

**ÇOK AMAÇLI GENETİK
ALGORİTMALAR : TEMELLERİ VE
UYGULAMALARI**

ENGİN UFUK ERGÜL

DOKTORA TEZİ

**ELEKTRİK-ELEKTRONİK
MÜHENDİSLİĞİ ANA BİLİM DALI**

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ÇOK AMAÇLI GENETİK ALGORİTMALAR:
TEMELLERİ VE UYGULAMALARI

ENGİN UFUK ERGÜL

DOKTORA TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI

AKADEMİK DANIŞMAN
Yrd. Doç. Dr. İLYAS EMİNOĞLU

SAMSUN - 2010

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bu çalışma jürimiz tarafından / ... /..... tarihinde yapılan sınav ile Elektrik-Elektronik Mühendisliği Anabilim Dalı'nda DOKTORA tezi olarak kabul edilmiştir.

Başkan :

Üye :

Üye :

Üye :

Üye :

ONAY :

Yukarıdaki imzaların adı geçen öğretim üyelerine ait olduğunu onaylarım.

..../..../2010

Prof. Dr. Hasan GÜMÜŞ
Fen Bilimleri Enstitüsü Müdürü

ÇOK AMAÇLI GENETİK ALGORİTMALAR: TEMELLERİ VE UYGULAMALARI

ÖZ

Bu çalışmada topluluk tabanlı bir optimizasyon yöntemi olan genetik algoritmalar (GA) incelenmektedir. Tezin ana amacı, çok amaçlı genetik algoritmaların (ÇAGA) puanlama ve elitizm mekanizmalarının iyileştirilmesidir. Bireylerin baskınlık gücünden yararlanan, çeşitlilik için farklı yöntemler içeren, iki yeni ÇAGA yöntemi (DOPGA+, DOPGA2) önerilmektedir. Bu yöntemler daha önce önerilen DOPGA yönteminde bazı değişiklikler yapılarak elde edilmiştir. Önerilen yöntemler literatürde sık kullanılan yöntemlerle (SPEA ve SPEA2) yakınsama ve çeşitlilik ölçütleri üzerinden karşılaştırılmışlardır. Ayrıca, literatürde yer alan bazı yöntemlerin (NSGA ve SPEA) puanlama yeteneğini artırmak için, ÇAGA yöntem iyileştirmeleri önerilmiş ve bunlar orjinal yöntemlerle karşılaştırılmışlardır.

Bireylerin seçilme baskısının seçme mekanizmasından önce değiştirilerek, ÇAGA' ların başarımının artırılması için gama düzeltmesi ile puan ölçekleme (GDPÖ) yöntemi önerilmiştir. Bu yöntem, çeşitli gama değerleri kullanılarak, tek amaçlı GA' lar ile literatürde sık kullanılan ve yeni önerilen ÇAGA yöntemlerine uygulanmıştır. Sonuç olarak, GDPÖ ile yakınsama yeteneğinin artırılabilceği görülmüştür. Elitizm mekanizması GA' ların başarımını önemli ölçüde artırmaktadır. Bu çalışmada, literatürde yer alan pasif sakla/aktar yapıdaki elitizm mekanizması, sakla/uyar/aktar yapıdaki etkin elitizm mekanizması ile yer değiştirilerek, ÇAGA' ların başarımının artırılabilceği görülmüştür. ÇAGA yöntemlerinin sıralama (puan atama) yeteneklerinin ölçülmesi için Ceza ve Ödül başarımlar ölçütleri tanımlanmıştır. Bu iki ölçüt ile bir ÇAGA yöntemi tarafından seçme mekanizmasına ne kadar nitelikli bilgi aktarıldığı sezgisel ve istatistiksel olarak tespit edilebilmektedir.

Son olarak, önerilen ve literatürde sık kullanılan ÇAGA yöntemlerini içinde barındıran bir kullanıcı arayüzü (MATGAT) tasarlanmıştır. Bu arayüz sayesinde kullanıcılar istedikleri parametreleri, seçme mekanizmasını ve test işlevini, seçtikleri ÇAGA yöntemine uygulayarak, başarımlarını ölçebilmektedirler.

Anahtar Sözcükler: Genetik Algoritmalar, DOPGA, DOPGA+, DOPGA2, İyileştirme, Puan Ölçekleme, Etkin Elitizm, Ödül ve Ceza.

MULTI-OBJECTIVE GENETIC ALGORITHMS: FUNDAMENTALS AND APPLICATIONS

ABSTRACT

In this study, the genetic algorithm (a population-based optimization method) has been investigated and improved. The main goal of the thesis is to improve the fitness assignment and elitism mechanisms of the multi-objective genetic algorithms (MOGAs). The proposed two MOGAs (DOPGA+, DOPGA 2) use the definition of domination power of the individuals. Hence, they do not need extra parameter for promoting diversity. The proposed methods are derived from the existing DOPGA method. The proposed methods have been compared with some well-known methods (SPEA and SPEA2) in terms of diversity and convergence properties. Furthermore, fitness assignment capabilities of some existing methods (NSGA and SPEA) have been modified and improved and the comparative simulation results have been reported.

To be able to change the selection pressures of the individuals before the selecting mechanism, a gamma correction based fitness scaling method (GCFS) is proposed in order to improve the performance of the MOGAs. This method is embedded into the single objective GAs and newly proposed MOGAs using different gamma values. As a result, the convergence ability of MOGAs with GCFS shown to be improved. The presence of elitism can also significantly improve the performance of MOGAs. The classical holding/sending back type passive elitism mechanism is replaced by the proposed holding/exciting and sending back type effective elitism mechanism (EFE). It is shown that the convergence performance of MOGAs can slightly be improved by using the EFE. The reward and punishment concept is introduced to measure fitness assignment capabilities of MOGAs with the help of two proposed metrics. How much useful information can be generated and passed into the selection mechanism by MOGA methods can now be determined heuristically and statistically.

Finally, a graphical user interface (MATGAT) is designed for easy-use. Users can measure the performance of selected MOGAs by implementing any parameters, any selection mechanism and any test functions with MATGAT software.

Key Words: Genetic Algorithms, DOPGA, DOPGA+, DOPGA2, Modification, Fitness Scaling, Effective Elitism, Reward and Punishment.

TEŞEKKÜR

Bu tezin ortaya çıkmasında ve hazırlanmasında en büyük desteği veren değerli hocam Yrd.Doç.Dr. İlyas EMİNOĞLU' na çok teşekkür ederim. Doktora tez izleme komitesinde yer alan değerli hocalarım Yrd.Doç.Dr. Hatice SEZGİN ve Yrd.Doç.Dr. Mustafa ULUTAŞ' a da ilgi ve yönlendirmelerinden dolayı çok teşekkür ederim. Tezde yer alan yöntemlerin programlanmasında desteğini esirgemeyen değerli arkadaşım Hanife USTA' ya da çok teşekkür ederim.

Doktora çalışmam boyunca desteklerini esirgemeyen, var olma nedenlerim eşim, oğlum ve aileme de çok teşekkür ederim. Ayrıca çalışma boyunca her türlü konuda yardımcı olan arkadaşlarıma da teşekkürü bir borç bilirim.

Bu doktora tezi, Ondokuz Mayıs Üniversitesi Bilimsel Araştırma Projeleri fonu tarafından **2007-BAP-MF-119** numaralı proje olarak desteklenmiştir, bu projeye katkısı olan herkese sonsuz teşekkürlerimi sunuyorum.

İÇİNDEKİLER

	Sayfa
1. GİRİŞ	1
2. GENEL BİLGİLER	7
2.1. Optimizasyon (En İyileme) Nedir ?	7
2.2. Topluluk Tabanlı Optimizasyon Yöntemleri	9
2.2.1. Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO)	9
2.2.2. Karınca Kolonisi Optimizasyonu (Ant Colony Optimization, ACO)	13
2.2.3. Farksal Gelişim (Differential Evolution-DE)	17
2.2.4. Benzetimli Tavlama (Simulated Annealing-SA)	19
2.2.5. Diğer Yöntemler	21
2.3. Genetik Algoritmalar	21
2.3.1. Klasik Optimizasyon Yöntemlerinden Farkları	25
2.3.2. Temel İşlevleri	26
2.3.2.1. Bireylerin (ya da Çözüm Önerilerinin) Kodlanması	26
2.3.2.2. Seçme Mekanizması	26
2.3.2.3. Çaprazlama İşlevi	30
2.3.2.4. Mutasyon İşlevi	31
2.3.2.5. Elitizm (Bellek) Mekanizması	32
2.3.3. Tek Amaçlı Genetik Algoritmalar	33
2.3.3.1. Çözüm Adaylarının (Bireylerin) Gösterilimi	34
2.3.4. Çok Modluluk ve Çeşitlilik Artırıcı Yöntemler	50
2.3.4.1. Paylaşım (Sharing) Yöntemi	51
2.3.4.2. Ayıklama (Clearing) Yöntemi	52
2.3.4.3. Eşleşme Kısıtlaması (Mating Restriction)	54
2.3.5. Çok Amaçlı Genetik Algoritmalar (ÇAGA)	55
2.3.5.1. Optimizasyon Parametreleri veya Karar Değişkenleri	55
2.3.5.2. Amaç İşlevi	56

2.3.5.3. Baskınlık Kavramı ve Pareto-Optimallığı	57
2.3.5.4. ÇAGA' ların Temel Bileşenleri	59
2.3.5.5. Arama ve Karar Verme	61
2.3.6. Uygulama Alanları	62
3. MATERYAL VE YÖNTEM	65
3.1. Pareto Tabanlı Olmayan ÇAGA Yöntemleri	65
3.1.1. Sabit Ağırlıklı Amaç İşlevi (Constant Weight Objective Function, CWOFF)	66
3.1.2. Değişken Ağırlıklı Amaç İşlevi (Variable Weight Objective Function, VWOJ)	70
3.1.3. Vektör Hesaplamalı Genetik Algoritma (Vector Evaluated Genetic Algorithm, VEGA)	72
3.2. Pareto Tabanlı ÇAGA Yöntemleri	74
3.2.1. MOGA (Multiple Objective Genetic Algorithm, Çok Amaçlı Genetik Algoritma)	75
3.2.2. NSGA (Non-Dominated Sorting Genetic Algorithm, Bastırılmayan Sıralamalı Genetik Algoritma)	78
3.2.3. SPEA (Strength Pareto Evolutionary Algorithm, Pareto' nun Gücü Evrimsel Algoritma)	80
3.2.3.1. Pareto Bireylerin Puanlanması	81
3.2.3.2. Pareto Olmayan Bireylerin Puanlanması	81
3.2.3.3. SPEA Elitizm Mekanizması	82
3.2.4. SPEA2 (Strength Pareto Evolutionary Algorithm 2, Pareto' nun Gücü Evrimsel Algoritma 2)	83
3.2.4.1. SPEA2 Puanlama Algoritması	83
3.2.4.2. SPEA2 Arşiv Yönetimi	86
3.2.5. Diğer Yöntemler	87
3.3. Çok Amaçlı Genetik Algoritma Test İşlevleri	88
3.3.1. SCH1 Test İşlevi	89
3.3.2. SCH2 Test İşlevi	90

3.3.3. FON Test İşlevi	91
3.3.4. POL Test İşlevi	92
3.3.5. Sık Kullanılan Diğer Test İşlevleri	93
3.3.5.1. ZDT1 Test İşlevi	94
3.3.5.2. ZDT2 Test İşlevi	95
3.3.5.3. ZDT3 Test İşlevi	95
3.3.5.4. ZDT4 Test İşlevi	96
3.3.5.5. ZDT5 Test İşlevi	97
3.3.5.6. ZDT6 Test İşlevi	99
3.4. Çok Amaçlı Genetik Algoritma Başarım Ölçütleri	102
3.4.1. Yakınsama İçin Kullanılan Başarım Ölçütleri	105
3.4.1.1. Hata Oranı (Error Ratio, ER)	106
3.4.1.2. Küme Kapsama Ölçütü (Set Coverage Metric, C)	107
3.4.1.3. Nesilsel Mesafe (Generational Distance, GD)	108
3.4.1.4. Maksimum Pareto-Optimal Cephe Hatası (Maximum Pareto-Optimal Front Error, MFE)	109
3.4.2. Dağılım veya Çeşitlilik İçin Kullanılan Başarım Ölçütleri.....	110
3.4.2.1. Aralık (Spacing)	110
3.4.2.2. Yayılım (Dağılım, Spread)	111
3.4.2.3. Maksimum Yayılım (Maximum Spread)	113
3.4.3. Yakınsamayı ve Dağılımı Birlikte Hesaplayan Ölçütler	114
3.4.3.1. Hiperküp (Hypervolume)	115
3.4.3.2. Başarım Yüzeyi Temelli İstatistiksel Ölçüt (Attainment Surface Based Statistical Metric)	116
3.4.3.3. Ağırlıklandırma Ölçütü (Weighted Metric)	118
3.4.3.4. Bastırılmayan Hesap Ölçütü (Non-Dominated Evaluation Metric)	119
4. BULGULAR	120
4.1. Çok Amaçlı Genetik Algoritma Yöntem Önerileri	121

4.1.1. DOPGA (DOmination Power of an individual Genetic Algorithm, Bireyin Baskınlık Gücü Genetik Algoritma)	121
4.1.1.1. DOPGA Puanlama Mekanizması	121
4.1.1.2. DOPGA Elitizm Mekanizması	125
4.1.1.3. DOPGA ve SPEA Yöntemlerinin Görsel Karşılaştırması ...	127
4.1.2. DOPGA+ (DOmination Power of an individual Genetic Algorithm +, Bireyin Baskınlık Gücü Genetik Algoritma +)	131
4.1.2.1. DOPGA+ Puanlama Mekanizması	131
4.1.3. DOPGA2 (DOmination Power of an individual Genetic Algorithm 2, Bireyin Baskınlık Gücü Genetik Algoritma 2)	134
4.1.3.1. DOPGA2 Arşiv Yönetimi	134
4.2. Çok Amaçlı Genetik Algoritma Yöntemlerinin Puanlama Mekanizmalarının İyileştirmeleri	136
4.2.1. <i>k</i> . En Yakın Komşuluk Yöntemi Kullanılarak Oluşturulan Yoğunluk Bilgisi	136
4.2.2. Baskınlık Gücü Kavramı	137
4.2.3. NSGA İyileştirmesi (NSGAmoD)	139
4.2.4. SPEA' nın Birinci İyileştirmesi (SPEAmoD1)	140
4.2.5. SPEA' nın İkinci İyileştirmesi (SPEAmoD2)	141
4.2.6. SPEA' nın Üçüncü İyileştirmesi (SPEAmoD3)	142
4.2.7. SPEA' nın Dördüncü İyileştirmesi (SPEAmoD4)	143
4.2.8. Benzetim Sonuçları	145
4.2.9. Genel Sonuçlar ve Değerlendirme	146
4.3. Gama Düzeltmesi ile Puan Ölçekleme Yöntemi (GDPÖ) ve Çok Amaçlı Genetik Algoritmaların Başarımına Etkisi	146
4.3.1. Gama Düzeltmesi ile Puan Ölçekleme Yöntemi (GDPÖ)	147
4.3.2. Uyarlanabilir (Adaptif) Gama Düzeltmesi ile Puan Ölçekleme (U_GDPÖ) Yöntemi	150
4.3.3. Tek Amaçlı GA' larda GDPÖ' nün Etkisinin İncelenmesi ve Benzetimler	152
4.3.4. ÇAGA' larda GDPÖ' nün Etkisinin İncelenmesi ve Benzetimler ...	154
4.3.4.1. Sabit Gama Değerleri Kullanıldığında Benzetim Sonuçları .	155

4.3.4.2. U_GDPÖ Kullanıldığında Benzetim Sonuçları	163
4.3.4.3. Turnuva Seçimi (TS) Kullanıldığında Benzetim Sonuçları ..	164
4.3.4.4. 300 Adım Çalıştırılma ile Elde Edilen Sonuçlar	165
4.3.5. Genel Sonuçlar ve Değerlendirme	169
4.4. Çok Amaçlı Genetik Algoritmalar İçin Etkin Elitizm Mekanizması	
Önerisi	170
4.4.1. Etkin Elitizm Mekanizması (Effective Elitism Mechanism, EFE) ...	172
4.4.2. Benzetim Sonuçları	175
4.4.3. Sonuçlar ve Değerlendirme	182
4.5. ÇAGA Yöntemlerinin Sıralama Yeteneklerinin Ölçülmesi İçin İki Yeni	
Ölçüt Önerisi	183
4.5.1. Ceza ve Ödül Ölçütleri	185
4.5.2. Benzetim Sonuçları	190
4.5.3. Sonuçlar ve Değerlendirme	193
4.6. ÇAGA' lar İçin Kullanıcı Arayüzü Tasarımı	195
5. SONUÇ VE ÖNERİLER	199
6. KAYNAKLAR	205
7. ÖZGEÇMİŞ	216
8. EK	217

SİMGE VE KISALTMALAR LİSTESİ

SİMGELER

bbg	Bağıl Baskınlık Gücü
C	Küme Kapsama Ölçütü
Δ	Yayılım/Dağılım Ölçütü
d(i,j)	i ve j Bireyleri Arasındaki Öklit Mesafesi
D	Yoğunluk
F	Son Puan
f_{si}	Paylaşılmış Puan
γ	Puan Ölçekleme Üs Değeri
gp	Gerçek Puan
<i>l</i>	Dizi Boyutu
m	Amaç Fonksiyonu Sayısı
N	Topluluk Boyutu
$\bar{N}$	Arşiv Boyutu
nc	Yarıçap (niche) Sayısı
P	Ana Topluluk
<i>P</i>	Ceza Ölçütü
P^a	Arşiv veya İkincil Topluluk
P[*]	Pareto-optimal Küme
PF[*]	Pareto-optimal Cephe
p_c	Çaprazlama Olasılığı
p_i	Seçilme Olasılığı
p_m	Mutasyon Olasılığı
Q	ÇAGA Tarafından Bulunan Pareto Bireyler Kümesi
q_i	Toplam (kümülatif) Olasılık
R	Ham Puan
<i>R</i>	Ödül Ölçütü
Sh	Paylaşım Fonksiyonu
σ_s	Paylaşım Yarıçapı

$\sigma_{\text{eşleşme}}$	Eşleşme Yarıçapı
tbg	Toplam Baskınlık Gücü
x	Parametre veya değişkenler
yp	Yapay Puan
w	Ağırlıklar

KISALTMALAR

ACO	Ant Colony Optimization
BG	Baskınlık Gücü
CWOF	Constant Weight Objective Function
ÇAGA	Çok Amaçlı Genetik Algoritma
DE	Differential Evolution
DOPGA	DOmination Power of individual Genetic Algorithm
DOPGA+	DOmination Power of individual Genetic Algorithm +
DOPGA2	DOmination Power of individual Genetic Algorithm 2
EFE	EFfective Elitism
ER	Error Ratio
FON	FONseca' nın test işlevi
GA	Genetik Algoritma
GD	Generational Distance
GDPÖ	Gama Düzeltmesi ile Puan Ölçekleme
GUI	Graphic User Interface
HV	Hiperküp (HyperVolume)
KV	Karar Verici
MATGAT	MATlab Genetic Algorithm Toolbox
MFE	Maximum Pareto-Optimal Front Error
MOGA	Multiple Objective Genetic Algorithm
NFL	No Free Lunch
NP	Normalize Puan
NPF	Normalize Puan Farkı
NSGA	Non-Dominated Sorting Genetic Algorithm
NSGAmo	Non-Dominated Sorting Genetic Algorithm Modification
PAGA	Preference Articulation Genetic Algorithm
PAM	Puan Atama Mekanizması
POL	POLoni' nin test işlevi
PÖY	Puan Ölçekleme Yöntemi
PSO	Particle Swarm Optimization

RBF	Radial Basis Functions
RWS	Roulette Wheel Selection
SA	Simulate Annealing
SCH1	SCHaffer' in 1. test işlevi
SCH2	SCHaffer' in 2. test işlevi
SPEA	Strength Pareto Evolutionary Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm 2
SPEAmod1	Strength Pareto Evolutionary Algorithm Modification 1
SPEAmod2	Strength Pareto Evolutionary Algorithm Modification 2
SPEAmod3	Strength Pareto Evolutionary Algorithm Modification 3
SPEAmod4	Strength Pareto Evolutionary Algorithm Modification 4
SUS	Stochastic Universal Sampling
TP	Toplam Puan
TS	Turnuva Seçimi
U_GDPÖ	Uyarlanabilir Gama Düzeltmesi ile Puan Ölçekleme
VEGA	Vector Evaluated Genetic Algorithm
VWOJ	Variable Weight Objective Function
ZDT	Zitzler-Deb-Thiele

ŞEKİLLER LİSTESİ

	Sayfa
Şekil 2.1. Örnek bir optimizasyon problemi (otomobil satın alma)	8
Şekil 2.2. Örnek bir kuş sürüsü	10
Şekil 2.3. Bir karınca kolonisinin davranışı	14
Şekil 2.4. Ant-Q program kodu	15
Şekil 2.5. Benzetimli tavlama program kodu	20
Şekil 2.6. GA' ların genel yapısı	24
Şekil 2.7. (a) Rulet çarkı seçimi (RWS), (b) Stokastik evrensel örnekleme (SUS)	28
Şekil 2.8. İkili turnuva seçimi örneği (en küçükleme problemi)	29
Şekil 2.9. Tek noktalı çaprazlama örneği	30
Şekil 2.10. Çok noktalı çaprazlama örneği	31
Şekil 2.11. Bit-tabanlı mutasyon örneği	32
Şekil 2.12. Amaç işlevi örneği (Rastrigin işlevi)	33
Şekil 2.13. Tek amaçlı GA uygulaması (Örnek-1)	46
Şekil 2.14. Tek amaçlı GA uygulaması (Örnek-2)	47
Şekil 2.15. Tek amaçlı GA uygulaması (Örnek-3)	48
Şekil 2.16. Tek amaçlı GA uygulaması (Örnek-4)	49
Şekil 2.17. Rasgele sayı üretme (Örnek-5)	50
Şekil 2.18. Yedi adet RBF işlevi ile oluşturulan çok modlu bir amaç işlevinin paylaşım yöntemi ile en iyi değerinin bulunması	52
Şekil 2.19. Yedi adet RBF işlevi ile oluşturulan çok modlu bir amaç işlevinin ayıklama yöntemi ile en iyi değerinin bulunması	54
Şekil 2.20. Parametre uzayı ve amaç uzayının gösterilimi	56
Şekil 2.21. Çizgili bölge Pareto-optimal cepheyi göstermektedir	58
Şekil 2.22. (a) Başlangıç topluluğu ve Pareto cephesi, (b) İstenilen GA durumu	59
Şekil 2.23. Çok amaçlı genetik algoritmaların sınıflandırılması	60
Şekil 3.1. CWOFF, arama uzayında bir arama yönüne sahiptir ve yalnızca A gibi bir birey bulabilir	67
Şekil 3.2. CWOFF algoritması	68

Şekil 3.3. $F(x) = 0.5f(x) + 0.5g(x)$ için CWOOF sonucu	68
Şekil 3.4. $F(x) = 0.25f(x) + 0.75g(x)$ için CWOOF sonucu	69
Şekil 3.5. $F(x) = 0.75f(x) + 0.25g(x)$ için CWOOF sonucu	69
Şekil 3.6. VWOJ Pareto çözümlerini bulmak için farklı yönlerde arama yapar	70
Şekil 3.7. VWOJ algoritması	71
Şekil 3.8. VWOJ örneği	71
Şekil 3.9. (a) VEGA blok diyagramı, (b) VEGA' nın arama yönü	72
Şekil 3.10. VEGA algoritması	73
Şekil 3.11. VEGA örneği	74
Şekil 3.12. (a) MOGA tarafından atanan ham puanlar (b) MOGA tarafından atanan ortalama puanlar (c) MOGA tarafından atanan ölçeklenmiş veya son puanlar	76
Şekil 3.13. MOGA puan atama algoritması	77
Şekil 3.14. (a) NSGA ham puanları, (b) NSGA paylaşılmış puanları	79
Şekil 3.15. NSGA puan atama algoritması	80
Şekil 3.16. (a) SPEA ile Pareto bireylerin puanlanması (b) SPEA ile tüm bireylerin puanlanması	82
Şekil 3.17. SPEA çeşitlilik işlemi	83
Şekil 3.18. (a) SPEA2 ham puanları, (b) SPEA2 son puanları	86
Şekil 3.19. SPEA2 arşiv güncellemesi	87
Şekil 3.20. SCH1 test işlevi	89
Şekil 3.21. SCH2 test işlevi	90
Şekil 3.22. FON test işlevi	91
Şekil 3.23. POL test işlevi	92
Şekil 3.24. ZDT1 test işlevinin Pareto-optimal cephesi	94
Şekil 3.25. ZDT2 test işlevinin Pareto-optimal cephesi	95
Şekil 3.26. ZDT3 test işlevi	96
Şekil 3.27. ZDT4 test işlevinin Pareto-optimal cephesi	97
Şekil 3.28. ZDT5 test işlevinin Pareto-optimal cephesi	98
Şekil 3.29. ZDT6 test işlevi	99

Şekil 3.30. (a) Çok amaçlı optimizasyonda ulaşılmaması istenen iki amaç (b) İdeal bir Pareto-optimal çözüm kümesi	103
Şekil 3.31. (a) Yakınsama iyi, ancak dağılım kötü (1. Algoritma) (b) Yakınsama kötü, ancak dağılım iyi (2. Algoritma)	104
Şekil 3.32. (a) A algoritması B algoritmasından daha iyi (b) A ve B algoritmalarını karşılaştırmak zor	105
Şekil 3.33. <i>Kareler:</i> Pareto-optimal cepheyi oluşturan bireyler (P^*) ve <i>Daireler:</i> ÇAGA yöntemi tarafından bulunan Pareto bireyler (Q)	106
Şekil 3.34. Bireyler arasındaki mesafeler	112
Şekil 3.35. En büyük yayılım ölçütü	114
Şekil 3.36. Pareto bireyler tarafından oluşturulan hiperküp	115
Şekil 3.37. Pareto bireyler tarafından oluşturulan başarımlar yüzeyi	116
Şekil 3.38. (a) Bir ÇAGA' nın üç farklı adımında bulunan bastırılmayan bireyler, (b) Bulunan üç farklı cepheye ilişkin başarımlar yüzeyleri	117
Şekil 3.39. Tipik bir kesişim hattı ve kesişim noktaları. Bu noktalara göre frekans dağılımı veya histogram	118
Şekil 3.40. Yakınsama ve çeşitliliğin birlikte değerlendirilmesi	119
Şekil 4.1. (a) DOPGA ham puanları, (b) DOPGA son puanları	123
Şekil 4.2. DOPGA puanlama algoritması	124
Şekil 4.3. DOPGA elitizm mekanizmasının blok diyagramı	126
Şekil 4.4. Arşivdeki Pareto bireylerin azaltılma adımları	126
Şekil 4.5. DOPGA genel algoritması	127
Şekil 4.6. SPEA'da, A ve C birbirinden ayırt edilemez	128
Şekil 4.7. DOPGA' da, A ve C birbirinden ayırt edilebilir	128
Şekil 4.8. SPEA; F , $F1$ ve $F2$ bireylerine aynı puanı atar, çünkü bu bireyler yalnızca B bireyi tarafından bastırılmaktadır	129
Şekil 4.9. DOPGA' da ise F , $F1$ ve $F2$ bireyleri aynı alt topluluğa ait olmadıkları için farklı puanlara sahiptirler	129
Şekil 4.10. DOPGA tarafından bulunan sonuçlar : (a) ZDT1, (b) ZDT2	130
Şekil 4.11. DOPGA tarafından bulunan sonuçlar : (a) ZDT3, (b) ZDT6	130
Şekil 4.12. (a) DOPGA+ ham puanlar, (b) DOPGA+ son puanlar	131
Şekil 4.13. DOPGA+ genel algoritması	132

Şekil 4.14. DOPGA+ tarafından bulunan sonuçlar : (a) ZDT1, (b) ZDT2	133
Şekil 4.15. DOPGA+ tarafından bulunan sonuçlar : (a) ZDT3, (b) ZDT6	133
Şekil 4.16. DOPGA2 genel algoritması	134
Şekil 4.17. DOPGA2 tarafından bulunan sonuçlar : (a) ZDT1, (b) ZDT2	135
Şekil 4.18. DOPGA2 tarafından bulunan sonuçlar : (a) ZDT3, (b) ZDT6	135
Şekil 4.19. NSGAmođ puanlaması (a) İlk adıđ: NSGA puanlaması, (b) İkinci adıđ: yoğunluk bilgisi, (c) NSGAmođ tarafından atanan son puan değeri	140
Şekil 4.20. SPEAmođ1 puanlaması (a) İlk adıđ: baskınlık gücü ile puanlama, (b) SPEA ile Pareto olmayan bireylerin puanlanması	141
Şekil 4.21. SPEAmođ2 puanlaması (a) İlk adıđ: SPEA puanlaması, (b) İkinci adıđ: yoğunluk bilgisi, (c) SPEAmođ2 tarafından atanan son puan değeri	142
Şekil 4.22. SPEAmođ3 puanlaması (a) İlk adıđ: MOGA puanlaması, (b) İkinci adıđ: sanal puanlar, (c) SPEAmođ3 tarafından atanan son puan değeri	143
Şekil 4.23. SPEAmođ4 puanlaması (a) İlk adıđ: MOGA puanlaması, (b) İkinci adıđ: sanal puanlar, (c) SPEAmođ4 tarafından atanan son puan değeri	144
Şekil 4.24. (a) Sayısal görüntü işlemede gama düzeltmesi (b) Genetik algoritmalarda gama düzeltmesi	149
Şekil 4.25. Seçme mekanizmasından önce GDPÖ kullanımı	149
Şekil 4.26. GDPÖ genel algoritması	150
Şekil 4.27. Beş farklı puan dağılım işlevi	151
Şekil 4.28. Uyarlanabilir GDPÖ algoritması	151
Şekil 4.29. Ackley işlevi ($x_1, x_2 \in [-2, 2]$ için)	152
Şekil 4.30. Tek amaçlı GA' da GDPÖ etkisi (Ackley işlevi)	153
Şekil 4.31. Tek amaçlı GA' da GDPÖ etkisi (Rastrigin işlevi)	154
Şekil 4.32. Yakınsama ölçütünün ortalama değeri	157
Şekil 4.33. Dağılım ölçütünün ortalama değeri	158
Şekil 4.34. SPEA yöntemi için gama-yakınsama değeri	159
Şekil 4.35. SPEA2 yöntemi için gama-yakınsama değeri	159
Şekil 4.36. DOPGA yöntemi için gama-yakınsama değeri	160
Şekil 4.37. DOPGA2 yöntemi için gama-yakınsama değeri	160

Şekil 4.38. $\gamma=1$ ve $\gamma=7$ için SPEA yöntemiyle ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyler	161
Şekil 4.39. $\gamma=1$ ve $\gamma=2$ için SPEA2 yöntemiyle ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyler	162
Şekil 4.40. $\gamma=1$ ve $\gamma=3$ için DOPGA yöntemiyle ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyler	162
Şekil 4.41. $\gamma=1$ ve $\gamma=2$ için DOPGA2 yöntemiyle ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyler	163
Şekil 4.42. $\gamma=7$ ile SPEA tarafından bulunan Pareto bireyler:(a)ZDT1,(b)ZDT2	165
Şekil 4.43. $\gamma=2$ ile SPEA2 tarafından bulunan Pareto bireyler:(a)ZDT1,(b) ZDT2	166
Şekil 4.44. $\gamma=3$ ile DOPGA tarafından bulunan Pareto bireyler:(a)ZDT1,(b)ZDT2	166
Şekil 4.45. $\gamma=2$ ile DOPGA2 tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2	167
Şekil 4.46. $U_GDP\ddot{O}$ ile SPEA tarafından bulunan Pareto bireyler:(a) ZDT1, (b) ZDT2	167
Şekil 4.47. $U_GDP\ddot{O}$ ile SPEA2 tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2	168
Şekil 4.48. $U_GDP\ddot{O}$ ile DOPGA tarafından bulunan Pareto bireyler: (a)ZDT1, (b) ZDT2	168
Şekil 4.49. $U_GDP\ddot{O}$ ile DOPGA2 tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2	169
Şekil 4.50. Turboşarj ünitesi	173
Şekil 4.51. Etkin elitizm mekanizmalı ÇAGA blok diyagramı	173
Şekil 4.52. Etkin elitizmde elit bireylerin mutasyon işlevi ile uyarılması	174
Şekil 4.53. O andaki elit bireylerler ile daha önceki adımdaki elit bireylerin çaprazlanması	175
Şekil 4.54. ZDT1 test işlevinde bulunan Pareto bireyler:(a) SPEA,(b)EFE_SPEA	178
Şekil 4.55. ZDT2 test işlevinde bulunan Pareto bireyler:(a) SPEA,(b)EFE_SPEA	178
Şekil 4.56. ZDT1 test işlevinde bulunan Pareto bireyler: a) SPEA2, (b) EFE_SPEA2	179
Şekil 4.57. ZDT2 test işlevinde bulunan Pareto bireyler: (a) SPEA2, (b) EFE_SPEA2	179

Şekil 4.58. ZDT1 test işlevinde bulunan Pareto bireyler: (a) DOPGA, (b) EFE_DOPGA	180
Şekil 4.59. ZDT2 test işlevinde bulunan Pareto bireyler: (a) DOPGA, (b) EFE_DOPGA	180
Şekil 4.60. ZDT1 test işlevinde bulunan Pareto bireyler: (a) DOPGA+, (b) EFE_DOPGA+	181
Şekil 4.61. ZDT2 test işlevinde bulunan Pareto bireyler: (a) DOPGA+, (b) EFE_DOPGA+	181
Şekil 4.62. Sekiz bireylik bir topluluk için ceza işlevi	187
Şekil 4.63. İdeal puan dağılımında ödül değeri sıfır olur	188
Şekil 4.64. Beş farklı topluluğun puan dağılımları	189
Şekil 4.65. Boxplot gösterilimi (a) Ceza ölçütü, (b) Ödül ölçütü	191
Şekil 4.66. Boxplot gösterilimi (a) Ceza ölçütü, (b) Ödül ölçütü	192
Şekil 4.67. MATGAT v1.0 çalıştırılmadan önce ekran görüntüsü	197
Şekil 4.68. MATGAT v1.0 çalıştırıldıktan sonra ekran görüntüsü	198

ÇİZELGELER LİSTESİ

	Sayfa
Çizelge 2.1. Rasgele sayılar ve seçilen bireyler	40
Çizelge 2.2. Ebeveyn bireyler, çaprazlama noktası ve çocuk bireyler	41
Çizelge 2.3. Bazı ÇAGA uygulamaları	63
Çizelge 3.1. ÇAGA test işlevleri	100
Çizelge 3.2. Örnek uygulama	108
Çizelge 4.1. Yakınsama ölçütünün (GD) tüm yöntemler ve iyileştirmeleri için tüm test işlevlerinde elde edilen ortalama değerleri	145
Çizelge 4.2. Dağılım ölçütünün (Δ) tüm yöntemler ve iyileştirmeleri için tüm test işlevlerinde elde edilen ortalama değerleri	145
Çizelge 4.3. Yakınsama ölçütünün (GD) tüm ÇAGA yöntemleri için tüm test işlevlerinde elde edilen ortalama değerleri	156
Çizelge 4.4. Dağılım ölçütünün (Δ) tüm ÇAGA yöntemleri için tüm test işlevlerinde elde edilen ortalama değerleri	157
Çizelge 4.5. U_GDPÖ uygulaması sonucunda hesaplanan yakınsama ölçüt sonuçları	163
Çizelge 4.6. U_GDPÖ uygulaması sonucunda hesaplanan dağılım ölçüt sonuçları	164
Çizelge 4.7. TS-2 / TS-4 için yakınsama ölçüt sonuçları	164
Çizelge 4.8. TS-2 / TS-4 için dağılım ölçüt sonuçları	165
Çizelge 4.9. Yakınsama ölçütünün (GD) ortalama (ilk satırlar) ve varyans (ikinci satırlar) değerleri	177
Çizelge 4.10. Dağılım ölçütünün (Δ) ortalama (ilk satırlar) ve varyans (ikinci satırlar) değerleri	177
Çizelge 4.11. Şekil 4.64' teki puan dağılımlarının ödül değerleri	189
Çizelge 4.12. SPEA ve DOPGA yöntemleri için ölçüt değerleri	191
Çizelge 4.13. SPEA2 ve DOPGA2 yöntemleri için ölçüt değerleri	192

1. GİRİŞ

Genel olarak, optimizasyon, daha iyi bir sonuç bulununcaya kadar olası tüm çözümlerin amaç işlevine göre aranması ve karşılaştırılması işlemidir. Optimizasyon amaç sayısı bakımından; tek amaçlı ve çok amaçlı olmak üzere ikiye ayrılır. Klasik optimizasyon yöntemleri tek bir benzetimde (adımda) tek bir çözümün bulunmasında iyidirler, ancak çok amaçlı optimizasyon problemlerinin çözülmesi için uygun değildirler. Ancak, gerçek dünya problemleri genelde birbirleriyle çelişen bir çok amaçtan oluşurlar. Bu yüzden çok amaçlı optimizasyon yöntemlerinin çözülebilmesi için topluluk tabanlı optimizasyon yöntemleri (Genetik algoritmalar vb.) daha uygundur. Bu yöntemler, tek bir benzetimde (adımda) çok sayıda en iyi çözümü bulabilirler.

Genetik algoritmalar (GA), en iyi çözümü arama işlevini, biyolojik kökenli bazı işlevleri (seçme, çaprazlama, mutasyon, elitizm vb.) taklit ederek yapar. GA, doğal evrimin doğrudan bir benzeşimini kullanır. Genetik algoritmaların diğer optimizasyon yöntemlerinden ve klasik arama yöntemlerinden en önemli farkı, tek bir çözüm yerine her adımda çözümlerden oluşan bir topluluk kullanmasıdır. Her adımda bir çözüm topluluğu kullanıldığı için, GA ile elde edilen sonuçlar da bir çözüm topluluğu olacaktır. Bir problemin potansiyel bir çözümü *birey (kromozom)* olarak adlandırılır. Bireyler parametreleri temsil eder. Bu parametreler ise bir kromozomun *genleri* olarak adlandırılır. Bu genler, ikili sayı veya gerçel sayı dizileri kullanılarak oluşturulabilir. Problem çözümünde bir bireyin ya da kromozomun ne kadar iyi olduğu, amaç işlevinde (kullanıcı tarafından belirlenir) aldığı değerle yakın ilişkili olan *puan (fitness)* değerine göre belirlenir. Evrim kuramına göre iyi bir kromozomdan iyi kromozomlar elde edileceği söylenmektedir. GA’ da da en iyi bireylerden her zaman daha iyi bireyler üretilebileceği beklenmektedir. Pratik bir GA uygulamasında bireylerden oluşan bir topluluk (genellikle rasgele) oluşturulur. Bu topluluğun boyutu, problemde probleme değişebilir. Her bir GA adımında, seçme mekanizması kullanılarak bir alt topluluk seçilir. Seçilen bireylerden oluşan bu alt topluluk, *ebeveynler* veya *eşleşme havuzu* olarak adlandırılır. Ebeveynlerin genleri çaprazlama işlevi kullanılarak karıştırılır ve birleştirilerek bir sonraki nesil oluşturulur. Bireylerin birbirinden farklı olmasını sağlamak ve aynı çözümlerin elde edilmesinden kurtulmak için mutasyon işlevi uygulanır. Ayrıca doğada en güçlülerin yaşamını sürdürmesinden esinlenilerek, GA’ da da her adımda en iyi birey/bireyler saklanır ve bir sonraki topluluğa eklenir. Böylece en

iyi bireylerden en iyi çözümler üretilmesi amaçlanmaktadır. Bu işleme ise *elitizm* adı verilir.

GA' lar, Pareto cephesinin (bulunması istenen çözüm bölgesi) şekline ve sürekli olup olmamasına çok duyarlı değillerdir. Bu yapılarından dolayı çok amaçlı optimizasyon için önemli/uygun bir hesapsal yapay zeka yöntemidir.

Holland' ın 1984 yılında adını koymasından sonra, GA' lar konusunda 1990' lara kadar çok fazla çalışma yapılmamıştır. Ancak Goldberg' in GA' yı gerçek bir probleme uygulaması ve çığır açan kitabı sayesinde GA' lar konusuna olan ilgi giderek artmıştır. Her geçen yıl, GA' lar konusunda yeni çalışmalar yapılmıştır. Bu çalışmalar; GA' nın başarımının artırılması, GA yöntemlerinin karşılaştırılması, test edilmesi vb. durumlarla ilgilidir. Ayrıca GA' lar diğer optimizasyon yöntemleri ile hibrit hale getirilerek, gerçek dünya problemlerine uygulanmıştır.

GA çözümleri ararken, çok fazla sayıda işleme ve çok fazla zamana ihtiyaç duymaktadır. Evrimsel algoritmaların temelleri 1950' lere kadar uzanmasına rağmen, son 15 yılda bu konudaki çalışmalarda önemli bir artış olmuştur. Bunun nedeni, düşük maliyetli ve güçlü bilgisayarların üretilmesidir. Bu sayede GA işlemleri daha hızlı bir şekilde yapılabilmektedir.

Bu çalışmada, gösterilimi daha kolay olduğu için, genellikle iki amaçlı optimizasyon problemleri kullanılmıştır. Çünkü boyut sayısı arttığında çözümlerin gösterilmesi zorlaşmaktadır. Ayrıca, kullanılan optimizasyon problemlerinin tamamı en küçükleme problemidir. Ancak şunu da belirtmek gerekir ki, ikililik prensibine göre, en küçükleme problemlerini en büyükleme problemlerine ve en büyükleme problemlerini de en küçükleme problemlerine dönüştürmek mümkündür.

Tezin ana bölümü çok amaçlı genetik algoritmalar (ÇAGA). ÇAGA yöntemlerinin iki ana özelliği sağlamaları gerekir: *i)* Pareto-optimal (istenilen) çözümlerden oluşan cepheye mümkün olduğunca iyi yakınsama sağlanmalıdır, *ii)* bulunan Pareto bireyler bu cephe üzerinde düzgün olarak dağılmalıdır. Literatürde yer alan ÇAGA yöntemleri, GA' ların çalışma biçimi hakkında çok fazla fikir sahibi olunmadan önerilmişlerdir. GA' ların nasıl çalıştığı konusu analitik olarak bilinmemektedir. Başlangıç topluluğunun bir nesilden diğerine nasıl hareket ettiği sorusu da yanıtlanması gereken temel bir sorudur. Ayrıca, hangi birey çiftlerinin çaprazlanması ile daha iyi bireyler elde edilebileceği de yanıtlanması gereken önemli bir konudur.

ÇAGA' larda bir diğer önemli konu ise puanlama mekanizmasından seçme mekanizmasına nitelikli bilgi aktarılabilme yeteneğidir. Nitelikli bilgi ile kastedilen bir topluluktaki bireylerin en iyiden en kötüye doğru tekil bir biçimde sıralanıp sıralanamadığıdır. Eğer bu sağlanabilirse, ÇAGA' nın başarımının artması beklenmektedir. Çünkü bu şekilde, seçme mekanizması en iyi bireyleri ayırt etmekte zorlanmayacaktır. Bu amaçla bir ÇAGA yöntemi tarafından bir topluluktaki bireylerin ne kadar iyi puanlandığının belirlenmesi önemli olacaktır.

Şu ana kadar, literatürde mühendislik hesaplamalarından çok mühendislik sezgilerine dayanılarak oluşturulmuş birçok ÇAGA yöntemi önerilmiştir. Temel olarak, bu yöntemler birbirlerinden puan atama mekanizmaları açısından farklılıklar göstermektedirler. Doğal olarak, çok sayıda ÇAGA yöntemi bulunmaktadır ve bu yöntemlerle ilgili ayrıntılı bilgilere [Coello Coello, 1999; Deb, 2001; Coello Coello ve ark., 2007] kaynaklarından ulaşılabilir. Çok sayıda ÇAGA yönteminin olması, puan atama konusunun çok amaçlı genetik algoritmalar için ucu açık bir soru olduğunu göstermektedir. Bu kadar çok sayıda PAM (puan atama mekanizması) olmasının bir başka açıklaması da, PAM yöntemleri ile çok amacın (veya çok boyutun) tek bir puan (fitness) değerine (ya da tek bir boyuta) dönüştürülmesidir ve bu tür bir boyut azaltımının basit ve tek bir yolu yoktur.

Elitizm mekanizması ÇAGA' ların başarımına önemli ölçüde etki etmektedir. Bu mekanizmada yapılabilecek iyileştirmeler başarımda artışa neden olacaktır. Çünkü en iyi bireylerin saklanması ve çaprazlama işlevine tabi tutulması ile oluşacak bireylerin de daha iyi olması beklenmektedir.

ÇAGA yöntemlerinin puan atama mekanizmaları üzerinde bir takım iyileştirmeler yapılarak başarımlarında artış sağlanabilir. Bu işlem yapılırken, yöntemlerin iyi olan yönleri de kullanılabilir.

Tezin ilk bölümünde optimizasyon ve özellikle çok amaçlı optimizasyon hakkında genel bilgiler verilmektedir. Çok amaçlı optimizasyon için kullanılan topluluk tabanlı bazı optimizasyon yöntemleri (PSO, DE, SA, ACO) kısaca incelenmektedir. Tezin ana konusu olan genetik algoritmalar bölümü ise daha ayrıntılı olarak ele alınmaktadır.

Genetik algoritmalar bölümünde, ilk olarak, GA' ların klasik optimizasyon yöntemlerinden farkları verilmektedir. İkinci olarak, genetik algoritmaların temel işlevleri olan çaprazlama, mutasyon ve seçme işlevleri incelenmektedir. Daha sonra tek

amaçlı bir optimizasyon problemi GA yardımıyla adım adım çözülmekte ve GA' nın daha iyi anlaşılması sağlanmaktadır. GA' larda yerel en iyilere takılma önemli sorunlardan biridir, bu sorunu gidermek amacıyla çeşitli yöntemler önerilmiştir. Bu amaçla önerilen paylaşım, ayıklama ve eşleşme kısıtlaması yöntemleri incelenmekte ve birer örnekle değerlendirme yapılmaktadır. Bir sonraki bölümde ÇAGA' ların temel kavramları ayrıntılı olarak anlatılmaktadır. Son olarak ise, ÇAGA' larla ilgili literatürde yer alan bazı gerçek dünya uygulamaları listelenmiştir.

Tezin ikinci bölümünde, literatürde yer alan Pareto tabanlı ve Pareto tabanlı olmayan ÇAGA yöntemleri ayrıntılı olarak incelenmekte, algoritmaları verilmekte ve uygulama örnekleri sunulmaktadır. Daha sonra ÇAGA yöntemlerinin karşılaştırılmalarında ve başarımlarının test edilmesinde kullanılan test işlevleri incelenmekte, grafikleri ve Pareto cepheleri verilmektedir. ÇAGA yöntemlerinin başarımlarının ölçülmesi için literatürde yer alan bazı başarımların ölçütleri de ayrıntılı olarak incelenmektedir.

Bulgular bölümünde ise yeni önerilen ÇAGA yöntemleri, ÇAGA yöntem iyileştirmeleri, puan ölçekleme yöntemi, etkin elitizm mekanizması ve başarımların ölçütleri ayrıntılı olarak anlatılmaktadır. Bu bölümde, ilk olarak daha önce önerilen **DOPGA** (D^Omination Power of an individual Genetic Algorithm) yöntemi [Eminoğlu, 2003] derinlemesine incelenmektedir. **DOPGA** yöntemi Pareto tabanlı ve elitist bir yöntemdir. Baskınlık gücü kavramını kullanarak bir topluluktaki bireylere iki adımda puan atar. Çeşitlilik için harici katsayı tahminine gereksinim duymaz, bu bilgi puanlama mekanizmasının içine gömülüdür. Bu yöntemin elitizm mekanizmasında bazı değişiklikler (ilk topluluk ile ikincil topluluk birleştirilmiştir) yapılmış ve etkinliği artırılmıştır. DOPGA yöntemi daha önce, başarımların ölçütleri kullanılarak literatürde yer alan ÇAGA yöntemleri ile karşılaştırılmamıştır. Bu bölümde DOPGA, sık kullanılan test işlevleri ve başarımların ölçütleri kullanılarak literatürde sık kullanılan ÇAGA yöntemleriyle (SPEA ve SPEA2) karşılaştırılmaktadır. Daha sonra, DOPGA yönteminde bazı değişiklikler yapılarak iki yeni yöntem önerisi (DOPGA+, DOPGA2) sunulmaktadır. DOPGA' da puanlama mekanizması k . en yakın komşuluk bilgisi eklenerek değiştirilmiş ve **DOPGA+** yöntemi önerilmiştir. DOPGA+ yönteminin elitizm mekanizması değiştirilerek ise **DOPGA2** yöntemi önerilmiştir. Tüm bu yöntemler çeşitli test işlevleri üzerinde çalıştırılmış ve sonuçlar verilmiştir. Bu yöntemler literatürde sık kullanılan SPEA ve SPEA2 yöntemleri ile karşılaştırılmışlardır.

ve sonuç olarak daha iyi yakınsama ve dağılım özelliklerine sahip oldukları görülmüştür. SPEA ve SPEA2 yönteminin karşılaştırmalarda kullanılmak üzere seçilmesinin nedenleri şunlardır: *i)* her ikisi de Pareto tabanlıdır, *ii)* her ikisi de harici bir katsayı kullanmadan çeşitlilik bilgisini puanlama mekanizmasının içine gömebilme yeteneğine sahiptir, *iii)* her ikisi de elitisttir, *iv)* her ikisi de son zamanlarda farklı yapay test işlevleri kullanılarak farklı ÇAGA yöntemleriyle karşılaştırılmışlardır.

ÇAGA yöntemlerinin puanlama mekanizmalarında bir takım iyileştirmeler yapılarak başarımları artırılmıştır. Bu iyileştirmeler NSGA ve SPEA yöntemleri üzerinde yapılmıştır. İyileştirmelerde, *k*. en yakın komşuluk ve baskınlık gücü kavramları kullanılmıştır. Tüm bu iyileştirmeler ve orjinal yöntemler çeşitli test işlevleri üzerinde çalıştırılmış ve sonuç olarak yapılan iyileştirmelerin orjinal yöntemlerden daha iyi olduğu belirlenmiştir.

Puan ölçekleme genellikle tek amaçlı GA' larda kullanılan bir yöntemdir. Bu yöntemle, seçme mekanizmasından önce bireylerin seçilme baskısı değiştirilmekte ve böylece başarımlar artırılmaktadır. Bu tezde, yeni bir puan ölçekleme yöntemi önerilerek, hem tek amaçlı hem de çok amaçlı genetik algoritmalara uygulanmıştır. Bu yöntem ***gama düzeltmesi ile puan ölçekleme (GDPÖ)*** olarak adlandırılmıştır. Sonuç olarak; hem tek amaçlı GA' larda hem de literatürde yer alan ve önerilen ÇAGA yöntemlerinde başarımın, fazladan hesapsal bir yük getirmeden artırılabilceği gözlenmiştir. GDPÖ yönteminin etkisi çeşitli seçme mekanizmaları üzerinde de değerlendirilmiştir. Sonuç olarak, GDPÖ' nün yalnızca puan orantılı bir seçme mekanizması (RWS veya SUS) kullanıldığı zaman etkili olduğu, turnuva seçimi kullanıldığı zaman etkisiz olduğu belirlenmiştir.

Elitizm mekanizmasının GA' ların başarımını önemli ölçüde etkilediği daha önceden yapılan çalışmalarla belirlenmiştir. Literatürde önerilen elitizm mekanizmaları *pasif sakla/aktar* bir yapıdadır ve elit bireyler hiçbir uyartıma tabi tutulmadan bir sonraki topluluğa aktarılmaktadır. Bu çalışmada ise bu yapı, *sakla/uyar/aktar* bir yapı ile değiştirilerek ***etkin elitizm mekanizması*** önerilmiştir. Elit bireyler çözüme en yakın bireylerdir ve bunların bir takım işlemlerle (çaprazlama ve mutasyon) uyarılmasıyla kendilerinden daha iyi bireyler elde edilme olasılığı diğer bireylere oranla daha fazladır. Etkin elitizm mekanizması bu saptamadan yola çıkılarak oluşturulmuştur. Etkin elitizm mekanizması ÇAGA yöntemlerine uygulanmış ve orjinal yani pasif elitizm

mekanizmalı biçimleri ile karşılaştırılmışlardır. Sonuç olarak, etkin elitizmin ÇAGA yöntemlerinin başarımını artırdığı gözlenmiştir.

ÇAGA yöntemlerinin ne kadar nitelikli puanlama yaptıklarının ve böylece seçme mekanizmasına ne kadar nitelikli bilgi gönderebildiklerinin belirlenmesi için ise **ceza ve ödül** başarımlar ölçütleri önerilmektedir. Ceza ölçütü, ÇAGA yöntemlerinin bir topluluktaki bireylere birbirinden farklı puanlar atayıp atamadıklarına, yani bireyleri en iyiden en kötüye doğru tekel olarak sıralayıp sıralamadıklarına bakılarak belirlenir. Ödül ölçütü ise atanan puan değerlerinin ne kadar düzgün dağılıp dağılmadığına bakılarak belirlenir. Cezası ve ödülü düşük olan yöntem, sıralama yeteneği açısından daha iyidir. Benzetimler sonucunda, DOPGA yönteminin SPEA yönteminden, DOPGA2 yönteminin ise SPEA2 yönteminden sıralama yeteneği açısından daha iyi olduğu görülmüştür. Ceza ve ödül ölçütleri ile bireylerin Pareto cephesine dağılımları arasında doğrusal bir ilişki vardır, yani ceza ve ödülü düşük olan yöntem Pareto-optimal cephe üzerinde daha iyi dağılan çözümler elde edilmesini sağlamaktadır. Ancak, ceza ve ödül arasında DOPGA2 ve SPEA2 yöntemleri için yapılan benzetimlerde yakınsama arasında bir ilişki bulunmadığı tespit edilmiştir. Ayrıca, seçme mekanizmasının olasılıksal olarak çalışması nedeniyle, çaprazlamaya girecek bireyler, PAM’ dan ne kadar nitelikli bilgi gelirse gelsin, rasgele ve olasılıksal olarak seçilecektir. Hangi birey çiftlerinin çaprazlamaya tabi tutulacağı konusu kesin olarak bilinmemektedir. Bu yüzden, PAM’ dan gelen nitelikli bilgiyi kullanabilecek uzman sistem tabanlı bir seçme mekanizması tasarlanması tartışmaya açılmıştır.

En son olarak, önerilen ve literatürde sık kullanılan ÇAGA yöntemleri bir GUI (Graphic User Interface, kullanıcı arayüzü)’ de toplanmış ve kullanıcıların kendi parametreleri ile optimizasyon problemlerini çözmeleri için bir alt yapı oluşturulmuştur.

2. GENEL BİLGİLER

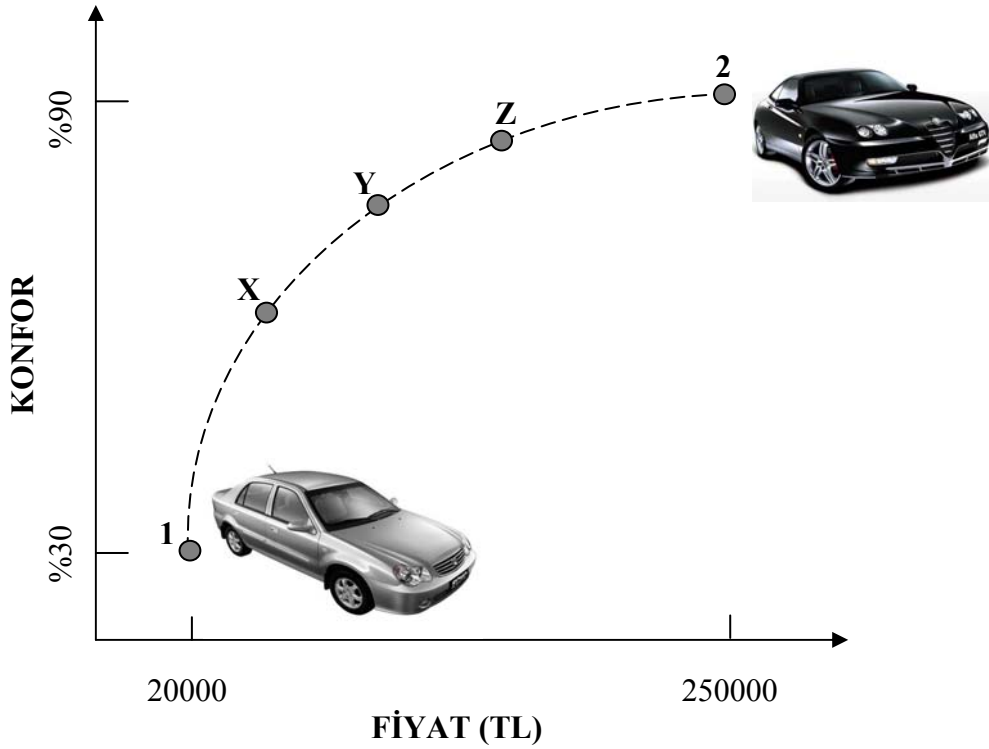
Bu bölümde, optimizasyon kavramı açıklanmakta ve topluluk tabanlı optimizasyon yöntemlerine ilişkin genel kavramlara değinilmektedir. Tezin ana konusu olan genetik algoritmalara ilişkin genel kavramlar ise ayrıntılı olarak açıklanmaktadır. Genetik algoritmaların ortaya çıkma nedeni, amacı, temel işlevleri, bu zamana kadar yapılan çalışmalar ve uygulamaları hakkında genel bilgiler verilmektedir.

2.1. Optimizasyon (En İyileme) Nedir ?

Optimizasyon, daha iyi bir sonuç bulununcaya kadar olası tüm çözümlerin amaç işlevine göre aranması ve karşılaştırılması işlemidir. Optimizasyon konusundaki araştırma ve uygulamaların çoğu yalnızca tek bir amaçla ilgilidir, ancak gerçek dünya problemlerinin çoğu tek amaçtan daha fazlasını gerektirmektedir. Birbiriyle çelişen birçok amacın olması (örneğin bir otomobilin hem konforlu hem de ucuz olmasının istenmesi) çoğu problem için doğaldır. Çok amaçlı bir problemde, bir tek en iyi sonuç yerine en iyi sonuçlardan oluşan ve birbirlerine karşı bir üstünlükleri olmayan bir en iyiler kümesi söz konusu olacaktır. Klasik optimizasyon yöntemleri tek bir benzetimde (adımda) tek bir çözümün bulunmasında iyidirler, ancak çok amaçlı optimizasyon problemlerinin çözülmesi için uygun değildirler. Diğer taraftan, genetik algoritmalar vb. topluluk tabanlı yöntemler, tek bir benzetimde (adımda) çok sayıda en iyi çözümü bulabilirler. Bu yüzden, çok amaçlı optimizasyon yöntemlerinin çözümü için en uygun yöntemler topluluk tabanlı yöntemlerdir.

Fiziksel bir sistemi modelleyen bir optimizasyon problemi yalnızca tek bir amaç işlevi ile gösterilebiliyorsa, bu durumda en iyi çözümün bulunması işlemi *tek amaçlı optimizasyon* olarak adlandırılır. Eğer bir optimizasyon problemi birden fazla amacı içinde barındırıyorsa, bir veya daha fazla en iyi çözümün bulunması işlemine ise *çok amaçlı optimizasyon* denir. Optimizasyon problemlerinin çoğu çok amaçlıdır. Tek bir amacın yerine bütün amaçlar önemlidir. Farklı amaçların olması birbiriyle çelişen farklı çözümlerin elde edilmesini sağlayacaktır. Bir amaca göre iyi olan bir çözüm diğer bir amaca göre kötü olabilir. Bu durumda, tek amaçlı duruma göre en iyi çözümün seçilmesi zor olacaktır. Örnek olarak, bir otomobil satın alma işiyle ilgili karar verilmesi

gerekiyor olsun. Satın alınmak istenen otomobillerin fiyatları 20000 TL ile 250000 TL arasında olsun. Bu otomobillere ilişkin fiyat-konfor grafiği Şekil 2.1’ de gösterilmektedir. En pahalı otomobil 2 numaralı çözüm, en ucuz otomobil 1 numaralı çözüm olarak gösterilmektedir. Eğer yalnızca fiyata göre bir karar verilirse, en iyi çözüm 1 numaralı çözüm (otomobil) olacaktır. Eğer bu durum tüm alıcılar için tek amaç ise, fabrikalar yalnızca bir çeşit otomobil (1 numaralı çözüm) üretecekler ve pahalı otomobil üretmeyeceklerdir. Doğal olarak, bu türden bir karar verme işlemi tek amaçlı olmayacaktır. Ucuz bir otomobilin daha az konfora sahip olacağı beklenmektedir. Konfora daha fazla önem veren zengin müşteriler ise 2 numaralı çözümü (otomobili) seçeceklerdir. Şekilde bu iki otomobil haricinde de çözümler gösterilmektedir. Tüm bu çözümler, birbirlerinden yalnızca tek bir amaç için iyi iken diğer amaç söz konusu olduğunda iyi olmamaktadırlar. Bu durumda her iki amacı da önemseyen kişilerin karar vermesi zor olacaktır.



Şekil 2.1. Örnek bir optimizasyon problemi (otomobil satın alma)

Her iki amaç ta dikkate alındığında en iyi sonuç hangisi olacaktır? Bu sorunun yanıtı, hiçbir çözümün diğerinden iyi olmadığıdır. Her iki amaç ta dikkate alınırsa,

çözümlerin birbirlerine karşı bir üstünlüğü yoktur. Yani çok amaçlı optimizasyonda bir tek en iyi çözüm yerine birçok en iyi çözüm bulunmaktadır. Bu durumda şöyle bir soru daha akla gelebilir: Kişi bu çözümlerden hangisini seçmeli? Yukarıdaki otomobil satın alma problemini düşünürsek, hangi aracın satın alınacağını yanıtlamak kolay değildir. Bu durumda çeşitli yöntemler (çok amaçlı karar verme – multi objective decision making) kullanılarak, koşullar göz önüne alınarak, tercih belirleme (preference articulation) ya da uzman bilgisi (expert knowledge) kullanılarak bu çözümlerden bir tanesi seçilebilir.

2.2. Topluluk Tabanlı Optimizasyon Yöntemleri

Çok amaçlı optimizasyon problemlerinin çözümü için birçok yöntem önerilmiştir. Bu yöntemlerden son yıllarda en dikkat çekenleri doğal yaşamda bazı hayvan/böcek topluluklarının birlikte hareket etmeleri ve istediklerine (yiyecek vb.) ulaşmalarını taklit eden topluluk tabanlı optimizasyon yöntemleridir. Bunların yanı sıra metallerin sıcaklıklarının değiştirilerek tavlanması (ısıtıp soğutarak sertliğini alma) yöntemiyle benzeşim kurularak ta çok amaçlı optimizasyon problemleri çözülebilmektedir. Aşağıda bu yöntemlerden bazıları kısaca ele alınmakta, özellikleri, kısa program kodları, artıları ve eksileri belirtilmektedir. Bu yöntemler arasında yer alan genetik algoritmalar (GA) ise tezin ana bölümünü oluşturduğu için daha ayrıntılı bir şekilde incelenmektedir.

2.2.1. Parçacık Sürü Optimizasyonu (Particle Swarm Optimization, PSO)

J. Kennedy ve R. Eberhart tarafından kuş sürülerinin uçuş biçimlerinden esinlenilerek önerilmiş bir topluluk tabanlı bir optimizasyon yöntemidir [Kennedy ve Eberhart, 1995]. Bu yöntemin ana fikri, yiyecek bulmaya çalışan kuş sürülerinin (veya topluluklarının) hareketlerinin benzetimlerinin yapılmasıdır. Şekil 2.2’ de gerçek bir kuş sürüsü görülmektedir. Bu sürüde, en önde yer alan kuşun hava akımından yararlanan diğer kuşlar daha az yorularak daha fazla mesafe gidebilmektedirler. En öndeki kuş yorulunca onun yakınındaki kuşla yer değiştirmektedir. Böylece tüm kuşlar yer

değiştirirler ve daha az yorularak tüm sürü daha fazla mesafe katederek yiyeceklere ulaşırlar.



Şekil 2.2. Örnek bir kuş sürüsü

PSO kuş sürülerinin davranışlarının bir benzetimidir. Kuşların yerini bilmedikleri yiyeceği aramaları, bir problemin çözümünün aranması ile benzerdir. Kuşlar yiyecek ararken yiyeceğe en yakın olan kuşu izlerler. PSO’ daki parçacık (particle) kavramı sürüdeki her bir kuşu temsil etmektedir. Çözümün her adımında her bir parçacığın puan (fitness) değeri, dolayısıyla yiyeceğe yani amaca ne kadar uzaklıkta olduğu hesaplanır. Her parçacık, GA’ lardaki elitizm mekanizmasında olduğu gibi kendisine ait bilgileri (hızı, puanı ve koordinatları) saklamak zorundadır. Her bir program adımında bir parçacığın konumu, kendi en iyi konumuna ve komşularının en iyi konumuna göre değişir. PSO algoritması rasgele bir parçacık sürüsü veya topluluğu ile başlatılır ve her adımda bir parçacığın konumu, kendi en iyi konumu (p_{eniyi}) ve o anda topluluktaki en iyi çözümün konumuna (g_{eniyi}) göre güncellenir.

Örneğin PSO algoritması, m adet parametreden oluşan n adet parçacıkla (çözümle) başlatılsın. Bu durumda topluluğun parçacık matrisi şöyle verilir:

$$x = \begin{bmatrix} x_{11} & x_{12} & \dots & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & \dots & x_{2m} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & & & x_{nm} \end{bmatrix}_{n \times m} \quad (2.1)$$

Bu matriste i . parçacık $x_i = [x_{i1}, x_{i2}, \dots, x_{im}]$ ile belirtilir. Önceki adımdaki en iyi parçacığın konumu $p_{eniyi} = [p_{i1}, p_{i2}, \dots, p_{im}]$ ile, her adımdaki en iyi parçacığın konumu ise $g_{eniyi} = [p_1, p_2, \dots, p_m]$ ile, i . parçacığın hızı ise $v_i = [v_{i1}, v_{i2}, \dots, v_{im}]$ ile gösterilsin. Buna göre topluluktaki tüm parçacıkların konumları ve hızları sırasıyla aşağıdaki formüllere göre güncellenir:

$$v_i^{k+1} = v_i^k + c_1.rasgele_1^k.(p_{eniyi_i}^k - x_i^k) + c_2.rasgele_2^k.(g_{eniyi}^k - x_i^k) \quad (2.2)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2.3)$$

Bu denklemlerde k adım sayısını göstermektedir. c_1 ve c_2 ise öğrenme katsayılarıdır. c_1 parçacığın kendi bilgilerine göre hareket etmesini, c_2 ise sürünün bilgilerine göre hareket etmesini sağlar. c_1 ve c_2 değerlerinin çok iyi seçilmesi gerekmektedir. Düşük seçilmeleri uzak bölgelerde arama yapılmasına yol açar ve amaca ulaşılmasını geciktirebilir. Yüksek seçilmeleri ise istenilen amaca ulaşılmasına yol açabilir. Araştırmacılar $c_1=c_2=2$ seçilmesinin iyi sonuçlar verdiğini belirtmişlerdir. Genel bir PSO algoritmasının kodu şöyle verilir [Tamer ve Karakuzu, 2006] :

For Her parçacık için başlangıç koşulları

End

Do For Her parçacık için puan değerini hesapla. Eğer puan değeri,

p_{eniyi} den daha iyi ise; şimdiki değeri yeni p_{eniyi} olarak ayarla

End

Tüm parçacıkların bulunduğu p_{eniyi} değerlerinin en iyisini, tüm parçacıkların

g_{eniyi} 'si olarak ayarla

For her parçacık için (2.2) denklemine göre parçacık hızını hesapla

(2.3) denklemine göre parçacık konumunu güncelle

End

While maksimum adım sayısına ulaşıncaya veya minimum hata koşulu

sağlanana kadar devam et

PSO hem sürekli doğrusal olmayan, hem de ayırık ikili optimizasyonda başarılı olarak kullanılmıştır [Kennedy ve Eberhart, 1995, 1997, 2001; Eberhart ve Shi, 1998; Engelbrecht, 2003]. Bir parçacık ya da birey tarafından bulunan en iyi sonuçların kaydının tutulması, geçmişte üretilen bastırılmayan bireylerin saklanması için kullanılabilir (bu aşama genetik algoritmalarındaki elitizm mekanizmasıyla benzerdir).

Son yıllarda, çok amaçlı birçok PSO yöntemi önerilmiştir. Ray ve Liew, Pareto baskınlık kavramını kullanan ve genetik algoritma tekniklerini PSO ile birleştiren bir algoritma önermişlerdir [Ray ve Liew, 2002]. Bu algoritma kalabalıklık yöntemini kullanarak çeşitliliği sağlamaktadır. Hu ve Eberhart ise bir kerede yalnızca bir amacın en iyilenebileceği dinamik komşuluk yaklaşımını önermişlerdir [Hu ve Eberhart, 2002]. Daha sonra Hu ve ark., ikincil topluluk ekleyerek dinamik komşuluk yaklaşımı PSO yönteminde bir takım iyileştirmeler yapmışlardır. Coello Coello ise bir parçacığın uçuş (hareket) yönünün belirlenmesi için Pareto baskınlık kavramını kullanmıştır ve karar uzayında daha iyi bir birey dağılımı elde edebilmek için kümeleme algoritmalarıyla sürüleri (ya da toplulukları) alt topluluklara bölmüştür [Pulido ve Coello Coello, 2004]. Srinivasan ve Seow ise PSO'yu evrimsel algoritma ile birleştiren *Evrimsel Algoritmadan esinlenen Parçacık Sürüsü* (PS-EA) yöntemini önermişlerdir [Srinivasan ve Seow, 2003; 2005]. Bu algortmada elit bireyler saklanmış ve birleştirme işlevi uygulanmamıştır. Raquel ve Naval ise küresel en iyinin seçilmesi ve arşivdeki bireylerin silinmesi için kalabalıklık mesafesi kavramını kullanmışlardır [Raquel ve Naval, 2005].

PSO' nun en büyük artısı, hem kavramsal olarak hem de uygulama açısından çok basit olmasıdır. Ayrıca kullanımı basittir ve yüksek bir yakınsama hızına sahiptir.

PSO' nun en önemli eksiği ise çok amaçlı optimizasyonda çeşitliliğin kontrol edilmesinin zor olmasıdır. Çok amaçlı PSO yöntemlerindeki çeşitlilikteki azalma, mutasyon (veya türbölans) işlevleri kullanılarak giderilmeye çalışılır.

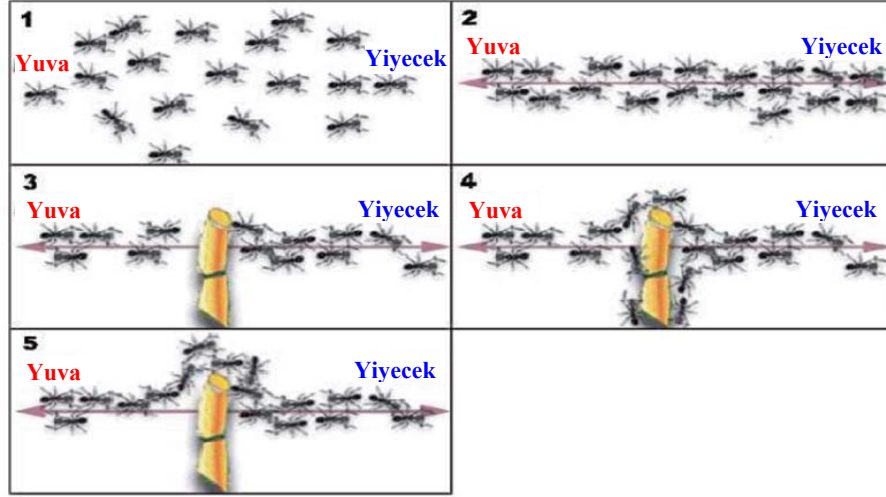
PSO ile ilgili birçok çalışma yapılmaya devam edilmektedir. Çalışmalarda PSO yönteminin iyileştirilmesi, başka PSO yöntemleri ile karşılaştırma, diğer çok amaçlı optimizasyon yöntemleriyle hibritleştirme ve PSO' nun gerçek dünya problemlerine uygulanması başlıca konulardır. PSO ile ilgili genel bilgi ve ayrıntılara [Kennedy ve Eberhart, 2001; Engelbrecht, 2003; Clerc 2006; Lazineka, 2009, Lim ve ark., 2009] kaynaklarından ulaşılabilir. Son yıllarda, PSO algoritmasına olan ilginin artmasıyla, bu konuyla ilgili kitap ve makale sayılarında önemli bir artış gözlenmektedir. Bununla

beraber PSO ile ilgili birçok açık problem vardır ve yeni alanlara uygulanması da gerekmektedir. Bu amaçla, IEEE Transactions on Evolutionary Computation dergisi 2009 yılı Ağustos ayında PSO ve ACO'yu da içeren sürü optimizasyonu yöntemleri ile ilgili özel bir sayı çıkarmıştır. (Editörler: Andries Engelbrecht, Xiaodong Li, Martin Middendorf, Luca Maria Gambardella, *IEEE Trans. EvComp.*, Vol.13, No:4)

2.2.2. Karınca Kolonisi Optimizasyonu (Ant Colony Optimization, ACO)

ACO algoritması, gerçek karıncaların yuvaları ile yiyecek noktaları arasındaki en kısa yolu (feromon maddesi yardımıyla) bulma yeteneklerinden esinlenilerek geliştirilmiştir [Colormi ve ark., 1991; Dorigo ve ark., 1996; Dorigo ve Caro, 1999; Dorigo ve Stützle, 2004]. Feromon karıncaların davranışını etkilemektedir, bu madde karıncaların çok miktarda feromon olan yerlere giden yolları (dolayısıyla en kısa yolları) bulmasını sağlamaktadır. Feromon izleri (kokuları), karıncalar arasında doğrusal olmayan bir iletişim olarak görülebilir.

Şekil 2.3, bir karınca kolonisinin tipik bir davranış örneğini göstermektedir. Karıncalar yuvalarının terk ederken, (1) rasgele yollar izlerler, (2) belirli bir zamandan sonra ortak bir yolu izlemeye başlarlar, (3,4) bir engelle karşılaştıkları zaman bazıları engelin sağına bazıları ise soluna doğru gitmeyi seçerler, (5) belirli bir zamandan sonra feromon maddesinden dolayı tüm koloni ortak bir yolu (feromon maddesi çok olan yolu yani en kısa yolu) takip eder. Kısa olan yolda feromon miktarı uzun yollara nispeten daha fazla birikmektedir. Kısa olan yoldan geçiş daha hızlı gerçekleşeceğinden, birim zamanda geçiş yapan karınca sayısı uzun yola göre daha fazla olacaktır. Dolayısıyla herhangi iki düğüm arasındaki yol üzerinde bulunan feromon miktarı, yolun uzunluğuyla ters orantılıdır [Keskintürk ve Söyler, 2006].



Şekil 2.3. Bir karınca kolonisinin davranışı [Coello Coello ve ark., 2007]

Karınca kolonilerinden uyarlanan ACO algoritmasında kullanılan üç ana fikir şöyledir:

- i) Feromon izleri ile doğrusal olmayan iletişim,
- ii) En kısa yollar feromon seviyesinin yüksek olduğu yerlerdir,
- iii) Karıncalar yüksek feromon olan yerleri tercih etmektedirler.

Ek olarak ACO algoritması gerçek karınca kolonilerinde olmayan yeteneklere de sahiptir. Örneğin:

- i) Her karınca hedeften ne kadar uzak olduğunu tahmin edebilme yeteneğine sahiptir.
- ii) Karıncalar çevreleri hakkında bilgiye sahiptir ve karar vermek için bunu kullanırlar. Bu yüzden, davranışları yalnızca uyarlanabilir (adaptif) değildir, aynı zamanda detaylıdır.
- iii) Algoritmanın her adımında yalnızca olası sonuçların üretildiğinden emin olmak için, karıncalar bellek mekanizmasına sahiptirler.

ACO gezgin satıcı probleminin (traveling salesman problem-TSP) çözümü için önerilmiştir [Dorigo ve ark., 1991]. Bu algoritma işlem zamanı ve çözüm kalitesi açısından diğer sezgisel yöntemlerden daha geride olduğundan algoritmanın başarımını

artırmak ve daha iyi çözümler elde etmek amacıyla çalışmalar yapılmıştır [Keskintürk ve Söyler, 2006]. Gambardella ve Dorigo Ant-Q sistemini [Gambardella ve Dorigo, 1995], Bullnheimer ve arkadaşları Rank Temelli Karınca Sistemi (ASrank)'ni [Bullnheimer ve ark., 1997], Dorigo ve Gambardella Karınca Koloni Sistemi (ACS)'ni [Gambardella ve Dorigo, 1997], Stützle ve Hoos Max-Min Karınca Sistem (MMAS)'ni [Stützle ve Hoos, 2000] ortaya koymuşlardır. Ayrıca birden çok karınca kolonisi ile çalışmayı esas alan algoritmalar da geliştirilmiştir [Tsai ve ark., 2004]. Bunların dışında birçok çalışmada farklı alanlardaki uygulamalara yönelik, yöntemin başarımını arttırmak amaçlı değişiklikler ve eklemeler yapılmıştır [Bullnheimer ve ark., 1999]. Genel Ant-Q algoritması Şekil 2.4' te gösterildiği gibi dört ana aşamadan oluşmaktadır [Gambardella ve Dorigo, 1995].

1. $Q(s,a)$ rasgele bir değerle başlatılır
2. For $i=1$ to N (N -bölüm sayısı)
 - For $i=1$ to m (birey sayısı)
 - Her birey için $s=s_0$ yap
 - Her bir bölüm için f defa tekrarla
 - For $i=1$ to m
 - (2.4) eşitliğine göre a ve s' yi belirle
 - a değerini uygula, r ve s' değerlerin gözlemle
 - $Q(s,a) \leftarrow Q(s,a) + \alpha [\gamma maks_a Q(s',a') - Q(s,a)]$
 - $s \leftarrow s'$
 - Döngüyü bitir
 - s sonlandırma kriterine ulaşıncaya kadar devam et
 - Döngüyü bitir
3. Bulunan m adet sonucu karşılaştır ve en iyiyi seç
 - En iyi çözümdeki tüm $Q(s,a)$ ' lar için
 - $Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma maks_a Q(s',a') - Q(s,a)]$
 - (α – öğrenme katsayısı, γ – azaltma çarpanıdır)
4. Döngüyü bitir.
5. En iyi sonucu göster.

Şekil 2.4. Ant-Q program kodu

İlk olarak, $Q(s,a)$ işlevleri başlatılır (sabit değerlerle başlatmak ta mümkündür). İkinci aşamada her bir dağıtıcıya (çözüme) bir değer atanması ile başlar. Her adımda, hareket seçme kuralına (2.4) bağlı olarak her dağıtıcı o andaki durumu için olası hareketlerden birini seçer.

$$s' = \begin{cases} \arg \max_{s'} C(s, s') & \text{eger } q \leq q_0 \\ s_{\text{rasgele}} & \text{aksi takdirde} \end{cases} \quad (2.4)$$

q , $[0,1]$ aralığında oluşturulan bir rasgele sayıdır. s ve s' , durumları göstermektedir. $C(s,s')$ birleştirilmiş işlevdir.

Her geçiş noktasında dağıtıcının işlev değeri aşağıdaki formüle göre güncellenir:

$$Q(s, a) \leftarrow Q(s, a) + \alpha [\gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (2.5)$$

Tüm dağıtıcıların işlev değerleri bulununcaya kadar devam eder. Üçüncü aşamada, çözümler hesaplanır ve en iyi sonuçlar ödüllendirilir. Ayrıca hareket değerleri de aşağıdaki formüle göre güncellenir:

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (2.6)$$

Buradaki α ve γ katsayıları kullanıcı tarafından belirlenmelidir, bu katsayıların belirlenmesi deneme yanılma yoluyla olur. Son olarak, dördüncü adımda sonlandırma kriterine göre algoritma sonlandırılır ve en iyi sonuçlar belirlenir.

Mariano ve Morales, Ant-Q algoritmasının çok amaç için kullanılan bir versiyonunu (Multi-Objective Ant-Q, MOAQ) önermişlerdir [Mariano ve Morales, 1999a, 1999b]. ACO ile ilgili birçok algoritma önerilmiş ve bunlar bazı gerçek dünya problemlerinin çözümü için kullanılmıştır. Karıca kolonisi optimizasyonu ile ilgili ayrıntılı bilgilere [Engelbrecht, 2003; NABIYEV, 2003; Dorigo ve Stützle, 2004; Karaboğa, 2004] kaynaklarından ulaşılabilir.

2.2.3. Farksal Gelişim (Differential Evolution, DE)

Kenneth Price ve Rainer Storn tarafından sürekli optimizasyon problemlerinin çözümü için tasarlanmış nispeten yeni bir algoritmadır [Storn ve Price, 1995, 1997; Price ve ark., 2005]. Bu algoritma Price' ın Tchebycheff polinomal uyarlama problemini çözmek için Storn' a danışması nedeniyle ortaya çıkmıştır. Bu problemin çözümünde Price, vektör topluluğunu bozmak için vektör farklarını kullanmayı düşünmüştür. Daha sonra bunun uygulanması için algoritma önerilmiş ve iyileştirilmiştir [Storn ve Price, 1995]. DE genellikle sürekli optimizasyon problemlerinin çözümü için kullanılan topluluk tabanlı bir algoritmadır. DE, geleneksel genetik algoritmalarla benzerliklere sahiptir. Bununla beraber, basit bir GA' daki gibi ikili kodlama kullanmaz. Bunun yerine DE, topluluktaki çözümlerin dağılımına bağlı mutasyon uygular. Bu yolla, mutasyon değerlerinin hesaplanması için, arama yönleri ve bireylerin konumuna bağlı olası adım boyutları seçilir.

DE, en güçlü olasılıksal gerçel parametrelili optimizasyon algoritmalarından biridir. Çok basit bir algoritmadır ve yalnızca birkaç satırlık kodlarla programlanabilir. Ek olarak, kullanımı kolaylaştıran çok az sayıda kontrol parametresi kullanır. Buna rağmen; doğruluk, yakınsama hızı, kararlılık ve gürbüz olma durumları açısından çok amaçlı ve çok modlu işlevlerin optimizasyonunda dikkate değer bir başarımlı göstermektedir.

DE tek amaçlı optimizasyon problemlerin çözümü için çok iyi bir algoritmadır ve çok sayıda araştırmacı çok geniş sayıda optimizasyon probleminde (genellikle doğrusal olmayan) DE' nin çok iyi olduğunu göstermişlerdir [Storn ve Price, 1995; Mezura-Montes ve ark., 2006]. Buna rağmen, DE' nin çok amaçlı optimizasyon için kullanılması ile ilgili araştırmalar yeni yeni artmaktadır. DE, PSO' ya benzer şekilde yüksek bir yakınsama hızına sahiptir, ancak gerçek Pareto cephesine ulaşmakta zorluklar yaşamaktadır. Ayrıca, çok amaçlı DE çeşitliliğin sağlanması için ek mekanizmalara (örneğin kalabalıklık tabanlı işlevler veya mutasyon işlevleri) ihtiyaç duymaktadır. DE yalnızca parametreleri gerçel sayı olan problemler için önerilmiştir, diğer kodlama çeşitlerinde kullanılması için yenilikler yapılması gerekmektedir.

DE algoritmasının önemli parametreleri: *NP* (Number of population-topluluk boyutu), *CR* (Crossover rate-çaprazlama oranı), *F* (Scaling factor-Ölçekleme sabiti) olarak sayılabilir. İlk olarak, *NP* adet başlangıç vektörü rasgele oluşturulur. Daha sonra

ise genetik algoritmalarla olduđu gibi aprazlama, mutasyon ve seme iřlevleri uygulanır. DE algoritmasının temel adımları řöyle verilebilir [Yiđit ve ark., 2007]:

Başlangı topluluđunun oluřturulması

Deđerlendirme

Tekrarla

Mutasyon

Yeniden birleřtirme / aprazlama

Deđerlendirme

Seme

Durdurma kriteri sađlanıncaya kadar

DE algoritması, diđer biyolojik kökenli algoritmalarla mutasyon ve yeniden birleřtirme (aprazlama) ařamalarında farklılık göstermektedir. DE algoritması topluluđu karıřtırmak için özüm vektörleri arasındaki ađırlıklandırılmıř farkları kullanmaktadır:

$$u_{i,G+1} = x_{i,G} + K \cdot (x_{r3,G} - x_{i,G}) + F \cdot (x_{r1,G} - x_{r2,G}) \quad (2.7)$$

Burada $r_1, r_2, r_3 \in \{1, 2, \dots, n\}$ ve $r_1 \neq r_2 \neq r_3$ olmak üzere rasgele seilir. DE algoritması, bir ama vektörüne mutasyon iřlemini, rastgele seilmiř ama vektör iftinin ađırlıklandırılmıř farkının bu ama vektörüne eklenmesiyle gerekleřtirir. Seme iřlemi, yeni üretilen vektörlerin hangi řartlar altında topluluđa girebileceđini tanımlayan bir kriterdir. DE algoritmasının seme iřleminde, yeni üretilen vektör ebeveynine göre daha geliřmiř veya en azından aynı geliřme seviyesinde deđilse ebeveyn vektör en az bir adım daha toplulukta kalmaya devam etmekte ve bařka bir vektörle yer deđiřtirmemektedir. Yeniden birleřtirme veya aprazlama iřleminin amađı, var olan ama vektör parametrelerinden faydalanarak yeni vektörleri oluřturmak suretiyle arařtırmanın bařarılı olması için yardımcı olmaktır [Yiđit ve ark., 2007].

Son on yılda, DE algoritmasına olan ilginin artmasıyla, bu konuyla ilgili kitap ve makale sayılarında önemli bir artıř gözlenmektedir. Bununla beraber, DE ile ilgili birok aık problem vardır ve yeni alanlara uygulanması da gerekmektedir. Bu amala,

IEEE Transactions on Evolutionary Computation dergisi 2011 yılında DE ile ilgili özel bir sayı çıkaracaktır (Editörler: Swagatam Das, P. N. Suganthan, Carlos A. Coello Coello). DE algoritması ile ilgili ayrıntılı bilgilere [Engelbrecht, 2003; Karaboğa, 2004; Price ve ark., 2005; Das ve ark., 2008; Chakraborty, 2008] kaynaklarından ulaşılabilir. Ayrıca, veri kümeleme yöntemleri için DE ve PSO' nun kullanılması ve DE ile ilgili ayrıntılı bilgilerin verildiği bir video sunumu http://videlectures.net/solomon_krink_depso/ bağlantısı kullanılarak izlenebilir (Erişim zamanı: 05.08.2009).

2.2.4. Benzetimli Tavlama (Simulated Annealing, SA)

Benzetimli tavlama, metallerin tavlanması yani ısıtılıp soğutularak sertliğinin alınması işleminden uyarlanan olasılıksal bir arama algoritmasıdır. Tavlama işlemi, bir katı cismin (genelde metaller, örneğin çelik) sıcaklığının atomlarının serbest olarak hareket edebileceği bir noktaya kadar artırılması, daha sonra atomların daha düşük enerji seviyelerinde yeniden düzenlenmesini sağlayacak bir şekilde sıcaklığın düşürülmesi olayıdır (örneğin bir kristalizasyon işlemi). Bu işlem sırasında metalin serbest enerjisi düşürülür. Soğutma işlemi hayati bir öneme sahiptir. Eğer metal çok hızlı bir şekilde soğutulursa veya sistemin başlangıç sıcaklığı çok düşükse, bir kristal elde edilemez, onun yerine metal yüksek enerjili amorf (şekilsiz, kristal yapısı olmayan) bir duruma gelir. Bu durumda, sistem küresel en küçük (en düşük enerji durumu) yerine yerel en küçük değere (yüksek enerji durumuna) ulaşır [Dowsland, 1993]. Metropolis ve ark., ısıtma kazanındaki katı bir cismin ısı dengeye ulaşmaya kadar geçirdiği değişimlerin benzetimini yapmak için bir algoritma önermişlerdir [Metropolis ve ark., 1953]. Metropolis' in bu yaklaşımından sonra, Kirkpatrick ve ark. [Kirkpatrick ve ark., 1983] ve Cerny [Cerny, 1985] birbirlerinden bağımsız olarak, tavlama işlemi ile optimizasyon arasındaki benzeşime dikkat çekmişlerdir. Bu araştırmacılar birkaç adet önemli benzeşim önermişlerdir: sistem durumu optimizasyon probleminin çözümü ile benzerdir, sistemin serbest enerjisi (azaltılacak) en iyilenecek amaç işlevinin aldığı değerle ilgilidir, düzensizlik GA' larda kullanılan mutasyon işlevi ile benzeşiktir, soğutma işlemi arama algoritmasındaki denetim mekanizmasıyla ilgilidir ve son olarak sistemin donma durumu arama algoritmasının (topluluk boyutu bir olduğunda) son

çözümünü göstermektedir. Bu benzeşimler kullanılarak benzetimli tavlama algoritması geliştirilmiştir. Benzetimli tavlamanın program kodu Şekil 2.5’ de verilmektedir.

1. Bir başlangıç (olası) çözümü seçilir (s_0),
2. Bir başlangıç sıcaklığı seçilir ($t_0 > 0$),
3. Bir soğutma çizelgesi (programı) seçilir CS ,
4. Tekrarla

Tekrarla
 Rasgele bir $s \in N(s_0)$ seç (N : komşuluk yapısı)
 $\delta = f(s) - f(s_0)$ (f : amaç işlevi)
 Eğer $\delta < 0$ ise o zaman $s_0 \leftarrow s$
 Aksi takdirde
 Rasgele x değeri oluştur $[(0,1)$ aralığında düzgün dağılım]
 Eğer $x < \exp(-\delta/t)$ ise o zaman $s_0 \leftarrow s$
 ITER verilen adım sayısına ulaşıncaya kadar
 $t \leftarrow CS(t)$
5. Sonlandırma kriteri sağlanıncaya kadar devam et.

Şekil 2.5. Benzetimli tavlama program kodu [Dowsland, 1993].

Şekil 2.5’ de verilen program kodunda en küçükleme problemi söz konusudur. Bu algoritma, o andaki durumun komşuluklarında yerel hareketler oluşturur ve yeni durum o andaki sıcaklığa (t) bağlı bir işleve göre belirlenir. Algoritmanın iki ana parametresi, adım sayısı (ITER) ve soğutma çizelgesi (CS)’ dir, çünkü bu parametreler algoritmanın başarımına önemli derecede etki etmektedirler.

Benzetimli tavlamanın çok amaçlı optimizasyon için kullanımı ilk olarak Serafini tarafından önerilmiştir [Serafini, 1994]. Serafini, iki amaçlı bir optimizasyon probleminin çözümü için hedef-vektör yaklaşımını kullanmıştır. Serafini’ nin kullandığı bu yöntem MOSA (Multi-Objective Simulated Annealing, Çok Amaçlı Benzetimli Tavlama) olarak adlandırılmıştır. Ray ve ark., benzetimli tavlama kullanarak çok amaçlı bir tasarım problemini ağırlıklı toplamla çözmüştür [Ray ve ark., 1995]. Ruiz-Torres ve ark. iki amaçlı paralel makine çalışma çizelgesi problemini çözmek için Pareto tanımını seçme kriteri olarak kullanan bir benzetimli tavlama kullanmışlardır [Ruiz-Torres, 1997]. Czyzak ve Jaszkiewicz ise Pareto Benzetimli Tavlama (PSA-Pareto Simulated

Annealing) isimli bir yöntem önermişlerdir [Czyzak ve Jaszkievicz, 1997a, 1997b]. Bu yöntemde, MOSA' dan farklı olarak her adımda tek bir çözüm yerine çözümlerden oluşan bir topluluk kullanılmıştır. Her iki yöntemde de bastırılmayan çözümler saklanmıştır. PSA yöntemi MOSA ile karşılaştırılmış ve Pareto cephesinde birçok birey üretmiştir ve bu bireyler bu cephe üzerinde düzgün dağılmışlardır. PSA birçok alana uygulanmıştır (örneğin hemşire çalışma çizelgesi problemi vb.).

Literatürde çok sayıda çok amaçlı benzetimli tavlama yöntemi yer almaktadır. Bu yöntemlerden bazıları şöyledir: SMOSA [Suppaitnarm ve ark., 2000], UMOSA [Ulungu ve ark., 1995, 1997], WMOSA [Suman, 2002, 2003] ve AMOSA [Bandyopadhyay ve ark., 2008]. Bu yöntemler çeşitli uygulamalar üzerinde değerlendirilmiş ve birbirleriyle karşılaştırılmışlardır. Benzetimli tavlama hakkında daha ayrıntılı bilgilere [Karaboğa, 2004; Tan, 2008] kaynaklarından ulaşılabilir.

2.2.5. Diğer Yöntemler

Literatürde yukarıda belirtilen yöntemler haricinde de optimizasyon (özellikle çok amaçlı) için kullanılabilecek yöntemler bulunmaktadır. Bu yöntemlerden bazıları şöyledir: Tabu algoritması (Tabu Search), yapay bağışıklık sistemleri (Artificial Immune Systems), kültürel algoritmalar (Cultural Algorithms) ve pekişik arama (Cooperative Search). Bu yöntemlerle ilgili ayrıntılı bilgilere [Nabiyev, 2003; Engelbrecht, 2003; Karaboğa, 2004; Coello, 2007] kaynaklarından ulaşılabilir.

2.3. Genetik Algoritmalar

Genetik algoritmalar (GA), en iyi çözümü arama işlevini biyolojik kökenli bazı işlevleri (seçme, çaprazlama, mutasyon, elitizm vb.) taklit ederek yapar. Genetik algoritmaların diğer optimizasyon yöntemlerinden ve klasik arama yöntemlerinden en önemli farkı, tek bir çözüm yerine her adımda çözümlerden oluşan bir topluluk kullanmasıdır. Her adımda bir çözüm topluluğu kullanıldığı için, genetik algoritmayla elde edilen sonuçlar da bir çözüm topluluğu olacaktır. Çok sayıda en iyi çözümün tek bir adımda (benzetimde) bulunabilmesi, genetik algoritmaların en önemli

özelliklerinden biridir. Genetik algoritmalar, Pareto cephesinin (bulunması istenen çözüm bölgesi) şekline ve sürekli olup olmamasına çok duyarlı değillerdir (örneğin ayrık ve içbükey Pareto cephelerini kolaylıkla bulabilmektedir) [Coello Coello, 1999].

Son yirmi yıl içinde, değişik mühendislik konularının bilgisayar aracılığı ile modellenmesi, benzetilmesi, en iyilenmesi (optimizasyon) ve gelecek davranışlarının tahmini için doğal olayların işleyiş ve davranış biçimlerinden esinlenerek ilgi çekici yöntemler geliştirilmiştir. Bunlar arasında canlıların genetik davranış biçimleri, genetik algoritmaların (GA) ortaya çıkmasında çok önemli rol oynamıştır.

Genetik algoritmalar doğada var olan bir yarışma ortamında, ancak daha iyi ve daha kuvvetli olan bireylere kazanma şansının verildiği biyolojik olaylara benzetilerek geliştirilmiş paralel ve küresel (global) bir arama tekniğidir. Arama uzayında aynı anda birçok noktayı değerlendirdiği için küresel çözüme ulaşma olasılığı fazladır. Genetik algoritmalar (GA) esasen kılavuzlanmış bir rasgele sayı üretme tekniğidir, yani parametreler için belirli sınırlar vardır. Genetik algoritmalar, ikili (veya gerçel sayı) dizi yapıları içinde en iyi olanların yaşayabilme şansı ile yapılandırılmış olsa da, rasgele bilgi alışverişi olaylarını birleştirerek bir arama algoritması oluştururlar. Genetik algoritmalar türevsel bir niteliğe sahip olmadığından analitik değildirler, tekrarlandığında aynı sonuç alınmayabilir.

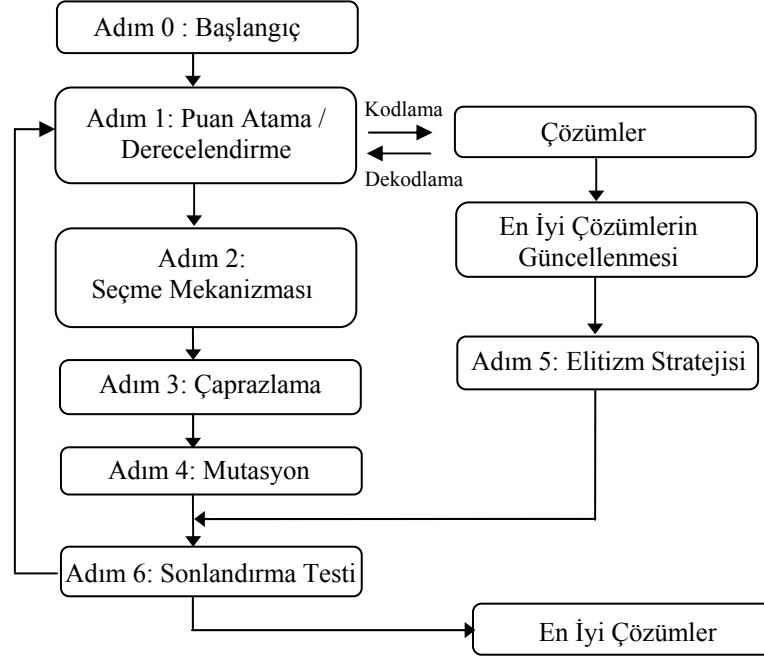
Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı olan John Holland bu konuda ilk çalışmaları yapan kişidir (Holland, 1975). Mekanik öğrenme konusunda çalışan Holland, Darwin'in evrim kuramından etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine böyle yapılardan oluşan bir topluluğun çoğalma, çiftleşme, mutasyon vb. genetik süreçlerden geçerek başarılı yeni bireyler oluşturabildiğini görmüştür. Araştırmalarını, arama ve en iyiyi bulma için, doğal seçme ve genetik evrimden yola çıkarak yapmıştır. Biyolojik sistemde bireyin bulunduğu çevreye uyum sağlaması örnek alınarak, en iyiyi bulma ve makine öğrenme problemlerinde bilgisayar yazılımı modellenmiştir. Holland'ın çalışmalarının sonucunu açıkladığı kitabının 1975'te yayınlanmasından sonra geliştirdiği yöntemin adı **Genetik Algoritmalar** (ya da kısaca **GA**) olarak yerleşmiştir.

Genetik algoritmalarla ilgili ilk gerçek uygulama 1984 yılında Holland'ın öğrencisi olan David Schaffer tarafından yapılmıştır (Schaffer, 1984). Schaffer tarafından önerilen vektör hesaplamalı genetik algoritma (VEGA), tek amaçlı genetik

algoritmanın basit bir iyileştirmesidir ve çoklu çözümleri genetik algoritmaların yalnızca birkaç adımda bulabileceğini göstermiştir. Bu çalışmadan sonra, GA araştırmacıları yaklaşık 5 yıl çok amaçlı optimizasyon konularına fazla ilgi göstermemişlerdir. 1989 yılında, David E. Goldberg yeni ufuklar açan kitabında (Goldberg, 1989) baskınlık kavramını kullanan 10 satırlık bir çok amaçlı genetik algoritma (ÇAGA) yöntemi önermiştir. Bu kitaptaki ipuçlarından yararlanan çok sayıda araştırmacı farklı ÇAGA yöntemleri geliştirmeye başlamıştır. Bunların arasında Srinivas ve Deb' in NSGA (Non-dominated Sorting GA, 1994) yöntemi, Fonseca ve Fleming' in MOGA (Multiple Objective GA) yöntemi, Horn, Nafplitis ve Goldberg' in NPGA (Niche Pareto GA) yöntemi ilk geliştirilen yöntemlerdendir. Bu baskınlık tabanlı yöntemler, birbirinden farklı gerçek dünya problemleri üzerinde test edilmiş ve bu yöntemlerin çok amaçlı problemlerin çözülebilmesini sağladığı görülmüştür. Bu çalışmalardan sonra, çok amaçlı optimizasyon yöntemlerinin çözülebilmesi için genetik algoritmaları farklı biçimlerde kullanan yeni yöntemler önerilmiştir. Bu yöntemlere, Kursawe' nin eş kalıtsallık (diploidy) yaklaşımı (Kursawe, 1990), Hajela ve Lin' in ağırlık tabanlı yaklaşımı (Hajela ve Lin, 1992), Osyczka ve Kundu' nun mesafe tabanlı GA' sı (Osyczka ve Kundu, 1995) örnek olarak verilebilir. Çok amaçlı optimizasyona olan ilginin artmasıyla, GA dergileri yayınlanmaya, bu konuyla ilgili uluslararası konferanslar düzenlenmeye başlamıştır. 2000' li yıllarda ise çok sayıda yeni ÇAGA yöntem önerileri [PAES (Knowles ve Corne, 2000), PESA (Corne ve ark., 2000), SPEA2 (Zitzler ve ark., 2001), NSGA-II (Deb ve ark., 2002b), DMOEA (Yen ve Lu, 2003) vb.] yapılmış ve bu yöntemler çok amaçlı optimizasyon problemlerinin çözümü için kullanılmışlardır. Son yıllarda ise GA' larla ilgili çalışmalarla beraber diğer bazı topluluk tabanlı optimizasyon yöntemleri ile karşılaştırmalar yapılmakta, yeni yapı önerileri verilmekte, önerilen veya önerilmiş ÇAGA yöntemleri test işlevleri yardımıyla karşılaştırılmaktadır.

Genetik algoritmaların genel yapısı Şekil 2.6' da verilmektedir. GA başlamadan önce optimizasyon problemini tanımlayan bir amaç işlevinin belirlenmesi gerekmektedir. Bu işlemten sonra, GA adımları, çözüm önerilerini yani bireyleri içeren bir başlangıç topluluğunun belirlenmesiyle ve probleme ilişkin parametrelerin girilmesiyle başlar. Başlangıç topluluğu rasgele belirleneceği gibi dışarıdan kullanıcı tarafından da girilebilir. Daha sonra, amaç işlevinin hesabı parametrelere ve girişlere göre yapılır. En iyi birey/bireyler bellekte saklanır (elitizm). Seçme mekanizmasıyla

(RWS, SUS veya TS) bir sonraki nesli yani çözüm önerilerini üretecek anne-baba bireyler seçilir. Çaprazlama ile çocuk bireyler (yeni çözüm önerileri) üretilir. Farklı bireylerin oluşturulması (çeşitlilik) için ise mutasyon işlemi uygulanır. En iyi birey/bireyler çözümlere eklenir. Girilen adım sayısına ve sonlandırma kriterine göre GA' nın bir adımı tamamlanmış olur.



Şekil 2.6. GA' ların genel yapısı

Genetik algoritmalar problem sayısına göre tek amaçlı veya çok amaçlı, parametrelerin kodlanma biçimine göre gerçel ya da ikili, problem tipine göre ise kısıtlamalı veya kısıtlamasız olarak adlandırılırlar. Bu tezde ikili kodlama kullanan/kısıtlamasız/çok amaçlı genetik algoritmalar kullanılmıştır.

ÇAGA' ların bazı temel ve istenilen karakteristikleri şöyle özetlenebilir: 1) birbiriyle çelişen çoklu amaçları kullanabilmelidir, 2) var olan kısıtlamalar puan atama mekanizmasına (PAM) eklenebilmelidir, 3) herhangi bir harici katsayıya gereksinim duyulmadan çeşitlilik bilgisi bir bireyin bağlı konumu dikkate alınarak puan atama mekanizmasına eklenebilmelidir, 4) elitizm mekanizması etkili bir biçimde kullanılmalıdır, 5) PAM tarafından farklı bireylere farklı puanlar atanabilmelidir, böylece topluluktaki bireyler en çok istenilenden en az istenilene doğru tekil bir biçimde sıralanabilir, 6) endüstri uygulamalarında sık görülen kötü başlangıç koşullarına

hesapsal olarak dayanıklı olmalıdır, 7) çalışma zamanları mümkün olduğu kadar az olmalıdır, 8) GA küresel arama yöntemidir ve çözüm/çözümleri kullanıcı tarafından belirlenen sınırlar içerisinde arar. GA' ların bulduğu sonuçlara göre parametre sınırları artırılıp azaltılarak bölgesel arama (region of interest) çözünürlüğü değiştirilebilir, böylece arama hızı ve başarımı artırılabilir, 9) probleme bağlı olarak ÇAGA yöntemlerinin melezleştirilmesi (hibritleştirilmesi) faydalı olabilir, 10) puan ölçekleme ÇAGA yöntemlerinin yakınsama özelliklerini iyileştirebilir.

2.3.1. Klasik Optimizasyon Yöntemlerinden Farkları

Genetik algoritmaların klasik optimizasyon yöntemlerinden farkları şöyle özetlenebilir:

- GA, çözüm uzayında aynı anda geniş bir alanda çok sayıda noktadan araştırmaya başlar. Arama uzayında, **yerel (local)** değil **küresel (global)** arama yaparak sonuca ulaşmaya çalışır. Bir tek noktadan değil, bir grup çözüm içinden arama yapar. Aramaya tek bir noktadan değil birçok noktadan başlamanın en büyük yararı; yerel en iyiye yakalanma olasılığını ortadan kaldırmasıdır. Klasik optimizasyon yöntemleri aramaya tek bir noktadan başladıkları için, ilk buldukları yerel en iyi noktada aramayı sonlandırırlar. Dolayısıyla çok amaçlı optimizasyon için uygun değildir.
- GA, arama uzayında bireylerin puan değerlerini bulmak için sadece *amaç - uygunluk işlevi* (objective - fitness function) ister. Böylelikle sonuca ulaşmak için türev ve diferansiyel işlemlere gerek duymaz.
- Bireyleri seçme ve birleştirme aşamalarında belirgin kurallar yerine olasılık kurallarını kullanır. Her çalışmada, GA çözüme yakın ama birbirinden farklı sonuçlar üretir.
- Diğer yöntemlerde olduğu gibi doğrudan parametreler üzerinde çalışmaz. GA' lar, optimize edilecek parametreleri kodlar ve parametreler üzerinde değil, bu kodlar üzerinde işlem yapar.
- Genetik algoritma, kör bir arama yöntemidir. GA en iyilemeye çalıştığı problemle ilgili herhangi bir bilgiye ihtiyaç duymaz. Sistemin matematik

model bilgilerini gerektirmez. Karmaşık matematik hesapları yerine yalnızca giriş-çıkış bilgilerine ihtiyaç duyar.

- Olasılık kurallarına göre çalışırlar. Programın ne kadar iyi çalıştığı önceden kesin olarak belirlenemez. Ancak, olasılıkla hesaplanabilir.

2.3.2. Genetik Algoritmaların Temel İşlevleri

GA' larda üç temel işlev bulunmaktadır: seçme, çaprazlama ve mutasyon. Aşağıda bu işlevler ayrıntılı bir şekilde ele alınmaktadır. Bu işlevler dışında, genetik algoritmalarda kullanılan bazı parametreler de vardır. Örneğin topluluk boyutu ya da birey sayısı. Bu parametre probleme göre belirlenir ve çok iyi seçilmesi gerekir. Parametrelerin çözünürlüğü ya da bit sayısı ise GA' nın başarımını etkileyen önemli bir faktördür. ÇAGA' larda önemli olan bir faktör ise arşiv ya da ikincil topluluk boyutudur ve problem tipine göre iyi ayarlanması gerekmektedir. Tezde kullanılan GA programları kullanıcı tarafından girilen maksimum nesil sayısına göre sonlandırılmaktadır.

2.3.2.1. Bireylerin (ya da Çözüm Önerilerinin) Kodlanması

Genetik algoritmalarda bireylerin kodlanması ikili sayı kodlama, gerçel sayı kodlaması veya gray kodlama biçimlerinden birisi ile yapılır. Bu tezde, bireyler ikili sayı biçiminde kodlanmıştır. Verilen parametreler gerçel sayı ise, bu durumda gerçel sayılar ikili sayılara dönüştürülerek kullanılmıştır.

2.3.2.2. Seçme Mekanizması

Seçme mekanizması, en basit şekilde, puan değerine bağlı olarak, sonraki topluluğun ebeveyn (anne-baba) adayları arasından bazı bireylerin seçilmesi olarak tanımlanır. Diğer bir deyişle, bir ebeveyn topluluğu bir seçme mekanizması tarafından seçilir. Böylece, hangi bireylerin çaprazlama ve mutasyon işlevlerine uğrayacağı belirlenmektedir. Burada sık kullanılan üç seçme mekanizmasından bahsedilecektir:

Rulet çarkı seçimi (Roulette Wheel Selection, RWS), stokastik evrensel örnekleme (Stochastic Universal Sampling, SUS) ve turnuva seçimi (Tournament Selection, TS). Bu yöntemlerden RWS ve SUS puan orantılı seçme mekanizmalarıdır.

Rulet Çarkı Seçimi (RWS): Bu yöntem, puan orantılı bir seçme mekanizmasıdır ve puan değerlerine bağlı olasılık dağılımına göre ebeveyn topluluğunu seçmektedir. Rulet çarkı, aşağıda belirtilen adımlar kullanılarak uygulanabilir (burada tek amaçlı GA için uygulanma biçimi verilmektedir, ÇAGA' larda bir bireyin puanına göre seçilme olasılığı değişmektedir):

Adım 1. Her birey için puan değerleri ($puan(i)$) hesaplanır ($i=1,2,\dots,N$). Burada N topluluk boyutu ya da birey sayısıdır.

Adım 2. Her bireyin puan değerleri birbirine eklenerek topluluğun toplam puanı (TP) hesaplanır:

$$TP = \sum_{i=1}^N puan(i) \quad (2.8)$$

Adım 3. Her birey için, seçilme olasılığı (p_i) hesaplanır:

$$p_i = \frac{puan(i)}{TP} \quad (2.9)$$

Adım 4. Her birey için, toplam (kümülatif) olasılık (q_i) hesaplanır:

$$q_i = \sum_{j=1}^k p_j \quad (2.10)$$

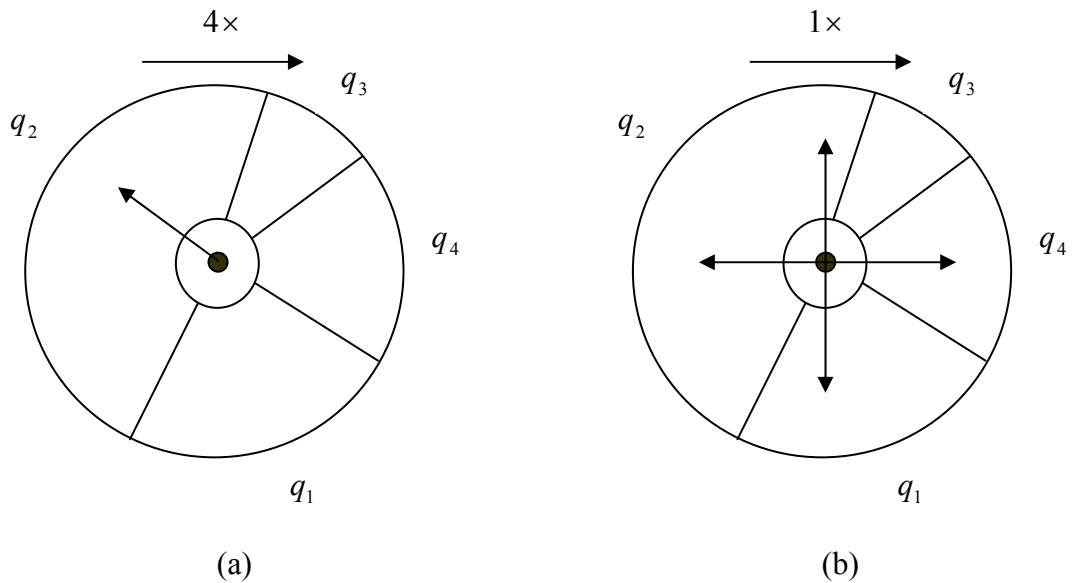
Seçme işlemi rulet çarkının N kez dönderilmesiyle başlar; her seferinde aşağıda belirtilen şekilde tek bir birey yeni topluluk için seçilir.

Adım 5. $[0,1]$ aralığında rasgele bir r sayısı oluşturulur (rulet bir kez dönüyor).

Adım 6. Eğer $r \leq q_1$, o zaman ilk bireyi seç (v_1); aksi takdirde k . birey v_k ($2 \leq k \leq N$) $q_{k-1} < r < q_k$ ile seçilir.

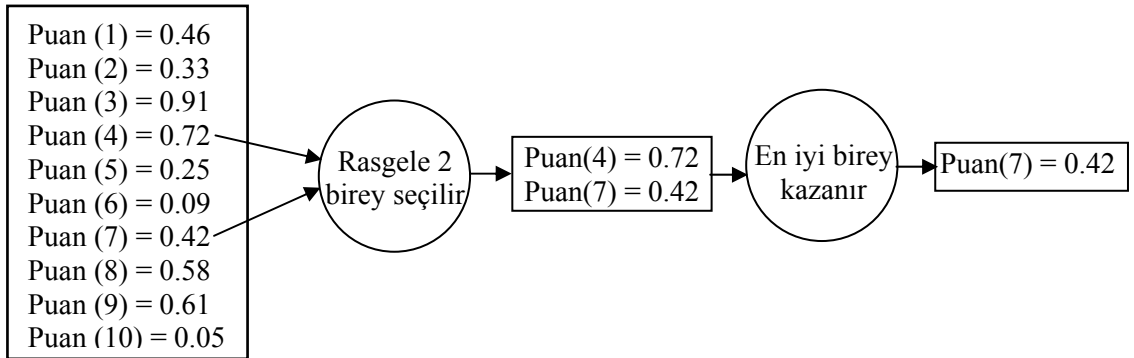
5. ve 6. adım, yeni topluluğu oluşturmak için N kez tekrarlanır. RWS büyük bir seçme hatasının oluşmasına müsaade etmektedir. Bu yöntemde, aynı bireyin N kez seçilebilme olasılığı vardır. RWS' nin gösterilimi (dört bireylik topluluk için) Şekil 2.7a' da verilmektedir.

Stokastik Evrensel Örnekleme (SUS): Rulet çarkının aksine SUS' ta tek bir çark dönüşü kullanır. RWS' deki Adım 1-4 arası ve Adım 6 aynen kullanılır. Asıl fark 5. adımdadır. Çark, bir rulet çarkı gibi oluşturulur ve topluluk boyutuna eşit sayıda işaretçi ile döner. Çarkın bir kez dönüşü seçilen bireyleri belirler. Şekil 2.7b, SUS seçme mekanizmasını (dört bireylik topluluk için) göstermektedir. RWS' deki aynı bireyi N defa seçme olasılığından kurtulmak için, SUS kullanılır. Kaç birey üretilmek isteniyorsa o kadar ok (işaretçi) yerleştirilir ve bir seferde o kadar birey seçilir. Burada oklar arasındaki mesafe eşit olmalıdır. Örneğin 10 bireyden 8 tanesi seçilmek isteniyorsa, aralık 8 eşit parçaya bölünür ve 8 kez rasgele sayı üretilir. SUS, genetik algoritmalarda en çok kullanılan seçme mekanizmasıdır.



Şekil 2.7. (a) Rulet çarkı seçimi (RWS), (b) Stokastik evrensel örnekleme (SUS)

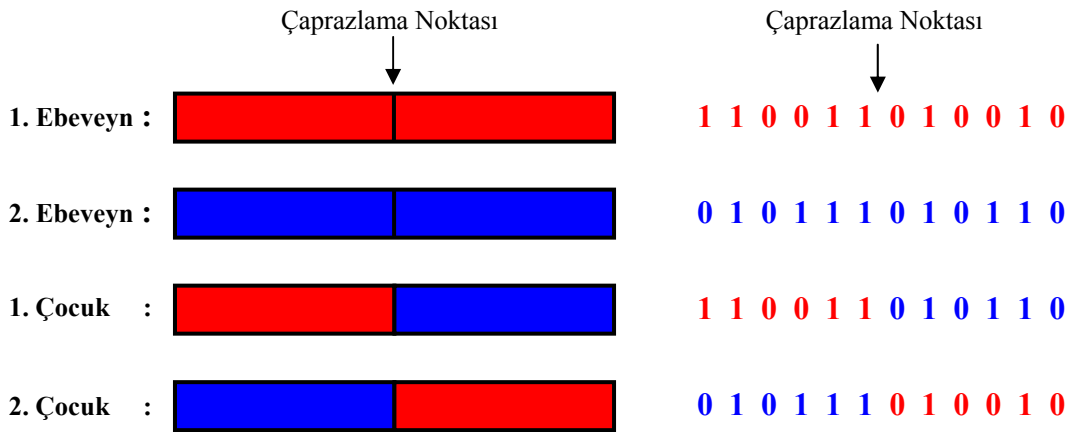
Turnuva Seçimi (TS): En çok kullanılan seçme mekanizmalarından birisi de turnuva seçimidir. Turnuva seçiminde, birkaç adet birey o anki topluluktan rasgele seçilir ve bu bireylerden en iyisi (probleme göre en düşük puanlı ya da en yüksek puanlı) yeni toplulukta yerini alır. Turnuvanın boyutu ayarlanarak, seçilme baskısının miktarı kontrol edilebilir ve böylece yakınsama hızı ayarlanabilir. Turnuva seçimi, temel olarak her seferinde yalnız bir kazanan olacağını varsayar. Seçme işlemi, gelecek topluluğun tamamlanmasına yetecek kadar sık tekrarlanır. Eğer turnuva seçimli basit bir GA düşünüyorsanız ve turnuvanın boyutu bireylerin sayısına eşitse, o zaman seçilen birey o anki topluluktaki en iyi birey olacaktır. Örneğin, 100 bireyden 10 tanesi seçilecekse, rasgele olarak bir grup birey seçilir (10 tane), bunlar bir torbaya konulur, bu 10 tane birey içinden en iyisi alınır ve 1. sıraya konulur. Daha sonra ikinci bir 10 birey seçilir ve bunun en iyisi de 2. sıraya konulur, bu işlem böyle devam eder. Oluşturulacak birey sayısı kadar torba olur. Ancak torbadaki birey sayısı değişebilir. Daha çok bireyden en iyisini seçmek en uygun olanıdır. Turnuva seçimi puan orantılı bir seçilme olasılığı kullanmaz, sadece o anda torbada bulunan en iyi birey seçilir. Şekil 2.8’ de, 10 bireylik bir topluluktaki bireylerin puanları (bir en küçükleme probleminde) ve bu bireyler arasından ikili TS kullanılarak yapılan bir seçme işlemine ilişkin örnek verilmektedir.



Şekil 2.8. İkili turnuva seçimi örneği (en küçükleme problemi)

2.3.2.3. Çaprazlama İşlevi

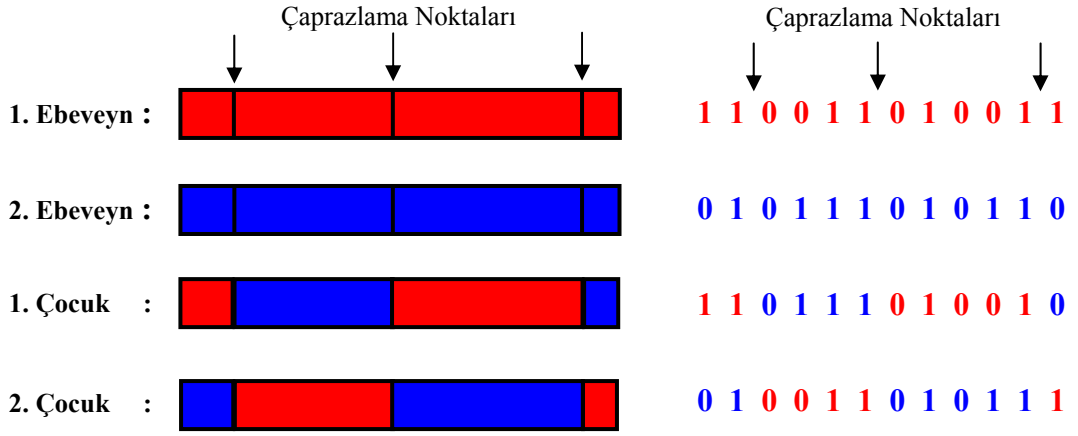
Çaprazlama işlevi, ebeveyn (eşleşme) havuzundan seçilen (seçme mekanizması kullanılarak) bireylerden yeni bireyler (çözümler) üretilmesi için kullanılır. Literatürde bir çok çaprazlama işlevi önerilmiştir, ancak hemen hemen tüm çaprazlama işlevlerinde, ebeveyn havuzundan rasgele seçilen iki birey, ikili kodları bir noktadan kesilerek bölümlere ayrılır ve bu iki bireyin kodları karşılıklı olarak yer değiştirilerek iki yeni birey elde edilir. Çaprazlama işlemi tek noktadan, daha fazla noktadan, düzgün (uniform) veya karışık (shuffled) yapılabilir. Tek noktalı çaprazlamada bireyler rasgele seçilmiş bir noktadan ikiye ayrılır ve karşılıklı ikili kodlar yer değiştirilir. Bu tezde tek-noktalı çaprazlama işlemi kullanılmıştır. Tek noktalı çaprazlama işlemine ilişkin bir örnek Şekil 2.9’ da verilmektedir.



Şekil 2.9. Tek noktalı çaprazlama örneği

Çaprazlama işlemi sonrasında elde edilen bireylerin çaprazlamaya uğrayan bireylerden daha iyi olmaları beklenmektedir, ancak bu her zaman mümkün değildir. Yani ebeveynlere kendilerinden daha iyi bireyler üretebilme şansı verilmektedir, ancak bu her zaman mümkün değildir. Eğer kötü bireyler üretilirse, bunlar zaten büyük olasılıkla bir sonraki GA adımında eleneceklerdir. Eğer çaprazlamaya uğrayacak bireylerin ikili kodları birbirine çok benzerse, çaprazlamadan sonra elde edilen bireyler de birbirine çok benzeyecektir, bu da topluluktaki çeşitliliği azaltacaktır. İkili kodları birbirinden farklı bireylerin çaprazlaması daha iyi sonuçlar verebilir. Usta ve ark., birbirine benzemeyen ikili gövdelere sahip ebeveyn bireylerin, birbirinden daha farklı

(çeşitlilik açısından) bireyler oluşturduğunu ve benzer olanlardan daha iyi sonuçlar ürettiğini belirlemişlerdir [Usta ve ark., 2008]. Çok amaçlı durum söz konusu olduğunda, aynı Pareto cephesinde yer alan ve aralarındaki mesafe (Öklit mesafesi) en büyük olan yani birbirine benzemeyen iki bireyden, çeşitliliği daha yüksek olan bireyler elde edilmiştir [Usta ve ark., 2008]. Eğer çaprazlamayla bireyler arasındaki benzerlik değiştirilemezse devreye mutasyon işlevi girecektir. Eğer çaprazlama olasılığı $p_c=1$ (%100) seçilirse, bu muhtemelen tüm bireylerin çaprazlamaya uğrayacağını göstermektedir. Çaprazlama işlevi birden çok noktadan da uygulanabilir. Aşağıda çok-noktalı bir çaprazlama örneği verilmektedir.

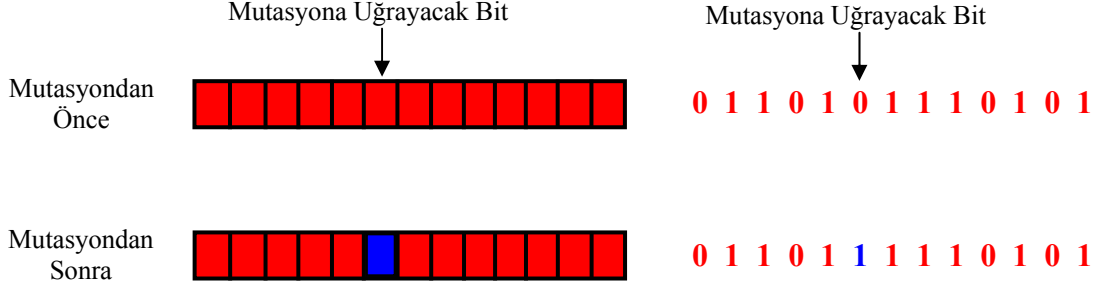


Şekil 2.10. Çok noktalı çaprazlama örneği

2.3.2.4. Mutasyon İşlevi

Mutasyon işlevi de çaprazlama işlevi gibi genetik algoritmanın arama yönünden sorumludur. Bit-tabanlı mutasyon ile, p_m mutasyon olasılığına bağlı olarak bireyin ikili kodundaki rasgele bir bit 0 ise 1, 1 ise 0 yapılır. Eğer rasgele seçilmiş sayı, mutasyon olasılığından düşük ise o takdirde mutasyon işlemi uygulanır. Mutasyon işlevi bir topluluğun çeşitliliğinin sağlanması için gereklidir. Çalışmalar mutasyon olasılığının düşük seçilmesi gerektiğini göstermiştir. Mutasyon olasılığının yüksek seçilmesi çeşitliliği azaltmakta yani farklı bireylerin elde edilebilme olasılığını azaltmaktadır. Genellikle mutasyon olasılığı $1/ikili_dizi_boyutu$ olarak seçilir. Örneğin, 10 bit uzunluklu 20 bireylik bir topluluk için, mutasyon olasılığı 0.01 (%1) seçilirse, bunun

anlamı en azından iki bitin mutasyona uğrayacağıdır. Şekil 2.11’ de mutasyon işlevinin nasıl bir değişikliğe yol açtığı gösterilmektedir.



Şekil 2.11. Bit-tabanlı mutasyon örneği

Mutasyon işlevi ile toplulukta birbirinin aynı bireylerin olması engellenebilir. Ayrıca mutasyona uğratılan bireyden daha iyi puan değerine sahip bir birey elde edilmesi de beklenmektedir. Literatürde birçok mutasyon çeşiti vardır. Bu tezde ise bit-tabanlı mutasyon işlevi kullanılmıştır. Ayrıca, bireyler ikili kodlama yerine gerçel sayılarla da kodlanabilir ve gerçek sayılarla kodlanmış GA’larda da birçok farklı mutasyon işlevi söz konusudur (Deb, 2001).

2.3.2.5. Elitizm (Bellek) Mekanizması

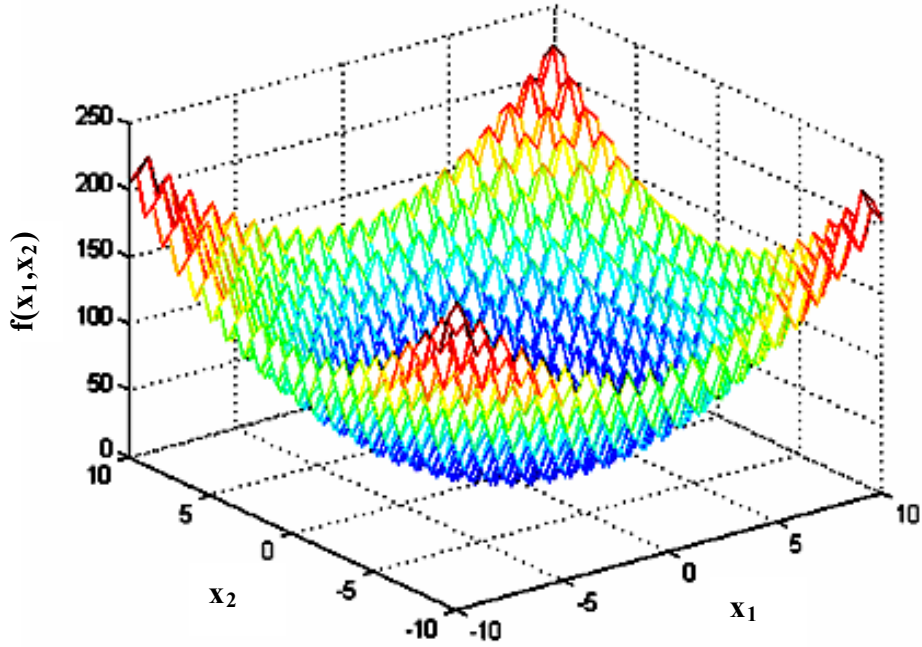
Elit birey, bir toplulukta puanı en iyi olan birey demektir. Elitizm ya da en iyi birey/bireylerin saklanması ve bir sonraki topluluğa eklenmesinin GA’ nın başarımını önemli ölçüde etkilediği görülmüştür [Zitzler, 1999a]. Ancak elitizmin dikkatli bir biçimde uygulanması gerekir. Eğer elitizm kontrollü bir biçimde uygulanmazsa çeşitlilik kaybı söz konusu olabilir. Tek amaçlı GA’larda yalnızca bir tek en iyi birey olduğu için elitizmin uygulanması kolaydır. Bu durumda elit birey saklandıktan sonra, bir sonraki adımda ya en kötü bireyle yer değiştirilmekte ya da basitçe topluluğun en sonundaki bireyle yer değiştirilmektedir. Ancak çok amaçlı GA’larda tek bir en iyi birey yerine en iyi bireylerden oluşan bir küme söz konusudur. Bu durumda elitizmin uygulanması tek amaçlı durumda olduğu gibi kolay ve tek bir biçimde değildir. ÇAGA’larda elit bireyler arşiv veya ikincil topluluk denilen bir kümede saklanmakta ve ÇAGA yöntemlerine göre farklılaşan stratejilerle bir sonraki topluluğa aktarılmaktadırlar.

2.3.3. Tek Amaçlı Genetik Algoritmalar

Tek amaçlı genetik algoritmalar tek amaçlı optimizasyon problemlerini kolaylıkla çözebilecek bir yapıya sahiptirler. Tek amaçlı genetik algoritmaların dolayısıyla GA'ların nasıl çalıştığını daha iyi anlatmak için, bu bölümde tek amaçlı ancak iki parametrelili bir problem ayrıntılı olarak GA yardımıyla çözülmektedir [Michalewicz, 1994; Gen ve ark., 1997]. Daha sonra bu problem çeşitli GA parametreleri ile çalıştırılarak çözümler sunulmaktadır. Parametreler ikili sayı olarak kodlanmışlardır. Kısıtlamasız bir problem (Rastrigin işlevi) analitik olarak aşağıdaki gibi verilmektedir:

$$\min f(x_1, x_2) = (x_1^2 - 10 \cos(2\pi x_1) + 10) + (x_2^2 - 10 \cos(2\pi x_2) + 10) + 5 \quad (2.10)$$

Burada f amaç işlevinin minimumu (tek amaç), genetik algoritma kullanılarak bulunmak istenmektedir. f amaç işlevi iki parametreye bağlıdır (x_1 ve x_2). Amaç işlevinin grafiği Şekil 2.12' de gösterilmektedir.



Şekil 2.12. Amaç işlevi örneği (Rastrigin işlevi)

Şekle bakılırsa veya amaç işlevi analitik olarak çözülürse, bu işlevin minimum değerinin 5 olduğunu görülür. Buna göre, tek amaçlı genetik algoritmanın bir adımının nasıl olduğu aşağıda ayrıntılı bir şekilde verilmektedir.

2.3.3.1. Çözüm Adaylarının (Bireylerin) Gösterilimi

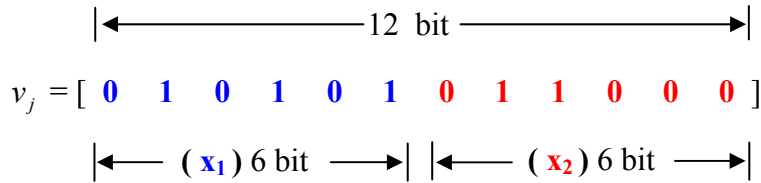
GA' nın ilk aşaması karar değişkenlerinin ya da parametrelerin ikili sayı dizileri şeklinde kodlanmasıdır (gerçel sayı veya gray kodla da kodlama yapılabilir) [Eminoğlu, 2003]. Eğer bir x_j parametresi için istenilen kesinlik değeri biliniyorsa (örneğin 1000), o zaman gerekli bit sayısı (m_j olarak gösterelim) şöyle hesaplanabilir:

$$2^{m_j} - 1 < (b_j - a_j) \times 10^3 \leq 2^{m_j} - 1 \quad (2.11)$$

Burada a_j , b_j ; x_j (x_1, x_2)' lerin alt ve üst sınırlarıdır. Böylece 6 bitlik parametre $2^6 - 1 = 63$ aralığa bölünmüş olur. İkili bir sayıyı gerçel sayıya çevirmek için aşağıdaki işlem yapılır (x_j parametrelerdir):

$$x_j = a_j + \text{ondalik}(x_j) \times \frac{b_j - a_j}{2^{m_j} - 1} \quad (2.12)$$

Sınırlara göre temsil edilen bit sayısı değişmelidir. Sınır büyükse daha fazla bit, küçükse daha az bit parametrenin temsili için kullanılır. Bu örnekte her parametre 6 bitlik ikili sayılar olarak kodlanmıştır. Yalnızca iki parametre olduğu için, bir bireyin (ya da çözüm önerisinin) boyutu toplam 12 bit olarak gösterilir:



x_1 ve x_2 'nin değerleri aşağıdaki gibidir:

	<u>İkili Sayı</u>						<u>Ondalık Sayı</u>
x_1	0	1	0	1	0	1	21
x_2	0	1	1	0	0	0	24

x_1 ve x_2 'nin gerçek değerleri aşağıdaki gibi hesaplanır:

$$x_1 = -10 + 21 \times \frac{10 - (-10)}{2^6 - 1} = -3.3333 \quad x_2 = -10 + 24 \times \frac{10 - (-10)}{2^6 - 1} = -2.3810 \quad (2.13)$$

Bu hesaplamalar topluluğun geri kalan bireyleri için de aynıdır. Aşağıda basit bir GA programının yalnızca bir adımı (iterasyonu) gösterilmektedir.

Adım 1. Başlangıç topluluğu (10 bireyden oluşan) aşağıdaki gibi rasgele oluşturulur:

	x_1						x_2					
v_1	0	1	0	1	0	1	0	1	1	0	0	0
v_2	1	0	0	1	1	1	1	1	0	1	0	1
v_3	0	0	0	1	1	0	1	1	1	0	0	1
v_4	0	1	1	0	1	1	1	0	0	1	0	0
v_5	0	1	0	0	1	1	0	0	1	1	1	1
v_6	1	0	1	0	1	1	0	0	0	1	1	1
v_7	1	1	0	1	0	0	1	0	0	1	1	1
v_8	0	1	1	0	1	1	1	0	1	1	1	1
v_9	0	0	0	0	0	0	1	0	1	1	0	0
v_{10}	1	0	0	0	0	1	1	0	1	1	1	0

Adım 2. Başlangıç topluluğundaki bireylerin ondalık değerleri hesaplanır:

$$\begin{aligned}
 v_{1o} &= [x_1, x_2] = [21, 24] & v_{2o} &= [x_1, x_2] = [39, 53] & v_{3o} &= [x_1, x_2] = [6, 57] \\
 v_{4o} &= [x_1, x_2] = [27, 36] & v_{5o} &= [x_1, x_2] = [19, 15] & v_{6o} &= [x_1, x_2] = [43, 7] \\
 v_{7o} &= [x_1, x_2] = [52, 39] & v_{8o} &= [x_1, x_2] = [27, 47] & v_{9o} &= [x_1, x_2] = [0, 44] \\
 v_{10o} &= [x_1, x_2] = [33, 46]
 \end{aligned}$$

Adım 3. Başlangıç topluluğundaki tüm parametrelerin gerçek değerleri hesaplanır:

$$\begin{aligned}
 v_1 &= [x_1, x_2] = [-3.3333, -2.3810] & v_2 &= [x_1, x_2] = [2.3810, 6.8254] \\
 v_3 &= [x_1, x_2] = [-8.0952, 8.0952] & v_4 &= [x_1, x_2] = [-1.4286, 1.4286] \\
 v_5 &= [x_1, x_2] = [-3.9683, -5.2381] & v_6 &= [x_1, x_2] = [3.6508, -7.7778] \\
 v_7 &= [x_1, x_2] = [6.5079, 2.3810] & v_8 &= [x_1, x_2] = [-1.4286, 4.9206] \\
 v_9 &= [x_1, x_2] = [-10.0000, 3.9683] & v_{10} &= [x_1, x_2] = [0.4762, 4.6032]
 \end{aligned}$$

Adım 4. Tüm bireylerin amaç işlevinde aldığı değerler hesaplanır:

$$\begin{aligned}
 f(v_1) &= f(-3.3333, -2.3810) = 49.1106 & f(v_2) &= f(2.3810, 6.8254) = 75.0234 \\
 f(v_3) &= f(-8.0952, 8.0952) = 134.541 & f(v_4) &= f(-1.4286, 1.4286) = 42.1010 \\
 f(v_5) &= f(-3.9683, -5.2381) = 52.6357 & f(v_6) &= f(3.6508, -7.7778) = 97.9231 \\
 f(v_7) &= f(6.5079, 2.3810) = 85.3403 & f(v_8) &= f(-1.4286, 4.9206) = 46.4809 \\
 f(v_9) &= f(-10.000, 3.9683) = 115.9453 & f(v_{10}) &= f(0.4762, 4.6032) = 59.2756
 \end{aligned}$$

Asıl amaç; $f(x_1, x_2)$ 'nin minimumunu ve rasgele oluşturulan başlangıç topluluğunun en iyi çözümünü ($f(v_4) = 42.1010$) bulmaktır. İkinci aşama ise, bireylerin puanlarının bulunmasıdır.

Adım 5. Bireylerin Puanlarının Hesaplanması

Amaç işlevinin değeri puan değerine dönüştürülür. $f(v_j)$ 'nin en küçük değeri tüm topluluk için en iyi çözümdür. Aslında, $f(v_j)$ aşağıdaki gibi bir en büyükleme problemine dönüştürülebilir (Böylece en yüksek puana sahip birey bulunabilir ve bireyler en iyiden en kötüye doğru sıralanabilir, en iyi bireyin puanı yüksek, en kötü bireyin puanı ise düşük olacaktır):

$$Puan_j = \frac{K_1}{f(v_j)} + K_2 \quad (2.14)$$

Burada $K_1 = 10$ ve $K_2 = 0.1$ seçilmiştir. K_2 sifıra bölme hatalarından kurtulmak için kullanılmıştır. $Puan_j$ (her bir bireyin puanı) aşağıdaki gibi hesaplanır: ($j=1 \dots 10$)

Puan_1 = 0.2032	Puan_6 = 0.1020
Puan_2 = 0.1331	Puan_7 = 0.1170
Puan_3 = 0.0743	Puan_8 = 0.2147
Puan_4 = 0.2370	Puan_9 = 0.0862
Puan_5 = 0.1896	Puan_10 = 0.1684

Adım 6. Puanların Normalize Edilmesi

Mutlak puan değerlerindense ($Puan_j$) normalize puan değerlerinin ($Puan_j_n$) kullanılması daha uygun olur. Normalize puan değerleri aşağıdaki gibi hesaplanır:

$$Puan_j_n = \frac{Puan_j}{En_Yüksek_Puan_j} \quad (2.15)$$

Burada $En_Yüksek_Puan_j$, $Puan_j$ ($j=1 \dots 10$) 'nin en yüksek değeridir. Buna göre bu örneğe ilişkin normalize puanlar şöyledir:

Puan_1_n = 0.8576	Puan_6_n = 0.4305
Puan_2_n = 0.5618	Puan_7_n = 0.4939
Puan_3_n = 0.3134	Puan_8_n = 0.9060
Puan_4_n = 1.0000	Puan_9_n = 0.3637
Puan_5_n = 0.8002	Puan_10_n = 0.7107

Açıkça görülmektedir ki, v_4 bireyi en güçlü, v_3 bireyi ise en zayıf bireydir. Bireylerin normalize puan değerleri onların ebeveyn olarak seçilebilme şanslarıyla doğrudan ilişkilidir (Bu, rulet çarkı benzeri bir oransal seçim mekanizması kullanıldığı sürece geçerlidir). Bir bireyin bağıl olasılığı (ebeveyn olarak seçilmek için) diğer bireylerle karşılaştırılır, toplam puan değeri burada hayati bir ölçüm sağlar.

Adım 7. Puanların Toplamı

Puanların toplamı aşağıdaki gibi hesaplanır:

$$Toplam_puan = \sum_{j=1}^{10} Puan_j_n = 6.4378 \quad (2.16)$$

Adım 8. Her Bir Bireyin Seçilme Olasılığı ve Toplam (kümülatif) Olasılığı

Her bir v_j ($j=1,\dots,10$) bireyi için seçilme olasılığı p_j aşağıdaki formülle hesaplanır:

$$p_j = \frac{Puan_j_n}{Toplam_Puan} \quad (2.17)$$

Buna göre birey puanları şöyle olur:

$$\begin{array}{lllll} p_1 = 0.1332 & p_2 = 0.0873 & p_3 = 0.0487 & p_4 = 0.1553 & p_5 = 0.1243 \\ p_6 = 0.0669 & p_7 = 0.0767 & p_8 = 0.1407 & p_9 = 0.0565 & p_{10} = 0.1104 \end{array}$$

Her bir birey v_j ($j=1,\dots,10$) için, kümülatif (toplanarak artan) olasılıklar

$q_j = \sum_{i=1}^j p_i$ şöyledir:

$$\begin{array}{lllll} q_1 = 0.1332 & q_2 = 0.2205 & q_3 = 0.2692 & q_4 = 0.4245 & q_5 = 0.5488 \\ q_6 = 0.6157 & q_7 = 0.6924 & q_8 = 0.8331 & q_9 = 0.8896 & q_{10} = 1 \end{array}$$

Kümülatif (toplanarak artan) olasılığın kullanılma nedeni; olasılıklar toplamı **1** olduğu için buna dönüşüm yapılması gerekliliğidir. Yani en büyük olasılık değeri **1** olur. Seçme mekanizmasındaki çarktaki dilim ne kadar büyük olursa, üretilen rasgele sayının bu dilime düşme olasılığı daha yüksek olacaktır. Böylece seçilme olasılığı da yüksek olacaktır.

Adım 9. Seçme Mekanizması Olarak Rulet Çarkı Yöntemi

Rulet çarkı 10 kez döndürülür (10 birey olduğu için) ve her seferinde yeni topluluk için (ebeveyn topluluğu) tek bir birey seçilir. Rasgele 10 sayı dizisi, $[0,\dots,1]$ aralığında Çizelge 2.1' deki gibi oluşturulsun. İlk rasgele sayı $r_1 = 0.7413$, q_7 ' den büyük, q_8 ' den küçüktür, bunun anlamı v_8 kromozomu yeni topluluk için seçilmiştir demektir. İkinci rasgele sayı $r_2 = 0.0596$, q_1 ' den küçüktür, bunun anlamı v_1 kromozomunun yeni topluluk için seçileceğidir ve bu böyle devam eder. Çizelge 2.1, yeni topluluk için seçilen bireyleri göstermektedir.

Çark N (birey sayısı ya da topluluk boyutu) defa döndürülürken, 1. dönüşten sonra 2., sonra 3. şeklinde dönüş yapılır. Bu yöntemin dezavantajı, aynı bireyi N defa seçme olasılığı olmasıdır. Bu da, çaprazlamaya aynı bireylerin girmesine, dolayısıyla da aynı bireylerin yeniden oluşturulmasına neden olacaktır ve topluluğun çeşitliliği azalacaktır. Çeşitliliğin mümkün olduğu kadar fazla olması gerekir. Çünkü bu sağlanmazsa, arama yerel en iyilere takılacaktır, GA' nın istenilen sonucu (küresel en iyiyi) bulması mümkün olmayacaktır.

Çizelge 2.1. Rasgele sayılar ve seçilen bireyler

Rasgele Sayı	Seçilen Birey	Seçilme Nedeni
$r_1 = 0.7413$	v_8	$q_7 < r_1 < q_8$
$r_2 = 0.0596$	v_1	$r_2 < q_1$
$r_3 = 0.4673$	v_5	$q_4 < r_3 < q_5$
$r_4 = 0.1363$	v_2	$q_1 < r_4 < q_2$
$r_5 = 0.9340$	v_{10}	$q_9 < r_5 < q_{10}$
$r_6 = 0.6286$	v_7	$q_6 < r_6 < q_7$
$r_7 = 0.4983$	v_5	$q_4 < r_7 < q_5$
$r_8 = 0.5760$	v_6	$q_5 < r_8 < q_6$
$r_9 = 0.9933$	v_{10}	$q_9 < r_9 < q_{10}$
$r_{10} = 0.0542$	v_1	$r_{10} < q_1$

Adım10. Yeni topluluğun (veya ebeveyn topluluğunun) gösterilimi

$$v'_1 = [0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1] (v_8)$$

$$v'_2 = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0] (v_1)$$

$$v'_3 = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1] (v_5)$$

$$v'_4 = [1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1] (v_2)$$

$$v'_5 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0] (v_{10})$$

$$v'_6 = [1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1] (v_7)$$

$$v'_7 = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1] (v_5)$$

$$v'_8 = [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1] (v_6)$$

$$v'_9 = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0] (v_{10})$$

$$v'_{10} = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0] (v_1)$$

Yukarıdan da görüleceği üzere en güçlü birey (v_4) rulet çarkı mekanizmasıyla seçilmemiştir ve oluşan yeni toplulukta **1., 5. ve 10.** bireylerden ikiyeşer kez üretilmiştir. Bu yüzden, bu topluluktan üretilen bireyler de aynı olacaktır. Bu nedenle genetik

işlevlere başvurulur, böylece çeşitlilik sağlanır. En güçlü bireyin seçilmemesi ise, elitizmin (en güçlü bireyin saklanması) gerekliliğinin açıklanması için çok yararlı olacaktır.

Adım 11. Çaprazlama İşlevi

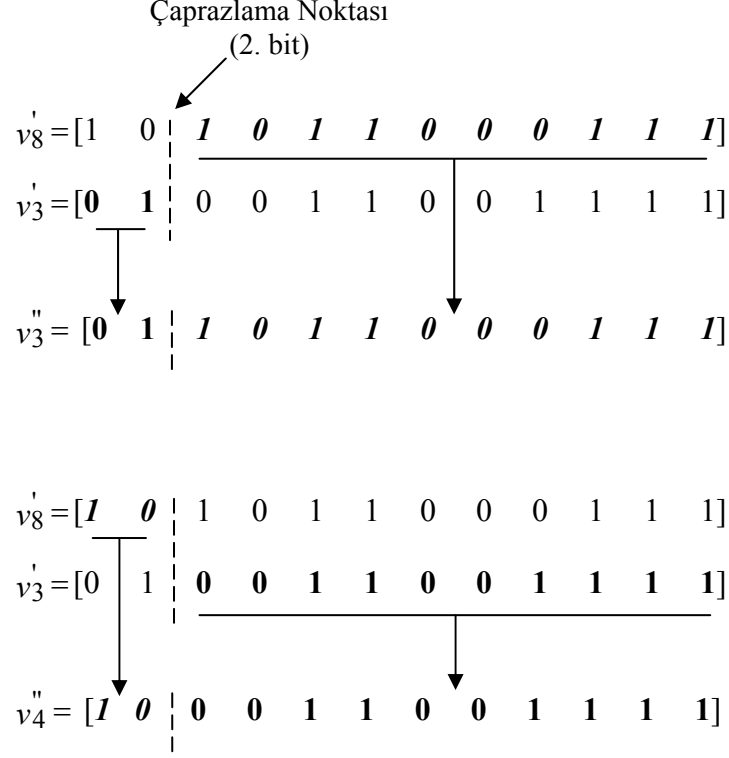
Şimdi, ebeveyn topluluğundaki bireyler için çaprazlama işlevi uygulanabilir. Çaprazlama olasılığı $p_c = 1$ seçilsin, dolayısıyla topluluktaki tüm bireylerin çaprazlamaya uğrayacağını beklenmektedir. Çaprazlama katsayısı 1 seçilmez ise, örneğin 0.8, 8 birey çaprazlamaya girecek, topluluğun tamamı için geriye kalan 2 bireyin saklanması ve işlem sokulması gerekecektir. Burada değişik bir çaprazlama yöntemi benimsenmiştir. Rasgele sayılar (r) $[0, \dots, 1]$ aralığında oluşturulmuş ve çaprazlama olasılığı (p_c) ile karşılaştırılmıştır. Rasgele sayıların oluşturulması ve karşılaştırılması bireylerin yarısı için tekrarlanmıştır. Eğer $r < p_c$ memnun edici ise, o zaman üç rasgele pozitif tamsayı oluşturulur. İlk ikisi, ebeveyn kromozomlarını belirler ve sonuncu rasgele tamsayı ise çaprazlama noktasını gösterir. Her ebeveyn, kendi ikili biçimini koruyarak iki çocuk / birey oluşturabilir.

Çaprazlama sadece iki birey arasında değil, daha fazla birey arasında da olabilir. Ayrıca çaprazlama birkaç noktadan da yapılabilir. Burada tek noktalı çaprazlama kullanılmıştır.

Çizelge 2.2. Ebeveyn bireyler, çaprazlama noktası ve çocuk bireyler

1. Ebeveyn	2.Ebeveyn	Çaprazlama Noktası	Çocuklar
3 (v_3')	7 (v_7')	8. bitten sonra	v_1'' ve v_2''
8 (v_8')	3 (v_3')	2. bitten sonra	v_3'' ve v_4''
7 (v_7')	8 (v_8')	6. bitten sonra	v_5'' ve v_6''
2 (v_2')	8 (v_8')	5. bitten sonra	v_7'' ve v_8''
5 (v_5')	5 (v_5')	7. bitten sonra	v_9'' ve v_{10}''

İkinci ebeveyn çifti v_8' ve v_3' 'dür ve çaprazlama noktası **2.** bitten sonradır. Çocuklar çaprazlama sonucu elde edilir (v_3'' ve v_4''). İlk çocuk (v_3''), ilk iki bitini ikinci ebeveynden (v_3'), diğer bitlerini ise ilk ebeveynden (v_8') alır.



Benzer olarak ikinci çocuk ta (v_4''), ilk iki bitini birinci ebeveynden (v_8'), diğer bitlerini ise ikinci ebeveynden (v_3') alır. Benzer olarak, çocukların geri kalanı da oluşturulur ve topluluğun şu andaki biçimi (çocuk topluluğu) şöyle olur :

$$\begin{aligned}
 v_1'' &= [0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1] \\
 v_2'' &= [0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1] \\
 v_3'' &= [0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1] \\
 v_4'' &= [1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1] \\
 v_5'' &= [1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1] \\
 v_6'' &= [0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1] \\
 v_7'' &= [1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0]
 \end{aligned}$$

$$v_8'' = [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1]$$

$$v_9'' = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0]$$

$$v_{10}'' = [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0]$$

Çaprazlama olasılığı **1** olmasına rağmen (yani tüm bireyler çaprazlamaya uğramasına rağmen), hala çocuk topluluğundaki bazı bireyler (1. ile 2. ve 9. ile 10. bireyler) aynı bitleri taşımakta ve ebeveyn topluluğundan değişiklik olmadan gelmektedir. Bu nedenle, mutasyon işlevinin uygulanması gerekmektedir. Bir sonraki aşama, mutasyon işlevinin topluluğu nasıl değiştireceğini göstermektedir.

Adım 12. Mutasyon İşlevi

Çeşitliliğin fazla olması istendiğinden, çaprazlamadan sonra mutasyon işlevi kullanılır. Mutasyon oranı düşük seçilir (örneğin %1 gibi). Her adım sonucunda amaca ne kadar yaklaşıldığına bakılması gerekir. Çaprazlama ve mutasyon, daha iyi bir çözüm elde edileceğini garanti etmez. Mutasyon işlevi, bit bit değişim temeline dayanır. Mutasyon, mutasyon oranına eşit bir olasılıkla, bir veya daha fazla geni (biti) değiştirir. Çocuk topluluğundaki ikinci bireyin (v_2'') 11. geninin (bitinin) mutasyon için seçildiğini varsayalım. Eğer gen **1** ise **0** olacaktır. Böylece, mutasyondan önce ve sonra birey şöyle olacaktır:

$$v_2'' = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad \boxed{1} \quad 1] \quad (\text{mutasyondan önce})$$

$$v_2'' = [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad \mathbf{0} \quad 1] \quad (\text{mutasyondan sonra})$$

Mutasyon olasılığı $p_m = 0.1$ (bu kasten böyle yüksek bir değer seçilmiştir) ise, buradan bitlerin %10' unun mutasyona uğrayabileceği anlaşılmaktadır. Çocuk topluluğunda 120 bit vardır (12 bit x 10 birey, bu yüzden topluluk başına 12 bitin mutasyona uğrayacağı beklenmektedir). Her bit mutasyona uğramak için aynı şansa sahiptir. Bu yüzden $[0, \dots, 1]$ aralığında değişecek bir rasgele sayılar dizisi r_j ($j=1, \dots, 120$) oluşturulması gerekmektedir. Kaç bit varsa o kadar rasgele sayı üretilir. Eğer mutasyon

olasılığı j . rasgele sayıdan daha büyükse ($p_m > r_j$), j . bit mutasyona uğrayacaktır.

Çocuk topluluğunun mutasyona uğramış biçimi aşağıdaki gibidir:

$$\begin{aligned}
 v_1'' &= [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1] \\
 v_2'' &= [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad \mathbf{0} \quad 1] \\
 v_3'' &= [0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad \mathbf{1} \quad 0 \quad 0 \quad 1 \quad 1 \quad 1] \\
 v_4'' &= [1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad \mathbf{0}] \\
 v_5'' &= [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad \mathbf{0}] \\
 v_6'' &= [0 \quad 1 \quad 0 \quad \mathbf{1} \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad \mathbf{0}] \\
 v_7'' &= [1 \quad 0 \quad 1 \quad \mathbf{1} \quad 1 \quad 1 \quad \mathbf{1} \quad 1 \quad 1 \quad 0 \quad 0 \quad 0] \\
 v_8'' &= [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad \mathbf{0} \quad 0 \quad 0 \quad 0 \quad 1 \quad \mathbf{0} \quad \mathbf{0}] \\
 v_9'' &= [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad \mathbf{0} \quad 1 \quad 1 \quad 0] \\
 v_{10}'' &= [1 \quad 0 \quad \mathbf{1} \quad 0 \quad 0 \quad 1 \quad 1 \quad \mathbf{1} \quad 1 \quad 1 \quad 1 \quad \mathbf{1}]
 \end{aligned}$$

Koyu renk bitler mutasyona uğramıştır. 120 bitten 14 tanesi mutasyona uğramıştır. Bu %10 olarak beklenen mutasyondan fazladır. Toplam 10 bireyden 9 tanesi mutasyona uğramıştır. Bazı bireylerde yalnızca bir bit, bazılarında ise iki veya üç bit mutasyona uğramıştır. Dikkat edilirse topluluğun mutasyona uğramış biçiminde aynı bireylerin artık olmadığı kolayca anlaşılır. Her birey diğerinden farklıdır, böylece çeşitlilik sağlanmıştır.

Adım 13. Elitizm veya Bellek Mekanizması (En İyi Bireyin Saklanması)

Rasgele oluşturulan birinci topluluktaki en güçlü bireyin (v_4), ebeveyn olarak seçilmediği görülmektedir. Bu, büyük olasılıkla, rulet çarkı seçme mekanizmasının olasılık tabanlı örnekleme hatasına bağlıdır. Diğer bir deyişle, buradaki seçme algoritması, tüm durumlarda en güçlü bireyin ebeveyn olarak seçilmesini garanti etmemektedir. Bir ebeveyn olarak seçilse bile, onun çocuğunun kendisinden daha güçlü olacağının garantisi yoktur. Bu yüzden, topluluktaki her en iyi bireyi ayrı bir yerde

saklamak gereklidir. Bellek olması, en iyilerin saklanması içindir. Genetik algoritmalar belleksizdir, bu yüzden bellek eklenmesi gerekir. Elitizmin, GA' nın başarımına etkisi çok yüksektir ve bu yüzden GA' nın doğal bir parçası olması gerekir. İlk topluluktaki en güçlü birey (v_4), basitçe mutasyona uğramış topluluktaki sonuncu bireyle (v_{10}'') yer değiştirilir. Aslında bunun en kötü bireyle yer değiştirilmesi daha iyi olur ancak bu algoritmaya ekstra bir yük getirmektedir. Topluluğun şimdiki durumu şöyledir:

$$\begin{aligned}
v_1'' &= [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1] \\
v_2'' &= [0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1] \\
v_3'' &= [0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1] \\
v_4'' &= [1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0] \\
v_5'' &= [1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0] \\
v_6'' &= [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0] \\
v_7'' &= [1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0] \\
v_8'' &= [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0] \\
v_9'' &= [1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0] \\
v_{10}'' &= [0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0] (v_4)
\end{aligned}$$

Üstteki topluluk dekodlanır ve her bireyin gerçek değeri aşağıdaki gibi hesaplanır:

$$\begin{aligned}
f(v_1'') &= f(-3.9683, -5.2381) = 52.6357 & f(v_2'') &= f(-3.9683, -5.8730) = 53.4553 \\
f(v_3'') &= f(-1.4286, 2.3810) = 44.0500 & f(v_4'') &= f(1.1111, -5.5556) = 53.8352 \\
f(v_5'') &= f(3.6508, -1.7460) = 42.4637 & f(v_6'') &= f(-2.6984, -8.0952) = 87.7368 \\
f(v_7'') &= f(4.9206, 7.7778) = 94.1878 & f(v_8'') &= f(-3.6508, -8.7302) = 116.6248 \\
f(v_9'') &= f(0.4762, 2.0635) = \mathbf{25.1583} & f(v_{10}'') &= f(-1.4286, 1.4286) = 42.1010
\end{aligned}$$

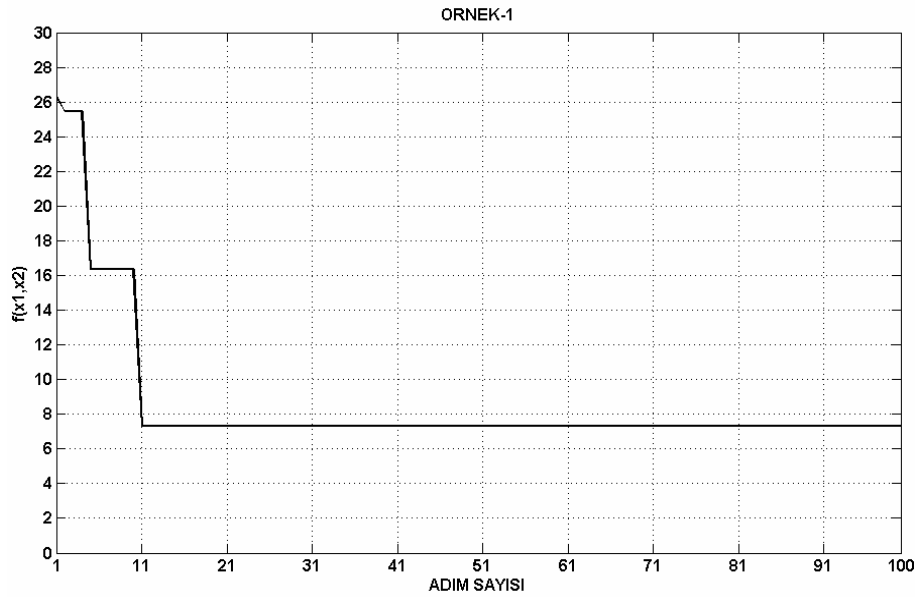
Hedef, $f(x_1, x_2)$ işlevinin minimumunu bulmaktır. Yeni en güçlü birey (v_9''), ilk elit (en iyi) bireyden ($v_4 = v_{10}''$) daha iyidir. Bu topluluğun en zayıf bireyi de (v_8''), ilk topluluktaki en zayıf bireyden (v_3) daha iyidir.

Burada genetik algoritmanın bir adımı tamamlanmıştır. Girilen adım sayısına göre program adımları çalıştırılacak ve yeni topluluklar elde edilecektir. Her bir adımdan sonra daha iyi bireyler elde edildiği için belli bir adım (nesil) sayısı veya sonlandırma kriterine göre istenilen sonuca ulaşılabacaktır.

Aşağıda birkaç adet örnekle tek amaçlı genetik algoritmanın başarımı incelenmektedir.

Örnek-1: GA parametreleri şöyle seçilmiştir; (elitizm var, SUS seçme mekanizması)

Birey Sayısı : 20
 Toplam Bit Sayısı : 10
 Çaprazlama Olasılığı : 1
 Mutasyon Olasılığı : 0.1
 Adım (Nesil) Sayısı : 100



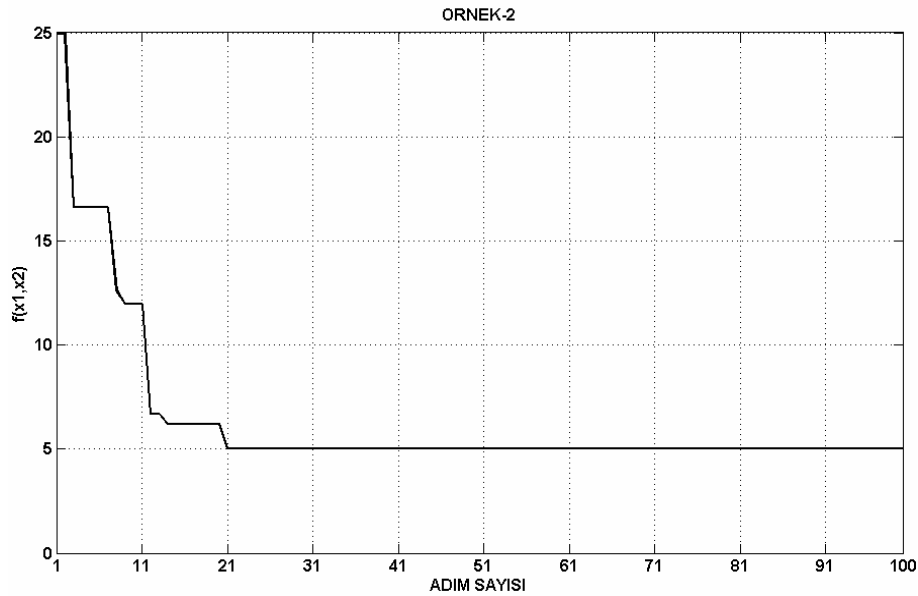
Şekil 2.13. Tek amaçlı GA uygulaması (Örnek-1)

Görüleceği üzere GA, 11. adımdan sonra **7.2824** değerine takılmış ve sonucu bulamamıştır. Bunun nedeni çözünürlük yani bit sayısının az seçilmesidir. Eğer bu değer artırılırsa GA' nın çözüme daha çabuk ve kesin olarak yaklaşacağı beklenmektedir.

Örnek-2: Bu örnek için GA parametreleri şöyle seçilmiştir; (elitizm var, SUS seçme mekanizması)

Birey Sayısı : 20
 Toplam Bit Sayısı : 20
 Çaprazlama Olasılığı : 1
 Mutasyon Olasılığı : 0.1
 Adım (Nesil) Sayısı : 100

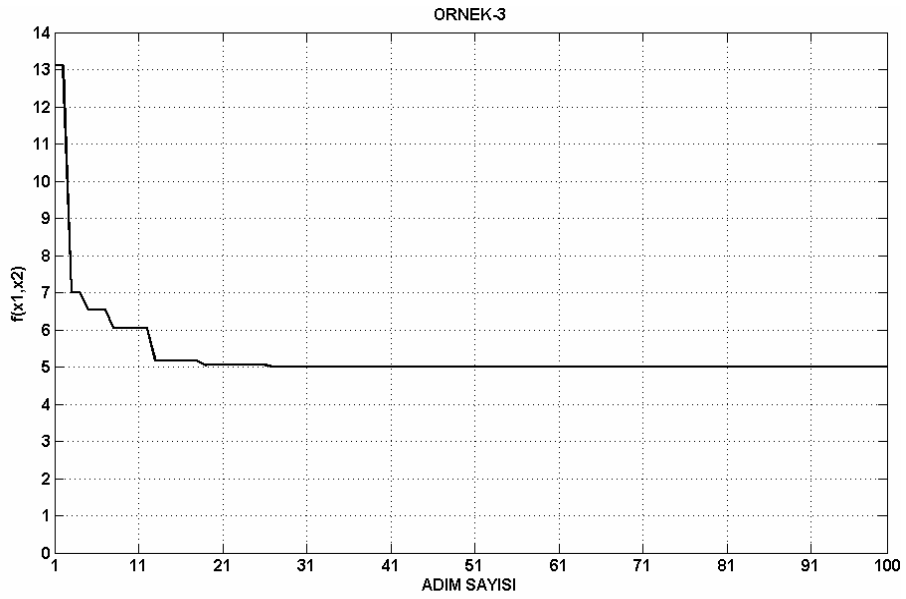
Bu örnekte GA, 21. adımda en küçük sonuca ulaşmıştır. Burada elde edilen sonuç **5.0379**' dur. Toplulukta yer alan bireylerin sayısı veya bit sayısı artırıldığında, GA' nın çözüme daha da çabuk yaklaştığı görülmektedir. Bu değerler çok artırılırsa GA' nın çalışma süresi de önemli ölçüde artacaktır.



Şekil 2.14. Tek amaçlı GA uygulaması (Örnek-2)

Örnek-3: Bu örnek için GA parametreleri şöyle seçilmiştir; (elitizm var, SUS seçme mekanizması)

Birey Sayısı	: 100
Toplam Bit Sayısı	: 30
Çaprazlama Olasılığı	: 1
Mutasyon Olasılığı	: 0.1
Adım (Nesil) Sayısı	: 100



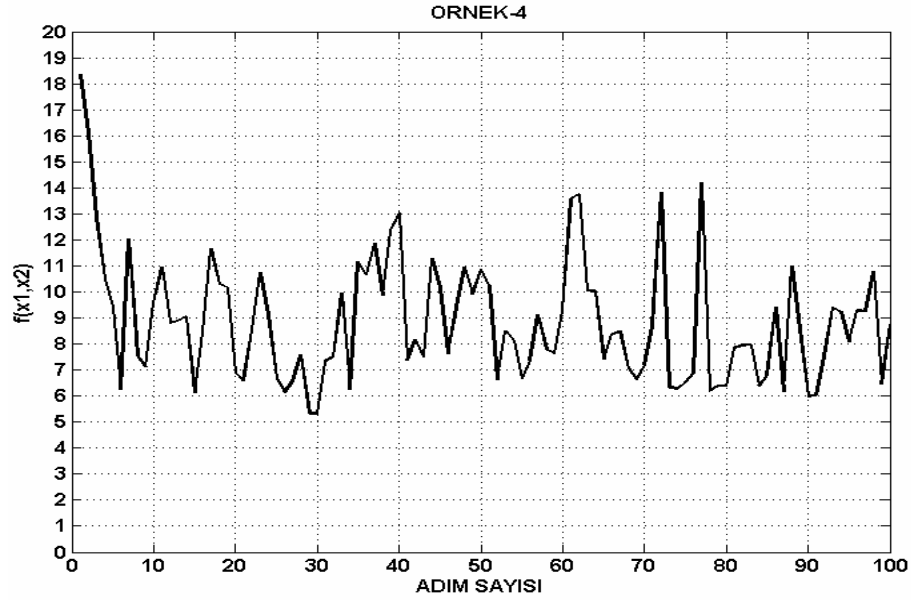
Şekil 2.15. Tek amaçlı GA uygulaması (Örnek-3)

Şekil 2.15' den görüleceği üzere, GA 38. adımda sonuca ulaşmıştır. Bu durumda GA tarafından elde edilen en küçük sonuç **5.0000369**' tur. GA, bu örneklerde belirli bir adıma göre sonlandırılmıştır. Sonlandırma için istenilen en küçük yaklaşım (hata) da kullanılabilir.

Örnek-4: Bu örnek için GA parametreleri şöyle seçilmiştir; (elitizm yok, SUS seçme mekanizması)

Birey Sayısı	: 100
Toplam Bit Sayısı	: 30

Çaprazlama Olasılığı : 1
 Mutasyon Olasılığı : 0.1
 Adım (Nesil) Sayısı : 100

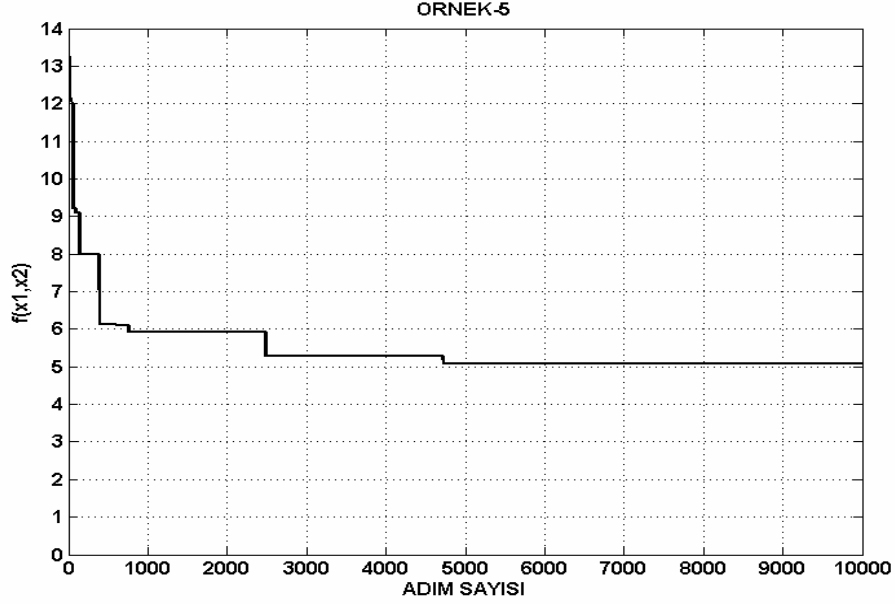


Şekil 2.16. Tek amaçlı GA uygulaması (Örnek-4)

Genetik algoritmalarda elitizm mekanizması çok önemlidir, çünkü her adımda bulunan elit bireyin saklanması ve bir sonraki adıma (topluluğa) aktarılması GA' nın başarımını etkilemektedir. Yukarıdaki örnekte, elitizm yoktur ve en iyi birey saklanmadığı için istenilen sonuca ulaşılsa bile bu sonuç bir diğer adımda kaybedilmektedir.

Örnek-5: GA esasen kılavuzlanmış bir rasgele sayı üretme tekniğidir. GA' nın rasgele sayı üretmeden farkını göstermek için, rasgele sayı üretimi ile ilgili bir program yazılmış ve bu program aşağıdaki parametrelerle çalıştırılmıştır.

Birey Sayısı : 20
 Toplam Bit Sayısı : 50
 Adım Sayısı : 10000



Şekil 2.17. Rasgele sayı üretme (Örnek-5)

Şekilden de görüleceği üzere rasgele sayı üretme algoritması yaklaşık 4800. adımda **5.07709** sonucuna ancak ulaşabilmektedir. Rasgele sayı üretimi ile yapılan optimizasyonda ancak çok ileri adımlarda sonuca ulaşabilmektedir. Bu da GA' nın rasgele sayı üretmeden farkını göstermektedir. GA esasen kılavuzlanmış bir rasgele sayı üretme tekniğidir. GA, sonuca rasgele sayı üretiminden daha kısa sürede ve daha iyi yaklaşmaktadır.

2.3.4. Çok Modluluk ve Çeşitlilik Artırıcı Yöntemler

Birbiriyle aynı olan bireylerin üretilmesi GA' nın sonuca ulaşmasını engeller. Bunun için GA' larda çeşitliliğin fazla olması gerekmektedir. Çeşitlilik fazla olmazsa GA yerel en iyilere takılacak ve küresel ya da genel en iyiyi bulamayacaktır. Çok modlu işlevlerde genel bir en iyinin yanında birden çok yerel en iyi değeri de bulunmaktadır. Çok modluluk, GA' nın çeşitliliğini test etmekte kullanılır. Normalde GA, bu yerel en iyilere takılarak küresel en iyiyi bulamayabilir. Yani sonuçlar yerel en iyiler civarında toplanabilir. Bu durum genetik sürüklenme olarak adlandırılır. Bu yüzden GA' da çeşitliliği artırıcı yöntemlere başvurulması gerekmektedir. Aşağıda çeşitlilik artırıcı bazı yöntemler açıklanmakta ve bunlara ilişkin uygulama örnekleri verilmektedir.

2.3.4.1. Paylaşım (Sharing) Yöntemi

Paylaşım yönteminde, bir bireyin puanı eğer etrafında benzer komşu bireyler varsa, cezalandırılır (yani azaltılır veya artırılır) [Goldberg ve Richardson, 1987]. Bu yüzden, kalabalık bir bölgede yer alan bir birey, en yüksek cezayı alacaktır ve daha az birey oluşturacaktır. Yani, kalabalık yerde olan bireyin puanı (amaç işlevinde aldığı değer) yarıçapa (niche) bağlı olarak azaltılacaktır. Paylaşım yöntemi, topluluğun yoğunluğuna bakarak, arama bölgesini değiştirir. Tipik olarak, bir i bireyinin f_{si} paylaşılmış puanı, onun ilk puanı f_i 'nin yarıçap sayısına bölümüyle bulunur. Bir bireyin yarıçap sayısı, onun topluluktaki diğer bireyler arasındaki (kendisi de dahil) paylaşım işlevi değerlerinin toplamına eşittir.

$$f_{si} = \frac{f_i}{\sum_j sh(d(i, j))} \quad (2.18)$$

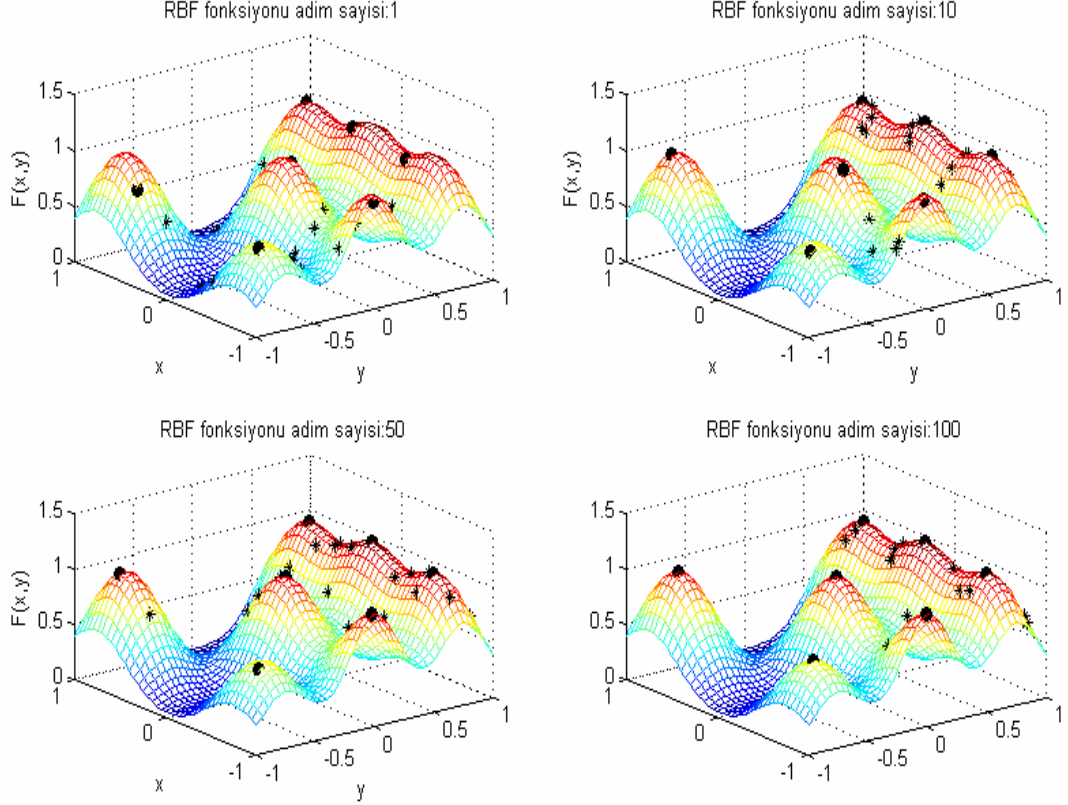
Burada N topluluk boyutunu gösterir. $d(i, j)$ i . ve j . bireyler arasındaki Öklit mesafesidir ve amaç uzayında şöyle hesaplanır (m : amaç sayısıdır):

$$d(i, j) = \sqrt{\sum_{k=1}^m \left(\frac{f_k^{(i)} - f_k^{(j)}}{f_k^{maks} - f_k^{min}} \right)^2} \quad (2.19)$$

f_k^{maks} ve f_k^{min} sırasıyla k . amaç için en büyük ve en küçük işlev değerleridir. Paylaşım işlevi, iki birey arasındaki benzerlik seviyesini ölçer. Elemanlar aynı ise **1** olarak döner, eğer mesafeleri $d(i, j)$ farklılık eşiğinden yüksekse **0** olarak döner, orta değerler ise farklılığın orta seviyelerindedir. Yaygın bir paylaşım işlevi aşağıdaki gibidir:

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_s} \right)^\alpha, & \text{eğer } d < \sigma_s, \\ 0, & \text{diğer durumlarda,} \end{cases} \quad (2.20)$$

Burada farklılığın eşiği σ_s gibi bir sabittir ve α ise paylaşım işlevini biçimini değiştiren bir sabittir (genelde 1 seçilir). Bu hesaplamalara göre, kalabalık bir bölgede yer alan bir bireyin puanı azaltılarak başka bir bölgeye kayması sağlanır. Çok modlu bir işlev ve bu işleve paylaşım yöntemi uygulanmasıyla elde edilen sonuçlar Şekil 2.18’ de gösterilmektedir.



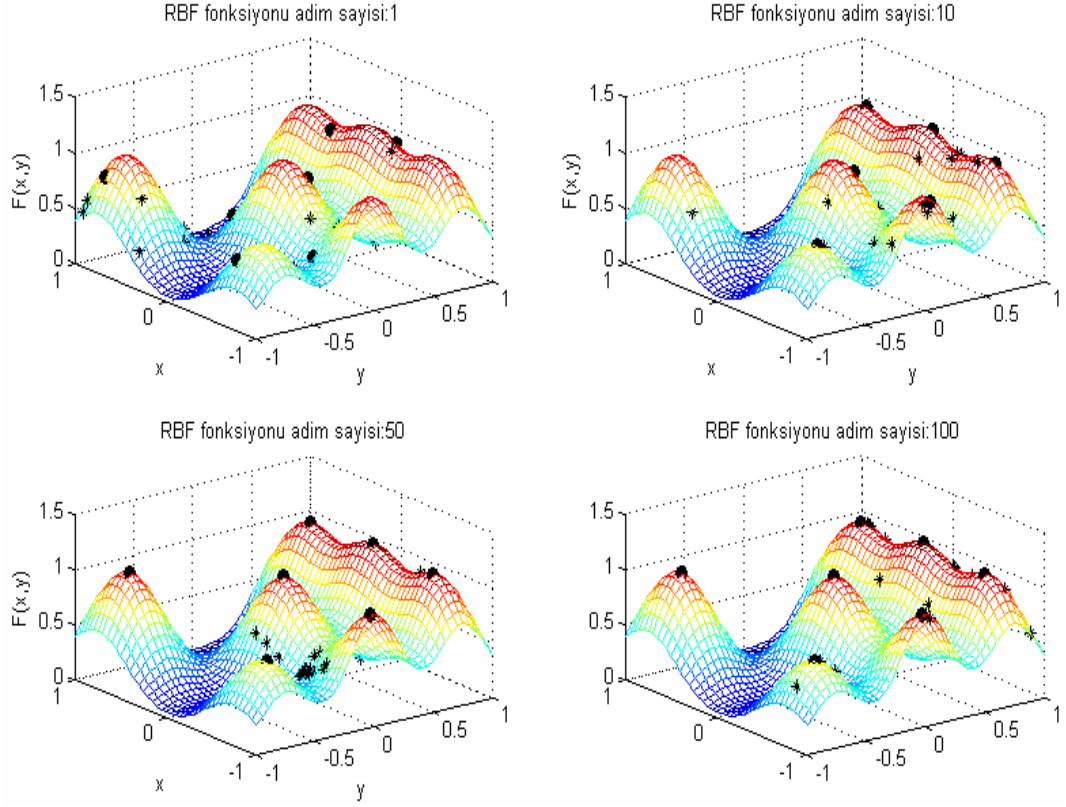
Şekil 2.18. Yedi adet RBF işlevi ile oluşturulan çok modlu bir amaç işlevinin paylaşım yöntemi ile en iyi değerinin bulunması

2.3.4.2. Ayıklama (Clearing) Yöntemi

Ayıklama yöntemi, paylaşım yönteminin doğal bir uzantısıdır (Petrowski, 1996). Cezalandırma yoktur ve bireylerden biri saklanıp, diğerleri yok edilmektedir. Yani, yarıçap içerisindeki bireylerden en iyileri tutulur, diğerlerini ise ortadan kaldırılır. Böylece yarıçap, seçilen bireyler tarafından temsil edilir. Burada, toplam birey sayısı değişmemektedir, bunu sağlamak için çıkan bireylerin yerine yeni bireyler eklenir.

Amaç; bireyleri düzgün olarak dağıtmaktır. Paylaşım yönteminde olduğu gibi, ayıklama yöntemi de bireylerin puanı bulunduktan sonra ve seçme işlemi uygulanmadan önce uygulanır. En basit ayıklama yöntemi, aynı alt topluluktaki diğer tüm bireylerin puanı sıfırlanırken, en iyi bireyin(lerin) puanını saklamaktır. Puan paylaşımında olduğu gibi, tek bir alt topluluktaki tüm bireylere kaynakları paylaşdırmak yerine, ayıklama yöntemi alt topluluktaki yalnızca en iyi bireylere dağıtım yapar. Pratikte, bir yarıçapın k kapasitesi, yarıçapın kabul edebileceği en yüksek eleman sayısını belirler. Ayıklama yöntemi, yarıçapın ve yarıçap kapasitesinin en doğru hesaplanmasını sağlayan en iyi yöntemdir. Bununla birlikte, ayıklama ve paylaşım yöntemlerinin her ikisi de çok iyi tahmin edilmesi gereken bir yarıçap bilgisi kullanırlar. Her alt topluluk baskın bir bireye sahiptir ve yarıçap, en yakın iki baskın birey arasındaki mesafenin yarısından daha küçük olmalıdır. Aksi takdirde, bir tepe kuralı için ($k=1$ için) bir birey atılır ve baskın bir birey için birkaç tepeli gösterim yetersiz olacaktır. 2-boyutlu bir problemde yarıçapın belirlenmesi nispeten kolaydır, ancak boyutlar arttığında bu yarıçapın belirlenmesi bir sorun teşkil edecektir. Bu parametre dışarıdan girilmesi gereken bir parametredir.

Şunu belirtmek gerekir ki, her birey için ayrı ayrı yarıçap belirlenir ve her bireyin merkezde olduğu bir çember vardır ve bu işlem her bir birey için yalnızca bir kez yapılmaktadır. Bu işlemler birinci puanlamadan sonra (amaç işlevine yerleştirme işleminden sonra) yapılır. Daha sonra bu yöntemlere göre ikinci bir puanlama yapılır. Yarıçapın belirlenmesinde alan daraltılarak şüphelerin ortadan kaldırılması gerekir. Tabi ki bunun için öncelikle ilk yarıçap tahmininin yapılması gerekir. Bunun GA' ya getirdiği dezavantaj ise ikincil puanlamadan dolayı bir işlem yüküdür. Burada göz önünde bulundurulması gereken diğer bir husus ta, elitizm mekanizmasının değiştirilmesi gerekliliğidir. Çünkü yarıçap küçüldükçe elit birey sayısı da artacaktır. O yüzden yarıçapın çok iyi belirlenmesi gerekir. Bu arada, GA' nın da yerel en iyilere takılmaması gerekir. Çok modlu bir işlev ve bu işleve ayıklama yöntemi uygulanmasıyla elde edilen sonuçlar Şekil 2.19' da gösterilmektedir.



Şekil 2.19. Yedi adet RBF işlevi ile oluşturulan çok modlu bir amaç işlevinin ayıklama yöntemi ile en iyi değerinin bulunması

2.3.4.3. Eşleşme Sınırlaması (Mating Restriction)

İlk olarak Goldberg tarafından tek amaçlı optimizasyon problemlerinde kötü bireylerin azaltılması veya önlenmesi amacıyla kullanılmıştır [Goldberg, 1989]. Eşleşme kısıtlaması, GA' nın başarımının artırılması amacıyla birey çiftlerinin çaprazlama işlemi için nasıl seçileceğini belirler. Goldberg eşleşme kriteri olarak, yapısal (genotip) benzerlik kullanan bir örnek vermiştir. Deb ve Goldberg ise fenotip mesafesini kullanan bir eşleşme kısıtlaması önermişlerdir [Deb ve Goldberg, 1989b]. Burada, iki birey ancak birbirlerine çok benziyorlarsa çaprazlamaya gönderilmektedir (Hesaplanan bir $\sigma_{\text{eşleşme}}$ ölçütünden daha az mesafeye sahip bireyler birbirine çok benzer kabul edilmiştir). Böylece birbirinden farklı bireylerin üretilmesi amaçlanmaktadır. Birçok ÇAGA yönteminde eşleşme kısıtlaması daha az istenilen bireylerin (örneğin Pareto olmayan bireylerin) azaltılması için kullanılmaktadır. Eşleşme kısıtlaması, literatürde farklı biçimlerde uygulanmıştır. Örneğin, Lis ve Eiben yalnızca birbirinden

farklı kodlara sahip bireylerin eşleşmesine izin vermişlerdir [Lis ve Eiben, 1996]. Bazı araştırmacılar $\sigma_{\text{eşleşme}} = \sigma_s$ olarak kullanmışlardır, ancak bunun neden böyle seçildiğine dair bir bilgi literatürde bulunmamaktadır [Coello Coello ve ark., 2007]. Buradaki önemli bir durum da, $\sigma_{\text{eşleşme}}$ katsayısının tahmin edilmesi gerektiğidir. Zitzler ve Thiele ise farklı $\sigma_{\text{eşleşme}}$ değerlerini kendi oluşturdukları test işlevlerinde kullanmışlar ancak, eşleşme kısıtlaması olmayan duruma göre herhangi bir iyileşme görmemişlerdir [Zitzler ve Thiele, 1998].

2.3.5. Çok Amaçlı Genetik Algoritmalar (ÇAGA)

Mühendislik problemleri optimizasyona gereksinim duyar ve genelde çok amaçlıdır. Bu çok amaçlı problemlerin çözülmesi tek amaçlı duruma göre zordur. Bu nedenle ÇAGA' nın birçok bileşeni vardır. Aşağıda bu bileşenler genel olarak açıklanmaktadır. Ayrıca ÇAGA' ların uygulama alanlarına ilişkin bazı örnekler verilmektedir.

2.3.5.1. Optimizasyon Parametreleri veya Karar Değişkenleri

Parametre veya karar vektörü x aşağıdaki gibi verilir:

$$x = [x_1, x_2, \dots, x_N]^T \quad (2.21)$$

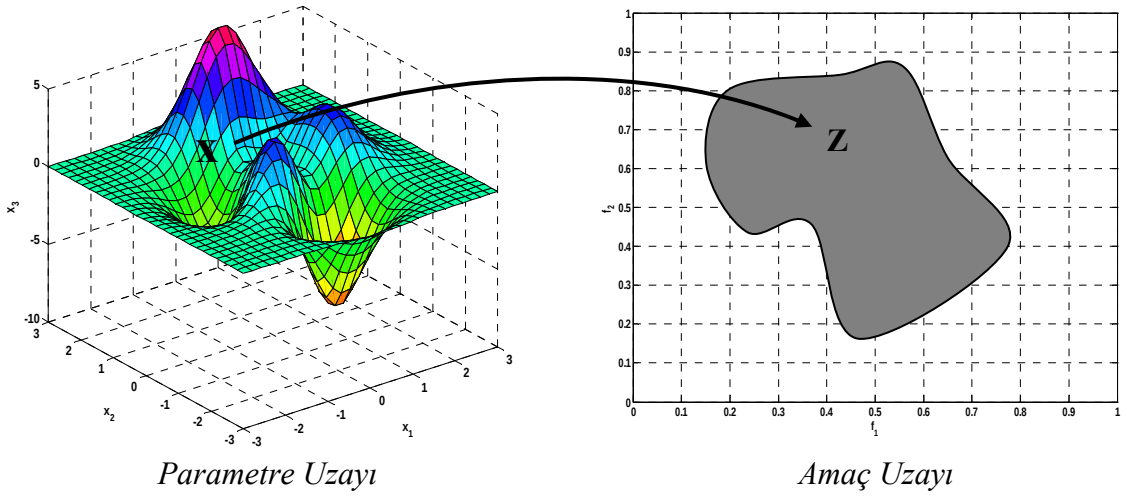
Bu T tranpozu ifade etmekte, N ise parametre sayısını göstermektedir. Her birey bu parametrelerin birleşiminden oluşmaktadır. Parametrelerin alt ve üst sınırları kullanıcı tarafından çok iyi belirlenmelidir. Eğer parametre sınırları yanlış seçilirse arama uzayı genişleyebilir ve arama işlemi uzun sürebilir, ya da istenilen en iyi çözüm/çözümlere ulaşamayabilir.

2.3.5.2. Amaç İşlevi

Hem doğrusal hem de doğrusal olmayan çok amaçlı optimizasyon problemleri için m amaçlı bir amaç işlevi şöyle verilir:

$$F(x) = [f_1(x), f_2(x), \dots, f_m(x)]^T \quad (2.22)$$

Her amaç işlevi ya en küçüklenecek ya da en büyüklenecektir. Karar ya da parametre uzayına ek olarak, çok amaçlı uzayda bir de amaç uzayı söz konusudur. Parametre uzayındaki her x değerine karşılık, amaç uzayında bir nokta bulunur ve bu nokta $f(x)=z$ olarak gösterilir. Şekil 2.20, her iki uzayı ve bunlar arasındaki haritalamayı göstermektedir.



Şekil 2.20. Parametre uzayı ve amaç uzayının kavramsal gösterilimi

Bu tezde kısıtlamasız, iki amaçlı işlevler kullanılmıştır. Optimizasyon problemleri ise en küçükleme (minimizasyon) problemleridir. Parametre sayıları kullanılan test işlevlerine göre değişmektedir.

2.3.5.3. Baskınlık Kavramı ve Pareto-Optimalliği

Bu bölümde çok amaçlı optimizasyonda gerekli olan bazı tanımlar (baskınlık, Pareto-optimal, Pareto-optimal cephe) bir en küçükleme problemi için verilmektedir.

Tanım 1 (Baskınlık): Bir karar vektörü (x_1), bir başka karar vektörüne (x_2) yalnız ve yalnız aşağıda verilen durumlar sağlandığında baskın olur ($x_1 \prec x_2$ olarak gösterilir):

- x_1 tüm amaç işlevlerinde x_2 ' den iyi ise (örneğin $f_k(x_1) \leq f_k(x_2)$, $\forall k = 1, \dots, m$ ve
- x_1 , x_2 ' den en azından bir amaç işlevinde çok iyi ise (örneğin $\exists k = 1, \dots, m : f_k(x_1) < f_k(x_2)$)

Benzer olarak, bir amaç vektörü (f_1) bir diğer amaç vektörüne (f_2), tüm amaçlarda ondan daha iyi ise veya en azından bir amaç işlevinde ondan çok iyi ise baskın olur ($f_1 \prec f_2$). Şekil 2.21' de, bireyler arasındaki baskınlık ilişkisi gösterilmektedir. Burada D, E ve F bireylerinin hepsi G bireyinden baskındır. Çünkü her iki amaçta da ondan iyidirler.

Tanım 2 (Pareto-optimal): Bir karar vektöründe (x^*) baskın olan başka bir karar vektörü yoksa Pareto-optimal olarak adlandırılır. Amaç vektörü ($f^*(x)$) ise, eğer x Pareto-optimal ise Pareto-optimaldir.

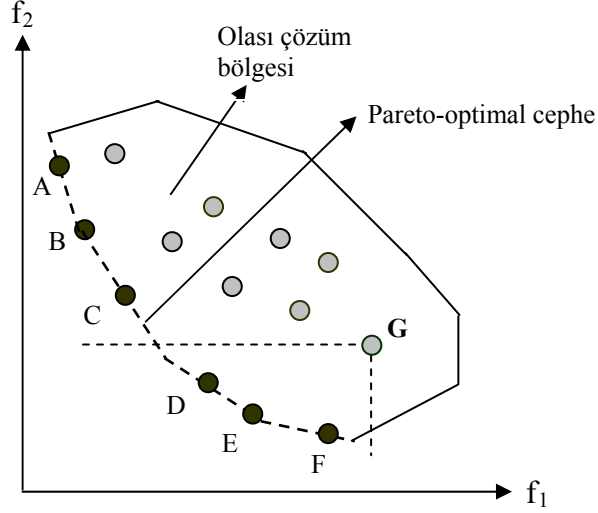
Pareto-optimalliği kavramı, ilk olarak Francis Ysidro Edgeworth tarafınan öne sürülen bu kavramı gerçekleştiren ve genelleyen İtalyan matematikçi ve ekonomist Vilfredo Pareto' dan gelmektedir.

Tanım 3 (Pareto-optimal küme): Pareto-optimal karar vektörleri veya Pareto-optimal bireylerden oluşan kümedir. P^* ile gösterilir.

Tanım 4 (Pareto-optimal cephe): Tüm amaç işlevleri dikkate alındığında, baskın olan tüm bireylerden / karar vektörlerinden oluşan cephedir. PF^* ile gösterilir.

Tanım 5 (Tercih Edilen Çözüm): Çözüm kümesi içerisinde, karar verici (decision maker) tarafından ek kriterler kullanılarak seçilen baskın (Pareto) bir çözümdür.

Şekil 2.21’ de iki amaçlı bir en küçüklenme problemi üzerinde, Pareto-optimal (veya kısaca Pareto) bireyler, bu bireylerden oluşan $P^*=\{A,B,C,D,E,F\}$ Pareto-optimal küme ve Pareto-optimal cephe gösterilmektedir.

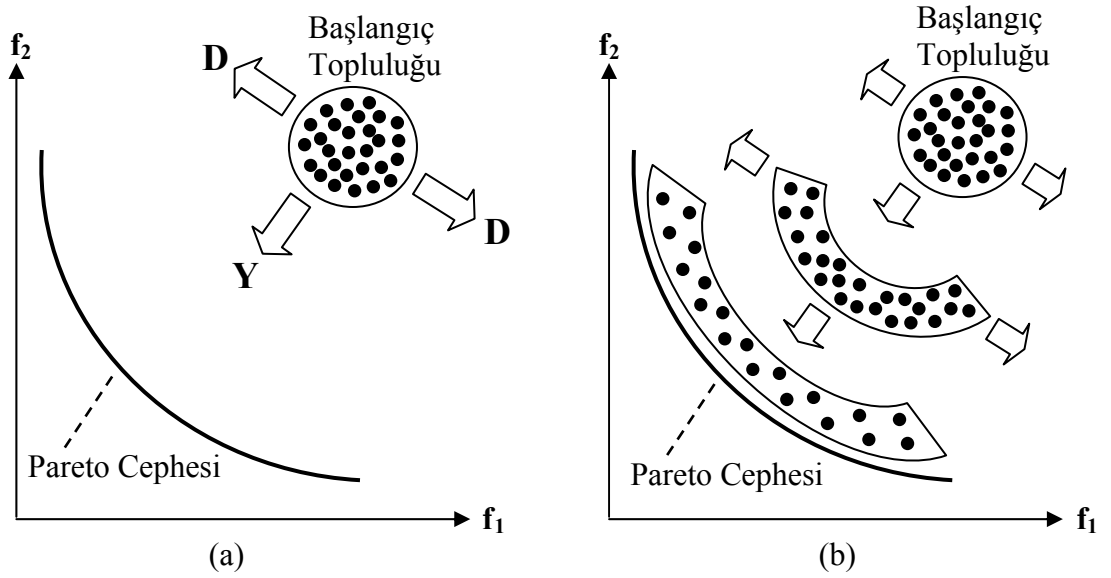


Şekil 2.21. Çizgili bölge Pareto-optimal cepheyi göstermektedir (bu bir en küçüklenme problemidir). Siyah noktalar Pareto bireyleri, gri noktalar Pareto olmayan bireyleri göstermektedir. Örneğin, “G” bireyi, “D, E” ve “F” Pareto bireylerinin etki alanına girmektedir

Tek amaçlı GA’larda tek bir en iyi sonuç vardır ve bireyler (çözüm önerileri) en iyiden en kötüye doğru tekil bir biçimde sıralanabilir. Ancak ÇAGA’larda tek bir en iyi çözüm yerine bir grup en iyi çözüm (*Pareto-optimal çözümler* olarak adlandırılırlar) vardır ve dolayısıyla en iyiden en kötüye doğru bireylerin sıralanması tekil değildir. Bu yüzden, birçok ÇAGA yöntemi geliştirilmiş ve geliştirilmeye devam edilmektedir. Sıralama yapmanın amacı, bir sonraki nesildeki bireylerin (çözüm önerilerinin) nasıl elde edileceğini belirleyen seçme mekanizmasına bilgi üretmektir. Yani seçme mekanizmasından önce bir sıralama yöntemine ilişkin hesaplamalar yer almaktadır.

Amaçlar tek tek hesaba katıldığında bireylerin sıralaması farklı olacaktır, bir amaca göre en iyi olan çözüm diğerlerine göre en kötü olabilir. Ancak tüm amaçlar hesaba katıldığında Pareto-optimal cephedeki bireylerin birbirlerine göre bir üstünlüğü olmayacaktır. **Çok amaçlı optimizasyonun asıl hedefi, Pareto-cephesini bulmak veya ona yaklaşmak ve bu cephe üzerinde düzgün bir dağılım sağlamaktır.** Var olan tüm yöntemler bunu sağlamaya çalışmaktadırlar. Bu hedefler, Şekil 2.22’de bir en

küçükleme problemi üzerinde görsel olarak gösterilmektedir. Bu şekilde **D**-düzgün dağılım ya da çeşitliliği, **Y**-yakınsamayı temsil etmektedir. Düzgün dağılımın artırılması için, başlangıç topluluğu **D** ile gösterilen oklar yönünde genişletilmelidir. Pareto cephesine yakınsama için ise başlangıç topluluğu **Y** ile gösterilen ok yönünde ilerlemelidir. Düzgün dağılım – yakınsama dengesi doğru olarak ayarlandığı zaman, Şekil 2.22b’ de gösterildiği gibi iyi bir çözüm topluluğu bulunabilir. Bununla birlikte, her problem ve her ÇAGA yöntemine göre böyle bir çözümü bulmak kolay değildir [Ishibuchi ve Shibata, 2004].

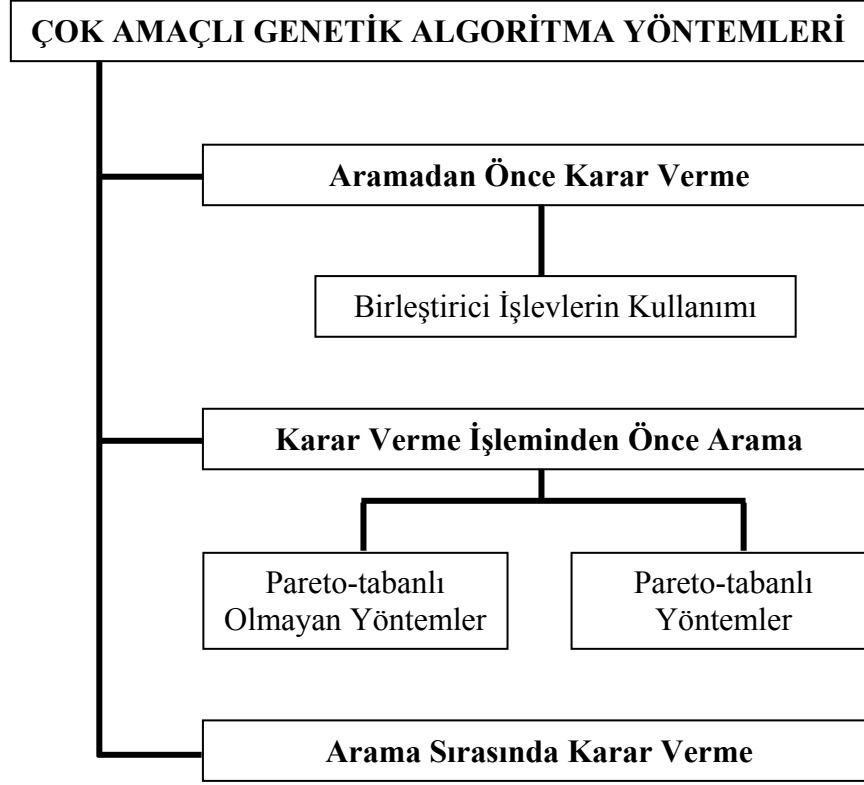


Şekil 2.22. (a) Başlangıç topluluğu ve Pareto cephesi, (b) İstenilen GA durumu

2.3.5.4. Arama ve Karar Verme

Çok amaçlı optimizasyonda, Pareto tanımından dolayı birden fazla en iyi çözüm bulunmaktadır. Bu yüzden, çok amaçlı optimizasyon problemleri iki farklı durumdan oluşur: Pareto-optimal birey / çözüm kümesinin aranması ve çok amaçlı karar verme (Horn, 1997). Genellikle, bu iki durum ayrı ayrı değerlendirilir. Tek amaçlı optimizasyon ile geniş ve karmaşık arama uzaylarının aranması zordur. Bu nedenle, çok amaçlı optimizasyon yöntemleri tanımlanmıştır. Bu yöntemler arasındaki temel farklar, amaç fonksiyonları ve puan atama mekanizmalarıdır. Bir topluluktaki (çözüm

kümesindeki) her bir bireye ilişkin puanın nasıl hesaplandığı ve atandığına bağlı olarak, çok amaçlı genetik algoritmalar Şekil 2.23’ te gösterildiği gibi üç kategoride sınıflandırılabilir.



Şekil 2.23. Çok amaçlı genetik algoritmaların sınıflandırılması [Vazquez, 1999]

Aramadan Önce Karar Verme: Çok amaçlı optimizasyon probleminin amaçları, karar vericiden gelen tercih bilgisine (preference information) göre tek bir amaçta birleştirilir. Yani çok amaçlı optimizasyon problemi, bu tür genetik algoritmalarda bir takım katsayılarla çarpılıp toplanarak tek bir amaçta birleştirilir. Bu tür genetik algoritmalara örnek olarak CWO (sabit ağırlıklı amaç işlevi) ve VWO (değişken ağırlıklı amaç işlevi) yöntemleri verilebilir. Bu tür yöntemler kolay uygulanabilir olmalarına rağmen çoğu optimizasyon probleminde istenilen sonuçların elde edilmesini sağlayamazlar.

Karar Verme İşleminin Önce Arama: Optimizasyon, herhangi bir tercih bilgisi verilmeden gerçekleştirilir. Arama işlemi sonucunda bir grup aday çözüm (ideal olarak Pareto-optimal çözümler) bulunur. Karar verici (KV) son seçimini bu çözümler

arasından yapar. Pareto-tabanlı olmayan yöntemler ve Pareto-tabanlı yöntemler bu tür çok amaçlı genetik algoritma yöntemlerindendir. Pareto-tabanlı olmayan yöntemlere örnek olarak Schaffer' in Vektör Hesaplamalı Genetik Algoritması (VEGA) örnek olarak verilebilir. Pareto-tabanlı yöntemler ise Pareto tanımını kullanarak bireylere puan atamaktadırlar. Bu yöntemlere örnek olarak MOGA, NSGA, SPEA vb. verilebilir.

Arama Strasında Karar Verme: Her bir optimizasyon adımından sonra, aramaya kılavuzluk edecek şekilde karar vericiden (KV) sağlanan ayrıntılı tercih bilgisine göre alternatif çözümler sunulur. Böylece arama uzayı küçültülebilir. Bu tür genetik algoritmalarla örnek olarak Fonseca ve Fleming' in PAGA (Preference Articulation Genetic Algorithm) yöntemi örnek olarak verilebilir [Fonseca ve ark., 1998]. Bu yöntemde, Pareto derecelendirmesi ile tercih belirleme yöntemi birleştirilmektedir.

Şu ana kadar, ÇAGA algoritmalarında KV' nin tercihlerinden yararlanılması üzerinde çok az çalışma yapılmıştır. Bu tür çalışmalar önemli yararlar sağlayabilir. Çok boyutlu problemlerin çözümünde hesapsal bir azaltım sağlanabilir, ayrıca KV' ye daha anlamlı bilgi gönderilebilir. Tercih bilgisine göre arama uzayı daraltılarak daha iyi sonuçların bulunması sağlanabilir. Tüm bu sorunlara çözüm sağlamak amacıyla IEEE Evolutionary Computation dergisi tarafından, 2010 yılında tercih belirleme tabanlı ÇAGA' lar hakkında özel bir sayı çıkarılacaktır (Editörler: Prof. Kalyanmoy Deb ve Prof. Murat Köksalan).

2.3.5.5. Çok Amaçlı Genetik Algoritmaların Temel Bileşenleri

ÇAGA' ların temel bileşenleri şunlardır:

- Sıralama / puan atama yöntemleri (NSGA, MOGA, SPEA, PAES vb.)
- İkincil topluluk (arşiv) ve yönetimi veya elitizm (ikincil topluluğun boyutu, ikincil topluluğu güncelleme, birincil topluluğu güncelleme).
- Çeşitliliği sürdürmek veya yarıçap belirleme (niching) yöntemleri (paylaşım, ayıklama, eşleşme kısıtlaması vb.).
- Seçme mekanizmaları (rulet çarkı seçimi (RWS), stokastik evrensel örnekleme (SUS), turnuva seçimi (TS) vb.).

- Parametrelerin kodlama teknikleri (ikili kodlama, gray kodlama, gerçel sayı kodlama vb.).
- Genetik işlevlerin kullanımı ve biçimleri (çaprazlama ve mutasyon).
- Topluluk boyutu (kullanıcıya bağlıdır).
- Optimize edilecek parametrelerin alt ve üst sınırları (problem hakkındaki öngörüye bağlıdır).

2.3.6. Uygulama Alanları

Klasik optimizasyon yöntemlerinin farklı alanlardaki problemlerin çözümü için kullanılması 1950' li yıllara dayanmasına rağmen, GA' ların endüstriyel problemlere uygulanması 80' li yılların ortasını bulmaktadır. Bununla birlikte 1990' dan sonra, GA' ların gerçek dünya problemlerine uygulanmasında önemli bir artış olmuştur. Bunun nedenlerinden biri GA' ların çok amaçlı optimizasyon problemlerini çözme yeteneğidir.

Çok amaçlı bir genetik algoritma yöntemi önerilirken, bu yöntemin iyi ve kullanışlı olup olmadığının belirlenmesi için algoritma test işlevleri üzerinde denenmektedir ve bazı çıkarımları buna göre yapılmaktadır. Diğer taraftan, algoritmanın pratikte de kullanılabilir olduğunun belirlenmesi için, algoritmanın başarımının gerçek problemler üzerinde de test edilmesi gerekir. Çizelge 2.3' te, çok amaçlı genetik algoritmaların uygulandığı bazı gerçek dünya problemleri ve bu uygulamayı yapanların bir listesi yer almaktadır. Bu uygulamalar üç temel bölümde sınıflandırılabilir:

- I. *Deneyisel Uygulamalar:*** Varolan diğer optimizasyon algoritmalarına karşı GA' nın üstünlüğünü test etmek için yapılan uygulamalardır. Gezgin Satıcı Problemi, Knapsack Problemi, Grafik Bölme Problemi vb.
- II. *Pratik Uygulamalar:*** Endüstride GA kullanılan uygulamalardır. Sayısal Optimizasyon Problemleri, Çizelge/Planlama Problemleri, Yerleşim Problemleri, Görüntü/Ses İşleme, Mikrodalga Eleman Tasarımı, Mekanik Bileşen Tasarımı, İşaret İşleme, Süzgeç Tasarımı, Kontrol Sistemi Tasarımı, Parametre ve Sistem Tanımlama, Robotik, Mühendislik Tasarımı (VLSI vb.), Nonlinear Sistem Tasarımı, Yapay Sinir Ağları ve Bulanık Mantığa Uyarlanma, Şekil (Parça) Tasarımı vb.

III.Sınıflandırıcı Sistemler: Bilgi elde edilmesi için GA' nın kullanıldığı uygulamalardır. Uzman sistemin bilgi tabanının oluşturulması için GA kullanımı vb.

Çizelge 2.3. Bazı ÇAGA uygulamaları

Araştırmacı(lar) (Yıl)	Uygulama Alanı
A.D. Belegundu ve ark. (1994)	Levhalı Seramik Alaşımlar
T.J. Stanley ve T. Mudge (1995)	Mikroişlemci Çip Tasarımı
C.S. Chang ve ark. (1995)	DC Demiryolu Sistemi
F. Jimenez ve J.L. Verdegay (1995)	Taşımacılık/Ulaşım
C.A. Coello ve ark. (1995)	Bir Robot Kolun Karşı Ağırlık Dengelemesi
A.J. Chipperfield ve P.J. Fleming (1996)	Gaz Türbin Motoru Denetleyici Tasarımı
T. Arslan ve ark. (1996)	VLSI Devre Tasarımı
S.Y. Hahn (1996)	Sürekli Mıknatıslı Motor Tasarımı
D.S. Weile ve ark. (1996)	Geniş Band Mikrodalga Soğurucu Tasarımı
D.S. Todd ve P. Sen (1997)	Yük Gemileri İçin Yükleyici Tasarımı
D. Lee (1997)	Deniz Aracı Tasarımı
W. A. Crossley (1997)	Helikopter Tasarımı
E. Zitzler ve L. Thiele (1998)	Sayısal Donanım-Yazılım Çoklu İşlemci Sistem Sentezi
K. Fujita ve ark. (1998)	Otomobil Motor Tasarımı
S. Obayashi ve ark. (1998)	Uçak Kanat Tasarımı
H.A. Güvenir ve E. Erel (1998)	Stok Sınıflandırılması
D. Cvetkovic ve I. Parmee (1998)	Uçak Dış Yüzey Tasarımı
C.M. Fonseca ve P.J. Fleming (1998)	Gaz Türbin Motoru Tasarımı
B. Paechter ve ark. (1998)	Bir Üniversitenin Ders Programlarının Hazırlanması
M. S. Bright (1998)	DSP Sistem Tasarımı
E. Zitzler ve L. Thiele (1998)	Sayısal Donanım-Yazılım Çoklu İşlemci Sistem Sentezi
K. Fujita ve ark. (1998)	Otomobil Motor Tasarımı

Çizelge 2.3. (devamı) Bazı ÇAGA uygulamaları

T. Bagchi (1999)	Çok Ölçütlü Gezgin Satıcı Çizelgelemesi
C. Poloni ve ark. (2000)	Aerodinamik Şekil Tasarımı
E. Schlemmer ve ark. (2000)	Hidroelektrik Generatör Tasarımı
P. di Barba ve ark. (2000)	Elektrostatik Mikromotor Tasarımı
L. Costa ve P. Oliveria (2000)	Levhalı Alaşım Sac Tasarımı
A.J. Blumel ve ark. (2000)	Otomatik Pilot Denetleyici Tasarımı
H. Meunier ve ark. (2000)	Radio Ağı Optimizasyonu
X. Li ve ark. (2000)	Tıbbi Görüntü İşleme
M. Lahanas ve ark. (2001)	Doz Optimizasyonu
N. Laumanns ve ark. (2001)	Yolcu Tren Tasarımı
D. Sasaki ve ark. (2001)	Süpersonik Kanat Tasarımı
Ishibuchi ve ark. (2001)	Dilsel Kural Çıkarımı
S.-Y. Ho ve H.-L. Huang (2001)	Yüz Modelleme ve Tanıma
J. Wright ve ark. (2002)	Bir Binanın Isı Sisteminin Tasarımı
H. A. Abbass (2002)	Göğüs Kanseri Tanısı
R. Balling ve ark. (2003)	Şehir Planlama
X. Hu ve ark. (2004)	Otomobil Parça Tasarımı
J. Regnier ve ark. (2005)	Elektromekanik Sistem Tasarımı
M. Trefzer ve ark. (2005)	CMOS İşlemsel Yükselteç Sentezi
N. Nariman-Zadeh ve ark. (2005)	Aerodinamik Optimizasyon
J. Regnier ve ark. (2005)	Elektromekanik Sistem Tasarımı
R. Verschae ve ark. (2005)	Yüz Modelleme ve Tanıma
L. Araujo (2006)	Doğal Dil İşleme
M. Calonder ve ark (2006)	Biyoinformatik
M. P. Kleeman ve ark. (2007)	Kuantum Kaskat Laser Tasarımı
C. Perales-Gravan ve ark. (2008)	AM Radio Alıcı Tasarımı
E. Mininno ve ark. (2008)	Gömülü Mikrodenetleyici Optimizasyonu
F. C. J. Allaire ve ark. (2008)	Gerçek Zamanlı Uçuş Planı Uygulaması
T. Siegfried ve ark., (2009)	Yer altı Suyu Yönetimi
J.G. Herrero ve ark. (2009)	Hava Trafik Kontrolü

3. MATERYAL VE YÖNTEM

Bu bölümde Pareto tabanlı olmayan ve Pareto tabanlı ÇAGA yöntemleri açıklanmakta, algoritmaları ile yöntemlere ilişkin örnekler verilmektedir. Tezde önerilen ÇAGA yöntemleri, bu bölümde verilen bazı iyi bilinen (state-of-art methods) ÇAGA yöntemleri ile sık kullanılan test işlevleri kullanılarak karşılaştırılmışlardır.

3.1. Pareto Tabanlı Olmayan ÇAGA Yöntemleri

Bu bölümde, Pareto tanımını kullanmayan bazı çok amaçlı genetik algoritma yöntemleri incelenmektedir. Tek amaçlı optimizasyondan farklı olarak, çok amaçlı optimizasyonda bir topluluğun genetik algoritma tarafından puanlanmasının tek bir yolu yoktur. Tek amaçlı genetik algoritmada, puanlama, probleme göre amaç işlevinde alınan değerlerin büyükten küçüğe veya küçükten büyüğe doğru sıralanmasıyla yapılabilir. Ağırlıklı toplam yaklaşımlarından Pareto-tabanlı yaklaşımlara kadar var olan puanlama yöntemleri, istenilen bireyler hakkında ne kadar bilgi verdiklerine göre sorgulanmaktadır. Bu yöntemler arasındaki temel fark, onların bireylerin puanlarını belirlerken kullandıkları yoldur. Diğer bir deyişle, her yöntem kendi mekanizmasına göre bireyleri en iyiden en kötüye doğru sıralar. Bireyleri sıralamanın veya puanlamanın en iyi yolu hala ucu açık bir sorudur ve daha çok araştırma yapılması gereklidir.

Çoğu gerçek dünya problemi için, çoklu amaçların anlık optimizasyonuna gereksinim vardır. Tek amaçlı durumda, amacın ya en büyük ya da en küçük değerleri aranmaktadır. Çok amaçlı durumda ise, topluluktaki bireyler (ya da çözüm önerileri) çok amaçlı bir genetik algoritma yöntemi (ÇAGA) tarafından puanlanır. Temel olarak bir ÇAGA, bir topluluktaki bireyleri puanlama işlemine veya tanımına bağlı olarak sıralamaktır. Yani ÇAGA ile çoklu amaç uzayı, tek bir puan değerine dönüştürülmektedir. Aşağıda bu yöntemlerden Pareto tanımını kullanmayan üç tanesi ayrıntılı olarak incelenmektedir.

Bu yöntemlerden ikisi (CWOFF ve VWOJ) ağırlıklı toplam yaklaşımları olarak adlandırılır, çünkü bu yöntemler bireyleri puanlarken Pareto tanımını hesaba katmamaktadırlar. CWOFF çok bilinen bir yaklaşımdır ve amaçları birleştirmek için bazı ön bilgilere gereksinim duyar (Michalewicz, 1994; Murata ve Ishibuchi, 1995).

CWOF' tan farklı olarak, VWOJ sabit ağırlıkları değil rasgele üretilmiş ağırlıkları kullanır (Murata ve Ishibuchi, 1995; Ishibuchi ve Murata, 1998). VEGA, çok amaçlı genetik algoritmalar için atılan ilk adım olarak bilinir ve karşılaştırma yapmak için standart bir araç olarak kullanılır (Schaffer, 1985).

Bu yöntemlerin incelenmesinde Schaffer' in SCH1 test işlevi kullanılmıştır:

$$f(x) = x^2 \quad \text{ve} \quad g(x) = (x - 2)^2 \quad (3.1)$$

Bu yöntemlerin bu test işlevi ile çalıştırılmasıyla elde edilen sonuçlar Şekil 3.3, Şekil 3.4, Şekil 3.5, Şekil 3.8 ve Şekil 3.11' de verilmektedir. Tüm uygulamalar için birey sayısı 20, bit sayısı 20, adım sayısı 500, $p_m = 0.1$, $p_c = 1$, parametrenin alt sınırı 0, üst sınırı ise 2 seçilmiştir.

3.1.1. Sabit Ağırlıklı Amaç İşlevi (Constant Weight Objective Function, CWOF)

CWOF, temel olarak, birkaç amaç işlevini ağırlıklandırarak tek bir amaç işlevinde birleştirir ve çok amaçlı problemi tek amaçlı probleme dönüştürür (Michalewicz, 1994; Murata ve Ishibuchi, 1995). Farklı ağırlıklar, farklı Pareto çözümlerinin elde edilmesini sağlar. Birkaç Pareto-optimal çözümün bulunması için, programın farklı ağırlıklarla birkaç kez çalıştırılması gerekir. CWOF klasik bir yaklaşımdır, burada çok amaçlı işlev f_i tek bir amaçta F birleştirilir :

$$F(x) = \sum_{i=1}^n w_i f_i(x) \quad \text{veya} \quad F(x) = w_1 f_1(x) + \dots + w_n f_n(x) \quad (3.2)$$

Burada $w_1, \dots, w_n$, n adet amaç için negatif olmayan ağırlıklardır ve bunların arasındaki ilişki aşağıdaki gibidir:

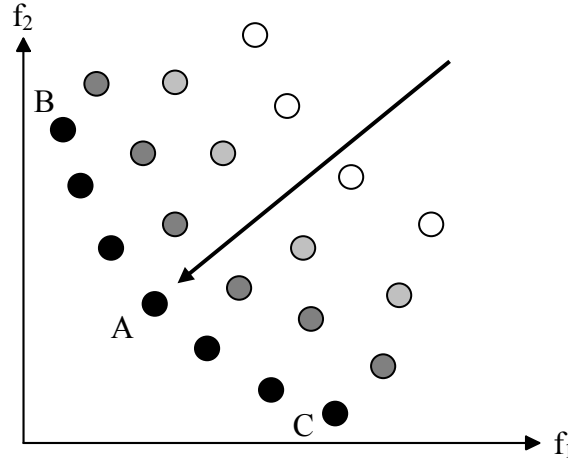
$$\sum_{i=1}^n w_i = 1 \quad (3.3)$$

Sabit ağırlıklar (w_i) ayarlanarak, genetik algoritmanın arama yönü ayarlanabilir. Bunun anlamı; CWOFF Pareto cephesindeki yalnızca bir veya birkaç bireye yüksek puan vermekte (seçime bağlı olarak düşük puan) geri kalan bireyler ise düşük puan almaktadır, çünkü CWOFF arama uzayında belirli bir yönü tercih etmektedir. Şekil 3.1, bu durumu göstermektedir. Eğer iki amaç tek bir amaçta birleştirilecekse ve ilk amaç diğerine göre daha fazla önemli ise, o zaman $F(x)$ (3.4) eşitliğindeki gibi olur. Eğer ikinci amaç daha önemliyse, $F(x)$ (3.5) eşitliğindeki gibi olur (en küçükleme problemi söz konusu ise). Eğer her iki amaçta aynı öneme sahipse, $F(x)$ (3.6) eşitliğindeki gibi olur.

$$F(x) = 0.75 f_1(x) + 0.25 f_2(x) \quad (3.4)$$

$$F(x) = 0.25 f_1(x) + 0.75 f_2(x) \quad (3.5)$$

$$F(x) = 0.5 f_1(x) + 0.5 f_2(x) \quad (3.6)$$



Şekil 3.1. CWOFF, arama uzayında bir arama yönüne sahiptir ve yalnızca A gibi bir birey bulabilir (veya birkaç Pareto çözümü)

CWOFF yalnızca birkaç Pareto çözümünü bulur, Pareto çözümlerinin tam bir kümesini bulamaz. Karmaşık bir sistem tasarım problemi için tipik bir CWOFF örneği (Linkens ve Nyongesa, 1995)' de verilmektedir. CWOFF yönteminin algoritması Şekil 3.2' de verilmektedir.

CWOF Algoritması

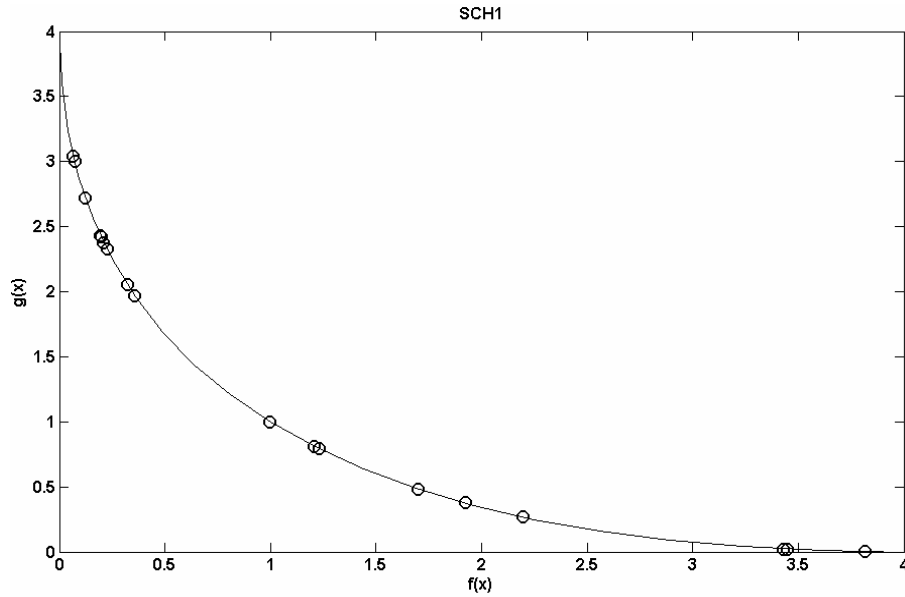
- 1) Başlangıç topluluğu oluşturulur (Rasgele veya dışarıdan girilerek)
- 2) Amaçların önemine göre w_i ağırlıkları belirlenir (w_i ' lerin sayısı amaçlar kadardır)
- 3) $F(x) = w_1 f_1(x) + w_2 f_2(x) + w_3 f_3(x) + \dots$

Amaçlar sabit ağırlıklarla çarpılır ve sonuçta tek bir amaç (diğer amaçların ağırlıklı toplamı) olarak düşünülür. Burada ağırlıklar sabittir ve ağırlıkların toplamı 1' dir. Ağırlığı yüksek olan amaç daha önemli olacaktır.

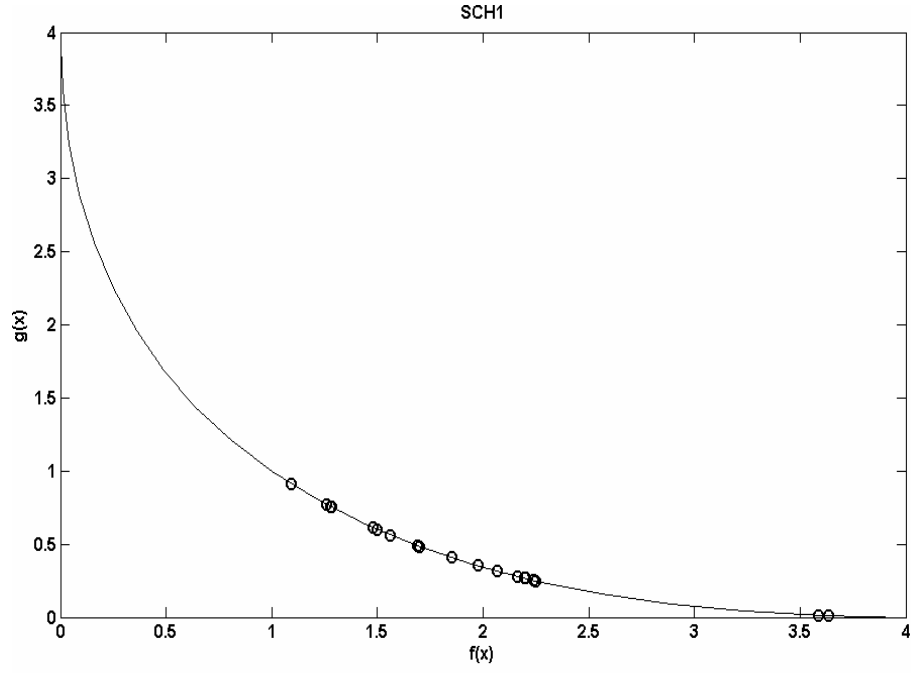
- 4) GA işlevleri olan seçme (RWS veya SUS), çaprazlama ve mutasyon uygulanır.

Şekil 3.2. CWOF algoritması

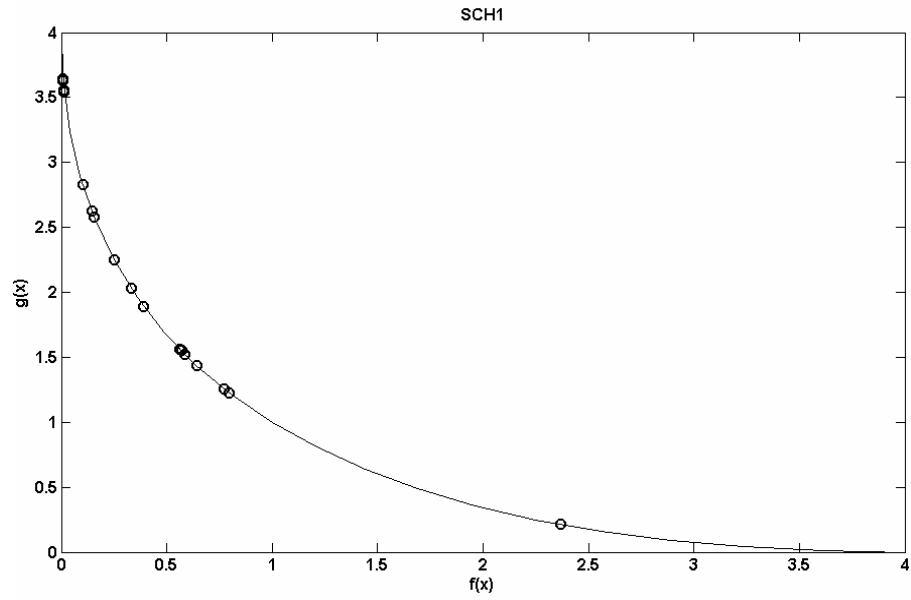
CWOF yönteminin, (3.4, 3.5 ve 3.6) eşitliklerinde verilen üç farklı durum için SCH1 test işlevinde çalıştırılmasıyla elde edilen sonuçlar Şekil 3.3, Şekil 3.4 ve Şekil 3.5' te verilmektedir.



Şekil 3.3. $F(x) = 0.5f(x) + 0.5g(x)$ için CWOF sonucu



Şekil 3.4. $F(x) = 0.25f(x) + 0.75g(x)$ için CWOFF sonucu



Şekil 3.5. $F(x) = 0.75f(x) + 0.25g(x)$ için CWOFF sonucu

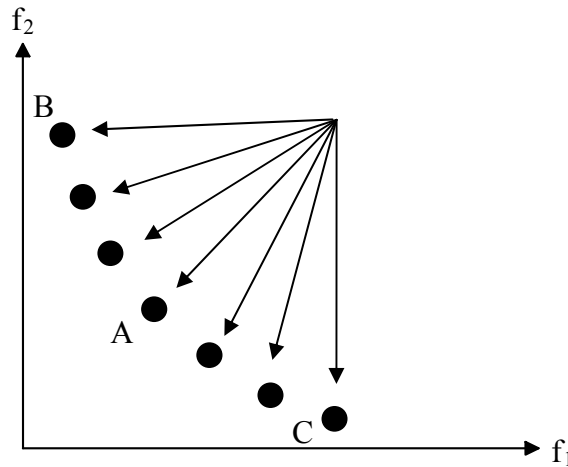
3.1.2. Değişken Ağırlıklı Amaç İşlevi

(Variable Weight Objective Function, VWOJ)

Bu yöntemde, ayarlanmış belirli ağırlıklardan çok rasgele seçilmiş ağırlık kümeleri kullanılır. VWOJ ilk kez (Murata ve Ishibuchi, 1995; Ishibuchi ve Murata, 1998)' te CWOFF' taki eksikleri gidermek için oluşturulmuştur. Bilindiği üzere, CWOFF arama uzayında tek bir yönde arama yapmaktadır ve tüm Pareto çözümlerini bulmak için yeterli değildir. Bunun nedeni, çok amaçlı uzayda farklı arama yönlerine gereksinim olmasıdır. Farklı arama yönlerini gerçekleştirmek için, rasgele belirlenmiş ağırlık değerleri önerilmiştir (Murata ve Ishibuchi, 1995; Ishibuchi ve Murata, 1998).

VWOJ (3.2) ve (3.3) nolu eşitlikleri kullanır ancak, ağırlıklar (3.7)' da verilen eşitliği de sağlamalıdır. Ağırlıklar artık sabit değildir ve GA' da artık çok amaçlı arama uzayında tek bir yönde arama yapmayacaktır. VWOJ farklı arama yönlerine sahiptir (Bkz. Şekil 3.6). CWOFF ve VEGA, VWOJ' un özel durumlarıdır. (3.6)' daki biçimdeki bir CWOFF, A Pareto çözümünü kolayca bulabilir ancak her iki köşede yer alan B ve C Pareto çözümlerini bulması ise çok zordur. VEGA, B ve C çözümlerini kolayca bulur; çünkü B , 2. amacın C ise 1. amacın en iyi çözümleridir; ancak VEGA iki amacın karışımı olan bir çözüm olan A benzeri bir çözümü ise çok zor bulacaktır.

$$w_i = \frac{rasgele_i}{(rasgele_1 + \dots + rasgele_n)} \quad i = 1, 2, \dots, n \quad (3.7)$$



Şekil 3.6. VWOJ Pareto çözümlerini bulmak için farklı yönlerde arama yapar

VWOJ Algoritması

- 1) Başlangıç topluluğu oluşturulur (Rasgele veya dışarıdan girilerek)
- 2) w_i ağırlıkları rasgele belirlenir (w_i ' lerin sayısı amaçlar kadardır)

$$w_i = \frac{rasgele_i}{(rasgele_1 + \dots + rasgele_n)} \quad i=1,2,\dots,N$$

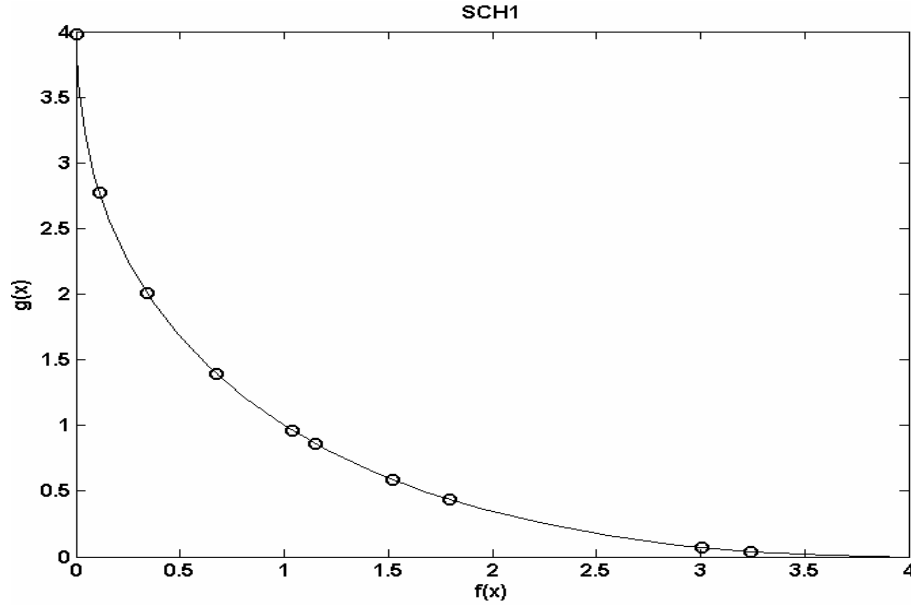
- 3) $F(x) = w_1 f_1(x) + w_2 f_2(x) + w_3 f_3(x) + \dots$

Yukarıdaki formülle bütün amaçlar rasgele belirlenen ağırlıklarla çarpılarak, tek bir amaç işlevi olarak düşünülür. Burada ağırlıklar sabit değildir ve rasgele belirlenir. Ancak bu rasgele ağırlıkların toplamı 1 olmalıdır. Ağırlığı yüksek olan amaç daha önemli olacaktır.

- 4) GA işlevleri olan seçme, çaprazlama ve mutasyon uygulanır.

Şekil 3.7. VWOJ algoritması

VWOJ yönteminin SCH1 test işlevinde çalıştırılmasıyla elde edilen sonuçlar Şekil 3.8' de verilmektedir.

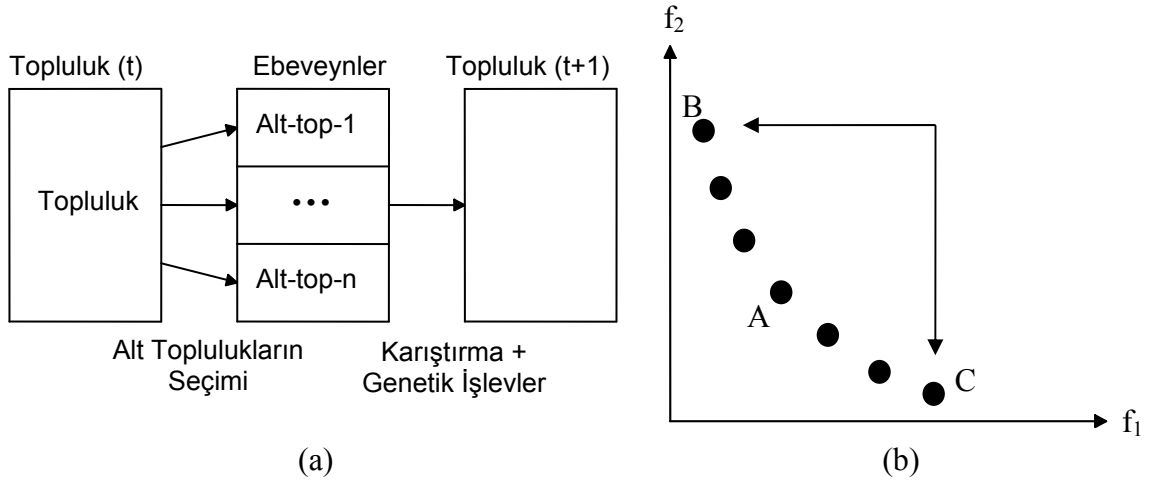


Şekil 3.8. VWOJ örneği

3.1.3. Vektör Hesaplamalı Genetik Algoritma (Vector Evaluated Genetic Algorithm, VEGA)

VEGA, çok amaçlı bir GA tasarlanması için atılan ilk adımdır (Schaffer, 1985). VEGA' nın arkasında yatan temel düşünce, bir sonraki topluluğun alt kümesinin tek bir amaç tarafından seçilmesi ve gelecek topluluğun oluşturulması için tüm alt kümelerin birlikte kullanılmasıdır. Gelecek topluluk, her amacın en iyi bireylerinden oluşmaktadır.

VEGA Pareto çözümlerini bulurken her bir amacı ayrı ayrı kullanır. Eğer n adet amaç varsa, o zaman o anki topluluktan n adet alt topluluk seçilmelidir. Her alt topluluk, tek bir amaç tarafından seçilir ve seçilen her bir alt topluluk sonraki topluluğun yalnızca bir bölümünü oluşturur. Bir sonraki adımdaki topluluk, alt toplulukların birleşimidir.



Şekil 3.9. (a) VEGA blok diyagramı, (b) VEGA' nın arama yönü

VEGA algoritması, (3.8) ifadesinde tanımlandığı gibi, amaç işlevlerinin bir çeşit doğrusal birleşimi olarak çalışır. VEGA o andaki topluluktan alt toplulukları seçerken (3.8)' de verilen eşitlikleri kullanır. $F(x)$, her alt topluluk seçimi için tek bir amaç tarafından belirlenir.

$$F(x) = 1 f_1(x) + \dots + 0 f_n(x) \quad (3.8a)$$

$$F(x) = 0 f_1(x) + 1 f_2(x) + \dots + 0 f_n(x) \quad (3.8b)$$

$$F(x) = 0 f_1(x) + \dots + 1 f_{n-1}(x) + 0 f_n(x) \quad (3.8c)$$

$$F(x) = 0 f_1(x) + \dots + 1 f_n(x) \quad (3.8d)$$

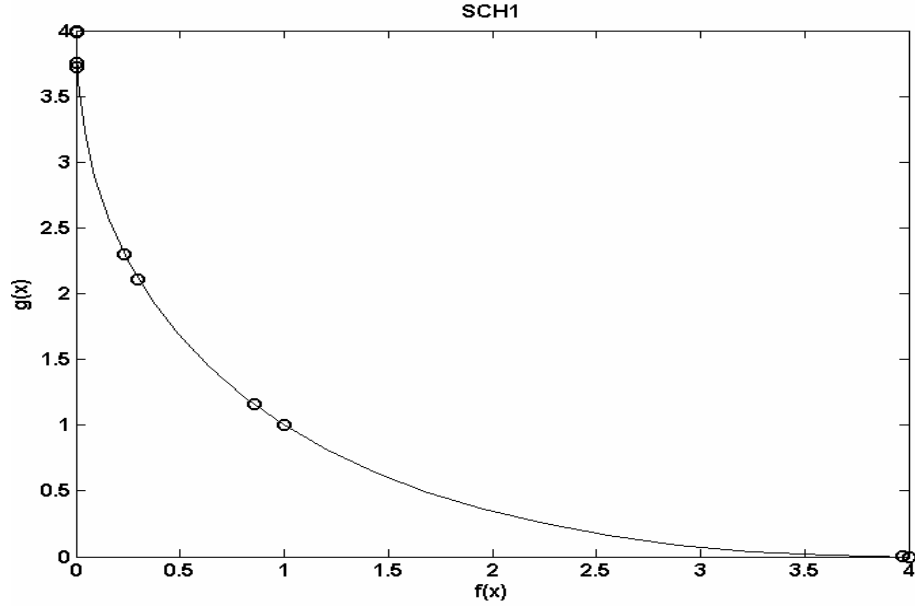
Belirli bir amaç işlevi tarafından seçilen her bir alt topluluk, daha sonra birleştirilir ve klasik GA işlevleri (çaprazlama ve mutasyon) uygulanır. VEGA Pareto-tabanlı bir yaklaşım değildir, bu yüzden ancak birkaç Pareto çözümü bulabilir. Şekil 3.9a’ da, VEGA’ nın blok diyagramı verilmektedir. Şekil 3.9b’ den de görüleceği üzere, VEGA’ nın arama yönlerinin hepsi amaç uzayının bir eksenine paraleldir. VEGA yönteminin algoritması ise Şekil 3.10’ da verilmektedir.

VEGA Algoritması

- 1) Amaç işlevi sayıcısını $i=1$ ve $q=N/M$ ifadesi yap. (N: topluluk boyutu, M: amaç sayısı). Amaç sayısı kadar alt topluluk elde edilecektir.
- 2) Amaç işlevleri ayrı ayrı kullanılarak bireylere ilişkin puanlar hesaplanır.
- 3) Eğer $i=M$ ise, 4. adıma geçilir. Aksi takdirde i bir artırılır ve 2. adıma dönülür.
- 4) Her bir amaç için belirlenen alt topluluklar birleştirilir. Yani her bir amaç ile elde edilen bireyler tek bir birey havuzunda (P) birleştirilir. En iyi puana sahip olan bireyler seçme işlemi ile belirlenir.
- 5) Yeni bir topluluk oluşturmak için birey havuzuna (P) çaprazlama ve mutasyon işlevleri uygulanır.

Şekil 3.10. VEGA algoritması

VEGA yönteminin SCH1 test işlevinde çalıştırılmasıyla elde edilen sonuçlar Şekil 3.11’ de verilmektedir.



Şekil 3.11. VEGA örneği

3.2. Pareto Tabanlı ÇAGA Yöntemleri

Son yıllarda özellikle 1998' den sonra, evrimsel algoritmalar ile çok amaçlı optimizasyon konusunda birçok araştırma yapılmıştır ve çok iyi yöntemler önerilmiştir. Bu yöntemlerin çoğu Pareto tanımına dayanan puan atama yöntemleri kullanmaktadırlar: bastırılmayan sıralama, baskınlık sayımı ve bastırılmayan çözümlerin tanımlanması [Huband ve ark., 2006].

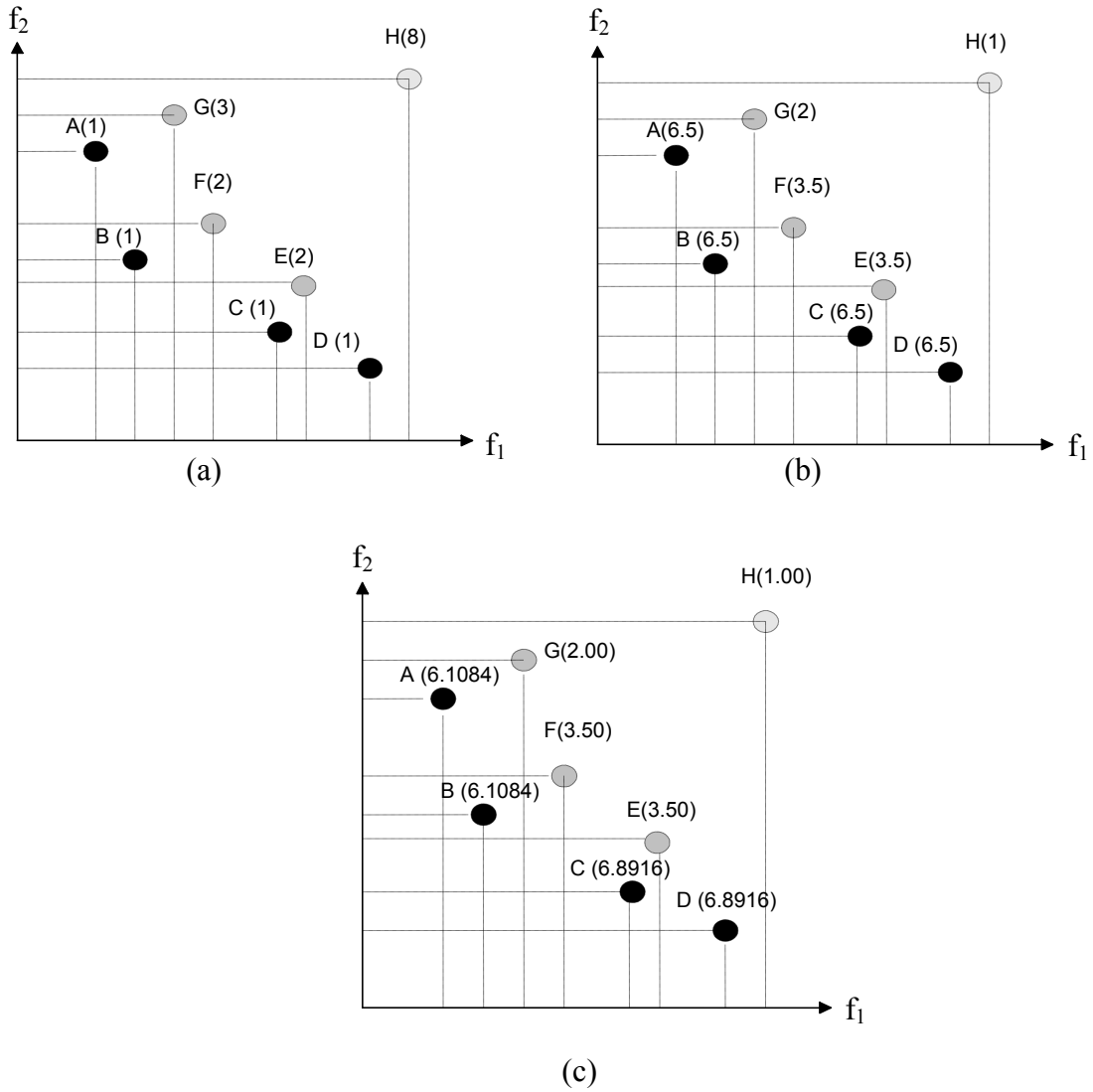
Puanlama (fitness assignment) hala ucu açık bir soru olduğu için ve Pareto tanımından dolayı literatürde çok sayıda yöntem mevcuttur [Coello Coello, 1999; Deb, 2001; Ghosh ve Dehuri, 2004; Coello Coello ve ark., 2007] ve önerilmeye de devam edilmektedir. Bu bölümde, tezde önerilen yöntemlerin karşılaştırılmasında kullanılan ve literatürde sık kullanılan ÇAGA yöntemleri ayrıntılı bir biçimde açıklanmaktadır. Yöntemlerin puanlama algoritmaları, elitizm/arşiv yöntemi mekanizmaları verilmektedir.

3.2.1. MOGA (Multiple Objective Genetic Algorithm, Çok Amaçlı Genetik Algoritma)

MOGA, Fonseca ve Fleming tarafından önerilmiştir [Fonseca ve Fleming, 1993; 1995; 1998]. Pareto tabanlı, elitist olmayan bir yöntemdir. MOGA’ da bir bireyin puanı o andaki toplulukta o bireye baskın olan bireylerin sayısına 1 eklenmesiyle bulunur (Şekil 3.13’ de Adım 1). Böylece baskın bireylerin puanı, toplulukta onlara baskın olan herhangi bir birey olmadığından dolayı 1 olarak atanır. MOGA algoritmasının ilk üç adımı şu sorunun yanıtını vermektedir: *"bir topluluktaki bireylerin puanı, onlara baskın olan bireylerin sayısı (artı 1) kullanılarak belirlenebilir mi ? "*.

Bu puanlama (alt topluluklara ayırma) işlemi tamamlandıktan sonra, bir bireyin ham puanı onun puanına göre belirlenir. Bunu yapmak için, ilk olarak puanlar azalan bir şekilde sıralanır. Daha sonra, her bireye doğrusal bir işlev yardımıyla bir ham puan atanır. Genellikle, bu işlev, N (en iyi puanlı birey için) ile I (en kötü puanlı birey için) arasında bir puan atayacak bir şekilde seçilir. Daha sonra, aynı alt topluluktaki bireyler için ham puanların ortalaması hesaplanır (Şekil 3.13’ de Adım 4). Bu işlem, daha iyi puanlı bireylerin daha yüksek bir atanmış puana sahip olmalarını garantiler. Pareto (bastırılamayan) bireylerin çeşitliliğinin sağlanması için, Fonseca ve Fleming bir paylaşım yöntemi önermişlerdir. Yarıçap sayısı hesaplandıktan sonra, bir bireyin paylaşılmış puan değeri, atanmış puanının yarıçap sayısına bölünmesi ile bulunur. Atanmış puan değerinin yarıçap sayısına (daima 1’ e eşit veya büyüktür) bölünmesi bireyin gerçek puanını azaltır. Daha sonra bu puan değerleri ölçeklenir; böylece ortalama paylaşılmış puan değeri, ortalama atanmış puan değeri ile aynı olur (Şekil 3.13’ te Adım 5). Bu hesaplamalardan sonra, diğer alt topluluklar için hesaplamalar tekrarlanır. Bu işlemler tüm alt topluluklardaki bireyler puanlanana kadar devam ettirilir [Fonseca ve Fleming, 1993; Deb, 2001].

Bireyleri MOGA tarafından puanlanan örnek bir topluluk Şekil 3.12’ de gösterilmektedir. Şekil 3.12a’ ya göre bireylerin en çok istenilenden en az istenilene doğru sıralaması şöyledir: $A(1), B(1), C(1), D(1) \succ E(2), F(2) \succ G(3) \succ H(8)$. Bireylerin ortalama puan değerleri Şekil 3.12b’ de verilmektedir. Ölçeklenmiş puan değerleri ise Şekil 3.12c’ de verilmektedir ve buna göre topluluktaki bireylerin en çok istenilenden en az istenilene doğru sıralaması şöyle olacaktır: $D, C \succ B, A \succ E, F \succ G \succ H$.



Şekil 3.12. (a) MOGA tarafından atanan ham puanlar
 (b) MOGA tarafından atanan ortalama puanlar
 (c) MOGA tarafından atanan ölçeklenmiş veya son puanlar ($\sigma_s=1.3$)

MOGA yönteminde çeşitliğin sağlanması için paylaşım yöntemi kullanılır. Paylaşım işlevi amaç uzayında uygulanır. Paylaşım yarıçapının (σ_s) çok iyi seçilmesi gerekmektedir. Eğer bu yarıçap probleme göre iyi seçilirse çeşitlilik sağlanabilir. MOGA'nın başarımı σ_s 'in seçimine önemli derecede bağlıdır. Fonseca ve Fleming, paylaşım yarıçapının dinamik olarak güncellenmesi için bir yöntem önermişlerdir, böylece program başında sabit bir yarıçap seçilmesine (tahminine) gerek kalmamaktadır. MOGA yöntemi, Pareto-optimal cephenin biçimine ve arama uzayındaki çözümlerin yoğunluğuna çok duyarlıdır.

MOGA Puan Atama Algoritması [Deb, 2001]

Adım 1: σ_s parametresi seçilir. Tüm olası dereceler için $\mu(j)=0$ yapılır [$j=1, \dots, N$ (topluluk boyutu)]. Birey sayacını $i=1$ yap.

Adım 2: i bireyine baskın olan birey sayısı (n_i) belirlenir. i . bireyin derecesi $r_i=1+n_i$ formülü ile hesaplanır. r_i derecesine sahip bireylerin sayısını bir artır $\mu(r_i)=\mu(r_i)+1$.

Adım 3: Eğer $i < N$ ise, i' yi bir artır ve Adım 1' e git. Aksi takdirde Adım 4' e git.

Adım 4: $\mu(r_i)>0$ olan en büyük r_i değerini kontrol ederek en yüksek dereceyi (r^*) belirle. Her birey için ortalama puan değerini hesapla:

$$F_i = N - \sum_{k=1}^{r_i-1} \mu(k) - 0.5(\mu(r_i) - 1) \quad (3.9)$$

$r_i=1$ derecesine sahip her i bireyi için, yukarıdaki eşitlik $F_i = N - 0.5(\mu(1) - 1)$ olur. Derece sayıcısını $r=1$ yap.

Adım 5: r derecesine sahip her bir i bireyi için aynı derecedeki bireyleri kullanarak yarıçap sayısını hesapla. $F_j' = F_j / n_{c_i}$ ile paylaşılmış puanı hesapla. Aynı ortalama puanı sağlamak için, paylaşılmış puan aşağıdaki gibi bir ölçekleme çarpanı ile çarpılır:

$$F_j'' = \frac{F_j \mu(r)}{\sum_{k=1}^{\mu(r)} F_k'} F_j' \quad (3.10)$$

Adım 6: Eğer $r < r^*$ ise, r' yi bir artır ve Adım 5' e git. Aksi takdirde işlemi sonlandır.

Şekil 3.13. MOGA puan atama algoritması

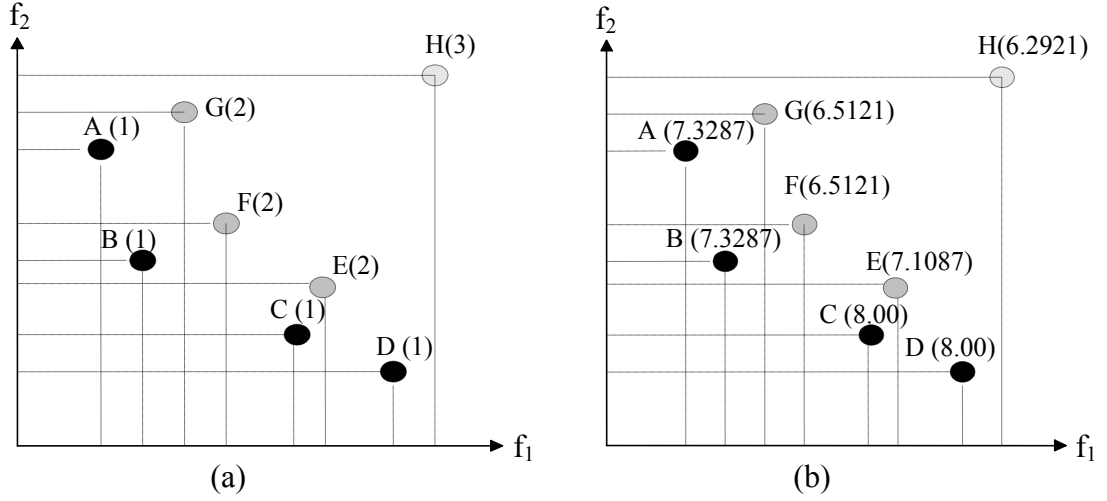
3.2.2. NSGA (Non-Dominated Sorting Genetic Algorithm, Bastırılmayan Sıralamalı Genetik Algoritma)

NSGA ilk olarak Goldberg tarafından önerilmiştir [Goldberg, 1989], ancak uygulaması Srivinas ve Deb tarafından yapılmıştır [Srivinas ve Deb, 1994]. Pareto tabanlı, elitist olmayan bir yöntemdir. NSGA ilk bastırılmayan bireylerden oluşan cepheye **1** puanını atar ve daha sonra bu bireyler o andaki topluluktan silinirler. Daha sonra, azaltılmış topluluktaki ikinci bastırılmayan cephedeki bireylere **2** puanı atanır. Bu işlem, tüm cephelerdeki bireylere puan atanana kadar devam eder.

NSGA şu sorunun yanıtını verir: *"bir topluluk birbirlerine baskın olmayan bireylerin oluşturduğu alt topluluklara ayrılabilir mi?"* (paylaşım yöntemi uygulanmadan). Bu sorunun sonucu olarak, NSGA birkaç alt topluluk oluşturur ve her alt topluluk aynı puanlara sahip bireylerden oluşur. NSGA algoritmasının 2. adımına göre puanlanmış bir topluluk Şekil 3.14a' da gösterilmektedir. Şekil 3.14a' da üç adet alt topluluk bulunmaktadır. İlk alt topluluk $\{A(1), B(1), C(1) \text{ ve } D(1)\}$, ikinci alt topluluk $\{E(2), F(2) \text{ ve } G(2)\}$ ve üçüncü alt topluluk ise $\{H(3)\}$ bireylerinden oluşmaktadır. Açıkça görüleceği üzere, ilk alt topluluk ikinciden, ikinci alt topluluk ise üçünden daha çok istenen bireylerden oluşmaktadır. Eğer bireyler aynı Pareto cephesine aitse, örneğin ilkinde, bu durumda hangi bireyin diğerlerinden daha iyi olduğuna karar vermek kolay değildir. Buna göre, bu yöntem aynı Pareto cephesine ait bireylerin ayırt edilmesi için hiçbir bilgi vermemektedir. Bireyleri çeşitliliğini sağlayabilmek için, NSGA yöntemine paylaşım yöntemi eklenmiştir [Deb, 2001; sayfa 149-160]. Orjinal NSGA paylaşımı parametre uzayında uygularken, bu tezde karşılaştırmalarda kullanılan NSGA' da paylaşım MOGA yönteminde olduğu gibi amaç uzayında uygulanmıştır.

Bireylerin paylaşılmış puanlarının bulunması için şu yol izlenir: ilk olarak; ilk Pareto cephesine ait (cephe sayıcısını **1** yap) tüm bireylerin puanı topluluk boyutuna (N) eşitlenir. Bu, herhangi bir topluluktaki herhangi bir bireyin alabileceği en büyük puan değeridir. Paylaşım yöntemine göre, eğer bir bireyin aynı cephede çok sayıda komşusu varsa, o bireyin puanı yarıçap sayısı kadar azaltılır ve paylaşılmış puan (Şekil 3.15' te Adım 3) kullanılarak hesaplanır. İlk cephedeki bireylerin puanları hesaplandıktan sonra, en küçük paylaşılmış puan değeri belirlenir ve cephe sayıcısı bir artırılır (Şekil 3.15' te Adım 4). Bundan sonra, ikinci cephedeki tüm bireylere birinci cephedeki en küçük

paylaşılmış puandan daha küçük bir puan değeri atanır. Bu işlem, ikinci cephedeki herhangi bir bireyin birinci cephedeki herhangi bir bireyden daha iyi bir paylaşılımış puan değeri almasını engeller. Paylaşım yöntemi daha sonra, ikinci cephedeki tüm bireylere uygulanır ve paylaşılımış puan değerleri hesaplanır. Bu işlem, topluluktaki tüm bireylere paylaşılmış puan değeri atanana kadar devam eder. Şekil 3.14b’ de, $\sigma_s = 1.3$ ve $\varepsilon = 0.22$ için örnek bir topluluktaki bireylerin paylaşılmış puan değerleri gösterilmektedir.



Şekil 3.14. (a) NSGA ham puanları
(b) NSGA paylaşılmış puanları ($\sigma_s = 1.3$ ve $\varepsilon = 0.22$ için)

Bu topluluktaki bireylerin en çok istenilenden en az istenilene doğru sıralaması Şekil 3.14a’ ya göre şöyledir: A(1),B(1),C(1),D(1) $\succ$ E(2),F(2),G(2) $\succ$ H(3). Bu sıralama Şekil 3.14b’ ye göre ise şöyle verilir: D,C $\succ$ B,A $\succ$ E $\succ$ F,G $\succ$ H .

NSGA yönteminde de Fonseca ve Fleming tarafından önerilen dinamik yarıçap güncellenmesi kullanılabilir, böylece yarıçap tahmini ortadan kaldırılabilir. NSGA’ da da MOGA’ da olduğu gibi yarıçap seçimi yöntemin başarımına önemli derecede etki etmektedir.

NSGA Puan Atama Algoritması [Deb, 2001]

- Adım 1:** Paylaşım yarıçapı (σ_s) ile küçük pozitif bir sayı ε seçilir ve $F_{\min}=N+\varepsilon$ ile program başlatılır. Cephe sayıcı $j=1$ yapılır (N : topluluk boyutu),
- Adım 2:** P topluluğu baskınlık durumuna göre sınıflandırılır:
 $(P_1, P_2, \dots, P_\rho) = \text{Sırala}(P, \leq)$,
- Adım 3:** Her $q \in P_j$ bireyi için
- 3a.** Puan atanır $F_j(q) = F_{\min} - \varepsilon$
- 3b.** Yarıçap sayısı nc_q , $nc_i = \sum_{j=1}^N sh(d_{ij})$ formülüne göre hesaplanır
- 3c.** Paylaşılmış puanlar hesaplanır: $F'_j(q) = F_j(q)/nc_q$
- Adım 4:** $F_{\min} = \min(F'_j(q) : q \in P_j)$ ve $j=j+1$ yapılır,
- Adım 5:** Eğer $j \leq \rho$ ise, Adım 3' e gidilir. Aksi takdirde, işlem tamamlanır.

Şekil 3.15. NSGA puan atama algoritması

3.2.3. SPEA (Strength Pareto Evolutionary Algorithm, Pareto' nun Gücü Evrimsel Algoritma)

SPEA, Pareto tanımına dayalı elitist bir ÇAGA yöntemidir [Zitzler, 1999; Zitzler ve Thiele, 1999]. Farklı bir kümede (arşiv veya ikincil topluluk) saklanan elit bireylerin sayısını, çözüm kümesinin karakteristiğini bozmadan kümeleme yöntemi kullanarak azaltır. Arşivdeki tüm bireyler seçme işleminde kullanılır. SPEA yönteminde, birey puanları iki aşamada atanır: *i*) Pareto bireylerin puanlanması, *ii*) Pareto olmayan bireylerin puanlanması.

3.2.3.1. Pareto Bireylerin Puanlanması

Bir Pareto bireyin puanı, baskın olduğu bireylerin sayısı ile orantılıdır. n bir Pareto bireyin baskın olduğu bireylerin sayısı ve N de Pareto olmayan bireylerin (birinci Pareto cephesi dışındaki tüm bireylerin) toplam sayısı olsun. Buna göre, bir Pareto bireyin puanı (ya da gücü, strength) şu şekilde hesaplanır: $f_i = \frac{n}{N+1}$. Pareto bireylerin puanları aynı zamanda güçlerine de eşittir ($f_i = s_i$). Bu durumda, eğer iki Pareto birey aynı sayıda bireye baskın oluyorsa puanları aynı olacaktır, bu durumda bu iki bireyi seçme mekanizmasının ayırt etmesi zor olacaktır.

SPEA yönteminin birey puanlamasını daha iyi açıklamak için Şekil 3.16' daki örneği inceleyelim. C bireyi, iki Pareto olmayan bireyden (E ve H) baskındır, bu yüzden $n=2$ olur ve toplulukta dört adet Pareto olmayan birey (E, F, G ve H) olduğundan $N=4$ tür. Buna göre, C bireyinin puanı $n/(N+1)=2/5$ olarak bulunur. Geriye kalan diğer Pareto bireylerin puanları da aynı yöntemle hesaplanır.

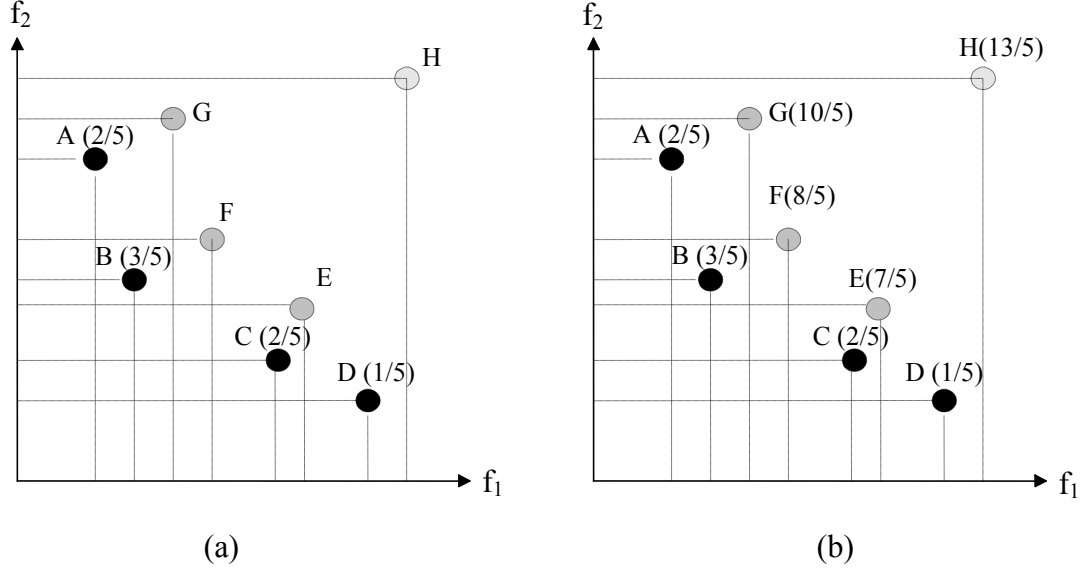
3.2.3.2. Pareto Olmayan Bireylerin Puanlanması

İlk adımda tüm Pareto bireyler puanlanmıştır, ancak Pareto olmayan bireyler puanlanmamıştır (Bkz. Şekil 3.16a). Pareto olmayan bir bireyin puanı; ona baskın olan tüm Pareto bireylerin puanları toplamına **1** eklenmesiyle bulunur ($f_j = 1 + \sum s_i$). Şekil 3.16b' de verilen E bireyi, yalnızca C Pareto bireyi tarafından bastırılmaktadır ve C bireyinin puanı da $2/5$ tir. Bu yüzden E bireyini puanı şöyle hesaplanır: $1 + (2/5) = 7/5$. Pareto olmayan diğer bireylerin puanları da bu şekilde hesaplanır. Bu puanlama yöntemi, en düşük puan değerini en çok istenilen bireye, en yüksek puan değerini ise en az istenilen bireye atar.

SPEA tarafından puanlanmış örnek bir topluluk Şekil 3.16b' de verilmektedir. Görüleceği üzere, A ve C bireyleri aynı puana sahiptir. A ve C bireylerinin aynı puana sahip olmalarının nedeni, ikisinin de iki adet Pareto olmayan bireye baskın olmalarıdır. Bu durumda bu iki bireyin ayırt edilmesi zor olacaktır. Ayrıca, eğer bir grup Pareto olmayan birey yalnızca bir Pareto bireye baskın olursa, bu durumda bunları birbirinden

ayırt etmenin hiçbir yolu yoktur. Şekil 3.16b’ deki bireylerin en çok istenilenden en az istenilene doğru sıralaması şöyledir:

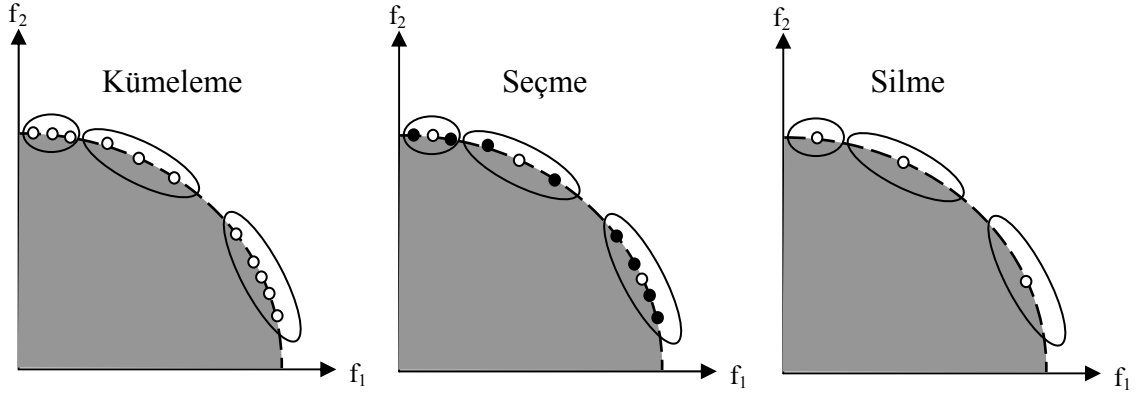
$$D(1/5) \succ \{A(2/5), C(2/5)\} \succ B(3/5) \succ E(7/5) \succ F(8/5) \succ G(10/5) \succ H(13/5) .$$



Şekil 3.16. (a) SPEA ile Pareto bireylerin puanlanması
(b) SPEA ile tüm bireylerin puanlanması

3.2.3.3. SPEA Elitizm Mekanizması

SPEA yönteminde, her bir adımda bulunan elit (ya da Pareto) bireyler arşiv olarak adlandırılan ikinci bir toplulukta ($\bar{P}$) saklanır. Arşiv boyutu sabittir. Daha sonra, her adımda, yeni bulunan elit bireyler arşiv ile karşılaştırılır ve baskın olanlar arşivde saklanmaya devam eder, baskın olmayanlar ise silinirler. İlerleyen adımlarda elit bireylerin sayısında artış olabilir. Bu durumda, yalnız arşiv boyut kadar birey arşivde saklanır. Eğer arşiv boyutu aşılsa, bir çeşit kümeleme yöntemi kullanılarak kalabalık bölgede olan elit bireyler arşiv boyutuna ulaşıncaya kadar silinir, böylece çeşitlilik te sağlanmış olur. Bu işlem şöyle yapılır: i) ilk olarak, birbirine yakın bireyler kümelenir, ii) her küme için o kümeyi temsil eden bir birey seçilir, iii) geriye kalan bireyler kümeden silinir. Bu kümeleme işlemi grafiksel olarak Şekil 3.17’ de gösterilmektedir. Orjinal SPEA’ da kümeleme amaç uzayında yapılmaktadır, ancak parametre uzayında da gerçekleştirilebilir.



Şekil 3.17. SPEA çeşitlilik işlemi

SPEA yönteminde, topluluk boyutu ile arşiv boyutu arasındaki dengenin sağlanması başarıma etki etmektedir. Eğer arşiv boyutu çok büyük seçilirse, elit bireylerin seçilme baskısı da büyük olacaktır ve bu durumda SPEA Pareto-optimal cepheye yakınsayamayabilir. Diğer taraftan, eğer daha küçük bir arşiv kullanılırsa, elitizmin etkisi kaybedilir. SPEA yöntemini önerenler arşiv boyutu ile topluluk boyutu arasında 1:4 oranını kullanmışlardır. Ayrıca, arşivdeki bireyler seçme mekanizmasında da kullanılır. Bir sonraki topluluğu oluşturacak bireyler, arşiv + ana topluluk birleşiminden seçilir.

3.2.4. SPEA2 (Strength Pareto Evolutionary Algorithm 2, Pareto' nun Gücü Evrimsel Algoritma 2)

SPEA2, SPEA yönteminin geliştirilmiş bir biçimdir. SPEA2, SPEA yöntemine göre daha iyi bir puanlama mekanizması, bir yoğunluk tahmin tekniği ve iyileştirilmiş bir arşiv (ikincil topluluk) yönetimi getirmektedir [Zitzler ve ark., 2001].

3.2.4.1. SPEA2 Puanlama Algoritması

Aynı arşiv üyeleri tarafından bastırılan bireylerin puanların aynı olmasını önlemek için, SPEA2' de her bir birey için, hem baskı altına alındığı hem de baskın

olduğu bireyler dikkate alınmaktadır. Hem arşivdeki ($\bar{P}_t$) hem de ana topluluktaki (P_t) her i bireyine, onu bastıran bireylerin sayısını gösteren bir güç (strength) değeri $S(i)$ atanır.

$$S(i) = |\{j \mid j \in P_t + \bar{P}_t \wedge i \succ j\}| \quad (3.11)$$

Burada $+$ kümelerin birleşimini, $\succ$ sembolü ise Pareto baskınlık ilişkisini göstermektedir. Bir i bireyinin ham puanı $R(i)$, S değerlerine bağlı olarak şöyle hesaplanır:

$$R(i) = \sum_{j \in P_t + \bar{P}_t, j \succ i} S(j) \quad (3.12)$$

Bu değer, o bireyi hem arşivde hem de ana toplulukta bastıran bireylerin güçleri kullanılarak belirlenir. Burada puan değerinin küçük olması önemlidir, örneğin $R(i)=0$ ise bu birey bir Pareto bireydir, $R(i)$ değerinin büyük olması ise, o bireyin birçok birey tarafından bastırıldığı anlamına gelmektedir. Bu ham puan atama işlemi her ne kadar Pareto tanımına dayanan bir çeşitlilik sağlasa da, birçok bireyin birbirilerine baskın olamadığı durumlarda sıkıntılara yol açabilir. Bu yüzden, aynı ham puan değerlerine sahip bireylerin ayırt edilebilmesi için SPEA2' de ek bir yoğunluk bilgisi kullanılmaktadır. SPEA2' de kullanılan yoğunluk tahmin tekniği ***k. en yakın komşuluk*** yönteminden uyarlanmıştır. Burada, yoğunluk bilgisi k . en yakın komşu bireye olan mesafenin bir (azalan) işlevidir. SPEA2 yöntemini önerenler, basit olarak k . en yakın komşuya olan mesafenin tersini yoğunluk bilgisi olarak kullanmışlardır. Her i bireyinin arşivde ve o andaki ana toplulukta olan her j bireyine olan olan mesafesi (amaç uzayında) hesaplanır ve bir listede saklanır. Bu liste artan bir sırada sıralandıktan sonra, k . eleman istenilen mesafeyi (σ_i^k) gösterir. Genellikle k , örnekleme oranının karekökü olarak alınır, burada $k = \sqrt{N + \bar{N}}$ ' dir (N: topluluk boyutu, $\bar{N}$: arşiv boyutu). Buna göre, i bireyinin yoğunluğu $D(i)$ şöyle hesaplanır:

$$D(i) = \frac{1}{\sigma_i^k + 2} \quad (3.13)$$

Paydaya yoğunluk değerinin sıfırdan büyük olmasını sağlamak için iki değeri eklenmiştir, böylece $D(i) < 1$ olur. Sonuç olarak, bir i bireyinin son puanı $F(i)$, ham puanına ($R(i)$) yoğunluk bilgisinin ($D(i)$) eklenmesiyle bulunur:

$$F(i) = R(i) + D(i) \quad (3.14)$$

Örnek bir topluluğun SPEA2 yöntemi tarafından puanlanması, Şekil 3.18' de iki adım olarak gösterilmektedir. İlk adımda, bireylerin güç değerleri hesaplanmaktadır. Bir bireyin gücü, bu bireyin baskın olduğu bireylerin toplam sayısı ile belirlenir:

$$S(A)=2, S(B)=3, S(C)=2, S(D)=1, S(E)=1, S(F)=1, S(G)=1, S(H)=0 \quad (3.15)$$

A bireyi iki bireye $\{G,H\}$ baskın olduğu için, A bireyinin gücü $S(A)=2$ olur. H bireyi hiçbir bireye baskın olmadığı için H bireyinin gücü $S(H)=0$ olur. Bir bireyin gücü, onun bağlı baskın olma yeteneğini gösterir. İkinci adım, ham puanların belirlendiği adımdır. Bir bireyin ham puanı, ona baskın olan bireylerin güçleri toplamıdır. Bu tanıma göre, Pareto ya da baskın bireyler $\{A,B,C,D\}$ “ 0 ” ham puan değerini alır. Örneğin A bireyinin ham puanı hiçbir birey tarafından bastırılmadığı için $R(A)=0$ olurken, H bireyi tüm bireyler tarafından bastırıldığı için $R(H)=11$ olur. Tüm bireylerin ham puanları aşağıda verilmektedir:

$$R(A)=R(B)=R(C)=R(D)=0 \quad (3.16a)$$

$$R(E)=S(C)=2 \quad (3.16b)$$

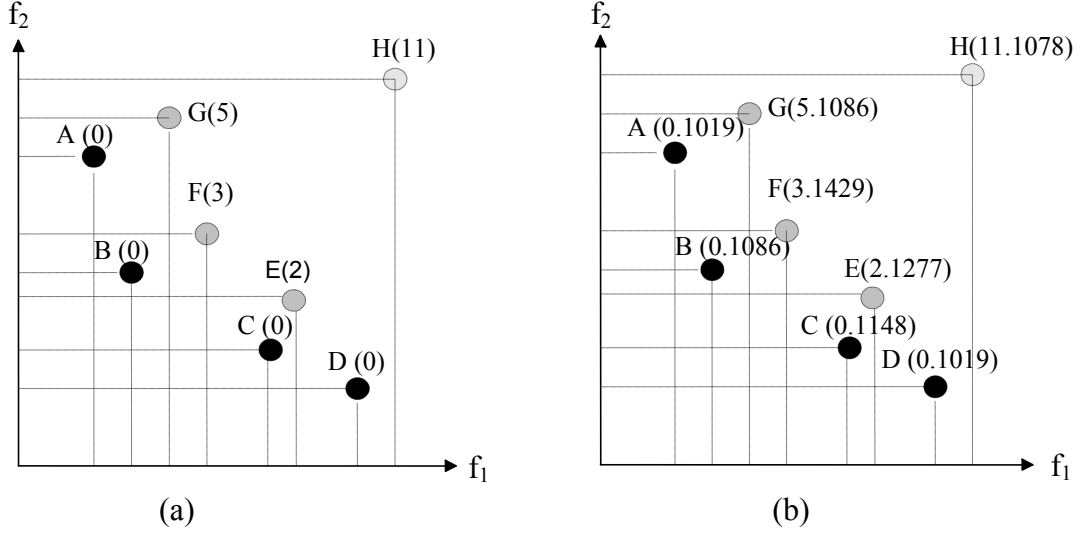
$$R(F)=S(B)=3 \quad (3.16c)$$

$$R(G)=S(A)+S(B)=2+3=5 \quad (3.16d)$$

$$R(H)=S(A)+S(B)+S(C)+S(D)+S(E)+S(F)+S(G)=2+3+2+1+1+1+1+0=11 \quad (3.16e)$$

Üçüncü adımda, her bireyin son puanı; ham puanına yoğunluk bilgisinin eklenmesiyle bulunur. Şekil 3.18a, SPEA2 yöntemi tarafından atanan ham puan

değerlerini göstermektedir, burada A, B, C ve D bireyleri aynı puana sahiptir ve bireylerin en çok istenilenden en az istenilene göre sıralaması şöyledir: $A, B, C, D \succ E \succ F \succ G \succ H$. Şekil 3.18b ise SPEA2 yöntemi tarafından atanan son puanları göstermektedir ve bireylerin en çok istenilenden en az istenilene göre sıralaması ise şöyledir: $A, D \succ B \succ C \succ E \succ F \succ G \succ H$.



Şekil 3.18. (a) SPEA2 ham puanları
(b) SPEA2 son puanları (arşiv boyutu=2, dolayısıyla $k=3.1623$)

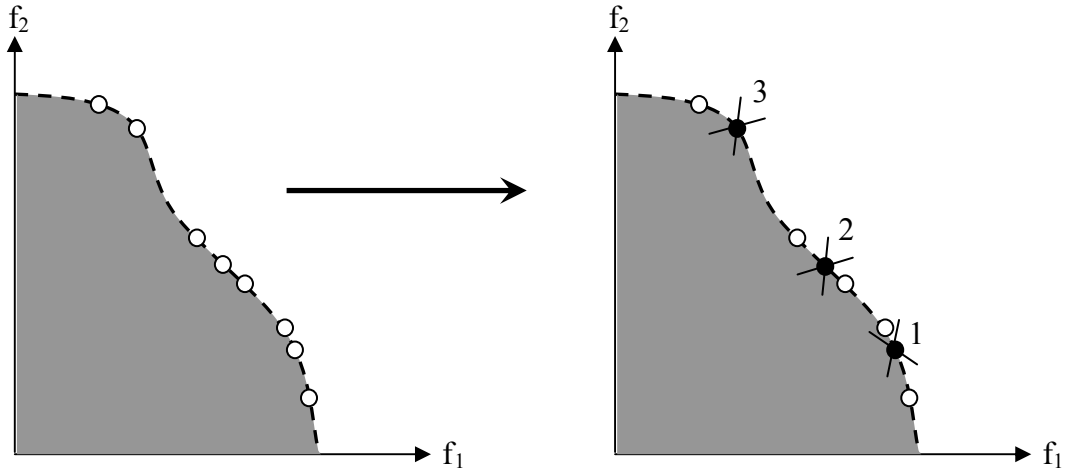
3.2.4.2. SPEA2 Arşiv Yönetimi

SPEA2 arşiv yönetiminde ilk olarak, arşivdeki veya ana topluluktaki puanı birden küçük olan tüm elit (Pareto) bireyler, bir sonraki neslin (adımın) arşivine kopyalanırlar (P-ana topluluk, $\bar{P}$ -arşiv, N-topluluk boyutu, $\bar{N}$ -arşiv boyutu) :

$$\bar{P}_{t+1} = \{i \mid i \in P_t + \bar{P}_t \wedge Puan(i) < 1\} \quad (3.17)$$

Eğer arşive kopyalanan Pareto birey sayısı, daha önceden belirlenen arşiv boyutuna ulaşırsa ($|\bar{P}_{t+1}| = \bar{N}$) artık seçme mekanizmasına geçilebilir. Aksi takdirde, iki durum söz konusu olacaktır: ya arşiv boyutuna ulaşılamaz ($|\bar{P}_{t+1}| < \bar{N}$) ya da arşiv

boyutu aşılmış olabilir ($|\bar{P}_{t+1}| > \bar{N}$). Eğer arşiv boyutuna ulaşamazsa, bir önceki adımdaki arşiv ve topluluktaki Pareto olmayan bireylerden en iyi $\bar{N} - |\bar{P}_{t+1}|$ adedi gelecek arşive kopyalanır. Bu işlem şöyle yapılır: arşiv ve ana topluluk ($P_t + \bar{P}_t$) birleştirilir ve bireyler puan değerlerine göre sıralanır. Sıralanmış listeye göre puanı 1' den büyük veya 1' e eşit olan $\bar{N} - |\bar{P}_{t+1}|$ adet birey yeni arşive ($\bar{P}_{t+1}$) kopyalanır. Eğer arşiv boyutu aşılsa yani Pareto bireylerin sayısı arşiv boyutundan fazla olursa, bu durumda bir çeşit arşiv güncelleme yöntemi kullanılarak arşivdeki bireylerden bazıları arşiv boyutuna ulaşılncaya kadar silinirler. Silme işleminde bireylerin birbirlerine olan mesafeleri göz önüne alınır. Bir bireye en yakın birey silinir. Eğer en az mesafeye sahip birkaç birey varsa, bu durumda en yakın 2. birey silinir, bu işlem böylece devam eder. Bu işlemin nasıl yapıldığına dair grafiksel anlatım Şekil 3.19' da gösterilmektedir.



Şekil 3.19. SPEA2 arşiv güncellemesi. Sağ taraftaki şekilde Pareto bireylerin bir kümesi gösterilmektedir. Sol taraftaki şekilde ise güncelleme işlevi kullanılarak bireylerin silinmesi gösterilmektedir. Bu örnekte arşiv boyutu 5 seçilmiştir

3.2.5. Diğer Yöntemler

Literatürde çok sayıda ÇAGA yöntemi önerilmiştir ve her geçen gün bu yöntemlerin sayısı artmaktadır. ÇAGA ile çok boyutlu durumdan puanlama ile tek boyuta inilmesinden dolayı, puanlama işlemini yapmanın tek bir yolu yoktur. Bu

nedenle ÇAGA yöntemlerinin sayısı fazladır. Literatürde önerilen ve sık kullanılan bazı yöntemler şöyledir: PAES (Knowles ve Corne, 2000), PESA (Corne ve ark., 2000), NSGA-II (Deb ve ark., 2002b), DMOEA (Yen ve Lu, 2003) vb. Bu yöntemler tezde kullanılmadığı için ayrıntılarına inilmeyecektir. Bu yöntemlerin etkinlikleri, genellikle test işlevleri kullanılarak araştırılmıştır. Bu yöntemlerle ve GA' larla ilgili birçok kaynağa internet üzerinden (<http://www.lania.mx/~ccoello/EMOO/>, Erişim zamanı: 10.10.2006) sitesinde ulaşılabilir.

3.3. Çok Amaçlı Genetik Algoritmalarda Kullanılan Test İşlevleri

Literatürde çok sayıda ÇAGA yöntemi önerilmiştir ve önerilmeye devam edilmektedir. Bu kadar çok sayıda yöntemin olması, bu yöntemlerden hangisi diğerlerinden daha iyidir şeklinde bir soru sorulmasına yol açmaktadır. Ayrıca, bir yöntemin etkinliğini test etmek için, o yöntemin zor problemler üzerinde denenmesi gereklidir. Bu amaçla, ÇAGA yöntemlerinin karşılaştırma çalışmalarında veya bir ÇAGA yönteminin etkinliğinin tespit edilmesinde kullanılmak üzere birçok yapay test işlevi önerilmiştir ve bunlar karşılaştırma çalışmalarında kullanılmışlardır [Deb, 1999; Deb, 2001; Huband ve ark., 2006; Coello Coello, 2007]. Bu test işlevleri, genetik algoritmaların nasıl çalıştığı konusunda çok fazla bilgi sahibi olunmadan ortaya atılmış analitik ifadelerdir. Bir ÇAGA' da istenilen iki özellik vardır: Pareto-optimal cepheye mümkün olduğunca yakınsama ve bu cephe üzerinde çözümlerin düzgün dağılımı. ÇAGA test işlevleri, genellikle bu iki amaca ulaşılmasını zorlaştıracak biçimlerde oluşturulmuştur. Örneğin, çok modluluk, ayrık Pareto-optimal cepheler, içbükeylik, dışbükeylik, düzgün dağılmamış Pareto-optimal cephe vb.

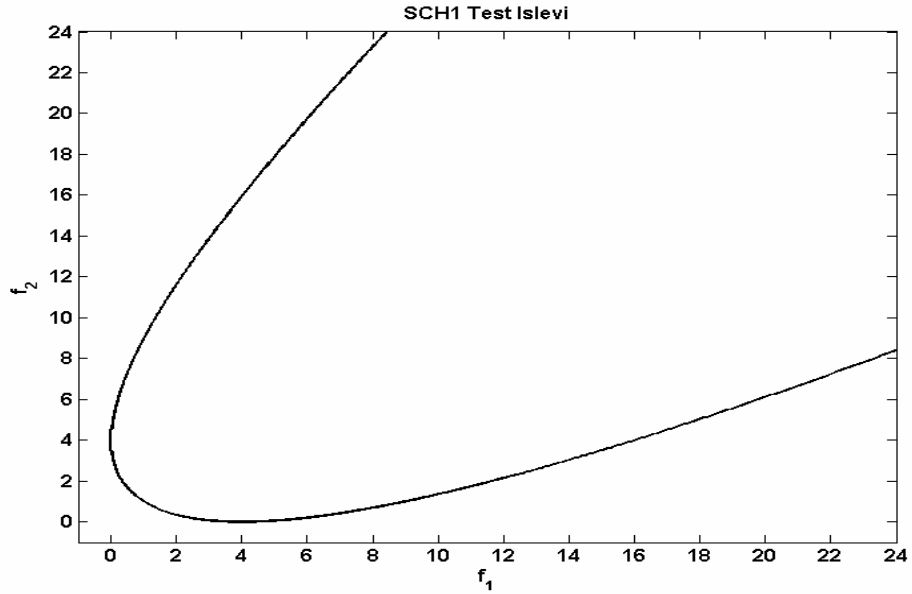
Kısıtlamalı ve kısıtlamasız çok amaçlı genetik algoritmalar için, bu tür test işlevlerinin oluşturulması amacıyla Deb tarafından sistematik bir yol da önerilmiştir [Deb, 1999]. Bu bölümde, kısıtlamasız test işlevlerinden literatürde en çok kullanılanları verilmektedir.

3.3.1. SCH1 Test İşlevi

Basit olmasına rağmen, en çok kullanılan tek parametrelili (değişkenli) test işlevidir. Schaffer tarafından önerilmiştir [Schaffer, 1984]. Bir en küçükleme problemidir ve matematiksel ifadesi aşağıdaki eşitlikte verilmektedir:

$$\begin{aligned} f_1(x) &= x^2 \\ f_2(x) &= (x-2)^2 \\ -A &\leq x \leq A \end{aligned} \quad (3.18)$$

Bu işlev, $x \in [0,2]$ aralığında Pareto-optimal çözümlere sahiptir ve Pareto-optimal çözüm kümesi $0 \leq f_1 \leq 4$ aralığındadır. Parametre sınırları için (A), farklı çalışmalarda farklı değerler kullanılmıştır. A değeri yükseldikçe, Pareto-optimal cepheye olan yakınsama zorlaşmaktadır. Bu test işlevinin $A=6$ değeri için grafiği Şekil 3.20' de verilmektedir.



Şekil 3.20. SCH1 test işlevi

3.3.2. SCH2 Test İşlevi

Schaffer tarafından önerilmiştir [Schaffer, 1984]. Birçok çalışmada ve uygulamada kullanılmıştır. Bir en küçükleme problemidir ve matematiksel ifadesi aşağıdaki eşitlikte verilmektedir:

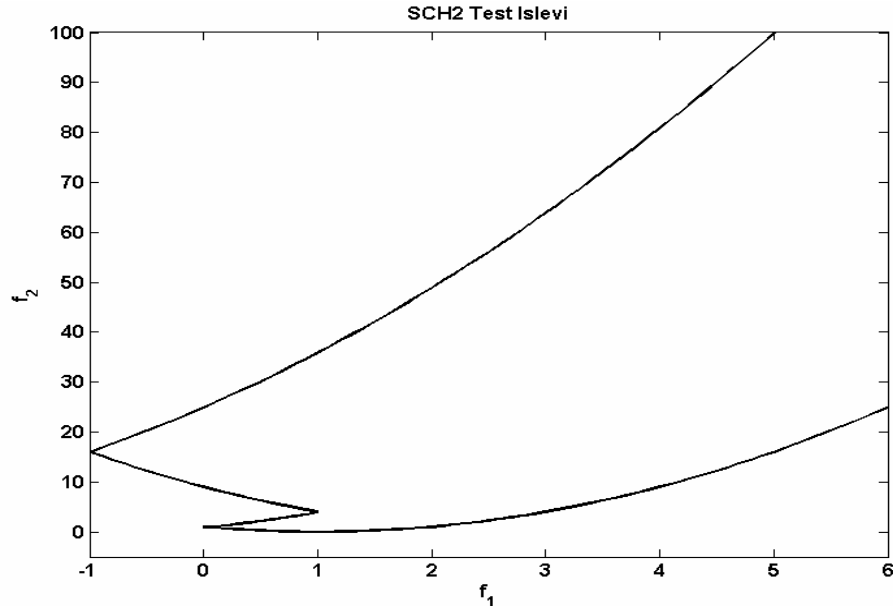
$$f_1(x) = \begin{cases} -x & x \leq 1 \\ x-2 & 1 < x \leq 3 \\ 4-x & 3 < x \leq 4 \\ x-4 & x > 4 \end{cases} \quad (3.19a)$$

$$f_2(x) = (x-5)^2 \quad (3.19b)$$

$$-5 \leq x \leq 10 \quad (3.19c)$$

Pareto-optimal cephe, iki ayırık bölgeden oluşmaktadır: $x \in \{ [1,2] \cup [4,5] \}$.

Şekil 3.21, bu test işlevinin grafiğini göstermektedir. Bu problemin zorluğu, iki ayırık bölgede yer alacak Pareto bireylerin çeşitliliğinin sağlanmasıdır.



Şekil 3.21. SCH2 test işlevi

3.3.3. FON Test İşlevi

Fonseca ve Fleming tarafından kullanılmış, n parametrelili iki amaçlı bir test işlevidir [Fonseca ve Fleming, 1995]. Bir en küçükleme problemidir ve matematiksel ifadesi aşağıdaki eşitlikte verilmektedir:

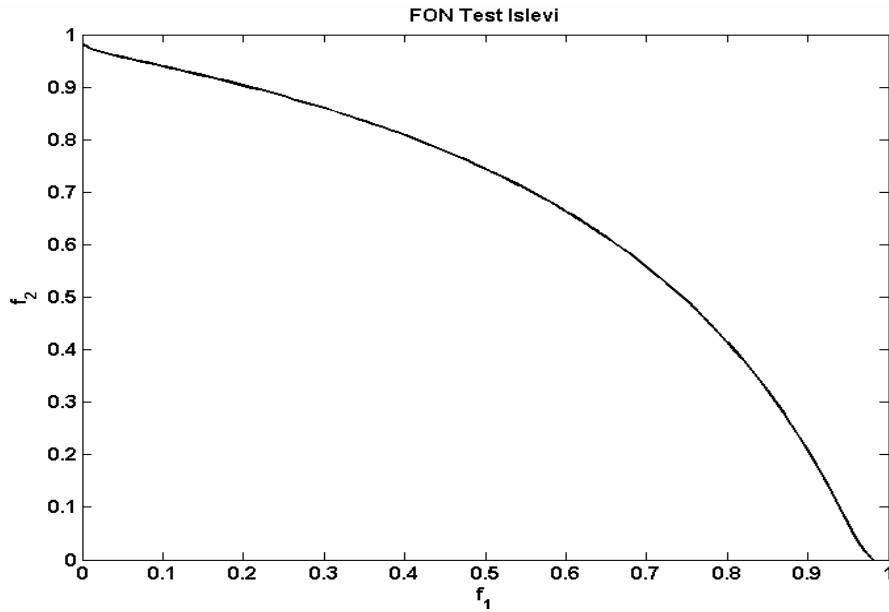
$$f_1(x) = 1 - \exp\left(1 - \sum_{i=1}^n \left(x_i - \frac{1}{\sqrt{n}}\right)^2\right) \quad (3.20a)$$

$$f_2(x) = 1 - \exp\left(1 - \sum_{i=1}^n \left(x_i + \frac{1}{\sqrt{n}}\right)^2\right) \quad (3.20b)$$

$$-4 \leq x_i \leq 4 \quad i = 1, 2, \dots, n \quad (3.20c)$$

Bu test işlevinin Pareto-optimal cephesi $x_i \in \left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right]$ aralığındadır.

Şekil 3.22' de 10 parametre için, FON test işlevinin grafiği gösterilmektedir. Pareto-optimal cephe dışbükeydir. Bu yüzden bazı algoritmalar (ağırlıklı toplam yaklaşımı gibi) Pareto-optimal bireylerin bu cepheye düzgün dağılmasını sağlayamayabilirler.



Şekil 3.22. FON test işlevi

3.3.4. POL Test İşlevi

Poloni tarafından önerilmiş, iki değişkenli ve iki amaçlı bir test işlevidir [Poloni ve ark., 2000]. Bir en küçükleme problemidir ve matematiksel ifadesi aşağıdaki eşitlikte verilmektedir:

$$f_1(x) = \left[1 + (A_1 - B_1)^2 + (A_2 - B_2)^2 \right] \quad (3.21a)$$

$$f_2(x) = \left[(x_1 + 3)^2 + (x_2 + 1)^2 \right] \quad (3.21b)$$

$$A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2 \quad (3.21c)$$

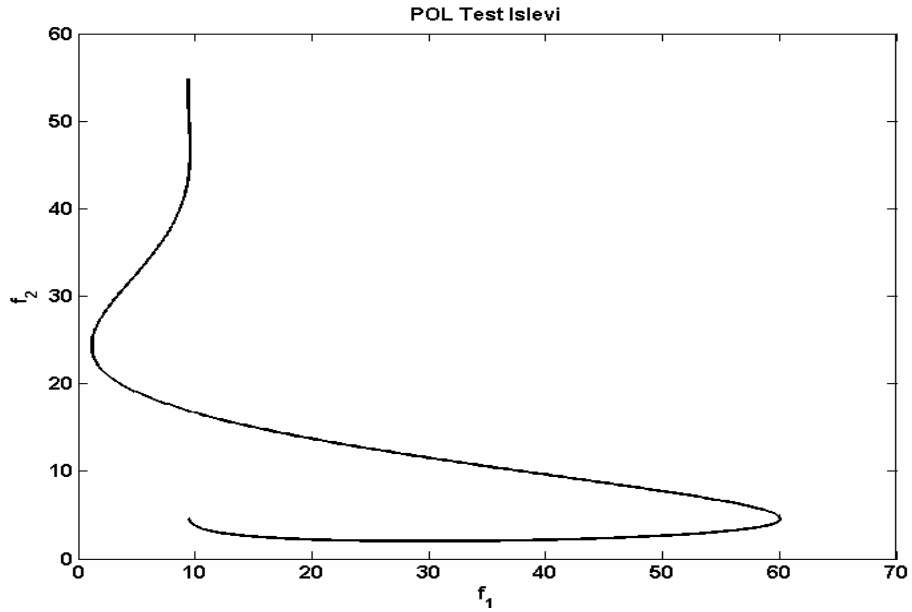
$$A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2 \quad (3.21d)$$

$$B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2 \quad (3.21e)$$

$$B_2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2 \quad (3.21f)$$

$$-\pi \leq (x_1, x_2) \leq \pi \quad (3.21g)$$

Bu test işlevi, Şekil 3.23’ te gösterildiği gibi, içbükey ve ayrık bir Pareto-optimal cepheye sahiptir. Diğer ayrık test problemlerinde olduğu gibi, çoğu çok amaçlı optimizasyon algoritmasının Pareto-optimal cepheyi bulması zordur.



Şekil 3.23. POL test işlevi

3.3.5. Sık Kullanılan Diğer Test İşlevleri

Deb ve ark. tarafından, sistematik olarak tasarlanmış çeşitli test işlevleri önerilmiştir [Deb ve ark., 1999]. Bu test işlevleri; *i*) Pareto-optimal cepheye yakınsama ve *ii*) topluluk içerisinde çeşitliliğin sağlanması açısından çok amaçlı genetik algoritma yöntemleri için zorluklara (sorunlara) yol açar. İlk durum için, çok modluluk, aldatıcılık ve yalıtılmış en iyi, en çok bilinen sorunlardır. İkinci durum, düzgün dağılmış bir Pareto cephesi elde edilmesi için önemlidir. Bununla beraber, Pareto-optimal cephenin karakteristikleri bir GA' nın düzgün olarak dağılmış Pareto-optimal bireyler bulmasını zorlaştırabilir: dışbükeylik veya içbükeylik, ayrıklık ve düzgün olmayan dağılım. Deb ve ark.' larının önerdiği yolun ışığı altında 6 adet test işlevi oluşturulmuş ve bunlar ÇAGA yöntemlerinin karşılaştırılması için kullanılmışlardır [Zitzler ve ark., 2000, Deb, 2002a]. Bu test işlevlerinin hepsi en küçüklenme problemidir. Bu test işlevleri f_1, g, h gibi üç adet işlevden oluşmaktadırlar:

$$\mathcal{T}(\vec{x}) = (f_1(x_1), f_2(\vec{x})) \quad (3.22a)$$

$$f_1(x) = f_1(x_1) \quad (3.22b)$$

$$f_2(\vec{x}) = g(x_2, \dots, x_m) \cdot h(f_1(x), g(x_2, \dots, x_m)) \quad (3.22c)$$

$$\vec{x} = (x_1, \dots, x_m) \quad (3.22d)$$

f_1 işlevi yalnızca ilk parametreye bağlıdır, g işlevi ise geriye kalan $m-1$ parametreye bağlı bir işlevdir. h ise f_1 ve g ' nin işlev değerlerine bağlıdır. Bu test işlevleri; bu üç işleve, parametre sayısına ve bu parametrelerin aldıkları değerlere göre birbirinden ayrılmaktadırlar.

Bu altı adet test işlevi ZDT1, ZDT2, ZDT3, ZDT4, ZDT5 ve ZDT6 olarak adlandırılır. ZDT adı öneren kişilerin soyadlarının baş harfleridir [Zitzler, Deb, Thiele].

ZDT test işlevleri birçok ÇAGA karşılaştırmasında sık olarak kullanılmıştır ve kullanılmaya devam edilmektedirler [Zhang ve ark., 2006; Chen ve Lu, 2008; Gao, 2009; Wang ve ark., 2009]. Bu tezde de bu test işlevlerinden dört tanesi yeni yöntemlerin karşılaştırılmasında kullanılmak üzere seçilmiştir.

3.3.5.1. ZDT1 Test İşlevi

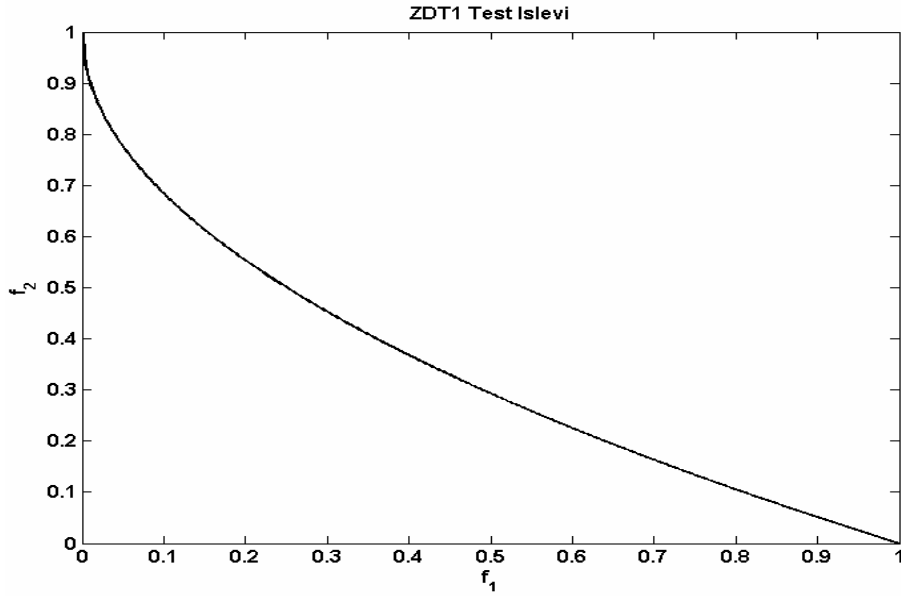
ZDT1 test işlevi dışbükey bir Pareto-optimal cepheye sahiptir:

$$f_1(x_1) = x_1 \quad (3.23a)$$

$$g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1) \quad (3.23b)$$

$$f_2(f_1, g) = 1 - \sqrt{f_1/g} \quad (3.23c)$$

Burada $m=30$ ve $x_i \in [0,1]$ ' dir. Pareto-optimal cephe $g(\vec{x})=1$ alınarak oluşturulur. Bu test işlevinin Pareto-optimal cephesi Şekil 3.24' te verilmektedir.



Şekil 3.24. ZDT1 test işlevinin Pareto-optimal cephesi

3.3.5.2. ZDT2 Test İşlevi

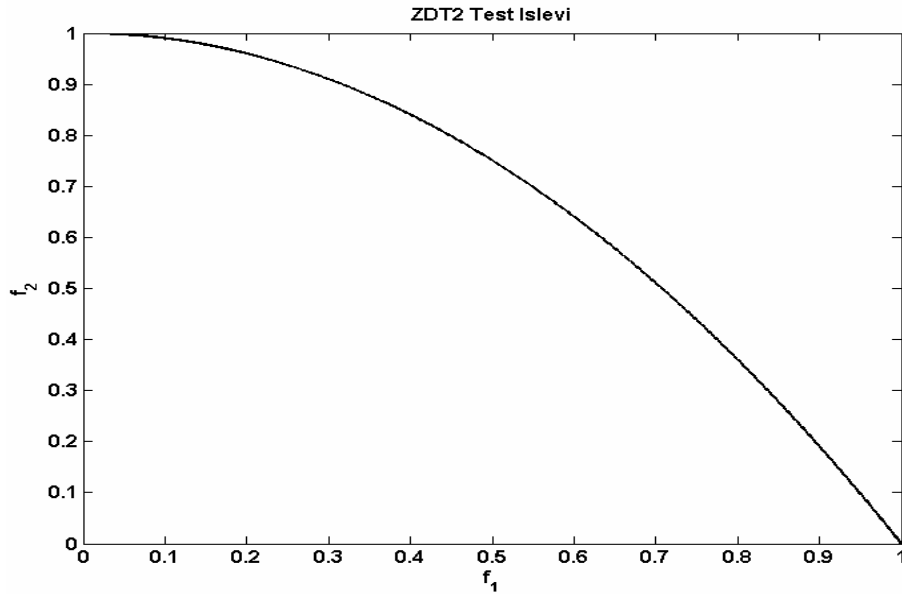
Bu test işlevi, ZDT1 test işlevinin içbükey bir biçimidir:

$$f_1(x_1) = x_1 \quad (3.24a)$$

$$g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1) \quad (3.24b)$$

$$f_2(f_1, g) = 1 - (f_1/g)^2 \quad (3.24c)$$

Burada $m=30$ ve $x_i \in [0,1]$ ' dir. Pareto-optimal cephe $g(\vec{x})=1$ alınarak oluşturulur. Bu test işlevinin Pareto-optimal cephesi Şekil 3.25' de verilmektedir.



Şekil 3.25. ZDT2 test işlevinin Pareto-optimal cephesi

3.3.5.3. ZDT3 Test İşlevi

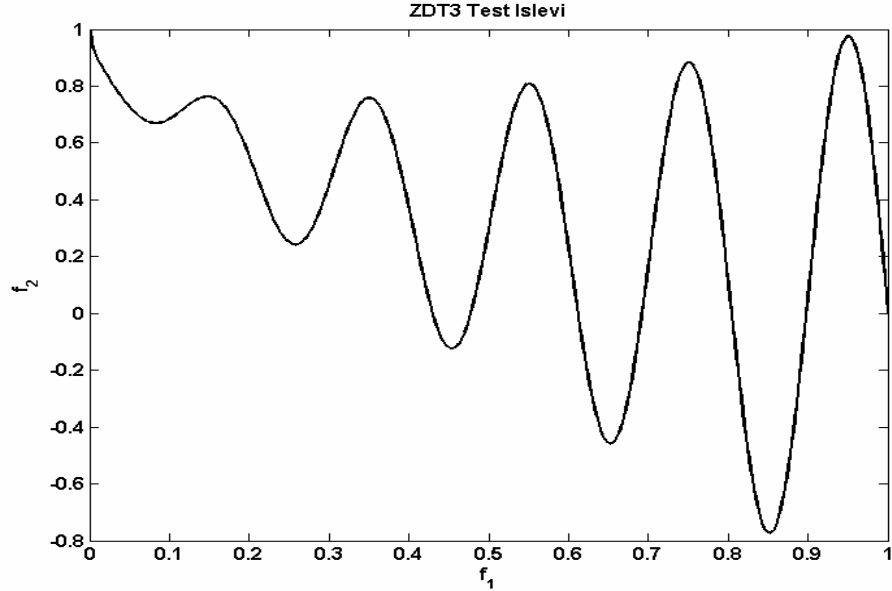
Bu test işlevi ayrık bir işlev özelliği gösterir, Pareto-optimal cephesi birkaç adet sürekli olmayan (ayrık) dışbükey parçadan oluşur:

$$f_1(x_1) = x_1 \quad (3.25a)$$

$$g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1) \quad (3.25b)$$

$$f_2(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1) \quad (3.25c)$$

Burada $m=30$ ve $x_i \in [0,1]$ ' dir. Pareto-optimal cephe $g(\vec{x})=1$ alınarak oluşturulur. f_2 ifadesindeki sinüs işlevi Pareto-optimal cephede ayırlığa neden olmaktadır. Bununla birlikte, parametre uzayında hiçbir ayırıklık yoktur. Bu test işlevinin grafiği Şekil 3.26' da verilmektedir.



Şekil 3.26. ZDT3 test işlevi

3.3.5.4. ZDT4 Test İşlevi

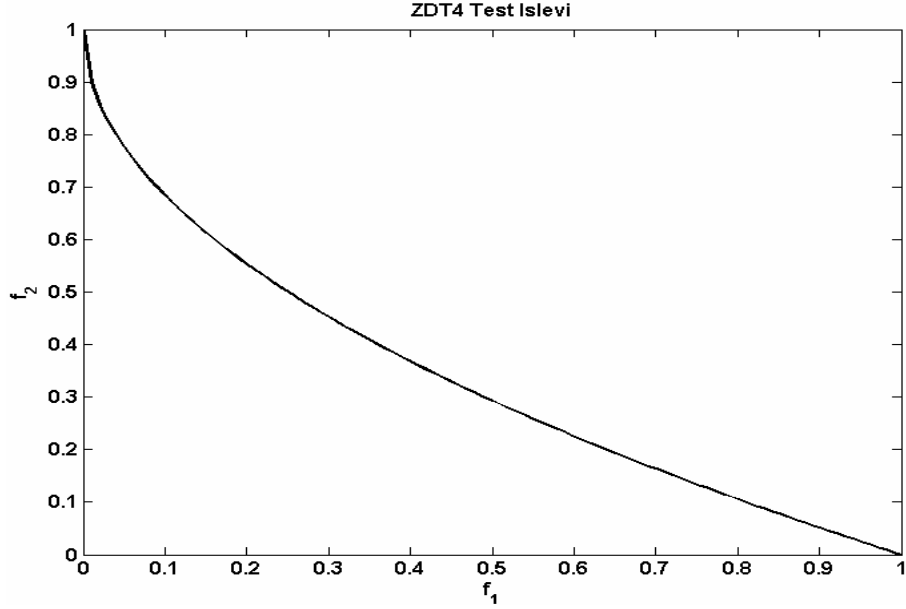
Bu test işlevi, 21^9 adet yerel Pareto-optimal cepheden oluşur ve bu yüzden ÇAGA' ların çok modluluğa olan tepkisinin ölçülebilmesi için kullanılır:

$$f_1(x_1) = x_1 \quad (3.26a)$$

$$g(x_2, \dots, x_m) = 1 + 10(m-1) + \sum_{i=2}^m (x_i^2 - 10 \cos(4\pi x_i)) \quad (3.26b)$$

$$f_2(f_1, g) = 1 - \sqrt{f_1/g} \quad (3.26c)$$

Burada $m=10$, $x_i \in [0,1]$ ve $x_2, \dots, x_m \in [-5,5]$ ’tir. Küresel Pareto-optimal cephe $g(\vec{x})=1$ alınarak oluşturulur. En iyi yerel Pareto-optimal cephe ise $g(\vec{x})=1.25$ alınarak oluşturulur. Şunu belirtmek gerekir ki; yerel Pareto-optimal cephelerin tamamı amaç uzayında ayırt edilemez. Bu test işlevinin Pareto-optimal cephesi Şekil 3.27’de verilmektedir.



Şekil 3.27. ZDT4 test işlevinin Pareto-optimal cephesi

3.3.5.5. ZDT5 Test İşlevi

Bu test işlevi ÇAGA’ları aldatıcı bir problem oluşturur ve diğer test işlevlerinden parametrelerinin ikili sayı dizisi olması nedeniyle ayrılır:

$$f_1(x_1) = 1 + u(x_1) \quad (3.27a)$$

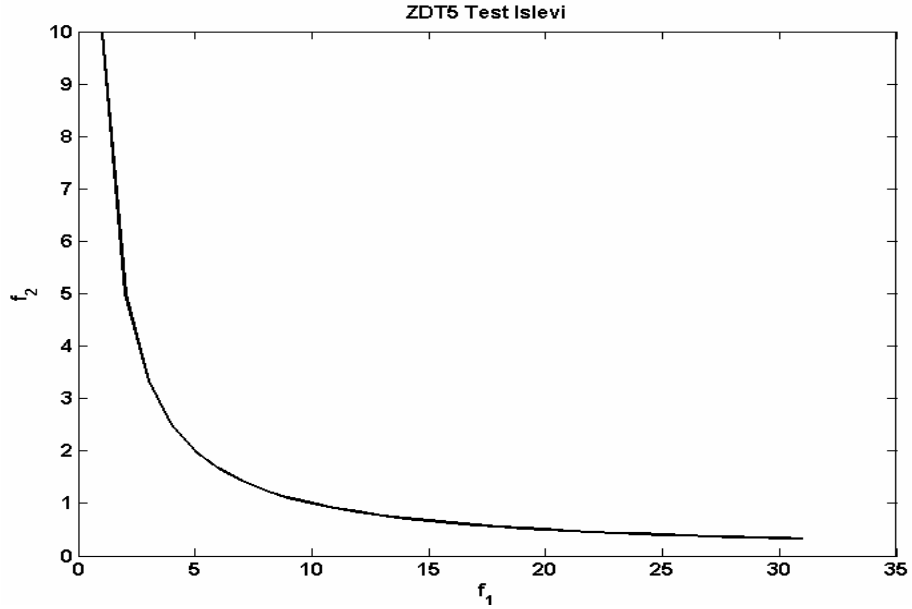
$$g(x_2, \dots, x_m) = \sum_{i=2}^m v(u(x_i)) \quad (3.27b)$$

$$f_2(f_1, g) = 1/f_1 \quad (3.27c)$$

Burada $u(x_i)$, x_i bit dizisindeki 1'lerin sayısını verir.

$$v(u(x_i)) = \begin{cases} 2 + u(x_i) & \text{eger } u(x_i) < 5 \\ 1 & \text{eger } u(x_i) = 5 \end{cases} \quad (3.28)$$

Burada $m=11$, $x_1 \in \{0,1\}^{30}$ ve $x_2, \dots, x_m \in \{0,1\}^5$ 'tir. Küresel Pareto-optimal cephe $g(\vec{x})=10$ alınarak oluşturulur. $g(\vec{x})=11$ alınarak ise en iyi aldatıcı Pareto-optimal cephe görülebilir. Küresel Pareto-optimal cephe gibi yerel cepheler de dışbükeydir. Bu test işlevinin Pareto-optimal cephesi Şekil 3.28' de verilmektedir.



Şekil 3.28. ZDT5 test işlevinin Pareto-optimal cephesi

3.3.5.6. ZDT6 Test İşlevi

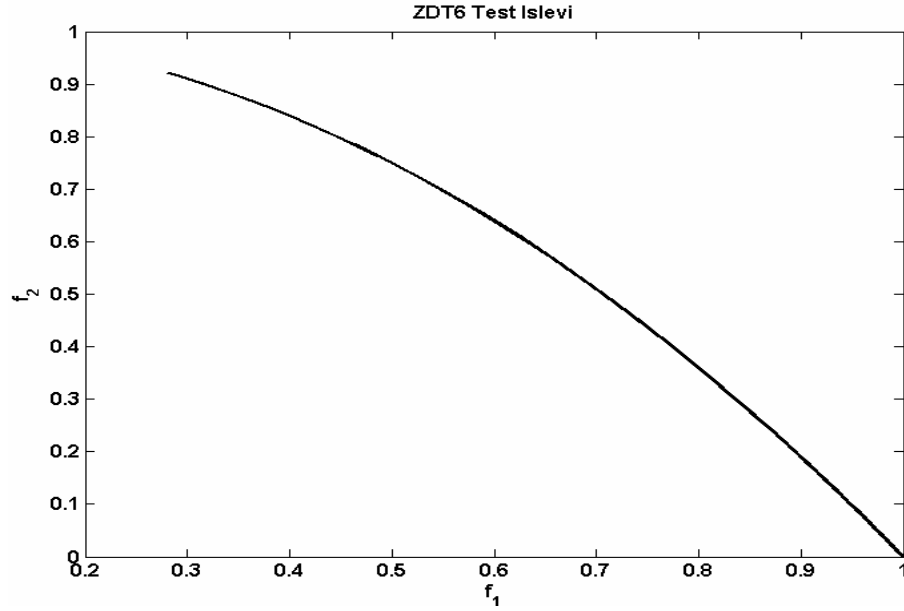
Bu test işlevi düzgün olmayan dağılıma neden olan iki zorluk içermektedir: ilk olarak, Pareto-optimal bireyler küresel Pareto-optimal cephe üzerinde düzgün bir biçimde dağılmamışlardır; ikinci olarak, Pareto-optimal cephe yakınlarında bireylerin yoğunluğu en azdır, uzakta ise en fazladır:

$$f_1(x_1) = 1 - \exp(-4x_1) \sin^6(6\pi x_1) \quad (3.29a)$$

$$g(x_2, \dots, x_m) = 1 + 9 \cdot \left(\left(\sum_{i=2}^m x_i \right) / (m-1) \right)^{0.25} \quad (3.29b)$$

$$f_2(f_1, g) = 1 - (f_1 / g)^2 \quad (3.29c)$$

Burada $m=10$, $x_i \in [0,1]$ ’tir. Pareto-optimal cephe $g(\vec{x}) = 1$ alınarak oluşturulur ve içbükeydir. Bu test işlevinin grafiği Şekil 3.29’da verilmektedir.



Şekil 3.29. ZDT6 test işlevi

Literatürde ÇAGA yöntemlerinin karşılaştırılması için birçok test işlevi önerilmiştir. Huband ve ark., bu test işlevlerini derleyerek, bir test işlevi araç kutusu haline getirmişlerdir. Ayrıca, bu test işlevlerinin özelliklerini ve problem tiplerini de belirtmişlerdir. Çizelge 3.1’ de, literatürde sık kullanılan bazı kısıtlamasız test işlevleri, parametre sınırları ve problem tipleri verilmektedir.

Çizelge 3.1. ÇAGA test işlevleri (m parametre sayısı)

Test İşlevi	m	Parametre Sınırları	Amaç İşlevleri	Pareto Cephesi
ZDT1	30	[0,1]	$f_1(X) = x_1$ $f_2(X) = g(X)[1 - \sqrt{x_1 / g(X)}]$ $g(X) = 1 + 9 \left(\sum_{i=2}^m x_i \right) / (m - 1)$	Dışbükey
ZDT2	30	[0,1]	$f_1(X) = x_1$ $f_2(X) = g(X)[1 - (x_1 / g(X))^2]$ $g(X) = 1 + 9 \left(\sum_{i=2}^m x_i \right) / (m - 1)$	İçbükey
ZDT3	30	[0,1]	$f_1(X) = x_1$ $f_2(X) = g(X)[1 - \sqrt{x_1 / g(X)} - \frac{x_1}{g(X)} \sin(10\pi x_1)]$ $g(X) = 1 + 9 \left(\sum_{i=2}^m x_i \right) / (m - 1)$	Dışbükey ve Süreksiz
ZDT4	10	$x_1 \in [0,1]$ $x_2, \dots, x_m \in [-5,5]$	$f_1(X) = x_1$ $f_2(X) = g(X)[1 - \sqrt{x_1 / g(X)}]$ $g(X) = 1 + 10(m - 1) + \sum_{i=2}^m (x_i^2 - 10 \cos(4\pi x_i))$	Dışbükey
ZDT6	10	[0,1]	$f_1(X) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(X) = g(X)[1 - (f_1(X) / g(X))^2]$ $g(X) = 1 + 9 \left[\left(\sum_{i=2}^m x_i \right) / (m - 1) \right]^{0.25}$	İçbükey ve Düzgün Dağıtılmış
MOP1 (SCH1)	1	$[-10^5, 10^5]$	$f_1 = x^2$ $f_2 = (x - 2)^2$	Dışbükey
MOP2	-	[-4,4]	$f_1(x_1, \dots, x_n) = 1 - \exp(-\sum_{i=1}^n (x_i - 1/\sqrt{n})^2)$ $f_2(x_1, \dots, x_n) = 1 - \exp(-\sum_{i=1}^n (x_i - 1/\sqrt{n})^2)$	İçbükey

Çizelge 3.1. (devamı) ÇAGA test işlevleri(m parametre sayısı)

MOP3	2	$[-\pi, \pi]$	$Maks.f_1(x_1, x_2) = -1 - (A_1 - B_1)^2 - (A_2 - B_2)^2$ $Maks.f_2(x_1, x_2) = -(x_1 + 3)^2 - (x_2 + 1)^2$ $A_1 = 0.5 \sin 1 - 2 \cos 1 + \sin 2 - 1.5 \cos 2$ $A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2$ $B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2$ $B_2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2$	Süreksiz
MOP4	3	$[-5, 5]$	$f_1(x_1, x_2, x_3) = \sum_{i=1}^2 -10 \exp^{-0.2 \sqrt{x_i^2 + x_{i+1}^2}}$ $f_2(x_1, x_2, x_3) = \sum_{i=1}^3 x_i ^{0.8} + 5 \sin(x_i^3)$	Süreksiz
MOP5	2	$[-30, 30]$	$f_1(x_1, x_2) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$ $f_2(x_1, x_2) = \frac{(3x_1 - 2x_2 + 4)^2}{8} + \frac{(x_1 - x_2 + 1)^2}{27} + 15$ $f_3(x_1, x_2) = \frac{1}{x_1^2 + x_2^2 + 1} - 1.1 \exp(-x_1^2 - x_2^2)$	Süreksiz
MOP6	2	$[0, 1]$	$f_1(x_1) = x_1$ $f_2(x_1, x_2) = (1 + 10x_2) \left(1 - \left(\frac{x_1}{1 + 10x_2} \right)^2 - \frac{x_1}{1 + 10x_2} \sin(8\pi x_1) \right)$	Süreksiz
MOP7	2	$[-400, 400]$	$f_1(x_1, x_2) = \frac{(x_1 - 2)^2}{2} + \frac{(x_2 + 1)^2}{13} + 3$ $f_2(x_1, x_2) = \frac{(x_1 + x_2 - 3)^2}{36} + \frac{(-x_1 + x_2 + 2)^2}{8} - 17$ $f_3(x_1, x_2) = \frac{(x_1 + 2x_2 - 1)^2}{175} + \frac{(-x_1 + 2x_2)^2}{17} - 13$	-
DTLZ1	-	$[0, 1]$	$f_1 = (1 + g) \prod_{i=1}^{M-1} \cos(y_i \pi / 2)$ $f_{m=2:M-1} = (1 + g) 0.5 \left(\prod_{i=1}^{M-m} y_i \right) (1 - y_{M-m+1})$ $f_M = (1 + g) 0.5 (1 - y_1)$ $g = 100 \left[k + \sum_{i=1}^k \left((z_i - 0.5)^2 - \cos(20\pi(z_i - 0.5)) \right) \right]$	Doğrusal
DTLZ2	-	$[0, 1]$	$f_1 = (1 + g) 0.5 \prod_{i=1}^{M-1} y_i$ $f_{m=2:M-1} = (1 + g) \left(\prod_{i=1}^{M-m} \cos(y_i \pi / 2) \right) \sin(y_{M-m+1} \pi / 2)$ $f_M = (1 + g) \sin(y_1 \pi / 2)$ $g = \sum_{i=1}^k \left((z_i - 0.5)^2 \right)$	İçbükey
DTLZ3	-	$[0, 1]$	DTLZ2 ile aynı, yalnızca g ifadesi DTLZ1' deki ile değiştirilir.	İçbükey

Çizelge 3.1. (devamı) ÇAGA test işlevleri (m parametre sayısı)

DTLZ4	-	[0,1]	DTLZ2 ile aynı, yalnızca, $y_i \in y$ ifadesi y_i^α ile değiştirilir ($\alpha > 0$ 'dır)	İçbükey
DTLZ5	-	[0,1]	DTLZ2 ile aynı, yalnızca $y_2, \dots, y_{M-1} \in y$ ifadesi $\frac{1+2gy_i}{2(1+g)}$ ifadesi ile değiştirilir.	-
DTLZ6	-	[0,1]	DTLZ5 ile aynı, yalnızca g ifadesi $g = \sum_{i=1}^k z_i^{0.1}$ ile değiştirilir.	-
DTLZ7	-	[0,1]	$f_{m=1:M-1} = y_m$ $f_M = (1+g) \left(M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right] \right)$ $g = 1 + 9 \sum_{i=1}^k \frac{z_i}{k}$	Süreksiz

3.4. Çok Amaçlı Genetik Algoritma Başarım Ölçütleri

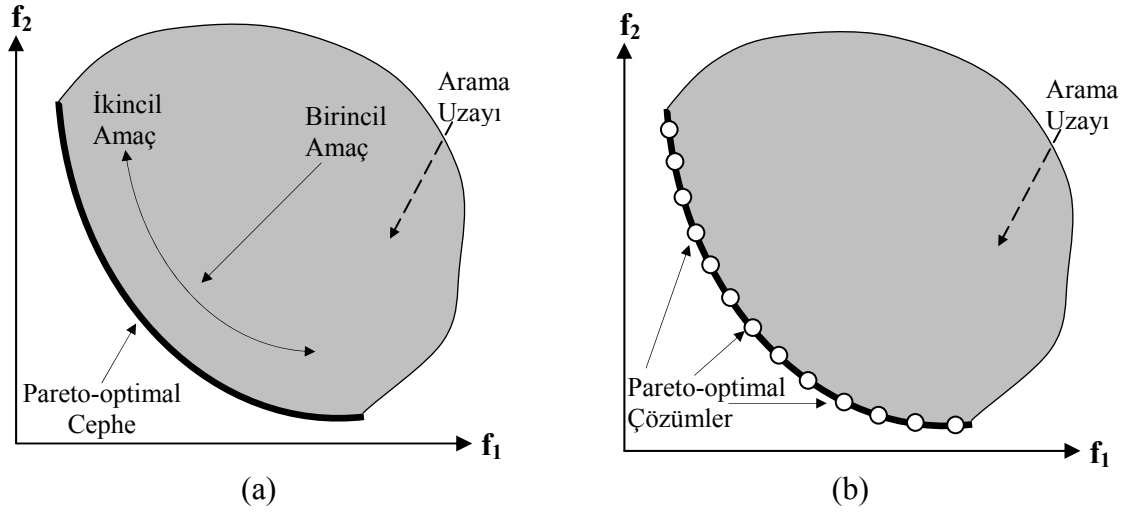
Bir optimizasyon problemini çözmek için yeni bir yöntem önerildiğinde, önerilen yöntemin çalıştığını gösteren görsel bir tanımlama yeterlidir. Bu tür yöntemlerin işleyişinin okuyucunun zihninde, bir resminin çizilebilmesi önemlidir. Bununla birlikte, yöntem popüler olduğunda ve birçok farklı uygulaması yapıldığında, başarımının karşılaştırılabilmesi için çeşitli test işlevlerine verdiği tepkiler belirlenmelidir.

Çok önceleri, ÇAGA'ların amaç uzayında buldukları bastırılmayan bireyler ile gerçek Pareto-optimal bireylerin birlikte gösterilerek karşılaştırma yapılmaktaydı. Bu çalışmalarda, bulunan sonuçların gerçek Pareto-optimal sonuçlara ne kadar yaklaştığı vurgulanırdı. Farklı ÇAGA yöntemlerinin önerilmesiyle birlikte, bu yöntemlerin çeşitli test işlevleri üzerinde başarımlarının karşılaştırılması gerekli olmuştur. Bir karşılaştırma çalışması yapmadan önce, uygun bir test işlevinin seçilmesine ihtiyaç vardır. Test işlevlerinin Pareto-optimal cephelerinin yerinin bilinmesi gereklidir (hem amaç hem de parametre uzayında).

Çok amaçlı optimizasyonun iki temel amacı vardır: (i) Pareto-optimal cepheye (önceden biliniyorsa) mümkün olduğunca yakın çözümler bulmak (*yakınsama*), (ii) bulunan bastırılmayan bireylerin Pareto cephesinde mümkün olduğunca düzgün dağılması (*dağılım veya çeşitlilik*). İlk amaç, Pareto-optimal bölgeye doğru bir aramaya

ihtiyaç duyarken; ikinci amaç, Pareto-optimal cephe boyunca bir aramaya ihtiyaç duyar (Bkz. Şekil 3.30a).

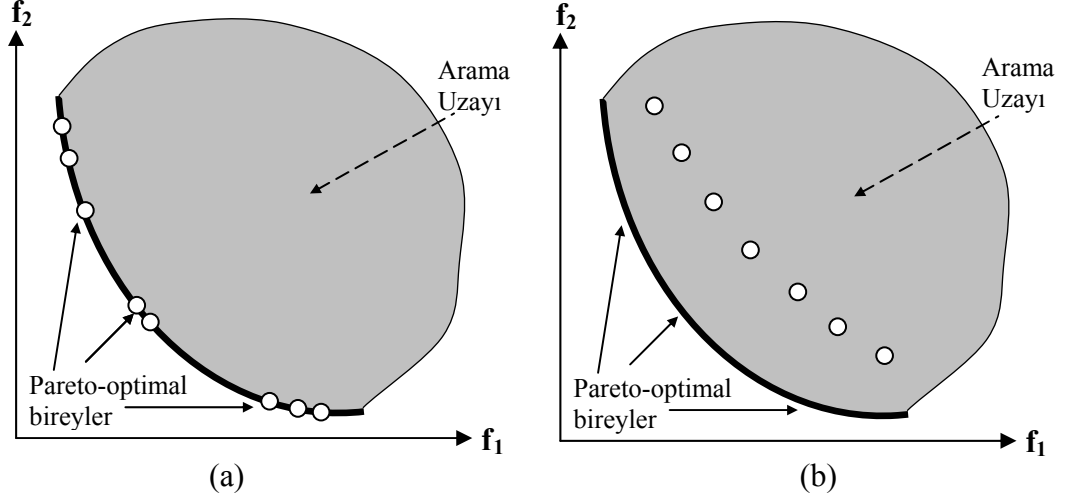
Çeşitliliği fazla olan bir küme, tüm Pareto-optimal cepheye düzgün olarak dağılmış bireylerden (çözümlerden) oluşur. Çeşitlilik ölçütü, aynı zamanda iki farklı ölçüte ayrılabilir: en uç bireylerin yayılımı ve dağılım (bireyler arasındaki bağıl mesafe) [Zitzler ve ark., 2000].



Şekil 3.30. (a) Çok amaçlı optimizasyonda ulaşılması istenen iki amaç
(b) İdeal bir Pareto-optimal çözüm kümesi

İyi bir ÇAGA yönteminin, gerçek Pareto-optimal cepheye yakın ve bu cepheye düzgün olarak dağılmış çözümler bulması gerekmektedir. Şekil 3.30b' de örnek bir problem üzerinde ideal bir ÇAGA' nın başarımı gösterilmektedir. Açıkça görüleceği üzere; bulunan Pareto bireyler Pareto-optimal cephe üzerinde yerleşmekte ve bu cephe üzerinde düzgün olarak dağılmaktadırlar. Bununla birlikte, problemden gelen farklı zorluklar ve seçilen algoritmanın yetersizliğinden dolayı, bu şekilde iyi yakınsamış ve iyi dağılmış Pareto bireylerin bir ÇAGA tarafından bulunması her zaman mümkün değildir. Şekil.31 (a) ve (b) ' de iki farklı algoritma tarafından bulunmuş Pareto-optimal bireyler gösterilmektedir. 1. algoritma Pareto-optimal cepheye çok iyi yaklaşmasına rağmen, bu cephede bireylerin dağılımı kötüdür. Yani bu algoritma, Pareto-optimal bölgenin orta kısımları için hiçbir bilgi sağlayamamaktadır. Diğer taraftan, 2. algoritma bireyleri düzgün bir şekilde dağıtmasına rağmen, bulunan sonuçlar ne yazık ki gerçek

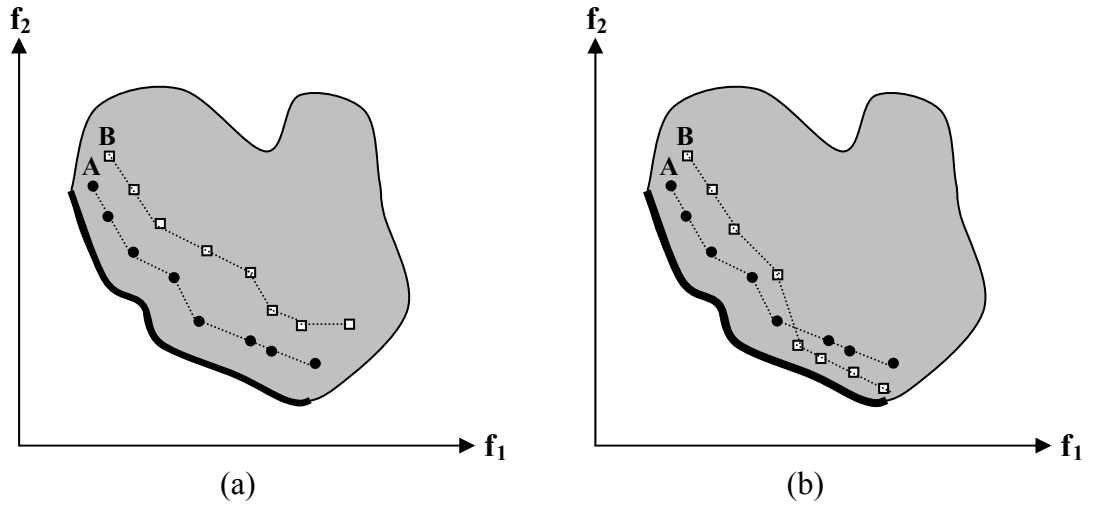
Pareto-optimal cepheye yakın değildir. Pareto-optimal sonuçlar bulunamamıştır. Bu yüzden, bu algoritmanın daha iyi olduğuna dair kesin bir sonuca varmak zordur.



Şekil 3.31. (a) Yakınsama iyi, ancak dağılım kötü (1. Algoritma)
(b) Yakınsama kötü, ancak dağılım iyi (2. Algoritma)

Pareto-optimal cepheye yakınsama ve bu cephe üzerinde düzgün dağılım birbiriyle çatışan iki farklı amaç olduğu için, bir algoritmanın başarımına karar verebilecek tek bir ölçüt yoktur. 1. algoritma birinci amaç için iyiysen, 2. algoritma ikinci amaç için iyidir. Eğer, Pareto-optimal cepheye yakınsama ile ilgili bir ölçüt tanımlarsak ve bir diğer ölçütü de düzgün dağılım için tanımlarsak, iki algoritma birbirini bastıramaz, yani birbirlerine karşı bir üstünlükleri yoktur. Dolayısıyla, bu iki amacı sağlamak için en azından iki ölçüte ihtiyaç vardır.

Şekil 3.32a' da B algoritması tarafından bulunan Pareto bireylere göre, A algoritması tarafından bulunan Pareto bireyler Pareto-optimal cepheye daha yakındır. Bu durumda, A algoritmasının B algoritmasından daha iyi olduğu açıkça görülmektedir. Bununla birlikte, Şekil 3.32b' de, A algoritmasının bulduğu bireylerden bazıları, B algoritmasının bulduğu bireylerden bazılarını bastırmaktadır ve bunun tersi de söz konusudur. Bu durum, çok amaçlı optimizasyon için bir başarımlı ölçütü tasarlarlarken dikkat edilmesi gereken bir zorluktur. Burada, iki algoritma da, yakınsama ve çeşitlilik açısından birbirlerine benzerdirler. Bu tür bir karşılaştırmayı yapabilmek için, ölçütlerin çok iyi tanımlanması gereklidir.



Şekil 3.32. (a) A algoritması B algoritmasından daha iyi
(b) A ve B algoritmalarını karşılaştırmak zor

Yukarıdaki tartışmalara göre, iki veya daha fazla algoritmayı karşılaştırırken, en azından iki başarımlı ölçütünün kullanılması gerektiği ve başarımlı ölçütlerinin tanımlarının çok iyi yapılması gerektiği anlaşılmaktadır. Aşağıda literatürde kullanılan başarımlı ölçütlerinin bazıları üç farklı kategoride incelenmektedir: ilk bölümde Pareto-optimal cepheye olan yakınsamayı bulmak için kullanılan ölçütler, ikinci bölümde çeşitliliği veya dağılımı ölçmek için kullanılan ölçütler ve üçüncü bölümde her iki amacı da ölçmek için kullanılan ölçütler verilmektedir. Literatürde çok sayıda başarımlı ölçütü önerilmiş ve önerilmeye devam edilmektedir. Tezde yer almayan diğer başarımlı ölçütleri ile ilgili ayrıntılı bilgilere (Deb, 2001; Coello Coello ve ark., 2007) kaynaklarından ulaşılabilir.

3.4.1. Yakınsama İçin Kullanılan Başarımlı Ölçütleri

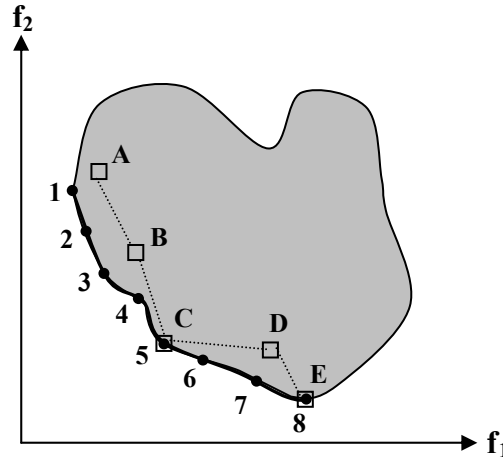
Bu ölçütler, önceden bilinen bir Pareto-optimal küme (P^*), ÇAGA yönteminin bulduğu N adet bireyden oluşan bir Q kümesinin yakınsamasını ölçmek için kullanılır. Farklı boyutlardaki bu iki farklı küme arasındaki yakınsamayı bulmak için, çok sayıda ölçüt tanımlanabilir. Aşağıda bu amaç için, literatürde yer alan ve kullanılan ölçütlerden bazıları verilmektedir. Bu ölçütler, eğer P^* kümesi çok elemanlı seçilirse (örneğin 500 bireylik), yakınsama için iyi bir tahmin sağlayabilmektedirler.

3.4.1.1. Hata Oranı (Error Ratio, ER)

Bu ölçüt, basitçe Q kümesinde yer alan çözümlerden (bireylerden) kaç tanesinin P^* Pareto-optimal kümesinin üyesi olmadığı sayılmasıyla hesaplanır [Veldhuzien, 1999] ve matematiksel olarak şöyle hesaplanır;

$$ER = \frac{\sum_{i=1}^{|Q|} e_i}{|Q|} \quad (3.30)$$

Burada eğer i bireyi P^* kümesinin elemanı değilse $e_i=1$, aksi takdirde ise $e_i=0$ olur. Şekil 3.33' te P^* ve Q kümeleri gösterilmektedir. Buna göre Q kümesinde, Pareto-optimal kümeye ait olmayan 3 birey (A, B, D) ve toplamda 5 birey olduğu için, hata oranı $ER=3/5=0.6$ olarak hesaplanır. (3.30) eşitliğine göre eğer ER değeri küçük bir değerse, bunun anlamı Pareto-optimal cepheye iyi bir yakınsama söz konusu olduğudur. ER ölçütü, 0 ile 1 arasında değerler alır. $ER=0$ ise, tüm bireyler Pareto-optimal cepheye aittir ve en iyi yakınsama elde edilmiştir, $ER=1$ ise hiçbir birey Pareto-optimal cepheye ait değildir ve yakınsama en kötüdür ya da yoktur.



Şekil 3.33. *Kareler:* Pareto-optimal cepheyi oluşturan bireyler (P^*) ve *Daireler:* ÇAGA yöntemi tarafından bulunan Pareto bireyler (Q)

Burada şunu da belirtmek gerekir ki; eğer Q ' nun bir elemanı Pareto-optimal olup eğer P^* kümesinde yer almıyorsa, bu durumda (3.30) eşitliğinde Pareto olmayan bir çözüm olarak düşünülmelidir. Bu yüzden, (3.30) eşitliğinde fazla sayıda bireyden oluşan bir P^* kümesi kullanılmalıdır. Bu ölçütün bir başka eksikliği de, Q ' nun hiçbir

elemanının Pareto-optimal kümeye üye olmaması durumudur. Bu eksikliklerinden dolayı, bu ölçüt çok fazla kullanılmamaktadır.

3.4.1.2. Küme Kapsama Ölçütü (Set Coverage Metric, C)

Bu ölçüt, Zitzler tarafından 1999 yılında sunulmuştur [Zitzler, 1999; Zitzler ve Thiele, 1999]. Bu ölçüt, A ve B gibi iki çözüm kümesindeki bireylerin bağıl dağılımı hakkında bir fikir elde etmek için kullanılır. Küme kapsama ölçütü $C(A,B)$, B kümesinde yer olan bireylerden kaç tanesinin A kümesinde yer alan bireyler tarafından bastırıldığının oranına bakılarak hesaplanır:

$$C(A,B) = \frac{|\{b \in B \mid \exists a \in A : a \leq b\}|}{|B|} \quad (3.31)$$

$C(A,B)=1$ ise, B kümesinin tüm elemanları, A tarafından bastırılmaktadır. Diğer taraftan $C(A,B)=0$ ise, B kümesinin hiçbir elemanı A tarafından bastırılmamaktadır. Bastırma işlevi, simetrik bir işlev olmadığı için $C(A,B)$ ölçütü, $1-C(B,A)$ ' ya eşit olmayabilir. Bu yüzden, A 'daki kaç bireyin B tarafından veya B 'deki kaç bireyin A tarafından kapsandığını anlamak için hem $C(A,B)$ hem de $C(B,A)$ ölçütlerinin hesaplanması gereklidir.

Zitzler 1999 yılında, bu ölçütü iki algoritmayı karşılaştırmak için kullanmış olsa da, bu ölçüt aynı zamanda $A=P^*$ ve $B=Q$ kullanılarak, bir algoritmanın başarımının ölçülmesi için de kullanılabilir. $C(P^*,Q)$ ölçütü, Q 'da yer alan bireylerin P^* 'daki bireyler tarafından bastırılmasına bağlı olarak hesaplanabilir. Şekil 3.33' teki P^* ve Q kümeleri için, $C(P^*,Q)=3/5=0.6$ olarak hesaplanır, çünkü üç birey (A,B,D) P^* kümesine ait bireyler tarafından bastırılmaktadırlar. $C(Q,P^*)$ ise sıfır olacaktır, çünkü P^* 'ın hiçbir elemanı Q tarafından bastırılmamaktadır.

3.4.1.3. Nesilsel Mesafe (Generational Distance, GD)

Bu ölçüt, Q kümesinde yer alan bireylerin P^* kümesinde yer alan bireylere olan ortalama mesafesini hesaplar [Deb, 2001; Deb ve ark., 2002b]:

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^p \right)^{1/p}}{|Q|} \quad (3.32)$$

$p=2$ yani iki amaçlı bir problem için, Q ’da yer alan bir i bireyinin, P^* ’da en yakın olduğu bireyle arasındaki Öklit mesafesi şöyle hesaplanır:

$$d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^M \left(f_m^{(i)} - f_m^{*(k)} \right)^2} \quad (3.33)$$

Burada $f_m^{*(k)}$, P^* ’ın k . bireyinin m . amaç işlevinde aldığı değerdir. Şekil 3.33 için, bulunan çözümlerden A bireyi 1 nolu Pareto-optimal çözüme, B bireyi 3 nolu Pareto-optimal çözüme, C bireyi 5 nolu Pareto-optimal çözüme, D bireyi 7 nolu Pareto-optimal çözüme ve E bireyi de 8 nolu Pareto-optimal çözüme en yakın mesafededir. Tüm bireyler için amaç işlevi değerleri Çizelge 3.2’de verilmektedir.

Çizelge 3.2. Örnek uygulama

Birey	f_1	f_2	Birey	f_1	f_2
1	1.0	7.5	8	8.4	1.2
2	1.1	5.5	A	1.2	7.8
3	2.0	5.0	B	2.8	5.1
4	3.0	4.0	C	4.0	2.8
5	4.0	2.8	D	7.0	2.2
6	5.5	2.5	E	8.4	1.2
7	6.8	2.0			

Buna göre bu bireyler arasındaki Öklit mesafeleri şöyle hesaplanabilir:

$$d_{A1} = \sqrt{(1.2 - 1.0)^2 + (7.8 - 7.5)^2} = 0.36 \quad (3.34a)$$

$$d_{B3} = \sqrt{(2.8 - 2.0)^2 + (5.1 - 5.0)^2} = 0.81 \quad (3.34b)$$

$$d_{C5} = \sqrt{(4.0 - 4.0)^2 + (2.8 - 2.8)^2} = 0.00 \quad (3.34c)$$

$$d_{D7} = \sqrt{(7.0 - 6.8)^2 + (2.2 - 2.0)^2} = 0.28 \quad (3.34d)$$

$$d_{E18} = \sqrt{(8.4 - 8.4)^2 + (1.2 - 1.2)^2} = 0.00 \quad (3.34e)$$

(3.32) eşitliğine göre nesil mesafe $GD=0.19$ olarak hesaplanır. Daha küçük GD değerine sahip olan algoritma, yakınsama açısından daha iyidir. Bu ölçütteki zorluk şudur; eğer mesafe değerleri arasında büyük bir dalgalanmaya sahip bir Q kümesi varsa, bu ölçüt gerçek mesafeyi veremeyebilir. Böyle bir durumda, GD ölçütünün varyansının hesaplanması gereklidir. Ayrıca, eğer amaç işlevi değerleri genlik olarak farklılaşıyorsa, mesafe hesabı yapılmadan önce normalize edilmelidirler. Mesafe hesabının güvenilir olması isteniyorsa, P^* kümesinin eleman sayısının büyük seçilmesi (örneğin 500) gereklidir.

Birçok makalede bu ölçüt kullanılmıştır [Zitzler ve Thiele, 1999; Deb, 2002b]. Bazı makalelerde bu ölçüt γ olarak adlandırılmış ve standart sapması da kullanılmıştır. Eğer bu değer standart sapması küçük olarak hesaplanmışsa, γ değeri güvenilir olarak kabul edilmektedir.

3.4.1.4. En Büyük Pareto-Optimal Cephe Hatası (Maximum Pareto-Optimal Front Error, MFE)

Bu ölçüt, Q' nun tüm elemanları arasındaki en kötü d_i mesafesini hesaplar [Veldhuizen, 1999]. Şekil 3.33' teki problem için, en kötü mesafe yukarıdaki hesaplamalardan da görüleceği üzere B bireyi tarafından oluşturulur ve $MFE=0.81$ ' dir. Bu ölçüt yakınsama için basit bir ölçüttür, ancak bireylerin dağılımı için yanlış bir bilgi verebilir.

3.4.2. Dağılım veya Çeşitlilik İçin Kullanılan Başarım Ölçütleri

ÇAGA tarafından bulunan Pareto bireyler arasındaki çeşitliliğin / dağılımın hesaplanması için birçok ölçüt önerilmiştir. Aşağıda bunlardan birkaç tanesi açıklanmaktadır.

3.4.2.1. Aralık (Spacing)

Schott 1995 yılında, bir ÇAGA yöntemi tarafından bulunan Pareto bireyler kümesinde, ardışıl bireyler arasındaki bağıl mesafenin hesaplanması için bir ölçüt önermiştir [Schott, 1995]:

$$S = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} (d_i - \bar{d})^2} \quad (3.35)$$

Burada, $d_i = \min_{k \in Q \wedge k \neq i} \sum_{m=1}^M |f_m^i - f_m^k|$ formülünden hesaplanır. $\bar{d}$ bu ifadenin ortalama değeridir ve şöyle hesaplanır; $\bar{d} = \sum_{i=1}^{|Q|} d_i / |Q|$. Buradaki mesafe hesabı şöyle yapılmaktadır: i . bireyle, Pareto-optimal kümedeki diğer tüm bireyler arasındaki amaç işlevi değerlerinin mutlak değer olarak farklarının en küçük değeridir. Burada şuna dikkat edilmelidir ki, mesafe hesabı, iki birey arasındaki en küçük Öklit mesafesinden farklıdır.

Bu ölçüt, farklı d_i değerleri arasındaki standart sapmaları ölçmektedir. Bireyler birbirlerinden düzgün aralıklarla dağılıyorlarsa, önerilen mesafe ölçütü küçük olacaktır. Bu yüzden, Pareto bireyleri arasında daha küçük bir aralık ölçütü veren yöntem daha iyi bir yöntem olacaktır. Şekil 3.33' te verilen problem için, d_A mesafesinin hesabı şöyle yapılır;

$$d_A = \min((1.6 + 2.7), (2.8 + 5.0), (5.8 + 5.6), (7.2 + 6.6)) = 4.3 \quad (3.36)$$

Benzer olarak, $d_B=3.5$, $d_C=3.5$, $d_D=2.4$ ve $d_E=2.4$ olarak hesaplanır. Bu sonuçlara göre A ve E bireyleri arasında hemen hemen düzgün bir dağılım olduğu anlaşılmaktadır. Bu yüzden, bulunan d_i değerlerinin standart sapması yüksek çıkacaktır. Buna göre, $\bar{d} = 3.22$ ve $S=0.73$ olarak bulunur. Amaç uzayında rasgele yerleşmiş Pareto bireylerden oluşan bir küme için, standart sapma daha küçük çıkacaktır.

Bu ölçüt, bulunan bastırılmayan bireylerin dağılımı hakkında yararlı bilgiler sağlamaktadır. Bir i bireyinin tüm diğer bireylerle olan mesafesinin hesaplanması ve en küçüğünün (d_i) bulunması nedeniyle işlem sayısı fazladır. Bununla beraber, mesafe hesabındaki simetriden yararlanılarak, hesaplamaların yarısından kurtulabilir, ancak yine de işlem sayısı bulunan Pareto bireylerin sayısının karesiyle orantılı olacaktır. Deb, ardışıl bireyler arasındaki d_i mesafesinin her amaç işlevinde ayrı ayrı olarak hesaplanması için, bir yol önermiştir [Deb ve ark., 2002b]. Bu yol şöyledir: ilk olarak, bulunan Pareto-optimal cephe, her amaç işlevindeki genlik değerine göre artan bir sırada sıralanır. Daha sonra, her bir çözüm ya da birey için, her bir amaçta en yakın iki komşusu arasındaki amaç işlevi değerlerinin farkının toplamı alınır. Bu durumda mesafe ölçüm yöntemi yukarıda bahsedilen mesafe ölçümünden daha hızlıdır. Farklı amaç işlevleri eklendiğinde, (3.35) eşitliğini kullanmadan önce amaçları normalize etmek önemlidir. Ayrıca, bu ölçüt, dağılımın boyutunu dikkate almamaktadır. Dağılım düzgün olduğu sürece, S ölçütü küçük bir değer alacaktır.

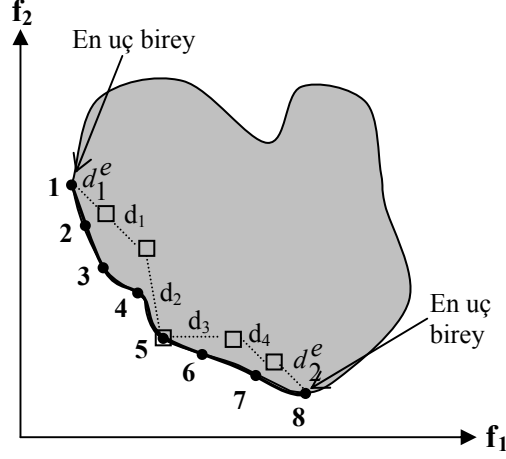
3.4.2.2. Yayılım (Dağılım, Spread)

Deb, yukarıdaki ölçütte belirtilen zorluğu gidermek için yayılım ölçütünü önermiştir [Deb ve ark., 2002b]:

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|Q|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |Q|\bar{d}} \quad (3.37)$$

Burada, d_i komşu bireyler arasındaki mesafeler ve $\bar{d}$ bu mesafelerin ortalama değeridir. d_i mesafeleri hesaplanırken; Öklit mesafe hesabı veya kalabalıklık mesafe

hesabı [Schott, 1995] kullanılabilir. d_m^e parametresi, m . amaç işlevi için, P^* ve Q kümelerinde en uç bireyler arasındaki mesafedir.



Şekil 3.34. Bireyler arasındaki mesafeler

Şekil 3.34' te, iki amaçlı bir problem için, d_1^e ve d_2^e en uç bireylere olan mesafeler ve i . ve $(i+1)$. ardışık bireyler arasındaki Öklit mesafeleri gösterilmektedir. Şekilden de görüleceği üzere, (3.37) eşitliğindeki $|Q|$ terimi $(|Q|-1)$ terimi ile yer değiştirilebilir. Bu ölçüt; $d_j^e = 0$ ve tüm d_i değerleri de ortalama değeri $\bar{d}$ ' ye göre aynı olan ideal bir dağılım için sıfır değerini alacaktır. Buradaki ilk koşulun anlamı; bulunan Pareto bireylerin gerçek Pareto cephesindeki en uç bireylerle aynı olması gerektiğidir. İkinci koşul ise, bireyler arasındaki dağılımın düzgün olması gerektiğini söyler. İdeal küme, herhangi bir çok amaçlı genetik algoritma ile belirlenebilir. Bundan dolayı, bireyler ideal olarak dağılmışsa $\Delta = 0$ olacaktır. Bulunan Pareto bireylerin dağılımı ideal olsun, ancak bireyler bir noktada kümelenmiş olsunlar. Böyle bir dağılımda tüm $|d_i - \bar{d}|$ değerleri sıfır olacaktır, ancak d_m^e değerleri sıfır olmayacaktır. Bu durumda,

$$\Delta = \sum_{m=1}^M d_e^m / \left(\sum_{m=1}^M d_e^m + (|Q|-1)\bar{d} \right) \text{ şekline dönüşecektir. Bu değer de } [0,1]$$

aralığında olacaktır. Payda, Pareto cephesine yaklaşımın boyutunu ölçtüğünden, Δ değeri d_m^e ile orantılı olarak artacaktır. Bu yüzden, bireyler ideal dağılıma ne kadar yakın olarak kümelenirse, Δ değeri de sıfırdan bire doğru artacaktır. Eğer bastırılmayan bireyler düzgün olarak dağılmamışsa, paydaki ikinci terim sıfır

olmayacağı için Δ değeri daha da artacaktır. Bundan dolayı, kötü dağılımlar için Δ değeri 1' den daha büyük olabilir.

Örneğin, Şekil 3.34' te verilen örnek için, Pareto cephesinin en uç sağındaki birey (8 nolu birey), Pareto bireylerin en ucundaki bireyle (E bireyi) aynıdır. Bu yüzden, $d_2^e = 0$ olacaktır. Pareto cephesindeki en uç sol noktadaki bireyle aynı olan bir birey bulunmadığı için $d_1^e = 0.5$ olarak Schott' un mesafe ölçümüyle bulunur. Şekil 3.34' teki d_i değerleri şöyle bulunur: $d_1=4.3$, $d_2=3.5$, $d_3=3.5$, $d_4=2.4$. Buna göre $\bar{d} = 3.43$ olarak hesaplanır. Yayılım ölçütü ise şöyle hesaplanır:

$$\Delta = \frac{0.5 + 0 + |4.3 - 3.43| + |3.5 - 3.43| + |3.5 - 3.43| + |2.4 - 3.43|}{0.5 + 0 + 4 \times 3.43} = 0.18 \quad (3.38)$$

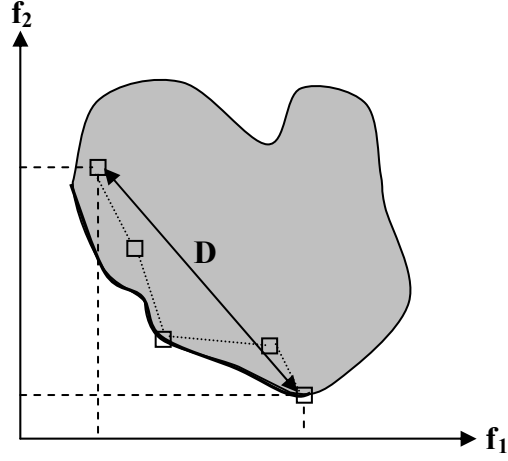
Bulunan bu değer sıfıra yakın olduğu için, dağılım çok kötü değildir. A bireyi, 1 nolu bireyle aynı olsaydı, bu durumda $\Delta = 0.15$ olurdu, yani dağılım şu andakinden daha iyi olabilirdi. Bu yüzden, daha küçük bir Δ değeri bulabilen ÇAGA yöntemi, düzgün dağılım açısından daha iyi olacaktır.

3.4.2.3. En Büyük Yayılım (Maximum Spread)

Zitzler 1999 yılında, bulunan bastırılmayan birey kümesinin en uzak işlev değerlerini kullanarak oluşturulan hiperküpin köşegen uzunluğunu kullanan bir ölçüt önermiştir [Zitzler ve Thiele, 1999]:

$$D = \sqrt{\sum_{m=1}^M \left(\max_{i=1}^{|Q|} f_m^i - \min_{i=1}^{|Q|} f_m^i \right)^2} \quad (3.39)$$

İki amaçlı problemler için, bu ölçüt, amaç uzayında en uzak noktalarda yer alan bireyler arasındaki Öklit mesafesine denk gelmektedir (bkz Şekil 3.35).



Şekil 3.35. En büyük yayılım ölçütü

Bu ölçütün normalize biçimi aşağıdaki gibidir:

$$\bar{D} = \sqrt{\frac{1}{M} \sum_{m=1}^M \left(\frac{\max_{i=1}^{|Q|} f_m^i - \min_{i=1}^{|Q|} f_m^i}{F_m^{\max} - F_m^{\min}} \right)^2} \quad (3.40)$$

Bu formülde, $F_m^{\max}$ ve $F_m^{\min}$ seçilen Pareto-optimal cephenin (P^*) m . amaç işlevinde aldığı en yüksek ve en düşük değerlerdir. Bu yüzden, bu ölçütün değeri 1 ise, geniş bir biçimde dağılmış bir bireyler kümesi bulunur. Bununla beraber, ne $\bar{D}$ ne de D aradaki bireyler için doğru bir dağılım hesaplayamaz.

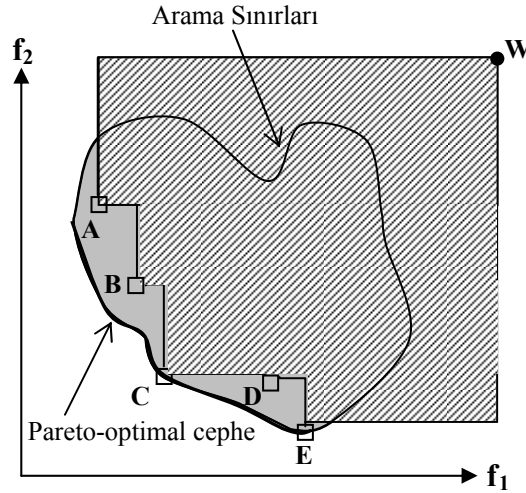
3.4.3. Yakınsamayı ve Dağılımı Birlikte Hesaplayan Başarım Ölçütleri

Hem yakınsamayı hem de düzgün dağılımı hesaplamak için kullanılabilen bazı ölçütler de bulunmaktadır. Böyle bir ölçüt, yakınsama ve dağılım ölçmeyi aynı anda sağlayabilir. Bununla birlikte, daha iyi bir sonuç alınması için yukarıda bahsi geçen ölçütlerden biri ile kullanılması daha iyi olabilir.

3.4.3.1. Hiperküp (Hypervolume)

Bu ölçüt, tüm amaç işlevlerinin en küçük yapılmaya çalışıldığı problemler için, Q' nun elemanları (Şekil 3.36' da taralı alan) tarafından kapsanan hacmi (amaç uzayında) hesaplamak için kullanılır [Zitzler ve Thiele, 1998]. Matematiksel olarak, her $i \in Q$ bireyi için, bir v_i hiperküpü W referans noktası ile oluşturulur ve bireyler de bu hiperküpün köşegenleridir. Referans noktası, en kötü amaç işlevi değerlerinden oluşan bir vektör oluşturulmasıyla kolayca bulunabilir. Bundan sonra, tüm hiperküpelerin birleşimi bulunur ve hiperküp (HV) ölçütü hesaplanır:

$$HV = \text{hacim} \left(\bigcup_{i=1}^{|Q|} v_i \right) \quad (3.41)$$



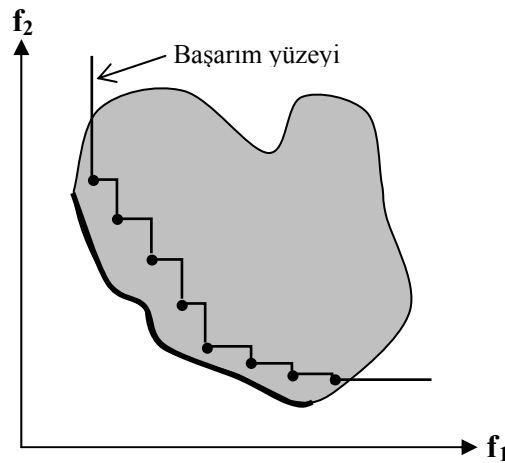
Şekil 3.36. Pareto bireyler tarafından oluşturulan hiperküp

Şekil 3.36, W referans noktasının seçimini göstermektedir. Hiperküp taralı bölge ile gösterilmektedir. Açıkça görülmektedir ki, büyük bir HV değerine sahip olan bir yöntem istenmektedir. Hem HV ölçütü seçilen referans noktasına bağlıdır. Şekil 3.33' te verilen örnek problem için, $W=(11.0,10.0)^T$ olmak üzere hiperküp şöyle hesaplanır:

$$\begin{aligned}
HV &= (11.0 - 8.4) \times (10.0 - 1.2) + (8.4 - 7.0) \times (10.0 - 2.2) \\
&\quad + (7.0 - 4.0) \times (10.0 - 2.8) + (4.0 - 2.8) \times (10.0 - 5.1) \\
&\quad + (2.8 - 1.2) \times (10.0 - 7.8) \\
&= 64.80
\end{aligned} \tag{3.42}$$

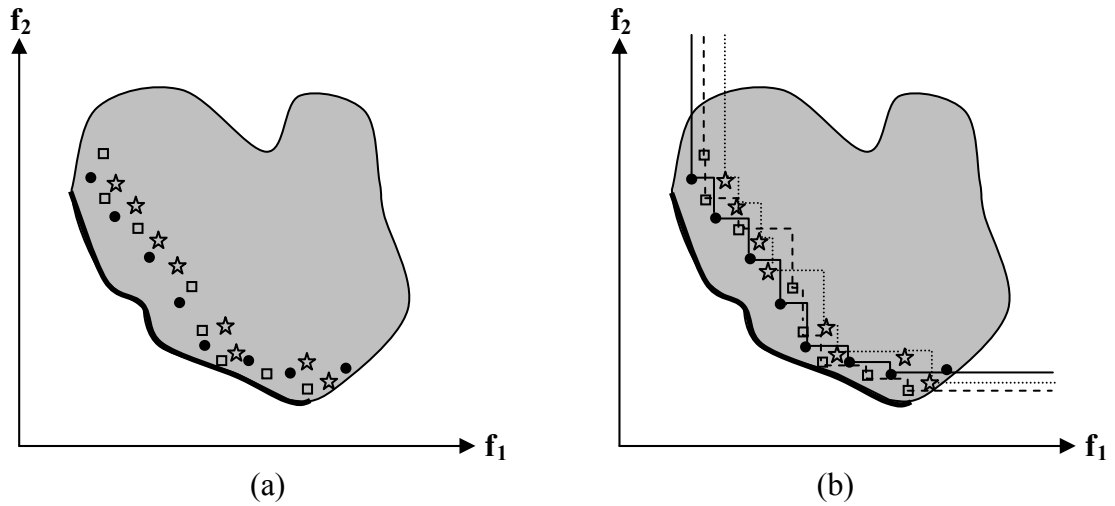
3.4.3.2. Başarım Yüzeyi Temelli İstatistiksel Ölçüt (Attainment Surface Based Statistical Metric)

Fonseca ve Fleming 1996 yılında, çok amaçlı optimizasyon için başarım yüzeyi kavramını önermişlerdir [Fonseca ve Fleming, 1996]. Çoğu çalışmada, bulunan Pareto bireyler genellikle bunları birleştiren bir eğriyle gösterilirler. Böyle bir eğri bir cephenin iyi bir gösterilimini sağladığı halde, cephenin ortalarındaki bireylerin iyi gösterilimini garanti etmez veya ortalarındaki bireylerin Pareto-optimal olduğuna dair bir garanti yoktur. Fonseca ve Fleming bulunan Pareto bireylerin bir eğriyle gösterilmesi yerine, arama uzayında Pareto bireyler tarafından bastırılan tüm bireylerin işaretlendiği bir zarf oluşturmuşlardır. Şekil 3.37, Pareto bireyler tarafından oluşturulmuş örnek bir zarfı göstermektedir. Oluşturulan bu zarf, başarım yüzeyi olarak adlandırılır ve hiperküp (HV) hesaplanması için kullanılan yüzeyi belirler. Hiperküp ölçütünde olduğu gibi, başarım yüzeyi de yakınsama ve dağılımı birlikte belirleyen bir ölçüttür.



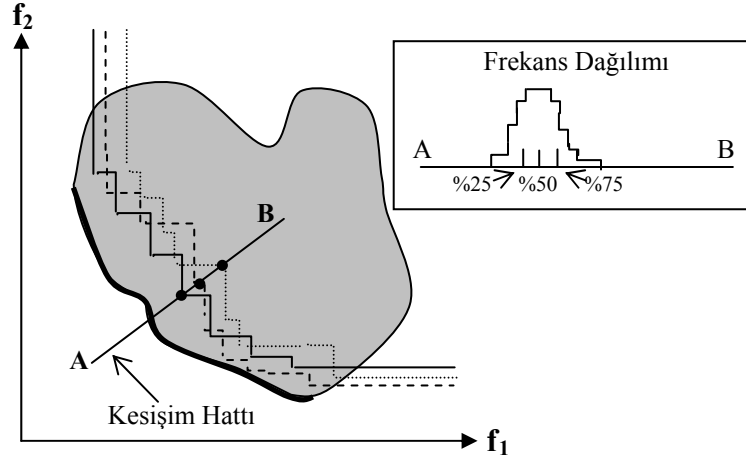
Şekil 3.37. Pareto bireyler tarafından oluşturulan başarım yüzeyi

Pratikte, bir ÇAGA algoritması çok adım üzerinden çalıştırılır ve her adımda ÇAGA farklı bir başlangıç topluluğu ve farklı parametrelerle çalışacaktır. Tüm adımlar bittikten sonra, bulunan Pareto bireyler kullanılarak, her adım için başarıım yüzeyi bulunabilir. Şekil 3.38a' da, aynı problem için aynı algoritmayla ama farklı başlangıç topluluklarıyla çalıştırılmış, aynı ÇAGA' nın üç farklı adımı gösterilmektedir. Böyle bir algoritma sonucunda başarıımın adımlar ilerledikçe artması beklenmektedir. Bu yüzden, böyle bir grafik gerçek Pareto-optimal cephe hakkında açık bir fikir vermemektedir. İki veya daha fazla yöntem karşılaştırılacağı zaman, Pareto-optimal cephe yakınlarına yığılmış bireyler hangi yöntemin daha iyi olduğu konusunda bir fikir vermemektedirler. Şekil 3.38b, Şekil 3.38a' da verilen cephelere ilişkin başarıım yüzeylerini göstermektedir, bu yüzeyler algoritmaları karşılaştırmak için bir ölçüt tanımlamak için kullanılabilir.



Şekil 3.38. (a) Bir ÇAGA' nın üç farklı adımında bulunan bastırılmayan bireyler
(b) Bulunan üç farklı cepheye ilişkin başarıım yüzeyleri

İlk olarak, tüm amaçlarda yönelmeleri gösteren sanal köşegen hatları seçilir. Her bir hat için, tüm başarıım yüzeylerinin kesişim noktaları hesaplanır. Bu noktalar, seçilen hat üzerinde olacaktır ve bu yüzden bir frekans dağılımına sahip olacaktır. Bu noktaları kullanarak, örneğin % 25, % 50 veya % 75 başarıım yüzeyleri gibi birkaç istatistiksel değer üretilebilir. Şekil 3.39, bir çapraz AB hattını ve buna bağlı olarak üç amaç yüzeyi için kesişen noktaları göstermektedir. Frekans dağılımını belirlemek için çok sayıda başarıım yüzeyi ile kesişen nokta tanımlanabilir. Burada yalnızca, 25,50 ve 75% başarıım yüzeyleri gösterilmektedir.



Şekil 3.39. Tipik bir kesişim hattı ve kesişim noktaları. Bu noktalara göre frekans dağılımı veya histogram

3.4.3.3. Ağırlıklandırma Ölçütü (Weighted Metric)

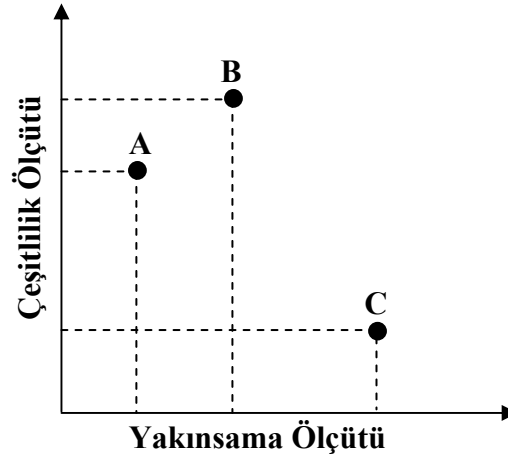
Hem yakınsamayı hem de dağılımı ölçmek için, bu değerleri ayrı ayrı ölçen ölçütleri ağırlıklandırarak birleştirmek basit bir yoldur. Örneğin;

$$W = w_1 GD + w_2 \Delta \quad (3.42)$$

Burada w_1 ve w_2 ağırlıklardır ve $w_1 + w_2 = 1$ dir. Burada yakınsamayı hesaplamak için kullanılan nesil mesafe ölçütü (GD) ile dağılımı ölçmek için kullanılan yayılım (Δ) ölçütü birleştirilmiştir. Bilindiği üzere, GD değeri küçük olan yöntem iyi bir yakınsamaya sahiptir ve Δ değeri küçük olan yöntem de iyi bir dağılım sağlar. Bu yüzden, toplam olarak küçük bir W değerine sahip olan yöntem her iki açıdan da iyi olacaktır. Kullanıcı, iki ölçütü birleştirmek için uygun ağırlıkları seçebilir. Bununla birlikte, bu ölçüt kullanılacaksa, seçilen iki ölçütün normalize olarak kullanılması daha iyi olacaktır.

3.4.3.4. Bastırılmayan Hesap Ölçütü (Non-Dominated Evaluation Metric)

Bir yöntemin ölçüt değerleri diğer yöntemin değerlerinden daha iyi ise, o zaman o yöntem diğerinden iyidir. Aksi takdirde, iki yöntem arasında hangisinin iyi olduğu konusunda kesin bir şey söylenemez. Şekil 3.40, bir problem için üç ÇAGA yönteminin başarımını göstermektedir. Şekil incelendiğinde, *A* yönteminin *B* yöntemine baskın olduğu görülmektedir. Yani *A* yöntemi, *B* yönteminden iyidir. Ancak, *A* veya *C* yöntemlerinden hangisinin iyi olduğu konusunda bir şey söylenemez.



Şekil 3.40. Yakınsama ve çeşitliliğin birlikte değerlendirilmesi

Bu bölümde açıklanan ölçütlerin dışında literatürde birçok ölçüt daha vardır ve önerilmeye devam edilmektedir. Bu ölçütlerden bazılarını [Knowles ve Corne, 2000; Knowles ve Corne, 2002; Coello Coello ve ark., 2007] kaynaklarından ulaşılabilir.

4. BULGULAR

Bu bölüm tezin ana bölümüdür ve önerilen yöntemler bu bölümde verilmektedir. Bu bölümde yapılan öneriler şu şekilde sıralanabilir:

- İlk olarak 2003 yılında önerilen DOPGA (D^Omination Power of an individual Genetic Algorithm) yöntemi [Eminoğlu, 2003] derinlemesine incelenmektedir. Bu yöntemin elitizm mekanizmasında bazı değişiklikler yapılmış ve etkinliği artırılmıştır. DOPGA yöntemi daha önce, başarımlar ölçütleri kullanılarak literatürde yer alan ÇAGA yöntemleri ile karşılaştırılmamıştır. Bu bölümde, DOPGA sık kullanılan test işlevleri ve başarımlar ölçütleri kullanılarak literatürde sık kullanılan ÇAGA yöntemleriyle (SPEA ve SPEA2) karşılaştırılmaktadır.
- DOPGA yönteminde bazı değişiklikler yapılarak iki yeni yöntem önerisi (DOPGA+, DOPGA2) sunulmaktadır. Bu değişiklikler, DOPGA yönteminin puanlama mekanizmasında veya elitizm mekanizmasında yapılmıştır.
- Literatürde yer alan bazı ÇAGA yöntemlerinin (NSGA ve SPEA) puanlama mekanizmalarında bir takım iyileştirmeler yapılarak başarımları artırılmıştır.
- Yeni bir puan ölçekleme yöntemi (gama düzeltmesi ile puan ölçekleme, GDPÖ) önerilmektedir. Bu yöntem ÇAGA yöntemlerine ek işlem yükü getirmeden uygulanmakta ve ÇAGA yöntemlerinin yakınsama başarımları artırılmaktadır.
- ÇAGA' larda elitizm mekanizmasının iyileştirilmesi için etkin elitizm mekanizması önerisi verilmektedir. Etkin elitizm mekanizması literatürde yer alan ve yeni önerilen yöntemlere uygulanarak, başarımları artırılmıştır.
- ÇAGA yöntemlerinin sıralama / nitelikli bilgi üretme yeteneklerinin ölçülmesi için iki yeni başarımlar ölçütü (ceza ve ödül) tartışmaya açılmaktadır.
- Son olarak ise ÇAGA yöntemlerini (önerilen ve var olan) içerisinde bulunduran bir kullanıcı arayüzü tasarımı ayrıntılı olarak açıklanmaktadır.

4.1. Çok Amaçlı Genetik Algoritma Yöntem Önerileri

Daha önce 2003 yılında önerilmiş olan DOPGA [Eminoğlu, 2003] yönteminde birtakım değişiklikler yapılarak çeşitli iyileştirmeler yapılmıştır. Bu bölümde ilk olarak, DOPGA yönteminin puanlama mekanizması ayrıntılı olarak verildikten sonra arşiv yönetimi aşaması verilmekte ve yapılan birtakım iyileştirmeler gösterilmektedir. DOPGA yönteminin puanlama mekanizması iyileştirilerek DOPGA+ yöntemi önerilmektedir. Hem DOPGA hem de DOPGA+’ da arşiv yönetimi, azaltımsal (subtractive) kümeleme yöntemi kullanılarak sağlanmaktadır. Son olarak, SPEA2 yönteminde kullanılan arşiv yönetimi biçimi DOPGA+ yöntemine eklenerek DOPGA2 yöntemi önerilmektedir.

Bu üç yöntemin ayrıntılı puanlama algoritmaları, elitizm veya arşiv yönetim mekanizmaları ve çok kullanılan bazı GA test işlevleri ile çalıştırılmaları neticesinde elde edilen sonuçlar aşağıda verilmektedir.

4.1.1. DOPGA (DOrnation Power of an individual Genetic Algorithm, Bireyin Baskınlık Gücü Genetik Algoritma)

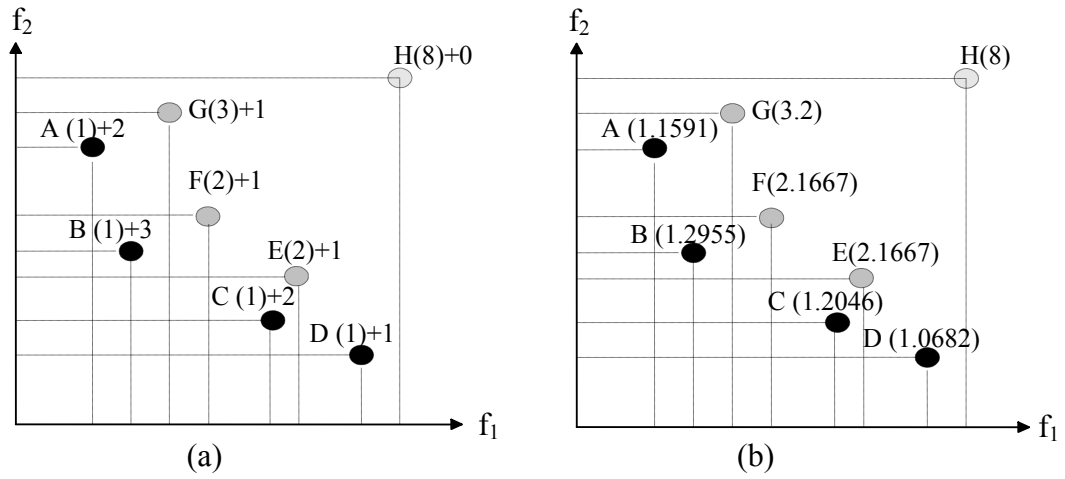
DOPGA yöntemi 2003 yılında, çok amaçlı optimizasyon için önerilmiş Pareto tabanlı ve elitist bir ÇAGA yöntemidir. DOPGA’ da bir bireyin puanı iki aşamada atanır. Arşiv yönetimi ya da ikincil topluluk yönetimi için ise azaltımsal (subtractive clustering) kümeleme yöntemi kullanılmaktadır.

4.1.1.1. DOPGA Puanlama Mekanizması

DOPGA’ da bir bireyin puanı iki aşamada elde edilmektedir. Aşağıda bu iki aşama ayrıntılı olarak açıklanmakta ve daha sonra puanlama algoritması verilmektedir. DOPGA ile bir bireyin ya da çözüm önerisinin seçme mekanizmasından önce puanı, baskınlık gücüne göre belirlenmektedir. Bu kavramla, bireylerin puanları birbirlerinden daha iyi ayırt edilebilmekte ve seçme mekanizmasına daha fazla bilgi aktarılabilmektedir.

Adım 1: İlk olarak, topluluk MOGA [Fonseca ve Fleming, 1995; 1998] yöntemi kullanılarak ön bir puanlama işlemine tabi tutulur. Böylece, topluluk aynı puana sahip birkaç bireyden oluşan alt topluluklara bölünür. Bu ilk puanlama ile **1** puanına sahip bireylerden oluşan ilk Pareto cephesi elde edilir ve diğer cephelere ait bireylere artan şekilde puan ataması yapılır. DOPGA yönteminin ilk puanlama adımını daha iyi açıklamak için Şekil 4.1a’ da verilen örnek bir topluluğun puanlamasını inceleyelim. Bu toplulukta, ilk Pareto cephesi dört bireyden oluşmaktadır: $\{A(1), B(1), C(1), D(1)\}$. $A(1)$ ’ in anlamı; o bireyin ilk Pareto cephesine üye olduğu ve hiçbir birey tarafından bastırılmadığıdır. İkinci Pareto cephesi iki bireyden oluşmaktadır: $\{E(2), F(2)\}$ ve E bireyi yalnızca bir birey tarafından bastırılmaktadır. Diğer bireylerde aynı şekilde puanlanmıştır. Bireylerin puan değerleri Şekil 4.1a’ da gösterilmektedir.

Adım 2: Bu adımda, her alt topluluk baskınlık gücü kavramı kullanılarak yeniden puanlanır. Baskınlık gücü, bireylerin konumuna yani Pareto olup olmadığına bakılmaksızın her bir bireye uygulanır. Örneğin Şekil 4.1a’ daki B bireyini düşünelim. B bireyi toplulukta n adet bireye baskın olsun. Bu birey, $B(1) + n$ olarak gösterilir ve bu örnekte $n=3$ ’ tür, çünkü B bireyi 3 adet bireye baskın olmaktadır. B bireyinin baskınlık gücü, bu üç bireyin bağıl puanlarının toplamı ile bulunur. Bu yolla, bireyler aynı puana sahip olsa veya aynı alt toplulukta olsalar bile, hangi bireyin diğerinden daha fazla istendiğini belirlemek mümkündür. Örneğin Şekil 4.1a’ da ilk Pareto cephesindeki bireylerin en düşük baskınlık gücünden (en çok istenilenden) en yüksek baskınlık gücüne (en az istenilene) doğru sıralaması şu şekildedir: $D(1)+1 \succ A(1)+2 \succ C(1)+2 \succ B(1)+3$. A ve C bireyleri iki adet bireye baskın olmalarına rağmen, A bireyi C bireyinden daha fazla istenmektedir, çünkü A bireyi daha az toplam puanlı bireylere baskın olmaktadır. $D(1)+1$, yalnızca $H(8)$ bireyine baskın olmaktadır ve bu yüzden diğer bireylere göre en düşük baskınlık gücüne sahiptir. Şekil 4.1b’ de DOPGA tarafından puanlanmış bir topluluk gösterilmektedir.



Şekil 4.1. (a) DOPGA ham puanları, (b) DOPGA son puanları

Şekil 4.1b' den de görüleceği üzere, ilk Pareto cephesine ait bireyler birbirinden farklı puanlar almıştır ve bireylerin önem sırası şöyledir: $D \succ A \succ C \succ B \succ E, F \succ G \succ H$. DOPGA puanlama mekanizmasında bireylerin komşulukları özellikle dikkate alınır. Örneğin, eğer bir birey aynı alt topluluktaki diğer bir bireyden daha yüksek bir baskınlık gücüne sahipse, o zaman o birey kendi alt topluluğunda en az istenilen birey olacaktır. Eğer bir birey yüksek bir baskınlık gücüne sahipse, bunun anlamı o bireyin kalabalık bir bölgede olduğudur, bu yüzden yakınında birkaç benzer birey bulunmaktadır. DOPGA böyle bir bireyi cezalandırır ve onu o alt toplulukta en az istenilen birey yapar. DOPGA, yakın bir komşuyu uzak olandan ayırt edebilir, çünkü yakın bir komşu uzak olan komşudan daha yüksek bir baskınlık gücüne (daha az istenilen) sahiptir. Bu yüzden DOPGA çeşitlilik için harici bir katsayı tahminine gereksinim duymaz.

DOPGA yönteminin puanlama mekanizmasının ayrıntılı algoritması Şekil 4.2' de verilmektedir.

DOPGA Puanlama Algoritması

Adım 1: Her bireye alt topluluk (*alt_top*) numarası ile ters orantılı olan bir yapay puan atanır (MOGA algoritması kullanılarak).

$$yp(birey_i) = \frac{1}{alt_top(j)} \quad (4.1)$$

$i=1\dots n$, burada n bireylerin toplam sayısı ve $j=1\dots m$, m ise toplam alt topluluk sayısıdır.

Adım 2: Bireyin *bağıl baskınlık* gücü hesaplanır. Bunun için, kaç tane bireyin bu birey tarafından bastırıldığı belirlenir ve baskın olan bireylerin yapay puanları toplanır.

$$bbg(birey_i) = \sum_{k=1}^r yp(birey_k) \quad (4.2)$$

Burada $k\dots r$, i bireyinin baskın olduğu tüm bireyleri göstermektedir ve $k\dots r \neq i$.

Adım 3: Aynı alt topluluktaki tüm bireyler için Adım 2 tekrar edilir. Aynı alt topluluktaki tüm bireylerin *bbg*()' leri toplanır ve her alt topluluk için toplam *baskınlık* gücü aşağıdaki gibi hesaplanır:

$$tbg(alt_top(j)) = \sum_i^p bbg(birey_i) \quad (4.3)$$

Burada $i\dots p$, *Alt_top(j)*' de yer alan tüm bireylerdir.

Adım 4: i bireyinin *gerçek puanı* aşağıdaki formül kullanılarak hesaplanır (paydaya sıfıra bölme durumunda kurtulmak için 1 eklenmiştir):

$$gp(i) = alt_top(j) + \frac{bbg(i)}{tbg(alt_top(j)) + 1} \quad (4.4)$$

Adım 5: Topluluktaki tüm bireyler puanlanana kadar Adım 2 - Adım 4 arası tekrarlanır.

Şekil 4.2. DOPGA puanlama algoritması

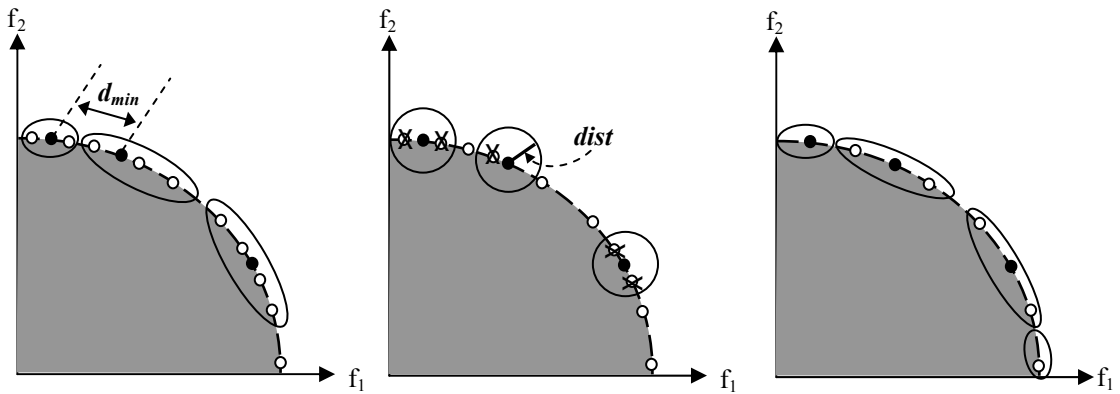
4.1.1.2. DOPGA Elitizm Mekanizması

DOPGA yönteminde kullanılan elitizm mekanizmasının blok diyagramı Şekil 4.3' te verilmektedir. Seçme mekanizmasından önce ana topluluktaki bireylerle, ikincil topluluk yani arşivdeki bireyler birleştirilmektedir. Eğer arşivdeki birey sayısı daha önce kullanıcı tarafından belirlenmiş boyutu aşarsa, azaltımsal kümelemenin [Chiu, 1994] değiştirilmiş bir biçimi uygulanır ve azaltılmış topluluk (kullanıcı tarafından belirlenen boyuta düşürülmüş) elde edilir. Eğer arşivin boyutu kullanıcı tarafından belirlenmiş boyuta ulaşmazsa, arşivdeki tüm bireyler ana toplulukla birleştirilir.

Azaltımsal kümeleme algoritması, kümeleri (veya küme merkezlerini) bulmak için uygulanır. Bir sonraki aşamada merkezler arasındaki en küçük mesafe bulunur. Doğal olarak, kümelerin merkezleri kalabalık bölgeleri temsil eder ve bu bölgeler de birbirlerine benzer bireylerden oluşmaktadırlar. Küme merkezlerini temsil eden bireyler ve küme merkezlerinden yeterli derecede uzak bireyler seçilir. Böylece, kümeleme algoritması tarafından seçilmiş bireyler arşive kopyalanır. Kümeleme işleminin grafiksel olarak ayrıntılı açıklaması Şekil 4.4' te verilmektedir.

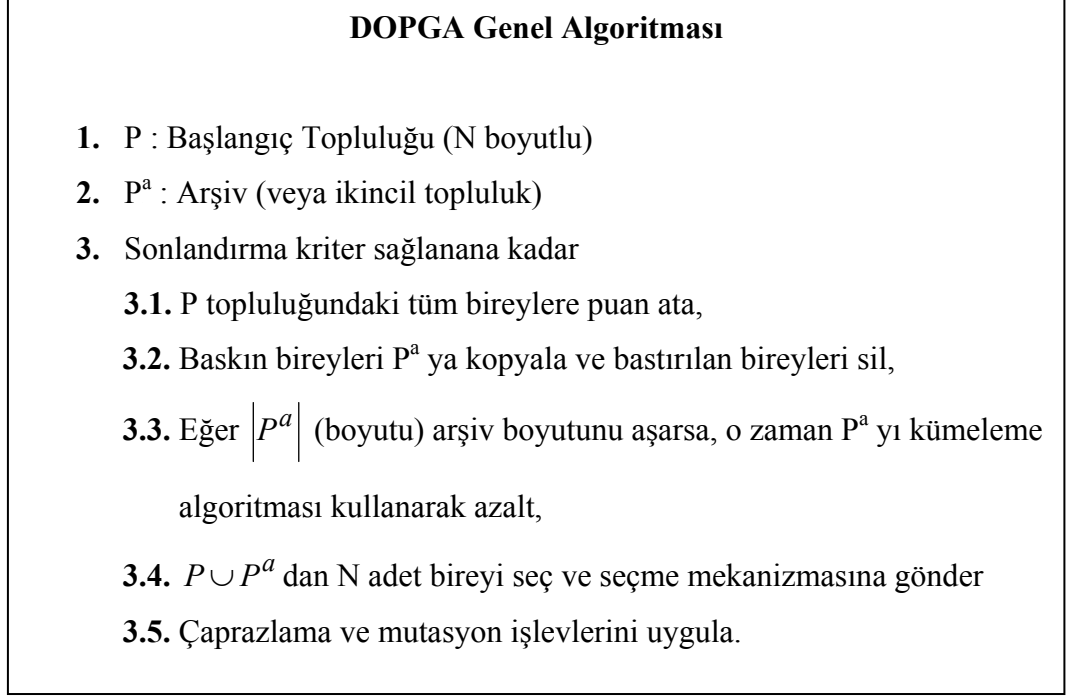


Şekil 4.3. DOPGA elitizm mekanizmasının blok diyagramı



Şekil 4.4. Arşivdeki Pareto bireylerin azaltılma adımları: i) Azaltımsal kümeleme kullanılarak merkezler bulunur, ii) Merkezler arasındaki en küçük mesafe bulunur (d_{min}) ve yeni bir mesafe tanımı yapılır $dist = k \times d_{min}$ (örneğin $k=0.9$ olsun), merkezlerden $dist$ mesafesinden daha uzak olan bireyler arşivde tutulur, diğerleri silinir, iii) Azaltılmış topluluk, bu bireyler ve merkezlerden oluşur

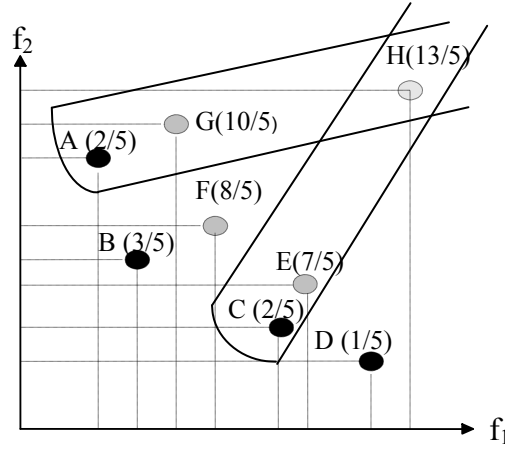
DOPGA yönteminin genel algoritması Şekil 4.5' te verilmektedir.



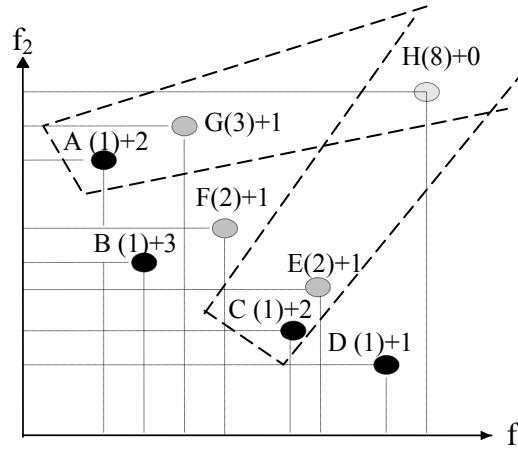
Şekil 4.5. DOPGA genel algoritması

4.1.1.3. DOPGA ve SPEA Yöntemlerinin Görsel Karşılaştırması

Pareto Bireylerin Karşılaştırılması: SPEA yöntemi, topluluktaki Pareto bireylere baskın oldukları birey sayısına göre puan atar. Örnek bir topluluk Şekil 4.6' da verilmektedir. Bu şekilden de görüleceği üzere aynı puana sahip bireyler olabilir. SPEA puanlamasına göre A ve C bireyleri aynı puana sahip olacaktır, çünkü ikisi de iki adet Pareto olmayan bireye, sırasıyla (G,H) ve (E,H) bireyelerine baskın olmaktadır. DOPGA puanlamasında ise bir Pareto (veya bir Pareto olmayan) bireyin baskın olduğu bireylerin sayısı önemlidir, ancak puan atama işlemi toplam birey sayısından ziyade toplam puana göre yapılır (Bkz. Şekil 4.7). DOPGA puanlamasında da A ve C bireyleri sırasıyla (G,H) ve (E,H) bireyelerine baskın olmaktadır ancak, iki bireyin baskın olduğu toplam puan aynı değildir (Bkz. Şekil 4.7). Bu yüzden, A ve C bireyleri kendi alt topluluklarında artık ayırt edilebilir.

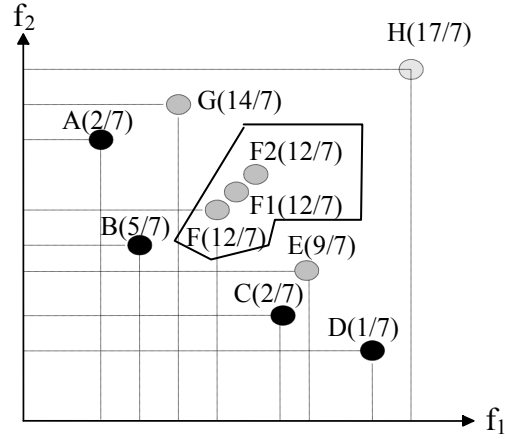


Şekil 4.6. SPEA’da, A ve C birbirinden ayırt edilemez

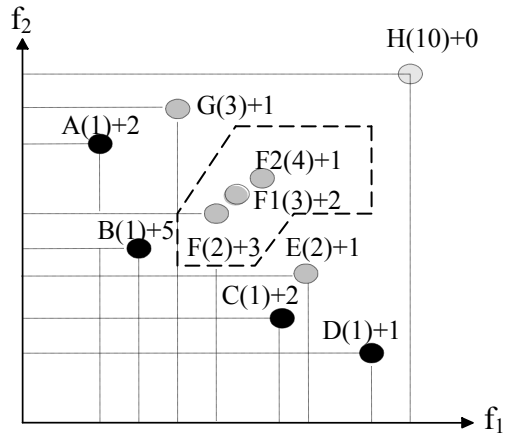


Şekil 4.7. DOPGA’da, A ve C birbirinden ayırt edilebilir

Pareto Olmayan Bireylerin Karşılaştırılması: Pareto olmayan bireylerin karşılaştırılması da bir diğer önemli noktadır. SPEA’da Pareto olmayan bir bireyin puanı, ona baskın olan Pareto bireylerin puanları toplamına bir eklenmesiyle bulunur. Bu nedenle, eğer bir grup Pareto olmayan birey yalnızca bir Pareto birey tarafından bastırılırsa, o zaman hepsi de aynı puana sahip olacaktır (Bkz. Şekil 4.8). F , $F1$ ve $F2$ bireyleri yalnızca B bireyi tarafından bastırılmaktadır ve bu yüzden SPEA bu bireylere aynı puanı atar. DOPGA’da ise, bu üç birey [$F(2)$, $F1(3)$ ve $F2(4)$] birbirinden kolaylıkla ayırt edilebilir, çünkü Şekil 4.9’ dan da görüleceği üzere hepsi de farklı alt topluluklara aittir. SPEA bu tip bireylere aynı puanı atarken, DOPGA hepsine farklı puan atayacaktır.

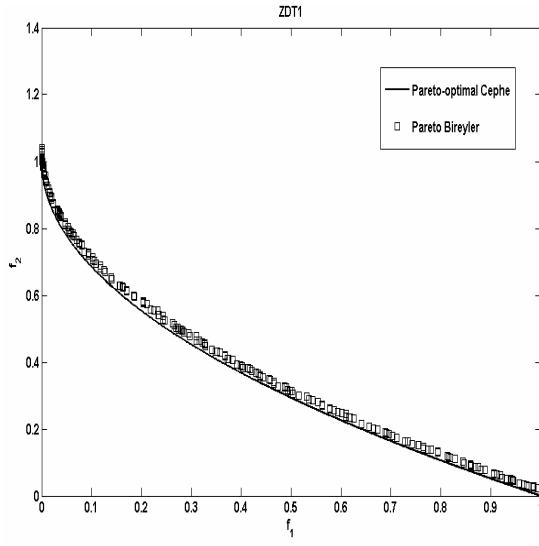


Şekil 4.8. SPEA; F , $F1$ ve $F2$ bireylerine aynı puanı atar, çünkü bu bireyler yalnızca B bireyi tarafından bastırılmaktadır

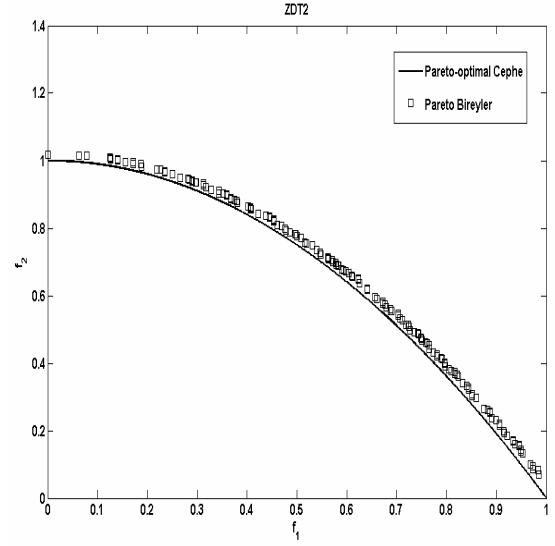


Şekil 4.9. DOPGA' da ise F , $F1$ ve $F2$ bireyleri aynı alt topluluğa ait olmadıkları için farklı puanlara sahiptirler

DOPGA yöntemi; ZDT1, ZDT2, ZDT3 ve ZDT6 test işlevleri üzerinde 300 adım çalıştırılmış, Şekil 4.10 ve Şekil 4.11' de verilen sonuçlar elde edilmiştir. Benzetimlerde; parametre sayısı ZDT1, ZDT2 ve ZDT3' te 30, ZDT6' da 10, bit sayısı 30, birey sayısı 150, arşiv boyutu 100, çaprazlama olasılığı 0.9, mutasyon olasılığı 0.01 seçilmiştir. Tek noktalı çaprazlama ve bit-tabanlı mutasyon işlevleri uygulanmıştır. Seçme mekanizması olarak SUS kullanılmıştır.

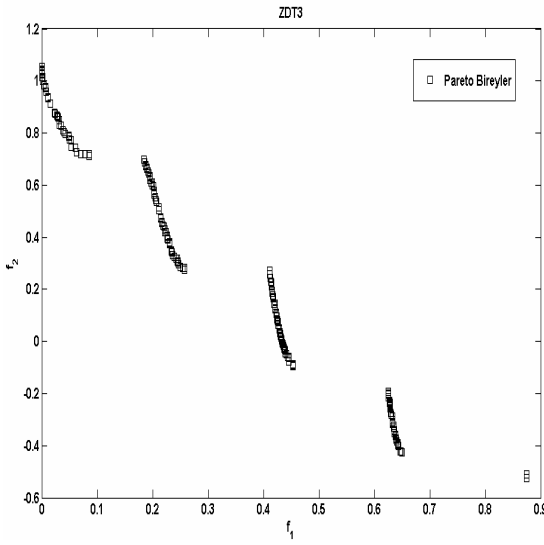


(a)

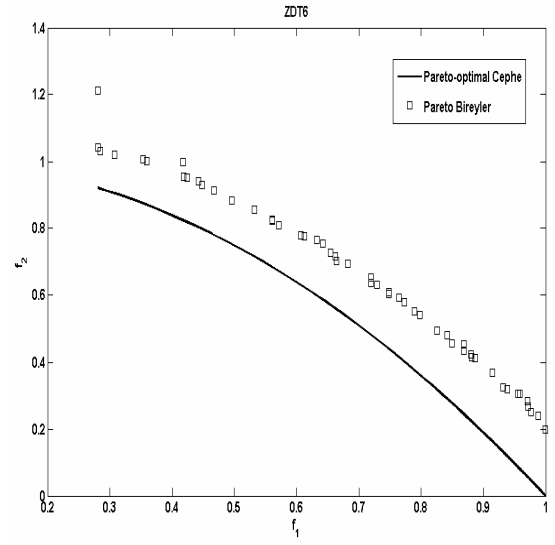


(b)

Şekil 4.10. DOPGA tarafından bulunan sonuçlar : (a) ZDT1, (b) ZDT2



(a)



(b)

Şekil 4.11. DOPGA tarafından bulunan sonuçlar: (a) ZDT3, (b) ZDT6

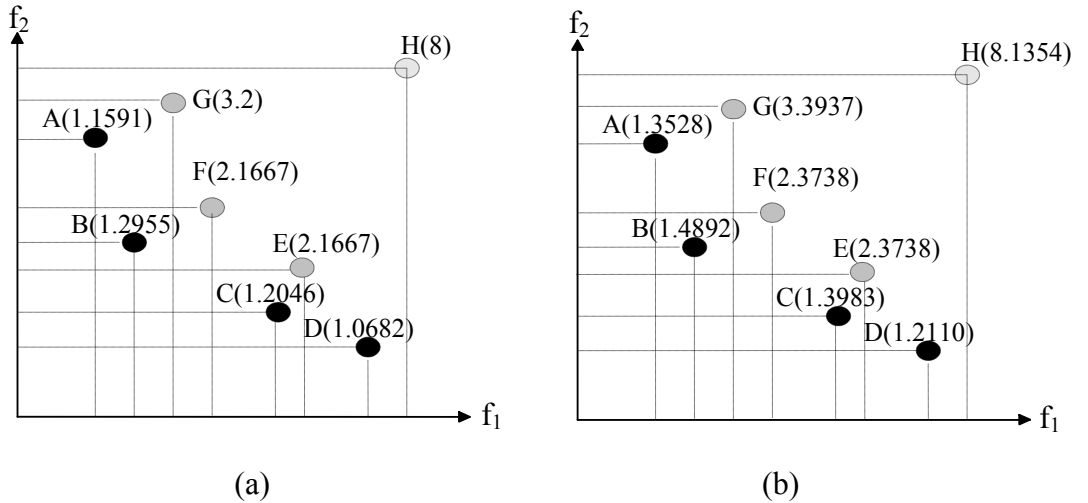
4.1.2. DOPGA+ (DOmination Power of an individual Genetic Algorithm +, Bireyin Baskınlık Gücü Genetik Algoritma +)

4.1.2.1. DOPGA+ Puanlama Mekanizması

DOPGA+ yönteminde bir bireyin puanı; DOPGA yöntemi ile elde edilen puanlara *k. en yakın komşuluk* yöntemi ile elde edilen (SPEA2' olduğu gibi [Zitzler ve ark., 2001]) yoğunluk bilgisinin eklenmesiyle bulunur. Buna göre bir bireyin puanı şöyle hesaplanır:

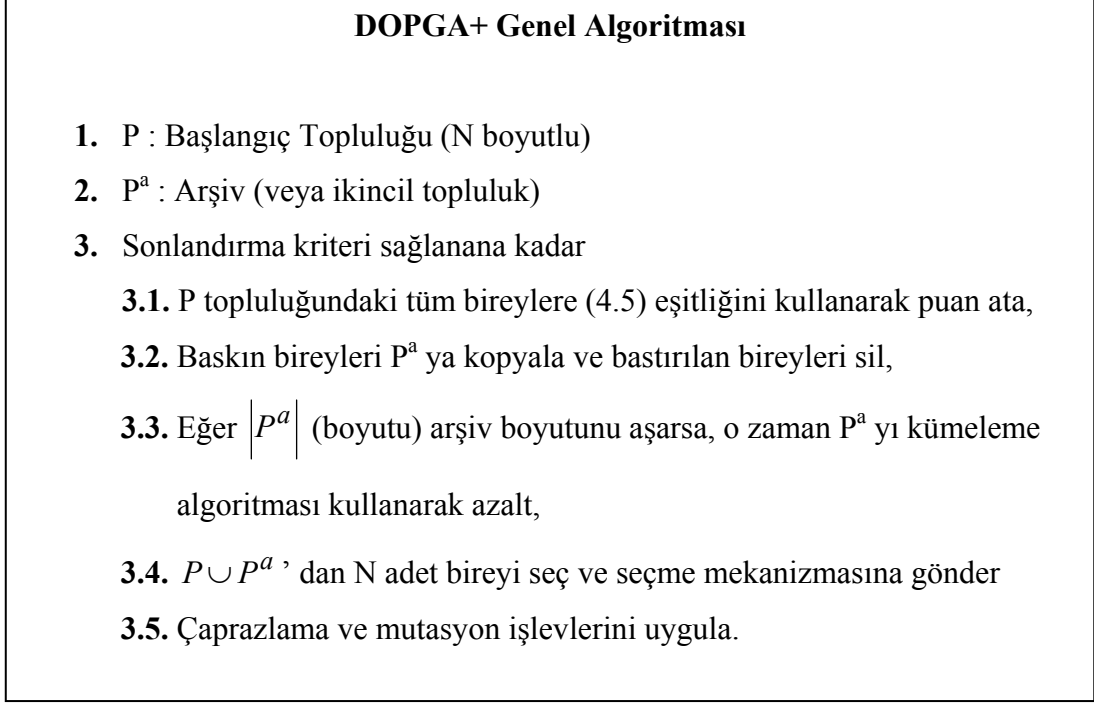
$$puan_{DOPGA+}(i) = puan_{DOPGA}(i) + D_{DOPGA+}(i) \quad i=1,2,\dots,top_boyutu \quad (4.5)$$

Bu eşitlikte *D* yoğunluk bilgini göstermektedir ve (3.13) nolu eşitlik kullanılarak bulunur. DOPGA+ tarafından puanlanan örnek bir topluluk Şekil 4.12 (a) ve (b)' de gösterilmektedir. Şekil 4.12b' den de görüleceği üzere, ilk Pareto cephesindeki bireyler farklı puan değerlerine sahiptirler ve buna göre bireylerin önem sırası şöyle olacaktır: $D \succ A \succ C \succ B \succ E, F \succ G \succ H$.



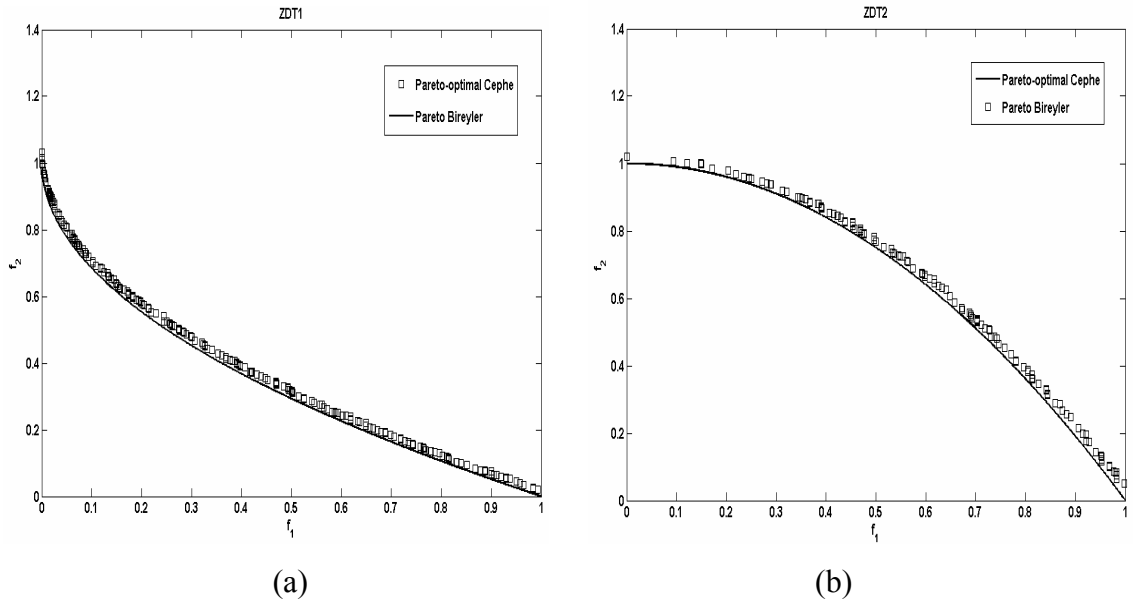
Şekil 4.12. (a) DOPGA+ ham puanlar, (b) DOPGA+ son puanlar

DOPGA+ yönteminde elitizm mekanizması olarak DOPGA yönteminde olduğu gibi azaltımsal kümeleme yöntemi kullanılmaktadır. DOPGA+ yönteminin genel algoritması Şekil 4.13’ te verilmektedir.

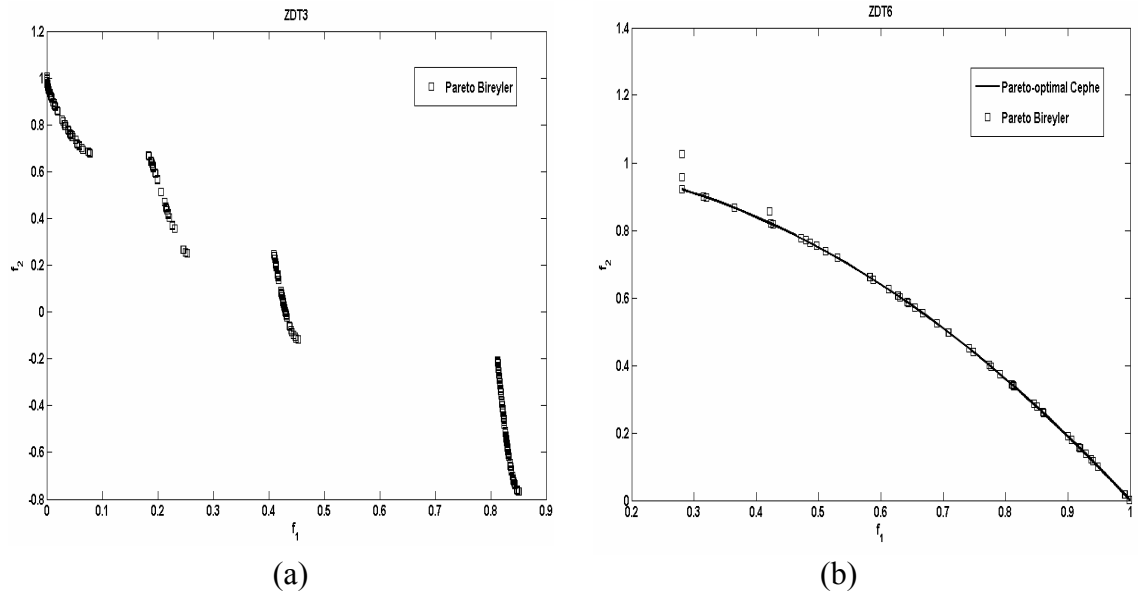


Şekil 4.13. DOPGA+ genel algoritması

DOPGA+ yöntemi; ZDT1, ZDT2, ZDT3 ve ZDT6 test işlevleri üzerinde 300 adım çalıştırılmış, Şekil 4.14 ve Şekil 4.15’ de verilen sonuçlar elde edilmiştir. Benzetimlerde; parametre sayısı ZDT1, ZDT2 ve ZDT3’ te 30, ZDT6’ da 10, bit sayısı 30, birey sayısı 150, arşiv boyutu 100, çaprazlama olasılığı 0.9, mutasyon olasılığı 0.01 seçilmiştir. Tek noktalı çaprazlama ve bit-tabanlı mutasyon işlevleri uygulanmıştır. Seçme mekanizması olarak SUS kullanılmıştır.



Şekil 4.14. DOPGA+ tarafından bulunan sonuçlar: (a) ZDT1, (b) ZDT2



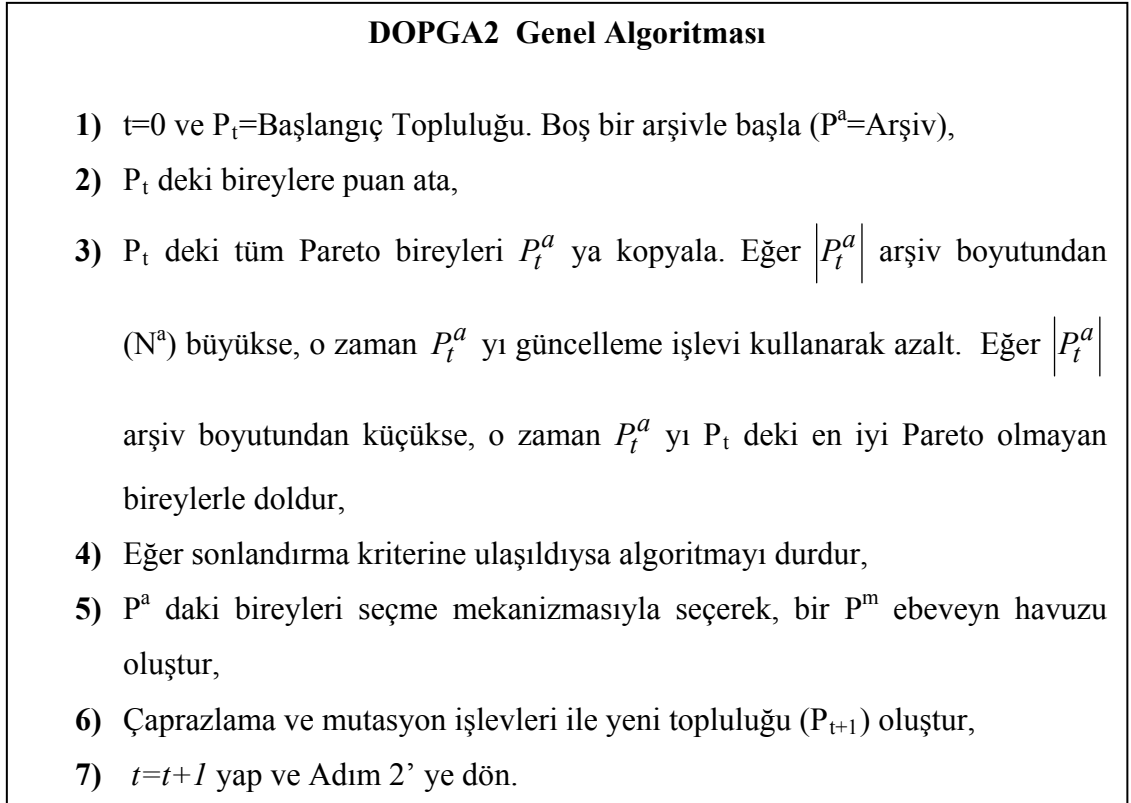
Şekil 4.15. DOPGA+ tarafından bulunan sonuçlar: (a) ZDT3, (b) ZDT6

4.1.3. DOPGA2 (DOnation Power of an individual Genetic Algorithm 2, Bireyin Baskınlık Gücü Genetik Algoritma 2)

DOPGA2 yönteminde bir topluluktaki bireyler DOPGA puanlama algoritması kullanılarak puanlanır. Ancak DOPGA2, DOPGA yöntemine göre daha iyi bir arşiv yönetimi sağlayarak, elitizm mekanizmasını daha iyi kullanmaktadır.

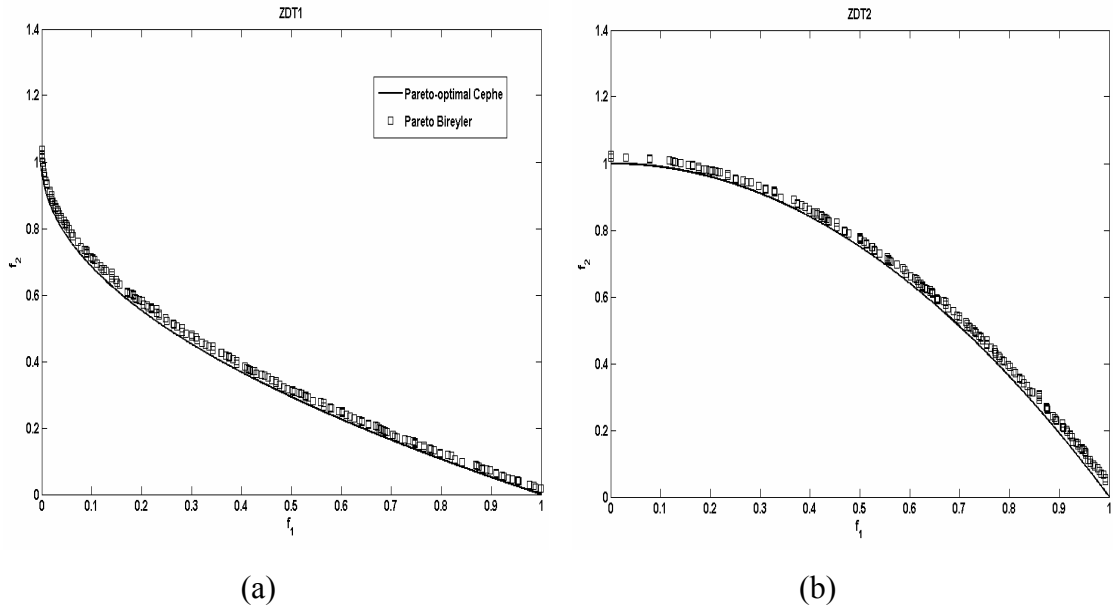
4.1.3.1. DOPGA2 Arşiv Yönetimi

DOPGA2 yönteminde, SPEA2 yönteminde olduğu gibi arşivde saklanan Pareto bireylerin sayısı bir arşiv güncelleme yöntemi kullanılarak azaltılır. Bu yöneme ilişkin ayrıntılı bilgi Bölüm 3.2.4.2’ de verilmektedir. Diğer bireylere en yakın mesafedeki birey ilk silinecek birey olarak seçilir, eğer aynı durumda birkaç birey varsa bu durumda ikinci en yakın birey silinir ve bu şekilde işlemler devam eder. DOPGA2 yönteminin genel algoritması Şekil 4.16’ da verilmektedir.

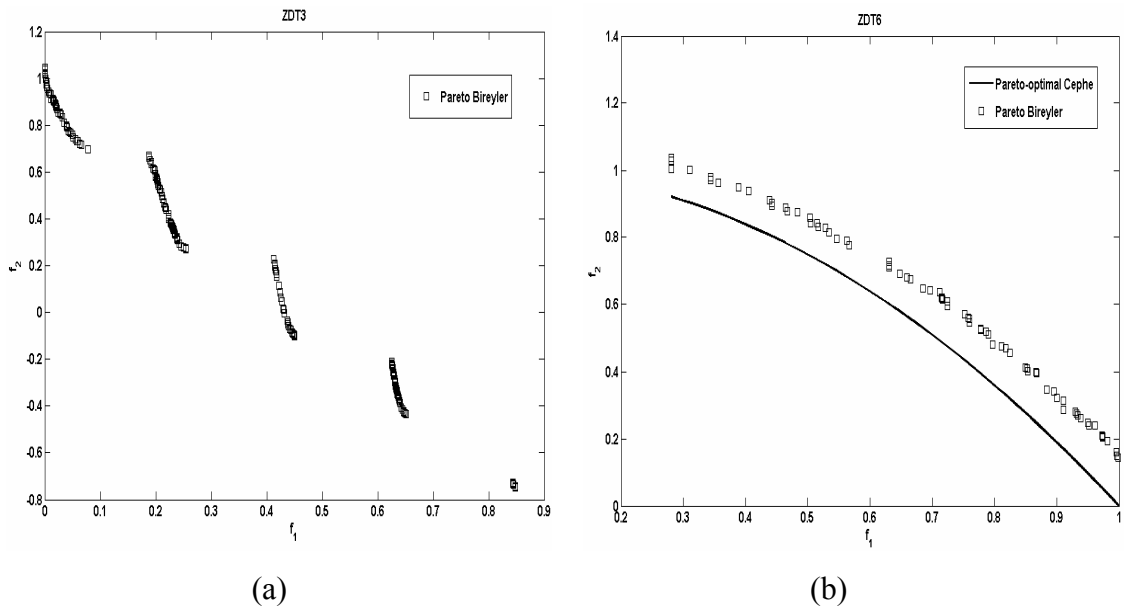


Şekil 4.16. DOPGA2 genel algoritması

DOPGA2 yöntemi; ZDT1, ZDT2, ZDT3 ve ZDT6 test işlevleri üzerinde 300 adım çalıştırılmış, Şekil 4.17 ve Şekil 4.18’ de verilen sonuçlar elde edilmiştir. Benzetimlerde; parametre sayısı ZDT1, ZDT2 ve ZDT3’ te 30, ZDT6’ da 10, bit sayısı 30, birey sayısı 150, arşiv boyutu 100, çaprazlama olasılığı 0.9, mutasyon olasılığı 0.01 seçilmiştir. Tek noktalı çaprazlama ve bit-tabanlı mutasyon işlevleri uygulanmıştır. Seçme mekanizması olarak SUS kullanılmıştır.



Şekil 4.17. DOPGA2 tarafından bulunan sonuçlar: (a) ZDT1, (b) ZDT2



Şekil 4.18. DOPGA2 tarafından bulunan sonuçlar: (a) ZDT3, (b) ZDT6

4.2. Çok Amaçlı Genetik Algoritma Yöntemlerinin Puanlama Mekanizmalarının İyileştirilmesi

ÇAGA yöntemlerinde iki türlü iyileştirme yapmak mümkündür: *i)* puanlama (puan atama) mekanizmalarının iyileştirilmesi, *ii)* elitizm mekanizmalarının iyileştirilmesi. Bu bölümde ilk tür iyileştirme kullanılarak literatürde yer alan bazı yöntemlerin puan atama mekanizmaları iyileştirilmiş ve başarımlarının (yakınsama ve düzgün dağılım açısından) değişip değişmediği incelenmiştir. İyileştirmeler için iki temel kavram kullanılmıştır. Birincisi, *k. en yakın komşuluk* yönteminden yararlanılarak eklenen yoğunluk bilgisi, ikincisi ise baskınlık gücü kavramıdır. Bu bölümde, NSGA [Srinivas ve Deb, 1994] ve SPEA [Zitzler, 1999; Zitzler ve Thiele, 1999] yöntemlerine ilişkin toplam 5 adet iyileştirme önerilmektedir. Bu iyileştirmelerden bir tanesi NSGA, dört tanesi ise SPEA yöntemine ilişkindir. Bu iyileştirmelerin yapılma amacı, puan atama mekanizmalarının iyileştirilmesiyle seçme mekanizmasına daha iyi ve nitelikli bilgi gönderebilmesini sağlamaktır. Ayrıca, NSGA yöntemi ve iyileştirmesi (NSGAmoD) elitist biçime getirilmişlerdir. İyileştirmeler, literatürde sık kullanılan dört test işlevi kullanılarak orjinal yöntemlerle karşılaştırılmışlardır. Karşılaştırmalarda GD (yakınsama için) ve Δ (düzgün dağılım için) başarımlar ölçütleri kullanılmıştır.

4.2.1. *k. En Yakın Komşuluk Yöntemi Kullanılarak Oluşturulan Yoğunluk Bilgisi*

Bu bölümde kullanılan yoğunluk tahmin tekniği, *k. en yakın komşuluk* yönteminin bir uyarlamasıdır ve herhangi bir bireyin yoğunluğu *k. en yakın komşusuna* olan mesafesinin azalan bir işlevidir [Zitzler ve ark., 2001]. Daha iyi bir hesaplama için, herhangi bir *i* bireyinin arşivde veya o andaki topluluktaki tüm bireylere olan mesafesi hesaplanır ve bir listede saklanır. Bu listenin artan bir sırada sıralanmasından sonra, *k. elemana* olan mesafe istenilen mesafeyi verir, bu mesafe σ_i^k olarak gösterilir. Yaygın olarak, *k* örnekleme oranının karekökü olarak seçilir $k = \sqrt{N + \bar{N}}$ (*N*: topluluk boyutu, $\bar{N}$: arşiv boyutu). Böylece, *i* bireyinin yoğunluğu $D(i)$ şöyle hesaplanır:

$$D(i) = \frac{1}{\sigma_i^k + 2} \quad (4.6)$$

4.2.2. Baskınlık Gücü Kavramı

Bir toplulukta yer alan bireylerin baskınlık güçleri aşağıda verilen algoritma kullanılarak bulunur. Bu bölümde, ayrıntılı algoritma ve sayısal örnekler verilmektedir.

Baskınlık Gücü Algoritması

Adım 1: Her bireye alt topluluk (*alt_top*) numarası ile ters orantılı olan bir yapay *p* puan atanır (MOGA algoritması kullanılarak).

$$yp(birey_i) = \frac{1}{alt_top(j)} \quad (4.7)$$

$i=1\dots n$, burada n bireylerin toplam sayısı ve $j=1\dots m$, m ise toplam alt topluluk sayısıdır.

Şekil 3.12a' da verilen örnek topluluk için yapay puanlar şöyle hesaplanır: (üç tane alt topluluk vardır ({A,B,C,D},{E,F},{G},{H})):

yp(A)=1	yp(B)=1	yp(C)=1	yp(D)=1
yp(E)=0.5	yp(F)=0.5	yp(G)=0.3333	yp(H)=0.25

Adım 2: Bireyin *bağlı baskınlık gücü* hesaplanır. Bunun için, kaç tane bireyin bu birey tarafından bastırıldığı belirlenir ve baskın olan bireylerin yapay puanları toplanır.

$$bbg(birey_i) = \sum_{k=1}^r yp(birey_k) \quad (4.8)$$

Burada $k\dots r$, i bireyinin baskın olduğu tüm bireyleri göstermektedir ve $k\dots r \neq i$.

Aynı örnek topluluk için *bbg* değerleri şöyle hesaplanır:

$$\begin{aligned} bbg(A) &= yp(G) + yp(H) = 0.3333 + 0.25 = 0.5833 \\ bbg(B) &= yp(F) + yp(G) + yp(H) = 0.5 + 0.3333 + 0.25 = 1.0833 \\ bbg(C) &= yp(E) + yp(H) = 0.5 + 0.25 = 0.75 \\ bbg(D) &= yp(H) = 0.25 \\ bbg(E) &= bbg(F) = bbg(G) = 0.25 \\ bbg(H) &= 0 \end{aligned}$$

Adım 3: Aynı alt topluluktaki tüm bireyler için Adım 2 tekrar edilir. Aynı alt topluluktaki tüm bireylerin *bbg*' leri toplanır ve her alt topluluk için *toplam baskınlık gücü* aşağıdaki gibi hesaplanır:

$$tbg(alt_top(j)) = \sum_i^p bbg(birey_i) \quad (4.9)$$

Burada $i \dots p$, $Alt_top(j)$ ' de yer alan tüm bireylerdir. Aynı örnek için;

$$\begin{aligned} tbg(alt_top_1) &= bbg(A) + bbg(B) + bbg(C) + bbg(D) = 0.5833 + 1.0833 + 0.75 + 0.25 = 2.6666 \\ tbg(alt_top_2) &= bbg(E) + bbg(F) = 0.25 + 0.25 = 0.50 \\ tbg(alt_top_3) &= bbg(G) = 0.25 \\ tbg(alt_top_4) &= bbg(H) = 0 \end{aligned}$$

Adım 4: i bireyinin *Baskınlık Gücü (BG)* aşağıdaki formülle hesaplanır (paydadaki 1 sıfıra bölme hatasından kurtulmak için konulmuştur):

$$BG(i) = \frac{bbg(i)}{tbg(alt_top(j)) + 1} \quad (4.10)$$

Örnek topluluktaki tüm bireylerin baskınlık güçleri şöyle olacaktır:

$$\begin{aligned}
 BG(A) &= 0.5833 / (2.6666 + 1) = 0.1591 & BG(E) &= 0.25 / (0.50 + 1) = 0.1667 \\
 BG(B) &= 1.0833 / (2.6666 + 1) = 0.2955 & BG(F) &= 0.25 / (0.50 + 1) = 0.1667 \\
 BG(C) &= 0.75 / (2.6666 + 1) = 0.2045 & BG(G) &= 0.25 / (0.25 + 1) = 0.2 \\
 BG(D) &= 0.25 / (2.6666 + 1) = 0.0682 & BG(H) &= 0 / (0 + 1) = 0
 \end{aligned}$$

Adım 5: Topluluktaki tüm bireylerin baskınlık gücü bulunana kadar Adım 2 - Adım 4 arası tekrarlanır.

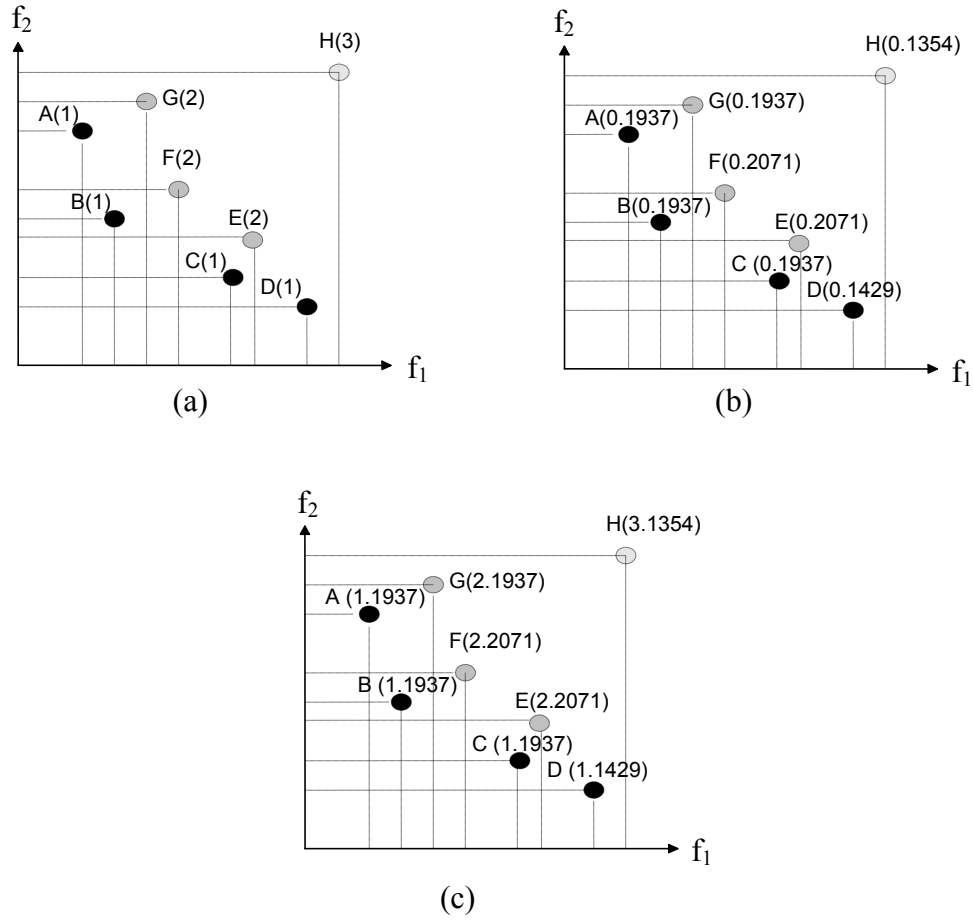
4.2.3. NSGA İyileştirmesi (NSGAmoD)

NSGA yönteminde puanlama işlemi gerçekleştirildikten sonra, bireylerin çeşitliliğinin sağlanması için paylaşım yöntemi (burada paylaşım yöntemi amaç uzayında gerçekleştirilmiştir) uygulanmaktadır [Goldberg ve Richardson, 1987; Deb, 2001].

Yöntemin bu iyileştirilmesinde ise, paylaşım yöntemi yerine yoğunluk bilgisi kullanılarak çeşitlilik sağlanmaktadır. NSGA iyileştirmesi (modification), **NSGAmoD** simgesiyle gösterilmektedir. Buna göre NSGAmoD iyileştirmesinde bir i bireyinin puanı şöyle hesaplanır :

$$F_{NSGA\ mod}(i) = NSGA_puan(i) + D_{NSGA\ mod}(i) \quad i=1,2,\dots,top_boyut \quad (4.11)$$

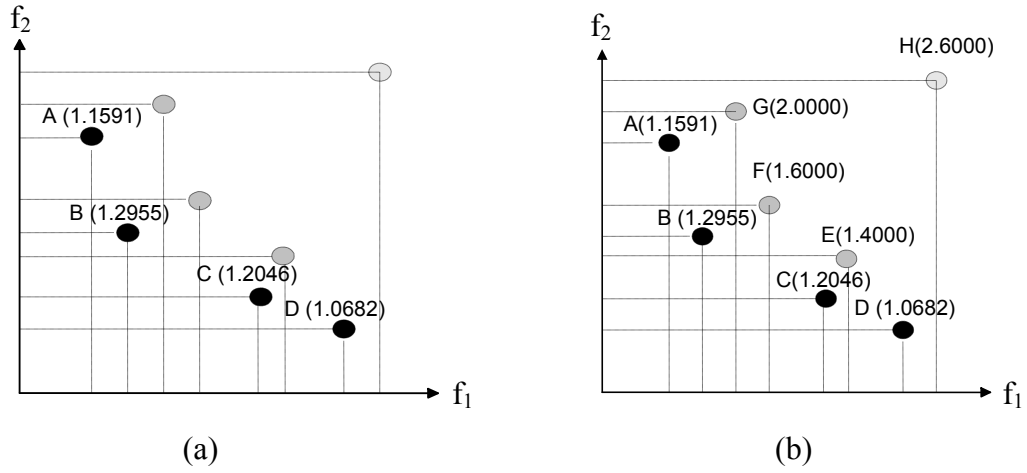
Burada D yoğunluk bilgisini göstermektedir. Hem orjinal NSGA hem NSGAmoD elitist bir biçime getirilmişlerdir. NSGAmoD ile puanlanan örnek bir topluluk Şekil 4.19' da gösterilmektedir. Buna göre, bireylerin en çok istenilenden en az istenilene doğru sıralaması şöyledir: $D \succ A, B, C \succ G \succ E, F \succ H$.



Şekil 4.19. NSGAmoD puanlaması: (a) İlk adım: NSGA puanlaması, (b) İkinci adım: yoğunluk bilgisi, (c) NSGAmoD tarafından atanan son puan değerleri

4.2.4. SPEA' nın Birinci İyileştirmesi (SPEAmoD1)

Bu iyileştirmede, bir toplulukta diğer bireylere göre daha önemli olan elit ya da Pareto bireyler daha derinlemesine bir puanlamaya tabi tutulmaktadır. Öncelikle Pareto bireyler baskınlık gücü kavramı kullanılarak puanlanmıştır, Pareto olmayan bireyler ise SPEA yöntemi kullanılarak puanlamaya tabi tutulmuştur. SPEAmoD1 ile puanlanan örnek bir topluluk Şekil 4.20' de gösterilmektedir. Buna göre bireylerin en çok istenilenden en az istenilene doğru sıralaması şöyledir: $D \succ A \succ C \succ B \succ E \succ F \succ G \succ H$.



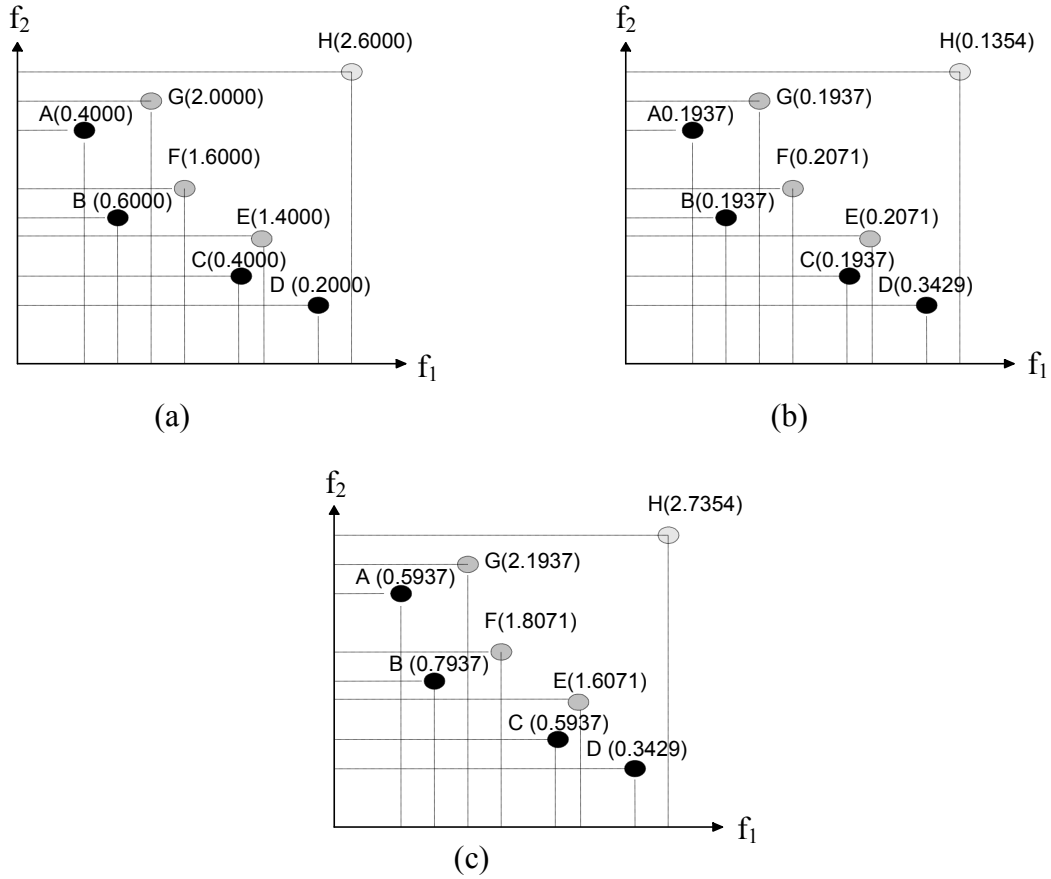
Şekil 4.20. SPEAmod1 puanlaması: (a) İlk adım: baskınlık gücü ile puanlama, (b) SPEA ile Pareto olmayan bireylerin puanlanması

4.2.5. SPEA' nın İkinci İyileştirmesi (SPEAmod2)

Bu iyileştirmede SPEA yöntemi ile elde edilen ham puan (raw fitness) değerlerine k . en yakın komşuluk yönteminden yararlanılarak hesaplanan yoğunluk bilgisi eklenmektedir. Bu iyileştirmede bir i bireyinin puan değeri şöyle hesaplanır:

$$F_{SPEAmod2}(i) = F_{SPEA}(i) + D_{SPEAmod2}(i) \quad i=1,2,\dots,top_boyut \quad (4.12)$$

SPEAmod2 ile puanlanan örnek bir topluluk Şekil 4.21' de gösterilmektedir. Buna göre bireylerin en çok istenilenden en az istenilene doğru sıralaması şöyledir: $D \succ A, C \succ B \succ E \succ F \succ G \succ H$.



Şekil 4.21. SPEAmod2 puanlaması: (a) İlk adım: SPEA puanlaması, (b) İkinci adım: yoğunluk bilgisi, (c) SPEAmod2 tarafından atanan son puan değerleri

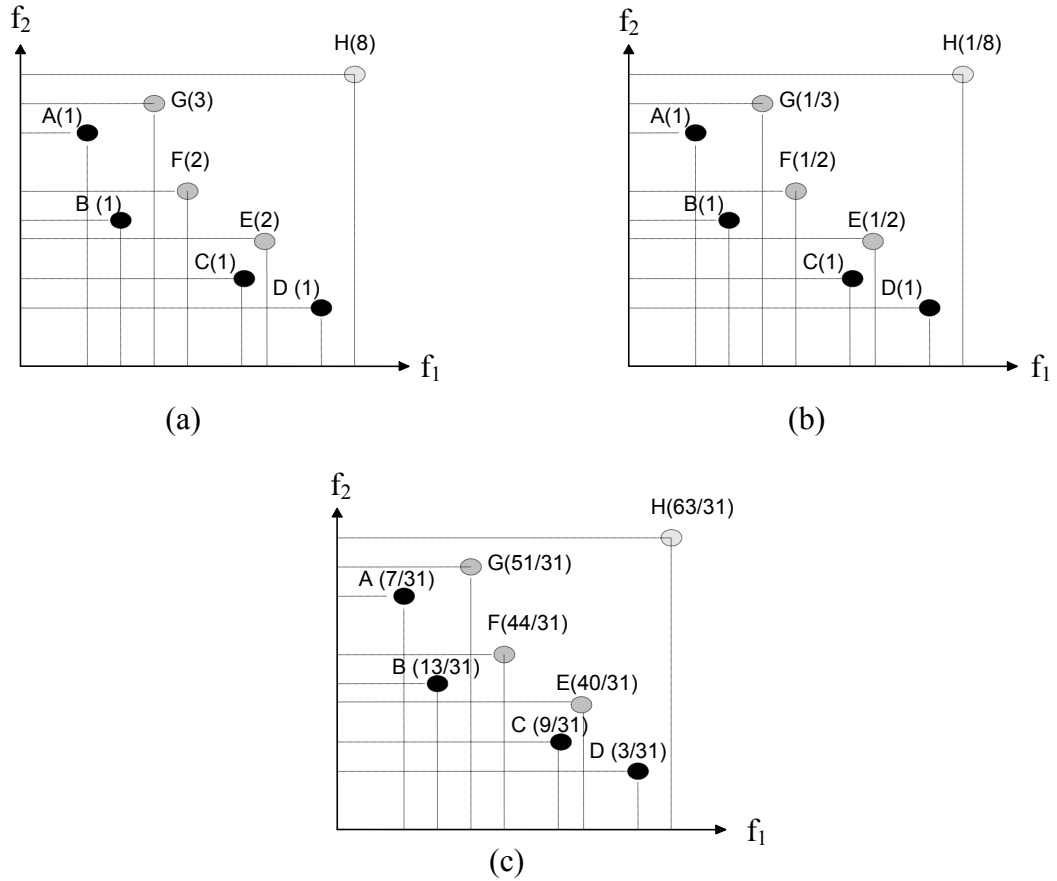
4.2.6. SPEA' nın Üçüncü İyileştirmesi (SPEAmod3)

Bu yöntem iyileştirme olarak görünmekle birlikte farklı kavramları da içerisinde barındırmaktadır. Burada, ilk olarak MOGA yöntemi kullanılarak topluluk alt topluluklarına ayrıştırılır. Daha sonra her bir bireye (ya da sadece Pareto olmayan bireylere) ilişkin yapay puan değerleri bulunur (Baskınlık gücü algoritması, Adım-1). İkinci adımda Pareto bireylerin puanı, baskın olduğu Pareto olmayan bireylerin yapay puanlarına göre (4.13)' te verilen formüle göre hesaplanır. Pareto olmayan bireylerin puanı ise, "*bastırıldıkları Pareto bireylerin puanları toplamı + 1*" formülüyle bulunur. Böylece Pareto olmayan bireylerde de bir farklılık (çeşitlilik) sağlanabilir.

$$\text{Pareto Bireyler İçin: } \text{Puan}_{-}P(i) = \frac{yp_{-}PO(j)}{Top_{-}yp_{-}PO + 1} \quad (4.13)$$

$$\text{Pareto Olmayan Bireyler İçin: } Puan_PO(j) = Top_Puan_P(i) + 1 \quad (4.14)$$

(4.13) ve (4.14) ifadelerinde; i : Pareto bireylerin indeksi, j : Pareto olmayan bireylerin indeksi, yp_PO : Pareto bireyin baskın olduğu Pareto olmayan bireylere ilişkin sanal puanlar toplamı, Top_yp_PO : Pareto olmayan tüm bireylere ilişkin sanal puanlar toplamı, $Puan_P$: Pareto bireylerin puanları, $Puan_PO$: Pareto olmayan bireylerin puanıdır. SPEAmod3 ile puanlanan örnek bir topluluk Şekil 4.22’ de gösterilmektedir. Buna göre bireylerin en çok istenilenden en az istenilene doğru sıralaması şöyledir: $D \succ A \succ C \succ B \succ E \succ F \succ G \succ H$.



Şekil 4.22. SPEAmod3 puanlaması: (a) İlk adım: MOGA puanlaması, (b) İkinci adım: sanal puanlar, (c) SPEAmod3 tarafından atanan son puan değerleri

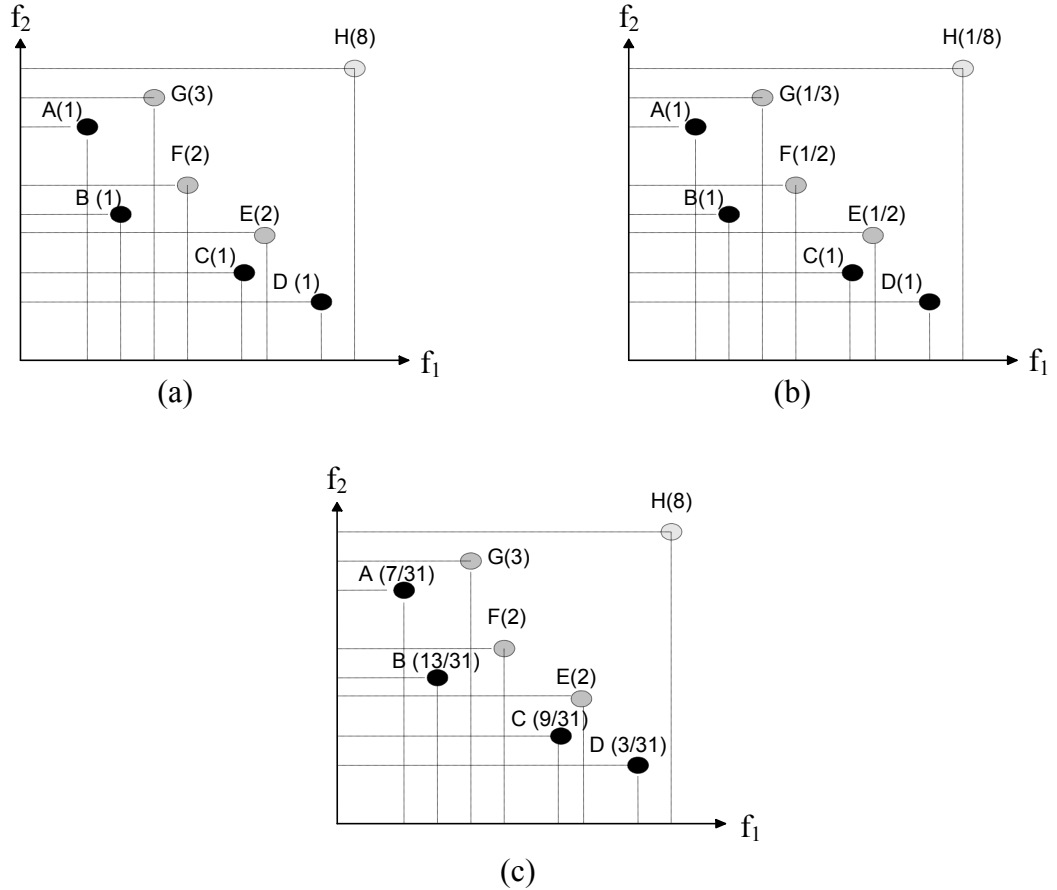
4.2.7. SPEA’ nın Dördüncü İyileştirmesi (SPEAmod4)

Bu yöntem, iyileştirme olarak görünmekle birlikte farklı kavramları da içerisinde barındırmaktadır. Burada, ilk olarak MOGA yöntemi kullanılarak topluluk, alt

topluluklarına ayrıştırılır. Daha sonra her bir bireye (ya da sadece Pareto olmayan bireylere) ilişkin yapay puan değerleri bulunur (Baskınlık gücü algoritması ile). İkinci adımda Pareto bireylerin puanı, baskın olduğu Pareto olmayan bireylerin yapay puanlarına göre aşağıda verilen formüle göre hesaplanır:

$$Puan_P(i) = \frac{yp_PO(j)}{Top_yp_PO + 1} \quad (4.15)$$

Pareto olmayan bireylerin puanı ise MOGA algoritması [Fonseca ve Fleming, 1995] tarafından bulunur. SPEAmod4 ile puanlanan örnek bir topluluk Şekil 4.23' te gösterilmektedir. Buna göre bireylerin en çok istenilenden en az istenilene doğru sıralaması şöyledir: $D \succ A \succ C \succ B \succ E, F \succ G \succ H$.



Şekil 4.23. SPEAmod4 puanlaması: (a) İlk adım: MOGA puanlaması, b) İkinci adım: sanal puanlar, c) SPEAmod4 tarafından atanan son puan değerleri

4.2.8. Benzetim Sonuçları

İyileştirmeler, orjinal yöntemlerle dört farklı test işlevi (ZDT1, ZDT2, ZDT3, ZDT6) kullanılarak karşılaştırılmıştır. Stokastik evrensel örnekleme (SUS), puanla orantılı bir seçme mekanizması olarak tüm yöntemlerine uygulanmıştır. Daha sonra 100 adım sonucunda elde edilen topluluklar kullanılarak, GD ve Δ ölçütleri hesaplanmıştır. Tüm yöntemler ve iyileştirmelerinin 20' şer kez çalıştırılmaları sonrasında elde edilen, yakınsama ölçütlerinin ortalama değerleri Çizelge 4.1' de ve dağılım (yayılım) ölçütlerinin ortalama değerleri ise Çizelge 4.2' de verilmektedir.

Benzetimlerde, tek noktalı çaprazlama ve bit-tabanlı mutasyon uygulanmıştır. Çaprazlama olasılığı $p_c = 0.9$, mutasyon olasılığı $p_m = 1/l$ (burada l dizi uzunluğudur) olarak seçilmiştir. Topluluk boyutu 100 birey, arşiv veya ikincil topluluk boyutu ise 25 seçilmiştir. Her parametrenin kodlanması için 30 bit kullanılmıştır. NSGA yöntemi için paylaşım yarıçapı 0.0158 seçilmiştir.

Çizelge 4.1. Yakınsama ölçütünün (GD) tüm yöntemler ve iyileştirmeleri için tüm test işlevlerinde elde edilen ortalama değerleri

GD	ZDT1	ZDT2	ZDT3	ZDT6
Elitist NSGA	0.0241	0.0535	0.0133	1.2495
Elitist NSGAmo	0.0213	0.0351	0.0129	0.9036
SPEA (orjinal)	0.1256	0.2432	0.1039	1.8338
SPEAmo1	0.0867	0.1050	0.0759	0.5331
SPEAmo2	0.0695	0.2195	0.0750	1.7353
SPEAmo3	0.0527	0.0859	0.0444	0.8190
SPEAmo4	0.0767	0.2168	0.0932	0.7108

Çizelge 4.2. Dağılım ölçütünün (Δ) tüm yöntemler ve iyileştirmeleri için tüm test işlevlerinde elde edilen ortalama değerleri

Δ	ZDT1	ZDT2	ZDT3	ZDT6
Elitist NSGA	0.5930	0.6542	0.6068	0.8990
Elitist NSGAmo	0.5886	0.5543	0.6046	0.8455
SPEA (orjinal)	0.6390	0.7280	0.6507	0.9182
SPEAmo1	0.6113	0.7635	0.6446	0.8239
SPEAmo2	0.6047	0.6537	0.5532	0.8957
SPEAmo3	0.8095	0.7190	0.7586	0.8248
SPEAmo4	0.6640	0.8483	0.7893	0.9155

4.2.9. Genel Sonuçlar ve Değerlendirme

NSGA ve SPEA yöntemlerinin puan atama mekanizmaları, orjinal yöntemlerin ürettiğinden daha iyi ve nitelikli bilgi (puan) üretebilmeleri amacıyla hesapsal ve yapısal olarak değiştirilmiştir (iyileştirilmiştir). İyileştirmeler iki temel biçimde yapılmıştır: **i)** *k*. en yakın komşuluk yöntemiyle üretilen yoğunluk bilgisi eklenmesi, **ii)** baskınlık gücü kavramı.

İyileştirme yapılmasındaki ana amaç; bir topluluktaki bireyler hakkında seçme mekanizmasına daha nitelikli bilgi ulaştırmak, böylece seçme mekanizmasının yeteneğini artırmaktır. Orjinal yöntemler ve iyileştirmeleri dört farklı ve sık kullanılan test işlevi kullanılarak test edilmişlerdir. Önerilen beş iyileştirme (NSGAmođ, SPEAmođ1, SPEAmođ2, SPEAmođ3, SPEAmođ4), orjinal yöntemlere göre Pareto cephesine daha fazla yakınsamakta ve bu cephe üzerinden bireylerin dağılımını daha düzgün yapmaktadır.

Sonuç olarak, literatürde yer alan NSGA ve SPEA yöntemlerinin puan atama (ya da sıralama) yetenekleri önemli derecede iyileştirilmiştir. Ayrıca, *k*. en yakın komşuluk yöntemiyle oluşturulan yoğunluk bilgisi tekniğı, paylaşım tabanlı çeşitlilik tekniğinden daha iyi başarımlar göstermektedir.

Literatürde yer alan tüm yöntemler üzerinde bazı iyileştirmeler yapılması mümkündür. Böylece yöntemlerin yakınsama ve/veya dağılım yetenekleri artırılabilir ve yöntemler daha etkin hale getirilebilir. Bu çalışmada, yalnızca beş adet iyileştirme verilmekle beraber daha çok sayıda iyileştirme yapılması da mümkündür. Yöntemlerin işlem sürelerinin kısaltılması için de iyileştirmeler yapılması, gelecek çalışma alanlarından birisi olacaktır.

4.3. Gama Düzeltmesi ile Puan Ölçekleme Yöntemi (GDPÖ) ve Genetik Algoritmaların Başarımlarına Etkisi

Literatürde, yerel en iyiye hızlı yakınsamanın önüne geçmek için bazı puan ölçekleme yöntemleri önerilmiştir. Bu yöntemler, genel olarak tek amaçlı genetik algoritmalara uygulanmıştır. Bu zamana kadar, bu puan ölçekleme yöntemlerinin ÇAGA' lara uygulanması gözden kaçırılmıştır. Bu bölümde, bir çeşit üstel (power-law)

puan ölçekleme [Gillies, 1985; Goldberg, 1989; Michalewicz, 1994] yöntemi kullanılarak genetik algoritmaların yakınsamasının / yakınsama hızının iyileştirilip iyileştirilemeyeceği incelenmektedir. Burada kullanılan yöntem, **gama (γ) düzeltmesi ile puan ölçekleme (GDPÖ)** olarak adlandırılmıştır. Bu yöntemin etkisi, ilk olarak tek amaçlı GA üzerinde incelenmiştir. Daha sonra ise, GDPÖ sık kullanılan bazı ÇAGA yöntemleri (SPEA ve SPEA2) ve yeni önerilen DOPGA ve DOPGA2 yöntemlerine uygulanarak, etkisi incelenmiştir.

GA' larda puan atama bloğunun çıkışı, seçme mekanizmasının girişini oluşturmaktadır. Seçme mekanizmasının çıkışı da seçme mekanizmasının puan atama mekanizması tarafından nasıl beslendiği ile doğrudan ilişkilidir. Bu yüzden, puan atama yöntemleri seçme mekanizmasının çıkışını etkilemektedir. Eğer seçme mekanizmasından önce bireylerin seçilme baskısı değiştirilebilirse, tek amaçlı GA' larda yakınsama hızı ve ÇAGA yöntemlerinde Pareto cephesine yakınsama artırılabilir.

Puan ölçekleme, GA' ların erken adımlarında seçilme baskısının azaltılması için kullanılır ve böylece tüm arama uzayının taranması ve çeşitliliğin artırılması sağlanabilir. Tersine, son adımlara doğru, puan ölçekleme seçilme baskısını artırır ve böylece en iyi çözüme yakınsama sağlanabilir [Hopgood ve Mierzejewska, 2008]. Literatürde puan ölçeklemenin ÇAGA' lara uygulanması ile ilgili eksikler gözükmemektedir. Bunun nedeni, ÇAGA yöntemlerinin genellikle en gelişmiş (state-of-art) yöntemler olarak görülmesinden kaynaklanmaktadır. Bu sınırlama, basit bir fikir olan puan ölçeklemenin ÇAGA' lara uygulanması önünde psikolojik bir engel oluşturmaktadır. Bu yüzden literatürde bu konuda bir eksiklik söz konusudur.

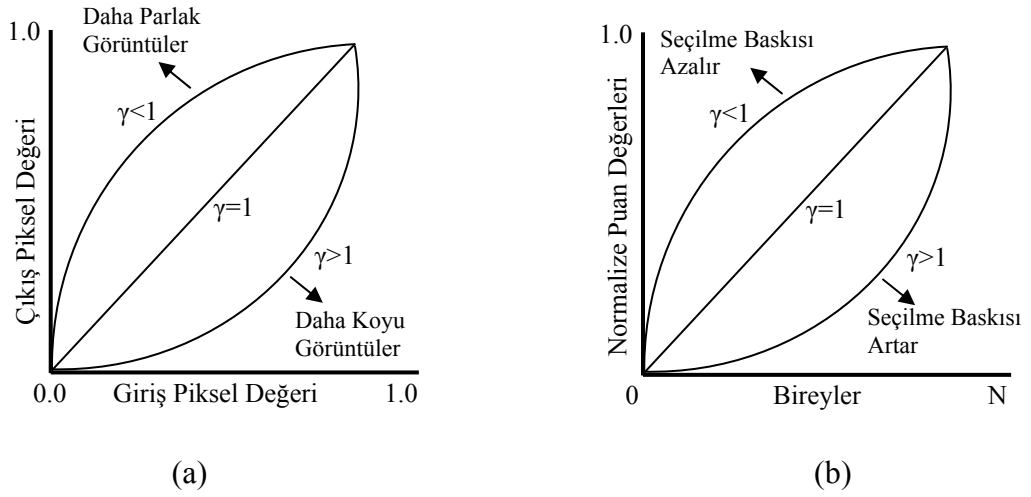
GDPÖ' de bir ÇAGA yöntemi tarafından üretilen puan değerlerinin gama (γ) üssü alınmakta ve topluluktaki bireylerin seçilme baskısı değiştirilebilmektedir. Böylece, en iyi bireylerin seçilme olasılığı artırılabilir ve en çok istenilen ve en az istenilen bireyler, seçme mekanizmasına gönderilmeden önce daha iyi ayırt edilebilir. İlerleyen bölümlerde GDPÖ ayrıntılı olarak açıklanmakta ve benzetim sonuçları verilmektedir.

4.3.1. Gama Düzeltmesi ile Puan Ölçekleme Yöntemi (GDPÖ)

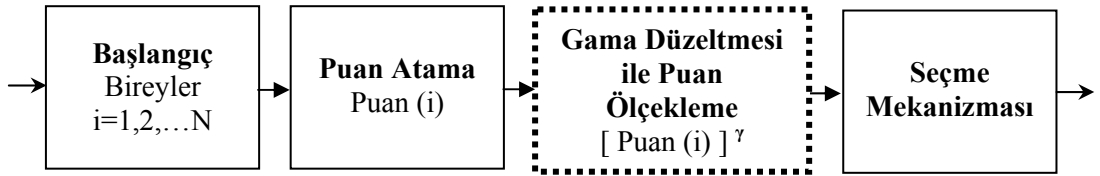
Yerel en iyiye erken yakınsama genetik algoritmalar için büyük bir sorundur. Eğer verilen toplulukta yüksek puana sahip bir *süper birey* bulunuyorsa, onun seçilme

olasılığı yüksek olacaktır ve çeşitlilik erken aşamada kaybedilecektir [Leclerc ve Potvin, 1995]. Sadjadi, ölçeklemenin genetik çeşitlilik ve yakınsamaya çarpıcı bir etkisi olduğu belirtmektedir [Sadjadi, 2004]. Buna göre, puan ölçekleme yöntemleri kullanılarak GA' ların başarımını artırmak mümkündür. Literatürde birçok puan ölçekleme yöntemi önerilmiştir: doğrusal ölçekleme [Goldberg, 1989; Michalewicz, 1994], derece (rank) ölçekleme [Goldberg, 1989], eksponansiyel ölçekleme [Hopfield, 1982; Kirkpatrick ve ark., 1983; Gen ve ark., 1996], sigma ölçekleme [Goldberg, 1989; Michalewicz, 1994], dönüşüm (transform) derecelendirme [Hopgood ve Mierzejewska, 2008], Boltzman ölçekleme [Hopgood, 2001] ve üstel (power-law) ölçekleme [Gillies, 1985; Goldberg, 1989; Michalewicz, 1994]. Bu puan ölçekleme yöntemleri genel olarak tek amaçlı GA' lara uygulanmış ve ÇAGA' lara olan etkisine şimdiye kadar değinilmemiştir. Burada önerilen gama düzeltmesi ile puan ölçekleme yöntemi, ÇAGA yöntemlerine uygulanabilen bir çeşit üstel puan ölçekleme yöntemidir. Sayısal görüntü işlemede kullanılan gama düzeltme işlevine tarihsel ve yapısal benzeşiminden dolayı bu isim verilmiştir. Üstel puan ölçeklemenin bir incelemesi ve SPEA yöntemine SUS seçme mekanizmasıyla uygulamasının etkisi [Ergül ve ark., 2009a] makalesinde rapor edilmiştir.

Bir sayısal görüntünün parlaklığı, gama düzeltmesi ile kalitesi iyileştirilerek değiştirilebilir (Şekil 4.24a) [Gonzales ve Woods, 2008]. Böylece, kullanıcılar için görüntü daha cazip / daha iyi bir duruma getirilir, daha anlaşılır olur. Bir ÇAGA yöntemi tarafından bir topluluktaki bireylere puan atanır, daha sonra bu puan değerleri GDPÖ tarafından değiştirilerek yeni puan değerleri elde edilir ve bu puan değerleri seçme mekanizmasına gönderilir (Şekil 4.25). Temel olarak, GDPÖ doğrusal olmayan basit bir işlemdir ve seçilen gama (γ) değerine bağlı olarak, puan atama bloğundan önce puan değerlerini (ve dolayısıyla topluluğun seçilme baskısını) değiştirir. Eğer gama (γ) değeri 1' den büyükse, seçilme baskısı artar. Eğer gama (γ) değeri 1' den küçükse, seçilme baskısı azalır (Şekil 4.24b). Bu şekilde, en iyi bireylerin seçilebilme olasılığı artırılabilir ve en iyi ile en kötü bireyler arasındaki ayrım seçme mekanizmasından önce daha iyi yapılabilir.



Şekil 4.24. (a) Sayısal görüntü işlemede gama düzeltmesi
(b) Genetik algoritmalarda gama düzeltmesi



Şekil 4.25. Seçme mekanizmasından önce GDPÖ kullanımı

Üstel (power-law) puan ölçekleme bir i bireyinin puanını aşağıdaki formüle göre değiştirir:

$$f_i(\text{yeni}) = (f_i)^\gamma \quad (4.16)$$

Böylece, γ değerine göre en iyi ve en kötü bireylerin arasındaki fark ayarlanır. Leclerc ve Potvin, tek amaçlı genetik algoritmalar için γ üs değerinin dinamik olarak değiştirildiği bir yöntem önermişlerdir [Leclerc ve Potvin, 1995].

Bu tezde önerilen GDPÖ yöntemi ile bir topluluktaki bireylerin puanları Şekil 4.26’ da verilen algoritmaya göre değiştirilir.

Gama Düzeltmesi ile Puan Ölçekleme (GDPÖ)

1) ÇAGA yöntemi tarafından atanan puan değerleri normalize edilir, ($i = 0,1,..., top_boyutu$):

$$Normalize_Puan = Puan(i) / En_Yüksek_Puan \quad (4.17)$$

2) Gama (γ) düzeltme işlevi kullanılarak bireylerin yeni puan değerleri hesaplanır:

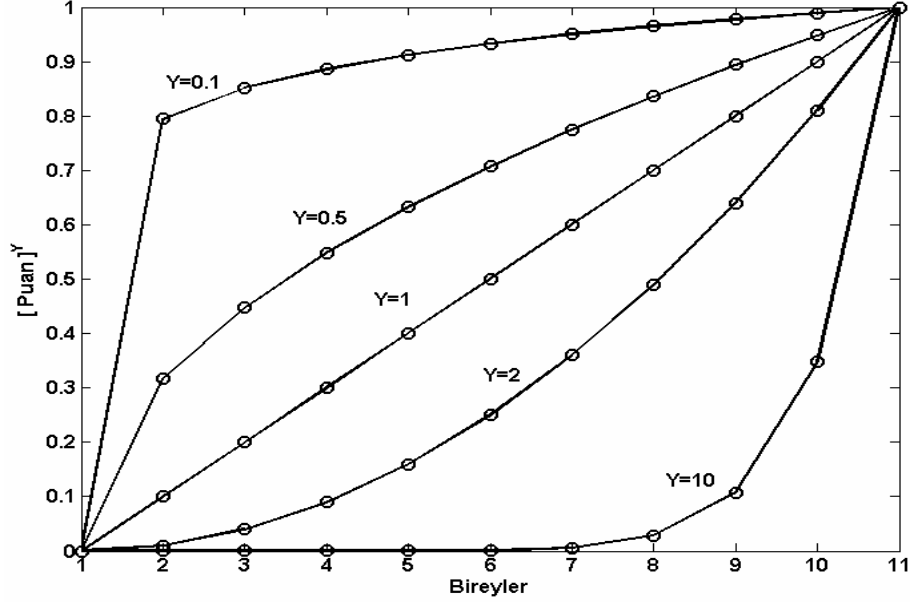
$$Yeni_Puan(i) = Normalize_Puan(i)^\gamma \quad (4.18)$$

3) Yeni puan değerleri seçme mekanizmasına gönderilir.

Şekil 4.26. GDPÖ genel algoritması

4.3.2. Uyarlanabilir (Adaptif) Gama Düzeltmesi ile Puan Ölçekleme (U_GDPÖ) Yöntemi

Puan değerleri seçme mekanizmasının giriş kümesidir ve seçme mekanizması giriş kümesinin içeriğine bağlı olarak bir çıkış kümesi (gelecek nesili oluşturacak ebeveynler) oluşturur. Şekil 4.27, beş farklı puan dağılımını göstermektedir (burada bireyler en önemliden en az önemliye doğru sıralanmışlardır). Bu farklı puan dağılımları seçme mekanizması üzerinde farklı seviyede seçilme baskısı oluşturur. Bir başka deyişle, bu beş farklı giriş kümesiyle beslenen seçme mekanizmasının çıkışı aynı olmayacaktır. Şekil 4.27' de, puan değerleri $\gamma=1$ ile üstel olarak değiştirilirse, dağılım düzgün olur. Eğer $\gamma>1$ ise puan işlevi içbükeydir ve ilk birkaç bireyin puanı birbirlerine çok yakındır, tersine son birkaç bireyin puanı birbirine uzaktır. Benzer olarak, eğer $\gamma<1$ ise puan işlevi dışbükey olur ve ilk birkaç bireyin puanı birbirinden uzak iken, tersine son birkaç bireyin puanı birbirine yakındır.



Şekil 4.27. Beş farklı puan dağılım işlevi

Herhangi bir problem için, hangi γ değerinin iyi olduğu bilinmemektedir. Bunun için GDPÖ' nin uyarlanabilir (adaptif) bir biçimi gerekmektedir, böylece seçilme baskısının bir nesilden diğer nesile dinamik olarak kontrol edilmesi γ değerinin güncellenmesiyle sağlanabilir. Gamma (γ) değeri Şekil 4.28' de verilen uyarlanabilir algoritmaya göre güncellenebilir. (4.19) nolu eşitlikte verilen gama değeri ayrıca düzgün dağılımın bir ölçütüdür.

Uyarlanabilir GDPÖ Algoritması

- 1) Puan değerleri 0 ile 1 arasına normalize edilir,
- 2) Normalize puan (NP) değerleri en önemliden en az önemliye doğru sıralanır,
- 3) Tüm topluluk için, ardışıl bireyler arasındaki normalize puan farkları (NPF) hesaplanır,
- 4) NPF' lerin ortalama değeri ve standart sapması hesaplanır. Gama (γ) değeri (4.19) eşitliğindeki gibi hesaplanır ve nesilden nesile değişir.

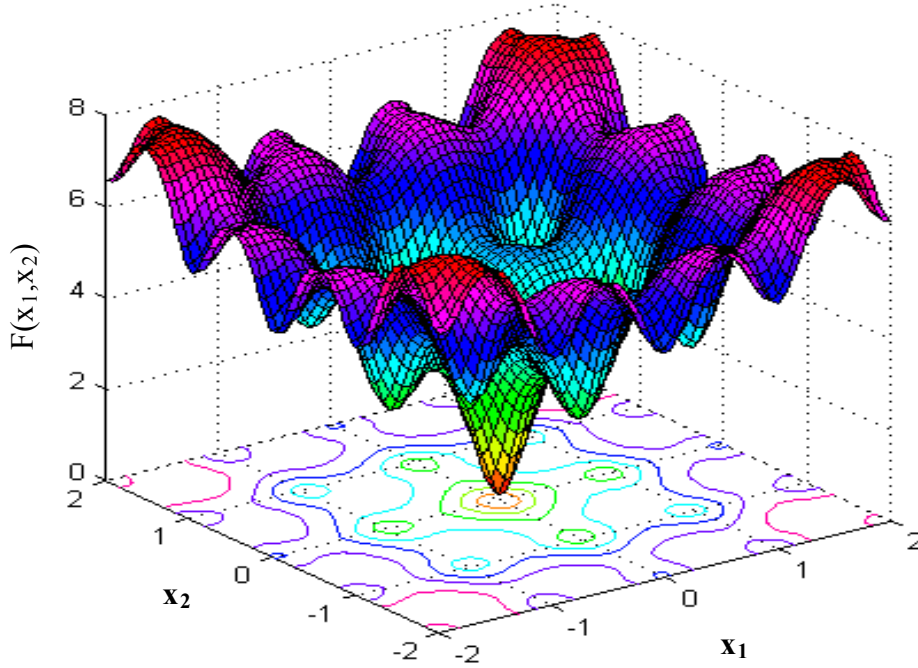
$$Gama(\gamma) = \frac{NPF' \text{ lerin Standart Sapması}}{NPF' \text{ lerin Ortalama Değeri}} \quad (4.19)$$

Şekil 4.28. Uyarlanabilir GDPÖ algoritması

4.3.3. Tek Amaçlı GA' larda GDPÖ' nün Etkisinin İncelenmesi ve Benzetimler

Tek amaçlı genetik algoritmelerde GDPÖ' nün etkisinin belirlenmesi için, iki adet test işlevi kullanılmıştır. Bu test işlevleri, Ackley işlevi ve Rastrigin işlevidir (ikisi de en küçükleme problemidir). Ackley işlevinin en küçük değeri 0, Rastrigin işlevinin en küçük değeri 5' tir. Rastrigin işlevinin matematiksel ifadesi (2.10) eşitliğinde, grafiği ise Şekil 2.12' de verilmektedir. Ackley işlevinin grafiği ise Şekil 4.29' da verilmektedir. Ackley işlevinin matematiksel ifadesi ise şöyledir:

$$F(x_1, x_2) = -20 \cdot \exp\left(-0.2 \cdot \sqrt{\frac{1}{2} \cdot \sum_{i=1}^2 x_i^2}\right) - \exp\left(\frac{1}{2} \cdot \sum_{i=1}^2 \cos(2\pi \cdot x_i)\right) + 20 + e \quad (4.20)$$



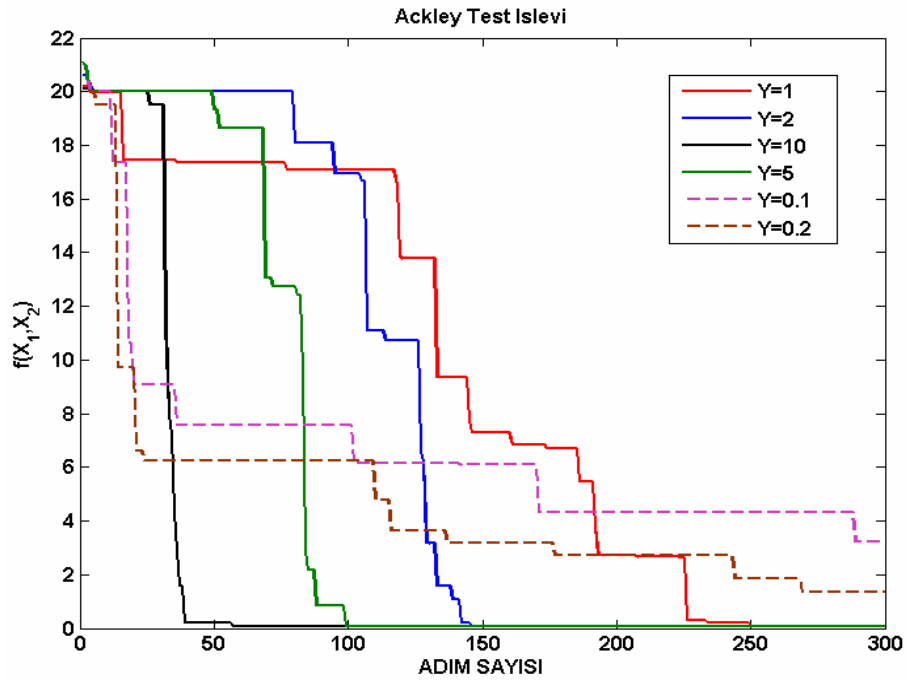
Şekil 4.29. Ackley işlevi ($x_1, x_2 = [-2, 2]$ için)

Bu test işlevlerinde tek amaçlı GA, altı farklı gama değeri (0.1, 0.2, 1, 2, 5 ve 10) kullanılarak GDPÖ ile çalıştırılmıştır. GA parametreleri benzetimlerde şöyle seçilmiştir:

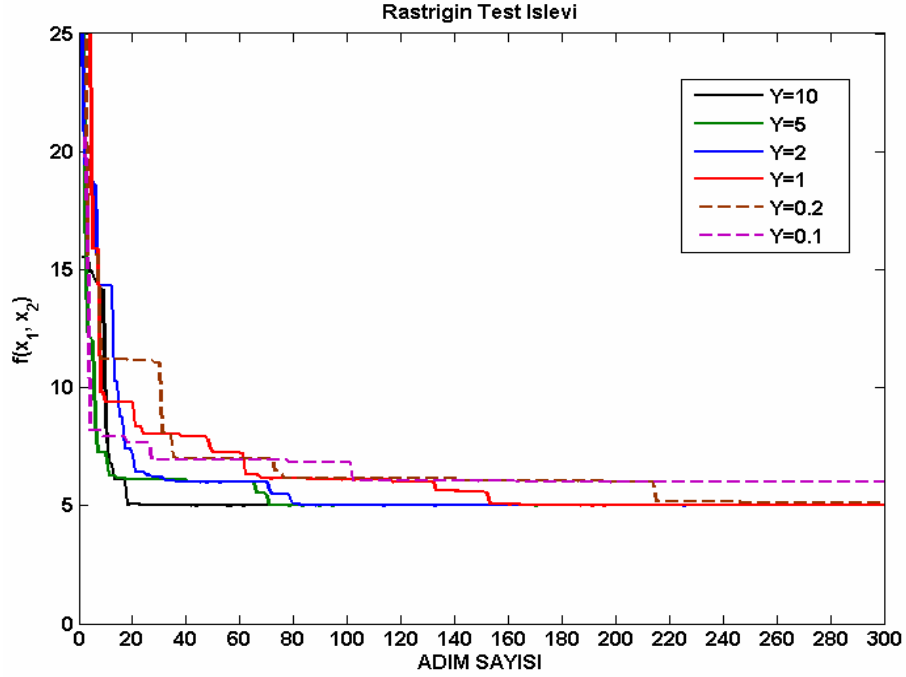
Birey sayısı	: 20
Bit sayısı	: 30
Adım (nesil) sayısı	: 300
Çaprazlama olasılığı	: 1
Mutasyon olasılığı	: 0.1
Parametrelerin alt sınırı (Ackley)	: -500
Parametrelerin üst sınırı (Ackley)	: 500
Parametrelerin alt sınırı (Rastrigin)	: -10
Parametrelerin üst sınırı (Rastrigin)	: 10

Tek amaçlı GA' nın yukarıda verilen parametre ve test işlevleriyle çalıştırılması sonucu elde edilen sonuçlar Şekil 4.30 ve Şekil 4.31' de verilmektedir.

Bu sonuçlardan görüleceği üzere gama değeri artırıldığı zaman GA çözümü daha az adımda ulaşmaktadır. Her iki test işlevinde de GA, $\gamma=10$ için en iyi sonuca daha hızlı yakınsamıştır.



Şekil 4.30. Tek amaçlı GA' da GDPÖ etkisi (Ackley işlevi)



Şekil 4.31. Tek amaçlı GA’ da GDPÖ etkisi (Rastrigin işlevi)

4.3.4. ÇAGA’ larda GDPÖ’ nün Etkisinin İncelenmesi ve Benzetimler

GDPÖ yöntemi, hesapsal olarak sık kullanılan SPEA ve SPEA2 yöntemleri ile önerilen DOPGA ve DOPGA2 yöntemlerine uygulanmıştır. Bu ÇAGA yöntemleri, ilk olarak 12 farklı sabit gama değeri (0.1,0.5,1,2,3,4,5,6,7,8,9 ve 10) kullanılarak dört adet sık kullanılan test işlevi (ZDT1, ZDT2, ZDT3 ve ZDT6, [Zitzler ve ark., 2000]) ile çalıştırılmışlardır. Daha sonra, uyarlanabilir GDPÖ uygulanarak tüm yöntemler aynı test işlevleri kullanılarak çalıştırılmıştır.

SPEA ve SPEA2 yönteminin karşılaştırmalarda kullanılmak üzere seçilmesinin nedenleri şunlardır: *i)* her ikisi de Pareto tabanlıdır, *ii)* her ikisi de harici bir katsayı kullanmadan çeşitlilik bilgisini puanlama mekanizmasının içine gömebilme yeteneğine sahiptir, *iii)* her ikisi de elitisttir, *iv)* her ikisi de son zamanlarda farklı yapay test işlevleri kullanılarak farklı ÇAGA yöntemleriyle karşılaştırılmışlardır. Benzetimlerde, iki temel seçme mekanizması, turnuva seçimi (TS) ve stokastik evrensel örnekleme (SUS) kullanılmıştır ve bu mekanizmaların etkinlikleri test edilip tartışılmıştır.

Daha önceden belirlenmiş bir nesil (adım) sayısı sonrası ($NS=100$ seçilmiştir) elde edilen topluluk kullanılarak yakınsamayı ve dağılımı ölçen ve literatürde sık kullanılan iki adet ölçüt (GD ve Δ) hesaplanmıştır [Deb, 2001; Deb ve ark., 2002b]. İlk ölçüt GD, bilinen bir Pareto cepheye olan yakınsamayı ölçmektedir. Bunun için test işlevlerinin Pareto cephelerine 500 adet düzgün dağılmış birey yerleştirilmiş ve yöntemler tarafından bulunan Pareto bireylerin bu bireylere olan Öklit mesafeleri kullanılarak GD ölçütü hesaplanmıştır. İkinci ölçüt Δ , bulunan sonuçların Pareto cepheye ne kadar düzgün dağıldığını belirlemektedir. Her iki ölçütün de değerinin küçük olması istenilmektedir.

Tüm yöntemler 12 farklı gama değeri ile her bir test işlevi için 20' şer kez çalıştırılmışlardır. Daha sonra, yakınsama ve dağılım ölçütlerinin ortalama değerleri ve varyansları hesaplanmıştır.

4.3.4.1. Sabit Gama Değerleri Kullanıldığında Benzetim Sonuçları

DOPGA ve DOPGA2 yöntemleri, SPEA ve SPEA2 yöntemleriyle dört farklı test işlevi kullanılarak karşılaştırılmıştır. Stokastik evrensel örnekleme (SUS), puanla orantılı bir seçme mekanizması olarak tüm ÇAGA yöntemlerine uygulanmıştır. Daha sonra 100 adım sonucunda elde edilen topluluklar kullanılarak GD ve Δ ölçütleri bulunmuştur. Tüm yöntemler 20' şer kez çalıştırıldıktan sonra, yakınsama için Çizelge 4.3 ve dağılım için Çizelge 4.4' te verilen sonuçlar elde edilmiştir. Sonuçların daha kolay anlaşılması ve yorumlanması için en iyi sonuçlar koyu renkle belirtilmiştir.

Çizelge 4.3' e göre, $\gamma=1$ için DOPGA ve DOPGA2, SPEA ve SPEA2' ye göre Pareto-optimal cepheye daha iyi yakınsamaktadırlar. Daha sonra farklı gama değerleri için benzetimler tekrarlanmıştır. Benzetimlerde, tek noktalı çaprazlama ve bit-tabanlı mutasyon uygulanmıştır. Çaprazlama olasılığı $p_c = 0.9$, mutasyon olasılığı $p_m = 1/l$ (burada l dizi uzunluğudur) olarak seçilmiştir. Topluluk boyutu 100 birey, arşiv veya ikincil topluluk boyutu ise SPEA ve DOPGA için 25, SPEA2 ve DOPGA2 için ise 100 olarak seçilmiştir. Her parametrenin kodlanması için 30 bit kullanılmıştır.

Çizelge 4.3, tüm ÇAGA yöntemleri için farklı gama değerlerinde elde edilen yakınsama ölçüt (GD) sonuçlarının ortalama değerlerini göstermektedir. Çizelge 4.3 ve Şekil 4.32' ye göre, SPEA yöntemi $\gamma=7$ ile tüm test işlevlerinde diğer gama değerlerine göre daha iyi bir yakınsama sağlamaktadır. Benzer olarak, SPEA2 ve DOPGA2 yöntemleri $\gamma=2$ ile tüm test işlevlerinde diğer gama değerlerine göre daha iyi yakınsama sağlamaktadırlar. DOPGA ise $\gamma=3$ ile ZDT2, ZDT3 ve ZDT6 test işlevlerinde, $\gamma=4$ ile ise ZDT1 test işlevinde diğer gama değerlerine göre daha iyi yakınsama sağlamaktadır.

Çizelge 4.4, tüm ÇAGA yöntemleri için farklı gama değerlerinde elde edilen dağılım ölçüt (Δ) sonuçlarının ortalama değerlerini göstermektedir. Çizelge 4.4 ve Şekil 4.33' e göre, SPEA yöntemi $\gamma=7$ ile tüm test işlevlerinde diğer gama değerlerine göre daha düzgün bir dağılım sağlamaktadır. SPEA2, DOPGA ve DOPGA2 yöntemleri ise farklı test işlevlerinde farklı gama değerlerinde iyi dağılım göstermektedirler.

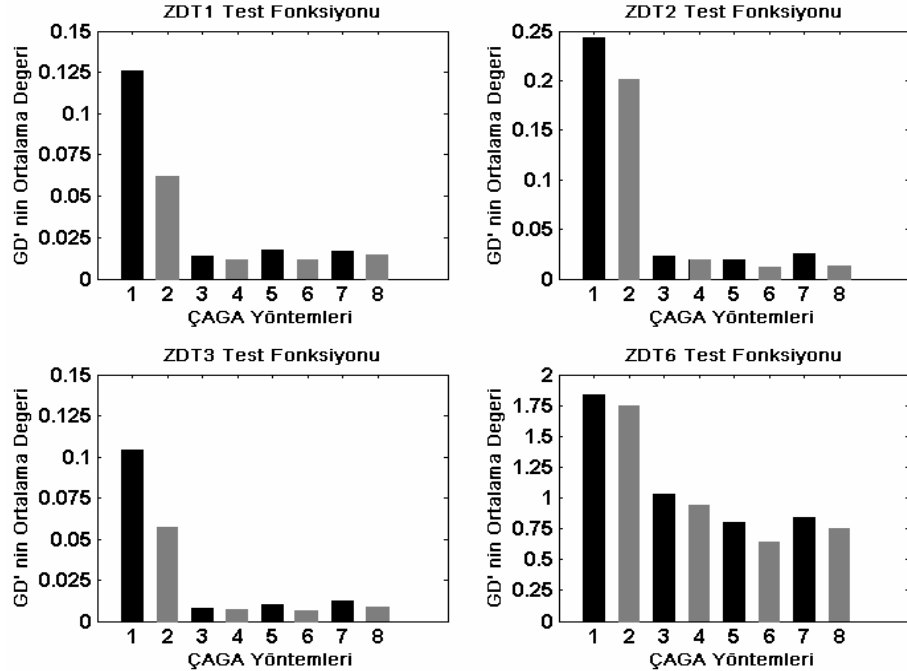
Çizelge 4.3. Yakınsama ölçütünün (GD) tüm ÇAGA yöntemleri için tüm test işlevlerinde elde edilen ortalama değerleri

	GD	$\gamma=0.1$	$\gamma=0.5$	$\gamma=1$	$\gamma=2$	$\gamma=3$	$\gamma=4$	$\gamma=5$	$\gamma=6$	$\gamma=7$	$\gamma=8$	$\gamma=9$	$\gamma=10$
SPEA	ZDT1	0.1409	0.1061	0.1256	0.1034	0.0857	0.0667	0.0612	0.0630	0.0569	0.0624	0.0691	0.0613
	ZDT2	0.3430	0.2768	0.2432	0.2592	0.2824	0.2578	0.2700	0.2885	0.1989	0.2668	0.2348	0.1999
	ZDT3	0.1299	0.1027	0.1039	0.1015	0.0752	0.0728	0.0559	0.0580	0.0554	0.0587	0.0577	0.0563
	ZDT6	1.9760	1.9839	1.8338	1.8747	2.0126	1.8992	2.1623	2.0070	1.7277	2.4558	2.5922	1.7486
SPEA2	ZDT1	0.1163	0.0226	0.0141	0.0110	0.0144	0.0205	0.0285	0.0344	0.0450	0.0482	0.0555	0.0599
	ZDT2	0.3809	0.0461	0.0223	0.0194	0.0267	0.0450	0.0643	0.0887	0.1164	0.1332	0.1611	0.1652
	ZDT3	0.0974	0.0150	0.0081	0.0073	0.0080	0.0124	0.0181	0.0229	0.0360	0.0403	0.0435	0.0498
	ZDT6	2.2259	1.1805	1.0266	0.9406	1.0912	1.3590	1.4712	1.4737	1.7379	1.7874	1.9064	1.8775
DOPGA	ZDT1	0.1968	0.0324	0.0178	0.0114	0.0094	0.0085	0.0103	0.0139	0.0183	0.0248	0.0274	0.0423
	ZDT2	0.4890	0.0531	0.0190	0.0117	0.0112	0.0126	0.0214	0.0279	0.0481	0.0555	0.0846	0.1080
	ZDT3	0.1323	0.0215	0.0099	0.0053	0.0046	0.0051	0.0062	0.0084	0.0126	0.0167	0.0209	0.0278
	ZDT6	2.0913	1.1704	0.8041	0.6391	0.6309	0.7426	0.8146	0.9908	1.1321	1.2200	1.3314	1.3952
DOPGA2	ZDT1	0.1063	0.0217	0.0121	0.0084	0.0132	0.0219	0.0275	0.0364	0.0442	0.0529	0.0551	0.0598
	ZDT2	0.0438	0.0506	0.0176	0.0129	0.0246	0.0442	0.0608	0.0932	0.1215	0.1309	0.1671	0.1727
	ZDT3	0.1046	0.0167	0.0074	0.0049	0.0085	0.0138	0.0183	0.0240	0.0355	0.0418	0.0426	0.0433
	ZDT6	1.9609	1.2431	1.0344	0.8488	1.0331	1.2941	1.5147	1.5527	1.5622	1.6837	1.7429	1.8621

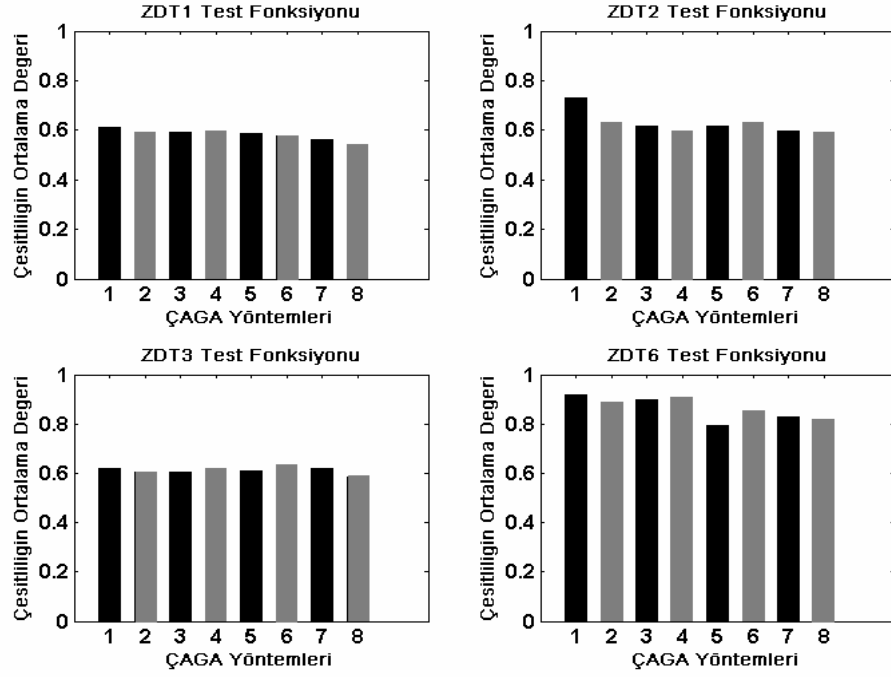
Çizelge 4.4. Dağılım ölçütünün (Δ) tüm ÇAGA yöntemleri için tüm test işlevlerinde elde edilen ortalama değerleri

	Δ	$\gamma=0.1$	$\gamma=0.5$	$\gamma=1$	$\gamma=2$	$\gamma=3$	$\gamma=4$	$\gamma=5$	$\gamma=6$	$\gamma=7$	$\gamma=8$	$\gamma=9$	$\gamma=10$
SPEA	ZDT1	0.6210	0.6391	0.6092	0.6332	0.5756	0.5884	0.6016	0.5808	0.5696	0.5805	0.5758	0.5908
	ZDT2	0.7298	0.7230	0.7280	0.6558	0.6968	0.6803	0.7044	0.6318	0.6187	0.6777	0.6359	0.6281
	ZDT3	0.6793	0.6491	0.6191	0.6550	0.6125	0.6196	0.6104	0.6166	0.6000	0.6074	0.6324	0.6066
	ZDT6	0.9186	0.9131	0.9182	0.9326	0.9119	0.8784	0.8801	0.9099	0.8737	0.9011	0.8976	0.8895
SPEA2	ZDT1	0.5831	0.5983	0.5916	0.5945	0.6073	0.5668	0.5854	0.5665	0.6053	0.5867	0.5761	0.6131
	ZDT2	0.7357	0.5684	0.6167	0.5951	0.6061	0.5974	0.6539	0.6461	0.6657	0.6591	0.6708	0.6367
	ZDT3	0.6063	0.5696	0.6045	0.6172	0.6331	0.6280	0.6193	0.6121	0.5913	0.6021	0.5911	0.5982
	ZDT6	0.9219	0.8749	0.8977	0.9017	0.8630	0.8911	0.9025	0.8957	0.9022	0.8260	0.9365	0.8773
DOPGA	ZDT1	0.5299	0.5573	0.5842	0.5775	0.5923	0.5760	0.5782	0.5684	0.5623	0.5538	0.5701	0.6035
	ZDT2	0.7582	0.5977	0.6142	0.6316	0.6070	0.5757	0.5726	0.5744	0.5752	0.6341	0.6116	0.6503
	ZDT3	0.5705	0.5843	0.6109	0.6358	0.6035	0.5726	0.5850	0.6029	0.5810	0.6223	0.6252	0.5876
	ZDT6	0.8916	0.8216	0.7945	0.8479	0.8299	0.8232	0.8549	0.8454	0.8826	0.8690	0.8586	0.9004
DOPGA2	ZDT1	0.5552	0.6030	0.5970	0.5898	0.6270	0.5707	0.6047	0.5699	0.5874	0.6101	0.5726	0.6099
	ZDT2	0.6342	0.5992	0.5976	0.6248	0.6234	0.5994	0.6268	0.6513	0.6665	0.7027	0.6868	0.6314
	ZDT3	0.5750	0.6044	0.6244	0.6394	0.6499	0.6375	0.6168	0.6084	0.5880	0.5772	0.7196	0.6194
	ZDT6	0.9075	0.8723	0.8839	0.9176	0.8792	0.8821	0.8868	0.9152	0.8889	0.8989	0.8986	0.8891

Tüm ÇAGA yöntemlerinin tüm test işlevlerinde elde edilen yakınsama ve dağılım ölçütlerinin ortalama değerleri bar / sütun grafikleri olarak Şekil 4.32 ve Şekil 4.33' te verilmektedir.

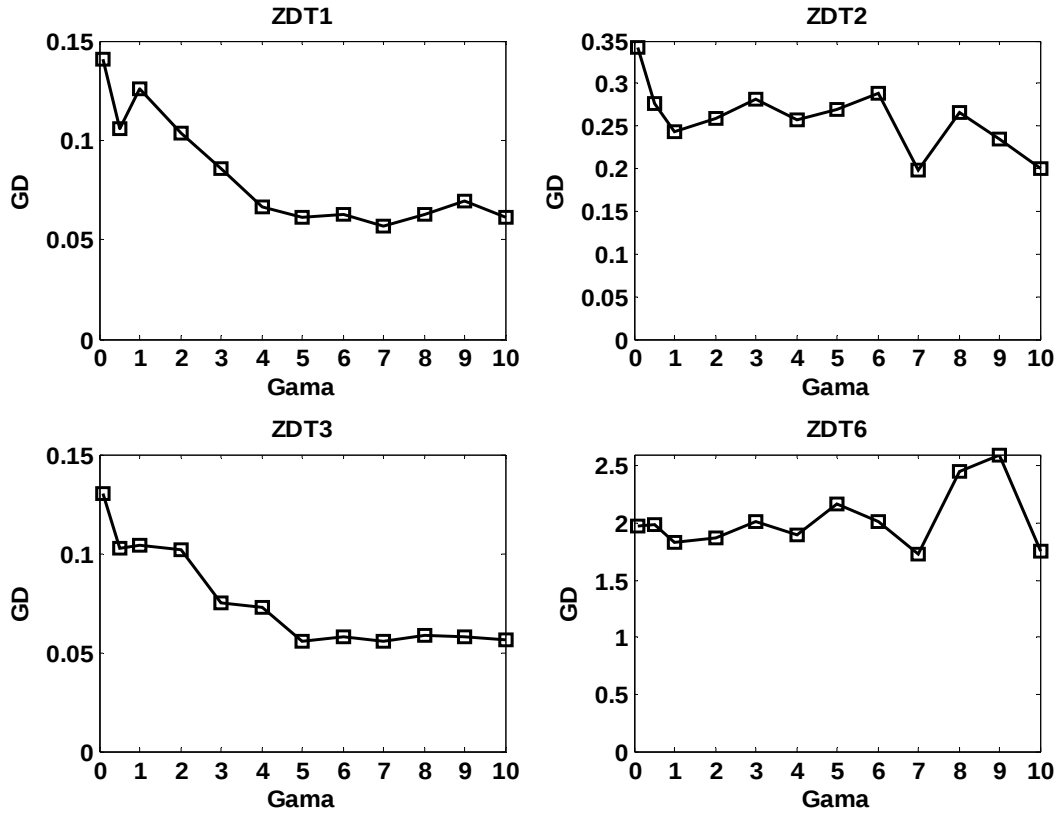


Şekil 4.32. Yakınsama ölçütünün ortalama değerleri. Bu şekilde sayılar şu yöntemleri simgelemektedir: 1-Orjinal SPEA, 2- $\gamma=7$ ile SPEA, 3- Orjinal SPEA2, 4- $\gamma=2$ ile SPEA2, 5- Orjinal DOPGA, 6- $\gamma=3$ ile DOPGA, 7- Orjinal DOPGA2 and 8- $\gamma=2$ ile DOPGA2

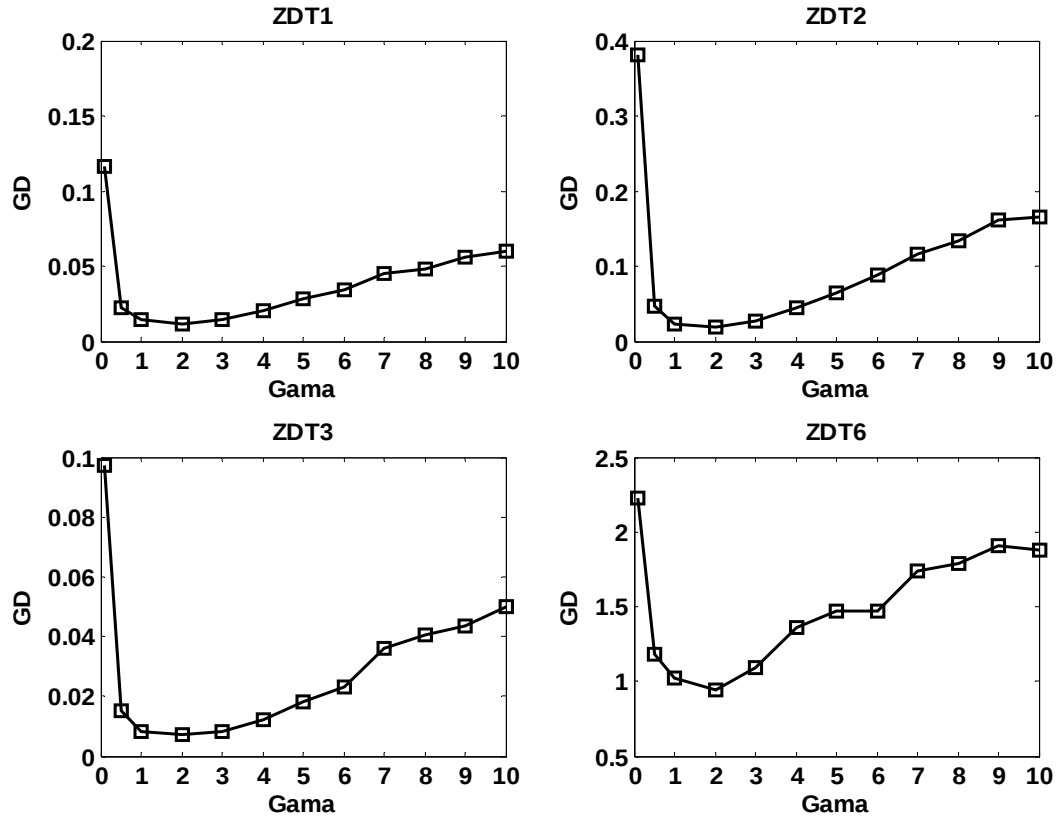


Şekil 4.33. Dağılım ölçütünün ortalama değerleri. Bu şekilde sayılar şu yöntemleri simgelemektedir: **1-**Orjinal SPEA, **2-** $\gamma=7$ ile SPEA, **3-** Orjinal SPEA2, **4-** $\gamma=2$ ile SPEA2, **5-** Orjinal DOPGA, **6-** $\gamma=3$ ile DOPGA, **7-** Orjinal DOPGA2 and **8-** $\gamma=2$ ile DOPGA2

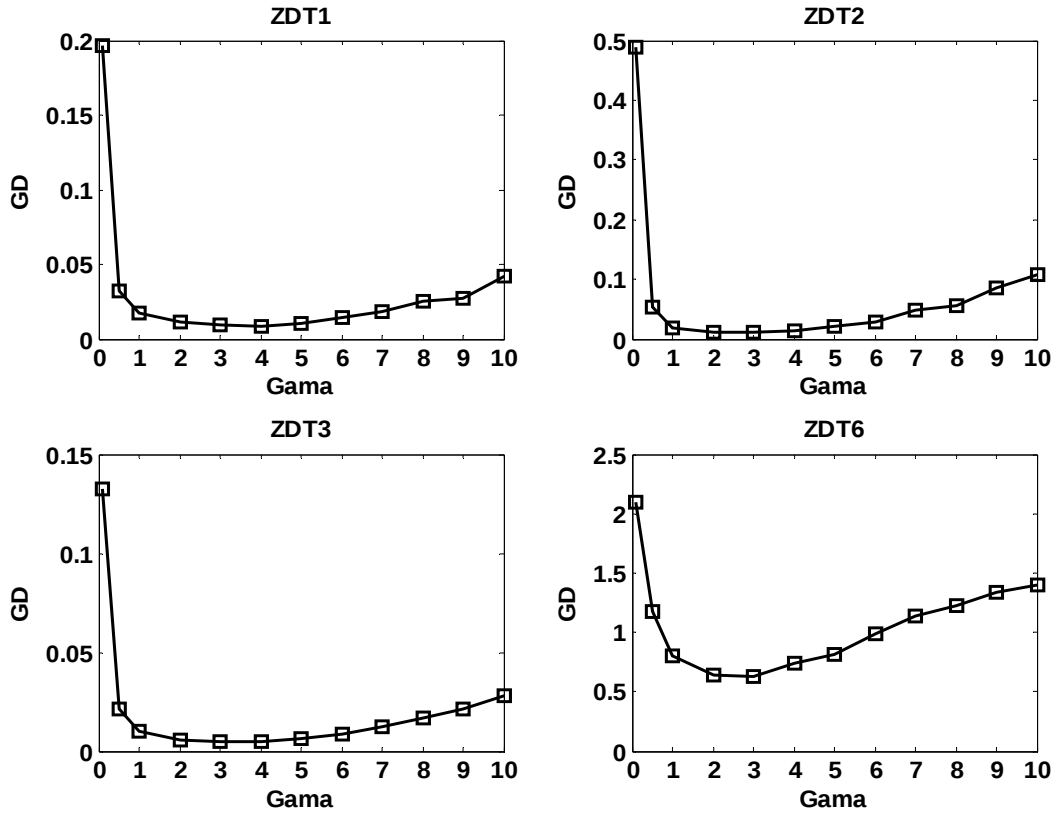
Şekil 4.34, Şekil 4.35, Şekil 4.36 ve Şekil 4.37, gama değerindeki değişime karşılık ÇAGA yöntemlerinin Pareto-optimal cephelere olan yakınsamalarındaki değişimlerin grafiklerini göstermektedir.



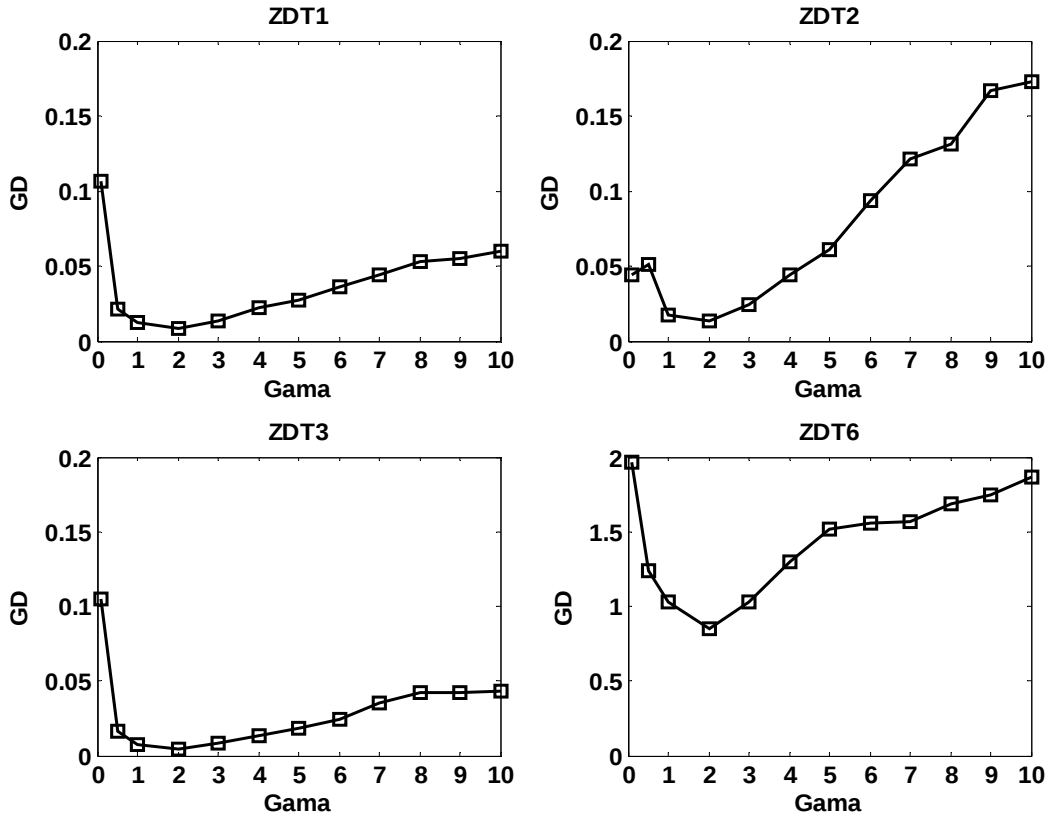
Şekil 4.34. SPEA yöntemi için gama-yakınsama değişimi



Şekil 4.35. SPEA2 yöntemi için gama-yakınsama değişimi

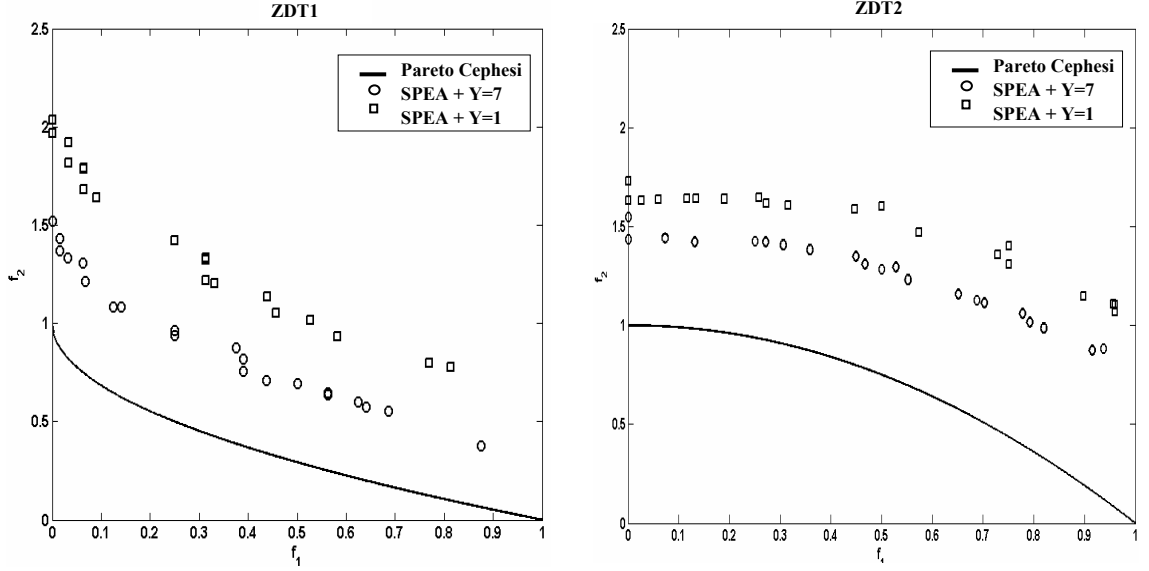


Şekil 4.36. DOPGA yöntemi için gama-yakınsama değişimi

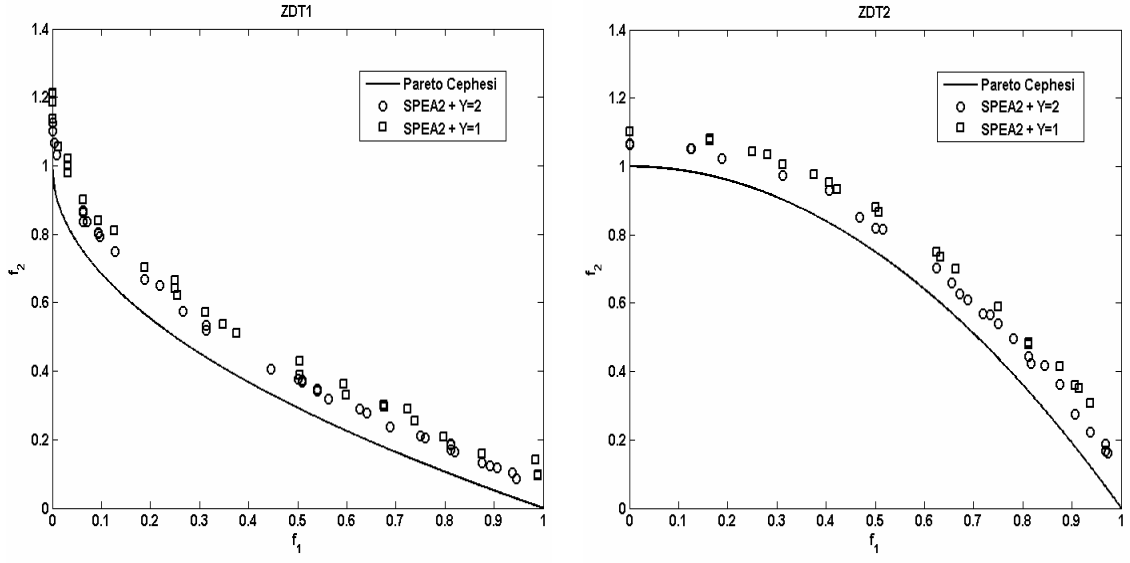


Şekil 4.37. DOPGA2 yöntemi için gama-yakınsama değişimi

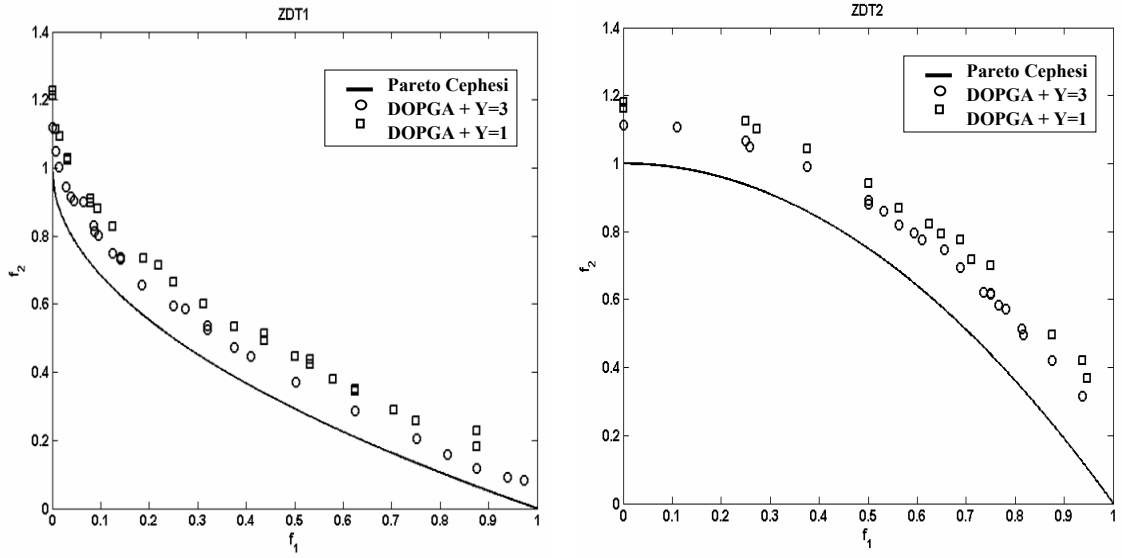
Şekil 4.38, Şekil 4.39, Şekil 4.40 ve Şekil 4.41, 100 adım sonunda ZDT1 ve ZDT2 test işlevleri için, SPEA, SPEA2, DOPGA ve DOPGA2 yöntemlerinin $\gamma=1$ ve en iyi gama değerleri için seçilmiş Pareto bireyleri göstermektedir. Şekil 4.38’ den de açıkça görüleceği üzere, SPEA yöntemi $\gamma=7$ ile $\gamma=1$ ’ e göre daha iyi bir yakınsama sağlamaktadır. Benzer olarak, Şekil 4.39’ da, SPEA2 yöntemi $\gamma=2$ ile $\gamma=1$ ’ e göre daha iyi bir yakınsama sağlamaktadır. Şekil 4.40’ tan da görüleceği üzere, DOPGA yöntemi $\gamma=3$ ile $\gamma=1$ ’ e göre daha iyi bir yakınsama sağlamaktadır. Son olarak, Şekil 4.41’ de DOPGA2 yöntemi $\gamma=2$ ile $\gamma=1$ ’ e göre daha iyi bir yakınsama sağlamaktadır.



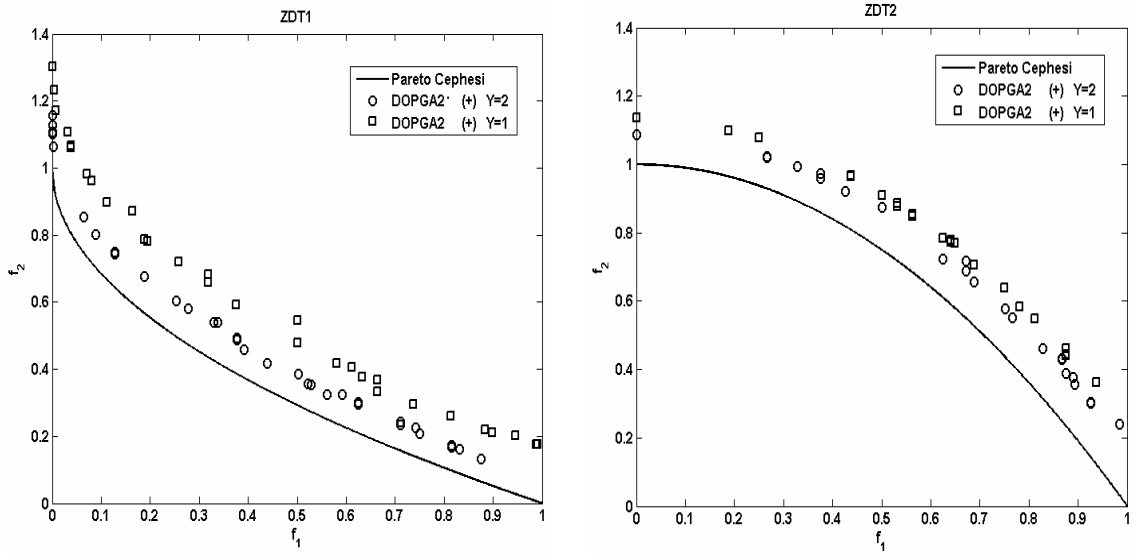
Şekil 4.38. $\gamma=1$ ve $\gamma=7$ için SPEA yöntemiyle ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyler



Şekil 4.39. $\gamma=1$ ve $\gamma=2$ için SPEA2 yöntemiyle ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyler



Şekil 4.40. $\gamma=1$ ve $\gamma=3$ için DOPGA yöntemiyle ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyler



Şekil 4.41. $\gamma=1$ ve $\gamma=2$ için DOPGA2 yöntemiyle ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyler

4.3.4.2. Uyarlanabilir GDPÖ Kullanıldığında Benzetim Sonuçları

Bu bölümde, uyarlanabilir GDPÖ (U_GDPÖ) tüm ÇAGA yöntemlerine uygulanmıştır. 20 çalıştırma sonucunda elde edilen yakınsama ve dağılım ölçütlerinin ortalama değerleri Çizelge 4.5 ve Çizelge 4.6' da verilmektedir. Çizelge 4.5' ten de açıkça görüleceği üzere, DOPGA tüm test işlevlerinde diğer yöntemlerden daha iyi bir yakınsama sağlamaktadır. Dağılım için ise kesin bir şey söylenemektedir.

Çizelge 4.5. U_GDPÖ uygulaması sonucunda hesaplanan yakınsama ölçüt sonuçları

<i>GD- Yakınsama Ölçütü</i>	ZDT1	ZDT2	ZDT3	ZDT6
SPEA + U_GDPÖ	0.0592	0.1871	0.0548	1.6026
SPEA2 + U_GDPÖ	0.0183	0.0277	0.0118	1.1507
DOPGA + U_GDPÖ	0.0093	0.0109	0.0048	0.6189
DOPGA2 + U_GDPÖ	0.0249	0.0414	0.0144	1.1453

Çizelge 4.6. U_GDPÖ uygulaması sonucunda hesaplanan dağılım ölçüt sonuçları

<i>Δ- Dağılım Ölçütü</i>	ZDT1	ZDT2	ZDT3	ZDT6
SPEA + U_GDPÖ	0.5614	0.6464	0.6540	0.8736
SPEA2 + U_GDPÖ	0.6121	0.6477	0.6303	0.8917
DOPGA + U_GDPÖ	0.5764	0.5700	0.6181	0.8053
DOPGA2 + U_GDPÖ	0.6010	0.6125	0.6115	0.8820

4.3.4.3. Turnuva Seçimi (TS) Kullanıldığında Benzetim Sonuçları

Bu bölümde SPEA ve DOPGA yöntemleri GDPÖ ile turnuva seçimi (ikili ve dörtlü turnuva) kullanılarak 20' şer kez çalıştırılmışlardır. Çizelge 4.7, seçme mekanizması olarak turnuva seçimi kullanıldığında beş farklı gama değeriyle SPEA ve DOPGA yöntemlerinin yakınsama ölçütünün ortalama değerlerini göstermektedir. Çizelge 4.8 ise, seçme mekanizması olarak turnuva seçimi kullanıldığında beş farklı gama değeriyle SPEA ve DOPGA yöntemlerinin dağılım ölçütünün ortalama değerlerini göstermektedir. Benzetim sonuçlarına göre, turnuva seçimi kullanıldığında GDPÖ etkili değildir. Çünkü, turnuva seçimi birey puanlarıyla orantılı olarak seçme yapan bir seçme mekanizması değildir ve turnuvaya giren bireylerden yalnızca en iyisini seçmektedir. Farklı gama değerlerinde yöntemlerin Pareto cephesine yakınsamaları ve dağılımları arasında çok büyük bir fark görülmemiştir.

Çizelge 4.7. TS-2 / TS-4 için yakınsama ölçüt sonuçları

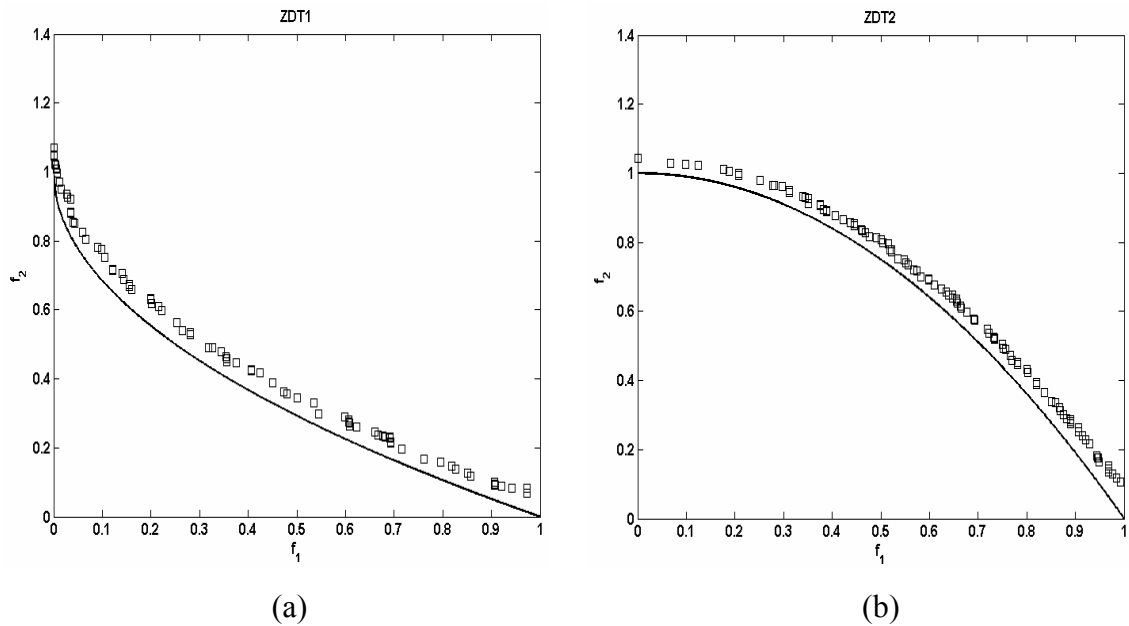
	TS	$\gamma=10$	$\gamma=2$	$\gamma=1$	$\gamma=0.5$	$\gamma=0.1$
SPEA	ZDT1	0.0638 / 0.0648	0.0649 / 0.0651	0.0646 / 0.0652	0.0655 / 0.0708	0.0673 / 0.0693
	ZDT2	0.2230 / 0.2305	0.2318 / 0.2360	0.2264 / 0.2329	0.2306 / 0.2444	0.2642 / 0.2352
	ZDT3	0.0458 / 0.0530	0.0504 / 0.0535	0.0503 / 0.0565	0.0517 / 0.0538	0.0502 / 0.0545
	ZDT6	2.4133 / 1.8219	2.4663 / 2.0009	2.4561 / 2.1569	2.4371 / 2.1423	2.6439 / 2.1117
DOPGA	ZDT1	0.0220 / 0.0359	0.0210 / 0.0315	0.0218 / 0.0330	0.0236 / 0.0343	0.0239 / 0.0329
	ZDT2	0.0519 / 0.0919	0.0505 / 0.0709	0.0508 / 0.0730	0.0536 / 0.0719	0.0554 / 0.0823
	ZDT3	0.0153 / 0.0239	0.0137 / 0.0235	0.0144 / 0.0262	0.0142 / 0.0247	0.0169 / 0.0258
	ZDT6	1.1331 / 1.3783	1.1264 / 1.3397	1.1272 / 1.3856	1.1661 / 1.3670	1.1877 / 1.3529

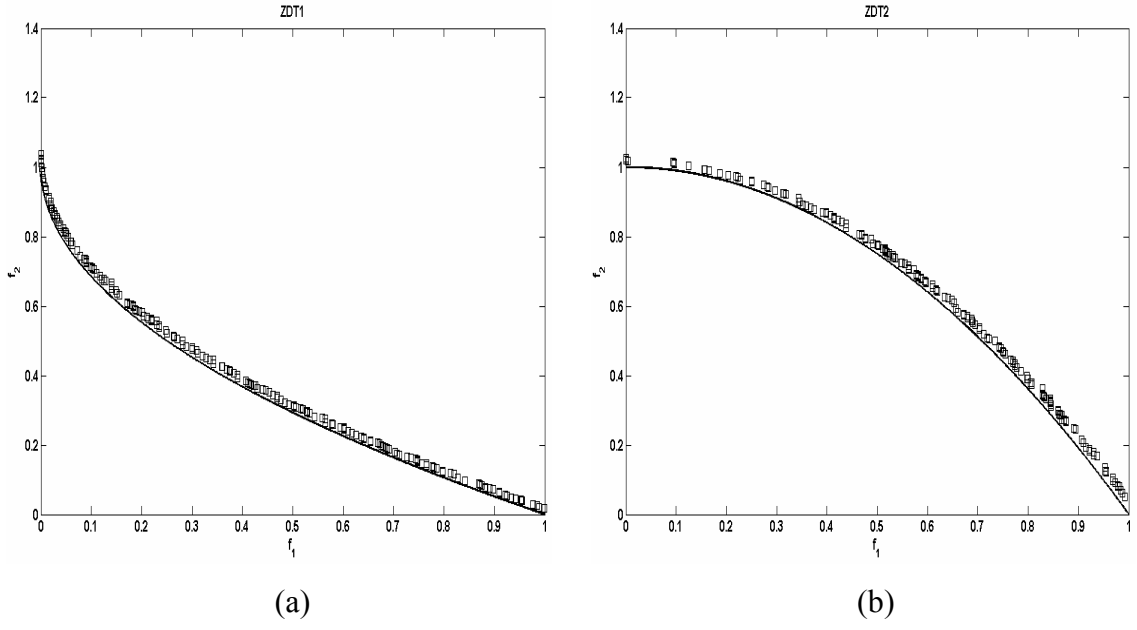
Çizelge 4.8. TS-2 / TS-4 için dağılım ölçüt sonuçları

	TS	$\gamma=10$	$\gamma=2$	$\gamma=1$	$\gamma=0.5$	$\gamma=0.1$
SPEA	ZDT1	0.6120 / 0.6184	0.6103 / 0.6047	0.6023 / 0.5990	0.6085 / 0.6302	0.6168 / 0.6145
	ZDT2	0.6975 / 0.6608	0.6993 / 0.7046	0.6995 / 0.6565	0.6851 / 0.6316	0.6946 / 0.6980
	ZDT3	0.5889 / 0.5922	0.5904 / 0.6082	0.6034 / 0.6225	0.5995 / 0.6019	0.5947 / 0.6026
	ZDT6	0.9076 / 0.9075	0.8794 / 0.8853	0.9114 / 0.9191	0.8804 / 0.9363	0.9201 / 0.9058
DOPGA	ZDT1	0.5885 / 0.5804	0.6016 / 0.6054	0.5934 / 0.5664	0.5985 / 0.6008	0.5817 / 0.5573
	ZDT2	0.6133 / 0.6162	0.6178 / 0.6156	0.6095 / 0.6035	0.6203 / 0.6082	0.5833 / 0.5558
	ZDT3	0.6110 / 0.6062	0.6147 / 0.6202	0.6397 / 0.5971	0.6078 / 0.5787	0.6321 / 0.5728
	ZDT6	0.9018 / 0.9026	0.8662 / 0.8890	0.8957 / 0.8658	0.8558 / 0.8532	0.8955 / 0.8511

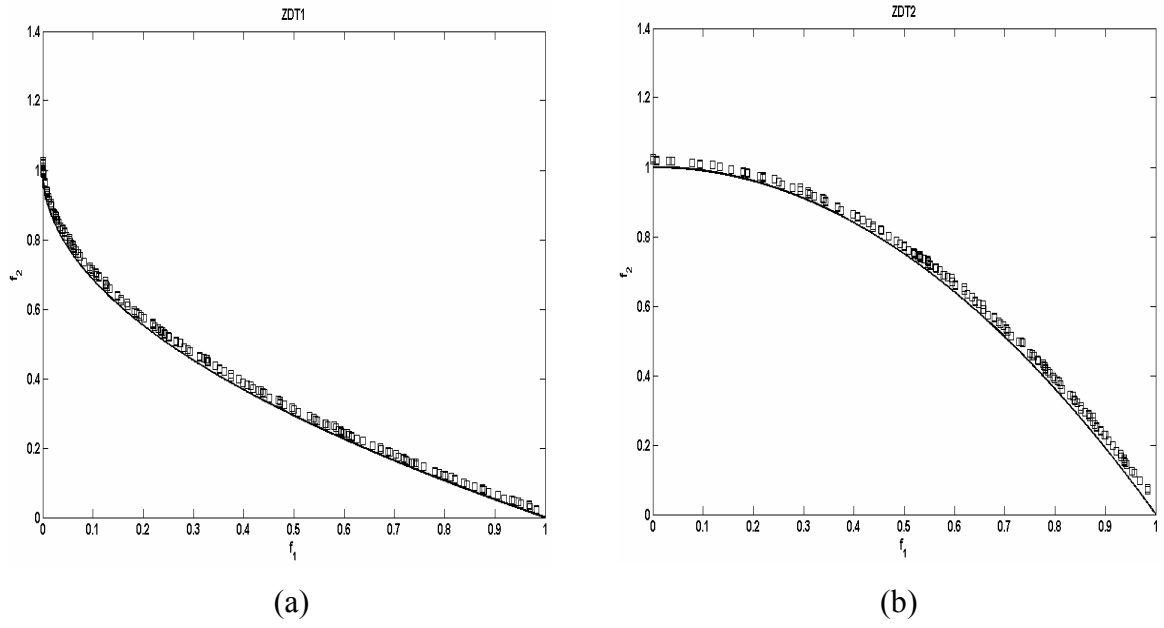
4.3.4.4. 300 Adım Çalıştırılma ile Elde Edilen Sonuçlar

Ölçütler kullanılmadan tüm yöntemlerin temel yeteneklerinin görsel olarak görülebilmesi için, tüm yöntemler sabit gamalı GDPÖ ve uyarlanabilir GDPÖ ile 300 adım çalıştırılmışlardır. Benzetimlerde, GA parametreleri daha önceki benzetimlerde kullanılanlarla aynı seçilmiştir. Şekil 4.42 – Şekil 4.49 arası; en iyi gama değerleri ve uyarlanabilir GDPÖ kullanıldığı durumlarda SPEA, SPEA2, DOPGA ve DOPGA2 yöntemleri tarafından ZDT1 ve ZDT2 test işlevlerinde bulunan Pareto bireyleri göstermektedirler.

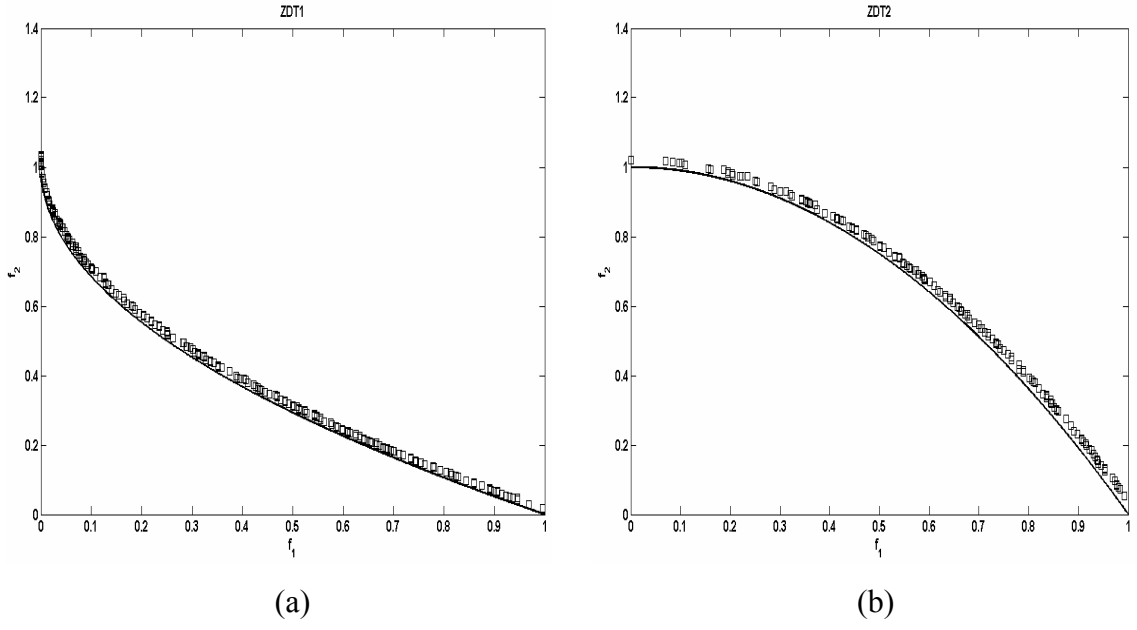
**Şekil 4.42.** $\gamma=7$ ile SPEA tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2



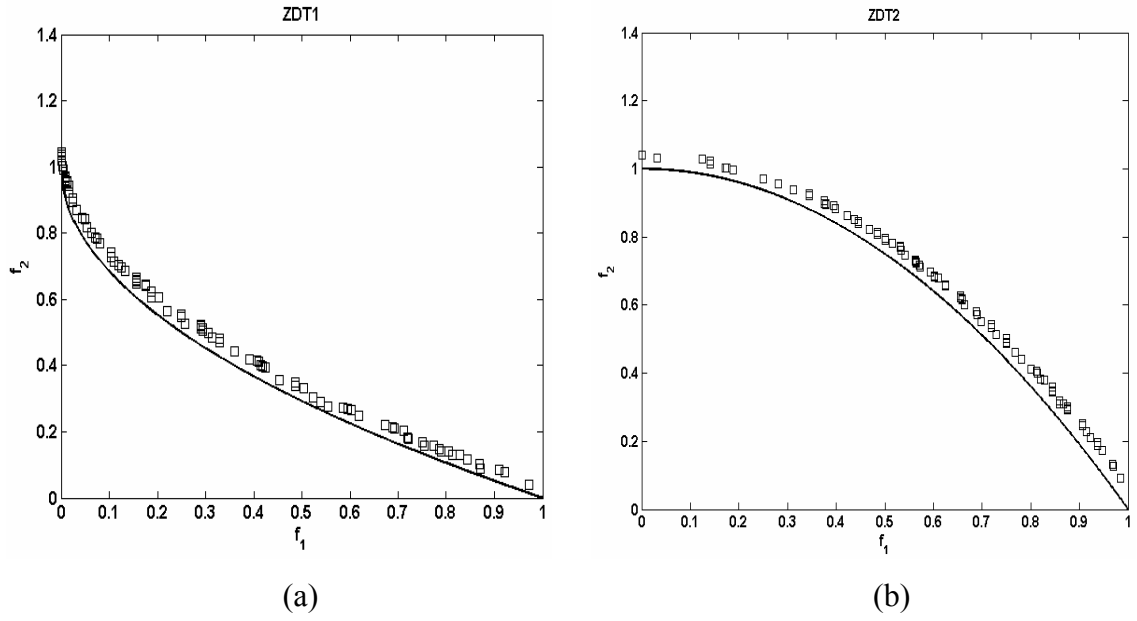
Şekil 4.43. $\gamma=2$ ile SPEA2 tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2



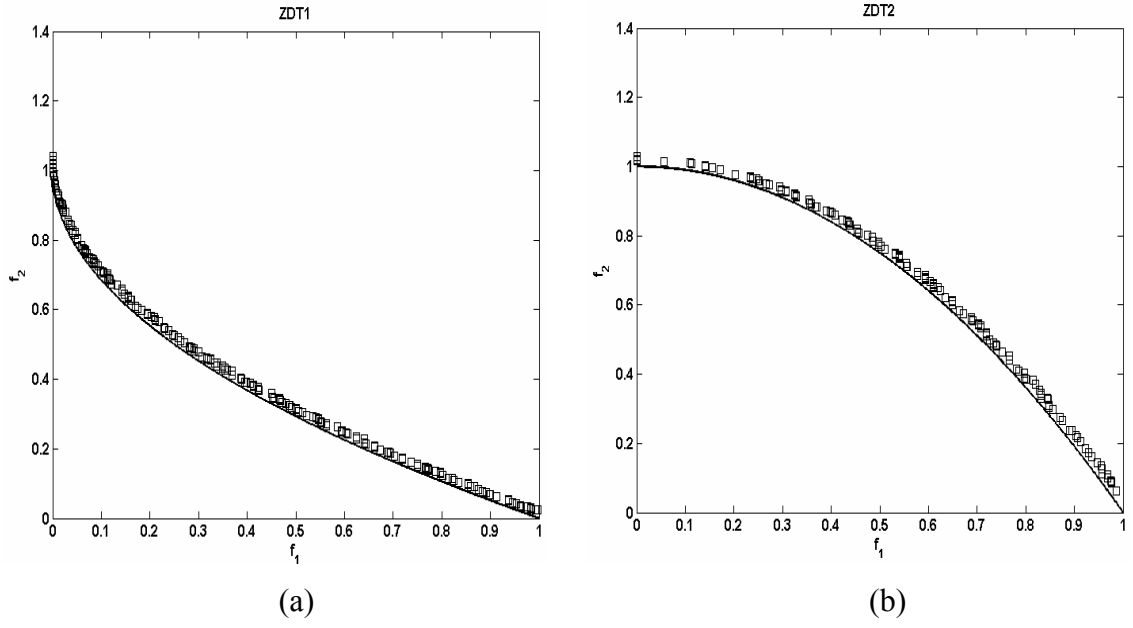
Şekil 4.44. $\gamma=3$ ile DOPGA tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2



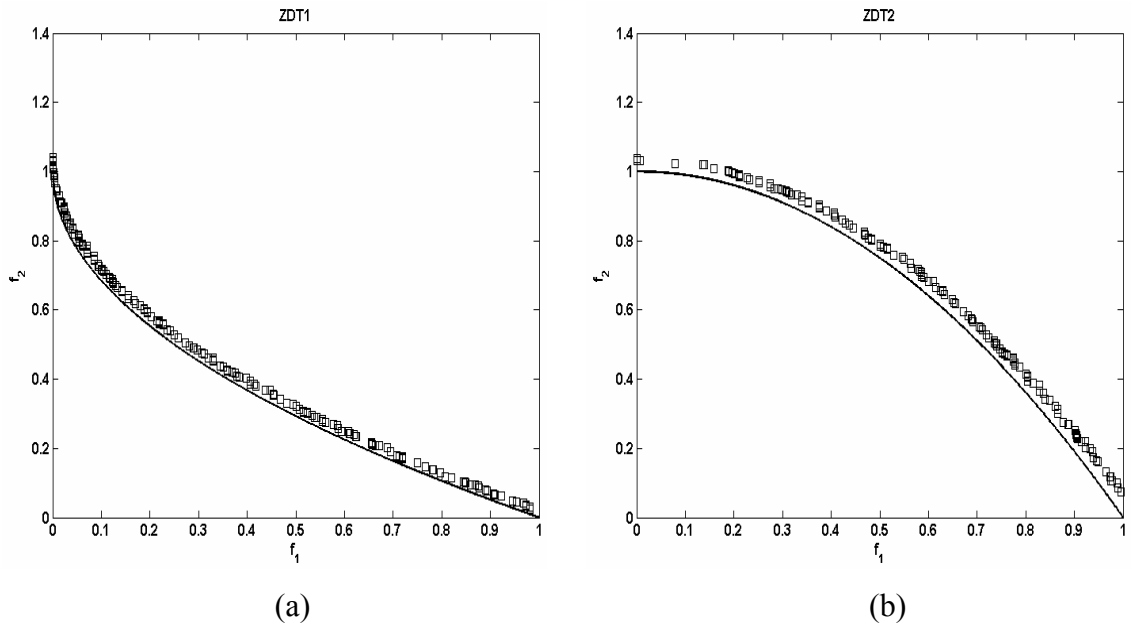
Şekil 4.45. $\gamma=2$ ile DOPGA2 tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2



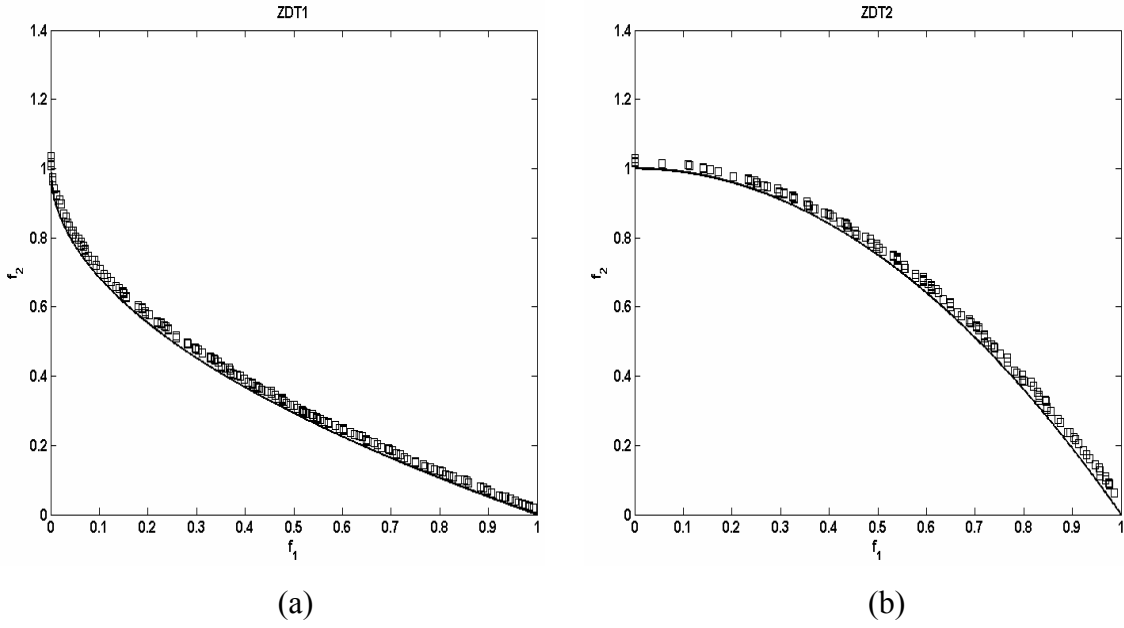
Şekil 4.46. U_GDPÖ ile SPEA tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2



Şekil 4.47. U_GDPÖ ile SPEA2 tarafından bulunan Pareto bireyler: (a)ZDT1, (b)ZDT2



Şekil 4.48. U_GDPÖ ile DOPGA tarafından bulunan Pareto bireyler: (a) ZDT1, (b) ZDT2



Şekil 4.49.U_GDPÖ ile DOPGA2 tarafından bulunan Pareto bireyler:a) ZDT1,b) ZDT2

4.3.5. Genel Sonuçlar ve Değerlendirme

GDPÖ ilk olarak, tek amaçlı GA üzerinde iki test işlevi kullanılarak çalıştırılmıştır. GDPÖ’ de altı farklı sabit gama değeri (0.1,0.2,1,2,5,10) kullanılmıştır. Sonuç olarak, kullanılan iki test işlevinde de $\gamma=10$ seçildiğinde daha iyi ve daha hızlı bir yakınsama sağlandığı görülmüştür.

Genel benzetim sonuçlarına göre DOPGA ve DOPGA2 yöntemleri, seçme mekanizması olarak SUS veya TS kullanılırsa dört test işlevinde de SPEA ve SPEA2 yöntemlerinden iyi sonuçlar vermektedirler.

GDPÖ yönteminin bir topluluktaki bireylerin seçilme baskısının değiştirebileceği dolayısıyla ÇAGA yöntemlerinin yakınsama yeteneğini artırabileceği görülmüştür. Sabit gama (γ) değerleri kullanılarak GDPÖ yöntemi, mevcut ve önerilen ÇAGA yöntemlerine uygulanmıştır. Bu durumda elde edilen genel sonuçlar şöyledir: i) ÇAGA yöntemlerinin başarımı gama (γ) değerine bağlı olarak değişmektedir, ii) SPEA yöntemi $\gamma=7$ ile uygulanan başarımlar ölçütlerine göre en iyi yakınsamayı ve dağılımı sağlamaktadır, iii) $\gamma=2$ ile SPEA2 ve DOPGA2, $\gamma=3$ ile DOPGA (ZDT1’ de $\gamma=4$ ile) en iyi yakınsamayı sağlamaktadırlar, iv) $\gamma=1$ ile yani orjinal sonuçlarla tüm yöntemler değerlendirildiğinde, DOPGA2 yönteminin ZDT1, ZDT2 ve ZDT3 test işlevlerinde,

DOPGA' nın ise ZDT6 test işlevinde en iyi sonuçları verdiği görülmüştür, v) tüm gama değerlerine göre değerlendirme yapıldığında ise, ZDT1 test işlevinde $\gamma=2$ ile DOPGA2; ZDT2, ZDT3 ve ZDT6 test işlevlerinde ise $\gamma=3$ ile DOPGA tüm yöntemler arasında en iyi sonuçları vermektedir, vi) dağılım (veya çeşitlilik) farklı gama değerlerinde önemli derecede değişmemektedir, vii) GDPÖ yalnızca puan orantılı bir seçme mekanizması (örneğin SUS) kullanıldığı zaman etkilidir, viii) TS kullanıldığı zaman GDPÖ etkili değildir.

Uyarlanabilir GDPÖ yöntemi (dinamik γ değerleri) kullanıldığında SPEA (ZDT1 hariç) ve DOPGA (ZDT1 ve ZDT3 hariç) yöntemleri sabit gama kullanıldığında elde edilen sonuçlardan daha iyi sonuçlar vermiştir. SPEA2 ve DOPGA2 ise sabit durumdakinden iyi sonuç vermemişlerdir. Hem uyarlanabilir hem de sabit GDPÖ sonuçları dikkate alındığında, ZDT2 ve ZDT6 test işlevlerinde DOPGA + Adaptif GDPÖ, ZDT1 test işlevinde DOPGA2, ZDT3 test işlevinde ise DOPGA; tüm yöntemlerden daha iyi yakınsama sonuçları vermektedir.

Sonuç olarak, GDPÖ bloğu, herhangi bir ek işlem yükü getirmeden ve yöntemlerin yapılarını değiştirmeden literatürde yer alan tüm ÇAGA yöntemlerine uygulanabilir ve yöntemlerin yakınsama başarımı artırılabilir. Ayrıca, benzetim sonuçlarından da anlaşılaçağı üzere, tüm ÇAGA yöntemleri hala iyileştirmelere açıktırlar.

4.4. Çok Amaçlı Genetik Algoritmalar İçin Etkin Elitizm Mekanizması

Önerisi

GA literatüründe yer alan klasik veya pasif elitizm mekanizması, sakla/aktar bir yapıya sahiptir. Tek amaçlı GA' larda her adımda yalnızca bir tek en iyi olduğu için elitizm mekanizması basittir ve klasik elitizm şöyle uygulanır: yalnızca topluluktaki en iyi birey bir sonraki topluluğa (nesile) aktarılır. Buna rağmen, ÇAGA' larda elitizm mekanizması tek amaçlı GA' larda olduğundan daha karmaşıktır, ancak temel yapı aynıdır. Bir tek en iyi bireyin yerine, bir grup en iyi birey seçilir ve bunlar arşiv veya ikincil topluluk olarak adlandırılan bir toplulukta saklanır ve belirli kurallar çerçevesinde bu bireylerin hepsi veya seçilen bir kısmı sonraki nesile aktarılır. İkincil topluluk / arşiv yönetimi (veya genel olarak elitizm) ÇAGA' ların başarımını önemli

ölçüde etkilemektedir, ancak bu durum çok iyi kontrol edilmelidir. Bazı ÇAGA' larda arşiv boyutu önceden tanımlanır ve yalnızca belirli sayıda elit birey arşivde saklanabilir. Arşive bazı durumlarda, elit bireylerin yanında elit olmayan bireyler de dahil edilebilir. Arşivde saklanan bireylerin çeşitliliği de önemli bir konudur, eğer arşiv birbirinin aynı veya birbirine yakın elit bireylerden oluşturulursa, çeşitlilik azalır. Klasik elitizm mekanizmasının iki önemli aşaması vardır:

i) Saklama Aşaması: *Hangi bireyler arşivde saklanacak ?* Bu aşamada, ilk (veya ana) topluluktaki Pareto (bastırılmayan) bireyler arşive gönderilir ve orada saklanır. Eğer arşiv boyutuna ulaşamazsa, bazı Pareto olmayan bireyler arşive kopyalanır. Eğer arşiv boyutu aşılsa, o zaman bazı elit bireyler arşivden silinir.

ii) Geri Gönderme Aşaması: *Hangi bireyler ilk topluluğa geri gönderilecek?* Stratejilerden bir tanesi, o andaki topluluktaki tüm elit bireylerin bir sonraki topluluğa eklenmesidir. Bir diğer strateji ise, yalnızca belli bir sayıda elit bireyin bir sonraki topluluğa gönderilmesidir [Zitzler, 1999]. Seçme işlemi, yalnızca arşivden veya ana topluluk + arşivden yapılabilir.

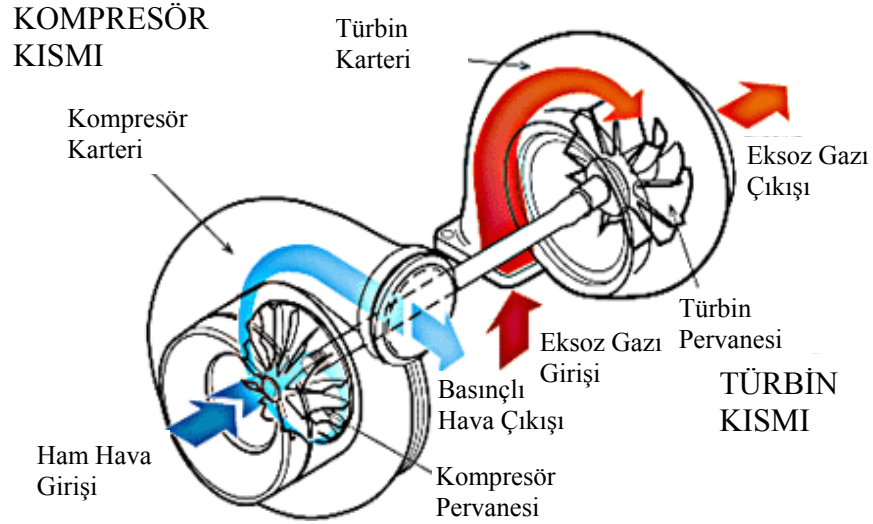
Literatürde birçok elitist ve elitist olmayan ÇAGA yöntemi önerilmiştir. Elitizmin ÇAGA yöntemlerinin başarımını artıran etkili bir yol olduğu deneysel olarak ispatlanmıştır. [Zitzler, 1999; Zitzler ve Thiele, 1999]' de iki farklı ÇAGA yönteminin elitist versiyonlarının daha iyi başarımlar gösterdiği belirtilmiştir. Rudolph, elitizm olduğunda bazı test işlevlerinde GA' ların küresel en iyiye daha iyi yakınsadığını ispat etmiştir [Rudolph, 1996; 2001]. Ayrıca, elit bireylerin varlığı daha iyi nesillerin üretilme olasılığını artırmaktadır.

Yeni elit bireylerin arşive eklenmesi ve bastırılan elit bireylerin arşivden silinmesi işlemine *arşiv güncellenmesi* adı verilir. ÇAGA yöntemlerinde birçok arşiv güncelleme yöntemi mevcuttur [Osyczka ve Kundu, 1995; Zitzler, 1999; Veldhuizen, 1999; Zitzler ve ark., 2001; Deb ve ark., 2002b]. Sonuç olarak şu söylenebilir; tüm klasik elitizm yöntemlerinin ortak noktası, elit bireylerin hiçbir uyartım olmadan arşivde saklanması ve daha sonra ana topluluğa aktarılmasıdır (elit saklama yordamı / yöntemi).

4.4.1. Etkin Elitizm Mekanizması (Effective Elitism Mechanism, EFE)

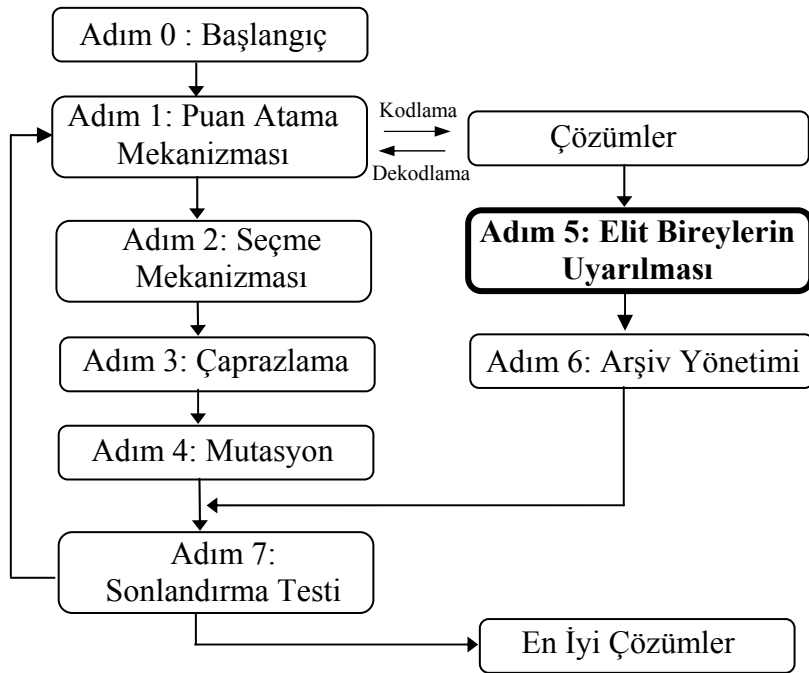
Usta, arşivdeki bir grup elit bireyi kendilerinden daha iyi ve çeşitlilik açısından farklı bireyler üretilmesi umuduyla genetik işlevler (çaprazlama / mutasyon) kullanarak uyarılmıştır [Usta, 2007]. Daha sonra bu uygulamalar daha da geliştirilerek, literatürde sık kullanılan SPEA yöntemine uygulanmış ve başarımı artırdığı tespit edilmiştir (Ergul ve ark., 2009b). Eğer uyarılmış elitler orjinallerinden daha iyi ise, o zaman arşiv, statik bir saklama yerinden ziyade etkin bir çözüm sağlayıcı olarak görülebilir. Bu yaklaşımın ardındaki ana motivasyon, elit bireylerin çözüme (herhangi bir optimizasyon probleminde) en yakın bireyler olmaları ve bu bireylerin genetik işlevler tarafından uyarılmasıyla daha uzaktaki bireylere göre daha iyi sonuçlar elde edilebileceği ümididir. İşte bu yapı etkin elitizm mekanizması olarak adlandırılmaktadır.

Etkin elitizm mekanizması, bir turboşarj ünitesi (veya turbodan) esinlenilerek önerilmiştir. Turbo, motora atmosferik basıncın üzerinde hava vererek yani cebri doldurum yaptırarak daha küçük hacimli motordan daha yüksek güç alınmasını sağlayan, hareketini egzoz gazının dışarı çıkma basıncından alan bir çeşit hava pompasıdır. Türbin ve kompresör olmak üzere iki adet pervaneye sahiptir. Türbin egzoz tarafında, kompresör emme tarafında yer almaktadır. Egzoz gazının çıkma basıncıyla dönen türbin aradaki bağlantı milinin yardımıyla kompresör pervanesini döndürür. Bu sayede motor silindrine önemli ölçüde artan bir hava girişi sağlanır. Eğer motora giren hava yoğunluğu artırılırsa, daha fazla güç üretilir. Dolayısıyla, bir turbo motorda ne kadar uyarım yapılırsa (hava basılırsa), o kadar fazla güç üretilir (Bkz Şekil 4.50). Benzer şekilde, GA' larda da elit bireyler bir takım GA işlevleri (çaprazlama ve mutasyon) uyarılırsa, GA' nın başarımı artırılabilir. Çünkü elit bireyler çözüme en yakın bireylerdir ve bunların GA işlevleri ile uyarılması neticesinde daha iyi sonuçlar elde edilebilir.



Şekil 4.50. Turboşarj ünitesi [<http://auto.howstuffworks.com/turbo2.htm>, 21.06.2009]

Tüm elit bireylerin seçilebilmesi ve seçme mekanizmasına gönderilmesi her zaman mümkün değildir. Seçilemeyen elit bireyler genetik işlevler tarafından uyarılamaz ve yeni çözümler üretmez. Bununla birlikte, etkin elitizm mekanizması tüm elit bireylere yeni çözümler üretebilmeleri için bir şans vermektedir. Etkin elitizm mekanizmalı bir ÇAGA' nın blok diyagramı Şekil 4.51' de verilmektedir.

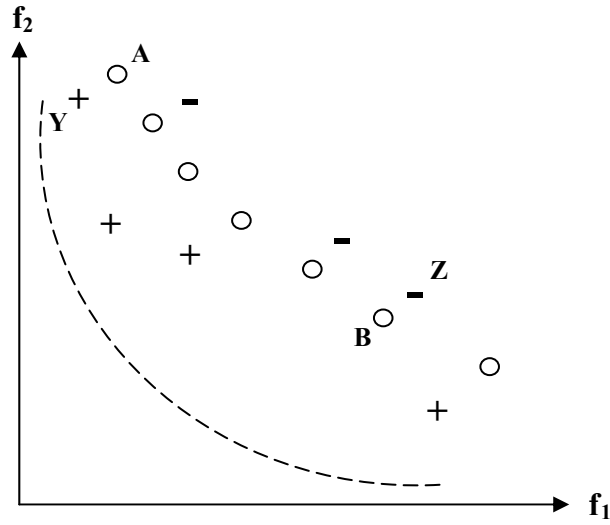


Şekil 4.51. Etkin elitizm mekanizmalı ÇAGA blok diyagramı

Etkin elitizm mekanizması ÇAGA yöntemlerine iki biçimde uygulanabilir:

- i) Elit bireylerin mutasyon işlevi ile uyarılması,
- ii) O andaki (veya adımdaki) elit bireylerin n adım önceki elit bireylerle çaprazlanması

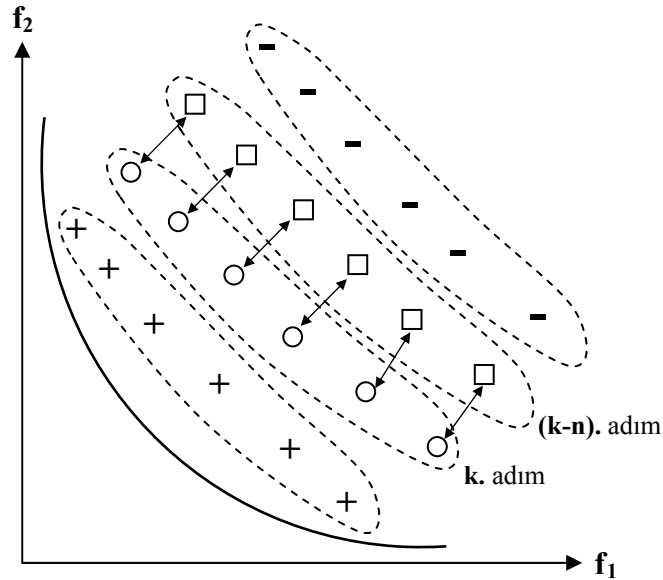
Elit bireylerin mutasyon işlevi ile uyarılmasına bir örnek Şekil 4.52’ de gösterilmektedir. Tüm elit bireyler yalnızca bir kez mutasyona uğratılmışlardır. (+) simgesi ile gösterilen bu mutantlardan bazıları orijinal elit bireylere göre Pareto cephesine daha fazla yakınsamışlardır. Örneğin; Y , A bireyinin mutantıdır ve Y bireyi her iki amaç dikkate alındığında A bireyine göre Pareto cephesine daha yakındır. Bu yüzden Y bireyi arşivde A bireyi ile yer değiştirilir. Z ise B bireyinin mutantıdır ve B bireyi hala Z bireyinden daha iyi olduğundan B bireyi arşivde kalmaya devam eder.



Şekil 4.52. Etkin elitizmde elit bireylerin mutasyon işlevi ile uyarılması (Yuvarlaklar Pareto (elit) Bireyler, artı ve eksi elit bireylerin mutantlarıdır)

Elit bireylerin çaprazlama işlevi ile uyarılmasına ilişkin bir örnek Şekil 4.53’ te verilmektedir. Bu aşamada, k . adımdaki elit bireylerle $(k-n)$. adımdaki elit bireyler çaprazlama işlemine tabi tutulur. Elit sayısı eşit değilse, en az elit sayısı kadar bireyler çaprazlamaya tabi tutulur, çaprazlamaya uğramayacak bireyler aynen saklanır. Çaprazlama işlemi sonucunda bulunan bireyler Şekil 4.53’ te “+” ile gösterilmektedir ve bu bireyler k . adımdaki orijinal elit bireylerden daha iyidirler. Dolayısıyla, arşivdeki

orjinal elitlerle yer değiştirirler. “-” ler ise diğer çapraz bireylerdir ve bunlar orjinal elitlerden iyi olmadıklarından hiçbir işlem yapılmadan kalırlar. Eğer uyartımla yani çaprazlama veya mutasyonla uyarılan bireyler orjinal elit bireylerden daha iyi değilse arşiv değişmeden kalacaktır.



Şekil 4.53. O andaki elit bireyler ile daha önceki adımdaki elit bireylerin çaprazlanması. Yuvarlaklar k . adımdaki elit bireyleri, kareler $(k-n)$. adımdaki elit bireyleri, artı ve eksiler ise çaprazlama sonucu elde edilen bireyleri göstermektedir

4.4.2. Benzetim Sonuçları

Literatürde sık kullanılan çok amaçlı SPEA ve SPEA2 yöntemleri ile önerilen DOPGA ve DOPGA+ yöntemleri hem pasif elitizm mekanizması hem de önerilen etkin elitizm mekanizması kullanılarak çalıştırılmış, pasif ve etkin elitizm karşılaştırması yapılmıştır. Etkin elitizm mekanizması kullanan yöntemlerin önüne **EFE (Effective Elitist)** ön eki getirilmiştir.

ÇAGA karşılaştırmalarında sık kullanılan dört test işlevi (ZDT1, ZDT2, ZDT3 ve ZDT6) benzetimlerde kullanılmıştır. Tek noktalı çaprazlama ve bit-tabanlı mutasyon kullanılmıştır. Çaprazlama olasılığı $p_c = 0.9$ ve mutasyon olasılığı $p_m = 1/l$ (l dizi boyutudur) olarak seçilmiştir. 100 adım (ya da nesil) sonucunda GD ve Δ metrikleri

kullanılarak deęerlendirmeler yapılmıřtır. Bařlangıç topluluęu boyutu 100 olarak seilmiřtir. Arřiv boyutu ise SPEA ve DOPGA iin 25, SPEA2 ve DOPGA+ iin ise 100 olarak belirlenmiřtir. Tm yntemler iin, son nesildeki ana topluluk ile arřivin birleřtirilmesi ile elde edilen birey topluluęu bařarım ltlerinin hesabı iin kullanılmıřtır. Her parametre 30 bit olarak kodlanmıřtır. ZDT2 ve ZDT6 test iřlevlerinde 10 parametre kullanılmıřtır. Seme mekanizması olarak ikili turnuva seimi kullanılmıřtır. Etkin elitizm iin, etkin mutasyon olasılıęı 0.05 seilmiřtir (deneysel olarak bu olasılıęın dřk olması gerektięi saptanmıřtır).

Tm etkin elitizm mekanizmalı yntemlerde, elit bireyler 31. adıma kadar (deneysel olarak belirlenmiřtir) mutasyon iřlevi ile uyarılmıřlardır, 30. adımdan sonra mutasyon iřlevi kullanılmamıřtır. nk mutasyon ile uyarım elit bireylerin Pareto cephesine yakınsamasını saęlarken, aprazlama ile uyarım ise dzgn daęılımı saęlamaktadır ve belli bir ařamaya kadar bu iki iřlevin birlikte kullanılması gerekmektedir. Eęer elit bireylerin mutantları kendilerinden iyi ise bunlar arřivde yer deęiřtirilmiřtir, aksi takdirde deęiřmeden kalmıřlardır. Hangi bireyin iyi olduęuna ama uzayında hesaplanan klit mesafesine gre karar verilmiřtir. Aynı zamanda, *k*. adımdaki elit bireylerle ondan 10 adım nceki elit bireyler aprazlama iřlevine tabi tutulmuřlardır. rneęin, 100. adımdaki elit bireylerle 90. adımdaki elit bireyler, 90. adımdaki elit bireylerle 80. adımdaki elit bireyler vs. aprazlanmıřlardır. Eęer uyarım sonucunda elde edilen apraz bireyler o andaki elit bireylerden iyi ise bunlar arřivde yer deęiřtirilmiřtir, aksi takdirde deęiřmeden kalmıřlardır. Daha sonra arřivdeki ve ana topluluktaki bireyler birleřtirilerek seme iřlevine tabi tutulmuřlardır.

Tm yntemler ve onların etkin elitizm mekanizmalı versiyonları, tm test iřlevlerinde 20' řer kez alıřtırılmıřlardır ve sonu olarak yakınsama ve daęılım ltlerinin ortalama deęerleri ve varyansları hesaplanmıřtır.

izelge 4.9, tm yntemler tarafından bulunan bireylerin Pareto cephesine yakınsamalarını gsteren yakınsama ltnn (GD) ortalama ve varyans deęerlerini gstermektedir. izelge 4.10, tm yntemler tarafından bulunan bireylerin Pareto cephesinde dzgn daęılıp daęılmadıklarını gsteren daęılım ltnn (Δ) ortalama ve varyans deęerlerini gstermektedir.

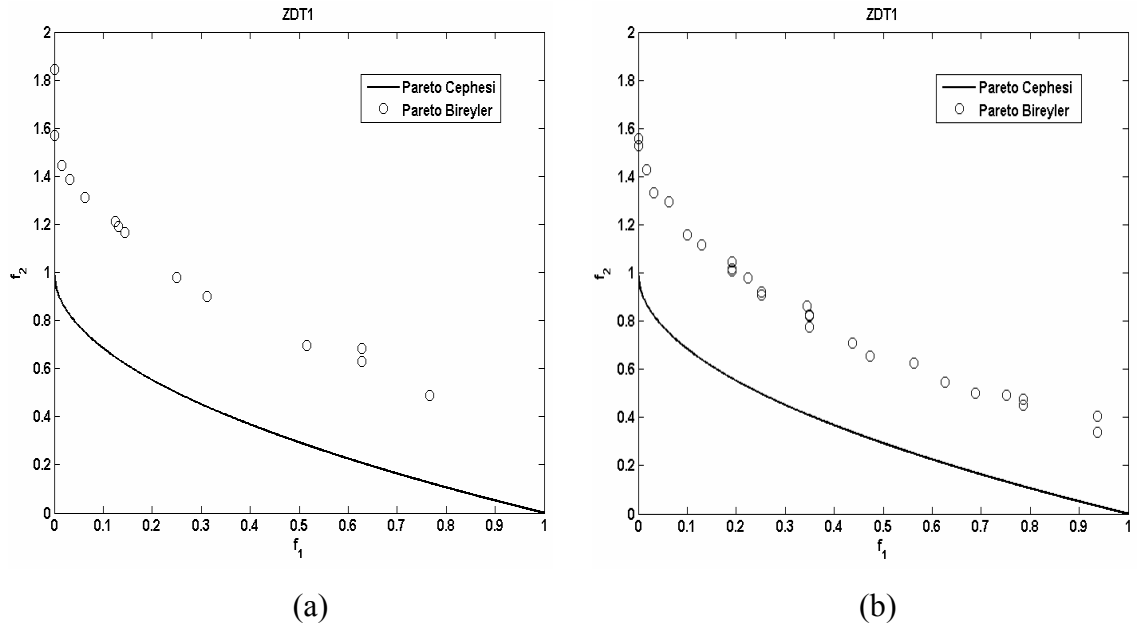
nerilen yntemin etkinlięinin gsterilmesi iin ZDT1 ve ZDT2 test iřlevlerinde tm yntemlerin pasif ve etkin elitizm mekanizmalı biimleri iin elde edilen tipik benzetim sonuları řekil 4.54 - řekil 4.61 arasında verilmektedir.

Çizelge 4.9. Yakınsama ölçütünün (GD) ortalama (ilk satırlar) ve varyans (ikinci satırlar) değerleri

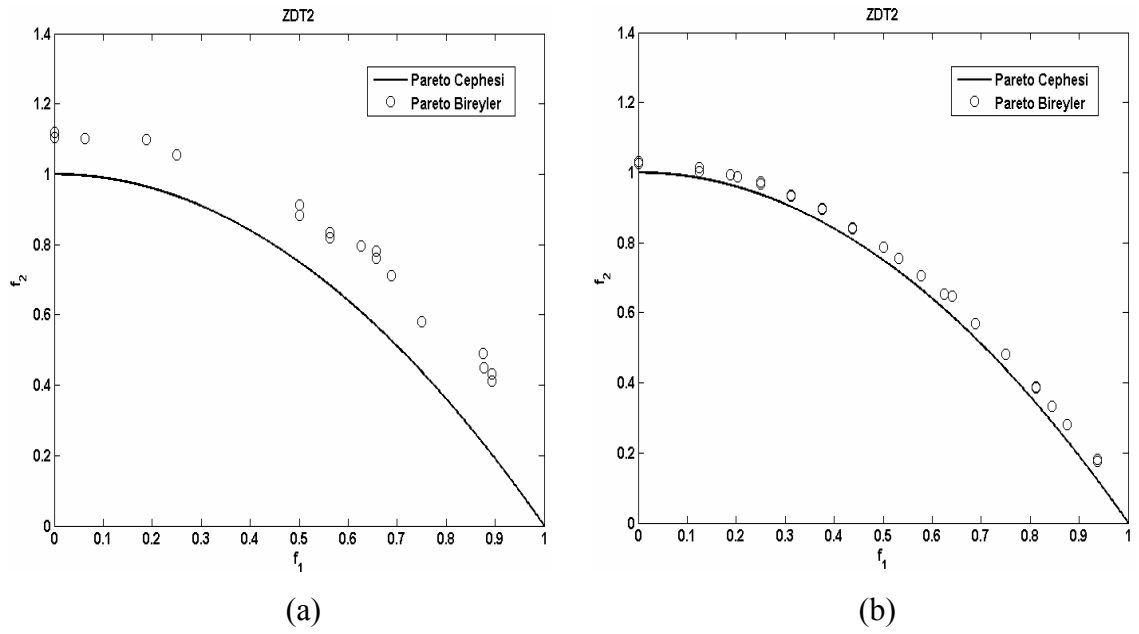
	ZDT1	ZDT2	ZDT3	ZDT6
SPEA	0.0627	0.0133	0.0517	1.2824
	2.1822e-4	2.3891e-4	2.5627e-4	0.2391
EFE_SPEA	0.0549	0.0086	0.0488	1.1638
	1.3940e-4	3.3388e-5	8.4167e-5	0.1942
SPEA2	0.0327	0.0044	0.0441	0.7672
	2.5225e-5	5.1226e-6	2.5372e-4	0.0232
EFE_SPEA2	0.0293	0.0031	0.0345	0.7238
	1.7667e-5	2.8834e-6	6.5541e-5	0.0410
DOPGA	0.0223	0.0016	0.0152	0.2864
	3.3784e-5	1.4443e-6	4.1772e-5	0.0030
EFE_DOPGA	0.0208	0.0010	0.0167	0.3902
	1.5020e-5	1.2260e-7	2.4062e-5	0.0259
DOPGA+	0.0219	0.0013	0.0148	0.3201
	3.3813e-5	2.3274e-7	2.2227e-5	0.0059
EFE_DOPGA+	0.0198	0.0011	0.0143	0.2892
	1.3532e-5	1.9397e-7	5.5985e-5	0.0027

Çizelge 4.10. Dağılım ölçütünün (Δ) ortalama (ilk satırlar) ve varyans (ikinci satırlar) değerleri

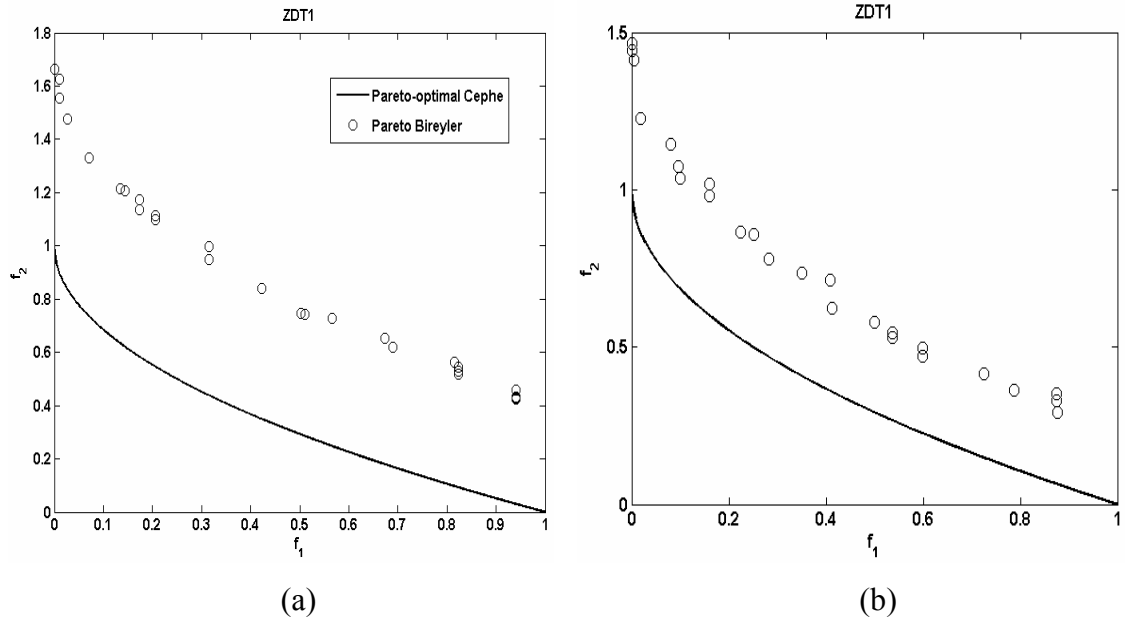
	ZDT1	ZDT2	ZDT3	ZDT6
SPEA	0.6170	0.6189	0.6320	0.8425
	0.0047	0.0032	0.0073	0.0079
EFE_SPEA	0.5945	0.6062	0.6148	0.8761
	0.0055	0.0056	0.0059	0.0070
SPEA2	0.5895	0.6115	0.6080	0.8285
	0.0041	0.0033	0.0112	0.0060
EFE_SPEA2	0.5761	0.6322	0.5874	0.8147
	0.0038	0.0031	0.0095	0.0089
DOPGA	0.5869	0.5894	0.6172	0.7168
	0.0041	0.0037	0.0099	0.0083
EFE_DOPGA	0.5971	0.6082	0.6504	0.8015
	0.0028	0.0019	0.0097	0.0167
DOPGA+	0.5289	0.5703	0.5895	0.7063
	0.0055	0.0044	0.0050	0.0052
EFE_DOPGA+	0.6026	0.6142	0.6119	0.7308
	0.0028	0.0021	0.0057	0.0059



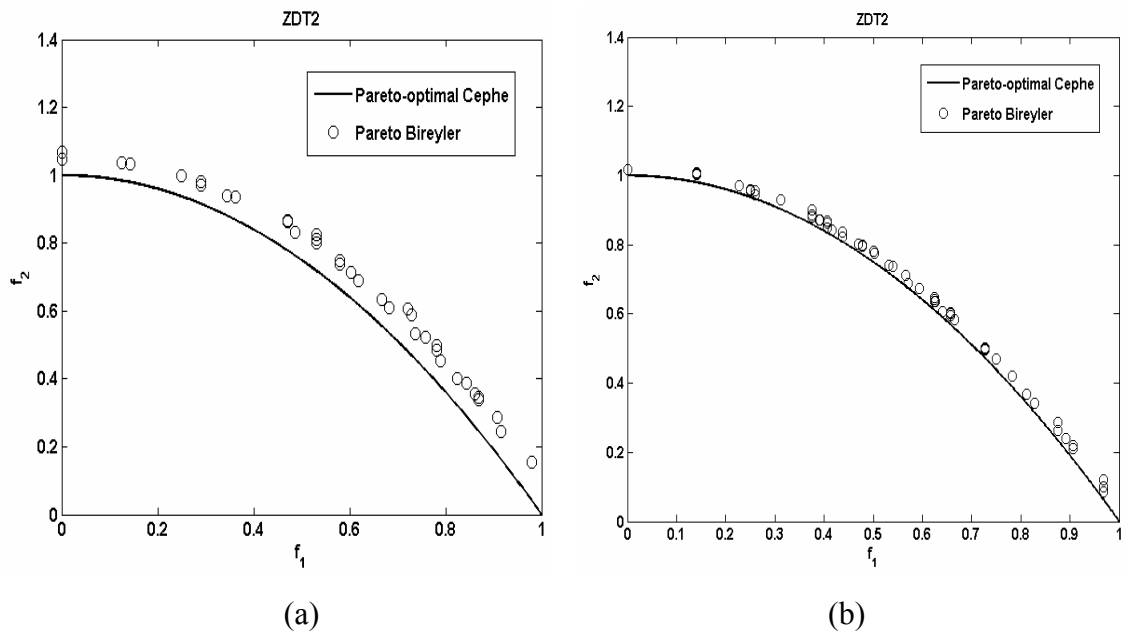
Şekil 4.54. ZDT1 test işlevinde bulunan Pareto bireyler: (a) SPEA, (b) EFE_SPEA



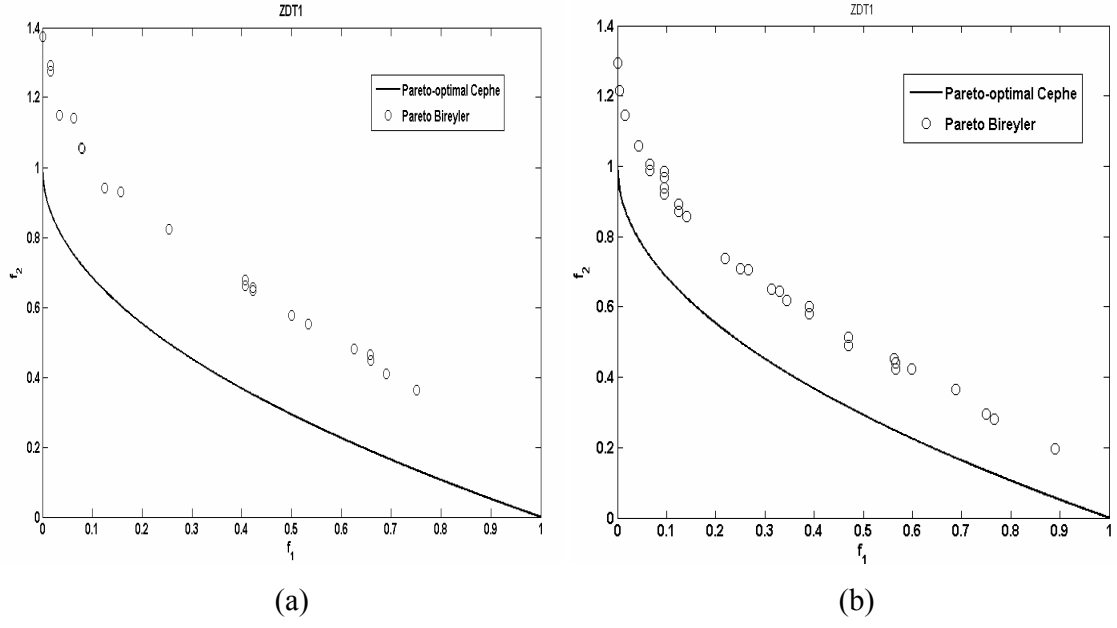
Şekil 4.55. ZDT2 test işlevinde bulunan Pareto bireyler: (a) SPEA, (b) EFE_SPEA



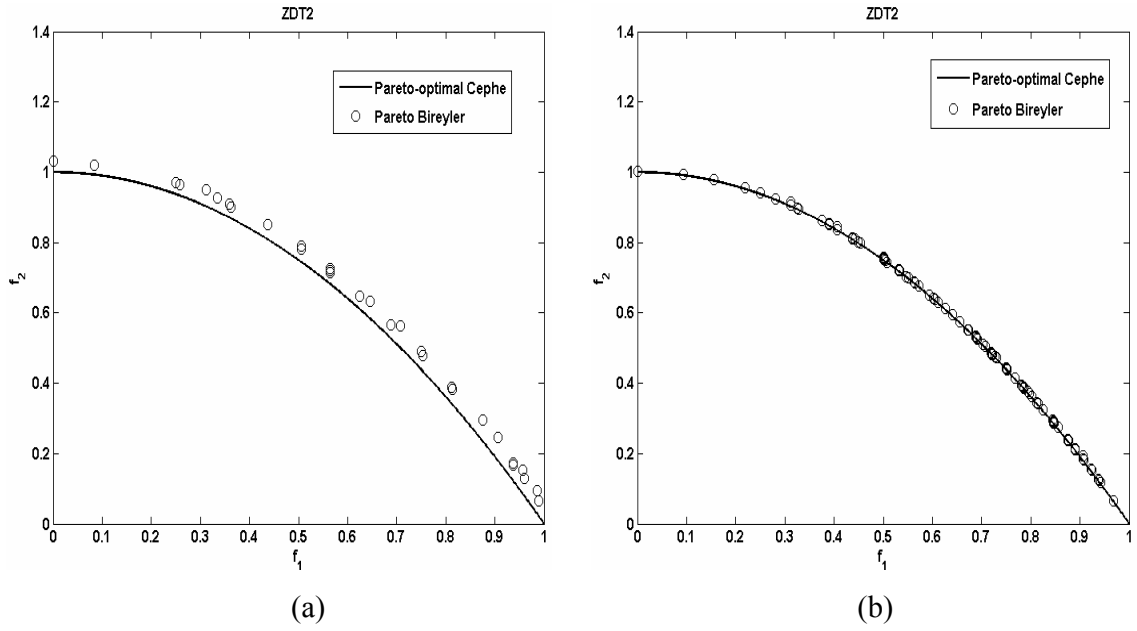
Şekil 4.56. ZDT1 test işlevinde bulunan Pareto bireyler: (a) SPEA2, (b) EFE_SPEA2



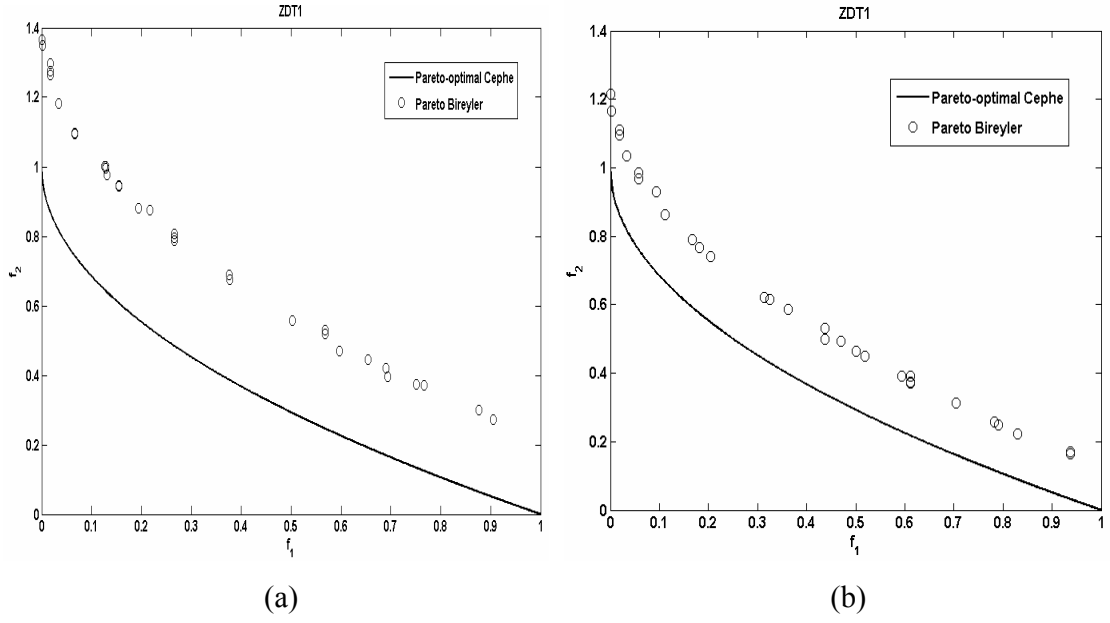
Şekil 4.57. ZDT2 test işlevinde bulunan Pareto bireyler: (a) SPEA2, (b) EFE_SPEA2



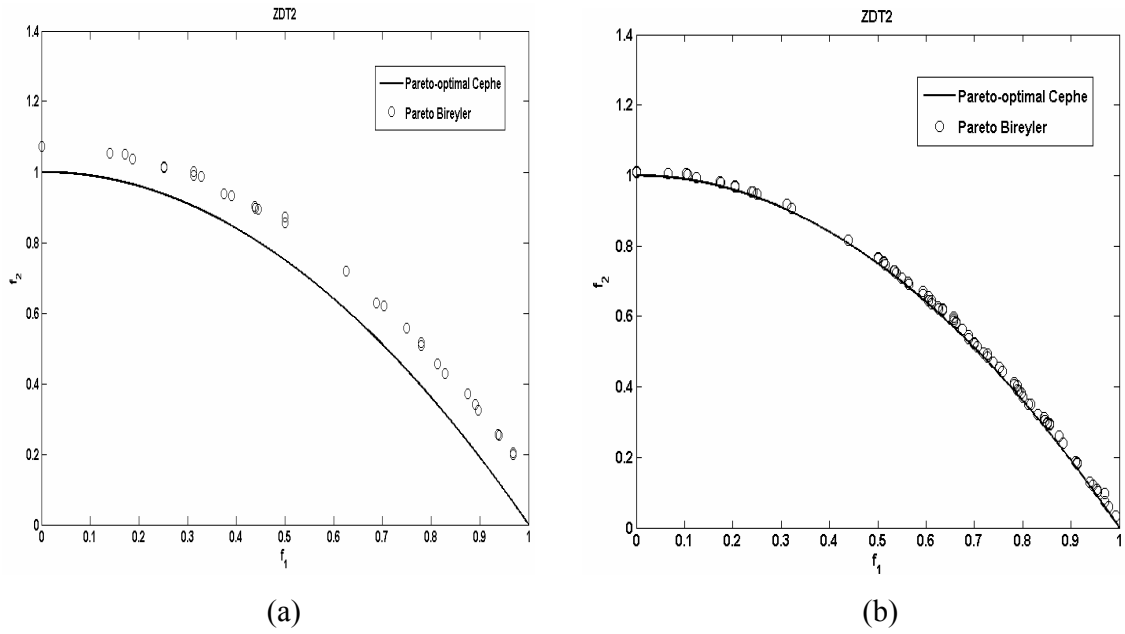
Şekil 4.58. ZDT1 test işlevinde bulunan Pareto bireyler: (a) DOPGA, (b) EFE_DOPGA



Şekil 4.59. ZDT2 test işlevinde bulunan Pareto bireyler: (a) DOPGA, (b) EFE_DOPGA



Şekil 4.60. ZDT1 test işlevinde bulunan Pareto bireyler: (a) DOPGA+, (b) EFE_DOPGA+



Şekil 4.61. ZDT2 test işlevinde bulunan Pareto bireyler: (a) DOPGA+, (b) EFE_DOPGA+

4.4.3. Sonuçlar ve Değerlendirme

Pasif elitizm mekanizması önerilen etkin elitizm mekanizmasıyla yer değiştirilmiştir. Elit bireyler en iyi bireylerdir ve dolayısıyla çözüme en yakın bireylerdir. Bu yüzden, arşivdeki elit bireylerin çaprazlama / mutasyon işlevleri ile uyarılması, kendilerinden yakınsama ve düzgün dağılım açısından daha iyi bireyler üretilmesini sağlayabilir. Bu amaçla klasik **sakla/aktar** elitizm yapısı, **sakla/uyar/aktar** bir etkin elitizm yapısı kullanılarak iyileştirilmiştir. Böylece, etkin elitizm mekanizmasında ilk topluluk ve arşivdeki bireyler çözüm üretebilmektedir.

Benzetimler sonucunda, EFE_SPEA yönteminin SPEA yönteminden yakınsama ve dağılım açısından daha iyi olduğu görülmüştür. Yalnızca, ZDT6 test işlevinde SPEA yöntemi EFE_SPEA' dan daha iyi bir dağılım sağlamıştır.

EFE_SPEA2 yönteminin SPEA2 yönteminden yakınsama açısından daha iyi olduğu görülmüştür. ZDT2 test işlevi hariç tüm test işlevlerinde, EFE_SPEA2 yöntemi SPEA2' den daha düzgün bir şekilde Pareto cephesine dağılmıştır.

EFE_DOPGA yöntemi ZDT1 ve ZDT2' de, DOPGA yöntemi ise ZDT3 ve ZDT6' da daha iyi yakınsama sağlamışlardır. DOPGA, EFE_DOPGA' dan daha iyi dağılım ölçütlerine sahiptir.

EFE_DOPGA+, tüm test işlevlerinde DOPGA+ yönteminden iyi yakınsama sağlamaktadır. Ancak DOPGA+ tüm test işlevlerinde dağılım açısından daha iyidir.

Tüm pasif ve etkin elitizm mekanizmalı yöntemler birlikte değerlendirildiğinde; ZDT1, ZDT3 ve ZDT6' da EFE_DOPGA+, ZDT2' de ise EFE_DOPGA diğer yöntemlerden yakınsama açısından daha iyidir. ZDT1, ZDT2 ve ZDT6' da DOPGA+, ZDT3' te ise EFE_SPEA2 yöntemi diğer tüm yöntemlerden daha iyi olarak Pareto cephesine dağılan bireyler oluşturmuşlardır.

Etkin elitizm mekanizması üzerinden bir takım iyileştirmelere gereksinim duyulmaktadır. Özellikle, mutasyon ve çaprazlama işlevlerinin hangi anda devreye gireceklerinin daha iyi belirlenmesi gerekebilir. Ayrıca elit bireylerin çaprazlanması için bir algoritma kurulması daha iyi sonuçlar elde edilmesini sağlayabilir. Hangi elit bireylerin hangi elit bireylerle çaprazlanacağını daha iyi saptanması, etkin elitizmin yeteneğini artırabilir. Bunun için hangi bireylerin çaprazlamaya gireceğine dair bilgi veren bir kural tabanının oluşturulması gelecek çalışmalardan birisi olacaktır. Ayrıca etkin elitizm mekanizmasının işlem yükünün azaltılması da bir çalışma konusudur.

Bu bölümde önerilen etkin elitizm mekanizması literatürde yer alan ÇAGA yöntemlerine uygulanarak, bu yöntemlerin başarımı, genel yapıları değiştirilmeden artırılabilir. Etkin elitizm mekanizmasının tek dezavantajı, ÇAGA yöntemlerinin işlem zamanlarında artışa neden olmasıdır.

4.5. ÇAGA Yöntemlerinin Sıralama Yeteneklerinin Ölçülmesi İçin İki Yeni Ölçüt Önerisi

ÇAGA yöntemlerinin, bir topluluktaki birbirinden farklı bireylere farklı puan atama yeteneğinin olması gerekir. Böylece, topluluk en çok istenilen bireyden en az istenilen bireye doğru tekil bir biçimde sıralanabilir ve ÇAGA yöntemi tarafından seçme mekanizmasına daha nitelikli bilgi aktarımı sağlanarak, genetik algoritmanın başarımı artırılabilir. Bu amaçla bu bölümde ilk olarak, ÇAGA yöntemlerinin sıralama yeteneklerinin ölçülebilmesi için **Ceza (P)** ve **Ödül (R)** başarımlar ölçütleri önerilmektedir. Bu ölçütlerle bir ÇAGA' nın seçme mekanizmasına (ya da karar vericiye) ne kadar nitelikli bilgi aktarabildiği ölçülebilmektedir. Sıralama yeteneğinin analitik olarak ölçülebilmesi için, puan dağılımlarının tekil bir biçimde sıralanıp sıralanmadığı ve düzgün dağılıp dağılmadığı göz önüne alınmıştır.

Tek amaçlı genetik algoritmada, bireyler en çok istenilenden en az istenilene doğru tekil bir biçimde sıralanır. Ancak çok amaçlı durumda, Pareto tanımından dolayı tekil bir sıralama elde edilmesi her zaman mümkün olmayabilir. İdeal durumda, ÇAGA yöntemleri bir topluluktaki bireyleri en çok istenilenden en az istenilene doğru tekil bir biçimde sıralamaya (bireyleri birbirinden ayırt etmeye) çalışırlar. Literatürde yer alan ÇAGA yöntemleri, bir topluluktaki bireylere farklı farklı puanlar atamaktadırlar. Bunun anlamı, belirli bir topluluk farklı ÇAGA' lar tarafından farklı puan değerleri veya farklı puan dağılımlarıyla gösterilebilir. Puanlama mekanizmasının çıkışı, karar vericinin veya seçme mekanizmasının giriş kümesini oluşturmaktadır. Seçme mekanizmasının çıkışı ise ÇAGA yöntemi tarafından nasıl beslenildiği ile ilişkilidir. Bu yüzden, ÇAGA puanlama yöntemleri seçme mekanizmasının çıkışını etkilemektedir. Doğal olarak, ÇAGA yöntemi tarafından ne kadar nitelikli bilgi üretildiği ve bunun ne kadarının seçme mekanizmasına aktarıldığı sorusu akla gelir. N adet farklı giriş kümesi (topluluktaki bireylerin puanlarını gösteren) ile seçme mekanizmasının beslendiğini

düşünelim, benzer olarak seçme mekanizması tarafından da N adet farklı çıkış üretilcektir. Diğer bir deyişle, N adet farklı puan, seçme mekanizmasına farklı seviyelerde seçilme baskısı uygulayacaktır. Bu nedenle, var olan yöntemlerin bireyleri sıralama yeteneği sayısal olarak hesaplanmalıdır. Şu ana kadar, literatürde yer alan ÇAGA yöntemleri, genellikle bazı test işlevleri ve başarımlar ölçütleri kullanılarak karşılaştırılmışlardır. Var olan ölçütler; Pareto cephesine yakınsama, bu cephede düzgün dağılım ve bulunan Pareto birey sayısını ölçmektedirler. [Zitzler ve ark., 2000] makalesinde, bir ÇAGA yönteminin başarımlarını ölçerken şu üç konuya dikkat edilmesi gerektiği belirtilmektedir: *i)* Bulunan Pareto bireylerin sayısı fazla olmalıdır, *ii)* küresel Pareto-optimal cepheye (biliniyorsa) mümkün olduğu kadar yaklaşılmalıdır, *iii)* bireylerin bu cephede dağılımı mümkün olduğunca düzgün olmalıdır. [Deb, 1999] makalesinde, test işlevlerinin GA' ların nasıl çalıştığı konusunda çok fazla bilgi sahibi olunmadan oluşturulduğunu belirtmektedir. Benzer olarak, önerilen ÇAGA yöntemleri de mühendislik hesaplamalarından daha çok mühendislik sezgilerine dayanmaktadır. Bu yüzden, literatürde çok sayıda ÇAGA yöntemi vardır ve önerilmeye de devam edilmektedir. Bu kadar çok yöntemin olması puanlama mekanizmasının hala **ucu açık bir soru** olduğunu göstermektedir.

Ceza ölçütü, bir ÇAGA yöntemi tarafından bir topluluktaki bireylere atanan puanların ne kadar farklı olduğunu ölçer. Yani, aynı puana sahip bireyler çok ise ceza da çok olacaktır. Mantıksal olarak, en iyi bireylerin aynı puana sahip olması en kötü bireylerin aynı puanlara sahip olmasından daha fazla ceza alınmasına neden olacaktır. Eğer bireyler en çok istenilenden en az istenilene doğru tekil bir şekilde sıralanabilirse, yani hiçbir birey aynı puana sahip değilse, o zaman seçme mekanizması hangi bireyin diğerinden daha iyi olduğu konusunda bir sıkıntıya girmeden karar verebilecektir. Bu durumda, ÇAGA sanal olarak tek amaçlı bir GA gibi davranacaktır.

Ödül ölçütü, bir topluluktaki bireylerin puan dağılımlarının ne kadar düzgün dağıldığını belirler. Ceza sıfır olsa bile ödül sıfırdan farklı olabilir. Düzgün puan dağılımı ideal bir durumdur. Puan dağılımının kısmi kalabalıklıkları veya kısmi seyreklilikleri sıralamanın düzgün olmadığı anlamına gelir.

İdeal bir durumda, yani bir ÇAGA yöntemi farklı bireylere farklı puanlar atamışsa, ceza sıfır olur. Eğer bireylerin puanları en çok istenilenden en az istenilene doğru düzgün bir biçimde dağılıyorsa, ödül ölçütü sıfır olacaktır ve bu ideal bir durumdur. İdeal bir ÇAGA yöntemi, aynı anda hem sıfır ceza hem de sıfır ödüle sahip

olmalıdır. Yani, ceza ve ödül ölçütlerinin en küçük olması istenmektedir. Eğer birkaç ÇAGA yöntemi karşılaştırılıyorsa, en düşük ceza ve ödüle sahip olanın en iyi yöntem olduğu söylenebilir. Ancak, seçme mekanizması olasılıksal olarak çalıştığı için tam bir değerlendirme yapmak mümkün olmayabilir.

İlerleyen bölümlerde ceza ve ödül ölçütleri ayrıntılı olarak açıklanmakta ve literatürde yer alan ve yeni önerilen ÇAGA yöntemleri bu ölçütler kullanılarak karşılaştırılmasıyla elde edilen sonuçlar verilmektedir.

4.5.1. Ceza ve Ödül Ölçütleri

Ceza ve ödül kavramlarına geçmeden önce, ÇAGA yöntemlerinin nasıl karşılaştırılabileceğine bakmak yararlı olabilir. Literatürde yer alan tipik bir karşılaştırma senaryosu aşağıda verilmektedir [Huband ve ark., 2006].

- Karşılaştırılacak ÇAGA yöntemleri belirlenir,
- Varolan test işlevlerinden birkaçı seçilir veya yenileri oluşturulur,
- ÇAGA yöntemi tarafından elde edilen sonuçları karşılaştırmak için birtakım başarımlar ölçütleri seçilir,
- Her bir test işlevinde tüm ÇAGA yöntemlerinin sonuçları bulunur,
- Sonuçlar kullanılarak ölçümler yapılır ve elde edilen veriler karşılaştırılır,
- Sonuçlar grafiksel olarak çizdirilir.

Ceza ve ödül ölçütleri, bir ÇAGA yönteminin puanlama mekanizmasının görevini ne kadar iyi yaptığı belirlemek için kullanılırlar. Puanlama (sıralama) yeteneğinin ölçülmesi için belirli bir ölçüt yoktur. Diğer bir deyişle, puanlamanın ne kadar iyi yapıldığını doğrudan ölçen bir ölçüt yoktur. Burada önerilen ölçütler, problem tipinden bağımsızdır ve Pareto-optimal cephenin önceden bilinmesine de ihtiyaç duymamaktadırlar. ÇAGA yöntemleri tarafından bir topluluktaki bireylere atanan puanların bilinmesiyle karşılaştırma yapılabilir. Buradaki fikir, *“bireylerin puanlarına bakılarak topluluk hakkında ne kadar bilgi edinilebilir”* sorusundan gelmektedir. Ayrıca, *bu puan değerleri kullanılarak karar vericiye ya da seçme mekanizmasına ne kadar*

bilgi gönderilebilir? Bir topluluktaki bireylere puan atanması, aslında bireylerin en çok istenilenden en az istenilene doğru sıralanması işlemidir. Bu sorulara yanıt bulabilmek için ceza ve ödül ölçütleri önerilmektedir.

Prensip olarak, farklı bireyler aynı puana sahipse, bu durumda bir ceza değeri oluşturacaklardır. Örneğin, en çok istenilen iki birey (iki Pareto birey gibi) aynı puana sahipse, daha az istenilen iki bireye (iki Pareto olmayan birey gibi) göre daha fazla ceza değeri üreteceklerdir. Yani daha önemli bireylerin aynı puana sahip olması daha fazla ceza değeri üretir, bu nedenle ceza eğrisi üstel bir biçimde azalan bir işlev olmalıdır. Aşağıda bu duruma uygun bir ceza işlevi verilmektedir. Bu işlev, her bireyin konumu veya önemi ile ilişkilidir.

$$Ceza(i) = \exp\left(-\frac{i}{top_boyut}\right) \quad i=1,2,...,top_boyut \quad (4.21)$$

Ceza değeri aşağıdaki algoritma kullanılarak hesaplanır:

- 1- ÇAGA yöntemini çalıştır ve bireylerin puan değerlerini al,
- 2- Puan değerlerini en önemliden en az önemliye (ÇAGA yöntemine göre değişebilir) doğru sırala,
- 3- Aynı puan değerlerine sahip olan bireyleri belirle,
- 4- Ceza işlevine (eşitlik 4.21) git ve Adım-3' teki bireylerin ceza değerlerini belirle, daha sonra bu ceza değerlerini topla,

$$Toplam_Ceza = \sum Ceza(m) \quad (m: \text{aynı puana sahip birey indisleri}) \quad (4.22)$$

Hesaplamaların daha iyi anlaşılması için aşağıda sayısal bir örnek verilmektedir. Sekiz bireylik bir topluluğa ait puan değerleri aşağıdaki gibi verilsin. Buna göre toplam ceza şöyle hesaplanır:

- 1- Birey puanları : [1 2 1 6 4 8 5 8]
- 2- Puanlar sıralanır : [1 1 2 4 5 6 8 8]
- 3- İlk iki ve son iki birey aynı puanlara sahip,

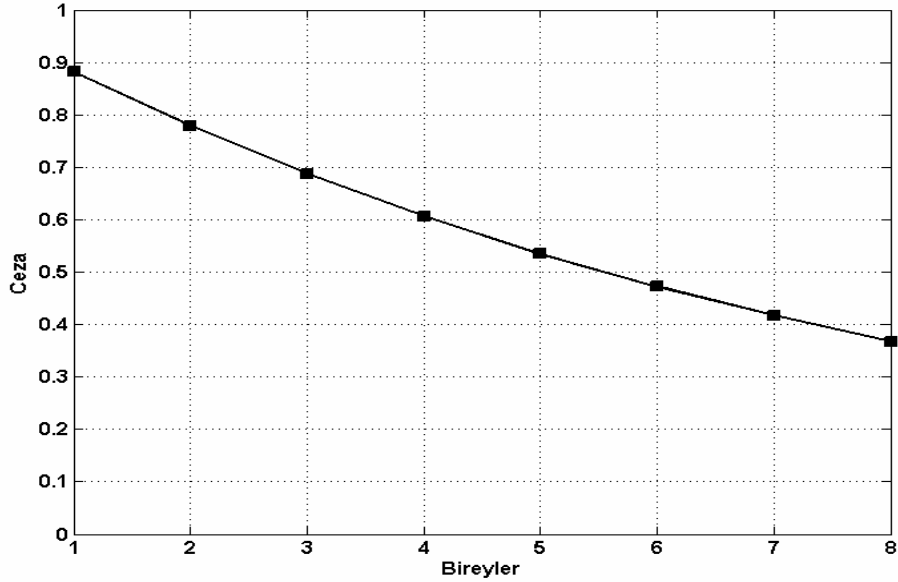
- 4- (4.21) numaralı eşitliğe göre 8 bireylik bir ceza eğrisi oluşturulur (Şekil 4.62) ve aynı puana sahip bireylerin ayrı ayrı ceza değeri hesaplanır:

$$\text{ceza}(1) = p(1) = \exp\left(\frac{-1}{8}\right) = 0.8825 \quad \text{ceza}(2) = p(2) = \exp\left(\frac{-2}{8}\right) = 0.7788 \quad (4.23)$$

$$\text{ceza}(7) = p(7) = \exp\left(\frac{-7}{8}\right) = 0.4169 \quad \text{ceza}(8) = p(8) = \exp\left(\frac{-8}{8}\right) = 0.3679 \quad (4.24)$$

$$\text{Toplam_Ceza} = \sum \text{Ceza}(m) = \sum p(m) = p(1) + p(2) + p(7) + p(8) \quad (4.25)$$

$$\text{Toplam_Ceza} = 0.8825 + 0.7788 + 0.4169 + 0.3679 = 2.4461 \quad (4.26)$$



Şekil 4.62. Sekiz bireylik bir topluluk için ceza işlevi

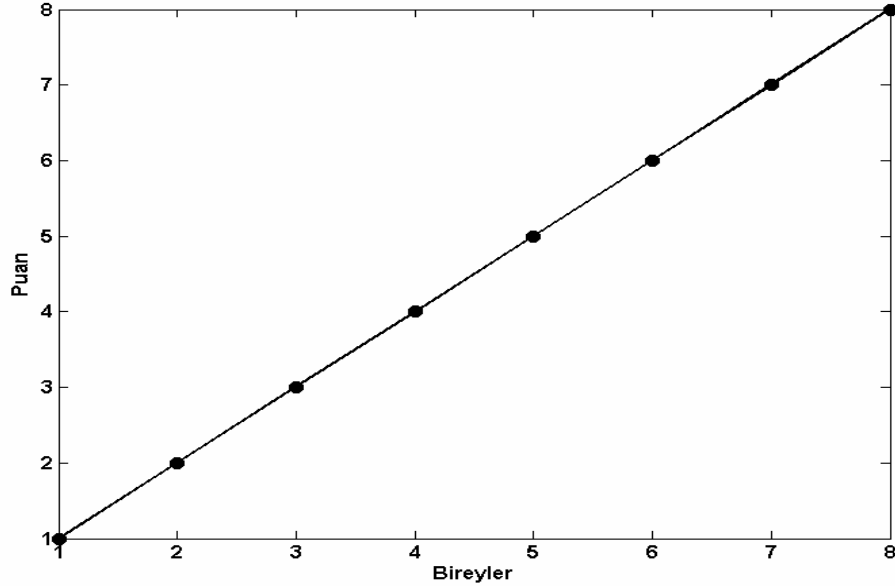
Ödül ölçütü ise, tüm bireyler birbirinden farklı olsa veya topluluk sıfır ceza değerine sahip olsa bile, bireylerin puan dağılımının ne kadar düzgün olduğunun belirlenmesinde kullanılır. Ödül ölçütü şöyle hesaplanır:

- 1- Puan değerleri 0 ile 1 arasına normalize edilir,
- 2- Normalize puan değerleri en önemliden en az önemliye doğru sıralanır,
- 3- Tüm toplulukta ardışıl bireyler arasındaki normalize puan farkları (NPF) hesaplanır,
- 4- Ardışıl bireyler arasındaki normalize puan farklarının standart sapması ve

ortalama değeri bulunur ve buna göre Ödül ölçütü aşağıdaki gibi hesaplanır:

$$\text{Ödül} = \frac{\text{Standart Sapma(NPF)}}{\text{Ortalama Deger(NPF)}} \quad (4.27)$$

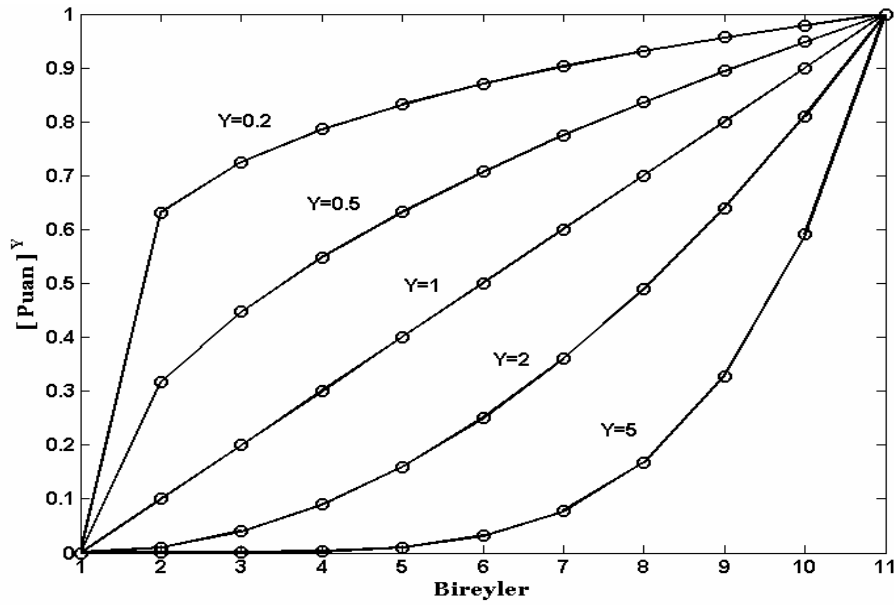
Eğer Şekil 4.63' te gösterildiği gibi ödül değeri sıfır olursa, puan dağılımı kesinlikle düzgündür. Burada 1 nolu birey en iyi birey, 8 nolu birey ise en kötü bireydir. Bu ideal puan dağılımında, bireyler eşit aralıklarla sıralanmış ve puan değerleri arasındaki farklar sabit olduğundan standart sapma sıfır olacaktır.



Şekil 4.63. İdeal puan dağılımında ödül değeri sıfır olur

ÇAGA yöntemleri karşılaştırıldığında, en düşük ceza ve ödül (birlikte düşünülmelidir) değerine sahip olan yöntem sıralama yeteneği açısından en iyi yöntem olacaktır. Puan değerleri seçme mekanizmasının giriş kümesidir ve seçme mekanizması giriş kümesinin içeriğine bağlı olarak bir çıkış kümesi oluşturur. Şekil 4.64, birbirinden farklı puan dağılımları olan ancak hepsi de sıfır ceza değerine sahip beş farklı topluluğu göstermektedir. Çünkü, tüm bu farklı puan dağılımları seçme mekanizması üzerinde farklı seviyelerde seçilme baskısı oluşturacaklardır. Bir başka deyişle, bu beş farklı küme ile beslendiğinde, seçme mekanizmasının çıkışları (bir sonraki ebeveynler) aynı olmayacaktır. Tüm bu dağılımlar sıfır ceza değerine sahip olmalarına rağmen, ödül

tanımından dolayı ödül değerleri birbirinin aynısı değildir. Şekil 4.64’ ten görüleceği üzere, puan değerinin $\gamma=1$ ile üssü alınırsa, bu durumda puan dağılımı düzgün olur. Eğer γ değeri 1’ den büyük olursa, puan dağılımının grafiği içbükey olur ve ilk birkaç bireyin puanı birbirine çok yakın olurken, tersine son birkaç bireyin puanı birbirinden uzak olur. Benzer olarak, eğer γ değeri 1’ den küçük ise, puan dağılımı dışbükey bir işlev olur. İlk birkaç bireyin puan değerleri birbirinden uzak yerleşirken, tersine son birkaç bireyin puanı çok yakın olur. Puan dağılımının kısmi seyreklikleri ve kısmi kalabalıklıkları, sıfırdan farklı bir ödül değeri olmasına neden olur ve bu da dağılımın düzgün olup olmadığını gösterir. Tüm bu beş farklı puan dağılımı ve bunlara ilişkin ödül değerleri Şekil 4.64 ve Çizelge 4.11’ de verilmektedir.



Şekil 4.64. Beş farklı topluluğun puan dağılımları

Çizelge 4.11. Şekil 4.64’ teki puan dağılımlarının ödül değerleri

Puan Dağılımının Üs Değeri	Ödül (R) Değeri
$\gamma=1$	0
$\gamma=2$	0.6055
$\gamma=5$	1.3867
$\gamma=0.5$	0.7981
$\gamma=0.2$	1.8788

4.5.2. Benzetim Sonuçları

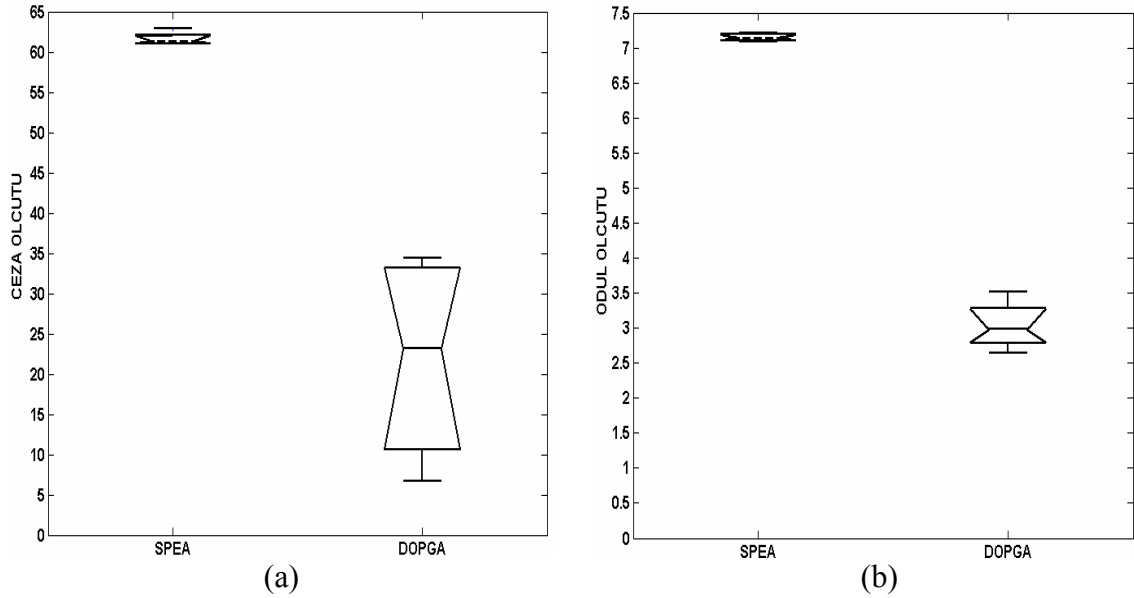
Literatürde sık kullanılan SPEA ve SPEA2 yöntemleri ile tezde önerilen DOPGA ve DOPGA2 yöntemlerinin sıralama yeteneklerinin ölçülmesi için, sık kullanılan dört test işlevi (ZDT1, ZDT2, ZDT3, ZDT6) üzerinde bu yöntemler çalıştırılmış ve her bir GA adımımda bireylere atadıkları puan değerleri ceza ve ödül ölçüt değerlerinin hesaplanmasında kullanılmıştır. Tüm ÇAGA yöntemleri 20' şer kez 100 adım çalıştırılmıştır. Her bir adımda bireylere atanan puan değerleri kullanılarak, ceza ve ödül ölçütleri ($20 \times 100 = 2000$ değer) hesaplanmıştır. Daha sonra bu ceza ve ödül değerlerinin ortalamaları sıralama yeteneklerinin karşılaştırılmasında kullanılmıştır. Ceza ve ödül ölçütlerinin yanı sıra, yakınsama ve dağılım ölçütleri de hesaplanarak ortalama değerleri alınmıştır.

Benzetimlerde, parametre sayısı 30, bit sayısı ya da çözünürlük 30, birey sayısı 100, nesil sayısı 100, çaprazlama olasılığı 0.9, mutasyon olasılığı $1/\text{bit_sayısı}$ olarak seçilmiştir. İkincil topluluk boyutu, SPEA ve DOPGA için 25, SPEA2 ve DOPGA2 için 100 olarak seçilmiştir. Seçme mekanizması olarak SUS kullanılmıştır.

Çizelge 4.12, SPEA ve DOPGA yöntemleri için elde edilen ölçüt değerlerini göstermektedir. Şekil 4.65' te ise SPEA ve DOPGA yöntemlerinin ceza ve ödül değerlerine göre elde edilen boxplot gösterilimleri verilmektedir. Ceza ve ödül ölçütlerine göre daha iyi olan yani hem ceza hem de ödül ölçütü daha düşük olan DOPGA yöntemi, SPEA yöntemine göre daha iyi sıralama yeteneğine sahiptir. Bu sonuçlara göre DOPGA yönteminin Pareto-optimal cepheye yakınsama ve bu cephe üzerinde dağılım açısından da SPEA yönteminde daha iyi olması beklenmektedir. Yakınsama ve dağılım ölçütlerinin sonuçlarına göre de DOPGA yöntemi başarımlı açısından SPEA yönteminden daha iyidir.

Çizelge 4.12. SPEA ve DOPGA yöntemleri için ölçüt değerleri

		CEZA (P)	ÖDÜL (R)	GD	Δ
SPEA	<i>ZDT1</i>	60.9579	7.1189	0.1277	0.6184
	<i>ZDT2</i>	61.2675	7.1624	0.3046	0.7556
	<i>ZDT3</i>	62.9278	7.2240	0.0977	0.6228
	<i>ZDT6</i>	61.1981	7.0937	2.0635	0.9112
DOPGA	<i>ZDT1</i>	31.9688	3.0415	0.0156	0.5797
	<i>ZDT2</i>	14.4255	2.9271	0.0223	0.5922
	<i>ZDT3</i>	34.4204	3.5166	0.0097	0.6111
	<i>ZDT6</i>	6.6732	2.6472	0.8116	0.8334

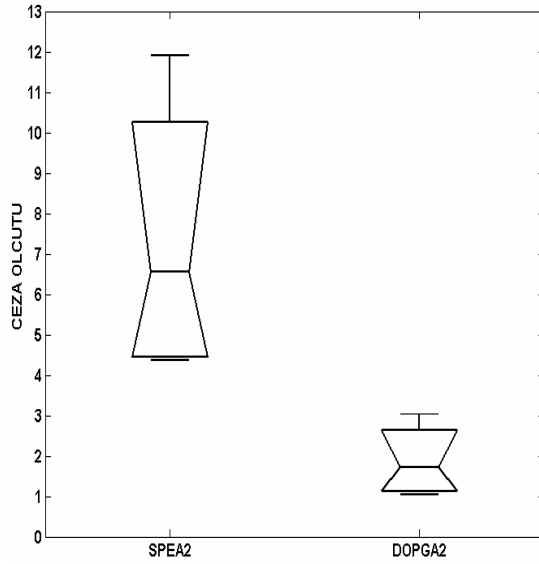
**Şekil 4.65.** Boxplot gösterilimi: (a) Ceza ölçütü, (b) Ödül ölçütü

Çizelge 4.13, SPEA2 ve DOPGA2 yöntemleri için elde edilen ölçüt değerlerini göstermektedir. Şekil 4.66’ da ise SPEA2 ve DOPGA2 yöntemlerinin ceza ve ödül değerlerine göre elde edilen boxplot gösterilimleri verilmektedir. Ceza ve ödül ölçüt değerlerine göre DOPGA2 yönteminin sıralama yeteneği SPEA2 yönteminden daha iyidir. Yakınsama ölçütleri açısından ZDT6 test işlevi hariç, SPEA2 yöntemi DOPGA2 yönteminden daha iyidir. DOPGA2 yöntemi, SPEA2 yöntemine göre dağılım ölçütü açısından tüm test işlevlerinde daha iyidir. Bu yöntemler için ceza ve ödül ölçütleri ile yakınsama ölçütü arasında bir ilişki olmadığı görülmektedir. ÇAGA yöntemlerinin

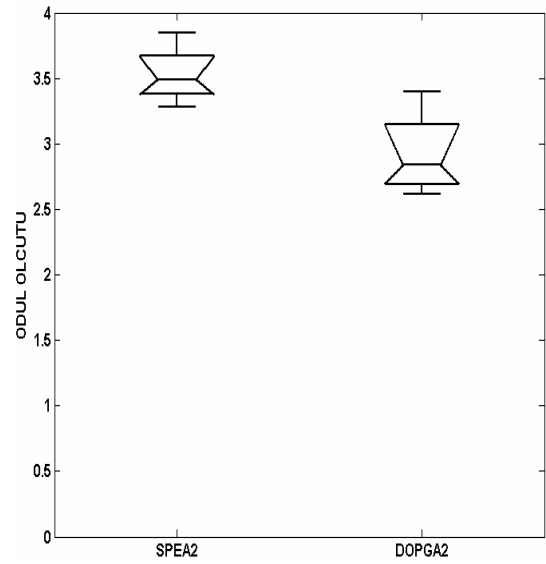
bireylere farklı puan atamalarının yanı sıra, en iyi bireylere atadıkları puan da önemli olabilir. Dağılım ölçütü ile ceza ve ödül ölçütleri arasında bir paralellik vardır, yani ceza ve ödülü düşük olan yöntem olan DOPGA2 yöntemi, dağılım açısından SPEA2 yönteminden daha iyi olmaktadır.

Çizelge 4.13. SPEA2 ve DOPGA2 yöntemleri için ölçüt değerleri

		CEZA (P)	ÖDÜL (R)	GD	Δ
SPEA2	<i>ZDT1</i>	4.3765	3.4980	0.0162	0.6157
	<i>ZDT2</i>	8.6303	3.4804	0.0231	0.6338
	<i>ZDT3</i>	4.4925	3.8453	0.0071	0.6203
	<i>ZDT6</i>	11.9064	3.2777	0.9816	0.8733
DOPGA2	<i>ZDT1</i>	1.1955	2.9062	0.0181	0.5690
	<i>ZDT2</i>	2.2558	2.7662	0.0262	0.5784
	<i>ZDT3</i>	1.0540	3.3938	0.0109	0.5963
	<i>ZDT6</i>	3.0341	2.6191	0.8965	0.8262



(a)



(b)

Şekil 4.66. Boxplot gösterilimi: (a) Ceza ölçütü, (b) Ödül ölçütü

4.5.3. Genel Sonuçlar ve Değerlendirme

Bu bölümde, ÇAGA yöntemlerinin sıralama yeteneklerinin ölçülmesi için iki yeni başarımlı ölçütü önerilmiştir. Bu ölçütler kullanılarak, ÇAGA yöntemlerinin bir topluluktaki bireyleri en çok istenilenden en az istenilene doğru sıralama yetenekleri tespit edilmiştir. Ceza ve ödül ölçütleri en az olan yöntem sıralama yeteneği açısından daha iyidir. Böylece seçme mekanizmasına veya karar vericiye en fazla / nitelikli bilgiyi iletmektedir. Bu ölçütler, test işlevlerinden bağımsızdırlar ve daha önceden bilinse dahi Pareto cephesini kullanmamaktadırlar.

SPEA ve DOPGA yöntemleri ile SPEA2 ve DOPGA2 yöntemleri ayrı ayrı değerlendirilmiştir. Bunun nedeni, SPEA2 ve DOPGA2 yöntemlerinde k . en yakın komşuluk yöntemi ile bulunan yoğunluk bilgisinin puanlamaya dahil edilmesidir. Sonuç olarak, DOPGA yönteminin SPEA yöntemine göre bireyleri en çok istenilenden en az istenilene doğru daha iyi sıraladığı belirlenmiştir. Yakınsama ve dağılım açısından da beklendiği üzere DOPGA yöntemi daha iyidir. DOPGA2 yönteminin sıralama yeteneği açısından SPEA2 yönteminden daha iyi olduğu görülmüştür. Ceza ve ödül ölçütleri ile yakınsama ölçütü arasında bir bağ bulunmamaktadır. Ancak, dağılım ile ceza-ödül arasında doğrusal bir ilişki vardır, yani ceza ve ödülü düşük olan yöntem dağılım açısından daha iyi olmaktadır. Ayrıca seçme mekanizmasının olasılıksal olarak davranması nedeniyle, sıralama bilgileri nitelikli bir biçimde kullanılmamaktadır. Yani hangi birey çiftlerinin çaprazlamaya tabi tutulacağı kesin olarak bilinmediğinden dolayı olasılıksal bir seçim yapılmaktadır. Bu da seçme mekanizmasının puanlama mekanizmasından gelen bilgileri kullanmadığı anlamına gelir. Bu durumda, olasılıksal olarak çalışan seçme mekanizması yerine uzman sistem veya kural tabanlı bir seçme mekanizması kullanıldığında ceza ve ödül ölçütleri daha anlamlı olacaktır. Yukarıdaki tutarsız sonuçları değerlendirmeden önce, GA'nın dinamik davranışı hakkında genel bir çerçeve çizmek adına hayali bir futbol maçı yapılsın.

GA' larda bir topluluğun dinamik davranışını açıklamak zordur. Bu soruyu yanıtlamak için hayali bir örnek verilsin [Eminoğlu, 2003]. GA' nın yapısındaki rasgelelelikten dolayı, bir topluluğun bir nesilden bir nesile nasıl hareket ettiğini belirlemek zordur. Bazen büyük bir problem bazı varsayımlarla daha küçük bir probleme indirgenebilir. Dolayısıyla, bir topluluğun hareketini incelemektense, bir çift bireyin hareketini incelemek daha kolay olabilir. Bu amaçla, olası bir 100 bireylik

topluluk iki bireye (bir çift ebeveyne) düşürülsün. Böylece, ebeveynlerin rasgele seçilmesi işlemine gerek kalmaz. İki bireyden oluşan ana topluluğun, çaprazlama ve mutasyon işlevleri ile bir nesilden diğer nesile nasıl hareket edeceğini merak edilmektedir. Bu bireyler, iki boyutlu bir yüzeye örneğin bir futbol sahasına rasgele yerleştirilsin. Örneğin, bir birey penaltı noktasında, diğeri de diğeri penaltı noktasında olsun. Her birey, sahanın x ve y eksenlerini gösteren iki değişkenden oluşsun. Ayrıca her bireyin sahadaki herhangi bir noktaya gidebilme yeteneği olsun. Çaprazlama ve mutasyon işlevlerini sırasıyla 1 ve 0.1 olasılıkla uygulansın ve bu hayali program yeterince uzun bir süre (örneğin bir milyon adım) çalıştırılsın. Her adımda, bu bireyler sahanın herhangi bir noktasına yerleşecek iki yeni birey üreteceklerdir. Sonuç olarak, sahada iki milyon adet birey olacaktır. Hayali bir gözlemci (örneğin hakem) sahaya baktığı zaman, ya düzenli olarak sahaya dağılmış bireyler ya da dağınık bir biçimde duran bireyler görecektir. Yeterli süre verildiğinde, ebeveynler istenilen bireyleri üretebilir. İkinci bir durum, belirli bir düzenin oluşmasıdır. Bunun anlamı, daha önceden tanımlanan olasılıklara rağmen, yeni bireylerin oluşturulmasında ebeveynlerin konumlarının önemli bir rol oynadığıdır. Eğer bireylerin başlangıç konumları, sonucun elde edilmesinde belirleyici bir rol oynuyorsa, farklı konumlar kullanılarak bir çok hayali benzetim yapılabilir. Eğer bu durum doğru ise, bu hayali oyunun ikinci aşamasına geçilebilir. İki birey sahaya rasgele yerleştirilsin ve çaprazlama ve mutasyon işlevlerini uygulansın. Elde edilen yeni bireyler, bir sonraki adımdaki bireyler (erkek çocuk ve kız çocuk) olacaktır. İlk ebeveynleri silmeden önce, baba ile oğlu sahada birleştiren hayali bir kırmızı çizgi ve anne ile kızı birleştiren hayali mavi bir çizgi çizilsin. Bu işlem, bir milyon adım tekrarlınsın. Sonuç olarak, tüm kırmızı ve mavi çizgiler birleştirilsin. Bu çizgiler bireylerin ilk nesilden son nesile doğru olan hareketlerini gösterecektir.

Üçüncü aşama, basit bir amaç fonksiyonu (bir en küçükleme problemi) ve bir çift bireyden oluşsun. Bu durumun farkı, ebeveynlerin iki erkek veya iki kız çocuk oluşturmalarına izin verilmesidir. Erkek bireyler ve kız bireyler kendi aralarında bir sonraki nesildeki ebeveynler olabilmek için amaç fonksiyonuna göre birbirleriyle yarışacaktır. En güçlü (amaç fonksiyonuna göre en iyi) erkek ve kız çocuk bir sonraki adımdaki ebeveynler olacaktır. Bu durumda baba ile kazanan erkek çocuk arasına kırmızı bir çizgi, anne ile kazanan kız çocuk arasına da mavi bir çizgi çizilecektir. Bu durumda hayali gözlemcinin bireylerin bir nesilden diğer nesile nasıl hareket ettiğini

anlaması çok zor olacaktır. Bu sorunun yanıtını vermek çok zordur. İkinci ve üçüncü aşamada, gözlemci elde edilen çizgilerden bir sonuç çıkarmakta oldukça zorlanacaktır.

Bu hayali oyun, (bir futbol sahasında topu takım arkadaşına atmaya çalışan bir futbolcu ve bu olaya göre karar veren bir hakem arasındaki benzeşim kullanılarak) yukarıda belirtilen kısıtlamalar olmadan bir topluluğun bir nesilden diğer nesile nasıl hareket ettiğinin (hangi birey çiftlerinin çaprazlama ve mutasyon işlevlerine uğratılması gerektiğinin) belirlenmesinin zor olduğunu göstermektedir.

4.6. ÇAGA' lar İçin Kullanıcı Arayüzü Tasarımı

ÇAGA yöntemlerini bir arada bulunduran bir kullanıcı arayüzü (graphic user interface, GUI) tasarlanmıştır. GUI, Matlab paket programında oluşturulmuştur. Bu GUI, literatürde yer alan ve bu tezde kullanılan ÇAGA yöntemleri ile, tezde önerilen ÇAGA yöntemlerinin kullanıcının belirlediği parametrelere göre çalıştırılmasına ve işlem sonunda hesaplanan ölçütler ile kullanıcının yöntemleri karşılaştırmasına olanak sağlamaktadır. GUI özel olarak *MATlab Genetic Algorithm Toolbox (MATGAT)* olarak adlandırılmıştır.

GUI' de yer alan 16 adet ÇAGA yöntemi sırasıyla şöyledir: DOPGA, DOPGA+, DOPGA2, SPEA, SPEA2, NSGA, MOGA, EFE_DOPGA, EFE_DOPGA+, EFE_SPEA, EFE_SPEA2, NSGAmođ, SPEAmođ1, SPEAmođ2, SPEAmođ3, SPEAmođ4.

Kullanıcılar programı başlatmadan önce GA parametrelerini girmek zorundadırlar. GUI' de kullanıcı tarafından giriş yapılacak GA parametreleri şunlardır: topluluk boyutu (population size), parametre sayısı (number of parameters), parametrelerin alt sınırı (lower limit of parameters), parametrelerin üst sınırı (upper limit of parameters), bit sayısı (number of bits), arşiv boyutu (archive size), mutasyon olasılığı (mutation probability), çaprazlama olasılığı (crossover probability), nesil / adım sayısı (generation), yarıçap boyutu (niche size, yalnızca NSGA ve MOGA için), turnuva boyutu (tournament size, yalnızca turnuva seçimi kullanıldığında), gama değeri (gamma value, GDPÖ kullanılmak istenirse girilir) ve etkin mutasyon olasılığı (effective mutation probability, yalnızca etkin elitizm için kullanılır).

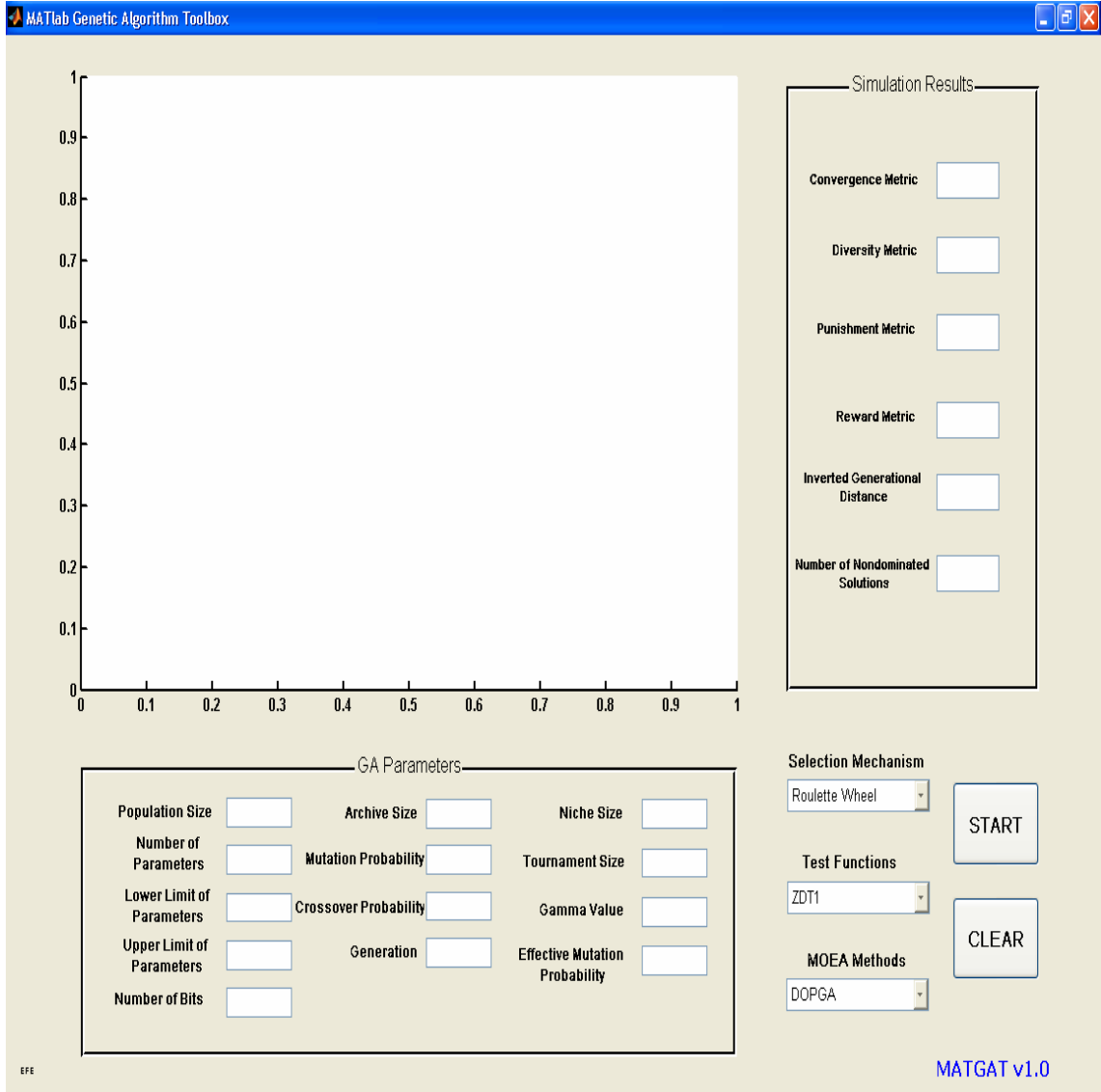
Kullanıcılar, dört farklı ve sık kullanılan test işlevini kullanabilirler. Ayrıca, literatürde yer alan üç seçme mekanizmasından (RWS, SUS, TS) birini seçebilirler.

Parametreler girildikten sonra, ÇAGA yöntemi, seçme mekanizması ve test işlevi ilgili menüler kullanılarak seçilir ve START tuşuna basılır. Girilen adım sayısı sonunda, test işlevinin Pareto-optimal cephesi ve ÇAGA yöntemi tarafından bulunan Pareto bireyler ekrana çizdirilir. Ayrıca, bazı ÇAGA başarımlarının sonuçları ekranda görüntülenir. Böylece kullanıcı seçtiği yöntemin başarımı hakkında bilgi sahibi olur. Bu GUI' de 6 adet başarımların ölçütü hesaplanmaktadır: GD (yakınsama ölçütü), Δ (çeşitlilik/dağılım) ölçütü, tezde önerilen ceza (punishment) ve ödül (reward) ölçütleri, IGD ölçütü [Zitzler ve ark., 2003] ve ÇAGA tarafından bulunan Pareto birey sayısı.

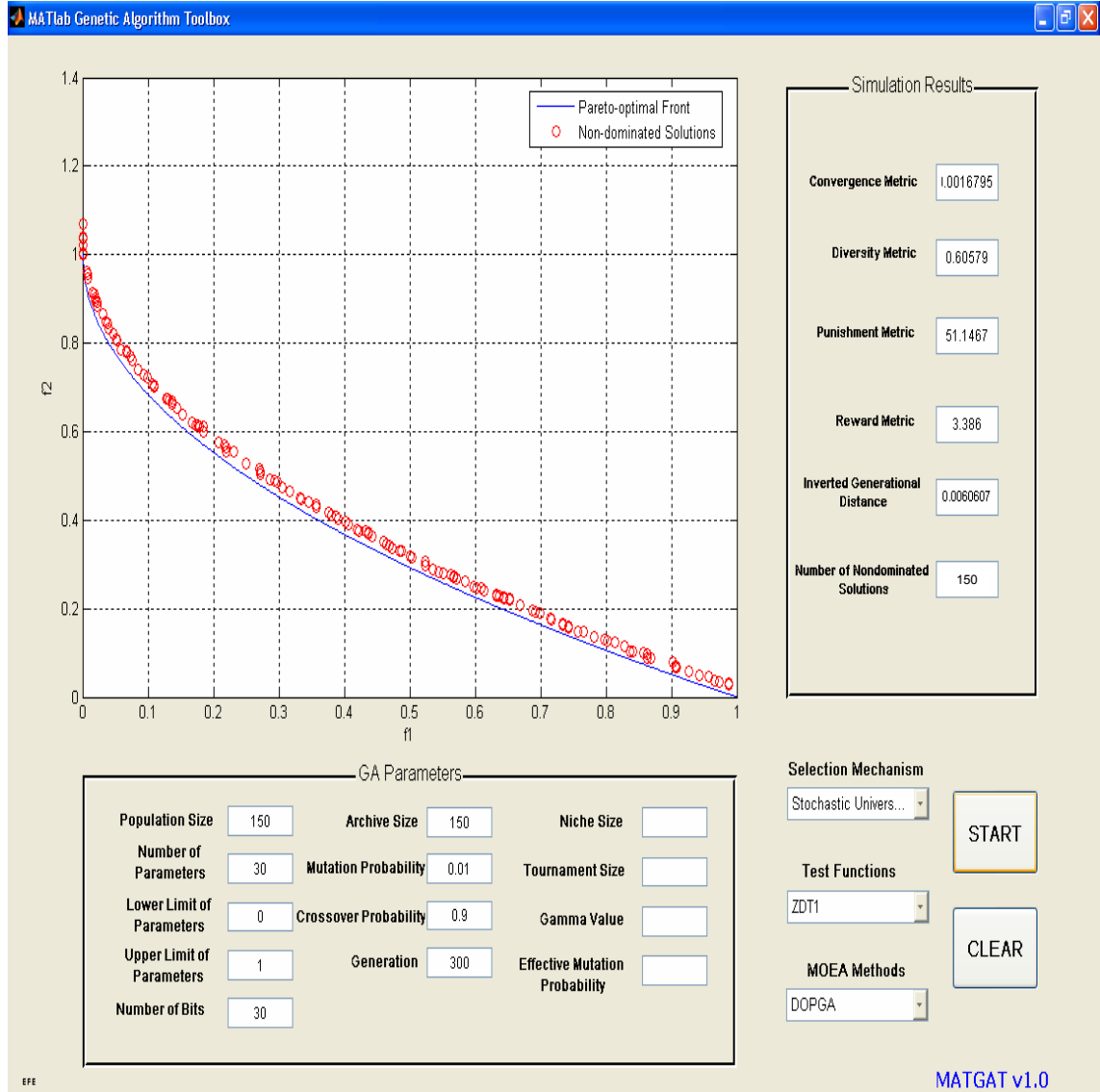
Yeni bir benzetim yapılmadan önce CLEAR tuşuna basılarak ekran temizlenebilir. Elde edilen sonuç grafiği kaydedilerek kullanılabilir.

Şekil 4.67 ve Şekil 4.68' de MATGAT v1.0 GUI' sinin iki adet ekran görüntüsü (çalıştırılmadan önce ve çalıştırdıktan sonra) verilmektedir. Bu, GUI üzerinden gelecekte çeşitli geliştirmeler yapılacaktır. Yeni ÇAGA yöntemleri, başarımların ölçütleri ve test işlevleri (kısıtlamalı ve kısıtlamasız) eklenecektir. Ayrıca, GA literatüründe kullanılan gösterim şekilleri (boxplot, başarımların yüzeyi vb.) GUI' ye eklenecektir. Kullanıcının kendi problemini de kullanabilmesi için alt programlar eklenecektir. Daha sonra, GUI tüm topluluk tabanlı yöntemleri (PSO, DE, ACO vb.) içerisinde barındıran bir yapıya dönüştürülecektir.

Genetik algoritmalarla ilgili ticari bazı yazılımlar da bulunmaktadır. Hartmut Pohlheim tarafından geliştirilmiş Geatbx (Genetic and Evolutionary Algorithm Toolbox for MATLAB) yazılımı, bunlara örnek olarak verilebilir (www.geatbx.com, Erişim zamanı: 04.01.2007). Bu yazılım ile tek amaçlı ve bazı çok amaçlı genetik algoritma yöntemleri kullanılarak optimizasyon problemleri çözülebilmektedir. Tezde geliştirilen MATGAT programının da güncellenerek (farklı özellikler ve programlar vb. eklenmesiyle) ticari bir yazılım haline getirilmesi amaçlanmaktadır.



Şekil 4.67. MATGAT v1.0 çalıştırılmadan önceki ekran görüntüsü



Şekil 4.68. MATGAT v1.0 çalıştırıldıktan sonra ekran görüntüsü

5. SONUÇ VE ÖNERİLER

Bu tezde, iki yeni ÇAGA yöntemi (DOPGA+ ve DOPGA2) önerilmektedir. Bu yöntemler daha önce önerilmiş olan DOPGA [Eminoğlu, 2003] yönteminde bazı değişiklikler yapılarak elde edilmiştir. Önerilen bu yöntemlerin ana özelliği; bireylerin baskınlık gücünün puanlama mekanizması içine gömülmesidir. Yöntemler birbirinden puanlama mekanizmaları veya elitizm mekanizmaları açısından farklılık göstermektedirler. DOPGA yöntemi, herhangi bir ÇAGA yöntemi ile başarımlar ölçütleri kullanılarak karşılaştırılmamıştır. Bu tezde hem DOPGA, hem de DOPGA+ ve DOPGA2 çeşitli test işlevleri kullanılarak literatürde sık kullanılan SPEA ve SPEA2 yöntemleri ile karşılaştırılmışlardır. Sonuç olarak, önerilen yöntemlerin, mevcut SPEA ve SPEA2 yöntemlerinden Pareto-optimal cepheye yakınsama ve çeşitlilik açısından daha iyi oldukları tespit edilmiştir. Bu üç yöntemin literatürde yer alan diğer ÇAGA yöntemleriyle aynı parametreler altında karşılaştırılması, üç veya daha fazla amaç içeren problemlere uygulanması, kısıtlı test problemlerine uygulanması ve endüstriyel problemlere uygulanması gelecek çalışma alanlarından bazıları olacaktır.

NFL (No Free Lunch) teoremine göre, hiçbir algoritma her zaman en iyi olamaz, ya da bir algoritma her problemde en iyi olamaz [Wolpert ve Macready, 1997]. Yani literatürde olan yöntemler ve bu tezde önerilen yöntemler, her test işlevinde veya her optimizasyon problemde en iyi olamaz. Her yöntemin diğer yöntemlerden daha iyi olabileceği bir problem vardır.

Bir topluluktaki bireylerin ÇAGA yöntemleri tarafından puanlanmasının birçok yolu vardır. Çünkü çok amaçlı bir düzlemdeki bir birey yalnızca bir puan değeri ile temsil edilmektedir. Bu yüzden, literatürde birçok ÇAGA yöntemi önerilmiştir ve önerilmeye devam edilmektedir. Önerilen yöntemler üzerinde de birtakım iyileştirmeler yapmak mümkündür. Bu çalışmada, NSGA ve SPEA yöntemlerinin puan atama mekanizmaları, orjinal yöntemlerin ürettiğinden daha iyi ve nitelikli bilgi (puan) üretebilmeleri amacıyla hesapsal ve yapısal olarak değiştirilmiştir (iyileştirilmiştir). İyileştirmeler iki temel biçimde yapılmıştır: **i)** k . en yakın komşuluk yöntemiyle üretilen yoğunluk bilgisi eklenmesi, **ii)** baskınlık gücü kavramı. Yöntemler ve iyileştirmeleri dört farklı ve sık kullanılan test işlevi kullanılarak test edilmişlerdir. Önerilen beş iyileştirme (NSGAmođ, SPEAmođ1, SPEAmođ2, SPEAmođ3, SPEAmođ4) orjinal yöntemlere göre Pareto cephesine daha fazla yakınsamakta ve bu cephe üzerinde

bireylerin daha düzgün dağılmasını sağlamaktadırlar. Ayrıca, k . en yakın komşuluk yöntemiyle oluşturulan yoğunluk bilgisi tekniği, paylaşım tabanlı çeşitlilik artırma tekniğinden daha iyi başarımlar göstermektedir. Yapısal olarak değiştirilmiş (uzman sistem benzeri) bir seçme mekanizması tasarımı gelecek çalışma alanlarından birisi olacaktır. Böylece, üretilen puan bilgilerinin daha nitelikli bir şekilde değerlendirilmesi mümkün olabilir.

Literatürde yer alan tüm yöntemler üzerinde bazı iyileştirmeler yapılması mümkündür. Böylece yöntemlerin yakınsama ve/veya dağılım yetenekleri artırılabilir ve yöntemler daha etkin hale getirilebilir. Bu çalışmada, yalnızca beş adet iyileştirme verilmekle beraber daha çok sayıda iyileştirme yapılması da mümkündür. Yöntemlerin işlem sürelerinin kısaltılması için de iyileştirmeler yapılması, gelecek çalışma alanlarından birisi olabilir.

GA yöntemlerinin başarımlarını artırmanın bir başka yolu da seçme mekanizmasından önce bireylerin seçilme baskısının değiştirilmesidir. Bu amaçla literatürde birçok puan ölçekleme yöntemi kullanılmıştır. Bu çalışmada ise literatürde yer alan üstel puan ölçekleme yönteminin değiştirilmiş bir versiyonu olan, gama düzeltmesi ile puan ölçekleme (GDPÖ) yöntemi önerilmiştir. Bu yöntem ilk olarak, tek amaçlı bir GA' da altı sabit gama değeri kullanılarak iki test işlevi üzerinde çalıştırılmıştır. Sonuç olarak, gama değerinin değişmesiyle yakınsama hızının değiştiği gözlenmiştir. Ayrıca, her iki test işlevinde de $\gamma=10$ seçildiğinde daha iyi ve daha hızlı bir yakınsama sağlandığı görülmüştür.

GDPÖ yönteminin bir topluluktaki bireylerin seçilme baskısının değiştirebileceği dolayısıyla ÇAGA yöntemlerinin yakınsama yeteneğini artırabileceği görülmüştür. Sabit gama (γ) değerleri kullanılarak GDPÖ yöntemi, mevcut ve önerilen ÇAGA yöntemlerine uygulanmıştır. Bu durumda elde edilen genel sonuçlar şöyledir: i) ÇAGA yöntemlerinin başarımları gama (γ) değerine bağlı olarak değişmektedir, ii) SPEA yöntemi $\gamma=7$ ile uygulanan başarımlar ölçütlerine göre en iyi yakınsamayı ve dağılımı sağlamaktadır, iii) $\gamma=2$ ile SPEA2 ve DOPGA2, $\gamma=3$ ile DOPGA (ZDT1' de $\gamma=4$ ile) en iyi yakınsamayı sağlamaktadırlar, iv) $\gamma=1$ ile yani orjinal sonuçlarla tüm yöntemler değerlendirildiğinde, DOPGA2 yönteminin ZDT1, ZDT2 ve ZDT3 test işlevlerinde, DOPGA' nın ise ZDT6 test işlevinde en iyi sonuçları verdiği görülmüştür, v) tüm gama değerlerine göre değerlendirme yapıldığında ise, ZDT1 test işlevinde $\gamma=2$ ile DOPGA2; ZDT2, ZDT3 ve ZDT6 test işlevlerinde ise $\gamma=3$ ile DOPGA tüm yöntemler arasında en

iyi sonuçları vermektedir, vi) dağılım (veya çeşitlilik) farklı gama değerlerinde önemli derecede değişmemektedir, vii) GDPÖ yalnızca puan orantılı bir seçme mekanizması (örneğin SUS) kullanıldığında zaman etkilidir, viii) TS kullanıldığında zaman GDPÖ etkili değildir.

Uyarlanabilir GDPÖ yöntemi (dinamik γ değerleri) kullanıldığında SPEA (ZDT1 hariç) ve DOPGA (ZDT1 ve ZDT3 hariç) yöntemleri sabit gama kullanıldığında elde edilen sonuçlardan daha iyi sonuçlar vermiştir. SPEA2 ve DOPGA2 ise sabit durumdakinden iyi sonuç vermemişlerdir. Hem uyarlanabilir hem de sabit GDPÖ sonuçları dikkate alındığında, ZDT2 ve ZDT6 test işlevlerinde **DOPGA + Adaptif GDPÖ**, ZDT1 test işlevinde DOPGA2, ZDT3 test işlevinde ise DOPGA; tüm yöntemlerden daha iyi yakınsama sonuçları vermektedir.

Sonuç olarak, GDPÖ bloğu, herhangi bir ek işlem yükü getirmeden ve yöntemlerin yapılarını değiştirmeden literatürde yer alan tüm ÇAGA yöntemlerine uygulanabilir ve yöntemlerin yakınsama başarımı artırılabilir. Ayrıca, benzetim sonuçlarından da anlaşılabileceği üzere, tüm ÇAGA yöntemleri hala iyileştirmelere açıktırlar.

Elitizm mekanizması GA'ların başarımını önemli ölçüde etkilemektedir. Elitizm mekanizması olan yöntemler, olmayanlara göre daha iyi başarımlar göstermektedirler. Bu nedenle, elitizm mekanizmasında yapılabilecek en ufak bir iyileştirme bile, başarımları etkileyecektir. Bu çalışmada, literatürde yer alan pasif elitizm mekanizması önerilen etkin elitizm mekanizmasıyla değiştirilmiştir. Elit bireyler en iyi bireylerdir ve dolayısıyla çözüme en yakın bireylerdir. Bu yüzden, arşivdeki elit bireylerin çaprazlama / mutasyon işlevleri ile uyarılması neticesinde kendilerinden yakınsama ve düzgün dağılım açısından daha iyi bireyler üretilebilmesi sağlanabilir. Bu amaçla klasik *sakla/aktar* elitizm yapısı, *sakla/uyar/aktar* bir etkin elitizm yapısı kullanılarak iyileştirilmiştir. Böylece, etkin elitizm mekanizmasında ana topluluk ve ikincil topluluktaki bireyler çözüm üretebilmektedir.

Etkin elitizm mekanizması, literatürde yer alan SPEA ve SPEA2 yöntemleriyle, önerilen DOPGA ve DOPGA+ yöntemlerine uygulanmıştır. Benzetimler sonucunda, EFE_SPEA yönteminin SPEA yönteminden yakınsama ve düzgün dağılım açısından daha iyi olduğu görülmüştür. Yalnızca, ZDT6 test işlevinde SPEA yöntemi EFE_SPEA' dan daha iyi bir dağılım sağlamıştır. EFE_SPEA2 yönteminin SPEA2 yönteminden yakınsama açısından daha iyi olduğu görülmüştür. ZDT2 test işlevi hariç tüm test

işlevlerinde EFE_SPEA2 yöntemi SPEA2' den daha düzgün bir şekilde Pareto cephesine dağılmıştır. EFE_DOPGA yöntemi ZDT1 ve ZDT2' de, DOPGA yöntemi ise ZDT3 ve ZDT6' da daha iyi yakınsama sağlamışlardır. DOPGA, EFE_DOPGA' dan daha iyi düzgün dağılım ölçütlerine sahiptir. EFE_DOPGA+, tüm test işlevlerinde DOPGA+ yönteminden iyi yakınsama sağlamaktadır. Ancak DOPGA+ tüm test işlevlerinde düzgün dağılım açısından daha iyidir.

Tüm pasif ve etkin elitizm mekanizmalı yöntemler birlikte değerlendirildiğinde; ZDT1, ZDT3 ve ZDT6' da EFE_DOPGA+, ZDT2' de ise EFE_DOPGA diğer yöntemlerden yakınsama açısından daha iyidir. ZDT1, ZDT2 ve ZDT6' da DOPGA+, ZDT3' te ise EFE_SPEA2 yöntemi diğer tüm yöntemlerden daha iyi olarak Pareto cephesine dağılan bireyler oluşturmuşlardır.

Etkin elitizm mekanizması üzerinden bir takım iyileştirmelere gereksinim duyulmaktadır. Özellikle, mutasyon ve çaprazlama işlevlerinin hangi anda devreye gireceklerinin daha iyi belirlenmesi gerekebilir. Ayrıca elit bireylerin çaprazlanması için kural tabanlı bir uzman sistemin karar verici olarak kullanılması daha iyi sonuçlar elde edilmesini sağlayabilir. Hangi elit bireylerin hangi elit bireyler çaprazlanacağını daha iyi saptanması etkin elitizmin yeteneğini artırabilir. Bunun için hangi bireylerin çaprazlamaya gireceğine dair bilgi veren bir kural tabanının oluşturulması gelecek çalışmalardan birisi olacaktır. Ayrıca etkin elitizm mekanizmasının işlem yükünün azaltılması da bir çalışma konusudur. Önerilen etkin elitizm mekanizması literatürde yer alan ÇAGA yöntemlerine uygulanarak, bu yöntemlerin başarımı genel yapıları değiştirilmeden artırılabilir. Etkin elitizm mekanizmasının tek dezavantajı ÇAGA yöntemlerinin işlem zamanlarında artışa neden olmasıdır.

ÇAGA yöntemlerinin sıralama (puan atama) yeteneklerinin ölçülmesi için iki yeni başarımlı ölçütü önerilmiştir. Bu ölçütler kullanılarak, ÇAGA yöntemlerinin bir topluluktaki bireyleri en çok istenilenden en az istenilene doğru sıralama yetenekleri tespit edilmiştir. Ceza ve ödül ölçüt değerleri en az olan yöntem sıralama yeteneği açısından daha iyidir. Böylece seçme mekanizmasına veya karar vericiye en fazla / nitelikli bilgiyi iletmektedir. Bu ölçütler, test işlevlerinden bağımsızdırlar ve daha önceden bilinse dahi Pareto cephesini kullanmamaktadırlar. ÇAGA karşılaştırmalarında sık kullanılan SPEA yöntemi ile DOPGA yöntemlerinin sıralama yetenekleri karşılaştırılmıştır. Benzetimler sonucunda, DOPGA yönteminin ceza&ödül değerlerinin SPEA yönteminden daha düşük olduğu, dolayısıyla da SPEA yöntemine göre bireyleri

en çok istenilenden en az istenilene doğru daha iyi sıraladığı belirlenmiştir. Yakınsama ve dağılım açısından da beklendiği üzere DOPGA yöntemi daha iyidir.

Puan atama mekanizmalarında k . en yakın komşuluk yönteminden yararlanılarak oluşturulan bir yoğunluk bilgisi kullanan SPEA2 ve DOPGA2 yöntemleri ise ayrı ayrı değerlendirilmiştir. DOPGA2 yönteminin sıralama yeteneği açısından SPEA2 yönteminden daha iyi olduğu görülmüştür. Ancak bu yöntemler için, ceza ve ödül ölçütleri ile yakınsama ölçütü arasında bir bağ bulunamamıştır. Pareto cephesi üzerinde dağılım ile ceza&ödül arasında doğrusal bir ilişki vardır, yani ceza ve ödülü düşük olan yöntem dağılım açısından daha iyi olmaktadır, yani Pareto-optimal cephede daha düzgün dağılmaktadır. Ayrıca seçme mekanizmasının olasılıksal olarak davranması nedeniyle, sıralama bilgileri nitelikli bir biçimde kullanılmamaktadır. Yani hangi birey çiftlerinin çaprazlamaya tabi tutulacağı kesin olarak bilinmediğinden dolayı olasılıksal bir seçim yapılmaktadır. Bu da seçme mekanizmasının puanlama mekanizmasından gelen bilgileri kullanmadığı anlamına gelir. Bu durumda, olasılıksal olarak çalışan seçme mekanizması yerine uzman sistem veya kural tabanlı bir seçme mekanizması kullanıldığında ceza ve ödül ölçütleri daha anlamlı olacaktır.

Uzman sistem veya kural tabanlı bir seçme mekanizmasının geliştirilmesi ile hangi birey çiftlerinin çaprazlama için seçileceği, dolayısıyla da hangi birey çiftlerinin çaprazlanmasıyla kendilerinden daha iyi bireyler elde edilebileceği konusu açıklığa kavuşturulabilir.

Bu çalışmada, literatürde sık kullanılan ve önerilen ÇAGA yöntemlerini içerisinde barındıran ve MATLAB üzerinde çalışabilen bir GUI (toolbox) tasarlanmıştır. GUI' nin bu ilk versiyonunda, kullanıcılar istedikleri ÇAGA yöntemini, seçme mekanizmasını, test işlevini seçerek ve kendi GA parametrelerinin girerek değerlendirme yapabilmektedirler. GUI üzerinde gelecekte çeşitli geliştirmeler yapılacaktır. Yeni ÇAGA yöntemleri, başarımlar ölçütleri ve test işlevleri (kısıtlamalı ve kısıtlamasız) eklenecektir. Ayrıca, GA literatüründe kullanılan gösterilim şekilleri (boxplot, başarımlar yüzeyi vb.) GUI' ye eklenecektir. Kullanıcının kendi problemini de kullanabilmesi için alt programlar eklenecektir. Bu GUI, ÇAGA' lar için geliştirilmiş olsa da literatürde yer alan diğer topluluk tabanlı yöntemlerin de programlanarak GUI' ye eklenmesi optimizasyon ile uğraşan akademik ve endüstriyel kullanıcılar için çok yararlı olacaktır. MATGAT programının güncellenerek ticari bir yazılım haline getirilmesi amaçlanmaktadır.

Son olarak, GA' ların işleyişini geliştirmek ve yapısını anlamak için yapılabilecek bazı geliştirmeler / gelecek çalışma alanları şöyle sıralanabilir:

- i. GA topluluğunun dinamiklerini yani GA' nın nasıl çalıştığını belirlemek, bu sayede GA' nın işleyiş ve etkinliğini değiştirmek, bu amaçla istatistiksel teknikleri kullanmak,
- ii. Elitizm mekanizmasının iyileştirilmesi,
- iii. Çok amaçlı GA' ları gerçek dünya problemlerine (örüntü tanıma, bilgisayarlı animasyon, hücresel sistemler, sanal gerçeklik, kriptografi, biyoinformatik, veri madenciliği vb.) uygulamak,
- iv. Önerilen ÇAGA yöntemlerini kısıtlamalı test problemlerine uygulamak,
- v. GA' ların işlem süresini ve bellek kullanımını azaltmak,
- vi. GA' ları diğer topluluk tabanlı optimizasyon yöntemleriyle (PSO, DE vb.) melezleştirmek (hibritleştirmek),
- vii. Seçme mekanizmasını değiştirmek (en iyi bireylerin seçilmesini sağlamak),
- viii. İki amaçtan daha fazla amaçlı optimizasyon problemlerine uygulamak,
- ix. Parametrelerin kontrolünü kendi kendine sağlayan bir ÇAGA yöntemi tasarlamak (kullanıcı kendisi parametrelerini ayarlamayacak),
- x. ÇAGA tarafından bulunan çözüm önerilerinden en iyisinin seçilmesi için yeni yöntemler geliştirilmesi (çok amaçlı karar verme)
- xi. Hangi bireylerin çaprazlamaya gireceği ve hangi bireylerin eşleşmesi gerektiğine ilişkin kavramların geliştirilmesi,
- xii. Yeni başarımlar ölçütleri (özellikle Pareto-optimal cephenin bilinmediği durumlarda da kullanılabilecek) geliştirilmesi,
- xiii. ÇAGA yöntemlerinin paralelleştirilmesi,
- xiv. Gerçek sayı ile çalışan GA' ların kullanılması,
- xv. Tercih belirleme (preference articulation) için yeni yöntemler geliştirilmesi,
- xvi. ÇAGA yöntemleri kullanılarak bulanık mantıkla denetlenen veya modellenen sistemlerin IF-THEN kurallarının (merkez , genişlik ve ağırlık gibi) ve RBF (radial basis functions) ağırları vb. yapıların son evre optimizasyonunun yapılması,
- xvii. Biyolojik kökenli yeni nesil topluluk tabanlı sürü optimizasyonu yöntemlerinin (PSO, ACO, DE vb.) endüstriyel problemlere uygulanması.

6. KAYNAKLAR

- Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K., 2008.** A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 3, 269-283.
- Bullnheimer B., Hartl R.F., Strauss C., 1997.** A New Rank Based Version of the Ant System: A Computational Study. *Central European Journal for Operations Research and Economics*, 7(1): 25-38.
- Bullnheimer, B., Hartl, R.F., Strauss, C., 1999.** An Improved Ant System Algorithm for The Vehicle Routing Problem". *Annals of Operations Research*, 89, 319-328.
- Chakraborty, U.K., 2008.** *Advances in Differential Evolution*. Springer, 340 s.
- Cerny, V., 1985.** A Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimization Theory and Applications*, 45(1): 41–51.
- Chen, M., Lu, Y., 2008.** A Novel Elitist Multiobjective Optimization Algorithm: Multiobjective Extramal Optimization. *European Journal of Operational Research*, Vol. 188, Issue 3, 637-651.
- Chiu, S., 1994.** Fuzzy Model Identification Based on Cluster Estimation. *Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, 267-278.
- Clerc, M., 2006.** *Particle Swarm Optimization*. Wiley-ISTE, 243 s.
- Coello Coello, C.A., 1999.** A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems*, Vol.1, No.3, 269-308.
- Coello Coello, C.A., 2006.** Evolutionary Multi-objective Optimization: A Historical View of the Field. *IEEE Computational Intelligence Magazine*, 28-36.
- Coello Coello, C.A., Lamont G.B., Veldhuizen, D.V., 2007.** *Evolutionary Algorithms for Solving Multi-Objective Problems*. Second Edition, Springer, 800 s.
- Colorni, A., Dorigo, M., Maniezzo, V., 1992.** Distributed Optimization by Ant Colonies. *Proceedings of the First European Conference on Artificial Life* (Editörler: F. J. Varela ve P. Bourguine), MIT Press, Cambridge, MA, 134–142.
- Corne, D.W., Knowles J.D., Oates, M.J., 2000.** The Pareto Envelope-Based Selection Algorithm for Multi-objective Optimisation. *Lecture Notes in Computer Science*; Vol. 1917, *Proceedings of the 6th International Conference on Parallel Problem*

- Solving from Nature, Springer-Verlag, London, İngiltere, 839 – 848.
- Czyzak, P., Jaskiewicz, A., 1997a.** The Multiobjective Metaheuristic Approach for Optimization of Complex Manufacturing Systems. Proceedings of the XIIth International Conference (Editörler: G. Fandel ve T. Gal), Hagen, Germany, Springer-Verlag, 591–592.
- Czyzak, P., Jaskiewicz, A., 1997b.** Pareto Simulated Annealing. Proceedings of the XIIth International Conference (Editörler: G. Fandel ve T. Gal), Hagen, Germany, Springer-Verlag, 297–307.
- Deb, K., 1989a.** Genetic Algorithms in Multi-Modal Function Optimization. Yüksek Lisans Tezi, University of Alabama, Tuscaloosa.
- Deb, K., Goldberg, D.E., 1989b.** An Investigation of Niche and Species Formation in Genetic Function Optimization. Proceedings of the Third International Conference on Genetic Algorithms (Editör: J. D. Schaffer), San Mateo, California, ABD, Morgan Kaufmann Publishers, 42–50.
- Deb, K., 1999.** Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, Vol.7, No.3, 205-230.
- Deb, K., 2001.** *Multiobjective Optimization Using Evolutionary Algorithms*. Wiley, First Edition, Chichester, U.K., 518 s.
- Deb, K., Thiele, L., Laumanns, M., Zitzler, E., 2002a.** Scalable Test Problems for Evolutionary Multi-Objective Optimization. Proceedings of the 2002 Congress on Evolutionary Computation CEC '02, Honolulu, ABD, 825-830.
- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002b.** A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, Vol.6, No.2, 182-197.
- Dorigo, M., Maniezzo, V., Colorni, A., 1991.** Positive Feedback as a Search Strategy. Technical Report 91-016. Dipartimento di Elettronica, Politecnico di Milano, Italy.
- Dorigo, M., Maniezzo, V., Colorni, A., 1996.** The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1): 29–41.

- Dorigo, M., Gambardella, L.M., 1997.** Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1, 53-66.
- Dorigo, M., Caro, G. D., 1999.** The Ant Colony Optimization Meta-Heuristic. *New Ideas in Optimization* (Editörler: D. Corne, M. Dorigo, ve F. Glover), McGraw-Hill, 493 s.
- Dorigo, M., Stützle, T., 2004.** *Ant Colony Optimization*. The MIT Press, USA, 305 s.
- Dowsland, K. A., 1993.** Simulated Annealing. *Modern Heuristic Techniques for Combinatorial Problems* (Editör: C. R. Reeves), Chp.2, John Wiley & Sons, 20-69.
- Eberhart R., Shi Y., 1998.** Comparison between Genetic Algorithms and Particle Swarm Optimization. *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, (Editörler: V. W. Porto, N. Saravanan, D. Waagen, ve A. Eibe), Springer-Verlag, 611–619.
- Eminoğlu, İ., 2003.** *Computational Intelligence Methods: Generic Interpretations, Optimisation and Application*. Doktora Tezi, The University of Sheffield, 268 s.
- Engelbrecht, A. P., 2007.** *Computational Intelligence: An Introduction, 2nd Edition*. John Wiley & Sons, 628 s.
- Ergul, E.U., Usta, H., Eminoglu, I., 2009a.** Effective Usage of Gamma Correction Operator in Genetic Algorithms. *International Symposium on INnovations in Intelligent SysTems and Applications – INISTA 2009*, Trabzon, Turkey, June 29-July 1, 447-451.
- Ergul, E.U., Usta, H., Eminoglu, I., 2009b.** Effective Elitism Mechanism for Multi-objective Evolutionary Algorithms. *International Symposium on INnovations in Intelligent SysTems and Applications – INISTA 2009*, Trabzon, Turkey, June 29-July 1, 502-506.
- Fonseca, C.M., Fleming, P.J., 1993.** Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, 416-423.
- Fonseca, C.M., Fleming, P.J., 1995.** An Overview of Evolutionary Algorithms in Multi-objective Optimization. *Evolutionary Computation*, Vol.3, No.1, 1-16.
- Fonseca, C.M., Fleming, P.J., 1996.** On The Performance Assessment and Comparison of Stochastic Multiobjective Optimizers. In *Proceedings of Parallel ProblemSolving from Nature IV (PPSN-IV)*, 584-593.

- Fonseca, C.M., Fleming, P.J., 1998.** Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms-Part 1-2. IEEE Transactions on System, Man and Cybernetics, Vol. 28, No.1, 26-47.
- Gambardella, L. M., Dorigo, M., 1995.** Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. Proceedings of the 12th International Conference on Machine Learning (Editörler: A. Prieditis ve S. Russell), Morgan Kaufmann, USA, 252–260.
- Gao, J., 2009.** WBMOIGA: Weight-Based Multiobjective Immune Genetic Algorithm and Its Application. Proceedings of the 2009 IEEE International Conference on Information and Automation, China, 1-6.
- Gen, M., Liu B., Ida, K., 1996.** Evolution Program for Deterministic and Stochastic Optimizations. European Journal of Operational Research, 94, 618-625.
- Gen M., Cheng, R., 1997.** *Genetic Algorithms and Engineering Design*. John Wiley & Sons, 432 s.
- Ghosh, A., Dehuri, S., 2004.** Evolutionary Algorithms for Multi-Criterion Optimization: A Survey. International Journal of Computing & Information Sciences, Vol. 2, No. 1, 38-57.
- Gillies, A.M., 1985.** Machine Learning Procedures for Generating Image Domain Feature Detectors. Doktora Tezi, University of Michigan, ABD.
- Goldberg, D.E., Richardson, J., 1987.** Genetic Algorithms with Sharing for Multimodal Function Optimization. In Proceedings of The First International Conference on Genetic Algorithms and Their Applications, 41-49.
- Goldberg, D.E., 1989.** *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, ABD, 432 s.
- Gonzales, R.C., Woods, R.E., 2008.** *Digital Image Processing*, 3rd Edition. Pearson Prentice Hall, New Jersey, USA, 954 s.
- Hajela, P., Lin, C.Y., 1992.** Genetic Search Strategies in Multi-criterion Optimal Design. Structral Optimization, 4(2), 99-107.
- Hansen, M.P., Jaskiewicz, A., 1998.** Evaluating The Quality of Approximations to The Non-dominated Set. Technical Report IMM-REP-1998-7, Institute of Mathematical Modelling, Technical Universtiy of Denmark, Lyngby.
- Holland, J., 1975.** *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, ABD.

- Hopfield, J.J., 1982.** Neural Networks and Physical Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Sciences, Vol. 79, 2554-2558.
- Hopgood, A., 2001.** *Intelligent Systems for Engineers and Scientists, Second Edition.* CRC Press, 488 s.
- Hopgood A., Mierzejewska, A., 2008.** Transform Ranking: A New Method for Fitness Scaling in Genetic Algorithms. Proceedings AI-2008: Research and Development in Intelligent Systems XXV, 349-354.
- Horn, J., Nafpliotis, N., Goldberg, D.E., 1993.** A Niche Pareto Genetic Algorithm for Multiobjective Optimization. Proceedings of the First IEEE Conference on Evolutionary Computation, 82-87.
- Hu, X., Eberhart, R., 2002.** Multiobjective Optimization Using Dynamic Neighborhood Particle Swarm Optimization. In Congress on Evolutionary Computation (CEC'2002), Volume 2, Piscataway, New Jersey, 1677–1681.
- Huband, S., Hingston, P., Barone, L., While, L., 2006.** A Review of Multiobjective Test Problems and A Scalable Test Problem Toolkit. IEEE Transactions on Evolutionary Computation, Vol.10, No.5, 1-30.
- Ishibuchi, H., Murata, T., 1998.** A Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling. IEEE Transactions on System, Man and Cybernetics, Vol.28, No.3, 392-403.
- Ishibuchi, H., Shibata, Y., 2004.** Mating Scheme for Controlling the Diversity-Convergence Balance for Multiobjective Optimization. Proceedings of 2004 Genetic and Evolutionary Computation Conference, Part I, Seattle, ABD, 1259-1271.
- Jensen, M. T., 2003.** Reducing the Run –Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms. IEEE Transactions on Evolutionary Computation, Vol.7, No.5, 503-515.
- Karaboğa, D., 2004.** *Yapay Zeka Optimizasyon Algoritmaları.* Nobel Yayın Dağıtım, 199 s.
- Kennedy, J., Eberhart, R.C., 1995.** Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Piscataway, New Jersey, 1942–1948.

- Kennedy, J., Eberhart, R.C., 1997.** A Discrete Binary Version of the Particle Swarm Algorithm. In Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Piscataway, New Jersey, 4104–4109.
- Kennedy, J., Eberhart, R.C., 2001.** *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 512 s.
- Keskintürk, T., Söyler, H., 2006.** Global Karınca Kolonisi Optimizasyonu. Gazi Üniv. Müh. Mim. Fak. Dergisi, Cilt 21, No 4, 689-698.
- Kirkpatrick, S., Gelatt C.D., Vecchi, M.P., 1983.** Optimization by Simulated Annealing. Science, Vol. 220, 671-680.
- Knowles, J.D., Corne, D. W., 2000.** Approximating the Non-Dominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation Journal 8(2), 149-172.
- Knowles, J.D., Corne, D. W., 2002.** On Metrics for Comparing Nondominated Sets. Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02, Vol. 1, 711-716.
- Kreinovich, V., Quintana C., Fuentes, O., 1993.** Genetic Algorithms: What Fitness Scaling is Optimal? Cybernetics and Systems, 24, 9–26.
- Kursawe, F., 1990.** A Variant of Evolution Strategies for Vector Optimization. Parallel Problem Solving from Nature I (PPSN-I), 193-197.
- Lazinica, A., 2009.** *Particle Swarm Optimization*. In-Tech basımevi, Hırvatistan, 476 s.
- Leclerc F., Potvin, J.Y., 1995.** A Fitness Scaling Method Based on a Span Measure. IEEE International Conference on Evolutionary Algorithms, Vol.2, 561-565.
- Lim, C.P., Lakhmi, C.J., Dehuri, S., 2009.** *Innovations in Swarm Intelligence*. Springer-Verlag, 253 s.
- Linkens, D.A., Nyongesa, H.O., 1995.** Genetic Algorithms for Fuzzy Control-Part:1-2. IEEE Proceedings of Control Theory Applications, Vol.142, No.3, 161-185.
- Lis, J., Eiben, A.E., 1996.** A Multi-Sexual Genetic Algorithm for Multiobjective Optimization. Proceedings of the 1996 International Conference on Evolutionary Computation (Editörler: T. Fukuda and T. Furuhashi), Nagoya, Japonya, 59–64.
- Mariano, C. E., Morales, E., 1999a.** MOAQ: an Ant-Q Algorithm for Multiple Objective Optimization Problems. Genetic and Evolutionary Computing Conference (GECCO 99) (Editörler: W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela ve R. E. Smith), Volume 1, San Francisco,

- California, ABD, Morgan Kaufmann, pages 894–901.
- Mariano, C. E., Morales, E., 1999b.** A Multiple Objective Ant-Q Algorithm for the Design of Water Distribution Irrigation Networks. Technical Report HC-9904, Instituto Mexicano de Tecnologia del Agua, 1-40.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., 1953.** Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6): 1087–1092.
- Mezura-Montes, E., Velazquez-Reyes, J., Coello Coello, C.A., 2006.** Comparing Differential Evolution Models for Global Optimization. *Genetic and Evolutionary Computation Conference (GECCO'2006)*, Volume 1, Seattle, Washington, USA, ACM Press, pages 485–492.
- Michalewicz, Z., 1994.** *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, Almany, 356 s.
- Murata, T., Ishibuchi H., 1995.** MOGA: Multiobjective Genetic Algorithms. 2nd IEEE International Conference on Evolutionary Computation, 289-294.
- Nabiyev V.V., 2003.** *Yapay Zeka*. Seçkin Yayıncılık, Ankara, 724 s.
- Osyczka, A., Kundu, S., 1995.** A New Method to Solve Generalized Multicriteria Optimization Problems Using The Simple Genetic Algorithm. *Structural Optimization*, 10(2), 94-99.
- Petrowski, A., 1996.** A Clearing Procedure as a Niching Method for Genetic Algorithms. *Proceeding of the third IEEE International Conference on Evolutionary Computation*, 798-803.
- Poloni, C., Giurgevich, A., Onesti, L., Pediroda, V., 2000.** Hybridization of a Multiobjective Genetic Algorithm, a Neural Network and a Classical Optimizer for Complex Design Problems in Fluid Dynamics. *Computer Methods in Applied Mechanics and Engineering* 186(2-4), 403-420.
- Raquel, C. R., Naval, P. C., 2005.** An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization. *Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, New York, USA, 257–264.
- Ray, T., Gokarn, R., Sha, O., 1995.** A Global Optimization Model for Ship Design. *Computers in Industry*, 26: 175–192.
- Ray, T., Liew, K., 2002.** A Swarm Metaphor for Multiobjective Design Optimization. *Engineering Optimization*, 34(2): 141–153.

- Rudolph, G., 1996.** Convergence of Evolutionary Algorithms in General Search Spaces. Proceedings of the Third IEEE Conference on Evolutionary Computation, 50-54.
- Rudolph, G., 2001.** Evolutionary Search Under Partially Ordered Fitness Sets. Proceedings of the International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent Systems, 818-822.
- Ruiz-Torres, A. J., Enscoe, E. E., Barton, R. R., 1997.** Simulated Annealing Heuristics for the Average Flow-Time and the Number of Tardy Jobs Bi-Criteria Identical Parallel Machine Problem. Computers and Industrial Engineering, 33(1-2), 257-260.
- Sadjadi, F., 2004.** Comparison of Fitness Scaling Functions in Genetic Algorithms with Applications to Optical Processing. Optical Information Systems II, Proc. SPIE, 5557, (Editörler: Javidi, B., Psaltis, D.), 356-364.
- Schaffer, J.D., 1984.** Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms. Doktora Tezi, Vanderbilt University, Nashville, ABD, 166 s.
- Schaffer, J.D., 1995.** Multi-objective Optimization with Vector Evaluated Genetic Algorithms. Proceedings of 1st International Conference on Genetic Algorithms, 93-100.
- Schott, J.R., 1995.** Fault Tolerant Design Using Single and Multi-Criteria Genetic Algorithms. Yüksek Lisans Tezi, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Boston, ABD, 203 s.
- Serafini, P., 1994.** Simulated Annealing for Multiple Objective Optimization Problems. Proceedings of the Tenth International Conference on Multiple Criteria Decision Making: Expand and Enrich the Domains of Thinking and Application (Editörler: G. Tzeng, H. Wang, U. Wen, ve P. Yu), Volume 1, Berlin, Springer-Verlag, 283-292.
- Srinivas, N., Deb, K., 1994.** Multi-objective Function Optimization Using Non-dominated Sorting Genetic Algorithms. Evolutionary Computation Journal 2(3), 221-248.
- Srinivasan, D., Seow, T. H., 2003.** Particle Swarm Inspired Evolutionary Algorithm (PS-EA) for Multiobjective Optimization Problem. In Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003), volume 4, Canberra, Australia, 2292-2297.

- Srinivasan, D., Seow, T. H., 2005.** Particle Swarm Inspired Evolutionary Algorithm (PS-EA) for Multi-Criteria Optimization Problems. *Evolutionary Multiobjective Optimization: Theoretical Advances And Applications* (Editörler: A. Abraham, L. Jain, ve R. Goldberg), Springer-Verlag, London, 147–165.
- Storn, R., Price, K., 1995.** Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley, California, ABD.
- Storn, R., Price, K., 1997.** Differential Evolution - A Fast and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11: 341–359.
- Storn, R., Price, K., Lampinen, J.A., 2005.** *Differential Evolution. A Practical Approach to Global Optimization*. Springer, Berlin, 538 s.
- Stützle, T., Hoos, H.H., 2000.** Max Min Ant System. *Journal of Future Generation Computer Systems*, 8 (16), 889–914.
- Suman, B., 2002.** Multiobjective Simulated Annealing—A Metaheuristic Technique for Multiobjective Optimization of a Constrained Problem. *Foundations of Computing and Decision Sciences*, 27(3): 171–191.
- Suman, B., 2003.** Simulated Annealing-Based Multiobjective Algorithms and Their Application for System Reliability. *Engineering Optimization*, 35(4): 391–416.
- Suppaitnarm, A., Seffen, K., Parks, G., Clarkson, P., 2000.** A Simulated Annealing Algorithm for Multiobjective Optimization. *Engineering Optimization*, 33(1): 59–85.
- Tamer, S., Karakuzu, C., 2006.** Parçacık Sürüsü Optimizasyon Algoritması ve Benzetim Örnekleri. ELECO 2006 Elektrik-Elektronik-Bilgisayar Sempozyumu, Elektronik Bildirileri Kitabı, Bursa, 302-306.
- Tan, C.M., 2008.** *Simulated Annealing*. In-Tech, 420 s.
- Tan, K. C., Lee, T. H., Khor, E. F., 2001.** Evolutionary Algorithms for Multiobjective Optimization: Performance Assessments and Comparisons. In *Congress on Evolutionary Computation (CEC2001)*, IEEE Press, 979–986, Piscataway, ABD.
- Tsai C.F., Tsai C. W., Tseng, C.C., 2004.** A New Hybrid Heuristic Approach for Solving Large Traveling Salesman Problem. *Information Sciences*, 166 (1-4), 67-81.

- Toscano Pulido, G., Coello Coello, C. A., 2004.** Using Clustering Techniques to Improve the Performance of a Particle Swarm Optimizer. Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I, Seattle, Washington, USA, 225–237.
- Ulungu, E., Teghem, J., Fortemps, P., 1995.** Heuristics for Multi-Objective Combinatorial Optimization by Simulated Annealing. Multiple Criteria Decision Making: Theory and Applications. Proceedings of the 6th National Conference on Multiple Criteria Decision Making (Editörler: J. Gu, G. Chen, Q. Wei ve S. Wang), Windsor, UK, Sci-Tech, 228–238.
- Ulungu, E., Teghem, J., Ost, C., 1998.** Efficiency of Interactive Multi-Objective Simulated Annealing through a Case Study. Journal of the Operational Research Society, 49: 1044–1050.
- Usta, H., 2007.** Genetik Algoritmalarında Mutasyon, Çaprazlama ve Bellek Mekanizmalarının Etkileri ve İyileştirilmesi. Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi, 124 s.
- Usta, H., Ergül, E.U., Eminoğlu, İ., 2008.** Çaprazlama İşlevinin Genetik Algoritmalarında Çeşitliliğe Olan Etkisinin İncelenmesi. Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU 2008), Isparta, 129-133.
- Vazquez, K.R., 1999.** Multiobjective Evolutionary Algorithms in Non-Linear System Identification. PhD Thesis, The University of Sheffield, 185 s.
- Veldhuizen, D.V., 1999.** Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. Doktora Tezi, Air Force Institute of Technology Dayton, 270 s.
- Veldhuizen D.V., Lamont, G.B., 2000.** On Measuring Multiobjective Evolutionary Algorithm Performance. In Congress on Evolutionary Computation (CEC2000), (Editörler: A. Zalzala, R. Eberhart), IEEE Press, Vol. 1, 204–211, Piscataway, ABD.
- Wang, X., Hao, M., Cheng, Y., Lei, R., 2009.** PDE-PEDA: A New Multiobjective Optimization Algorithm. Journal of Universal Computer Science, Vol. 15, No. 4, 722-741.
- Wolpert, D.H., Macready, W.G., 1997.** No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, 67-82.

- Yen, G.G., Lu, H., 2003.** Dynamic Multiobjective Evolutionary Algorithm: Adaptive Cell-Based Rank and Density Estimation. *IEEE Evolutionary Computation*, Volume 7, No.3, 253-274.
- Yiğit, N., Karaboğa, N., Gürer, B., 2007.** Adaptif Süzgeçlerde Farksal Gelişim Algoritması Kullanılarak Gürültü Giderme. III. İletişim Teknolojileri Ulusal Sempozyumu, Adana, 157-161.
- Zhang, L.B., Zhou, C.G., Xu, X.X., Sun, C.T., Liu, M., 2006.** Multi-objective Evolutionary Algorithm Based on Max-Min Distance Density. *International Conference on Computational Intelligence and Security*, Vol. 1, 312-315.
- Zitzler, E., Thiele, L., 1998.** Multiobjective Optimization Using Evolutionary Algorithms-A Comparative Case Study. In *Parallel Problem Solving from Nature V (PPSN-V)*, 292-301.
- Zitzler, E., 1999.** Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Doktora Tezi, Swiss Federal Institute of Technology, Zurich, Switzerland, 110 s.
- Zitzler, E., Thiele, L., 1999.** Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Trans. on Evolutionary Computation*, Vol.3, No. 4, 257-271.
- Zitzler, E., Deb, K., Thiele, L., 2000.** Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, Vol. 8, No. 2, 173-195.
- Zitzler, E., Laumanns, M., Thiele, L., 2001.** SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103, Swiss Federal Institute of Technology, Zurich, Switzerland, 1-21.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Fonseca, V.G., 2003.** Performance Assessments of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, Vol.7, No.2, 117-132.
- <http://www.lania.mx/~ccoello/EMOO/>** (Erişim zamanı: 10.10.2006)
- <http://www.geatbx.com>** (Erişim zamanı: 04.01.2007)
- <http://auto.howstuffworks.com/turbo2.htm>** (Erişim zamanı: 21.06.2009)
- http://videlectures.net/solomon_krink_depso/** (Erişim zamanı: 05.08.2009)

7. ÖZGEÇMİŞ

ÖZGEÇMİŞ

Adı Soyadı : Engin Ufuk ERGÜL

Doğum Yeri : Şereflikoçhisar / Ankara

Medeni Hali : Evli

Bildiği Yabancı Diller : İngilizce

Eğitim Durumu

Lise : Şereflikoçhisar Lisesi – 1994

Lisans : Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü – 1999

Yüksek Lisans : Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Ana Bilim Dalı – 2003

Çalıştığı Kurumlar ve Yıl :

- Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi
Elektrik-Elektronik Mühendisliği Bölümü (2000-2002)
- Amasya Üniversitesi Meslek Yüksekokulu (2002- ...)

İletişim Bilgileri : euergul@gmail.com

- Yayınları :**
- 1- **Ergul, E.U.**, Usta, H., Eminoglu, I., 2009. Effective Usage of Gamma Correction Operator in Genetic Algorithms, International Symposium on INnovations in Intelligent SysTems and Applications – INISTA 2009, Trabzon, Turkey, June 29-July 1, 447-451.
 - 2- **Ergul, E.U.**, Usta, H., Eminoglu, I., 2009. Effective Elitism Mechanism for Multi-objective Evolutionary Algorithms, International Symposium on INnovations in Intelligent SysTems and Applications – INISTA 2009, Trabzon, Turkey, June 29-July 1, 502-506.
 - 3- Usta, H., **Ergül, E.U.**, Eminoğlu, İ., 2008. Çaprazlama İşlevinin Genetik Algoritmelerde Çeşitliliğe Olan Etkisinin İncelenmesi. Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU 2008), Isparta, 18-20 Haziran, 129-133.
 - 4- Gezeğin, C., **Ergül, E.U.**, Bergil, E., 2006. Endüstriyel Otomasyon Teknolojileri Eğitiminde Yeni Bir Eğitim Ortamı Oluşturulması ve Eğitim Yönteminin Geliştirilmesi, UMES- Ulusal Teknik Eğitim, Mühendislik ve Eğitim Bilimleri Genç Araştırmacılar Sempozyumu, Kocaeli, 20-22 Haziran, 297-300.
 - 5- Özgönel, O., **Ergül, E.U.**, 2005. Güç Elektronik Devrelerinin İHM Tabanlı Benzetimi. Elektrik-Elektronik-Bilgisayar Mühendisliği 11. Ulusal Kongresi, İstanbul, 22-25 Eylül, 394-397.

8. EK:

Tüm ÇAGA yöntemlerinin puanlama mekanizmalarının anlatımında kullanılan örnek topluluktaki bireylerin amaç işlevlerinde aldıkları değerler aşağıdaki çizelgede verilmektedir.

	A	B	C	D	E	F	G	H
f₁	1	2	5	7	6	4	3	8
f₂	6	4	2	1	3	5	7	8