiğine bakılarak faktörler arasında etkileşimin bir hayli arttığı gözlemlenebilmesi ile birlikte Popülasyon Sayısı NP ve Tc nin diğer faktörlerle etkileşiminin çok olduğu bunu yanında cp-MR ve Tc-cp etkileşimleri gözlemlenebilmektedir. Dikkat çekici olarak CR-MR etkileşiminin yok denecek kadar az olduğu gözlemlenebilmektedir. Bu durumda ana etkiler grafiğine bakarak uygun deney tasarımı, Tablo 3.13 de gösterildiği gibi Deneme10, Deneme17, Deneme24 ve Deneme34 ' te kullanılan parametreler şeklinde belirlenebilmektedir.

Tablo 3.13. Epsilon constraint metodunun 100.000 fonksiyon hesaplaması için en uygun parametre seti

Tc Seviyesi	Teta Seviyesi	cp Seviyesi	CR Seviyesi	MR Seviyesi	NP Seviyesi
-1	1	-1	1	-1	-1
-1	1	-1	1	1	-1
-1	1	-1	-1	-1	-1
-1	1	-1	-1	1	-1

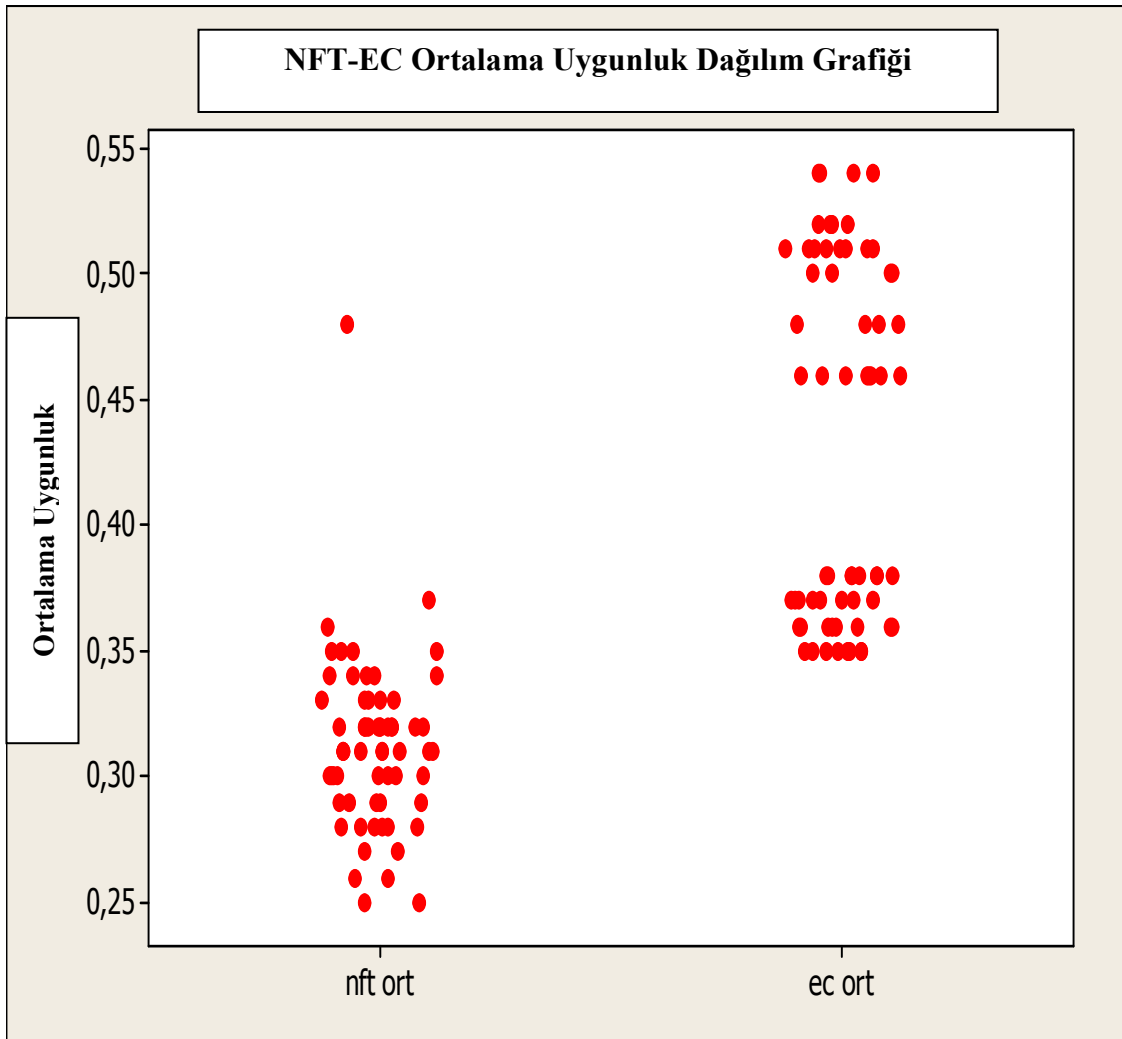


Şekil 3.12. Epsilon constraint metodunun 200.000 fonksiyon hesaplaması için etkileşim grafiği

Etkileşim grafiğine bakılarak faktörler arasında etkileşimin bir hayli arttığı gözlemlenebilmesi ile birlikte Popülasyon Sayısı NP, Tc ve cp nin diğer faktörlerle etkileşiminin çok olduğu bunu yanında cp-MR ve Tc-cp etkileşimleri gözlemlenebilmektedir. Yine CR-MR etkileşiminin yok denecek kadar az olduğu gözlemlenebilmektedir. Bu durumda ana etkiler grafiğine bakarak uygun deney tasarımı, Tablo 3.14 de gösterildiği gibi Deneme20, Deneme22, Deneme28 ve Deneme45 te kullanılan parametreler şeklinde belirlenebilmektedir.

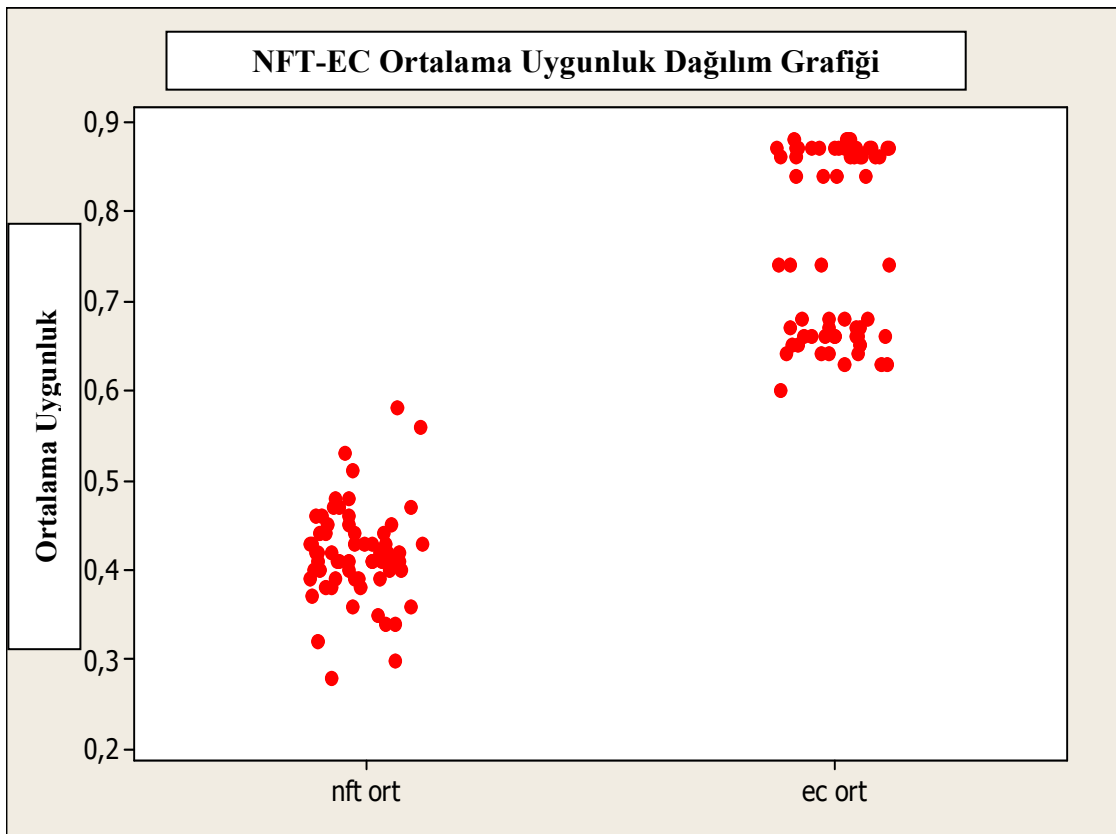
Tablo 3.14. Epsilon constraint metodunun 200.000 fonksiyon hesaplaması için en uygun parametre seti

Tc Seviyesi	Teta Seviyesi	cp Seviyesi	CR Seviyesi	MR Seviyesi	NP Seviyesi
1	1	-1	1	-1	1
1	1	-1	-1	-1	1
1	1	-1	1	1	1
1	1	-1	-1	1	1



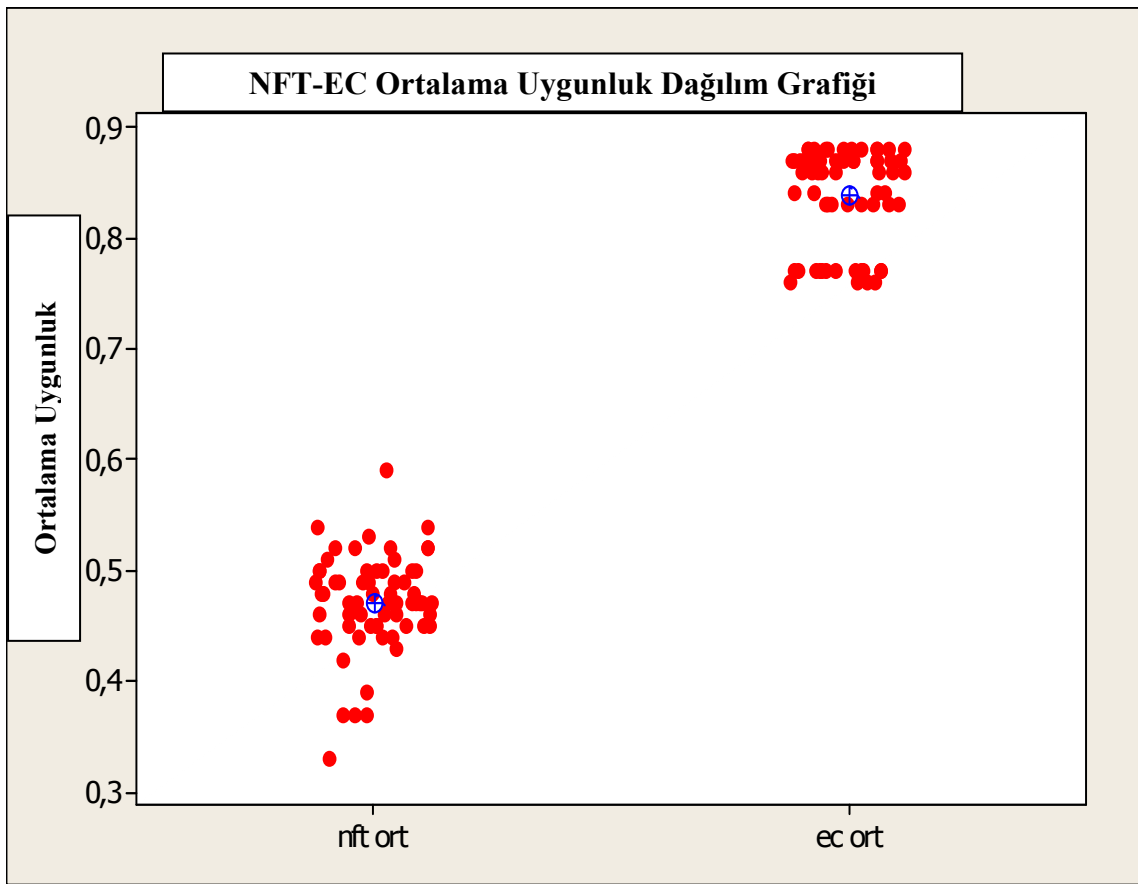
Şekil 3.13. 20.000 Fonksiyon hesaplamasına göre nft–ec ortalama uygunluk dağılım durumu

20.000 fonksiyon hesaplamasındaki sonuçların grafik üzerine dağılımına bakıldığında NFT metodunun kullanılması neticesinde ortaya çıkan uygunluk değerlerinin, EC metodunun kullanılması neticesinde ortaya çıkan uygunluk değerlerine göre düşük olduğu gözlemlenebilmektedir. NFT metodunda, sadece bir denemede nispeten maksimum ortalama uygunluk oranı elde edilirken, EC metodunda birçok sonuç NFT ile elde edilmiş maksimum değere çok rahat bir şekilde ulaşabilmekte ve hatta geçebilmektedir. Her iki metotta da ortalama uygunluk seviyelerinin optimum seviyeden uzak oldukları da gözlemlenebilmektedir.



Şekil 3.14. 100.000 Fonksiyon hesaplamasına göre nft–ec ortalama uygunluk dağılım durumu

100.000 fonksiyon hesaplamasındaki sonuçların grafik üzerine dağılımına bakıldığında, her iki metodunda ortalama uygunluk oranı seviyelerini arttırmış olmaları ile birlikte özellikle bu artışın EC metodunda bir hayli fazla olduğu açık şekilde görülebilmektedir. NFT metodunun kullanılması neticesinde ortaya çıkan uygunluk değerlerinin, EC metodunun kullanılması neticesinde ortaya çıkan uygunluk değerlerine göre yine çok düşük olduğu gözlemlenebilmektedir. EC metodunda ortalama uygunluk değeri artışı % 30 iken; NFT metodunda ortalama uygunluk değeri artışı % 10 oranında kalmaktadır. NFT metodu, daha çok ortalama uygunluk oranları düşük olan parametre setlerinin uygunluk oranlarını iyileştirirken, EC metodunda toplu bir şekilde parametre setlerinin uygunluk oranlarını iyileştirdiği gözlenebilmektedir.



Şekil 3.15. 200.000 Fonksiyon hesaplamasına göre nft–ec ortalama uygunluk dağılım durumu

Grafikten de anlaşılacağı üzere, 200.000 fonksiyon hesaplamasındaki sonuçların grafik üzerine dağılımına bakıldığında NFT metodunun kullanılması neticesinde ortaya çıkan uygunluk değerlerinin, EC metodunun kullanılması neticesinde ortaya çıkan uygunluk değerlerine göre çok düşük olduğu gözlemlenebilmektedir.. 100.000 fonksiyon hesaplamasına nazaran, EC metodunda, 200.000 fonksiyon hesaplamasında optimuma yakın onlarca parametre setinin olduğu gözlenebilmektedir. NFT metodunda tüm denemelerdeki maksimum uygunluk oranı 0,6' ya yakın çıkarken, EC Metodunda bu değer 0,9' a kadar çıkabilmektedir. Bu da iki metot arasındaki farkın belirgin bir şekilde ortaya koyulduğunu göstermektedir.

4. BÖLÜM

TARTIŞMA, SONUÇLAR ve ÖNERİLER

Bu tez çalışmasında Diferansiyel Gelişim Algoritması (DGA) ile birlikte kısıt yönetimi metotlarından olan NFT ve EC metotları sınırlamalara sahip olan 18 Benchmark test problemi üzerinde uygulanmıştır. Algoritma başlangıç popülasyonlarının rassal olarak üretilmesi ile çalışmaya başlamaktadır. Algoritma her bir problem için 25 replikasyon adedince 20.000-100.000 ve 200.000 fonksiyon hesaplama sayılarınca çalıştırılmıştır.

DGA ile birlikte kısıtlamaları aşmak için kullanılan metotların koşma sonuçları birbirleri ile karşılaştırılmıştır.

DGA, NFT ve EC metotları Visual C++ da programlanmıştır ve Intel(R) Core i7-2630QM 2.00 GHz 8 GHz belleği olan bilgisayarda çalıştırılmışlardır. DGA' nın, EC ve NFT metotlarının ayrı ayrı kullanılması ile ortaya çıkan performansları Tablo 3.15 Tablo 3.16 ve Tablo 3.17 de verilmiştir. Tabloların sırası ile NFT ve EC için 1. sütunu deneme numarasını, 2. sütunu problemler üzerinde bulunan en iyi çözümlerin ortalamalarını, 3. sütun her bir problem için bulunan maksimum uygunluk oranlarının ortalamalarını, 4. sütun her bir problem için bulunan minimum uygunluk oranlarının ortalamalarını, 5. sütun ilgili denemede bulunan en küçük standart sapmayı, 6. Sütun her bir problem için bulunan uygunluk oranlarının ortalamalarını göstermektedir.

Tablo 3.15. 20.000 Fonksiyon hesaplaması için NFT-EC performansı

	NFT					EC				
	avgbest	maxfeas	minfeas	stdbest	feasrate	avgbest	maxfeas	minfeas	stdbest	feasrate
1	2,28E+12	2,35E+13	-5,00E+01	1,24E-02	0,286667	6,04E+12	1,27E+12	-6,35E+01	1,93E-03	0,364444
2	3,96E+12	9,06E+12	-5,90E+01	5,21E-03	0,302222	2,17E+12	6,42E+10	-6,35E+01	2,72E-02	0,522222
3	1,96E+12	4,79E+12	-6,47E+01	1,39E-02	0,317778	5,32E+11	4,39E+11	-8,87E+02	2,72E-02	0,54
4	2,72E+12	2,07E+13	-7,85E+02	3,71E-02	0,32	5,45E+12	3,04E+12	-6,12E+01	1,93E-03	0,351111
5	6,70E+12	4,72E+13	-5,03E+01	2,15E-02	0,273333	7,88E+12	1,04E+13	-6,08E+01	1,93E-03	0,373333
6	3,56E+12	7,06E+13	-4,14E+01	2,97E-02	0,277778	2,17E+12	6,42E+10	-6,35E+01	2,72E-02	0,522222
7	1,45E+12	2,44E+13	-4,64E+01	1,39E-02	0,322222	7,91E+12	1,59E+13	-6,04E+01	1,93E-03	0,38
8	8,74E+12	2,46E+14	-9,61E+01	3,59E-02	0,253333	2,60E+12	1,25E+12	-6,35E+01	2,72E-02	0,508889
9	7,39E+12	1,22E+14	-1,42E+02	4,84E-03	0,322222	7,91E+12	1,59E+13	-6,04E+01	1,93E-03	0,38
10	1,36E+12	9,14E+12	-5,29E+01	1,67E-03	0,291111	3,63E+12	3,57E+11	-6,35E+01	2,72E-02	0,495556
11	2,53E+12	1,44E+13	-5,55E+01	3,74E-03	0,291111	4,16E+12	4,66E+12	-6,23E+01	2,72E-02	0,506667
12	2,87E+12	1,10E+14	-5,72E+01	4,25E-03	0,302222	7,91E+12	1,59E+13	-6,04E+01	1,93E-03	0,38
13	2,89E+12	2,00E+13	-5,16E+01	9,99E-03	0,306667	7,91E+12	1,59E+13	-6,04E+01	1,93E-03	0,38
14	3,58E+12	3,32E+13	-6,02E+01	2,13E-04	0,337778	7,50E+12	1,49E+13	-6,35E+01	2,72E-02	0,46
15	2,14E+12	4,89E+12	-6,17E+01	8,96E-02	0,3	6,79E+12	4,33E+12	-6,23E+01	1,93E-03	0,368889
16	3,10E+12	2,11E+13	-1,18E+02	2,31E-02	0,266667	7,55E+12	3,33E+13	-6,32E+01	1,93E-03	0,377778
17	3,06E+12	4,84E+12	-3,87E+02	7,56E-02	0,315556	3,63E+12	3,57E+11	-6,35E+01	2,72E-02	0,495556
18	1,20E+12	1,02E+13	-6,23E+01	3,25E-02	0,368889	7,88E+12	1,04E+13	-6,08E+01	1,93E-03	0,373333
19	2,04E+12	3,64E+12	-5,97E+01	5,99E-03	0,326667	2,60E+12	1,25E+12	-6,35E+01	2,72E-02	0,508889
20	2,11E+12	2,29E+13	-5,52E+01	2,78E-03	0,313333	4,54E+12	2,21E+13	-6,12E+01	1,93E-03	0,353333
21	2,92E+12	8,11E+12	-6,35E+01	4,00E-02	0,357778	7,50E+12	1,49E+13	-6,35E+01	2,72E-02	0,46
22	5,81E+12	2,48E+13	-5,49E+01	1,31E-02	0,284444	4,54E+12	2,21E+13	-6,12E+01	1,93E-03	0,353333
23	4,42E+12	5,48E+13	-5,68E+01	1,13E-02	0,322222	7,55E+12	3,33E+13	-6,32E+01	1,93E-03	0,377778
24	2,38E+12	6,66E+11	-7,20E+02	5,97E-02	0,322222	3,63E+12	3,57E+11	-6,35E+01	2,72E-02	0,495556
25	2,93E+12	9,03E+13	-6,35E+01	3,07E-02	0,348889	7,50E+12	1,49E+13	-6,35E+01	2,72E-02	0,46
26	1,56E+12	5,73E+11	-6,23E+01	3,58E-02	0,342222	2,17E+12	6,42E+10	-6,35E+01	2,72E-02	0,522222
27	1,13E+12	4,79E+12	-4,46E+01	2,06E-02	0,275556	5,45E+12	3,04E+12	-6,12E+01	1,93E-03	0,351111
28	1,89E+12	1,49E+13	-5,55E+01	1,82E-02	0,335556	4,54E+12	2,21E+13	-6,12E+01	1,93E-03	0,353333
29	1,66E+12	6,20E+12	-6,45E+01	2,05E-02	0,295556	5,55E+12	6,36E+12	-6,23E+01	2,72E-02	0,462222
30	3,40E+12	3,83E+12	-6,69E+01	3,30E-02	0,331111	7,55E+12	3,33E+13	-6,32E+01	1,93E-03	0,377778
31	6,08E+12	2,27E+13	-6,28E+01	2,58E-03	0,315556	5,55E+12	6,36E+12	-6,23E+01	2,72E-02	0,462222
32	1,56E+12	2,74E+12	-8,27E+02	3,16E-02	0,328889	2,60E+12	1,25E+12	-6,35E+01	2,72E-02	0,508889

Tablo 3.15. Devamı

DNM	NFT					EC				
	avgbest	maxfeas	minfeas	stdbest	feasrate	avgbest	maxfeas	minfeas	stdbest	feasrate
33	2,05E+12	3,96E+12	-3,16E+01	2,97E-02	0,482222	2,60E+12	1,25E+12	-6,35E+01	2,72E-02	0,508889
34	2,46E+12	7,86E+12	-6,34E+01	4,04E-02	0,344444	3,63E+12	3,57E+11	-6,35E+01	2,72E-02	0,495556
35	3,69E+12	1,09E+13	-6,35E+01	2,44E-03	0,311111	4,16E+12	4,66E+12	-6,23E+01	2,72E-02	0,506667
36	1,30E+12	1,19E+13	-5,47E+01	5,89E-03	0,331111	6,79E+12	4,33E+12	-6,23E+01	1,93E-03	0,368889
37	7,32E+12	1,12E+13	-5,24E+01	9,22E-03	0,313333	5,43E+12	3,11E+12	-6,35E+01	2,72E-02	0,48
38	3,55E+12	2,61E+13	-6,54E+01	4,75E-03	0,328889	5,32E+11	4,39E+11	-8,87E+02	2,72E-02	0,54
39	2,69E+12	5,68E+12	-5,25E+01	1,14E-02	0,306667	6,04E+12	1,27E+12	-6,35E+01	1,93E-03	0,364444
40	1,69E+12	5,02E+12	-6,35E+01	5,38E-02	0,306667	4,16E+12	4,66E+12	-6,23E+01	2,72E-02	0,506667
41	1,80E+12	4,71E+12	-1,32E+02	5,26E-02	0,324444	6,04E+12	1,27E+12	-6,35E+01	1,93E-03	0,364444
42	2,45E+12	1,10E+13	-5,92E+01	5,74E-03	0,306667	7,75E+12	2,26E+13	-6,04E+01	1,93E-03	0,364444
43	7,48E+12	1,13E+14	-4,74E+01	2,42E-02	0,284444	7,75E+12	2,26E+13	-6,04E+01	1,93E-03	0,364444
44	7,67E+12	4,19E+13	-3,30E+01	2,28E-02	0,26	5,43E+12	3,11E+12	-6,35E+01	2,72E-02	0,48
45	2,71E+12	3,58E+12	-5,83E+01	7,49E-03	0,3	4,54E+12	2,21E+13	-6,12E+01	1,93E-03	0,353333
46	9,79E+12	1,33E+13	-6,30E+01	3,41E-02	0,26	6,04E+12	1,27E+12	-6,35E+01	1,93E-03	0,364444
47	8,86E+12	8,97E+12	-5,63E+01	6,07E-03	0,3	5,32E+11	4,39E+11	-8,87E+02	2,72E-02	0,54
48	9,86E+12	1,97E+13	-4,46E+01	2,56E-02	0,28	7,50E+12	1,49E+13	-6,35E+01	2,72E-02	0,46
49	3,87E+12	9,82E+12	-4,86E+01	2,01E-02	0,284444	5,43E+12	3,11E+12	-6,35E+01	2,72E-02	0,48
50	5,38E+12	3,58E+13	-4,94E+01	9,15E-03	0,286667	2,17E+12	6,42E+10	-6,35E+01	2,72E-02	0,522222
51	8,32E+12	2,45E+13	-5,49E+02	2,31E-02	0,248889	5,55E+12	6,36E+12	-6,23E+01	2,72E-02	0,462222
52	5,87E+11	6,76E+12	-4,86E+01	7,48E-03	0,351111	5,32E+11	4,39E+11	-8,87E+02	2,72E-02	0,54
53	3,48E+12	2,85E+13	-5,42E+01	5,20E-03	0,304444	7,75E+12	2,26E+13	-6,04E+01	1,93E-03	0,364444
54	1,74E+12	7,47E+12	-6,02E+01	3,01E-02	0,308889	6,79E+12	4,33E+12	-6,23E+01	1,93E-03	0,368889
55	2,56E+12	2,54E+13	-8,13E+01	2,07E-02	0,315556	5,55E+12	6,36E+12	-6,23E+01	2,72E-02	0,462222
56	5,70E+12	1,35E+14	-5,09E+01	1,50E-02	0,286667	7,88E+12	1,04E+13	-6,08E+01	1,93E-03	0,373333
57	2,45E+12	1,12E+13	-5,46E+02	1,38E-02	0,315556	7,75E+12	2,26E+13	-6,04E+01	1,93E-03	0,364444
58	2,45E+12	2,15E+13	-5,70E+02	2,99E-03	0,324444	4,16E+12	4,66E+12	-6,23E+01	2,72E-02	0,506667
59	5,49E+12	3,50E+13	-5,62E+01	6,70E-03	0,295556	7,88E+12	1,04E+13	-6,08E+01	1,93E-03	0,373333
60	4,43E+12	4,58E+12	-2,33E+02	4,29E-02	0,351111	5,43E+12	3,11E+12	-6,35E+01	2,72E-02	0,48
61	2,25E+12	1,40E+12	-2,91E+02	6,83E-02	0,311111	6,79E+12	4,33E+12	-6,23E+01	1,93E-03	0,368889
62	1,38E+12	1,25E+13	-5,31E+01	2,90E-02	0,351111	5,45E+12	3,04E+12	-6,12E+01	1,93E-03	0,351111
63	2,78E+12	1,43E+13	-5,31E+01	1,36E-03	0,34	7,55E+12	3,33E+13	-6,32E+01	1,93E-03	0,377778
64	4,76E+12	2,82E+13	-6,55E+01	9,50E-03	0,32	5,45E+12	3,04E+12	-6,12E+01	1,93E-03	0,351111

Tablo 3.16. 100.000 Fonksiyon hesaplaması için NFT-EC performansı

DNM	NFT					EC				
	avgbest	maxfeas	minfeas	stdbest	feasrate	avgbest	maxfeas	minfeas	stdbest	feasrate
1	2,33E+12	4,33E+12	-5,83E+01	4,62E-06	0,417778	7,66E+11	3,73E+11	-6,84E+01	1,19E-07	0,662222
2	3,19E+12	5,93E+12	-6,81E+01	1,58E-04	0,411111	1,69E+10	1,86E-03	-1,99E+00	2,72E-02	0,864444
3	1,43E+12	2,50E+12	-6,84E+01	1,40E-03	0,424444	1,61E+02	1,93E-03	-1,23E+02	1,84E-02	0,873333
4	2,09E+12	2,65E+12	-7,85E+02	5,26E-02	0,402222	4,80E+11	3,69E+10	-6,84E+01	1,19E-07	0,684444
5	6,46E+12	4,72E+13	-6,15E+01	6,08E-04	0,357778	8,63E+11	9,99E+12	-6,84E+01	1,19E-07	0,671111
6	6,40E+11	3,50E+12	-5,04E+01	7,85E-11	0,562222	1,69E+10	1,86E-03	-1,99E+00	2,72E-02	0,864444
7	8,73E+11	6,53E+12	-6,21E+01	1,72E-25	0,482222	2,36E+12	3,20E+12	-6,84E+01	1,19E-07	0,633333
8	8,09E+12	2,46E+14	-4,74E+01	1,48E-02	0,335556	2,19E+09	1,16E-04	-3,88E+01	2,72E-02	0,868889
9	6,75E+12	1,22E+14	-6,83E+01	1,45E-03	0,431111	2,36E+12	3,20E+12	-6,84E+01	1,19E-07	0,633333
10	9,19E+11	3,13E+12	-6,80E+01	1,95E-06	0,431111	5,31E+08	5,06E+00	-4,08E+01	2,72E-02	0,877778
11	2,02E+12	5,86E+12	-6,83E+01	6,97E-06	0,4	1,40E+11	4,56E+00	-6,46E-01	2,72E-02	0,868889
12	2,94E+12	2,66E+13	-6,84E+01	1,36E-03	0,431111	2,36E+12	3,20E+12	-6,84E+01	1,19E-07	0,633333
13	3,17E+12	1,79E+13	-6,06E+01	1,20E-06	0,431111	2,36E+12	3,20E+12	-6,84E+01	1,19E-07	0,633333
14	3,14E+12	3,32E+13	-6,84E+01	1,72E-06	0,408889	8,45E+11	3,80E+02	-4,33E+01	2,72E-02	0,862222
15	1,85E+12	4,89E+12	-6,17E+01	8,96E-02	0,455556	1,54E+12	1,05E+12	-6,84E+01	1,19E-07	0,653333
16	2,68E+12	2,11E+13	-5,54E+01	1,63E-04	0,377778	1,66E+12	3,46E+12	-6,84E+01	1,19E-07	0,662222
17	2,34E+12	1,82E+12	-3,87E+02	7,57E-02	0,353333	5,31E+08	5,06E+00	-4,08E+01	2,72E-02	0,877778
18	1,18E+12	2,20E+12	-6,23E+01	2,41E-02	0,468889	8,63E+11	9,99E+12	-6,84E+01	1,19E-07	0,671111
19	1,64E+12	6,22E+12	-6,84E+01	8,62E-04	0,444444	2,19E+09	1,16E-04	-3,88E+01	2,72E-02	0,868889
20	1,46E+12	2,29E+13	-6,84E+01	4,60E-06	0,426667	1,24E+10	6,04E+02	-6,84E+01	1,19E-07	0,744444
21	1,57E+12	2,13E+12	-6,35E+01	4,98E-02	0,468889	8,45E+11	3,80E+02	-4,33E+01	2,72E-02	0,862222
22	5,36E+12	3,15E+12	-6,71E+01	7,24E-04	0,388889	1,24E+10	6,04E+02	-6,84E+01	1,19E-07	0,744444
23	4,38E+12	7,38E+12	-6,80E+01	2,47E-04	0,42	1,66E+12	3,46E+12	-6,84E+01	1,19E-07	0,662222
24	2,08E+12	2,68E+11	-7,20E+02	5,97E-02	0,284444	5,31E+08	5,06E+00	-4,08E+01	2,72E-02	0,877778
25	2,78E+12	9,28E+13	-6,35E+01	3,05E-02	0,451111	8,45E+11	3,80E+02	-4,33E+01	2,72E-02	0,862222
26	1,37E+12	5,73E+11	-6,23E+01	4,09E-02	0,435556	1,69E+10	1,86E-03	-1,99E+00	2,72E-02	0,864444
27	7,15E+11	8,16E+11	-5,69E+01	1,77E-18	0,531111	4,80E+11	3,69E+10	-6,84E+01	1,19E-07	0,684444
28	1,34E+12	3,76E+12	-6,84E+01	4,75E-05	0,471111	1,24E+10	6,04E+02	-6,84E+01	1,19E-07	0,744444
29	1,32E+12	3,81E+12	-6,84E+01	3,90E-03	0,402222	8,18E+10	6,50E+01	-1,95E+02	1,06E-02	0,866667
30	3,05E+12	2,37E+12	-6,77E+01	2,04E-02	0,44	1,66E+12	3,46E+12	-6,84E+01	1,19E-07	0,662222
31	5,88E+12	2,27E+13	-6,82E+01	3,12E-05	0,404444	8,18E+10	6,50E+01	-1,95E+02	1,06E-02	0,866667
32	1,33E+12	2,74E+12	-8,27E+02	3,16E-02	0,404444	2,19E+09	1,16E-04	-3,88E+01	2,72E-02	0,868889

Tablo 3.16. Devamı

DNM	NFT					EC				
	avgbest	maxfeas	minfeas	stdbest	feasrate	avgbest	maxfeas	minfeas	stdbest	feasrate
33	1,87E+12	1,54E+12	-3,16E+01	2,97E-02	0,582222	2,19E+09	1,16E-04	-3,88E+01	2,72E-02	0,868889
34	1,80E+12	1,30E+12	-6,35E+01	4,04E-02	0,417778	5,31E+08	5,06E+00	-4,08E+01	2,72E-02	0,877778
35	2,88E+12	8,67E+12	-6,80E+01	1,58E-04	0,408889	1,40E+11	4,56E+00	-6,46E-01	2,72E-02	0,868889
36	1,03E+12	9,24E+12	-6,84E+01	1,55E-03	0,437778	1,54E+12	1,05E+12	-6,84E+01	1,19E-07	0,653333
37	7,00E+12	1,12E+13	-6,75E+01	1,36E-03	0,386667	7,96E+10	1,16E-04	-8,35E+02	2,72E-02	0,844444
38	1,89E+12	4,66E+13	-6,80E+01	4,35E-03	0,431111	1,61E+02	1,93E-03	-1,23E+02	1,84E-02	0,873333
39	2,65E+12	7,75E+11	-6,63E+01	3,06E-05	0,388889	7,66E+11	3,73E+11	-6,84E+01	1,19E-07	0,662222
40	1,23E+12	1,74E+12	-6,35E+01	5,39E-02	0,388889	1,40E+11	4,56E+00	-6,46E-01	2,72E-02	0,868889
41	1,53E+12	3,51E+11	-1,32E+02	4,94E-02	0,371111	7,66E+11	3,73E+11	-6,84E+01	1,19E-07	0,662222
42	2,08E+12	7,58E+12	-6,80E+01	1,82E-04	0,386667	1,58E+12	7,50E+11	-6,84E+01	1,19E-07	0,637778
43	7,28E+12	1,13E+14	-5,79E+01	5,47E-05	0,382222	1,58E+12	7,50E+11	-6,84E+01	1,19E-07	0,637778
44	7,64E+12	4,19E+13	-4,58E+01	2,56E-02	0,32	7,96E+10	1,16E-04	-8,35E+02	2,72E-02	0,844444
45	2,33E+12	3,58E+12	-6,80E+01	1,34E-04	0,406667	1,24E+10	6,04E+02	-6,84E+01	1,19E-07	0,744444
46	7,58E+12	1,13E+13	-5,01E+01	7,56E-05	0,375556	7,66E+11	3,73E+11	-6,84E+01	1,19E-07	0,662222
47	8,16E+12	8,97E+12	-6,84E+01	1,35E-03	0,42	1,61E+02	1,93E-03	-1,23E+02	1,84E-02	0,873333
48	7,44E+12	9,44E+12	-6,29E+01	2,33E-14	0,508889	8,45E+11	3,80E+02	-4,33E+01	2,72E-02	0,862222
49	3,38E+12	5,59E+12	-5,66E+01	1,31E-05	0,411111	7,96E+10	1,16E-04	-8,35E+02	2,72E-02	0,844444
50	4,79E+12	8,72E+12	-6,43E+01	1,36E-05	0,408889	1,69E+10	1,86E-03	-1,99E+00	2,72E-02	0,864444
51	8,42E+12	4,71E+12	-5,38E+01	2,25E-03	0,342222	8,18E+10	6,50E+01	-1,95E+02	1,06E-02	0,866667
52	3,80E+11	2,10E+12	-6,84E+01	8,36E-06	0,475556	1,61E+02	1,93E-03	-1,23E+02	1,84E-02	0,873333
53	2,95E+12	2,85E+13	-6,83E+01	1,01E-05	0,408889	1,58E+12	7,50E+11	-6,84E+01	1,19E-07	0,637778
54	1,25E+12	3,61E+12	-6,56E+01	2,42E-02	0,413333	1,54E+12	1,05E+12	-6,84E+01	1,19E-07	0,653333
55	2,14E+12	4,87E+12	-8,13E+01	2,20E-02	0,448889	8,18E+10	6,50E+01	-1,95E+02	1,06E-02	0,866667
56	5,86E+12	1,35E+14	-6,60E+01	9,12E-05	0,362222	8,63E+11	9,99E+12	-6,84E+01	1,19E-07	0,671111
57	2,06E+12	1,80E+13	-6,79E+01	6,02E-03	0,451111	1,58E+12	7,50E+11	-6,84E+01	1,19E-07	0,637778
58	1,79E+12	2,79E+13	-6,72E+01	2,33E-04	0,406667	1,40E+11	4,56E+00	-6,46E-01	2,72E-02	0,868889
59	5,26E+12	3,50E+13	-7,21E+01	3,93E-04	0,413333	8,63E+11	9,99E+12	-6,84E+01	1,19E-07	0,671111
60	3,53E+12	4,58E+12	-2,33E+02	4,24E-02	0,464444	7,96E+10	1,16E-04	-8,35E+02	2,72E-02	0,844444
61	2,02E+12	7,51E+11	-2,91E+02	7,40E-02	0,3	1,54E+12	1,05E+12	-6,84E+01	1,19E-07	0,653333
62	9,86E+11	4,27E+12	-6,35E+01	2,90E-02	0,431111	4,80E+11	3,69E+10	-6,84E+01	1,19E-07	0,684444
63	2,06E+12	4,07E+12	-6,84E+01	4,94E-09	0,457778	1,66E+12	3,46E+12	-6,84E+01	1,19E-07	0,662222
64	4,05E+12	2,82E+13	-3,12E+02	9,41E-03	0,422222	4,80E+11	3,69E+10	-6,84E+01	1,19E-07	0,684444

Tablo 3.17. 200.000 Fonksiyon hesaplaması için NFT-EC performansı

DNM	NFT					EC				
	avgbest	maxfeas	minfeas	stdbest	feasrate	avgbest	maxfeas	minfeas	stdbest	feasrate
1	3,14E+12	4,50E+12	-6,84E+01	1,32E-11	0,482222	2,77E+10	4,19E+01	-2,07E+02	0,00E+00	0,831111
2	3,07E+12	5,26E+12	-6,84E+01	9,27E-06	0,448889	1,96E+03	8,09E-05	-2,51E+02	1,26E-02	0,873333
3	1,21E+12	1,38E+12	-6,84E+01	5,24E-04	0,486667	2,85E+01	3,89E-04	-2,65E+02	3,65E-30	0,868889
4	1,98E+12	2,14E+12	-7,85E+02	5,74E-02	0,455556	2,22E+10	3,33E+01	-6,84E+01	0,00E+00	0,842222
5	6,30E+12	4,72E+13	-6,79E+01	2,46E-05	0,424444	1,29E+10	5,82E+02	-6,84E+01	7,23E-14	0,831111
6	6,32E+11	7,17E+11	-5,74E+01	3,44E-28	0,493333	1,96E+03	8,09E-05	-2,51E+02	1,26E-02	0,873333
7	8,06E+11	1,99E+12	-6,21E+01	0,00E+00	0,48	1,25E+12	1,04E+13	-6,84E+01	7,23E-14	0,766667
8	7,99E+12	2,46E+14	-5,31E+01	2,95E-06	0,515556	-6,48E+01	1,19E-03	-2,00E+02	2,06E-30	0,855556
9	6,51E+12	1,22E+14	-6,84E+01	1,39E-03	0,493333	1,25E+12	1,04E+13	-6,84E+01	7,23E-14	0,766667
10	7,13E+11	7,26E+11	-6,84E+01	1,87E-07	0,497778	1,23E+02	9,09E-04	-1,23E+02	1,81E-02	0,875556
11	1,90E+12	3,15E+12	-6,84E+01	1,28E-06	0,444444	1,70E+10	8,07E-04	-1,99E+00	2,72E-02	0,882222
12	2,68E+12	2,66E+13	-6,84E+01	8,69E-05	0,486667	1,25E+12	1,04E+13	-6,84E+01	7,23E-14	0,766667
13	3,01E+12	4,28E+12	-6,84E+01	5,36E-18	0,517778	1,25E+12	1,04E+13	-6,84E+01	7,23E-14	0,766667
14	3,04E+12	3,32E+13	-6,84E+01	4,42E-07	0,471111	6,62E+10	6,49E-05	-8,35E+02	2,72E-02	0,873333
15	1,82E+12	4,89E+12	-6,17E+01	8,96E-02	0,495556	4,33E+11	5,32E+10	-6,84E+01	7,23E-14	0,771111
16	2,62E+12	2,11E+13	-6,10E+01	1,35E-05	0,462222	4,34E+11	1,61E+12	-8,87E+02	0,00E+00	0,76
17	2,04E+12	1,82E+12	-3,87E+02	7,57E-02	0,371111	1,23E+02	9,09E-04	-1,23E+02	1,81E-02	0,875556
18	1,11E+12	2,45E+12	-6,35E+01	2,41E-02	0,493333	1,29E+10	5,82E+02	-6,84E+01	7,23E-14	0,831111
19	1,32E+12	5,08E+12	-6,84E+01	1,33E-04	0,495556	-6,48E+01	1,19E-03	-2,00E+02	2,06E-30	0,855556
20	1,33E+12	2,29E+13	-6,84E+01	1,21E-06	0,488889	6,01E+00	5,08E+01	-6,84E+01	0,00E+00	0,884444
21	1,31E+12	7,57E+11	-6,35E+01	4,98E-02	0,524444	6,62E+10	6,49E-05	-8,35E+02	2,72E-02	0,873333
22	5,04E+12	8,56E+12	-1,13E+02	9,06E-05	0,453333	6,01E+00	5,08E+01	-6,84E+01	0,00E+00	0,884444
23	4,25E+12	7,38E+12	-1,14E+02	2,16E-05	0,448889	4,34E+11	1,61E+12	-8,87E+02	0,00E+00	0,76
24	1,99E+12	1,79E+11	7,20E+02	5,97E-02	0,328889	1,23E+02	9,09E-04	-1,23E+02	1,81E-02	0,875556
25	2,58E+12	1,50E+12	-6,35E+01	3,05E-02	0,504444	6,62E+10	6,49E-05	-8,35E+02	2,72E-02	0,873333
26	1,27E+12	2,78E+11	-6,23E+01	3,61E-02	0,471111	1,96E+03	8,09E-05	-2,51E+02	1,26E-02	0,873333
27	6,11E+11	8,16E+11	-6,35E+01	0,00E+00	0,446667	2,22E+10	3,33E+01	-6,84E+01	0,00E+00	0,842222
28	6,99E+11	1,58E+12	-6,84E+01	1,64E-06	0,517778	6,01E+00	5,08E+01	-6,84E+01	0,00E+00	0,884444
29	1,17E+12	3,81E+12	-6,84E+01	2,23E-04	0,48	1,75E+09	6,49E-05	-3,88E+01	2,72E-02	0,873333
30	2,76E+12	2,37E+12	-6,82E+01	2,30E-02	0,497778	4,34E+11	1,61E+12	-8,87E+02	0,00E+00	0,76
31	5,72E+12	2,27E+13	-6,84E+01	5,14E-06	0,466667	1,75E+09	6,49E-05	-3,88E+01	2,72E-02	0,873333
32	1,17E+12	1,86E+12	-8,27E+02	3,16E-02	0,435556	-6,48E+01	1,19E-03	-2,00E+02	2,06E-30	0,855556

Tablo 3.17. Devamı

DNM	NFT					EC				
	avgbest	maxfeas	minfeas	stdbest	feasrate	avgbest	maxfeas	minfeas	stdbest	feasrate
33	1,84E+12	1,54E+12	-3,16E+01	2,97E-02	0,588889	-6,48E+01	1,19E-03	-2,00E+02	2,06E-30	0,855556
34	1,43E+12	8,14E+11	-6,35E+01	4,04E-02	0,471111	1,23E+02	9,09E-04	-1,23E+02	1,81E-02	0,875556
35	2,77E+12	3,69E+12	-6,82E+01	6,07E-06	0,473333	1,70E+10	8,07E-04	-1,99E+00	2,72E-02	0,882222
36	6,97E+11	2,49E+12	-6,84E+01	6,15E-06	0,535556	4,33E+11	5,32E+10	-6,84E+01	7,23E-14	0,771111
37	6,93E+12	7,92E+12	-6,84E+01	1,35E-03	0,433333	7,23E+09	1,26E-02	-1,99E+00	2,72E-02	0,862222
38	1,71E+12	4,66E+13	-6,84E+01	4,29E-03	0,486667	2,85E+01	3,89E-04	-2,65E+02	3,65E-30	0,868889
39	2,57E+12	7,75E+11	-6,84E+01	3,44E-06	0,44	2,77E+10	4,19E+01	-2,07E+02	0,00E+00	0,831111
40	1,13E+12	8,57E+11	-6,35E+01	5,39E-02	0,453333	1,70E+10	8,07E-04	-1,99E+00	2,72E-02	0,882222
41	1,44E+12	3,51E+11	-1,32E+02	5,25E-02	0,368889	2,77E+10	4,19E+01	-2,07E+02	0,00E+00	0,831111
42	1,90E+12	6,50E+12	-6,83E+01	1,54E-05	0,446667	6,45E+11	3,73E+11	-6,84E+01	7,23E-14	0,768889
43	7,12E+12	1,13E+14	-6,50E+01	7,27E-06	0,455556	6,45E+11	3,73E+11	-6,84E+01	7,23E-14	0,768889
44	7,55E+12	4,19E+13	-4,83E+01	3,00E-03	0,371111	7,23E+09	1,26E-02	-1,99E+00	2,72E-02	0,862222
45	2,13E+12	2,15E+12	-1,20E+02	1,33E-05	0,453333	6,01E+00	5,08E+01	-6,84E+01	0,00E+00	0,884444
46	7,33E+12	6,85E+12	-6,35E+01	9,37E-13	0,535556	2,77E+10	4,19E+01	-2,07E+02	0,00E+00	0,831111
47	7,92E+12	8,97E+12	-6,84E+01	3,54E-04	0,484444	2,85E+01	3,89E-04	-2,65E+02	3,65E-30	0,868889
48	7,27E+12	4,63E+12	-6,29E+01	0,00E+00	0,473333	6,62E+10	6,49E-05	-8,35E+02	2,72E-02	0,873333
49	3,42E+12	3,21E+12	-6,49E+01	6,58E-07	0,46	7,23E+09	1,26E-02	-1,99E+00	2,72E-02	0,862222
50	4,65E+12	5,18E+12	-6,84E+01	2,51E-10	0,477778	1,96E+03	8,09E-05	-2,51E+02	1,26E-02	0,873333
51	8,34E+12	4,71E+12	-5,90E+01	4,85E-05	0,437778	1,75E+09	6,49E-05	-3,88E+01	2,72E-02	0,873333
52	3,01E+11	1,69E+12	-6,84E+01	1,41E-06	0,526667	2,85E+01	3,89E-04	-2,65E+02	3,65E-30	0,868889
53	2,86E+12	2,85E+13	-6,84E+01	1,89E-06	0,457778	6,45E+11	3,73E+11	-6,84E+01	7,23E-14	0,768889
54	1,07E+12	3,61E+12	-6,56E+01	3,02E-02	0,468889	4,33E+11	5,32E+10	-6,84E+01	7,23E-14	0,771111
55	1,91E+12	4,87E+12	-8,13E+01	2,77E-02	0,515556	1,75E+09	6,49E-05	-3,88E+01	2,72E-02	0,873333
56	5,72E+12	1,35E+14	-8,43E+01	1,33E-05	0,435556	1,29E+10	5,82E+02	-6,84E+01	7,23E-14	0,831111
57	1,89E+12	1,80E+13	-6,83E+01	5,99E-03	0,504444	6,45E+11	3,73E+11	-6,84E+01	7,23E-14	0,768889
58	1,56E+12	2,79E+13	-6,76E+01	7,33E-05	0,468889	1,70E+10	8,07E-04	-1,99E+00	2,72E-02	0,882222
59	5,06E+12	3,50E+13	-7,21E+01	2,54E-05	0,46	1,29E+10	5,82E+02	-6,84E+01	7,23E-14	0,831111
60	3,45E+12	4,58E+12	-2,33E+02	4,39E-02	0,506667	7,23E+09	1,26E-02	-1,99E+00	2,72E-02	0,862222
61	1,88E+12	6,82E+11	-2,91E+02	7,40E-02	0,391111	4,33E+11	5,32E+10	-6,84E+01	7,23E-14	0,771111
62	8,45E+11	4,27E+12	-6,35E+01	2,90E-02	0,502222	2,22E+10	3,33E+01	-6,84E+01	0,00E+00	0,842222
63	1,51E+12	1,73E+12	-6,84E+01	0,00E+00	0,511111	4,34E+11	1,61E+12	-8,87E+02	0,00E+00	0,76
64	3,80E+12	2,82E+13	-3,12E+02	9,35E-03	0,473333	2,22E+10	3,33E+01	-6,84E+01	0,00E+00	0,842222

Bu tez çalışmasında DGA ile birlikte kısıt yönetimi metotlarından olan NFT ve EC metotları sınırlamalara sahip olan 18 Benchmark test problemi üzerinde başarıyla uygulanmış ve performansları karşılaştırılmıştır.

Denemeler neticesinde ortaya çıkan bireysel ortalama uygunluk dağılımlarına bakıldığında, EC metodunun NFT ye olan üstünlüğünü çok açık bir şekilde gözlemlenebilmektedir. EC de 0,9 ortalama uygunluk oranına kadar çıkabildiği, NFT metodunun ise 0,6 civarında kaldığı gözlenebilir. Her iki metodun performansı problemin zorluğuna göre değişmektedir.

EC metodunda ilk etapta fonksiyon hesaplama sayısının 100.000 den 200.000 çıkarılmasıyla elde edilen maksimum ortalama uygunluk oranının(0,88) değişmemesine rağmen, 100.000 fonksiyon hesaplamasında 0,88 e çıkan 4 parametre seti söz konusu iken, 200.000 fonksiyon hesaplamasında bu sayı ortalama uygunluk değerlerinin tam sayıya yuvarlanması da göz önünde bulundurulursa 20 ye kadar çıkabildiği yani 20 farklı parametre setinin optimal çözüme yakın sonuçları verdiği gözlenmiştir.

Algoritmanın performans açısından gidişatını görmek adına, fonksiyon hesaplama sayısı arttırdığında(300.000-500.000-1.000.000) ortalama uygunluk oranlarının özellikle EC metodunda arttığı gözlemlenmiştir. Fonksiyon hesaplama sayısı 200.000 den 500.000 e çıkarıldığında ortalama uygunluk oranı 0,88 den 0,991 e; 1.000.000 a çıkarıldığında ise 0,997 ye yükseldiği gözlemlenmiştir. Bu sonuçlardan, EC metodunun deney tasarımında kullanılan parametrelere ek olarak uygunluğun yükseltilmesi adına yapılabilecek en uygun ince ayarın, optimum sonucu elde etmenin gereken zamana kıyasla daha önemli olduğu problemlerde, fonksiyon hesaplama sayısındaki değişim olduğu çıkarılmaktadır.

NFT metodunda, fonksiyon hesaplama sayısının arttırılmasının(500.000-1.000.000) ortalama uygunluk oranlarının yükselmesi yani problemlerdeki optimum çözümlere yaklaşılması adına pek fazla etki etmediği gözlenmiştir. Fonksiyon hesaplama sayısı 200.000 den 500.000 e çıkarıldığında ortalama uygunluk oranı 0,58 den 0,59 a; 1.000.000 a çıkarıldığında ise ancak 0,61 e yükselbildiği gözlemlenmiştir.

NFT metodunun deney tasarımında kullanılan CR, MR ve NP parametrelerine ek olarak metodun kendi formülasyonundaki alpha, NFT_0 , lamda vb. parametreleri üzerinde yapılan artı ya da eksi yöndeki deęişimlere rağmen ortalama uygunluk oranlarının 0,6 nın üzerine çok az miktarda çıkabildięi görülmüştür. Buna ek olarak daha zor problemler söz konusunda olduęunda arttırılması düşünölen popölasyon boyutunun 10 dan 30 a çıkarılmasına rağmen, ortalama uygunluk oranının 0,6 deęerinin üzerine küçük miktarlarda çıkabildięi yani yerel optimuma takıldıęı gözlenmiştir.

NFT metodunun her bir problem için en uygun parametre seti ile yapılan denemeler neticesinde ortaya çıkan uygunluk oranları göz ününe alındıęında en başarısız olduęu problemin Ek A da verilen C5 olduęu gözlenmiştir. Aynı şekilde EC metodunun her bir problem için en uygun parametre seti ile yapılan denemeler neticesinde ortaya çıkan uygunluk oranları göz ününe alındıęında, 200.000 fonksiyon hesaplamasına göre ortalama uygunluk deęeri 0,68 den 500.000 fonksiyon hesaplamasında 0,84 e; 1.000.000 fonksiyon hesaplamasında ise 0,96 çıkarak nispeten en başarısız olduęu problemin Ek A da verilen C17 olduęu gözlenmiştir.

Hesaplama karmaşıklığı açısından bakıldıęında, NFT ve EC metodlarının farklı fonksiyon hesaplama sayılarında, algoritma durdurulana kadar ki geöen zamanlara göre özellikle düşük fonksiyon hesaplamalarında NFT metodunun EC metoduna göre nispeten daha hızlı sonuçlandıęı; yüksek fonksiyon hesaplama sayılarında EC metodunun NFT metoduna göre daha hızlı olduęu; ortalama fonksiyon hesaplama sayıları göz önüne alındıęında ise birbirlerine yakın sürede sonuçlandıkları gözlemlenmiştir(200.000 fonksiyon hesaplama sayısında en uygun parametre setlerine göre yapılan denemede; NFT metodunun 18 problem için ortalama 10,9 sn de sonuçlandıęı; EC metodunun ise 18 problem için ortalama 11,6 sn de sonuçlandıęı gözlenmiştir).

Her iki metodunda, zaman açısından bakıldıęında, en fazla zaman harcadıęı problemin C15 olduęu gözlenmiştir.

NFT metodunun, araştırma sürecinde farklı parametre setlerine göre yapılan denemelere göre algoritmanın sonlanmasına kadar geöen sürede EC metodu kadar verimli çalışmadıęı saptanmıştır. Yani aynı sürede EC metodu, hem maksimum

uygunluk ve ortalama uygunluk deęerleri aısından hem de optimuma yakın daha fazla sonuçlar elde edebilmesi göz önüne alındığında, NFT metodunun oldukça başarısız kaldığı gözlenmiştir.

EC metodunda 20.000-100.000 ve 200.000 fonksiyon hesaplama sayılarında CR-MR parametrelerinin etkileşimlerine bakıldığında, etkileşimin yok denecek kadar az olduğu gözlemlenebilir. Aksine, NFT metodunda CR-MR etkileşiminin, MR nin deęeri en yüksek deęerine doğru çıktıka CR nin deęerinin düşerek çözümün uygunluk deęerinin azaldığı ya da MR nin ortalama deęere yükselmesi ile CR nin en yüksek deęerine ulaşarak çözümün uygunluk deęerinin arttığı şeklindeki örnekleri de göz önüne alındığında bir hayli fazla olduğu görülmüştür.

NFT metodunun, 192 deneme neticesinde dięer çözümlere göre nispeten en uygun 1 er sonuç bulabildiğı; EC metodunun ise, fonksiyon hesaplama sayısının artması ile birlikte uygunluk oranı optimuma yakın onlarca sonuç bulabilmekte olduğu görülmüştür.

EC metodunda, 20.000 ve 100.000 fonksiyon hesaplama sayılarında popülasyon sayısı 30 seçildiğı zaman daha uygun çözümler ortaya çıkabildiğı gözlemlenebilir. Yani popülasyon sayısının 100 seçilmesi ile birlikte çözümün uygunluk oranı ciddi bir şekilde düşebilmektedir. Ancak 200.000 ve daha fazla fonksiyon hesaplama sayılarında popülasyon sayısının 100 seçilmesi çözümün uygunluk oranını 30 seçilmesinden daha fazla arttırmakta olduğu gözlemlenebilir. Buradan nispeten düşük fonksiyon hesaplama sayılarında düşük popülasyon sayısı seçilmesi; yüksek fonksiyon hesaplama sayılarında ise yüksek popülasyon sayısı seçilmesi gerektiğı sonucuna varılabilmektedir.

Bu tez alışmasında, fonksiyon hesaplama sayısı arttıka her iki metotta da farklı parametre setlerinin en uygun sonuçları veren parametrelerinin başarılı bir şekilde optimize edildiğı; EC metodunun kullanılması ile elde edilen ortalama uygunluk oranı, NFT metodunun kullanılması ile elde edilen ortalama uygunluk oranının yaklaşık iki katına yakın ıkması EC metodunun NFT metoduna olan üstünlüğünü açık bir şekilde göstermektedir.

EKLER

EK A

Bu bölümde, kısıtlamalara sahip 18 Benchmark test problemi tanımlanmıştır.

C1

$$\min f(x) = \max(x)$$

$$h_1(x) = \frac{1}{D} \sum_{i=1}^D (-x_i \sin(\sqrt{|x_i|})) = 0$$

$$h_2(x) = \frac{1}{D} \sum_{i=1}^D (-x_i \cos(0.5\sqrt{|x_i|})) = 0$$

$$x \in [-500, 500]^D$$

C2

$$\min f(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z) + 10)$$

$$h_1(x) = \frac{1}{D} \sum_{i=1}^D (-x \sin(2\sqrt{|x_i|})) = 0$$

$$h_2(x) = \frac{1}{D} \sum_{i=1}^D (x \sin(0.5\sqrt{|x_i|})) = 0$$

$$x \in [-5, 5]^D$$

$$z = x - 10$$

C3

$$\min f(x) = \frac{1}{D} \sum_{i=1}^D (-z_i \cos(2\sqrt{|z_i|}))$$

$$h(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1}) + (x_i - 1)) = 0$$

$$x \in [-100, 100]^D$$

$$z = x - 1$$

C4

$$\min f(x) = \max(x)$$

$$h(x) = 418.9828872724338D - \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|})) = 0$$

$$x \in [-500, 500]^D$$

C5

$$\min f(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2) = 0$$

$$h(x) = \sum_{i=1}^{D-1} (x_i - x_{i+1})^2 = 0$$

$$x \in [-1000, 1000]^D$$

C6

$$\min f(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2) = 0$$

$$g(x) = \sum_{i=1}^D z \sin(0.1z) \leq 0$$

$$x \in [-100, 100]^D$$

$$z = x - 1$$

C7

$$\min f(x) = \max(x)$$

$$h_1(x) = \frac{1}{D} \sum_{i=1}^D (x_i \cos(\sqrt{|x_i|})) = 0$$

$$h_2(x) = \sum_{i=1}^{D/2-1} (x_i - x_{i+1})^2 = 0$$

$$h_3(x) = \sum_{i=D/2+1}^{D-1} (x_i^2 - x_{i+1})^2 = 0$$

$$h_4(x) = \sum_{i=1}^D x = 0$$

$$x \in [-50, 50]^D$$

C8

$$\min f(x) = \frac{1}{D} \sum_{i=1}^D (-x_i \sin(\sqrt{|x_i|}))$$

$$g_1(x) = 10 + \frac{1}{D} \sum_{i=1}^D (-y_i \sin(\sqrt{|y_i|})) \leq 0$$

$$g_2(x) = 10 + \frac{1}{D} \sum_{i=1}^D (-z_i \sin(\sqrt{|z_i|})) \leq 0$$

$$x \in [-100, 100]^D$$

$$y = x - 100, z = x + 50$$

C9

$$\min f(x) = \frac{1}{D} \sum_{i=1}^D (-x_i \sin(\sqrt{|x_i|}))$$

$$g_1(x) = -50 + \frac{1}{100D} \sum_{i=1}^D x_i^2 \leq 0$$

$$g_2(x) = \frac{50}{D} \sum_{i=1}^D \sin\left(\frac{1}{50} \pi x\right) \leq 0$$

$$g_3(x) = 75 - 50 \left(\sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \right) \leq 0$$

$$x \in [-500, 500]^D$$

C10

$$\min f(x) = \max(x)$$

$$g_1(x) = 10 - \frac{1}{D} \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10] \leq 0$$

$$g_2(x) = \frac{1}{D} \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10] - 15 \leq 0$$

$$h(x) = \frac{1}{D} \sum_{i=1}^D [y_i^2 - 10 \cos(2\pi y_i) + 10] - 20 \leq 0$$

$$x \in [-5.12, 5.12]^D$$

$$y = x - 0.5$$

C11

$$\min f(x) = \sum_{i=1}^{D-1} (z_i^2 - z_{i+1})^2 + (z_i - 1)^2$$

$$g(x) = 418.9828872724338D - \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|})) = 0$$

$$x \in [0, 5000]^D$$

$$z = x - 419.9687462275036$$

C12

$$\min f(x) = 418.9828872724338D - \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|})) = 0$$

$$h(x) = \sum_{i=1}^{D-1} (x_i - x_{i+1})^2 = 0$$

$$x \in [-500, 500]^D$$

C13

$$\min f(x) = \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|})) = 0$$

$$h_1 = \sum_{i=1}^{D-1} (x_i - x_{i+1})^2 = 0$$

$$h_2 = \sum_{i=1}^{D-1} (x_i - x_{i+1})^2 = 0$$

$$x \in [-10000, 10000]^D$$

C16

$$\min f(x) = \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|})) = 0$$

$$h(x) = \sum_{i=1}^{D-1} (x_i^2 - x_{i+1})^2 = 0$$

$$g(x) = \sum_{i=1}^D (x - 100 \cos(0.1x) + 10) \leq 0$$

$$x \in [-1000, 1000]^D$$

C14

$$\min f(x) = - \left| \frac{\sum_{i=1}^D \cos^4(x_i) - 2 \prod_{i=1}^D \cos^2(x_i)}{\sqrt{\sum_{i=1}^D i x_i^2}} \right|$$

$$g_1(x) = 0.75 - \prod_{i=1}^D x_i \leq 0$$

$$g_2(x) = \sum_{i=1}^D x_i - 7.5D \leq 0$$

$$x \in [0, 10]^D$$

C17

$$\min f(x) = 418.9828872724338D - \sum_{i=1}^D (x_i \sin(\sqrt{|x_i|})) = 0$$

$$h(x) = \sum_{i=1}^{D-1} (x_i - x_{i+1})^2 = 0$$

$$g(x) = \sum_{i=1}^D (1000 \cos(0.01x) - x) \leq 0$$

$$x \in [-1000, 1000]^D$$

C15

$$\min f(x) = \sum_{i=1}^D -z \sin(0.1z)$$

$$g(x) = \sum_{i=1}^D (x - 100 \cos(0.1x) + 10) \leq 0$$

$$h(x) = \sum_{i=1}^{D-1} (x_i - x_{i+1})^2 = 0$$

$$x \in [-100, 100]^D$$

$$z = x - 60$$

C18

$$\min f(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

$$g(x) = 0.5 - \exp(-0.1 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - 3 \exp(\frac{1}{D} \sum_{i=1}^D \cos(0.1x)) + \exp(1)$$

$$x \in [-140, 140]^D$$

KAYNAKLAR

1. Coello, C.A.C, Jan, 2002. "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state-of-the-art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, nos. 11-12, pp. 1245-1287.
2. Mallipeddi, R. Suganthan, P.N. August 2010. "Ensemble of constraint handling techniques" *IEEE Transactions on Evolutionary Computation*, vol. 14 , 561-579.
3. Jin, Y. Jan. 2005. "A comprehensive survey of fitness approximation in evolutionary computation", *Soft Comput.*, vol. 9, no. 1, pp. 3-12.
4. Michalewicz, Z., Schoenauer, M. 1996. "Evolutionary algorithms for constrained parameter optimization problems", *Evol. Comput.*, vol. 4, no. 1, pp. 1-32.
5. Koziel, S., Michalewicz, Z. 1999. "Evolutionary algorithms, homomorphous mappings, and constrained parameter optimization", *Evol. Comput.*, vol. 7, no. 1, pp. 19-44.
6. Wang, Y., Cai, Z., Guo, G., Zhuo, Y. Jun 2007. "Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems ," *IEEE Trans. Syst., Man, Cybern.*, vol. 37, no. 3, pp. 560-575.
7. Gözüdeli, Y., Akcayol M. A., 2007. "Genetik algoritma ile web sayfası düzeninin gerçek zamanlı optimizasyonu", *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, Cilt 22, No 2, 431-439.
8. Keskindürk, T., 3- 5 Temmuz 2006. "Diferansiyel gelişim algoritması", 214-217, YA/EM 2006 – Yöneylem Araştırması / Endüstri Mühendisliği – XXVI. Ulusal Kongresi – Kocaeli.
9. Koyuncu, C. A., 2006. "Farksal Gelişim Algoritmasının İncelenmesi ve İşaret Kestiriminde Kullanılması". Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yayınlanmamış Yüksek Lisans Tezi.

10. Keskinürk, T., 2006a. "Gezgin satıcı probleminin farksal gelişim algoritması ile çözümü", s:651-657. VI. Ulusal Üretim Araştırmaları Sempozyumu, İstanbul Kültür Üniversitesi.
11. Taşgetiren, M.F., Liang, Y.C., Pan, Q.K., Suganthan, P., 2009. "Discrete/Binary Approach. C.G. Onwubolu, ve D. Davendra, (Ed.), Differential evolution: a handbook for global permutation-based combinatorial optimization". (V.175) içinde (s:139-162). Berlin: Springer
12. Storn, R., Price, K., 1995. "Differential Evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces", International Computer Science Institute, Berkeley, TechnicalReport, 95-012.
13. Keskinürk, T., 2006. "Diferansiyel gelişim algoritması", İstanbul Ticaret Üniversitesi FenBilimleri Dergisi, Cilt 9, 85-99.
14. Mayer, D.G., Kinghorn, B.P., Archer, A.A., 2005. "Differential Evolution– An easy and efficient evolutionary algorithm for model optimisation", Agricultural Systems, 83, 315-328.
15. Karaboğa N., Koyuncu C. A., 2005. "Diferansiyel gelişim algoritması kullanılarak sinyal kestirimine yönelik adaptif SDY süzgeç tasarımı", s.91-95, II. İletişim Teknolojileri Ulusal Sempozyumu-İTUSEM 2005, 17-19 Kasım, Adana.
16. Shiakolas, P. S., Koladiye, D., Kebrle, J., 2005. "On the optimum synthesis of six-bar linkages using differential evolution and the geometric centroid of precision positions technique", Mechanism and Machine Theory, 40, 319-335.
17. Karaboga, D., 2004. Yapay Zeka Optimizasyonu Algoritmaları, AtlasYayın Dağıtım, 199 s.
18. Sun, J., et al., 2005. DE/EDA: "A new evolutionary algorithm for global optimization", Information Sciences,169, 249-262.

19. Qin, A. K., Huang, V. L., Suganthan, P. N. Apr. 2009. "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398-417.
20. Wang, Y., Cai, Z., Zhuo, Y., Zeng, W. Feb. 2008. "An adaptive tradeoff model for constrained optimization evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 80-92.
21. Smith, A. E., David, W. Coit, September 1995. "Penalty functions," Section C 5.2 of *Handbook of Evolutionary Computation*, A Joint Publication of Oxford University Press and Institute of Physics Publishing , pp. 1-13.
22. Taşgetiren, M.F., Suganthan P. N., Pan, Q.K., Mallipeddi R., Sarman S., 2010-CCIB. "An ensemble of differential evolution algorithms for constrained function optimization," WCCI 2010 IEEE World Congress on Computational Intelligence , Barcelona, Spain, July, 18-23.
23. Sebastiaan B., Pascal R. M. Storchi and Ben J M Heijmen. November, 2009. "The equivalence of multi-criteria methods for radiotherapy plan optimization," *Phys. Med. Biol.* 54, 7199–7209, 17.
24. Be'rubé', J. F., Gendreau, M., Potvin, J. Y. 7 December 2007. "An exact ϵ -constraint method for bi-objective combinatorial optimization problems: application to the traveling salesman problem with profits," *European Journal of Operational Research* 194 , 39–50.
25. Ehrgott M., Ruzika S., August 30, 2004. "An improved epsilon constraint method for multiobjective programming".
26. Laumanns M., Thiele L., and Zitzler E., 2005. "An adaptive scheme to generate the pareto front based on the epsilon-constraint method," In: Branke J., Deb K., Miettinen K., Steuer R. (eds.): *Practical Approaches to Multi-Objective Optimization*. Dagstuhl Seminar Proceedings, Vol. 04461

27. Şanyılmaz, M., 2006. "Deney Tasarımı ve Kalite Geliştirme Faaliyetlerinde Taguchi Yöntemi ile Bir Uygulama". Dumlupınar Üniversitesi, Yayınlanmamış Yüksek Lisans Tezi. s:2-9.
28. Karakoç, Ö., 2006. "Deneilerin Faktöriyel Tasarımı". Marmara Üniversitesi, Yayınlanmamış Yüksek Lisans Tezi. s:3-14.
29. Şanyılmaz, M., 2006. "Deney Tasarımı ve Kalite Geliştirme Faaliyetlerinde Taguchi Yöntemi ile Bir Uygulama". Dumlupınar Üniversitesi, Yayınlanmamış Yüksek Lisans Tezi. s:2-9.
30. Kasap, Ş., 2006. "F-4 Uçaklarında Yakıt Tüketimini Etkileyen Faktörlerin 2^k Deneileri ve Taguchi Yöntemiyle Belirlenmesi". Anadolu Üniversitesi, Yayınlanmamış Yüksek Lisans Tezi. s:12-15.
31. Becerra R. L., and Coello C. A. C., 2006. "Solving hard multiobjective optimization problems using epsilon constraint with cultured differential evolution," Conference: Parallel Problem Solving from Nature - PPSN , pp. 543-552.
32. Taşgetiren M. F., 2002. "A genetic algorithm with an adaptive penalty function for orienteering problem", Journal of Economic and Social Research 4 (2), 1-26.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Serdar Kasım KARATAŞ

Uyruğu: Türkiye (TC)

Doğum Tarihi ve Yeri: 8 Kasım 1984, Kayseri

Medeni Durumu: Bekâr

Tel: +90 0532 347 75 59

email: 4010930805@erciyes.edu.tr

Yazışma Adresi: Ziya Gökalp Mah. Milli Egemenlik Cad. Konak Apt. No:19/A

Kocasinan/KAYSERİ

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	EÜ Fen Bilimleri Enstitüsü Bil. Müh.	2011
Lisans	OMÜ Eğitim Fakültesi BÖTE	2008
Lise	Sami Yangın Anadolu Lisesi, KAYSERİ	2003

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2008- Halen	Mehmet Sepici 60.Yıl Cumhuriyet İÖO	Bilişim Teknolojileri Öğretmeni

YABANCI DİL

İngilizce, Almanca