

YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ

GENETİK ALGORİTMA İLE HAT DENGEME

Endüstri Mühendisi Abit BALIN

FBE Endüstri Mühendisliği Anabilim Dalı Endüstri Mühendisliği Programında Hazırlanan

YÜKSEK LİSANS TEZİ

Tez Danışmanı : Yrd. Doç. Dr. Hayri BARAÇLI

İSTANBUL, 2010

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ	v
KISALTIMA LİSTESİ	vi
ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	viii
ÖNSÖZ	ix
ÖZET	x
ABSTRACT	xi
1. GİRİŞ	1
2. GENETİK ALGORİTMA	3
2.1 Evrimsel Hesaplama	3
2.2 Genetik Algoritmanın Tarihçesi	3
2.3 Genetik Algoritmaların Temel Kuramı	5
2.4 Genetik Algoritmanın Çalışma Prensipleri	6
2.5 Genetik Algoritmanın Temel Kavramları	9
2.5.1 Gen	9
2.5.2 Kodlama	9
2.5.3 Kromozom	9
2.5.4 Birey	9
2.5.5 Popülasyon ve Jenerasyon	10
2.5.6 Uygunluk Fonksiyonu	11
2.5.7 Bireyler	11
2.5.8 Diversity (Farklılık/Saçkınlık)	12
2.5.9 Uygunluk Değeri ve En İyi Uygunluk Değeri	12
2.5.10 Ebeveyn ve Çocuk	13
2.6 Genetik Algoritma Kullanım Sebepleri	13
2.7 Genetik Algoritmanın Diğer Yöntemlerden Farkı	13
2.8 Genetik Algoritmada Kullanılan Teknikler	14
2.8.1 Kromozomların Şifrelenmesi	14
2.8.1.1 İkili Kodlama	14
2.8.1.2 Permütasyon Kodlama	15
2.8.1.3 Değer Kodlama	15
2.8.1.4 Ağaç Kodlama	15
2.8.2 Hedef fonksiyonunun belirlenmesi	17
2.8.3 Başlangıç Toplamı	18
2.8.4 Çaprazlama (Takas)	20
2.8.4.1 Tek Noktalı Çaprazlama	22
2.8.4.2 İki Noktalı Çaprazlama	23
2.8.4.3 Çok Noktalı Çaprazlama	25
2.8.4.4 Tek Biçimli-Üniform Çaprazlama	25

2.8.4.5	Aritmetik Çaprazlama.....	27
2.8.4.6	Tersleme	27
2.8.4.7	Karıştırmalı Çaprazlama	27
2.8.4.8	PMX Çaprazlama.....	28
2.8.4.9	OX Çaprazlama	28
2.8.4.10	CX Çaprazlama.....	29
2.8.4.11	LOX Çaprazlama	30
2.8.4.12	Ağaç Çaprazlama.....	30
2.8.4.13	Ara Birleşmeli Çaprazlama.....	31
2.8.4.14	Doğrusal Birleşmeli Çaprazlama.....	32
2.8.5	Mutasyon	33
2.8.6	Uygunluk Fonksiyonu	37
2.9	Genetik Algoritmanın Çalışma Şekli.....	38
2.9.1	İlk Popülasyonun Oluşturulması	38
2.9.2	Yeni Jenerasyonun Oluşturulması	39
2.9.3	Çaprazlama	40
2.9.4	Mutasyon	41
2.9.5	Algoritma İçin Durdurma Kriteri.....	41
2.10	Genetik Algoritmelerde Parametre Seçimi	42
2.10.1	Popülasyon Büyüklüğü	43
2.10.2	Çoğalma Parametresi	43
2.10.3	Çaprazlama Olasılığı	44
2.10.4	Mutasyon Olasılığı.....	45
2.10.5	Kuşak Aralığı.....	45
2.10.6	Seçim Stratejisi	46
2.10.7	Fonksiyon Ölçeklemesi	46
2.10.8	Genetik Algoritmada Parametre Kontrolü.....	47
2.11	Genetik Algoritma Uygulama Alanları	49
2.11.1	Eniyileme “Optimizasyon”	49
2.11.2	Otomatik Programlama ve Bilgi Sistemleri.....	50
2.11.3	Mekanik Öğrenme	50
2.11.4	Finans ve Pazarlama	50
2.11.5	Rotalama Problemleri	51
2.11.6	Gezgin Satıcı Problemi	52
2.11.7	Çizelgeleme Problemleri	52
2.11.8	Hücrel Üretim Problemleri.....	53
2.11.9	Taşıma Problemleri.....	53
2.11.10	Tesis Yerleşimi Problemi	53
2.11.11	Montaj Hattı Dengeleme Problemi.....	54
3.	MATLAB ve GENETİK ALGORİTMA	55
3.1	MATLAB Genetik algoritma Tools	60
3.2	Rastrigin’s Function.....	61
3.3	Genetik Algoritma İle Grafik Çizimi.....	63
3.4	MATLAB Programının Genetik Algoritma Tool’unun Çalışma Şekli	64
3.4.1	İlk Popülasyonun Oluşturulması	64
3.4.2	Çaprazlama	65
3.4.3	Mutasyon	66

3.4.4	Seçim	67
4.	UYGULAMA	68
4.1	Literatür Çalışmaları	68
4.2	Matematiksel Model	70
4.3	Optimizasyon İçin Kullanılan İki Aşamalı Yöntem	71
4.3.1	Yöntemin İlk Aşaması	71
4.4	Yöntemin İkinci Aşaması	79
4.4.1	Genetik Algoritma	80
4.4.2	GA'nın Çalışma Prensipleri	80
4.4.3	İkinci Aşama Optimizasyon İçin Problemin Formüle Edilmesi	81
4.4.4	Genetik Algoritma Sonuçları	82
5.	SONUÇLAR ve DEĞERLENDİRME	87
	KAYNAKLAR	89
	ÖZGEÇMİŞ	91

SİMGE LİSTESİ

f_k : herhangi bir k rotasında otobüs hareket sıklığı (otobüs sefer sayısını belirler),

A_k : herhangi bir k rotasında rotayı etkileyen diğer girdiler,

CAP: otobüs ağı içinde kullanılan otobüslerin yolcu kapasiteleri,

q_{ij}^k : Herhangi bir k Rotasında iki durak arası (i-j) yolcu akışı,

g_{ij}^k : k rotasında i ve j arasında akan yolcu sayısını belirleyen genel fonksiyon,

V^{ab} : a ve b düğümleri arasındaki orijine olan uzaklık,

N: otobüs ağındaki düğüm sayıları,

L_k : k rotasındaki iki durak arası mesafe sayıları,

SR: Otobüslerin kullanacağı rota sayısı,

T_k : herhangi bir k rotasında bir döngü için geçen seyahat zamanı (ertelemeler ve beklemler dâhildir),

X_{ij} : farklı rotaların kullandığı aynı duraklar (i-j),

M: Toplam Otobüs sayısı

N_k : k rotasındaki otobüs sayısı

W_0 :Filo yani toplam otobüs sayısı “ilk aşama optimizasyon işeminde belirlenmişti”.

KISALTIMA LİSTESİ

GA: Genetik Algoritma

H: Alt dizi

GSP: Gezgin Satıcı Problemi

PMX: Partially Matched Crossover

OX: Order Crossover

CX: Cycle Crossover

LOX: Linear Order Crossover

İETT: İstanbul Elektrik Tramvay Tünel

ŞEKİL LİSTESİ

Şekil 2.1 GA Akış Diyagramı.....	9
Şekil 2.2 GA Temel Kavramlar	11
Şekil 2.3: GA da Low Diversity ve High Diversity (Matlab GATOOL, 2004).....	12
Şekil 2.4 Ağaç Kodlama (Nabiyev, 2005).....	16
Şekil 2.5 (a) Ağaç Yapısında Ebeveyn Diziler (b) Ağaç Yapısında Yavru Dizi (Taskin ve Emel, 2009).....	31
Şekil 2.6 Ara Birleşmeli Çaprazlama karar uzayındaki durumu (Şen, 2004).	32
Şekil 2.7 Doğrusal Birleşim Üyeleri (Şen, 2004).	33
Şekil 2.8 Ağaç Kodlamada Çaprazlama Operatörünün ve Mutasyon İşleminin Kullanılması	36
Şekil 2.9 İlk Popülasyonun Oluşturulması (Matlab GATOOL, 2004).....	39
Şekil 2.10 Genetik Algoritma Çocuk Oluşumu (Matlab GATOOL, 2004)	40
Şekil 2.11 Genetik Algoritma ile Elit-Çaprazlama-Mutasyon Gösterimi (Matlab GATOOL, 2009)	41
Şekil 3.1 MATLAB ToolBoxes (MATLAB 7.9.0 (R2009), 2009).....	58
Şekil 3.2 MATLAB Simulink (MATLAB 7.9.0 (R2009), 2009).....	59
Şekil 3.3 MATLAB Genetik Algoritma Tool Ara Yüzü (MATLAB 7.9.0 (R2009), 2009)....	60
Şekil 3.4 Rastrigin's Fonksiyon Grafiği. (GA Tools, 2004)	62
Şekil 3.5 Rastrigin's Fonksiyon Grafiği . (GA Tools, 2004)	62
Şekil 3.6 Genetik Algoritma İle Grafik Çizimi "En iyi ve Ortalama Uygunluk Fonksiyonu Değeri" (GA Tools, 2004)	64
Şekil 3.7 Genetik Algoritmanın ilk popülasyon oluşturma matrisi	65
Şekil 3.8 MATLAB Genetik Algoritma Tool'unda İlk Popülasyonun Operatörü Algoritması	65
Şekil 3.9 MATLAB Genetik Algoritma Tool'unda Çaprazlama Operatörü Algoritması	66
Şekil 3.10 MATLAB Genetik Algoritma Tool'unda Mutasyon Operatörü Algoritması	67
Şekil 3.11 MATLAB Genetik Algoritma Tool'unda Seçim Operatörü Algoritması	67
Şekil 4.1 Temsili Otobüs Ağı 1	79
Şekil 4.2 Temsili Otobüs Ağı 2	80

ÇİZELGE LİSTESİ

Çizelge 2.1 GA ve Standart Algoritma Arasındaki Farklar.....	7
Çizelge 2.2 Başlangıç Toplum Kromozomları (Şen, 2004).	19
Çizelge 2.3 LOX Çaprazlama Örneği.....	30
Çizelge 2.4 Bazı Çoğalma Operatörlerinin Seçim Basınçları	44
Çizelge 2.5 Bazı Çaprazlama Operatörlerinin Arama Güçleri	44
Çizelge 4.1 82, 92, 97A Rotalarına Ait Durak Mesafe Matrisi (Metre).....	74
Çizelge 4.2 82 Numaralı İETT Rota Seyahat Talep Matrisi	74
Çizelge 4.3 92 Numaralı İETT Rota Seyahat Talep Matrisi	75
Çizelge 4.4 97A Numaralı İETT Rota Seyahat Talep Matrisi	76
Çizelge 4.5 Üç İETT Hattı İçin Toplam Seyahat Talep Matrisi.....	76
Çizelge 4.6 İETT de Üç Hat İçin Kullanılan Gerçek Değerler Listesi.....	78
Çizelge 5.1 Karşılaştırma Çizelgesi.....	87

ÖNSÖZ

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 algoritmalar problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler.

Böyle önemli bir konuda çalışmak bana son derece önemli bir akademik bakış açısı ve tecrübe kazandırdı. Bu önemli konuyu çalışırken vermiş olduğu değerli katkılardan ve izlemiş olduğu etkin rehberlik politikasından dolayı Hocam Yrd. Doç. Dr. Hayri BARAÇLI'ya,

Uygulama kısmında sizinle paylaşacağım bilgilere ulaşmamı sağlayan İETT'de görev yapan başta Ramazan YILDIRIM olmak üzere Başkanlık Koordinasyon Şube Müdürü Sayın Ayhan AYVAZ ve diğer İETT çalışanlarına,

Tüm eğitimim boyunca sürekli yanımda olan; özellikle manevi desteklerini her an arkamda hissettiğim annem ve babam olmak üzere, onlarsız bu çalışmayı asla gerçekleştiremeyeceğim ağabeylerim, ablam ve kardeşime,

Teşekkürlerimi sunuyorum...

ÖZET

Günümüzün rekabet ortamında, işletmeler en az miktarda kaynak kullanarak, müşteri gereksinimlerinin hızlı bir şekilde belirlenmesi ve çözülmesi için farklı teknikler kullanmaktadır. Bu şekilde dünyada diğer şirketlerle daha iyi koşullarda rekabet etme yeteneği kazanmaktadırlar. Bu yüzden, hızlı değişen müşteri talepleri karşısında özellikle üretim planlarını veya hizmet planlarının yani faaliyet planlarını en çabuk oluşturan şirketler rekabette bir adım öne geçmektedir. Sürekli değişen ve globalizasyon ile birlikte bu değişimin hızlanması rekabet ortamında hızlı karar vermeyi ve kararların hızlı bir şekilde uygulanmasını gerektirmektedir.

Genetik Algoritmalar (GA) doğal seçim ilkelerine ve popülasyon genetiğine dayanan etkin bir arama yöntemidir. Bu GA'lar işletme mühendisliğine ve bilim dünyasındaki problemlere başarılı bir şekilde uygulanabilmiştir.(Golberg, 1994) GA'lar rastsal operatörler kullanırlar ve bu operatörleri muhtemel sonuçların elde edileceği çalışma alanında yeni noktaların oluşturulması için kullanır. (Erick Cantu Paz, 1995

Son yıllarda GA'lar çok zor problemlerin çözümünde kullanılmıştır. Zor problemler büyük popülasyonlara ihtiyaç duyar ve bu büyük popülasyonlarda çok büyük hesaplama maliyetlerine neden olurlar. GA'lar ile uğraşan daha önceki çalışmaların konsantre oldukları önemli konu, operasyon zamanlarının minimizasyonu veya azaltılması ile birlikte kabul edilebilir bir sonucun ortaya konulabilmesidir. (Erick Cantu Paz,1995) Bu, GA'ların birçok paralel yapı şeklinde oluşturularak yeniden yapılandırılması kaydı ile gerçekleşir.

Bu çalışmanın ana amacı otobüslerin rotalanmasının optimizasyonu sağlamak ve onların sistemde ne sıklıkla yer almaları gerektiği konusunda etkin bir metot ortaya koymaktır. Bu yüzden kullanıcı makul bir hesaplama zamanıyla fizibil bir çözüm elde edecektir. Optimize edilmiş metot bize değişik kısıtlar altında çoklu amaç fonksiyonunun hesaplanmasını sağlayacaktır. Bunu yaparken de yöneylem tekniklerinden olan hat dengeleme yöntemleriyle Genetik Algoritma yöntemini birlikte kullanacaktır.

Anahtar Kelimeler: Optimizasyon, Hat Dengeleme, Toplu Taşımlar, Genetik Algoritma ve MATLAB.

ABSTRACT

Almost all companies use different methods to respond customer expectancy by using minimum sources that can give them a strong ability to compete with other companies in the world. Therefore, in the face of rapidly changing customer demands, a quick action by company plans to compete is a step ahead. With globalization and rapidly changing competitive environment, the acceleration of these changes and the rapid decision-making requires the quick implementation of decisions.

Genetic algorithms are an efficient search method based on the principles of natural selection and population genetics. Genetic algorithms are effectively used for the problems in management engineering and sciences. (Golberg-1994) Genetic algorithms use random operators and these potential operators are used in the study area for the creation of a new point. (Erick Cantu Paz, 1995)

Recently Genetic algorithms have been used to solve many difficult problems. Difficult problems require great populations and grate populations necessitate great counting cost. The important issue which is covered in previous studies in genetical algorithms is minimization of operational process and supplementation of acceptable result (Erick Cantu Paz,1995). This can be achieved by re-arranging genetical algorithms in the form of many parallel structures.

The objectives of this study is to supply an optimization framework for bus scheduling models and to provide a suitable method on what frequency they ought to be in this system. Therefore the user can obtain a feasible solution with counting time. The optimized method provides us the calculation multiple objective function under different restrictions. This will be done via using genetical algorithms method with the method of line balancing which is one of the operation researches techniques

Key Words: optimization, Line Balancing, Public Transportation, Genetic Algorithm and MATLAB.

1. GİRİŞ

Günümüzün rekabet ortamında, işletmeler ve kamu şirketleri en az miktarda kaynak kullanmak, müşteri gereksinimlerinin hızlı bir şekilde belirlenmesi ve çözülmesi amaçlı gerekli teknikleri kullanmak ürün ve hizmet üretmenin olanaklarını araştırmaktadırlar. Bu yüzden, hızlı değişen müşteri talepleri karşısında özellikle üretim planlarını veya hizmet planlarının yani faaliyet planlarını en çabuk oluşturan şirketler rekabette bir adım öne geçmektedir. Her geçen gün artan rekabet koşullarında şirketler ayakta kalabilmek, etkili yöntemler seçmek için optimizasyon tekniklerinden yararlanıp ve sahip oldukları bilgiyi doğru kullanmak zorundadır. Sürekli değişen ve globalizasyon ile birlikte bu değişimin hızlanması rekabet ortamında hızlı karar vermeyi ve kararların hızlı bir şekilde uygulanmasını gerektirmektedir. Bu süreci hızlandıracak yöntemlerden biri Genetik Algoritmadır.

Genetik Algoritmalar (GA) doğal seçim ilkelerine ve popülasyon genetiğine dayanan etkin bir arama yöntemidir. Bu GA'lar işletme mühendisliğine ve bilim dünyasındaki problemlere başarılı bir şekilde uygulanabilmiştir.(Golberg, 1994) GA'lar rastsal operatörler kullanırlar ve bu operatörleri muhtemel sonuçların elde edileceği çalışma alanında yeni noktaların oluşturulması için kullanır. (Erick Cantu Paz, 1995)

Son yıllarda GA'lar çok zor problemlerin çözümünde kullanılmıştır. Zor problemler büyük popülasyonlara ihtiyaç duyar ve bu büyük popülasyonlarda çok büyük hesaplama maliyetlerine neden olurlar. GA'lar ile uğraşan daha önceki çalışmaların konsantre oldukları önemli konu, operasyon zamanlarının minimizasyonu veya azaltılması ile birlikte kabul edilebilir bir sonucun ortaya konulabilmesidir. (Erick Cantu Paz,1995) Bu, GA'ların birçok paralel yapı şeklinde oluşturularak yeniden yapılandırılması kaydı ile gerçekleşir.

Bu çalışma literatürde birçok farklı modelin ve algoritmanın otobüslerin rota optimizasyonu ve otobüslerine kadar sıklıkla sistemde olmaları gerektiği konusunu analiz etmektedir. Buradaki problemin çözümü için farklı sezgisel prosedürler ile birlikte yöneylem araştırması tekniklerinden de faydalanılmış ve ortaya konulan yöntem uygulanabilir olduğu kadar etkileşimi de sağlamaktadır.

Bu çalışmanın ana amacı otobüslerin rotalanmasının optimizasyonu sağlamak ve onların sistemde ne sıklıkla yer almaları gerektiği konusunda etkin bir metot ortaya koymaktır. Bu yüzden kullanıcı makul bir hesaplama zamanıyla fizibil bir çözüm elde edecektir. Optimize edilmiş metot bize değişik kısıtlar altında çoklu amaç fonksiyonunun hesaplanmasını

sağlayacaktır. Bunu yaparken de yöneylem tekniklerinden olan hat dengeleme yöntemleriyle Genetik Algoritma yöntemini birlikte kullanacaktır.

2. GENETİK ALGORİTMA

2.1 Evrimsel Hesaplama

Özellikle geçen 20–30 yılda evrim prensibine dayalı algoritmalara ilgi oldukça artmış ve bu konularla ilgili araştırmacıların ve literatürde ki çalışmaların sayısı hızla çoğalmıştır. Son zamanlarda bu prensibe dayanan tekniklerin hepsini temsilen ortak bir terim olarak “evrimsel hesaplama” terimi yaygın olarak kullanılmaya başlanmıştır (Kalinli, Karaboga, 2004).

Bugün bilgisayar yöntemleri biyolojik değerlendirmeden esinlenerek evrimsel hesaplama olarak adlandırılan bir şemsiye altında gruplandırılmıştır. Evrimsel hesaplamanın ana elemanlar aşağıda tanımlanmaktadır.

- Değerlendirme stratejileri
- Evrimsel programlama
- Genetik Algoritmalar

Evrimsel hesaplama sınıfına giren algoritmalar; GA’lar, evrimsel programlama, evrimsel stratejiler, genetik programlama ve diferansiyel gelişim algoritmalarıdır. Ayrıca literatürde bu paradigmanın çeşitli özelliklerini birlikte kullanan çok sayıda melez sistemler de mevcuttur. Bundan dolayı sınıflandırma yapmak oldukça zordur. Bir problemi çözmeye kullanılacak herhangi bir evrimsel algoritma aşağıdaki beş elemana ihtiyaç duymaktadır (Kalinli, Karaboga, 2004):

- Problemler için çözümlerin genetik temsili,
- Çözümlerin başlangıç popülasyonunu oluşturacak bir yöntem,
- Çözümleri uygunluk açısından değerlendirmeye tabii tutacak değerlendirme fonksiyonu, yani çerçeve,
- Genetik kompozisyonu değiştirecek operatörler,
- Kontrol parametrelerinin değerleri (popülasyon büyüklüğü, operatörleri uygulama ihtimalleri vs.).

2.2 Genetik Algoritmanın Tarihçesi

Genetik Algoritma adı ilk olarak 1975 yılında Michigan Üniversitesinde Profesör John Holland tarafından ortaya konulmuştur. Genetik algoritma bilgisayar tabanlı araştırma ve doğal genetik ve doğal seçim mekaniği ile çalışan bir optimizasyon algoritmasıdır (Goldberg, 1994).

GA doğadaki canlıların geçirdiği süreci örnek alır ve iyi nesillerin kendi yaşamlarını

korurken, kötü nesillerin yok olması ilkesine dayanır. Matematiksel modellerin yapılmadığı ve kesin çözümün olmadığı problemlerde genetik algoritmalarından yararlanır. Bu algoritma, anne ve baba bireylerinden (bir önceki nesil) doğan yeni bireylerin şartlara uyum sağlayıp yaşamlarını devam ettirmesine dayanır. (Elmas, 2002). Yani anne ve babadan genlerini alan yeni bireylerin iyi veya kötü olması mümkündür. Buna bağlı olarak iyi genler hayatlarını sürdürdürebilirken kötü genler ise hayatlarını sürdürememektedir.

Genetik algoritmalar, uygunluk (Fitness) işlevi, yeni çözümler üretmek için çaprazlama ve değiştirme gibi operatörleri kullanır. Genetik algoritmanın önemli özelliklerinden bir tanesi de bir grup üzerinde çözümü araması ve bu sayede çok sayıda çözümün içinden en iyiyi seçmesidir. (Elmas, 2002).

Golberg'in tanımına göre GA, rastlantısal arama yöntemleri kullanılarak çözüm bulmaya çalışan, parametre kodlama esasına dayanan sezgisel bir arama tekniğidir.

Michigan Üniversitesinde psikoloji ve bilgisayar bilimi uzmanı olan John Holland bu konuda ilk çalışmaları yapan kişidir. Makine öğrenmesi konusunda çalışmalar yapan Holland, evrim kuramından etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünmüştür. Genetik Algoritma ilk ismini biyoloji, ikinci ismini ise bilgisayar biliminden almaktadır (Holland, 1992).

Darwin'in evrim kuramında etkilenerek canlılarda yaşanan genetik süreci bilgisayar ortamında gerçekleştirmeyi düşünen John Holland, tek bir mekanik yapının öğrenme yeteneğini geliştirmek yerine böyle yapılarda oluşan bir topluluğun çoğalma, çiftleşme, mutasyon, vb. Genetik süreçlerden geçerek başarılı (öğrenebilen) yeni bireyler oluşturabildiğini görmüş ve geliştirdiği yöntemin adı Genetik Algoritmalar olarak yerleştirmiştir.

1985 yılında David E. Goldberg gaz boru hatlarının denetimi üzerine doktora tezi yapmıştır. Goldberg' in gaz borusu hatlarının denetimi üzerine yaptığı çalışma ona 1985 Ulusal Bilim Fonu Genç Araştırmacı ödülünü kazandırmış, ayrıca genetik algoritmaların pratik kullanımının da olabildiğini kanıtlamıştır (Goldberg, 1994).

1992 yılında John Koza genetik algoritmayı kullanarak genetik programlamayı geliştirmiştir (Koza, 1992).

GA'lar günümüzde güçlü arama algoritmaları olarak kendilerini etkin bir şekilde kullanılmaktadır. GA'lar çok çeşitli problemlerde başarıyla uygulanabiliyor olması,

sistemde etkili olmalarını ve tercih edilmelerinde ki etkindir.

2.3 Genetik Algoritmaların Temel Kuramı

Genetik algoritmaların nasıl arama yaptığı alt dizi kavramıyla açıklanmaktadır. Alt diziler, genetik algoritmaların davranışlarını açıklamak için kullanılan teorik yapılardır. Bir alt dizi, belirli dizi kümeleri arasındaki benzerliği tanımlayan bir dizidir. Alt diziler, $\{0, 1, *\}$ alfabeti kullanılarak tanımlanır. Örneğin H alt dizisi, ilk konumunda 0, ikinci ve dördüncü konumunda 1 değeri olan kromozomlar kümesi içindir.

$$H = 0 1 * 1 *$$

* sembolü dizinin o konumunun hangi değeri alıp almadığının önemli olmadığı anlamındadır. Dizi o konumda 0 veya 1 değeri alabilir. Eğer bir x dizisi, alt dizinin kalıbına uyarsa x dizisine “H’nin bir örneğidir” denir. Alt dizilerin iki özelliği mevcuttur. Bu özellikler aşağıda verilmiştir (Taşkın ve Emel, 2009).

- Al
t dizi derecesi: Bir H alt dizisinin derecesi $o(H)$ ile gösterilir ve mevcut alt dizi kalıbında bulunan sabit konumların sayısıdır. Bu sayı ikili alfabede 0 ve 1 değerlerinin sayısının toplamına eşittir.
- Al
t dizi uzunluğu: Bir H alt dizisinin uzunluğu $\delta(H)$ ile gösterilir ve mevcut alt dizi kalıbında bulunan belirli ilk ve son konumlar arasındaki uzaklıktır.

Alt dizi derecesi ve alt dizi uzunluğu kavramlarının genetik algoritmaların temel teoreminde son derece önemli bir yeri vardır. Alt dizi derecesi düşük, alt dizi uzunluğu kısa olan diziler “yapı blokları” olarak adlandırılır. John Holland, genetik algoritmaların işleyişinde uygun yapı bloklarının tanımlanmasını ve bu yapı bloklarının daha uygun yapı blokları elde etmek amacıyla birleştirilmesini önermektedir. Bu fikir yapı blokları hipotezi olarak bilinmektedir. Genetik algoritmanın temel teoremi ise şöyle açıklanmaktadır (Taşkın ve Emel, 2009):

“Popülasyon ortalamasının üstünde uyum gücü gösteren, kısa uzunluğa ve düşük dereceye sahip alt diziler zamanın ilerlemesiyle üstsel olarak çoğalırlar.”

Bu çoğalma, genetik işlemler aracılığı ile gerçekleşmektedir ve sonucunda ana-babadan daha üstün özellikler taşıyan bireyler ortaya çıkmaktadır. Bu çözüm kalitesinin kuşaktan kuşağa artması iki nedene bağlanmaktadır. Bu nedenler şöyle açıklanabilir (İşlier, 2001):

- Ba
şarısız olan bireylerin üreme şansı azaltıldığı için kötüye gidiş zorlaşmaktadır.
- Ge
netik algoritmaların yapısı kötüye gidişi engellemekle kalmamakta, genetik algoritmaların temel teoremi uyarınca, zaman içinde hızlı bir iyiye gidiş de sağlayabilmektedir.

Genetik algoritmaların işleme adımları incelendiğinde bu nedenler daha iyi anlaşılacaktır. Genetik algoritmalar yapısı gereği, kötü bireyleri yani uygun olmayan çözümleri, operatörleri sayesinde elemektedir. Bu işlemler bir döngü içerisinde durdurma kriteri sağlanana kadar devam etmektedir (Ying-ping, 2006).

2.4 Genetik Algoritmanın Çalışma Prensibi

Genetik algoritma, doğal seçime dayalı optimizasyon problemlerinin çözümü için kullanılan bir metottur, bu proses biyolojik gelişim şeklindedir (Nils, 1998). GA’ma sürekli olarak bir önceki bireyleri modifiye ederek ilerler. Her basamakta yeni popülasyon bireyleri oluşturmak için var olan popülasyondan rastgele bir şekilde iki ebeveyn seçer. Başarılı bir seçimden sonra “gelişim” optimal bir sonuç bize verir. GA’yı birçok değişik optimizasyon problemini çözmek için kullanabilirsiniz. Bu standart optimizasyon algoritmaları için en iyi bir yöntem demek değildir.

GA, üç temel kural üzerine hareket eder. Her kural yeni nesli eski popülasyondan oluşturmaya çalışır (Matlab GATOOL).

- Se
çim Kuralı; bu kural yeni neslin oluşturacağı genleri seçer “ebeveyn”,
- Ç
a
prazlama kuralı; bu kural yeni neslin oluşturmak için ebeveynleri birleştirir
- M
utasyon kuralı; bu kural rastlantısal olarak bireysel olarak her ebeveyn üzerinde değişiklik gerçekleştirerek yeni neslin daha güçlü olmasını sağlar.

GA, standart optimizasyon algoritmalarından iki ana noktada farklıdır. Aşağıda bu değişiklikler belirtilmiştir (Matlab GATOOL).

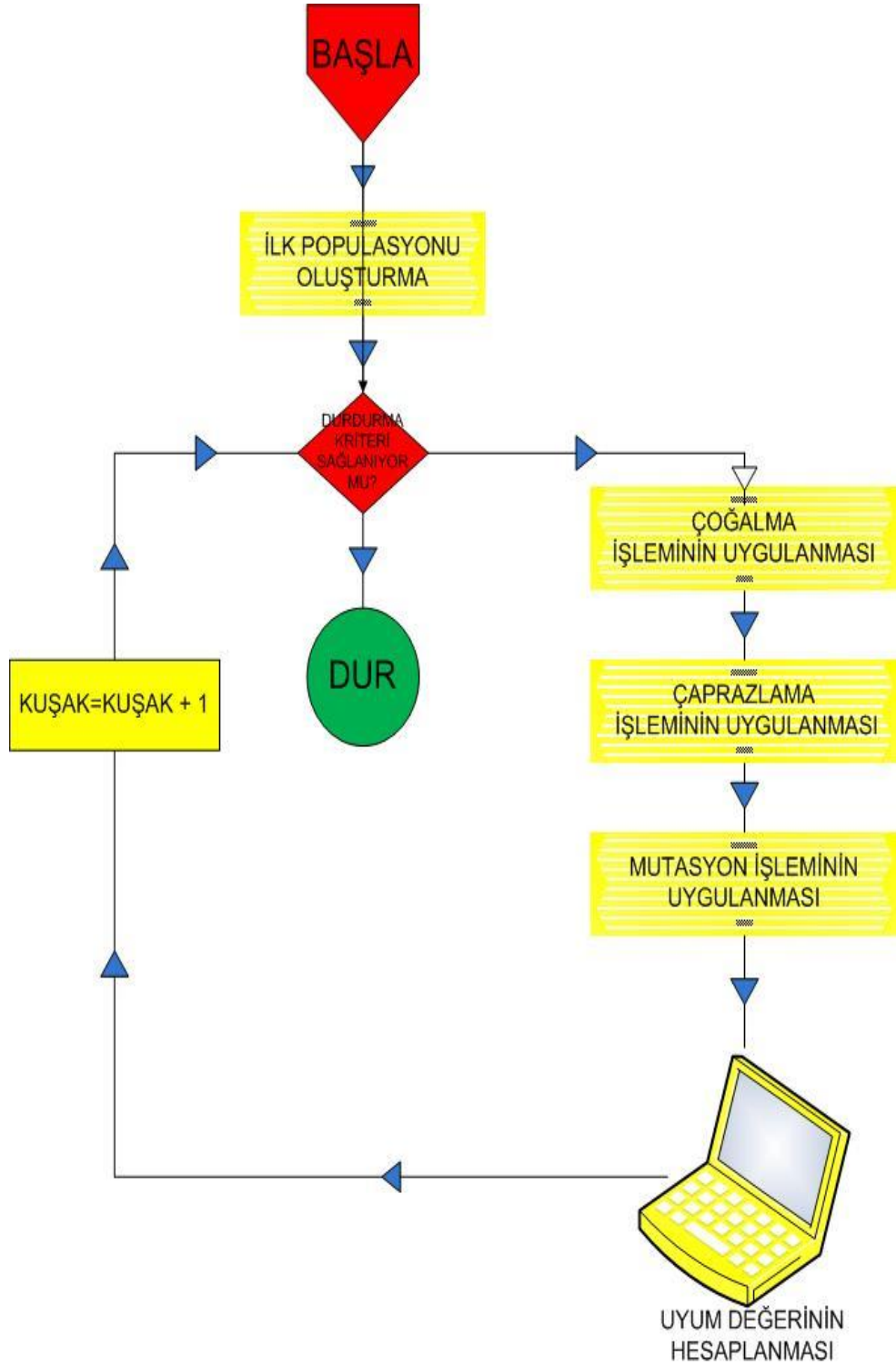
Çizelge 2.1 GA ve Standart Algoritma Arasındaki Farklar

Standart Algoritma,	Genetik Algoritma
Her iterasyonda sadece bir nokta oluşturur noktaların sırası optimal sonuca ulaşır.	Her iterasyonda popülasyon noktaları oluşturulur. Popülasyon optimal sonuca ulaşır.
Deterministik hesaplamalarla sıralamada bir sonraki nokta belirtilir.	Rastsal seçimlerle bir sonraki popülasyon belirlenir.

Genetik algoritmanın ilk adımı, ilk popülasyonun oluşturulması, uyum değerinin hesaplanmasıdır. Daha sonra mevcut popülasyona, temel genetik operatörler Çoğalma, Çaprazlama ve mutasyon uygulanır. Her kuşak için uyum değeri hesaplanır. Bu durum durdurma kriteri uygulanana kadar devam eder. Şekil 2.1de GA'nın akış diyagramı verilmektedir. Genetik algoritmanın adımlarını şöyle sıralayabiliriz. (Taşkın ve Emel, 2009)

- Arama uzayındaki tüm olası çözümlerden bir grup çözüm dizini olarak kodlanır.
- Genellikle rastsal bir çözüm kümesi seçilir ve başlangıç popülasyonu olarak kabul edilir.
- Her bir dizi için bir uyum değeri hesaplanır. Bulunan uyum değerleri dizilerin çözüm kalitesini gösterir.
- Bir grup dizi belirli bir olasılık değerine göre rastsal olarak seçilir.
- Yeni dizilerin uyum değerleri hesaplanarak, çaprazlama ve mutasyon işlemlerine tabi tutulur. Oluşan yeni popülasyon ile yer değiştirilir.

- lirlenmiş durdurma kriteri sağlanana kadar yukarıdaki işlemler devam ettirilir. Be
- öngü, durdurma kriteri sağlanınca sona erdirilir. Amaç fonksiyonuna göre en uygun olan dizi seçilir. D



Şekil 2.1 GA Akış Diyagramı

2.5 Genetik Algoritmanın Temel Kavramları

GA'da; çözüme ulaşmak için algoritma yapısının oluşturulması ve parametrelerin belirlenmesi için kullanılan kavramların ve değerlerinin iyi belirlenmesi gerekir. Aşağıda bu kavramlar konusuna değinilmiştir.

2.5.1 Gen

Kromozom yapısında kendi başına birer genetik bilgi taşıyan en ufak yapı taşına gen denir (Mitsuo, Cheng, Lin, 2008). Bir diğer şekliyle gen, kendi içinde probleme ait en küçük bilgi taşıyan yapı taşına denir.



2.5.2 Kodlama

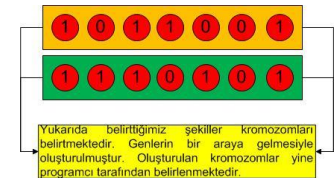
GA'nın kullanıldığı programlama yapısında bu gen programcının tanımlamasına bağlıdır. Bir genin içerdiği bilgi sadece ikili tabandaki sayıları içerebildiği gibi onluk taban ve onaltılık tabandaki sayı değerlerini de içerebilir(Elmas, 2007).

Çözümlerin kodlanması, probleme özgü genetik algoritmanın kullanacağı şekle çevrilmesine olanak tanır. Genetik algoritmayla çözülecek problemler tiplerinden dolayı, farklı kodlama biçimlerine gereksinim duyar(Taşkın ve Emel, 2009).

Yukarıda da belirtildiği gibi kodlama biçimini programa ve programcıya bağlı olduğu için bütün problemler için geçerli en uygun kodlama biçimini söylemek mümkün değildir.

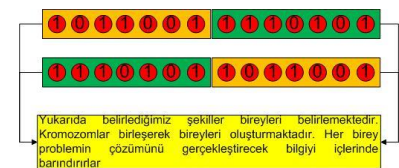
2.5.3 Kromozom

Bir ya da birden fazla gen yapısının bir araya gelerek problemin çözümüne ait tüm bilgiyi içeren dizilere kromozom denir (Mitsuo, Cheng, Lin, 2008).



2.5.4 Birey

Kromozomların bir araya gelmesiyle oluşan yapıya birey denir. Bireyler üzerinde çalışılan problem için, olası çözüm bilgilerini içermektedirler. Her birey problem için bir çözüm



adayıdır.(Nilsson, 1998)

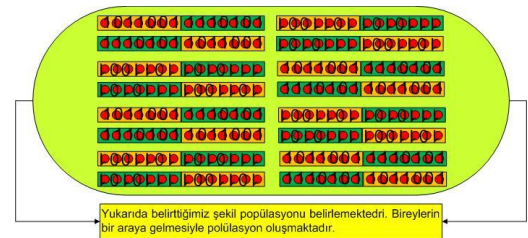
Birey, uygunluk fonksiyonuna atanacak herhangi bir noktayı ihtiva eder. Uygunluk fonksiyonunun değeri bireylerin skorlarına bağlıdır. Örneğin, bir uygunluk fonksiyonu; (Matlab GATOOL, 2004).

$$f(x_1, x_2, x_3) = (2x_1 + 1)^2 + (3x_2 + 4)^2 + (x_3 - 2)^2$$

(2, -3, 1) değişkenleri bireylere verilse fonksiyon skor'u $f(2, -3, 1) = 51$ bulunur.

2.5.5 Popülasyon ve Jenerasyon

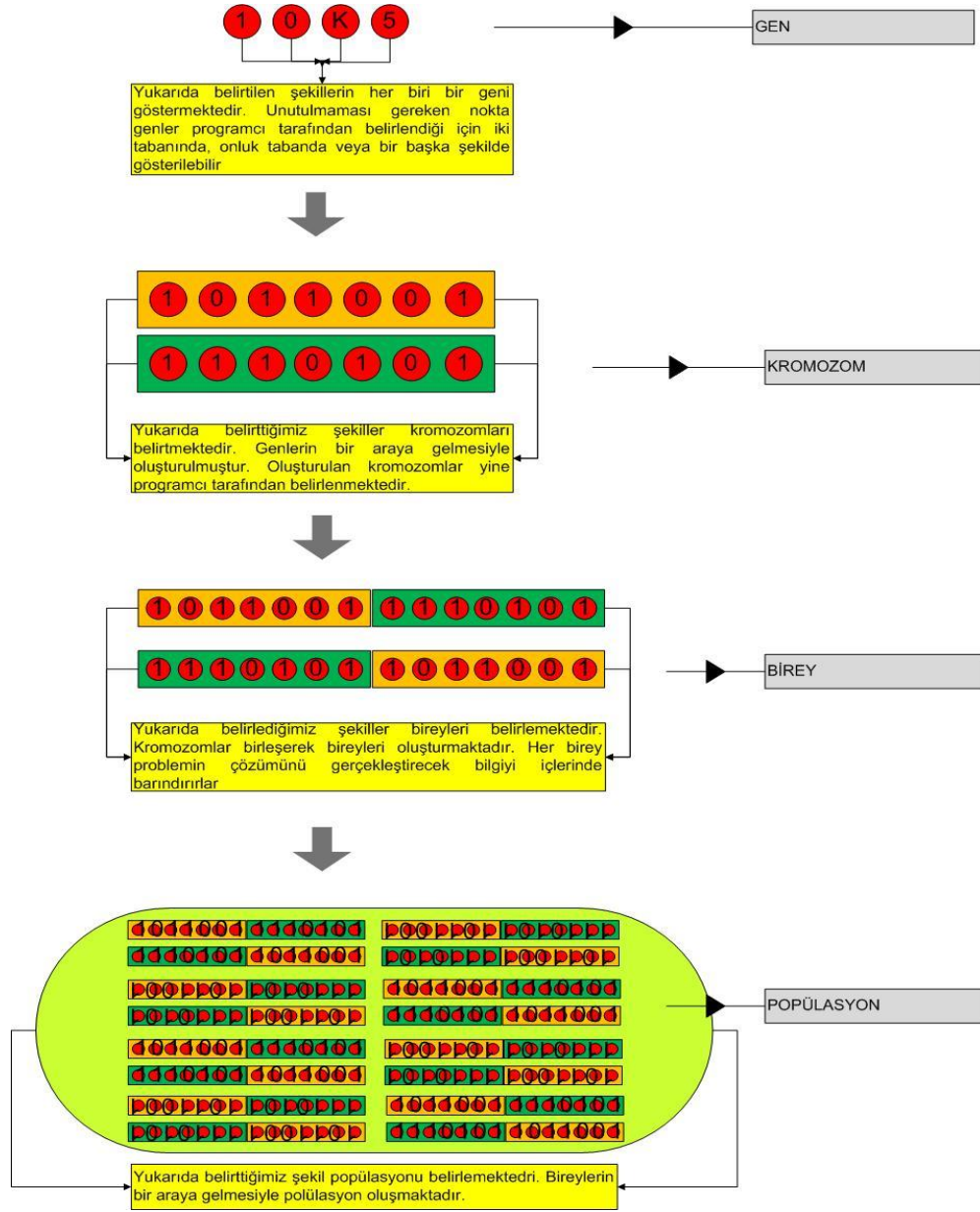
Bireylerin bir araya gelmesiyle popülasyon meydana gelir. Bireyler çözülmeye çalışılan problem konusunda gerekli bilgi yapı taşlarına sahip olup problemi çözmek amaçlı bir araya gelip popülasyonu oluştururlar.



Bir popülasyondaki kromozom sayısı belli olup problemin özelliğine göre programcı tarafından belirlenir. GA'nın işleyişi esnasında bu popülasyon kümesinden bir takım kromozomlar yok olmakta ve yerlerine yeni kromozom yapıları eklenerek popülasyon büyüklüğü sabitlenmektedir (Elmas, 2007). Popülasyon büyüklüğü problemin çözüm zamanını doğrudan etkilemektedir. Popülasyon büyüklüğü bu nedenle programcı tarafından iyi belirlenmesi gerekmektedir.

Popülasyonu yukarıda tanımlamıştık. Popülasyon bireylerin bir araya gelmesiyle oluşur. Örneğin, eğer popülasyon büyüklüğü 100 ise ve uygunluk fonksiyonundaki değişkenler 3 ise popülasyonu büyüklüğü 100 olan 3 matris ile gösterilir (Matlab GATOOL, 2004).

Her iterasyon, bilgisayar var olan popülasyondan yeni popülasyon oluşturmak için değişik serisel hesaplama yapar. Her başarılı hesaplama serisi yeni popülasyon diye adlandırılır (Nilsson, 1992).



Şekil 2.2 GA Temel Kavramlar

2.5.6 Uygunluk Fonksiyonu

Uygunluk fonksiyonu optimize etmeye çalıştığımız fonksiyondur. Standart optimizasyon algoritması için bu objektif fonksiyon olarak da bilinir. Matlab Toolbox bu uygunluk fonksiyonunun duruma göre maksimum veya minimum değerlerini bulmaya çalışmaktadır (Matlab GATOOL, 2004).

2.5.7 Bireyler

Birey, uygunluk fonksiyonuna atanacak herhangi bir noktayı ihtiva eder. Uygunluk fonksiyonunun değeri bireylerin skorlarına bağlıdır. Örneğin, bir uygunluk fonksiyonu;

(Matlab GATOOL).

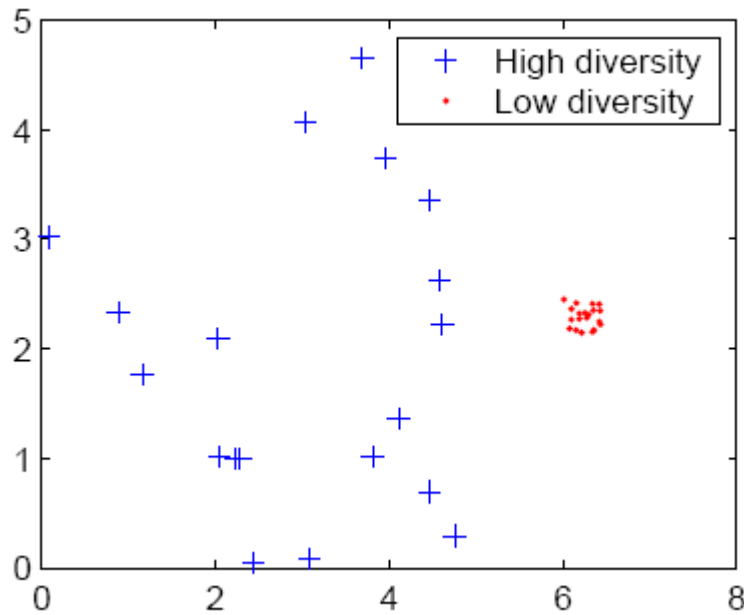
$$f(x_1, x_2, x_3) = (2x_1 + 1)^2 + (3x_2 + 4)^2 + (x_3 - 2)^2$$

(2, -3, 1) deęişkenleri bireylere verilse fonksiyon skor'u $f(2, -3, 1) = 51$ bulunur.

2.5.8 Diversity (Farklılık/Saçkınlık)

Diversity, popülasyondaki bireyler arasındaki ortalama mesafeyi ifade eder. Eęer ortalama mesafe yüksek ise *high diversity* (Yüksek Farklılık/Saçkınlık) diye adlandırılır. Eęer ortalama mesafe az ise *low diversity* (Düşük Farklılık/Saçkınlık) diye adlandırılır. Şekil 2.3 de sol taraftaki popülasyon yüksek farklılığı, sağdaki popülasyonda düşük farklılığı bize göstermektedir (Matlab GATOOL, 2004).

Diversity GA için önemlidir çünkü Algoritma'nın araştırma yapacağı alan konusunda sınırları belirler.



Şekil 2.3: GA da Low Diversity ve High Diversity (Matlab GATOOL, 2004).

2.5.9 Uygunluk Deęeri ve En İyi Uygunluk Deęeri

Bireylerin Uygunluk deęeri, uygunluk fonksiyonunun deęerine belirli bireyler için verdiği deęerdir. En iyi uygunluk deęeri ise uygunluk fonksiyonunun popülasyon içinde herhangi bir birey için bulduęu problemin yapısı gereęi en maksimum/minimum deęeri belirtir. (Jain, Martin, 1998).

2.5.10 Ebeveyn ve Çocuk

Yeni bir nesli oluşturmak için GA mevcut popülasyondaki bireylerden seçimler gerçekleştirir. Bu seçilen bireylere Ebeveyn denir. Ve bu seçilen ebeveyn den oluşturulan yeni jenerasyon oluşturulur. Bu yeni jenerasyona Çocuk denir (Mitsuo, Cheng, Lin, 2008). Hepimizin bildiği gibi ve daha öncede belirttiğimiz gibi yeni jenerasyon yani Çocuklar en iyi uygunluk değerine sahip ebeveynlerden oluşturulur.

2.6 Genetik Algoritma Kullanım Sebepleri

Problemin zorluk derecesinin bilinmesi problemin çözümü için en iyi yöntemin uygulanmasını sağlar. Polinomal olan denklemlerin çözümlenmesi ve incelenmesi kolay olan çözümlerdir. Sorunun çözüme kavuşturulmasına yardımcı yöntemler mevcuttur. Polinomal olmayan denklemler veya problemlerin çözümlenmesi zor sistemlerdir. Bu polinomal olmayan sistemlerin kısa sürede çözüme kavuşturan yöntemler mevcut değildir. Bu nedenle polinomal olmayan sistemlerin çözüme kavuşturulması için gerçek çözüme en yakın sonucu bulmak amacıyla yaklaşık çözüm algoritmaları geliştirilmiştir.(Elmas, 2002)

Yaklaşık çözüm algoritmaları, problemin gerçek sonucunu bulamayan fakat gerçek çözüme en yakın bir çözümü kısa bir süre içerisinde ulaşabilir. Geçek yaşamda karşılaşılan sorunlar da bizden gerçek sonucu değil de kısa sürede gerçek çözüme yakın bir sonuç elde etmemizi çoğunlukla ister. Bu nedenle gerçek sistemde karşılaşılan polinomal olmayan problemlerin çözümünde sezgisel yöntemler yardımıyla geliştirilen algoritmalar kullanılmaktadır. Bizim gerçek hayatta karşılaştığımız problemler karşısında algoritmanın ne kadar sürede sonuca ulaştığı önemli bir noktadır. Gelişen ve değişen bir rekabet ortamında gereklerine hızlı ve optimal bir sonuçla cevap vermek sistemin ayakta kalmasını ve rekabette başarı sağlayacaktır.

GA, sezgisel bir yöntem olduğundan dolayı verilen bir problem için kesin sonucu veremeyebilir ancak bilinen yöntemlerle çözümlenemeyen veya çözüm zamanı problemin büyüklüğü ile üssel olarak artan problemlerde kesin sonuca çok yakın çözümler verebilir (Melanie, 1999). GA, fonksiyon en iyilemesi, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarı ile uygulama alanları bulunmaktadır (Elmas, 2002).

2.7 Genetik Algoritmanın Diğer Yöntemlerden Farkı

- GA'lar problemlerin çözümünü parametrelerin değerleriyle değil, kodlarıyla arar. Parametreler kodlanabildiği sürece çözüm üretilebilir. Bu sebeple GA'lar ne yaptığı konusunda bilgi içermez, nasıl yaptığını bilir.

- GA'lar aramaya tek bir noktadan değil, noktalar kümesinden başlar. Bu nedenle çoğunlukla yerel en iyi çözümde sıkışıp kalmazlar.
- GA'lar türev yerine uygunluk fonksiyonunun değerini kullanır. Bu değer kullanılması ayrıca yardımcı bir bilginin kullanılmasını gerektirmez.
- GA'lar gerekirci kuralları değil olasılıksal kuralları kullanır.

2.8 Genetik Algoritmada Kullanılan Teknikler

Popülasyona temel genetik operatörlerin uygulanma aşaması GA'lar da ele alınan probleme özgü kodlamanın yapılması, ilk popülasyon oluşturulması ve uyum değeri hesaplanmasından sonra başlar. Genetik operatörler her zaman bir sonraki kuşaklarda daha iyi sonuçlar ortaya koyar (Wright).

GA'lar da kullanılan temel genetik operatörler; Çoğalma, çaprazlama ve mutasyon operatörleridir. Genetik algoritmalarda, probleme uygun olan farklı kodlama yöntemlerinin kullanılması, farklı çoğalma, çaprazlama ve mutasyon operatör çeşitlerinin kullanılması bir nedenidir. Ayrıca operatörler problem tipine göre, çözüm bulma hızı ve optimal değere yakınlık açısından farklılık göstermektedir. Bu nedenle problem çözümünde probleme uygun operatörlerin kullanılması son derece önemlidir (Taşkın ve Emel, 2009). Bunlara bağlı olarak sistemde önemli bir diğer konularda kromozomların şifrelenmesi, hedef fonksiyonun belirlenmesi ve durdurma kriterlerinin belirlenmesi oldukça önem taşımaktadır.

2.8.1 Kromozomların Şifrelenmesi

Kromozomların kodlanması GA ile çözmeye başladığımızda karşımıza çıkan önemli bir problemlerden bir tanesidir. Her probleme göre kromozomların kodlama şeklide değişebilmektedir. Kullanılan kodlama yöntemleri aşağıda sunulmaktadır.

2.8.1.1 İkili Kodlama

Bu yöntem ilk GA uygulamalarında kullanıldığı için hala en çok kullanılan yöntemlerdendir. Burada her kromozom, 0 ve 1 'lerden oluşan bit dizisidir ve ikili diziyle ifade edilir. Bu dizideki her bit, çözümün bir özelliğini taşır. Dizinin tümü ise bir sayıya karşılık gelir (Nabiyev, 2005).

Kromozom A 101110010110

Kromozom B 010110100000

İkili kodlamayla oluşturulan kromozomlar küçük problemler için çok büyük kromozom vektörü gerektirmektedir. Ele alınacak problemlere göre çeşitli şifreleme yöntemleri vardır.

2.8.1.2 Permütasyon Kodlama

Bu kodlama, Gezgin Satıcı Problemi (GSP) ve iş sıralama problemleri gibi permütasyon problemlerinde kullanılır. Burada her kromozom bir numaralar dizisidir (Nabiyev, 2005).

Kromozom A 35127604

Kromozom B 01562347

Bu problemlerde olan ikili düzende kodlama kullanılabilir. Örneğin, her onluk sayı için ikili düzendeki sayılar yazılır (Nabiyev, 2005). (Yapısına göre bilgisayarlar verileri ikili biçimde işlemektedir.)

0	icin	000
1	için	001
2	için	010
...
7	için	111 gibi

Bu durumda 8 genden (örneğin 8 şehirden) oluşan permütasyon kodlamalı kromozomlarımız 24 genden oluşacaktır ve her üç gen bir şehri ifade edecektir.

2.8.1.3 Değer Kodlama

Bu kodlama, gerçel gibi kompleks sayıların yer aldığı problemlerde kullanılır. Bu tür problemler için ikili kodlama çok zordur. Burada her kromozom bazı, değerler dizisidir. Bu değerler ise problemle ilişkilidir; örneğin gerçel sayı, karakter veya kompleks nesnelere olabilir (Nabiyev, 2005).

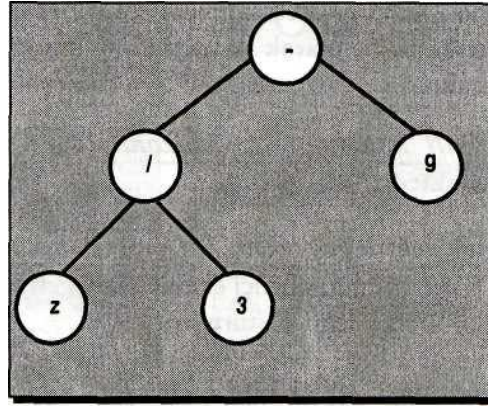
Kromozom A	3.2746.1212.456 0.2992.115
Kromozom B	ABCJFKHDERJFDLFFEGHK
Kromozom C	(geri), (sağ), (ileri), (geri), (sol)

Bu kodlama bazı özel problemler (örneğin bir yapay sinir ağının ağırlık katsayılarının bulunması) için çok uygundur.

2.8.1.4 Ağaç Kodlama

Genellikle ağaç kodlama genetik programlamada programlar ve ifadeler oluşturmak için kullanılır (Mitsuo, Cheng, Lin, 2008). Ağaç kodlamada her kromozom, adından da anlaşıldığı

gibi nesnelar ve nesnelar arası işlemleri içeren bir ağaç yapısından oluşur (Nabiyev, 2005).



Şekil 2.4 Ağaç Kodlama (Nabiyev, 2005)

Ağaç kodlama, program geliştirmek için uygundur. Örneğin, LISP ve Prolog gibi programlama dillerinde ağaçtan yapısal olarak sık bir biçimde kullanılır. Ağaç kodlamada çaprazlama ve mutasyon çok kolay bir şekilde uygulanabilir (Nabiyev, 2005).

Televizyon yarışmalarından iyi bildiğimiz, verilen hedef değere en yakın sonucu üretecek aritmetik ifadenin bulunması, bu kodlamanın kullanıldığı problemlere verilebilecek örneklerdendir. Bu ifade, sayılardan ve "+,-,*,/" aritmetik işlemlerden oluşmaktadır. İlk dört giriş sayısı 1 ile 10 arasından ve son iki sayı ise "25, 50, 75, 100" arasından seçilir. Burada önemli bir sınırlama vardır. Bu sınırlama yarışmacının her sayıyı yalnızca bir defa kullanabilmesidir. Örneğin (2, 3, 5, 8, 25, 100) sayıları ile 467 sayısını veren ifadeyi bulmaya çalışalım. $(5*100) - (25 + 8) = 467$ mümkün çözümlerden biridir. Fakat her zaman kesin sonucun bulunması mümkün değildir. Bu durumda ise genetik algoritma en yakın sonucu veren ifadeyi bulmaya çalışır. Problemin durum uzayının belli sınırlamalar olduğunda bile 33.664.168 noktadan oluştuğu düşünülürse, onun basit arama yöntemiyle çözülemeyeceği kesindir (Nabiyev, 2005). Buna başka bir örnek olarak ise yalnız tek bir kez l'den 9'a kadar sayıları ve yine dört aritmetik işlemi "+,-,*,/" kullanılarak 100 değerinin elde edilmesini gösterebiliriz. Örneğin,

$$((1*2)+3)*4*5-6+7+8-9=100\text{veya}$$

$$(1+2+3+4 + 5)*6-7+8+9 = 100\dots$$

8 farklı çözümü olan bu problemde de her sayı tek bir kez kullanılmaktadır.

2.8.2 Hedef fonksiyonunun belirlenmesi

Karar uzayının deęişik noktalarında dolaşan toplum kromozomlarından her birinin aranılan çözüme ne kadar yakın olduğunu belirtebilmek için mutlaka bir hedef fonksiyonuna, $h(x)$, gerek vardır. En iyinin yönüne göre yani en küçükleme veya en büyükleme olmasına göre hedef fonksiyonunun küçük veya büyük deęerlerine ulaşmak arzu edilir. GA toplumdaki her bir kromozomun (üyenin) göreceli dinçliklerinin hesaplanabilmesi için bunların mutlak (sadece artı deęerli) dinçlik eęerlerine yani hedef fonksiyonunda aldıkları deęerlere ihtiyaç vardır (Şen, 2004).

Hedef fonksiyonundan dinçlik derecelerini verecek bir fonksiyonun, $d(x)$, bir şekilde elde edilmesi gereklidir. Bu iki fonksiyon arasındaki kapalı ilişkiyi (Şen, 2004);

$$d(x) = F[h(x)]$$

şeklinde gösterebiliriz. Burada, F dönüşümü o şekilde olmalıdır ki sonuçta dinçlik derecelerini veren fonksiyonun deęerleri tüm karar deęişkenleri için mutlaka artı işaretli olmalıdır. Bu son ifadenin açık şekli, $d(x_i)$, i -inci karar deęişkeninin dinçlik deęeri olmak üzere, bu

$$d(x_i) = \frac{h(x_i)}{\sum_{j=1}^N h(x_j)}$$

şeklinde hesaplanabilir. Aslında hedef fonksiyonu deęerlerinin tümünün artı işaretli olması dinçlik derecesi o karar deęişkeninin toplum içindeki bulunma yüzdesini (ihtimalini) verir. Buna göreceli dinçlikte denilebilir. Bu tanımın en büyük mahzuru hedef fonksiyonunun eksi işaretli olan deęerlerini göz önünde tutamamasıdır. Hedef fonksiyonunda ortaya çıkabilecek eksilikleri ortadan kaldırmak için başka bir dinçlik atamasının açık ifade doğrusal olarak,

$$d(x) = ah(x) + b$$

şeklinde verilir. Burada a ve b sırası ile ölçekleme ve öteleme katsayılarıdır. EI 'nin en küçükleme olması durumunda a 'nın işareti eksi, en büyüklemede ise artı olarak alınır. Ancak, b 'nin deęeri ortaya çıkacak dinçlik katsayılarının eksi işaretli olmamasını temin etmek içindir. Denklem $d(x)=ah(x) + b$ 'daki işleme doğrusal ölçeklendirme adı da verilir (Şen, 2004).

Teklif edilen başka bir yaklaşımda da dinçlik derecelerinin hedef fonksiyonu deęerlerinin küçükten büyüğe doğru dizilmesindeki mertebelere göre verilmesidir (Şen, 2004).

2.8.3 Başlangıç Toplumu

Daha öncede söylendiği üzere GA'lar en azından 15-20 civarında ve tercihen daha da fazla kromozom ile işe başlar ve bu başlangıç toplumunu ifade eder. Bu toplumda Nbtopy kadar kromozom bulunursa, NbtopyXNgenNhane boyutunda elemanları 0 ve 1'den meydana gelen bir matris yazılır. Buna başlangıç toplumu, BTOP, denilecek olursa,

$$BTOP = \text{Yuvarlat}[\text{Rastgele}(Nbtopy, Nhane)]$$

matrisi elde edilir. Burada, Rastgele “(Nbtopy, Nhane)” komutu NbtopyXNhane boyutunda ve elemanları 0 ile 1 arasında üniform yayılı olan bir matris meydana getirir. Bu tür standart rastgele sayı üretme fonksiyonları her bilgisayar yazılımında vardır. Buradaki “Yuvarlat[]” komutu elde edilen rastgele sayıların en yakın tam sayıya (0 ya da 1) yuvarlatır. Bu matrisin her satırı bir kromozoma karşı gelir (Şen, 2004).

Daha sonra değişkenler hedef fonksiyonuna yerleştirilerek hedef değeri hesaplanır. Geniş çözüm uzayı, GA'da güzel rastgele örneklemelerin elde edilmesini sağlar. Çizelge 3.2'de önceki kısımda söylenen enlem ve boylam genlerinden meydana gelen kromozomların başlangıç toplumu rastgele olarak Nbtopy = 24 tanesi hedef değerleri ile beraber gösterilmiştir. Her gen 7 haneli olunca iki tane karar değişkeni bulunduğundan kromozomlardan 2x7= 14 tane hane bulunur. Aynı çizelgenin 4. kolonunda hedef değerlerinin mertebeleri (küçükten büyüğe doğru sıralama düşünülerek) ve en son sütununda da Denklem;

$$d(x_i) = \frac{h(x_i)}{\sum_{j=1}^N h(x_j)}$$

'a göre hesaplanmış dinçlik dereceleri gösterilmiştir (Şen, 2004).

Çizelge 2.2 Başlangıç Toplumunu Kromozomları (Şen, 2004).

Sıra numarası	Kromozomlar	Hedef değerleri	Mertebe	Dinçlik dereceleri $d(x_i) = \frac{h(x_i)}{\sum_{j=1}^{N_u} h(x_j)}$
1	10111000101010	-11818	6	0.059508
2	00111000101010	-3626	21	0.018258
3	10111000101011	-11819	5	0.059513
4	10111001101010	-11882	4	0.059830
5	10101000101010	- 10793	7	0.054347
6	11111000101010	- 15924	2	0.080183
7	01110000100010	-7202	17	0.036265
8	01010000100010	-5154	19	0.025952
9	01110000101010	-7210	11	0.036305
10	01110110100010	-7586	12	0.038198
11	01110100100010	-7458	15	0.037554
12	01000110100010	-4514	20	0.022730
13	00110110100010	-3490	22	0.017573
14	01110010101010	L-7338	16	0.036950
15	01100100100010	-6434	18	0.032398
16	01110110100010	-7586	14	0.038198
L 17	01110110101110	-7598	13	0.038259
18	01111000110010	-7730	9	0.038923
19	111111010101010	- 16298	1	0.082067
20	01110101011101	-7517	10	0.037851
21	00101110001010	- 2954 $_j$	23	0.014874
22	00101011001010	-2762	24	0.013908
23	10000110101010	-8618	8	0.043395
24	11101110110100	-15284	3	0.078961

GA'lar aynı anda çözüm olabilecek birçok karar noktası ile çalışırlar ki, bu karar noktalarının hepsi ikili sayı sistemine göre kodlanarak toplumu meydana getirir. İlk başta hemen sorulması gerekli olan soru, acaba başlangıç toplumunda kaç tane kromozom (çözüm alternatifi) olsun? Buna cevap ancak yapılan çalışmaların sonucundaki gözlemlerle verilir, ama başlangıçta en azından 10 - 15 tane kromozom almak uygundur. Bunu 50 - 60'a kadar da artırabiliriz (Şen, 2004).

Her GA hazırlığında sorunun karar değişkenlerinin her biri birer ikili sayı dizisi olarak düşünülür. Bütün karar değişkenleri dizisinin ard arda aralıksız biçimde sıralanması ile karar değişkenlerini içeren kromozom elde edilir (Chen, 2006).

Bundan sonraki adım toplumu teşkil eden kromozomların içine konacak sayıların (0 veya 1)

nasıl seçileceğidir? Bunun için tamamen bağımsız uniform rastgele sayı üretme yöntemlerinden yararlanılır. Başlangıç toplumdaki kromozom sayısı N_{btop} olduğuna göre tüm toplumda $N_{gen} \times N_{hane} \times N_{btop}$ kadar haneye ihtiyaç olduğu anlaşılır. İşte bu sayıdaki hane konumlarını 0 veya 1 sayıları ile doldurmak için uniform ve tamamen rastgele sayı üreticilerinden yararlanılır. Eğer sorunun çözümü hakkında hiçbir ipucu yoksa karar uzayının başlangıçta tamamen rastgele olarak bombardımana tutulması uygundur. Varılması istenilen Eİ noktasının yaklaşık konumu hakkında ipuçları bulunuyorsa kromozom değerlerinin böyle bir çözüm noktası etrafında rastgele ama daha dar bir karar uzayında seçilmesi uygundur (Şen, 2004).

GA çalışması sırasında takip edilmesi gerekli kısmi de olsa bir yön yoktur. Noktalar konumlarından yeni konumlarına rastgelelik içeren evrim teorisi işlemleri ile geçerler. Başlangıç toplumunun tesis edilmesi ile daha sonraki aşamalarda ihtimal ilkelerine dayalı üç ardışık GA işlemi vardır. Bunlardan ilki toplumdaki seçme, sonrakiler de o toplum içindeki kromozomların daha da dinçleştirilmesi için çaprazlama ve rakam değiştirme (mutasyon) işlemleridir. Seçme, toplum içindeki çözüm seçenekleri olan kromozomlar arasında bir sonraki aşamada daha da iyiye gidebilecek olanların seçilmesi işlemidir. Bu seçme işlemi esnasında dinçlik dereceleri yüksek olan kromozomların seçilmesine dikkat edilmelidir. Dinçlik derecesi fazla olanın bir sonraki toplumda da rol oynaması ihtimali fazladır. Bu tür seçim işlerinde pastadan en büyük pay en fazla dinçlik derecesine sahip olandan alınabileceğinden, daha sonra da rulet tekerleğini pastaya benzeterek seçimler yapılır (Şen, 2004).

2.8.4 Çaprazlama (Takas)

Eşlerin bir araya gelerek birleşmesi ile yeni ürünlerin ortaya çıkmasına GA'nın çaprazlama aşaması denir. Çaprazlama, kromozomların genlerini birbirleriyle değiştirmelerini sağlayan bir işlemdir (Holland, 1994).

İki dizinin bir araya gelerek karşılıklı gen yapılarının değişimi ile yeni dizilerin oluşumunu sağlayan operatördür (Elmas, 2007).

Çözüm araması sırasında çözüme aday olan iki farklı karar uzayı noktası arasında yaptığı bir çaprazlama ile öncekilerden daha iyi yani en iyi çözüme daha yakın olabilecek iki tane yeni çözüm noktası (karar noktası) ortaya çıkarılır. Bu iki yeni çözüm noktası önceki noktalardan kalıtımı olarak doğar. Böylece çözüm toplumundaki karar noktaları daha iyiye doğru

evrimleşecek biçimde yeni bir toplum meydana getirir. Burada iki çözüm noktasının sayısal değerleri bazı haneleri aralarında çaprazlama yaparak yeni çözüm noktalarının ortaya çıkmasına sebep olurlar (Şen, 2004).

Genetik algoritmanın performansını etkileyen önemli parametrelerden biri olan çaprazlama operatörü doğal popülasyonlardaki çaprazlamaya karşılık gelmektedir. Çoğalma işlemi sonucunda elde edilen yeni popülasyondan rastsal olarak iki dizi seçilmekte ve karşılıklı çaprazlama işlemine tabi tutulmaktadır. Çaprazlama işleminde dizi uzunluğu L olmak üzere, 1 ile L aralığında k tamsayısı rastsal olarak seçilmektedir. Bu tamsayı değeri çaprazlama noktasını belirtir (Taskin ve Emel, 2009).

Hangi şifrelemenin kullanılacağına karar verdikten sonra, çaprazlama veya gen takası adımına geçilir. Çaprazlama, genleri ebeveyn kromozomlardan seçer ve yeni bir döl oluşturur. Bu işlem rastgele çaprazlama noktası (veya noktaları) seçer ve bu noktadan (veya noktalardan) itibaren alt dizilerin yer değişmesiyle yeni kopyaların oluşturulmasını sağlar (Nabiyev, 2005).

Önce çaprazlamaya tabi tutulacak kromozom eşleri rastgele seçilir. Daha sonra bu kromozom eşlerinin hangi genlerden itibaren kesileceği yine rastgele seçim ile belirlenir. Kesilen genler kromozom eşleri arasında değiştirilir. Burada amaç eldeki toplumdaki farklı toplumlar (nesiller) elde etmektir. Bu işlemde çaprazlanacak kromozom eşleri ve bunların hangi genlerden itibaren çaprazlanacağı rastgele olarak belirlenir. Bu konudaki değişik işlemler aşağıdaki dört işlemle yapılır. Okların sağ taraflarında her kromozomun temsil ettiği ondalık sayısal değerleri gösterilmiştir (Şen, 2004).

Uygun bir çaprazlama noktası seçildiğinde, ebeveynlerden uyum değeri daha yüksek diziler elde edilebilmektedir. Fakat uygun çaprazlama noktası hakkında bilgi genellikle bilinmediğinden rastsal bir nokta seçilmektedir. Rastsal bir noktanın seçilmesi bu arama işleminin rastsal olduğu anlamına gelmemektedir. L uzunluğundaki iki ebeveyn diziyeye en basit çaprazlama yöntemi olan tek noktalı çaprazlama uygulandığında, arama uzayında en fazla $2*(L-1)$ tane farklı dizi elde edilebilmektedir. Fakat arama uzayında toplam 2^L tane dizi mevcuttur. Rastsal bir çaprazlama noktasıyla üretilen yavru diziler ebeveynlerin uyum değeri yüksek alt dizi kombinasyonlarından oluşabilmektedirler. Tersine durum da söz konusu olabilmektedir. Fakat her iki durumda da uyum değeri yüksek dizilere ulaşılmaktadır. Çaprazlama operatörü ile uyum değeri yüksek diziler oluşturulursa bu dizilerden bir sonraki kuşakta eşleşme havuzunda daha fazla kopyası bulunacaktır. Çaprazlama operatörü ile uyum

değeri yüksek diziler oluşturulmadığı durumda ise bu diziler bir sonraki kuşağa kadar hayatta kalamayacaklardır. Çünkü çoğalma operatörü ile eşleşme havuzuna bu diziler seçilmeyecektir (Taskin ve Emel, 2009).

2.8.4.1 Tek Noktalı Çaprazlama

Bir çaprazlama noktası seçilir. Bu noktaya kadar olan bitler (genler) birinci ebeveynden, geriye kalanlar ise diğerinden alınarak yeni bir kromozom oluşturulur (Nabiyev, 2005).

Kromozom-1: 1 1 0 1 1 \ 0 0 1 0 0 1 1 0 1 1 0

Kromozom-2: 1 1 0 1 1 \ 1 1 0 0 0 0 1 1 1 1 0

Döl-1: 1 1 0 1 1 \ 1 1 0 0 0 0 1 1 1 1 0

Döl-2: 1 1 0 1 1 \ 0 0 1 0 0 1 1 0 1 1 0

Öncelikle iki kromozomun ilk ve son rakamları arasında bir yerde rastgele olarak çapraz geçiş noktası tespit edilir. Bir kromozomun çapraz geçiş noktasının solunda bulunan hanelerdeki tüm rakamlar olduğu gibi diğer kromozom ile yer değiştirilir. Önce ilk iki esas kromozomdan birindeki çapraz noktanın solundaki tüm haneler yeni çocuğun kromozomuna geçer. Benzer şekilde diğer kromozomunun çapraz geçiş noktasının solundakiler de diğer yeni doğacak kromozoma geçer. Bundan sonra birinci esas kromozom çapraz geçiş noktasının sağındaki haneler ikinci yeni doğanın önceki hanelerinin sağına geçer. Böylece, ilk yeni doğanın kromozomu tamamlanmıştır (Şen, 2004).

Benzer olarak ikinci esas kromozomun çapraz geçiş noktası sağındaki haneler de birinci yeni doğanın sağına yerleşir. Böylece, yeni doğan iki kromozom kısmi olarak öncekilerin hanelerini ihtiva eder. Esas kromozomlar toplam olarak Nkötü tane yeni doğuşa sebep olmalıdır ki toplum büyüklüğü sabit kalsın ($N_{top} = \text{sabit}$). Buna basit veya tek çapraz geçiş noktalı doğumlar adı verilir. Bu işlemde kromozomlar rastgele bir yerinden kesilir ve sonra ilgili genler ile yer değiştirilir (Şen, 2004).

1↓	↑	0↓	↑	0↓	↑	/	1	0	1	1	→	75
1	1	1					0	1	0	0	→	116

Yukarıdaki kromozom ikilisi 3. haneden hemen sonra kesilmiş ve koyu genler yer değiştirilerek aşağıdaki yeni kromozomlar elde edilmiştir. Yukarıdaki izahlardan anlaşılacağı üzere her bir kromozoma bir sayı karşı gelir ve kromozomlardaki genlerden bir tanesinin bile

değişmesi ile bu sayılar da değişir.

$$\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 0 & 1 & 1 & \longrightarrow & 123 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & \longrightarrow & 68 \end{array}$$

Tek kesimli çaprazlama ile birbirinden farklı sadece iki yeni kromozom elde edilebilir.

Seçim sürecinden gelen diziler rastsal olarak eşlenirler. L dizi uzunluğu olmak üzere, seçilen her dizi çifti için, [1..... L-1] aralığında rastsal olarak bir tamsayı üretilir. k tamsayısı da çaprazlama noktasını göstermektedir. Çaprazlama noktası belirlendikten sonra, çaprazlama noktasından sonra gelen genler yer değiştirir. Tek noktalı çaprazlamanın uygulanabilmesi için her iki dizinin de aynı gen uzunluğunda olması gerekmektedir. Tek noktalı çaprazlama bir örnekle aşağıda açıklanmaktadır (Taskin ve Emel, 2009).

$$A1 = 0 \ 1 \ 1 \ 0 \ 1 \qquad A2 = 1 \ 1 \ 0 \ 0 \ 0$$

Çaprazlama noktasının belirlenmesi amacıyla, bir ile dört arasında rastsal bir sayı seçilir ve bu sayı (k) dört olarak elde edilsin.

$$A1 = 0 \ 1 \ 1 \ 0/1 \qquad A2 = 1 \ 1 \ 0 \ 0/0$$

Dizilerin beşinci genler yer değiştirir ve böylece iki yeni dizi oluşturulmuş olur.

$$A1 \text{ (yeni)} = 01100 \qquad A2 \text{ (yeni)} = 110 \ 0 \ 1$$

2.8.4.2 İki Noktalı Çaprazlama

Burada iki kırılma noktası seçilir. İlk noktaya kadar olan bitler birinci ebeveynden iki nokta arasındaki bitler ikinci ebeveynden, kalanlar ise tekrar birinci ebeveynden yeni kromozoma kopyalanır (Nabiyev, 2005).

$$1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1 + 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1 = 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 1$$

İki noktalı çaprazlamada her kromozom üç parçaya bölünür. Bu parçalardan karşılıklı her ikisinin çaprazlama yer değiştirmesi ile birbirinden farklı 6 tane yeni kromozom elde edilir. Aşağıda iki noktalı çaprazlamaya maruz bırakılan kromozomlarda düz, italik ve koyu kısımlar olmak üzere üç parça gösterilmiştir. (Şen, 2004).

$$000010111000110$$

$$001111100001001$$

Çaprazlamanın sırası ile düz, italik ve koyu parçalar arasında yapılması ile birbirinden farklı

kromozomlar düz kısımlar arasında

001010111000110

000111100001001

italik kısımlar arasında

000111100000110

001010111001001

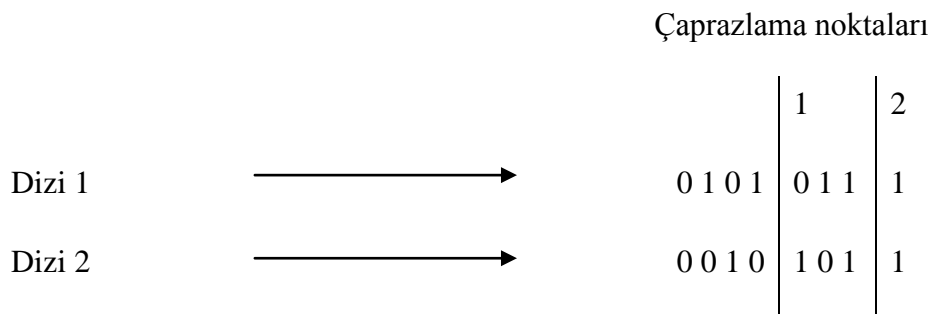
ve son olarak da koyu kısımlar arasında

000010111001001

001111100000110

şeklinde olmak üzere 6 tanedir. Yeni kromozomların tümünün kullanılması gerekli değildir. Bu seçimde ya düz, ya italik veya koyu parçaların değiştirilmesinden 6 yeni kromozomdan istenilen kadar rastgele seçimi ile topluma katılabilir.

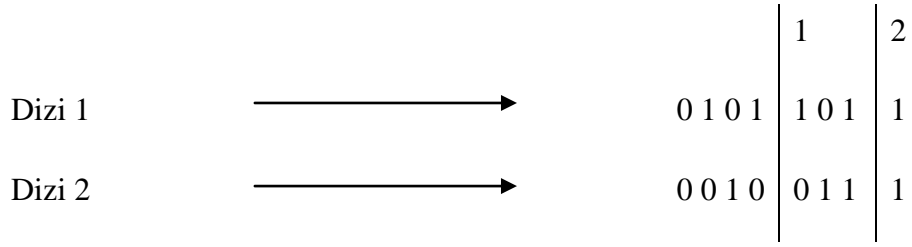
İki noktalı çaprazlama yönteminde, iki tane çaprazlama noktası rastsal olarak seçilmektedir. Çaprazlama işlemi, seçilen iki nokta arasındaki genlerin yerlerinin değişimini içerir. Bu yöntemin genişletilmiş hali çok noktalı çaprazlama operatörüdür. Bu genişletilmiş durumun vardığı son nokta ise düzgün (uniform) çaprazlama operatörüdür (Chen, 2006). Düzgün çaprazlama operatöründe, her ebeveyndeki her gen 0,5 olasılıkla seçilir. Aşağıda iki noktalı çaprazlama yöntemini anlatan bir örnek verilmektedir (Taskin ve Emel, 2009).



(a)

Çaprazlama işleminden sonra:

Çaprazlama noktaları

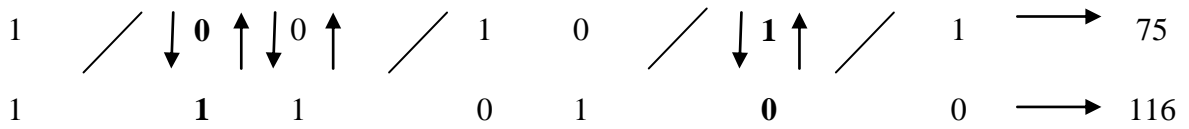


(b)

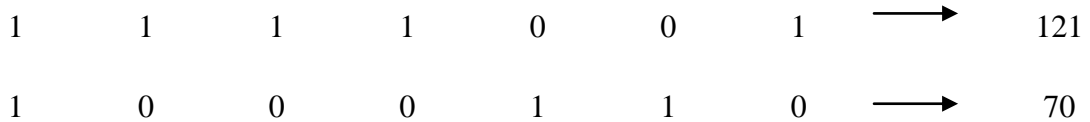
Şekil 3.1. (a) İki Noktalı Çaprazlama İşlemi Öncesi (b) İki Noktalı Çaprazlama İşlemi Sonrası

2.8.4.3 Çok Noktalı Çaprazlama

Kromozomların çapraz olarak ikiden daha fazla yerden rastgele kesilerek genlerin yer değiştirilmesi ile sağlanır (Şen, 2004).



Bu çaprazlama sonucunda yeni kromozom ve sayılar aşağıdaki gibidir.



Burada da karşılıklı parçalar arasında çaprazlama ile çok sayıda yeni kromozom elde edilir. Eğer n parça sayısını gösteriyorsa n^2 kadar yeni kromozom elde edilebilir.

2.8.4.4 Tek Biçimli-Üniform Çaprazlama

Bu çaprazlama biçiminde bitler rastgele şekilde her iki ebeveynden yeni kromozoma kopyalanmaktadır (Nabiyev, 2005). Rastgele hanelerin iki kromozom arasında yer değiştirmesi ile bu işlem gerçekleştirilir.

$$11001011 + 11011101 = 11011111$$

Burada her hane için yazı tura atılır. Örneğin yazı gelirse çaprazlama yapılsın, tura gelirse

çaprazlama yapılmasın kuralına uyulur. Koyu renkli genler için yazı geldiğinden bu genler için çaprazlama uygulanarak kromozomların yeni şekilleri şöyle olur (Şen, 2004).

1	↓	0↑	0	↓	1↑	0	1	↓	↑	→	75
1		1	1		0	1	0		0	→	116
1		1	0	0	0	0	1	0		→	98
1		0	1	1	1	1	0	1		→	93

Tekdüze çaprazlamada ilk iki kromozom hanelerine rastgele bakılır. Rastgele olan hane belirlendikten sonra karşılıklı olarak rakamlar değiştirilir. Bunun için önce rastgele bir altlık teşkil edilir. Bu altlık rastgele olarak 0 ve 1'leri ihtiva eden ve başlangıç kromozomları uzunluğunda olan bir dizidir. Altlıkta bir haneye karşı gelen rakam 0 ise, buna karşı gelen kromozom hanesindeki rakam birinci kromozoma geçirilir. Altlıktaki rakam 1 ise buna karşı gelen kromozomdaki rakam ikinci yeni kromozoma geçirilir. Buna misal aşağıda verilmiştir (Şen, 2004).

00101011000110

01111100001100

Altlık 00110110001110

00111101001100

01101010000110

Tekdüze çaprazlama aşağıda verilen altlık için bir kesimli çaprazlama olur.

Altlık 00000011111111

Benzer olarak iki noktalı çaprazlamaya sebep olabilecek altlıkta

Altlık 00000011110000

şeklinde dir. Buradan da tekdüze çaprazlamanın, tekli, ikili, ve diğer çaprazlamaların bir genellemesi olduğu anlaşılır. Yapılan çok sayıdaki araştırmada tekdüze çaprazlamanın tekli veya ikili çaprazlamadan daha etkin olduğu görülmüştür. Aslında daha özel olarak iki noktalı çaprazlamanın da tek noktalıdan daha etkin olduğu sonucuna varılmıştır (Şen, 2004).

Çoklu çaprazlamada sadece rastgele seçilmiş olan haneler arasında karşılıklı yer değiştirmeler yapılmıştır. Halbuki tekdüze çaprazlama işleminde her hane çaprazlamaya aynı derecede açık bırakılır (Şen, 2004).

Tekdüze çaprazlama ile yeni kromozomların iki ara kromozomun 1 rakamları yerine ilk kromozomdan karşı gelen rakamlar, 0 rakamlarına da ikinci kromozomdan karşı gelen değerler alınarak teşkil edilir. İkinci yeni kromozomun teşkili de ara kromozomun 1'lerine ikinci kromozomun rakamları, 0'larına da birinci kromozomdaki rakamlar alınarak yapılır (Şen, 2004).

2.8.4.5 Aritmetik Çaprazlama

Yeni bir kromozom oluşturmak için değişik aritmetik işlemler uygulanır (Nabiyev, 2005).

$$11001011 + 11011111 = 11001001 \text{ (AND) VE işlemi}$$

2.8.4.6 Tersleme

Önceki işlemlerden tamamen farklı olan tersleme kromozomun soldan sağa okunacak yerde sağdan sola doğru gen dizisi ile okunması durumunda ortaya çıkan kromozomdur. Burada da önceki kromozomun temsil ettiği sayıdan tamamen farklı bir sayı elde edilir (Şen, 2004).

Örneğin

$$1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad \longrightarrow \quad 98$$

gibi bir kromozom tersine çevrilirse aşağıdaki yeni kromozom elde edilir.

$$0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad \longrightarrow \quad 35$$

2.8.4.7 Karıştırmalı Çaprazlama

Bunlardan birincisi tek noktalı çaprazlamada kromozomların ilk kısımları aralarında aynı sıra dahilinde rakamlarla değiştirilecek yerde rakamlar tamamen bağımsız ve tekdüze bir rastgele sayı üreticisi ile yeni baştan 0 veya 1 rakamları ile doldurulur ama esas kromozomların ikinci kısımları aynen kalırsa buna karıştırmalı çaprazlama denir. Böylece yeni kromozomların başlangıç kısımları birbirlerinden tamamen bağımsızdır (Şen, 2004). Örneğin,

$$\begin{array}{cccc|cccc} 1 & 0 & 0 & / & 1 & 0 & 1 & 1 & \longrightarrow & 75 \\ 1 & 1 & 1 & & 0 & 1 & 0 & 0 & \longrightarrow & 116 \end{array}$$

kromozomlarının karıştırmalı çaprazlaması sonucunda

0	1	0	1	0	1	1	→	43
1	0	1	0	1	0	0	→	84

elde edilir.

2.8.4.8 PMX Çaprazlama

Goldberg ve Lingle tarafından önerilen PMX (Partially Matched Crossover) çaprazlama yöntemi, sıra temelli çaprazlama yöntemlerinden birisidir. Bu tip yöntemler gezgin satıcı problemi ve araç rotalama problemi gibi problemlerde yaygın olarak kullanılmaktadırlar. Bu yöntemde iki ebeveyn diziden yalnızca bir adet yavru dizi elde edilmektedir. Ebeveynlerden birinin rastsal olarak seçilen bir alt dizisi, oluşturulacak yavru diziyeye aynı konumlara olmak şartıyla kopyalanır. Kalan konumlar ise henüz kullanılmamış değerlerle doldurulur. Bu yöntem bir örnekle aşağıdaki gibi anlatılabilir (Taskin ve Emel, 2009).

Aşağıda iki ebeveyn dizi verilmektedir. Bu yöntemde rastsal olarak bir alt dizinin belirlenmesi gerekmektedir. Bu alt dizi birinci ebeveyn'den olsun ve 4 6 3 olarak belirlensin. Bu alt dizi P2 dizisindeki 1 7 2 alt dizisine karşılık gelmektedir.

P1	(1 2 4 6 3 7 5 8)
P2	(5 4 1 7 2 6 8 3)

Yavru dizinin 3., 4., ve 5. konumları belirlenmiştir. İkinci dizide 4 6 ve 3 sayılarının bulunduğu konumlar ise yıldız (*) işaretiyle işaretlenir.

$$P' \quad (5 * 4 6 3 * 8 *)$$

Dört(4), bir değerine; 6 yedi değerine; ve 3 de iki değerine karşılık geldiğinden yeni dizinin * işaretli konumlarına bu sayılar yerleştirilir. Sonuç olarak bir adet yavru dizi elde edilmiştir.

$$P' \quad (5 1 4 6 3 7 8 2)$$

2.8.4.9 OX Çaprazlama

Sıra temelli diğer bir yöntem olan OX (Order Crossover) çaprazlama yöntemi, 1980'li yıllarda Davis tarafından gezgin satıcı problemleri için önerilmiş, daha sonra da Oliver ve arkadaşları tarafından geliştirilmiştir. Bu yöntemde amaç, verilen iki ebeveyn diziden bir tanesinde

izlenen sırayı koruyup, diğer ebeveyn diziden bir alt dizi seçerek iki yavru dizi oluşturmaktır (Taskin ve Emel, 2009).

Aşağıda iki tane ebeveyn dizi verilmektedir. Amaç her dizide seçilen alt dizilerin korunarak yavru diziler elde edilmesidir. Bu alt diziler birinci ebeveyn dizide 7 9 2 olarak belirlenmiştir. Bu konumların diğer dizideki karşılığı da 2 6 3 konumlarıdır.

P1 (1 3 5 / 7 9 2 / 4 6 8)

P2 (1 5 9 / 2 6 3 / 7 4 8)

P1' (* * * / 7 9 2 / * * *)

P2' (* * * / 2 6 3 / * * *)

Birinci ebeveynde ikinci kesim noktasından başladığında sıralama şöyle olmaktadır:

4 6 8 1 3 5 7 9 2

P2' dizisinde bulunan 2 6 3 alt dizisi bu diziden çıkartılır. Sonuç olarak 4 8 1 5 7 9 elde edilir. Bu kalan dizi P2' dizisine ikinci kesim noktasından sonra eklenir. Aynı işlem P1' dizisi için de gerçekleştirilerek iki tane yavru dizi elde edilir. Yavru diziler şunlardır:

P1' (5 6 3 7 9 2 4 8 1)

P2' (5 7 9 2 6 3 4 8 1)

2.8.4.10 CX Çaprazlama

CX (Cyde Crossover) çaprazlama yöntemi de diğer iki yöntem gibi sıra temelli çaprazlama yöntemi olup Oliver adlı araştırmacı tarafından önerilmiştir. Bu yöntemde yavru dizilerin her bir konumu sırayla birinci ebeveyn diziden ve ikinci ebeveyn diziden gelmektedir. Bir örnek verilerek yöntem aşağıda açıklanmıştır (Taskin ve Emel, 2009).

P1 8 2 1 7 4 5 10 6 3

P2 1 2 3 4 5 6 7 8 9 10

Bu yöntemde çaprazlama noktaları seçilmeyip, ilk ebeveyn dizinin en sol konumundan işleme başlanmaktadır. Buna göre 9 değerinin bulunduğu konum seçilir.

P1' 9 * * * * * * * *

Her konum iki ebeveynden birinden seçileceği için, 9 değerinin bulunduğu konumun karşılığı

olan 1 değeri C dizisinden alınır ve yavru diziye yerleştirilir.

$$P1' \quad 9 * * 1$$

Aynı şekilde, 1 değerinin bulunduğu konumun karşılığı olan 4 değeri seçilerek yavru diziye yerleştirilir. Bu durum devam ettiğinde aşağıdaki örüntü elde edilmiş olur.

$$P1' \quad 9 * * 1 * 4 * * 6 *$$

6 değerinin bulunduğu konumun karşılığı 9 değeridir. Ancak 9 değerinin bulunduğu konum seçildiği için bir daha seçilemez. Başka bir ifadeyle döngü tamamlanmıştır. Birinci döngü sonunda, kalan yıldız işaretli konumlar ise diğer dizi üyeleriyle yer değiştirir. Sonuç olarak aşağıdaki diziler elde edilmiştir.

$$P1' \quad 9 \ 2 \ 3 \ 1 \ 5 \ 4 \ 7 \ 8 \ 6 \ 10$$

$$P2' \quad 1 \ 8 \ 2 \ 4 \ 7 \ 6 \ 5 \ 10 \ 9 \ 3$$

2.8.4.11 LOX Çaprazlama

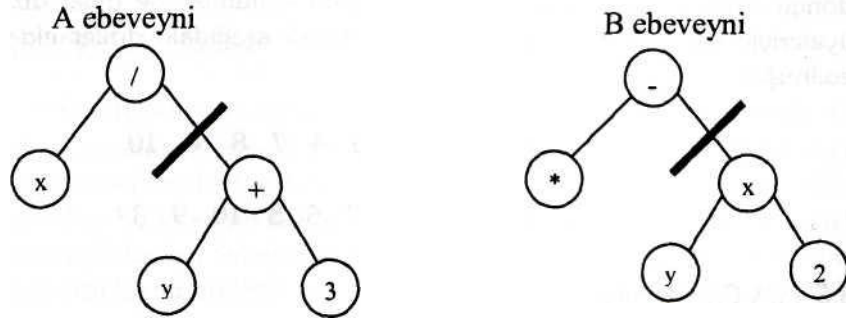
Falkenauer ve Bouffouix (1991) tarafından iş çizelgeleme problemleri için önerilen LOX (Linear Order Crossover), OX çaprazlamanın farklı bir biçimidir. LOX, her iki ebeveyn çiftten gelen dizi sırasını korumak amacıyla tasarlanmıştır. Başka bir ifadeyle, LOX çaprazlamaya ebeveyn çiftleri üç bölgeye ayıran iki noktalı çaprazlama yöntemi de denilebilir. Aşağıda LOX çaprazlama ile ilgili bir örnek verilmiştir (Taskin ve Emel, 2009):

Çizelge 2.3 LOX Çaprazlama Örneği

Ebeveyn	a	b	c	d	e	f	g	h	i	j
Ebeveyn	c	b	a	g	h	i	j	f	d	e
Yavru 1	a	b	c	g	h	i	d	e	{	i
Yavru 2	c	b	a	d	e	f	g	h	i	i

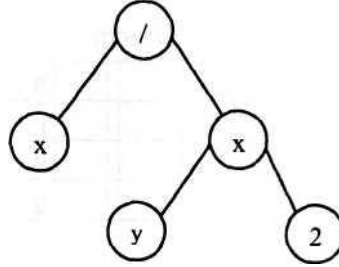
2.8.4.12 Ağaç Çaprazlama

Bu yöntemde iki ebeveyn için çaprazlama noktaları belirlenir. Birinci ebeveyndeki çaprazlama noktasından sonra gelen kısım atılır. İkinci ebeveyndeki çaprazlama noktasından sonra gelen kısım birinci ebeveyne eklenir ve yavru oluşturulur. Bu durum bir örnekle aşağıdaki gibi açıklanabilirdi (Taskin ve Emel, 2009);



(a)

Burada çaprazlama noktası siyah çizgi ile belirtilmiştir. Çaprazlama noktasından sonra gelen kısım atılarak yerine ikinci ebeveyndeki çaprazlama noktasından sonra gelen kısım eklenecektir.



(b)

Şekil 2.5 (a) Ağaç Yapısında Ebeveyn Diziler (b) Ağaç Yapısında Yavru Dizi (Taskin ve Emel, 2009)

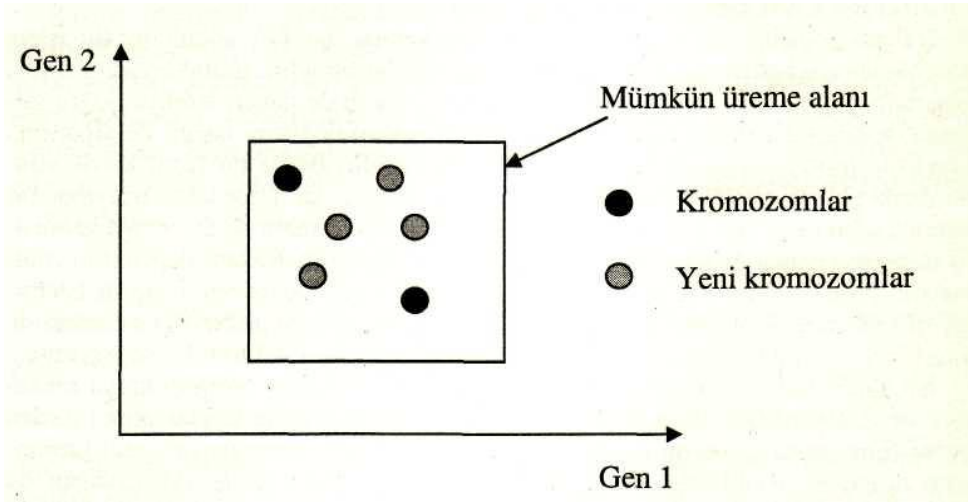
2.8.4.13 Ara Birleşmeli Çaprazlama

Eğer bir kromozoma yakın başka kromozomlar üretilmek isteniyorsa bu taktirde iki kromozomdan uygun kısımların alınarak aşağıdaki denkleme göre bir araya getirilmesi ile yeni ve istenen türde kromozomlar üretilebilir (Şen, 2004). İki ana kromozom K1 ve K2 ise

bunların bir “ α ” aracılığı ile karıştırılması sonucunda yeni üye,

$$K_y = K_1 \alpha (K_2 - K_1)$$

şeklinde elde edilir. Yapılan çalışmalara göre - $0.25 < \alpha < 1.25$ arasında rastgele bir sayı olarak seçilmelidir. Birleştirme esnasında Denklem her iki kromozomun karşılıklı genleri için uygulanırken yeniden α değeri rastgele olarak seçilmelidir. Geometrik olarak bu şekilde üretilen yeni kromozomların karar uzayındaki konumları ana kromozomlardan fazla uzakta olamaz (Şekil 2.6). Bu yakınlıklar α değerine göre değişir (Şen, 2004).



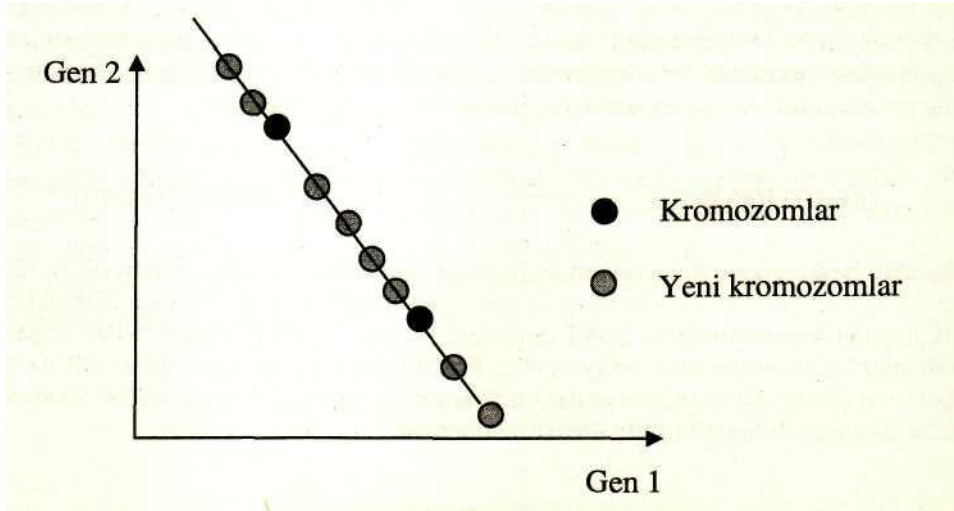
Şekil 2.6 Ara Birleşmeli Çaprazlama karar uzayındaki durumu (Şen, 2004).

2.8.4.14 Doğrusal Birleşmeli Çaprazlama

Bu önceki çaprazlamaya benzer ama tüm birleşme durumları sabit bir “ α ” değerinin kullanılması ile elde edilir. Şekil 2.7'de gösterildiği üzere ilk kromozomları karar uzayında birleştiren doğrunun üzerinde yeni kromozomlar elde edilir (Şen, 2004). Bu doğru denklemi

$$K_y = K_1 + \alpha(K_2 - K_1)$$

şeklinde ifade edilir.



Şekil 2.7 Doğrusal Birleşim Üyeleri (Şen, 2004).

2.8.5 Mutasyon

Mutasyon kromozom üzerinde rastgele bir bit değişikliğini gerçekleştiren operatördür (Melanie, 1999).

Rakam değişikliği (mutasyon) kromozomun bir hanesindeki 0 değerini 1 veya 1 değerini 0 yapma işlemidir. Bu işlem kromozomun bir hanesinde yapılabileceği gibi birden fazla hanede de yapılabilir. Böylece ortaya bir rakamın bile değişmesi ile çok farklı bir sayı çıkar. Rakam değişimleri ile GA'lar ikinci türden hedef fonksiyonu yüzeyini araştırır (Şen, 2004). Böylece başlangıçta bulunmayan yeni kromozom türleri elde edilir.

Mutasyon işlemi, problemin popülasyondaki çözümlerinin yerel optimuma düşmesini engellemek için kullanılır. Mutasyon yeni dölü rastgele değiştirir. İkili şifreleme için, rastgele seçilen birkaç bit 0'dan 1'e veya 1'den 0'a değiştirilebilir, yani seçilen bitler terslenir (Nabiyev, 2005).

Basit genetik algoritmada mutasyon, dizideki herhangi bir konumun değerinin rastsal olarak değiştirilmesidir. İkili kodlama ile kodlanmış dizilerde mutasyon, 1 değerinin 0'a, 0 değerinin de 1'e çevrilmesi demektir (Taskin ve Emel, 2009).

Eğer bir kromozoma birli bir rakam değişimi uygulanıyorsa, bu takdirde '1' sayısı '0' sayısına veya aksine dönüştürülür. Rakam değişiminin yapılacağı hane konumu $N_{top} \times N_{hanex} \times N_{gen}$ toplam hane toplumundan rastgele olarak seçilir (Şen, 2004). Rakam değişimiyle nokta sayısının artırılması GA'nın doğal olarak taradığı çözüm uzayının yani algoritmanın tarama alanında dokunulamayan yerlerin de incelenmesi sağlanmış olur.

Dizilerdeki genler çaprazlamada olduğu gibi belirli bir mutasyon oranıyla mutasyona uğrattırılır. Problemlerde, mutasyon oranları genellikle düşük tutulmaktadır. Bunun nedeni, çaprazlama sonucunda elde edilen uyum değeri yüksek dizileri kaybetmemektir (Taskin ve Emel, 2009). GA işlemleri sırasında rakam değişimi işlemi nadir olarak yapıldığından düşük ihtimaller göz önünde tutularak uygulanır. Pratik çalışmalar bu ihtimal değerinin 0.01 ile 0.001 arasında değişmesi sonucuna varılmıştır (Şen, 2004).

Değişim oranının yüksek olması çözüm uzayım çok genişleterek sistem çözümünün yanlış yerlerde aranmasına neden olur. Bu nedenler değişim operatörünün uygulanma olasılığı %0.5 - %15 arasında değişmektedir (Elmas, 2007).

Genelde, her iterasyonda hanelerin %1'i ile %0,1'i miktarında rakam değişimi yapılır. En son iterasyonda artık rakam değişimi yapılmaz. Genel olarak, en iyi çözümlenmelerde de rakam değişikliğine müsaade edilmez. En iyi kromozomların dışındakiler arasından mesela %5 yani 0.05 miktarında rakamların değiştirilmesi yapılır. Daha önceki bölümlerde toplumun iki boyutlu matris şeklinde gösterildiğinden söz edilmiştir. Bu durumda yukarıda verilen misalde yedi çift rastgele sayı üretmelidir ki, bunlarla rakam değişikliği yapılacak yerlerin satır ve sütunları tespit edilebilsin. Mesela, ilk rastgele çift (4,11) ise 4-üncü sıra ve 11-inci satırdaki sayının rakam değişikliğine uğratılması gereklidir. Bu konumda bulunan '0' rakamı '1' rakamına çevrilir. Bu durum aşağıdaki kromozomda koyu haneli olarak gösterilmiştir (Şen, 2004).

0001011000**0**010 -----► 0001011000**1**010

Bundan başka 6 tane daha rakam değişimi gereklidir. Bunların konumu (9,3); (2,2); (2,1); (5,14); (8,10) ve (5,8) rastgele sayı çifti olabilir. Yapılan rakam değişikliklerinin çoğu kromozomların hedef değerlerini artırır. Zaman zaman hedef değer azalmaları gibi durumlarda ortaya çıkar.

Bir GA işlemi sırasında rakam değiştirme işlemi incelenirken iki durum göz önünde tutulmalıdır. Bunlar rakam değiştirme türleri ve oranıdır. Rakam değiştirme ne kadar olmalıdır? Bir hanedeki rakamlardan bir tanesinin değiştirilmesi buna karşı gelen değişken değerinin % 50 değiştirilmesine karşı gelebilir. Rakam değişimi yapılan bir rakamın beklenen rakam değişimi (Şen, 2004),

$$E(G) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2^i}$$

formülü ile hesaplanır. Buna göre 4 haneli bir gende 0.25 oranının kullanılması ile rakam değişimi olarak $(1/4)(0.5+0.25+0.0125+0.0675) = 0.23563$ kadar beklenebilir. Diğer taraftan, 8 hanelik bir gende beklenen 2 rakam değiştirmedeki değişme miktarı $(1/8)(1/2+1/22+1/23+1/24+1/25+1/26+1/27+1/28) = 0.24902$ kadar olur. Genel olarak, bir gendeki hane sayısına bağlı olarak rakam değiştirmenin değişken değerinde çok fazla değişikliğe sebep olmaz. GA rakam değişim oranının seçimine çok hassastır.

Literatürde ayrıca sıra temelli mutasyon yöntemlerine de rastlanmaktadır. Bu yöntemler arasında, tersinme (inversion), yerleştirme (insertion), karşılıklı değişim (reciprocal exchange) ve kaydırma (displacement) bulunmaktadır. Bu yöntemler birleştiğinde optimizasyon problemlerinin çözümleri için uygun yöntemlerdir (Taskin ve Emel, 2009).

Bu yöntemlerden kaydırma yöntemi, bir alt turun seçilerek rastsal bir konuma yerleştirilmesini içermektedir. Karşılıklı değişim yönteminde ise rastsal olarak iki gen seçilmekte ve bu genler birbirleriyle yer değiştirmektedir. Yerleştirme yöntemi, rastsal olarak bir genin seçilerek rastsal bir konuma yerleştirilmesidir. Tersinme yönteminde ise dizi boyunca iki nokta seçilmekte ve seçilen iki nokta arasındaki konumlar yer değiştirmektedir. Bu yöntem için aşağıda bir örnek verilmiştir (Taskin ve Emel, 2009).

Mutasyon da çaprazlama gibi şifrelemeye bağlıdır (Nabiyev, 2005). İleriki aşamada ikili kodlanmış kromozomlar için çaprazlama ile birlikte mutasyon işlemleri grafiksel olarak gösterilmiştir.



Pe

mutasyon kodlanmış kromozomlarda çaprazlama yapılırken bir çaprazlama noktası seçilir. Birinci ebeveynden bu noktaya kadar alınır, yeni kromozom kalan kısmını ise diğer ebeveynden kopyalanarak oluşturulur.

$$(123456789) + (453689721) \Rightarrow (123459721) = (123459768)$$

Burada mutasyon işlemi iki gen seçilmesi ve bunların yerlerinin değiştirilmesiyle gerçekleştirilir.

$$(123456897) \Rightarrow (183456297)$$



De

ğer kodlanmış kromozomlarda çaprazlama yapılırken ikili kodlanmış kromozomlarda yer alan tüm takas türleri burada da uygulanması mümkündür. Burada mutasyon,

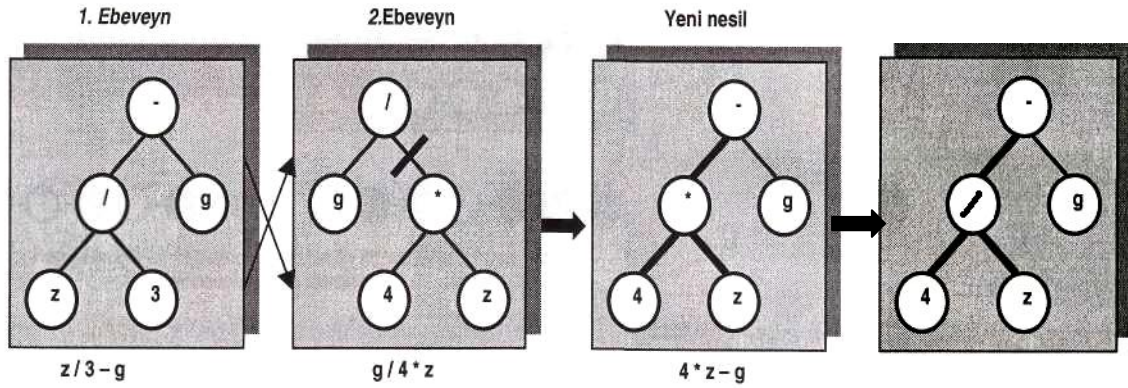
seçilen değerlerden küçük bir sayı çıkartılarak ya da eklenerek yeni kromozom oluşturulması ile gerçekleştirilir.

$$(1.29 \ 5.68 \ \mathbf{2.86} \ \mathbf{4.11} \ 5.55) \Rightarrow (1.29 \ 5.68 \ \mathbf{2.73} \ \mathbf{4.22} \ 5.55)$$



A

ğaç kodlanmış kromozomlarda ise çaprazlama, iki ebeveyn seçilir ve her iki ebeveyninden birer takas noktası belirlenir parçalama işlemi uygulanır. Belirlenen bu noktaların altındaki kısımlar, aralarında değiştirilerek yeni bireyler oluşturulur. Mutasyon: Seçilen düğümlerdeki numaralar veya işlemler değiştirilir.



Sekil 2.8 Ağaç Kodlamada Çaprazlama Operatörünün ve Mutasyon İşleminin Kullanılması

Literatürde birçok değişim operatörü görülebilir. Önemli olan değişim operatörünün uygulandığı yer ve uygulama biçimine göre GA'nın kullanım şekli probleme göre değişebilir. Aşağıda kullanılması muhtemel değişim operatörlerinden bir kaç gösterilmiştir.



K

omşu iki işi değiştirme; burada rastgele seçilen iki komşu gen yer değiştirmesi şeklinde,

Değişim Öncesi

Değişim Sonrası

1. Çocuk 2571943 => 1.Çocuk257**9**1432



Ke

yfi iki işi değiştirme; burada rastgele seçilen iki gen yer değiştirmesi şeklinde,

Değişim Öncesi Değişim Sonrası

1. Çocuk 25719432 => 1.Çocuk 24719532



Ke

yfi üç iş değiştirme; burada rastgele seçilen üç gen rasgele yer değiştirmesi şeklinde,

Değişim Öncesi Değişim Sonrası

1. Çocuk 25719432=>1.Çocuk 24719532



Ar

aya gen ekleyerek değiştirme; burada rastgele seçilen genin rastgele sayıda sağa veya sola kaydırılması şeklinde,

Değişim Öncesi Değişim Sonrası

1. Çocuk 25719432 => 1.Çocuk 235719432

2.8.6 Uygunluk Fonksiyonu

Hedef fonksiyonuna sahip olmak, oluşturulmuş kromozomların karar uzayının değişik yerlerinde buldukları noktanın aranılan sonuca ne kadar yakın olup olmadıkları konusunda karar vermek amaçlı kullanılır (Behzadi, Alesheikh). Mutlaka bir $h(x)$ gerek vardır. Aranılan sonucun yönüne göre yani en küçükleme veya en büyükleme olmasına göre hedef fonksiyonunun küçük veya büyük değerlerine bulmak amaçlanır.

GA toplumdaki her bir kromozomun (üyenin) göreceli dinçliklerinin hesaplanabilmesi için bunların mutlak (sadece artı değerli) dinçlik değerlerine yani hedef fonksiyonunda aldıkları değerlere ihtiyaç vardır. Hedef fonksiyonundan dinçlik derecelerini verecek bir fonksiyonun, $d(x)$, bir şekilde elde edilmesi gereklidir (Şen, 2004). Bu iki fonksiyon arasındaki kapalı ilişkiyi,

$$d(x_i) = F[h(x_i)]$$

şeklinde gösterebiliriz. Burada, F dönüşümü o şekilde olmalıdır ki sonuçta dinçlik derecelerini veren fonksiyonun değerleri tüm karar değişkenleri için mutlaka artı işaretli olmalıdır. Bu son ifadenin açık şekli, $d(x_i)$, i -inci karar değişkeninin dinçlik değeri olmak

üzere,

$$d(x_i) = \frac{h(x_i)}{\sum_{j=1}^{N_u} h(x_j)}$$

şeklinde hesaplanabilir. Burada bakılması gereken nokta hedef fonksiyonu değerlerinin tümünün artı işaretli olmasıdır. Buna bağlı olarak değişkenlerin toplum içindeki bulunma ihtimalini dinçlik derecesi ortaya koyar.

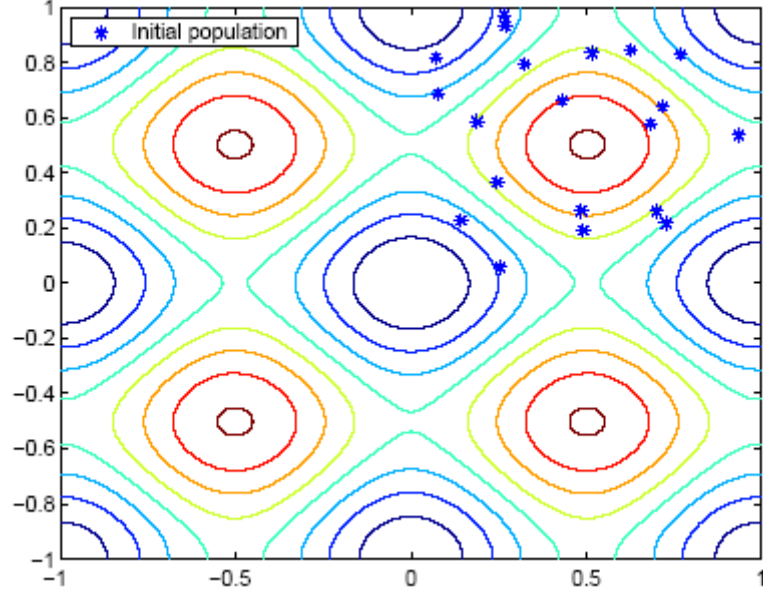
2.9 Genetik Algoritmanın Çalışma Şekli

GA'nın nasıl çalıştığını kısaca özetlemeye çalışırsak; (Matlab GATool, 2004)

1. GA, rassal olarak ilk popülasyonu oluşturarak başlar.
2. Algoritma düzenli olarak yeni popülasyonu ya da yeni jenerasyonu oluşturur. Her basamakta, algoritma var olan popülasyon içindeki bireyleri kullanarak yeni jenerasyonu oluşturur. Ve yeni bir jenerasyon oluşturmak için sistemde şöyle devam eder;
 - a. Popülasyondaki her bir üyeye uygunluk değerini hesaplayarak bir skor verir.
 - b. Verilen skorları bir terazi şeklinde tartarak kullanılabilir bir aralık oluşturur.
 - c. Uygunluklarına göre ebeveynler belirlenir.
 - d. Çocuklar, ebeveynlerden oluşturulur. Çocuklar rassal olarak değişiklik geçirirler ve bir ebeveyn kullanılır "mutasyon/mutation" veya iki ebeveyn kullanılarak çaprazlama (crossover) yöntemi kullanılır.
 - e. Yeni oluşturulan popülasyonla eski popülasyon yer değiştirilerek yeni jenerasyon oluşturulur.
3. Herhangi bir durdurma kriteri sağlandığında algoritma durur.

2.9.1 İlk Popülasyonun Oluşturulması

Bir sonraki şekilde görüldüğü gibi ilk popülasyon oluşturulması tamamen rastsal olarak gerçekleştirerek başlar.



Şekil 2.9 İlk Popülasyonun Oluşturulması (Matlab GATool, 2004).

Bu örnekte, ilk popülasyon 20 bireyden oluşturulmaktadır. Bu bireyler, popülasyon seçenekleri içerisinde popülasyon değerinin varsayılan değerlerini içerir. Dikkat edilmesi gereken nokta burada görülen bireylerin hepsinin sağ-üst çeyrekte yer almış olmalarıdır. Bunun sebebi onların koordinatları 0 ile 1 arasında yer almasıdır. Bunu sebebi popülasyon seçenekleri içerisinde ilk aralıklarının varsayılan değerlerinin $[0;1]$ olmasıdır (Matlab GATool, 2004).

Fonksiyon için minimum noktanın nerede yaklaşık olarak olduğunu bilerseniz, ilk aralığı ona göre koymalısınız. Bu yüzden noktanın bulunduğu konum bu aralığın ortası olur. Örneğin, eğer bir fonksiyonun minimum noktasının $[0,0]$ noktasına yakın bir yer olduğuna inanırsanız. İlk Aralığı $[-1;1]$ olarak koyabilirsiniz. Fakat, bu örnekte de gösterildiği gibi, GA, ilk aralığı için optimal seçenektan bile daha az minimum bulabilir (Matlab GATool, 2004).

2.9.2 Yeni Jenerasyonun Oluşturulması

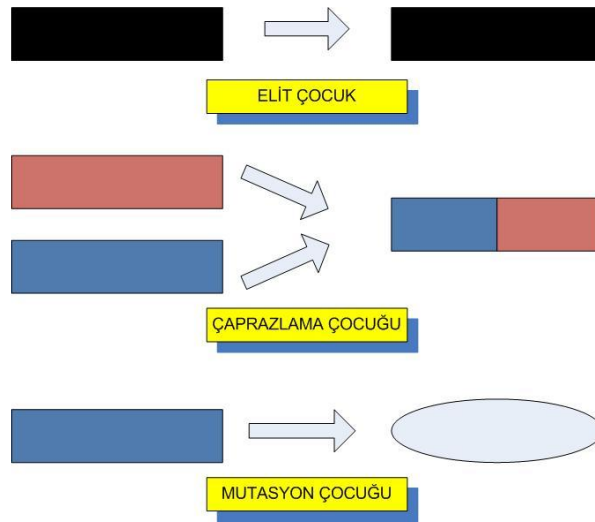
Her basamakta, GA her zaman var olan popülasyondaki üyeleri kullanarak çocukları ortaya koyar bu şekilde yeni jenerasyon oluşturur. Algoritma mevcut popülasyonda bir grup birey seçer ve bunlara ebeveyn der. Bu ebeveynler kendi genlerini ve amaçlarını çocuklara geçirirler. Oluşturulan yeni bireylerin ebeveynlerde olduğu gibi iyi uygunluk değerlerine sahip olması gerekir ve bunu algoritma sağlar (Bielli, Caramia ve Carotenuto, 2000).

Genetik algoritma bir sonraki jenerasyonu oluşturmak için üç yöntem kullanır (Matlab

GATOOOL);

- Eli
t çocuk “Elitizm”; var olan jenerasyon içinde en iyi uygunluk değerine sahip olan birey “elit birey/çocuk olur. Bu birey otomatik olarak bir sonraki jenerasyona alınır.
- Ça
prazlama “crossover”; bir çift ebeveynin kombinasyonu ile oluşturulan çocuklardır.
- M
utasyon Çocukları; bir ebeveynin rastsal olarak gen değişikliği ya da mutasyonu sonucu oluşan çocuklardır.

Aşağıdaki şekil şematik olarak GA ile üç çeşit çocuk oluşumunu göstermektedir.



Şekil 2.10 Genetik Algoritma Çocuk Oluşumu (Matlab GATOOOL, 2004)

Bir sonraki basamak algoritmanın çaprazlama ve mutasyon işlemleriyle nasıl yeni çocukların oluşturulduğu açıklanacaktır.

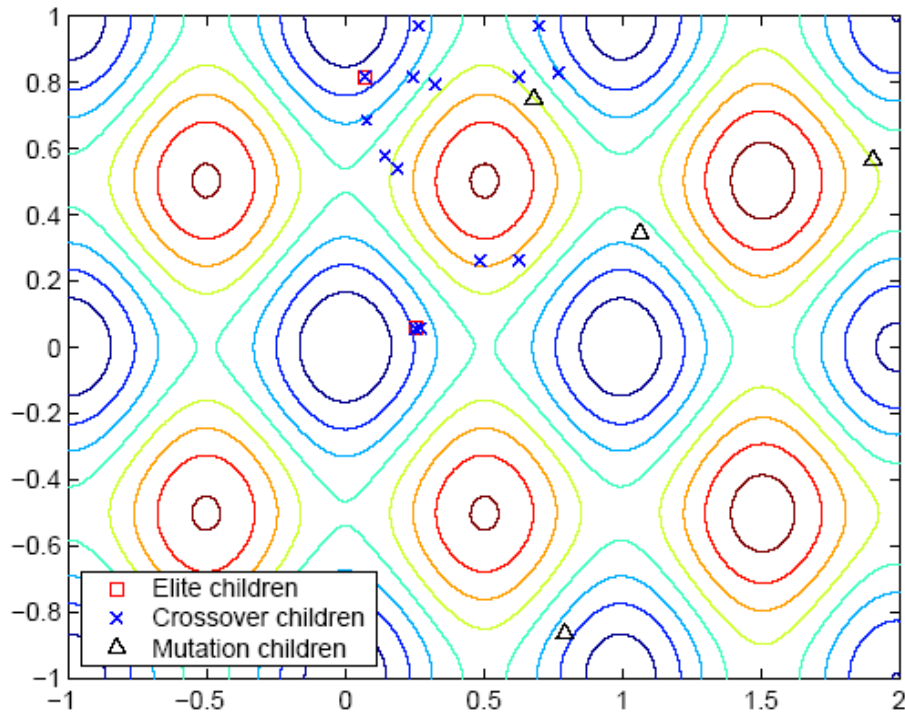
2.9.3 Çaprazlama

Algoritma var olan popülasyondan bir çift ebeveyni kombine ederekten gerçekleştirilmektedir. Çocukların amaçlarının koordine edilmesi çaprazlama fonksiyonu rastgele olarak her iki ebeveyn üzerinde aynı noktalardan işaretleyip çaprazlayarak oluşturur”(Chakroborty, Dwivedi, 2002).

2.9.4 Mutasyon

Algoritma mutasyon çocuklarının oluştururken rastsal olarak belirlenen bir ebeveynin geninin değiştirilmesiyle oluşturur (Erick, 1995).

Bir sonraki şekil ilk popülasyonu, ve bununla birlikte Elitizm, Çaprazlama ve Mutasyonla oluşturulan çocukları göstermektedir (Matlab GATOOL, 2004).



Şekil 2.11 Genetik Algoritma ile Elit-Çaprazlama-Mutasyon Gösterimi (Matlab GATOOL, 2009)

2.9.5 Algoritma İçin Durdurma Kriteri

Algoritma aşağıdaki 5 kriter doğrultusunda sistemi durdurmaktadır (Matlab GATOOL, 2004).



Je

nerasyon sınırlaması: Her ne zaman yeni jenerasyon sayısı eski jenerasyon kadar olduğu zaman algoritmayı durdurur.

- Za
mansal sınırlama: algoritmanın zamansal bir sınırlaması vardır. Her ne zaman algoritma bu sınıra ulaştığında otomatikman kendisini durdurmaktadır.
- A
maç sınırlaması: her ne zaman algoritma amaç fonksiyonunun en iyi değerine ulaştığında veya eşit olduğunda otomatikman kendini durdurur.
- St
all jenerasyon sınırlaması: algoritma ilerleyen zamanda amaç fonksiyonunda herhangi bir gelişme görmez ise kendini otomatikman durdurur;
- St
all zaman sınırlaması: her ne zaman algoritma amaç fonksiyonu belirlenen zaman içerisinde bir gelişme gösteremez ise otomatikman kendini durdurur.

Algoritma herhangi bir durumda buradaki herhangi bir kriter ile karşılaştığı an kendini otomatikman durdurur. Fakat buradaki kriterleri kendimiz istediğimiz gibi düzenleyebiliyoruz.

2.10 Genetik Algoritmalarda Parametre Seçimi

Parametreler, genetik algoritma performansı üzerinde önemli etkiye sahiptir. Optimal kontrol parametreleri bulmak için birçok çalışma yapılmıştır. Fakat tüm problemler için genel olarak kullanılacak parametreler bulunamamıştır (Holland, 1992).

Genetik Algoritmalarda parametrelerin kontrolü ve uygun parametre değerlerinin belirlenmesi, ele alınan problemin uygun çözümünün elde edilmesinde önemli rol oynar. (Taskin ve Emel, 2009)

Bu parametreler, kontrol parametreleri olarak adlandırılmaktadır. Kontrol parametreleri popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, kuşak aralığı, seçim stratejisi ve fonksiyon ölçeklemesi olarak sayılabilir (Yeniay, 2001: 40; Sinreich, 1999; Taşkın ve Emel, 2009). Bu parametreler, parametre kontrol teknikleri ve genel kurallar aşağıda açıklanmıştır.

- Pr
oblem tipine uygun bir kodlama tasarlanmalıdır.

- Popülasyon, yapı bloklarının yani uyum değeri ortalamasının üzerinde olan dizilerin rekabet etmesine olanak tanıyacak büyüklükte olmalıdır.
- Çoğalma operatörü, yapı bloklarının artmasına imkân verecek biçimde tasarlanmalıdır.

2.10.1 Popülasyon Büyüklüğü

Genetik algoritma kullanıcısı tarafından verilen en önemli kararlardan birisidir. Bu değer çok küçük olduğunda, genetik algoritma yerel bir optimum'a takılabilmektedir. Popülasyonun çok büyük olması ise çözüme ulaşma zamanını arttırmaktadır. Bu konuda Goldberg 1985'de, yalnızca kromozom uzunluğuna bağlı bir popülasyon büyüklüğü hesaplama yöntemi önermiştir:

$$N = 1,65 * 20,21 * l \quad l = \text{kromozom uzunluğu,}$$

N = popülasyon büyüklüğü

Ayrıca Schaffer ve arkadaşları 1989'da çok sayıda test fonksiyonları üzerinde yaptıkları araştırmalar sonucunda, 20–30 arası bir popülasyon büyüklüğünün iyi sonuçlar verdiğini belirtmişlerdir.

Her zaman büyük bir popülasyon gerekip gerekmediği sorusuna literatürde şöyle bir açıklama getirilmiştir (Taskin ve Emel, 2009):

“Herhangi bir probleme genetik algoritma ile çözüm aranması durumu söz konusu olsun. Problem için toplam yapılacak hesaplama sayısı da S olarak verilsin. Karmaşık arama ve optimizasyon problemlerinde hesaplamalar en fazla zaman kaybına yol açan işlemlerdir. Bu ifade ile S sayısının önemi anlaşılmaktadır. N büyüklüğünde bir popülasyonun kullanılması durumunda genetik algoritma S/N sayıda kuşak üretecektir. Bu nedenle küçük bir popülasyon büyüklüğü seçildiğinde genetik algoritma daha fazla kuşak boyunca çalışacaktır.”

2.10.2 Çoğalma Parametresi

Çoğalma operatörü ile popülasyonda bulunan çözümlerden uyum değeri yüksek çözümlerin kopyası yapılarak, uyum değeri düşük çözümler elenir (Whitley). Bu işlemi gerçekleştiren birçok operatör mevcuttur. Bunların tümü seçim basıncı denilen tek bir parametre ile gösterilir. Bu parametre çoğalma işlemi sonucunda en iyi çözümün sahip olduğu kopya sayısını göstermektedir. Bazı çoğalma operatörlerinin seçim basıncı aşağıdaki tabloda gösterilmiştir

(Taşkın ve Emel, 2009):

Çizelge 2.4 Bazı Çoğalma Operatörlerinin Seçim Basıncı

Çoğalma Operatörü	Seçim Basıncı
Oransal Operatör(Rulet Tekerleği Yöntemi)	Fmaks/fort
Turnuva Seçimi	2

Büyük bir seçim basıncına sahip bir çoğalma operatörü kullanıldığında, eldeki en iyi çözüm birçok kopyaya sahip olacaktır. Bu da zamanla popülasyonun çeşitliliğini kaybetmesi anlamına gelmektedir (Taşkın ve Emel, 2009).

2.10.3 Çaprazlama Olasılığı

Çaprazlamanın amacı, mevcut iyi kromozomların özelliklerini birleştirerek daha uygun kromozomlar yaratmaktır (Patriksson, Labbé, 2002). Kromozom çiftleri çaprazlama olasılığı ile çaprazlamaya uğramak üzere seçilirler. Çaprazlamanın artması, yapı bloklarının artmasına neden olmakta fakat aynı zamanda bazı iyi kromozomların da bozulma olasılığını arttırmaktadır.

Çaprazlama operatörleri “Arama Gücü” ile gösterilirler. Her çaprazlama operatörünün farklı bir arama gücü vardır. Arama Gücü, çaprazlama operatörünün bir ebeveyn diziyeye uyguladığı zaman erişilebilen arama uzayı oranıdır. L dizi uzunluğu olmak üzere farklı çaprazlama operatörlerinin arama güçleri aşağıdaki tabloda “Tablo2,5'da” verilmiştir (Taskin ve Emel, 2009).

Çizelge 2.5 Bazı Çaprazlama Operatörlerinin Arama Güçleri

Çaprazlama Operatörü	Arama Gücü
Tek noktalı çaprazlama operatörü	$(l-1)/2l-1$
İki noktalı çaprazlama operatörü	$((l-1)/2)/2l-1$
Düzensiz çaprazlama operatörü	1

Arama gücünün en yüksek çaprazlama operatörü “düzgün çaprazlama operatörü” olduğu yukarıdaki tablodan görebiliyoruz. Bu operatörler ile önceden bulunmuş yapı bloklarının korunma olasılıklarının düşük olduğu görülmektedir.

Literatürde farklı optimizasyon problemleri için farklı çaprazlama olasılık değerleri görülmektedir. Örneğin, ikili kodlanmış problemlerin çözümünde De Jong, Grefenstette ve Schaffer adlı araştırmacılar, 0.6 ile 0.95 arası değişen değerler önermişlerdir. Fakat tüm bu önermeler ikili kodlanmış problemler için geçerlidir. Diğer kodlama yöntemlerinin kullanıldığı problemler için olasılık değerleri tamamen farklı olabilir (Taşkın ve Emel, 2009).

Çaprazlama operatörü ile seçim basıncı arasında başarılı bir denge kurulması genetik algoritmanın başarılı sonuçlar vermesini sağlayacaktır (Whitley,).

2.10.4 Mutasyon Olasılığı

Mutasyonun amacı popülasyondaki genetik çeşitliliği korumaktır. Mutasyon olasılığı ile bir kromozomdaki her bitte meydana gelebilir. Eğer mutasyon olasılığı artarsa, genetik arama rastsal bir aramaya dönüşür. Fakat bu aynı zamanda kayıp genetik malzemeyi tekrar bulmada yardımcı olmaktadır (Yeniay, 2001).

Araştırmacılar mutasyon olasılık değerleri bulmak için yoğun çalışmalar yapmaktadırlar. Bremermann, ikili kodlama yöntemi ile kodlama yapılmış fonksiyon optimizasyonu problemleri için optimal mutasyon oranının $1/L$ olduğunu kanıtlamamıştır. Burada l ikili değişkenlerin sayısıdır. Ayrıca diğer araştırmacılardan De Jong ve Grefenstette, ikili kodlama yöntemi ile kodlanmış sürekli optimizasyon problemleri için 0.001 ile 0.01 arasında değişen mutasyon oranları önermişlerdir. Schaffer ve arkadaşları ise yine aynı tip problemler için 0.005 ile 0.01 arası değerleri uygun mutasyon olasılık değerleri olarak bulduklarını çalışmalarında (schaffer, J.D., Caruana, J.D., Eshelman, L.J. Ve Das, R., “a study of Control Parameters Affecting Online Performance of Genetic Algorithms for function Optimizasion”, Proceedings of the third International Conference on Genetic Algorithms, San Mateo, CA: Morgan Kaufmann, 1989, s. 51-60) belirtmişlerdir. Bu önermelerin tamamı, deneysel çalışmalar ile bulunmuş olduğundan genel olarak kullanımları sınırlıdır (Taskin ve Emel, 2009).

2.10.5 Kuşak Aralığı

Her kuşaktaki yeni kromozom oranına kuşak aralığı denmektedir. Genetik operatörler için kaç

tane kromozomun seçildiğini gösterir. Yüksek bir değer birçok kromozomun yer değiştirdiği anlamına gelmektedir.

2.10.6 Seçim Stratejisi

Eski kuşağı yenilemenin çeşitli yöntemleri mevcuttur. Kuşaksal stratejide, mevcut popülasyondaki kromozomlar tamamen yavrular ile yer değiştirir. Popülasyonun en iyi kromozomu da yenilediğinden dolayı bir sonraki kuşağa aktarılamaz ve bu yüzden bu strateji en uygun (elitist) stratejisiyle beraber kullanılmaktadır. En uygun stratejisinde, popülasyondaki en iyi kromozomlar hiçbir zaman yenilenmemektedir, bundan dolayı çoğalma için en iyi çözüm her zaman elverişlidir. Denge durumu stratejisinde ise, her kuşakta yalnızca birkaç kromozom yenilenmektedir. Genellikle, yeni kromozomlar popülasyona katıldığında en kötü kromozomlar yenilenir (Taşkın ve Emel, 2009).

2.10.7 Fonksiyon Ölçeklemesi

Doğrusal ölçekleme, üstsel ölçekleme gibi yöntemler mevcuttur. Probleme göre en uygun ölçekleme yönteminin seçilmesi genetik algoritmanın etkin işlemesi açısından önem taşımaktadır (Holland, 1992).

Genetik algoritmalarda etkin parametrelerin belirlenmesi için yapılan çalışmaların biri de Engin ve Fırlalı adlı araştırmalara aittir. Genetik Algoritma parametrelerini akış tipi çizelgeleme problemlerinde incelenmişlerdir. Akış tipi çizelgeleme problemleri için etkin çaprazlama operatörlerini deneysel olarak bulmuşlardır. Bunun için çalışmada: işlem zamanları 1-1000dakika arasında düzgün dağılıma göre rastsal olarak üretilen 2-makine n-iş problemleri, diğeri de J. Carlier tarafından geliştirilen ve işlem zamanı 1-1000 dakika arasında m-makine ($m > 2$) n-iş problemi olmak üzere iki farklı problem grubu ele alınmıştır. İki farklı grup problem üzerinde yapılan problemler üzerinde yapılan deneyler sonucunda, işlem süreleri yüksek olan çizelgeleme problemlerinde çok makine-çok iş problemleri için en uygun çaprazlama operatörünün doğrusal sıralı çaprazlama (LOX) olduğu; iki makine-çok iş problemlerinde ise sıralı çaprazlama yöntemlerinin (OX) iyi performans gösterdiği belirlenmiştir (Taşkın ve Emel, 2009).

Bu tür problemlerde hangi yöntemin kullanıldığı önemli değildir. Önemli olan hangi parametre değerlerin kullanıldığıdır. Deneme yanılma yöntemleri bu parametrelerin belirlenmesinde oldukça yaygın kullanılmıştır. Bilinmesi gereken önemli bir nokta bu yöntemlerin oldukça zaman aldığıdır.

2.10.8 Genetik Algoritmada Parametre Kontrolü

Ele alınan bir problem ile bu problemi çözen genetik algoritma arasındaki bağlantıyı oluşturan kodlama biçimi ve uyum fonksiyonu tanımlandıktan sonra bile belirlenmesi gereken birçok parametre vardır. Bunlar arasında, ebeveyn çözümlerin yavru çözümlere nasıl dönüştürüleceği, ne tür genetik operatörlerin kullanıldığı vb. sayılabilir. Literatürde çeşitli parametre kontrol teknikleri bulunmaktadır. Parametre kontrol teknikleri bir takım kriterlere göre sınıflandırılabilir. Bu kriterler aşağıda sıralanmaktadır (Taşkın ve Emel, 2009).

1. m olarak ne değiştiriliyor? (Kodlama, uyum fonksiyonu, genetik operatörler, mutasyon oranı, vb.) Ta
2. değişiklik nasıl yapılıyor? (Deterministik bulgusal yardımıyla veya geri besleme tabanlı bulgusal yardımıyla.) Bu
3. genişimin düzeni nedir? (Popülasyon düzeyinde, dizi düzeyinde.) De
4. genişimi etkilemek için hangi istatistik veya kanıt kullanılıyor? (Operatörlerin performansını gösterme-monitör, popülasyon çeşitliliği.) De

Parametre kontrol tekniklerini birinci kriter gere göre sınıflamak için öncelikle bir genetik algoritmanın tüm bileşenlerinin listelenmesi gerekir. Bunlar aşağıdaki gibi dir (Taskin ve Emel, 2009);

1. zilerin kodlanması, Di
2. yum fonksiyonu, U
3. genişim operatörleri ve ilgili olasılıkları, De
4. çim operatörü ve ilgili kuralları, Se
5. yüklük cinsinden popülasyon. Bü

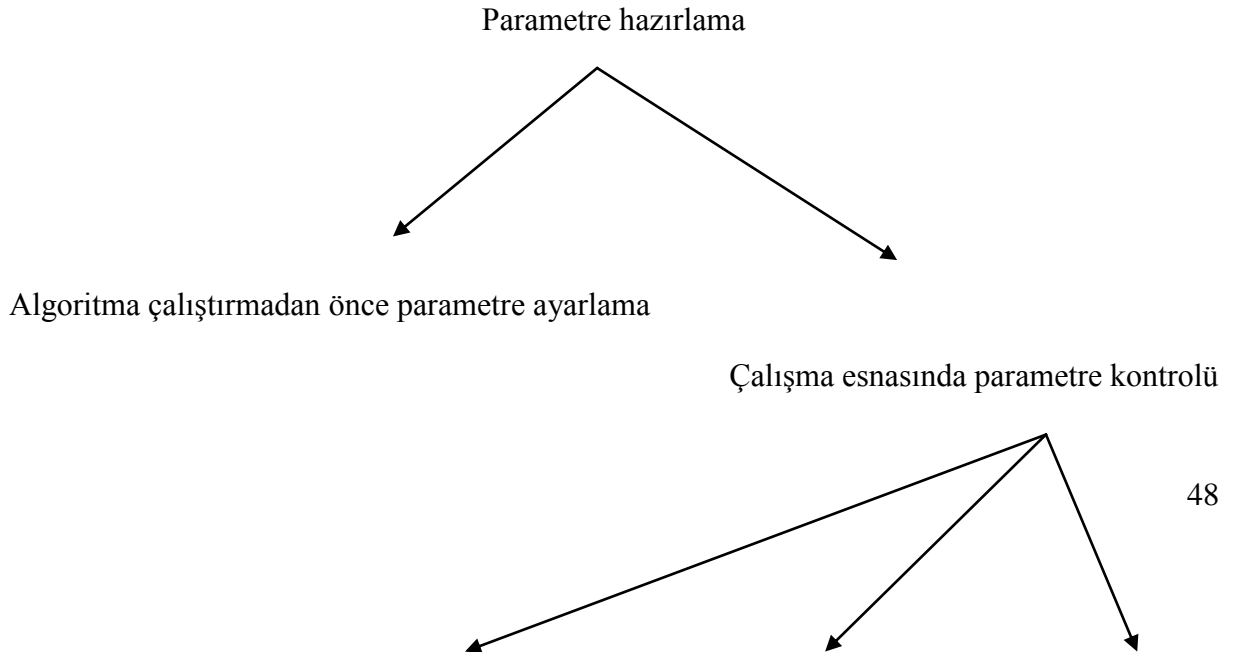
Parametre değerlerini değiştirmede kullanılan her yöntem aşağıdaki sınıflardan birine aittir (Michalewicz ve Fogel, 2004);

1. terministim parametre kontrolü: bu tip kontrol, bir strateji parametresinin (örneğin De

evrimsel aramanın nasıl yapıldığını kontrol eden bir parametre) değeri bazı deterministik kurallar tarafından değiştirilmesi ile gerçekleştirilir. Bu tip parametre kontrolünde, kural arama işleminden herhangi bir geri besleme alınmamaktadır.

2. A
daptif (Uyarlamalı) parametre kontrolü: bu tip kontrol, strateji parametresine olan değişimin yönünü ve/veya büyüklüğünü saptamak için arama işleminden geri besleme alınmıyorsa vardır.

3. Ke
kendinden uyarlamalı parametre kontrolü: Evriminde evrimi fikri, parametrelerin kendinden uyarlamasının uygulanmasında kullanılabilir. Burada uygulanacak parametreler, bireyin/bireylerin veri yapısı/yapılarına kodlanır ve çaprazlama/mutasyon operatörleri uygulanır. Daha uygun bireyler daha uygun bireylere yol açar. Sonuç olarak ise daha uygun parametre değerleri bulunur. Bu parametre aşağıda gösterilen sınıflamaya yol açar:



Deterministik

Adaptif

Kendinden uyumlu

Parametre deęişim tipi Deterministik, Adaptif ve Kendinden Uyumlu olmak üzere üç şekilde olduęu gösterilmektedir. Parametre deęişim bileşeni ise beş sınıftan oluşur. Bunlar; kodlama, uyum fonksiyonu, deęişim operatörleri (çaprazlama ve mutasyon), seçim işlemi ve popülasyon olarak sayılabilir.

2.11 Genetik Algoritma Uygulama Alanları

Genetik algoritmanın genel prensiplerinin oluşturulmasından sonra bu konuda bir çok başarılı bilimsel çalışma ortaya konulmuştur. Aşağıda Genetik Algoritmanın uygulandıęı alanlar verilebilir:

2.11.1 Eniyileme “Optimizasyon”

Bir arama yöntemi olan genetik algoritmalar, farklı bilim dallarındaki optimizasyon problemlerini çözmeye kullanılmaktadır. Genetik algoritmaların uygulandıęı optimizasyon problemleri, fonksiyon optimizasyonu ve birleşik (combinatorial) optimizasyonu altında toplanabilir (Taskin ve Emel, 2009).

Genetik Algoritma araştırmalarının önemli bir bölümü fonksiyon en iyilemesi ile ilgilidir. GA, geleneksel en iyileme yöntemlerine göre zor, süresiz ve gürültü içeren fonksiyonları çözmeye daha etkindirler (Elmas, 2007).

Genetik algoritma araştırmalarının önemli bir bölümü fonksiyon optimizasyonu ile ilgilidir. Genetik algoritmalar, geleneksel optimizasyon tekniklerine göre zor, süresiz ve gürültü (noisy) içeren fonksiyonları çözmeye daha etkindirler (Beasley, Bull ve Martin, 1993).

GA'nın uygulandıęı dięer bir en iyileme problemi, istenen amaçlara ulaşmak üzere, sınırlı kaynakların etkin tahsis edilmesi ile ilgili birleşik en iyileme problemleridir. Gezgin satıcı problemleri, araç yön bulma problemi, iş atölyesi çizelgeleme problemi, yerleşim tasarımı problemi, birleşim en iyileme problemlerine örnektir (Taskin ve Emel, 2009).

2.11.2 Otomatik Programlama ve Bilgi Sistemleri

Genetik algoritmanın 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 (Elmas, 2007).

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 (Behzadi, Alesheikh).

2.11.3 Mekanik Öğrenme

Sınıflama sistemi GA'nın mekanik öğrenme konusunda bir uygulamasıdır. Basit dizi kurallarını öğrenen bir mekanik öğrenme sistemi olan sınıflama sisteminin kural ve mesaj sistemi, özel bir üretim sistemi olarak adlandırılabilir. Bir üretim sistemi "eğer-sonra" kural yapısını kullanır. Bir üretim kuralı, "eğer" yapısından sonra belirtilen durum için "sonra" yapısından sonra gelen faaliyetin gerçekleştirilmesini içerir. GA, sınıflama sistemlerinde kural-bulma mekanizması olarak kullanılmaktadır. Ayrıca, sinir ağlarında ve proteinini yapısal analizinde de kullanılmaktadır (Elmas, 2007).

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 başlamak. Onun yerine uygun model yapısını belirlemek için doğrudan mevcut veriden hareketle bir araç kutusu yaklaşımı kullanır (Whitley).

2.11.4 Finans ve Pazarlama

Genetik algoritma finansal modelleme uygulama alanları için son derece uygundur. Özellikle hisse senedi fiyatlarındaki değişim kalıplarını tahmin etmede ve bulmada, kaynak tahsisi ve

uluslar arası sermaye tahsisi stratejilerini belirlemede GA kullanılabilmektedir. Pazarı ve tüketiciyi tanımada son derece önemli rol oynayan veri madenciliği; veriyi bilgiye, bilgiyi de güvenli kararlara dönüştürür. Veri madenciliği; verimlilik, karlılık, müşteri tatmini ve rekabet edebilme yeteneği gibi yaşamsal konularda işletme üzerinde çok önemli etkileri bulunmaktadır. Veri madenciliğinde kullanılan tekniklerden biride GA'dır. GA tabanlı yaklaşım kullanılarak veri yığınlarından modeller elde edilmektedir (Elmas, 2007).

Finans problemlerinde genel olarak, amaç fonksiyonları tahmin etme gücüne veya bir kıyaslama sonucuna bağlı getirilerdeki gelişmeleri içerir. Kullanılan araç ve problemler arasında mükemmel bir eşleşme mevcuttur. Özellikle hisse senedi fiyatlarındaki değişim kalıplarını tahmin etmede ve bulmada, kaynak tahsisi ve uluslar arası sermaye tahsisi stratejilerini belirlemede genetik algoritmalar kullanılabilmektedir (Taskin ve Emel, 2009).

Tüketici profilini çıkarmak için çok büyük veri tabanlarını işletme amaçları doğrultusunda hızlı ve etkin bir biçimde kullanmak gerekmektedir. Burada kullanılan teknik veri madenciliğidir (Melanie, 1999).

Veri madenciliğinde kullanılan tekniklerden birisi de genetik algoritmadır. Genetik algoritma tabanlı yaklaşım kullanılarak veri yığınlarından modeller elde edilmektedir (Taskin ve Emel, 2009).

2.11.5 Rotalama Problemleri

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 (Taskin ve Emel, 2009).

Daha karmaşık bir araç rotalama problemi olan zaman pencereli rotalama probleminde ise amaç müşteri talebini belirli zaman aralıkları içerisinde minimum toplam maliyetle karşılamaktır (Patriksson, Labbé, 2002).

Genetik algoritmalar özellikle zaman pencereli araç rotalama problemlerinin çözümü için kullanılmaktadır (Chen, 2006).

2.11.6 Gezgin Satıcı Problemi

Genetik algoritmaların, birleşik optimizasyon problemlerine uygulamaları ile ilgili çeşitli çalışmalar mevcuttur. En yoğun yapılan çalışmalardan biri de gezgin satıcı problemleri için yapılmaktadır. Gezgin satıcı probleminde amaç, kat edilen toplam mesafeyi minimize eden bir yolculuk planı oluşturmaktır. Birçok problem tipi gezgin satıcı problemi gibi modellenebilmektedir. Bunlara örnek olarak; devre tasarımı, posta taşıyıcılarının, havayolu uçaklarının, okul otobüslerinin rotalarının bulunması verilebilir (Taskin ve Emel, 2009).

Gezgin satıcı probleminin bir özelliği de değişken sayısı arttıkça üstsel artış gösteren zaman ihtiyacı içinde çözüme ulaştırılabilmesidir. Bu durum bir örnekle şöyle açıklanabilir; bir satış görevlisinin ziyaret etmek durumunda olduğu n tane şehir olsun. Burada tüm şehirler arasındaki maksimum izlenecek rota sayısı $(n-1)!$ dir. Tüm mümkün rotaları basitçe inceleyen ve en kısa olan rotayı bulan bir algoritma kullanılır. Fakat şehir sayısı arttıkça algoritmanın hesaplama için gereksinim duyduğu zaman daha da büyük bir oranda artmaktadır. Ziyaret edilmesi gereken 25 şehir varsa, algoritmanın inceleyeceği rota sayısı $24!$ dir. Bu da yaklaşık $6,2 \times 10^{22}$ sayısına karşılık gelmektedir. Saniyede bir milyon rota inceleme kapasitesine sahip bir bilgisayar, bu problemi, $6,2 \times 10^{17}$ saniyede yani, $1,96 \times 10^{10}$ yılda çözebilmektedir (Mitsuo, Cheng, Lin, 2008).

Herhangi bir problem için kullanılan algoritmanın en yaygın performans ölçütü, algoritmanın çözüme ulaşma süresidir. Gezgin satıcı gibi değişken sayısı arttıkça çözüm zamanı üstsel olarak artan problemlerde bu daha da önemlidir. Genetik algoritmalar birleşik optimizasyon problemlerini klasik yöntemlere göre çok daha kısa sürede çözmektedir. Sonuçta optimala yakın ve kabul edilebilir bir çözüm bulunmaktadır (Holland, 1992).

2.11.7 Çizelgeleme Problemleri

Genetik algoritmaların çizelgeleme problemine ilk uygulama çalışması, Davis tarafından 1985 yılında yapılmıştır. 1987'de Liepins ve arkadaşları, belirli teslim tarihleri ve işlem süreleri olan işlerin çizelgelenmesi problemini araştırmışlardır. Bu problem en basit çizelgeleme problemi adlandırılmaktadır. 1993'de Gupta ve arkadaşları, akış zamanını minimize etme amacını taşıyan tek makine modeli üzerindeki çalışmalarını yayınlamışlardır. Lee ve Kim 1995'de gecikme ve sarkma cezalarını da modele katan çalışmalarını sunmuşlardır. Cheng ve arkadaşları gene aynı yıl, özdeş paralel makinalardan oluşan model üzerindeki çalışmalarını yayınlamışlardır (Taskin ve Emel, 2009).

Bunun dışında; iş atölyesi çizelgelemesi problemi için Biegel ve Davern'nin 1990'da, akış

atölyesi problemi için Badami ve Parks'ın 1991'de, süreç planlama problemi için Vancza ve Markus'un 1991'de yayınlanmış çalışmaları bulunmaktadır (Chen, 2006).

Genel olarak genetik algoritmalar, çizelgeleme problemlerine optima yakın çözüm bulmuşlardır. Fakat çözüm bulma süreleri diğer çözüm yöntemlerine göre oldukça hızlı olmuştur (Taskin ve Emel, 2009).

2.11.8 Hücresel Üretim Problemleri

Hücresel üretim kavramı, üretim sistemlerinin verimliliğini arttırmada anahtar faktörlerden biridir. Hücresel üretim, parça ailelerini belirledikten sonra, her parça ailesini ayrı bir üretim hücresinde imal ederek hücreler arası taşımaları en aza indirmeyi amaçlamaktadır. Genetik algoritmalar, hücreler arası taşımanın minimum olduğu bir hücre kuruluşu amaçlanmasında kullanılabilir. Bu konuda Tate ve Smith'in, Kamrani ve Parsai'nin yayınlanmış çalışmaları bulunmaktadır Ayrıca, Joines'in 1996'da yayınlanmış çalışması mevcuttur (Taşkın ve Emel, 2009).

İşler'in bu konudaki çalışmasında ise, üretim hücrelerinin yapısını temsil eden bilgilerin gösterimi, iki bölüm halindedir. Birinci bölümde, tezgah-hücre ilişkileri, bunun devamı olan ikinci bölümde de parça-hücre ilişkileri de almaktadır. Genetik algoritmalar ile birden fazla çözüm aynı anda ele alınmakta ve bu sayede farklı bölgeler eş zamanlı olarak taranmaktadır. Bunun sonucunda da daha kısa zamanda daha uygun sonuçlar elde edilmektedir (Holland, 1992).

2.11.9 Taşıma Problemleri

Taşıma problemi; tedarikçilerden tüketicilere, talebi karşılamak üzere, minimum maliyetle tek tipte mamul gönderilmesini içermektedir. M tane tedarikçi ve n tane de tüketici mevcuttur. Tek tedarikçiden her bir tüketiciye bir birim mamul ulaştırma maliyeti bilinmektedir. Problem, tüm talebin karşılanması ve maliyet minimizasyonu şartıyla mamulün arz yerinden talep yerine optimum tahsisini sağlamaktır. Son zamanlarda, çeşitli taşıma problemlerinin çözümü için evrimsel (evolutionary) yaklaşımlarla çözüm önerileri sunulmaktadır. Michalewicz ve arkadaşları, doğrusal ve doğrusal olmayan taşıma problemleri için genetik algoritma kullanımını ilk öneren araştırmacılarıdır. Ayrıca, Gen ve Li de genetik algoritmaları taşıma problemlerinin çözümü için kullanmışlardır (Taskin ve Emel, 2009).

2.11.10 Tesis Yerleşimi Problemi

Tesis yerleşim problemleri araç/gereçleri veya diğer kaynakları belirli bir kritere göre

optimum performans sağlayacak şekilde yerleştirme kararını içermektedir. Bu gibi kararlar, araç/gereçlerin genellikle farklı ürünleri üretme esnasında kullanılmasından dolayı karmaşık hale gelmektedir. Her ürünün kendine özgü gereksinimleri olabilir ve tüm ürünler için toplam üretim maliyetinin optimum olması sağlanacak şekilde yerleşim tasarlanabilir. Yerleşim kararları hızlı ve doğru verilmelidir. Çünkü kararların zayıflığı üretim esnasında ortaya çıkmakta ve bu da artı maliyetlere yol açmaktadır. Örneğin, üretimde robot kullanan işletmelerin tesis yerleşimi tasarımında karmaşıklık söz konusudur. Tek bir robot bir makineden diğerine parçalar taşıırken hareketsiz bir noktada sabitlenir ve yalnızca bir eksen etrafında hareket eder. Robotun hareketine göre, makineler tek-sıra, doğrusal çift-sıra, dairesel tek-sıra ve çoklu-sıra gibi dört farklı yerleşim şekliyle yerleştirilebilir. Burada, dairesel tek-sıra, doğrusal tek-sıranın özel bir durumudur. Ayrıca doğrusal çift-sıra da çoklu sıra probleminin bir alt kümesidir (Taskin ve Emel, 2009).

Tesis yerleşim problemleri bunun gibi bir çok zorluğu içermektedir. Genetik algoritmalar, bu tip problemlerin çözümünde uygun bir çözüm yöntemi olabilmektedir. Bu alanda Tam'ın 1992 yılında, Chan ve Tansri'nin 1994 yılında, Tom ve Chan'nin 1998 yılında, İşlier'in 1998 yılında ve Al Hakim'in 2000 yılında yayınlanmış çalışmaları bulunmaktadır (Taskin ve Emel, 2009).

2.11.11 Montaj Hattı Dengeleme Problemi

Montaj işlemi endüstrilerde çok önemli bir rol oynamaktadır. Nof ve arkadaşlarının 1997'de yayınlanan çalışmalara göre üretilen mamullerin montajı, toplam üretim zamanının % 50'sine, toplam birim üretim maliyetinin % 20'sine ve işçilik maliyetlerinin % 30-% 50'sine karşılık gelmektedir. Bundan dolayı montaj hattı dengeleme problemi, firmalar açısından yaşamsal öneme sahiptir (Taskin ve Emel, 2009).

3. MATLAB ve GENETİK ALGORİTMA

MATLAB, temel olarak nümerik hesaplama, grafiksel veri gösterimi ve programlamayı içeren teknik ve bilimsel hesaplamalar için yazılmış yüksek performansa sahip bir yazılımdır. Matlab programının tipik kullanım alanları: Matematik ve hesaplama işlemleri, algoritma geliştirme, modelleme, simülasyon ve ön tipleme, veri analizi ve görsel efektlerle destekli gösterim, bilimsel ve mühendislik grafikleri, uygulama geliştirme şeklinde özetlenebilir.

Not: bu bölümde belirtilen bilgilerin hepsi aşağıda gösterilen kaynaklarının derlenmesi ile oluşturulmuştur.

- Genetic Algorithm and Direct Search Toolbox For Use with MATLAB ve
- TEACHING GENETIC ALGORITHM USING MATLAB. Int. J. Elect. Enging. Educ., Vol. 36, pp. 139–153. Manchester U.P., 1999. Printed in Great Britain,

MATLAB adı, *MATrix LABORatory (Matrix Laboratuvarı)* kelimelerinden gelir. MATLAB, ilk olarak Fortran Linpack ve Eispack projeleriyle geliştirilen ve bu programlara daha etkin ve kolay erişim sağlamak amacıyla 1970'lerin sonlarında yazılmıştır. İlk başlarda bilim adamlarına problemlerin çözümüne matris temelli teknikleri kullanarak yardımcı olmaktadır. Bugün ise geliştirilen yerleşik kütüphanesi ve uygulama ve programlama özellikleri ile gerek üniversite ortamlarında (başta matematik ve mühendislik olmak üzere tüm bilim dallarında) gerekse sanayi çevresinde yüksek verimli araştırma, geliştirme ve analiz aracı olarak yaygın bir kullanım alanı bulmuştur. Ayrıca işaret işleme, kontrol, fuzzy, sinir ağları, wavelet analiz gibi birçok alanda ortaya koyduğu **Toolbox** adı verilen yardımcı alt programlarla da özelleştirilmiş ve kolaylaştırılmış imkânlar sağlamış ve sağlamaya da devam etmektedir.

Web adresi: "<http://www.mathworks.com>"

MATLAB'i çok gelişmiş özellikleri olan, programlanabilen bir bilimsel hesap makinesine benzetilebilir. MATLAB'de yazılan programlar, MATLAB dili kullanılarak yazılır ve MATLAB programı içinden çalıştırılır. Ayrıca yazdığımız programları DLL ve EXE olarak oluşturabilindiği gibi C/C++ kodlarına da çevirebilirsiniz.

MATLAB diğer programlama dillerine benzer bir giriş (input)'ları vardır. Bu input'ları 4 ana başlık altında toplayabiliriz.

- Sa
yılar (Numbers)
- De
ğişkenler (Variables)
- İşl
eçler (Operators)
- Fo
nksiyonlar (Functions)

MATLAB’de her şey bir dizi (array) olarak işleme konur. Bir dizi, skaler, vektör, matris veya metinsel dizge (karakter dizisi) olabilir.

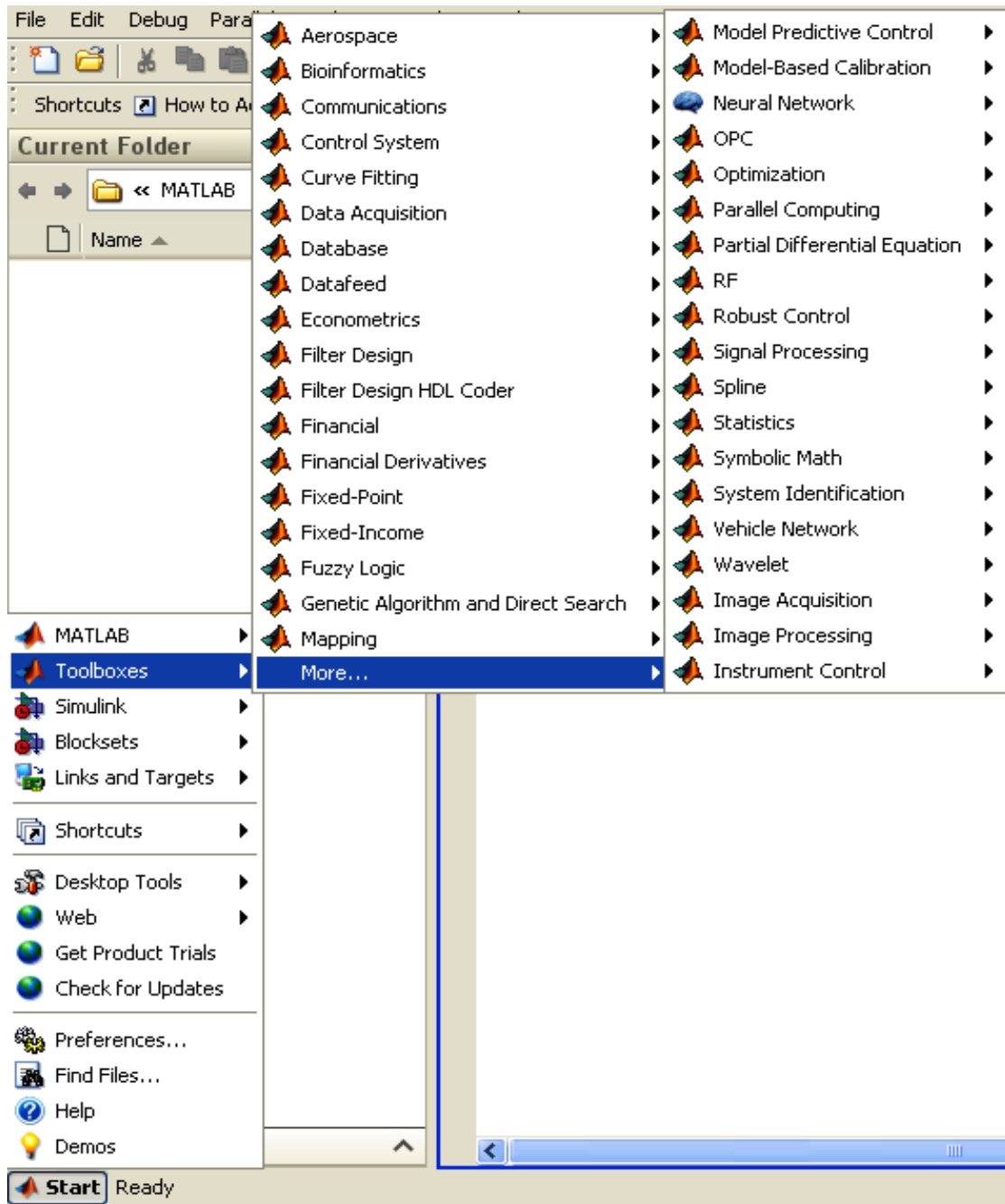
- 1x
1 dizisi skaler (scaler) veya metin dizgesi (string) gösterir. $a=3$ $b=-20.56$ $c=3e-4$ $d=2+5*i$ $e='Bir\ tamsayı\ giriniz.'$ gibi. Metin dizgeleri (text string), '...' arasında verilir.
- nx
1 veya 1xn dizisi bir vektör (vector) gösterir. $x=[0, 2, 4, 6]$ ifadesinde x, 1x4 boyutunda bir vektördür. [] köşeli parantezler içindeki sayıların arasında virgül (,) veya en az bir veya daha fazla boşluk olmalıdır.
- nx
m veya mxn dizisi bir matris (matrix) gösterir. [] köşeli parantezler içindeki sayı grupları arasında noktalı virgül (;) olmalıdır. ; işareti matrisin sütunlarını ayırır. Sayı gruplarında virgül (,) veya en az bir veya daha fazla boşluk olabilir.

MATLAB programı ile

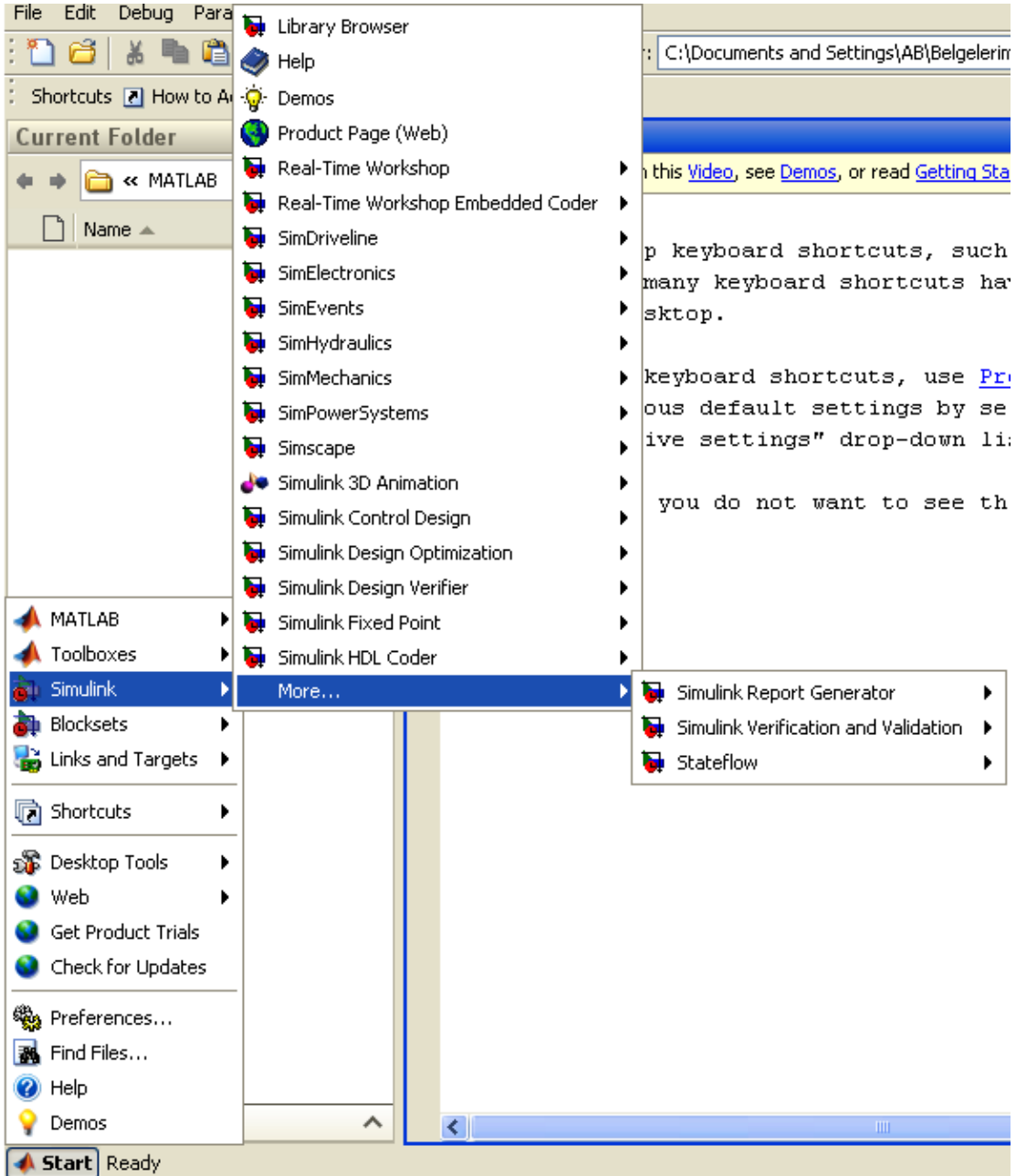
- M
atematiksel Ölçüm ve Hesaplamalar (Maths and Computation)
- Al
goritma Geliştirme (Algorithm Development)
- Ve
ri Elde Etme (Data Acquisition)
- G
örüntü İşleme (Image Processing)

- Sa
 - Sü
 - Da
 - Ya
 - Ve
 - O
 - M
- yısal İşaret İşleme (Signal Processing)
- zgeç Tasarımı (Filter Design)
- lgacıklar (Wavelets)
- pay Sinir Ağları (Artificial Neural Networks)
- ritabanı (Database)
- ptimizasyon (Optimization)
- odelleme ve Simülasyon (Modeling, Simulation)

MATLAB'da kusursuz tasarımlar yapabilirsiniz. Ayrıca MATLAB'ı vazgeçilmez yapan durumlardan en önemlileri : Toolbox 'lar ve Simulink



Şekil 3.1 MATLAB ToolBoxes (MATLAB 7.9.0 (R2009), 2009)



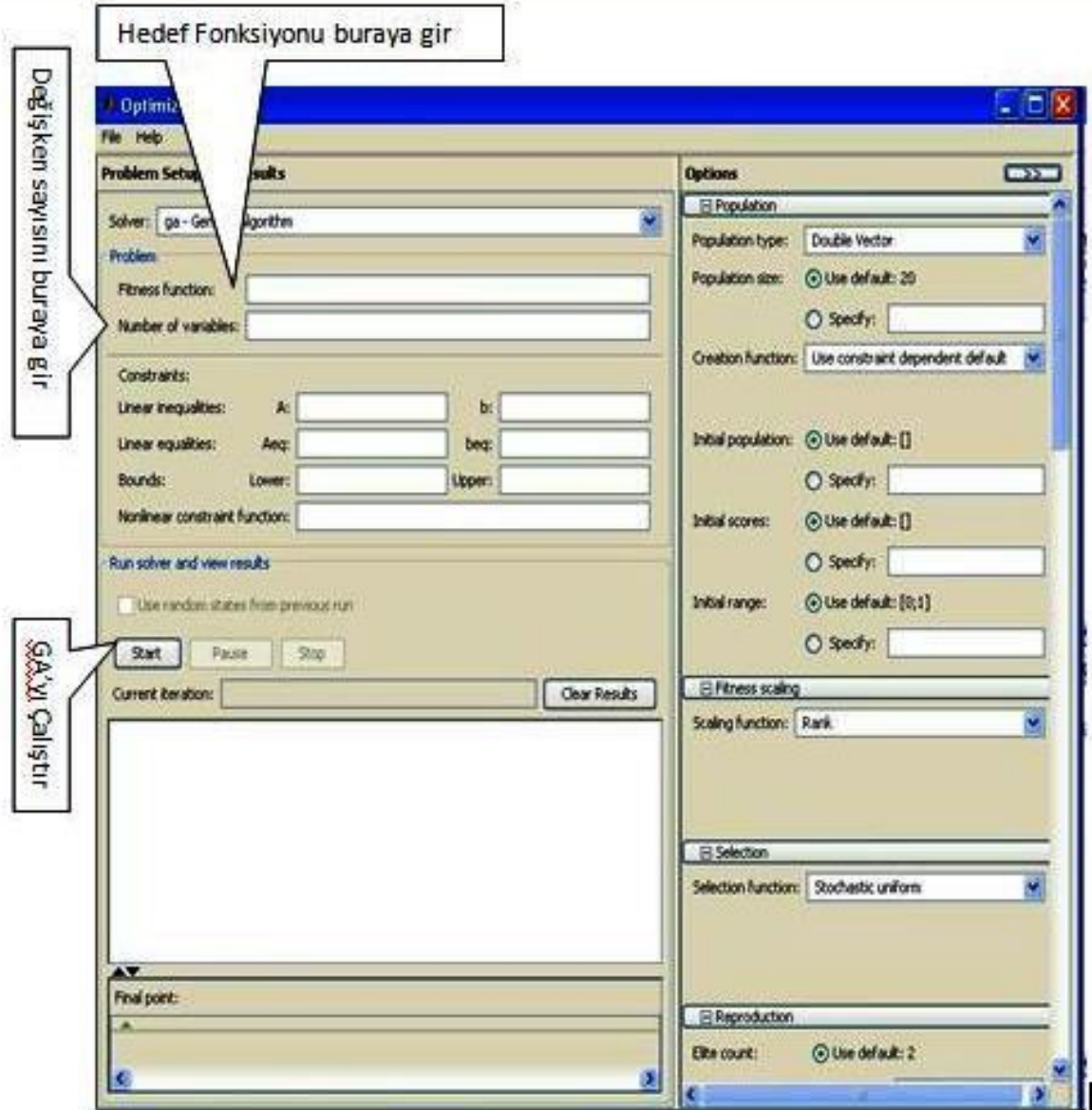
Şekil 3.2 MATLAB Simulink (MATLAB 7.9.0 (R2009), 2009)

Yukarda anlatılan bilgiler ışığında, MATLAB ile neler yapabileceğiniz konusunda bir miktar fikir vermiştir. MATLAB'in içindeki pek çok yazılım aracı ile (toolbox) özel uygulamalarımız için hazır yazılmış onlarca fonksiyon bulabiliriz. Ayrıca pek çok mühendislik problemlerinin çözümünde MATLAB'in avantajlarını kullanabiliriz.

3.1 MATLAB Genetik algoritma Tools

MATLAB Genetik Algoritma Tool'u bizlere genetik algoritma kullanmak için bir ara yüz sunmuştur. Bu ara yüz ile biz komut sistemini kullanmadan var olan problemi çözmeye fırsat verir. Şimdi ben burada GA Tool'unu nasıl çalıştığı konusunda kısa bilgi vereceğim.

MATLAB başlatılarak Genetik Algoritma Tool'unu açıyoruz. Bu işlem aşağıdaki gibi bir ara yüz açacaktır.



Şekil 3.3 MATLAB Genetik Algoritma Tool Ara Yüzü (MATLAB 7.9.0 (R2009), 2009)

Genetik algoritmayı kullanmak için İlk olarak Aşağıda belirtilen değerleri girmeniz

gerekmektedir.

Amaç Fonksiyonu (Fitness Funktion): Minimize etmeye çalıştığımız amaç fonksiyonudur. Bu amaç fonksiyonunu şöyle girilir "@fitnessfun". Bu şekilde girildiğinde sistem Fonksiyonun yazıldığı Mfile giderek fonksiyona ulaşır.

Değişken Sayısı (Number of Variables): çözüm için gerekli değişken sayısı da buradan girilir. Sonra GA çalıştırılır START butonuna basarak. Bu Genetik Algoritma tool'u sonuçları Status and Results panelinde gösteriri.

Biz Option panelini kullanarak optionlarımızı değiştirebiliriz.

3.2 Rastrigin's Function

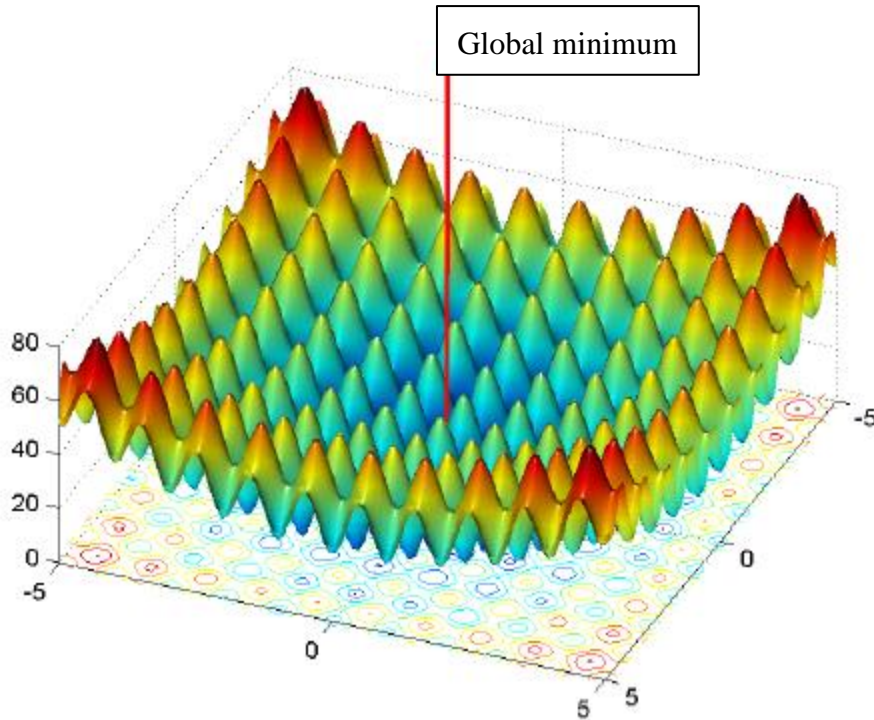
Rastrigin's Fonksiyonu aşğıdaki gibi gösterirsek,

$$Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$$

(3.

1)

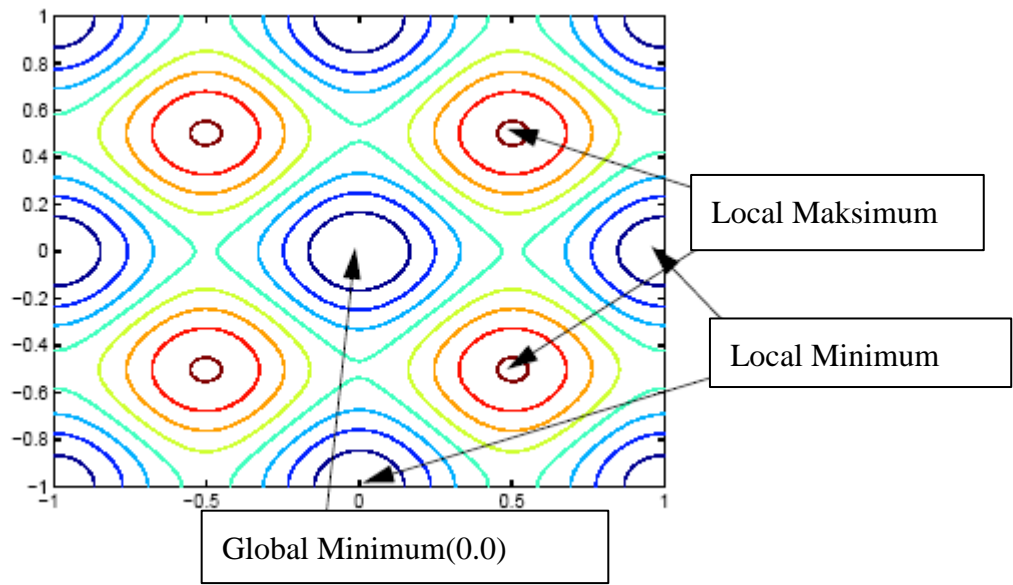
Sistem otomatikmen daha önce tanımlanan *rastriginsfcn.m* M_file'a giderek bu fonksiyonun değerini hesaplar. Aşağıda şekil ortaya çıkacaktır.



Şekil 3.4 Rastrigin's Fonksiyon Grafiği. (GA Tools, 2004)

Yukarıdaki şekilden de anlaşılacağı gibi bu fonksiyonun birçok local mimum noktası bulunmaktadır. Fakat bu fonksiyonun sadece bir tane Global minimum noktası vardır. Buda x ve y panelindeki [0,0] noktasıdır. bu nedenden dolayı herhangi bir lokal minimum değeri içim bu fonksiyonun değeri 0 dan büyüktür. Merkezden ne kadar uzakta olunursa fonksiyonun değeri o kadar büyüktür denilir.

Aşağıdaki şekilde bu fonksiyonun Global ve Lokal minimum değerlerini gösteren başka bir şekildir.



Şekil 3.5 Rastrigin's Fonksiyon Grafiği . (GA Tools, 2004)

Örneğin biz bu fonksiyonun minimum değerini bulmaya çalışalım. Aşağıdaki adımları sırasıyla uygulayalım.

1.

U

uygunluk fonksiyonu satırına Rastrigin's Fonksiyonunu girelim ve değişkenler satırına da 2 değerini yazalım.

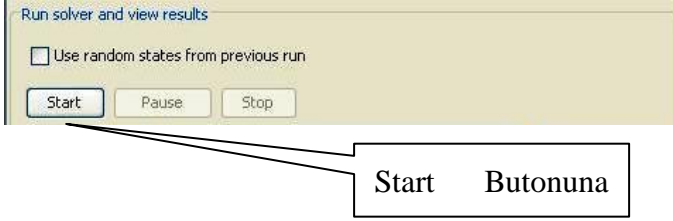
Problem	
Fitness function:	@rastriginsfcn
Number of variables:	2

Ekranında bu şekilde görülmesi gerekmektedir.

2.

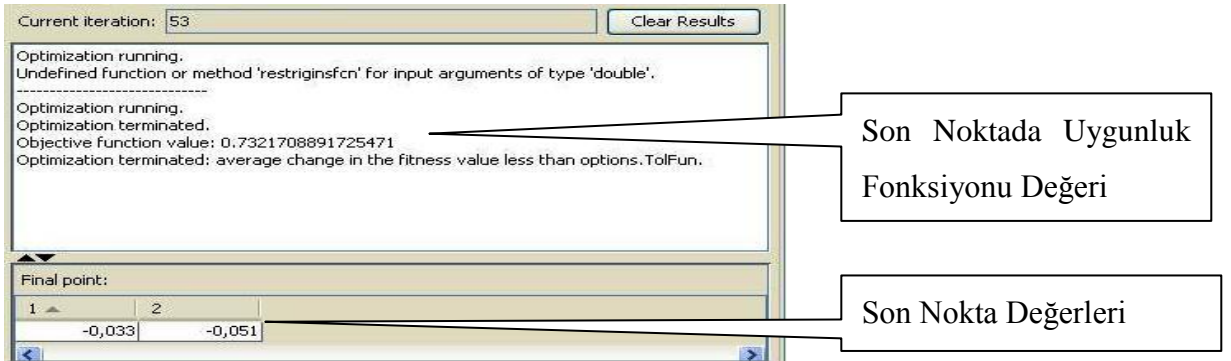
So

nra run solver ve view result panelinde START butonuna basılır.



Algoritma çalışırken Current Generation bölgesi görülecektir ve buradan jenerasyon durumu takip edilebilir. Eğer istersek biz “Pause” butonuna basarak işlemi durdura biliriz fakat “Pause” butonuna bastıktan sonra “Resume” butonu ortaya çıkacaktır. Tekrar devam edecek isek bu butona basarak işlemi sonlandıracağız.

Her ne zaman işlem sonlandığında “Statu and Results” paneli aşağıdaki gibi görünecektir.



Bu ekran ile uygunluk fonksiyonun değeri görülebilmektedir.

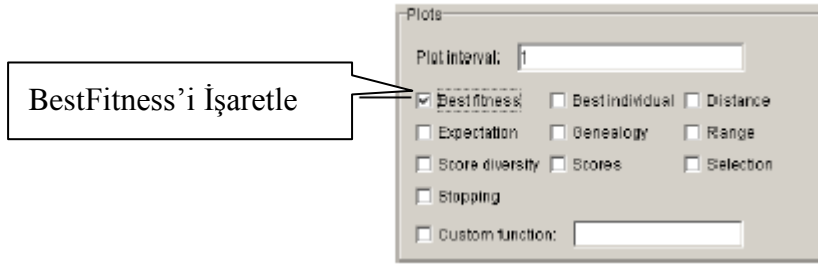
Uygunluk Fonksiyonu= 0,7321708891725471

Belirlediğimiz değişkenlerin yani son karar verilen nokta değerleri 1= -0,033 ve 2= -0,051 olarak bulunmuş olur.

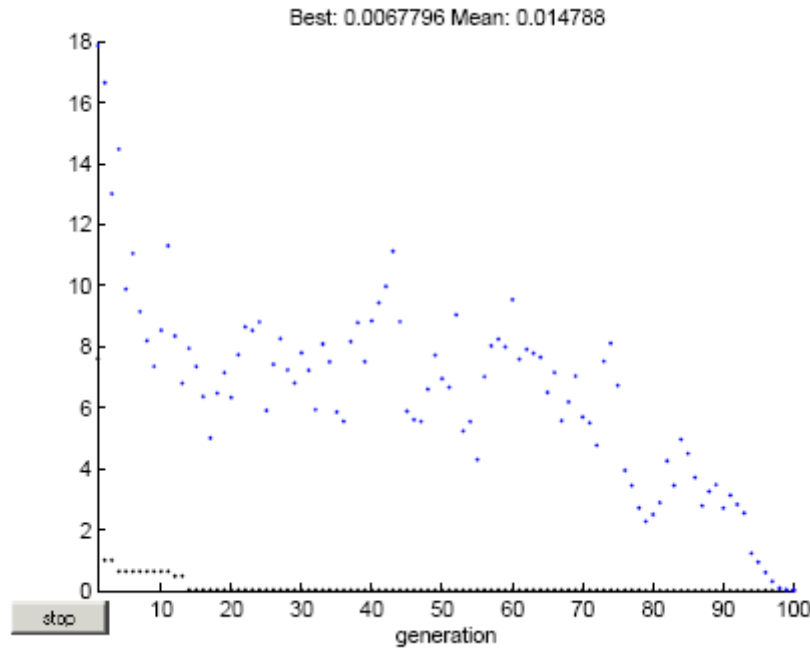
3.3 Genetik Algoritma İle Grafik Çizimi

Grafik çizim paneli “Plot panel” bize birçok grafik çizimi için yardımcı olmaktadır. Bu Grafikler bize çalışan algoritma hakkında bilgiler vermektedir. Bu tür bilgileri kullanarak çeşitli noktalarda değişiklik yapabilir ve algoritmanın performansının artırılması sağlanabilir. Eğer bir örnek verecek olursak, uygunluk fonksiyonun en iyi veya ortalama değerini her

jenerasyon için çizmek istersek, Best Fitness Value yanındaki kutuya tıklarsak aşağıdaki şekilde görüldüğü gibi;



Ne zaman biz algoritmayı çalıştırır isek Genetik Algoritma Tool bize uygunluk fonksiyonunun en iyi ve ortalama değerini her jenerasyon için gösterecek ve program sonlandığında ise aşağıdaki gibi bir grafik elde edilecektir.



Şekil 3.6 Genetik Algoritma İle Grafik Çizimi “En iyi ve Ortalama Uygunluk Fonksiyonu Değeri” (GA Tools, 2004)

3.4 MATLAB Programının Genetik Algoritma Tool’unun Çalışma Şekli

3.4.1 İlk Popülasyonun Oluşturulması

Popülasyonun kromozomlardan oluştuğunu belirtmiştik ve yine aynı şekilde kromozomların genlerden oluştuğunu ve genlerinde bitlerden (0,1) oluştuğunu daha önce belirtmiştik.

MATLAB programının ilk popülasyonu nasıl oluşturduğunu ortaya koyacak olursak; örneğin m sayıdaki bir popülasyonda herhangi bir X_i çözümü için ($i=1, 2, 3, \dots, m$), X 'in büyüklüğü $[a,b]$ ve kromozom uzunluğu l olsun. İlk popülasyon rassal olarak oluşturulacaktır. Popülasyon oluştururken ilk olarak ikili kodlama sistemini kullanarak sırasıyla [ikili kodlama sistemiyle kromozom, gerçek X in değeri,]

$$\text{pop} = \begin{pmatrix} \text{binary string 1} & x_1 & f(x_1) \\ \text{binary string 2} & x_2 & f(x_2) \\ \vdots & \vdots & \vdots \\ \text{binary string } \text{popsize} & x_{\text{popsize}} & f(x_{\text{popsize}}) \end{pmatrix}$$

0110.....10		
X değişkeninin ikili kodlama sistemiyle	X 'in gerçek	X 'e bağlı uygunlu değeri

Şekil 3.7 Genetik Algoritmanın ilk popülasyon oluşturma matrisi

Kodlama sistemiyle bakacak olursak;

```
function [ pop ] = initialise( popsize, stringlength, fun );
    pop = round( rand( popsize, stringlength + 2 ) );
    pop(:, stringlength + 1) = sum( 2.^( size( pop(:, 1:stringlength), 2 ) - 1 : -1:0 ) .
        * pop(:, 1:stringlength) ) * ( b - a ) / ( 2.^ stringlength - 1 ) + a;
    pop(:, stringlength + 2) = fun( pop(:, stringlength + 1) );
end
```

Şekil 3.8 MATLAB Genetik Algoritma Tool'unda İlk Popülasyonun Operatörü Algoritması

Burada ilk ikili kodlama sistemi rassal olarak üretilmiş ve sonra $(stringlength + 1)$ 'inci ve $(stringlength + 2)$ 'inci kolonlardaki gerçek X değerleri ile değiştirilmektedir. Burada fun fonksiyonumuzu göstermekte ve bu fonksiyon M-file da daha önce oluşturulmuş idi.

3.4.2 Çaprazlama

Çaprazlama işleminde daha önce de belirttiğimiz gibi iki ebeveyn den iki tane yeni çocuğun üretilmesi şeklinde oluyor idi. Burada pc 'yi biz çaprazlama olasılığı olarak koyarsak, algoritma aşağıdaki gibi olmaktadır;

```

function [child1, child2] = crossover(parent1, parent2, pc);
if (rand < pc)
    cpoint = round(rand*(stringlength - 2)) + 1;
    child1 = [parent1(:,1:cpoint) parent2(:,cpoint + 1:stringlength)];
    child2 = [parent2(:,1:cpoint) parent1(:,cpoint + 1:stringlength)];
    child1(:, stringlength + 1) = sum(2.^(size(child1(:,1:stringlength),2) - 1:-1:0).
        *child1(:,stringlength))*(b - a)/(2.^stringlength - 1) + a;
    child2(:, stringlength + 1) = sum(2.^(size(child2(:,1:stringlength),2) - 1:-1:0).
        *child2(:,1:stringlength))*(b - a)/(2.^stringlength - 1) + a;
    child1(:, stringlength + 2) = fun(child1(:, stringlength + 1));
    child2(:, stringlength + 2) = fun(child2(:, stringlength + 1));
else
    child1 = parent1;
    child2 = parent2;
end
end
end

```

Şekil 3.9 MATLAB Genetik Algoritma Tool'unda Çaprazlama Operatörü Algoritması

Burada yeni çaprazlama ile oluşan yeni jenerasyonu eski olan ebeveyn ile karşılaştırması yapılmaktadır. Genel anlamda, rassal olarak bir değer üretiliyor ve bu değer Pc çaprazlama olasılığı ile karşılaştırılıp küçük çıkması halinde çaprazlama gerçekleştiriliyor eğer büyük çıkarsa ebeveynler aynı şekilde kalıyor. Burada çaprazlama noktası $cpoint$ 1 ile kromozom uzunluğu arasında bir noktadır. Round fonksiyonuyla çaprazlama noktası $cpoint$ kromozomum maksimum ve minimum uzunluğu arasında yani (1 ile (kromozom uzunluğu) $stringlength - 1$ noktaları arasında) belirleniyor. Bu şekilde yeni bir aile bireyleri oluşturulmuş olunuyor.

3.4.3 Mutasyon

Mutasyon işlemi herhangi bir bireyden yeni bir bireyin oluşturulması şeklinde olmaktadır. Pm mutasyon olasılığı olsun. Bir mutasyon noktası $mpoint$ belirlenir ve buradaki bit 1 is 0'a veya 0 ise 1'e dönüştürülür. Yeni jenerasyonun algoritması aşağıdaki gibidir.

```

function [child] = mutation(parent, pm):
    if (rand < pm)
        mpoint = round(rand*(stringlength - 1)) + 1;
        child = parent;
        child[mpoint] = abs(parent[mpoint] - 1);
        child(:, stringlength + 1) = sum(2.^ (size(child(:, 1:stringlength), 2) - 1: - 1:0).
            *child(:, 1:stringlength))*(b - a) / (2.^ stringlength - 1) + a;
        child(:, stringlength + 2) = fun(child(:, stringlength + 1));
    else
        child = parent;
    end
end
end

```

Şekil 3.10 MATLAB Genetik Algoritma Tool’unda Mutasyon Operatörü Algoritması

3.4.4 Seçim

Seçim operatörü de hangi bireyin hayatta kalıp bir sonraki jenerasyona kalacağına karar verir. Burada *rulet tekerleği* yöntemi kullanılmıştır. Burada il yapılan her seçilmiş bireyin olasılık değerine bakılır. Vektör *prob* üzerinde bu olasılıklar toplanır. Bir sonraki aşamada *rns* vektörü üzerinde normalize edilmiş rassal değerler oluşturulur. Bu normalize etme aşamasında *rns* ve *prob* vektörleri karşılaştırılarak yapılır. Ve hangi bireyin bir sonraki jenerasyon da yer alacağına karar verilir. Mutasyon operatörü için kullanılan algoritma aşağıdaki gibidir.

```

function [newpop] = roulette(oldpop);
    totalfit = sum(oldpop(:, stringlength + 2));
    prob = oldpop(:, stringlength + 2) / totalfit;
    prob = cumsum(prob);
    rns = sort(rand(popsiz, 1));
    fitin = 1; newin = 1;
    while newin <= popsiz
        if (rns(newin) < prob(fitin))
            newpop(newin, :) = oldpop(fitin, :);
            newin = newin + 1;
        else
            fitin = fitin + 1;
        end
    end
end
end

```

Şekil 3.11 MATLAB Genetik Algoritma Tool’unda Seçim Operatörü Algoritması

4. UYGULAMA

Otobüs sitemlerini tasarlarken karşımıza sistematik olarak otobüs ağı tasarımı, otobüs sıklığı düzenlemesi, zaman tablosu planlanması, otobüs zaman çizelgesi ve otobüs şoförü zaman çizelgesi çıkmaktadır. Fakat her zaman karşımıza şu iki konu öncelikli olarak çıkmaktadır. Rotaların düzenlenmesi ve otobüs sıklıklarının ayarlanması bir diğeri de sistematik olarak sistemin performansının operatörlerin ve kullanım noktalarına göre belirlenmesi önem taşımaktadır. Eğer herhangi bir tasarruf sağlayacak ise bunun yolu otobüs hatlarının tekrar tasarlanmasından ve doğru ölçümlendirilmiş-hesaplanmış müşteri taleplerinin yani insanların gidecekleri noktaların en iyi bir şekilde ortaya konması sonucunda gerçekleşir.

In Kidwia (1998) transit network için şu üç ana noktayı temel almıştır; transit rota tasarımı, transit atama-devir ve transit çizelgeleme. Yapılan bu çalışmada toplu taşıma otobüslerinin çizelgeleme problemini formülasyonun gerçekleştirilmesi ve iki şekilde çözümlenmesini içermektedir. İlk şekliyle, otobüsler otobüslerin interaktif prosedürüne göre bireysel rotalanmaları, ikinci şekliyle otobüs filosundaki otobüs sayısının azaltılması ve bu azaltma esnasında Genetik Algoritmasının kullanımını gerçekleştirmek şeklinde olacaktır.

Genetik Algoritma bir optimizasyon yöntemi olup doğal seçim ve doğal genetik yapılanması üzerine yapılandırılmıştır. Genetik algoritma; parametre değerlerini kullanmak yerine, araştırmasını popülasyon arasında yapar ve kodlama parametreleriyle çalıştığı için diğer araştırma yöntemlerinden farklıdır. GA, çıktıları olasılıklara dayanan sezgisel bir yöntemdir. Fakat diğer geleneksel yöntemler sabit olmayan yani artabilen veya azalabilen bilgiyi kullanır. Sonuç olarak bu modelimizi gerçek sisteme uygulamaya çalışılıp ve karşılaşılan sonuçlar ortaya konulmuştur.

4.1 Literatür Çalışmaları

Literatüre baktığımızda bu tür problemler konusunda değişim yaklaşımlar ortaya konmuş ve değişik hesaplama yöntemleri kullanılmıştır. 1967 yılında Lampkin ve Saalmans sıklığın belirlenmesi konusunda bir kısıt optimizasyon problemi formüle etmiştir. Bu arada toplam seyahat zamanını düşürmeyi amaçlamış ve sonucun ortaya konulması için rastsal arama prosedürü kullanmıştır (Patriksson, Labbé, 2002).

1972 yılında Rea ortaya koyduğu modelle optimum otobüs ağını ortaya koymak için otobüs hareket sıklıklarını tekrar tekrar ayarlamak ve her otobüs hattı üzerindeki otobüs çeşitlerini sisteme uyumluluklarını sağlayarak belirlemeye çalışmıştır. Bu çalışmasıyla bazı hatlardaki

otobüs sayısını yükseltmişken bazı hatlarda azaltmıştır. 1974 yılında Silman et al, otobüs sayısı ve birkaç otobüs hattı üzerinde optimum otobüs hareket sıklıklarını belirlemiş ve bununla yolcu konforunu ve toplam yolculuk zamanını azaltmayı başarmıştır. 1977 de Hsu ve Surti, verilen otobüs sayısına göre yeteri kadarki otobüs hareket sıklığını ortaya koymuştur. Yine 1977 yılında Scheele yaptığı çalışmada otobüs trafik modeli bulmak için matematik programlama algoritmanın birleşik minimizasyon şeklini kullanmıştır. Optimal otobüs hareket sıklığını projection metodunu kullanarak çözmüştür. Dubois et al 1979 yılında otobüs hareket sıklığını belirlemek için iki basamak prosedür'ünü kullanmıştır. 1979 yılında Mandl, sistemde rotaların tümünde sabit bir otobüs hareket sıklığı olduğunu varsayarak işlem gerçekleştirmiştir.

1980 yılında Dhingra, bir similasyon modeli geliştirerekten efektif çalışan bir otobüs ağı ve çeşitli rota düzeylerine etki eden otobüs hareket sıklığı çalışması ortaya koymuştur. 1981 yılında Furth ve Wilson tarafından ortaya konan model otobüs zaman periyodu ve otobüs rotaları arasındaki paylaşımı gerçekleştirmiştir. Bununla birlikte net sosyal kazanç konusunu toplam kazanç kısıt'ı üzerinde, toplam otobüs sayısı üzerinde ve araç yükleme seviyesi-miktarı üzerinde işlemiştir. 1982 de Han ve Wilson ortaya koydukları model ile yolcu rota tercih davranışını, yolcuların bekleme sürelerini ve otobüs kalabalık minimize etmeyi sabit otobüs sayısı ile her rota için yeterli otobüs kapasitesi kısıtları doğrultusunda amaçlamışlardır.

1990 yılında Baaj ve 1994 yılında Shih ve Mahmassani, 1982 yılında ki Han ve Wilson ile aynı modeli kullanmışlardır. Aralarında ki farklılık yolcuların kullanacakları yolları seçme mantığı konusunda farklılık göstermektedir. 1994 yılında Shih ve Mahmassani yaptıkları çalışmada optimal araç sayısını, araç hareket sıklıklarının rota koordinasyonu için ayarlanması, transfer zamanlı ve aktarma merkezleri konseptlerini de kullanmışlardır. 1994 yılında Dashora expert-system tabanlı bir model kullanarak maksimum ve minimum miktarı gözeterekten farklı rotaların yerlerinin belirlenmesini amaçlamış ve bununla birlikte eklenebilir otobüs yerleşim faktör “ bekleme zamanlarını kriterlerini gözetmiştir.

1993 yılında Xiong ve Schneider, 1995 yılında Chakraborty et al, 1998 de Pattnaik et al, 2001 de Chien et al, 2001 de Khalage et al, 2002 de Bielli et al, 2003 de Tom ve Mohan ve 2003 yılında Ngamchai ve Lovell yaptıkları çalışmalarla bu konuda isimlerini duyurmaktadırlar.

4.2 Matematiksel Model

Aşağıda göstereceğimiz otobüs hattı çizelgeleme problemi için genel fonksiyondur.

Notasyonlar (Han ve Wilson, 1982):

f_k : herhangi bir k rotasında otobüs hareket sıklığı (otobüs sefer sayısını belirler),

A_k : herhangi bir k rotasında rotayı etkileyen diğer girdiler,

CAP: otobüs ağı içinde kullanılan otobüslerin yolcu kapasiteleri,

q_{ij}^k : Herhangi bir ka noktasında iki durak arası (i-j) yolcu akışı,

g_{ij}^k : k rotasında i ve j arasında akan yolcu sayısını belirleyen genel fonksiyon,

V^{ab} : a ve b düğümleri arasındaki orijine olan uzaklık,

N: otobüs ağındaki düğüm sayıları,

L_k : k rotasındaki iki durak arası mesafe sayıları,

SR: Otobüslerin kullanacağı rota sayısı,

T_k : herhangi bir k rotasında bir döngü için geçen seyahat zamanı (ertelemeler ve beklemler dâhildir),

X_{ij} : farklı rotaların kullandığı aynı duraklar (i-j),

M: Toplam Var Olan Otobüs sayısı

$$\text{Min } J = J (q_{ij}^k, f_k, A_k,) \quad (4.1)$$

$$\text{Yolcu Akış Devri-Atama: } q_{ij}^k = g_{ij}^k (V^{ab}, f_r, A_r) \quad (4.2)$$

$k \in \text{SR}, ij \in L_k, r \in X_{ij}, \text{ ve } a, b \in N$

$$\text{Yükleme Fizibilitesi: } \text{CAP} \times f_k \geq (q_{ij}^k)_{\max} \quad k \in \text{SR} \quad (4.3)$$

$$\text{Toplam Otobüs Sayısı-Filo: } \sum_{k=1}^{SR} T_k \times f_k \leq M \quad (4.4)$$

Amaç fonksiyonu yolcuların bekleme zamanlarını ve kalabalık durumunu içermelidir. Bu yüzden birçok otobüs tam kapasiteyle taşıma yapmak ve tam anlamıyla bekleme zamanlarının ve kalabalıklık durumunun spesifikasyonu oldukça zor olacaktır (Han ve Wilson, 1982).

Bu yüzden çok karabalık olan herhangi bir rotada doluluk seviyesinin minimizasyonun basitleştirilmesi burada ele alınmıştır. Bu durum bekleme zamanlarıyla ve kalabalıklık durumuyla her ne kadar farklı olsa da aynı zamanda ilişkilidir bununla birlikte yoğunluk gösteren otobüs ağlarında kullanımı konusunda kullanımı son zamanlar da benzerdir.

Yükleme fizibil kısıtı verilen zaman diliminde yolcular istedikleri zaman otobüse binebilirlerini gerektiriyor çünkü yeteri kadar kapasite herhangi bir rota için mümkün olarak belirlenmiştir. Bu tüm yolcuların en baştaki otobüse binebilirler anlamına gelmemekte çünkü her rotada yolcular rastsal dalgalanmalar yüzünden bazı otobüslerin tam dolu gidebilir. Bazı yolcular eğer ilk otobüste yer bulamazlar ise kesinlikle arkadan gelen başka bir rotanın otobüsüne binebilirler. Yolcunun tercihi farz edilen alışın atama kuralına göredir. “ her nerede bir veya daha fazla alternatif var ise onun seyahat zamanı minimum seyahat zamanı içinde otobüs hareket sıklığı kuralı uygulanır.” Bu alternatif yollar üzerinde servislerin ilişkili frekans yansımalarını yerleştirme formülüdür

4.3 Optimizasyon İçin Kullanılan İki Aşamalı Yöntem

İlk olarak otobüs yerleşim hattı problem formülasyonu verilmiştir. Burada iki şekilde bahsetmiştik. İlk olarak her rotada gerekli olan otobüs sayısına bağlı olan otobüs hareket sıklığı gerekmektedir. Buna bağlı olarak her rota için ayrı ayrı olarak yükleme fizibilliği belirlenmiştir. Bunlardan sonra belirlenen otobüs sayısının toplamı ile gerekli toplam otobüs sayısı yani filo belirlenmiştir.

İkinci olarak ilk belirlenen toplam otobüs sayısı yani filo kullanılarak, GA yöntemi yardımıyla her rotayı kapsayan bir minimizasyon gerçekleştirilecektir.

4.3.1 Yöntemin İlk Aşaması

Burada amaç olarak;

- İki nokta arasında gidecek yolcu sayısını otobüs başına azaltmak q_{ij}^k ,
 - Ge rekli otobüs sayısı veya sıklığını azaltmak f_k ,
 - Ot obüsle alakalı atanan diğer noktaları azaltmak A_k ,
- şeklinde sıralanabilir.

Sistemi yani problemi ilk olarak şöyle formüle edebiliriz:

$$\text{Min: } Z = \sum_{k=1}^{SR} (T_k \times f_k)$$

$$\text{Yolcu Akış Devri-Atama: } q_{ij}^k = g_{ij}^k (V^{ab}, f_r, A_r) \quad k \in SR \quad (4.5)$$

$$\text{Yükleme Fizibilitesi: } CAP \times f_k \geq (q_{ij}^k)_{\max} \quad k \in SR \quad (4.6)$$

Bu problemin çözümü aşağıda verilen algoritma ile çözülecektir.

Adım 1: Merkezler arası mesafe matrisi, rota ağı ve rotaların ortak yani kesişen durakları-noktaları belirlenmesi,

Adım 2: Varılacak noktaya olan talep matrisini bir sonraki basamakta gösterilen şekilde oluştur.

Adım 3: Her rota için hangi iki durak arasında maksimum akışın olduğunun ortaya konması ve gerekli olan sefer sayısının aşağıdaki formüle göre belirlenir;

$$f_k^{n+1} = \frac{(q_{ij}^k)_{\max}}{2 \times \text{Otobüs Kapasitesi}} \quad (4.7)$$

Eğer çıkan sonuç kesirli bir sayı olur ise bir sonraki tamsayıya yuvarlanılır.

Adım 4: eğer $f_k^{n+1} \sim f_k^n$ durumu her rota için küçük çıkar ise adım 5'e geç. Eğer büyük ise $n=n+1$ yapılır ve adım 2'ye dönülür.

Adım 5: Her rota için gerekli sefer sayısı çıkarılır.

Adım 6: Sefer sayısının ihtiyacını karşılayacak otobüs sayısını aşağıdaki formül ile bulunur;

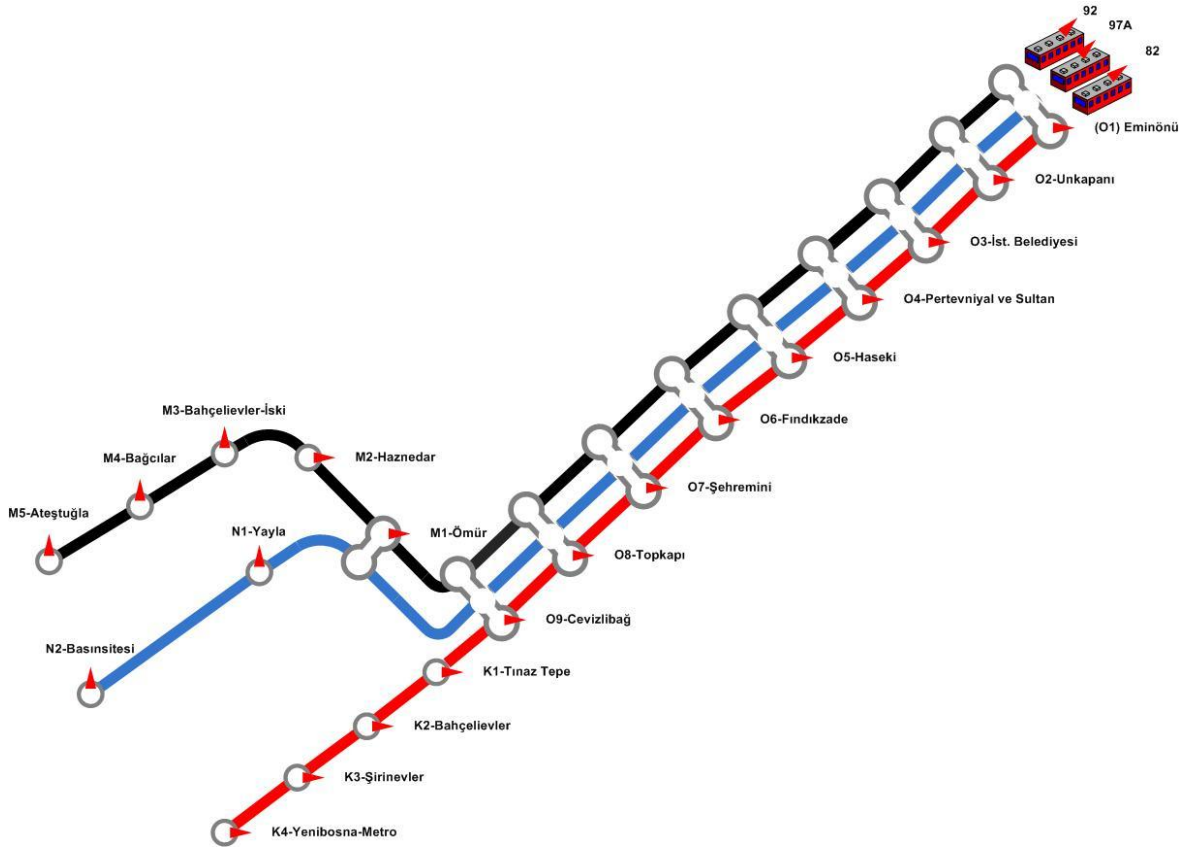
$$N_k = \frac{f_k \times T_k}{\text{Zaman Peryodu}} \quad (4.8)$$

Eğer çıkan sonuç kesirli bir sayı olur ise bir sonraki tamsayıya yuvarlanılır.

Adım 7: Her rota da bulunan otobüsler toplanılarak gerekli toplam otobüs sayısını yani filoyu (W_o) bulunur.

Buradaki algoritmanın herhangi bir kanıtlama şeklinin olmamasına rağmen, bu güne kadarki deneyime bağlı olarak yaklaşık bir sonuç ortaya konulabilmektedir.

Burada ortaya konulan İstanbul da İETT 'ye ait üç tane otobüs toplu taşıma hattı ele alınmıştır. Ve Gösterilen değerler 01.03:2010 tarihine aittir. Aşağıdaki şekilde otobüs hattının rotası gösterilmiştir.



Şekil 4.1 İETT'ye Ait Optimize Edilecek Rota Ağı

Bu şekilde görüldüğü gibi rota üzerinde her üç rotanın da ortak olarak kullandığı 9 durak (O1, O2, O3, O4, O5, O6, O7, O8 ve O9) bulunmaktadır ve iki rotamızda 10 tane durağı (O1, O2, O3, O4, O5, O6, O7, O8, O9 ve M1) ortak olarak kullanmaktadır. Burada normalde her rotada

bulunan gerçek durak sayısı bir gruptandırma yapılaraktan 82 numaralı rota için normalde 24 olan durak sayısı en çok kullanılan 13 durak etrafında gruptandırılmıştır. 92 numaralı rota için normalde 40 olan durak sayısı en çok kullanılan 14 durak etrafında gruptandırılmıştır. 97A numaralı rota için normalde 27 olan durak sayısı en çok kullanılan 14 durak etrafında gruptandırılmıştır.

Aşağıdaki matrisler bize bu üç hatta ait hafta içi herhangi bir günün en yoğun uç saati yani sabah saat 7:00 den saat 11:00'e kadarki süre içerisinde bu belirlenen üç hatta seyahat etmek isteyen müşteri sayıları ortaya konmuştur.

Çizelge 4.1 82, 92, 97A Rotalarına Ait Durak Mesafe Matrisi (Metre)

	O1	O2	O3	O4	O5	O6	O7	O8	O9	K1	K2	K3	K4	N1	N2	M1	M2	M3	M4	M5
O1		1530	2380	3080	7330	4130	4880	5680	7230	8880	11730	13430	13830	12580	13580	11180	12380	13380	18330	19030
O2	1530		950	1550	2200	2600	3350	4150	5700	7350	6750	11900	12300	11050	12050	9650	10850	11850	14800	17500
O3	2380	950		700	1350	1750	2500	3300	4850	6500	9350	11050	11450	10200	11200	8800	10000	11000	13950	16650
O4	3080	1550	700		650	1050	1800	2600	4150	5800	8650	9950	10350	9500	10500	8100	9300	10300	13250	15950
O5	7330	2200	1350	650		400	1250	2050	3600	5250	8000	9700	10100	8850	9850	7450	8650	9650	12600	13300
O6	4130	2600	1750	1050	400		750	1550	3100	4750	7600	9300	9700	8450	9450	6300	7500	8500	11450	14150
O7	4880	3350	2500	1800	1250	750		800	2350	4000	6850	7950	8350	7700	8700	5500	6700	7700	10650	13350
O8	5680	4150	3300	2600	2050	1550	800		1550	3200	6050	7750	8150	6900	7900	3950	5150	6150	9100	11800
O9	7230	5700	4850	4150	3600	3100	2350	1550		1650	4500	6200	6600	5350	6350	2300	3500	4500	7450	10150
K1	8880	7350	6500	5800	5250	4750	4000	3200	1650		2850	4550	4950	3700	4700	0	0	0	0	0
K2	11730	6750	9350	8650	8000	7600	6850	6050	4500	2850		1700	2100	0	0	0	0	0	0	0
K3	13430	11900	11050	9950	9700	9300	7950	7750	6200	4550	1700		400	0	0	0	0	0	0	0
K4	13830	12300	11450	10350	10100	9700	8350	8150	6600	4950	2100	400		0	0	0	0	0	0	0
N1	12580	11050	10200	9500	8850	8450	7700	6900	5350	3700	0	0	0		1000	0	0	0	0	0
N2	13580	12050	11200	10500	9850	9450	8700	7900	6350	4700	0	0	0	1000		0	0	0	0	0
M1	11180	9650	8800	8100	7450	6300	5500	3950	2300	0	0	0	0	0	0		1200	2200	5150	7850
M2	12380	10850	10000	9300	8650	7500	6700	5150	3500	0	0	0	0	0	0	1200		1000	3950	6650
M3	13380	11850	11000	10300	9650	8500	7700	6150	4500	0	0	0	0	0	0	2200	1000		2950	5650
M4	18330	14800	13950	13250	12600	11450	10650	9100	7450	0	0	0	0	0	0	5150	3950	2950		2700
M5	19030	17500	16650	15950	15300	14150	13350	11800	10150	0	0	0	0	0	0	7850	6650	5650	2700	

Çizelge 4.2 82 Numaralı İETT Rota Seyahat Talep Matrisi

DURAKLAR	O1	O2	O3	O4	O5	O6	O7	O8	O9	K1	K2	K3	K4	N1	N2	M1	M2	M3	M4	M5
O1		32	84	251	22	35	21	105	46	22	23	17	101	0	0	0	0	0	0	0
O2			5	15	2	3	2	7	3	2	2	1	6	0	0	0	0	0	0	0
O3				26	3	4	3	12	5	3	3	2	10	0	0	0	0	0	0	0
O4					17	27	17	84	37	17	16	12	52	0	0	0	0	0	0	0
O5						5	3	15	7	3	3	2	9	0	0	0	0	0	0	0
O6							6	29	13	6	6	4	16	0	0	0	0	0	0	0
O7								21	10	5	5	3	12	0	0	0	0	0	0	0
O8									88	43	42	27	90	0	0	0	0	0	0	0
O9										51	52	35	100	0	0	0	0	0	0	0
K1											39	27	73	0	0	0	0	0	0	0
K2												57	154	0	0	0	0	0	0	0
K3													211	0	0	0	0	0	0	0
K4														0	0	0	0	0	0	0
N1															0	0	0	0	0	0
N2																0	0	0	0	0
M1																	0	0	0	0
M2																		0	0	0
M3																			0	0
M4																				0
M5																				

Çizelge 4.3 92 Numaralı İETT Rota Seyahat Talep Matrisi

DURAKLAR	O1	O2	O3	O4	O5	O6	O7	O8	O9	K1	K2	K3	K4	N1	N2	M1	M2	M3	M4	M5
O1		52	137	413	36	57	34	172	113	0	0	0	0	0	0	41	3	1	9	170
O2			9	25	3	4	3	11	7	0	0	0	0	0	0	3	1	1	1	18
O3				42	5	7	4	20	9	0	0	0	0	0	0	3	1	1	2	22
O4					27	44	27	138	60	0	0	0	0	0	0	21	2	1	6	131
O5						8	5	24	22	0	0	0	0	0	0	8	1	1	2	45
O6							9	47	22	0	0	0	0	0	0	8	1	1	2	41
O7								34	20	0	0	0	0	0	0	7	1	1	2	36
O8									145	0	0	0	0	0	0	54	6	1	11	262
O9										0	0	0	0	0	0	37	5	1	14	334
K1											0	0	0	0	0	0	0	0	0	0
K2												0	0	0	0	0	0	0	0	0
K3													0	0	0	0	0	0	0	0
K4														0	0	0	0	0	0	0
N1															0	0	0	0	0	0
N2																0	0	0	0	0
M1																	3	1	7	169
M2																		1	1	14
M3																			1	2
M4																				61
M5																				

Çizelge 4.4 97A Numaralı İETT Rota Seyahat Talep Matrisi

DURAKLAR	O1	O2	O3	O4	O5	O6	O7	O8	O9	K1	K2	K3	K4	N1	N2	M1	M2	M3	M4	M5
O1		46	122	368	32	51	30	153	101	0	0	0	0	2	164	37	0	0	0	0
O2			8	22	2	4	2	10	6	0	0	0	0	1	16	2	0	0	0	0
O3				38	4	6	4	18	8	0	0	0	0	1	21	3	0	0	0	0
O4					24	39	24	123	54	0	0	0	0	2	120	19	0	0	0	0
O5						7	4	21	20	0	0	0	0	1	42	7	0	0	0	0
O6							8	42	19	0	0	0	0	1	38	7	0	0	0	0
O7								31	18	0	0	0	0	1	34	6	0	0	0	0
O8									129	0	0	0	0	1	246	48	0	0	0	0
O9										0	0	0	0	1	309	33	0	0	0	0
K1											0	0	0	0	0	0	0	0	0	0
K2												0	0	0	0	0	0	0	0	0
K3													0	0	0	0	0	0	0	0
K4														0	0	0	0	0	0	0
N1															18	0	0	0	0	0
N2																0	0	0	0	0
M1																	0	0	0	0
M2																		0	0	0
M3																			0	0
M4																				0
M5																				

Çizelge 4.5 Üç İETT Hattı İçin Toplam Seyahat Talep Matrisi

DURAKLAR	O1	O2	O3	O4	O5	O6	O7	O8	O9	K1	K2	K3	K4	N1	N2	M1	M2	M3	M4	M5
O1		130	343	1032	90	143	85	430	260	22	23	17	101	2	164	78	3	1	9	170
O2			22	62	7	11	7	28	16	2	2	1	6	1	16	5	1	1	1	18
O3				106	12	17	11	50	22	3	3	2	10	1	21	6	1	1	2	22
O4					68	110	68	345	151	17	16	12	52	2	120	40	2	1	6	131
O5						20	12	60	49	3	3	2	9	1	42	15	1	1	2	45
O6							23	118	54	6	6	4	16	1	38	15	1	1	2	41
O7								86	48	5	5	3	12	1	34	13	1	1	2	36
O8									362	43	42	27	90	1	246	102	6	1	11	262
O9										51	52	35	100	1	309	70	5	1	14	334
K1											39	27	73	0	0	0	0	0	0	0
K2												57	154	0	0	0	0	0	0	0
K3													211	0	0	0	0	0	0	0
K4														0	0	0	0	0	0	0
N1															18	0	0	0	0	0
N2																0	0	0	0	0
M1																	3	1	7	169
M2																		1	1	14
M3																			1	2
M4																				61
M5																				

Yukarıda gösterilen matrislerde herhangi bir noktadan herhangi bir noktaya gitmek isteyen müşteri sayısını göstermektedir. Örneğin 97A rotasında O3 durağından O8 durağına gitmek isteyen müşteri sayısı 18 kişi bulunmaktadır. 82 rotasında seyahat eden müşterilerden ise 12 kişi iken 92 numaralı rotada seyahat eden müşteri sayısı ise 20 kişidir. Toplam olarak bu iki durak arasında seyahat etmek isteyen müşteri sayısı ise 50 kişidir.

Burada dikkati çeken her üç hattın son durakları arasındaki seyahat eden yolcu sayısının fazlalığı olmaktadır. Bunun sebebi yapılan analiz Akbil verilerine dayanmasıdır. “Akbil İstanbul da toplu taşımalar da seyahat etmek için kullanılan seyahat kartı.” Şoför akbili kullanan müşterilerin tam anlamıyla nereden nereye gittikleri belirlenememesi ve buna bağlı olarak bu hatlarda şoför akbil’ini kullanan müşterileri son duraklar arasında seyahat etmek isteyen müşteri olarak kabul etmemizden kaynaklanmaktadır.

Her rota için hangi iki durak arasında maksimum akışın olduğunun ortaya konması ve gerekli olan sefer sayısının aşağıdaki formüle göre belirlenir ise

$$f_k^{n+1} = \frac{(q_{ij}^k)_{\max}}{2 \times \text{Otobüs Kapasitesi}}$$

Rota 82 için

$$f_1^1 = \frac{(q_{1,4}^1)}{2 \times \text{Otobüs Kapasitesi}} = \frac{251}{60} = 4.18 = 5 \text{ otobüs}$$

$$f_1^2 = \frac{(q_{2,4}^1)}{2 \times \text{Otobüs Kapasitesi}} = \frac{15}{60} = 0.25 = 1 \text{ otobüs}$$

$$f_1^3 = \frac{(q_{3,4}^1)}{2 \times \text{Otobüs Kapasitesi}} = \frac{26}{60} = 0.43 = 1 \text{ otobüs}$$

$$f_1^4 = \frac{(q_{4,8}^1)}{2 \times \text{Otobüs Kapasitesi}} = \frac{84}{60} = 1.4 = 2 \text{ otobüs}$$

$$f_1^5 = \frac{(q_{5,8}^1)}{2 \times \text{Otobüs Kapasitesi}} = \frac{15}{60} = 0.25 = 1 \text{ otobüs}$$

$$f_1^6 = \frac{(q_{6,8}^1)}{2 \times \text{Otobüs Kapasitesi}} = \frac{29}{60} = 0.48 = 1 \text{ otobüs}$$

$$f_1^7 = \frac{(q_{7,8}^1)}{2 \times \text{Otobüs Kapasitesi}} = \frac{21}{60} = 0.35 = 1 \text{ otobüs}$$

$$f_1^8 = \frac{(q_{8,13}^1)}{2 \times \text{Otobüs Kapasitesi}} = \frac{90}{60} = 1.50 = 2 \text{ otobüs}$$

$$f_1^9 = \frac{100}{60} = 1.67 = 2 \text{ otobüs}$$

$$f_1^{12} = \frac{100}{60} = 1.67 = 2 \text{ otobüs}$$

$$f_1^{10} = \frac{73}{60} = 1.22 = 2 \text{ otobüs}$$

$$f_1^{13} = \frac{211}{60} = 3.51 = 4 \text{ otobüs}$$

$$f_1^{11} = \frac{154}{60} = 2.56 = 3 \text{ otobüs}$$

Rota 92 için;

$$f_2^1 = 7 \text{ otobüs}$$

$$f_2^5 = 1 \text{ otobüs}$$

$$f_2^2 = 1 \text{ otobüs}$$

$$f_2^6 = 1 \text{ otobüs}$$

$$f_2^3 = 1 \text{ otobüs}$$

$$f_2^7 = 1 \text{ otobüs}$$

$$f_2^4 = 3 \text{ otobüs}$$

$$f_2^8 = 5 \text{ otobüs}$$

$$f_2^9 = 6 \text{ otobüs}$$

$$f_2^{10} = 3 \text{ otobüs}$$

$$f_2^{11} = 1 \text{ otobüs}$$

$$f_2^{12} = 1 \text{ otobüs}$$

$$f_2^{13} = 1 \text{ otobüs}$$

Rota 97A için

$$f_3^1 = 7 \text{ otobüs}$$

$$f_3^2 = 1 \text{ otobüs}$$

$$f_3^3 = 1 \text{ otobüs}$$

$$f_3^4 = 3 \text{ otobüs}$$

$$f_2^5 = 1 \text{ otobüs}$$

$$f_3^6 = 1 \text{ otobüs}$$

$$f_3^7 = 1 \text{ otobüs}$$

$$f_3^8 = 5 \text{ otobüs}$$

$$f_3^9 = 6 \text{ otobüs}$$

$$f_3^{10} = 1 \text{ otobüs}$$

$$f_{1\text{-top}} = 27$$

$$f_{2\text{-top}} = 32$$

$$f_{3\text{-top}} = 27$$

Sefer sayısının ihtiyacını karşılayacak otobüs sayısını aşağıdaki formül ile bulunur;

Örneğimizde günün en yoğun olduğu 3 saatlik bir zaman periyodu alındığını daha önce

belirtmiştik.

Aşağıda ki çizelgede gösterilen değerler sistem içerisinde kullanılan gerçek değerlerdir.

Çizelge 4.6 İETT de Üç Hat İçin Kullanılan Gerçek Değerler Listesi

	82	92	97A
Ticari Hız (km/h)	32,35	27,85	25,85
Sefer Müddeti (Gidiş-Dönüş) dk	115	165	130
Sefer Km (Gidiş Dönüş)	62	76,6	56
Mevcut Araç Sayısı (İş günü)	5+7=12	10+9=19	6+8=14

Bir İETT şoför'ün aylık gideri 5.200 TL

Bir otobüs km'de 0.50 lt yakıt tüketir

$$N_1 = \frac{f_1 \times T_1}{\text{Zaman Peryodu}} = \frac{26 * 60}{3 * 60} = 8.81 = 9 \text{ otobüs}$$

$$N_2 = \frac{f_2 \times T_2}{\text{Zaman Peryodu}} = \frac{32 * 85}{3 * 60} = 15.11 = 16 \text{ otobüs}$$

$$N_3 = \frac{f_3 \times T_3}{\text{Zaman Peryodu}} = \frac{27 * 65}{3 * 60} = 9.75 = 10 \text{ otobüs}$$

Çıkan sonuç kesirli bir sayı olduğu için biz sonucu bir sonraki tamsayıya yuvarlarız. Çünkü otobüs sayılarını kesirli kullanamayız. Rotalarda kullanılan otobüs tamsayı şeklinde olmalıdır.

Her rota da bulunan otobüsler toplanılarak gerekli toplam otobüs sayısını yani filo (W_o) bulunur.

$$W_o = N_1 + N_2 + N_3$$

$$W_o = 9 + 16 + 10 = 35 \text{ otobüs gereklidir.}$$

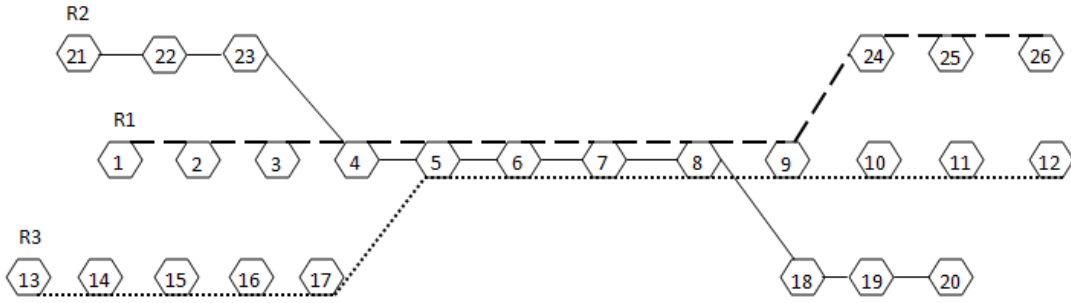
4.4 Yöntemin İkinci Aşaması

Bir önceki aşamada gerekli olan toplam otobüs sayısı yani filo her rota için ayrı ayrı hesaplanarak bulundu. Fakat tüm rotaların birlikte değerlendirilerek filo sayısında herhangi bir minimizasyon işlemi gerçekleştirilmedi. Minimizasyon işlemi yapmamızdaki amaç rotalarda seyir halinde bulunan otobüslerin gereğinden fazla olma olasılığıdır. Tabii eğer herhangi bir gereğinden fazlalık söz konusu olmadığı durumlarda herhangi bir minimizasyon işleminin gerçekleştirilmesi beklenemez. Bu durumu çözmek amaçlı birçok konu olmasına rağmen biz iki konu üzerine yoğunlaşacağız.

1.

Fa

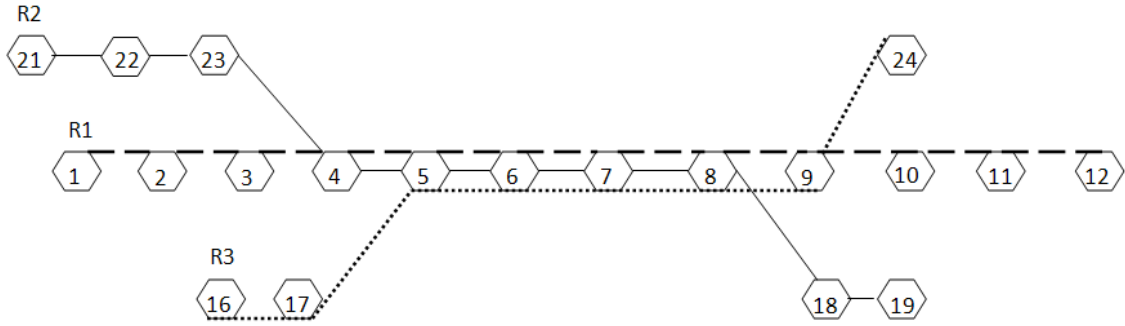
Farz edelim ki elimizde 3 tane rota (R1, R2 ve R3) bulunmakta. Bu üç rotanın da ortak olan otobüs durakları ve rotaların belirli bir kısmının ortak olduğunu varsayalım. Bu rotalarda ilk aşamada belirlenen minimum sayıdaki otobüsler burada maksimum yolcu akışını gerçekleştirsin. Maksimum yolcu talebinin bu duraklar genelde oluşturmakta ve rotaların araç kapasitelerini bu noktalar belirlemektedir. Eğer biz bu Üç rotadan herhangi birisinden bir veya iki otobüs çıkarırsak rotalardaki otobüsler birbirlerini tamamlayacak ve belki herhangi bir kargaşa ortaya çıkmayacak ve talep karşılanabilir.



Şekil 4.1 Temsili Otobüs Ağı 1

2. Farz edelim ki bu elimizde ki 3 rota (R1, R2 ve R3) den biri R1 öteki iki R2 ve R3 rotalarından uzun olsun. Burada sabit zaman aralığında gelen otobüs sayısı direk olarak otobüs hattı uzunluğuyla (tur zamanı da olabilir) doğru orantılı olmuştur. Bu uzun rota R1 üzerindeki otobüs sayısı aşırı yüklenmiştir durumda olacaktır. Eğer biz bu üç rotadan R1 den iki otobüs çıkarıp R2 veya R3'e bir otobüs eklersek, belki de bu ortak noktada gerekli olan yolcu talebini karşılayacaktır ve herhangi bir kargaşa da

oluşmayacaktır. Bu durumda gerekli toplam otobüs sayısından 1 tane otobüs çıkarılarak filo sayısında bir düşüş sağlayacak minimizasyon gerçekleşmiş olacaktır. İşte bu filo sayısındaki düşüşünün olma olasılığının varlığı ikinci aşama optimizasyon işlemi GA ile yapılacaktır.



Şekil 4.2 Temsili Otobüs Ağı 2

4.4.1 Genetik Algoritma

Genetik Algoritma adı ilk olarak 1975 yılında Michigan Üniversitesinde Profesör John Holland tarafından ortaya konulmuştur. Genetik algoritma bilgisayar tabanlı araştırma ve doğal genetik ve doğal seçim mekaniği ile çalışan bir optimizasyon algoritmasıdır (Goldberg, 1989).

4.4.2 GA'nın Çalışma Prensipleri

Genetik algoritma ilk olarak kromozom uzunluğunu belirleyerek ilk popülasyonu rastsal olarak oluşturularaktan başlar. Sonra her kromozom değeri hesaplanır. Sonra ilk popülasyon üç operatör “yeniden üretim, çaprazlama ve mutasyon” tarafından işlenip beklenen daha iyi bir popülasyon elde edilir. Eğer durdurma kriteri yakalanmadığı takdirde işlemler tekrarlanır. Bu durum durdurma kriteri elde edilinceye kadar devam eder. Bu operatörlerin bir döngüsü sonucunda bir jenerasyon oluşturulmuş olur. Aşağıda bir genetik algoritmanın basitçe prosedürü gösterilmiştir.

```

begin
  Initialize population of string;
  Compute fitness of population;
  Repeat
  Reproduction;

```

Crossover;
Mutation;
Compute fitness of population;
Until (Termination criteria);
End

Biz bu çalışmamızda MATLAB programını kullanarak probleme çözüm bulmaya çalışacağız.

1. Kromozom Uzunluğu
2. Kodlama Ve Kod çözümü
3. Değerlendirme
4. Genetik Algoritma Operatörleri
5. Sonlandırma Kriteri

4.4.3 İkinci Aşama Optimizasyon İçin Problemin Formüle Edilmesi

İlk aşama optimizasyon da belirttiğimiz gibi her rotadaki otobüs sayısı N_k dır. Gerekli otobüs sayısını bu N_k değeri etrafında yapmamız gerekir yoksa anlamsız bir araştırma olacaktır. Bunun için araştırmanın yapılacağı belirli bir alanın belirlenmesi gerekir. Bu çalışmada daha önceden belirlenen N_k değerinin etrafında yapılacak araştırma alanı için 8 değeri kullanılacaktır.

Örneğin, eğer herhangi bir k rotasında N_k 20 alınır ise araştırma alanı 16 ile 23 değerleri arasında yapılacaktır. Bu araştırma alanının 8 olması yapılacak araştırmayı daha gerçekçi kılmaktadır. Genetik Algoritmada değişkenler ikili sayı sistemiyle kromozomlar oluşturulacak ve 2^n kadar farklı değerdeki değişken için n kadar bit gerekecektir. Burada bir bit 2 otobüse iki bit 4 otobüse üç bit ise 8 otobüse ve n bit 2^n otobüse karşılık gelmektedir.

Eğer biz örneğimizin çözümü için her rotanın yeniden yapılanmasını sağlayacak araştırma alanını 2 bit olarak ele alsaydık, bu araştırma alanımızı 4 otobüsle sınırlandırıp çok dar bir alana sıkıştırıp optimal sonucu bu alanın dışında bırakabiliriz. Diğer taraftan eğer biz örneğimizin çözümü için her rotanın yeniden yapılanmasını sağlayacak araştırma alanını 4 bit olarak ele alsaydık, bu araştırma alanımızı 16 otobüse çıkarıp çok geniş bir alanda optimal sonucu aramamıza sebep olur ve bu bize fizibil olmayan sonuçlar ortaya koyabilir. Örneğin

herhangi bir rota da otobüs sayısını sıfır veya negatif bir değer yapabilir.

Bundan dolayı biz araştırma alanımızı 8 otobüs den oluşturursak, her rota için 3 bit gerekmektedir. Bunların yanında kromozom uzunluğumuz herhangi bir k rotası için 3 x k bit olmayı gerektirmektedir. Çünkü araştırma alanımızı 8 otobüs seçmiştik ve herhangi bir k rotası için gerekli minimum ve maksimum otobüs sayısı limiti aşağıdaki gibi hesaplanır. (Kıdwai, Deb, Marwah ve Karim, 2005)

$$N_{kmin} = N_k - 4 \quad (4.9)$$

$$N_{kmax} = N_k + 3 \quad (4.10)$$

N_k ilk aşama optimizasyonda k rotası için belirlenen otobüs sayısıdır. İkinci aşama optimizasyon için sistem aşağıdaki gibi formülize edilebilir.

Minimize

$$N_{kmin} \leq N_k \leq N_{kmax} \quad k \in SR$$

(4.11)

$$\sum_{k=1}^{SR} N_k < W_0 \quad (4.12)$$

$$\text{Yükleme Fizibilitesi: } CAP \times f_k \geq (q_{ij}^k)_{max} \quad k \in SR \quad (4.13)$$

N_k :k rotasındaki otobüs sayısı

W_0 :Filo yani toplam otobüs sayısı “ilk aşama optimizasyon işleminde belirlenmişti”.

4.4.4 Genetik Algoritma Sonuçları

Bizim sistemimizde yani problemin çözümünde kullanacağımız otobüs sayısının tamsayı olması gerekmektedir. GA için kullanılan adımlar aşağıdaki gibi sıralayabiliriz.

Adım 1- Her rota için gerekli minimum otobüs sayısı N_{kmin} belirlenir;

$$N_{kmin} = N_k - 4 \quad (4.9)$$

Adım 2- GA operatörlerinde seçim operatörü, çaprazlama operatörü ve mutasyon operatörü seçilir. Popülasyon büyüklüğünün belirlenmesi, çaprazlama ve mutasyon olasılıklarının belirlenmesi ve hesaplanabilir maksimum jenerasyon sayısının belirlenmesi.

Adım 3- GA rastgele başlangıç popülasyonunu oluşturur. Rota sayısına bağlı kromozom uzunluğu hesaplanır. Mesela k kadar rotamız olsun kromozom uzunluğu $k \times 3$,

Adım 4- Dördüncü adımda kromozomun kod çözümü gerçekleştirilir ve gerçek otobüs sayısı belirlenir bunun için aşağıdaki formül belirlenir; otobüs sayısı her rota için sırasıyla

$$N_k = N_{kmin} + \text{çözülmüş kod değeri (3 bit)} \quad (4.14)$$

Adım 5- Toplam otobüs sayısını yani filo sayısını tüm rotalar için bulunan N_k değerlerini toplayarak bulunur. Eğer bunun filo değeri en başta bulunan filo sayısına eşit veya büyük olur ise atanan uygunluk değeri çok düşük değerlidir. Eğer küçük çıkar ise adım 6'ya geçilir.

Adım 6- N_k değerini her rota için kullanarak sıklık değeri f_k hesaplanır.

Adım 7- Eğer yükleme fizibil kısıt $CAP \times f_k \geq (q_{ij}^k)_{max}$ ihlal ediliyor ise atanan uygunluk değerine küçük bir değer atanır. Eğer ihlal etmiyor ise uygunluk değeri aşağıda ki formül yardımıyla hesaplanır;

$$\text{Uygunluk Değeri} = \frac{C}{1 + \sum N_k} \quad (4.15)$$

C: Objektif fonksiyonu normalleştirmek için kullanılan sabit bir değer.

Adım 8- Eğer kromozom halindeki tüm popülasyon işlendiğinde, jenerasyonda ki en iyi ve ortalama uygunluk değerini hesaplanır ve bitirme kriteriyle karşılaştırılır. Eğer durdurma kriterini sağlamıyor ise bir sonraki kromozom işleme sokulur.

Adım 9- Eğer bulunan popülasyon beklenen maksimum popülasyona eşit ise problem sonlandırılır. Verilen minimum filo sayısı optimum olarak belirlenir-işaretlenir. Eğer değil ise sistem yeniden; yeniden üretim, çaprazlama ve mutasyon operatörleri son popülasyon üzerine tekrar uygulanır ve yeni bir kromozom popülasyonu bulunur. Bulunan yeni popülasyon tekrar hesaplanır.

Uygulamamızda:

Amaç:

max-

$z=(CAP)$

$$* Bilet * W_o * Ort. Tur - Yakıt \left(\frac{lt}{km} \right) * (RotaUz_{82} (km) * N_1 + RotaUz_{92} (km) * N_2 + RotaUz_{97A} (km) * N_3) - Şof. Maaş. \frac{(ay)}{Gün * Saat} * Peryot * W_o$$

MATLAB için uygunluk fonksiyonumuz;

$$Min_z = -(60 * 1.5 * 1.5 (31 + x(1) + x(2) + x(3)) - 50 * 3 * 20 * (16 + x(1) + x(2) + x(3)) + (5200 / (30 * 8)) * 3 * (16 + x(1) + x(2) + x(3)));$$

	82	92	97A
Ticari Hız (km/h)	32,35	27,85	25,85
Sefer Müddeti (Gidiş-Dönüş) dk	115	165	130
Sefer Km (Gidiş Dönüş)	62	76,6	56
Mevcut Araç Sayısı (İş günü)	12	19	14
Bir İETT şoför'ün aylık gideri	5.200 TL		
Bir otobüs km'de 0.50 lt yakıt tüketir			

$$W_o = 35$$

$$N_{kmin} = N_k - 4$$

$$N_{kmax} = N_k + 3$$

$$N_{1min} = N_1 - 4 = 9 - 4 = 5 \text{ ve } N_{1max} = N_1 + 3 = 9 + 3 = 12$$

$$N_{2min} = N_2 - 4 = 16 - 4 = 12 \text{ ve } N_{2max} = N_2 + 3 = 16 + 3 = 19$$

$$N_{3min} = N_3 - 4 = 10 - 4 = 6 \text{ ve } N_{3max} = N_3 + 3 = 10 + 3 = 13$$

Başlangıç Popülasyonu: 20

Seçim Operatörü: Binary tournament

Çaprazlama Operatörü: Tek noktalı Çaprazlama Operatörü

Çaprazlama Olasılığı: 0.80

Mutasyon Operatörü: Matlab de varsayılan

Mutasyon Olasılığı: 0.01

Hesaplanabilir maksimum Jenerasyon Sayısı: 100

Kromozom Uzunluğu = $k * 3 = 3 * 3 = 9$

k: 3 "Rota sayısı"

Elde edilen sonuç: [1 0 1 0 1 0 1 0 0]

$$N_1 = N_{1min} + \text{çözülmüş kod değeri}(3 \text{ bit}) = 5 + 5 = 10$$

$$N_2 = N_{2min} + \text{çözülmüş kod değeri}(3 \text{ bit}) = 12 + 2 = 14$$

$$N_3 = N_{3min} + \text{çözülmüş kod değeri}(3 \text{ bit}) = 6 + 4 = 10$$

$$W_o = 10 + 14 + 10 = 34$$

" $W_o = 35$ "eski değer < " $W_o = 34$ "yeni değer

$$N_1 = \frac{f_1 \times T_1}{\text{Zaman Peryodu}} \Rightarrow 10 = \frac{f_1 \times 60}{3 * 60} \Rightarrow f_1 = 30$$

$$N_2 = \frac{f_2 \times T_2}{\text{Zaman Peryodu}} \Rightarrow 14 = \frac{f_2 * 85}{3 * 60} \Rightarrow f_2 = 29.6$$

$$N_3 = \frac{f_3 \times T_3}{\text{Zaman Peryodu}} \Rightarrow 10 = \frac{f_3 * 65}{3 * 60} \Rightarrow f_3 = 27.7$$

Burada da kısıtımızın karşılandığı görülmektedir. Bulunan değerlerin üçü de yani gerekli sefer sayısı maksimum 32'den küçüktür. Ve aynı zaman da duraklar arasındaki maksimum gerekli sefer sayısından da büyüktür.

$$\begin{aligned} \max - z &= 60 * 1,50(tl) * 34 * 1.5 - 0.50 \left(\frac{lt}{km} \right) \\ &* (62(km) * 10 + 76.6(km) * 14 + 130(km) * 10) - \frac{5200(ay)}{30 * 8} * 3 * 34 \end{aligned}$$

$$= 4590 - 1496,20 - 2210 = 883,80 \text{ TL}$$

5. SONUÇLAR ve DEĞERLENDİRME

Geleneksel yaklaşımlarla bir otobüs hattının belirlenmesi probleminin ortaya konması ve bununla birlikte gerekli olan problemin çözümü için teori ile reel’i birleştirmek oldukça zordur. Burada yaptığımız oldukça kompleks bir yapıya ait küçük çaplı bir örnektir. Genetik Algoritma yapısı gereği geniş çaplı ve kompleks yapıları çözmek amaçlı bir hesaplama yöntemidir.

Biz bu çalışmamızda hafta içi herhangi bir günün son derece yoğun olan 3 saatlik bir periyodunu inceleme altına aldık. Burada eğer söz konusu periyotta mevcut talebe gerekli otobüs arzını gerçekleştirmeyi başarılırsa elde edilen optimum otobüs sayısı günün diğer saatlerinde gerçekleşen talebi de rahatlıkla karşılayabilir. Bunun için günün en yoğun saatleri olan sabah 7:00 ile 10:00 arasındaki saat dilimi inceme için uygun bulunmuştur.

Burada çalışmamızın iki temel hedefi bulunmaktadır. İlk olarak, bilimsel ilkeler ışığında kriterler belirlendi. Daha sonra bu kriterlere uygun olarak hatlarda bulunana durak bazlı talebi karşılamak üzere gereken optimum otobüs sayısı elde edildi. İncelenen hat bir toplu taşıma hattı olduğu için bu aşamada dengelemeye çalıştığımız hattın rotalarının yeniden çizilmesi ve otobüs duraklarının atamalarının yeniden gerçekleştirilmesi uygun görülmemiştir. Bunun sebebi İstanbul’da otobüs güzergâhlarının diğer normal araçlara oranla çok kısıtlı olmasıdır. Aşağıdaki çizelge 5.1 mevcut durumun optimal durum ile karşılaştırılmasını gösterir.

Çizelge 5.1 Karşılaştırma Çizelgesi

Rotalar	82	92	97A
Mevcut Araç Sayısı (İş günü)	12	19	14
Optimum Hale Getirilmiş Araç Sayısı	9	16	10
Genetik Algoritma İle Optimal Hale Gelmiş Araç Sayısı	10	14	10
Toplam Kazanç	2+5+4=11		

Bu çizelgede de görüldüğü gibi 82 numaralı rotada işlem gören 12 otobüs 9 otobüse indirilmiştir. Yine aynı şekilde yapılan hesaplamalar sonucu 92 numaralı rotada işlem gören otobüs sayısı 19’dan 16’ya ve 97A numaralı rotada işlem gören otobüs sayısı da 14’den 10’a

düşürülmüştür. Burada kazanılan toplam otobüs sayısı 10 olarak hesaplanabilir.

İkinci olarak, bu çalışma, optimize edilmiş otobüs sayısının MATLAB Genetik Algoritma Tool' u kullanılarak bu üç hatta işlem gören otobüs sayısında herhangi bir azaltmanın yapılabileceğinin sorgulanmasını hedeflemektedir. Burada yapılan işlemler sonrasında yukarıdaki çizelge 5.1 incelendiğinde 82 numaralı rotada işlem gören otobüs sayısı dokuz otobüsten on otobüse çıkarılmıştır. Yine aynı şekilde yapılan hesaplamalar sonucu 92 numaralı rotada işlem gören otobüs sayısı 16'dan 14'e indirilmiş ve 97A numaralı rotada işlem gören otobüs sayısı da hiçbir değişiklik yapılmamıştır. Burada kazanılan toplam otobüs sayısı bir tane olarak hesaplanabilir. Kazanılan toplam otobüs sayısı 11 olmuştur. Ayrıca bu çalışmanın sonunda İETT' nin karı da hesaplanmıştır.

Geliştirilen bu model, İETT'nin sahip olduğu toplu taşıma ağından üç tanesini içermektedir. Bu modelde toplam otobüs sayısını azaltmak için Genetik Algoritma kullanılmıştır. Geliştirdiğimiz model sonucunda otobüs sayısı belirlenen hatlarda sırasıyla 10, 14 ve 10 şeklinde belirlenmiştir. Otobüs sayısındaki azalma ise sırasıyla 2,5 ve 4 adettir. Bu model otobüs sayısında azalma sağlarken aslında çeşitli geliştirmelerle daha büyük hatlarda da bu model uygulanarak toplam İETT otobüs sayısında büyük azalma sağlanabilir.

Bununla birlikte ileriki çalışmalarda bu model daha karmaşık ve daha geniş çaplı problemlere uygulanabilir. Sistemin durum analizi aşamasında simülasyon gibi analiz ve modelleme tekniklerinin kullanılması incelenen sisteme uygun geliştirmelerin yapılmasında yardımcı olacaktır. Ayrıca çalışma için veri toplama ve veri analizi aşamasında karşılaşılan güçlükleri hafifletmek için İETT'de bulunan veritabanlarındaki veriler düzenlenmeli, bu veritabanından çekilen bilgilerin efektif olarak kullanılması için de veri madenciliği gibi verilerin toplanması, analizi ve yorumlanması konularında yardımcı olan tekniklerin kullanılmasına eğilmelidir. Problemin çözümü aşamasında daha fizibil çözümler bulmak için genetik algoritma parametrelerinde değişiklikler yaparak farklı deney setleri oluşturulmalı ve bu deney setlerinin sonuçları incelenerek en iyi sonucu veren set problemlere uygulanmalıdır.

KAYNAKLAR

Kitaplar

Holland, John H. (1992) **Adaptation in Natural and Artificial Systems : An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence Complex Adaptive Systems.** MIT Press

Elmas Çetin. (2007). **Yapay Zeka Uygulamaları.** Seçkin

Lakhmi C. Jain; N.M. Martin. (1998). **Fusion of Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications.** CRC Press, CRC Press LLC

Melanie Mitchell. (1999) **An Introduction to Genetic Algorithms.** A Bradford Book The MIT Press Cambridge, Massachusetts London, England

Mitsuo Gen, Runwei Cheng, Lin Lin. (2008). **Network Models and Optimization.** Springer-Verlag London Limited

Nabiyev Vasif V. (2005). **Yapay Zeka Problemler, Yöntemler Ve Algoritmalar.** Seçkin

Nils J. Nilsson. (1998) **Artificial Intelligence, A New Syntesis.** Morgan Kaufmann Publishers. Inc

Patriksson Michael ve Martine Labbé. (2002) **Transportation Planning.** Kluwer Academic Publishers New York, Boston, Dordrecht, London, Moscow

Şen Zekai. (2004). **Genetik Algoritmalar ve En İyileme Yöntemleri.** Su Vakfı İstanbul

Taşkın Çağatay ve Gül G. Emel. (2009). **Sayısal Yöntemlerde Genetik Algoritmalar.** Alfa Aktüel

Ying-ping Chen. (2006). **Extending the Scalability of Linkage Learning Genetic Algorithms Theory & Practice.** Springer-Verlag Berlin Heidelberg Printed in The Netherlands.

Makaleler

Beasley David, David R. Bully, Ralph R. Martinz.(1993). **A Sequential Niche Technique for Multimodal Function Optimization.** UWCC COMMA Technical Report No: 93001

Bielli Maurizio, Massim İliano Caramia ve Pasquale Carotenuto. (2000). **Genetic Algorithms**

in Bus Network Optimization. Pengamon Transportation Research Part C 10 (2000) 19-34.

Chakroborty Partha ve Tathagat Dwivedi. (2002). **Optimal Route Network Design For Transit Systems Using Genetic Algorithms.** Engineering Optimization, 2002, Vol. 34(1), pp. 83–100.

Erick Cantu Paz (1995) **A Summary of Research on Parallel Genetic Algorithms.** IlliGAL Report No. 95007

Farhan A. Kidwai, Kalyanmoy Deb, Baldev R. Marwah ve Mohamed R. Karim, (2005) **A Genetic Algorithm Based Bus Scheduling Model For Transit Network.** Proceedings of the Eastern Asia Society for Transportation Studies, Vol. 5, pp. 477 - 489, 2005

Han, A. F ve Wilson, N. M. H (1982). **The Allocation of Buses in Heavily Utilized Networks With Overlapping Routes,** Transportation Research Part B, Vol. 16, No.3, pp 221 -232.

Genetic Algorithm and Direct Search Toolbox For Use with MATLAB ve

Goldberg, D.E. (1994) **Genetic and evolutionary algorithms come of age.** Communications of the ACM

Kalinli Adem ve Dervis Karaboga. (2004). **Training recurrent neural networks by using parallel tabu search algorithm based on crossover operation.** Engineering Applications of Artificial Intelligence Volume 17, Issue 5, August 2004, Pages 529-542

Koza Jhon. R. (1992). **A genetik approach to Econometric Modelling.** In Bourgine, P. And B. Walliser(eds.), Economic and cognitive Science, Perganom Press, s. 57-75

S. Behzadi , Ali A. Alesheikh. **A Pseudo Genetic Algorithm For Solving Best Path Problem.** Dept. of GIS, Faculty of Geodesy and Geomatics Engineering, K.N. Toosi University of Technology Valiasr Street, Mirdamad Cross, Tehran, P.C. 19967-15433

TEACHING GENETIC ALGORITHM USING MATLAB. Int. J. Elect. Enging. Educ., Vol. 36, pp. 139–153. Manchester U.P., 1999. Printed in Great Britain,

Whitley Darrel. **A Genetik Algoritma Tutorial. Computer Science Department,** Colarado State University Fort Collins, CO 80523

Wright Alden H. **Genetic Algorithms for Real Parameter Optimization.** Department of Computer Science University of Montana Missoula, Montana 59812

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Soyadı, Adı **BALIN, ABİT**
Doğum Tarihi – Yeri 20.01.1983 -ELAZIĞ

İŞ/STAJ - TECRÜBE

- 2007-2009
- Şirket - Görev **Şitoğlu Gıda MALATYA**
“Endüstri Mühendisliği, Proje Yönetimi, Fabrika Yönetimi, Kalite Güvence, Lojistik”
Tarih (05.06.2005- (05.08.2005)
 - Şirket - Görev **Türk Hava Yolları Yeşilköy İstanbul**
“BT Proje Yönetimi, Network, Sistem Yönetimi, Yönetim Bilgi Sistemleri (MIS) ve kontrol açısından incelenmesi, bu firmanın örgütteki yerini ve yaptığı işleri öğrenme.”
Tarih (15.05.2005)-(03.09.2006)
 - Şirket - Görev **Uzel Makina Topkapı İstanbul**
“İşletmenin tedarik, stok, üretim süreci, mamuller, dağıtım, maliyetler, verimlilik, mümkün olduğunda pazarlama, finans ve iş gören yönü ile yönetim fonksiyonları, planlama, örgütleme, yürütme, koordinasyon ve kontrol açısından incelenmesi, bu firmada mühendislerinin örgütteki yerini ve yaptığı işleri öğrenme.”
Tarih 15.03.2005)-(13.06.2006)
 - Şirket - Görev **Veksan Aydınlatma Hadım Köy İstanbul**
“Araştırma ve Geliştirme, Ürün Geliştirme, Bakım / Onarım Mühendisliği, Bakım Teknisyenliği, Endüstri Mühendisliği, Fabrika Yönetimi, İş Güvenliği Mühendisi, Kalite Güvence”

EĞİTİM

- Tarih
 1. 2008 – 2010
 2. 2004 – 2008
 3. 2002 – 2004
 4. 1997 – 2000
- Eğitim Kurumunun Adı
 1. YILDIZ TEKNİK ÜNİVERSİTESİ YÜKSEK LİSANS (İSTANBUL)
 2. YILDIZ TEKNİK ÜNİVERSİTESİ (İSTANBUL)
 3. DOĞU AKDENİZ ÜNİVERSİTESİ (KIBRIS)
 4. MİH LİSESİ
- Bölümü,
 1. Endüstri mühendisliği
 2. Endüstri Mühendisliği
 3. Yıldız Teknik Üniversitesine Yatay Geçiş
 4. Sözel Bölümü
- YAPTIĞI PROJELERDE NÖRNEKLER
 - Yüksek lisans Tezi: Genetik Algoritma ve Hat Dengeleme
 - Lisans Bitirme Tezi: Yerel Yönetimlerde Kurumsallaşma
 - Proje1: Esnek yönetim
 - Proje2: Risk Yönetimi
 - Proje3: Toplam Kalite Yönetimi / 6 Sigma