

**YÜKSEK GERİLİM TEKNİĞİNDE
GENETİK ALGORİTMA UYGULAMALARI**

**YÜKSEK LİSANS TEZİ
Müh. Alper GÜÇLÜ**

**Anabilim Dalı: Elektrik Mühendisliği
Programı: Elektrik Mühendisliği**

Tez Danışmanı: Doç.Dr. Özcan KALENDERLİ

NİSAN 2004

**YÜKSEK GERİLİM TEKNİĞİNDE
GENETİK ALGORİTMA UYGULAMALARI**

YÜKSEK LİSANS TEZİ

Müh. Alper GÜÇLÜ

(504011002)

Anabilim Dalı: Elektrik Mühendisliği

Programı: Elektrik Mühendisliği

Tez Danışmanı: Doç.Dr. Özcan KALENDERLİ

NİSAN 2004

ÖNSÖZ

Bu tez çalışmasında, yüksek gerilim tekniğinde elektrot sistemlerinde delinme ve ekonomik bakımdan elektriksel alan hesaplamaları, elektriksel alanın basınç ve açıklıkla değişimi, elektriksel alana geometrik pürüzlerin etkisi ve darbe akım üreticinin maksimum akımının bulunması gibi uygulamalar için yeni bir yaklaşım, Genetik Algoritmalar kullanılmıştır.

Bana bu çalışmayı yapma olanağı veren, çalışmalarımı büyük bir sabır ve özenle inceleyerek, karşılaştığım zorlukları aşmak için bana yol gösteren değerli hocam Doç. Dr. Özcan Kalenderli'ye, sevgili meslektaşım Suna Bolat'a ve Ali Miharbi'ye, ayrıca çalışmalarım boyunca bana destek olan arkadaşlarıma ve aileme içten teşekkürlerimi sunarım.

Nisan, 2004

Alper GÜÇLÜ

İÇİNDEKİLER

KISALTMALAR.....	v
TABLO LİSTESİ.....	vi
ŞEKİL LİSTESİ.....	vii
SEMBOL LİSTESİ.....	ix
ÖZET.....	xi
SUMMARY.....	xii
1. GİRİŞ.....	1
1.1. Giriş ve Çalışmanın Amacı.....	1
2. GENETİK ALGORİTMALAR.....	3
2.1. Genetik Algoritmaların Tarihsel Gelişimi.....	3
2.2. Biyolojik Temeller.....	6
2.2.1. Temel Tanımlar.....	6
2.3. Genetik Algoritmalar Nedir?.....	7
2.4. Genetik Algoritmaların Diğer Optimizasyon Yöntemleri ile Karşılaştırılması.....	10
2.5. Uygulama Alanları.....	14
2.5.1. Mühendislik.....	15
2.5.2. Yönetim.....	15
2.5.3. Finans.....	15
2.5.4. Ar-Ge.....	15
2.6. Seçim.....	15
2.6.1. Rulet Çarkı Seçimi.....	15
2.6.2. Rank Seçimi.....	17
2.6.3. Kararlı Hal Seçimi.....	20
2.6.4. Turnuva Seçimi.....	20
2.6.5. Seçkinlik.....	20
2.7. Genetik Algoritma İşlemcileri.....	20
2.7.1. Kodlama.....	20
2.7.1.1. İkili Kodlama.....	20
2.7.1.2. Dizilim (permütasyon) Kodlaması.....	21
2.7.1.3. Değer (Reel) Kodlaması.....	22
2.7.1.4. Ağaç (tree) Kodlaması.....	22
2.7.2. Çaprazlama.....	23

2.7.2.1. İkili Kodlama İçin Çaprazlama.....	23
2.7.2.2 Reel Kodlama İçin çaprazlama.....	26
2.7.3. Mutasyon.....	29
2.7.3.1. Reel Değişkenli Bireyler İçin Mutasyon.....	29
2.7.3.2. İkili Kodlanmış Bireyler İçin Mutasyon.....	30
2.8. Örnekler.....	31
2.8.1. Örnek 1.....	31
2.8.2. Örnek 2.....	34
3. YÜKSEK GERİLİM UYGULAMALARI.....	39
3.1 Statik Elektrik Alanı.....	39
3.1.1 Küresel Elektrot Sistemi.....	39
3.1.1.1 Küresel Elektrot Sisteminin Delinme Bakımından İncelenmesi.....	41
3.1.2 Silindirsel Elektrot Sistemi.....	43
3.1.2.1 Eş Eksenli Silindirik Sistemin Ekonomik Bakımdan İncelenmesi.....	45
3.1.2.2 Eş Merkezli Silindirik Sistemin Delinme Bakımından İncelenmesi.....	46
3.2 Düzgün Alanda Delineme Gerilimi.....	47
3.3 Farklı Geometrilerin Elektriksel Alan Üzerindeki Etkileri.....	50
3.4 Darbe Akım Üreticinin Akım Karakteriğinin İncelenmesi.....	52
4. SONUÇLAR VE TARTIŞMA	56
KAYNAKLAR	59
EK A.....	64
EK B	67
EK C.....	79
ÖZGEÇMİŞ	89

KISALTMALAR

GA	: Genetik Algoritmalar
EP	: Evrimsel Programlama
ES	: Evrimsel Stratejiler

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. Çeşitli yıllarda basılan GA yayınlarının sayısı	5
Tablo 2.2. Genetik algoritmanın akış tablosu.....	9
Tablo 2.3. Seçilme olasılıkları ve uygunluk değerleri.....	17
Tablo 2.4. Uygunluk değerlerinin seçilme baskısı ile değişimi.....	19
Tablo 2.5. İkili kodlama kullanılarak oluşturulan bireyler.....	21
Tablo 2.6. Dizilim kodlaması kullanılarak oluşturulan bireyler.....	21
Tablo 2.7. Gerçel kodlama kullanılarak oluşturulan bireyler	22
Tablo 2.8. Ağaç yapı kodlaması kullanılarak oluşturulan bireyler.....	23
Tablo 2.9. Mutasyon öncesi ve sonrası bireyler.....	30
Tablo 2.10 Farklı kodlamalarda mutasyon etkisi.....	31
Tablo 2.11 150 nesil boyunca problem verileri.....	38
Tablo 3.1 Küresel elektrot sisteminde analitik ve GA sonuçları.....	40
Tablo 3.2 Küresel Elektrot Sisteminin Delinme Bakımından İncelenme Sonuçları.....	42
Tablo 3.3 Silindirs el elektrot sisteminde analitik ve GA sonuçları.....	44
Tablo 3.4 p geometrik karakteristiğinin hesaplanması.....	46
Tablo 3.5 Silindirs el Elektrot Sisteminin Delinme Bakımından İncelenme Sonuçları.....	46
Tablo 3.6 Değerleri bilinen gazın pkr ve minimum delinme dayanımının hesaplanması.....	49
Tablo 3.7 a'nın ve p'nin sabit olduğu durumlarda Ud değişimi.....	50
Tablo 3.8 Farklı geometrideki pürüzlerde alan ifadeleri.....	51
Tablo 3.9 Hesaplama sonuçları.....	52
Tablo 3.10 Maksimum akım hesaplama sonuçları.....	55

ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1 : İkili kodlama yapısındaki birey.....	8
Şekil 2.2 : Genetik algoritma program blok şeması.....	9
Şekil 2.3 : Çok optimum noktalı fonksiyon.....	11
Şekil 2.4 : Parazitlerin karıştığı fonksiyon.....	11
Şekil 2.5 : Optimizasyon yöntemlerinin verimleri.....	13
Şekil 2.6 : Rulet çarkı sistemi.....	16
Şekil 2.7 : Üretilen sayıların uygun düştüğü birey aralıkları.....	17
Şekil 2.8 : Sıralama öncesi durum.....	18
Şekil 2.9 : Sıralama sonrası durum.....	18
Şekil 2.10 : Lineer ve lineer olmayan rank seçimlerinin karşılaştırılması.....	19
Şekil 2.11 : Tek noktalı çaprazlama.....	24
Şekil 2.12 : Çok noktalı çaprazlama.....	25
Şekil 2.13 : Ayrık çaprazlaması ile oluşabilecek yavruların olası konumları..	26
Şekil 2.14 : Ara çaprazlamada ebeveynlerin ve yavruların dağılımları.....	27
Şekil 2.15 : Ara çaprazlaması ile oluşabilecek yavruların olası konumları.....	28
Şekil 2.16 : Doğrusal çaprazlaması ile oluşabilecek yavruların olası konumları.....	29
Şekil 2.17 : Mutasyonun etkisi.....	30
Şekil 2.18 : Bireylerin rulet çarkında kapladığı alanlar.....	32
Şekil 2.19 : Rulet çarkında kaplanacak alanlar.....	36
Şekil 3.1 : Küresel elektrot sistemi.....	39
Şekil 3.2 : Küresel elektrot sisteminde α ve $E_{\max}$ 'ın r_1 ile değişimi.....	42
Şekil 3.3 : Silindirselsel elektrot sistemi.....	43
Şekil 3.4 : Silindirselsel elektrot sisteminde α ve $E_{\max}$ 'ın r_1 ile değişimi.....	47
Şekil 3.5 : Delinme geriliminin p_a ile değişimi.....	48
Şekil 3.6 : Delinme geriliminin basınç-elektrot açıklığı ile değişimi	50
Şekil 3.7 : Çift ekponansiyel darbe akımı.....	52
Şekil 3.8 : Dikdörtgen darbe akımı.....	52
Şekil 3.9 : Doğru bileşensiz sinüzoidal darbe akımı.....	53
Şekil 3.10 : Doğru bileşeni ekponansiyel olarak azalan darbe akımı.....	53
Şekil 3.11 : Yüksek akım generatörü yapısı.....	54
Şekil 3.12 : Darbe üretici eşdeğer devresi.....	54
Şekil B.1 : Küresel elektrot sistemi maksimum alan için fonsiyon değeri- nesil değişimi.....	68
Şekil B.2 : Küresel elektrot sistemi maksimum alan için fonsiyon değeri- nesil değişimi.....	68
Şekil B.3 : Küresel elektrot sisteminde α fonsiyon değeri - nesil değişimi..	69
Şekil B.4 : Silindirselsel elektrot sistemi maximum alan için fonsiyon değeri- nesil değişimi.....	70
Şekil B.5 : Silindirselsel elektrot sistemi minimum alan için fonsiyon değeri-	

	nesil deęiřimi.....	71
řekil B.6	: Silindirsel elektrot sisteminde α fonsksiyon deęeri - nesil deęiřimi.....	72
řekil B.7	: Geometrik karakteristikte aęırlık-nesil sayısı deęiřimi.....	73
řekil B.8	: Delinme gerilimi ortalama deęerinin nesil sayısı ile deęiřimi.....	74
řekil B.9	: İletken yarı kre przde fonksiyon(E) ortalama deęeri - nesil deęiřimi.....	75
řekil B.10	: Yalıtkan yarı kre przde fonksiyon(E) ortalama deęeri-nesil sayısı deęiřimi.....	76
řekil B.11	: Dik elipsoidal przde fonksiyon(E) ortalama deęeri-nesil deęiřimi.....	77
řekil B.12	: Darbe akım fonksiyonu ortalaması-nesil sayısı deęiřimi.....	78

SEMBOL LİSTESİ

A, B	: İntegral sabitleri
A, B	: Paschen sabitleri
a	: Problem sınır aralığının bölüm sayısı
a	: Elektrotlar arası açıklık
Alfa	: Ölçek katsayısı
b,c,F	: Elips parametreleri
C	: Darbe kondansatörü
c_k , c_{k1} , c_{k2}	: Çaprazlama konumu
d	: Ara çaprazlama parametresi
E	: Elektriksel alan şiddeti
E_d	: Delinme dayanımı
E_{max}	: Maksimum elektriksel alan şiddeti
E_{min}	: Minimum elektriksel alan şiddeti
E₀	: Başlangıç elektriksel alan
G	: Yalıtkan malzeme ağırlığı
I	: Akım
I_m	: Maksimum akım
K	: Sabit
krom_aralik	: Bireyin verilen problem aralığındaki değeri
krom_max	: Uygunluğu en iyi olan birey
krom_10	: Onluk taban düzenindeki birey
l	: Silindirsel elektrot uzunluğu
mask1	: Çaprazlama işlemi için yardımcı birey
mask2	: Çaprazlama işlemi için yardımcı birey
max	: Problem üst sınırı
min	: Problem alt sınırı
mutasyon_bit	: Mutasyonun meydana geldiği bit
mutasyon_krom	: Mutasyonun meydana geldiği birey
mutasyon_mask	: Mutasyon için yardımcı birey
n_{bit}	: Birey uzunluğu
nesil	: Nesil sayısı
n_pop	: Topluluk birey sayısı
R	: Küre çapı
r	: Küresel/silindirsel elektrot değişken çapı
r	: 0-1 arası rastgele üretilen sayı
r₁	: İç küre çapı, iç silindir çapı
r₂	: Dış küre çapı, dış silindir çapı
p	: Geometrik karakteristik
pa_{kr}	: Kritik basınç-aralık çarpımı
p_{birikim}	: Birikimli olasılık
p_c	: Çaprazlama olasılığı
p_i	: i. bireyin seçilme olasılığı

$P_{\text{kayıp}}$	: Kayıp güç
$P_{\text{seçilme}}$	: Seçilme olasılığı
P_m	: Mutasyon olasılığı
pop_size	: Topluluk büyüklüğü
POS	: Pozisyon
SP	: Seçilme baskısı
t	: Zaman
toplam_uygunluk	: Topluluğun uygunluk değerlerinin toplamı
U	: Elektrotlara uygulanan gerilim
u_i	: i. bireyin uygunluğu
U_{dmin}	: Minimum delinme gerilimi
$\text{uygunluk}[i]$	: i. bireyin uygunluk değeri
V_1	: İç küre gerilimi, iç silindir gerilimi
V_2	: Dış küre gerilimi, dış silindir gerilimi
Z	: Empedans
x	: Bireyin verilen aralıktaki değeri
x^I	: Onluk tabandaki birey
$x^I_{\text{üst sınır}}$	: Üst sınırın onluk tabandaki karşılığı
Σu	: Toplam uygunluk
γ	: Towsend'in 2. iyonlaştırma katsayısı
α	: Towsend'in 1. iyonlaştırma katsayısı
αa	: Elektronun toplam iyonlaştırma katsayısı
ϵ	: Dielektrik katsayısı
ϵ_0	: Dielektrik katsayısı
π	: Pi sayısı

1. GİRİŞ

Optimizasyon en genel anlamıyla, karışık problemlerin çözümlenmesinde izlenecek yolu tanımlayan süreçtir. Bu süreç, birbirleriyle bağlantılı olan birçok değişkeni değerlendirmeyi ve bunların sonuca yansımalarının kontrolünü gerektirir. Genellikle, incelenecek problemin tipine bağlı olarak bir amaç fonksiyonun minimumunu veya maksimumunu aramak suretiyle optimizasyon gerçekleştirilir.

Geçmişten günümüze, teoride ve pratikte karşılaşılan problemlere bir çözüm getirmek amacıyla, bir çok optimizasyon yöntemi kullanılmıştır. Bu optimizasyon yöntemlerinden birisi de temelleri 1960'lı yıllarda atılan Genetik Algoritmalarıdır.

Genetik Algoritmalar (GA), doğadaki evrim kurallarının yapay sistemlere uygulanması ile ortaya çıkan ve rastlantısal geçiş kurallarını araç olarak kullanan optimizasyon yöntemidir. İncelenen problemlerde karşılaşılan ve diğer optimizasyon yöntemlerinde sorun yaratabilen süreklilik, türev varlığı, tek bir fazla optimuma sahip olma, lineer olup olmama gibi kısıtlamalar GA için sorun teşkil etmemektedir. GA'lar kullanıcıya sağladığı bu avantajları sayesinde kendisine kısa sürede geniş bir kullanım alanı yaratmıştır. Geçmişten günümüze yapılan çalışmalar incelendiğinde mühendisliğin çeşitli alanlarında uygulamaları bulunan genetik algoritmaların yüksek gerilim tekniğinde uygulamalarına rastlanmamıştır.

Yüksek gerilim tekniğinde, boşalma olaylarını ve yalıtım problemlerini inceleyebilmek için öncelikle statik elektrik alanı bilmek gerekir. Küresel ve silindirselsel elektrot sistemlerinin tasarım ve hesaplamalarında, bu sistemleri zorlayacak maksimum ve minimum elektriksel alanları bilinmelidir. Ayrıca bu sistemlerin tasarımında, maliyeti en aza indirmek, istenilen diğer bir koşul olarak önümüze çıkmaktadır. Sistem geometrisinde bulunabilecek pürüzlerin ve basınç-elektrot açıklığı gibi değişkenlerin elektriksel alanları değiştirebileceği konusu da göz ardı edilmemelidir.

Yüksek gerilimlere bağı olarak ortaya çıkan darbe akımları sistemlerde aşırı sıcaklıklara ve mekanik kuvvetlere neden olurlar. Bu etkilerde sistemin dayanımını zorlamaktadır. Darbe dayanım deneylerinde akım darbelerinin tepe değerlerinin ve bu değere ulaşma sürelerinin hesaplanması gerekir.

Yukarıda sözü edilen yüksek gerilim uygulamaları problemlerinin her biri diğer yöntemlerde olduğu gibi GA'lar ile de çözümlenebilir. Böylece yüksek gerilim tekniğı literatüründe pek fazla yer bulamamış olan GA için yeni bir kullanım alanı yaratılmasına yardımcı olunabilir.

Bu çalışmada, yüksek gerilim tekniğindeki tek amaç fonksiyonlu tek değişkenli uygulamaların bazılarına farklı bir bakış açısı getirmek amacıyla Visual Basic programlama dilinde özgün GA programı yazılmış ve program sonucunda elde edilen veriler ile analitik yolla hesaplanan veriler karşılaştırılarak yöntemin doğruluğı sınanmıştır.

2. GENETİK ALGORİTMALAR

2.1 Genetik Algoritmaların Tarihsel Gelişimi

Yaşayan organizmalar eksiksiz birer problem çözücüdürler ve en iyi bilgisayar programının bile gösteremeyeceği bir çok yönlülük ortaya koyarlar. Bu gözlem, özellikle algoritmalar konusuna yıllarını veren bilgisayar uzmanlarının hayal kırıklığına uğramasına ve sinirlenmelerine neden oluyordu [18].

Pragmatik araştırmacılar evrimsel gelişmenin belirgin gücünü algılayarak onu çekememek yerine, yapay sistemlerde bu özelliği kullanma yolunu tercih ettiler. Doğal seçim, yazılım için en büyük problemlerden birini eliyordu: problemin tüm özelliklerini tanımlamak ve programı onlarla bağlantı kurabilmesi için harekete geçirmek. Böylece, yapısı hiç kimse tarafından tam olarak anlaşılmassa bile evrim mekanizması yardımıyla, başlangıcı olan herhangi bir program çözülebiliyordu. Bu -sözüm ona genetik algoritmalar (GA) - jet uçakları gibi karmaşık sistemlerin tasarımlarında ilerlemeler yapmak için uygun gözüküyordu [18].

Modern GA'ların temeli 1960'lı yıllarda evrim stratejileri adı altında Alman bilim adamı Rechenberg tarafından atıldı ve uçakların aerodinamik profilinin tasarımında denendi. Bu denemede amaç fonksiyonu kullanılarak uçak kanadının biçim optimizasyonu mekanizması bulunmuş oldu ve bu çalışma Schwefel tarafından devam ettirildi. Evrim stratejileri fikriyle iki sonuç ortaya çıkmıştı: ebeveyn ve ebeveynin başkalaşması ile ortaya çıkan yavru.

Evrimsel programlama Fogel, Owens ve Wals tarafından geliştirilen bir tekniktir ve modern GA'lara oldukça benzemektedir. Evrimsel programlamanın genetik algoritmalarından farkı, bir sonraki nesli oluştururken yalnızca mutasyon operatörünü kullanmasıdır.

Doğadaki birçok organizma ilk olarak iki süreçten geçer: doğal seçim ve cinsel üreme. İlki toplumun üyelerinden hangilerinin yaşayacağını ve üreme işlemine gireceğini belirler, ikincisi ise üreme sonunda oluşacak yavruların genlerinin hangi ebeveynlerden geleceğini belirleyerek üremeyi gerçekleştirir. Sonuç olarak oluşacak yavru ebeveynlerinin genlerinin karışımıdır. Bu karışım, oluşacak yavruların tek bir ebeveynin genlerinin aynısını taşıyan ve sadece mutasyonla değişen bir yavrudan daha hızlı bir şekilde evrimleşmesine neden olur (tek hücreli canlıların genleri çiftleşme ile birbirine geçmemesine rağmen genetik başkalaşım geçirirler ve tek hücreli organizmaların başkalaşimleri biyolojide aynı işlevi gören ancak farklı evrim kökenli terimi (analogous terms) ile ifade edilir) [1].

Seçme işlemi basitti, eğer bir organizma uygunluk testinde başarısız olursa ölüyordu. Benzer şekilde bilgisayar uzmanları da algoritmaların performanslarını zayıflatan sorunları ayıklamalıydılar. Örneğin, eğer çeşitli sayıları artan bir şekilde sıralayacak bir program tasarlanmışsa, girişteki sayının çıkıştan büyük olup olmadığı kontrol edilip, büyükse çıkışa gönderilmesi, değilse gönderilmemesi gerekiyordu.

İnsanlar en iyi ürünleri, yarış atlarını veya süs bitkilerini yetiştirmek için melezleme ve seçme süreçleri üzerinde çalıştılar. Tüm bunları bilgisayar programlarına aktarmak görüldüğü kadar kolay bir iş değildi. Herşeyden önce bilgisayar programında herhangi bir canlının DNA'sını temsil edecek bir genetik koda ihtiyaç vardı [18].

Genetik algoritmaların bilgisayar ile buluşmasındaki ilk denemeler 1960'lı yıllarda meydana geldi fakat başarılı olamadı, çünkü o dönemin biyolojik metinlerini takip ediyorlardı ve yeni gen kombinasyonlarının üretilmesinde mutasyona çiftleşmeden daha çok güveniyorlardı. Dolayısıyla denemeler sonucu alınan veriler tatmin edici değildi. Daha sonraları, Berkeley'deki Kaliforniya Üniversitesinden Hans J. Bremermann çiftleşme olgusuna “yavruların karakteristikleri iki ebeveynin genlerine bağlı olarak belirlenir” eklemesinde bulundu.

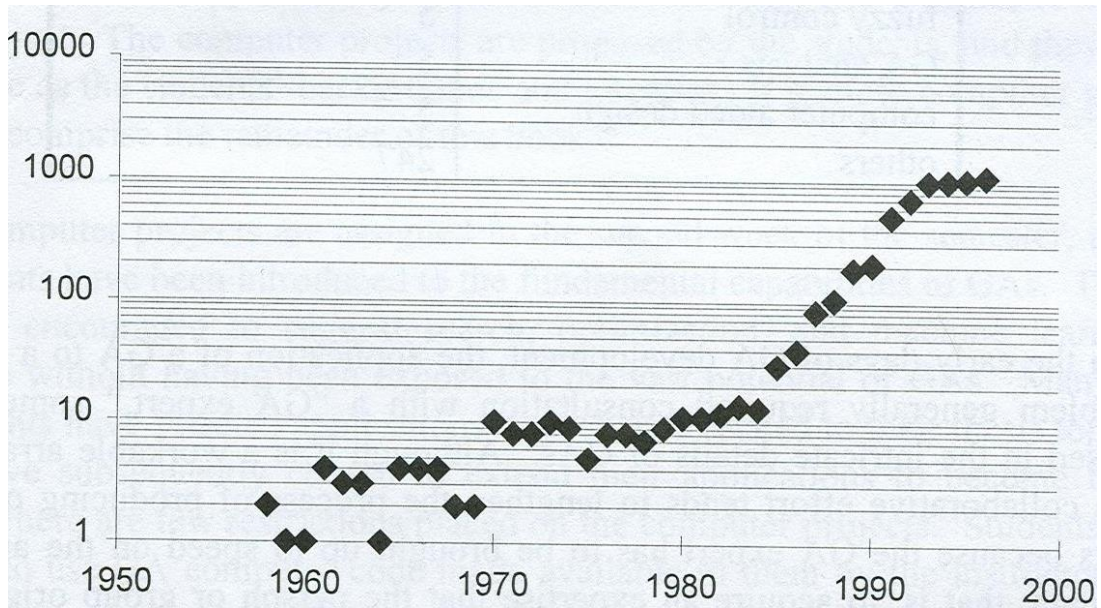
Aynı dönemde genetik algoritmaların kurucusu sayılan John H. Holland uyum sürecinin matematiksel analizleri üzerinde incelemelerde bulunuyordu ve gen guruplarının rekombinasyonunun (yeniden bir araya gelmesinin) uyum sürecinin önemli bir kısmını oluşturduğunu anlamıştı. Holland 1960'ların ortalarında çiftleşme

ve mutasyon işlemleri ile evrim süreci için uyumlu duruma gelen genetik algoritmalar tekniğini geliştirdi. Bunu izleyen yıllarda ise genetik algoritmaların bilgisayar programlarında temsil edileceği genetik kodlama yöntemi üzerinde çalışmalar yaptı. Holland'ın 1975 yılında yayınladığı “Doğal ve Yapay Sistemlerin Uyumu” adlı kitabı mutasyon, seçim, çiftleşme gibi genetik algoritmaların temel operatörlerinin birarada kullanıldığı ilk kitaptı [3].

David Goldberg çeşitli türdeki problemlerin çözümünde genetik algoritmaların kullanımı üzerinde araştırmalar yaparak çalışmalarını 1989 yılında “Araştırma, Optimizasyon ve Makine Öğretisinde Genetik Algoritmalar” adlı kitabında topladı. Kitapta kullanılan Pascal tabanlı kodlama daha sonraki birçok araştırmacı için yol gösterici oldu ve onların işlerini kolaylaştırdı [2]

İlerleyen yıllarda genetik algoritmalar ile uğraşanların sayısı ve basılan yayınların sayısı gittikçe arttı (Tablo 2.1). Bunun sonucunda endüstriyel alanlardaki problemlerin çözümünde genetik algoritmalar kullanılmaya başladı. 1985 yılında başlayan genetik algoritmaları konu alan ICGA, PPSN, ICEC, ANN&GA, FOGA, EP, COGANN, EC, GP ve SEAL gibi konferanslar halen yapılmaya devam etmektedir [2,5].

Tablo 2.1 Çeşitli yıllarda basılan GA yayınlarının sayısı [2]



2.2 Biyolojik Temeller

Doğal seçim ilkesinin temelleri genetik olayların keşfinden uzun zaman önce Charles Darwin tarafından ortaya atıldı ve formül haline getirildi. Darwin, kalıtımın karıştığını ve kaynaştığını varsaydı. Buradan hareketle ebeveynlerdeki özelliklerin yavrularında -aynı sınıfların birbiriyle karışması gibi- karıştığını farzettii.

Mendel yasaları, ancak 1900 yılında H. de Vries, K. von Tschermak tarafından tekrar çözümlendikten sonra bilim dünyasında yerini alabildi. Genetik bilgisi, kalıtım bilgisinin kromozomlar tarafından taşındığını ve kromozomların genlerden oluştuğunu deneysel çalışmalarla söyleyen T. Morgan tarafından geliştirilmiştir. Daha sonraları yapılan çalışmalar sonucunda da Mendel yasalarının tüm cinsel (birleşmeyle) üreyen organizmalar için geçerli olduğunu göstermiştir.

Mendel yasaları tekrar çözümlendikten sonra bile Darwin'in doğal seçim teorisi ile birbirinden bağımsız, ayrık iki konu gibi düşünülmüş, hatta bu iki olgu birbirine karşıymış gibi gösterilmişti. 1920'lerde Cetverikov, Mendel'in genetiğı ile Darwin'in doğal seçim ilkesinin birbiriyle çatışmadığını ve bu iki olgunun uyum içerisinde birleşerek modern evrim teorisini oluşturduğunu kanıtladı.

Modern evrim teorisini örnekle açıklamak amacıyla tavşanlar ele alınabilir. Herhangi bir andaki tavşan topluluğunu oluşturan tavşanlardan (bireylerden) bir kısmı diğerlerinden daha hızlı ve daha akıllıdır. Bu hızlı ve akıllı tavşanların tilkiler tarafından yakalanıp yenme şansı daha azdır, dolayısıyla yaşamlarını devam ettirebilirler. Ayrıca daha yavaş ve daha az akıllı olan tavşanlardan da bazıları şanslı oldukları için yaşamda kalırlar. Sözü edilen bireyleri kapsayan bu topluluk zamanla üremeye başlar. Üreme sonucunda genetik bakımdan daha iyi koşullara sahip bir karışım ortaya çıkar. Sonuç olarak yavru tavşanlar bir önceki topluluktan daha akıllı ve daha hızlı olurlar. Bu özelliklere sahip tavşanların tilkilere yem olmama ve yaşama şansı daha yüksektir. Aynı başkalaşımı tilkilerin de geçireceğini gözardı etmemek gerekir, aksi halde tilkiler hiçbir zaman tavşanları yakalayamazlardı.

2.2.1 Temel Tanımlar

Genetik biliminde kullanılan ve ilerdeki bölümlerde sözü edilecek bazı kavramlar ve terimler aşağıda açıklanmıştır.

Kromozom: Birbiri ardından gelen nesiller arasındaki benzerliği oluşturan ve biyolojik çeşitliliği sağlayan genlerin bulunduğu yapıdır. Genetik algoritmalarda incelenecek problemin her bir çözümünü birey (kromozom) temsil eder.

Lokus: Kromozomlar üzerinde genlerin bulunduğu yere verilen isimdir.

Gen: Kromozomlar üzerinde bulunan ve genetik bilginin taşınmasını sağlayan birimlerdir.

Allel: Bir genin tek bir lokustaki farklı formlarından herbiridir.

Fenotip: Bir organizmanın dış görünüşünü ve durumunu veren yapıdır. Örneğin bir bireyin şekli, büyüklüğü, dış ve iç yapısı, rengi, kimyasal bileşimi, davranışı gibi özelliklerinin toplamı onun fenotipidir. Fenotip, genotip ile çevre koşullarının karşılıklı ilişkileri sonucu oluşur. Genetik algoritma problemlerinde her bir bireyin temsil ettiği verilen çözüm aralığındaki değerdir.

Genotip: Bir organizmanın sahip olduğu genlerin toplamına onun genotipi denir. Bunlardan bir kısmı (dominant (baskın) genler) kendi varlığını belli ederken, diğer kısmının (resesif (çekinik) genlerin) etkisi gizli kalır. Her bir bireyin temsil ettiği ve verilen çözüm kümesine indirgenmeden önceki değerdir.

Çaprazlama: Mayoz bölünmenin bir aşamasında kromozomlar arasında kırılmalar ve yeniden birleşmeler yoluyla genetik bilginin değiş tokuşudur.

Mutasyon: Bir organizmanın gen veya kromozomlarında fiziksel veya kimyasal kökenli etmenlerin neden olduğu ani yapısal veya sayısal değişikliklerdir.

Rekombinasyon: Çaprazlama sonucu ebeveynlerden farklı olarak ortaya çıkan yeni gen veya kromozom çeşitlilikleridir.

2.3 Genetik Algoritmalar Nedir ?

Doğadaki evrim kuramının yapay sistemlere uygulanması sürecinde evrimsel hesaplamalar başlığı altında, genetik algoritmalar (GA), evrimsel programlama (EP) ve evrimsel stratejiler (ES) olmak üzere üç yöntem ortaya çıkmıştır [2,5] Bunlar arasında günümüzde en çok bilinen ve en çok kullanılan yöntem genetik

algoritmalarıdır. Az önceki açıklamalardan da anlaşılacağı gibi genetik algoritmalar, doğal seçim ve doğal kalıtım kurallarına dayanır. Doğal seçim, çevre koşullarına uyum sağlamış olan canlıların yaşamlarını sürdürmesi, uyum sağlayamamış olanların ise elenmesi yani ölmesi anlamına gelir. Doğal kalıtım ise, seçilmiş olan bireylerin birbirleriyle çiftleşerek yeni yavrular meydana getirmesi olarak kısaca özetlenebilir.

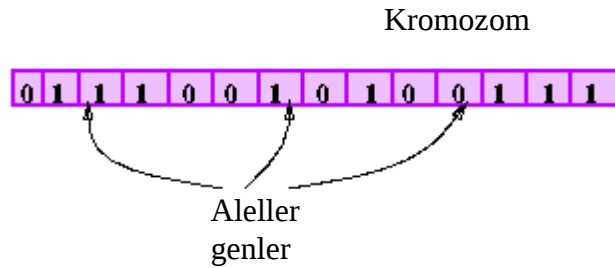
Genetik algoritmalarla problem çözümünde ilk topluluğu oluşturacak n adet birey tamamen rastlantısal belirlenir. Daha sonraki aşamalar ise iki çatı altında toplanabilir:

a) Genetik işlemler: çaprazlama ve mutasyon

b) Evrimsel işlem: seçim

Problem çözümüne noktalar topluluğundan başlayan genetik algoritmalar öncelikle bu noktaları kodlarlar. Farklı tipte kodlama sistemleri olmakla birlikte genelde ikili kodlama kullanılır. Daha sonraki bölümlerde (2.7.1) kodlama sistemleri ayrıntılı olarak ele alınacaktır. Çözüm kümesini (topluluğu) oluşturan her bir birey “kromozom” veya “birey” olarak adlandırılır. Kromozomlar gen adı verilen sembol katarlarından oluşur. Örneğin Şekil 2.1’de ikili kodlama yapısına sahip 0 ve 1 sembollerinden oluşan bir “birey” görülüyor.

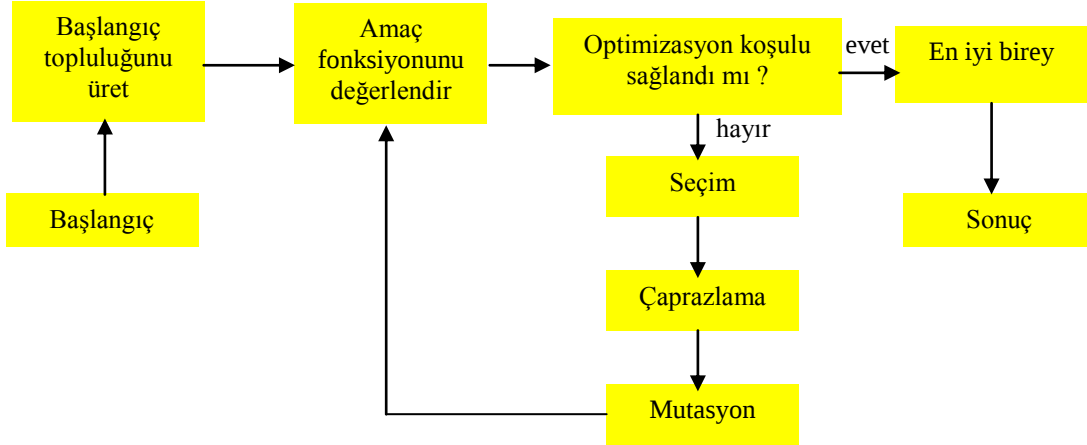
Kodlama bilgisi:



Şekil 2.1 İkili kodlama yapısındaki birey

Topluluğu oluşturan her bir bireyin temsil ettiği değer “amaç fonksiyonu”nda (objective function) yerine konularak bireylerin “uygunluk değerleri” (fitness value) yani bir sonraki nesilde yaşama olasılıkları bulunur. Bir sonraki adım, uygunluk değerleri gözönünde bulundurularak bir sonraki topluluğu meydana getirecek “ebeveynler” seçilmesidir. Önceden belirtilen oranda ebeveyn çaprazlamaya girerek birbirleriyle gen alış verişinde bulunurlar ve ortaya bir sonraki toplumun bireylerini oluşturacak olan “yavrular” çıkar. Yine önceden tanımlanmış mutasyon oranları

nedeniyle ortaya çıkan yavrularda başkalaşım lar meydana gelir. Bu aşamadan sonra yeni topluluğu meydana getirecek bireyler hazır dır. Bu dö ngü istenilen koş ullar sağ lanıncaya kadar devam eder ve her bir yin eleme iş lemi “ nesil” adını alır. Bu dö ngü Ş ekil 2.2’ de ve Tablo 2.2’ de gösterilmiştir. Probleme baş larken kullanılacak topluluk rastlantısal olarak üretilir. Genetik algoritmalar rastlantısal kuralları kullanarak gittikçe daha iyi bir çözüme yönlendirler.



Ş ekil 2.2 Genetik algoritma programı blok ş eması

Tablo 2.2 Genetik algoritmanın akış tablosu

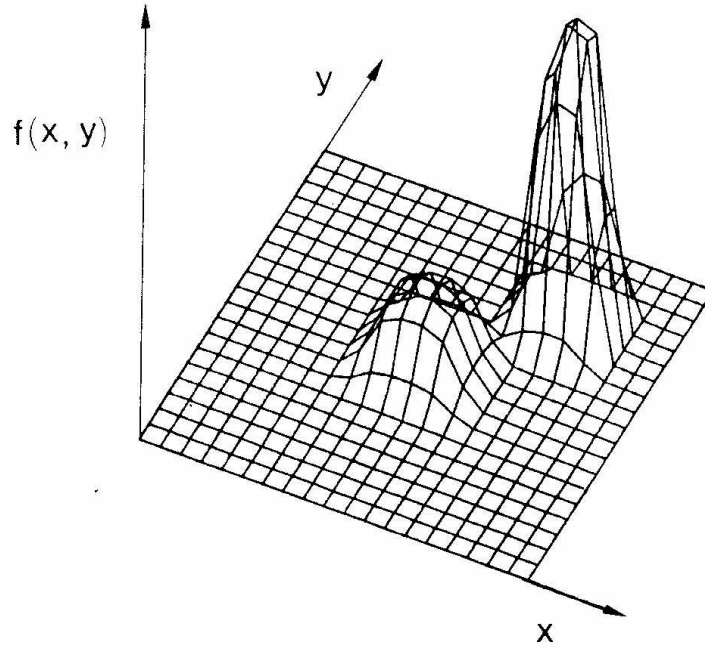
1. *[Başlama]* n bireyli bir topluluğu rastlantısal olarak üret (problem için uygun çözü m)
2. *[Uygunluk]* Topluluktaki herbir bireyin uygunluk değ erini kaydet.
3. *[Yeni topluluk]* Yeni topluluk tamamlanana kadar aş ağıdaki dö ngüye bağı lı olarak yeni topluluğu yarat.
 1. *[Seçim]* Uygunluk değ erlerine göre topluluk iç inden iki ebeveyn birey seç.
 2. *[Çaprazlama]* Çaprazlama olasılıklarına bağı lı olarak yeni yavru oluşturacak ebeveynleri çaprazlamaya sok. Eđer çaprazlama olmayacaksa yavru ebeveynlerinin tamamen aynısıdır.
 3. *[Mutasyon]* Mutasyon olasılığına bağı lı olarak mutasyonu gerçekleştir.

4. *[Kabul]* Oluşan yavruları yeni topluluğa yerleştir
5. *[Yerleşim]* Yeni üretilen topluluğu algoritmanın tekrar işlemesi için kullanıma hazırla.
6. *[Sınama]* Eğer istenilen koşullar sağlanmışsa, *dur*, ve o anki topluluğun en iyi çözümünü döndür.
7. *[Döngü]* Adım 2'ye dön.

2.4 Genetik Algoritmaların Diğer Optimizasyon Yöntemleri ile Karşılaştırılması

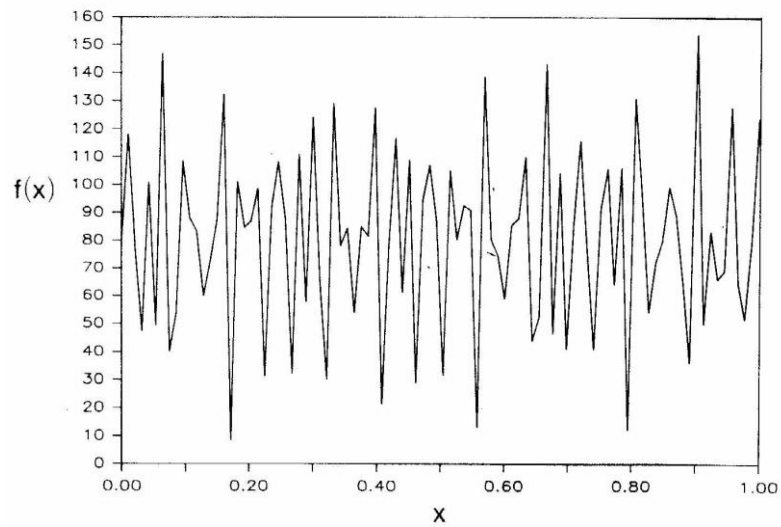
Literatürde tanımlanan üç tip araştırma yöntemi vardır: hesap tabanlı (calculus-based), listeli (enumerative) ve rastlantısal (random) [5].

Hesap tabanlı yöntemler sonuca yavaş bir biçimde yaklaşırlar. Doğrudan ve dolaylı olmak üzere kendi içerisinde iki alt guruba ayrılırlar. Dolaylı yöntemler, amaç fonksiyonunun türevini sıfıra eşitleyerek elde edilen ve lineer olmayan eşitlikleri çözerek yerel optimum noktalarına ulaşırlar. Doğrudan yöntemler ise yerel türevlerin değerlerine bağlı olarak uygun yönlerde tahmini ilerlemeler yaparak sonuca giderler. Her iki yöntem de yinelemeli (iteratif) yaklaşım tekniğini kullanır. Seçilen herhangi bir noktadan başlayan yineleme (iterasyon), bu nokta etrafındaki “komşu noktaları” değerlendirir ve amaç fonksiyonunda en iyi değeri (minimum problemleri için en küçük, maksimum problemleri için en büyük değeri) sağlayan nokta bulunur. Bulunan bu nokta yeni başlangıç noktasıdır ve aynı işlemler bu noktadan başlayarak tekrar edilir. Ulaşılan son noktadan daha ileriye gidilemeyinceye kadar işlemler devam ettirilir ve bulunan bu nokta yerel optimum noktası olarak adlandırılır. Görüleceği gibi bu yöntem yalnızca yerel optimumu bulmaya yöneliktir ve fonksiyonun genel optimum noktası hakkında sağlıklı bilgi vermez. Ayrıca, birden fazla yerel optimum noktasına sahip fonksiyonlarda (Şekil 2.3) hangi optimum noktasına ulaşılabileceği, yinelemede (iterasyonda) başlangıç noktasının seçimine bağlıdır.



Şekil 2.3 Çok optimum noktalı fonksiyon [3]

Hesap tabanlı yöntemler türevlerin varlığına bağlıdır. Gerçek dünyada yapılan birçok incelemede ise karşımıza sürekli olmayan, birden çok optimumu olan ve parazitlerin karıştığı fonksiyonlar çıkmaktadır (Şekil 2.4). Sonuç olarak süreksiz fonksiyonların türevlerinin alınamaması ve yerel optimum noktasının bulunması nedeniyle hesap tabanlı yöntemlerle inceleme yapmak yetersiz kalmaktadır.

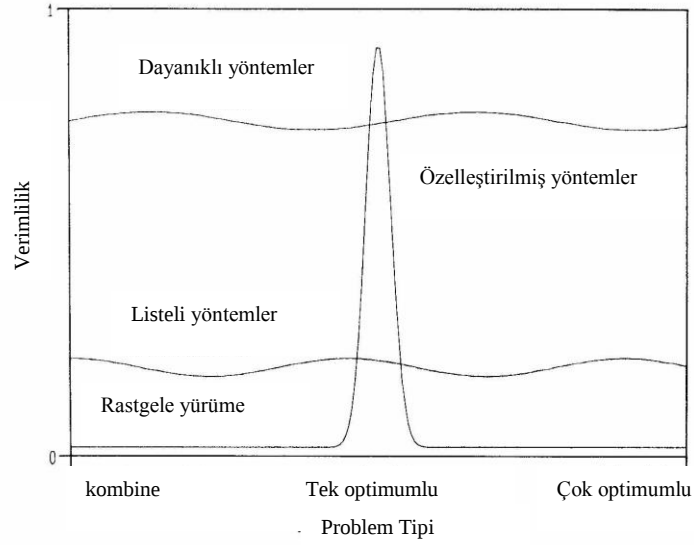


Şekil 2.4 Parazitlerin karıştığı fonksiyon [3]

Listeli (Enumerative) yöntem, sonlu veya ayrıklaştırılabilir sonsuz çözüm uzayının herbir noktasını eş zamanlı olarak değerlendirir. Bu yöntem çözüm kümesinin küçük olduğu problemler için gayet basit ve kullanışlı bir yöntemdir fakat uygulamada karşılaşılan büyük çözüm kümelerine sahip problemlerde birçok noktanın aynı anda değerlendirilmesi, yöntemin veriminin önemli ölçüde düşmesine neden olmaktadır. Listeli yöntemin en ilgi çekici dalı olan dinamik programlama bile orta büyüklükteki ve karmaşıklıkta problemlerin çözümünde başarı sağlayamamaktadır. Yöntemin bu özelliği, mucidi Bellman tarafından “boyutsallığın laneti” olarak ifade edilmiştir [5].

Hesap tabanlı ve listeli yöntemlerin eksikliklerinin ortaya çıkması rastlantısal yaklaşım algoritmalarına gösterilen ilginin artmasına neden olmuştur. Rastlantısal algoritmaların uzun süre çalışması durumunda alınacak sonuçlar listeli yaklaşımdan alınabilecek sonuçlardan daha iyi değildir. Burada dikkat edilecek nokta, tamamen rastlantısal arama (strictly random search methods) yöntemleri ile rastlantısal noktaları değerlendirmek suretiyle ilerleyen rastlantısal tekniklerin birbirine karıştırılmamasıdır. Örneğin genetik algoritmalar -bu gibi rastgele arama yöntemlerinden farklı olarak- rastlantısallığı bir araç olarak kullanırlar. Rastlantısallığı bir araç olarak kullanmak başlangıçta tuhaf gibi görünse de, doğa kendi içerisinde buna benzer pekçok örneği barındırmaktadır.

Sonuç olarak geleneksel yöntemler dayanıklı (robust) değildir fakat bu onların kullanışlı olmadığı anlamına gelmez. Bu yöntemler birçok uygulamada birlikte (hibrit) kullanılabilirler. Şekil 2.5’de kullanılabilecek yöntemlerin verimlerinin çeşitli problem tiplerine göre değişimi gösterilmiştir. Türevsel teknikler dar bir problem sınıfı için iyi sonuç verir fakat diğer tip problemlerde verimi ciddi bir biçimde düşer (eğer kullanılabiliyorsa). Diğer yandan listeli yöntemler Şekil 2.5’ den görüldüğü gibi tüm problem tiplerinde hemen hemen aynı verimsizliği gösterirler. Problem tipine bağlı olmaksızın en iyi verimi sağlayan dayanıklı (robust) yöntemlerdir (problem tipine göre hibrit yaklaşımlar uygulanarak verim daha da arttırılabilir).



Şekil 2.5 Optimizasyon yöntemlerinin verimleri [3]

Genetik algoritmaları geleneksel optimizasyon yöntemlerinden farklı kılan başlıca nedenler;

- Genetik algoritmalar parametrelerin kendilerini değil, parametre takımının kodlanmış biçimini kullanırlar.
- GA'lar aramaya tek bir noktadan değil, bir noktalar topluluğundan başlarlar.
- GA'lar amaç fonksiyonunun (objective function) türevlerini ve bir takım ek bilgilerini değil, doğrudan amaç fonksiyonun kendisini kullanırlar.
- GA'lar deterministik kuralları değil, raslantısal geçiş kurallarını kullanırlar.

Genetik algoritmalar çözüme başlarken optimizasyon problemindeki parametrelerin - sonlu bir katar olarak- kodlanmış haline ihtiyaç duyar. Bu parametrelerin kodlanması daha sonraki bölümlerde de açıklanacağı gibi çeşitli şekillerde olabilir.

Aramaya tek bir noktadan değil de birçok noktadan başlamanın en büyük yararı yerel optimuma yakalanma olasılığını ortadan kaldırmasıdır. Geleneksel yöntemler aramaya tek bir noktadan başladıkları için ilk buldukları yerel optimum noktasında aramayı sonlandırırlar. Dolayısıyla birden çok optimuma sahip fonksiyonlar için

kullanışlı değildir. Genetik algoritmalar ise tüm yerel optimum noktaları bularak bunları karşılaştırır ve fonksiyonun evrensel optimum noktasına ulaşır.

Genetik algoritmalarda diğer yöntemlerde olduğu gibi türev bilgileri ve ek bilgilere ihtiyaç yoktur, yalnızca amaç fonksiyonu ve uygunluk değerleri yaklaşımın hangi yönde ilerleyeceğini belirler.

Rastlantısal geçiş kurallarına bağlı olarak çalıştığından model değiştirilmese bile her çalıştığında farklı çözümler elde edilir. Geleneksel yöntemler ise aynı değerler ile her çalıştığında aynı çözüme ulaşılır.

Ayrıca seçim, çaprazlama ve mutasyon aşamaları da genetik algoritmaları diğer yöntemlerden farklı kılar. Topluluktaki daha uygun bireylerin seçilerek daha sonraki nesillere aktarılması ve uygun olmayanların elenmesi, algoritmanın en iyi çözümü bulması için izlenmesi gereken bir süreçtir. Aynı şekilde çaprazlama sayesinde de bir önceki çözüm kümesindeki sonuçlar birbirleriyle karışarak bir sonraki nesilde daha uygun sonuçları oluştururlar. Mutasyon ise algoritmanın bir kısır döngüye girmemesi için güvenlik aracı görevi görür. Herhangi bir yerel optimum etrafında toplanıp kalan bireyler, mutasyon ile başkalaşım geçirerek diğer çözümlere yönelebilirler.

Özetle, genetik algoritmalar sözü edilen tüm bu özellikleri sayesinde diğer yöntemlerde karşımıza çıkan süreklilik, türev varlığı, tek bir optimuma sahip olma, lineer olup olmama gibi kısıtlamalardan kurtulmuş olur. Bu yöntemin eksikliği ise büyük çözüm kümeli problemlerde her bir çözüm aynı anda değerlendirileceğinden sonuca yaklaşım hızının yavaş kalmasıdır.

2.5 Uygulama Alanları

GA uygulamaları genel olarak üç gruba ayrılır: (1) optimizasyon, (2) sınıflandırılmış sistemlerin öğrenimi kapsamındaki makina öğretisi, (3) genetik programlama. Amaç fonksiyonu ile ifade edilen optimizasyon uygulamaları, arzu edilen herhangi bir sonuca ulaşmada yol gösterici konumundadırlar. Bu tip uygulamalara örnek olarak kar marjının en yüksek seviyeye çekilmesi veya herhangi bir hata teriminin minimuma getirilmesi verilebilir. İkinci gruba giren uygulamalar ise GA'lar, etkin çözüme ulaşmak için, makinaların işleyiş kurallarını düzenleyen

yapıyı oluşturur. Genetik programlama, yapılan bir işte kullanılan programın geliştirilmesi ve daha verimli bir hale getirilmesine olanak sağlar. Bu işlemi yaparken de uygunluk fonksiyonunu kullanır.

GA yaklaşımının endüstrideki uygulama alanları ise kısaca şöyledir:

2.5.1 Mühendislik: Elektrik enerji sistemleri [13,14], reaktif güç kompanzasyonu [20], bina tasarımı (kirişlerin büyüklükleri), çeşitli elektriksel tasarımlar, mekanik tasarımlar (ağırlık, hacim ve fiyat), proses kontrolü, ağ tasarımları (bilgisayar ağları).

2.5.2 Yönetim: Dağıtım, planlama, proje yönetimi, kurye yönlendirme (courier routing), paketleme, görev atama, zaman çizelgeleri.

2.5.3 Finans: Portfolyo dengelemesi, bütçeleme, tahmin, yatırım analizi ve ödeme takvimi.

2.5.4 Araştırma-Geliştirme (R & D): Eğri ve yüzey uydurma, yapay sinir ağları bağlantı matrisleri, fonksiyon optimizasyonu, bulanık mantık, topluluk modellemesi, moleküler modelleme ve ilaç hazırlama (drug tasarımı).

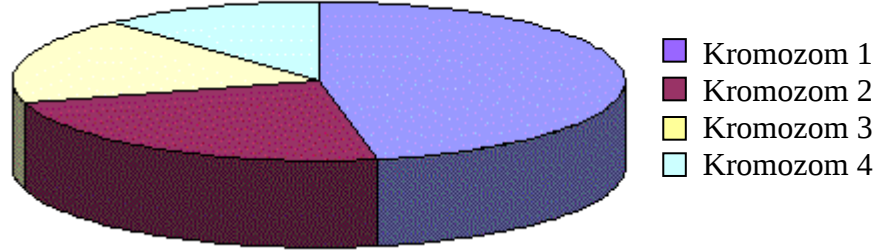
2.6 Seçim (Selection)

Bilindiği gibi çaprazlamaya girecek olan ebeveynler topluluğu (popülasyonu) oluşturan bireyler arasından seçilir. Buradaki temel problem çaprazlamaya girecek olan bireylerin seçiminin ne şekilde olacağıdır. Darwin'in evrim teorisine göre, daha uygun (fitness değeri yüksek) olan bireyler yaşamaya devam edip yavru oluşturabilirler. Bu ilke doğrultusunda, ebeveyn seçiminde rulet çarkı seçimi (roulette wheel selection), Boltzman seçimi, turnuva seçimi (tournament selection), rank seçimi, kararlı hal seçimi (steady state selection) gibi birçok yöntem kullanılabilir. Bu yöntemlerden bir kısmı aşağıda açıklanmıştır.

2.6.1 Rulet Çarkı Seçimi

Bu yöntemde ebeveynler uygunluk (fitness) değerlerine göre seçilirler, dolayısıyla uygunluk değeri yüksek olan bireylerin seçilme olasılığı daha fazla olacaktır. Şekil

2.6’da dört bireyden oluşan bir topluluğa ilişkin rulet çarkı görülmektedir. Uygunluk değeri yüksek olan bireylerin çarkta kaplayacağı alan daha geniş olacağından çark rastgele çevrildiğinde bu bireylerin seçilme olasılığı diğerlerinden yüksek olacaktır.



Şekil. 2.6 Rulet çarkı sistemi

Bu seçim aşağıdaki algoritma ile özetlenebilir;

1. *[Uygunluk]* Topluluğu oluşturan tüm bireylerin uygunluk değerleri bulunur.- U_i
2. *[Toplam]* Uygunluk değerleri toplanır ve bu değer toplam uygunluk değeri olarak adlandırılır. - ΣU .
3. *[Olasılık]* Her bir bireyin seçilme olasılığı bulunur. - $p_i = U_i / \Sigma U$
4. *[Birikimli Olasılık]* Her bir bireyin birikimli olasılıkları bulunur. P_{bi}

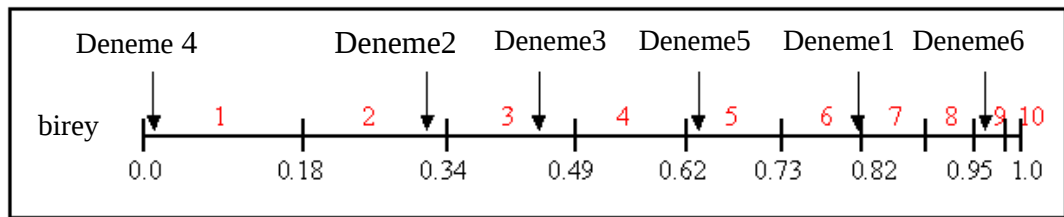
$$P_{bi} = \Sigma p_j \quad i = 1, 2, \dots, \text{pop_size}, j = 1, 2, \dots, i$$
5. *[Seçim]* (0,1) aralığında rastgele pop_size adet sayı üretilir - r .
6. *[Döngü]* Eğer $r \leq p_{b1}$ ise birinci bireyi ebeveyn olarak seç; diğer durumlarda üretilmiş olan r sayısı hangi birikimli olasılık aralığında ise $(p_{i-1} < r \leq p_i)$ i . bireyi seç ($2 \leq i \leq \text{pop_size}$).

Örneğin Tablo2.3’de uygunluk değerleri ve seçilme olasılıkları verilen onbir birey ele alınırsa;

Tablo.2.3 Seçilme olasılıkları ve uygunluk değerleri

Birey numarası	1	2	3	4	5	6	7	8	9	10	11
Uygunluk değeri	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
Seçilme olasılığı	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0

Altı adet rastgele sayı üretildiğinde: 0.81, 0.32, 0.96, 0.01, 0.65, 0.42.

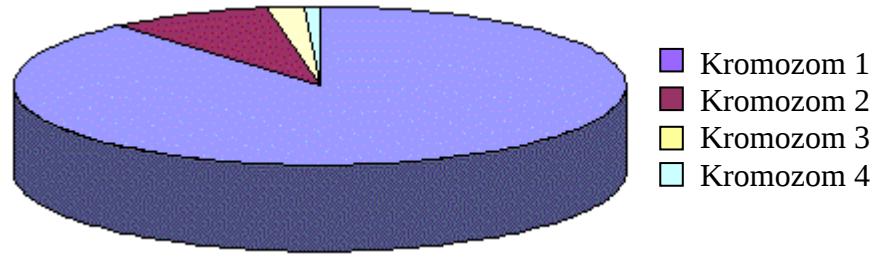


Şekil 2.7 Üretilen sayıların (deneme i) uygun düştüğü birey aralıkları

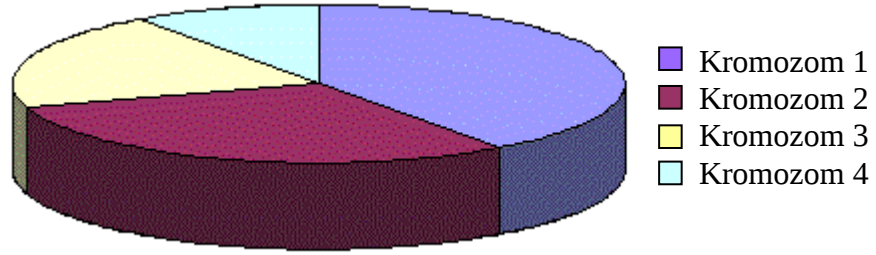
Seçim sonraki topluluğu oluşturan bireylerin 1, 2, 3, 5, 6, 9 numaralı bireyler olacaktır. Şekil 2.7’de görülmektedir.

2.6.2 Rank Seçimi

Uygunluk değerleri birbirinden çok farklı olan durumlar rulet çarkı seçimi için problem yaratır. Örneğin, en iyi bireyin uygunluk değeri 90% ise çark döndürüldüğünde diğer bireyler çok az oranda seçilecek ve algoritmanın kısır döngüye girme ihtimali doğacaktır. Rank seçimi öncelikle topluluğu sıralar ve diğer bireylerin uygunluk değerlerini bu sıralamaya göre düzenler. En kötü bireyin uygunluk değeri 1, daha az kötü olanın 2 ... ve en iyi bireyin uygunluk değeri n_{pop} olacaktır. Burada n_{pop} topluluğu oluşturan birey sayısını gösterir. Sıralama öncesi ve sonrası durumlar Şekil 2.8 ve Şekil 2.9 da görülmektedir.



Şekil.2.8 Sıralama öncesi durum (uygunluk değerlerine göre grafik)



Şekil.2.9 Sıralama sonrası durum

Topluluktaki birey sayısı $npop$, bireyin topluluktaki pozisyonu Pos (en az uygun birey için $Pos = 1$, en uygun birey için $Pos = npop$), seçilme baskısı parametresi SP olmak üzere her bir birey için uygunluk değeri aşağıdaki formüllerle hesaplanır.

Lineer rank seçimi için;

$$Uygunluk(Pos) = 2 - SP + 2 \cdot (SP - 1) \cdot (Pos - 1) / (npop - 1) \quad (2.1)$$

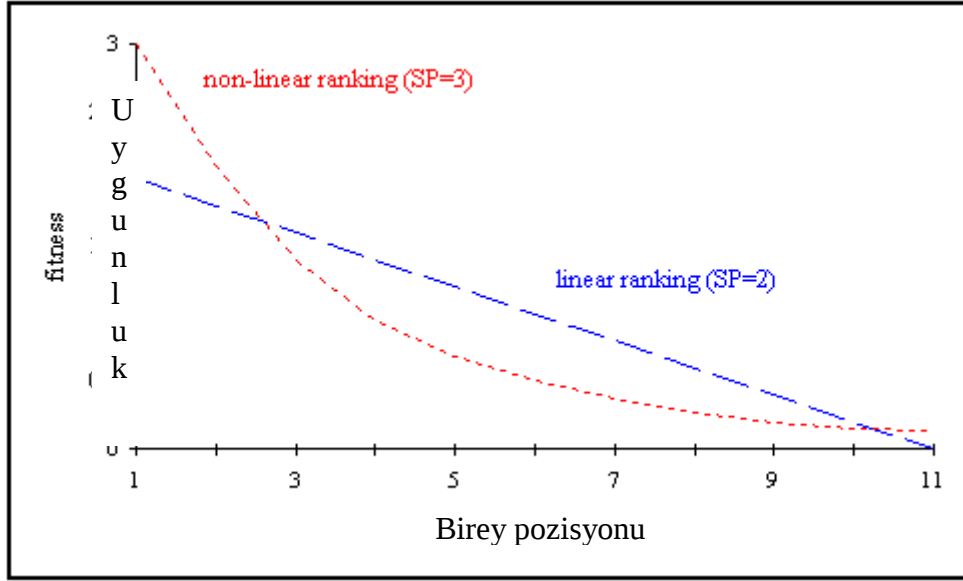
Lineer olmayan rank seçimi için;

$$Uygunluk(Pos) = npop \cdot X^{(Pos - 1)} / \sum(X^{(i - 1)}); i = 1:npop \quad (2.2)$$

Buradaki X ise aşağıdaki polinomun köküdür.

$$0 = (SP - 1) \cdot X^{(npop - 1)} + SP \cdot X^{(npop - 2)} + \dots + SP \cdot X + SP \quad (2.3)$$

Lineer rank seçimi için SP, [1.0, 2.0] aralığında iken lineer olmayan rank seçimi için bu aralık [1.0, npop-2.0] dır.



Şekil.2.10 Lineer ve lineer olmayan rank seçimlerinin karşılaştırılması [53]

Rulet çarkı seçiminde verilen örneğin aynısını rank seçimi için tekrar ele alırsak Tablo.2.4 yardımı ile seçilme baskısı parametresinin değeri azaldıkça bireylerin uygunluk değerlerinin birbirine yaklaştığı görülebilir.

Tablo.2.4 Uygunluk değerinin seçilme baskısı ile değişimi [53]

Amaç fonksiyonu	Uygunluk değerleri (parametre: seçilme baskısı)				
	Lineer rank		Rank yok	Lineer olmayan rank	
	2,0	1,1	1,0	3.0	2.0
1	2.0	1.10	1,0	3.00	2.00
3	1.8	1.08	1,0	2.21	1.69
4	1.6	1.06	1,0	1.62	1.43
7	1.4	1.04	1,0	1.99	1.21
8	1.2	1.02	1,0	0.88	1.03
9	1.0	1.00	1,0	0.65	0.87
10	0.8	0.98	1,0	0.48	0.74
15	0.6	0.96	1,0	0.35	0.62
20	0.4	0.94	1,0	0.26	0.53
30	0.2	0.92	1,0	0.19	0.45
95	0.0	0.90	1,0	0.14	0.38

Rank seçiminde tüm bireylerin seçilme olasılığı vardır fakat en iyi birey diğer bireylerden çok fazla farklılık göstermediğinden bu yöntem çok yavaş bir yaklaşımdır.

2.6.3 Kararlı Hal Seçimi

Bu yöntem ebeveyn seçimi için partikular değildir. Kararlı hal seçiminde uygunluk derecesi yüksek olan bireyler belirlenerek yavru oluşturmak üzere çaprazlamaya girerken, uygunluk derecesi düşük olan bireyler atılarak yerlerine yeni oluşan yavrular konulur.

2.6.4 Turnuva Seçimi

Bu yöntemde, bir grup birey rastlantısal olarak seçilir ve bu grup içerisindeki en iyi birey çiftleşmeye gönderilir. Gruptaki bireylerin sayısı, turnuva büyüklüğü (tournament size) olarak adlandırılır. En genel turnuva büyüklüğü 2'dir ve bu ikili turnuva olarak adlandırılır.

2.6.5 Seçkinlik (Elitism)

Çaprazlama sonucunda oluşan yavruların meydana getirdiği yeni toplulukta, geçmiş toplulukta bulunan ve uygunluk derecesi yüksek olan bireyler bulunmayabilir. Bu durumlarda seçkinlik (elitism) sıkça kullanılmaktadır. Bunun anlamı uygunluk derecesi en yüksek olan bireyin değişikliğe uğramadan, yani çaprazlama ve mutasyona uğramadan, yeni topluluğa doğrudan kopyalanmasıdır. Bu yöntem kullanıldığında genetik algoritmanın performansı daha yüksek olur ve sonuca daha hızlı yaklaşılr.

2.7 Genetik Algoritma İşlemcileri

2.7.1 Kodlama (Encoding)

Genetik algoritmaları kullanarak problem çözümüne başlanıldığında, bireylerin kodlanması sorunu karşımıza çıkar. Kodlama türleri, çalışılan problemin türüne bağlı olarak çeşitlilik gösterir.

2.7.1.1 İkili Kodlama (Binary Encoding)

En yaygın olarak kullanılan kodlama biçimidir. Bunun en önemli nedeni, genetik algoritmalar ile ilgili ilk çalışmalarda ikili kodlamanın kullanılmış olmasıdır. Bu tür kodlamada her bir birey 0 ve 1 bitlerinden oluşmuş bir katar olarak ifade edilir (Tablo 2.5).

Tablo 2.5 İkili kodlama kullanılarak oluşturulan bireyler

Kromozom A	101100101100101011100101
Kromozom B	111111100000110000011111

İkili kodlamada çok küçük değerli alellere rağmen çok çeşitli bireyler yaratmak mümkündür. Diğer bir yandan da bazı problemler için bu kodlama türü yeterince uygun (doğal) olmadığı için çaprazlama ve/veya mutasyon aşamasından sonra düzeltme yapılması gerekebilir. Knapsack problemlerinde olduğu gibi, değeri ve büyüklüğü sabit olan cisimlerin, hacmi ve boyutları belli olan uzay parçasına maksimum sayıda yerleştirilmesi problemi için bu kodlama tipi kullanılabilir.

2.7.1.2. Dizilim (Permütasyon) Kodlaması

Dizilim (permütasyon) kodlaması seyyar satıcı (travelling salesman problem) problemleri ve görev sıralama problemleri gibi sıralamayı gerektiren problemlerde kullanılır. Bu tür kodlamada her bir birey, sıralama gösteren sayılardan oluşan katar ile ifade edilir (Tablo 2.6).

Tablo 2.6 Dizilim kodlaması kullanılarak oluşturulan bireyler

Kromozom A	1 5 3 2 6 4 7 9 8
Kromozom B	8 5 6 7 2 3 1 4 9

Dizilim kodlaması yalnızca sıralama problemlerinde kullanışlıdır. Bu kodlamaya seyyar satıcı problemi örnek olarak verilebilir. Birçok şehre gitmesi gereken bir satıcının, katedeceği mesafenin en kısa olması için izlemesi gereken yol üzerindeki

şehirlerin sırası ne olmalıdır? Bu problem için birey kodu, satıcının ziyaret edeceği şehirlerin sırasını gösterir.

2.7.1.3. Reel Kodlama (Gerçek Kodlama)

Gerçek kodlama karışık (complicated) değerlerin (örneğin reel sayıların) bulunduğu problemlerde kullanılır. Bu tür problemlerde ikili kodlama kullanılması problemin daha zor bir biçim almasına neden olur. Gerçek kodlamada her bir birey problemle bağlantılı olan form numaralarının, reel sayılar veya karakterlerin oluşturduğu katarla ifade edilir(Tablo 2.7).

Tablo 2.7 Gerçek kodlama kullanılarak oluşturulan bireyler

Kromozom A	1.2324 5.3243 0.4556 2.3293 2.4545
Kromozom B	ABDJEIFJDHDIERJFDLDFLFEGT
Kromozom C	(back), (back), (right), (forward), (left)

Gerçek kodlama bazı özel problemler için çok iyi sonuçlar vermekle birlikte, bu tür kodlama kullanıldığında, yeni çaprazlama ve mutasyon tanımlamaları yapılması gereklidir. Örneğin, yapay sinir ağlarında istenilen çıkışların elde edilmesi için girişlerin ağırlıklarının bulunması. Bu problemde bireyi temsil eden katarı oluşturan reel sayılar giriş ağırlıklarını temsil eder.

2.1.7.4. Ağaç Kodlaması

Ağaç kodlaması genellikle genetik programlama için gerekli ifadelerin daha kolay geliştirilmesini sağlamak amacıyla kullanılan bir yöntemdir. Bu tip kodlamada her bir birey, programlama dilinde kullanılan fonksiyonlar veya komutlar, nesnelerin birbirleriyle bağlantılarını gösteren ağaç yapıdan oluşmaktadır (Tablo 2.8).

Tablo 2.8 Ağaç yapı kodlaması kullanılarak oluşturulan bireyler [53]

Kromozom A	Kromozom B
<pre> graph TD A((+)) --- B((x)) A --- C(/) C --- D((5)) C --- E((y)) </pre>	<pre> graph TD A[do until] --- B[step] A --- C[wall] </pre>
(+ x (/ 5 y))	(do_until step wall)

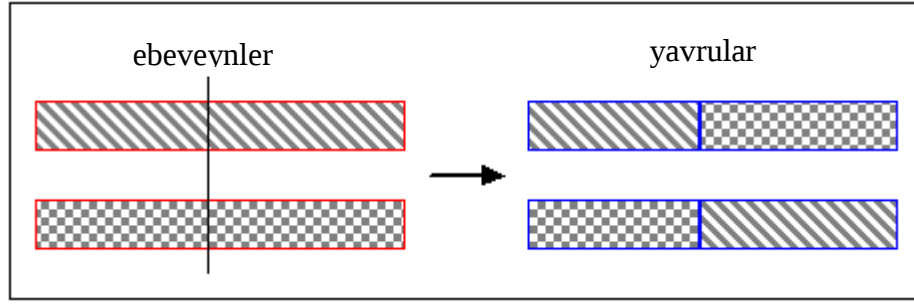
Daha öncedende söz edildiği gibi ağaç yapı kodlaması programlamanın değiştirilebilmesi ve geliştirilebilmesi için oldukça elverişlidir. Ağaç yapı nedeniyle çaprazlama ve mutasyon olacağı zaman, kullanılan programı oluşturan bölümler kolayca parçalanabilir, bu yüzden LISP programlama dili sık sık bu kodlama biçimini kullanmaktadır. Giriş ve çıkış değerleri verilen bir fonksiyonun bulunması örnek olarak verilebilir. Burada amaç tüm giriş değerleri için verilen çıkış değerine en yakın sonucu sağlayacak fonksiyonun hesaplanmasıdır.

2.7.2 Çaprazlama

2.7.2.1 İkili Kodlama İçin Çaprazlama

Tek-noktalı Çaprazlama

Tek-noktalı çaprazlamada, çaprazlamanın oluşacağı tek bir konum vardır ve bu konum rastgele belirlenir. Çaprazlama giren bireylerin bu noktaya kadar ve bu noktadan sonraki bit'leri yer değiştirerek iki yeni yavru oluşturulur. Tüm bu söylenenlerin daha iyi anlaşılabilmesinde Şekil 2.11 etkili olacaktır.



Şekil 2.11 Tek-noktalı çaprazlama

Aşağıda ikili kodlama ile verilmiş bireyler için çaprazlama konumu 5 kabul edilerek tek-noktalı çaprazlama yapılırsa;

birey 1	0	1	1	1	0	0	1	1	0	1	0
birey 2	1	0	1	0	1	1	0	0	1	0	1

Çaprazlama sonrası oluşan yeni bireyler (yavrular):

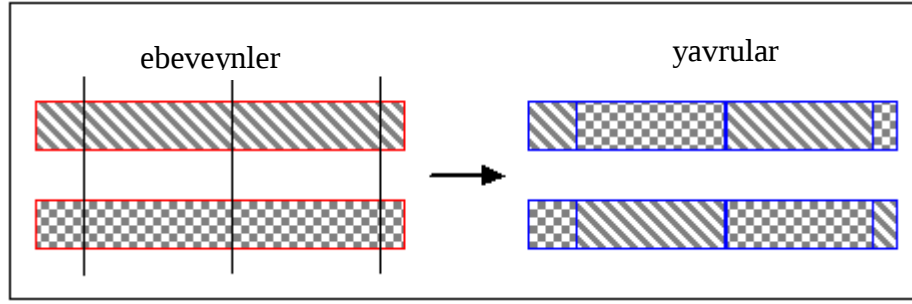
Yavru 1	0	1	1	1	0	1	0	0	1	0	1
Yavru 2	1	0	1	0	1	0	1	1	0	1	0

Çok-noktalı çaprazlama

Çok-noktalı çaprazlamada, çaprazlamanın yapılacağı m adet konum rastgele belirlenir ve çiftleşen bireylerin bu konumlar arasındaki bit'leri karşılıklı yer değiştirilerek yeni bireyler elde edilir. Şekil 2.12 bu işlemi göstermektedir.

Çaprazlama pozisyonu ($m = 3$) 2 6 10 olmak üzere çaprazlama yapıldığında;

birey 1	0	1	1	1	0	0	1	1	0	1	0
birey 2	1	0	1	0	1	1	0	0	1	0	1
yavru 1	0	1	1	0	1	1	0	1	1	1	1
yavru 2	1	0	1	1	0	0	0	0	1	0	0



Şekil 2.12. Çok-noktalı çaprazlama

Çok-noktalı çaprazlamanın ana fikri, ebeveynleri bir çok parçaya ayırarak çaprazlamaya sokup, ebeveynlerden oldukça farklı olan yavrular meydana getirmektir. Çok-noktalı çaprazlamanın sözü edilen bozucu etkisi nedeniyle araştırma alanı genişler ve başlangıçta uygun olan bireylere yakınsama ihtimali azalır. Dolayısıyla bu süreç araştırmayı daha dayanıklı (robust) hale getirir.

Düzgün Çaprazlama (Uniform Crossover)

Tek ve çok noktalı çaprazlamalarda, çaprazlamanın yapılacağı konum(lar) belirtiyor ve bu noktalar arasındaki bit'ler yer değiştirilerek yavrular oluşturuluyordu. Düzgün çaprazlamada ise, bireylerle aynı uzunlukta olan ve bit'leri rastgele üretilen bir maske kullanılır. Kullanılan bu maske, çaprazlama sonucunda oluşacak yavrunun bit'lerinin hangi ebeveyninden geleceğini gösterir.

birey 1	0	1	1	1	0	0	1	1	0	1	0
birey 2	1	0	1	0	1	1	0	0	1	0	1

Örneğin maskede bulunan 1, birey 1'den gelmesi gereken bit'i, maskede bulunan 0 ise birey 2 den gelmesi gereken bit'i gösterdiğini kabul edersek;

maske 1	0	1	1	0	0	0	1	1	0	1	0
maske 2	1	0	0	1	1	1	0	0	1	0	1

Çaprazlama sonucu oluşan yeni bireyler:

yavru 1	1	1	1	0	1	1	1	1	1	1	1
yavru 2	0	0	1	1	0	0	0	0	0	0	0

Düzgün çaprazlamanın algoritması, ayırık çaprazlama ile özdeştir.

Karışık Çaprazlama (Shuffled Crossover)

Karışık çaprazlama düzgün çaprazlamayla yakından alakalıdır. Tek-noktalı çaprazlamada olduğu gibi, çaprazlamanın olacağı konum belirlenir. Fakat değişkenler yer değiştirmeden önce iki ebeveyn arasında rastgele olarak karıştırılır. Çaprazlama sonunda oluşan yavrulardaki değişkenler karıştırılmamıştır.

2.7.2.2 Reel Kodlama İçin Çaprazlama

Ayrık Çaprazlama

Ayrık çaprazlama, değişkenlerin bireyler arasında değiştirilmesi ilkesine dayanır. Aşağıda üç değişkenli (üç boyutlu) iki birey gözönüne alınmıştır.

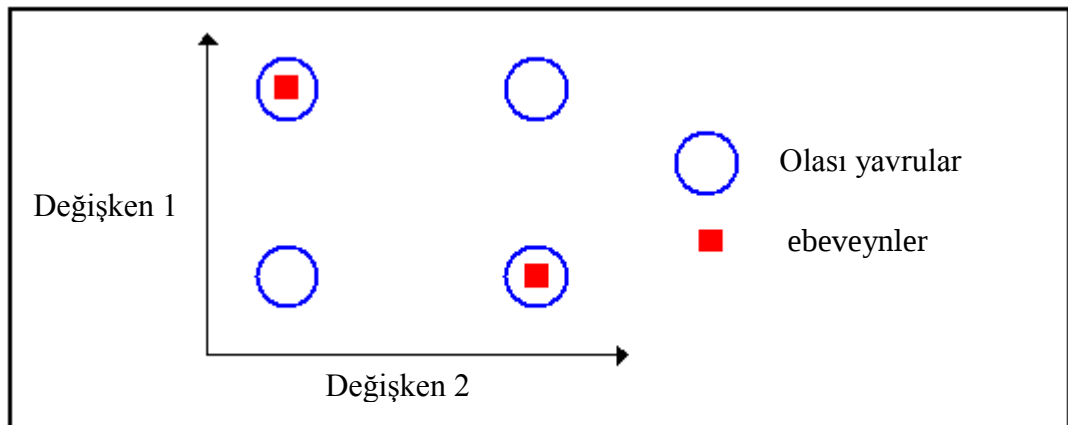
birey 1	12	25	5
birey 2	123	4	34

Modellerdeki her bir değişken, yavrulardaki değişkenlerin hangi ebeveynden geleceğini göstermek üzere,

model 1	2	2	1
model 2	1	2	1

Çaprazlama sonucu oluşan bireyler:

yavru 1	123	4	5
yavru 2	12	4	5



Şekil 2.13. Ayrık çaprazlamada oluşabilecek yavruların olası konumları [53]

Ayrık çaprazlamada, ebeveyley tarafından belirlenen hiperküp uzay parçasının uç noktaları (köşeleri) oluşmaktadır. Şekil 2.13 ayrık çaprazlamanın geometrik etkisini göstermektedir.

Ayrık çaprazlama çeşitli değişkenler (ikili, reel veya sembol) için kullanılabilir.

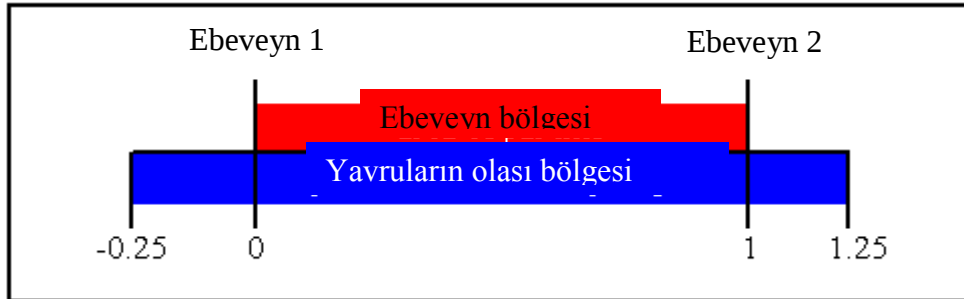
Ara (intermediate) Çaprazlama

Ara çaprazlama sadece reel değişkenler için kullanır (ikili değişkenler için kullanılamaz). Burada yavruların değişkenleri, ebeveynlerin değişkenlerine bağlı olarak herhangi bir şekilde (ebeveynler arasında veya etrafında) seçilebilir.

Yavrular aşağıdaki formüle göre oluşturulursa:

$$\text{yavru} = \text{ebeveyn 1} + \text{Alfa} (\text{ebeveyn 2} - \text{ebeveyn 1}),$$

burada Alfa ölçek katsayısı olmak üzere $[-d, 1 + d]$ aralığında rastgele seçilir. Ara çaprazlamada $d = 0$, genişletilmiş ara çaprazlamada $d > 0$. İyi bir seçim için genellikle $d = 0.25$ olarak alınır. Ebeveynlerdeki değişkenlerin herbiri için bir alfa değeri seçilir ve yukarıdaki formül kullanılarak oluşacak yavrunun değişkenleri belirlenir.



Şekil.2.14 Ara çaprazlamada ebeveynlerin ve yavruların dağılımları [53]

Üç değişkenli iki birey:

birey 1 12 25 5

birey 2 123 4 34

olmak üzere alfa aşağıdaki modellerdeki gibi seçildiğinde:

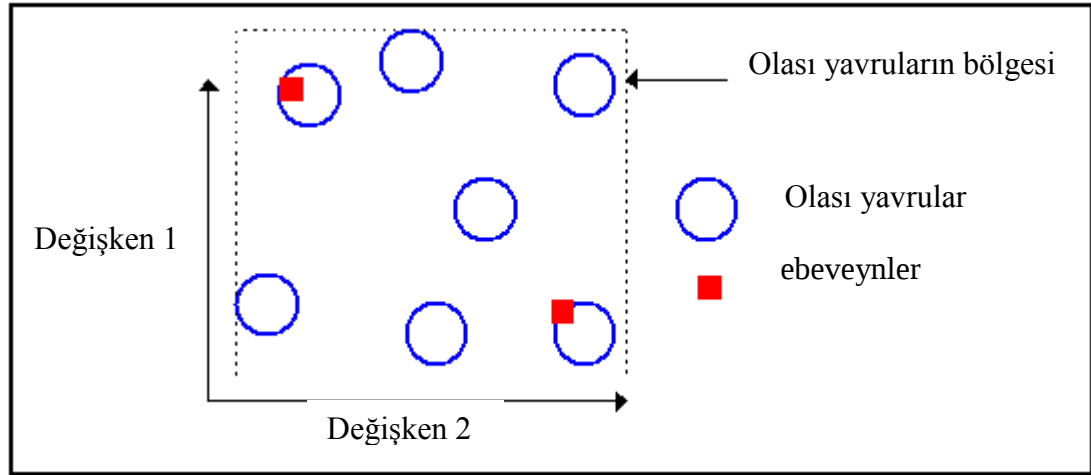
model 1 0.5 1.1 -0.1

model 2 0.1 0.8 0.5

Oluşacak yeni bireyler şu şekilde hesaplanır:

yavru 1	67.5	1.9	2.1
yavru 2	23.1	8.2	19.5

Ara çaprazlama, oluşabilecek yavruların bulunabileceği noktalar ebeveynlerin tanımladığı hiperküpden biraz daha geniş bir uzay parçasına yayılır. Şekil 2.15’de bu rahatça görülmektedir.



Şekil.2.15 Ara çaprazlama ile oluşabilecek yavruların olası konumları [53]

Doğrusal Çaprazlama (Line recombination)

Doğrusal çaprazlama, ara çaprazlamasına çok benzemekle birlikte tek farklılığı, tüm değişkenler için yalnız bir alfa değeri kullanılmasıdır.

birey 1	12	25	5
---------	----	----	---

birey 2	123	4	34
---------	-----	---	----

Bu örnek için alfa:

model 1	0.5
---------	-----

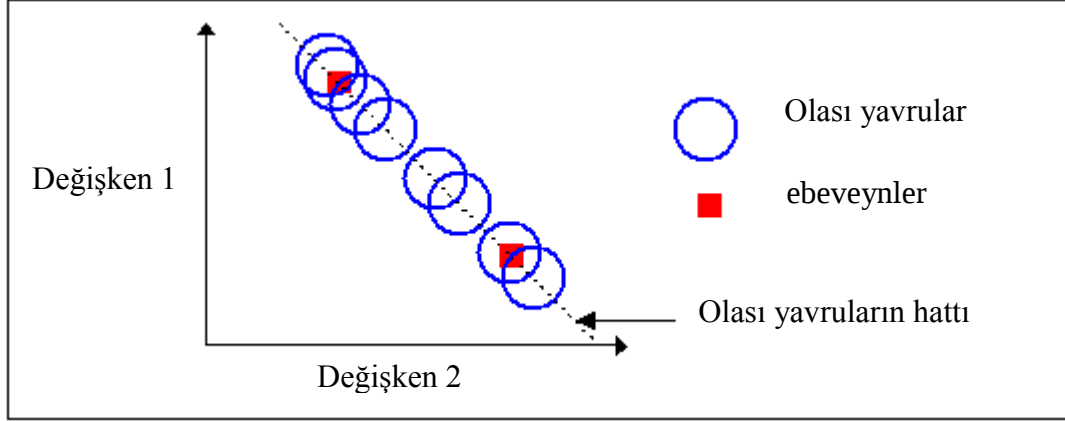
model 2	0.1
---------	-----

olarak seçilirse oluşacak yavrular:

yavru 1	67.5	14.5	19.5
---------	------	------	------

yavru 2	23.1	22.9	7.9
---------	------	------	-----

Doğrusal çaprazlama, ebeveynler tarafından tanımlanan bir doğru boyunca noktalar üretir. Şekil 2.16 doğrusal çaprazlama sonucu oluşan bireylerin dağılımını göstermektedir.



Şekil.2.16 Doğrusal çaprazlama ile oluşabilecek yavruların olası konumları [53]

2.7.3 Mutasyon

Çaprazlama aşamasından sonra mutasyon olma olasılığına bağlı olarak mutasyon işlemi meydana gelir. Mutasyonun oynadığı rolün önemi hala bir tartışma konusudur. Bazı araştırmacılar mutasyonun ikincil derecede önemli olğuna inansa da, doğa baz alınarak geliştirilen bu yöntem mutasyonsuz düşünülemez.

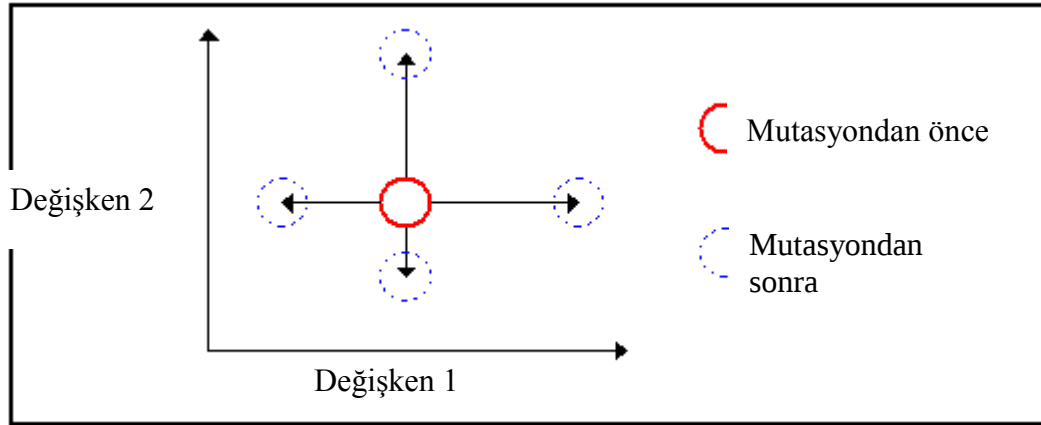
Mutasyon sonucunda asıl (orjinal) toplulukta olmayan özellikler kazanılabilir, dolayısıyla bu genetik algoritmanın hızlıca bir sonuca ulaşmasını (yakınsamasını) önler. Mutasyon olasılığı arttıkça algoritma öngörülen araştırma uzayından uzaklaşacaktır.

Birden çok optimum noktasına sahip fonksiyonlarda, yerel optimum noktasına yakınsayan ve kısır döngüye giren genetik algoritma mutasyon sayesinde bu nokta etrafından uzaklaşabilir. Böylece evrensel optimum noktasına ulaşma olasılığı artar.

En iyi çözümün yitirilmemesi için, son yineleme işlemi sırasında mutasyon meydana gelmez.

2.7.3.1 Reel Değişkenli Bireylerde Mutasyon

Şekil 2.17 reel değişkenli bireyler için iki boyuttaki olası mutasyonu gösteriyor.



Şekil 2.17 Mutasyonun etkisi [53]

Mutasyon adımının büyüklüğünü belirlemek oldukça zordur. En uygun adım aralığı üzerinde çalışılan probleme göre farklılık gösterir.[..]. Küçük adım aralığı seçmek genellikle başarılı olmakla birlikte bazı durumlarda adım aralığını büyültmek sonuca ulaşma hızını oldukça artırır.

2.7.3.2. İkili Kodlanmış Bireyler İçin Mutasyon

İkili kodlanmış bireylerde meydana gelecek mutasyon 0 olan bit'i 1'e, 1 olan bit'i de 0'a dönüştürür. Mutasyonun olacağı konum rastlantısal olarak belirlenir. Tablo 2.9 ikili kodlamaya göre yazılmış 11 değişkenli bir bireyin dördüncü bit'inde mutasyon meydana gelmesi işlemini gösteriyor.

Tablo 2.9 Mutasyon öncesi ve sonrası bireyler

Mutasyondan önce	0	1	1	1	0	0	1	1	0	1	0
Mutasyondan sonra	0	1	1	0	0	0	1	1	0	1	0

Mutasyonun etkisi kullanılan kodlama biçimine bağlıdır. Örneğin 1-10 arasındaki sayılar için tercih edilen farklı kodlama sistemlerinde mutasyon öncesi ve sonrası değişimler Tablo 2.10'da görülmektedir.

Tablo 2.10. Farklı kodlamalarda mutasyon etkisi [53]

aritmetik		logaritmik	
ikili kodlama	gray kodlaması	ikili kodlama	gray kodlaması
5.0537	4.2887	2.8211	2.3196
4.4910	3.3346	2.4428	1.8172

2.8. Örnekler

Şimdiye kadar teorik olarak anlatılan genetik algoritmalar, yöntemin daha iyi anlaşılabilmesi için iki basit optimizasyon problemine uygulanacaktır.

2.8.1 Örnek 1.

Örnek olarak elektrik mühendisliğinde sıkça karşılaşılan en büyük kayıp gücün hesaplanması probleminin GA ile çözümü açıklanacaktır. Kayıp güç, elektrik devrelerinde kendini ısı olarak belli eden ve Joule kaybı olarak bilinen bir büyüklüktür. Bu büyüklük, devreden akan I akımın karesi ve Z empedansı ile doğru orantılı olarak değişir ve $P_{\text{kayıp}} = I^2 \cdot Z$ fonksiyonu ile ifade edilir. Burada sayısal bir örnek olarak, $P_{\text{kayıp}} = I^2 \cdot Z$ fonksiyonunda akımın $I \in [0, 31]$ aralığında değişmesi durumunda, en büyük kayba yol açan I akım değerinin bulunması problemi çözülecektir. Örneğin $Z = 2 \text{ k}\Omega = \text{sabit}$ ve I değişken olarak alındığında fonksiyon $P_{\text{kayıp}} = 2I^2$ halini alır. Şimdi genetik algoritmaları kullanarak problem çözümüne geçebiliriz.

GA'nın ilk adımını, değişkenin ikili tabana göre kodlanması oluşturur. Verilen $[0, 31]$ aralığı (aralıkta $2^5 = 32$ sayı olduğundan) 5 bit'lik bir kodlamaya uygun düşer. Dolayısıyla 0: "00000" ve 31: "11111" olacaktır.

Topluluğun birey sayısını örneğin $n = 4$ olarak alalım. Buna göre topluluğu oluşturacak ilk dört bireyin herbir bit'i yazı tura atmak suretiyle tamamen rastlantısal bir biçimde seçilsin ve seçilen bireylerin ikili ve onlu tabandaki karşılıkları şunlar olsun:

Birey 1: 00011 $\longrightarrow I_1 = 3 \text{ kA}$

Birey 2: 01010 $\longrightarrow I_2 = 10 \text{ kA}$

Birey 3: 01111 $\longrightarrow$ $I_3 = 15 \text{ kA}$

Birey 4: 10110 $\longrightarrow$ $I_4 = 22 \text{ kA}$

Bir sonraki toplulukta oluşacak olan ve yavru olarak adlandırılan bireyleri belirlemek için çaprazlamaya girecek olan ebeveynlerin seçiminde, rulet çarkı yöntemi önemli bir rol oynar. Her bir bireyin temsil ettiği değer, $P_{\text{kayıp}}$ amaç fonksiyonunda yerine konulur ve bireylerin uygunluk değerleri (u) bulunur. Bulunan uygunluk değerlerine bağlı olarak, bireylerin rulet çarkında kaplayacakları alanlar tanımlanır. Dolayısıyla problem en büyük (maksimum) değer bulma problemi olduğundan bireyin uygunluk derecesi ne kadar büyükse çarkta kaplayacağı alan o kadar büyük olacaktır (Problem en küçük (minimum) değer bulma problemi olsaydı bireylerin uygunluk değerlerine bağlı olarak çarkta kaplayacakları alanlar ters orantılı olacaktı). Seçilen bireylerle bulunan kayıplar, toplam kayıp ve bunların çarkta kapladıkları alanlar:

$$P_{\text{kayıp}}(I_1) = 18 \text{ kW}$$

$$P_{\text{kayıp}}(I_2) = 200 \text{ kW}$$

$$P_{\text{kayıp}}(I_3) = 450 \text{ kW}$$

$$P_{\text{kayıp}}(I_4) = 968 \text{ kW}$$

$$\Sigma P_{\text{kayıp}} = 18 + 200 + 450 + 968 = 1636$$

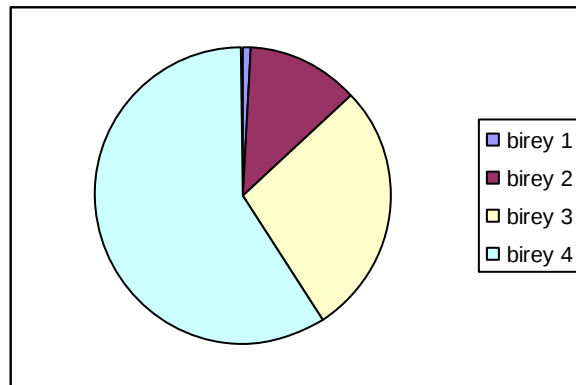
$$u_1 = 18/1636 = 0,011$$

$$u_2 = 200/1636 = 0,122$$

$$u_3 = 450/1636 = 0,275$$

$$u_4 = 968/1636 = 0,592$$

olur. Bu değerler, rulet çarkının her çevrilisinde hangi olasılıkla hangi bireyin seçileceğini belirtir, örneğin bir numaralı birey 0,011 olasılıkla seçilecektir.



Şekil 2.18 Bireylerin rulet çarkında kapladığı alanlar

Topluluktaki birey sayısı sabit kalacağından, çark dört kez çevrilerek çiftleşmeye girecek bireyler (ebeveynler) belirlenecek yani çiftleşme havuzu oluşturulacaktır. Buna göre rulet çarkı dört kez döndürülmüş ve örneğin birinci bireyin 0 kere, ikinci bireyin 1 kere, üçüncü bireyin 1 kere, dördüncü bireyin 2 kere geldiği görülmüş olsun. Buradan çiftleşmeye girecek olan ebeveynler: 01010, 10110, 10110, 01111 olarak bulunur. Bir sonraki aşama çiftleşecek ebeveynlerin eşleştirilmesi ve çaprazlama olacağı konumun belirlenmesidir. Rastlantısal olarak eşleştirme yapılmış ve bunun sonucunda (ebeveyn 1, ebeveyn 2) ve (ebeveyn 3, ebeveyn 4) çiftleşme grupları oluşturulmuştur. Çaprazlamanın olacağı konumlar da rastlantısal olarak belirlenmiş ve birinci grup için $k = 2$, ikinci grup için $k = 4$ olarak bulunmuştur. Belirtilenlere göre çaprazlama yapıldığında aşağıdaki sonuçlar elde edilmiştir (çaprazlama konumu “/” ile belirtilmiştir).

Çiftleşme grubu 1 ($k = 2$)

Ebeveyn 1: 01/010

Ebeveyn 2: 10/110 olmak üzere oluşacak bireyler 10010 ve 01110

Çiftleşme grubu 2 ($k = 4$)

Ebeveyn 3: 1011/0

Ebeveyn 4: 0111/1 oluşacak yavru bireyler 10111 ve 01110

Çiftleşme sonucu oluşan yavrular:

Yavru 1: 10010

Yavru 2: 01110

Yavru 3: 10111

Yavru 4: 01110

şeklindedir. Son aşama olarak ise mutasyon devreye girer. Mutasyon olma olasılığı topluluğun birey sayısına bağlı olarak belirlenir. Örnekte mutasyon olasılığı 0,001 olarak yani bin bit’ten bir tanesi değişime uğrayacak şekilde seçilmiştir. Mutasyon olasılığına bağlı olarak mutasyona uğrayacak bit rastlantısal bir şekilde (örneğin yazı

tura atılarak) bulunur. Bu işlem yapılmış ve oluşan ikinci yavrunun üçüncü bit'inde mutasyon olacağı ortaya çıkmıştır. Dolayısıyla yavru 2: 01010 halini alır.

Mutasyon adımında tamamlanmasından sonra bir sonraki topluluğu (nesil) oluşturacak bireyler belirlenmiş olur. Buna göre yeni topluluk şu şekildedir:

Birey 1: 10010 $\longrightarrow$ $I_1 = 18 \text{ kA}$ $\longrightarrow$ $P_{\text{kayıp}}(I) = 648 \text{ kW}$

Birey 2: 01010 $\longrightarrow$ $I_2 = 10 \text{ kA}$ $\longrightarrow$ $P_{\text{kayıp}}(I) = 200 \text{ kW}$

Birey 3: 10111 $\longrightarrow$ $I_3 = 23 \text{ kA}$ $\longrightarrow$ $P_{\text{kayıp}}(I) = 1058 \text{ kW}$

Birey 4: 01110 $\longrightarrow$ $I_4 = 14 \text{ kA}$ $\longrightarrow$ $P_{\text{kayıp}}(I) = 392 \text{ kW}$

Yukarıdaki örnekte sadece bir kez tekrarlama (iterasyon) yapılmış, başlangıç topluluğundan bir sonraki topluluk (nesil) elde edilmiştir. Yapılan işlemlerde herşey çok fazla rastlantısal olarak görünse de, uygunluk değeri yüksek olan bireylerin seçilme ve çiftleşme olasılıkları yüksek olduğu için bir sonraki topluluğu oluşturacak olan bireylerin uygunluk değerlerinin ortalamasının da artacağı açıktır. Genetik algoritmaların çalışmasının tam olarak gözlenebilmesi için tekrarlama sayısı artırılmalıdır.

2.8.2 Örnek 2.

$f(x) = x \cdot \sin(10\pi \cdot x) + 1,0$ fonksiyonunu $x \in [-1, 2]$ aralığında maksimum yapan x değerinin genetik algoritmalar kullanılarak bulunması.

Çözüm : Bir önceki örnekte olduğu gibi problem çözümüne kodlama işlemi ile başlanır. Kodlama işlemi için x değişkeninin alabileceği değer aralığını 3 milyon aralığa bölersek ihtiyacımız olan kromozomlardaki bit sayısı 22 olacaktır:

$$2097152 = 2^{21} < 3000000 \leq 2^{22} = 4194304$$

Bu durumda sınır değerler -1 ve 2 sırasıyla 0000000000000000000000 ve 1111111111111111111111 kromozomları ile ifade edilecektir. İkili kodlama sistemi ile kodlanmış kromozomun $[-1, 2]$ aralığında hangi değeri temsil ettiğini bulmak için önce kromozom onluk sisteme çevrilir (x^I). Daha sonra ise aşağıdaki formülden yararlanılarak geçiş sağlanır.

$$x = -1,0 + x^I \cdot \frac{3}{4194303}$$

Bu formülde,

-1,0: aralığın alt sınırı

3: [-1, 2] arasındaki aralık sayısı

4194303: $x^I_{\text{üst sınır}} - 1$

Örneğin, 1000101110110101000111 kromozomunun onluk tabandaki karşılığı

$x^I = 2288967$ ve buradan

$$x = -1,0 + 2288967 \cdot \frac{3}{4194303} = 0,637197$$

olarak bulunur.

Bir sonraki adımı olarak başlangıç toplumunun 22 bit'lik kromozomlarının rastlantısal olarak üretilmesi oluşturur. Örneğin dört kromozom;

Kromozom 1: 1000101110110101000111 $\longrightarrow$ $x_1 = 0,637197$

Kromozom 2: 0000001110000000010000 $\longrightarrow$ $x_2 = -0,958973$

Kromozom 3: 1110000000111111000101 $\longrightarrow$ $x_3 = 1,627888$

Kromozom 4: 0000000000000000000111 $\longrightarrow$ $x_4 = -0,999994$

Kromozomların rulet çarkında kaplayacağı yerlerin oranlarını bulmak için bu noktalar fonksiyonda yerine konularak her bir kromozomun uygunluk değerleri (u) bulunur. Buna göre,

$$f(x_1) = 1,586345$$

$$f(x_2) = 0,078878$$

$$f(x_3) = 2,250650$$

$$f(x_4) = 0,999812$$

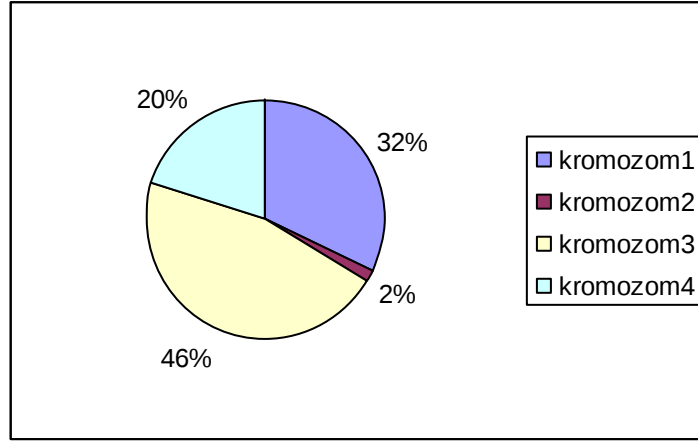
$$\Sigma f = 1,586345 + 0,078878 + 2,250650 + 0,999812 = 4,915685$$

$$u_1 = 1,586345/4,915685 = 0,322711$$

$$u_2 = 0,078878/4,915685 = 0,016046$$

$$u_3 = 2,250650/4,915685 = 0,457851$$

$$u_4 = 0,999812/4,915685 = 0,203392$$



Şekil 2.19 Rulet çarkında kaplanacak alanlar

Çark dört kez çevrildiğinde çaprazlamaya girecek ebeveynler belirlenmiş olur. Bu işlem yapıldığında çarkın ibresi kromozom 4’de bir kere, kromozom 1’de iki defa, kromozom 3’de bir defa durmuştur. Buna göre kromozom 1 ile kromozom 3 ve kromozom 1 ile kromozom 4 eşleşmeleri rastlantısal işlemler sonucu oluşturulmuş ve çift-noktalı çaprazlama seçilmiş ve çaprazlama konumları olarak birinci grup için ($c_{k_1} = 3-9$), ikinci grup için ise ($c_{k_2} = 10-19$) rastlantısal olarak seçilmiştir.

Çifileşme grubu 1 ($c_{k_1} = 3-9$)

Kromozom 1: 100/**010111**/0110101000111

Kromozom 3: **111**/000000/**0111111000101**

Çaprazlama sonucu oluşacak kromozomlar

Yavru 1: 1000000000110101000111

Yavru 2: 1110101110111111000101

Çiftleşme grubu 2 ($c_{k_2} = 10-19$)

Kromozom 1: 1000101110/**110101000**/111

Kromozom 4: **0000000000**/000000000/**111**

Çaprazlama sonucu oluşacak kromozomlar

Yavru 3: 1000101110000000000111

Yavru 4: 0000000000110101000111

Mutasyon olasılığı; $p_m = 0,001$ olarak seçildiğinde yine rastlantısal sonuçlara göre mutasyonun 4. yavrunun ilk bit'inde olacağı saptanmıştır. Buna göre yeni topluluğun kromozomları

Kromozom 1: 1000000000110101000111

Kromozom 2: 1110101110111111000101

Kromozom 3: 1000101110000000000111

Kromozom 4: 1000000000110101000111

şeklini alır.

Bu problem tekrarlama sayısı arttırılarak ve topluluk genişletilerek yapıldığında genetik algoritmaların sonuca ne kadar iyi yakınsadığı ortaya çıkar. Buna göre topluluğun büyüklüğü (kromozom sayısı); $pop_size = 50$, çaprazlama olma olasılığı; $p_c = 0.25$, mutasyon olasılığı $p_m = 0.01$ olmak üzere, yukarıdaki fonksiyonu için 150 topluluk (kuşak, nesil) sonraki en iyi kromozom $krom_{max} = 1111001101000100000101$ yani $x_{max} = 1.85$ ve $f(x_{max}) > 2,85$ olacağı görülmüştür. Bu problemin verileri Tablo2.11'de gösterilmiştir.

Tablo.2.11 150 nesil boyunca problem verileri

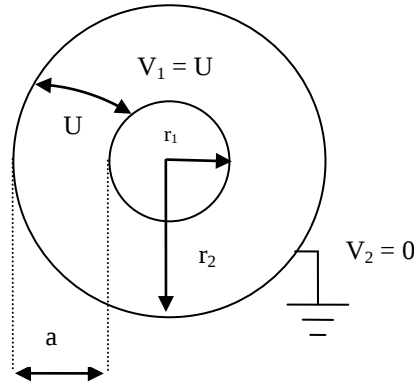
Topluluk numarası	Fonksiyon deęerleri
1	1,441942
6	2,250003
8	2,250283
9	2,250284
10	2,250363
12	2,328077
39	2,344251
40	2,345087
51	2,738930
99	2,849246
137	2,850217
145	2,850227

3. YÜKSEK GERİLİM UYGULAMALARI

3.1 Statik Elektrik Alanı

3.1.1 Küresel Elektrot Sistemi

Bir tabakalı ve eş merkezli küresel elektrot sisteminde yarıçapları sırasıyla r_1 , r_2 ve elektrotlarına uygulanan gerilimi U olan yapı Şekil 3.1 de gösterilmiştir.



Şekil 3.1 Eş merkezli küresel elektrot sistemi

Burada V potansiyeli sadece r 'ye bağlı olarak değiştiğinden, küresel koordinat sisteminde Laplace denklemi

$$\frac{d^2V}{dr^2} + \frac{2}{r} \cdot \frac{dV}{dr} = 0 \quad (3.1)$$

şeklini alır. Bu denklemin genel çözümü,

$$V = A + \frac{B}{r} \quad (3.2)$$

olur. A ve B sabitleri sınır koşullarından bulunur. $r = r_1$ için $V = V_1 = U$ ve $r = r_2$ için $V = V_2 = 0$ olduğu göz önünde tutulursa,

$$A = -U \frac{r_1}{r_2 - r_1} \quad (3.3)$$

$$B = U \frac{r_1 r_2}{r_2 - r_1} \quad (3.4)$$

değerleri elde edilir. Bunlar (3.2) denkleminde yerine konulursa potansiyel ifadesi

$$V = U \frac{r_1}{r_2 - r_1} \left(\frac{r_2}{r} - 1 \right) \quad (3.5)$$

olur. Alan şiddeti için de $E = -dV/dr$ bağıntısından

$$E = U \frac{r_1 r_2}{r_2 - r_1} \frac{1}{r^2} \quad (3.6)$$

denklemini elde edilir.

$$r = r_1 \text{ için } E = E_{\max} = U \frac{r_2}{r_1} \frac{1}{r_2 - r_1} \quad (3.7)$$

$$r = r_2 \text{ için } E = E_{\min} = U \frac{r_1}{r_2} \frac{1}{r_2 - r_1} \quad (3.8)$$

olur.

Tablo 3.1 küresel elektrot sistemlerinde maksimum ve minimum elektriksel alan şiddetlerinin analitik ve genetik algoritmalarla bulunması ile elde edilen sonuçları göstermektedir. Yine aynı tabloda farklı topluluk büyüklüklerinin aranan elektriksel alan şiddetlerini ne şekilde etkilediğini gösteren bağıl hata değerleri görülmektedir.

Genetik algoritma hesaplama parametreleri ise EK B' de gösterilmiştir.

Tablo 3.1 Küresel Elektrot Sisteminde Analitik ve GA Yaklaşımı Sonuçları

Analitik Hesap		Genetik Algoritma Sonucu			Bağlı hata	
$E_{\max}$ (V/m)	$E_{\min}$ (V/m)	n_{pop}	$E_{\max}$ (V/m)	$E_{\min}$ (V/m)	$\varepsilon_{\max}$ (%)	$\varepsilon_{\min}$ (%)
107.1428571	0.476190476	20	75.3723724	0.509711132	29.65	7.0393
		35	85.3388227	0.546032847	20.35	14.6668
		50	106.9599749	0.485103269	0.1706	1.8716
		100	107.1428571	0.476842225	0	0.1326
		200	107.1428571	0.476190476	0	0.0227

3.1.1.1 Küresel Elektrot Sisteminin Delinme Bakımından İncelenmesi

Elektrot sistemlerindeki maksimum zorlanma, ortamın delinme dayanımına eşit veya büyük olduğunda, eğrilik yarıçapı küçük olan elektrot üzerinde boşalma başlar ve bu elektrotların tam delinmesiyle sonuçlanır.

Dış küre çapı r_2 sabit ve uygulanan gerilim U sabit kalmak koşuluyla $E_{\max}$ ' ın r_1 ' e göre değişimi şu şekilde olur;

$$E_{\max} = \frac{U}{\frac{r_1}{r_2}(r_2 - r_1)} = \frac{U}{\alpha} \quad (3.9)$$

$$\alpha = \frac{r_1}{r_2}(r_2 - r_1) \quad (3.10)$$

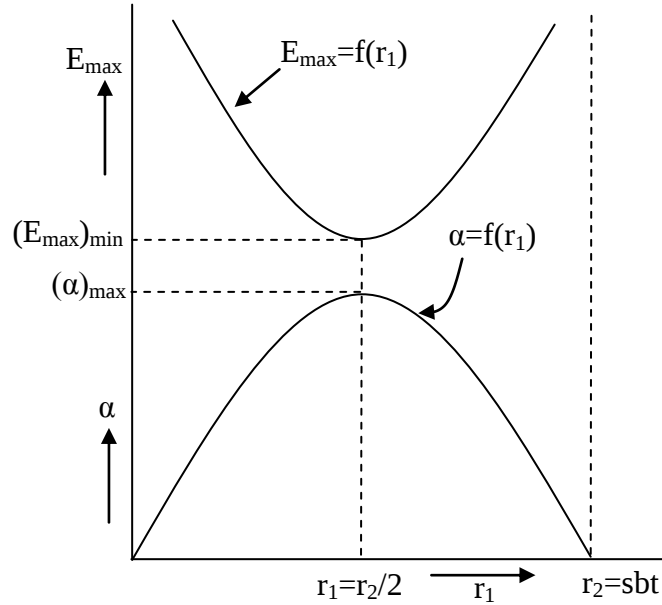
Bu ifadelerden görüldüğü gibi $E_{\max}$ değerinin minimum olması için α değerinin maksimum olması gerekir. α 'yı maksimum yapan değeri bulmak için bu ifadenin türevi alınıp sıfıra eşitlenirse $r_1 = r_2/2$ bulunur. Buradan,

$$\alpha_{\max} = \frac{r_1}{2} = \frac{r_2}{4} \quad (3.11)$$

$$(E_{\max})_{\min} = \frac{4U}{r_2} = \frac{2U}{r_1} \quad (3.12)$$

olarak bulunur.

$U = 100$ (V) , $r_2 = 15$ m olan bir küresel elektrot sisteminin α ve $E_{\max}$ değerlerinin r_1 ile değişim grafiği Şekil 3.2’de gösterilmiştir. Ayrıca aynı sistem için analitik ve GA ile hesaplamalar yapılmış ve sonuçlar Tablo 3.2’de verilmiştir.



Şekil 3.2 Küresel elektrot sisteminde α ve $E_{\max}$ 'ın r_1 ile değişimi

Tablo 3.2 Küresel Elektrot Sisteminin Delinme Bakımından İncelenme Sonuçları

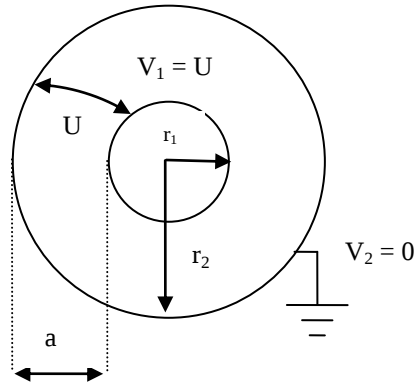
Analitik hesap			Genetik Algoritma Sonucu		Bağıl Hata	
p_c (%)	α (m)	$(E_{\max})_{\min}$ (V/m)	α (m)	$(E_{\max})_{\min}$ (V/m)	ϵ_{α} (%)	$\epsilon_{E_{\max\min}}$ (%)
15	3.75	26.666666	3.74981522	26.667980	0.0000492746	0.00004927
25			3.74999999	26.666666	0.0000000002	0
45			3.74999987	26.666675	0.0000000336	-0.00000033
60			3.74991572	26.667266	0.0000022746	-0.00002287

Hesaplamalar için büyüklükler EK B de verilmiştir.

3.1.2 Silindrsel Elektrot Sistemi

Silindrsel elektrot sistemleri uygulamada küresel elektrot sistemlerine göre daha fazla kullanılmaktadır. Yüksek gerilim hava hatları, kablolar ve geçit izalatörleri bu sisteme örnek olarak verilebilir [9].

Yarıçapları sırasıyla r_1 ve r_2 , uzunluğu l ve elektrotları arasına uygulanan gerilimi U olan eş eksenli silindrsel elektrot sistemi Şekil 3.2' de görölmektedir.



Şekil 3.3 Eş eksenli silindrsel elektrot sistemi

V potansiyeli sadece r 'ye bağılı olarak değıştığınden, silindrsel koordinat sisteminde Laplace denklemi

$$\frac{d^2V}{dr^2} + \frac{1}{r} \cdot \frac{dV}{dr} = 0 \quad (3.13)$$

$$E = -\frac{dV}{dr} \quad \text{buradan} \quad \frac{dE}{dr} = -\frac{d^2V}{dr^2}$$

$$\frac{dE}{dr} + \frac{1}{r}E = 0 \rightarrow \frac{dE}{E} = -\frac{dr}{r}$$

şeklini alır. Bu denklemin genel çözümü,

$$V = A + B \ln r \quad (3.14)$$

A ve B integral sabitleri sınır koşullarından bulunur.

$$A = \frac{U}{\ln \frac{r_2}{r_1}} \ln r_2 \quad \text{ve} \quad B = -\frac{U}{\ln \frac{r_2}{r_1}} \quad (3.15)$$

Değerleri elde edilir. Bunlar (3.14) denkleminde yerine konulursa potansiyel bağıntısı

$$V = \frac{U}{\ln \frac{r_2}{r_1}} \ln \frac{r_2}{r} = \frac{U}{\ln \frac{r_2}{r_1}} (\ln r_2 - \ln r) \quad (3.16)$$

olur. Alan şiddeti için $E = -dV/dr$ bağıntısından,

$$E = \frac{U}{\ln \frac{r_2}{r_1}} \frac{1}{r} \quad (3.17)$$

denklemini elde edilir. $r = r_1$ ve $r = r_2$ değerleri yerine konulursa

$$r = r_1 \text{ için } E = E_{\max} = \frac{U}{r_1 \ln \frac{r_2}{r_1}} \quad (3.18)$$

$$r = r_2 \text{ için } E = E_{\min} = \frac{U}{r_2 \ln \frac{r_2}{r_1}} \quad (3.19)$$

Genetik algoritma ve analitik hesaplama sonuçları Tablo 3.3’de görülmektedir.

Tablo 3.3 Silindirsel Elektrot Sisteminde Analitik ve GA Yaklaşımı Sonuçları

	Analitik hesap		Genetik Algoritma Sonucu		Bağıl Hata	
P_m (%)	$E_{\max}$ (V/m)	$E_{\min}$ (V/m)	$E_{\max}$ (V/m)	$E_{\min}$ (V/m)	$\varepsilon_{E_{\max}}$ (%)	$\varepsilon_{E_{\min}}$ (%)
0.1	36.92653731	2.461795821	36.83512	2.53709	0.002475	0.030582
0.5			36.61504	2.46446	0.008462	0.001084
1			32.93092	2.46291	0.108204	0.000454

Hesaplamalar için seçilen büyüklükler EK B’ de gösterilmiştir.

3.1.2.1 Eş Eksenli Silindrsel Sistemin Ekonomik Bakımdan İncelenmesi

Eş eksenli silindrsel sistemlerin en önemli uygulamalarından biri de bir damarlı yüksek gerilim kablolarıdır. Yüksek gerilim kablolarının en elverişli biçimde boyutlandırılması hesaplanırken delinme bakımından en elverişli düzen veya ekonomik bakımdan en elverişli düzen göz önünde bulundurulur. Bir diğer çözüm de iki düzene göre çözümün ortak kullanılması, yani iki çözüm arasında bir ara çözüm bulunmasıdır [9].

Delinme bakımından en elverişli düzendeki gibi dış çapın sabit kaldığı kabulünü burda da uygularsak, sistemde kullanılacak yalıtkan madde miktarının iç silindir yarı çapına bağlı olarak değişeceği görülecektir. Sisteme uygulanan U geriliminin ve yalıtkan malzemenin E_d delinme dayanımının sabit olduğu kabul edilerek malzeme kullanımının en az olduğu geometrik karakteristik değeri hesaplanabilir.

Sistemin birim uzunluğuna düşen yalıtkan malzeme ağırlığı,

$$G = \pi \gamma (r_2^2 - r_1^2) = \pi \gamma r_1^2 (p^2 - 1) \quad (3.20)$$

Sistemin gerilim ifadesi ise,

$$U = E_d r_1 \ln p \rightarrow r_1 = \frac{U}{E_d \ln p} \quad (3.21)$$

şeklindedir. Buradan, birim uzunluktaki ağırlık için

$$G = \pi \gamma \left(\frac{U}{E_d} \right)^2 \frac{p^2 - 1}{(\ln p)^2} = K \frac{p^2 - 1}{(\ln p)^2}, \quad K = \text{sabit} = \pi \gamma \left(\frac{U}{E_d} \right)^2 \quad (3.22)$$

bağıntısı elde edilir. Bu bağıntıda G 'yi minimum yapan p geometrik karakteristiği türev işlemi veya çizimle bulunursa

$$p^2 \ln p - p^2 + 1 = 0 \quad (3.23)$$

elde edilir.

Hesaplama büyüklükleri ve GA parametreleri EK B'de verilen p karakteristiğinin bulunan sonuçları Tablo 3.4'de gösterilmiştir.

Tablo 3.4 p geometrik karakteristiğinin hesaplanması

Geometrik karakteristik	Analitik Hesap	Genetik Algoritma	Bağıl Hata
p	2.2190	2.217	0.00090130

3.1.2.2 Silindrsel Elektrot Sisteminin Delinme Bakımından İncelenmesi

Dış silindir çapı r_2 'nin ve U'nun sabit olduğu durumda $E_{\max}$ 'ın r_1 'e göre değişimi,

$$E_{\max} = \frac{U}{r_1 \ln \frac{r_2}{r_1}} = \frac{U}{\alpha} \quad (3.24)$$

α 'nın maksimum olduğu değer için $(E_{\max})_{\min}$ olacaktır.

α 'nın türevi alınıp sıfıra eşitlenirse

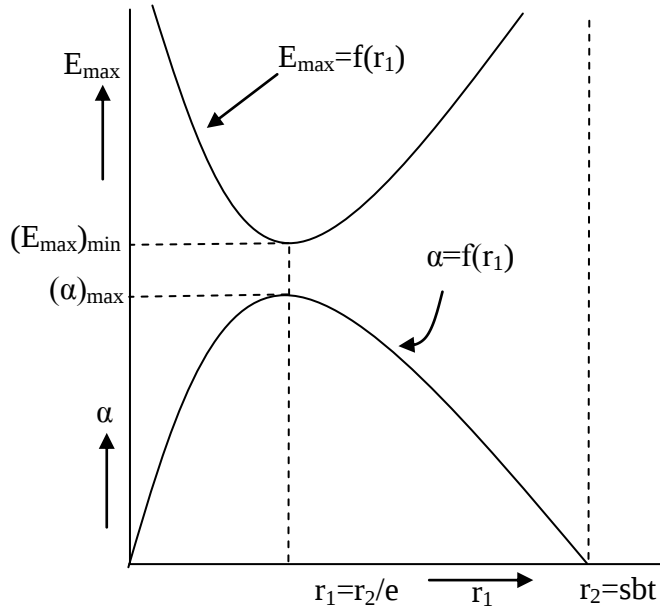
$$\alpha_{\max} = \frac{r_2}{e} = r_1 \quad (3.25)$$

$$(E_{\max})_{\min} = \frac{U}{r_1} \quad (3.26)$$

$U = 100$ (V) , $r_2 = 15$ m olan bir silindrsel elektrot sisteminin α ve $E_{\max}$ değişim grafiği Şekil 3.4'de gösterilmiştir. Ayrıca aynı sistem için analitik ve GA ile hesaplamalar yapılmış ve sonuçlar Tablo 3.5'de hesaplama parametreleri EK B'de verilmiştir.

Tablo 3.5 Silindrsel Elektrot Sisteminin Delinme Bakımından İncelenme Sonuçları

	Analitik hesap	Genetik Algoritma Sonucu	Bağıl Hata
α (m)	5.518191618	5.5179248768	0.00004833
$(E_{\max})_{\min}$ (V/m)	18.12187886	18.12275488	0.00004834



Şekil 3.4 Silindirsel elektrot sisteminde α ve $E_{\max}$ 'ın r_1 ile değişimi

3.2 Düzgün Alanda Delinme Gerilimi

Bir elektronun elektrik alan doğrultusunda 1 cm'lik yol alması durumunda çarpma suretiyle meydana getirdiği iyon çifti sayısına Townsend'in birinci iyonlaştırma katsayısı denir ve α ile gösterilir.

Bir pozitif iyonun elektrik alan doğrultusunda 1 cm'lik yol alması durumunda çarpma suretiyle meydana getirdiği iyon çifti sayısına iyonun iyonlaştırma katsayısı denir ve β ile gösterilir. Bu sayı α nın yanında çok küçük olduğundan çoğu kez ihmal edilir. Bunun yanında bir pozitif iyonun katot yüzeyinden kopardığı elektron sayısı γ ile gösterilir, Townsend'in ikinci iyonlaştırma katsayısı olarak adlandırılır ve ihmal edilemez [9,11].

Bir elektronun katottan anoda gidene kadar meydana getirdiği elektron sayısı $e^{\alpha a}$, pozitif iyon sayısı $e^{\alpha a} - 1$ ve bu pozitif iyonların katottan çözdüğü elektron sayısı $\gamma(e^{\alpha a} - 1)$ olacaktır.

Bu sayı bire eşit olması, katottan anoda yol alan elektronun kendisi için bir adet yedek elektron hazırladığı anlamına gelir ki bu da boşalmanın kendi kendini beslemesi demektir. Bu nedenle,

$$\gamma(e^{\alpha a} - 1) = 1$$

denklemini boşalmanın kendi kendini besleme koşulu olarak adlandırılır. Bu denklem,

$$\alpha a = \ln \left(1 + \frac{1}{\gamma} \right) \quad (3.27)$$

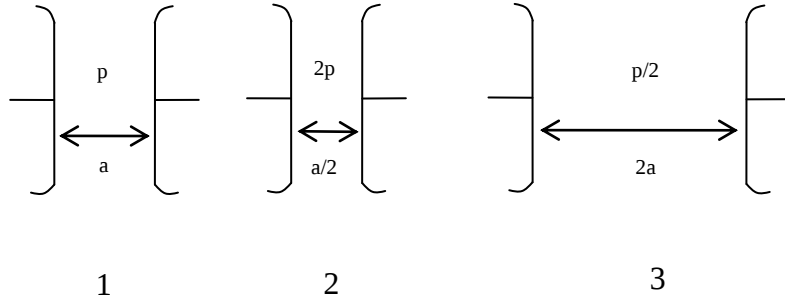
şeklinde de yazılabilir. Buradaki αa , elektronun toplam iyonlaştırma katsayısını gösterir.

$\alpha = p \cdot f_1(E/p)$ ve $\gamma = f_3(E/p)$ olduğu düşünülüp E yerine de U_d/a konulursa (3.27) numaralı denklem şu hali alır

$$pa f_1(U_d / pa) = \ln \left[1 + \frac{1}{f_3(U_d / pa)} \right] \quad (3.28)$$

Buradaki U_d boşalmanın kendi kendini besleme koşuluna karşı gelen delinme gerilimidir.

Düzgün alanda ve sabit sıcaklıkta delinme geriliminin, basınç ile elektrotlar arası açıklık çarpımının bir fonksiyonu olduğu (3.28) denkleminde görülmektedir. Buradan yola çıkılarak basıncı p ve açıklığı a olan bir düzlemsel elektrot ile basıncı 2p ve açıklığı a/2 olan veya basıncı p/2 ve açıklığı 2a olan düzlemsel elektrot sistemlerinin delinme gerilimleri birbirine eşittir (Şekil 3.5) [9,12].



Şekil 3.5 Delinme geriliminin pa ile değişimi

Bu bağıntı ilk defa Paschen tarafından deneysel olarak bulunduğu için Paschen yasası olarak adlandırılır.

(3.27) numaralı denklemde α yerine Ape^{-Bpa/U_d} konur $\gamma = \text{sabit}$ kabul edillir ve U_d yalnız bırakılırsa (3.28) numaralı denklemden

$$U_d = \frac{Bpa}{\ln \left[\frac{Apa}{\ln(1+1/\gamma)} \right]} = f(pa) \quad (3.29)$$

olarak bulunur. Bu ifadedeki γ değeri E/p ve elektrot malzemesine bağlı olmakla birlikte değeri 0.02 ile 0.0002 arasında değişir.

Denklem (3.23)'de A, B, γ sabit olduğundan, $U_d = f(pa)$ eğrisinin minimumunu bulmak için $dU_d / d(pa) = 0$ yapılarak $(pa)_{kr}$ ve buna karşılık gelen $(U_d)_{min}$ değerleri bulunabilir (3.30 ve 3.31).

$$(pa)_{kr} = \frac{e \cdot \ln(1+1/\gamma)}{A} \quad (3.30)$$

$$(U_d)_{min} = B(pa)_{kr} \quad (3.31)$$

Tablo 3.6'da $(pa)_{kr}$ ve $(U_d)_{min}$ değerleri bilinen bir gazın GA programı ile bulunan sonuçlar gösterilmiştir.

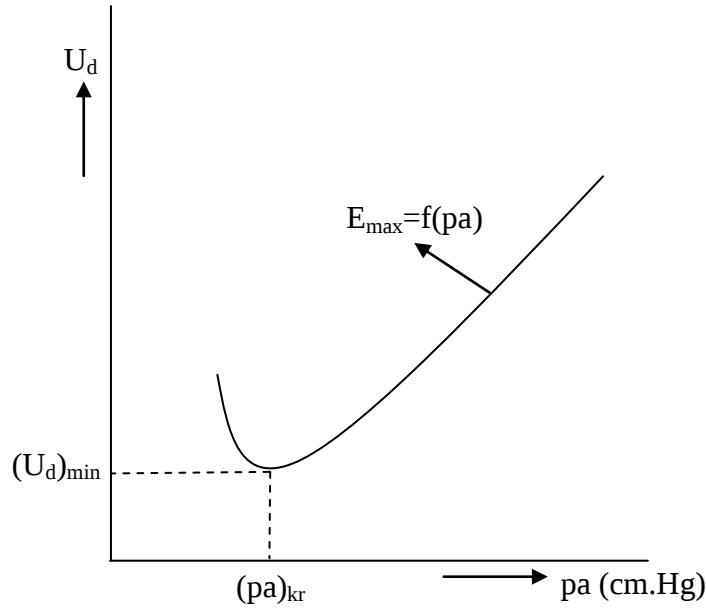
Tablo 3.6 Değerleri bilinen gazın $(pa)_{kr}$ ve minimum delinme dayanımının GA ile hesaplanması

Gaz	Analitik Hesap		GA sonuçları		
	$(pa)_{kr}$	$(U_d)_{min}$	$(pa)_{kr}$	$(U_d)_{min}$	Bağıl Hata
	(mmHg.cm)	(V)	(mmHg.cm)	(V)	(%)
	0.89	325.087646	0.889999	325.0878019	0.0000022

Hesaplama için seçilen büyüklüklerin değerleri EK B' de gösterilmiştir.

Buraya kadar anlatılanlardan anlaşıldığı gibi bir yalıtkan maddenin delinme dayanımı, ilk bakışta sanıldığı gibi sabit bir değer değildir. $U_d = f(pa)$ denkleminde üç büyüklükten biri parametre olarak alınırsa diğer iki büyüklük arasında bağıntılar görülebilir.

Şekil 3.6' da delinme geriliminin basınç-elektrot açıklığı ile değişim grafiği gösterilen sistemde, basınç sabit iken delinme geriliminin elektrot açıklığı ile, elektrot açıklığı sabit iken delinme geriliminin basınç ile değişimini içeren analitik hasaplama ve genetik algoritma hasaplama sonuçları Tablo 3.7' de görülebilir.



Şekil 3.6 Delinme geriliminin basınç-elektrot açıklığı ile değişimi

Tablo 3.7 a'nın ve p'nin sabit olduğu durumlarda U_d değişimi

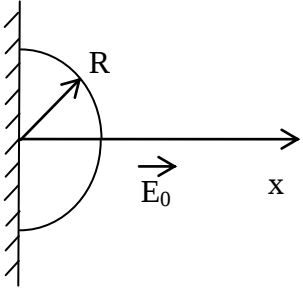
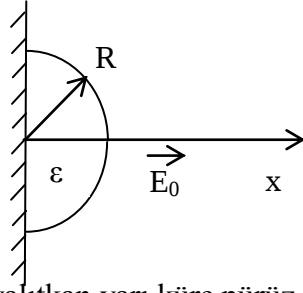
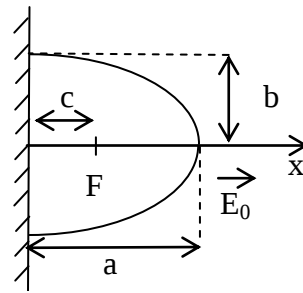
Analitik hesap			Genetik algoritma			Bağıl hata	
a(cm)	p(mmHg)	U _d	a(cm)	p(mmHg)	U _d	ε _{a,p}	ε _{U_d}
p: sbt	a: sbt	(V)	p: sbt	a: sbt	(V)	(%)	(%)
5	0.178	325.087646	5	0.17835561	325.087819	0.00199	0.000053
0.117105	7.6	325.087646	0.1188427	7.6	325.119097	0.01483	0.000096

Hesaplamalara ait büyüklükler EK B'de verilmiştir.

3.3 Farklı Geometrilerin Elektriksel Alan Üzerindeki Etkileri

İletken veya yalıtkan malzemelerin tasarımında malzeme geometrisi önemli bir rol oynamaktadır. Malzeme üzerinde var olabilecek farklı geometrideki pürüzler malzemenin delinme dayanımını değiştirecek niteliktedirler. Tablo 3.8'de farklı geometrik şekillerdeki pürüzlerin şekilleri ve elektriksel bağıntıları görülmektedir.

Tablo 3.8 Farklı geometrideki pürüzlerde alan ifadeleri

Durum	$E_{\max}$	Analitik İfade
 iletken yarı küre pürüz	$3E_0$	$E = E_0 \left(1 + 2 \frac{R^3}{x^3}\right)$
 yalıtkan yarı küre pürüz	$\frac{3\varepsilon}{2\varepsilon_0 + \varepsilon} E_0$	$E = E_0 \left(1 + 2 \frac{\varepsilon - \varepsilon_0}{2\varepsilon_0 + \varepsilon} \frac{R^3}{x^3}\right)$
 dik elipsoid pürüz	$\frac{\frac{c^3}{ab^2} E_0}{\frac{1}{2} \ln \frac{a+c}{a-c} - \frac{c}{a}}$	$E = E_0 + E_{\max} \frac{ab^2}{c^3} \left(\frac{cx}{x^2 - c^2} - \frac{1}{2} \ln \frac{x+c}{x-c} \right)$

Tablo 3.8’de verilen ifadelerin analitik hesapla ve GA ile bulunan sonuçları Tablo 3.9’da gösterilmektedir. Hesaplama ve GA parametreleri EK B’de verilmiştir.

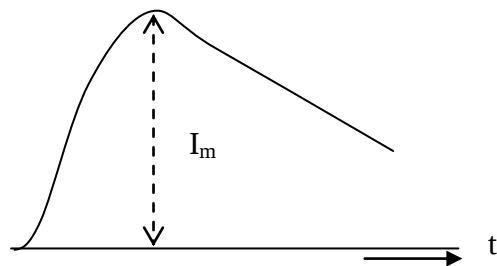
Tablo 3.9 Hesaplama sonuçları

	$E_{\max}$ (V) (Analitik hesap)	$E_{\max}$ (V) (GA)	Bağıl Hata (%)
İletken yarı küre pürüz	30	29,995989	0.0001334000
Yalıtkan yarı küre pürüz	15.71428	15.716238	0.000012460
Dik elipsoidal pürüz	18.11649	18.11637	0.000006623

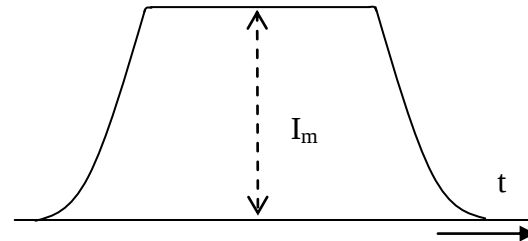
3.4 Darbe Akım Üreticinin Akım Karakteristiğinin İncelenmesi

Darbe akımları yüksek gerilimlere bağlı olarak ortaya çıkmaktadır. Bu tür akımlar delinmeler sonucu sıkça meydana gelerek yüksek sıcaklıklara ve büyük mekanik kuvvetlere neden olurlar. Bir darbe akımı genellikle çok kısa sürede tepe değerine ulaşarak yavaş bir şekilde söner.

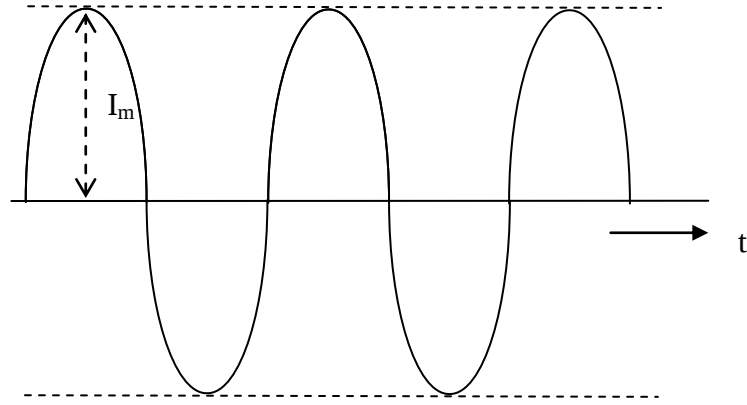
Uygulamada meydana geliş şekillerine göre çeşitlilik gösterirler. Yıldırım sonucu meydana gelen akımlar çift ekponansiyel darbe akımı (Şekil 3.7), uzun enerji iletim hatlarının boşalması sırasında meydana gelen akımlar dikdörtgen darbe gerilimi (Şekil 3.8), alternatif akım devrelerinde kısa devre sonucu meydana gelen akımlara doğru bileşensiz sinüzoidal akım (Şekil 3.9) veya doğru bileşeni üstel olarak azalan sinüzoidal darbe gerilimi (Şekil 3.10) denmektedir [6,9,10,12].



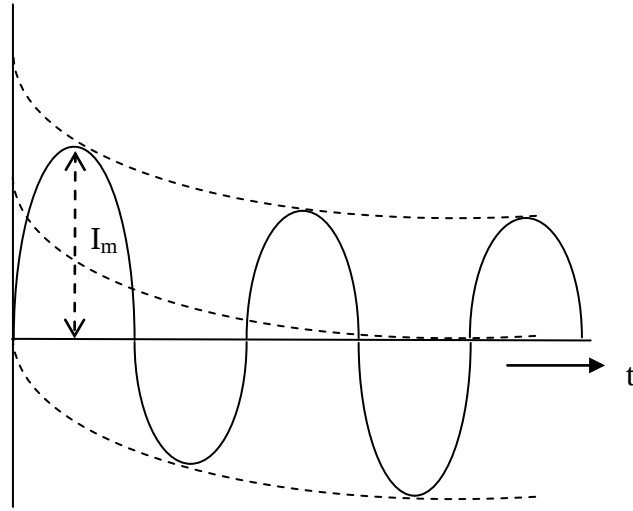
Şekil 3.7 Çift Ekponansiyel Darbe akımı



Şekil 3.8 Dikdörtgen darbe akımı



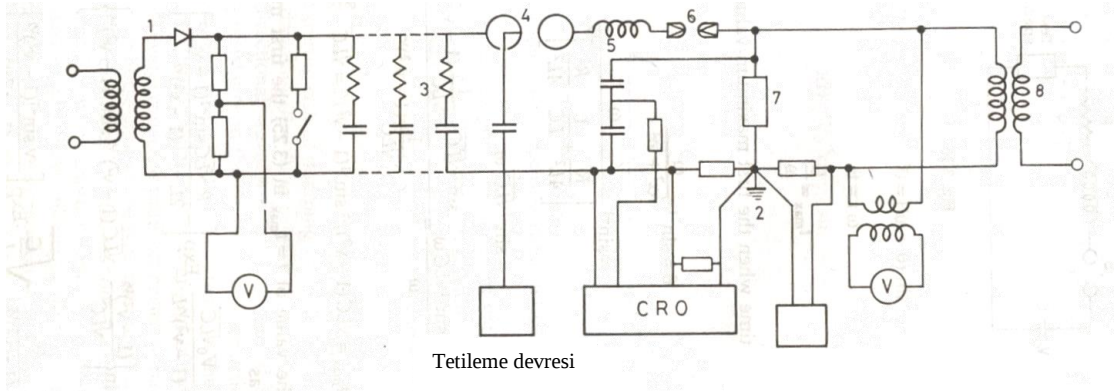
Şekil 3.9 Doğru bileşensiz sinüzoidal darbe akımı



Şekil 3.10 Doğru bileşeni ekponansiyel olarak azalan sinüzoidal darbe akımı

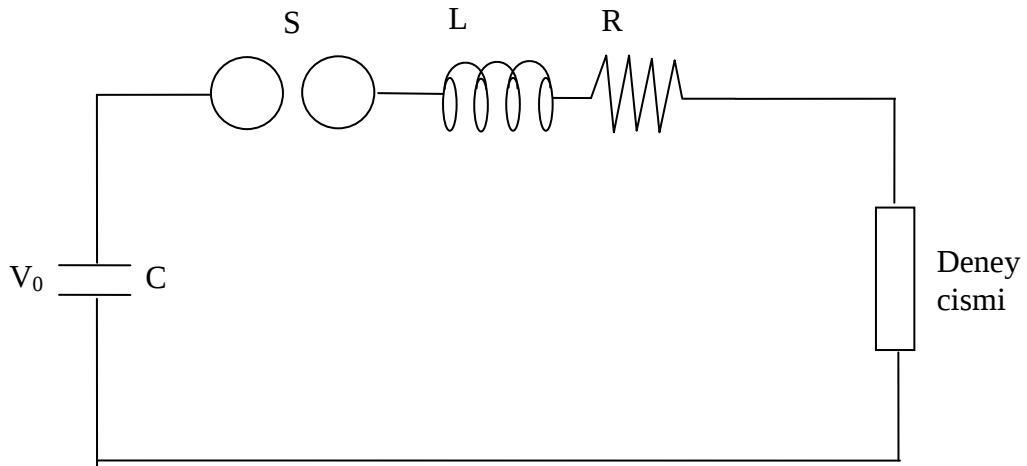
İstenilen biçim ve genlikli yüksek darbe akımlarının üretilebilmesi güç kaynağına ek olarak enerji depolayıcı sistemlere ihtiyaç vardır. Bu sistemler kapasitif enerji depolama, endüktif enerji depolama ve mekanik enerji depolama şeklinde karşımıza çıkarlar [6,10].

Yüksek akım generatörleri, birbirine paralel olarak bağlanmış birçok darbe kondansatöründen oluşmaktadır (Şekil 3.11).



Şekil 3.11 Yüksek Akım Generatörü yapısı

(Şekil 3.12)'de eşdeğer devresi gösterilen darbe üreticinde C darbe kondansatörü U_0 doğru gerilimi ile doldurulur ve S küresel elektrotlarında atlama meydana gelmesi ile L, R ve deney cismi üzerinden boşalır.



Şekil 3.12 Darbe üretici eş değer devresi

Bu eşdeğer devreye göre S küresel elektrotlarında atlama meydana geldiğinde darbe akımını Laplace dönüşümü ile,

$$I(s) = \frac{V_0}{s} \frac{1}{R + sL + 1/cs} \quad (3.32)$$

$$= \frac{V}{L} \frac{1}{(s + \alpha)^2 + w^2}$$

Bu denklemde $\alpha = \frac{R}{2L}$ ve $w = \left(\frac{1}{LC} - \frac{R^2}{4L^2} \right)^{\frac{1}{2}}$ olarak ifade edilir. (3.33)

(3.32) numaralı denklemin ters Laplace'ı alındığında darbe akım denklemi

$$i(t) = \frac{V}{wL} e^{-\alpha t} \sin wt \quad (3.34)$$

halini alır.

Eşdeğer devre ve GA parametreleri EK B'de verilen bir sistemin elde edilen maksimum darbe akım değerleri Tablo 3.10'da verilmiştir.

Tablo 3.10 Maksimum akım hesaplama sonuçları

	Analitik hesap	GA sonucu	Bağıl Hata (%)
t (μs)	260	264.26973	0.01642
I _{max} (kA)	10.13	10.131797	0.00017739

4. SONUÇ VE ÖNERİLER

Yüksek gerilim tekniğinde genetik algoritma uygulamaları konulu tez çalışması, iki temel bölümden oluşmaktadır.

Birinci bölümde genetik algoritmalar hakkında genel bilgi verilmiştir. Genetik algoritmalar temeli doğal seçim ve doğal kalıtım kurallarına dayalı bir optimizasyon yöntemidir ve geleneksel optimizasyon yöntemlerine göre üstünlükleri bulunmaktadır.

Geleneksel yöntemler aramaya tek bir noktadan başladıkları için ilk buldukları yerel optimum noktasında aramayı sonlandırırlar. Dolayısıyla birden çok optimuma sahip fonksiyonlar için kullanışlı değildir. Genetik algoritmalar ise aramaya tek bir noktadan değil, aynı anda birçok noktadan başlayarak, incelenen fonksiyonun yerel optimum noktasına takılıp kalmazlar. Tüm yerel optimum noktaları bularak bunları karşılaştırır ve fonksiyonun evrensel optimum noktasına ulaşır.

Genetik algoritmalarda diğer yöntemlerde olduğu gibi türev bilgileri ve ek bilgilere ihtiyaç yoktur. Yalnızca amaç fonksiyonu ve uygunluk değerleri yaklaşımın hangi yönde ilerleyeceğini belirler. Dolayısıyla incelenen fonksiyonun sürekli veya süreksiz olması bu yöntem için sorun yaratmaz.

GA rastlantısal geçiş kurallarına bağlı olarak çalıştığından, model değiştirilmese bile her çalıştığında farklı çözümler elde edilir. Geleneksel yöntemler ise aynı değerler ile her çalıştığında aynı çözüme ulaşılır.

Ayrıca kodlama, çaprazlama ve mutasyon aşamaları da genetik algoritmaları diğer yöntemlerden ayıran diğer özelliklerdir. Bu aşamalarda kullanılan parametreler yöntemin sonuca ulaşmasında önemli roller üstlenmektedir. Kodlama ile problemin her bir çözümü sonlu bir katar halinde kodlanarak temsil edilir. Çaprazlama sayesinde de bir önceki çözüm kümesindeki sonuçlar birbirleriyle karışarak bir sonraki kuşakta daha uygun sonuçları oluştururlar. Mutasyon ise algoritmanın bir kısır döngüye girmemesi için güvenlik aracı görevi görür. Herhangi bir yerel optimum etrafında toplanıp kalan bireyler mutasyon ile başkalaşım geçirerek diğer çözümlere yönelebilirler.

Genetik algoritmalar, sözü edilen tüm bu özellikleri sayesinde diğer yöntemlerde karşımıza çıkan süreklilik, türev varlığı, tek bir optimuma sahip olma, lineer olup olmama gibi kısıtlamalardan kurtulmuş olur. Diğer yandan her bir çözüm aynı anda değerlendirildiği için yaklaşım hızı büyük çözüm kümeli problemlerde yavaş kalmaktadır.

Çalışmanın ikinci bölümünde, genetik algoritmaları yüksek gerilim tekniğindeki uygulamalarda kullanabilmek için uygun algoritma programı Visual Basic programlama dilinde yazılmış, uygulamalara değinilmiş ve analitik yolla hesaplanan değerler ile yazılan genetik algoritma programı sonucu elde edilen sonuçlar karşılaştırılmıştır. Yapılan bu hesaplamalarda topluluk sayısı, nesil sayısı, çaprazlama olasılığı, mutasyon olasılığı gibi genetik algoritma parametreleri değiştirilerek sonucu ne şekilde etkilediği görülmüş ve buna göre uygun parametreler seçilmiştir.

Elektrot sistemlerindeki maksimum zorlanma, ortamın delinme dayanımına eşit veya büyük olduğunda, eğrilik yarıçapı küçük olan elektrot üzerinde boşalma başlar ve bu elektrotların tam delinmesiyle sonuçlanır. Yüksek gerilim hava hatları, kablolar, geçit izolatörleri ve daha birçok sistem tasarımları için küresel ve silindrsel elektrot sistemlerini delinme bakımından zorlayacak elektriksel alanların bilinmesi gerekir. Ayrıca bu tasarımların ekonomik olarak da uygun olması gerekliliği göz önünde bulundurulmalıdır. Yapılan çalışmada maksimum alan, minimum alan ve ekonomik bakımdan en uygun düzen için geometrik karakteristik ifadeleri çıkartılmış daha sonrasında bu ifadelerin çözümü analitik yoldan ve GA ile bulunarak yaklaşımın doğruluğu sınanmıştır.

Düzgün alanda ve sabit sıcaklıkta delinme geriliminin, basınç ile elektrotlar arası açıklık çarpımının bir fonksiyonudur. Paschen yasası olarak bilinen ve delinme geriliminin basınç-açıklık ile değişiminin ifadesi olan bu denklem çıkartılarak genetik algoritmalar kullanılarak hesaplama yapılmış ve yaklaşımın doğruluğu görülmüştür.

İletken veya yalıtkan malzemelerin tasarımında malzeme geometrisi önemli bir rol oynamaktadır. Malzeme üzerinde var olabilecek farklı geometrideki pürüzler malzemenin delinme dayanımını değiştirecek niteliktedirler. Farklı geometrilerdeki pürüzlerin iletken ve yalıtkan malzemelerin delinme dayanımlarını ne şekilde etkilediğine ilişkin ifadeler verilerek bu ifadeler GA ile çözümlenmiştir.

Son uygulama olarak da darbe akım üreticinin akım karakteristikleri incelenmiştir. Darbe akımları yüksek gerilimlere bağlı olarak ortaya çıkmaktadır. Bu tür akımlar delinmeler sonucu sıkça meydana gelerek yüksek sıcaklıklara ve büyük mekanik kuvvetlere neden olurlar. Bir darbe akımı genellikle çok kısa sürede tepe değerine ulaşarak yavaş bir şekilde söner. Eşdeğer devresi çıkartılan bir akım üreticinin hesaplamaları analitik ve genetik algoritmalar programı ile yapılarak akım değerinin maksimum değeri bulunmuştur.

Yapılan çalışmalar, yüksek gerilim tekniğindeki tek amaç fonksiyonlu tek değişkenli bu uygulamalar için, genetik algoritmaların kullanılmasının, etkili ve kesin sonuçlar verdiğini göstermektedir. Genetik algoritmalar kullanılarak, doğrusal olmayan sistemler için daha kolay ve hızlı şekilde çözüme ulaşılabilir. Genetik algoritma ile bulunan sonuçlar ve analitik hesap ile bulunan sonuçların uyumu, yöntemin doğruluğunu göstermektedir.

Daha önce de belirtildiği gibi bu çalışmada tek amaç fonksiyonlu tek değişkenli uygulamalara değinilmiş ve yazılan program bu kriterler göz önünde bulundurularak hazırlanmıştır. Benzer işlemleri çok amaç fonksiyonlu ve çok değişkenli uygulamalarda gerçekleştirebilmek için programın uygun bölümleri uygun şekilde düzenlenmelidir.

EK M

KAYNAKLAR

- [1] **Baytur, H.**, 2002. Genetik, Süleyman Demirel Üniv. Yayınları, Isparta.
- [2] **Gen, M. and Cheng, R.**, 1997. Genetic Algorithms and Engineering Design, John Wiley&Sons Inc., New York.
- [3] **Golberg, D.E.**, 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Longman Inc., ???.
- [4] **Halvorson, M.**, 2000. Microsoft Visual Basic 6.0 Profesional, Arkadaş Yayınları, Ankara.
- [5] **Karr, C.L. and Freeman, M.**, 1999. Industrial Applications of Genetic Algorithms, CRC Press, New York.
- [6] **Kind, D.**, 1978. An Introduction to High Voltage Experimental Technique: textbook for electr. Engineers, Braunschweig: Vieweg
- [7] **Man, K.F. and Tang, K.S Kwong, S., Halang W.A.**, 1997. Genetic Algorithms for Control and Signal Processing, Springer-Verlag, London
- [8] **Michalewicz, Z.**, 1992. Genetic Algorithm + Data Structures = Evolution Programs, Springer-Verlag, New York.
- [9] **Özkaya, M.**, 1996. Yüksek Gerilim Tekniği, Cilt 1, Birsen Yayınevi, İstanbul.
- [10] **Özkaya, M.**, 1996. Yüksek Gerilim Tekniği, Cilt 2, Birsen Yayınevi, İstanbul.
- [11] **Starr, C.**, 1991. Biology: Concepts and Applications, Wodsworth Publishing Comp., California
- [12] **Wadhwa, C.L.**, 1995. Hig Voltage Engineering, New Age International Limited Publishers, India
- [13] **Aygen, Z.E.**, 2002. Elektrik Enerji Sistemlerinde Genetik Algoritma Kullanarak Optimizasyona Yeni Bir Yaklaşım, *Doktora Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [14] **Artaç, T.**, 2003. Genetik Algoritmalar ile Dağıtım Şebekelerinin Optimum tasarımı, *Yüksek Lisans Tezi*, , İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

- [15] **Çelebi, M.**, 2003. Yeni Bir Genetik Algoritma Yaklaşımı ile Çıkık Kutuplu Senkron Makinanın Ağırlık Optimizasyonu, Doktora Tezi, Y.T.Ü Fen Bilimleri Enstitüsü, İstanbul
- [16] **Demirel .**, 2000. Genetik Algoritmalar ile ..., Doktora Tezi, İ.T.Ü Fen Bilimleri Enstitüsü, İstanbul
- [17] **Dawei Li; Li Wang**, 2002. A study on the optimal population size of genetic algorithm, 2002 Proceedings of the 4th World Congress on Intelligent Control and Automation, Vol. 4, pp. 3019 - 3021.
- [18] **Dinger, R. H.**, 1998. Engineering design optimization with genetic algorithms, Northcon/98 Conference Proceedings, pp. 114 – 119.
- [19] **Etaner, Ş.**, 1994. Genetik Algoritmalar, *Otomasyon*, 108-112.
- [20] **Franceschini, G., Piazzzi, A., Tassoni, C.**, 1994. A genetic algorithm approach to design flux observer for induction servo motors, IECON '94., 20th International Conference on Industrial Electronics, Control and Instrumentation, Vol. 3, pp. 2132 – 2136.
- [21] **Frenzel, J. F.**, 1993. Genetic algorithms, IEEE Potentials, Vol. 12, Issue: 3, pp. 21 – 24.
- [22] **Fukuyama, Y., Hsaio-Dong Chiang**, 1995. A parallel genetic algorithm for service restoration in electric power distribution systems, Proceedings of 1995 IEEE International Conference on Fuzzy Systems, Vol. 1, pp. 275 – 282.
- [23] **Haida, T., Akimoto, Y.**, 1991. Genetic algorithms approach to voltage optimization, Proceedings of the First International Forum on Applications of Neural Networks to Power Systems, pp. 139 – 143.
- [24] **Herrmann, J. W.**, 1999. A genetic algorithm for minimax optimization problems, CEC 99 Proceedings of the 1999 Congress on Evolutionary Computation, Vol. 2, p. 1103.
- [25] **Holland, J.**, 1992. Genetic Algorithms, *Scientific American*, 44-50.

- [26] **Horrocks, D. H., Spittle, M. C.**, 1992. Genetic algorithms, IEE Colloquium on Circuit Theory and DSP, pp. 4/1 - 4/5.
- Iba, K.**, 1994. Reactive power optimization by genetic algorithm, IEEE Transactions on Power Systems, Vol. 9, Issue: 2, pp. 685 – 692.
- [27] **Johnson, J. M., Rahmat-Samii, V.**, 1997. Genetic algorithms in engineering electromagnetics, IEEE Antennas and Propagation Magazine, Vol. 39, Issue: 4, pp. 7 – 21.
- [28] **Karr, C.L.**, 1994. Least Median Squares Curve Fitting Using a Genetic Algorithm, Engineering Applications Artificial Intelligence, Vol.8, No.13, 177-189.
- [29] **Lee, K.Y., Bai, X. and Parks, Y.M.**, 1998. Optimal Reactive Power Planning Using Evolutionary Algorithms: A Comparative Study for Evolutionary Programming, Evolutionary Strategy, Genetic Algorithms and Linear Programming, *IEEE Transactions on Power Systems*, Vol.13, No.1, 101-108.
- [30] **Ovaska, S. J., Bose, T., Vainio, O.**, 2003. Genetic algorithm-aided design of predictive filters for electric power applications, 2003 IEEE International Conference on Systems, Man and Cybernetics, Vol. 2, pp. 1463 – 1468.
- [31] **Ramirez-Rosado, I. J., Bernal-Agustin, J. L.**, 1998. Genetic algorithms applied to the design of large power distribution systems, IEEE Transactions on Power Systems, Vol. 13, Issue: 2, pp. 696 – 703.
- [32] **Reformat, M., Kuffel, E., Woodford, D., Pedrycz, W.**, 1998. Application of genetic algorithms for control design in power systems, IEE Proceedings- Generation, Transmission and Distribution, Vol. 145, Issue: 4, pp. 345 – 354.
- [33] **Samaan, N., Singh, C.**, 2002. Adequacy assessment of power system generation using a modified simple genetic algorithm, IEEE Transactions on Power Systems, Vol. 17, Issue: 4, pp. 974 – 981.

- [34] **Sareni, B., Krahenbuhl, L., Nicolas, A.**, 1998. Niching genetic algorithms for optimization in electromagnetics. I. Fundamentals, IEEE Transactions on Magnetism, Vol. 34, Issue: 5, Sept. 1998, pp. 2984 – 2987.
- [35] **Srinivas, M., Patnaik, L. M.**, 1994. Genetic algorithms: a survey, Computer, Vol. 27, Issue: 6, pp. 17 – 26.
- [36] **Stender, J.**, 1994. Introduction to genetic algorithms, IEE Colloquium on Applications of Genetic Algorithms, pp. 1/1 – 1/4.
- [37] **Wang, Z. H., Yin, X. G., Zhang, Z., Yang, J. C.**, 2003. Pseudo-parallel genetic algorithm for reactive power optimization, IEEE Power Engineering Society General Meeting, Vol. 2, p. 907.
- [38] **Wang, H. T., Liu, Z. J., Low, T. S., Ge, S. S., Bi, C.**, 2000. A genetic algorithm combined with finite element method for robust design of actuators, IEEE Transactions on Magnetism, Vol. 36, Issue: 4, pp. 1128 – 1131.
- [39] **Weile, D. S., Michielssen, E.**, 1997. Genetic algorithm optimization applied to electromagnetics: a review, IEEE Transactions on Antennas and Propagation, Vol. 45, Issue: 3, pp. 343 – 353.
- [40] **Wenzel, D., Borsi, H., Gockenbach, E.**, 1996. A new approach for partial discharge recognition on transformers on-site by means of genetic algorithms, Conference Record of the 1996 IEEE International Symposium on Electrical Insulation, Vol. 1, pp. 57 - 60.
- [41] **Vasconcelos, J. A., Ramirez, J. A., Takahashi, R. H. C., Saldanha, R. R.**, 2001. Improvements in genetic algorithms, IEEE Transactions on Magnetism, Vol. 37, Issue: 5, pp. 3414 – 3417.
- [42] **Yarushkina, N. G.**, 2002. Genetic algorithms for engineering optimization: theory and practice, ICAIS 2002 IEEE International Conference on Artificial Intelligence Systems, pp. 357 – 362.

- [43] **Ying-Hong Liao; Chuen-Tsai Sun**, 2001. An educational genetic algorithms learning tool, *IEEE Transactions on Education*, Vol. 44, Issue: 2, p. 20.
- [44] **Bäck, T.**, 1993. Optimal Mutation Rates in Genetic Search, *Proceedings of the Fifth International Conference on Genetic Algorithms*, Kaufmann Publishers, California, pp. 2-8.
- [45] <http://cs.felk.cvut.cz/~xobitko/ga/>
- [46] <http://www.personal.usyd.edu.au/~desm/ga>
- [47] <http://www.estec.esa.nl/outreach/gatutor>
- [48] http://www.epcc.ed.ac.uk/computing/training/document_archive/GA-slides
- [49] <http://www.aic.nrl.navy.mil/galist/>
- [50] <http://garage.cps.msu.edu/>
- [51] <http://www.geneticprogramming.com/>
- [52] <http://cs.gmu.edu/research/gag/pubs.html>
- [53] http://www.systemtechnik.tu-ilmenau.de/~pohlheim/GA_Toolbox

EK A GENETİK ALGORİTMA PROGRAM ALGORİTMA DETAYI

Visual Basic 6.0 kullanılarak yazılan genetik algoritma programında rulet çarkı yöntemi, tek nokta çaprazlaması, düzgün mutasyon işlemleri kullanılmıştır.

Program algoritması kısaca aşağıdaki gibi özetlenebilir.

1) Dışarıdan girilecek parametreler için arayüz oluşturulması

2) n_{bit} belirlenmesi

$$n_{bit} = n_{bit} + 1 \quad (n_{bit} = 0,1,2, \dots) \quad (A.1)$$

Koşul: $2^{n_{bit}} > a$ olduğunda döngüyü durdur ve o andaki n_{bit} 'i seç

3) c_k 'nın belirlenmesi

$$c_k = \frac{n_{bit}}{2} \quad (\text{eğer } n_{bit} \text{ çift ise}) \quad (A.2)$$

$$c_k = \frac{n_{bit} + 1}{2} \quad (\text{eğer } n_{bit} \text{ tek ise}) \quad (A.3)$$

4) n_{bit} uzunluğunda n_{pop} tane bireyin rastgele olarak üretilmesi $\Rightarrow$ krom_10

5) Üretilen bireylerin referans formülde yerine konularak belirtilen aralıktaki gerçek değerlerinin bulunması

$$krom_aralik[i] = \min + krom_10[i] \cdot \frac{\max - \min}{(2^{n_{bit}} - 1)} \quad (A.4)$$

6) Aralık (gerçek) değerlerin amaç fonksiyonunda yerine konulması ve herbir bireyin uygunluk değerinin bulunması

$$uygunluk[i] = f(krom_aralik[i]) \quad (A.5)$$

7) Toplam uygunluk değerinin bulunması

$$toplam_uygunluk = \sum_{i=0}^{n_{pop}-1} uygunluk[i] \quad (A.6)$$

8) Herbir bireyin seçilme olasılığının bulunması

$$p_secilme[i] = \frac{uygunluk[i]}{toplaml_uygunluk} \quad (A. 7))$$

9) Herbir bireyin birikimli olasılığının bulunması

$$p_birikim[i] = \sum_{i=0}^{n_{pop}-1} p_secilme[i] \quad (A.8)$$

10) 0-1 arasında rastgele n_{pop} tane sayı üretilmesi (üretilen sayı r olsun)

$$p_birikim[i-1] < r < p_birikim[i] \Rightarrow i. \text{ bireyi seç} \rightarrow \text{krom_temp}$$

11) Çaprazlamaya girecek $p_c \cdot n_{pop}$ tane bireyin rastgele seçilmesi $\rightarrow$ krom_temp2

12) Çaprazlamaya girmeyecek olan bireylerin bir sonraki nesil için doğrudan kopyalanması

13) Çaprazlama işleminin gerçekleşmesi

mask1 : n_{bit} uzunluğunda ve c_k noktasının sağını 0'ların, c_k noktasının solunu

1'lerin oluşturduğu maske bireyi

mask2 : n_{bit} uzunluğunda ve c_k noktasının sağını 1'lerin, c_k noktasının solunu

0'ların oluşturduğu maske bireyi

$$\text{birey}[k] \text{ VE mask1} = \text{birey}[k]_1$$

$$\text{birey}[k] \text{ VE mask2} = \text{birey}[k]_2$$

$$\text{birey}[j] \text{ VE mask1} = \text{birey}[j]_1$$

$$\text{birey}[j] \text{ VE mask2} = \text{birey}[j]_2$$

$$\text{birey}[k]_1 \text{ VEYA } \text{birey}[j]_2 \rightarrow \text{yavru}[k]$$

$$\text{birey}[k]_2 \text{ VEYA } \text{birey}[j]_1 \rightarrow \text{yavru}[j]$$

14) Mutasyon olasılığı gözönünde bulundurularak mutasyon geçirecek bireyin bulunması

$$mutasyon_krom = \frac{1}{n_{bit} \cdot p_m} \quad (A.9)$$

15) Mutasyon geçirecek bireyin hangi bit'inin mutasyona uğrayacağını bulunması

$$mutasyon_bit = \text{mod}_{n_{bit}} \left(\frac{1}{p_c} \right) \quad (\text{A.10})$$

16) n_{bit} uzunluğunda ve $mutasyon_bit$. bit'i 1, diğer bit'leri 0 olan mutasyon maske bireyinin oluşturulması $\rightarrow mutasyon_mask$

17) $mutasyon_krom$ XOR $mutasyon_mask$ işlemi gerçekleştirilerek mutasyon işleminin uygulanması

18) Topluluğu oluşturan yeni nesil hazır

19) Yeni nesil için aynı işlemleri 5. adımdan itibaren tekrar et

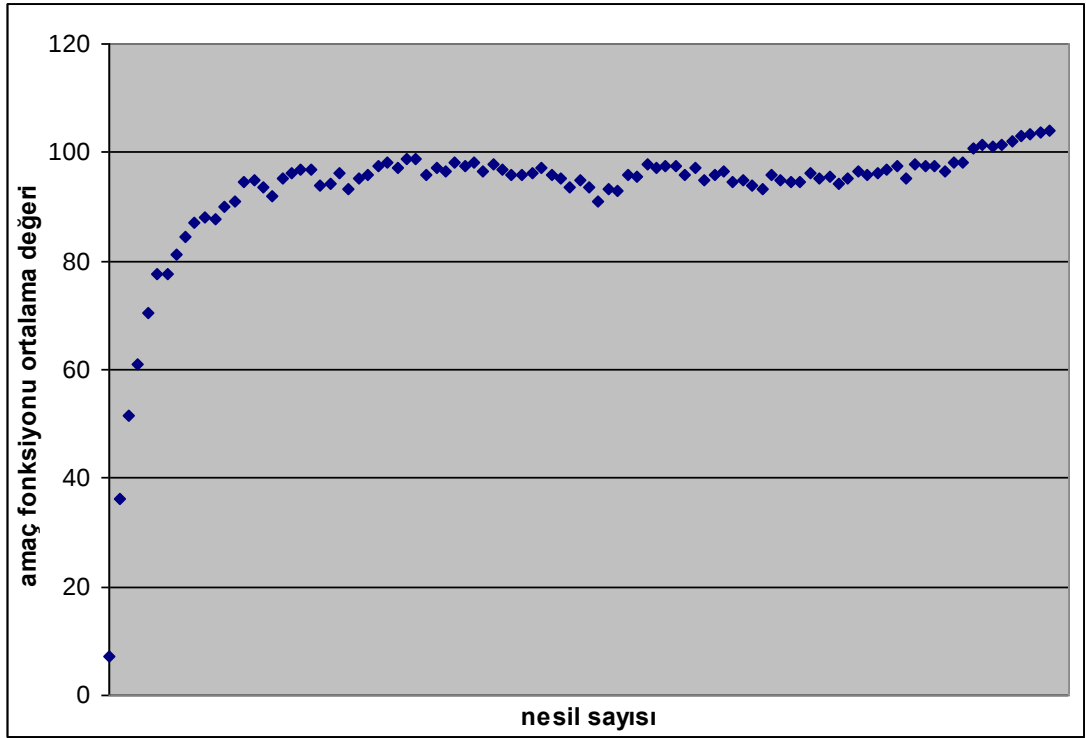
20) Girilen nesil sayısı kadar tekrarlama yapıldığında döngüyü durdur ve o nesildeki uygunluğu en yüksek bireyi çözüm olarak seç

EK B HESAPLAMA PARAMETRELERİ

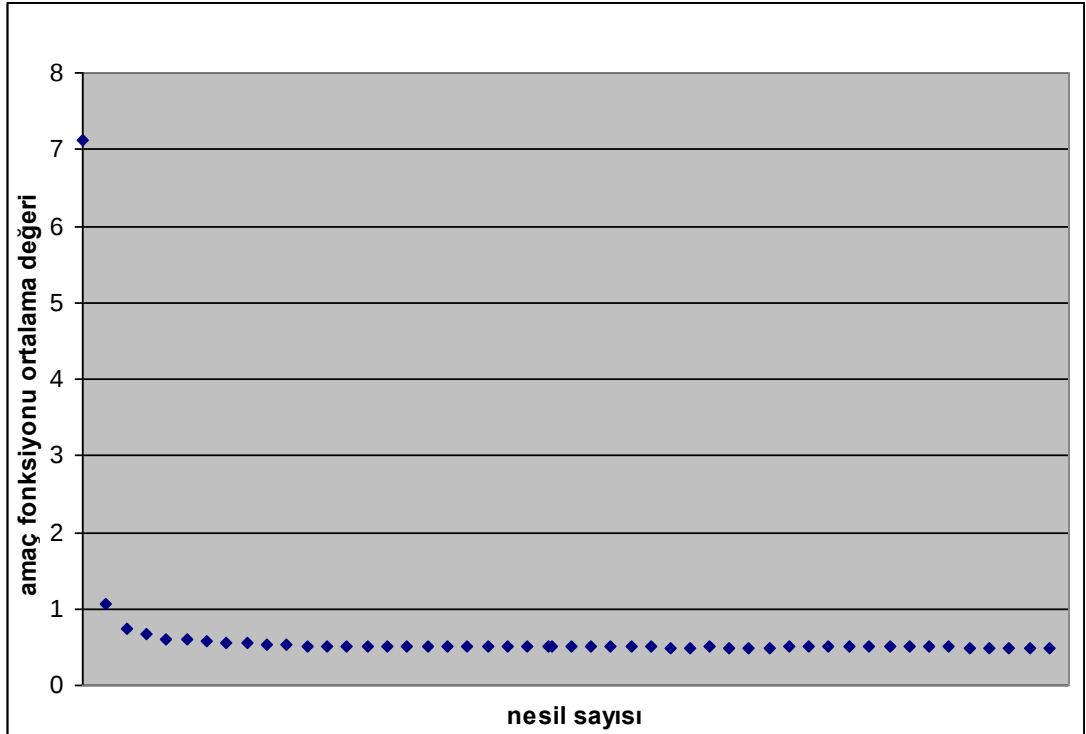
Küresel Elektrot Sisteminde Analitik ve GA Yaklaşımında Kullanılan Parametreler

Hesaplamanın Yapıldığı Tarih	: 05-16-2004
Hesaplamanın Yapıldığı Saat	: 11:34:04
n_pop	: 20/35/50/100/200
min (r_1)	: 1
max (r_2)	: 15
p_c	: 0.2
p_m	: 0.01
a	: 10000
nesil_sayisi	: 50

Genetik algoritma sonuçlarının analitik sonuçlara en yakın olduğu durum n_pop=100 olduğu durumdur. Bu durum için GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.1 ve B2’de gösterilmiştir. Grafikler n_pop = 100 esas alınarak çizdirilmiştir. Belirtilen değerler her bir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.



Şekil B.1 Küresel elektrot sistemi maksimum alan için amaç fonksiyon değerinin nesil ile değişimi

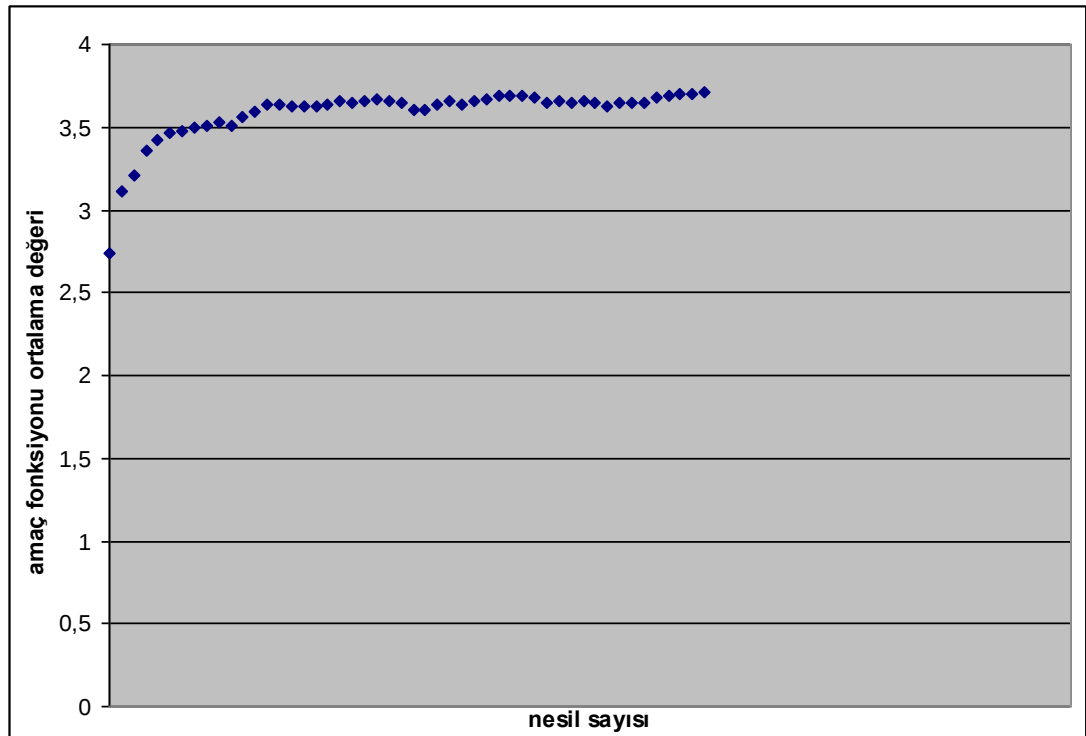


Şekil B.2 Küresel elektrot sistemi minimum alan için fonksiyon değeri-nesil değişimi

Küresel Elektrot Sisteminde α_{max} ve $(E_{max})_{min}$ hesabında kullanılan parametreler

Hesaplamanın Yapıldığı Tarih	: 06-06-2004
Hesaplamanın Yapıldığı Saat	: 18:30:11
n_pop	: 250
min (r_1)	: 1
max (r_2)	: 15
p_c	: 0.15/0.25/0.40/0.60
p_m	: 0.01
a	: 10000
nesil_sayisi	: 50

Genetik algoritma sonuçlarının analitik sonuçlara en yakın olduğu durum $p_c = 0.25$ olduğu durumdur. Bu durum için GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.3’de gösterilmiştir. Grafikler $p_c = 0.25$ esas alınarak çizdirilmiştir. Belirtilen değerler her bir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.

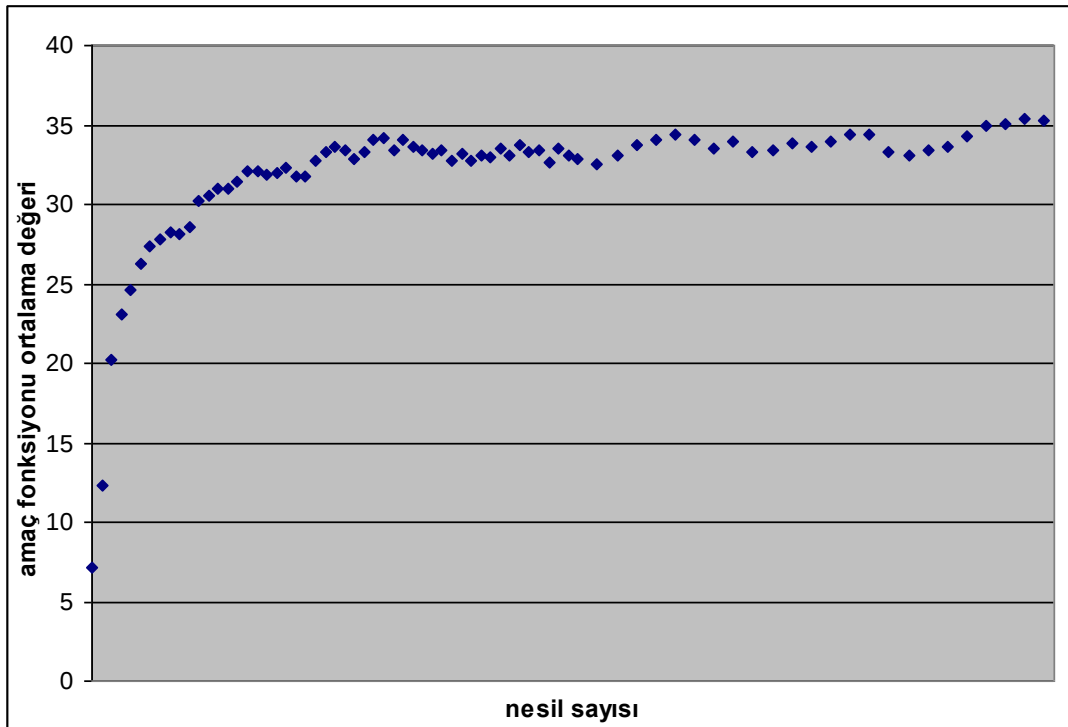


Şekil B.3 Küresel elektrot sisteminde α fonksiyon değeri - nesil değişimi

Silindrsel Elektrot Sisteminde Analitik ve GA Yaklaşımında Kullanılan Parametreler

Hesaplamanın Yapıldığı Tarih	: 05-15-2004
Hesaplamanın Yapıldığı Saat	: 22:15:25
n_pop	: 100
min (r_1)	: 1
max (r_2)	: 15
p_c	: 0.25
p_m	: 0.01/0.005/0.001
a	: 10000
nesil_sayisi	: 100

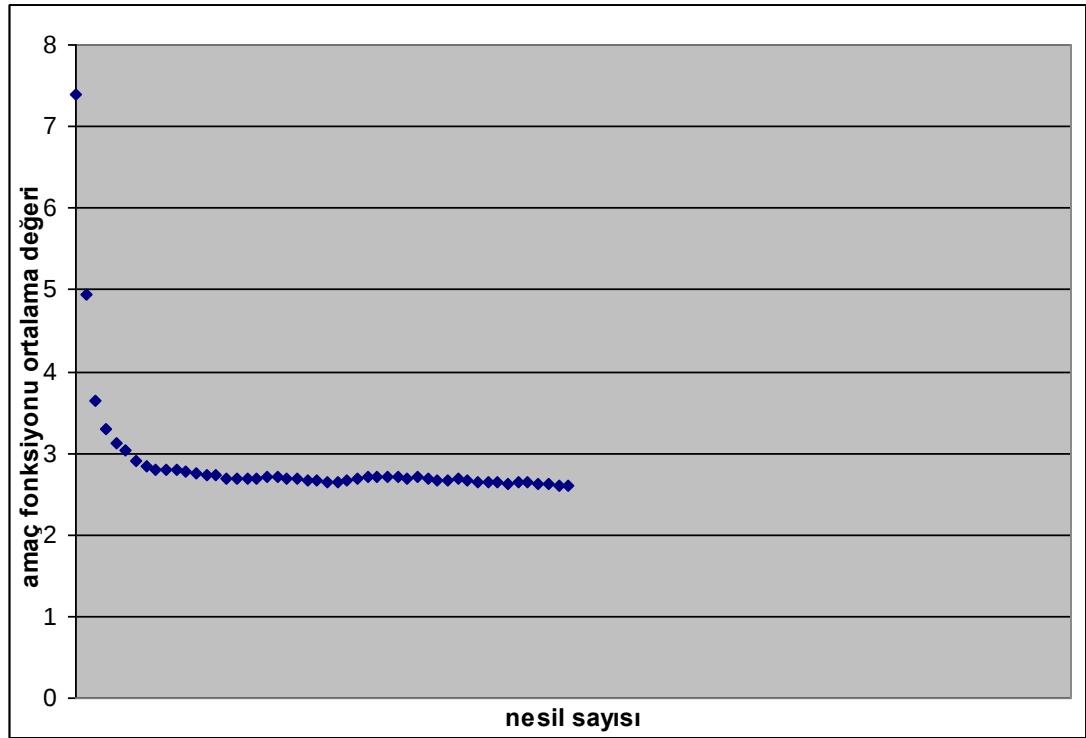
Genetik algoritma sonuçlarının analitik sonuçlara en yakın olduğu durum $p_m = 100$ olduğu durumdur. Bu durum için GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.4’de gösterilmiştir. Grafik $p_m = 0.001$ esas alınarak çizdirilmiştir. Belirtilen değerler her bir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.



Şekil B.4 Silindrsel elektrot sistemi maksimum alan için fonksiyon değeri-nesil değişimi

Hesaplamanın Yapıldığı Tarih	: 05-15-2004
Hesaplamanın Yapıldığı Saat	: 22:15:25
n_pop	: 200
min (r_1)	: 1
max (r_2)	: 15
p_c	: 0.25
p_m	: 0.01/0.005/0.001
a	: 10000
nesil_sayisi	: 50

Genetik algoritma sonuçlarının analitik sonuçlara en yakın olduğu durum $p_m = 100$ olduğu durumdur. Bu durum için GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.5’de gösterilmiştir. Grafik $p_m = 0.001$ esas alınarak çizdirilmiştir. Belirtilen değerler her bir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.



Şekil B.5 Silindrsel elektrot sistemi minimum alan için fonksiyon değeri-nesil değişimi

Silindirsel Elektrot Sisteminde α_{max} ve $(E_{max})_{min}$ hesabında kullanılan parametreler

Hesaplamanın Yapıldığı Tarih : 05-15-2004

Hesaplamanın Yapıldığı Saat : 23:15:07

n_pop : 200

min (r_1) : 1

max (r_2) : 15

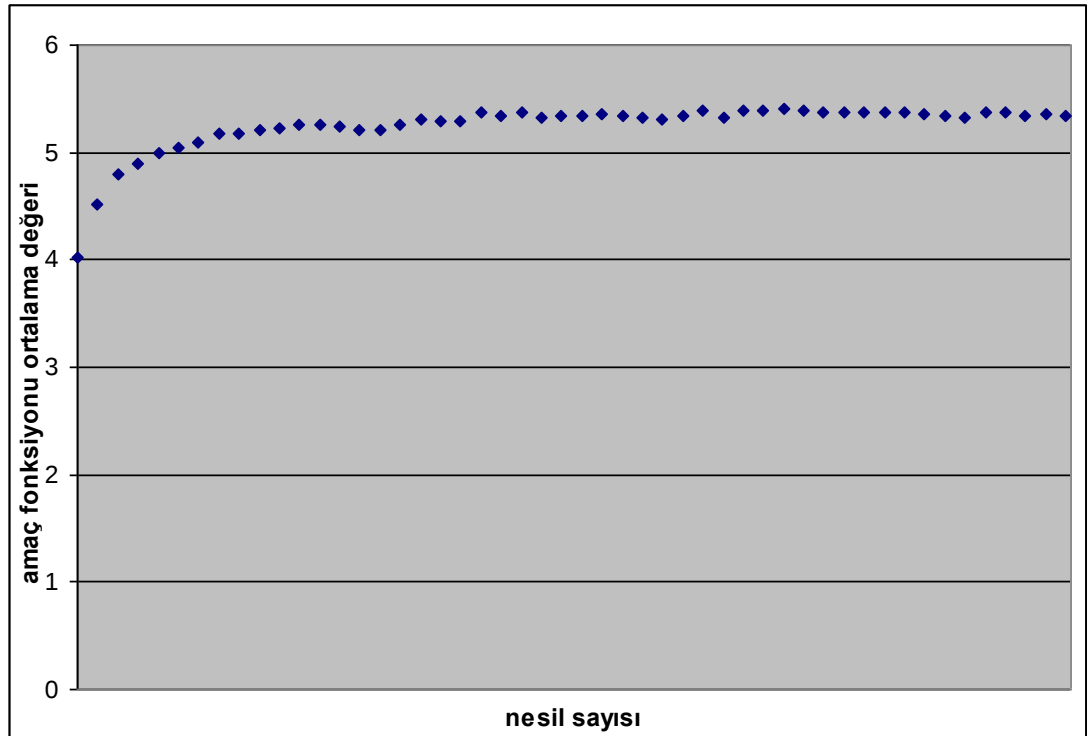
p_c : 0.25

p_m : 0.01

a : 10000

nesil_sayisi : 50

GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.6'da gösterilmiştir. Belirtilen değerler her bir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.



Şekil B.6 Silindirsel elektrot sisteminde α fonksiyon değeri - nesil değişimi

p geometrik karakteristiğinin hesabı için kullanılan parametreler

Hesaplamanın Yapıldığı Tarih : 05-15-2004

Hesaplamanın Yapıldığı Saat : 22:40:58

n_pop : 200

min : 1

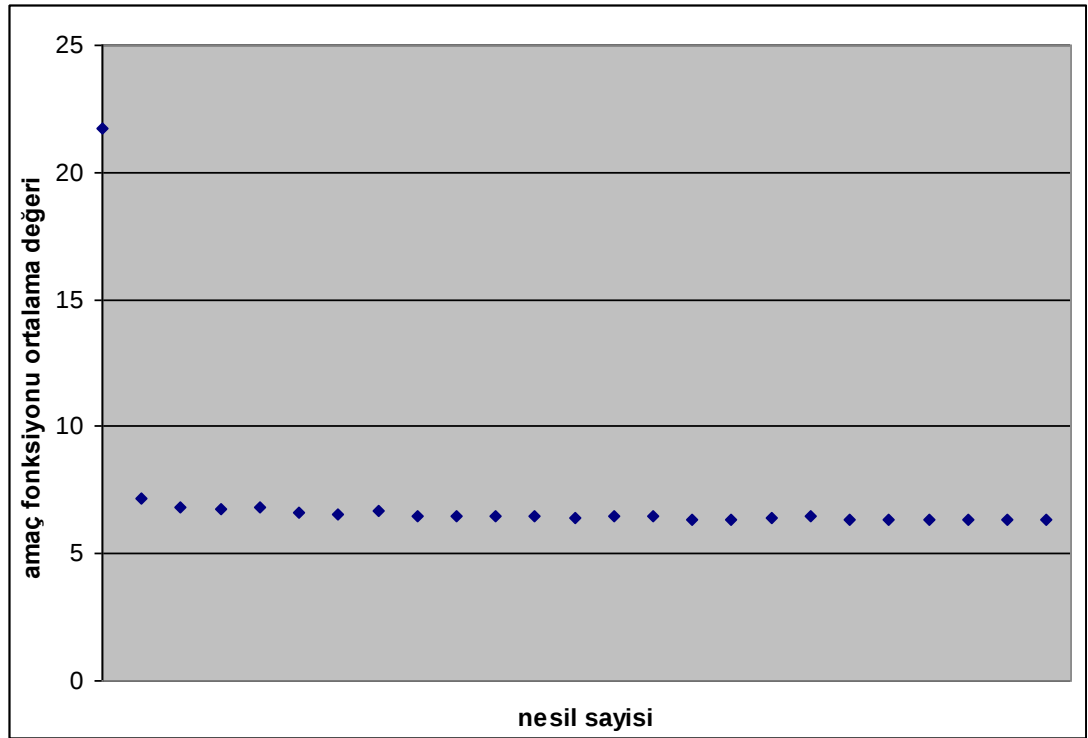
max : 5

p_c : 0.2

p_m : 0.001

a : 10000

nesil_sayisi : 50



Şekil B.7 Geometrik karakteristikte ağırlık-nesil sayısı değişimi

Düzgün alanda delinme geriliminin pa ile değişimi için kullanılan parametreler

Hesaplamanın Yapıldığı Tarih : 05-19-2004

Hesaplamanın Yapıldığı Saat : 19:48:58

n_pop : 100

min : 0

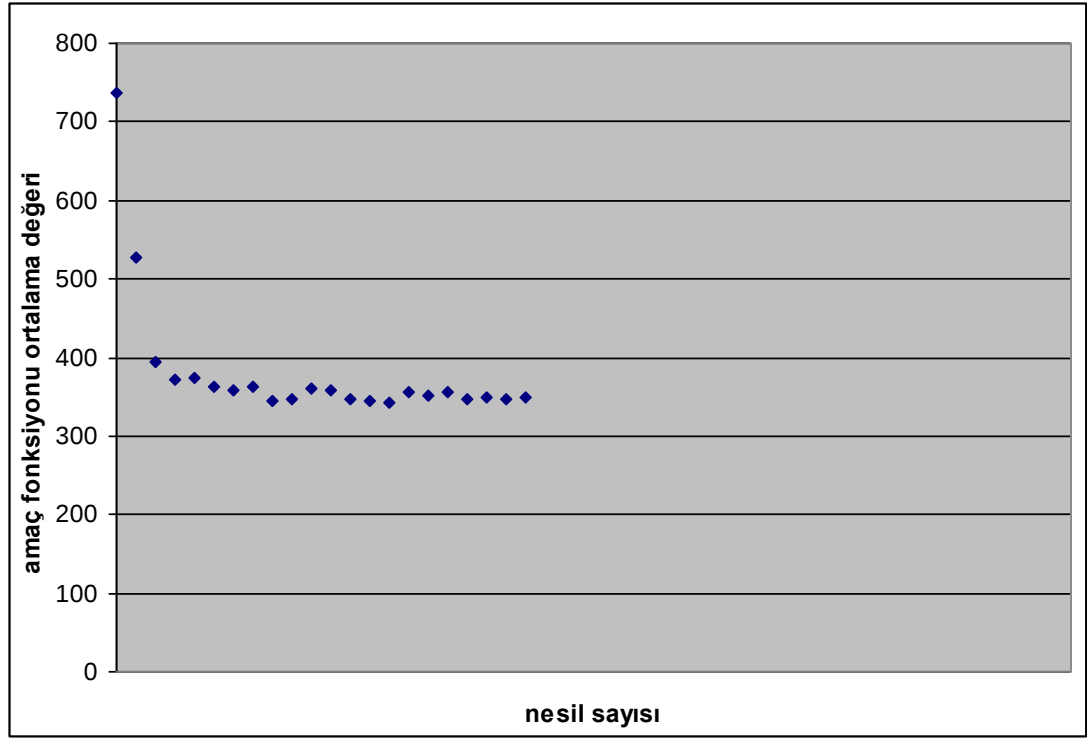
max : 3

p_c : 0.2

p_m : 0.01

a : 10000

nesil_sayisi : 50



Şekil B.8 Delinme gerilimi ortalama değerinin nesil sayısı ile değişimi

Farklı geometrideki pürüzlerin elektriksel alana etkilerinin hesabı için kullanılan parametreler

İletken yarı küre pürüz

Hesaplamanın Yapıldığı Tarih : 05-16-2004

Hesaplamanın Yapıldığı Saat : 11:43:30

n_pop : 200

min (r_1) : 0

max (r_2) : 10

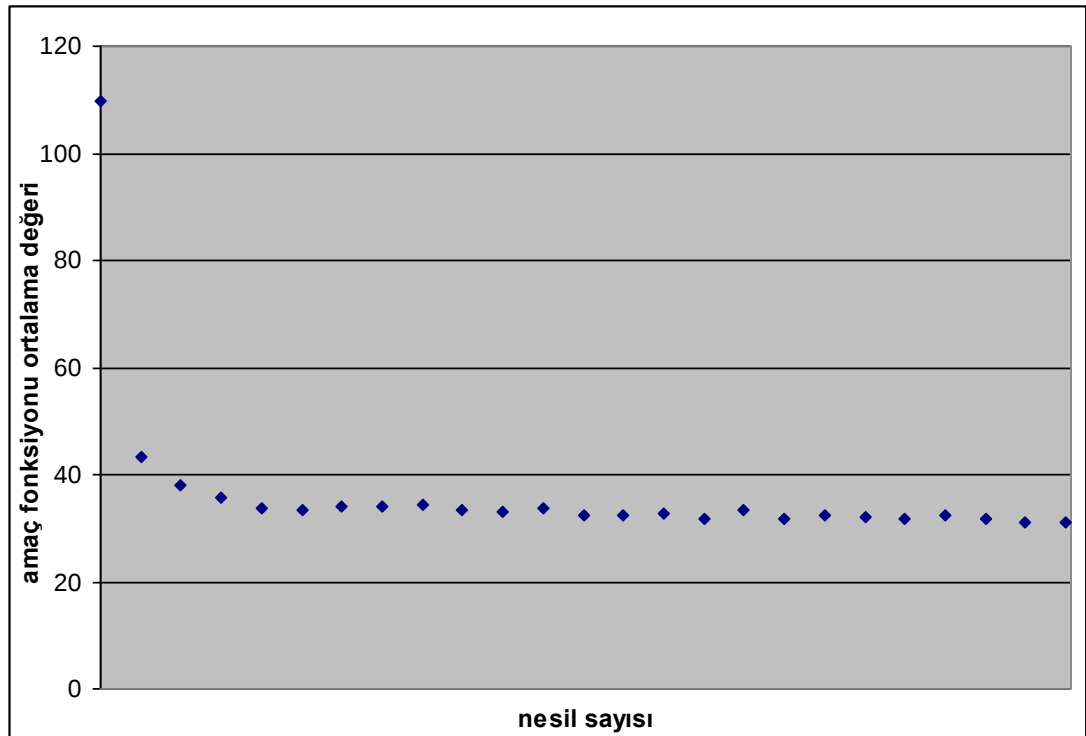
p_c : 0.2

p_m : 0.01

a : 10000

nesil_sayisi : 50

GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.8’de gösterilmiştir. Belirtilen değerler herbir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.



Şekil B.9 İletken yarı küre pürüzde fonksiyon(E) ortalama değeri - nesil değişimi

Yalıtkan yarı küre pürüz

Hesaplamanın Yapıldığı Tarih : 05-17-2004

Hesaplamanın Yapıldığı Saat : 20:15:02

n_pop : 100

min (r_1) : 1

max (r_2) : 15

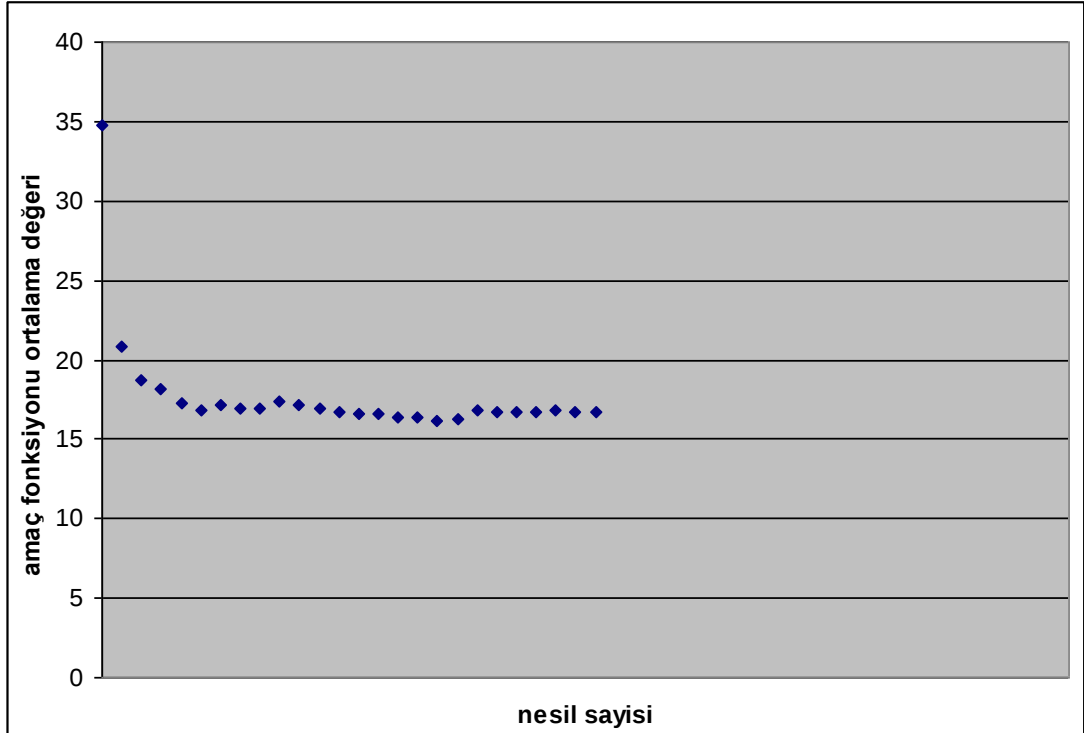
p_c : 0.2

p_m : 0.01

a : 10000

nesil_sayisi : 100

GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.9'da gösterilmiştir. Belirtilen değerler her bir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.



Şekil B.10 Yalıtkan yarı küre pürüzde fonksiyon(E) ortalama değeri-nesil sayısı değişimi

Dik elipsoidal pürüz

Hesaplamanın Yapıldığı Tarih : 05-19-2004

Hesaplamanın Yapıldığı Saat : 12:48:59

n_pop : 200

min (r_1) : 3

max (r_2) : 5

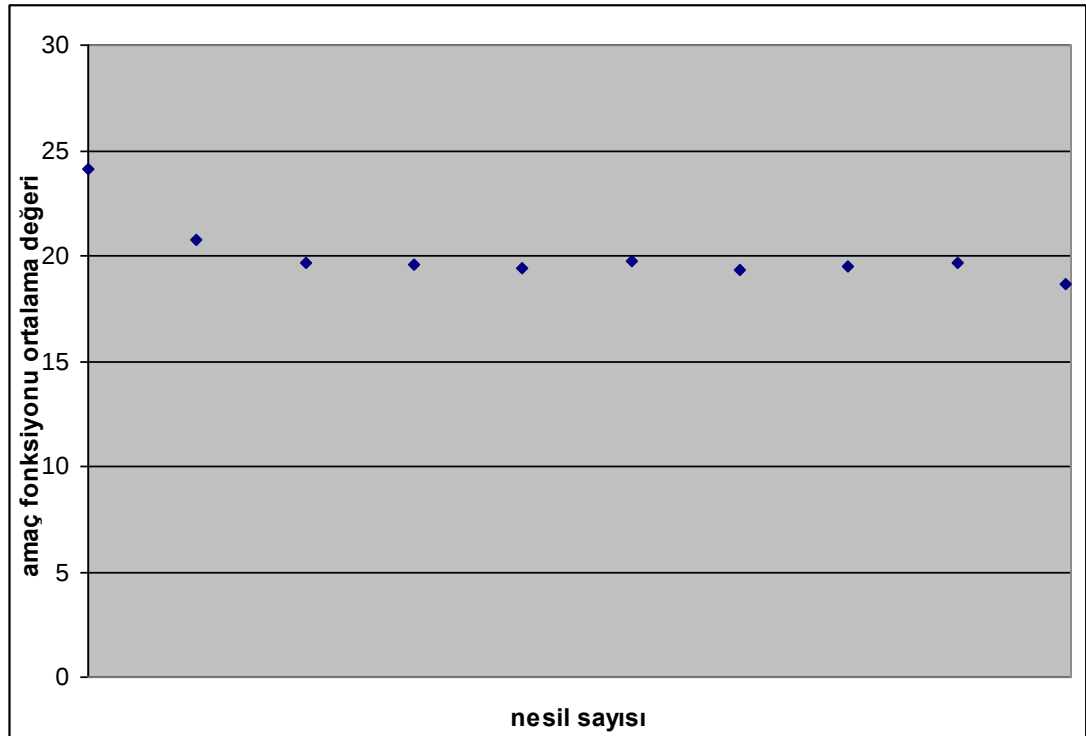
p_c : 0.25

p_m : 0.01

a : 10000

nesil_sayisi : 50

GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.10'da gösterilmiştir. Belirtilen değerler her bir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.



Şekil B.11 Dik elipsoidal pürüzde fonksiyon(E) ortalama değeri-nesil değişimi

Akım darbe fonksiyonunun maksimumunun bulunmasında kullanılan parametreler

Hesaplamanın Yapıldığı Tarih : 05-19-2004

Hesaplamanın Yapıldığı Saat : 22:59:27

n_pop : 200

min : 1

max : 400

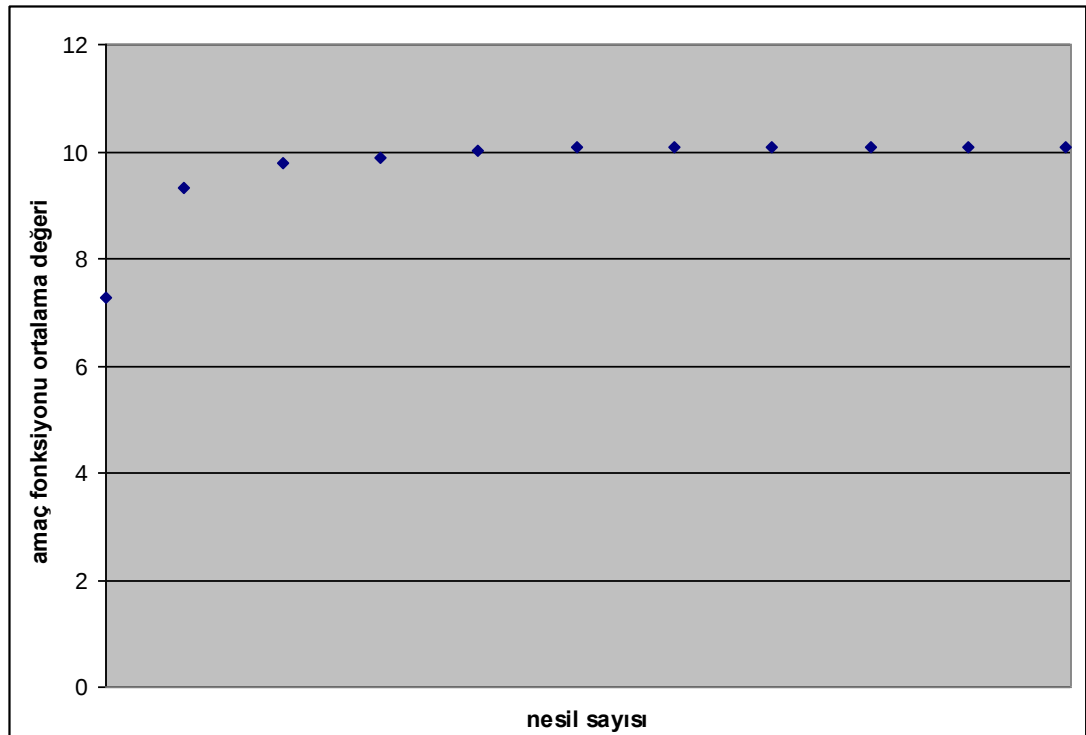
p_c : 0.2

p_m : 0.01

a : 10000

nesil_sayisi : 50

GA yaklaşım performansının nesillere göre değerlendirilmesi Şekil B.10'da gösterilmiştir. Belirtilen değerler her bir topluluğu oluşturan bireylerin amaç fonksiyonundaki değerlerinin ortalamasıdır.



Şekil B.12 Darbe akım fonksiyonu ortalaması-nesil sayısı değişimi

EK C GENETİK ALGORİTMA VISUAL BASIC PROGRAMI

'Tanimlamalar

```
Dim n_bit As Long
Dim c_k As Long
Dim mask1 As Long
Dim mask2 As Long
Dim abc_mask() As Long
Dim def_mask() As Long
Dim ghi_mask() As Long
Dim mutasyon_mask() As Long
Dim toplam_pop As Integer
Dim caprazlanan_pop As Integer
Dim mutasyon_bit() As Integer
Dim mutasyon_krom() As Integer
Dim p_secilme() As Double
Dim p_birikim() As Double
Dim krom_10() As Long
Dim krom_temp() As Long
Dim krom_temp2() As Long
Dim krom_aralik() As Double
Dim uygunluk() As Double
Dim maxuygunluk As Double
Dim toplam_uygunluk As Double
Dim sifirlama As Double
Dim rastgele_sayilar() As Double
Dim nesil_sayisi As Integer
Dim mut_tekrar As Integer
```

Const PI = 3.14159265358979

'SHL fonksiyonun tanımlanması; çaprazlamada kullanılacak olan mask1 ve mask2 için gerekli

'inval:mask1 ve inshift:ck olacak

'Sola kaydırma

```
Function SHL(ByVal inVal As Long, ByVal inShift As Byte) As Long
    SHL = inVal * (2 ^ inShift) ' Bit shift left
End Function
```

'Saga kaydırma bu programda kullanılmıyor cunku mask2 mask1 in degili alinarak bulunuyor

```
Function SHR(ByVal inVal As Long, ByVal inShift As Byte) As Long
    SHR = inVal \ (2 ^ inShift) ' Bit shift right
```

End Function

'Onluk tabandan ikilik tabana donusturme

Function LongToBinStr(x As Long, n As Long) As String

Dim temp_str As String

temp_str = ""

For I = 0 To n - 1

temp_str = temp_str + ""

Next I

For I = 0 To n - 1

temp_str = (x And (2 ^ I)) \ (2 ^ I) & temp_str

Next I

LongToBinStr = temp_str

End Function

'a nin disaridan girilmesi(x:a) ve log2 isimli fonksiyonun tanimi, bu fonk. yard. ile
nbit hesaplanacak

Function Log2(x As Double) As Double

If x <> 0 Then

Log2 = Log(x) / Log(2)

Else:

MsgBox "Hatalı deęer!... a'yı tekrar girin.", vbOKOnly + vbCritical, "Hata"

Log2 = -1 'hata icin -1

End If

End Function

'Amac fonksiyonun girilmesi

Function TargetFunction(x) As Double

'TargetFunction = 100 * 15 / (x ^ 2 * (15 - 1)) + sifirlama 'küresel

elektrotlarda max/min

'TargetFunction = x * Sin(10 * PI * x) + 1 + sifirlama 'örnek 2

'TargetFunction = 100 / (x * Log(15 / 1)) + sifirlama 'silindirsel

elektrotlarda max./min

'TargetFunction = 10 * (1 + ((2 * 10 * 10 * 10) / x ^ 3)) + sifirlama 'iletken yarı

küre pürüz

'TargetFunction = 10 * (1 + (1928.571 / x ^ 3)) + sifirlama 'yalitkan yarı

küre puruz

'TargetFunction = 5 + (53.678 * (3 * x / ((x ^ 2) - 9) - (0.5 * Log((x + 3) / (x - 3))))) +

sifirlama 'elipsoid pürüz

'TargetFunction = (125 / (5768.087 * 0.002)) * (Exp(-250 * x * 10 ^ -6)) *

Sin(5768.087 * x * 10 ^ (-6)) + sifirlama 'darbe akımı max

'TargetFunction = (((x ^ 2) - 1) / ((Log(x)) ^ 2)) + sifirlama 'p geo.min

TargetFunction = (x / 15) * (15 - x) + sifirlama 'alfa max küre

'TargetFunction = x * Log(15 / x) + sifirlama 'alfa max silindir

'TargetFunction = 365 * x / (Log(12 * x / Log(51))) + sifirlama 'pa kritik

'TargetFunction = 1825 * x / (Log(60 * x / Log(51))) + sifirlama 'pa kritik

a sbt

```

'TargetFunction = 2774 * x / (Log(91.2 * x / Log(51))) + sifirlama      ' pa
kritik p sbt
'TargetFunction = 10 * (1 + (5.714285714 / x ^ 3)) + sifirlama  'yalıtkan yarı küre
max
'TargetFunction = 10 + 220.24636 * 12.73148148 * (1.2 * x / ((x ^ 2) - 1.44)) - (0.5 *
Log((x + 1.2) / (x - 1.2))) + sifirlama
End Function

Private Sub Command1_Click()
    End
End Sub

Private Sub Hesapla_Click()
    ReDim out(1000) As String
    Dim out1 As String
    Dim out2 As String

    BigText.Text = " "

    Form1.MousePointer = 11      'kum saatinin goruntulenmesi

    nesil_sayisi = 1

    Dim int_arg As Double

    sifirlama = 0
    toplam_pop = Val(n_pop.Text)
    int_arg = toplam_pop * Val(p_c.Text)

    'Caprazlanan_pop mutlaka cift sayi olmasi icin gerekli tanim
    caprazlanan_pop = Int(int_arg) - (Int(int_arg) Mod 2)

    'Yazdirma islemleri
    out1 = out1 + "n_pop = " + n_pop.Text + vbCrLf
    out1 = out1 + "min = " + min.Text + vbCrLf
    out1 = out1 + "max = " + max.Text + vbCrLf
    out1 = out1 + "p_c = " + p_c.Text + vbCrLf
    out1 = out1 + "caprazlanan_pop = " + Str(caprazlanan_pop) + vbCrLf
    out1 = out1 + "p_m = " + p_m.Text + vbCrLf

    n_bit = Int(-Log2(Val(a.Text))) * (-1)

    out1 = out1 + "n_bit = " + Str(n_bit) + vbCrLf

    If n_bit = -1 Then Exit Sub

    'Eger max=<min ise hata mesaji ver
    If Val(max.Text) <= Val(min.Text) Then
        MsgBox "max değeri min'den büyük olmalı.", vbOKOnly + vbCritical, "Hata"
    
```

```
Exit Sub
End If
```

```
'Eger girilen caprazlama olasiligi 1 den buyukse hata ver
If Val(p_c.Text) > 1 Then
MsgBox "p_c 0 ile 1 arası olmalı.", vbOKOnly + vbCritical, "Hata"
Exit Sub
End If
```

```
'Eger girilen mutasyon olasiligi 1 den buyukse hata ver
If Val(p_m.Text) > 1 Then
MsgBox "p_m 0 ile 1 arası olmalı.", vbOKOnly + vbCritical, "Hata"
Exit Sub
End If
```

```
'Caprazlama konumunun tanimi
c_k = n_bit / 2
out1 = out1 + "c_k = " + Str(c_k) + vbCrLf
'out1 = out1 + "nesil_sayisi = " + nesil.Text + vbCrLf + vbCrLf
'out1 = out1 + Str(nesil_sayisi) + ". nesil" + vbCrLf
```

```
'Caprazlama maskelerinin SHL fonksiyonu yardimiyla tanimi
mask1 = 0
mask1 = SHL(Not mask1, c_k)
mask2 = Not mask1
```

```
'out2 = out2 + "mask1 = " + LongToBinStr(mask1, n_bit) + vbCrLf
'out2 = out2 + "mask2 = " + LongToBinStr(mask2, n_bit) + vbCrLf
BigText.Text = out1 + out2
```

```
'npop adet rastgele sayi uretilmesi
ReDim krom_10(Val(n_pop.Text))
For I = 0 To Val(n_pop.Text) - 1
    krom_10(I) = Int(Rnd * (2 ^ n_bit + 1))
    'out(nesil_sayisi) = out(nesil_sayisi) + "krom_10[" + Str(I) + "] = " +
LongToBinStr(krom_10(I), n_bit) + vbCrLf
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf
```

```
10 'Nesil_sayisi = 1 ise 20 numaraya atlayarak devam et
If nesil_sayisi = 1 Then GoTo 20
For I = 0 To toplam_pop - 1
    'out(nesil_sayisi) = out(nesil_sayisi) + "krom_10[" + Str(I) + "] = " +
LongToBinStr(krom_10(I), n_bit) + vbCrLf
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf
```

```
20 ' Nesil_sayısı > 1 ise buraya dönerek devam et!
'Donguyu buradan itibaren tekrarlatmalıyız !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```

'Bireyin gercek (verilen araliktaki)degerinin bulunmasi
ReDim krom_aralik(Val(n_pop.Text))
For I = 0 To Val(n_pop.Text) - 1
    krom_aralik(I) = min + krom_10(I) * (max - min) / (2 ^ n_bit - 1)
    'out(nesil_sayisi) = out(nesil_sayisi) + "krom_aralik[" + Str(I) + "] = " +
    Str(krom_aralik(I)) + vbCrLf
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf

'Uygunluk ve toplam uygunluk degerlerinin bulunmasi
ReDim uygunluk(Val(n_pop.Text))

toplam_uygunluk = 0
temp_min = 0
For I = 0 To Val(n_pop.Text) - 1
    'uygunluk(I) = (1 / TargetFunction(krom_aralik(I)))      'min
    uygunluk(I) = TargetFunction(krom_aralik(I))            'max
    If temp_min > uygunluk(I) Then temp_min = uygunluk(I)    ' maximum
    'If temp_min < uygunluk(I) Then temp_min = uygunluk(I)    ' minimum problemi
    'icin
Next I

sifirlama = -temp_min

maxuygunluk = 0
For I = 0 To Val(n_pop.Text) - 1
    'uygunluk(I) = 1 / TargetFunction(krom_aralik(I))        'min. icin uygunluk
    uygunluk(I) = TargetFunction(krom_aralik(I))            'max. icin uygunluk
    If uygunluk(I) > maxuygunluk Then maxuygunluk = uygunluk(I)
    toplam_uygunluk = toplam_uygunluk + uygunluk(I)
    out(nesil_sayisi) = out(nesil_sayisi) + "uygunluk[" + Str(I) + "] = " +
    Str(uygunluk(I)) + vbCrLf 'max
    'out(nesil_sayisi) = out(nesil_sayisi) + "uygunluk[" + Str(I) + "] = " + Str(1 /
    uygunluk(I)) + vbCrLf 'min
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf + "toplam_uygunluk = " +
    Str(toplam_uygunluk) + vbCrLf + vbCrLf

'Herbir bireyin secilme olasiliklerinin bulunmasi
ReDim p_secilme(Val(n_pop.Text))
For I = 0 To Val(n_pop.Text) - 1
    p_secilme(I) = uygunluk(I) / toplam_uygunluk
    'out(nesil_sayisi) = out(nesil_sayisi) + "p_secilme[" + Str(I) + "] = " +
    Str(p_secilme(I)) + vbCrLf
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf

'Bireylerin birikimli olasiliklerinin bulunmasi
ReDim p_birikim(Val(n_pop.Text))

```

```

For I = 0 To Val(n_pop.Text) - 1
For J = 0 To I
    p_birikim(I) = p_birikim(I) + p_secilme(J)
Next J
    'out(nesil_sayisi) = out(nesil_sayisi) + "p_birikim[" + Str(I) + "] = " +
Str(p_birikim(I)) + vbCrLf
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf

' 0 ile 1 arasinda n_pop adet rastgele sayi uretilmesi
ReDim rastgele_sayilar(Val(n_pop.Text))
For I = 0 To Val(n_pop.Text) - 1
    rastgele_sayilar(I) = Rnd
    'out(nesil_sayisi) = out(nesil_sayisi) + "rastgele_sayilar[" + Str(I) + "] = " +
Str(rastgele_sayilar(I)) + vbCrLf
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf

'Rulet carki yontemi ile bireylerin secilmesi
ReDim krom_temp(Val(n_pop.Text))
For I = 0 To Val(n_pop.Text) - 1
For J = 0 To Val(n_pop.Text) - 1
    If p_birikim(J) > rastgele_sayilar(I) Then
        krom_temp(I) = krom_10(J)
        J = Val(n_pop.Text) - 1
    End If
Next J
    'out(nesil_sayisi) = out(nesil_sayisi) + "krom_temp[" + Str(I) + "] = " +
LongToBinStr(krom_temp(I), n_bit) + vbCrLf
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf

'Caprazlamaya girecek olan pc*npop adet bireyin secilmesi
ReDim krom_temp2(caprazlanan_pop)
For I = 0 To caprazlanan_pop - 1
    random = Int(Rnd * (toplam_pop - I))
    krom_temp2(I) = krom_temp(random)
    krom_temp(random) = krom_temp(toplam_pop - 1 - I)
Next I

For I = 0 To caprazlanan_pop - 1
    'out(nesil_sayisi) = out(nesil_sayisi) + "krom_temp2[" + Str(I) + "] = " +
LongToBinStr(krom_temp2(I), n_bit) + vbCrLf
Next I
'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf

'Caprazlamaya girmeyen bireyleri direkt kopyalama
For I = 0 To toplam_pop - caprazlanan_pop - 1
    krom_10(I) = krom_temp(I)
Next I

```

```

'caprazlama islemi
For I = toplam_pop - caprazlanan_pop To toplam_pop - 2 Step 2
    Index_temp2 = I - (toplam_pop - caprazlanan_pop)
    capraz_temp1 = (krom_temp2(Index_temp2) And mask1) Or
(krom_temp2(Index_temp2 + 1) And mask2)
    capraz_temp2 = (krom_temp2(Index_temp2) And mask2) Or
(krom_temp2(Index_temp2 + 1) And mask1)
    krom_temp2(Index_temp2) = capraz_temp1
    krom_temp2(Index_temp2 + 1) = capraz_temp2
    krom_10(I) = krom_temp2(Index_temp2)
    krom_10(I + 1) = krom_temp2(Index_temp2 + 1)
Next I

If nesil_sayisi >= Val(nesil.Text) - Int(Val(nesil.Text) * 0.1) Then GoTo 70

'Mutasyon hangi kromozomun hangi bitinde meydana geliyor
mut_tekrar = Int(n_bit * Val(p_m.Text) * n_pop)
ReDim mutasyon_krom(mut_tekrar)
ReDim mutasyon_mask(mut_tekrar)
ReDim mutasyon_bit(mut_tekrar)
ReDim abc_mask(mut_tekrar)
ReDim def_mask(mut_tekrar)
ReDim ghi_mask(mut_tekrar)
For k = 1 To mut_tekrar

    mutasyon_bit(k) = (k / (Val(p_m.Text))) Mod n_bit
    mutasyon_krom(k) = Int(k / (n_bit * Val(p_m.Text)))

'mutasyon tekrari

    abc_mask(k) = 0
    abc_mask(k) = SHL(Not abc_mask(k), mutasyon_bit(k))
    def_mask(k) = 0
    def_mask(k) = SHL(Not def_mask(k), (mutasyon_bit(k) + 1))
    ghi_mask(k) = Not def_mask(k)
    mutasyon_mask(k) = (abc_mask(k) And ghi_mask(k))
    'Print LongToBinStr(mutasyon_mask(k), n_bit)
    'out(nesil_sayisi) = out(nesil_sayisi) + "mutasyon_mask[" + Str(k) + "] = " +
LongToBinStr(mutasyon_mask(k), n_bit) + vbCrLf
    'out(nesil_sayisi) = out(nesil_sayisi) + vbCrLf

    'If mutasyon_krom(k) - 1 < 0 Then mutasyon_krom(k) = mutasyon_krom(k) + 1
    'out1 = out1 + "mutasyon_bit = " + Str(mutasyon_bit) + vbCrLf
    'out1 = out1 + "mutasyon_krom = " + Str(mutasyon_krom) + vbCrLf
    'out(nesil_sayisi) = out(nesil_sayisi) + "mutasyon_krom[" + Str(k) + "] = " +
Str(mutasyon_krom(k)) + vbCrLf
    'out(nesil_sayisi) = out(nesil_sayisi) + "mutasyon_bit[" + Str(k) + "] = " +
Str(mutasyon_bit(k)) + vbCrLf

```

```

krom_10(mutasyon_krom(k) - 1) = krom_10(mutasyon_krom(k) - 1) Xor
mutasyon_mask(k)
krom_10(mutasyon_krom(k) - 1) = krom_10(mutasyon_krom(k) - 1)
'Print LongToBinStr(krom_temp(mutasyon_krom(k) - 1), n_bit)
Next k

```

```

For I = 0 To toplam_pop - 1
'Print LongToBinStr(mutasyon_krom(k), n_pop)
'out(nesil_sayisi) = out(nesil_sayisi) + "krom_10[" + Str(I) + "] = " +
LongToBinStr(krom_10(I), n_bit) + vbCrLf
Next I

```

```

70
nesilno = Str(nesil_sayisi) + ". nesil" + vbCrLf
BigText.Text = BigText.Text + nesilno + out(nesil_sayisi) + vbCrLf

```

```

nesil_sayisi = nesil_sayisi + 1
'Print nesil_sayisi - 1
If nesil_sayisi <= Val(nesil.Text) Then GoTo 10

```

```

Text.Text = maxuygunluk      'max
'Text.Text = 1 / maxuygunluk  'min

```

```

Form1.MousePointer = 0      'kum saatinin kaldırılması
End Sub

```

```

Private Sub Image1_Click(Index As Integer)

```

```

End Sub

```

```

Private Sub mnuKapatItem_Click()
End

```

```

End Sub

```

```

'çözümleme sonuçlarının ayrı bir dosyaya kaydedilmesi

```

```

Private Sub mnuKaydetItem_Click()
CommonDialog1.Filter = "Text files (*.TXT)/*.TXT"
CommonDialog1.ShowSave
If CommonDialog1.FileName <> "" Then
Open CommonDialog1.FileName For Output As #1
Print #1, BigText.Text
Close #1
End If

```

```

End Sub

```

```

'hesaplamanın yapıldığı saatin eklenmesi

```

```

Private Sub mnuSaatItem_Click()
Wrap$ = Chr$(13) & Chr$(10)
BigText.Text = Time$ & Wrap$ & BigText.Text
End Sub

```

```

'hesaplamanın yapıldığı tarihin eklenmesi

```

```

Private Sub mnuTarihItem_Click()
    Wrap$ = Chr$(13) & Chr$(10)
    BigText.Text = Date$ & Wrap$ & BigText.Text
End Sub

Private Sub n_pop_KeyPress(KeyAscii As Integer)
    If (KeyAscii >= vbKey0 And KeyAscii <= vbKey9) Or KeyAscii = vbKeyBack Then
        Exit Sub
    Else
        KeyAscii = 0
        Beep
    End If

End Sub

Private Sub min_KeyPress(KeyAscii As Integer)
    If (KeyAscii >= vbKey0 And KeyAscii <= vbKey9) Or KeyAscii = vbKeyBack Or
    KeyAscii = 46 Then
        Exit Sub
    Else
        KeyAscii = 0
        Beep
    End If

End Sub

Private Sub max_KeyPress(KeyAscii As Integer)
    If (KeyAscii >= vbKey0 And KeyAscii <= vbKey9) Or KeyAscii = vbKeyBack Or
    KeyAscii = 46 Then
        Exit Sub
    Else
        KeyAscii = 0
        Beep
    End If

End Sub

Private Sub p_c_KeyPress(KeyAscii As Integer)
    If (KeyAscii >= vbKey0 And KeyAscii <= vbKey9) Or KeyAscii = vbKeyBack Or
    KeyAscii = 46 Then
        Exit Sub
    Else
        KeyAscii = 0
        Beep
    End If

End Sub

Private Sub p_m_KeyPress(KeyAscii As Integer)

```

```
If (KeyAscii >= vbKey0 And KeyAscii <= vbKey9) Or KeyAscii = vbKeyBack Or  
KeyAscii = 46 Then
```

```
    Exit Sub
```

```
Else
```

```
    KeyAscii = 0
```

```
    Beep
```

```
End If
```

```
End Sub
```

```
Private Sub a_KeyPress(KeyAscii As Integer)
```

```
If (KeyAscii >= vbKey0 And KeyAscii <= vbKey9) Or KeyAscii = vbKeyBack Then
```

```
    Exit Sub
```

```
Else
```

```
    KeyAscii = 0
```

```
    Beep
```

```
End If
```

```
End Sub
```

Program Kullanıcı Arayüzü

Genetik Algoritma

Dosya Zaman

Büyüklikler **Çözümleme Sonuçları**

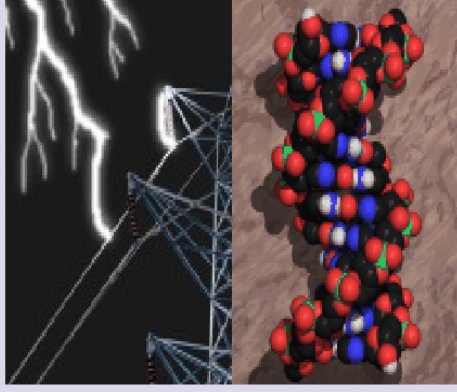
n_pop: 50
min: 1
max: 15
p_c: 0.2
p_m: 0.01
a: 10000
nesil: 100

Hesapla

Çıkış

Sonuç: 3.74999344113625

n_pop = 50
min = 1
max = 15
p_c = 0.2
caprazlanma_pop = 10
p_m = 0.01
n_bit = 14
c_k = 7
1. nesil
uygunluk(0) = 2.9890344866602
uygunluk(1) = 3.68743908423858
uygunluk(2) = 3.57635543062685
uygunluk(3) = 3.35112539677067
uygunluk(4) = 3.40569530434175
uygunluk(5) = 2.48989978238755
uygunluk(6) = 1.10037891212533
uygunluk(7) = 2.6012573997088
uygunluk(8) = 2.14678805894278
uygunluk(9) = 2.96633431016011
uygunluk(10) = 1.45672748209066
uygunluk(11) = 3.71699405738815
uygunluk(12) = 1.67563121893065
uygunluk(13) = 2.35885619213499
1. nesil sonu



ÖZGEÇMİŞ

Alper Güçlü, 1979 yılında Çanakkale’da doğdu. İlk ve orta eğitimini sırasıyla Sakarya İlköğretim Okulu, Biga Ortaokulu ve Mehmet Akif Ersoy Lisesi’nde tamamladı. 1996 – 2001 yılları arasında, bir yıl hazırlık sınıfı dahil, İstanbul Teknik Üniversitesi, Elektrik – Elektronik Fakültesi, Elektrik Mühendisliği Bölümü’nde tamamlayarak Elektrik Mühendisi ünvanını aldı. 2001 – 2002 döneminden beri, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Mühendisliği Ana Bilim Dalı’nda lisans üstü eğitimini sürdürmektedir.