

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

116573

GENETİK ALGORİTMALARDA IRAKSAMA VE YEREL
ÇÖZÜMDE KALMA PROBLEMLERİNİN GİDERİLMESİ

Ali KARCI

T.C. YÜKSEKÖĞRETİM KURULU
DOKÜMANTASYON MERKEZİ

DOKTORA TEZİ

ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

116573

Bu tez, 17.06.2002 tarihinde aşağıda belirtilen jüri tarafından oybirliği / ~~oyçokluğu~~ ile başarılı / ~~başarısız~~ olarak değerlendirilmiştir.

Danışman : Doç.Dr. Ahmet ARSLAN

Üye : Doç.Dr. Z.Hakan AKPOLAT

Üye : Doç.Dr. Saadetdin HERDEM

Üye : Doç.Dr. Mustafa POYRAZ

Üye : Doç.Dr. Erhan AKIN

.....
.....
.....
.....
.....

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun/...../..... tarih ve sayılı kararıyla onaylanmıştır.

TEŐEKKÖR

Yüksek lisans ve doktora çalışmalarında bana yol gösteren ve bu eğitimlerimi tamamlamamda yapmış olduđu katkılarından dolayı Doç.Dr. Ahmet ARSLAN' a ; tavsiye ve önerilerinden dolayı Arş.Gör. Ahmet ÇINAR' a, Öğr. Gör. İbrahim TÜRKOĞLU' na teşekkür ederim.

Doktora çalışmaları sırasında bana karşı göstermiş oldukları sabır ve desteklerinden dolayı çok kıymetli aileme sonsuz şükranlarımı sunarım.



İÇİNDEKİLER

TEŞEKKÜR

İÇİNDEKİLER.....	I
ŞEKİLLER LİSTESİ.....	III
SİMGELER LİSTESİ.....	VII
ÖZET.....	VIII
ABSTRACT.....	IX
1. GİRİŞ.....	1
1.1.Tezin Amacı.....	3
1.2.Tezin Yapısı.....	4
2. GENETİK ALGORİTMALAR.....	6
2.1. Genetik Algoritmanın Tarihçesi.....	6
2.2. Genetik Algoritmanın Tanımı.....	6
2.3. Kodlama yöntemleri.....	8
2.4. Uygunluk Değerlendirme Teknikleri.....	10
2.5. Genetik Operatörler.....	13
2.5.1. Seçme.....	13
2.5.2. Çaprazlama.....	19
2.5.3. Mutasyon.....	21
2.6. Genetik Algoritma Çeşitleri.....	22
2.6.1. Mikro-Genetik Algoritmalar.....	22
2.6.2. Basit Genetik Algoritmalar.....	23
2.6.3. Duragan Durum Genetik Algoritmalar (Genitor).....	23
2.6.4. CHC.....	23
2.6.5. Hibrid Genetik Algoritmalar.....	23
2.6.6. Paralel Genetik Algoritmalar.....	24
3. DÜZENLİ POPULASYON YÖNTEMİ VE GELİŞİGÜZEL POPULASYON YÖNTEMİNE OLAN ÜSTÜNLÜKLERİ.....	25
3.1. Düzenli Populasyon.....	26
3.2. Yeni Mutasyon ve Çaprazlama Tanımları.....	29
3.3. Düzenli Populasyonun Gelişigüzel Populasyona Göre Üstünlükleri.....	31
3.4. Adalı Düzenli Populasyon.....	38
3.5. Önemli Nokta.....	39
4. ÇAPRAZLAMA VE MUTASYONUN DÜZENLİ POPULASYONA ETKİLERİ.....	40

4.1. n-Noktalı Çaprazlamanın Etkisi.....	41
4.2. Düzenli Çaprazlamanın Etkisi.....	47
4.3. Çaprazlama ile Şema İnşası.....	48
4.4. Mutasyon Operatörünün Etkisi.....	51
5. UYGULAMA SONUÇLARI.....	52
5.1. Griewank Fonksiyonun Sonuçları.....	52
5.2. Michalewicz Fonksiyonun Sonuçları.....	56
5.3. Rosenbrock Fonksiyonun Sonuçları.....	59
5.4. Rastrigin Fonksiyonun Sonuçları.....	62
5.5. Peaks Fonksiyonun Sonuçları.....	64
5.6. Yeni Operatörlerin Sonuçları.....	66
5.7. Adalı Düzenli Populasyonun Uygulama Sonuçları.....	68
6. SONUÇ VE DEĞERLENDİRME.....	71
KAYNAKLAR.....	75
ÖZGEÇMİŞ.....	78

ŞEKİLLER LİSTESİ

Şekil 2.1. Genetik algoritmanın şematik gösterimi.....	9
Şekil 2.2. K_i , Kromozom _i , $1 \leq i \leq 10$, göstermek üzere basit bir seçim işlemi.....	14
Şekil 2.3. Rulet tekerleği seçimi.....	16
Şekil 2.4. Stokastik evrensel örnekleme yöntemi ile seçim yapmak.....	17
Şekil 2.5. Tek-noktalı çaprazlama işlemi.....	20
Şekil 2.6. İki-noktalı çaprazlama işlemi.....	21
Şekil 2.7. Düzenli çaprazlama işlemi.....	21
Şekil 3.1. $r=1$ için gelişigüzel üretilmiş olan kromozomdan ikinci kromozomun elde edilmesi.....	26
Şekil 3.2. $r=2$ için gelişigüzel bir kromozomdan üretilen kromozomlar.....	26
Şekil 3.3. $r=3$ için gelişigüzel bir kromozomdan 7 tane kromozomun üretilmesi.....	27
Şekil 3.4. Yoğun çaprazlama.....	30
Şekil 3.5. Transpozisyon mutasyon.....	30
Şekil 3.6. Griewank fonksiyonu için popülasyonların değişimleri.....	35
Şekil 3.7. Michalewicz fonksiyonu için popülasyonların değişimleri.....	36
Şekil 3.8. Rosenbrock fonksiyonu için popülasyonların değişimleri.....	36
Şekil 3.9. Popülasyonların değişik fonksiyonlar için korelasyon grafikleri.....	37
Şekil 3.10. Adalı düzenli popülasyonda kromozomların yerleşimi.....	38
Şekil 3.11. Değişken ve ayırım noktalarının çakışması durumu.....	39
Şekil 4.1. H_3 şemasında tanımlayıcı genler : g_1, g_2, g_3 (L kromozom uzunluğu ve L_1 tanımlama uzunluğu).....	41

Şekil 4.2. (a) Bitişik tanımlayıcı genler arasına çift sayıda kesme noktasının düşmesi; (b) Bitişik tanımlayıcı genler arasına tek sayıda kesme noktasının düşmesi.....	43
Şekil 5.1. Programın ana mönüsü.....	53
Şekil 5.2. İki değişkenli Griewank fonksiyonu.....	53
Şekil 5.3. Düzenli populasyonun ($r=1$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	54
Şekil 5.4. Düzenli populasyonun ($r=2$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	54
Şekil 5.5. Düzenli populasyonun ($r=3$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	55
Şekil 5.6. Düzenli populasyonun ($r=4$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	55
Şekil 5.7. Düzenli populasyonun ($r=5$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	56
Şekil 5.8. Gelişigüzel populasyonun Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	56
Şekil 5.9. İki değişkenli Michalewicz fonksiyonu.....	57
Şekil 5.10. Düzenli populasyonun ($r=3$) Michalewicz fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	58
Şekil 5.11. Düzenli populasyonun ($r=4$) Michalewicz fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	58
Şekil 5.12. Düzenli populasyonun ($r=5$) Michalewicz fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	58
Şekil 5.13. Gelişigüzel populasyonun Michalewicz fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	59
Şekil 5.14. İki değişkenli Rosenbrock fonksiyonu.....	60

Şekil 5.15. Düzenli populasyonun ($r=3$) Rosenbrock fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=250.....	60
Şekil 5.16. Düzenli populasyonun ($r=4$) Rosenbrock fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=250.....	61
Şekil 5.17. Düzenli populasyonun ($r=5$) Rosenbrock fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=250.....	61
Şekil 5.18. Gelişigüzel populasyonun Rosenbrock fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=250.....	61
Şekil 5.19. İki değişkenli Rastrigin fonksiyonu.....	63
Şekil 5.20. Düzenli populasyonun ($r=4$) Rastrigin fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	63
Şekil 5.21. Düzenli populasyonun ($r=5$) Rastrigin fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	63
Şekil 5.22. Gelişigüzel populasyonun Rastrigin fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	64
Şekil 5.23. İki değişkenli Peaks fonksiyonu.....	65
Şekil 5.24. Düzenli populasyonun ($r=4$) Peaks fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	65
Şekil 5.25. Düzenli populasyonun ($r=5$) Peaks fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	65
Şekil 5.26. Gelişigüzel populasyonun Peaks fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	66
Şekil 5.27. Yoğun çaprazlamanın Peaks fonksiyonu sonuçları.....	67

Şekil 5.28. Tek-Noktalı çaprazlamanın Peaks fonksiyonu sonuçları.....	67
Şekil 5.29. Random mutasyon operatörünün popülasyonun değişimine olan etkisi.....	68
Şekil 5.30. Transpozisyon mutasyon operatörünün popülasyonun değişimine olan etkisi.....	68
Şekil 5.31. Adalı düzenli popülasyonun ($r=2$) Griewank fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	69
Şekil 5.32. Adalı düzenli popülasyonun ($r=3$) Griewank fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	69
Şekil 5.33. Adalı düzenli popülasyonun ($r=4$) Griewank fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	69
Şekil 5.34. Adalı düzenli popülasyonun ($r=5$) Griewank fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.....	70

SİMGELER LİSTESİ

N	:	Populasyon boyutu.
n	:	Kromozom uzunluğu.
r	:	Düzenli populasyon parametresi.
$P(i)$	:	i . bireyin oluşma olasılığı
μ_t	:	Populasyonun t anındaki beklenen değeri
σ_t	:	Populasyonun t anındaki standart sapma değeri
H_k	:	Mertebesi k olan şema.
$m_t(H)$	:	t anında H şeması içindeki birey sayısı
$f_t(H)$	:	t anında H şemasının ortalama uygunluk değeri
$\bar{f}_t$	:	t anında populasyonun ortalama uygunluk değeri
$m_{t+1}(H)$	:	$t+1$ anında H şeması içinde var olabilecek birey sayısı
$P(u_i)$	:	u_i olayının oluşma olasılığı.
$P_y(H_k)$	:	H_k şemasının yaşama olasılığı.
$P_b(H_k)$	:	H_k şemasının bozulma olasılığı.
$P_{\text{çift}}(H_k)$	:	H_k şemasında çaprazlama için tanımlayıcı iki gen arasına çift sayıda çaprazlama noktasının gelme olasılığı.
$P_{\text{tek}}(H_k)$	:	H_k şemasında çaprazlama için tanımlayıcı iki gen arasına tek sayıda çaprazlama noktasının gelme olasılığı.
$P_{\text{eş}}$	:	Çaprazlama işleminde yer değiştiren tanımlayıcı genlerin aynı olma olasılığı.
$P_{\text{in}}(H_k s_j)$	:	s_j olayının oluşmasından sonra H_k şemasının inşa edilme olasılığı.
μ	:	Bir genin mutasyona uğrama olasılığı.

ÖZET
DOKTORA TEZİ
GENETİK ALGORİTMALARDA İRAKSAMA VE YEREL
ÇÖZÜMDE KALMA PROBLEMLERİNİN GİDERİLMESİ

Ali KARCI

Fırat Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Mühendisliği Anabilim Dalı
2002, Sayfa : 78

Bu çalışmada başlangıç popülasyonunu üretmek için bir yöntem önerilmiştir. Bu yöntemin önerilme amacı ise genetik algoritmaların global çözümden ıraksayarak yerel çözümde kalma problemini gidermektir. Bu yöntemde başlangıç popülasyonu üretilme yöntemini tamamen gelişigüzel olmaktansa, kısmen gelişigüzel hale getirip bu iki problemin giderilmesi hedeflenmiştir. Bu şekilde üretilen popülasyona Düzenli Popülasyon adı verildi.

Yapılan uygulamalarda düzenli popülasyonunun prematüre çözümler verme ihtimali vardır. Prematüre çözümler problemini gidermek için düzenli popülasyon için mutasyon, çaprazlama ve seçme operatörleri tekrar tanımlanmıştır.

Düzenli popülasyonun yapısından faydalanılarak adalı düzenli popülasyon tanımı yapıldı ve aynı fonksiyonlara uygulandı. İterasyon bakımından gelişigüzel ve düzenli popülasyonun kötü olduğu durumda adalı düzenli popülasyon çok daha kısa sürede çözüme ulaştı.

Düzenli Popülasyonun özellikleri: 1) Çeşitlilik vardır, 2) Nesiller arasında lineer bağımlılık yoktur, 3) Yerel çözümde kalmaz, 4) Global çözümden uzaklaşmaz, 5) Çeşitliliği arttırmaktadır, 6) Adalı popülasyona dönüştürüldüğünde daha iyi sonuçlar vermektedir, 7) Başlangıç popülasyonunda bütün kromozomların aynı olma olasılığı sıfırdır, 8) Adalar arasında kromozomların başlangıçta aynı olma olasılığı sıfırdır, 9) başlangıç popülasyonu çözüm uzayının medyanına göre simetrik bir popülasyondur, 10) çözüm uzayın medyanına göre simetrik bir popülasyondur, 11) düzgün bir popülasyondur.

Anahtar Kelimeler: Evrimsel Hesaplama, Genetik Algoritmalar, Gelişigüzel Başlangıç Popülasyonu, Şema Teorisi.

ABSTRACT

PhD Thesis

REMOVING DIVERSING AND TRAPPING IN LOCAL SOLUTION PROBLEMS IN GENETIC ALGORITHMS

Ali KARCI

Firat University

Graduate School of Natural and Applied Sciences

Department of Electric-Electronics Engineering

2002, Page : 78

In this thesis, a method was proposed to generate a partially random initial population and generated population was called uniform population. The expectation of initial population is equal to median of solution space and deformity coefficient of population is zero and population is symmetric with respect to median of solution space. So, it can be called as uniform population. The aim of this thesis is to overcome the diversing from global solution and trapping in local solution problems.

After application of this population to some multi-model function to find maximum point of them, uniform population have potential to be trapped in premature solution. Hence mutation and crossover operators were redefined for this population. In general, results of uniform population were better than or equal to results of random population.

Island model of genetic algorithms was constructed by making use of structure of uniform population and it was called island uniform population. The process of island uniform population made number of iteration decay rapidly.

The properties of uniform population: 1) have diversity, 2) there is no linear dependency between two consecutive generations, 3) there is no potential to be trapped in local solution, 4) there is no potential to go far away global solution, 5) increasing the diversity of population, 6) island uniform population gives better results, 7) the probability of all initial chromosomes to be same is zero, 8) the probability of all pairs of chromosomes that each of them is from different island, to be same is zero, 9) population is symmetric with respect to median of solution space, 10) uniform population, 11) symmetric population with respect to median of solution space.

Keywords: Evolutionary Computation, Genetic Algorithms, Random Initial Population, Schema Theory.

1. GİRİŞ

Tasarım yapma yeteneği, canlı varlıklar içerisinde sadece insana mahsus bir özelliktir (Schank, 1987). Bir çok yaratık bir şeyler üretme yeteneğine sahip olmasına rağmen, bir kuşun yuva yapması, bir örümceğin ağ örmesi, bir arının bal yapması, karıncaların işbirliği içerisinde çalışması gibi, fakat bunlar bu icraatlarını gerçekleştirirken yaptıkları işi nasıl yapacaklarını hiç düşünmezler. Bir makina gibi ezberlerinde bu eylemleri gerçekleştirirler. Bunların aksine, insan düşünme yeteneğine sahip olduğundan zaman içerisinde yapmış olduğu işlerde sürekli bir yenilik vardır ve bu yenilik içinde düşünme yeteneğini kullanmaktadır.

Günümüze kadar yapılan yenilikler içerisinde jet motorun icadı, buhar türbinlerinin icadı, patlayıcı silahların icadı, nükleer enerjinin kontrol edilebilir olduğunun ortaya çıkması, elektriğin icadı, v.b. belli başlı olanları sayılabilir. Özellikle elektrik icat edildikten sonra elektrik ile çalışan cihazların icat edilmesi yeniliklerin ortaya çıkma hızını oldukça arttırmıştır.

Elektrik ile çalışan ve insanın üretmiş olduğu yeniliklere ivme kazandıran en önemli olay bilgisayarın icat edilmesidir. Bu cihazın icadından sonra insanın yapmış işlerin çoğu bir-bir bilgisayar ortamına taşınmaya başlanmıştır ve bu sayede insanın uzun zamanda yapmış olduğu işler daha kısa zamanda yapılmaya başlanmıştır.

Zaman içerisinde yapılan işlerin bilgisayar ortamına aktarılması yeterli görülmemeye başlandı ve bunun sonucunda insanın düşünme kabiliyetinin bilgisayar ortamına aktarılabilir olup olmadığı tartışılmaya başlandı. Bu tartışmanın sonucunda “Yapay Zeka Kavramı” ortaya çıktı. Günümüzde yapay zeka, bir çok alt bölümden oluşmaktadır ve bu alt bölümler

- Makina öğrenmesi
- Veri madenciliği
- Yapay sinir ağları
- Bulanık mantık
- Evrimsel hesaplama

şeklinde sıralanabilir.

Başka bir bakış açısına göre yapay zekanın iki tane geniş alt alanı vardır (Schank, 1987). Bu alanlardan biri insanın zekiliğini ele almak ve bundan esinlenerek hesapsal modeller üretmek ve bu modeller yardımı ile problem çözmektir (Newell, 1990; McClelland v.d., 1986; Minsky, 1985; Anderson, 1983; Anderson, 1976; Newell v.d., 1963). Böyle bilinçsel modelleme işi, insanın psikolojik performansı ile ilgilendir, fakat buradaki temel amaç programı zekileştirmek değildir.

Yapay zekanın diğerk bir geniş kolu da insanların becerikli olduđu işlerde/görevlerde bilgisayarın da iyi olmasını sağlayacak tekniklerin geliştirilmesidir (Erman v.d., 1980; Lindsay v.d., 1980; Berliner, 1977; Shortliffe, 1976; Samuel, 1959;). Bilinçsel modele benzer olarak yapay zekanın bu metodolojisi de insanın zekiliğine dayanmaktadır. Bununla birlikte bilgisayarın zekice çalışması için insanın zekiliğinin özetlenerek alınmasına gerek yoktur.

Yapay zekanın her iki alanının ortak özelliğı arama işlemidir. Yapay zekanın içermiş olduđu öğrenme, keşfetme, örüntü tanıma, sınıflandırma, teşhis ve tasarım yöntemleri arama tabanlı yöntemlerdir.

Bu yöntemlerde kullanılan arama yöntemleri zayıf ve güçlü arama olarak ikiye ayrılırlar. Güçlü arama yönteminde, programcı veya bilgi mühendisi problemi tanımlayan bütün bilgileri arama mekanizmasına yerleştirir. Bundan dolayı bu yöntem problem bağımlı olur, çünkü çözülecek problemin bilgileri ile arama mekanizması donatılmıştır. Zayıf arama yöntemleri, problem bilgileri bakımından zengin olmadıklarından, problemden (görevden) bağımsız arama yöntemleridir ve bundan dolayı etkili yöntemler değildir. Güçlü arama yöntemleri, birden fazla zayıf arama yöntemini, problemin farklı aşamalarında kullanabilirler.

Evrimsel hesaplama yöntemi, arama tabanlı bir problem çözme yöntemidir. Bu yöntem, evrim teorisinin genel mantığını hesaplama ortamına taşır ve canlıların genetik özellikleriyle birleştirerek bir hesaplama yöntemi sunar. Doğada aynı türe birden fazla birey bulunduđu gibi bu yöntemde başlangıçta problemin gelişigüzel birden fazla çözümünden başlar. Doğadaki güçlü olan kazanır ilkesine dayanarak, gelişigüzel üretilen çözümlerden iyi olanlar bir sonraki nesile aktarılırken, kötü olanlar elenir.

Evrimsel hesaplama yönteminde, başlangıç çözümler gelişigüzel üretildikten sonra elde edilen gelişigüzel çözümler kümesine populasyon adı verilir. Nasıl ki doğada zaman içerisinde bazı bireyler yok olurken bazı bireyler varlığını sürdürmenin yanında bir de nesil üretirken, evrimsel hesaplamada da buna benzer olarak başlangıç çözümler kümesi bazı işlemlere tabi tutulur ve bu işlemler sonucunda iyi olan çözümler bir sonraki nesile aktarılırken, kötü olan çözümler elenir. Elenen çözümler yerine iyi çözümlerin kopyaları bir sonraki nesile aktarılırlar. Her nesil bir nesil olarak adlandırılır.

Bir çözümün iyi olup olmadığı çözülecek problemin tabiatına bağlıdır. Problemin özünü tanımlayan ve çözümlerin iyi mi kötü mü olduğuna karar verebilmesini sağlayan bu kritere uygunluk fonksiyonu değeri veya amaç fonksiyonu değeri denir. Bundan dolayı her problemin kendisine has bir uygunluk fonksiyonu vardır. Her nesilde her çözümün uygunluk değeri tekrar hesaplanır ve kalacak olan bireyler ayıklanır ve elenecek olan bireyler çözümler kümesinden silinirler. Bunların yerine uygunluk değeri nispeten daha iyi olan çözümlerin kopyaları

özmler kümesine eklenir. Evrimsel hesaplama yönteminde elde edilen bu gelişigüzel özmlerin her birine birey veya kromozom adı verilir.

Uygunluk değeriine dayanarak bir sonraki nesilde hangi kromozomların var olacağına ve hangilerin eleneceğine karar veren yöntem Seçme (Seleksiyon) işlemi denir ve bu işlem evrimsel hesaplama kullanılan operatörlerden biridir. Evrimsel hesaplama temel mantığını, canlıların genetik yapısı üzerine dayandırdığından dolayı buna Genetik Algoritma adı verilir.

Genetik algoritmalarda kullanılan seçme operatörü yanında, iki tane daha operatör kullanılır: Bunlar aprazlama ve mutasyon operatörleridir. Bu üç operatöre genetik operatörler adı verilir ve bir sonraki bölümde bu operatörler daha detaylı olarak ele alınacaklardır.

Genetik algoritma ile bir problem özölürken, bu problem için birden fazla gelişigüzel özüm oluşturulur ve bunların her biri popülasyon içindeki birer kromozomu temsil ederler. Bu kromozomlardan en az birinin özöme dönüşmesi veya kabul edilebilir bir hata ile özöme komşu olma durumuna dönüşmesi için genetik operatörler her nesilde kromozomlara uygulanırlar. Bu işleme genetik algoritmanın “süreci” denir.

Genetik algoritmalarda, başlangı popülasyonu gelişigüzel üretildiğinden dolayı bazı durumlarda genetik algoritma süreci özöme yaklaşma yerine özömden ıraksayabiliyor veya yerel özümü bulup oradan kurtulup genel özöme gidemeyebiliyor. Bu problemler özömden ıraksama ve yerel özümde kalma problemleri olarak bilinirler.

1.1. Tezin Amacı

Bu tezde, iki problemi gidermek için başlangı popülasyonu üretilme yöntemini tamamen gelişigüzel olmaksızın, kısmen gelişigüzel hale getirip bu iki problemin giderilmesi hedeflenmiştir. Bu şekilde üretilen popülasyona Düzenli Popülasyon adı verildi. Bu popülasyona düzenli popülasyon adının verilmesi, elde edilen kromozom topluluğunun arpıklık katsayısı, ortalaması olasılıksal olarak incelendiğinde ortalama değeri verilen özüm uzayın medyanına eşit ve arpıklık katsayısı da sıfıra eşit olmasından dolayıdır. Bu kriterler kromozom topluluğunun düzenli bir dağılıma sahip olduğunu gösterir.

Yapılan uygulamalarda düzenli popülasyonun prematüre özömler verme ihtimalinin olduğu görüldü. Prematüre özömler, genel özöme ulaşmadan ve genel özöme giderken yol üzerinde herhangi bir yerel özüm olmadığı halde genel özöme ulaşmadan bütün kromozomların birbirinin aynısı olması durumudur. Prematüre özömler problemini gidermek için düzenli popülasyon üzerinde mutasyon, aprazlama ve seçme operatörleri yeniden tanımlanmıştır.

Düzenli populasyon yönteminde elde edilen kromozom topluluğu, çok güzel bir özelliğe sahiptir. Bu özellikte, bir kromozomdan 2^r (r parça sayısı) tane kromozom elde edilmektedir ve hepsi birbirinden farklıdır. Elde edilen her kromozomu bir alt popülasyona atarak birden fazla popülasyon oluşturulur. Bu popülasyonların içermiş olduğu kromozomların birbirinden farklı olduğu kesindir. Eğer gelişigüzel oluşturulan kromozomların hepsi aynı olursa, alt popülasyonların içerisindeki kromozomların hepsi aynı olur. Bu durumda popülasyon sayısı kadar farklı kromozom vardır. Bu ise, düzenli popülasyonun avantajlarından biridir.

Alt popülasyonlar kullanılarak hesaplama yapılması durumunda, genetik operatörler alt popülasyonlar içerisinde uygulanır ve popülasyonlar arasında uygulanmazlar. Bu şekilde birden fazla çözüm elde edilir ve bir popülasyon prematüre çözümde kalmışsa, bir diğerinin çözüme ulaşma şansı artırılmaya çalışılmıştır.

Prematüre çözüm popülasyonun çeşitliliğini çözüme kavuşmadan kaybetmesidir. Bu durumda seçim işlemi değiştirilir ve seçim işlemi sırasında bir kez seçilen kromozom ile aynı olan kromozomlar seçilmez ve bunun yerine gelişigüzel kromozom üretilir ve bu şekilde çeşitlilik korunmuş olur. Bu yapılan seçme işlemine, yarıçaplı seçim denilir ve bu tezde kullanılan bir seçme türüdür.

1.2. Tezin Yapısı

Bu tez toplam altı bölümden oluşmaktadır. Birinci bölüm giriş bölümü olup böyle bir çalışmaya neden ihtiyaç duyulduğunu tasvir eden bir bölümdür. Diğer bölümler aşağıdaki gibidir.

Bölüm 2: Bu bölümde genetik algoritmanın tanımı ve tarihçesi verildikten sonra genetik algoritmalarda kullanılan operatörlerin tanımları verilmiştir. Çok kullanılan genetik operatörlerin bir kısmının açıklaması yapılmıştır ve genetik algoritmaların çeşitleri üzerinde durulmuştur.

Bölüm 3: Bu bölümde düzenli popülasyon yönteminin tasviri yapıldıktan sonra gelişigüzel popülasyonun taşıdığı iyi tarafların düzenli popülasyonda da olduğu gösterilmiştir. Ayrıca bu yöntemin gelişigüzel yöntemle olan üstünlüğü olasılıksal olarak ispatlanmıştır. Aynı zamanda bu bölümde genetik operatörlerden, mutasyon ve çaprazlama için yeni tanımlar yapılmıştır. Düzenli popülasyonun alt popülasyonlara bölünebilme özelliğinden yola çıkılarak adalı düzenli popülasyon detaylı olarak açıklanmıştır.

Bölüm 4: Bu bölümde düzenli popülasyonun teorik analizi yapılmıştır. Teorik analiz yapılırken, mutasyon ve çaprazlama operatörlerin popülasyon üzerine olan etkisi incelenmiştir. Ayrıca bazı durumlarda mutasyon ve çaprazlamanın gelişigüzel popülasyon üzerine olan etkileri düzenli popülasyon için de geçerliken, bazı durumlarda daha farklı etkilere sahip oldukları gösterilmiştir.

Bölüm 5: Bu bölümde beş tane çok modelli fonksiyonun genel tanımları verildikten sonra bu fonksiyonların maksimum değerini hesaplamak için düzenli, adalı düzenli ve gelişigüzel popülasyonlar kullanılmıştır. Deneysel sonuçlardan yola çıkılarak her iki popülasyon arasında kıyaslar yapılmıştır.

Bölüm 6: Bu bölümde ise, tez çalışmasında önerilen yöntemin sonuçları ve gelişigüzel popülasyona olan üstünlükleri tartışılmıştır. Bölüm 3, 4 ve 5’ de bu üstünlükler verilirken daha detaylı olarak açıklandığından bu bölümde sadece üstünlüklerin özeti yapılmıştır.



2. GENETİK ALGORİTMALAR

2.1. Genetik Algoritmanın Tarihçesi

Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı olan John Holland bu konuda ilk çalışmaları yapan kişidir. Makine öğrenmesi konusunda çalışan Holland, Darwin'in evrim kuramından etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşündü (Holland, 1975). Tek bir makinenin öğrenme yeteneğini geliştirmek yerine böyle yapılarda oluşan bir topluluğun çoğalma, çiftleşme, mutasyon, vb. genetik süreçlerden geçerek başarılı (öğrenebilen) yeni bireyler oluşturabildiğini gördü. Araştırmalarını, arama ve optimumu bulma için, doğal seçme ve genetik evrimden yola çıkarak yapmıştır. İşlem boyunca, biyolojik sistemde bireyin bulunduğu çevreye uyum sağlayıp daha uygun hale gelmesi örnek alınarak, optimumu bulma ve makine öğrenme problemlerinde, bilgisayar yazılımı modellenmiştir.

Çalışmalarının sonucunu açıkladığını kitabının 1975'te yayınlanmasından sonra geliştirdiği yöntemin adı Genetik Algoritmalar (GA) olarak yerleşti (Holland, 1975). Ancak 1985 yılında Holland'ın öğrencisi olarak doktora sınavı veren David E. Goldberg adlı inşaat mühendisi 1989 da konusunda bir klasik sayılan kitabını (Goldberg, 1989) yayınlamaya dek genetik algoritmaların pek pratik yararı olmayan bir araştırma konusu olduğu düşünülüyordu. İlk olarak Holland' da makina öğrenme sistemlerine yardımcı olarak kullanılmış daha sonra De Jong Goldberg ve diğerleri tarafından analiz edilmiştir. Goldberg, GA' nın çok sayıda kollara ayrılmış gaz borularında, gaz akışını düzenlemek ve kontrol etmek için uygulamasını tanımlamıştır. Ayrıca kendisinin kullandığı makina öğrenmesi, nesne tanıma, görüntü işleme ve işlemsel arama gibi alanlarda kullanıldığını vurgulamıştır.

Goldberg'in gaz boru hatlarının denetimi üzerine yaptığı doktora tezi ona sadece 1985 National Science Foundation Genç Araştırmacı ödülünü kazandırmakla kalmadı, genetik algoritmaların pratik kullanımının da olabilirliğini kanıtladı. Ayrıca kitabında genetik algoritmalara dayalı tam 83 uygulamaya yer vererek GA'nın dünyanın her yerinde çeşitli konularda kullanılmakta olduğunu gösterdi.

2.2. Genetik Algoritmanın Tanımı

Genetik algoritma, doğadaki evrim mekanizmasını örnek alan bir arama metodudur ve bir veri grubundan özel bir veriyi bulmak için kullanılır. Genetik Algoritmalar, Evrimsel Genetik ve Darwin'in doğal seçmeye (seleksiyona) benzerlik kurularak geliştirilmiş iteratif, ihtimali bir arama metodudur.

Genetik algoritmalar, gelişigüzel yürüyüş veya gelişigüzel üretilmiş çözümler arasında arama yapma mantığına dayalı bir evrimsel hesaplama yöntemidir. Bu yöntemde, problemin çözümü yapılırken bu problem ile ilgili gerekli bilgilerin hepsi kodlamada kullanılmaz. Bundan dolayı, bu yöneme zayıf arama yöntemi veya kör arama yöntemi adı da verilir.

Genetik algoritma kullanılarak problem çözümü yapmak için gelişigüzel birden fazla çözüm üretilir ve bu çözümlerin oluşturmuş olduğu kümeye populasyon denir ve populasyon dinamik bir özelliğe sahiptir ve zaman içerisinde değişiklik gösterir. Her çözüme birey veya kromozom adı verilir. Her kromozom genlerden oluşur ve bir gen, bir kromozom oluşturulurken kodlanabilen en küçük bilgiye verilen isimdir.

Genetik algoritmalar doğada geçerli olan en iyinin yaşaması kuralına dayanarak sürekli iyileşen çözümler üretir. Bunun için “iyi”nin ne olduğunu belirleyen bir *uygunluk* (fitness) fonksiyonudur. Diğer bir deyişle problemin özünü tasvir eden kriter, uygunluk fonksiyonu veya amaç fonksiyonudur. Uygunluk değeri bütün kromozomlar için hesaplanır ve populasyon içerisinde birden fazla kromozom olduğundan genetik algoritmaların bir diğer önemli özelliği olarak bir grup çözümle uğraşmasıdır. Bu sayede çok sayıda çözümün içinden iyileri seçilip kötüler elenebilir.

Genetik algoritmaları diğer algoritmalarından ayıran en önemli özelliklerden biri de seçmedir. Genetik algoritmalarda çözümün uygunluğu onun seçilme şansını artırır ancak bunu garanti etmez. Seçim de ilk grubun oluşturulması gibi rasgeledir ancak bu rasgele seçimde seçilme olasılıklarını çözümlerin uygunluğu belirler.

Genetik Algoritmaları (GA) diğer yöntemlerden ayıran önemli noktalar vardır ve bu noktalar şu şekilde özetlenebilir.

- GA , sadece bir tane çözüm üzerinden işleme başlamaz, gelişigüzel üretilmiş olan birden fazla çözüm üzerinde arama yapar.
- GA , arama uzayında çözümlerin kaliteli olup olmadığını uygunluk fonksiyonu ile kontrol eder. Bundan dolayı türev, integral veya kompleks matematiksel işlemlere ihtiyaç duymaz.
- Bireyleri seçme ve birleştirme aşamalarında belirli kurallar değil olasılık kuralları kullanır.
- Diğer yöntemlerde olduğu gibi doğrudan parametreler üzerinde çalışmaz. GA, en iyileme edilecek parametreleri kodlar ve bu kodlar üzerinde işlem yapar.
- GA ne yaptığı konusunda bilgi içermez, nasıl yaptığını bilir. Bu nedenle kör bir arama metodudur. Problemin tabiatı hakkında fazla bilgiye sahip değildir.

- Olasılık kurallarına göre çalıştığından dolayı programın performansı hakkında önceden bir fikir yürütmek doğru olmaz, ancak olasılıksal bir sonuç verilebilir.
- GA, bir çözüm arama yöntemidir.

Şekil 2.1' de genetik algoritmanın şematik gösterimi görülmektedir. İlk olarak gelişigüzel olarak popülasyon oluşturulur. Popülasyon oluşturulduktan sonra değerlendirme işlemi bütün kromozomlara uygulanır. Diğer bir deyişle bütün kromozomların uygunluk değerleri hesaplanır. Uygunluklar hesaplandıktan sonra uygunluğu iyi olan bireylerin bir sonraki nesilde varlıklarını devam ettirmeleri için seçme işlemi gerçekleştirilir. Bundan sonra bazı ebeveynlere çaprazlama ve bazı kromozomlara mutasyon işlemi uygulanır. Çaprazlama ve mutasyon işlemlerinin hangi kromozomlara uygulanması gerektiği uygunluk değerinden yola çıkılarak bulunabilir. Genelde uygunluk değeri kötü olan bireylere mutasyon işlemi uygulanırken, uygunluk değerleri iyi olan kromozom çiftlerine çaprazlama işlemi uygulanarak gelecek için umut verici olan evlatlar (yeni bireyler) elde edilir.

Genetik algoritmanın yapısal modeli aşağıdaki gibi olur.

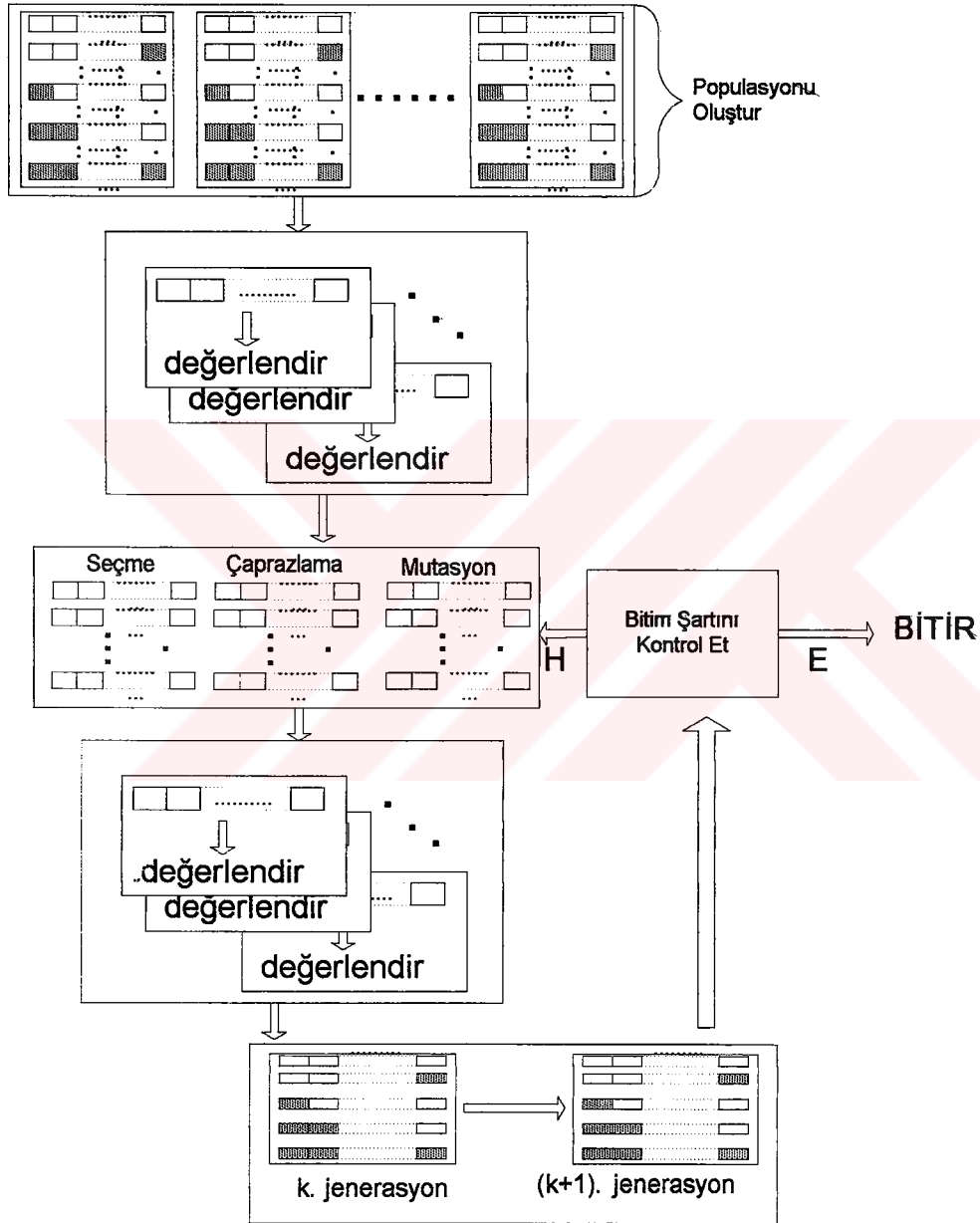
GA(P)

1. $t \leftarrow 0$
2. $P(t) \leftarrow \text{Gelişigüzel_Popülasyon_Oluştur.}$
3. $\text{Değerlendir}(P(t))$
4. Sonuç bulununcaya ve süre bitinceye kadar aşağıdaki bloğu tekrar et
 - a. $t \leftarrow t+1$
 - b. $B(t) \leftarrow \text{Seleksiyon}(P(t))$
 - c. $R(t) \leftarrow \text{Çaprazlama}(B(t))$
 - d. $M(t) \leftarrow \text{Mutasyon}(R(t))$
 - e. $\text{Değerlendir}(M(t))$
 - f. $P(t) \leftarrow \text{Yaşayanlar}(M(t), P(t-1))$

2.3. Kodlama yöntemleri

Kodlama planı, genetik algoritmanın önemli bir kısmını teşkil eder. Çünkü bu plan bilginin çerçevesini şiddetle sınırlayabilir. Öyle ki probleme özgü bilginin bir kromozomsal gösterimiyle temsili sağlanır. *Kromozom* genellikle, problemdeki değişkenlerin belli bir düzende sıralanmasıdır. Kromozomu oluşturmak için sıralanmış her bir değişkene “gen” adı verilir. Buna göre bir *gen* kendi başına anlamlı genetik bilgiyi taşıyan en küçük genetik yapıdır. Mesela; 101 bit dizisi bir noktanın x-koordinatının ikilik düzende kodlandığı gen olabilir. Aynı şekilde bir

kromozom ise bir ya da daha fazla genin bir araya gelmesiyle oluşan ve problemin çözümü için gerekli tüm bilgiyi üzerinde taşıyan genetik yapı olarak tanımlanabilir. Örnek vermek gerekirse; 100011101111 kromozomu x_1 , y_1 , x_2 , y_2 koordinatlarından oluşan iki noktanın konumu hakkında bilgi verecektir. Buna ikili kodlama ismi verilir, çünkü her gen bir bitten oluşmaktadır. Kodlama türleri:



Şekil 2.1. Genetik algoritmanın şematik gösterimi.

- İkili Kodlama (Binary Encoding)
- Tamsayı Kodlama (Integer Encoding)

- Karakter Kodlama (Character Encoding)
- Gerçel Sayı Kodlama veya Değer Kodlama (Floating-Point Encoding).

şeklinde dört gruba ayrılırlar. Bir gen, ikili kodlama bir biti, tamsayı kodlamada bir tamsayıyı, karakter kodlamada bir karakteri ve gerçel sayı kodlamada bir gerçel sayıyı temsil eder.

Bu parametreleri kodlarken dikkat edilmesi gereken en önemli noktalardan biri ise kodlamanın nasıl yapıldığıdır. Örnek olarak, kimi zaman bir parametrenin doğrusal ya da logaritmik kodlanması GA performansında önemli farka yol açar.

Kodlamanın diğer önemli bir hususu ise kodlama gösteriminin nasıl yapıldığıdır. Bu da yeterince açık olmamakla birlikte GA performansını etkileyen bir noktadır.

2.4. Uygunluk Değerlendirme Teknikleri

Başlangıç topluluğu bir kez oluşturulduktan sonra evrim başlar. Genetik algoritma bireylerin uygunluk ve iyiliklerine göre ayrılıp fark edilmesine gerek duyar. Uygunluk, topluluktaki bir kısım bireyin problemi nasıl çözeceği için iyi bir ölçüdür. O problem parametrelerini kodlamayla ölçülür ve uygunluk fonksiyonuna giriş olarak kullanılır. Yüksek ihtimalle uygun olan bu üyeler tekrar üreme, çaprazlama ve mutasyon operatörleriyle seçilirler.

Bazı problemler için bireyin uygunluğu, bireyden elde edilen sonuç ile tahmin edilen sonuç arasındaki hatadan bulunabilir. Daha iyi bireylerde, bu hata sifıra yakın olur. Bu hata genellikle, girişin tekrar sunulacak kombinasyonlarının ortalaması veya toplamıyla hesaplanır (değerler değişkenlerden bağımsızdır). Beklenen ve üretilen değer arasındaki korelasyon etkeni, uygunluk değerini hesaplamak için kullanılabilir.

Amaç fonksiyonu (değerlendirme fonksiyonu, uygunluk fonksiyonu) her bir kromozomun durumunu değerlendirmek için mekanizmayı sağlayan ana bir kaynaktır. Bu GA ve sistem arasında önemli bir bağlantıdır. Fonksiyon giriş olarak kodu çözülmüş şekilde kromozom alır ve kromozomun performansına bir ölçü unsuru olarak bir amaç değer üretir. Bu işlem bütün bireyler için gerçekleştirilir ve hesaplama sonucunda bireylerin ayırt edilebilmesini sağlayan ayrıık değerler elde edilir ve bu değerler bireyin maliyetini belirler. Populasyondaki diğer bireyler de hesaba katıldığı zaman, her bireyin uygunluk değeri kendi maliyetinden elde edilir.

GA' nın maksimizasyon tabiatı ve minimizasyon problemleri arasında yüzleşmeden daha önemli olan nokta maliyet ile uygunluk değeri arasındaki ayırımın iyi yapılması gerekiyor. Aslında, maliyet probleme has olan bir durum olduğundan, belki de gelecekte bu değer hiç değişmeyecektir. Bundan başka maliyet genellikle sıralı bir kümedeki değerleri düşünür. Diğer yünden maliyet-uygunluk eşleşmesinin yapılması seçme süreci ile ilişkilidir. Bu, negatif olmayan gerçel sayılar kümesinin bir alt kümesine olan eşleştirmedir.

Normalleştirilmiş uygunluk değeri, bir sonraki nesilde uygunluk değerinin ait olduğu bireyin kaç tane çocuğunun olacağını gösterir. Normalleştirilmiş uygunluk değeri $[0,1]$ aralığında değer alır ve bundan dolayı bazı durumlarda olasılık değeri olarak ele alınabilmektedir. Bununla birlikte, bu durum bazı seçme yöntemleri için doğru olabilir.

Diğer bir durumda izafi uygunluk değeridir ve her birey için bu değerinde elde edilebilmesine yönelik olarak bütün uygunluk değerleri toplandıktan sonra ortalaması alınır ve her bireyin uygunluğu bu ortalama değere bölünerek ait olduğu bireyin uygunluk değeri elde edilir. İzafi uygunluk değeri halihazırdaki popülasyonun bir önceki nesile göre iyiye mi gidiyor, yoksa kötüye mi gidiyor bunun anlaşılmasında yardımcı olur.

Popülasyon içindeki bireyler sıralanırken, maliyet-uygunluk eşleştirmesi ölçeği korunabilir olduğu gibi korunmayabilir de. Belli başlı iki tane uygunluk değeri atama yöntemi vardır.

Ölçekleme Uygunluk: Uygunluk, maliyetin bir fonksiyonu olarak hesaplanır ve popülasyondaki en iyi kromozoma diğer kromozomların kontrolü verilirken, uygunluk değerlerinin negatif olmadığından emin olunması gerekiyor.

Rank Uygunluk: Popülasyondaki bireylerin maliyetlerinin sıralanması ile eşleştirme elde edilir ve her bireye kendi pozisyonu ile ilişkili bir uygunluk değeri verilir.

Ölçekleme, daha geleneksel bir yöntemdir. Ham uygunluk değeri, maliyetin monotonik fonksiyonu olarak hesaplanır ve daha sonra lineer olarak ölçeklendirilirler. İlk zorluk bu aşamada ortaya çıkar; ölçekleme, farklı bireyler arasındaki performansı korumayı amaçlarken, başlangıç dönüşümü ve devamındaki dengeleme, sonuçta bireylere atanan ham uygunluk değerlerini önemli derecede etkilerler.

Ölçeklendirme ile diğer bireylerden daha güçlü olan bireye diğer uygunluk değerlerine göre çok büyük değer verilebilir ve seçmede hızlı bir şekilde popülasyonda bu bireyin baskısı hissedilir. Tersine olarak, eğer bireylerin çoğu birbirlerine benzer olarak daha çok veya daha az iş yapıyorlarsa, diğer bireylere göre en iyi bireyin avantajı minimumdur ve bunun sonucunda arama işlemi amaçsız bir yola girebilir.

Yukarıdaki iki durumda arzu edilmediği zamanlarda en iyi kromozomun ham uygunluk değerinin başka bir yolla kontrol edilmesi gerekir. Örneğin, Sigma-ölçekleme ortalamadan yola çıkarak arzu edilen ofset miktarını hesaplar ve her nesildeki ham uygunluk değerleri arasındaki standart sapmaları da hesaplar, böylece en iyi bireyin uygunluk değeri daha anlamlı bir şekilde belirlenmiş olur.

Rank, problem formülasyonunda elimine ettiği bazı hassaslıklardan dolayı aynı zorluk ile karşılaşır. Çünkü en iyi bireye her zaman aynı ham uygunluk değeri atanır. Benzer bir şekilde

bütün bireyler birbirlerine çok yakın oldukları zaman (birbirlerine çok benzemeye başladıkları zaman) hala en iyi birey kalan için açık bir şekilde tercih edilir.

Rank-tabanlı uygunluk atanması, genellikle lineer veya eksponansiyel olarak seçilen rank-uygunluk eşleştirmesi ile karakterize edilir. N tane kromozomu olan bir populasyon için en iyi bireyin rank işlemi yapılacak olursa, bireyler $p_0, p_1, \dots, p_{N-1}$ için rank τ ve izafi uygunluk $f=f(\tau)$ olmak üzere eşleştirme lineer için

$$f(\tau)=s-(s-1)\frac{2\tau}{N-1} \quad (2.1)$$

şeklinde olur ve $1 < s \leq 2$ ve bu en iyi birey için istenen izafi uygunluktur.

$$\sum_{i=0}^{N-1} f(i) = N \quad (2.2)$$

şartı sağlanırken, bütün bireylerin uygunluk değerlerinin negatif olmaması için s parametresine bir üst sınır gerekiyor.

Eksponansiyel seçildiği zaman

$$f(\tau)=\rho^{\tau} \cdot s \quad (2.3)$$

ve $s > 1$ ve en iyi birey için arzu edilen uygunluk değeri s' tir ve ρ aşağıdaki şartı sağlar.

$$\sum_{i=0}^{N-1} \rho^i = \frac{N}{s} \quad (2.4)$$

s için bir üst sınır olmadığından eksponansiyel eşleştirme, lineer eşleştirmeden daha esnekler.

$1 < s \leq 2$ için rank-tabanlı uygunluk atanmasında lineer ve eksponansiyel arasındaki en önemli fark eksponansiyel eşleştirme kötü bireyleri fazla cezalandırmaz. Sonuç olarak eksponansiyel eşleştirme genellikle daha geniş bir alanda arama işlemini devam ettirir.

Uygunluk fonksiyonu inşa edilirken bazı problemler ortaya çıkabilir ve bu problemler kısaca aşağıda vurgulanmıştır:

- Uygunluk fonksiyonu, problemin minimizasyon veya maksimizasyon olmasına bağlıdır.
- Ortam uygunluk fonksiyonun hesaplanmasını etkileyebilir (örneğin kısmen hesaplama).
- GA' nın icra edilmesi sırasında uygunluk fonksiyonu dinamik olarak değişebilir.

- Uygunluk fonksiyonu çok karmaşık olduğu durumlarda sadece yaklaşım ile değeri hesaplanabilir.
- Uygunluk fonksiyonu seçme işleminin sağlıklı yapılabilmesi için bireylere oldukça farklı değerler vermelidir.
- Uygunluk fonksiyonu, problemin sınırlayıcılarını hesaba katmalıdır. Örneğin, eğer mümkün olmayan çözüm geldiğinde uygunluk fonksiyonu o bireye kötü anlamına gelecek bir değer vermelidir.

2.5. Genetik Operatörler

Genetik algoritmanın işleme sürecinde popülasyonun çeşitliliği koruması veya daha iyi çözümleri yakalayabilmesi için büyük sıçrama gerektiği durumlarda kullanılan iki tane genetik operatör kullanılırken bir sonraki nesilde varlıklarını idame edecek bireylerin belirlenmesi için bir tane genetik operatör kullanılır. Bunlar:

- Seçme
- Çaprazlama
- Mutasyon

şeklindedir. Bu operatörlerden seçme operatörü kromozom tabanlıyken çaprazlama ve mutasyon *gen* tabanlı operatörlerdir. Seçme işlemi, bir nesilden bir sonraki nesile geçilirken kromozomların uygunluk değerlerine bakılarak hangilerin eleneceğine ve hangilerin bir sonraki nesilde varlıklarını sürdüreceğine karar verme mekanizmasına seçme adı verilir. Hatta uygunluk değeri iyi olan bireylerden elenen bireylerin yerlerine birer kopyaları alınır veya iyi olan bireyler çaprazlandıktan sonra elenen bireyler yerine yeni bireyler konulur.

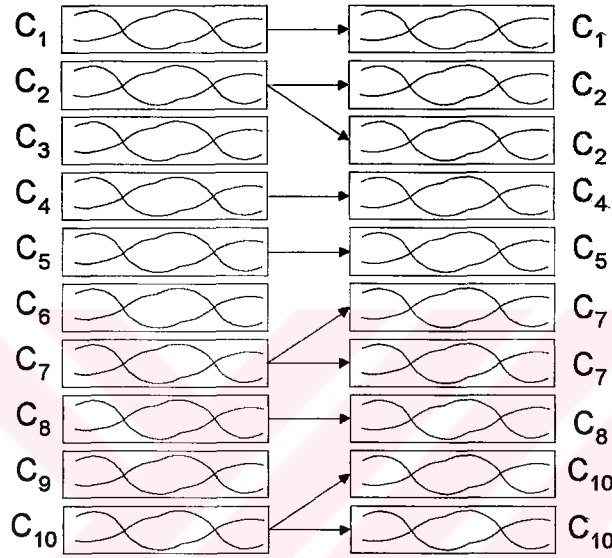
2.5.1. Seçme

t anındaki popülasyon P_t olmak üzere seleksiyon operatörü (S) aşağıdaki gibi bir bağıntıdır.

$$S: P_t \rightarrow P_t$$

Seçme operatörü, halihazır popülasyondan bazı kriterlere göre kromozomların bir kısmı seçip bir sonraki jenerasyona aktarırken bir kısmını eleyen bir operatördür. Şekil-2.2' de basit bir seçme işlemi görülmektedir. Bu şekilde $C_1, C_2, C_4, C_5, C_7, C_8$ ve C_{10} kromozomları bir sonraki nesilde varlıklarını sürdürürken (seçilen bireyler olurken), C_3, C_6 ve C_9 kromozomları

bir sonraki jenerasyona geçebilecek kadar iyi olmadıklarından dolayı elenen kromozomlardır. Bu kromozomların yerlerine sırası ile C_2 , C_7 ve C_{10} kromozomlarının birer kopyası alınmıştır. Genetik algoritmanın süreci bu şekilde devam ederse, belli adım sonunda bütün kromozomlar aynı olacaktır. Bu durum çoğu zaman prematüre çözümler vereceğinden dolayı pek istenmeyen bir durumdur. Başka bir deyişle, seçme operatörü temelde iki farklı şekilde kullanılır. Bunlardan birincisi jenerasyonsal seçme ve bu durumda popülasyonun hepsi yeni bir popülasyon ile yer değiştirir. İkinci bir kullanım şekli de kararlı-durum seçmesidir ve bu durumda her jenerasyondaki çok az bireyin yerine yeni birey alınır.



Şekil 2.2. C_i , Kromozom, $1 \leq i \leq 10$, göstermek üzere basit bir seçim işlemi.

Seçme operatörünün prematüre çözümler vermesini engellemek veya iyi çözüm elde etmek için bu operatörün değişik uygulanma şekilleri vardır. Bunlar

- Oransal Seçim
- Rulet Tekerleği Seçimi
- Stokastik Evrensel Örnekleme
- Rank Seçimi
- Kararlı Durum Seçimi
- Turnuva Seçimi
- Genel Elitizm
- Yerel Elitizm
- Kalabalıklaştırma
- Deterministik Kalabalıklaştırma
- Paylaşım

Seçme işlemi, bireylerin uygunlukları ile orantılı olarak bir bireyin bir sonraki nesil için bir veya birden fazla birey üretmesi işlemidir. Bununla birlikte seçilecek birey sayısı tamsayı olduğundan genellikle üretilebilecek birey sayısı ya istenenden çoktur ya da azdır. Bu durumda seçme hatası denilen bir hata sistematik olarak her nesilde vuku bulacaktır.

Seçme işleminin hatasından kurtulmak için farklı seçme yöntemleri kullanılmıştır ve bunlardan bir kısmı bundan sonraki bölümlerde kısaca açıklanacaklardır.

Oransal seçim: Oransal seçim yönteminde popülasyonun ortalama uygunluk değeri bulunur ve bu değer $\tilde{F}$ ile gösterilsin. $\forall p \in P$ için $F(p)$ değeri p kromozomunun uygunluk değerini göstermek üzere p kromozomundan seçilecek kopya sayısı

$$\left\| \frac{F(p)}{\tilde{F}} \right\| \quad (2.5)$$

kadar olur. Bu değer tamsayıya yuvarlandıktan sonra kaç çıkıyorsa, ilgili kromozomdan o kadar kopya seçilir. Bu yöntemin amacı, uygunluk değeri yüksek olan bireylerin bir sonraki nesil popülasyonu için daha önemli olduğunu ve bundan dolayı bu bireylerin bir sonraki nesilde daha baskın (daha çok) olmaları gerekiyor ve bu sebeple bu bireyleri tespit edip seçmektir. Eğer bu kesirli değer küçük ise, bir sonraki nesilde bu bireyin var olması popülasyonun toplam kalitesini fazlaca yükseltmeyeceğinden dolayı bu bireyin seçilme sayısı mümkün olan en küçük sayıya indirgenmeye çalışılmaktadır ve hatta mümkünse, bu bireyden hiç seçmemek eğilimindedir.

Rulet Tekerleği Seçimi: Rulet tekerleği seçimi, bir çok araştırmacı tarafından veya programcı tarafından tercih edilen bir seçim yöntemidir (Goldberg, 1989). N tane kromozom içeren bir popülasyon verilsin; her bireyin izafi uygunluk değeri hesaplanır ve bu değer seçme olasılığında kullanılır. X , popülasyon havuzu üzerinde tanımlanmış bir gelişigüzel değişken olsun ve $f_i \in R^+$ değeri havuzdaki x_i kromozomun uygunluk değerini belirtsin. Maksimizasyon problemleri için

$$P(X = x_i) = \frac{f_i}{\sum_{i=1}^N f_i} \quad (2.6)$$

değeri x_i kromozomun seçilme olasılığıdır. Minimizasyon problemleri için halihazırdaki en büyük uygunluk değerine göre izafi standart sapma uygunluk değeri olarak kullanılır. O anki popülasyon için f_{max} maksimum uygunluk değeri olsun.

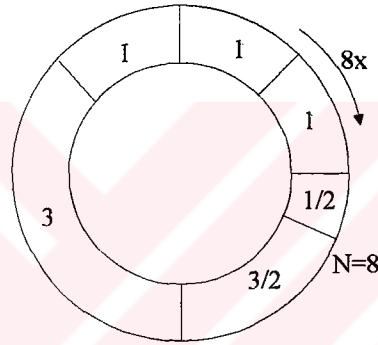
$$f_i' = f_{\max} - f_i \quad (2.7)$$

için x_i bireyini seçme olasılığı

$$P(X = x_i) = \frac{f_i'}{\sum_{i=1}^N f_i'} \quad (2.8)$$

şeklinde olur.

Şekil 2.3' te 8 kromozom için rulet tekerleği görülmektedir ve bu tekerlek üzerinde her bireyin seçilebilme olasılığı görülmektedir. Dikkat edilirse, herhangi bir anda sadece bir tane kromozom seçilir, çünkü bir tane başlık (yönlü doğru parçası) vardır.



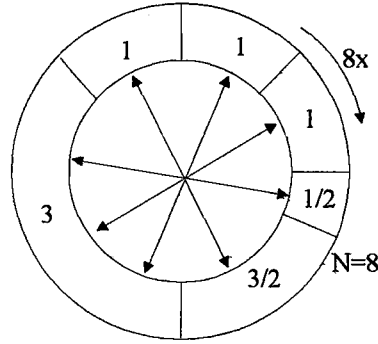
Şekil 2.3. Rulet tekerleği seçimi.

Stokastik Evrensel Örnekleme: Seçme işleminin sistematik hatasından kurtulmak için stokastik olarak seçme işlemi uygulanır. Stokastik seçme hatası mümkün oldukça küçük olarak tutulurken, bir bireyden uygunluğu ile orantılı olarak beklenen bireyler üretilir. Bu iki amacı ilk olarak Baker (1987) birleştirip Stokastik Evrensel Örnekleme yöntemini sunmuştur.

Bu yöntemde Rulet tekerleği yönteminde olduğu gibi tekerlek çevrilir ve durduğu zaman dikkat edilirse (Şekil 2.4) birden fazla yönlü doğru parçası vardır. Her doğru parçasının işaret etmiş olduğu alandaki sayının ya taban fonksiyonu ya da tavan fonksiyonu değeri alınır ve bu değer o bireyden kaç tane bireyin üretileceğini gösterir.

Rank Seçimi: Rulet tekerleği seçiminde eğer uygunluk değerlerinin kapsadığı aralık çok büyürse, bu durumda bazı kromozomların uygunluk değeri çok küçük kalabilir ve bunun sonucunda bu kromozomların seçilme olasılığı sıfıra yakın bir değer olabilir. Bu problemi gidermek için eğer popülasyondaki kromozom sayısı P ise uygunluğu en iyi olan kromozoma P değeri, ondan sonraki kromozoma $P-1$ değeri ve bu şekilde devam ederek en kötü

kromozoma 1 değeri verilir. Bunun anlamı en kötü kromozomun seçilme olasılığı sıfıra yakın bir değerden



Şekil 2.4. Stokastik evrensel örnekleme yöntemi ile seçim yapmak.

$$\frac{2}{P(P+1)}$$

(2.9)

olur. Bu işlem yapıldıktan sonra rulet tekerleği işlemi bu yeni uygunluk değerlerine göre yapılır.

Kararlı Durum Seçimi: Bu seçim yönteminin temel amacı, iyi kromozomların bir sonraki nesilde varlıklarını sürdürmelerini sağlamaktır. Böylece için bu algoritmanın uygulamasını gerçekleştiren kişinin belirlediği bir oranda iyi kromozomlar seçilir ve kötü kromozomların yerine iyi kromozomlardan çaprazlama ile veya mutasyon ile yeni bireyler üretilerek bir sonraki nesil oluşturulur.

Turnuva Seçimi: Genetik algoritmalarda kullanılan seçim mekanizmalarından biri de Turnuva seçimidir ve bu yöntemde bireyler bir eşleştirme havuzuna atılırlar (Hancock, 1994). Eşleştirme havuzundaki bireylerden evlatlar üretilir ve bu evlatlar bir sonraki neslin temelini teşkil ederler. Seçme mekanizmasının temel amacı iyi bireylerin seçilmesi yönünde baskı uygulamasıdır ve bunun sonucunda daha iyi bireylerin seçilmesi sağlanır. Seçme baskısı, iyi bireylerin seçilmesi yönünde hissedilen baskıdır ve bu baskı arttıkça iyi bireylerin seçilme şansı artar. Seçme baskısı çözüme yakınsama oranını artırır. Yüksek seçme baskısı beraberinde yüksek yakınsama oranını getirirken, düşük seçme baskısı yanında düşük yakınsama oranını getirir ve bunun sonucunda daha fazla iterasyon ile belki çözüme gidilebilir. Çok yüksek seçme baskısı da istenmeyen bir durumdur ve prematüre çözümlere sebep olur.

Bu seçme yönteminde uygunluk değeri en yüksek olan birey eşleştirme havuzuna eklenir ve sonra geriye kalan bireyler için aynı işlem gerçekleştirilir. Bu işleme s tane birey seçilip eşleştirme havuzuna eklenene kadar devam edilir, çünkü turnuvanın boyutu s olduğu kabul

edilmiştir. Bu şekilde havuzdaki populasyonun ortalama uygunluk değeri daha yüksek olacaktır ve bunun sonucunda seçme baskısı ortaya çıkacaktır.

Global Elitizm ve Yerel Elitizm: Bu yöntem seçme algoritması olmaktan çok en iyi çözümü (kromozomu) korumaya yönelik bir yöntemdir. İkiye ayrılmasının sebebi, global olarak bir bireyin korunmasını sağlamak veya yerel olarak bireylerin korunmasını sağlamak amacındadır.

Kalabalıklaştırma: Nesiller arasındaki dağılım farkını minimuma indirerek prematüre çözümlerin oluşmasını engelleme yöntemine Kalabalıklaştırma (Crowding) denir (De Jong, 1975). Bu yöntemde populasyonun belli bir orana göre bir kısmı seçilir ve bir sonraki nesil oluşturulur. Bu yöntemin temel prensibi aşağıdaki gibi özetlenebilir:

Var olan populasyondan evlatlar üretilir ve sonuç evlatları populasyondaki bireylerin yerlerini alırken, her evlat için populasyonda gelişigüzel olarak belli sayıda birey seçilir ve en çok benzeyenin yerine evlat konulur. Benzerlik metriği tanım kümesinden bağımsız olabileceği gibi tanım kümesine bağımlı da olabilir. Yerine yeni birey seçilecek olan, en çok benzeyen birey seçilerek bir nesilden bir sonraki nesile geçilirken populasyon dağılımları arasındaki farkın minimum olmasına çalışılır. Bu durum genetik algoritmanın tek çözüme meyilli olarak davranmasını engeller.

Deterministik Kalabalıklaştırma: Kalabalıklaştırma yöntemini geliştirmek için evlatlar ataları ile karşılaştırılırlar ve eğer evladın uygunluk değeri atanın uygunluk değerinden daha iyi ise, ata yerine evlat populasyonda bir birey olarak yer alır ve atanın populasyondaki ömrü son bulur ve buna deterministik kalabalıklaştırma yöntemi adı verilmiştir (Mahfoud, 1992). Bu durum genetik algortmada seçme operatörünün baskısının populasyonda hissedilmesini sağlar.

Paylaşım: Genetik algortmada populasyonun dağınıklık özelliğini kaybetmesi istenmez. Bunun için seçim operatörünün değişik tanımlamaları ortaya çıkmıştır ve bunlardan biri de Paylaşım yöntemidir (Holland, 1975; Goldberg v.d., 1987).

Bu yöntemin temel amacı, populasyondaki benzer bireylerin uygunluk değerlerinin azalması ve çözüm için kaliteli aday olma özelliklerinin yavaş-yavaş kaybolmasıdır. Bu yöntemde benzeri olmayan bireyler bir sonraki nesile taşınmaları için cesaretlendirilirken, benzeri olan bireylerin bir sonraki nesilde var olmaları ile ilgili cesaretlerinin kırılması durumudur. Bu populasyon dağınıklık baskısına sebep olur; bu şekilde bireylerin geçici olarak yerel çözümlerde beklemeleri sağlanmaya çalışılır.

i bir birey olmak üzere bu bireyin populasyonun geriye kalan kısmı ile örtüşen gen sayısı m_i olsun. i bireyinin ham uygunluk değeri f_i gösterilsin. i bireyinin paylaşılmış uygunluk değeri

$$f_{sh,i} = \frac{f_i}{m_i} \quad (2.10)$$

olur. Ortak olan gen sayısını bulmak için bütün bireyler üzerinde paylaşma fonksiyonunun toplamı yapılır ve

$$m_i = \sum_{j=1}^N sh(d_{i,j}) \quad (2.11)$$

toplamı bize bu değeri verir. $d_{i,j}$ değeri i ve j bireylerinin birbirine olan uzaklığıdır. Bir noktada, benzerliğin tersi olarak da ele alınabilir.

Eğer paylaşma fonksiyonu uzaklığın belli bir yarıçap σ_{sh} içerisinde kaldığını tespit ederse, paylaşma fonksiyonu 0 ile 1 arasında bir değer üretir ve bu değer daha büyük benzerliklerde artar, yani $d_{i,j}$ azalırken, fonksiyon değeri artar. Diğer durumda 0 değerini verir. Aslında populasyonun her bireyi bir bölgenin merkezi olarak kabul edilir ve bulunduğu bölgenin yarıçapı σ_{sh} için populasyonun diğer bireylerinden uzaklığı σ_{sh} küçük olan her birey için o bireyin paylaşılmış uygunluk değeri azalır ve aşağıdaki gibi verilebilir.

$$sh(d_{i,j}) = \begin{cases} 1 - \left(\frac{d_{i,j}}{\sigma_{i,j}} \right)^\alpha & d_{i,j} < \sigma_{sh} \text{ ise} \\ 0 & \text{aksi halde} \end{cases} \quad (2.12)$$

2.5.2. Çaprazlama

Çaprazlama operatörü $\otimes$ simgesi ile gösterilmek üzere ve P populasyonu için $\exists p_1, p_2, p_3, p_4 \in P$ için

$$\otimes: P \times P \rightarrow P \times P \text{ veya } \otimes: P \times P \rightarrow P$$

şeklinde olan bir operatördür ve

$$p_1 \otimes p_2 = \{p_3, p_4\} \text{ veya } p_1 \otimes p_2 = \{p_3\}$$

olur.

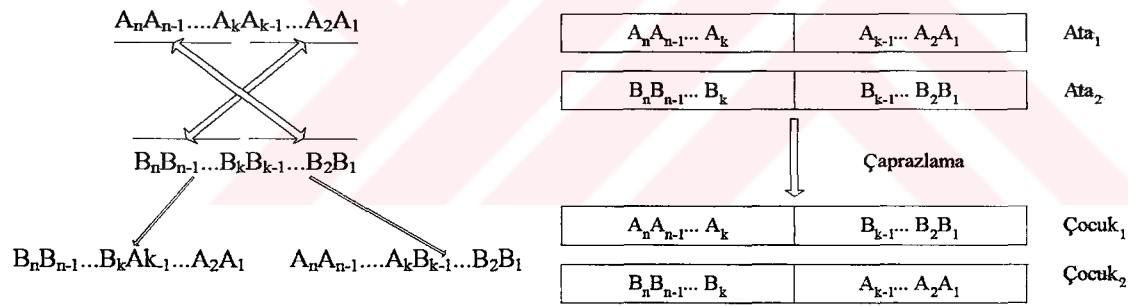
Bu tez çalışmasında kullanılan çaprazlama çeşitleri:

- Tek-noktalı çaprazlama
- Çift (iki)-noktalı çaprazlama
- Düzenli çaprazlama
- Yoğun çaprazlama

Tek-Noktalı Çaprazlama: Bu çaprazlama yönteminde kromozomun uzunluğu içinde gelişigüzel bir sayı üretilir ve bu noktadan seçilen iki ata kromozom için bölme noktası olur. Bir kromozomun en önemli parçası alınırken diğer kromozomun en önemsiz parçası alınarak bir tane çocuk oluşturulur.

Şekil 2.5’ de $A_n A_{n-1} \dots A_k A_{k-1} \dots A_2 A_1$ ve $B_n B_{n-1} \dots B_k B_{k-1} \dots B_2 B_1$ seçilen atalar görülmektedir. Gelişigüzel bir çaprazlama noktası seçildikten sonra atalar $A_n A_{n-1} \dots A_k | A_{k-1} \dots A_2 A_1$ ve $B_n B_{n-1} \dots B_k | B_{k-1} \dots B_2 B_1$ şeklinde olsunlar. Bu iki atadan elde edilen çocuklar $A_n A_{n-1} \dots A_k B_{k-1} \dots B_2 B_1$ ve $B_n B_{n-1} \dots B_k A_{k-1} \dots A_2 A_1$ şeklinde olurlar ve bu durum Şekil 2.5’ te görülmektedir.

İki-noktalı çaprazlamada gelişigüzel olarak kromozomun uzunluğu içinde kalacak şekilde iki tane nokta belirlenir. Bir kromozomda iki nokta arasında kalan kısım alınırken diğer kromozomda iki nokta dışında kalan kısımlar alınır ve bu şekilde bir tane çocuk oluşturulur. Diğer çocuğun oluşturulması için ataların rolleri değiştirilir.



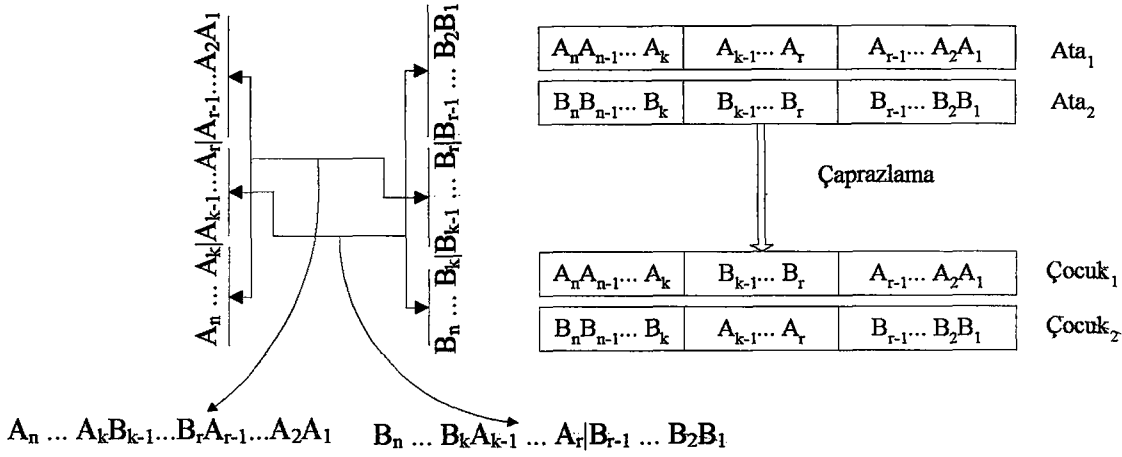
Şekil 2.5. Tek-noktalı çaprazlama işlemi.

Şekil 2.6’ da iki-noktalı çaprazlama işlemi şekil ile gösterilmiştir. Ata kromozomlar $A_n \dots A_k A_{k-1} \dots A_r A_{r-1} \dots A_2 A_1$ ve $B_n \dots B_k B_{k-1} \dots B_r B_{r-1} \dots B_2 B_1$ olmak üzere gelişigüzel oluşturulan iki nokta belirlendikten sonra atalar $A_n \dots A_k | A_{k-1} \dots A_r | A_{r-1} \dots A_2 A_1$ ve $B_n \dots B_k | B_{k-1} \dots B_r | B_{r-1} \dots B_2 B_1$ şeklinde olur. Bunun sonucunda çaprazlama işlemi yapıldığında elde edilen çocuklar $A_n \dots A_k B_{k-1} \dots B_r A_{r-1} \dots A_2 A_1$ ve $B_n \dots B_k A_{k-1} \dots A_r | B_{r-1} \dots B_2 B_1$ şeklinde olurlar.

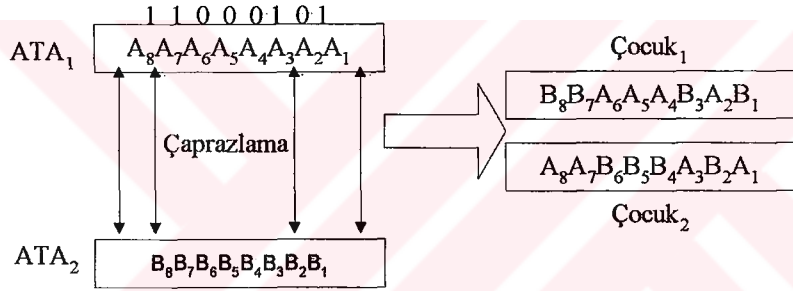
Düzenli çaprazlama yönteminde her gen için gelişigüzel bir sayı üretilir ve bu sayı 0 veya 1 değeri alabilmektedir. Eğer bu sayı 1 değeri almışsa, iki ata arasında bu gen yer değiştirecektir. Eğer bu sayı 0 ise, her iki atanın bu geni sabit kalacak demektir.

Örneğin ata kromozomlar $A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1$ ve $B_8 B_7 B_6 B_5 B_4 B_3 B_2 B_1$ olmak üzere her gen için gelişigüzel üretilen sayılar dizisi 11000101 olsun. Bu durumda yapılan çaprazlama

sonucunda elde edilen çocuklar $B_8B_7A_6A_5A_4B_3A_2B_1$ ve $A_8A_7B_6B_5B_4A_3B_2A_1$ olurlar. Şekil 2.7’ de düzenli çaprazlama işleminin grafiksel gösterimi görülmektedir.



Şekil 2.6. İki-noktalı çaprazlama işlemi.



Şekil 2.7. Düzenli çaprazlama işlemi.

2.5.3. Mutasyon

GA’ da işlemler devam ettikçe kromozomlar birbirine benzemeye başlarlar ve bunun sonucunda algoritma bitime gider. Eğer elde edilen çözüm optimal veya optimal çözüme yakın bir çözüm ise problem yoktur. Fakat, elde edilen çözüm optimal çözümden çok uzak ise bu çözüme prematüre çözüm denir ve kromozomlar içerisinde bazılarının büyük sıçramalara tabi tutulmaları gerekmektedir. Bunu yapmanın yollarından biri çaprazlama idi ve diğeri de mutasyon işlemidir.

P popülasyonu için mutasyon operatörü M aşağıdaki gibi tanımlanır.

$M:P \rightarrow P$

ve bağıntı bir fonksiyondur. Bir fonksiyonun tersinin var olması için o fonksiyonun bire-bir ve örten olması gerekir. Mutasyon işlemi için verilen fonksiyon bire-bir veya örten olmak zorunda olmadığından dolayı mutasyon işlemi her zaman tersinir bir işlem değildir. $\exists p, q \in P$ için

$$M[p]=q$$

olur. Bu işlemin anlamı p kromozomuna mutasyon işlemi uygulandıktan sonra q kromozomu elde edilmektedir.

Bir gene mutasyon işlemi uygulamak ya o genin tümleyeni alınır veya aynı kromozom içerisinde iki genin yerleri birbiri ile değiştirilir. En çok kullanılan mutasyon yöntemi, genin tümleyenin alınmasıdır.

2.6. Genetik Algoritma Çeşitleri

Bu tezde önereceğim genetik algoritma çeşidi dışında en çok kullanılan yedi tane genetik algoritma çeşidi vardır. Bunlar:

- Mikro-Genetik Algoritmalar
- Basit Genetik Algoritmalar
- Kararlı Durum GA (Genitor)
- CHC
- Hibrid Genetik Algoritmalar
- Paralel Genetik Algoritmalar

2.6.1. Mikro-Genetik Algoritmalar

Mikro-Genetik algoritmalarda, eğer kullanılan popülasyonun boyutu 10'dan daha az ise, bu algoritmaya Mikro-Genetik Algoritma denir. Diğer genetik algoritmalarda kullanılan kodlama yöntemleri, değerlendirme yöntemleri ve genetik operatörler aynıdır. Tek fark, kromozom sayısının az olmasıdır.

Kromozom sayısının az tutulmasının sebebi, GA iteratif bir yöntem olduğundan dolayı, algoritmanın icra edilmesi zamanı, popülasyon boyutunun lineer veya kuadratik veya kübik bir fonksiyonudur. Bundan dolayı, popülasyon boyutu küçültüldüğünde işlem zamanı doğal olarak kısalmaktadır.

2.6.2. Basit Genetik Algoritmalar

Bu genetik algoritmaların genel özellikleri şu şekilde sıralanabilir.

- Çaprazlama şekli tek-noktalı veya çift-noktalı.
- Seçim şekli rulet tekerleği veya turnuva.
- Mutasyon işlemi ile beraber elitizm kullanılır.

2.6.3. Durağan Durum Genetik Algoritmalar (Genitor)

Genitor veya Durağan Durum Genetik Algoritmanın genel özellikleri şu şekilde sıralanabilir.

- Rank tabanlı seçim.
- İki noktalı çaprazlama.
- Üretilen iki çocuktan gelişigüzel birinin seçilmesi.
- Mutasyon.
- Ekle ve en kötünün yerini değiştir.

2.6.4. CHC

Bu genetik algoritmanın özellikleri şu şekildedir.

- Populasyon boyutu küçük seçilir. En fazla 50 kromozomlu populasyon üretilir.
- Kısıltmalı seçim ile beraber gelişigüzel çiftleştirme.
- Yeterince farklı olan bireylerin çiftleşmelerine izin verilir.
- En iyi çözümle beraber gelişigüzel çözüm ile bir sonraki iterasyona başlanır.
- Yarı düzenli çaprazlama.

2.6.5. Hibrid Genetik Algoritmalar

Basit genetik algoritmaya yerel arama özelliğinin katılması ile elde edilen bir yöntemdir. Yerel arama adımı üç tane parametre ile tanımlanır. Yerel aramanın frekansı, yerel aramanın olasılığı ve yerel aramanın iterasyon sayısıdır. Algoritmanın tanımlanması için ilk eleman yerel arama frekansıdır ve bu global ve yerel arama arasındaki geçiş sayısını belirler.

Bu yöntemin amacı, global olarak populasyon biraz iyileştirildikten sonra yerel olarak yapılabilecek iyileştirilmelerin yapılmasıdır. Ondan sonra eğer hala çözüme ulaşılmadıysa, genel olarak biraz daha iyileştirme sağlanır ve bunun peşinde yerel iyileştirme yapılır. Bu şekilde çözüm bulununcaya kadar devam edilir.

2.6.6. Paralel Genetik Algoritmalar

Bu genetik algoritma, paralel mimariler üzerinde kullanılan bir yöntemdir. Kendi içerisinde çeşitlilik gösterir ve bunlar şu şekilde sıralanabilir.

- **Sahip-Köle (Master-Slave):** Bu yöntemde bir tane populasyon vardır ve çaprazlama, mutasyon ve seçme işlemi bu populasyonun üzerinde bulunduğu işlemci tarafından gerçekleştirilirken, köle işlemciler değerlendirme işlemini gerçekleştirirler.
- **Birden Fazla Populasyon Olması:** Birden fazla populasyon vardır ve her populasyon kendi içerisinde genetik süreçten geçer.
- **İyi Parçacıklama:** Bu yöntemde her işlemci kendi üzerinde yerel olarak seçme ve çiftleştirme yeteneğine sahiptir. Aynı zamanda bu yöntem hücresel paralel genetik algoritma olarak da bilinmektedir.
- **Hiyerarşik Kombinasyonlar:** Birden fazla populasyon vardır ve her populasyonun bulunduğu işlemci üzerinde çaprazlama, mutasyon ve seçme işlemleri gerçekleştirilirken bu işlemcinin köle işlemcisi durumundaki işlemciler değerlendirme işlemini gerçekleştirirler. Bu şekilde Sahip-Köle yöntemi ile Birden Fazla Populasyon yöntemlerinin beraber kullanılmasını sağlar.

3. DÜZENLİ POPULASYON YÖNTEMİ VE GELİŞİGÜZEL POPULASYON YÖNTEMİNE OLAN ÜSTÜNLÜKLERİ

Genetik algoritmaların temel mantığında şunlar vardır; başlangıçta çözülecek problem için gelişigüzel olarak çözümler kümesi üretilir ve daha sonra bu çözümlerin iyileştirilmesi için genetik süreç başlatılır. Genetik süreç içerisinde başlangıç popülasyonu gelişigüzel üretildiğinden, her zaman çözüme ulaşmak yerine bazen yerel çözümde kalınırken bazen de çözümden uzaklaşma durumu ile karşılaşılabilir. Çünkü üretilen popülasyon gelişigüzel bir yapıya sahiptir ve bu popülasyonun çözümden uzakta bir araya toplanmış kromozomları olabilir. Hatta, bütün kromozomları bu özelliğe sahip olabilirler. Bu durumda çözüme ulaşmak çok zordur ve fazla zaman alıcı olur. Bundan dolayı, başlangıç popülasyonunun bu dezavantajlı durumunun giderilmesi gerekir.

Bu problemleri gidermek veya genetik algoritmanın performansını arttırmak için çok sayıda araştırmacı bu konuda araştırma yapmıştır. Bu araştırmalar dört gruba ayrılabilirler: Yeni genetik algoritma çeşidi tanımlayanlar (Kee, v.d., 2001; Huang, 2001; Holland, 2000), genetik operatörler üzerinde işlem yapanlar (Furutani, 2001; Agapie, 2001; Prügel-Bennett, 2001; Hancock, 1994), şema teorisi üzerinde çalışma yapanlar (Goldberg, v.d., 2001; Goldberg, v.d., 2001; Hart, 2001; Stephens, v.d., 1999) ve uygunluk fonksiyonu üzerinde çalışmalar yapanlar (Branke, 2001) şeklinde ayırım yapılabilir. Hiç kimse başlangıç popülasyonu üzerinde işlem yapmayı denememiştir ve bu tezde bu alanda çalışma yapılacaktır.

Sözü edilen iki problemin giderilmesi için popülasyonda yer alan kromozomların kromozom uzayına düzgün bir şekilde dağıtılması gerekir. Bu şekilde bir popülasyon elde edildikten sonra çözüm kromozomuna yakın olan en az bir tane kromozom vardır. Bu şekilde elde edilen popülasyonda yerel çözümde kalma veya çözümden ıraksama gibi problemler olmayacaktır.

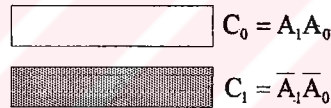
Bu tezin temel amacı, bu özelliğe sahip bir başlangıç popülasyonu üreten yöntem geliştirmektir. Bu amaçla yapılan çalışmalar sonucunda Düzenli Popülasyon adı verilen başlangıç popülasyonu üretme yöntemi geliştirilmiştir. Bu yöntem, başlangıç popülasyonun kromozom uzayına düzenli bir şekilde dağıtılması için böl-ve-yönet mantığı ile tamamen gelişigüzel olmayan bir yöntemdir.

Bu yöntemin kullanımında literatürde geçen genetik operatörlerle beraber bu operatörlerin tez çalışmasında tanımlanmış şekilleri kullanılmıştır. Bununla beraber, düzenli popülasyonun kendi içerisinde barındırmış olduğu alt popülasyonlara bölünebilme özelliğinden yola çıkılarak paralel genetik algoritmaların bir şekli olan Adalı Genetik Popülasyon elde edilmiştir ve bu genetik algoritmanın ardışıl şekli tezde gerçekleştirilmiştir.

3.1. Düzenli Populasyon

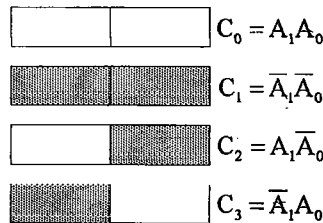
Bu yöntemin ilk adımı olarak bir kromozom gelişigüzel üretilir, bu kromozomun tümleyeni alınır ve bu tümleyen kromozom da populasyon içerisinde bir kromozom olarak yerini alır. Bu şekilde populasyon %50 oranında gelişigüzel üretilirken, geriye kalan %50 ise bu gelişigüzel üretilen kromozomların tümleyenleri olurlar (Şekil 3.1). Bu durumda her kromozom bir parça olarak ele alınmıştır ve parça sayısını gösteren parametre $r=1$ olur (Karcı v.d., 1999; Karcı v.d.; Karcı v.d. 2000; Karcı v.d. 2002).

$r=2$ olduğu durumda populasyon biraz daha düzenli hale gelmektedir. Bu durumda gelişigüzel üretilen kromozom iki parçaya bölünür. İlk olarak sağ yarının tümleyeni ile sol yarının birbirine eklenmesi ile bir kromozom elde edilir. Ondan sonra sol yarının tümleyeni ile sağ yarının birbirine eklenmesi ile başka bir kromozom elde edilir. Son olarak gelişigüzel üretilen kromozomun tümleyeni alınarak başka bir kromozom elde edilir (Karcı v.d.; Karcı v.d. 2002; Karcı v.d. 2000; Karcı v.d., 1999;). Bu şekilde populasyon üretilecek olursa, populasyonun %25 oranında gelişigüzel üretilmesinin yanında %75' i bu gelişigüzel olarak üretilen kromozomlardan üretilen kromozomlar olurlar. Kısaca gelişigüzel olarak üretilen bir kromozomdan 3 tane daha kromozom elde edilir ve bu yöntemin görsel şekli Şekil 3.2' de görülmektedir.



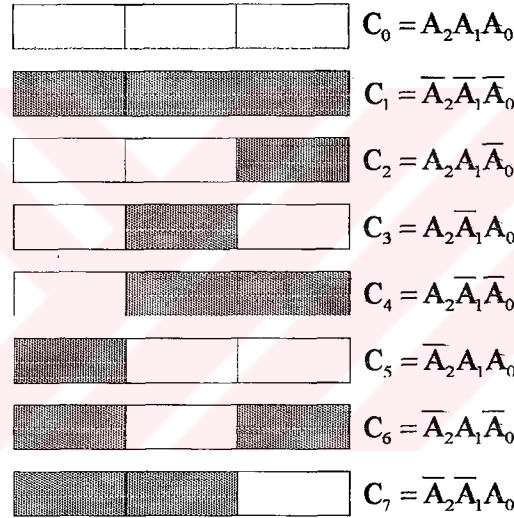
Şekil 3.1. $r=1$ için gelişigüzel üretilmiş olan kromozomdan ikinci kromozomun elde edilmesi.

$r=2$ için eğer ϕ tane gelişigüzel üretilen kromozom varsa, populasyonun boyutu $\phi 2^2=4\phi$ olur. Bu adımda elde edilen populasyon $r=1$ durumundaki populasyondan daha düzenli bir populasyondur. Çünkü dört tane kromozom uzay içerisinde düzenli bir şekilde dağıtılmaya çalışılırken $r=1$ durumunda sadece iki kromozom uzayda düzenli bir şekilde dağıtılmaya çalışılır.



Şekil 3.2. $r=2$ için gelişigüzel bir kromozomdan üretilen kromozomlar.

Populasyon daha düzenli hale getirilmek istendiğinde r değeri artırılır ve $r=3$ olmak üzere gelişigüzel bir kromozom üretilir ve bu kromozomun üç parçaya bölündüğü kabul edilir. Bölme işleminden sonra en sağda kalan parçanın tümleyeni ile ortada ve solda kalan parçaların birbirlerine eklenmesi sonucu yeni bir kromozom elde edilir. Sonra sağ parça, ortadaki parçanın tümleyeni ve sol parçanın eklenmesi ile başka bir kromozom elde edilir. Sol parça, ortadaki parçanın tümleyeni ve sağ parçanın tümleyeninin birbirlerine eklenmesi ile yeni bir kromozom daha elde edilir. Bu şekilde devam edilerek bir kromozomdan 7 tane kromozom elde edilir. Yani gelişigüzel üretilen kromozom sayısı n tane ise populasyon boyutu $8n$ olur. Bunun anlamı populasyon %12.5 oranında gelişigüzel üretilirken, %87.5 oranında gelişigüzel değil, üretilen kromozomlardan türetilen kromozomlardır. Bundan dolayı populasyon biraz daha düzenli hale gelmiş olur. Şekil 3.3' te $r=3$ durumunun görsel şekli görülmektedir (Karcı v.d.; Karcı v.d. 2002; Karcı v.d. 2000; Karcı v.d., 1999).



Şekil 3.3. $r=3$ için gelişigüzel bir kromozomdan 7 tane kromozomun üretilmesi.

$r=m$ için gelişigüzel üretilen kromozom m tane parçaya bölünür ve aynı mantık ile diğer kromozomlar elde edilir. Bunun sonucunda bir tane gelişigüzel kromozomdan 2^m-1 tane kromozom elde edilir.

Eğer n tane gelişigüzel üretilen kromozom var ve $r=m$ ise populasyonun boyutu (3.1) bağıntısında verilen formül ile hesaplanabilir.

$$(2^m-1)\phi + \phi = \phi 2^m \quad (3.1)$$

Buraya kadar sunulan kısımlar ikili kodlama için verilmiştir. Diğer kodlama yöntemlerinin tanımlamaları da yapılabilir.

Gerçel sayı tabanlı kodlamada her gen bir gerçel sayı ile ifade edilir. $R=(r_0r_1r_2...r_{n-1})$ bir gerçel sayı olmak üzere bu sayı mantisa ve üstel olmak üzere iki kısımdan oluşur. Bu bölümde kullanılan gerçel sayıların normalize edildikleri kabul edilecektir ve

$$r_k = r_{mn}r_{m(n-1)}...r_{m1}.r_{cs}r_{c(s-1)}...r_{c1}$$

olmak üzere

$$\begin{aligned}\bar{R} &= \bar{r}_0\bar{r}_1...\bar{r}_{n-1} \\ \bar{r}_k &= (9-r_{mn})...(9-r_{m1}).(9-r_{cs})...(9-r_{c1})\end{aligned}\quad (3.2)$$

olur. Bu yöntemde genin kendisi parçalara bölünerek tümleyen alma işlemi gerçekleştirilmiştir. Fakat çoğu zaman bu işlemi yapmak yerine bu gerçel sayısının üst sınırından gerçel sayının kendisi çıkartılarak bu şekilde tümleyeni alınmış olur.

Diğer bir kodlama yöntemi de karakter dizisinden oluşan kromozomların tanımlanmasıdır. Bu yöntemde ise bir karakterin en büyük ASCII değeri 255 olduğundan o karakterin değeri 255'ten çıkartılarak tümleyeni alınmış olur.

$R=c_n c_{n-1}...c_1$ karakter kodlamalı bir kromozom olmak üzere

$$\begin{aligned}\bar{R} &= \bar{c}_n \bar{c}_{n-1} \dots \bar{c}_1 \text{ ve } 1 \leq k \leq \phi \text{ için.} \\ \text{ord}(\bar{c}_k) &= \text{ord}(Z) - [\text{ord}(c_k) - \text{ord}(A)]\end{aligned}\quad (3.3)$$

olur ve $\text{ord}(\cdot)$ fonksiyonu kendisine parametre olarak gelen karakterin ASCII değerini hesaplar.

Populasyon biraz daha matematiksel olarak ifade edilmesi için $C_{\phi 0}$ gelişigüzel üretilen ϕ kromozom olsun ve $1 \leq \phi \leq (N/2^r)$. I endeks kümesi ve $I=\{i \mid i=0,1,...,2^r-1\}$. $C_{\phi i}$ kromozomu ise ϕ kromozomdan türetilen i . kromozomu temsil eder. Düzenli populasyonun parametresi r verildiğinde gelişigüzel üretilen her kromozom r tane parçaya bölünür ve endeks kümesinin elemanları r -bit ile ifade edilebilirler. I endeks kümesinin her elemanın en sağdaki biti en sağdaki parçayı, onun solundaki biti sağdan ikinci parçayı temsil eder ve bu şekilde devam edilerek en soldaki bit ise en soldaki parçayı temsil eder. $i \in I$ için i 'nin her biti temsil ettiği parçanın uzunluğu kadar genişletilir ve i_e ile gösterilir. Bu durumda gelişigüzel üretilen $C_{\phi 0}$ kromozomundan türetilen kromozomlar

$$C_{\phi i} = C_{\phi 0} \oplus i_e, \quad 0 \leq i \leq 2^r - 1$$

şeklinde olurlar. Başlangıç populasyonu P olmak üzere

$$P = \{C_{\phi i} \mid C_{\phi i} = C_{\phi 0} \oplus i_e, \quad 0 \leq i \leq 2^r - 1 \text{ ve } 1 \leq \phi \leq N/2^r\}$$

olur.

Gelişigüzel populasyonun sağlamış olduğu iki tane önemli özellik vardır. Bunlar populasyonun çeşitlilik içermiş olması ve nesiller arasında lineer bağımsızlığın olmasıdır. Üçüncü bölümde P populasyonun bu iki özelliği sağladığı detaylı olarak gösterildi. Bununla beraber bu populasyonun düzenlilik özelliğine de sahip olduğu ve uzaya düzgün bir şekilde dağıtıldığı aynı bölümde detaylı olarak gösterildi. Aynı zamanda bu populasyon uzayın medyanına göre simetrik özelliğine de sahiptir.

3.2. Yeni Mutasyon ve Çaprazlama Tanımları

Önerilen düzenli populasyon yönteminde kullanılan bir tane çaprazlama ve iki tane farklı mutasyon çeşidi bu tezde tanımlanmıştır.

Düzenli çaprazlama çeşidi genel olarak en etkili çaprazlama çeşididir. Fakat düzenli populasyonda düzenli çaprazlamanın çaprazlama noktası, düzenli populasyonda kromozom bölme noktası ile çakışırsa, çaprazlamanın etkili olma olasılığı, populasyon boyutu N olmak üzere

$$1 - \frac{\binom{2^r}{2}}{\binom{n}{2}} = 1 - \frac{2^r(2^r - 1)}{N(N-1)} \quad (3.4)$$

olur. Çaprazlama için iki tane kromozom seçilecektir. Bu kromozomların aynı kromozomdan türetilmiş kromozomlar olma sayısı $\binom{2^r}{2}$ olur ve populasyon içerisinde farklı iki kromozomu

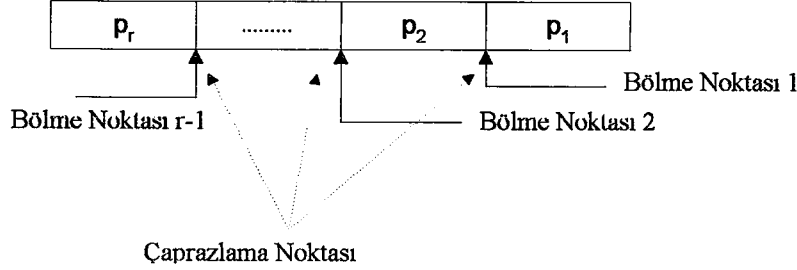
seçme sayısı da $\binom{N}{2}$ olur. Seçilen kromozomların türetilmiş veya türetilen kromozom olma

sayısının iki kromozomun bütün populasyon içerisinde seçilme sayısına bölünmesi ile elde edilen sayıya çaprazlama sonucunda elde edilecek bireylerin o anda var olan kromozomlarla aynı olma olasılığıdır. Bu durum mutasyonun olmadığı populasyonlar için geçerlidir, fakat iyi sonuç vermeyecek her türlü durumun elenmesi daha iyi sonuçların elde edilmesini sağlar. Populasyonda var olan kromozomlar tekrar üretilirse, bunun sonucunda populasyonun birbirine benzemesi belki de hızlanacaktır. Bu durumu ortadan kaldırmak için yoğun çaprazlama diye adlandırılan bir çaprazlama çeşidi bu tezde tanımlanmıştır.

Önerilen çaprazlama şeklinde eğer çaprazlama noktası bölme noktalarından biri ile çakışırsa, tekrar çaprazlama noktası gelişigüzel olarak belirlenmeye çalışılır. Bu şekilde yapılacak çaprazlamanın mümkünse, yeni bir birey oluşturmasını sağlamaktır. Bölme noktaları

belirlenen r değerine göre kromozomun parçalara bölündükten sonra parçaları birbirinden ayıran noktadır.

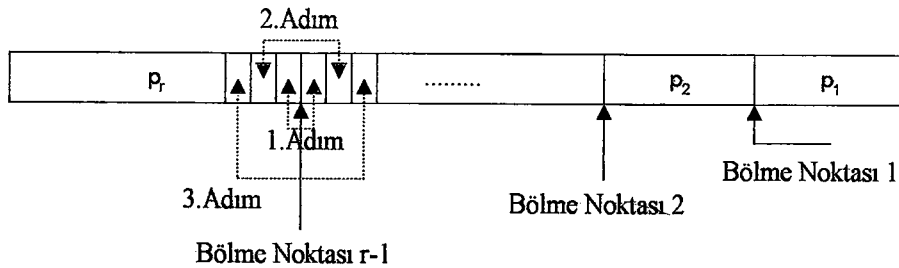
Şekil 3.4' te yoğun çaprazlama operatörünün şekli gösterilmiştir. Yoğun çaprazlama denilmesinin sebebi, kromozom içindeki bütün noktalar için değil de noktaların çoğunda çaprazlama yapıldığından dolayı bu isim verilmiştir.



Şekil 3.4. Yoğun çaprazlama.

Her zaman yoğun çaprazlamanın düzenli çaprazlamadan daha iyi olması mümkün değildir. Çünkü popülasyonun yapısı çaprazlama operatörlerin başarısını etkileyen faktörlerden biridir.

İşlemlerin yerel çözümde kalıp tıkanmasını engellemek için iki tane mutasyon operatörü tanımlanmıştır. Bunlardan birincisi Transpozisyon Mutasyondur. Bu mutasyonun çalışma şekli, en soldaki $(r-1)$ bölme noktasına göre simetrik olarak genlerin yer değiştirmesi ile elde edilen bir operatördür. Şekil 3.5' te transpozisyon mutasyonun şekli görülmektedir. İlk iterasyonda $(r-1)$ noktasına bitişik olan noktalar yer değiştirir (birinci adım). İkinci iterasyonda 2.Adım' da gösterilen genler yer değiştirir. Üçüncü iterasyonda 3.Adım' da gösterilen genler yer değiştirir. Bu şekilde işleme devam edilir. Ne zaman 1.Adım' da gösterilen genlere tekrar dönülür diye sorulacak olursa, iterasyon sayısının p_r parçasının uzunluğuna göre modu alınır ve sonuca 1 eklenerek yer değiştirecek genin $(r-1)$ noktasına göre yer değişimi elde edilir.



Şekil 3.5. Transpozisyon mutasyon.

Transpozisyon mutasyon operatörü, $(r-1)$ bölme noktasına göre simetrik çalışan bir mutasyon operatörüdür. Bir de sağ tarafta gelişigüzel bir genin seçilmesi durumu vardır. Bu mutasyon çeşidine de Antisimetrik Transpozisyon mutasyon operatörü adı verildi.

Bu iki operatörün tanımlanmasının sebebi, eğer işlem yerel çözümde kalmışsa, büyük sıçrama yaptırarak global aramada kolaylık sağlamaktır.

3.3. Düzenli Populasyonun Gelişigüzel Populasyona Göre Üstünlükleri

Uygulama sonuçları verilmeden önce gelişigüzel populasyonun avantajlı yönleri olduğu ve bu yönlerin düzenli populasyonda da var olduğunun gösterilmesi gerekiyor. Bu özellikler: populasyonun çeşitliliği, kromozomlar arasında lineer bağımsızlığın olması. Populasyonun çeşitliliğini gösterebilmek için o populasyonun varyansına bakılır ve bireyler arasında lineer bağımsızlığın olup olmadığını kontrol edebilmek için de nesiller arasında korelasyona bakılır.

Düzenli populasyon yönteminin olasılıksal analizinin yapılabilmesi için bazı terimlerin verilmesi gerekir.

N : Populasyondaki birey sayısı

n : bireyin bit uzunluğu

$P(i)$: i . bireyin oluşma olasılığı

μ_t : Populasyonun t anındaki beklenen değeri

σ_t : Populasyonun t anındaki standart sapma değeri

Olasılıksal hesaplamalara geçmeden önce düzenli populasyon yönteminde elde edilen başlangıç populasyonundaki bireylerden en az birinin çözümün olduğu bölgede olduğu düşünülecektir. Çünkü gelişigüzel oluşturulan bireyin tümleyeni alındığında, bu iki bireyden en az biri çözüm bölgesinde olur. Gelişigüzel üretilen bir birey dört parçaya bölünerek populasyon üretilecek olsun. Bu durumda gelişigüzel üretilen kromozom $C_0 = A_3 A_2 A_1 A_0$ olmak üzere

$$\begin{array}{llll} C_1 = A_3 A_2 A_1 \bar{A}_0 & C_2 = A_3 A_2 \bar{A}_1 A_0 & C_3 = A_3 A_2 \bar{A}_1 \bar{A}_0 & C_4 = A_3 \bar{A}_2 A_1 A_0 \\ C_5 = A_3 \bar{A}_2 A_1 \bar{A}_0 & C_6 = A_3 \bar{A}_2 \bar{A}_1 A_0 & C_7 = A_3 \bar{A}_2 \bar{A}_1 \bar{A}_0 & C_8 = \bar{A}_3 A_2 A_1 A_0 \\ C_9 = \bar{A}_3 A_2 A_1 \bar{A}_0 & C_{10} = \bar{A}_3 A_2 \bar{A}_1 A_0 & C_{11} = \bar{A}_3 A_2 \bar{A}_1 \bar{A}_0 & C_{12} = \bar{A}_3 \bar{A}_2 A_1 A_0 \\ C_{13} = \bar{A}_3 \bar{A}_2 A_1 \bar{A}_0 & C_{14} = \bar{A}_3 \bar{A}_2 \bar{A}_1 A_0 & C_{15} = \bar{A}_3 \bar{A}_2 \bar{A}_1 \bar{A}_0 & \end{array}$$

bireyleri elde edilir. Bir bireydeki gen sayısı k olmak üzere bu bireyler arasında

$$\begin{array}{l} C_0 + C_{15} = 2^{n-1} \\ C_1 + C_{14} = 2^{n-1} \\ C_2 + C_{13} = 2^{n-1} \\ C_3 + C_{12} = 2^{n-1} \end{array} \quad (3.5)$$

$$C_4 + C_{11} = 2^{n-1}$$

$$C_5 + C_{10} = 2^{n-1}$$

$$C_6 + C_9 = 2^{n-1}$$

ilişkisi vardır. Bu bilgilerden sonra üretilecek olan populasyonun olasılık hesapları aşağıdaki gibi verilebilir (düzenli dağılım özelliği).

$$P(C_0) = \frac{1}{2^n} \quad (3.6)$$

dır ve C_0 bireyi gelişigüzel üretildikten sonra $C_1, \dots, C_{15}$ bireylerinin ne oldukları artık bellidir. Bundan dolayı

$$P(C_i|C_0) = 1 \quad 1 \leq i \leq 15 \quad (3.7)$$

olur ve $P(C_i|C_0)$ olasılığı C_0 bireyin oluşturulma olayı olduktan sonra C_i bireyin oluşturulma olayının meydana gelme olasılığıdır (şartlı olasılık). Buradan,

$$P(C_i \cap C_0) = \frac{1}{2^n} \quad 1 \leq i \leq 15 \quad (3.8)$$

eşitliği elde edilir. Düzenli populasyon yönteminde bir bireyin gelişigüzel olarak üretilme olasılığının dağılımı düzenli bir yapıdadır. $P(C_i)$, $1 \leq i \leq 15$, olasılığını hesaplamak için C_0 bireyinin üretilme durumu göz önüne alındığı zaman $P(C_i)$ için alt sınır $P(C_i) \geq P(C_0)$ olur. Bunun anlamı

$$P(C_i, 0) \geq \frac{1}{2^n} \quad 1 \leq i \leq 15 \quad (3.9)$$

olur. Burada r parçaya bölme işlemi için $2^r - 1$ tane birey gelişigüzel üretilen bir bireyden üretilir. Bu durumda $1 \leq i \leq 15$ için $P(C_i)$ olasılık değeri $1/2^r$ olur. $C = \{C_0, C_1, \dots, C_{15}\}$ olmak üzere C kümesinin ortalama değeri (beklenen değeri - $\mu(C)$) hesaplanabilir.

$$\begin{aligned} \mu(C) &= \sum_{i=0}^{15} C_i \cdot P(C_i) \\ &\geq \frac{1}{2^r} ((C_0 + C_{15}) + (C_1 + C_{14}) + (C_2 + C_{13}) + (C_3 + C_{12}) + \\ &\quad (C_4 + C_{11}) + (C_5 + C_{10}) + (C_6 + C_9) + (C_7 + C_8)) \end{aligned} \quad (3.10)$$

$$= \frac{1}{2^r} \cdot 2^{r-1} \cdot (2^n)$$

$$= \frac{2^n}{2} = 2^{n-1}$$

olur. Bu değer sadece bir tane gelişigüzel üretilen birey için bulunmuştur. Gelişigüzel oluşturulan birey sayısı N tane eleman içeren bir populasyon için $N/2^r$ olur. Toplam populasyon için $\mu(C)$ değeri $N/2^r$ ile çarpılıp tekrar bu değere bölüneceği için aynı sonuç elde edilir. Buradan $\mu = \mu(C)$ olduğu görülmüş olur. (3.5) eşitliklerinden de yola çıkılarak populasyonun ortalama değerinin $\mu(C) = 2^{n-1}$ olacağı rahatlıkla görülebilir.

Burada amaç populasyonun düzenli dağılma özelliğine sahip olup olmadığının gösterilmesidir. Bunun için her bireyin gelme olasılığına $P(C_i)$ denilirse

$$P(C_i) \frac{n}{2} 2^n = \frac{2^n}{2} \Rightarrow P(C_i) = \frac{1}{n}$$

olacağından populasyon düzenli olma özelliğini kaybetmemektedir.

Bu populasyonun varyansı ve standart sapmasını hesaplamak için C kümesinden faydalanılacaktır. İlk olarak C kümesinin varyansının- $\sigma^2(C)$ - hesaplanması gerekir

$$\sigma^2(C) = \sum_{i=0}^{15} (C_i - \mu(C))^2 \quad (3.11)$$

$1 \leq i \leq 7$ ve $8 \leq j \leq 15$ için her farklı (i,j) çifti için (her i ve j değeri sadece bir kez kullanılarak) $C_i - \mu(C) = -(C_j - \mu(C))$ olduğundan (3.5 eşitliklerinde olduğu gibi)

$$\sigma^2(C) = 2 \sum_{i=0}^7 (C_i - \mu(C))^2 \quad (3.12)$$

olur. Burada farkların karesi alındığından, elde edilecek olan sonucun karekökü alınıp standart sapma olan $\sigma(C)$ değeri elde edildiğinde

$$\mu(C) > \sigma(C) \quad (3.13)$$

olur. Buradan değişim katsayısının

$$\frac{\sigma(C)}{\mu(C)} < 1 \quad (3.14)$$

olduğu görülür. Bunun anlamı hızlı yayılan bir topluluk olmadığıdır. Bu populasyonun düzgün olup olmadığı 3. momente bakılarak da anlaşılabilir.

$$\sigma^3(C) = \sum_{i=0}^{15} (C_i - \mu(C))^3 \quad (3.15)$$

olur ve $\sigma^3(C)$ değerini hesaplamak için $1 \leq i \leq 7$ ve $8 \leq j \leq 15$ ve her farklı (i,j) çifti için $(C_i - \mu(C)) = -(C_j - \mu(C))$ olduğundan (her i ve j değeri sadece bir kez kullanılarak)

$$\sigma^3(C) = 0 \quad (3.16)$$

olur. Buradan çarpıklık katsayısı da sıfır olarak bulunur. Bu elde edilen populasyonun beklenen değer etrafında düzgün olarak dağıldığını gösterir. Çünkü her bireyin oluşma olasılığı düzenli dağılım özelliğine sahip ve populasyonun çarpıklık katsayısı 0 çıkarken ortalama değeri o uzayın medyanına eşit olduğundan bu populasyon düzenli bir populasyondur. Bundan dolayı gelişigüzel populasyondan daha avantajlı bir yapıya sahiptir.

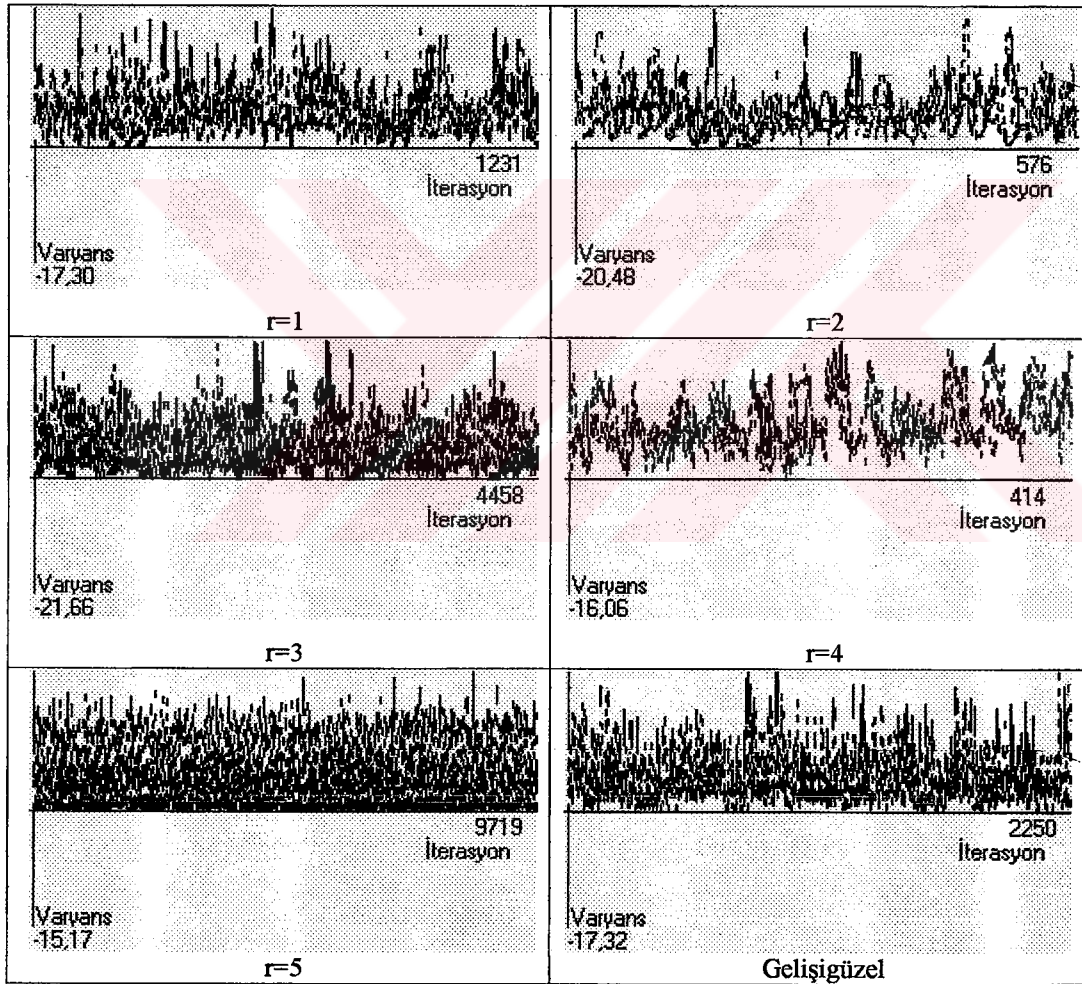
Gelişigüzel başlangıç populasyonu üretme yönteminde hiperuzayda bu kadar düzgün dağılmış bir başlangıç populasyonu elde etme olasılığı sıfıra yakındır. Bundan dolayı genetik arama çok zaman alırken ıraksama ve yerel çözümde kalması kaçınılmaz olur. Fakat düzenli populasyon yöntemi çok düzgün bir başlangıç populasyonu elde ettiğinden dolayı hem bu iki problemin ortadan kalkmasını sağlar ve hem de daha kısa zamanda çözüme ulaşılır.

Diğer iki önemli husus, populasyonun çeşitliliğine bakmaktır. Bunun anlamı: zaman içerisinde populasyonun çeşitliliğini kaybedip kaybetmediğini tespit etmektir. Bunun için yapılan uygulama sonucunda Şekil 3.6' da Griewank fonksiyonu için populasyonların değişimleri (varyansları) görülmektedir. Bu tezde her iki populasyonu karşılaştırmak için beş tane çok modelli fonksiyon kullanılmıştır ve bu fonksiyonların tanımları beşinci bölümde detaylı olarak verilmiştir. Bu şekilden de anlaşılacağı gibi gelişigüzel populasyonda gözlemlenen çeşitlilik olayı düzenli populasyonda da görülmektedir. Çünkü bütün r değerleri için elde edilen populasyonların, bu fonksiyon için elde edilen değişimleri ile gelişigüzel populasyonda yine aynı fonksiyon için elde edilen değişimleri arasında benzerlikler görülmektedir. Bu da göstermektedir ki, düzenli populasyon genetik populasyonun çeşitliliğini kaybetmemektedir.

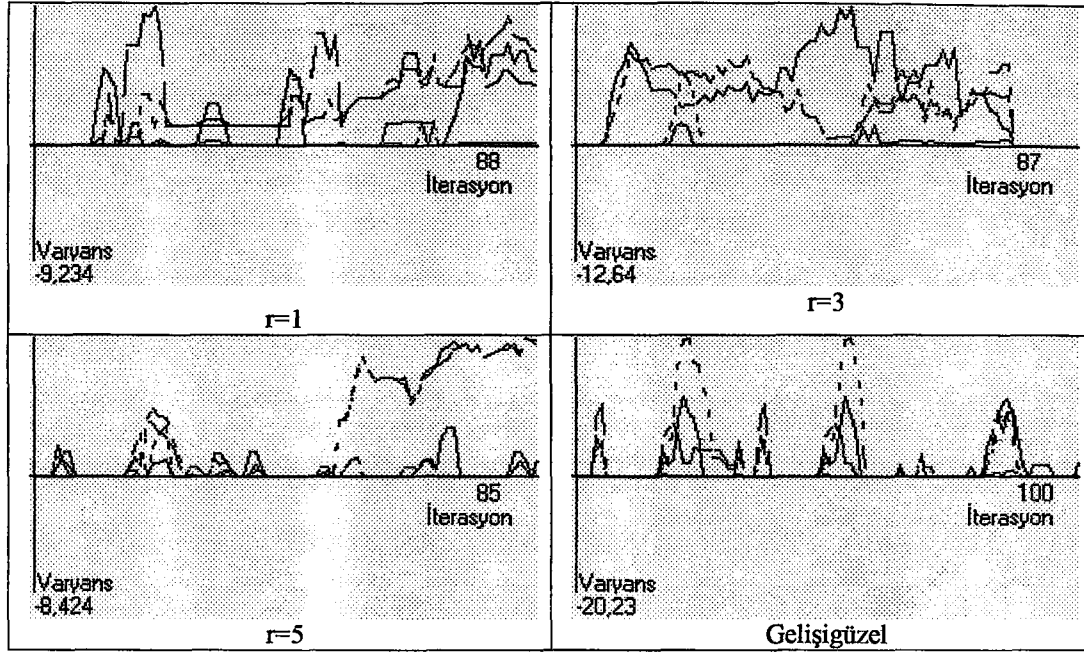
Bunun için yapılan diğer bir uygulama sonucunda Şekil 3.7' de Michalewicz fonksiyonu için populasyonların değişimleri (varyansları) görülmektedir. Bu şekilden de anlaşılacağı gibi gelişigüzel populasyonda gözlemlenen çeşitlilik olayı düzenli populasyonda da görülmektedir. Çünkü bazı r değerleri için elde edilen populasyonların bu fonksiyon için elde edilen değişimleri ile gelişigüzel populasyonda bu fonksiyon için elde edilen değişimleri arasında benzerlikler

görülmektedir. Bunun anlamı düzenli populasyon genetik populasyonun çeşitliliğini kaybetmemektedir. Şekil 3.8' de ise Rosenbrock fonksiyonu için populasyonların değişimleri verilmiştir ve burada görüleceği gibi populasyonların hepsinde çeşitlilik bulunmaktadır.

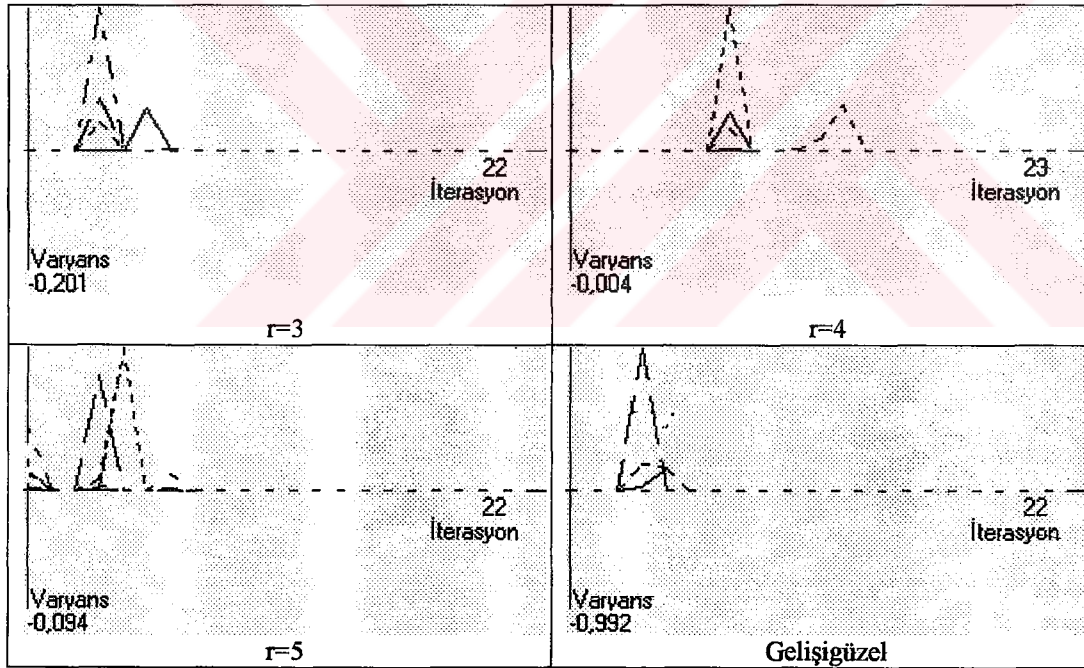
Şekil 3.8' de populasyon değişimi belli bir bölgede vardır ve ondan sonra bu değişim sıfıra düşmektedir. Bunun sebebi, bu fonksiyonun bir tane maksimum noktası bulunmaktadır. Belli iterasyon sayısından sonra populasyon birbirine benzemektedir. Bu durum her iki populasyon türü için de geçerlidir. Geriye kalan iki fonksiyon için de düzenli ve gelişigüzel populasyonların değişimlerine bakıldığında benzer özellikler taşımaktadırlar.



Şekil 3.6. Griewank fonksiyonu için populasyonların değişimleri.



Şekil 3.7. Michalewicz fonksiyonu için populasyonların değişimleri.



Şekil 3.8. Rosenbrock fonksiyonu için populasyonların değişimleri.

İkinci önemli özellik ise, bireyler arasında lineer bağımsızlığın olmasının sağlanmasıdır. Bunu görebilmek için iki nesil arasındaki korelasyona bakılır. Korelasyon değeri r değerinden bağımsız ve bütün r değerleri için sıfır çıkmaktadır. Aynı durum gelişigüzel populasyon için de geçerlidir, yani gelişigüzel populasyonda iki nesil arasındaki korelasyon sıfır çıkmaktadır. Şekil

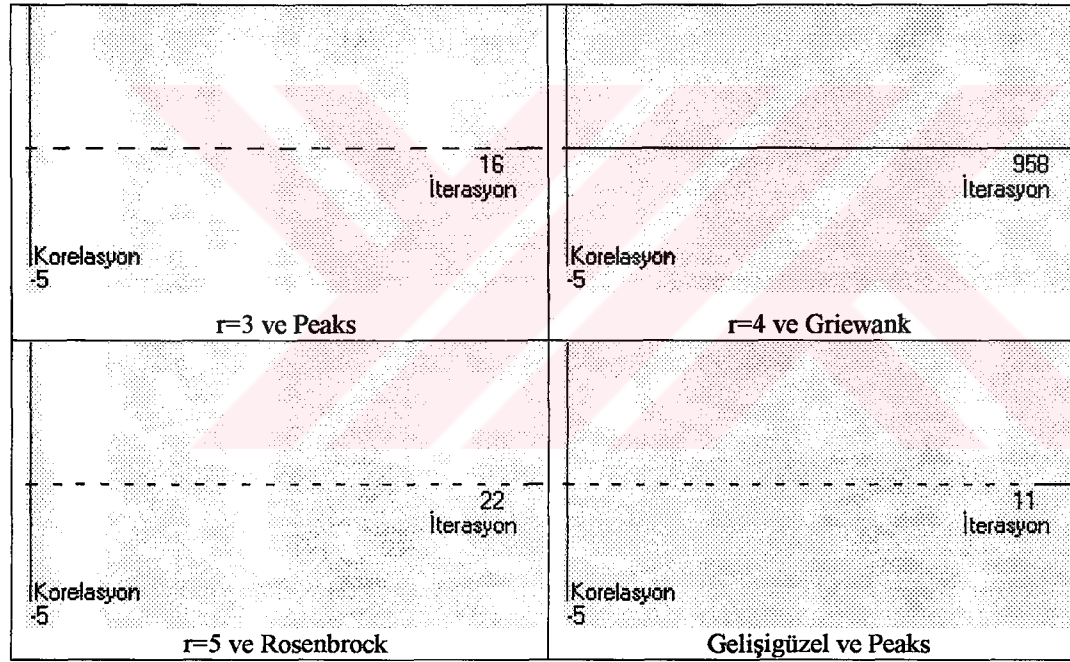
3.9' da deęişik r deęerleri iin dzenli populasyonun ve geliřigzel populasyonun korelasyon grafikleri grlmektedir.

Kısaca tekrar belirtilecek olursa, dzenli populasyon, geliřigzel populasyonun sahip olduęu eřitlilik ve nesiller arasındaki lineer baęımsızlık zellikleri yanı sıra aynı zamanda dzenlilik zelliięine de sahiptir. Bu zellikler, bu populasyonun getirmiş olduęu bir stnlktr.

Dzenli ve geliřigzel populasyonların zellikleri kısaca ařaęıdaki gibi zetlenebilir.

Geliřigzel Populasyon

- Bireyler arasında eřitlilięin olması.
- Nesiller arasında lineer baęımlılıęın olmaması.
- Populasyon yerel zmde kalma problemine sahiptir.



Şekil 3.9. populasyonların deęişik fonksiyonlar iin korelasyon grafikleri.

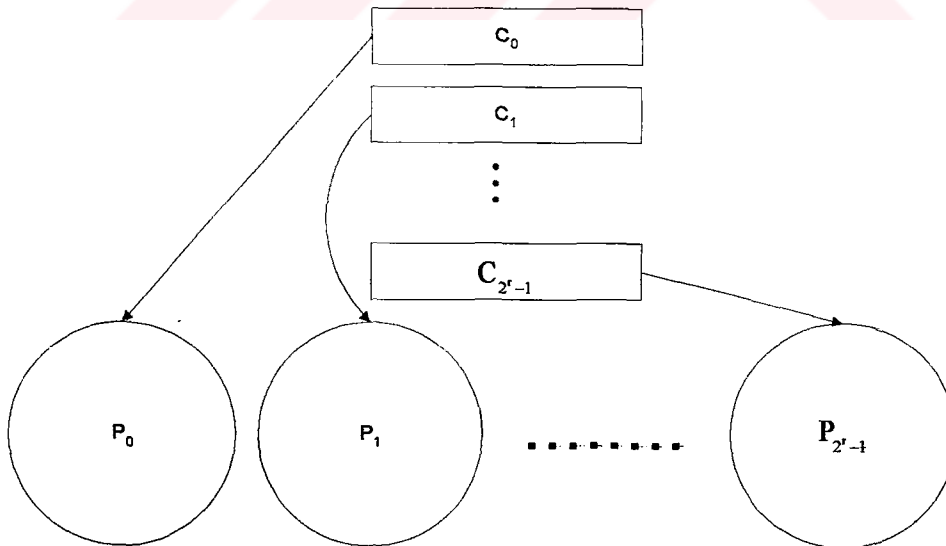
Dzenli Populasyon

- Bireyler arasında eřitlilięin olması.
- Nesiller arasında lineer baęımlılıęın olmaması.
- Populasyonun arpıklık katsayısı sıfır ve beklenen deęeri uzayın medyanına eřittir.
- Populasyon yerel zmde kalma problemine sahip deęildir.
- Populasyon en iyi zmden uzaklařmaz.

3.4. Adalı Düzenli Populasyon

Düzenli populasyonda gelişigüzel üretilen kromozom r değerine bağlı olarak parçalara ayrılıp daha sonra bütün kromozomun ve parçaların belli bir düzen içerisinde tümleyenleri alınarak yeni kromozomlar türetiliyordu. Adalı düzenli populasyonda ise, gelişigüzel üretilen kromozomlar bir adaya atılırken, bu kromozomların bütün tümleyenleri başka bir adaya atılır, en sağda kalan parçanın tümleyeni alınarak kromozomlar başka bir adaya atılır ve bu şekilde 2^r tane ada oluşur. Şekil 3.10' da gelişigüzel üretilen bir kromozomdan adalara hangi düzende türetilen kromozomlar ve gelişigüzel kromozomların adalara eklenişi görülmektedir.

Gelişigüzel üretilen kromozom C_0 olmak üzere $C_1, C_2, \dots, C_{2^r-1}$ tane kromozom C_0 kromozomundan türetilir. Bu kromozomlar Şekil 3.10' da görüldüğü gibi her kromozom kendisi ile aynı endekse sahip adaya yerleştirilir. Bu şekilde gelişigüzel üretilen bir kromozom durumu gösterilmiştir. Birden fazla gelişigüzel kromozom üretildiği zaman aynı durum söz konusudur. Dikkat edildiğinde görülecek bir gerçek vardır; o da eğer gelişigüzel üretilen bütün C_0 kromozomları birbirinden farklı ise adaların hepsinde bulunan kromozomlar birbirinden farklıdır. Eğer C_0 kromozomların hepsi aynı ise, bu durumda sadece her adadaki kromozomlar birbirinin aynı olur ve adalar arasında kromozom benzerliği olmayacaktır. Adalı düzenli populasyonun bu özelliği, gelişigüzel populasyondan daha iyidir. Çünkü bu adalı populasyonlar gelişigüzel olarak inşa edilecek olursa, populasyondaki bütün kromozomların aynı olma ihtimali vardır.

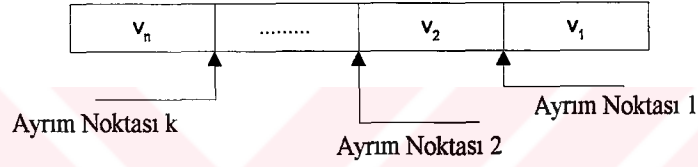


Şekil 3.10. Adalı düzenli populasyonda kromozomların yerleşimi.

Adalı popülasyonun diğör önemli bir özelliğı de paralel veya dağıtık mimariler üzerinde kullanılmaya çok müsait bir yapıya sahip olmasıdır. Daha önce vurgulandığı gibi adalar arasında başlangıçta kromozom benzerliğı olmadığından her makina farklı bir popülasyon ile aynı problemi çözecektir.

3.5. Önemli Nokta

Düzenli popülasyon yöntemi kullanılırken, dikkat edilmesi gereken önemli nokta eğer bir kromozomda birden fazla değişken varsa, o kromozomun değişken sayısı kadar parçaya ayrılması durumundaki r değeri nin hükmü sanki $r=1$ gibi etki eder. Bundan dolayı bu yöntemi kullanacak kişilerin bu noktayı her zaman hatırla tutmaları gerekmektedir (Şekil 3.11).



Şekil 3.11. Değişken ve ayırım noktalarının çakışması durumu.

4. ÇAPRAZLAMA VE MUTASYONUN DÜZENLİ POPULASYONA ETKİLERİ

Genetik algoritmalarda kodlama şekilleri daha önce belirtildiği gibi dört şekilde yapılmaktadır. Bu kodlama şekilleri ikili, tamsayı, gerçel sayı ve karakter kodlamalardır. Özellikle ikili kodlama kullanılıyorsa, nesiller arası geçişlerde genetik operatörlerin etkisi sonucunda nasıl bireylerin ortaya çıkacağı olasılıksal olarak daha kolay hesaplanabilmektedir.

İkili kodlamada her gen 0 veya 1 ile ifade edilir, yani kodlama alfabesi $A=\{0,1\}$ olur. Bir kromozom grubuna veya oluşabilecek kromozom grubuna şema denir ve şema örnekleri $101*...*$, $1101*..*$, $*01*..*1$ şeklinde verilebilir ve * bir genin 0 veya 1 olmasının önemli olmadığı durumlarda kullanılır. Örneğin, $001***$ şeması bir kromozom kümesini ifade eder ve bu kromozomlar

$$001***=\{001000, 001001, 001010, 001011, 001100, 001101, 001110, 001111\}$$

şeklinde dir. Bir şemada 0 veya 1 olan genlere tanımlayan genler denir ve bir şemada tanımlayan gen sayısına o şemanın mertebesi denir. En soldaki tanımlayan genin endeksinden en sağdaki tanımlayan genin endeksinin çıkarılması sonucunda elde edilen değere tanımlama uzaklığı denir. Bir şemada * olan gen sayısı o şemanın temsil ettiği uzayın boyutunu verir. Eğer bir şemada * olan gen sayısı 2 ise, bu şema hiperdüzlemi temsil etmektedir; eğer * olan gen sayısı 3 veya daha fazla ise, bu şema bir hiperuzayı temsil eder. Bu tez çalışmasında, genel olarak bu iki terim birbirinin yerine kullanılacaktır ve şemalar H ile gösterilecektir. H_t şeması t anında var olan bir şemadır.

Mutasyon ve çaprazlama operatörlerin etkisi iki şekilde analiz edilmektedir. Birincisi, bu çaprazlama operatörün şemanın yaşamı üzerine olan etkisidir ve ikincisi de mutasyon operatörünün şemanın yaşamı üzerine olan etkisidir. Operatörlerin bu etkilerinden yola çıkılarak t anındaki birey sayısı verildiğinde $t+1$ anındaki birey sayısı yaklaşık olarak hesaplanabilmektedir veya evlat sayısı için bir alt sınır hesabı yapılabilmektedir. Bunun için

$$\begin{aligned} m_t(H) &: t \text{ anında } H \text{ şeması içindeki birey sayısı} \\ f_t(H) &: t \text{ anında } H \text{ şemasının ortalama uygunluk değeri} \\ \bar{f}_t &: t \text{ anında populasyonun ortalama uygunluk değeri} \\ m_{t+1}(H) &: t+1 \text{ anında } H \text{ şeması içinde var olabilecek birey sayısı} \end{aligned}$$

olmak üzere

$$m_{t+1}(H) \geq m_t(H) \frac{f_t(H)}{\bar{f}_t} P_y(H) \quad (4.1)$$

olur (Spears, 1998; Goldberg, 1987; DeJong, 1975) ve $P_y(H)$ terimi H şemasının mutasyon veya çaprazlama sonucunda bozulmama olasılığıdır. Burada dikkat edilmesi gereken bir nokta vardır. Mutasyon ve çaprazlama ile H şemasının bozulmaması veya başka şemalardan H şemasının elde edilmesi gibi iki durumun olması eşitsizliğe sebep olur.

Genlerin değerini aldığı alfabe A olsun ve kromozom uzunluğu L olsun, $|A|=2$ olduğu biliniyor. H_k şeması en fazla 2^{L-k} tane farklı kromozom içerebilir. Örneğin, $H_4=10*1*0$ şemasında $2^{6-4}=2^2=4$ tane farklı kromozom vardır ve bu kromozomlar 100100, 100110, 101100, 101110 olur.

4.1. n-Noktalı Çaprazlamanın Etkisi

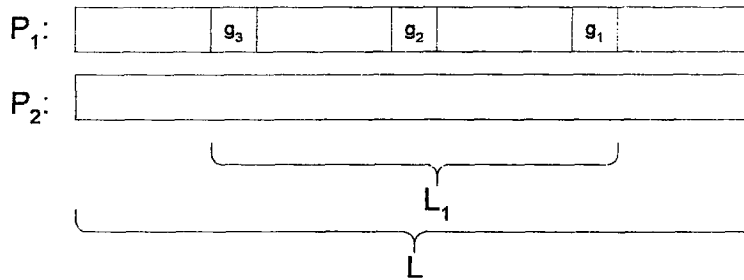
İlk olarak çaprazlama operatörünün etkisi incelenebilir. Çaprazlama işlemi U gelişigüzel değişken ile tasvir edilsin. H_k için 2^k tane farklı çaprazlama olayı gerçekleşebilir ve $\{u_0, u_1, \dots, u_{2^k-1}\}$ olur ve olasılığın tanımı gereği bütün olayların olasılıklarının toplamı

$$\sum_i P(U = u_i) = \sum_i P(u_i) = 1 \quad (4.2)$$

olur.

Çaprazlama için seçilen iki atadan biri gelişigüzel ve diğeri H_k şemasının içinde olsun. Seçilen atalar P_1 ve P_2 olmak üzere $P_1 \in H_k$ ve $P_2 \notin H_k$ veya $P_2 \in H_k$ için çaprazlama uygulandığında iki tane evlat oluşur (Şekil 4.1). Çaprazlama sonucunda oluşan evlatlardan en az biri H_k şeması içinde ise, H_k şeması yaşamına devam ediyor denir ve eğer hiçbir evlat bu şema içinde değilse, bu durumda H_k şeması bozulmuş denir. Bu şemanın yaşamına devam etme olasılığı P_y ve bozulma olasılığı P_b olsun. Açıkça

$$\begin{aligned} P_y(H_k) &= \sum_i P(U = u_i) P(H_k \text{ yaş.} | u_i) \\ &= \sum_i P(u_i) P_y(H_k | u_i) \end{aligned} \quad (4.3)$$



Şekil 4.1. H_3 şemasında tanımlayan genler : g_1, g_2, g_3 (L kromozom uzunluğu ve L_1 tanımlama uzunluğu).

ve

$$\begin{aligned} P_b(H_k) &= \sum_i P(U = u_i)P(H_k \text{ bozulma} | u_i) \\ &= \sum_i P(u_i)P_b(H_k | u_i) \end{aligned} \quad (4.4)$$

olduğu görülebilir (DeJong, 1975; Goldberg, 1987; Spears, 1998). Çaprazlama işleminden sonra H_k şeması yaşamına devam edebilir veya bozulabilir. Bundan dolayı

$$P_y(H_k) + P_b(H_k) = 1 \quad (4.5)$$

olur.

Bu bölümde n-noktalı çaprazlamanın ve düzenli çaprazlamanın etkileri incelenecektir. n-noktalı çaprazlamada, n tane kesme noktası çaprazlama olaylarını belirler. Bu olaylar çift ve tek olaylar olarak ikiye ayrılırlar. Çift olayın oluşması için tanımlayan bitişik iki gen arasına çift sayıda kesme noktasının denk gelmesi durumudur. Bu durumda H_k yaşamına devam eder, çünkü tanımlayan genler yer değiştirmezler. Şekil 4.2 (a)' da bu durum görülmektedir. Aksine tek olayın anlamı, bitişik tanımlayan genler arasına tek sayıda kesme noktasının düşmesi durumudur. Tek olay durumu Şekil 4.2(b)' de görülmektedir. Her iki durumda da L kromozom uzunluğunu gösterirken, L_1 uzunluğu tanımlama uzaklığını göstermektedir. Tek olayın oluşması durumunda tanımlayan genler yer değiştirir ve bu durumda da

$$P_{\text{çift}}(H_k) = \sum_{u_i \text{ çift}} P(u_i) \quad (4.6)$$

$$P_{\text{tek}}(H_k) = \sum_{u_i \text{ tek}} P(u_i) \quad (4.7)$$

bağıntıları elde edilir.

Şekil 4.2' de çift ve tek olayların H_3 üzerindeki etkileri görülmektedir. Bundan dolayı

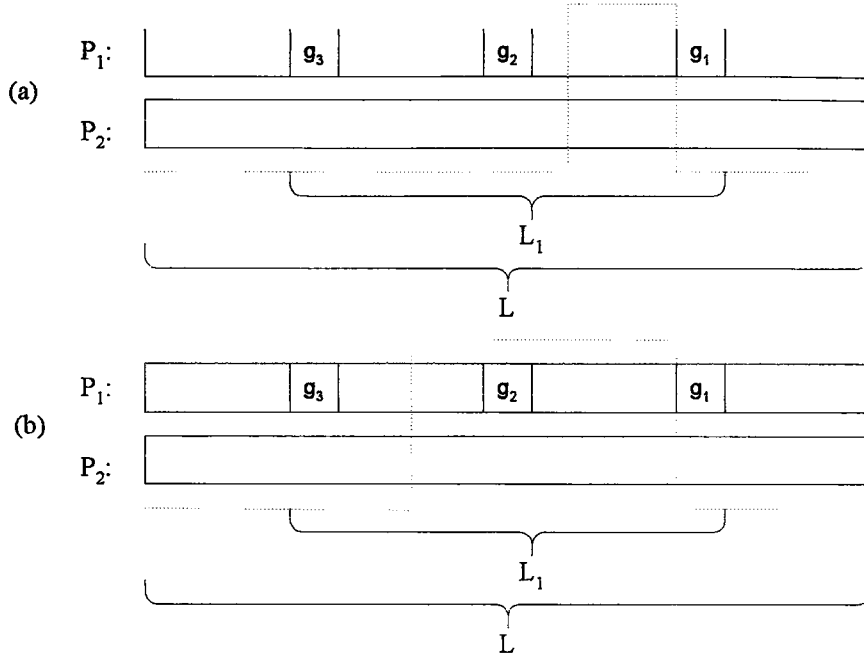
$$P_{\text{çift}}(H_k) + P_{\text{tek}}(H_k) = 1 \quad (4.8)$$

elde edilir. Buradan

$$P_b(H_k) = \sum_{u_i \text{ çift}} P(u_i)P_b(H_k | u_i) + \sum_{u_i \text{ tek}} P(u_i)P_b(H_k | u_i) \quad (4.9)$$

ve

$$P_y(H_k) = \sum_{u_i \text{ çift}} P(u_i)P_y(H_k | u_i) + \sum_{u_i \text{ tek}} P(u_i)P_y(H_k | u_i) \quad (4.10)$$



Şekil 4.2. (a) bitişik tanımlayan genler arasında çift sayıda kesme noktasının düşmesi;
(b) bitişik tanımlayan genler arasında tek sayıda kesme noktasının düşmesi.

bağıntıları elde edilir. Çift olay olduğu zaman

$$P_b(H_k|u_i)=0, \quad \forall u_i \text{ çift}$$

ve

$$P_y(H_k|u_i)=1, \quad \forall u_i \text{ çift}$$

olur. Bu bilgilerin kullanımı ile denklem 4.9 ve denklem 4.10

$$P_b(H_k) = \sum_{u_i \text{ tek}} P(u_i) P_b(H_k | u_i) \quad .$$

ve

$$P_y(H_k) = \sum_{u_i \text{ çift}} P(u_i) + \sum_{u_i \text{ tek}} P(u_i) P_y(H_k | u_i)$$

şeklini alırlar. Yaşam ve bozulma olasılıkları

$$\begin{aligned} P_b(H_k) &\leq \sum_{u_i \text{ tek}} P(u_i) \\ &= 1 - \sum_{u_i \text{ çift}} P(u_i) \\ &= 1 - P_{\text{çift}}(H_k) \end{aligned} \tag{4.11}$$

ve

$$P_y(H_k) \geq \sum_{u_i \text{ çift}} P(u_i) = P_{\text{çift}}(H_k) \quad (4.12)$$

olurlar. $P_{\text{çift}}(H_k)$, yaşam olasılığı için alt sınırı teşkil ederken bozulma olasılığı içinde $1 - P_{\text{çift}}(H_k)$ üst sınırı teşkil etmektedir. Şu ana kadar verilen bilgiler gelişigüzel populasyon için geçerli olan bilgilerdi. Bu bilgiler düzenli populasyon için de geçerli olmasına rağmen bazı özel durumlar karşımıza çıkabilmektedir. Bunun için çaprazlama operatörünün bu etkileri yanında bir de düzenli popülasyona has olan etkilerinin verilmesi gerekir. Bunun için popülasyonun yapısının daha rahat anlaşılması için $r=4$ ve K_0 gelişigüzel üretilen bir kromozomdan

$$\begin{aligned} K_0 &= C_3 C_2 C_1 C_0 & K_1 &= C_3 C_2 C_1 \bar{C}_0 \\ K_2 &= C_3 C_2 \bar{C}_1 C_0 & K_3 &= C_3 C_2 \bar{C}_1 \bar{C}_0 \\ K_4 &= C_3 \bar{C}_2 C_1 C_0 & K_5 &= C_3 \bar{C}_2 C_1 \bar{C}_0 \\ K_6 &= C_3 \bar{C}_2 \bar{C}_1 C_0 & K_7 &= C_3 \bar{C}_2 \bar{C}_1 \bar{C}_0 \\ K_8 &= \bar{C}_3 C_2 C_1 C_0 & K_9 &= \bar{C}_3 C_2 C_1 \bar{C}_0 \\ K_{10} &= \bar{C}_3 C_2 \bar{C}_1 C_0 & K_{11} &= \bar{C}_3 C_2 \bar{C}_1 \bar{C}_0 \\ K_{12} &= \bar{C}_3 \bar{C}_2 C_1 C_0 & K_{13} &= \bar{C}_3 \bar{C}_2 C_1 \bar{C}_0 \\ K_{14} &= \bar{C}_3 \bar{C}_2 \bar{C}_1 C_0 & K_{15} &= \bar{C}_3 \bar{C}_2 \bar{C}_1 \bar{C}_0 \end{aligned} \quad (4.13)$$

kromozomları türetilir. Çaprazlama için iki tane kromozom seçilir ve bu kromozomların bir gelişigüzel üretilmiş kromozomdan türetilmiş kromozomlar grubundan seçilme olasılıkları r bir kromozomdaki parça sayısı ve N de popülasyon boyutu olmak üzere

$$P(\text{ata seçme}) = \frac{\binom{2^r}{2}}{\binom{N}{2}} = \frac{2^r (2^r - 1)}{N(N - 1)} \quad (4.14)$$

olur. $r=1$ olduğu durumda eğer gelişigüzel üretilen kromozomların hepsi birbirinden farklı ve tanımlayan genlerden en az bir tanesi yer değiştiriyorsa, $P_y(u_i)=0$ olur. Bu durumda $P_b(u_i)=1$ olur. Eğer gelişigüzel üretilen kromozomlardan en az iki tanesi aynı ise bu durumda $P_y(u_i)=0$ olmaz ve

$$P_y(u_i) \geq \frac{6}{N(N - 1)} \quad (4.15)$$

olur ve bozulma olasılığı ise

$$P_b(u_i) \leq 1 - \frac{6}{N(N-1)} \quad (4.16)$$

olur. Eğer $r > 1$ ve her iki ata aynı kromozomdan türetilen kromozomlar grubundan seçilmiş ve bir parçaya düşen kesme nokta sayısı çift ise

$$P_y(u_i) = \frac{\frac{1}{r} \cdot \frac{1}{2} \cdot \binom{2^r}{2}}{\binom{N}{2}} = \frac{2^r(2^r - 1)}{rN(N-1)} \quad (4.17)$$

olur. Eğer bir parçaya düşen kesme nokta sayısı tek ise,

$$P_y(u_i) \geq \frac{\binom{2^{r-2}}{2} \cdot \frac{1}{2}}{\binom{N}{2}} = \frac{2^{r-2}(2^{r-2} - 1)}{2N(N-1)} \quad (4.18)$$

olur. Buradan bozulma olasılığı

$$P_b(u_i) \leq 1 - \frac{2^r(2^r - 1)}{rN(N-1)} - \frac{2^{r-2}(2^{r-2} - 1)}{2N(N-1)} \quad (4.19)$$

olur. Eğer seçilen iki ata aynı kromozomdan türetilmiş kromozom grubundan değillerse, gelişigüzel populusyona olan çaprazlama etkileri bu populusyon için de geçerli olur. Gelişigüzel populusyonda çaprazlamanın etkisinin hesaplanabilmesi için $P_{çift}(H_k)$ değerinin bilinmesi gerekir (Spears, 1998; DeJong, 1975). $k=2$ için

$$P_{çift}(H_2, L, L_1, n) = \sum_{j=0}^{\lfloor n/2 \rfloor} \binom{n}{2j} \left(\frac{L_1}{L} \right)^{2j} \left(\frac{L-L_1}{L} \right)^{n-2j} \quad (4.20)$$

bağıntısı elde edilir. L_1/L oranı bir kesme noktasının iki tane tanımlayan gen arasına denk gelme olasılığıdır ve $(L-L_1)/L$ ise denk gelmeme olasılığıdır. Denklem 4.7 genelleştirildiğinde ve $k=3$ için aşağıdaki tekrarlı bağıntı elde edilir (Spears, 1998).

$$P_{çift}(H_3, L, L_1, L_2, n) = \sum_{j=0}^{\lfloor n/2 \rfloor} \binom{n}{2j} \left(\frac{L_1}{L} \right)^{2j} \left(\frac{L-L_1}{L} \right)^{n-2j} P_{çift}(H_2, L_1, L_2, 2j) \quad (4.21)$$

Daha yüksek mertebeli şemalar içinde benzer tekrarlı bağıntılar elde edilebilir (Spears, 1998) ve bu bağıntı (4.22)'de görülmektedir.

$$P_{\text{çift}}(H_k, L, L_1, \dots, L_{k-1}, n) = \sum_{j=0}^{\lfloor n/2 \rfloor} \binom{n}{2j} \left(\frac{L_1}{L} \right)^{2j} \left(\frac{L-L_1}{L} \right)^{n-2j} P_{\text{çift}}(H_{k-1}, L_1, \dots, L_{k-1}, 2j) \quad (4.22)$$

Bitişik olan iki tanımlayan gen arasında tek sayıda kesme noktası olmasına rağmen eğer her iki kromozomda yer değiştiren tanımlayan genler aynı ise, şema bozulmaz ve yaşamına devam eder. Bu popülasyonun homojenliğinin bir sonucudur. $P_{\text{çift}}(H_k)$ olasılığı bir şemanın yaşamını sürdürmesi için bir alt sınır teşkil etmektedir; bundan dolayı $P_{\text{çift}}(H_k)$ olasılığı $P_y(H_k)$ olasılığı için bir alt sınırdır. Bitişik olan tanımlayan genler arasında sadece çift sayıda kesme noktası olması durumunun aksine, tek sayıda kesme noktasının olması durumlarının da göz önüne alınması gerekir, çünkü popülasyonun homojenliği şemanın yaşama şansını etkileyen bir faktördür. Denklem 4.3'ün hesaplanması için bütün durumların göz önüne alınması gerekir. n-noktalı çaprazlamada H_2 şemasının yaşama olasılığı

$$P_y(H_2, L, L_1, n) = \sum_{j=0}^n \binom{n}{j} \left(\frac{L_1}{L} \right)^j \left(\frac{L-L_1}{L} \right)^{n-j} P_y(H_2 | u_i) \quad (4.23)$$

olur. Böylece bütün durumların göz önüne alınmasından sonra H_k için tekrarlı bağıntı denklem 4.24'deki gibi olur.

$$P_y(H_k, L, L_1, \dots, L_{k-1}, n) = \sum_{j=0}^n \binom{n}{j} \left(\frac{L_1}{L} \right)^j \left(\frac{L-L_1}{L} \right)^{n-j} P_y(H_{k-1}, L_1, \dots, L_{k-1}, j) \quad (4.24)$$

(4.24) tekrarlı bağıntısı mertebesi küçük olan şemaya doğru ilerletilince en sonunda (4.23) bağıntısına ulaşır. Buradan $P_y(H_k | u_i)$ değerinin hesaplanabilmesi önemlidir; çünkü bunu hesaplamak hem çaprazlama olaylarına bağlı ve hem de popülasyonun homojenliğine bağlıdır. Gelişigüzel popülasyonlarda, popülasyonun homojenliğinden yola çıkarak hesaplamanın yapılması için K tanımlayan genler kümesi olmak üzere X genin değiştiği kabul edilsin. Çaprazlamadan sonra şemanın yaşama şansı iki duruma bağlıdır.

- İki atanın X genlerinin aynı olmasıdır.
- İki atanın $K-X$ genlerinin aynı olmasıdır.

Bu durumda

$$P_y(H_k|u_i)=P_{eş}(X)+P_{eş}(K-X)-P_{eş}(K) \quad (4.25)$$

olur. $P_{eş}(X)$ olasılığı her iki atanın X genlerinin aynı olma olasılığıdır.

Düzenli populasyonun gelişigüzel popülasyondan olan farkından yola çıkarak $P_{eş}(X)$ hesaplanabilir ve bunun bağıntısı (4.26)' da görülmektedir.

$$P_{eş}(X) = \frac{1}{2} \cdot \frac{\binom{2^r}{2}}{\binom{N}{2}} = \frac{2^r(2^r-1)}{2N(N-1)} \quad (4.26)$$

Bu durum mutasyon operatörünün hiç kullanılmadığı durumlar için geçerlidir. Genlerin bağımsız olduğu kabul edilirse (gelişigüzel popülasyonlarda), (4.25) bağıntısı

$$P_y(H_k | u_i) = \prod_{d \in X} P_{eş}(d) + \prod_{d \in (K-X)} P_{eş}(d) - \prod_{d \in K} P_{eş}(d) \quad (4.27)$$

şeklini alır. Bu bağıntı

$$P_y(H_k | u_i) = P_{eş}^x + P_{eş}^{k-x} - P_{eş}^k$$

şeklinde sadeleştirilebilir. İkili kodlama için $P_{eş} \geq 1/2$ olur. Fakat (4.26) bağıntısına bakıldığında zaman ikili kodlama kullanıldığı halde, sanki kodlama alfabesinin boyutu 2^r den daha büyümüş gibi $P_{eş} \leq 1/2$ olur. Bundan dolayı da düzenli popülasyonun çeşitliliği gelişigüzel popülasyona göre daha fazla sağladığı açık bir şekilde görülmektedir.

4.2. Düzenli Çaprazlamanın Etkisi

Düzenli çaprazlamanın popülasyon üzerine olan etkisinin incelenmesi için P_0 olasılığı tanımlayan genlerin yer değiştirme olasılığı olsun. Düzenli çaprazlamada çift çaprazlama olayı H_k şemaların k tane tanımlayan genlerinin yer değiştirmesi veya hiçbir tanımlayan genin yer değiştirmemesi durumunda oluşur. Bu durumda

$$P_{\text{çift}}(H_k, P_0) = P_0^k + (1 - P_0)^k \quad (4.28)$$

elde edilir. K kümesi k tanımlayan genlerin kümesi olsun. Denklem 4.28 daha genel olarak

$$P_y(H_k, P_0) = \sum_{X \subseteq K} P_0^{|X|} (1 - P_0)^{|K|-|X|} P_y(H_k | u_i) \quad (4.29)$$

şeklinde yazılır. Burada önemli olan $P_y(H_k | u_i)$ olasılığının hesaplanmasıdır, çünkü bu değer hesaplandıktan sonra eşitliğin değerinin hesaplanması kolaylaşacaktır. $P_y(H_k | u_i)$ olasılığının hesaplanması için populasyonun homojenliğinden faydalanılır ve

$$P_y(H_k, P_0) = \sum_{j=0}^k \binom{k}{j} P_0^j (1 - P_0)^{k-j} (P_{eş}^j + P_{eş}^{k-j} - P_{eş}^k) \quad (4.30)$$

elde edilir.

Düzenli populasyon göz önüne alındığında çaprazlama sonucunda şemanın bozulmaması için çaprazlamanın parça tabanlı olması gerekiyor. Eğer yer değişim parça bazlı ise,

$$P_y(H_k, P_0) = \frac{\frac{1}{2} \binom{2^r}{2}}{\binom{N}{2}} = \frac{2^r (2^r - 1)}{2N(N - 1)} \quad (4.31)$$

olur. Bu populasyonun homojenliği hesaba katılınca, Denklem 4.31' de sol tarafta elde edilecek değer bu olasılık hesabı için bir alt sınırı teşkil edecektir. Eğer şemanın bozulması olasılığı hesaplanmak istenirse, denklem 4.31' de verilen değer birden çıkarıldığında elde edilen değer şemanın bozulma olasılığının üst sınırını teşkil edecektir. Bu durumda

$$P_b(H_k, P_0) \leq 1 - \frac{2^r (2^r - 1)}{2N(N - 1)} \quad (4.32)$$

bağıntısı elde edilir. Eğer yer değiştiren tanımlayan genler parça tabanlı değilse, bu durumda populasyonun homojenliği şemanın yaşama durumunu belirler. Yer değiştirecek genler aynı değilse, şema bozulur, aksi halde şema yaşamına devam eder.

4.3. Çaprazlama ile Şema İnşası

Bu bölümde incelenmek istenen durum şudur: H_m ve H_q şemalarından H_k şemasının inşa edilmesidir. İlk olarak H_m ve H_q şemalarının tanımlayan genlerinin ortak olmadığı kabul edilsin ve $k=m+q$ olur.

Bu durumlar S gelişigüzel değişkeni ile ifade edilsinler. H_k şeması için 2^k tane olay vardır ve bu olaylar $\{s_0, s_1, \dots, s_{2^k-1}\}$ olurlar. $P_{in}(H_k|s_j)$ terimi s_j olayının oluşmasından sonra H_k şemasının inşa edilme olasılığını gösterebilir.

$$P_{in}(H_k|s_j) = P_{in}(H_k|H_m \wedge H_q)$$

Bağıntısı H_m ve H_q şemalarından H_k şemasının inşa edilmesi olasılığı olsun. İlk olarak n -noktalı çaprazlama sonucunda H_k şemasının inşa edilmesi durumu incelenebilir. n -noktalı çaprazlamaya göre H_k şemasının inşa edilmesi olasılığı

$$P_{in}(H_k, L, L_1, \dots, L_{k-1}, n | s_j) = \sum_{x=0}^n \binom{n}{x} \left(\frac{L_1}{L} \right)^x \left(\frac{L-L_1}{L} \right)^{n-x} P_{in}(H_{k-1}, L_1, \dots, L_{k-1}, x | s_j) \quad (4.33)$$

bağıntısı ile ifade edilir ve $k=2$ için (4.34) bağıntısı elde edilebilir.

$$P_{in}(H_2, L, L_1, n | s_j) = \sum_{x=0}^n \binom{n}{x} \left(\frac{L_1}{L} \right)^x \left(\frac{L-L_1}{L} \right)^{n-x} P_{in}(H_k | s_j \wedge u_i) \quad (4.34)$$

Burada önemli olan $P_{in}(H_k|s_j \wedge u_i)$ değerinin hesaplanmasıdır. Çaprazlamada kullanılan (4.25) bağıntısı yerine

$$P_{in}(H_k | s_j \wedge u_i) = P_{eş}(X) + P_{eş}(K - X) - P_{eş}(K)$$

bağıntısı kullanılır. Tanımlayan genlerin bağımsız olduğu kabul edilirse

$$P_{in}(H_k | s_j \wedge u_i) = \prod_{d \in X} P_{eş}(d) + \prod_{d \in (K-X)} P_{eş}(d) - \prod_{d \in K} P_{eş}(K) \quad (4.35)$$

bağıntısı elde edilir. Bu bağıntı daha da sade yazılabilir ve

$$P_{in}(H_k | s_j \wedge u_i) = P_{eş}^x + P_{eş}^{k-x} - P_{eş}^k \quad (4.36)$$

şeklinde olur. Düzenli populasyon düşünüldüğünde

$$P_{in}(H_k | s_j \wedge u_i) = 1 - \frac{1}{2} \cdot \frac{\binom{2^r}{2}}{\binom{N}{2}} = 1 - \frac{2^r(2^r - 1)}{2N(N - 1)} \quad (4.37)$$

bağıntısı elde edilir.

Düzenli çaprazlamanın şema inşası üzerindeki etkisi şu şekilde açıklanabilir. Düzenli çaprazlama altında bir şemanın yaşamına devam etme olasılığı

$$P_y(H_k, P_0) = \sum_{X \subseteq K} P_0^{|X|} (1 - P_0)^{|K - X|} (P_{eş}(X) + P_{eş}(K - X) - P_{eş}(K))$$

şeklindedir. Bu bağıntı üç parçaya bölünebilir. İlk parça H_k şemasının orijinal karakter katarı içinde yaşamını devam etme olasılığıdır ve bu olasılık

$$P_{y,orj}(H_k, P_0) = \sum_{X \subseteq K} P_0^{|X|} (1 - P_0)^{|K - X|} P_{eş}(X) \quad (4.38)$$

şeklindedir. İkinci parçada H_k şemasının diğer karakter dizisi içinde yaşamını devam ettirme olasılığıdır ve bu olasılık

$$P_{y,dig}(H_k, P_0) = \sum_{X \subseteq K} P_0^{|X|} (1 - P_0)^{|K - X|} P_{eş}(K - X) \quad (4.39)$$

şeklindedir. Son parça ise H_k şemasının her iki karakter dizisinde olma olasılığıdır ve bu olasılık

$$P_{y,iki}(H_k, P_0) = \sum_{X \subseteq K} P_0^{|X|} (1 - P_0)^{|K - X|} P_{eş}(K) = P_{eş}(K) \quad (4.40)$$

şeklindedir. Buradan şemanın yaşama olasılığı

$$P_y(H_k, P_0) = P_{y,orj}(H_k, P_0) + P_{y,dig}(H_k, P_0) + P_{y,iki}(H_k, P_0) \quad (4.41)$$

şeklinde olur. Bütün bu bilgiler ışığında bir şemanın inşa edilme olasılığı rahatlıkla hesaplanabilir. Bir şemanın inşa edilme olasılığı

$$P_{in}(H_k, P_0 | s_j) = P_{y,orj}(H_m, P_0) P_{y,dig}(H_q, P_0) + P_{y,dig}(H_m, P_0) P_{y,orj}(H_q, P_0) - P_{y,iki}(H_m, P_0) P_{y,iki}(H_q, P_0) \quad (4.42)$$

olur. Elde edilen bağıntının iki tane yaşama olayının ayrıştırılmış şekli olduğu rahatlıkla görülebilir. Tanımlayan genler arasında bağımsızlık olduğu kabul edilirse, son terim

$$P_{eş}(M)P_{eş}(Q) = \prod_{d \in M} P_{eş}(d) \prod_{d \in Q} P_{eş}(d) = \prod_{d \in K} P_{eş}(d) \quad (4.43)$$

şeklini alır. (4.38), (4.39) ve (4.40) eşitliklerinde $P_{eş}(d)$ yerine $P_{eş}$ yazıldığında

$$P_{y,orj}(H_k, P_0) = \sum_{x=0}^k \binom{k}{x} P_0^x (1-P_0)^{k-x} P_{eş}^x$$

$$P_{y,dig}(H_k, P_0) = \sum_{x=0}^k \binom{k}{x} P_0^x (1-P_0)^{k-x} P_{eş}^{k-x}$$

$$P_{y,iki}(H_k, P_0) = \sum_{x=0}^k \binom{k}{x} P_0^x (1-P_0)^{k-x} P_{eş}^k$$

şeklini alırlar. Şema inşa etmede gelişigüzel populasyon için elde edilen olasılık bağıntılarında eşitlik varken, düzenli populasyon için bu eşitlikler eşitsizlik durumuna gelecektir. Çünkü düzenli populasyonda hiçbir zaman populasyonun homojenliği yüzde-yüz olmaz. Fakat gelişigüzel populasyonda bu durumun oluşma olasılığı sıfır değildir. Populasyon homojenliğinin yüzde-yüz olmama durumu, populasyon çözüme giderken ki durum değildir. İşlemlerin başlangıcı veya çeşitlilik korunduğu sürece geçerlidir.

4.4. Mutasyon Operatörünün Etkisi

Düzenli populasyonda hatırlanacağı gibi örneğin $r=1$ için gelişigüzel üretilen kromozom $K_0 = c_L c_{L-1} \dots c_1$ ise başlangıç populasyonu içerisinde $\bar{K}_0 = \bar{c}_L \bar{c}_{L-1} \dots \bar{c}_1$ kromozomu da olacaktır. Populasyonun genel karakteristiği bu şekilde devam edecektir. Bu durumda H_k şemasının mutasyon operatörü uygulandıktan sonra yaşamına devam etme ve bozulma olasılıkları

$$P_y(H_k) \geq 1 - \mu \frac{k}{L} \quad \text{ve} \quad P_b(H_k) \leq \mu \frac{k}{L} \quad (4.44)$$

şeklinde olur ve μ parametresi bir genin mutasyona uğrama olasılığıdır. $r=2$ olduğu durumda $K_0 = c_L c_{L-1} \dots c_1$ gelişigüzel üretilen bir kromozom ise başlangıç populasyonunda $K_1 = c_L c_{L-1} \dots c_{\lfloor L/2 \rfloor + 1} \bar{c}_{\lfloor L/2 \rfloor} \dots \bar{c}_1$, $K_2 = \bar{c}_L \bar{c}_{L-1} \dots \bar{c}_{\lfloor L/2 \rfloor + 1} c_{\lfloor L/2 \rfloor} \dots c_1$ ve $K_3 = \bar{c}_L \bar{c}_{L-1} \dots \bar{c}_{\lfloor L/2 \rfloor + 1} \bar{c}_{\lfloor L/2 \rfloor} \dots \bar{c}_1$ kromozomları da olacaktır. Bu durumda mutasyon operatörünün H_k şeması üzerindeki etkisi (4.44) denklemindeki eşitsizlik şartını daha güçlü hale getirir. r değerini artırmak populasyonda çeşitliliği artıracığından, şemaların yaşama şansını artırır.

5. UYGULAMA SONUÇLARI

Bu tez çalışmasında amaç, genetik algoritmalarda başlangıç popülasyonu oluşturulurken tamamen gelişigüzellik yerine, kısmen gelişigüzellik ile popülasyon oluşturmaktır. Bu şekilde elde edilen popülasyona düzenli popülasyon adı verildi ve bu popülasyonun gelişigüzel popülasyonun sağlamış olduğu avantajlı özelliklere sahip olduğu üçüncü bölümde teorik ve uygulamalı olarak gösterildi. Aynı bölümde bu popülasyonun gelişigüzel popülasyona göre olan üstünlükleri gösterildi. Bu bölümde ise beş tane çok modelli fonksiyonun maksimum noktasını bulmak için bu iki popülasyonun uygulama sonuçları verilerek karşılaştırmaları yapılacaktır. Kullanılan fonksiyonlar Griewank, Michalewicz, Rosenbrock, Rastrigin ve Peaks fonksiyonlarıdır.

Uygulama sonuçları verilmeden önce, bu çalışmada düzenli popülasyon ile gelişigüzel popülasyonu karşılaştırmak için daha önce verilen test fonksiyonları üzerinde uygulama gerçekleştirmek için bir paket program (Delphi 4.0, Inprise) hazırlanmıştır. Bu paket programın ana menüsü Şekil 5.1’ de görülmektedir. Ana menüde dört tane kısım vardır. Bunlar seçim türünü belirleme kısmı, çaprazlama belirleme kısmı, popülasyon tipini belirleme kısmı ve diğer parametreleri belirleme kısmıdır. Diğer parametreleri belirleme kısmında mutasyon çeşidi, deney sayısı, düzenli popülasyonda r değerini belirleme, popülasyon boyutu, mutasyon oranı gibi parametreler vardır.

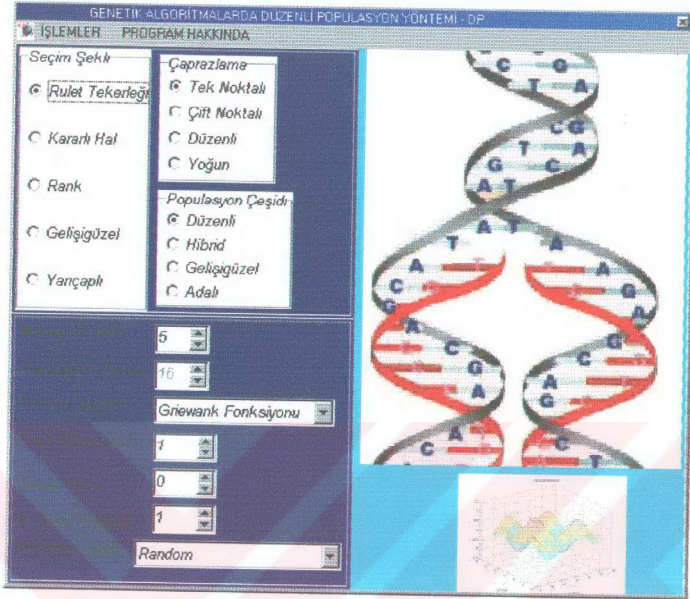
Uygulama sonuçları, her iki popülasyona klasik genetik operatörlerin uygulanması ile gösterilmiştir. Bundan sonra bu çalışmada tanımlanan genetik operatörlerin sonuçları sunulmuştur. Uygulanan genetik operatörler: Rulet tekerleği seçimi, düzenli çaprazlama ve gelişigüzel mutasyon şeklindedir.

5.1. Griewank Fonksiyonun Sonuçları

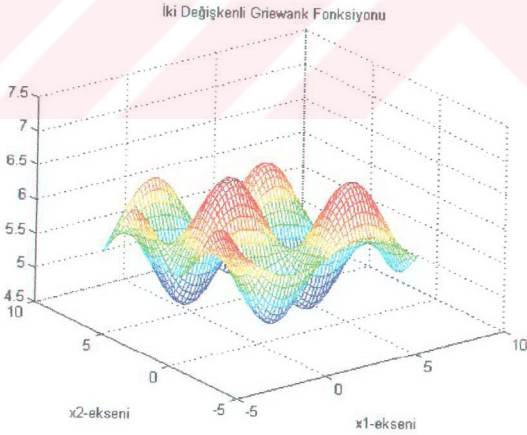
İlk olarak Griewank fonksiyonun genel tanımı verilebilir ve denklem 5.1’ de bu fonksiyonun genel tanımı görülmektedir.

$$f(x_1, \dots, x_n) = \frac{1}{4000} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1 \quad (5.1)$$

ve bu çalışmada kullanılan dört değişkenli Griewank fonksiyonu denklem (5.2)’ te görüldüğü gibidir. İki değişkenli Griewank fonksiyonun grafiği Şekil 5.2’ de görülmektedir.



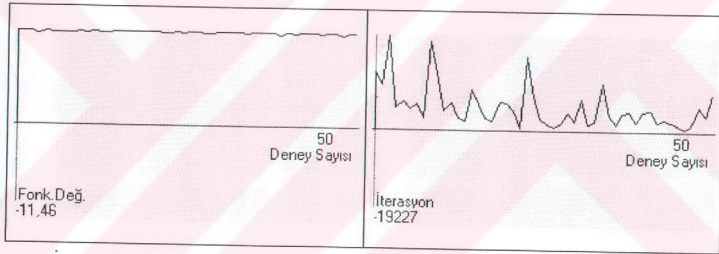
Şekil 5.1. Programın ana menüsü.



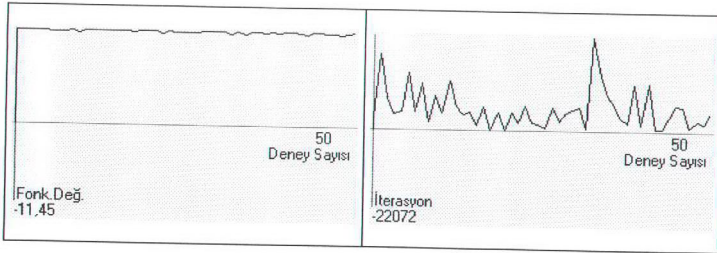
Şekil 5.2. İki değişkenli Griewank fonksiyonu.

$$f(x_1, x_2, x_3, x_4) = \frac{1}{4000} \left((x_1 - 100)^2 + (x_2 - 100)^2 + (x_3 - 100)^2 + (x_4 - 100)^2 \right) - \cos(x_1 - 100) \cos\left(\frac{x_2 - 100}{\sqrt{2}}\right) \cos\left(\frac{x_3 - 100}{\sqrt{3}}\right) \cos\left(\frac{x_4 - 100}{2}\right) + 1 \quad (5.2)$$

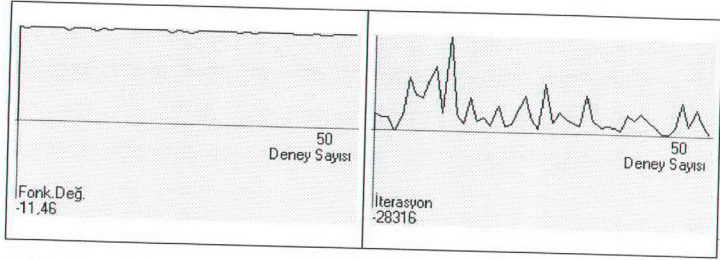
Griewank fonksiyonun $x_1, x_2, x_3, x_4 \in [-6.0, 6.0]$ aralığında maksimum noktası düzenli ve gelişigüzel populasyon ile elde edilmiştir. Düzenli populasyon kullanılırken $r=1, 2, 3, 4$ ve 5 değerleri için denenmiştir. Şekil 5.3' te görüldüğü gibi $r=1$ olduğu durumda düzenli populasyonun Griewank fonksiyonu sonucu görülmektedir. 50 deneme sonucunda elde edilen maksimum noktada fonksiyonun değeri 11.46 olurken ortalama iterasyon sayısı ise 4861 olmuştur. Şekil 5.4' de düzenli populasyonun $r=2$ olduğu durumda sonuçları görülmektedir. Elde edilen maksimum nokta $(-3.5765, -2.0, -3.5765, -1.0)$ şeklindedir. Düzenli populasyonun $r=2$ olduğu durumdaki sonuçlar Şekil 5.4' te görülmektedir. Elde edilen maksimum nokta $(x_1^*, x_2^*, x_3^*, x_4^*) = (-3.7333, -2.0, -4.0, -1.0)$ dır ve bu noktada elde edilen fonksiyon değeri 11.4481' dir. Bu sonucun bulunması için 50 deneme yapılarak ortalama iterasyon sayısı ise 5179



Şekil 5.3. Düzenli populasyonun ($r=1$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.

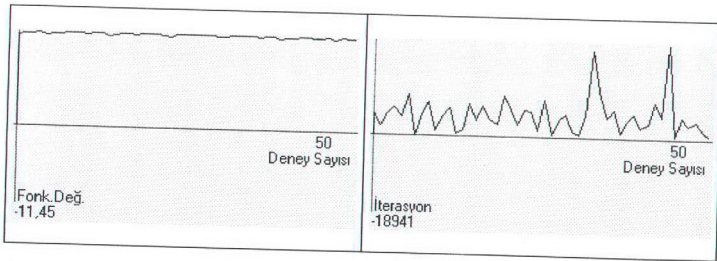


Şekil 5.4. Düzenli populasyonun ($r=2$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.

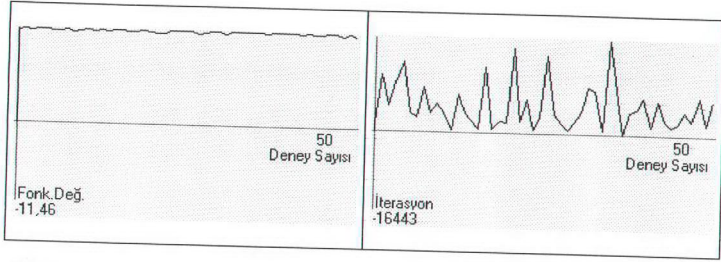


Şekil 5.5. Düzenli popülasyonun ($r=3$) Griewank fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.

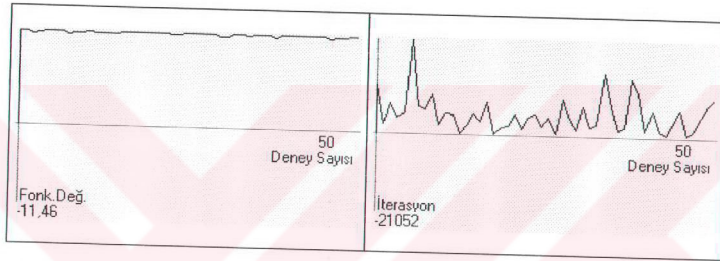
olmuştur. Şekil 5.5' te düzenli popülasyonun $r=3$ olduğu durumdaki sonuçları görülmektedir. 50 deneme sonucunda elde edilen maksimum nokta $(x_1^*, x_2^*, x_3^*, x_4^*) = (-3.5765, -2.0, -3.5765, -1.0)$ ' dir ve bu noktadaki fonksiyonun değeri ise 11.4693 olarak elde edilmiştir. Bu sonuçlar elde edilirken ortalama iterasyon sayısı 5908 olmuştur. Şekil 5.6' da düzenli popülasyonun $r=4$ olduğu durumdaki sonuçları görülmektedir. Fonksiyonun maksimum değeri $(x_1^*, x_2^*, x_3^*, x_4^*) = (-3.702, -2.0, -3.9529, -1.0)$ noktasında olup değeri 11.454' tür ve elde edilen ortalama iterasyon sayısı ise 4324' tür. Düzenli popülasyon için son olarak $r=5$ olduğu durumda elde edilen sonuçlar Şekil 3.7' de görülmektedir ve elde edilen maksimum fonksiyon değeri 11.4687 olup $(x_1^*, x_2^*, x_3^*, x_4^*) = (-3.6235, -2.0, -3.749, -1.0)$ noktasında elde edilmiştir. 50 deneme sonucunda ortalama iterasyon sayısı 4693' tür. Şekil 5.8' de ise gelişigüzel popülasyonun sonuçları görülmektedir. Bu yöntem fonksiyonun maksimum değerini 11.4633 olarak elde edip bu değere $(x_1^*, x_2^*, x_3^*, x_4^*) = (-3.8275, -2.0, -3.8275, -1.0)$ noktasında ulaşmıştır. 50 deneme sonucunda ortalama iterasyon sayısı 4740'dır. MATLAB (MathWork Inc.) ile bu fonksiyonun maksimum değerini 10.8524 olarak bulup bu değere $(x_1^*, x_2^*, x_3^*, x_4^*) = (-0.5819, 2.1575, 1.9062, -0.7342)$



Şekil 5.6. Düzenli popülasyonun ($r=4$) Griewank fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.7. Düzenli populasyonun ($r=5$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.8. Gelişigüzel populasyonun Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.

noktasında ulaştı.

Her iki populasyon türünün elde etmiş olduğu değer MATLAB'ın elde etmiş olduğu değerden daha kalitelidir. İterasyon sayısına gelince r büyüdükçe sonuca daha hızlı kavuşmaktadır. r 'nin küçük olduğu durumlarda gelişigüzel populasyonun iterasyon sayısı daha iyi olmaktadır.

5.2. Michalewicz Fonksiyonun Sonuçları

Test fonksiyonu olarak kullanılan ikinci test fonksiyonu Michalewicz fonksiyonudur ve tanımı denklem 5.3' te görüldüğü gibidir.

$$f(x_1, \dots, x_n) = \sum_{i=1}^n \sin(x_i) \sin^{20} \left(\frac{\pi x_i^2}{\pi} \right) \quad (5.3)$$

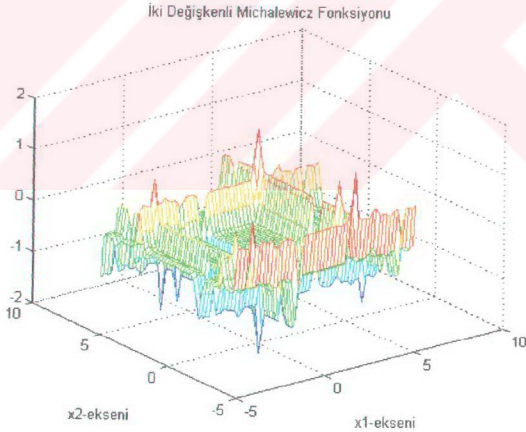
ve bu çalışmada kullanılan Michalewicz fonksiyonu dört değişkenli bir fonksiyondur ve dört değişkenli Michalewicz fonksiyonu denklem 5.4' te görülmektedir.

$$f(x_1, x_2, x_3, x_4) = \sin(x_1) \sin^{20}\left(\frac{x_1^2}{\pi}\right) + \sin(x_2) \sin^{20}\left(\frac{2x_2^2}{\pi}\right) + \sin(x_3) \sin^{20}\left(\frac{3x_3^2}{\pi}\right) + \sin(x_4) \sin^{20}\left(\frac{4x_4^2}{\pi}\right) \quad (5.4)$$

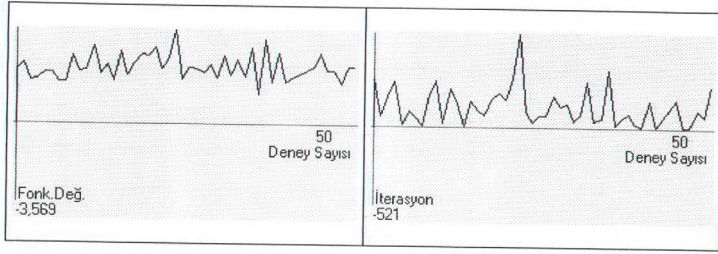
Şekil 5.9' da iki değişkenli Michalewicz fonksiyonun grafiği görülmektedir.

Michalewicz fonksiyonun $x_1, x_2, x_3, x_4 \in [-6.0, 6.0]$ aralığında maksimum noktası düzenli ve gelişigüzel populasyon ile elde edilmiştir. Düzenli populasyon kullanılırken $r=3, 4$ ve 5 değerleri için denenmiştir.

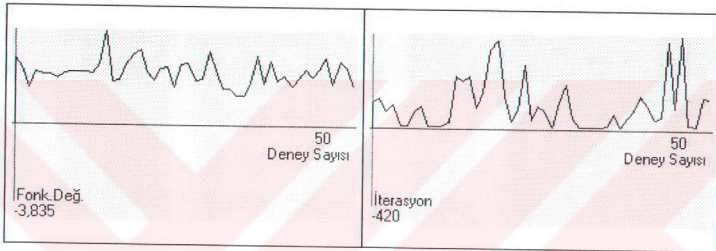
Şekil 5.10' da düzenli populasyonun $r=3$ için sonuçları görülmektedir ve fonksiyonun maksimum değerine (3.57) $(x_1^*, x_2^*, x_3^*, x_4^*) = (4.9529, -1.5686, -2.2549, 4.9431)$ noktasında ulaşılmıştır. Bunun için ortalama iterasyon sayısı 117 olmuştur. Şekil 5.11' de düzenli populasyonun $r=4$ durumunda elde edilen sonuçlar görülmektedir. $(x_1^*, x_2^*, x_3^*, x_4^*) = (4.949, -1.5854, -1.3137, 4.8431)$ noktasında fonksiyonun maksimum değeri elde edilmiştir ve bu değer 3.8351' dir. Ortalama iterasyon sayısı ise 109' dur. $r=5$ düzenli populasyonun sonucu



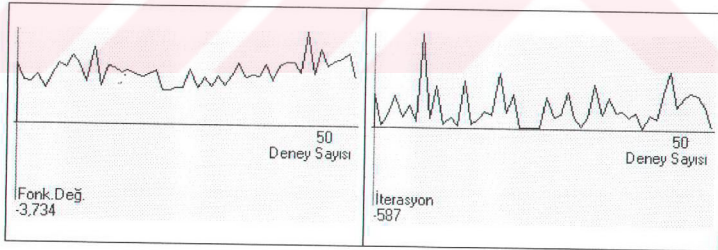
Şekil 5.9. İki değişkenli Michalewicz fonksiyonu.



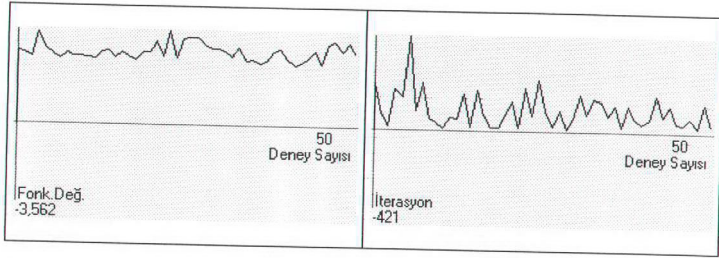
Şekil 5.10. Düzenli popülasyonun ($r=3$) Michalewicz fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.11. Düzenli popülasyonun ($r=4$) Michalewicz fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.12. Düzenli popülasyonun ($r=5$) Michalewicz fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.13. Gelişigüzel popülasyonun Michalewicz fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.

Şekil 5.12’ de görülmektedir ve $(x_1^*, x_2^*, x_3^*, x_4^*) = (4.949, -1.549, 4.9804, 4.8431)$ noktasında fonksiyonun maksimum değeri bulunmuştur ve bu değer 3.7347’ dir. Ortalama iterasyon sayısı 129’ dur. Şekil 5.13’ te ise gelişigüzel popülasyonun sonuçları görülmektedir. Bu popülasyon $(x_1^*, x_2^*, x_3^*, x_4^*) = (-2.2196, -1.5725, 5.2863, -1.9294)$ noktasında ekstrem noktasını buldu ve fonksiyonun bu noktadaki değeri 3.562’ dir. MATLAB ile aynı fonksiyonun değişkenler için aynı sınır değerlerine sahip tanım kümeleriyle $(0,0,0,0)$ noktasının maksimum nokta olduğunu ve bu noktada fonksiyonun değerini 0 olarak vermiştir. Dikkat edildiğinde her iki popülasyonun sonuçları MATLAB’ ın sonuçlarından çok daha iyi olduğu görülmektedir. Her iki popülasyonun kıyasına gelince iterasyon sayısında gelişigüzel popülasyon daha başarılyken, fonksiyonun maksimum değerini bulmada ise düzenli popülasyon daha iyidir.

5.3. Rosenbrock Fonksiyonun Sonuçları

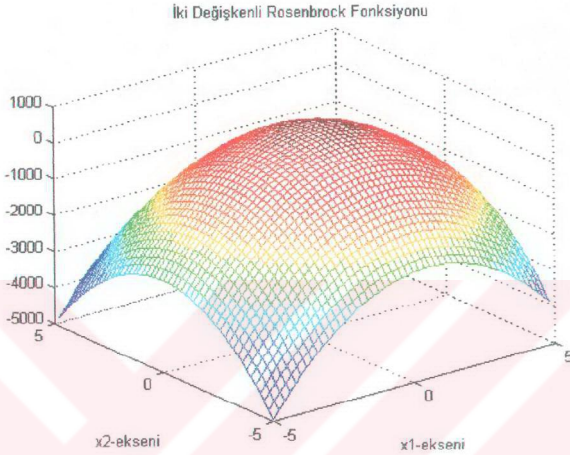
Üçüncü olarak kullanılan test fonksiyonu Rosenbrock fonksiyonudur ve bu fonksiyon da çok-modelli bir fonksiyondur. Tanımı denklem 5.5’ te verildiği gibidir ve bu çalışmada kullanılan dört değişkenli Rosenbrock fonksiyonu denklem 5.6’ da görülmektedir.

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2) + (1 - x_i)^2) \quad (5.5)$$

$$f(x_1, x_2, x_3, x_4) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 100(x_3 - x_2^2)^2 + (1 - x_2)^2 + 100(x_4 - x_3^2)^2 + (1 - x_3)^2 + 100(6 - x_4^2)^2 + (1 - x_4)^2 \quad (5.6)$$

İki değişkenli Rosenbrock fonksiyonun grafiği Şekil 5.14’ te görülmektedir ve bu çalışmada kullanılan Rosenbrock fonksiyonun değişken tanım aralığı $x_1, x_2, x_3, x_4 \in [-6.0, 6.0]$ şeklindedir.

Bu fonksiyon için r parametresinin 3, 4 ve 5 değerleri için sonuçlar alınıp gelişigüzel popülasyonun sonuçları ile karşılaştırılacaktır. Şekil 5.15’ te $r=3$ olduğu durumdaki düzenli popülasyonun sonuçları görülmektedir.



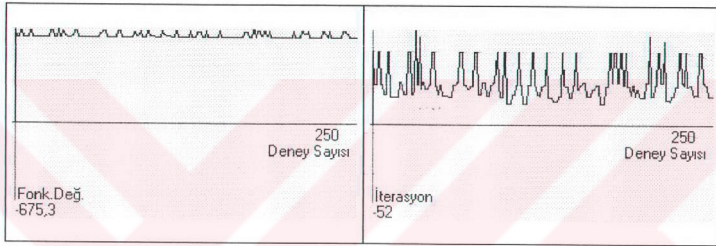
Şekil 5.14. İki değişkenli Rosenbrock fonksiyonu.



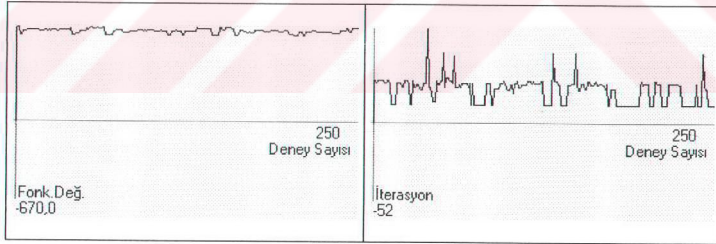
Şekil 5.15. Düzenli popülasyonun ($r=3$) Rosenbrock fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=250.



Şekil 5.16. Düzenli popülasyonun ($r=4$) Rosenbrock fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=250.



Şekil 5.17. Düzenli popülasyonun ($r=5$) Rosenbrock fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=250.



Şekil 5.18. Gelişigüzel popülasyonun Rosenbrock fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=250.

Şekil 5.15, Şekil 5.16 ve Şekil 5.17 şekilleri düzenli popülasyonun sırası ile $r=3, 4$ ve 5 değerleri için elde edilen sonuçları göstermektedir. Şekil 5.18' de ise gelişigüzel popülasyonun sonucu görülmektedir. İterasyon bakımından birbirlerine fazla bir üstünlükleri görülmemekle beraber elde edilen fonksiyon sonuçları bakımından fark vardır. Sonuçlar

$r=3$ için $(x_1^*, x_2^*, x_3^*, x_4^*) = (0.1255, 0.502, 0.3922, 0.5059)$ ve $f(x_1^*, x_2^*, x_3^*, x_4^*) = 673.8848$.

$r=4$ için $(x_1^*, x_2^*, x_3^*, x_4^*) = (0.0, 0.502, 0.502, 0.502)$ ve $f(x_1^*, x_2^*, x_3^*, x_4^*) = 676.743$.

$r=5$ için $(x_1^*, x_2^*, x_3^*, x_4^*) = (-0.1255, 0.498, 0.5294, 0.498)$ ve $f(x_1^*, x_2^*, x_3^*, x_4^*) = 675.3301$.

ve gelişigüzel popülasyon sonucu

$(x_1^*, x_2^*, x_3^*, x_4^*) = (-0.0314, 0.4196, 0.4706, 0.2471)$ ve $f(x_1^*, x_2^*, x_3^*, x_4^*) = 670.0186$.

Bu fonksiyonun verilen değişken tanım aralığında teorik maksimum değeri 676.6984 olup bu

değere $(x_1^*, x_2^*, x_3^*, x_4^*) = \left(\frac{1}{101}, \frac{51}{101}, \frac{51}{101}, \frac{51}{101}\right)$ noktasında ulaşmaktadır.

5.4. Rastrigin Fonksiyonun Sonuçları

Rastrigin fonksiyonu ise dördüncü test fonksiyonudur ve bu fonksiyon da çok-modelli bir fonksiyondur ve tanımı denklem 5.7’de verilirken, bu çalışmada kullanılan Rastrigin fonksiyonu ise denklem 5.8’de verilmiştir.

$$f(x_1, \dots, x_n) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) \quad (5.7)$$

$$f(x_1, x_2, x_3, x_4) = 20 + (x_1^2 - 10 \cos(2\pi x_1)) + (x_2^2 - 10 \cos(2\pi x_2)) \\ + (x_3^2 - 10 \cos(2\pi x_3)) + (x_4^2 - 10 \cos(2\pi x_4)) \quad (5.8)$$

Şekil 5.19’ da Rastrigin fonksiyonun iki değişkenli durumunun şekli görünürken, Şekil 5.20’ de ise aynı fonksiyonun izdüşüm durumunun şekli görülmektedir.

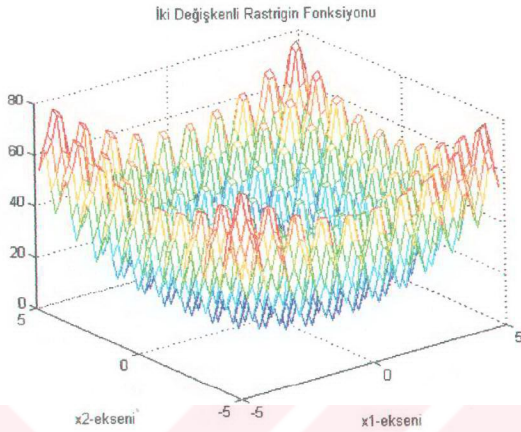
Şekil 5.20 ve Şekil 5.21’ de düzenli popülasyonun $r=4$ ve 5 değerleri için sonuçlar görülürken Şekil 5.22’ de ise gelişigüzel popülasyonun sonucu görülmektedir. Sonuçlar özetlenecek olursa,

$r=4$ için $(x_1^*, x_2^*, x_3^*, x_4^*) = (5.5216, 5.5215, 5.5176, 5.5216)$ ve $f(x_1^*, x_2^*, x_3^*, x_4^*) = 201.7511$.

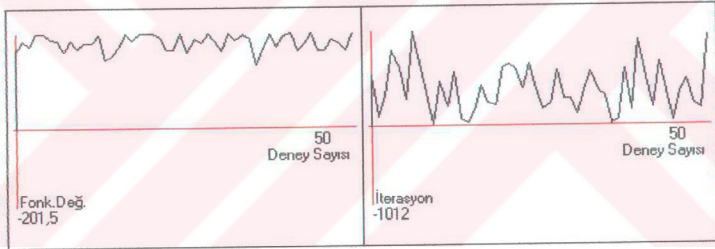
$r=5$ için $(x_1^*, x_2^*, x_3^*, x_4^*) = (5.5294, 5.5333, 5.5255, 5.5333)$ ve $f(x_1^*, x_2^*, x_3^*, x_4^*) = 201.6057$.

ve gelişigüzel popülasyon sonucu

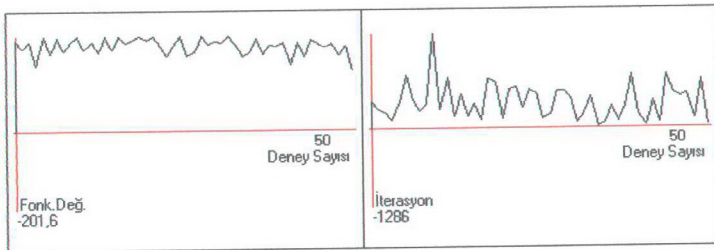
$(x_1^*, x_2^*, x_3^*, x_4^*) = (5.502, 5.5294, 5.502, 5.5294)$ ve $f(x_1^*, x_2^*, x_3^*, x_4^*) = 201.3499$.



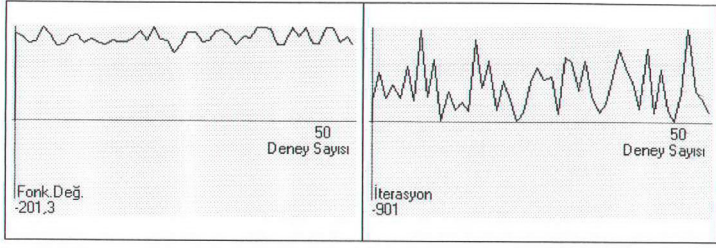
Şekil 5.19. İki değişkenli Rastrigin fonksiyonu.



Şekil 5.20. Düzenli popülasyonun ($r=4$) Rastrigin fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.21. Düzenli popülasyonun ($r=5$) Rastrigin fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.22. Gelişigüzel populasyonun Rastrigin fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.

Şekillerden görüldüğü gibi iterasyon sayıları arasında önemli bir fark görülmezken elde edilen fonksiyon sonuçları arasında fark vardır ve düzenli populasyonun elde etmiş olduğu sonuçlar daha kalitelidir.

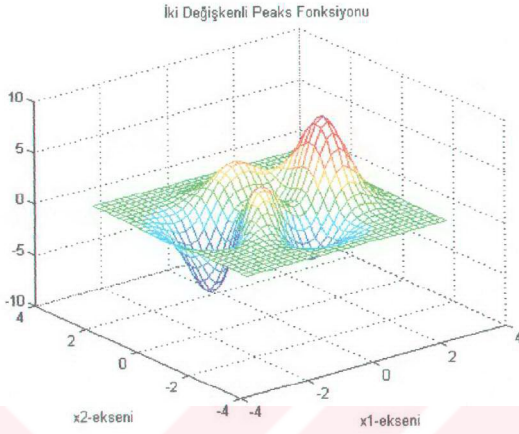
5.5. Peaks Fonksiyonun Sonuçları

Son olarak iki değişkenli olarak kullanılan Peaks fonksiyonun tanımı denklem 5.9' da görülürken, bu tez çalışmasında aynı fonksiyonun kullanılan şekli ise denklem 5.10' da görülmektedir.

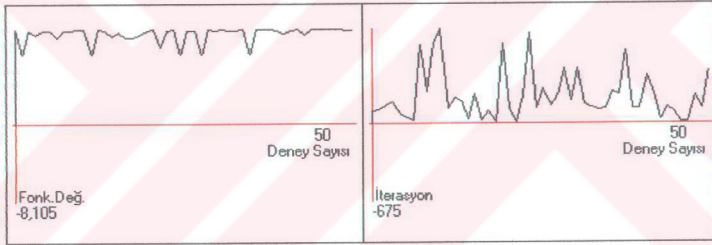
$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^{n-1} 3(1 - x_i)^2 e^{-x_i^2 - (x_{i+1}+1)^2} - \left(10 \left(\frac{x_i}{5} - x_i^3 - x_{i+1}^5 \right) e^{-x_i^2 - x_{i+1}^2} + \frac{1}{3} e^{-(x_i+1)^2 - x_i^2} \right) \quad (5.9)$$

$$f(x_1, x_2) = 3(1 - x_1)^2 e^{-x_1^2 - (x_2+1)^2} - \left(10 \left(\frac{x_1}{5} - x_1^3 - x_2^5 \right) e^{-x_1^2 - x_2^2} + \frac{1}{3} e^{-(x_1+1)^2 - x_2^2} \right) \quad (5.10)$$

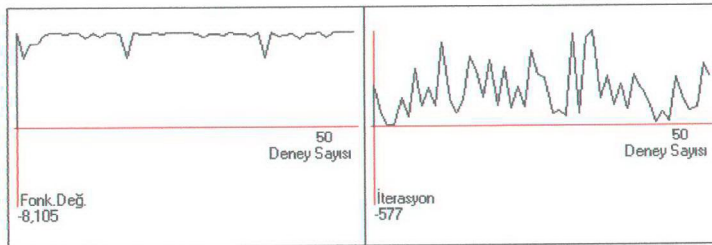
Bu çalışmada kullanılan Peaks fonksiyonun değişkenlerinin tanımlı olduğu aralık [-3.0,4.0] şeklindedir. İki değişkenli Peaks fonksiyon Şekil 5.23' te görülmektedir. Bu fonksiyonun maksimum değerini bulurken düzenli populasyonun r parametresi 4 ve 5 değerleri için denendi ve birde gelişigüzel populasyonun sonucu elde edildi.



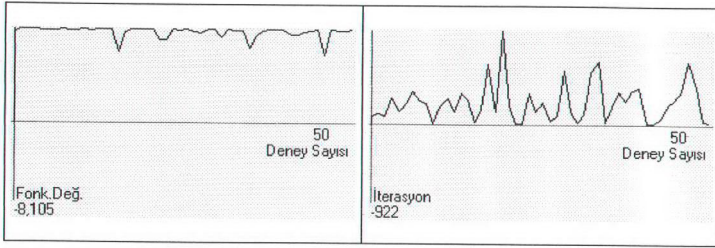
Şekil 5.23. İki değişkenli Peaks fonksiyonu.



Şekil 5.24. Düzenli popülasyonun ($r=4$) Peaks fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.25. Düzenli popülasyonun ($r=5$) Peaks fonksiyonu sonuçları: popülasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.26. Gelişigüzel populasyonun Peaks fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.

Şekil 5.24 ve Şekil 5.25’ te düzenli populasyonun Peaks fonksiyonu sonuçları görülmektedir. Şekil 5.26’ da ise gelişigüzel populasyonun sonucu görülmektedir. Sonuçlar arasında fazla büyük bir fark yoktur. Bu sonuçlar şu şekilde özetlenebilir.

$r=4$ için $(x_1^*, x_2^*)=(-0.0039, 1.5765)$ ve $f(x_1^*, x_2^*)=8.1056$.

$r=5$ için $(x_1^*, x_2^*)=(-0.0039, 1.5804)$ ve $f(x_1^*, x_2^*)=8.106$.

ve gelişigüzel populasyon sonucu

$(x_1^*, x_2^*)=(-0.0078, 1.5765)$ ve $f(x_1^*, x_2^*)=8.1058$.

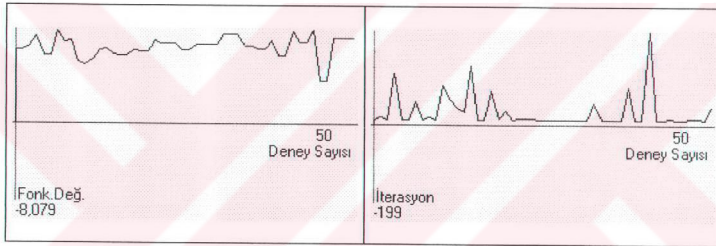
5.6. Yeni Operatörlerin Sonuçları

Üç tane yeni operatör tanımı üçüncü bölümde yapılmıştı. Bunlardan bir tanesi çaprazlama operatörü ve bu operatör tek noktali çaprazlamanın düzenli populasyon için olan şeklidir. Diğer ikisi ise mutasyon operatörleridir ve mutasyon operatörlerinin populasyonun değişimini artırıp artırmadığı kontrol edilecektir. Çünkü bu mutasyon çeşitlerinin amacı popülasyonda değişimlere sebep olmaktır. İlk olarak çaprazlama operatörünün tek-noktali çaprazlamaya göre olan farkına bakmak için tek-noktali çaprazlama ve yoğun çaprazlama Peaks fonksiyonuna uygulandı ve uygulama sonuçları Şekil 5.27 ve Şekil 5.28’ de görülmektedir.

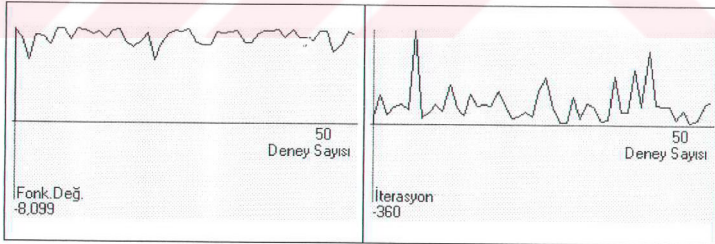
Her iki şekil dikkatlice incelendiği zaman görülecektir ki yoğun çaprazlama işlemi hızlandırıyor. Çünkü aynı kromozomların oluşmasını veya değişikliğe sebep olmayan çaprazlama adımlarını clemiş oluyor ve bunun yanında tek-noktali çaprazlamanın zamanı daha fazladır. Yoğun çaprazlamanın Şekil 5.27’ deki ortalama iterasyon sayısı 27 iken Şekil 5.28’ deki tek-noktali çaprazlamanın 78’ dir. Bu iki operatörün diğer fonksiyonları uygulanmasından sonra elde edilen iterasyon sayıları sırası ile şu şekildedir. İlk olarak yoğun çaprazlamanın sonuçları verilebilir: Rastrigin fonksiyonu için ortalama 72 iterasyon, Peaks fonksiyonu için

ortalama 27 iterasyon, Michalewicz fonksiyonu için ortalama 11 iterasyon, Griewank fonksiyonu için ortalama 542 iterasyon ve son olarak da Rosenbrock fonksiyonu için ise ortalama 11 iterasyon sonucunda işlem bitmektedir. Tek-noktalı çaprazlama durum bu kadar iyi değildir. Rastrigin fonksiyonu için ortalama 302 iterasyon, Peaks fonksiyonu için ortalama 78 iterasyon, Michalewicz fonksiyonu için 101 iterasyon, Griewank fonksiyonu için ortalama 2034 iterasyon ve Rosenbrock fonksiyonu için ise 12 iterasyon sonucunda işlem bitmektedir. Bütün fonksiyonlar için yoğun çaprazlamanın ortalama iterasyon sayısı tek-noktalı çaprazlamanın ortalama iterasyon sayısından düşük çıkmıştır.

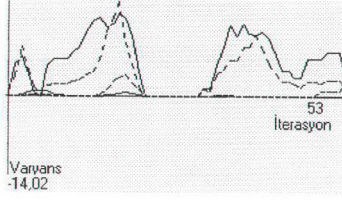
Yoğun çaprazlama operatörünün tanımlanmasının temel amacı bir kromozomda değişken ayırım noktası ile çaprazlama noktasının çakışmasını engellemektir. Bu durumda bu operatörün tek-noktalı çaprazlama göre daha kısa sürede işlemleri bitirtmesi gerekir. Sonuçlara bakıldığında yoğun çaprazlamanın tek-noktalı çaprazlamaya göre daha hızlı olduğu rahatlıkla görülebilir.



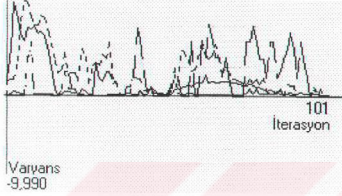
Şekil 5.27. Yoğun çaprazlamanın Peaks fonksiyonu sonuçları.



Şekil 5.28. Tek-Noktalı çaprazlamanın Peaks fonksiyonu sonuçları.



Şekil 5.29. Random mutasyon operatörünün popülasyonun değişimine olan etkisi.



Şekil 5.30. Transpozisyon mutasyon operatörünün popülasyonun değişimine olan etkisi.

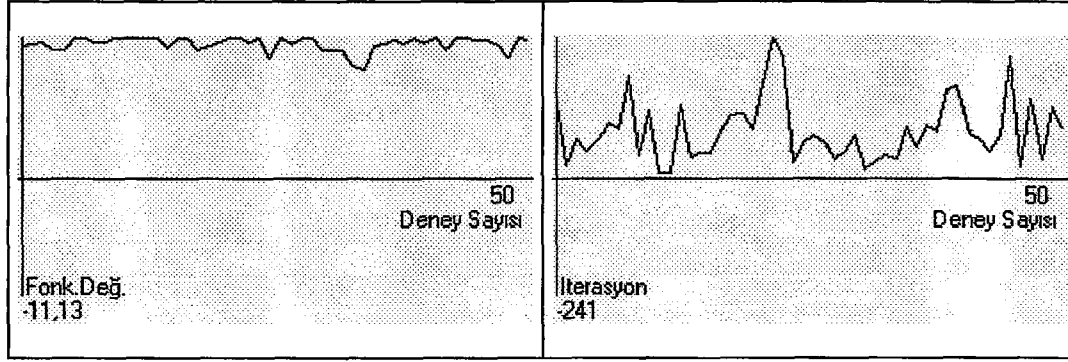
Mutasyon operatörlerinden sadece transpozisyon mutasyon operatörü ile random mutasyon operatörünün karşılaştırması yapılmıştır. Çünkü antisimetrik transpozisyon mutasyon operatörü transpozisyon mutasyon operatörüne benzemektedir.

Gelişigüzel mutasyon operatörünün popülasyonun çeşitliliğine olan katkısı Şekil 5.29’ da görülmektedir ve transpozisyon mutasyon operatörünün popülasyonun çeşitliliğine olan etkisi ise Şekil 5.30’ da görülmektedir. Her iki şekilde dikkati çeken durum; transpozisyon mutasyon operatörünün daha fazla çeşitliliğe sebep olduğudur.

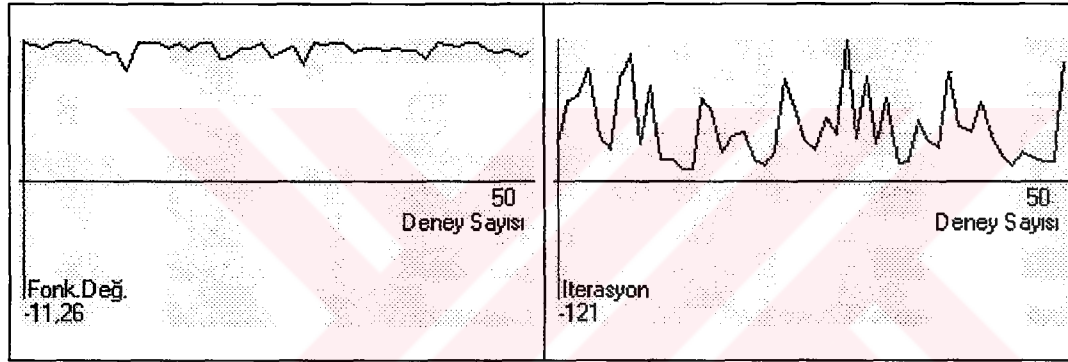
5.7. Adalı Düzenli Popülasyonun Uygulama Sonuçları

Griewank fonksiyonun maksimum noktasını bulmak için düzenli popülasyonun veya gelişigüzel popülasyonun uygulanması çok zaman almaktadır. Ortalama iterasyon sayıları binlerin üzerindeyken aynı problem adalı düzenli popülasyon ile çözüldüğünde iterasyon sayısından oldukça azalma meydana gelmektedir ve bu durum Şekil 5.31, Şekil 5.32, Şekil 5.33 ve Şekil 5.34’ de görülmektedir.

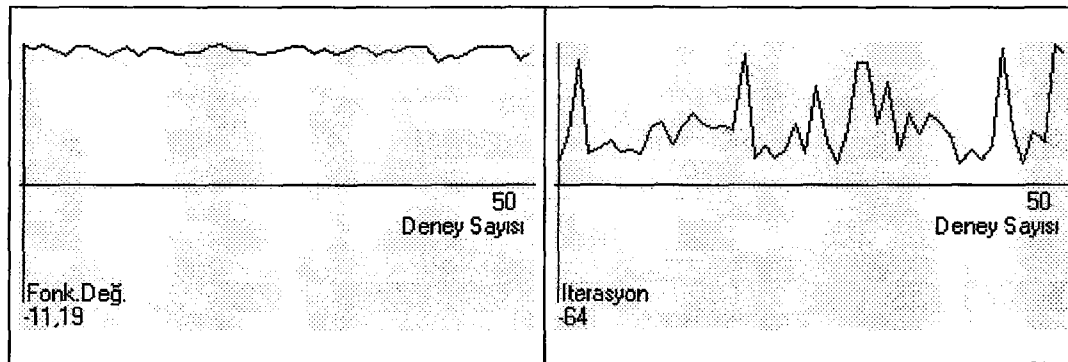
Griewank fonksiyonunun seçilmesinin sebebi düzenli ve gelişigüzel popülasyonların bu fonksiyonda sonuca ulaşmak için binler seviyesinde iterasyon yapmalarıdır (28 bin iterasyona kadar çıktığı oldu). Fakat şekiller incelendiği zaman, adalı düzenli popülasyonun aynı fonksiyonda daha kısa sürede sonuca ulaştığı görülebilir.



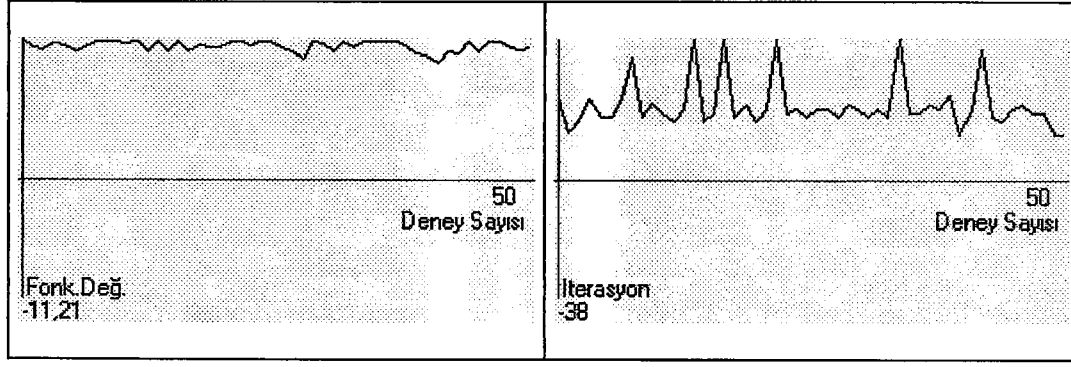
Şekil 5.31. Adalı düzenli populasyonun ($r=2$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.32. Adalı düzenli populasyonun ($r=3$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.33. Adalı düzenli populasyonun ($r=4$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.



Şekil 5.34. Adalı düzenli populasyonun ($r=5$) Griewank fonksiyonu sonuçları: populasyon boyutu=128, seçim şekli=rulet, mutasyon=random, çaprazlama=düzenli, deney sayısı=50.

6. SONUÇ VE DEĞERLENDİRME

Genetik algoritmalar, verilen problemin çözümü aranırken, birden fazla gelişigüzel başlangıç çözümü ile arama işlemine başlanır ve çözümün ne olduğu genetik algoritma tarafından bilinmez. Bundan dolayı başlangıç popülasyonu gelişigüzel üretildiği zaman elde edilen kromozomların hepsi yerel çözümlere daha yakın bölgelere denk düşebilirler. Bu nedenle global çözüme yaklaşımdan çok uzaklaşan çözümler veya yerel çözümde kalan çözümler elde edilebilir. Bu iki problem genetik algoritmalarda önemli bir yere sahiptirler. Bu tez çalışmasının temel amacı bu iki problemi giderecek bir yöntem geliştirmektir.

Bu amaçla başlangıç popülasyonu tamamen gelişigüzel olarak üretilmek yerine kısmen gelişigüzel üretilmesi daha uygun olduğu yapılan çalışmalar sonucunda görüldü. Popülasyonun gelişigüzel olmayan kısmı da belli bir yöntemle göre üretildi ve bu yöntemle düzenli popülasyon ismi verildi. Bu yöntem üçüncü bölümde tanıtıldı ve burada kısaca şu şekilde tanımlanabilir. N popülasyon boyutu, n kromozom uzunluğu, K_0 gelişigüzel üretilen kromozom, r düzenli popülasyon parametresi olmak üzere bu kromozomdan aşağıdaki kromozomlar üretilebilir (Kodlama ikili olduğu durum için).

$$K_0 + K_{2^r-1} = 2^n - 1$$

$$K_1 + K_{2^r-2} = 2^n - 1$$

.....

$$K_{2^r-k} + K_{2^r-k+1} = 2^n - 1$$

2^n-1 değeri ikili kodlama ve n uzunluklu kromozomlar için kromozomların temsil ettiği uzayın medyanının 2 ile çarpılmış durumudur. Bunun anlamı elde edilen popülasyon uzayın medyanına göre simetrik bir popülasyondur ve uzaya düzgün bir şekilde dağıtılmış durumdadır. Popülasyonun düzenliliği ile ilgili olasılıksal ispatlar üçüncü bölümde daha detaylı bir şekilde gösterildi (çarpıklık katsayısının sıfır olması, ortalama değerinin medyana eşit olması).

Gelişigüzel popülasyonun sağlamış olduğu iki tane önemli özellik vardır. Bunlar popülasyonun çeşitlilik içermiş olması ve nesiller arasında lineer bağımsızlığın olmasıdır. Üçüncü bölümde P popülasyonun bu iki özelliği sağladığı detaylı olarak gösterildi. Bununla beraber bu popülasyonun düzenlilik özelliğine de sahip olduğu ve uzaya düzgün bir şekilde dağıtıldığı aynı bölümde detaylı olarak gösterildi. Aynı zamanda bu popülasyon uzayın medyanına göre simetrik özelliğine de sahiptir.

Bu çalışmada her iki populasyonun karşılaştırmasını yapmak için beş tane çok modelli fonksiyon üzerinde uygulamaları gerçekleştirildi ve genellikle düzenli populasyonun daha başarılı sonuçlar elde ettiği beşinci bölümde gösterildi. Bu sonuçlarla ilgili yorumlar kısaca verilecek olursa:

1- Griewank fonksiyonunda her iki populasyon türünün elde etmiş olduğu değerin birbirine çok yakın olduğu ve r değeri büyüdükçe iterasyon sayısının gelişigüzel yönteme göre daha az olduğu görüldü. Her iki yöntemde elde edilen sonuçlar Matlab' ın elde etmiş olduğu sonuçtan daha iyi olduğu gözlemlendi.

2- Michalewicz fonksiyonunda yapılan uygulamada her iki populasyonun elde etmiş olduğu sonuçların Matlab' ın elde ettiği sonuçlardan çok daha iyi olduğu görüldü. Her iki populasyonun kıyasına gelince iterasyon sayısında gelişigüzel populasyon daha başarılıyken, fonksiyonun maksimum değerini bulmada ise düzenli populasyon daha iyidir.

3- Rosenbrock fonksiyonun uygulamasında her iki populasyonun iterasyon sayıları arasında fazla bir fark yok iken, elde edilen fonksiyon değerlerine göre düzenli populasyon daha başarılıdır.

4- Rastrigin fonksiyonun uygulamasında her iki populasyonun iterasyon sayıları arasında önemli bir fark görülmezken elde edilen fonksiyon sonuçları arasında fark vardır ve düzenli populasyonun elde etmiş olduğu sonuçlar daha kalitelidir.

5- Peaks fonksiyonun sonuçlarına gelince elde edilen fonksiyon değerleri arasında önemli bir fark yok iken iterasyon sayısı bakımından düzenli populasyon gelişigüzel popülasyona göre daha kısa zamanda sonuca ulaştığı görüldü.

Düzenli populasyon ile beraber kullanılacak mutasyon ve çaprazlama operatörleri için yeni tanımlamalar gerçekleştirildi. Bu yeni tanımlamaların biri çaprazlama ile ilgiliyken iki tanesi mutasyon ile ilgiliydi. Tek-noktalı çaprazlamanın düzenli populasyon için bir noktada iterasyon sayısında artışa sebep olma ihtimali olduğundan bu operatörün düzenli populasyon için olan tanımı yapıldı ve buna yoğun çaprazlama ismi verildi. Beşinci bölümde görüldüğü gibi yoğun çaprazlamanın tek-noktalı çaprazlamaya göre daha az iterasyon sayısı ile sonuca ulaştığı gösterildi.

Her fonksiyon için elde edilen ortalama iterasyon sayıları beşinci bölümde verilmiştir. Her fonksiyon için yoğun çaprazlamanın tek-noktalı çaprazlamaya göre daha hızlı olduğu rahatlıkla görülebilmektedir.

Random mutasyon operatörünün populasyonun çeşitliliğine olan katkısı Şekil 5.29' da görülmektedir ve transpozisyon mutasyon operatörünün populasyonun çeşitliliğine olan etkisi ise Şekil 5.30' da görülmektedir. Bu iki şekilden de anlaşılacağı gibi transpozisyon mutasyon

operatörünün popülasyonun değişimini sürekli koruduğudur. Bunun sonucunda popülasyonda çeşitlilik sürekli olarak korunur.

Düzenli popülasyonun yapısından faydalanılarak adalı popülasyon yapısı elde edilebilir. Düzenli popülasyondan adalı düzenli popülasyon elde etmek için 2^f tane ada kullanılır. K_{ϕ_i} kromozom i . adaya yerleştirilir. Bu işlem bütün ϕ değerleri için gerçekleştirilir. Bu durumda her adada bulunan kromozom sayısı $N/2^f$ tane olur ve her adada olacak kromozom sayısının belli bir değerden büyük olmasına özen gösterilmelidir. Aksi halde mikro genetik algoritmanın sahip olduğu özellikler her ada için geçerli olabilir.

Beşinci bölümde Griewank fonksiyonun sonuçlarına bakıldığı zaman her iki popülasyon türü için de iterasyon sayılarının çok yüksek olduğu görülür. Adalı düzenli popülasyonun aynı fonksiyon uygulamalarında ise iterasyon sayısından büyük bir oranda düşüş olduğu gözlemlenmektedir.

Griewank fonksiyonunun seçilmesinin sebebi düzenli ve gelişigüzel popülasyonların bu fonksiyonda sonuca ulaşmak için binler seviyesinde iterasyon yapmalarıdır (28 bin iterasyona kadar çıktığı oldu). Fakat şekiller incelendiği zaman adalı düzenli popülasyonun aynı fonksiyonda daha kısa sürede sonuca ulaştığı görülmektedir ve en fazla iterasyon sayısının yüzler seviyesine düştüğü görülmektedir. Bu çok önemli bir noktadır; çünkü genetik sürecin zamanı kısalmaktadır.

Diğer önemli bir noktada çaprazlama ve mutasyon operatörlerinin düzenli popülasyonda şemalar üzerine olan etkileridir. Bu iki operatörün etkileri dördüncü bölümde detaylı olarak verilmiştir. Bu bölümde elde edilen en önemli sonuç; kullanılan alfabenin eleman sayısı 2 olmasına rağmen düzenli popülasyonun etkisi ile sanki popülasyonun boyutu 2^f den büyümüş gibi bir etki ortaya çıkmaktadır. Böylece çeşitliliğin arttığı teorik olarak da gösterilmiş oldu.

Gelişigüzel ve düzenli popülasyonların özellikleri tekrar aşağıdaki gibi özetlenebilir.

Gelişigüzel Popülasyon:

- Çeşitlilik vardır.
- Nesiller arasında lineer bağımlılık yoktur.
- Yerel çözümde kalabilir.
- Global çözümden uzaklaşabilir.
- Başlangıç popülasyonunda bütün kromozomların aynı olma olasılığı sıfır değildir.

Düzenli Popülasyon:

- Çeşitlilik vardır.
- Nesiller arasında lineer bağımlılık yoktur.

- Yerel çözümde kalmaz.
- Genel çözümde uzaklaşmaz.
- Çeşitliliği arttırmaktadır.
- Adalı popülasyona dönüştürüldüğünde daha kısa zamanda sonuç vermektedir.
- Başlangıç popülasyonunda bütün kromozomların aynı olma olasılığı sıfırdır.
- Adalar arasında kromozomların başlangıçta aynı olma olasılığı sıfırdır.
- İkili kodlama kullanıldığı halde sanki kodlama alfabesinin boyutu 2' den büyükmiş gibi bir etkiye sahiptir.
- Elde edilen popülasyon düzgün bir popülasyondur.
- Uzayın medyanına göre simetrik bir popülasyondur.
- Düzenli popülasyon, genellikle gelişigüzel popülasyondan daha az iterasyonla daha iyi sonuç vermektedir.
- Adalı düzenli popülasyon her zaman gelişigüzel popülasyondan daha kısa zamanda sonuç verir.

İleriye yönelik yapılan çalışmalarda, Adalı düzenli popülasyonun her alt adanın boyutu yeterince büyük olacak şekilde ayarlandıktan sonra mühendislik ve bilimsel problemlere uygulanabilir. Her adadan elde edilen en iyi sonuçların bir popülasyon olarak bir araya toplanması ve bu yeni elde edilen en iyilerin popülasyonu bir daha genetik süreçten geçirilebilir. Bu şekilde iki-seviyeli genetik algoritma elde edilir.

Bu tez çalışmasının konusu ile ilgili olarak şu ana kadar üç tane ulusal konferansta bildiri yayınlanmıştır ve bir tane uluslararası konferansta bildiri yayınlanmıştır. Aynı konuda iki tane de makale kabul edilmiştir.

KAYNAKLAR

- Agapie, A., 2001, "Theoretical Analysis of Mutation-Adaptive Evolutionary Algorithms", *Evolutionary Computation*, MIT PRESS, 9, no.2, 127-146.
- Anderson, J.R., 1976, "*Language, memory and thought*", Hillsdale, NJ: Lawrence Erlbaum.
- Anderson, J.R., 1983, "*The Architecture of Cognition*", Cambridge, MA: Harvard University Press.
- Baker, J.E., 1987, "Reducing Bias and Inefficiency in the Selection Algorithm", in Grefenstette, 14-21.
- Berliner, H., 1977, "Experiences in Evaluation with BKG- a Program That Plays Backgammon", in the proceedings of International Joint Conference on Artificial Intelligence.
- Branke, J.; Schmidt, C.; Schmeck, H., 2001, "Efficient Fitness Estimation in Noisy Environment", GECCO-2001: Genetic and Evolutionary Computation Conference, 243-250.
- De Jong, K.A., 1975, "An Analysis of the Behaviour of a Class of Genetic Adaptive Systems", *Dissertation Abstracts International*, 36, no.10, 5140B.
- Erman, L.D.; Hayes-Roth, F.; Lesser, V.R.; Reddy, R.D., 1980, "The HEARSAY-II Speech Understanding System, : Integrating Knowledge to resolve Uncertainty", *ACM Computing Survey*, 12, 213-253.
- Furutani, H., 2001, "Study of Crossover in OneMax Problem", GECCO-2001: Genetic and Evolutionary Computation Conference, 320-327.
- Goldberg, D., 1989, "*Genetic Algorithms in Search, Optimization and Machine Learning*", Addison-Wesley.
- Goldberg, D.E.; Richardson, J., 1987, "Genetic Algorithms with Sharing for Multimodal Function Optimization", *Proceedings of the 2nd International Conference on Genetic Algorithms*, Hillsdale, NJ: Lawrence Erlbaum Associates, 41-49.
- Goldberg, D.E.; Sastry, K., 2001, "A Practical Schema Theorem for Genetic Algorithm Design and Tuning", GECCO-2001: Genetic and Evolutionary Computation Conference, 328-335.
- Goldberg, D.E.; Sastry, K.; Latoza, T., 2001, "On the Supply of Building Blocks", GECCO-2001: Genetic and Evolutionary Computation Conference, 336-342.
- Hancock, P.J.B., 1994, "An Empirical Comparison of Selection Methods in Evolutionary Algorithms", in Fogarty, 80-94.
- Hart, W.E., 2001, "A Convergence Analysis of Unconstrained and Bound Constrained Evolutionary Pattern Search", *Evolutionary Computation*, MIT Press, 9, no.1, 1-23.

- Holland, J.H., 1975, "*Adaptation in Natural and Adaptive Systems*", University of Michigan Press.
- Holland, J.H., 2000, "Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions", *Evolutionary Computation*, MIT Press, 8, no.4, 373-391.
- Huang, C.-F., 2001, "Independent Sampling Genetic Algorithms", *GECCO-2001: Genetic and Evolutionary Computation Conference*, 367-374.
- Karacı, A.; Çınar, A.; Ergen, B., 1999, "Genetik Algoritma Kullanılarak Minimum Düğüm Kapsama Probleminin Çözümü", *Elektrik-Elektronik-Bilgisayar Mühendisliği 8. Ulusal Kongresi*, 6-12 Eylül, Gaziantep, Türkiye, 20-23.
- Karacı, A.; Arslan, A., "Bidirectional Evolutionary Heuristic for the Minimum Vertex-Cover Problem", *Journal of Computer and Electrical Engineering*, Elsevier.
- Karacı, A.; Çınar, A., 2000, "Comparison of Uniform Distributed Initial Population Method and Random Initial Population Method in Genetic Search", *15th International Symposium on Computer and Information Sciences*, October, 11-13, Istanbul, Turkey, 159-166.
- Karacı, A.; Arslan, A., 2002, "Genetik Algoritmalarla Düzenli Populasyon", *GAP IV: Mühendislik konferansı*, 06-08 Haziran 2002, Harran Üniversitesi, Şanlıurfa.
- Kee, E., Airey, S., Cyre, W., 2001, "An Adaptive Genetic Algorithm", *GECCO-2001: Genetic and Evolutionary Computation Conference*, 391-397.
- Kuo, T.; Hwang, S.-Y., 1996, "A Genetic Algorithm with Disruptive Selection", *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, 26, no.2, 299-306.
- Lindsay, R.K.; Buchanan, B.G.; Feigenbaum, E.A.; Lederberg, J., 1980, "*Applications of Artificial Intelligence for Chemical Inference: The DENDRAL Project*", New York, McGraw-Hill.
- Mahfoud, S.W., 1992, "Crowding and Preselection Revisited", *Parallel Problem Solving from Nature*, 2, The Netherlands, Elsevier Science, 27-36.
- McClelland, J.; Rumelhart, D., 1986, "Parallel Distributed Computing: Explorations into the Microstructure of Cognition", Vol.2, Psychological and biological models, Cambridge, MA: MIT Press.
- Minsky, M., 1985, "*The Society of Mind*", New York, Simon and Schuster.
- Newell, A., 1990, "*Unified Theories of Cognition*", MA: Harvard University Press.
- Newell, A.; Simon, H.A. (1963), "GPS: A Program That Simulates Human Thought", in *Computer and Thought*, New York: McGraw-Hill, 279-293.
- Prügel-Bennett, A., 2001, "Modeling Crossover-Induced Linkage in Genetic Algorithms", *IEEE Trans. on Evolutionary Computation*, 5, no.4, 376-387.

- Samuel ,A., 1959, "Some Studies in :Machine Learning Using the Game of Checkers", IBM Journal of Research and Development, 3, 210-229.
- Schank, R.C., 1987, "What is AI anyway?" The AI Magazine, 8 (4), 59-65.
- Shortliffe, E.H., 1976, "Computer-based Medical Consultations : Mycin", New York.
- Spears, W.M., 1998, "The Role of Mutation and Recombination in Evolutionary Algorithms", Ph.D. dissertation, George Mason University.
- Stephens, C.; Waelbroeck, H., 1999, "Schemata Evolution and Building Blocks", Evolutionary Computation, MIT Press, 7, no.2, 109-124.



ÖZGEÇMİŞ

Lisans eğitimini Bilkent Üniversitesi Bilgisayar ve Enformatik Mühendisliği bölümünde 1994' te tamamladı. Aynı yıl zorunlu askerlik hizmetine başladı ve bu hizmeti 1995 yılının sonunda tamamladı. 1995-1996 eğitim yılının bahar döneminde Fırat Üniversitesi Bilgisayar Mühendisliği bölümünde yüksek lisans eğitimine başladı ve aynı dönemde aynı bölümde araştırma görevlisi oldu. Yüksek lisans eğitimini 1997-1998 eğitim yılının bahar döneminde tamamladı ve 1998-1999 eğitim yılının güz döneminde aynı üniversitenin Elektrik-Elektronik Mühendisliği bölümünde doktora eğitimine başladı.

İlgi Alanları : Evrimsel Hesaplama, Çizge Teorisi, Spektral Çizge Teorisi, Paralel ve Dağıtık Sistemler, Paralel ve Dağıtık Hesaplama, Arabağlama Ağları.

Yayınları: Yapmış olduğu yayınlar özellikle iki kategoride ele alınabilir. Evrimsel hesaplama alanında 7 tane ve paralel ve dağıtık sistemlerde (hesaplama) ise 9 tane olmak üzere toplam 16 tane yayın yapmıştır.

2001 international Conference on Parallel and Distributed Systems, Jun. 26-29, 2001, KyongJu, Korea konferansında program komitesinde görev aldı ve The 17 th ACM Symposium on Applied Computing March 10 - 14, 2002, Madrid, Spain Special Track on Parallel and Distributed Systems and Networking ise konferans başkanlığını yürütmüştür.

Journal of Information Science and Engineering dergisinin Special Issue on Parallel and Distributed Systems özel sayısında hakemlik yapmıştır.