



BANDIRMA 17 EYLÜL ÜNİVERSİTESİ

GENETİK ALGORİTMA

GÜZ DÖNEMİ

BANDIRMA,2017

201410306014 ÖZNUR CENGİZ

150306024 HİLAL KOCA

İÇİNDEKİLER

1.GENETİK ALGORİTMA.....	4
1.1.Genetik Algoritma Uygulama Alanları Nereledir?	6
1.2. Genel Uygulama Alanları.....	6
1.3. İşletmelerdeki Uygulama Alanları	7
2.Genetik Algoritma Kullanılarak Noktadan Noktaya Yol ve Rota Planlama	8
2.1.Genetik Algoritma Yönteminin Probleme Uygulanışı	9
2.2.Performans Analizi	11
2.3.Sonuçlar ve Öneriler	13
3. GENETİK ALGORİTMALAR İLE OPTİMİZASYON	14
3.1 Optimizasyon	14
3.2 Genetik Algoritmalar ve Evrimsel Hesaplama.....	18
3.3. Rasgele Araştırma Algoritması	20
3.4. İkili kodlu genetik algoritmalar	21
3.4.1. İkili kodlu genetik algoritmaların operatörleri	21
3.4.2. Amaç fonksiyonu ve parametreler.....	23
3.5. Başlangıç popülasyonu.....	26
3.6.Doğal seçim	27
3.7. Çaprazlama.....	30
3.8. Mutasyon	31
3.8.1.Gelecek nesil	32
3.8.2. Yakınsama	36
3.9. Gerçek kodlu genetik algoritma.....	37
Şekil 20: GA'nın akış diyagramı.....	38
3.10.Amaç fonksiyonu ve parametreler.....	38
3.11. Başlangıç Popülasyonu.....	39

3.12.Doğal seçim	40
3.13.GENETİK ALGORİTMA SEÇİM YÖNTEMLERİ	41
3.13.1.Seçim.....	41
3.13.2. Çaprazlama.....	42
3.13.3. Mutasyon	42
4.Gezgin Satıcı Probleminin Genetik Algoritmalarla Uygulaması	42
5.Java’da Basit Bir Genetik Algoritma Kodlaması	46
6. GENETİK ALGORİTMA TABANLI PID KONTROLÖR	47
SİMÜLATÖRÜ TASARIMI	47
6.1.PID KONTROLÖRLER VE TASARIM YÖNTEMLERİ.....	48
6.1.1 PID Kontrolörler	48
6.2. PID Kontrolör Tasarım Yöntemleri	50
6.2.1. Ziegler-Nichols yöntemi	50
6.2.2. Cohen-Coon yöntemi	51
6.2.3. Wang-Juang-Chan yöntemi.....	51
6.3 GENETİK ALGORİTMALAR	51
7.PID SİMÜLATÖR	52
7.1.PID SONUÇLARI	53
8.ÖRNEKLER	54
ÖRNEK II	54
ÖRNEK III	54
ÖRNEK IV	55
9. SONUÇLAR.....	59
KAYNAKÇA.....	60

1.GENETİK ALGORİTMA

GİRİŞ

Genetik algoritmalar, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar.

Genetik algoritmaların temel ilkeleri ilk kez Michigan Üniversitesi'nde John Holland tarafından ortaya atılmıştır. Holland 1975 yılında yaptığı çalışmaları “Adaptation in Natural and Artificial Systems” adlı kitabında bir araya getirmiştir. İlk olarak Holland evrim yasalarını genetik algoritmalar içinde eniyileme problemleri için kullanmıştır.

Genetik algoritmalar problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir. Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdır. Her biri çok boyutlu uzay üzerinde bir vektördür.

Genetik algoritmalar problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler. Diğer eniyileme yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, böyle yapılardan meydana gelen bir küme oluştururlar. Problem için olası pek çok çözümü temsil eden bu küme genetik algoritma terminolojisinde nüfus adını alır. Nüfuslar vektör, kromozom veya birey adı verilen sayı dizilerinden oluşur. Birey içindeki her bir elemana gen adı verilir. Nüfustaki bireyler evrimsel süreç içinde genetik algoritma işlemcileri tarafından belirlenirler.

Problemin bireyler içindeki gösterimi problemde değişiklik gösterir. Genetik algoritmaların problemin çözümündeki başarısına karar vermedeki en önemli faktör, problemin çözümünü temsil eden bireylerin gösterimidir. Nüfus içindeki her bireyin problem için çözüm olup olmayacağına karar veren bir uygunluk fonksiyonu vardır. Uygunluk fonksiyonundan dönen değere göre yüksek değere sahip olan bireylere, nüfustaki diğer bireyler ile çoğalmaları için fırsat verilir. Bu bireyler çaprazlama işlemi sonunda çocuk adı verilen yeni bireyler üretirler. Çocuk kendisini meydana getiren ebeveynlerin (anne, baba) özelliklerini taşır. Yeni bireyler üretilirken düşük uygunluk değerine sahip bireyler daha az seçileceğinden bu bireyler bir süre sonra nüfus dışında bırakılırlar. Yeni nüfus, bir önceki nüfusta yer alan uygunluğu yüksek bireylerin bir araya gelip çoğalmalarıyla oluşur. Aynı

zamanda bu nüfus önceki nüfusun uygunluğu yüksek bireylerinin sahip olduğu özelliklerin büyük bir kısmını içerir. Böylelikle, pek çok nesil aracılığıyla iyi özellikler nüfus içerisinde yayılırlar ve genetik işlemler aracılığıyla da diğer iyi özelliklerle birleşirler. Uygunluk değeri yüksek olan ne kadar çok birey bir araya gelip, yeni bireyler oluşturursa arama uzayı içerisinde o kadar iyi bir çalışma alanı elde edilir. Probleme ait en iyi çözümün bulunabilmesi için;

Bireylerin gösterimi doğru bir şekilde yapılmalı,

Uygunluk fonksiyonu etkin bir şekilde oluşturulmalı,

Doğru genetik işlemciler seçilmeli.

Bu durumda çözüm kümesi problem için bir noktada birleşecektir. Genetik algoritmalar, diğer eniyileme yöntemleri kullanılırken büyük zorluklarla karşılaşılır, oldukça büyük arama uzayına sahip problemlerin çözümünde başarı göstermektedir. Bir problemin bütünsel en iyi çözümünü bulmak için garanti vermezler. Ancak problemlere makul bir süre içinde, kabul edilebilir, iyi çözümler bulurlar. Genetik algoritmaların asıl amacı, hiçbir çözüm tekniği bulunmayan problemlere çözüm aramaktır. Kendilerine has çözüm teknikleri olan özel problemlerin çözümü için mutlak sonucun hızı ve kesinliği açısından genetik algoritmalar kullanılmazlar. Genetik algoritmalar ancak;

Arama uzayının büyük ve karmaşık olduğu,

Mevcut bilgiyle sınırlı arama uzayında çözümün zor olduğu,

Problemin belirli bir matematiksel modelle ifade edilemediği,

Geleneksel eniyileme yöntemlerinden istenen sonucun alınmadığı alanlarda etkili ve kullanışlıdır.

Genetik algoritmalar parametre ve sistem tanılama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanıma, mühendislik tasarımları, planlama, yapay zeka uygulamaları, uzman sistemler, fonksiyon ve kombinasyonel eniyileme problemleri ağ tasarım problemleri, yol bulma problemleri, sosyal ve ekonomik planlama problemleri için diğer eniyileme yöntemlerinin yanında başarılı sonuçlar vermektedir.

Diğer yöntemlerden farkı;

1-Genetik algoritmalar problemlerin çözümünü parametrelerin değerleriyle değil, kodlarıyla arar. Parametreler kodlanabildiği sürece çözüm üretilebilir. Bu sebeple genetik algoritmalar ne yaptığı konusunda bilgi içermez, nasıl yaptığını bilir.

2-Genetik algoritmalar aramaya tek bir noktadan değil, noktalar kümesinden başlar. Bu nedenle çoğunlukla yerel en iyi çözümde sıkışıp kalmazlar.

3-Genetik algoritmalar türev yerine uygunluk fonksiyonunun değerini kullanır. Bu değer kullanılması ayrıca yardımcı bir bilginin kullanılmasını gerektirmez.

4-Genetik algoritmalar gerekirci kuralları değil olasılıksal kuralları kullanır.

1.1.Genetik Algoritma Uygulama Alanları Nereledir?

Genetik algoritmalar, deneysel çalışmalarda optimizasyon aşamasında, endüstriyel uygulamalarda ve sınıflandırmalarda uygulama alanı bulunmaktadır. Mühendislik alanında en çok optimizasyon amaçlı olarak kullanılmakta ve diğer klasik yöntemlere göre daha iyi sonuç vermektedir. Genetik algoritmalar, bazı doğal olayları modelleyen skolastik algoritmalar. Bu algoritmalar biyolojik evrimin işleyiş biçimini taklit eder. Konuyla ilgili çalışmalar incelendiğinde, araştırma alanı geniş ve karmaşık ise, konuyla ilgili bilgi az veya eldeki bilgi araştırma alanını daraltmada yeterli değilse, matematiksel analiz elde edilemiyorsa veya geleneksel araştırma metotları ile başarısız olunmuş veya iyi sonuç alınmamışsa genetik algoritmalarından faydalandığı görülmüştür.

Genetik Algoritması uygulama alanları ikiye ayrılır:

1.1 Genel Uygulama Alanları

1. 2. İşletmelerdeki Uygulama Alanları

1.2. Genel Uygulama Alanları

➤ Otomatik Programlama ve Bilgi Sistemleri

Genetik algoritmaların yaygın olarak kullanıldığı alanlardan biri, belirli ve özel görevler için gerekli olan bilgisayar programlarını geliştirmedir. Ayrıca, diğer hesaplama gerektiren yapıların tasarımı için de kullanılmaktadır. Bunlara örnek

olarak, bilgisayar çipleri tasarımı, ders programı hazırlanması ve ağların çizelgelenmesi verilebilir.

Genetik algoritmalar kullanılarak dağıtılmış bilgisayar ağlarının tasarımı da gerçekleştirilmektedir. Bu problem tipinde ağ güvenilirlik parametrelerini (çap, ortalama uzaklık ve bilgisayar ağ güvenilirliği gibi) optimize etmek için birden fazla amaç fonksiyonu kullanılmaktadır. Genetik algoritmalar ile 100 düğüme kadar olan ağlar başarıyla tasarlanmıştır. Ağ tasarımında genetik algoritmaların kullanılması, tasarım sürelerinin ve maliyetlerinin azalmasında önemli bir katkı sağlamıştır. Özellikle, maksimum miktardaki verinin minimum iletişim hattıyla taşınmasında yüksek bir performans göstermiştir. Ayrıca genetik algoritmaların kullanımıyla, çeşitli alanlara dağıtılmış bir sistem için en uygun dosya tahsisatı gerçekleştirilmektedir.

➤ Mekanik Öğrenme

Mekanik öğrenme; ilki, gözlenmiş bir veri takımını anlamak ve yorumlamak, ikincisi de görülmemiş objelerin özelliklerini tahmin etmek olan iki temel amaç için model kurmayı amaçlar. Parametrik istatistikten ziyade çok büyük veri takımlarının yönetimi üzerinde çalışır. Kullandığı metotların çoğu dağılımdan bağımsız metotlar olarak sınıflanabilir. Uygun model seçimi için işe problem hakkındaki varsayımlarla başlar. Onun yerine uygun model yapısını belirlemek için doğrudan mevcut veriden hareketle bir araç kutusu yaklaşımı kullanır. Sınıflama sistemi, genetik algoritmaların mekanik öğrenme alanında bir uygulamasıdır. Basit dizi kurallarını öğrenen bir mekanik öğrenme sistemi olan sınıflama sisteminin kural ve mesaj sistemi, özel bir üretim sistemi olarak adlandırılabilir. Bu üretim sistemi, “eğer-sonra” kural yapısını kullanır. Bir üretim kuralı, “eğer” yapısından sonra belirtilen durum için, “sonra” yapısından sonra gelen faaliyetin gerçekleştirilmesini içerir. Genetik algoritmalar, sınıflama sistemlerinde kural-bulma mekanizması olarak kullanılmaktadırlar. Genetik algoritmalar ayrıca, sinir ağlarında ve proteinin yapısal analizinde de kullanılmaktadır.

1.3. İşletmelerdeki Uygulama Alanları

➤ Pazarlama

Tüketicilere ait verileri analiz etmek, çeşitli tüketici kalıpları çıkarmak ve bu kalıplara dayanarak pazarlama stratejileri uygulamak, pazarlamanın en önemli fonksiyonlarından biridir. Tüketicilerin profilleri çıkarılarak, belirli satın alma kalıpları yakalanabilmektedir. Ancak tüketici profilini çıkarabilmek için, çok büyük veri

tabanlarını işletme amaçları doğrultusunda hızlı ve etkin biçimde kullanmak gerekmektedir. Burada kullanılan teknik veri madenciliğidir. Veri madenciliği, çok geniş veri tabanlarından veriyi süzme tekniğidir. Pazarı ve tüketiciyi tanımada son derece önemli rol oynayan veri madenciliği, veriyi bilgiye bilgiyi de güvenli kararlara dönüştürür. Veri madenciliğinin verimlilik, karlılık, müşteri tatmini ve rekabet edebilme yeteneği gibi yaşamsal konularda işletme üzerinde çok önemli etkileri bulunmaktadır. Rekabet edebilme yeteneği karar alma kalitesine bağlıdır ve bundan dolayı işletmeler sürekli karar kalitelerini geliştirmeye çalışırlar. Veri madenciliğinde kullanılan tekniklerden birisi de genetik algoritmadır. Genetik algoritma tabanlı yaklaşım kullanılarak veri yığınlarından modeller elde edilmektedir

➤ Sistem Güvenilirliği Problemi

Bir sistemin güvenilirliği, belirli koşullar altında belirli bir zaman aralığında sistemin başarılı olarak çalışma olasılığı olarak tanımlanmaktadır. Çoğu sistem, çeşitli işlemlerde kritik bir role sahiptir ve eğer sistemde arıza olursa sonuçları oldukça ciddi olmaktadır. Bu alanda optimizasyon, etkisiz parçaların sisteme en iyi şekilde tahsis edilebilme veya yararlanabilme yolunu bulmayı içermektedir. Parçalara, güvenilirliklerinin etkin olarak ölçülebilmesi için olasılıklar atanmaktadır

➤ Araç Rotalama Problemi

Birleşti optimizasyon problemlerinin örneklerinden biri de araç rotalama problemidir. Temel araç rotalama problemi, talebi belirli olan müşterileri kapsar. Tek bir depodan araçlar ayrılmakta ve müşteri taleplerini karşılayarak tekrar depoya dönmektedir. Her aracın kapasite kısıtı vardır. Bu temel probleme ayrıca, her aracın alacağı yol da mesafe kısıtı olarak eklenebilir. Her bir müşterinin talebini yalnızca bir araç karşılamaktadır. Problem, bu kısıtlar altında minimum toplam maliyeti veren rotaları bulmaktır. Daha karmaşık bir araç rotalama problemi olan zaman pencereli rotalama probleminde ise amaç müşteri talebini belirli zaman aralıkları içerisinde minimum toplam maliyetle karşılamaktır.

2.Genetik Algoritma Kullanılarak Noktadan Noktaya Yol ve Rota Planlama

Giriş

Bu çalışmada bir kroki üzerinde bulunan noktalar arası rota ve yolun genetik algortima ile bulunması amaçlanmıştır. Genetik algoritma rastgele arama metodu olduğu için tek bir çözüm aramak yerine bir çözüm kümesi üzerinde çalışır. Optimum çözüme olası çözümlerin bir bölümü üzerinde gidilir. Böylece çalışmadaki sonuçlar her zaman en iyi olmaz. Çalışmada genetik algoritmanın kullanılmasının nedeni, genetik algoritmanın problemin doğasıyla ilgili herhangi bir bilgiye ihtiyaç duymamasıdır. Temelinde gezgin satıcı problemine benzeyen çalışmanın , gezgin satıcı problemine benzer problemler içinde çözüm olması amaçlanmıştır.



Şekil.1 Arayüz Görünümü

2.1.Genetik Algoritma Yönteminin Probleme Uygulanışı

Problemin çözümü için popülasyon büyüklüğü karar verilmelidir. Popülasyon büyüklüğü seçilirken aşırı yüksek seçilirse gelişim yavaşlar, aşırı küçük seçilmesi durumunda da araştırma uzayı yetersiz olur. Problemimize en uygun popülasyon büyüklüğü 30 birey olarak seçilmiştir. Programın arayüzünde birey sayısı değiştirilebilir. Her kromozom 20 genle temsil edilip, genler kodlanırken değer kodlama yöntemi kullanılmıştır. Her gen yönleri temsil eden 0;Batı, 1;Kuzey, 2;Doğu ve 3;Güney ile belirtilmiştir. Örnek olarak 5 nolu kromozomun genleri:01102033021011023221 gibi 20 gene sahiptir.

Popülasyon büyüklüğü tamamlanıp, kromozomlar kodlandıktan sonra uygunluklarına göre seçim yapıldı. Uygunluk değerleri hesaplanırken noktalar arası seyahat olduğu için kromozomun en son noktası ile ulaşmak istenen arasındaki farka bakılır. Bu yüzden uygunluk fonksiyonumuz 2 fonksiyonun toplamına eşittir. Birinci fonksiyonumuz en son nokta uzaklığı $f(y)$, ikinci fonksiyonumuz ise

kromozomun aldığı yolların toplamı olan toplam mesafe $f(z)$ fonksiyonudur. Uygunluk fonksiyonumuz $f(x)=5*f(y)+ f(z)$ olarak belirlenmiştir. Burada son nokta uzaklığını fonksiyonu problem için daha önemli olduğundan katsayısı artırılmıştır. Örneğin 6 nolu kromozom son noktası 21 , ulaşılacak istenen nokta 22 ise $f(y)=20$, ve aldığı yolların mesafesini ölçen $f(z)=550$ olduğunu düşünelim ,6 nolu kromozomun uygunluk değeri $f(x)=5*20+550=650$ olur. Kromozom ulaşılacak istenilen noktaya varmış olsaydı bu değer 550 olacaktı. Burada görüldüğü gibi son noktaya ulaşmak uygunluk değerini hesabının doğruluğunu kanıtlıyor.

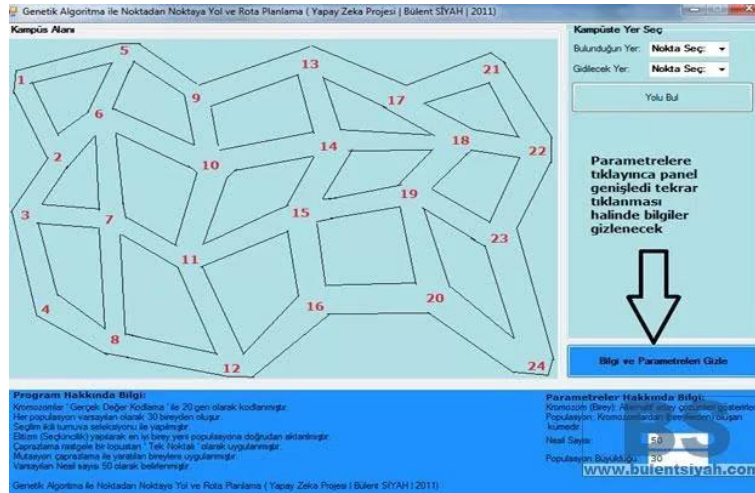
Seçilimde ikili turnuva seleksiyonu kullanıldı. İkili turnuva seçiminde populasyon içinden rastgele iki birey seçilir ve uygunlarına göre iyi olan alınır, daha sonra tekrar rastgele iki birey seçilir ve yine uygunluklarına göre en iyi olan alınır. Böylece elde olan iki tane birey çaprazlanarak yeni topluma katılır. Çaprazlama yapılırken rastgele bir lopus seçilir ve iki kromozom o lopustan değiştirilir. Örnek rastgele 2 ve 16 nolu bireyler seçildi bunlardan uygunluğu en iyi olan 2, tekrar rastgele iki birey seçildi 11 ve 7, bunlara arasındada uygun olan 7 , bu kazanan 2 ve 7 bireyi rastgele bir noktadan değişirme hazırlar. Rastgele noktamızın 9 olduğunu düşünürsek her kromozom 20 gen olduğu için ilk 9 gen 2 nolu kromozomdan geriye kalan 11 gen 7 nolu kromozomdan alınır(2 ile 7 ikili turnuva sonucunda eşleşen kromozomları). Aynı işlem 7nin ilk 9 geni alınır geriye kalan 11 gende 2den alınır. Turnuva metodun da seçilen birey tekrar sisteme dahil edilir yani çaprazlamaya uğrayan birey tekrar çaprazlamaya katılabilir.

Popülasyondaki en iyi birkaç birey doğrudan yeni topluma aktarıldı. Geriye kalan bireyler çaprazlama ile yaratıldı. Çaprazlamaya uğrayan bireylerden biri mutasyona uğratıldı. Mutasyon yapılmasının nedeni önceki çözümlerin kopyalanmasını önlemek. Mutasyon için rastgele bir gen seçilir ve değiştirilir. Böylece yeni populasyon oluşturuldu. Problemin çözümü için belirlenen iterasyon sayısı kadar döngü devam eder. Döngü sonlanınca problemin en uygun çözümü elde edilmiş olur. Böylece optimum değer elde edilmiş oldu fakat çalışmanın diğer amacı olan alternatif yollar üretmek için son nesilden önceki nesillerin tümünden en iyi olanlarda seçilip , seçilen yollar arasından birbirinden farklı olan diğer yollarda projede tutularak arayüzde yansıtıldı.

2.2.Performans Analizi

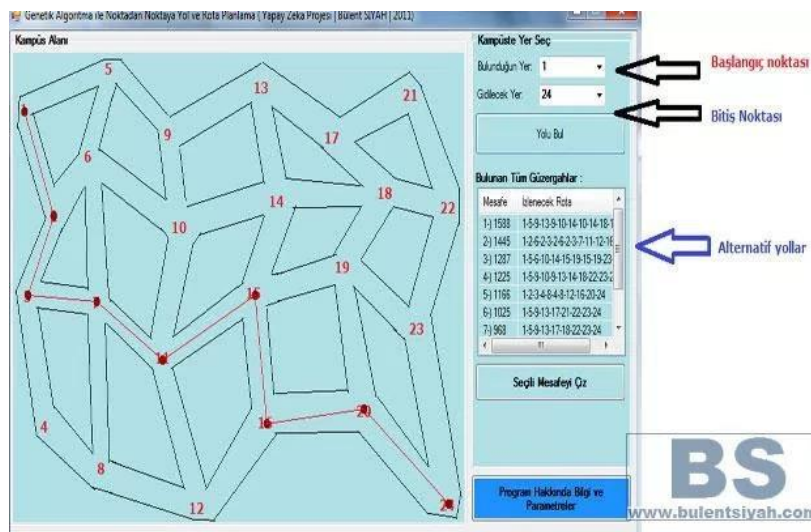
Problemin çözümünde elde edilen bazı veriler;

Başlangıç noktasından bitiş noktasına gitmeye çalışınca elde edilen veriler grafikte gösterilmiştir. Y eksenini Uzaklık piksel , x eksenini nesil sayısını gösterir.

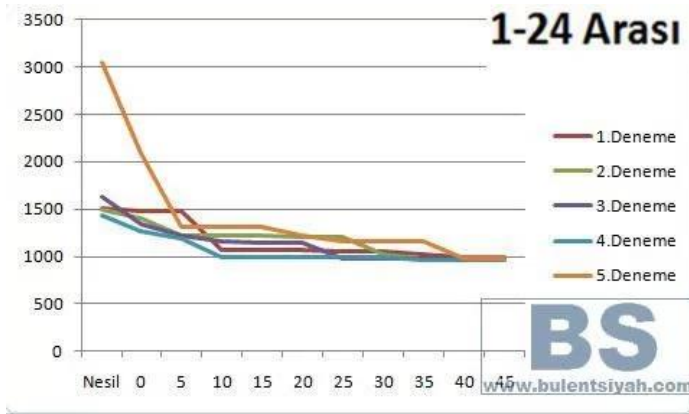


Şekil 2. Arayüz Görünümü (Parametreler Butonuna tıklandıktan sonra Görüntülenir)

1 ile 24 arasındaki elde edilen veriler aşağıda listelenmiştir. 50 nesil sayısı ile sonuçlar gözlenmiştir.

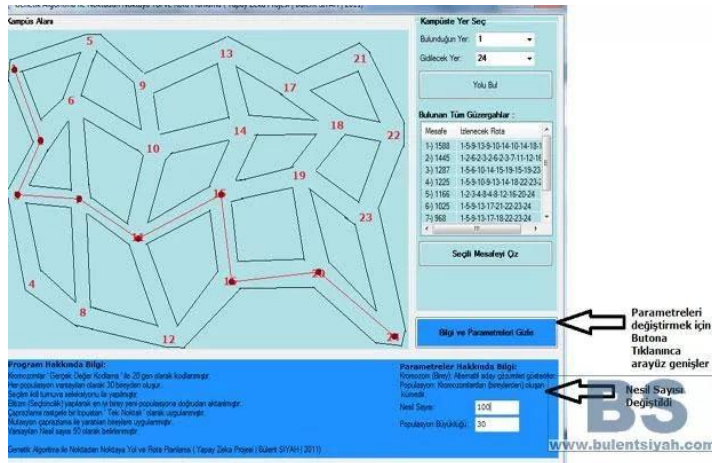


Şekil 3. 1-24 arası nesil sayısı=50 Programın Arayüzü



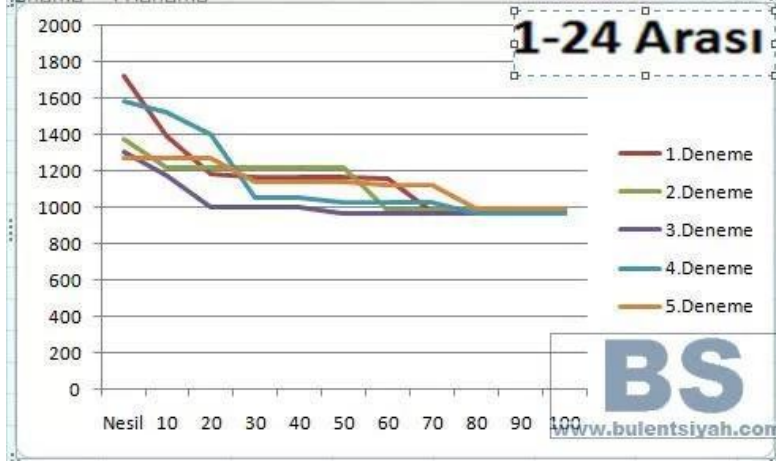
Şekil 4. 1-24 arası elde edilen veriler.

Nesil sayısı değiştirmek için arayüzde şöyle değiştirilir.



Şekil 5. Nesil sayısı değiştirilişi.

1 ile 24 arasındaki elde edilen veriler 100 nesil sayısı ile sonuçlar aşağıda gözlenmiştir.



Şekil 6. 1-24 arası elde edilen veriler(Nesil sayısı 100)

2.3.Sonuçlar ve Öneriler

Elde edilen tüm sonuçlara göre bir probleme genetik algoritma uygulanmasında uygun parametreler seçilmediği taktirde en uygun çözümden uzaklaşır. Çalışmada popülasyon büyüklüğü ayarlanırken varsayılan değer olan 30 ile 100 değeri arasında gözlemlerim; popülasyon büyüdükçe çözüm olması muhtemel bireylerin çoğalması ve çözüm uzayının genişlemesiyle en iyiye daha da yaklaşılmıştır.

Uygunluk fonksiyonu oluşturulurken çalışma için önemli noktalar belirlenmemesi halinde uygun olmayan bireyler uygun sanılıp problemin çözümünden uzaklaşılır. Bu çalışmada uygunluk fonksiyonu 2 tane fonksiyonun toplamına eşit. Çünkü en kısa yol bulunmasının dışında önemli olan parametre kromozomun istenilen noktaya ulaşip ulaşmadığıdır. Yönetimin probleme uygulanışında bir örnek ile bu durumun anlatılmaya çalışıldı.

Seçilim için ikili turnuva seçilimi haricinde rulet tekeli seçilimi çalışmada kullanılmamasının sebebi rulet tekeri yöntemi ile iyi olan birey nesiller sonra kendisiyle aynı bireyler üreterek çözüm uzayını o nokta etrafında toplamasıdır. Bu durum çalışmada daha iyi olması muhtemel çözümlerin araştırılmasını engellediğinden kullanılmamıştır.

Çaprazlamada ise çaprazlama oranının probleme uygun ayarlanmaması halinde çalışmada hatalar gözlemlendi. Örneğin en iyi bireylerin yok olması istenilen bir durum değildi ve tüm bireyler çaprazlama ile oluşturulması durumunda bu hata açığa çıktı. Bu durumu engellemek için elitizm yapılarak en iyi bireyler doğrudan yeni topluma aktarılacak elenmeleri engellenmiş oldu. Çaprazlama sayesinde çözüm uzayı genişledi. En iyi bireyler doğrudan yeni topluma aktarılmasına rağmen ikili turnuva

yönteminde kullanılan kümeden çıkarılmadılar. Çaprazla için seçilen bireyler de ikili turnava yöntemi için bulunan kümesinden çıkarılmadı.

Mutasyonun çözüm yoğunluğunu dağıttı gözlendi. Fakat mutasyon oranı fazla tutulması halinde en uygun değerden uzaklaşıldığı izlendi. Bu yüzden çalışmamızda mutasyon oranı 0.01 olarak tutulmasına karar verildi.

Projenin geliştirilmesi halinde bir bina içerisinde özellikle çok sayıda geçiş bulunduğu karmaşık binalarda istenilen yerler arasında en uygun şekilde yol ve rota bulunabilir. Çalışmanın ileriki aşamalarında program arayüzüne kullanıcı tarafından kroki, harita veya plan yüklenmesi sağlanabilir. Gezgini satıcı problemine benzer problemler için de kullanılabilir. Örneğin bilgisayar ağlarında olası bir kopmada diğer en uygun yollar bulunabilir.

3. GENETİK ALGORİTMALAR İLE OPTİMİZASYON

3.1 Optimizasyon

Optimizasyon bir şeyin daha iyisini gerçekleştirme sürecidir. Bir mühendis veya bilim adamı yeni bir fikir ortaya koyar. Optimizasyon bu fikrin geliştirilmesine yardımcı olur.

Optimizasyon; fikirleri ilerletmek için ortaya atılan bilgileri kullanarak, başlangıç kavramlarını (parametrelerini) değiştirmektir. Eğer fikri etkileyen parametreler elektronik formata dönüştürülürse bilgisayar mükemmel bir optimizasyon aracı olur. Bir takım bilgiler bilgisayara girilir ve bir çözüm elde edilir. Elde edilen bu çözümün değerlendirilmesi optimizasyon aracılığıyla gerçekleştirilir. Bir problemin birden fazla çözümü varsa, en iyi çözümü bulmak gerekir. Aslında “en iyi” tanımlaması izafi kavramdır. Optimal çözüm, problemi formüle eden kişiye bağlıdır. Eğitim, fikirler, ideolojiler ve sosyal durumlar “en iyi” tanımlamasında etkili olan faktörlerdir. Bazı problemlerin tam cevabı bulunurken, bazıları optimal noktalar olarak bilinen, değişik minimum ve maksimum noktalarına sahiptir. İşte burada “en iyi” izafi olabilir. Yer kürede yaşayan insanoğlunun hayatı, optimizasyon problemleri ve çözümleri ile doludur. Bununla ilgili birkaç örnek verilebilir. Hem işe zamanında gitmek, hem de

uyku miktarını artırmak için ne zaman kalkılmalıdır? İşe giderken en iyi yol neresidir? Bir projeyi yürütürken ilk olarak nereden başlanmalıdır? Bir şeyin uzunluğunu kısaltmak, ağırlığını azaltmak bir ürünün cazibesini artırmak ve maliyetini düşürmek optimizasyon tekniğinin ilgi alanı içerisinde.

Optimizasyonun temel aşamaları şekil 7’de gösterilmiştir.



Şekil 7: Optimizasyonun temel aşamaları

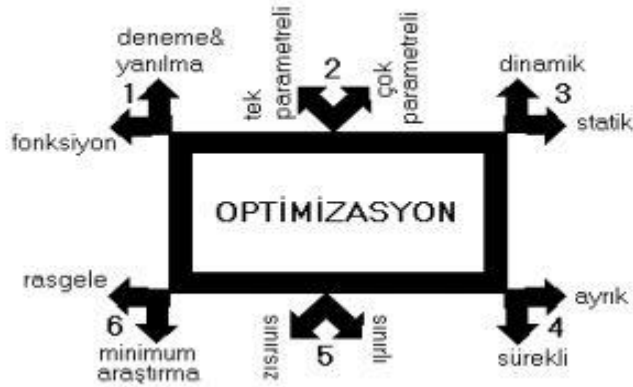
Bir deney düzeneğinde, maksimum veya minimum sonuç/çıkış elde edebilmek için, cihazın giriş karakteristiklerinin ayarlanması işlemi de bir optimizasyon sürecidir. Bir fonksiyonun girişinde, çeşitli parametreler çıkışında, uygunluk veya maliyet değerleri vardır. Fonksiyon veya süreç; maliyet (cost) fonksiyonu, amaç (objective) fonksiyonu veya uygunluk (fitness) fonksiyonu olarak tanımlanır. Eğer süreç deneysel ise, giriş parametreleri fiziksel büyüklüklerden oluşur (Pierre 1992). Hayat; tesadüf gibi görünen olaylarda, verilen kararlar nedeniyle çok ilginçtir. Kuantum teorisi, sonsuz boyut olduğunu ve her boyutun bir kararı temsil ettiğini söylemektedir. Hayat, lineer olmadığından kaos önemli bir rol oynar. Başlangıç Şartlarında küçük bozucu etkiler, çok farklı ve tahmin edilemeyen çözümlere sebep olmaktadır. Bir ürün tasarımında yüksek dereceli karmaşıklıklar ortaya çıkmaktadır. Bilimin gelişmesiyle birlikte geçmişte çözülemeyen karmaşıklıkların yeni çözümleri üretilmiştir. Gelişmenin bu sürecinde optimizasyon önemli bir rol oynamıştır. Çoğu optimizasyon metotları, kök veya sıfır araştırma işlemini kullanır. Matematiksel yaklaşımda, kök bulmak için fonksiyonun sıfır olduğu yerler araştırılırken, optimizasyon da türevin sıfır olduğu yerleri araştırmak gerekmektedir. Türev hesaplama her zaman kolay bir iş değildir. Teknik problemlerin birçoğu, köklerini bulmak üzere formülize edilebilir. Fakat bir kısım optimizasyon yöntemleri bu kökleri bulmada yetersiz kalmaktadır.

Optimizasyonda diğer bir zorluk; elde edilen bir sonucun, global veya lokal bir çözüm olup olmadığının belirlenmesidir. Örneğin; bulunan kökün optimal bir çözüm

olduğunu anlamak zordur. Çünkü bütün kökler, fonksiyonu sıfır yapmaktadır. Lineer olmayan bir fonksiyonun da minimumunu bulmak oldukça zordur. Bu tip problemler ya lineer bir yaklaşımla veya optimizasyon bölgesini küçük bir bölge ile sınırlamakla çözülür.

Optimizasyon algoritmaları şekil 2'de gösterildiği gibi altı grupta ele alınabilir. Ancak kesin hatlarıyla altı gruba ayrıldığı söylenemez. Örneğin; dinamik optimizasyon problemi sınırlı veya sınırsız olabilir. Bazı parametreler ayrık veya sürekli olarak tanımlanabilir (Broyden 1965). şekil 2'de verilen optimizasyon algoritmaları aşağıdaki gibi açıklanabilir:

1) Deneme-yanılma optimizasyonu; işlem hakkında çok fazla bilgi olmaksızın çıkışı etkileyen parametrelerin ayarlanmasıdır. Örneğin TV "de en iyi görüntü ve ses, deneme yanılma yoluyla ayarlanır. TV'deki görüntü ve sesin, antenin hangi eğiminde iyileşeceği anten uzmanları tarafından sadece tahmin edilir. Deneysel çalışma yapanlar ve çoğu büyük kâşifler bu yolu kullanmışlardır. Bunun aksine, matematiksel fonksiyonun optimizasyonunda, matematiksel formül ile süreç tanımlanır. Fonksiyonun optimum çözümünü bulmada değişik metotlar uygulanır. Bu yaklaşım teorisyenler tarafından tercih edilir.



Şekil 8: Optimizasyon algoritmalarının grupları

2) Tek ve çok parametrelili optimizasyon; Sadece bir parametre varsa, optimizasyon bir boyutludur. Birden fazla parametreye sahip fonksiyon için çok boyutlu optimizasyon gereklidir. Boyut sayısı artarsa, optimizasyonun zorluk derecesi de artar. Çok boyutlu optimizasyon metodunda, bir boyutlu optimizasyon metodu yaklaşımı kullanılır.

3) Statik ve dinamik optimizasyon; Statik optimizasyon zamandan bağımsızdır, dinamik optimizasyon ise zamana bağlı olarak çıkış üretir. Örneğin; bir şehrin kenar mahallesinde oturan bir insanın merkezdeki işine gitmesi için birçok yol olduğunu kabul edilirse en iyi yolun hangisi olduğu sorgulanabilir. Mesafe açısından bakılacak olursa problem statiktir. Çözüm, haritayı ve arabanın kilometre/saati kullanılarak bulunabilir. Pratikte değişkenlerin çokluğu nedeniyle problem, pek de basit değildir. En kısa yol, en hızlı yol değildir. En hızlı yolu bulmak dinamik bir problemdir ve zamana, havanın durumuna, kazalara vb. bağlıdır.

4) Sürekli ve ayırık parametrelili optimizasyon; Sürekli parametreler sonsuz değer alırken ayırık parametreler sınırlı değerler alır. Örneğin yapılacak işler bir liste halinde verilmiştir. Bu işlerin yapılması birbirinden bağımsız olduğundan ayırık parametrelili düşünülebilir. Ayırık parametrelili optimizasyon kombinasyonel bir optimizasyon olarak da adlandırılabilir. Bir çizgide $f(x)$ 'in minimum değerini bulmaya çalışmak, sürekli parametrelili optimizasyon olarak tanımlanır.

5) Sınırlı ve sınırsız optimizasyon; Sınırlı optimizasyon, parametreleri bir tanım aralığında değerlendirir. Sınırsız optimizasyonda ise parametreler herhangi bir değerde olabilir. Değişkenlerin sınırları kaldırılarak sınırlı parametreler sınırsız parametrelere çevrilirler. Çoğu nümerik optimizasyon rutinleri sınırsız parametrelerle çalışırlar. Örnek olarak $f(x)$ fonksiyonunu ele alalım ve sınırlar $-1 \leq x \leq 1$ arasında olsun. Bu fonksiyon $x=\sin(u)$ tanımı kullanılarak sınırsız optimizasyona dönüştürülür. Burada u 'nın değeri ne olursa olsun x ; $(-1,1)$ aralığında değişecektir. Sınırlı optimizasyon, lineer denklemler ve lineer sınırlarla parametreleri optimize ettiği zaman, program lineer program olarak adlandırılır. Sınırlar ve maliyet denklemleri nonlinear ise, program da nonlinear programlama problemi olur.

6) Rasgele ve minimum araştırma algoritmaları: Bazı algoritmalar parametrelerin başlangıç değerlerini ayarlayarak uygunluk değerlerini minimize etmeye çalışır. Bu araştırma tekniği, hızlı olmakla beraber lokal minimumlara ulaşabilir. Bunlar nümerik metotlara dayanan klasik optimizasyon algoritmalarıdır. Bir parametreden hareketle diğer parametreyi tespit etmek, bazı deterministik adımlarla gerçekleştirilmektedir. Diğer taraftan rasgele metotlar; parametrelerin optimum çözümünü bulmada ihtimal hesaplarını kullanırlar. Bu metotlar yavaş olmakla birlikte global minimumu bulmada daha başarılıdırlar.

Yukarıdaki gruplandırmanın sonucunda optimizasyon metotları; Deterministik metotlar, istatistiksel metotlar olmak üzere iki ana gruba ayrılabilir (Haataja 1994).

Deterministik optimizasyon metotları, lokal maksimum veya minimuma yakınsayan algoritmalar. Türevsel hesaplamalar veya türevsel yaklaşımlar deterministik metotlara örnek verilebilir. Rasgele araştırma algoritmaları gibi istatistiksel metotlar ise global minimum veya maksimumu bulmada bazı stratejileri ve rasgele sayıları kullanırlar (Palko 1996). Son yıllarda PC"lerin hızlarındaki artış bu algoritmaların uygulama sahasında sıkça görülmesine neden olmuştur. (Wurtz F. ve ark. 1997)

3.2 Genetik Algoritmalar ve Evrimsel Hesaplama

Evrimsel hesaplama bir optimizasyon işlemidir. Öyle ki, amaç, hayatta kalacak bireylerin yeteneklerini geliştirmektir. Evrimsel hesaplama (EC), bir arama işleminde doğal seçimin benzetimidir. Doğada, organizmalar hayatta kalmak ve çoğalmak için yeteneklerini etkileyen belli karakteristiklere sahiptir. Bu karakteristikler organizmanın kromozomlarında içerilen bilginin uzun stringleri şeklinde temsil edilirler. Eşlenerek çoğaltmadan sonra, yavru kromozomlar her çiftten gelen bilginin bir kombinasyonundan oluşur. Umut verici bir biçimde sonuç, her çiftin en iyi karakteristiklerini içeren yavru kromozomlar olacaktır. Doğal seçim işlemi, uygunluk değeri en yüksek bireyin seçilmesini sağlar. Evrim doğal seçim aracılığıyla, popülasyon içinden rasgele bir şekilde seçilmiş bireyler, uygun kromozom değerleri arama olarak düşünülebilir. Bu amaçla, bir evrim algoritması (EA) verilen bir probleme en uygun çözüm için olasılıklı bir aramadır.

Evrimsel algoritmaları; genetik algoritmalar (GA), genetik programlama (GP), evrimsel programlama (EP), evrimsel strateji (EV) ve benzerlerini içeren alt bölümlere ayrılabilir. Genetik algoritma (GA)"nın terminolojisinin anlaşılması için "doğal seçim" in (seleksiyonun) anlaşılması gerekir. Dünyayı gözlemleyecek olursak, olup biten olaylarda doğal seçim göze çarpar. Birbirinden ayrı muazzam organizmalar ve bu organizmalardaki karmaşıklık, inceleme ve araştırma konusudur. Organizmaların niçin böyle olduğu ve nasıl bu aşamaya geldiği sorgulanabilir. Bir başka açıdan meseleye şöyle bakılabilir. Sanki bugünün dünyasında, çok büyük optimizasyon algoritmaları oluşturulmuş ve binlerce iterasyon sonunda yer yüzünde optimum çözüm elde edilmiştir. Amaç fonksiyonu, yaşam mücadelesini temsil eder. İnsanoğlu bunu maksimize etmek ister (Grant 1985). Adaptasyon ve uygunluğun seviyesi, dünyada uzun süre yaşayabilmenin göstergesi haline gelmiştir. Evrim süreci, hayat şartlarına en uygun olanın yaşamasını sağlayan büyük bir algoritmadır. Eğer çevreyi değiştirme zeka ve yeteneğine sahip olunursa hayatta, global maksimum elde edilebilir (Haupt 1998).

Gen, temel kalıtım birimidir. Organizmanın genleri; DNA (Deoxyribo Nucleic Acid) formunda bir çift kromozomdur. Organizmanın her bir hücresi benzer miktarda kromozomlar içerirler. Her bir vücut hücresindeki kromozom miktarı farklıdır. Örneğin; sivrisinek de 6, kurbağada 26, insanda 46 ve alabalıkta 94 adet kromozom vardır. Genler, her biri farklı karakteristiği temsil eden iki fonksiyonel formda meydana gelir. Bu formların her biri “allel” olarak bilinir (Curtis 1975). Örneğin allel değerine göre insanın gözünün biri mavi olurken diğeri kahverengi olmaktadır.

Kromozomlardaki allel"lerin kombinasyonları, fertlerin kişisel özelliklerini belirler. Fert üzerinde allel"lerden biri baskın olurken diğeri pasif olmaktadır. Genetik algoritmalar, biyolojik süreci modelleyerek fonksiyonları optimize eden evrim algoritmalarıdır. GA parametreleri, biyolojideki genleri temsil ederken, parametrelerin toplu kümesi de kromozomu oluşturmaktadır. GA"ların her bir ferdi kromozomlar (bireyler) şeklinde temsil edilen popülasyonlardan oluşur. Popülasyonun uygunluğu, belirli kurallar dâhilinde maksimize veya minimize edilir. Her yeni nesil, rasgele bilgi değişimi ile oluşturulan diziler içinde hayatta kalanların birleştirilmesi ile elde edilmektedir (Angeline 1995).

Bu metot, uzun çalışmaların neticesinde ilk defa John Holland (1975) tarafından uygulanmaya başlandı. En son onun öğrencisi olan David Goldberg popüler oldu. David Goldberg tezinde; gaz boru hattının kontrolünü içeren bir problemin çözümünü genetik algoritma ile gerçekleştirdi (Goldberg,1989).

GA'ların avantajları;

- Sürekli ve ayrık parametreleri optimize etmesi
- Türevsel bilgiler gerektirmemesi
- Amaç fonksiyonunu geniş bir spektrumda araştırması
- Çok sayıda parametrelerle çalışma imkânı olması
- Paralel PC'ler kullanılarak çalıştırılabilmesi
- Karmaşık amaç fonksiyonu parametrelerini, lokal minimum veya maksimumlara takılmadan optimize edebilmesi
- Sadece tek çözüm değil, birden fazla parametrelerin optimum çözümlerini elde edebilmesi olarak sıralanabilir.
- GA'lar arama ve optimizasyon için sezgisel yöntemlerdir. Geniş arama algoritmalarının aksine, genetik algoritmalar en iyiyi seçmek için tüm farklı durumları üretmez. Bundan dolayı, mükemmel çözüme ulaşamayabilir. Fakat zaman kısıtlamalarını hesaba katan en yakın çözümlerden biridir. GA lar şartlara uyum sağlayabilir. Bunun anlamı, önceden hiç bilgisi olmamasına

karşın, olayları ve bilgiyi öğrenme ve toplama yeteneğine sahip olmasıdır. Her problemin çözümü için GA kullanmak iyi bir yol değildir. Birkaç parametrelili analitik fonksiyonun çözümünde klasik metotlar daha hızlıdır. Böyle durumlarda, nümerik metotlar tercih edilmelidir. Paralel bilgisayarlar kullanılırsa GA daha hızlı sonuç verebilir. Gelecek bölümde GA'larla benzerlik gösteren Saf Rasgele Araştırma Algoritması (pure random search) kısaca anlatılacaktır. GA'lara özel operatörler tanıtılacak ve birkaç örnekle

çalışması gösterilecek. Ayrıca, ikili kodlarla çalışan (ikili kodlu) GA ve gerçek kodlarla çalışan (gerçek kodlu) GA'ların çalışma ve işleyişi anlatılacaktır.

3.3. Rasgele Araştırma Algoritması

Rasgele Arama, belki de en basit arama işlemidir. Bir başlangıç arama noktasından veya başlangıç noktalarının kümesinden başlayan arama işlemi, arama uzayında rasgele noktaları araştırır ve kabul edilebilir bir çözüme ulaşıncaya veya maksimum iterasyon sayısı ulaşıncaya kadar devam eder. Rasgele aramayı gerçekleştirmek son derece basit olmakla beraber, verimsiz olabilir. Uygun çözüm elde edinceye kadar geçen zaman çok uzun olabilir. Rasgele araştırma için bir algoritma çalışması ve işleyişi aşağıda sunulmuştur.

Adım 1. N başlangıç arama noktaları kümesini seç. $C_g = \{ C_{g,n} = | n=1,2,...,N \}$.

Burada, $C_{g,n}$ I değişkenlerinin vektörü ve $g=0$ dır. Her bir $C_{g,n}$ elemanı,

$U(\min, \max)$ değişken değerlerin sınırı olmak üzere, verilen aralıklarda

üretilir.

Adım 2. Her bir $C_{g,n}$ vektörünün ("uygunluk") $F(C_{g,n})$ doğruluğunu değerlendir.

Adım 3. En iyi noktayı bul C_g , $best = \min\{ F(C_{g,n}) \}$

Adım 4. if $C_g, best < C_{best}$ then $C_{best} = C_g, best$ C_{best} tümünün en iyi çözümü

Adım 5. if C_{best} kabul edilebilir bir çözüm ise veya maksimum iterasyon sayısı

aşılmış ise o zaman "dur" ve çözüm olarak C_{best} "dön"

Adım 6. Her bir $C_{g,n}$, $\Delta C_{g,n}$ ile karıştır. Burada, $\Delta C_{g,n} \approx N(0, \sigma^2)$ ve σ^2 li küçük bir değişimi ifade eder

Adım 7. $g = g + 1$ artır ve adım 2 ye git.

3.4. İkili kodlu genetik algoritmalar

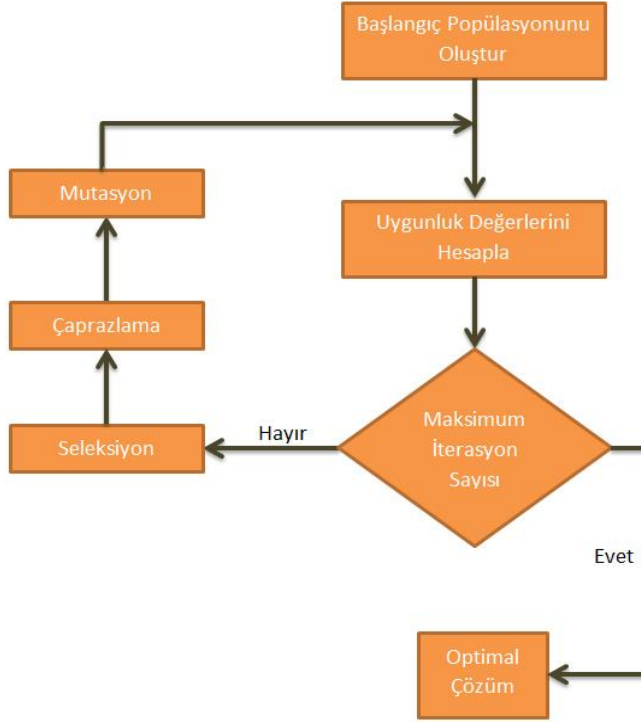
Genetik Algoritmalar (GAs), verilen bir veri kümesi için en iyi çözümü (parametreler kümesini) bulacak, doğrusal olmayan optimizasyon aracıdır. GA algoritma, mümkün çözümler kümesinin rasgele üretilmesiyle başlar. Parametreleriyle her bir çözüm, arama uzayında (kromozom veya nesil uzayı), uygunluk fonksiyonunun özel bir noktasını üretir. Her iterasyondaki bu farklı nesiller kümesi popülasyon olarak adlandırılır. Netice olarak, bir popülasyonun en iyi çözümünün bir parçasından (yarısı veya dörtte biri denebilir), çocuklar (yeni nesil) üretilir. Bu yeni nesil eskilerden daha iyi olması beklenir.

•İkili Kodlama	<table><tr><td>Kromozom1</td><td>1101100100110110</td></tr><tr><td>Kromozom2</td><td>1101111000011110</td></tr></table>	Kromozom1	1101100100110110	Kromozom2	1101111000011110		
Kromozom1	1101100100110110						
Kromozom2	1101111000011110						
•Değer Kodlama	<table><tr><td>Kromozom1</td><td>1.2324 5.3243 0.4556 2.3293 2.4545</td></tr><tr><td>Kromozom2</td><td>ABDJEIFJDHDIERJFDLDFLFEGT</td></tr><tr><td>Kromozom3</td><td>(back), (back), (right), (forward), (left)</td></tr></table>	Kromozom1	1.2324 5.3243 0.4556 2.3293 2.4545	Kromozom2	ABDJEIFJDHDIERJFDLDFLFEGT	Kromozom3	(back), (back), (right), (forward), (left)
Kromozom1	1.2324 5.3243 0.4556 2.3293 2.4545						
Kromozom2	ABDJEIFJDHDIERJFDLDFLFEGT						
Kromozom3	(back), (back), (right), (forward), (left)						
• Permütasyon Kodlama	<table><tr><td>Kromozom1</td><td>1 5 3 2 6 4 7 9 8</td></tr><tr><td>Kromozom2</td><td>8 5 6 7 2 3 1 4 9</td></tr></table>	Kromozom1	1 5 3 2 6 4 7 9 8	Kromozom2	8 5 6 7 2 3 1 4 9		
Kromozom1	1 5 3 2 6 4 7 9 8						
Kromozom2	8 5 6 7 2 3 1 4 9						

Şekil 9: Kodlama

3.4.1. İkili kodlu genetik algoritmaların operatörleri

Diğer optimizasyon metotlarında olduğu gibi ikili kodlu GA'da da amaç fonksiyonu, parametreler ve sınırlar tanımlanır. Aynı şekilde yakınsama kontrol edilerek algoritma son bulur. İkili kodlu GA'nın akış diyagramı şekil 10'da verilmektedir.



Şekil 10: İkili kodlu GA'nın akış diyagramı

Genel bir GA aşağıdaki program kod ile özetlenebilir.

1. $y = 0$ ata.
2. Cg ilk neslini oluştur.
3. While(yakınsama olmazken)
 - a. Her bireyin uygunluğunu değerlendir.
 - b. $g = g + 1$
 - c. Cg-1 den ebeveynleri seç.
 - d. Yavru Og yi şekillendirmek için çaprazlama seçilen ebeveynleri yeniden birleştir.
 - e. Og deki yavruyu mutasyona uğrat.
 - f. Önceki nesil Cg-1 ve yavru Og den yeni nesli seç.

GA ile topoğrafik haritadaki yükseltiler veya vadiler araştırılabilir. Optimizasyon algoritması vadiyi bulmak için, amaç fonksiyonunun minimum değerini ve en yüksek

bölgeyi bulmak için amaç fonksiyonunun maksimum değerini arar. Klasik metotlarda en yüksek noktayı bulmak zordur. GA ise bu noktayı rahatlıkla bulabilir.

3.4.2. Amaç fonksiyonu ve parametreler

Amaç fonksiyonu, giriş parametrelerine göre çıkış üreten bir fonksiyondur. Bu fonksiyon matematiksel veya deneysel olabilir. Burada amaç, giriş parametreleri için uygun değer bulununcaya kadar çıkış üretmektir. Örneğin banyo küveti doldurulurken farkında olmadan optimizasyon yapılır. Giriş parametreleri, sıcak ve soğuk su çeşmeleridir. Amaç fonksiyonu deneysel sonuçlardır. Çünkü el, suyun içine sokularak suyun sıcaklığı ayarlanır (Haupt 1998).

GA'ya, optimize edilecek parametre dizileri ve kromozomlar (bireyler) tanımlanarak başlanır. Kromozom, Npar adet parametreye sahip ise optimizasyon problemi Npar boyutlu olur. Parametreler P1, P2, P3,.....PNpar ise; Kromozom = [P1, P2, P3,.....PNpar] olarak tanımlanır.

Örneğin; topoğrafik bir haritayı ele alalım. Amaç fonksiyonu giriş parametreleri, enlem (x) ve boylam (y) kabul edilirse, kromozom = [x,y] şeklinde tanımlanır. Burada Npar=2' dir. Her bir kromozom; P1, P2 parametrelerinin değerlerine göre hesaplanır.

$$F(\text{kromozom}) = F(P1, P2, \dots, PNpar)$$

Çoğu zaman amaç fonksiyonu oldukça karmaşıktır. Parametrelerin hangisinin önemli olduğuna karar verilmelidir. Parametreler çok fazla olursa GA, iyi sonuç vermeyecektir. Örneğin; arabalarda yakıt tasarrufu için önemli parametreler araba boyutları, motor boyutları ve malzeme ağırlıkları şeklinde belirlenebilir. Boya rengi veya koltuk tipi gibi diğer parametreler ya çok az etkili ya da hiç etkili değildir. Bazen parametrelerin seçimi ve değer tespiti, tecrübeyle veya deneme-yanılma yoluyla yapılır. Örneğin;

$$F(w, x, y, z) = 2x + 3y + z / 100000 + \sqrt{w} / 9876$$

şeklinde analitik bir fonksiyonun parametrelerinin, 0–10 arasında değiştiğini kabul edelim.

Amaç fonksiyonu içerisinde w ve z parametreleri yüksek değerlere bölündüğünden 0–10 arasında değişmesi, sonucu pek fazla etkilemeyecektir. Eğer w ve z

parametreleri, ihmal edilirse, 4 boyutlu problem 2 boyutlu olarak modellenenebilir. Optimizasyon problemlerinin çoğunda parametrelerin belirli sınırlar içerisinde kalması istenir. Örneğin; otomobilin ağırlığı sıfır olmayacak, boyu 10 m'yi geçmeyecek vb. sınırlamalar getirilebilir. Sınırsız parametreler ise herhangi bir değer alabilir. İki türlü sınır belirlenebilir. Birincisi; $0 \leq x \leq 10$ arasında değişen bir parametrede, eğer $x=11$ değerini alırsa $x=10$ yapılır ve $x=-1$ değerini alırsa $x=0$ yapılır. İkincisi; $x = 5\sin y + 5$ olarak tanımlanırsa, y 'nin herhangi bir değerinde $0 \leq x \leq 10$ arası değişecektir. Böyle bir dönüştürmeyle sınırlı optimizasyon problemi sınırsız optimizasyon problemine çevrilmiş olur (Holland, J.H. 1992). Bağımlı parametreler, optimizasyon algoritmaları için özel problemler meydana getirir. Çünkü bir parametrenin değişimi diğer parametreleri de etkileyecektir. Örneğin; arabanın boyunu artırmak, ağırlığını da artıracaktır. Fourier serisi katsayılarında olduğu gibi bağımsız parametreler birbirlerini etkilemezler. GA literatüründe parametre etkileşimi, "epistasis" olarak adlandırılır. Biyolojide epistasis, genlerin birbirini etkilemesi olarak tanımlanır. Epistasis çok az olursa, minimum araştırma algoritmaları iyi sonuçlar üretir. GA'da Epistasis orta ve yüksek derecede olursa iyi sonuç vermemektedir, çok yüksek veya çok düşük olursa iyi sonuç vermemektedir. (Haupt 1998)

3.4.2.1. Parametrelerin tanımlanması ve kodlama

şekil 4'den, GA'nın üç işlem için döngüye girdiğini görebiliriz.

- En iyi gen stringinin seçimi
- Genetik operasyon (çaprazlama, mutasyon)
- Yeni gen stringleri (çocuklar) ile eski popülasyonun kötü gen stringlerinin yer değiştirilmesi.

Optimizasyon döngüsü başlamadan önce, optimize edilmesi gereken parametreler istenilen şekle dönüştürülmek zorundadır. Buna kodlama (encoding) denir. Kodlama GA için önemli bir konudur. Çünkü sistemden gözlemlenen bilgiye bakış açısı büyük ölçüde sınırlandırılabilir. Gen stringi probleme özel bilgiyi depolar. Gen olarak adlandırılan her bir öge, genellikle değişkenler stringi olarak ifade edilir. Değişkenler ikili veya reel sayı şeklinde gösterilebilir ve aralığı probleme özel olarak tanımlanır.

İkili Kodlama

1	1	0	1	0	0	1	0
---	---	---	---	---	---	---	---

Reel Sayı Kodlaması

43	4	9	16	30	22	59	12
----	---	---	----	----	----	----	----

Şekil 11: Gen stringlerinde parametrelerin kodlanması

İkili kodlama sıklıkla kullanılırken, şekil 11’de gösterildiği gibi Hamming cliffs dezavantajına sahiptir. Bir Hamming cliff, iki bitişik sayısal değer ayrı bit gösterimlerine sahip olduğunda şekillenir. Mesela onluk sayılar 7 ve 8 i düşünün. ikili gösterimlerle ilgili olarak (4 bit gösterimini kullanan) $7=0111$ ve $8=1000$ 4 hamming mesafesidir. (Hamming mesafesi birbirine benzemeyen ilgili bitlerin sayısıdır.) Bu, değişkenlerdeki küçük bir değişimin uygunluktaki küçük bir değişiklikte sonuç vermesi gerektiği zaman, bir problemi ortaya koyar. Eğer örneğin, uygun çözüm 7 olsun. Fakat yeni en iyi çözüm 8 olduğunda; 8 i elde etmek için birçok bitin değiştirilmeye ihtiyaç duyulduğu aşikârdır. Hâlbuki uygunluk değerinde küçük bir değişim söz konusudur.

Alternatif bir bit gösterimi Gray Kodlama kullanmaktır. Ardışık sayısal değerlerin gösterimleri arasındaki Hamming Mesafesi bunlardan biridir.

İkili sayılar dönüşüm kullanılarak kolaylıkla Gray kodlamaya çevrilebilirler. GA’lar ayrıca integer veya reel değerleri temsil eder ve bir kromozomda değişkenlerin sırasının önemli bir rol oynadığı yerdeki komut tabanlı temsiller kullanılarak geliştirilmiştir. Gaz için klasik temsil şeması, sabit uzunluklu ikili vektörlerdir. Bir I -boyutlu arama uzayının durumunda, her bir birey bir bit stringi olarak kodlanan her bir değişkenle I değişkenlerinden oluşur. Eğer değişkenlerin ikili değeri varsa, her bir kromozomun uzunluğu I bittir. Nominal değerli değişkenlerin durumunda, her bir nominal değer D boyutlu bit vektörleri olarak kodlanır. Bu değişken için $2D$, her bir D -bit stringi farklı bir nominal değerle temsil edilir. Ayrık nominal değerlerinin toplam sayısıdır. Sürekli değerli değişkenlerin durumunda, her bir değişken D boyutlu bir bit vektöre haritalanmalıdır.

Sürekli uzay aralığı, sonlu bir $[\alpha, \beta]$ aralığına sınırlandırılması gerekir. Standart ikili kod çözmeyi kullanarak her bir kromozom C_n in sürekli değişkeni $C_{n,i}$ sabit

uzunluklu bir bit stringi kullanılarak kodlanır. Amaç fonksiyonunun uygunluk değerinin hesaplanması için her iterasyonda kromozomun 10"luk sayı sistemine çevrilmesi gerekir.

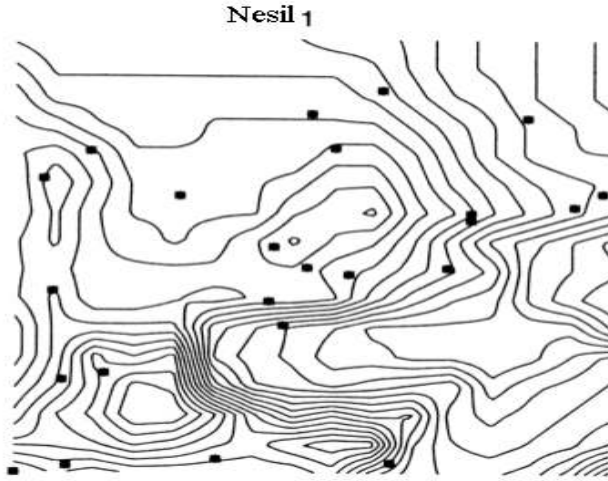
3.5. Başlangıç popülasyonu

Kromozom, başlangıçta verilen popülasyon miktarına göre rasgele üretilir. Bu durumda NipopxNbit adet bitlerle matris oluşturulur. Nipop, başlangıç popülasyonunun toplam sayısıdır, Nbit"de kromozomun bit sayısıdır. Matrisin her bir satırı, bir kromozomu temsil eder. Kromozomlar ise, enlem ve boylamları temsil eder. Rasgele üretilen popülasyondaki parametreler, amaç fonksiyonunda yerlerine konularak ve uygunluk değerleri hesaplanarak değerlendirilir. Yirmi dört adet kromozomun uygunluk değerleri Çizelge 1"de verilmiştir. Kromozomların yerleşimleri şekil 12"da topoğrafik harita üzerinde gösterilmiştir.

Çizelge 1: Başlangıç popülasyonunun sıralanışı (Haupt 1998)

Kromozomlar	Uygunluk değerleri
00000000000000	-13000
11111011010010	-11800
00010110000010	-13255
11000011001010	-12347
01111111101001	-12560
01000111010001	-12700
01010110000100	-13338
11101111001110	-11890
01111100111100	-12953
00100001011110	-12891
10001110111010	-12759
10111000111100	-12320
11011011101000	-11797
00100110011101	-13778
00010100011011	-13360
01110010101011	-12220
11000011001100	-12452
10011101110000	-12335
10100000000011	-12857

00001101010110	-13166
00010000110101	-13164
01101100110010	-12927
01101111000010	-13079
10001001011111	-12756



Şekil 12: Başlangıç popülasyonunun topoğrafik harita üzerinde dağılımı(Haupt 1998)

3.6.Doğal seçim

Başlangıç popülasyonu fazla ise bazı iteratif adımların uygulanması gerekir. Kromozom olarak en uygun olanı hayatta kalmalı diğerleri de yok olmalıdır. Doğal seçim, algoritmanın her bir iterasyonunda (jenerasyonunda) meydana gelir. Burada ne kadar kromozomun elde tutulacağına karar vermek biraz keyfidir. Kromozomlarda gelecek nesil içerisinde yer alma mücadelesi başlamaktadır. Popülasyon içerisinde bütün kromozomların seçilmesi gelecek nesile çok fazla katkısı olmamaktadır. GA'da değişik seçim metotları vardır.

a) Rasgele seçim: Kromozomların eşleştirilmesinde rasgele sayı üretici kullanılır. Kromozomlar 1'den başlayarak N^{iyi} 'ye kadar sıralanır. Birinci eşleştirmeyi bulmak için iki adet rasgele sayı üretilir.

$Kromozom = roundup\{N^{iyi} \times rasgele\ sayı\}$

Burada roundup fonksiyonu, sayıları en yüksek sayıya yuvarlar. Örneğin; rasgele olarak üretilen 6 sayı 0.1535, 0.6781, 0.0872, 0.1936, 0.7021 ve 0.3933 ise bu sayılar, 6 ile çarpılıp bir üst tamsayıya yuvarlanarak; 1, 5, 1, 2, 5, 3 değerleri elde

edilir. Buna göre kromozom1-kromozom5, kromozom1- kromozom2 ve kromozom5-kromozom3 eşleştirilecektir.

b) Ağırlıklı seçim: İlk olarak amaç fonksiyonunda, kromozomların uygunluk değerleri hesaplanır. Hesaplanan uygunluk değerleri en küçükten en büyüğe doğru sıraya konur. $N_{pop} \leq N_{ipop}$ ise tutulur, geriye kalanlar atılır. Burada N_{pop} değeri N_{ipop} 'a kadar olabilir. Genelde popülasyonun %50'sinin seçilmesi ($N_{pop} = N_{ipop}/2$) uygun seçenektir. Seçilen N_{pop} 'un yarısı N^* yi, yarısı da $N_{kötü}$ olarak ayrılır. N^* yi olanlar eşleştirme havuzuna konurken $N_{kötü}$ olanlar eşleştirme havuzundan atılır. Toplam popülasyon sayısı $N_{ipop} = 24$ 'dür. GA'nın her bir iterasyonunda popülasyonun 12'si tutulur ve bu kromozomların altı tanesi eşleştirme havuzuna atılır.

c) Ağırlıklı rasgele seçim: Bu yöntem, eşleştirme havuzundaki kromozomların, amaç fonksiyonunda hesaplanan uygunluk değerlerine göre seçilme ihtimallerini dikkate alır. En düşük uygunluk değerine sahip kromozom, eşleştirme ihtimali en yüksek kromozom olurken en yüksek uygunluk değerine sahip kromozom eşleştirme ihtimali en düşük olarak düşünülür. Üretilen rasgele sayı, hangi kromozomun seçileceğini belirler. Bu yöntem kendi arasında ikiye ayrılır:

Sıra Ağırlık: Kromozomlar büyükten küçüğe doğru sıralanır ve denklem kullanılarak kromozomların sıralamasındaki yerlerine göre P_n olasılıkları hesaplanır.

Bu yöntemle uygunluk değerlerinden bağımsız olarak, sadece sıralamayı dikkate alan P_n normalize değerleri elde edilmektedir. Bu metot uygulanarak alınan sonuç Çizelge 2 'de gösterilmiştir.

Çizelge 2: Eşleştirme olasılıkları (Haupt 1998)

n	Kromozom	P_n	$\sum_{i=1}^n P_i$
1	00100110011101	0.2857	0.2857
2	00010100011011	0.2381	0.5238
3	01010110000100	0.1905	0.7143
4	00010110000010	0.1429	0.8572
5	00001101010110	0.0952	0.9524
6	00010000110101	0.0476	1.0000

Kromozomların seçilmesinde kümülatif ihtimaller kullanılır. Rasgele olara 0–1 arasında bir sayı üretilir. Bu sayı, kümülatif ihtimallerin en başından başlanarak

sırayla karşılaştırılır. Kümülatif ihtimal değeri, üretilen rasgele sayıdan büyükse, eşleştirme havuzu için kromozom seçilir. Benzer tarzda 6 tane rasgele sayı üretilerek eşleştirme gerçekleştirilir. Üretilen rasgele sayılar

0.1535, 0.6781, 0.0872, 0.1936, 0.7021 ve 0.3933"dir. Bu sayılara göre kromozom1-kromozom3, kromozom1-kromozom1 ve kromozom3-kromozom2 eşleştirilirler. Kromozomlar kendisi ile eşleştirilirse, bu kromozomun gelecek nesilde daha etkin olarak yer aldığı söylenebilir. Bu yaklaşım tarzı doğaya daha uygundur.

Değer ağırlık: Kromozomların uygunluk değerleri amaç fonksiyonundan hesaplanır. Eşleştirme havuzunda $N_{iyi}+1$ "inci sırada bulunan kromozomun amaç fonksiyonunda hesaplanan uygunluk değeri, bütün eşleştirme havuzundaki kromozomlardan çıkartılır. $N_{iyi}+1$ "inci, sıradaki kromozomun uygunluk değeri – 13079"dur.

Daha önce üretilen rasgele sayılarkullanılırsa kromozom1-kromozom3, kromozom1-kromozom1 ve kromozom3- kromozom1 eşleştiği görülür.

Çizelge 3: Eşleştirme havuzunda kromozomların sıralanışı (Haupt 1998)

n	Kromozom	Cn	Pn	$\sum_{i=1}^n P_i$
1	00100110011101	-13778+13079= -699	0.4401	0.4401
2	00010100011011	-13360+13079= -281	0.1772	0.6174
3	01010110000100	-13338+13079= -259	0.1632	0.7805
4	00010110000010	-13255+13079= -176	0.1109	0.8915
5	00001101010110	-13166+13079= -87	0.0547	0.9461
6	00010000110101	-13164+13079= -85	0.0539	1.0000

d) Eşik değer seçim: Eşik değerden küçük olan kromozomlar yaşamaya devam eder, eşik değeri aşan kromozomlar ölür. Bu tekniğin avantajı, popülasyonun sıralanmasına gerek olmamasıdır.

e) Rulet tekerleği seçim: Rulet tekerleği seçim metodunun uygulanması için, ilk önce kromozomların toplam uygunluk değeri hesaplanır. Kromozomlar, toplam uygunluk değerine bölünerek her bir kromozom için 0–1 arasında değişen seçim ihtimalleri bulunur. Daha sonra, kümülatif ihtimaller hesaplanır. Popülasyon sayısı kadar "rasgele" 0-1 arasında sayılar üretilir.

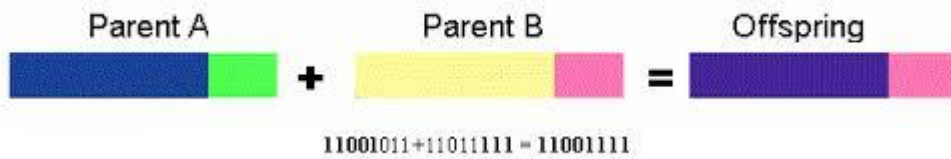
Üretilen rasgele sayı, birinci kromozomun kümülatif seçim ihtimalinden küçük ise, birinci kromozom seçilir. Eğer değilse, ikinci kromozomun veya diğerlerinin kümülatif ihtimalleriyle karşılaştırılarak hangisinden küçükse o kromozom seçilir. Böylece Rulet seçim metodu gerçekleştirilmiş olur.

f) Turnuva Seçimi: Bu yöntemde bir grup k bireyi rasgele olarak seçilir. Bu k bireyleri daha sonra bir turnuvaya katılır ve en iyi uygunluklu birey seçilir. Çaprazlama için iki turnuva tutulur. Bunlardan biri her bir ebeveyni seçmek içindir. Bir kereden daha fazla üretmek için seçilebilecek bir ebeveyn mümkündür. Turnuva seçiminin avantajı, popülasyonun daha kötü bireylerinin seçilmeyecek olması ve bundan dolayı sonraki neslin genetik yapısına katılmayacak olmasıdır. Ayrıca en iyi bireyler tekrar üretim sürecinde baskın olmayacaktır.

g) Seçkinlik(Elitism): Bu işlem, şu anki nesilden sonraki nesle aktarılan bir grup bireyin seçimini kapsar. Mutasyona uğratılmaksızın, sonraki nesle aktarılan bireylerin sayısı, nesil boşluğu olarak işaret edilir. Eğer nesil boşluğu sıfırsa; yeni nesil, yeni bireylerin girişleriyle oluşur. Pozitif nesil boşlukları için, diyelim ki k, k bireyleri sonraki nesle aktarılır. Seçilen genler sonraki yeni nesli oluşturmak için çaprazlama ve mutasyon genetik operasyonlarından geçer.

3.7. Çaprazlama

İki adet yeni nesil elde etmek için kromozomların bulunduğu eşleme havuzundan iki adet kromozom seçilir. Eşleme sürecinde, seçilen kromozomlardan bir ve birden fazla yeni nesil oluşturma olayına “çaprazlama” denir. En yaygın olarak kullanılan iki kromozomdan iki tane yeni nesil elde edilmesidir. Öncelikle kromozomun ilk biti ile son biti arasında çaprazlama noktası rasgele seçilir. Birinci kromozomun çaprazlama noktasının solundaki bitlerin hepsi ikinci kromozoma geçerken, benzer tarzda ikinci kromozomun çaprazlama noktasının solundaki bitler de birinci kromozoma geçerler. Sonuçta meydana gelen yeni kromozomlar, her iki kromozomun karakterlerini içermektedirler. Çaprazlama oranı, bir popülasyonda ne kadar kromozomun çaprazlamaya tabi tutulacağını belirler.



Şekil 13: Tek noktalı çaprazlama



Şekil 14: Gerçek kodlu GA'da çaprazlama

Gerçek kodlu GA'da sonraki neslin reel sayılarını değiştirmedigine dikkat edin. Çünkü çaprazlama noktası her zaman reel sayılar arasındadır. (1,5,10 ve 14) Sonuçta meydana gelen nesiller her iki kromozomun karakterlerini içermektedirler. Bu durum Çizelge 4 "de gösterilmiştir.

Çizelge 4: Eşleştirmesi yapılan kromozom çiftleri (Haupt 1998)

İşlem Sırası	6xrasgele(2-1)	Kromozom
1	2-1	1
2	5-5	5
3	6-3	3
4	4-5	4
5	1-1	1
6	4-5	4

3.8.Mutasyon

GA operatörlerinin ikincisi olan mutasyon, kromozomdaki bitlerin küçük bir yüzdesini değiştirir. Mutasyon ile kromozomdaki bitler "1" ise "0", "0" ise "1" yapılır. NpopxNbit"lik bir popülasyon matrisinde, mutasyon noktaları rasgele seçilir.

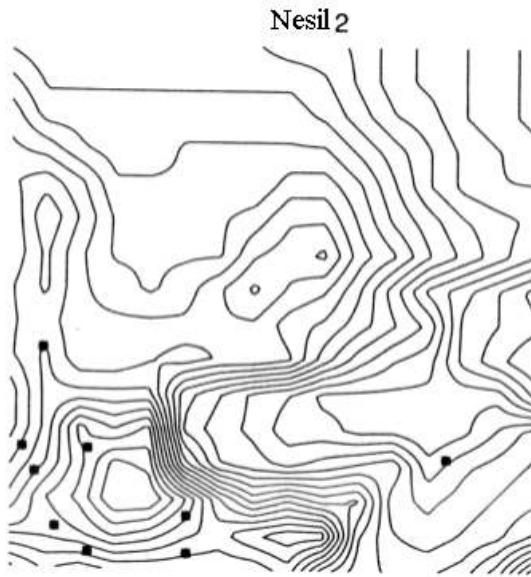
yazılan bitler, mutasyona uğrayanlardır. Çizelge 6'de görüldüğü gibi bir sonraki adımda kromozomlar uygunluk değerlerine göre yeniden sıralanır ve eşleştirme havuzuna konur. Başlangıç popülasyonun ortalama değeri, -12738 iken birinci nesilden sonra bu değer, -13334'e ulaşmıştır. Çizelge 6'deki kromozomların topoğrafik harita üzerinde yerleşimi şekil 15'de verilmiştir. Eşleştirme havuzunda kromozomların sadece altı tanesi tutulmaktadır. Tekrar çaprazlama, mutasyon ve sıralamadan sonra elde edilen üçüncü nesil Çizelge 7'de gösterilmektedir. Üçüncü neslin sonunda popülasyon genel ortalaması -13403 olmaktadır. Dördüncü neslin sonunda ise -13676 ve yerleşimi şekil 16'da görülmektedir.

Çizelge 5: Çaprazlamaya ve mutasyona uğramış yeni nesil (Haupt 1998)

Kromozomlar	Uygunluk değerleri
00100110011101	-13778
11010100011011	-11956
01010110000100	-13338
00010110001010	-13553
00001100010111	-13289
00010000110101	-13164
00100110000100	-13372
01010110001101	-13632
00000110011101	-13036
00100110011101	-13778
00100110011101	-13778
01010110000100	-13338

Çizelge 6: İkinci nesilden sonra sıralama (Haupt 1998)

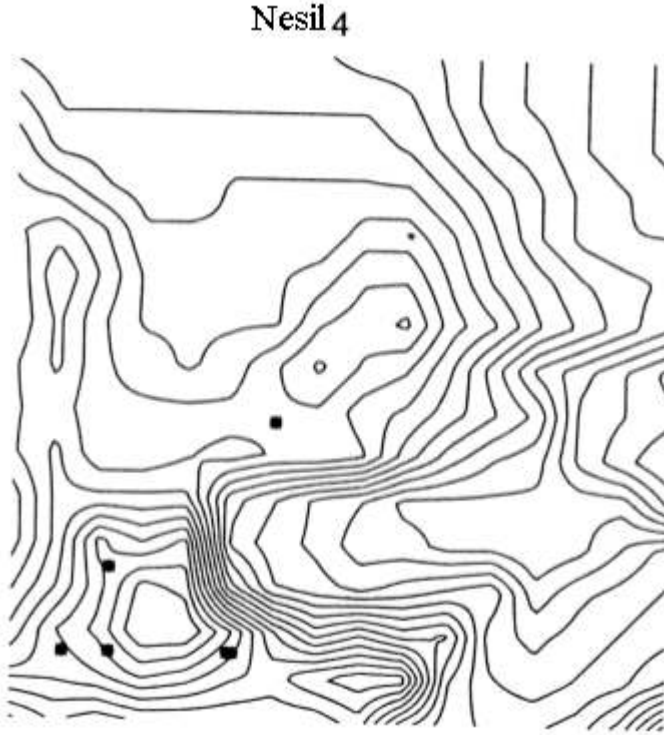
Kromozomlar	Uygunluk değerleri
00100110011101	-13778
00100110011101	-13778
00100110011101	-13778
01010110001101	-13632
00010110001010	-13552
00100110000100	-13372
01010110000100	-13338
01010110000100	-13338
00000110011101	-13036
00001100010111	-13289
00010000110101	-13164
11010100011011	-11956



Şekil 16: İkinci neslin topoğrafik harita üzerinde dağılımı (Haupt 1998)

Çizelge 7: Üçüncü nesilden sonra sıralama (Haupt 1998)

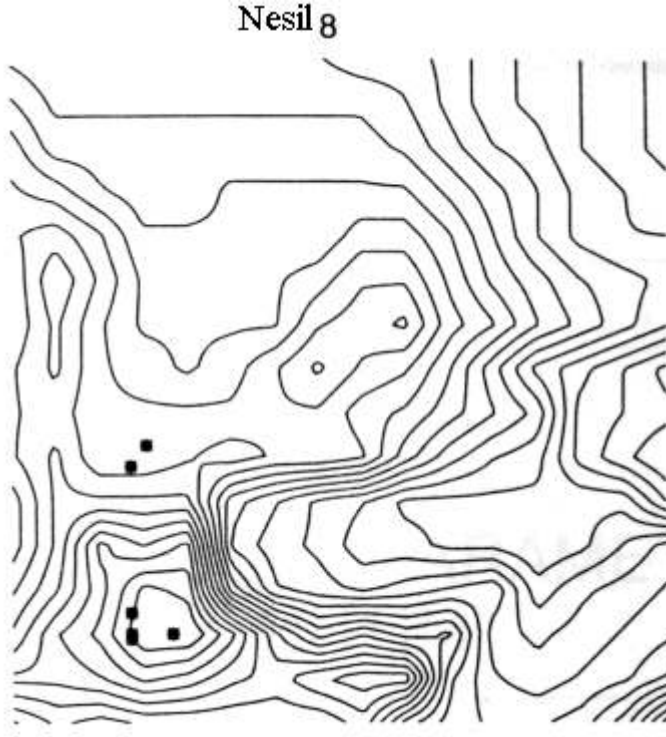
Kromozomlar	Uygunluk Değerleri
00100110011101	-13778
00100110011101	-13778
00100110011101	-13778
00100110011101	-13778
00100110011101	-13778
01010110001101	-13632
00010110001100	-13584
00010110001010	-13553
01010110001010	-13539
00100111010100	-13921
01101111011101	-13602
01100110011101	-13255



Şekil 17: Dördüncü neslin topoğrafik harita üzerinde dağılımı (Haupt 1998)

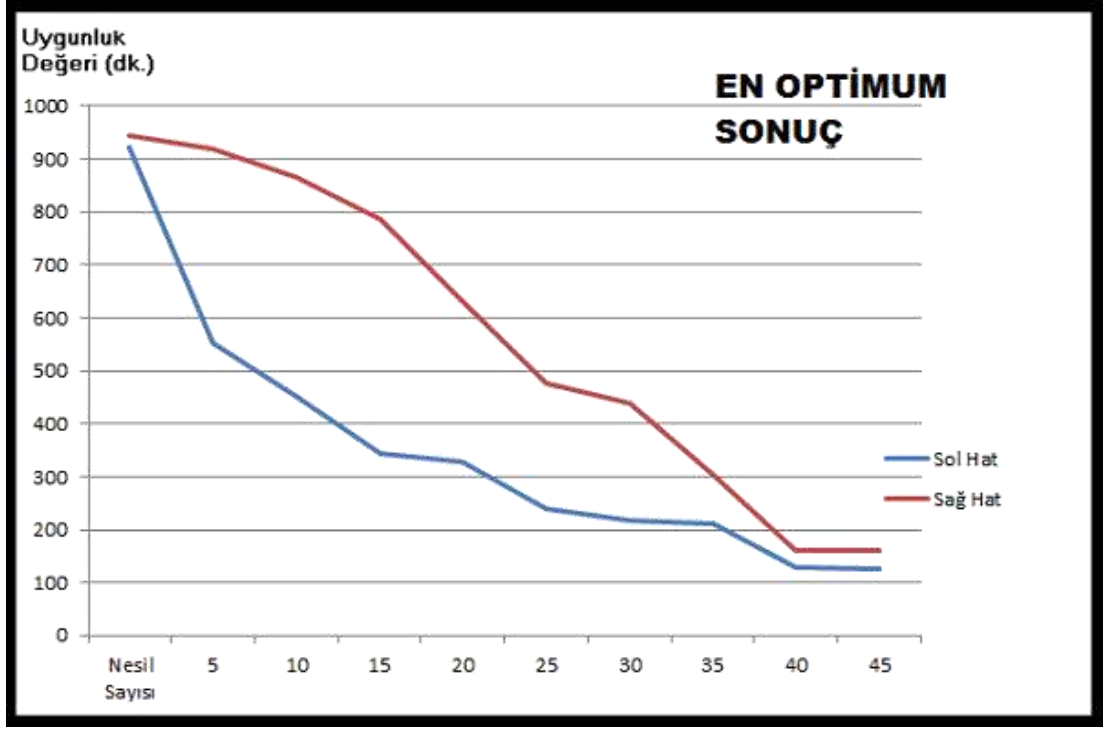
3.8.2. Yakınsama

İterasyon sayısına veya uygun bir çözüme ulaşıncaya kadar işleme devam edilir. Bütün kromozomların uygunluk değerleri değişmiyorsa algoritma durdurulmalıdır. Çoğu GA'lar genel ortalama, standart sapma ve minimum uygunluk değeri gibi istatistikleri tutarlar. Bu istatistikler, yakınsama testi için bir ölçü olabilir.



Şekil 18: Sekizinci neslin topoğrafik harita üzerinde dağılımı (Haupt 1998)

Son popülasyondaki kromozomların yerleşimi şekil 18'de gösterilmektedir. Sekizinci neslin sonunda topoğrafik harita üzerinde iki minimum noktası etrafında yoğunlaşma görülmektedir. Örnekte dokuzuncu iterasyonun sonunda global minimum -14199 olarak elde edilmiştir. Algoritmanın minimuma yakınsaması ve her bir neslin genel ortalaması şekil 19'de verilmektedir. Popülasyonun ortalama uygunluk değeri, algoritmanın yakınsaması hakkında bilgi vermesi bakımından önemli bir kriterdir. Örneğin maksimizasyon probleminde, farklı iki seçim metodu karşılaştırılıyor ise ortalama uygunluk değeri yüksek olan seçim metodu daha iyi sonuç vermektedir denilebilir.

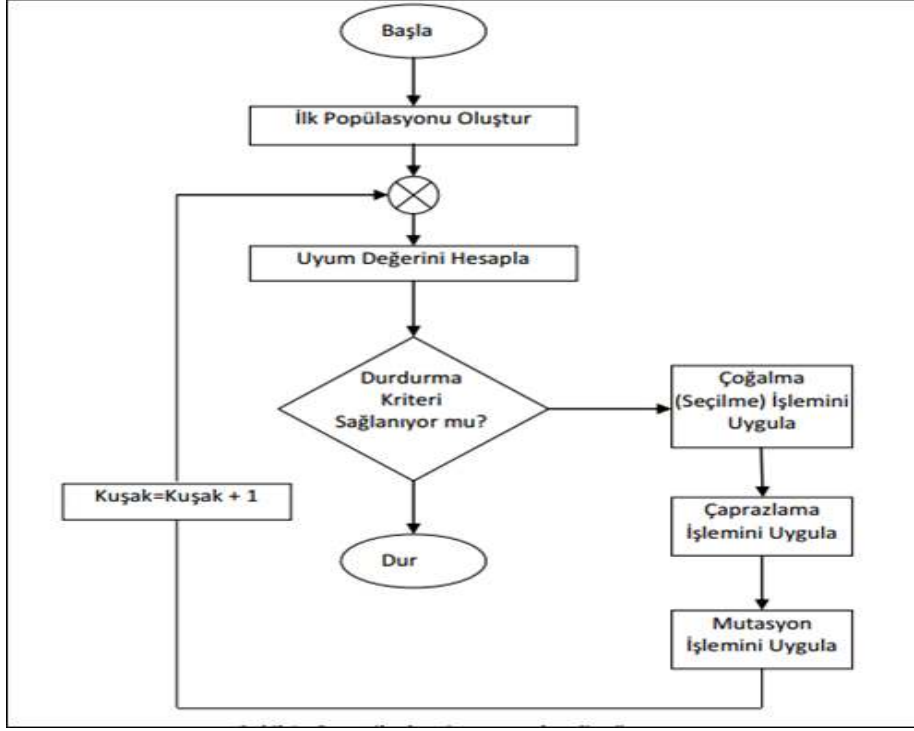


Şekil 19: Genetik Algoritma ile Çift Taraflı Montaj Hattı Dengeleme

3.9. Gerçek kodlu genetik algoritma

Problemlerin çözümünde rakamların hassasiyeti önemli bir yer tutmaktadır. İkili kodlu GA'da, parametrelerin "1" ve "0" larla ifade edilmesi, kromozomların boyutlarını oldukça artırdığından sınırlı hassasiyete sahiptir. Bunun yerine gerçek rakamlarla kodlama yapabilen, gerçek kodlu GA'yı kullanmak avantajlıdır. Gerçek kodlu GA, hem daha hassas hem de PC belleğinde daha az yer kaplamaktadır. Çunkaş ve Akkaya (2002), değişik test fonksiyonları kullanarak ikili kodlu GA ve gerçek kodlu GA'nın temel farklılıklarını irdelenmişler. Gerçek kodlu GA'ların ikili kodlu GA'ya göre hızlı çalıştığını ve global optimumu daha kısa sürede bulduğunu göstermişlerdir.

Gerçek kodlu GA bazı literatürlerde "Sürekli Parametrelili GA" olarak da bilinmektedir. Bu çalışmada "Gerçek Kodlu GA" tanımlaması kullanılmıştır. Gerçek kodlu GA, ikili kodlu GA'ya çok benzemektedir. Aralarındaki en önemli farklılık parametrelerin "1" ve "0" lar yerine gerçek rakamlarla ifade edilmesidir.



Şekil 20: GA'nın akış diyagramı

3.10.Amaç fonksiyonu ve parametreler

Optimizasyonda amaç, parametrelerin en uygun değerlerini bulmaktır. Optimize edilecek parametre değerleri, bir dizi haline getirilip kromozom şekline sokularak GA ile çözülür. N^{par} parametreye sahip bir kromozom $1 \times N^{\text{par}}$ boyutlu matris biçiminde yazılır.

$$\text{Kromozom} = [P_1, P_2, P_3, \dots, N^{\text{par}} P]$$

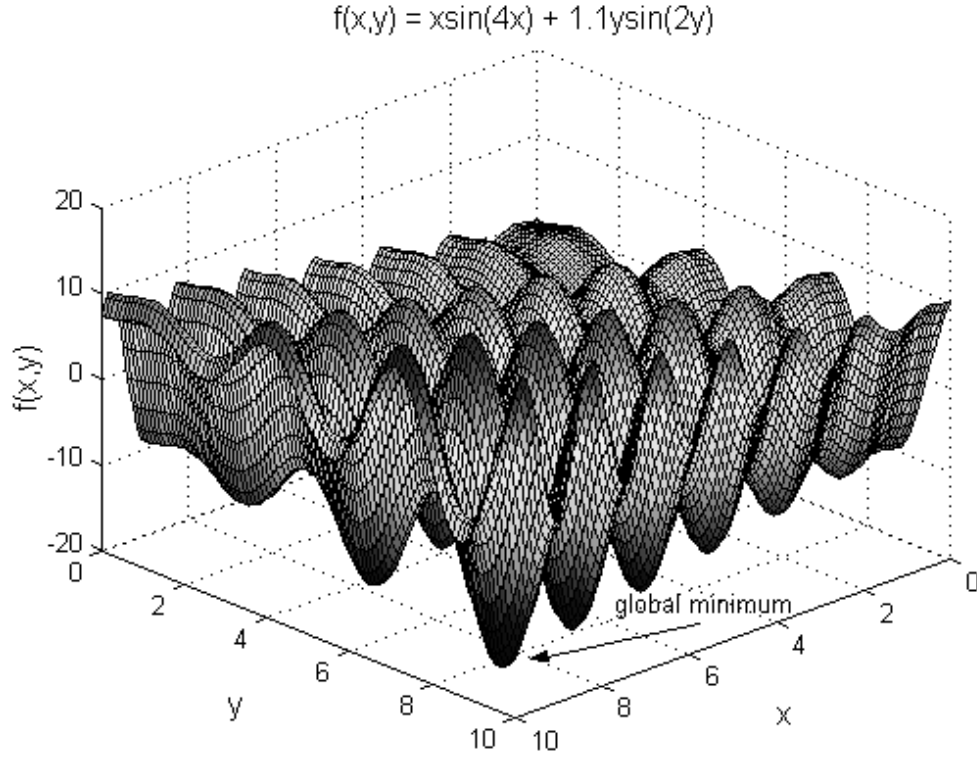
Gerçek kodlu GA'da parametrelerin hepsi "ondalık" rakamlarla ifade edilir ve her bir kromozomun bir uygunluk değeri (C) vardır.

$$C = f(\text{kromozom}) = f(P_1, P_2, P_3, \dots, N^{\text{par}} P)$$

Örneğin; amaç fonksiyonu

$$C = f(x, y) = x \sin(4x) + 1,1y \sin(2y)$$

ve parametre sınırları $0 \leq x \leq 10$, $0 \leq y \leq 10$ olarak tanımlanırsa Kromozom=[x, y] şeklinde ifade edilebilir. Buradaki problem iki parametrelili bir optimizasyon problemi olup Şekil 21'de grafik olarak gösterilmiştir. Bu problem, Bölüm bir önceki örneğe göre daha zor bir optimizasyon problemidir ve klasik metotlarla bu fonksiyonun global minimumunu bulmak zordur. GA kullanılırsa global minimumu bulmak kolaylaşır.



Şekil 21: $f(x, y) = x \sin(4x) + 1.1 y \sin(2y)$ "nin üç boyutlu görünümü

3.11. Başlangıç Popülasyonu

GA'nın çalışmaya başlayabilmesi için, Nipop kromozomdan oluşan başlangıç popülasyonuna ihtiyaç vardır. Popülasyonun kromozomları $N_{par} \times N_{ipop}$ luk bir matris ile temsil edilir. Kromozomlar denklem (3.5) kullanılarak rasgele üretilir.

$$IPOP = (PH - PL) \times \text{rasgele}\{N_{ipop}, N_{par}\} + PL \quad (3.5)$$

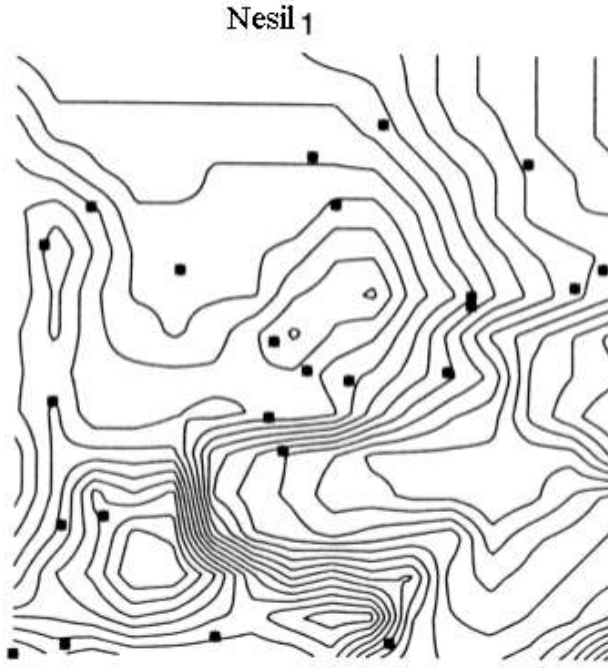
Burada;

PH → Parametrenin üst sınır değeri

PL → Parametrenin alt sınır değeri

$\text{rasgele}\{N_{ipop}, N_{par}\}$ → $N_{ipop} \times N_{par}$ olacak şekilde matris formunda 0-1 arasında üretilen rasgele sayıdır.

Bu tanımlamalar her parametre için kullanılabilir. Parametrelerin bu şekilde değerleri belirlendikten sonra amaç fonksiyonunda yerlerine konularak fertlerin uygunluk değerleri hesaplanır. Denklem (3.4)'de görüldüğü gibi x ve y den oluşan iki tane parametre vardır. Sınır değerleri $PL = 0$ ve $PH = 10$ dur. Amaç fonksiyonu biraz karmaşık olduğundan, başlangıç popülasyonunun yüksek tutulması iyi sonuç vermektedir. Başlangıç popülasyon sayısı $N_{pop}=48$ seçilir ve bu durumda popülasyon matrisi 48×2 'dir. Başlangıç popülasyonun büyüklüğü araştırma uzayının daha geniş seçilmesini sağlar.



Şekil 22: Birinci nesil

3.12. Doğal seçim

Gelecek nesilde, hangi kromozomun yer alacağını belirlemek gerekir. İkili kodlu GA'da yapıldığı gibi kromozomlar, büyükten küçüğe doğru sıralanır. Gerçek kodlu GA'da gelecek iterasyonda kullanılmak üzere N_{pop} kadar kromozom tutulur ve geri kalanı dikkate alınmaz. Doğal seçim, en iyi kromozom uygunluk değeri bulununcaya kadar devam etmelidir. N_{pop} kromozomları, kendi aralarında N_{iyi} ve

$N_{kötü}$ olarak büyükten küçüğe doğru sıralamaya tabi tutulur. N_{iyi} , eşleştirme havuzuna atılırken, $N_{kötü}$ hariçte tutulur. Örnekte 48 adet kromozomun ortalama uygunluk değeri 0.9039 ve en iyi kromozomun uygunluk değeri -16.26'dır. Alt sıradan başlanarak 24 tanesi atılırsa, geri kalan kromozomların ortalama uygunluk

değeri -4.27 olmaktadır. Çizelgede gösterildiği gibi her bir iterasyonda Npop= 24 kromozom kullanılır ve bunun 12 tanesi Niyi, 12 tanesi Nkötü olarak tanımlanır.

3.13.GENETİK ALGORİTMA SEÇİM YÖNTEMLERİ

3.13.1.Seçim

Kromozomların eşleştirilmesinde ikili kodlu GA'da tanımlanan Ağırlıklı eşleştirme metodu kullanılmış ve uygunluk değerlerine göre kromozomların seçilme ihtimalleri Çizelge 8'de verilmiştir. Çizelgede görüldüğü gibi eşleştirme çoğunlukla yukarıdan aşağıya doğru yapılmaktadır. Çünkü sıralamanın altındaki kromozomların uygunluk değerleri küçüktür. Rasgele üretilen 6 tane sayı aşağıya sıralanmıştır.

(0.4679,0.5344), (0.2872,0.4985), (0.1783,0.9554), (0.1537,0.7483),
(0.5717,0.5546), (0.8024,0.8907). Üretilen bu sayılar kullanılarak eşleştirme aşağıdaki gibi gerçekleştirilir.

Anne = [3, 2, 1, 1, 4, 5]

Baba = [3, 3, 10, 5, 3, 7]

Çizelge 8: Kromozomların uygunluk değerlerine göre seçilme ihtimalleri

n	Pn	$\sum_{i=1}^n P_i$
1	0.2265	0.2265
2	0.1787	0.4052
3	0.1558	0.5611
4	0.1429	0.7040
5	0.1230	0.8269
6	0.0367	0.8637
7	0.0308	0.8945
8	0.0296	0.9241
9	0.0247	0.9488
10	0.0236	0.9724
11	0.0168	0.9892
12	0.0108	1.0000

3.13.2. Çaprazlama

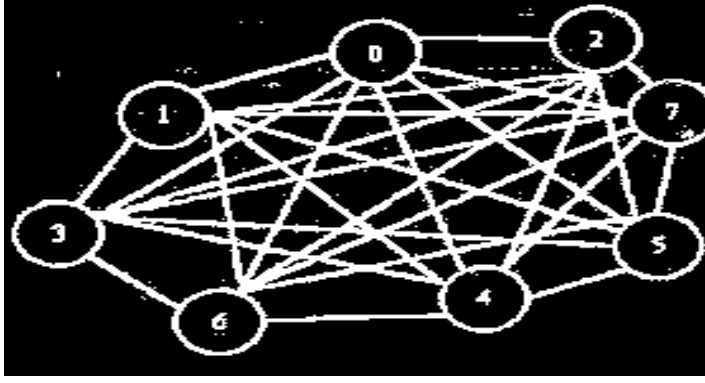
Çaprazlama için çok farklı yaklaşımlar vardır. Adewuya (1996) ve Michalewicz (1994) tarafından ilginç metotlar ortaya konulmuştur. Kromozomun birveya birden fazla yerinden bölünerek işaretleme yapılması ve işaretlenen yerlere göre kromozomun karşılıklı genlerinin yerlerinin değiştirilmesi “tek düzen metot” olarak adlandırılır.

3.13.3. Mutasyon

Genetik algoritmalar, bazen çok hızlı yakınsarlar. Yakınsama sonucunda global maksimum bulunursa sonuç iyi, lokal minimumlar bulunursa beklenen sonuç elde edilememiş demektir. Hızlı yakınsamadan kurtulmanın yolu, araştırma uzayında, mutasyon aracılığıyla yeni çözümler elde etmektir. İkili kodlu genetik algortmada”da, mutasyon oranı %1-%5 arasında değiştirilirse iyi sonuçlar elde edilir (Çunkaş ve Akkaya 2004). Gerçek kodlu GA”da mutasyon oranı ise daha yüksektir. Parametrelerin toplam sayısı ile mutasyon oranı çarpılarak mutasyona girecek parametre sayısı tespit edilir. Bir matriste, satır ve sütunu ifade edecek şekilde rasgele rakamlar üretilerek mutasyon gerçekleştirilir. Örneğin; mutasyon değerini $\mu=0.04$ olarak alınırsa, mutasyona uğrayacak parametre sayısı $0.04 \times 24 \times 2 = 2$ olarak hesaplanır.

4. Gezgin Satıcı Probleminin Genetik Algoritmalarla Uygulaması

GA”ların uygulanmasına ilişkin incelenecek olan GSP”de 8 şehir olduğunu varsayalım. Hatırlatalım ki problemde amaç, satıcının her şehre yalnızca bir kere uğrayarak turunu en kısa yoldan tamamlamasıdır. Satıcının başladığı şehre geri dönme zorunluluğunun olmadığını varsayalım. (Yani burada amaç döngü değil, Hamilton yolunun bulunmasıdır.) şehirler birbirine bağlı olmakta ve uzaklıklar verilmektedir bu şartlar içerisinde toplam alınan yolun minimum değerini bulmaya çalışacağız.



Şekil 23:Gezgin Satıcı Örneği

Başlangıç popülasyonun oluşturulması: GSP’de 8 şehir olduğunu kabul etmiştik. şehirleri “0” dan “7” ye kadar numaralandırdığımızı düşünürsek, her gen bir şehre karşılık gelecektir. Bu durumda kromozomlarımız ve başlangıç popülasyonumuz rastlantısal olarak aşağıdaki gibi oluşturulabilir. (Popülasyon büyüklüğü “5” olarak seçilmiştir.)

10275463

71302564

24170536

41653720

60534217

Popülasyon içindeki her kromozomun amaç fonksiyonunun değerinin hesaplanması:

İlk kromozomumuza bakarsak, bu kromozomun amaç fonksiyonu değeri, Çartta verilen şehirlerarası yol uzunluklarına bağlı olarak aşağıdaki şekilde hesaplanır:

10275463

1”den 0”a 5 km

2”den 7”ye km

5”den 4”e 3 km

6”dan 3”e 5 km

0”dan 2”ye 3 km

7"den 5"e 4 km

4"den 6"ya 6 km

İse; bu kromozomun gösterdiği yolun toplam uzunluğu 27 km, yani amaç fonksiyonun değeri "27" olacaktır. Artık popülasyonumuzun GA operatörlerinin uygulanması için hazırdır.

Bu operatörleri uygulayarak popülasyonumuzun her yeni nesilde evrilmesini ve daha iyi sonuçlar elde edilmesini sağlayabiliriz.

Tekrar üretme, çaprazlama ve mutasyon operatörlerinin uygulanması: Goldberg probleminde uygulanan çaprazlama işlemini GSP'nin kromozomlarına uygularsak bazı güçlüklerle karşılaşabiliriz. Örneğin, rastgele bir şekilde üçüncü ve dördüncü kromozomları seçelim ve çaprazlama noktası 5 olsun. Bu durumda çaprazlama şu şekilde olacaktır.

Çaprazlama öncesi durum

Çaprazlama sonrası durum

2 4 1 7 0 **5 3 6**

2 4 1 7 0 **7 2 0**

4 1 6 5 3 **7 2 0**

4 1 6 5 3 **5 3 6**

Görüldüğü gibi yeni oluşan kromozomlar, GSP problemi için olumlu bir sonuç vermez. Çünkü satıcımız, tüm şehirleri gezmeli ve her şehre yalnızca bir kez uğramalıydı, fakat çaprazlama sonrası kromozomlarda bazı şehirlere hiç gidilmemekte yada şehirlere uygun gelen düğümlere iki kere başvurulmaktadır. Örneğin çaprazlama sonrası oluşan ilk kromozomumuza dikkat edecek olursak, 3., 5. ve 6. şehirlere hiç gidilmediğini ve 0., 2. ve 7. şehirlere iki defa uğrandığını gözlemledik. Böyle bir olumsuz durumda ikinci bir işleme gerek duyulur. Yani kısıtlamaları sağlamayan uygunsuz kromozomların standartlaştırılması gerekmektedir. Standartlaştırma için aşağıdaki basit kural uygulanabilir: "Kromozom içinde tekrar eden ilk şehir ziyaret edilmeyen en küçük numaralı şehir ile değiştirilsin". Bu standartlaştırma kuralı akla ilk gelen kurallardan biridir. Herhangi başka bir kural da standartlaştırma amacı için uygulanabilirdi. Evrimsel programlama böyledir; hemen hemen her şey plansız programsızdır. Tıpkı doğadaki evrim gibi...

Sorunlu kromozomların standartlaştırılması ile şu sonucu elde ederiz: Çaprazlama ve standartlaştırma sonrası durum;

34156720

41027536

Böyle bir durumda ikili düzende kodlanmış olan kromozomların daha avantajlı olabileceği görülmektedir. Ancak ikili düzende kodlama da kromozomların çok uzun olması, ya da problemin yapısına uygun olmaması gibi sorunlar çıkabilmektedir. Şimdi de Goldberg problemiindeki gibi mutasyon operatörüne geçelim. Ancak bu rasgele seçilen kromozomlar üzerindeki iki genin yerlerini değiştirmekle gerçekleştirilir. Rastgele olarak seçtiğimiz 5. kromozomumuzun ikinci ve yedinci genlerini seçtiğimizi varsayalım. Bu durumda; mutasyondan önceki durum mutasyondan sonraki durum

60534217 67534210

4. ve 5. Yeni kromozomların amaç fonksiyonu değerlerinin bulunması ve kötü kromozomların popülasyona çıkartılması:

Yeni oluşturulan fonksiyonların değerleri 2. adımda yapılanlara benzer olarak hesaplanır. Seçici bir model söz konusu ise, bu adımdan sonra popülasyonun içindeki en kötü değerli kromozom, popülasyondan çıkartılır.

(2-3) arasındaki adımların tekrarı.

Konunun başında optimizasyon problemlerinde, optimuma her zaman ulaşamadığını ya da tüm olası sonuçların incelenmesini, bilgisayarlarla dahi mümkün olmadığını söylemiştik. Burada anlatmaya çalıştığımız Genetik Algoritmalar, Evrim Kuramını temel olarak, bu tür problemler için optimum sonuçları ya da optimumlara yakın sonuçları elde etmek için kullanılmaktadır. Örneğin bu yöntemle 200–300 şehir için bile yaklaşık 20 dk içinde optimuma çok yakın sonuçlar elde edilebilmektedir.

5.Java'da Basit Bir Genetik Algoritma Kodlaması

x^2 nin maksimum değerini simple genetik algoritma kullanarak Java'da kodlayalım.

Genetik algoritma; diğer optimizasyon yöntemlerinde olduğu gibi çözüm için tek bir yapının yerine, böyle yapılardan meydana gelen bir küme oluştururlar. Problem için olası pek çok çözümü temsil eden bu kümeye population (nüfus) deriz. Nüfuslar birey adı verilen dizilerden oluşur.Nüfus içindeki her bireyin problem için çözüm olup olmayacağına karar veren bir uygunluk fonksiyonu (fitness function) vardır.

Uygunluk fonksiyonu yüksek olan bireylere, populasyondaki diğer bireyler ile çoğalmaları için fırsat verilir. Bu bireyler çaprazlama (crossover) işlemi sonunda çocuk adı verilen yeni bireyler üretirler.

Çözüm adımlarımızı sıralarsak;

- 1) Initialization
- 2) Evaluation
- 3) Selection
- 4) Crossover
- 5) Mutation
- 6) Repeat

6. GENETİK ALGORİTMA TABANLI PID KONTROLÖR

SİMÜLATÖRÜ TASARIMI

Özet:

Sistem denetimi alanında PID (oransal-integral-türevsel) kontrolörler önemli yer tutmaktadır. Sistemin zaman veya frekans bölgesi verilerinden faydalanılarak; sistem özelliklerine uygun, belirli ölçütler altında, bu tür kontrolörlerin tasarımı için Ziegler-Nichols, Cohen-Coon, Chien Hrones Reswick(CHR), Wang-Juang-Chan gibi değişik yöntemler mevcuttur.

Bilgisayar ve yapay zekâ alanındaki gelişmelere paralel olarak genetik algoritmaların da kontrol sistemlerindeki uygulamaları artmaktadır. Bu çalışmada, PID kontrolörlerin tasarımları hem klasik yöntemlerle hem de genetik algoritmalarla gerçekleştirilerek karşılaştırmalı sonuçlar analiz edilmiştir. Bu doğrultuda, eğitim amaçlı da kullanılabilecek grafiksel ara yüz programı tasarlanmıştır. Program ile transfer fonksiyonları girilen sistemler için uygun P, PI ve PID kontrolör katsayıları hem seçilen klasik yöntemlerle hem de genetik algoritmalarla hesaplanmakta; sisteme ait birçok parametre ve cevap eğrileri hem sayısal hem de grafiksel olarak karşılaştırmalı sunulabilmektedir

GİRİŞ

Sistemlerin, istenilen özelliklerde çalışmasını sağlayan kontrolör/denetleyici devrelerinin analizi ve tasarımı, kontrol teorisinin başlıca alanlarından biridir. Bu alanda, farklı yapı ve karakteristiklerde kontrolörler geliştirilmiştir. Bunlardan birisi de endüstriyel amaçlı en çok kullanılan temel PID (oransal-integral-türevsel) kontrolör yapısıdır. Girişin oran-integral-türev değerlerini ve/veya bunların kombinasyonlarını kullanarak sistemlerin denetlenmesini sağlayan bu tip kontrolörler genel haliyle P, PI, PD ve PID olarak kullanılmaktadır. PID kontrolörlerin parametreleri hem deneysel hem de analitik olarak elde edilebilmektedir. Ayrıca tasarım aşamaları fazla işlemler içermemektedir. Bu nedenle endüstriyel uygulamalarda yaygın olarak kullanılmaktadırlar.

PID kontrolörlerin tasarımları için değişik yöntemler mevcuttur. Ziegler-Nichols, Cohen-Coon, Chien Hrones Reswick, Wang-Juang-Chan yöntemleri en çok bilinenleridir. Bu yöntemlerde; sistemin zaman veya frekans bölgesi

cevaplarından faydalanılarak, farklı ölçütlere göre uygun kontrolör katsayıları belirlenebilmektedir.

Teknolojideki gelişmelere paralel olarak bilgisayar destekli analiz ve tasarımlar, kontrol sistemlerinde de yoğun şekilde kullanılmaktadır. Bu alanda yapay zekâ tekniklerinden de yüksek oranda faydalanılmaktadır. Kullanılan yapay zekâ tekniklerinden birisi de genetik algoritmalarıdır.

Bu çalışmada; eğitim amaçlı da kullanılabilecek bir PID simülatör programı geliştirilmiştir. Program ile kullanıcı/öğrencinin tanımladığı geri beslemeli sistemler için, seçilen klasik tasarım yöntemleri ve genetik algoritmalar ile istenilen kontrolörler tasarlanabilmektedir.

Tasarlanan kontrolörlere göre sistem parametreleri, zaman domenii cevap eğrileri veya frekans domenii diyagramları da karşılaştırmalı olarak kullanıcıya sunulmaktadır. Böylece kullanıcılar/öğrenciler parametrelerin etkilerini inceleyebilmekte, performans analizlerini yapabilmekte ve kullanım alanlarına (amaçlarına) uygun PID kontrolör tasarımlarını gerçekleyebilmektedirler.

6.1.PID KONTROLÖRLER VE TASARIM YÖNTEMLERİ

6.1.1 PID Kontrolörler

Endüstriyel alanda çok kullanılan PID kontrolörün genel yapısı Şekil 24'de verilmektedir. PID kontrolörün girişindeki hata (referans ile geribesleme arasındaki fark) işareti $e(t)$ ve çıkışındaki kontrol işareti $d(t)$ ise sürekli ve ayrık zaman domenindeki matematiksel ifadeleri

$$d(t) = k_p \left(e(t) + \frac{1}{T_i} \int_0^t e(T).dt + T_d \frac{d}{dt} e(t) \right) \quad (1)$$

$$d[n] = K_c \left(e[n] + \frac{T_s}{T_i} \int_0^n e[i] + \frac{T_d}{T_s} (e[n] - e[n-1]) \right) \quad (2)$$

şeklindedir. (1-2) eşitliklerinde $\int$ integral, $\frac{d}{dt}$ türev ve Δ örnekleme zamanıdır. Zaman domeninde

$$d(t) = K_p \cdot e(t) + K_1 \int_0^t e(T).dr + K_d \frac{d}{dt} e(t) \quad (3)$$

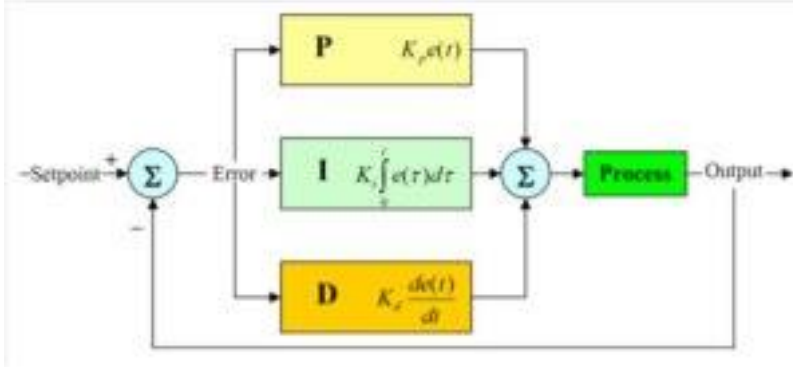
ile ifade edilen PID kontrolörün s- domenindeki transfer fonksiyonu

$$G_c(s) = K_p + \frac{K_1}{s} + K_d s$$

Veya (1) ile verilen eşitliğe göre

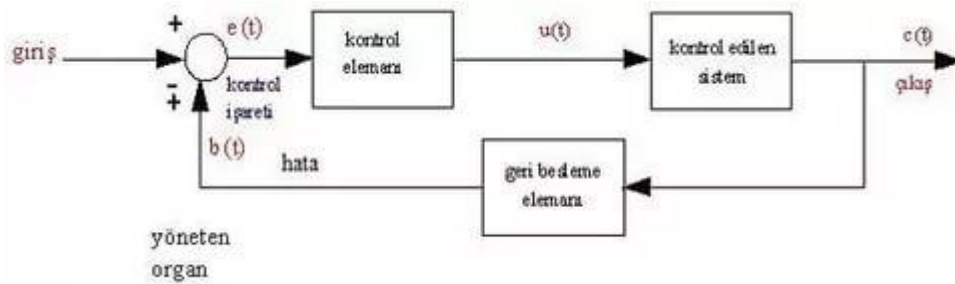
$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d \cdot s \right)$$

Olarak elde edilmektedir.



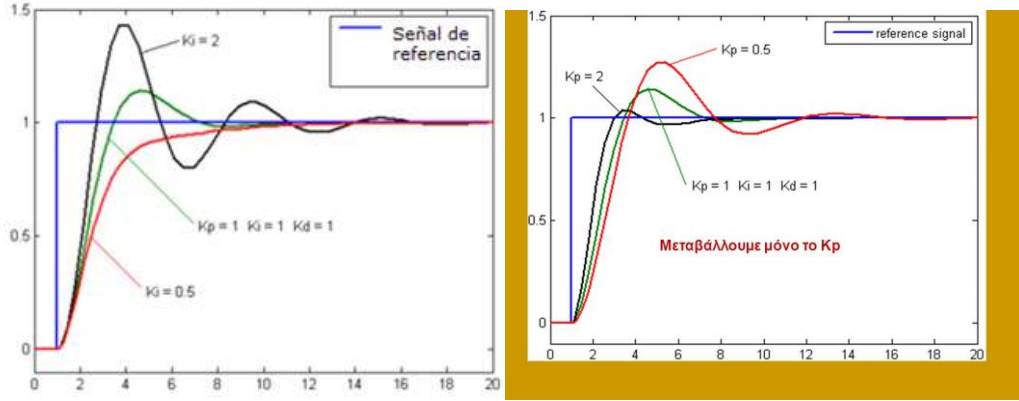
Şekil 24:Geri beslemeli kontrol sistemi ve PID kontrolör

Herhangi bir kontrol sisteminin genel olarak amacı; denetlediği sistemi, kısa sürede verilen referans değerinde çalıştırmak (referans değere getirmek), yani hemen cevap vererek kararlı halde hatasız çalışmasını sağlamaktır. Bu nedenle kontrolörlerin tasarımı önemli yer tutmaktadır. Şekil 25'deki örnek sistem üzerinde PID kontrolör katsayılarının etkileri, Tablo 1'de verilmektedir. Tablo 9'deki birim basamak cevapları incelendiğinde oran, integral ve türev parametrelerinin (kontrolörlerinin) etkileri Tablo 10'deki gibi özetlenebilmektedir



Şekil 25:Negatif geribeslemeli PID kontrolörlü örnek sistem

Çizelge 9: Örnek sistemin değişik PID parametrelerindeki birim basamak cevapları



Çizelge 10: PID kontrolör parametrelerinin sistem cevabına olan etkileri

	Oransal (P) (K_p)	İntegral (I) (K_i)	Türevsel (D) (K_D)
Yükselme zamanı	azalır	Azalır (az)	Az etki
Aşım değeri	artar	artar	Azalır
Yerleşme zamanı	Artar(az)	artar	azalır
Kararlı hal hatası	azalır	Azalır (çok)	Az etki

6.2. PID Kontrolör Tasarım Yöntemleri

PID kontrolör parametrelerinin belirlenmesi/hesaplanması için Ziegler-Nichols, CohenCoon, Chien Hrones Reswik (CHR), Wang-Juang-Chan gibi değişik yöntemler mevcuttur.

6.2.1. Ziegler-Nichols yöntemi

PID kontrolör tasarımında en çok kullanılan yöntemlerden birisi Ziegler-Nichols yöntemidir. 1942 yıllarında J.G. Ziegler ve N.B. Nichols tarafından önerilen bu yöntemde PID katsayıları basamak veya frekans cevabı verilerinden hesaplanabilmektedir (Ziegler ve Nichols, 1942). PID katsayılarının Nyquist frekans cevabından elde edilmesi Ziegler-Nichols osilasyon yöntemi ve birim basamak cevabından elde edilmesi de Ziegler-Nichols reaksiyon eğrisi yöntemi olarak da adlandırılmaktadır.

6.2.2. Cohen-Coon yöntemi

Ziegler-Nichols reaksiyon eğrisi yöntemindeki gibi; sistemin basamak cevabından faydalanılarak PID kontrolör katsayıları, Çizelgedeki eşitlikler kullanılarak hesaplanabilmektedir

6.2.3. Wang-Juang-Chan yöntemi

Daha önceden bahsedilen yöntemlerde olduğu gibi ϕ , λ ve ω parametrelerinin bilinmesi durumunda Wang-Juang-Chan yönteminde PID katsayıları Tablodaki eşitliklerle hesaplanmaktadır

6.3 GENETİK ALGORİTMALAR

Evrimsel hesaplamanın parçası olan genetik algoritmalar, ilk defa 1975 yıllarında John Holland tarafından geliştirilmiş ve kullanılmıştır (Holland, 1975). Bu algoritmaların temeli, doğadaki evrim/biyolojik sürece (güçlü/iyi olan nesillerin, yeni şartlara uyum sağlayarak yaşamlarını sürdürmeleri; zayıf/kötü olanların hayatta kalamamaları) dayanmaktadır. Genetik algoritmalar, ilgili sürecin modellenmesiyle ortaya çıkan iterasyonlu ve olasılıklı bir çözüm yöntemidir.

Çözüm için rastlantısal arama tekniklerini kullanan ve parametre kodlama esasına dayanan sezgisel bir yöntem olan genetik algoritmalar; birçok alandaki optimizasyon problemlerinin çözümünde, makine öğrenmesinde, bilgi sistemlerinde vb. kullanılmaktadır.

Çizelge 11: Wang-Juang-Chan yöntemiyle PID katsayılarının elde edilmesi

	K_p	T_i	T_d
PID	$\left(0.7303 + \frac{0.5307T}{L}\right)(0.5L+T)$	$0.5L+T$	$\frac{0.5LT}{0.5L + T}$
	$K(L+T)$		

Genetik algoritmalar ile çözüm aşamaları, aşağıdaki gibi özetlenebilir.

i. Başlangıç popülasyonunu oluşturma: İlgili problemin olabilecek çözümlerini gösteren ve genelde rastgele olan bireylerden (kromozomlardan) bir başlangıç popülasyonu (nesil) oluşturulur.

ii. Uygunluk değerlerini hesaplama: Nesildeki her bir bireyin uygunluk değerleri hesaplanır.

iii. Durdurma ölçütü sınaması:

- Sağlanmıyorsa yeni nesil oluşturma ve ikinci adıma dönme:
- Uygunluk değerlerine göre seçme: Eski nesildeki her bir bireyin hesaplanan uygunluk değerlerine göre seçme işlemi yapılır.
- Seçilen bireyleri uygunlaştırma: Seçilen bireyler, genetik işlemlerle (çaprazlama, mutasyon vb.) uygunlaştırılır.
- Sağlanıyorsa en uygun çözümü seçme: Popülasyonda hesaplanan uygunluk değerlerine göre en iyi birey (kromozom, çözüm) seçilir.
- Özetle; genetik çevrim, en iyi uygunluk değerine sahip birey bulununcaya kadar devam eder.

7. PID SİMÜLATÖR

Bu çalışmada; kullanıcı tarafından tanımlanan geribeslemeli sistemler için otomatik olarak uygun PID kontrolör tasarımı gerçekleştiren yazılım geliştirilmiştir. MATLAB (Mathworks, 2007) ortamında tasarlanan ve eğitim amaçlı kullanılabilecek olan etkileşimli PID simülatör programıyla uygun P, PI ve PID kontrolör katsayıları klasik yöntemler (Ziegler-Nichols, CohenCoon, Chien-Hrones-Reswick ve Wang-Juang-Chan) ve genetik algoritmalarla elde edilmektedir.

Programın;

- “Dosya” menüsüyle Tanımlanan sistemler için yeni tasarımlar gerçekleştirilebilmekte, sonuçlar kaydedilebilmekte veya yazdırılabilmektedir.
- “Yöntem” menüsüyle Kontrolör tasarım yöntemi seçilebilmektedir.
- “Ayarlar” menüsüyle Tasarımda kullanılacak genetik algoritmaların özellikleri/ölçütleri belirlenmektedir.
- “Yardım” menüsüyle Program hakkında kullanım ve yardım bilgileri edinilebilmektedir.

Giriş ekranından, bu ekranda denetlenecek olan sistemin ve geribesleme yolunun transfer fonksiyonu s- domeninde girilmekte ve aynı zamanda geribesleme türünün pozitif veya negatif oluşu da girilen “+” veya “-” sembolleriyle belirtilmektedir. Bu veriler doğrultusunda “Tanımla” butonuyla sistem tanımlanarak P, PI ve PID onay kutucuklarıyla istenilen/tasarlanacak kontrolör türü seçilmektedir. “Yöntem” ve “Ayarlar” menülerinden ilgili klasik tasarım yöntemi belirlenip genetik algoritma seçenekleri düzenlenerek “Tasarla” butonu ile tasarım işlemi gerçekleştirilmektedir. Geliştirilen program; hem seçilen klasik yöntemle hem de genetik algoritma ile elde edilen kontrolör katsayılarını listelenmektedir. Ayrıca tüm sisteme ait birçok önemli parametre (sistemin sıfırları – kutupları – sönüm faktörleri – doğal frekansları ve DC kazançları) karşılaştırmalı olarak gösterilmektedir. Ayrıca “Sistem cevapları” menüsünden seçim yapılarak kontrolörsüz, klasik yöntemle tasarlanan kontrolörlü, genetik algoritmalarla tasarlanan kontrolörlü ve her üçünün de karşılaştırmalı birim basamak, birim dürtü (impulse), kullanıcı tarafından tanımlanan özel işaret cevapları; Bode, Nyquist, Nichols diyagramlarıyla köklerin yer eğrileri çizdirilebilmektedir. Böylece cevap eğrileri üzerinde kullanıcı/öğrenci zaman domeninde yükselme ve yerleşme zamanlarını, aşım, kararlı durum gibi değerleri karşılaştırmalı olarak görebilmektedir. Aynı zamanda diyagramlarla da frekans domeni özelliklerini inceleyebilmektedir.

7.1.PID SONUÇLARI

Gerçekleştirilen çalışmada; PID kontrolör tasarımları ve eğitimlerinde kullanılabilecek bir simülatör programı geliştirilmiştir. Program ile geribeslemeli sistemler için P, PI ve PID kontrolörler klasik (Ziegler-Nichols, Cohen-Coon, Chien Hrones Reswick, Wang-Juang-Chan) yöntemler ve genetik algoritmalar ile tasarlanabilmektedir. Tasarım sonucu ilgili kontrolör katsayıları ve sisteme ait birçok parametre karşılaştırmalı olarak elde edilebilmektedir. Ayrıca tasarlanan kontrolörlere göre sisteme ait zaman ve frekans bölgesi cevapları/eğrileri yine karşılaştırılmalı olarak kullanıcıya sunulmaktadır. Aynı zamanda parametre değişiklikleri yapılarak sisteme etkileri incelenebilmektedir. Böylece istenilen sistemler için PID türü kontrolörlerin tasarımı etkin bir şekilde, hızlı ve kolaylıkla gerçekleştirilebilmektedir.

8.ÖRNEKLER

• $x \in [0,15]$ olmak üzere $f(x)=x^2$ fonksiyonun verilen aralıkta maksimizasyonu yapılması istenmektedir.

– İkili Kodlama

– Toplumun birey sayısı $n=4$

• Birey 1: 1101, $x = 13$, $x^2 = 169$

• Birey 2: 0100, $x = 4$, $x^2 = 16$

• Birey 3: 1011, $x = 11$, $x^2 = 121$

• Birey 4: 1000, $x = 8$, $x^2 = 64$

ÖRNEK II

– Rulet çemberi seçimi

• Birey 1: $169/370 = \%46$

• Birey 2: $16/370 = \%4$

• Birey 3: $121/370 = \%33$

• Birey 4: $64/370 = \%17$

– Eşleşme

• Aday 1: 1101 (Birey 1)

• Aday 2: 0100 (Birey 2)

• Aday 3: 0100 (Birey 2)

• Aday 4: 1000 (Birey 4)

ÖRNEK III

– Yeni toplum

• Birey 1: 1100, $x = 12$, $x^2 = 144$

• Birey 2: 0101, $x = 5$, $x^2 = 25$

• Birey 3: 0010, $x = 2$, $x^2 = 4$

- Birey 4: 1100, $x = 12$, $x^2 = 144$

Tek bir iterasyon yapılmış ve başlangıç toplumundan

bir sonraki kuşak oluşturulmuştur.

ÖRNEK IV

Bir Açıklayıcı Örnek – GA Nasıl Çalışır

X parametresinin 0 ve 15 arasında değişken olduğu ($15x - x^2$) fonksiyonun maksimum değerini bulalım. Kolaylık için biz x 'i sadece integer değerlerden alacağız. Böylece, kromozomlar dört bit" le ifade edilebilir.

Çizelge 11:Kromozomların bitle ifade edilişi

Integer	Binary code	Integer	Binary code	Integer	Binary code
1	0001	6	0110	11	1011
2	0010	7	0111	12	1100
3	0011	8	1000	13	1101
4	0100	9	1001	14	1110
5	0101	10	1010	15	1111

Farz edelim ki; kromozom popülasyon boyutu 6, çaprazlama olasılığı 0,7 ve mutasyon olasılığı 0.001 dir.(seçilen bu olasılıklar GAlarda yeterince tipiktir.)

Uygunluk fonksiyonu şöyle tanımlanır,

$$f(x) = 15x - x^2$$

GA rasgele bir şekilde üretilen birler ve sıfırlarla altı adet 4-bit string ile doldurularak kromozomların başlangıç popülasyonunu oluşturur. Başlangıç popülasyonu Tablo 11 de gösterilen gibi görünebilir. (Gerçek pratik bir problem, tipik olarak binlerce kromozomlu bir popülasyona sahip olacaktır.)

Sonraki adım her bir birey kromozomun uygunluğunu hesaplamaktır.Bu durumda başlangıç popülasyonunun ortalama uygunluğu 36 dır. Bunu geliştirmek için,

başlangıç popülasyonu genetik operatörler kullanılarak değiştirilir: Seçim,
Çaprazlama ve Mutasyon

Çizelge 12: Rasgele bir şekilde üretilen başlangıç kromozomlarının popülasyonu

Chromosome label	Chromosome string	Decoded integer	Chromosome fitness	Fitness Ratio, %	Cumulative ratio
x_1	1100	12	36	16.5%	16.5
x_2	0100	4	44	20.2%	36.7
x_3	0001	1	14	6.4%	43.1
x_4	1110	14	14	6.4%	49.5
x_5	0111	7	56	25.7%	75.2
x_6	1001	9	54	24.8%	100.0

Doğal seçimde, sadece en uygun türler hayatta kalabilir, üreyebilir ve sonraki nesle onların genleri ile geçilebilir. GA lar benzer bir yaklaşım kullanır. Fakat doğadan farklı bir şekilde, kromozom popülasyonunun boyutu bir nesilden sonrakine değiştirilmemiş kalır.

Tablo 12 deki son kolon, popülasyonun toplam uygunluğuna bireysel kromozomların uygunluk oranını gösterir. Bu oran, eşleşme için kromozomların seçilme şansını tanımlar. Böylece, x_3 ve x_4 kromozomlarının çok düşük bir seçilme olasılığı varken, x_5 ve x_6 kromozomlarının yeterli şansı vardır.

Seçim

Eşleşme için bir kromozom seçminde, rulet tekerleği seçimi tekniğini kullanarak, $[0,100]$ aralığında rasgele bir sayı üretilir ve kümülatif oranının rasgele sayıyı içerdiği kromozom seçilir. Bu, uygunluklarıyla orantılı olarak tekerlek üzerinde bir alana sahip her bir kromozomun yer aldığı bir tekerleğin dönmesi gibidir. Örneğimizde, biz

altı kromozomlu bir başlangıç popülasyonuna sahibiz. Böylece, sonraki nesilde aynı popülasyon boyutunu kurmak için, altı rasgele sayı üretilecektir.(bu, rulet tekerleğinin altı kez dönmesi gibidir.) İlk iki dönme, x_6 ve x_2 ebeveyn olmak için seçebilir; ikinci dönme çifti, x_1 ve x_5 kromozomlarını seçebilir ve son iki dönüş x_2 ve x_5 kromozomlarını seçebilir.

Çaprazlama

Bir çift ebeveyn kromozom seçildikten sonra, çaprazlama operatörü uygulanır. Öncelikle çaprazlama operatörü rasgele bir şekilde, iki ebeveyn kromozomu kıracak ve bu noktadan sonra kromozomların parçalarını değiştirecek bir çaprazlama noktası seçer. Sonuç olarak iki yeni yavru oluşturulur.

Mesela, iki kromozom x_6 ve x_2 ikinci genden sonra çaprazlanabilir. Her biri iki yavru üretir. Aşağıdaki gibi gösterilir:

1	0	0	1
---	---	---	---

x_6

x'_6

1	0	0	0
---	---	---	---

0	1	0	0
---	---	---	---

x_2

Ebeveynler

0	1	0	1
---	---	---	---

Çocuklar

Eğer bir çift kromozom çaprazlanmazsa, yavrular bireylerin tam kopyası olarak oluşturulur. 0.7 çaprazlama olasılığı genellikle iyi sonuç üretir.

Mutasyon

Doğada nadiren olan Mutasyon, gendeki bir değişikliği temsil eder. Bu, uygunlukta önemli bir gelişmeye neden olabilir. Fakat çoğunlukla, oldukça faydalı sonuçları vardır. Mutasyonun rolü, arama algoritmasının bir local optimuma takılmamasının garantisini sağlamaktır. Seçim sırası ve çaprazlama operatörleri herhangi bir homojen çözüm kümesinde durgunlaşabilir. Böyle şartlar altında, tüm kromozomlar özdeşleşir ve bu yüzden popülasyonun ortalama uygunluğu geliştirilemeyebilir. Çözüm sadece optimal (veya local olarak oldukça optimal) olmak görünebilir. Çünkü arama algoritması daha fazla ilerlemeyebilir. Mutasyon rasgele bir aramaya eşdeğerdir ve genetik farklılıkların kaybının korunmasında bize yardım eder. Mutasyon operatörü bir kromozomda rasgele bir şekilde seçilen geni çevirir. Mesela, aşağıda gösterildiği gibi, x_1 ' ikinci geninde ve kromozom x_2 de üçüncü geninde mutasyona uğratılmış olabilir.

1	1	1	1
---	---	---	---

 x_1'

1	1	0	1
---	---	---	---

0	1	0	0
---	---	---	---

 x_2
Parents

0	1	0	0
---	---	---	---

Children

Mutasyon, bazı olasılıklı kromozomda, herhangi bir gende meydana gelebilir. Mutasyon olasılığı doğada oldukça küçüktür ve GALarda oldukça küçük tutulur. Tipik olarak 0.001 ile 0.01 aralığında. Genetik algoritmalar, popülasyonun ortalama uygunluğunun sürekli geliştirilmesini,

temin eder ve bir nesil sayısından sonra (tipik olarak birkaç yüz) popülasyon bir

yakın-optimal çözümü geliştirir. Örneğimizde son çözüm

1	0	0	0
---	---	---	---

sadece ve kromozomlarından oluşur.

0	1	1	1
---	---	---	---

$$\text{Bu } f(x) = 15x - x^2 = 56 \text{ dır.}$$

Bu örnekte, problem sadece bir değişkene sahiptir. Çözümünü göstermek kolaydır.

9. SONUÇLAR

Evrimin modellenmesiyle birlikte Genetik algoritmaların ele alınmaları çok yakın tarihe dayanmaktadır. Son zamanlarda bu konu ile ilgili çalışmalar oldukça artmaktadır. Bilgisayar teknolojisindeki gelişmelere paralel olarak, GA'ların pratikte uygulanabilirliği artmıştır. Her geçen gün GA'nın adına yeni sempozyumlar düzenlenirken yeni uygulama alanlarına tatbik edildiği de görülmektedir. GA'nın geliştirilmesi alanında; hibridizasyon, parametrelerin optimal seçimi ve adaptasyonu gibi konularda araştırmalar devam etmektedir(Wurtz 1997). İlk bölümde de anlatıldığı gibi klasik metotlar bazı sınırlamalara sahiptir. Bu metotlardaki ciddi sınırlamalardan biri optimizasyon algoritmasının seri çalışmasıdır. GA'da ise PC'leri paralel çalıştırılarak hızlı sonuçlar elde edilebilir. Birkaç tane algoritmanın aynı anda çalıştırılması ve popülasyonları arasında göç olabilmesi paralel işlemcilerle yapılır. Buna ek olarak popülasyondaki fertlerin maliyet değerleri aynı anda hesaplanarak GA'nın yakınsaması hızlandırılır. GA karar mekanizmasında kullanılabilir mi? sorusu sorulduğunda sübjektif maliyet fonksiyonlu GA akla gelmektedir. Bu algoritma market stoklarının değerlendirilmesi, savaşta askeri stratejilerin geliştirilmesi gibi karar organlarında kullanılabilir.

GA geçici bir heves veya bir moda olmayıp uzun süre gündemde kalacak bir yöntemdir. Tabiatın hesaplama algoritmaları için iyi bir model olduğu bilinmektedir.Bütün bunlardan sonra tabiatta iyilikler ve güzellikler azalmaktadır ve belki belli birmüddet sonra tabiatta global optimuma ulaşacaktır.

KAYNAKÇA

- www.horozerk.com (Bahar 2006)
- <http://compe.hku.edu.tr>
- www.biyologlar.com
- <http://www.slideshare.net> (İsmail Akbudak)
- <http://docslite.org> (Uludağ Üniversitesi)
- www.bulentsiyah.com
- www.ab.org.tr (Muğla Sıtkı Koçman Üniversitesi)
- www.ahmetcevahircinar.com.tr