

ATATÜRK ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME (SAYISAL YÖNTEMLER) ANABİLİM DALI

Kemal ÇAKAR

**GENETİK ALGORİTMALAR YARDIMIYLA ACİL SERVİS
İSTASYONU YERLEŞİMİNİN OPTİMİZASYONU**

DOKTORA TEZİ

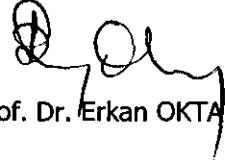
TEZ YÖNETİCİSİ
Prof. Dr. ERKAN OKTAY

ERZURUM - 2009

TEZ KABUL TUTANAĐI

SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĐÜNE

Bu alıřma İřletme Anabilim Dalının Sayısal Yöntemler Bilim Dalında jürimiz tarafından Doktora Tezi olarak Kabul edilmiştir.



Prof. Dr. Erkan OKTAY

Danışman / Jüri



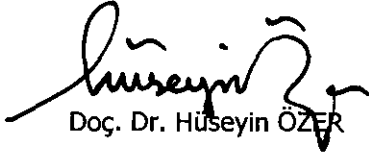
Prof. Dr. Mahmut KARTAL

Jüri



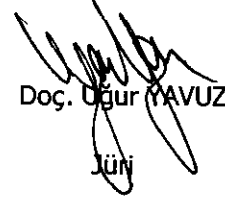
Do. Dr. Üstün ÖZEN

Jüri



Do. Dr. Hüseyin ÖZER

Jüri



Do. Uğur MAVUZ

Jüri

Yukarıdaki imzalar, adı geen öğretim üyelerine aittir. 23/09/2009

Prof. Dr. Mustafa YILDIRIM

Enstitü Müdürü

İÇİNDEKİLER

	Sayfa No
ÖZET	V
ABSTRACT	VI
TEŞEKKÜR	VII
ŞEKİLLER DİZİNİ	VIII
TABLolar DİZİNİ	IX
GİRİŞ	1

BİRİNCİ BÖLÜM

1. OPTİMİZASYON TEORİSİ ve GLOBAL OPTİMİZASYON

1.1 Optimizasyon Problemlerinin Sınıflandırılması.....	5
1.2 Global Optimizasyon.....	10
1.3. Durum Uzayı Araması (State Space Search).....	12

İKİNCİ BÖLÜM

2. EVRİMSEL ALGORİTMALAR

2.1. Evrimsel Algoritmalar ve Gelişimi	17
2.2. Evrimsel Algoritmaların Genel Özellikleri ve İşleyişi.....	
2.3. Evrimsel Algoritmaların Bileşenleri	23
2.4. Evrimsel Algoritmaların Avantajları	29
2.5. Evrimsel Algoritmaların Uygulama Alanları	30
2.5. Evrimsel Strateji	31
2.4. Evrimsel Programlama.....	35

ÜÇÜNCÜ BÖLÜM

3. GENETİK ALGORİTMALAR

3.1. Genetik Algoritmaların Gelişimi.....	39
3.2. Genetik Algoritmaların Çalışma Esasları.....	41
3.2. Genetik Algoritma Terminolojisi.....	45
3.2.1. Uygunluk.....	46

3.2.2. Popölasyon	47
3.2.3. Veri Yapıları.....	48
3.2.4. Arama Uzayı (Search Space).....	48
3.2.4. Arama Stratejileri.....	48
3.3. Kodlama.....	49
3.3.1. İkili Kodlama.....	49
3.3.2. Sekizli (octal) Kodlama.....	50
3.3.3. Hekzadesimal Kodlama.....	50
3.3.4. Permütasyon Kodlama (Gerçek Sayı Kodlama).....	50
3.3.5. Değer Kodlama.....	52
3.3.6. Ağaç Kodlama.....	52
3.3.7. Gri Kodlama.....	53
3.4. Genetik Algoritma Operatörleri.....	53
3.4.1. Yeniden Üretme/Seçim.....	54
3.4.1.1. Rulet Tekerleği Seçimi.....	55
3.4.1.2. Rastgele Seçim.....	56
3.4.1.3. Sıralama (Rank) Seçimi.....	56
3.4.1.4. Turnuva Seçimi.....	57
3.4.1.5. Boltzmann Seçimi.....	58
3.4.2. Çaprazlama.....	59
3.4.2.1. Tek noktalı çaprazlama.....	60
3.4.2.2. Çift noktalı çaprazlama.....	60
3.4.2.3. Çok noktalı çaprazlama.....	61
3.4.2.4. Uniform çaprazlama.....	62
3.4.2.5. Üç ebeveynli çaprazlama.....	62
3.4.2.6. Karıştırmalı (shuffle) çaprazlama.....	62
3.4.3. Mutasyon.....	63
3.4.3.1. Döndürme (flipping)	64
3.4.3.2. Değiş-tokuş.....	64
3.4.3.3. Gaussian mutasyon.....	65
3.4.4. Yenileme/Yer Değiştirme (Replacement)	65
3.4.4.1. Tesadüfi yenileme (Random replacement).....	66

3.4.4.2. Zayıf ebeveyn yenileme.....	66
3.4.4.3. Her iki ebeveyni yenileme.....	66
3.5. Yakınsama Kriterleri (Arama Sonlandırma)	66
3.5.1 En iyi birey.....	67
3.5.2. En kötü birey.....	67
3.5.3. Uygunluk Toplamı.....	67
3.5.4. Orta uygunluk.....	67
3.6. Genetik Algoritmalar da Parametre Seçimi.....	67
3.6.1. Popülasyon büyüklüğü.....	68
3.6.2. Çaprazlama olasılığı.....	68
3.6.3. Mutasyon olasılığı.....	69
3.6.4. Kuşak Aralığı.....	69
3.6.5. Seçim Stratejisi.....	69
3.6.6. Fonksiyon Ölçeklemesi.....	70
3.7. GA'nın Geleneksel Optimizasyon Yöntemleri ile Karşılaştırılması.....	70
3.8. Genetik Algoritmaların Uygulama Alanları.....	71
3.9. Genetik Algoritmaların Sınıflandırılması.....	73
3.9.1. Basit genetik algoritma	73
3.9.2. Paralel genetik algoritma	74
3.9.3. Hibrit genetik algoritma	75
3.9.4. Adaptif genetik algoritma	75
3.9.5. Hızlı dağınık (messy) genetik algoritma	76

DÖRDÜNCÜ BÖLÜM

4. ACİL SERVİS İSTASYONU YERLEŞİM MODELLERİ

4.1. Ambulans Yerleşim Modelleri.....	77
3.1.1. Statik Ambulans Yerleşim Modelleri	80
3.1.2. Deterministik Statik Ambulans Yerleşim Modelleri.....	82
3.1.3. Probabilistik Statik Ambulans Yerleşim Modelleri	85
3.1.4. Dinamik Ambulans Yerleşim Modelleri.....	90

BEŞİNCİ BÖLÜM

5. ACİL SERVİS YERLEŞİM PROBLEMİ VE UYGULAMASI

5.1. Genetik algoritma Metodu Kullanılarak Trabzon İl Merkezi Ambulans	
Yerleşim Planı Çıkarılması.....	94
5.2 Varsayımlar ve Oluşturulan Model	98
BULGULAR ve ÖNERİLER	102
KAYNAKLAR.....	104
EKLER	
EK-1. Mesafe Matrisi.....	109
EK-2.Genetik Algoritma MATLAB Programı	110
ÖZGEÇMİŞ.....	114

ÖZET
DOKTORA TEZİ
GENETİK ALGORİTMALAR YARDIMIYLA ACİL SERVİS
İSTASYONU YERLEŞİMİNİN OPTİMİZASYONU

Kemal ÇAKAR

Danışman : Prof. Dr. Erkan OKTAY

2009 – SAYFA : 114

Jüri : Prof. Dr. Mahmut KARTAL

Prof. Dr. Erkan OKTAY

Doç. Dr. Üstün ÖZEN

Doç. Dr. Uğur YAVUZ

Doç. Dr. Hüseyin ÖZER

Acil Sağlık Hizmetlerinde, çağrı geldiğinde operatörler için önemli olan vakanın şiddetini ve aciliyetini tespit etmek ve bundan sonra görevlendirilecek istasyon ve ambulans tipini belirlemektir. Acil durumlarda zaman hayati bir role sahip olduğu için talebin karşılanması ve hızlı müdahalenin sağlanması için uygun yerlerde , sayıda ve kapasitede istasyon yerleşimi yapılması önemlidir.

Çalışmada ilk olarak optimizasyon teorisi, global optimizasyon yöntemleri ve durum uzayı araması (state-space search) tekniği incelenmiştir. İkinci kısımda, genetik algoritmaların dahil olduğu evrimsel algoritmalar ailesinin gelişimi, işleyişi, terminolojisi, parametre seçimi ve geleneksel optimizasyon yöntemlerinden farkları ve avantajları genel olarak incelenmiştir. Üçüncü kısım, özel olarak genetik algoritmaların işleyişi, terminolojisi, parametre seçimi ve geleneksel optimizasyon yöntemleri ile bir karşılaştırmasını ele almaktadır. Dördüncü kısımda literatürde yer alan acil servis yerleşim modelleri gözden geçirilmiştir. Beşinci ve son kısımda ise minimum taşıma mesafesini verecek şekilde statik probabilistik bir istasyon yerleşim modeli geliştirilmiştir. Oluşturulan bu model Trabzon İl Ambulans Servisi verilerine göre Trabzon il merkezindeki acil yardım istasyonları yerleşimi problemine uygulanmış ve genetik algoritma metodu kullanılarak çözülmüştür. Elde edilen çözüm durum uzayı araması tekniği kullanılarak doğrulanmıştır.

Genetik algoritma metodu ile ulaşılan çözümde mevcut istasyon yerleşimi durumuna göre daha kısa taşıma mesafesi öngören bir yerleşim modeli önerilmektedir.

ABSTRACT
Ph. D. THESIS
OPTIMIZATION OF LOCATION OF EMERGENCY SERVICE
STATION USING GENETIC ALGORITHMS

Kemal ÇAKAR

2009 – PAGE : 114

Jury : Prof. Dr. Mahmut KARTAL

Prof. Dr. Erkan OKTAY

Assoc. Prof. Dr. Üstün ÖZEN

Assoc. Prof. Dr. Uğur YAVUZ

Assoc. Prof. Dr. Hüseyin ÖZER

In the Emergency Health Service Systems, the important issue for the operators is to determine the level of the importance and urgency of call and decide service station and ambulance type to send. Because the response time has crucial importance in any emergency case, it is important to locate service station into right places with right number of vehicle and enough capacity.

In this study, firstly, optimization theory, global optimization methods, and state-space search are investigated. In the second section, development stage, mechanism, terminology, and parameter selection of evolutionary algorithms family in which genetic algorithms involved are investigated. In third section, mechanism and terminology of genetic algorithms are investigated as well as a comparison between genetic algorithms and traditional optimization algorithms. In fourth section emergency service location models are studied. In the last section a static probabilistic model assuring minimum traveling distance is developed. This model applied to current location of Trabzon emergency service station problem with the data from Trabzon Emergency Service Unit and the problem is solved using genetic algorithm method. The results also tested using state-space search method.

The results of the genetic algorithms present some improvements providing less traveling distance and consequently quick response time.

TEŞEKKÜR

Doktora tez çalışması süresinde her konuda yardımcı olan ve tezin yürütülmesinde yol gösteren danışman hocam Prof. Dr. Erkan OKTAY'a ve uygulama konusunda önerilerini esirgemeyen Prof. Dr. Sibkat KAÇTIOĞLU'na tüm içtenliğimle teşekkür ederim.

Çalışmanın uygulama aşamasında gerek veri sağlayan ve gerekse sistemin işleyişi ile ilgili detaylı bilgi veren Trabzon İl Ambulans Servisi Başhekimi Dr. Faik ÖZŞAHİN ve ekibine, genetik algoritma çözümünde kullanılacak programın MATLAB uygulamaları için yardımlarını esirgemeyen KOSGEB-KTÜ Teknoloji Geliştirme Merkezi Müdürü Dr. Mehmet TURHAL'a ve tez çalışmaları süresince göstermiş oldukları anlayış için başta M. Mustafa KURT ve Ebru SARIALIOĞLU olmak üzere tüm çalışma arkadaşlarıma teşekkürlerimi sunuyorum.

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 1-1: Optimizasyon algoritmaları kategorileri.....	6
Şekil 1-2: Global Optimizasyon Algoritmalarının Sınıflandırılması.....	11
Şekil 2-1: Evrimsel algoritma işleyişi.....	20
Şekil 2-2: Evrimsel strateji algoritma işleyişi ($\mu+\lambda$).....	34
Şekil 3-1: Genetik algoritma döngüsü.....	42
Şekil 3-2: Genetik algoritma işleyişi.....	43
Şekil 3-3: Alel, Gen ve Kromozomlar.....	45
Şekil 3-4: İkili Kodlama.....	50
Şekil 3-5: Sekizli (octal) Kodlama.....	51
Şekil 3-6: Hekzadesimal Kodlama.....	51
Şekil 3-7: Permütasyon Kodlama.....	51
Şekil 3-8: Değer Kodlama.....	52
Şekil 3-9: Ağaç Kodlama.....	53
Şekil 3-10: Tek noktalı çaprazlama.....	60
Şekil 3-11: Çift noktalı çaprazlama.....	61
Şekil 3-12: Uniform çaprazlama.....	62
Şekil 3-13: Döndürme (flipping).....	64
Şekil 3-14: Değiş-tokuş.....	65
Şekil 5-1: Bir ambulans sistemindeki olaylar dizisi.....	94
Şekil 5-2: Trabzon Şehir Merkezi Haritası.....	97

ÇİZELGELER DİZİNİ

	Sayfa No
Çizelge 2.1. Evrimsel algoritma terminolojisi.....	22
Çizelge 5.1. Trabzon İl Merkezi Acil Servis İstasyonlarının durumu.....	96
Çizelge 5.2. İlk popülasyon matrisi.....	100
Çizelge 5.3. Trabzon İl Merkezi Acil Servis İstasyonlarının yerleşimi....	102

GİRİŞ

Acil servis sistemleri doğal afet, yangın, hastalık ve diğer aciliyeti olan olaylar için kurulan polis, itfaiye ve ambulans sistemleri ve diğer kurtarma ekiplerini kapsar. Yukarıda sıralanan durumların ortak yanı hemen hepsinin insan hayatı ile yakından ilişkili olmasıdır. İnsan yaşamının kurtarılmasında, can ve mal kaybının en aza indirilmesinde bu organizasyonların çalışma sistemlerinin iyi organize edilmiş olması ve değişen koşullara uyum sağlamasının önemi büyüktür. Tüm bu sistemlerin diğer bir ortak yanı ise kullandıkları performans başarı kriterleridir. Vakaya müdahale süresi ve hizmet kalitesi bu kriterlerin en önemlileri konumundadır.

Gelişen teknoloji, artan araç kullanımı, artan nüfus ve nüfus yoğunluğu gibi nedenlerle insan hayatı için tehlike arz eden acil durumların sayısı da giderek artmaktadır. Değişen koşullar da bu hizmetlerin daha etkin verilmesini giderek daha karmaşık ve zor bir hale getirmektedir.

Uzun yıllardan bu yana ülkemizde acil sağlık sistemleri tek numara altında hizmet vermektedir. Sistemin işleyişi için illerde komuta kontrol merkezleri oluşturulmuş ve yönlendirme için gerekli personel, araç ve donanım sağlanmıştır. Acil yardım servis istasyonlarında acil yardım ve nakil ambulansları bulundurulmaktadır.

Ambulans hizmetlerinde en önemli performans kriteri olan müdahale süresini etkileyen en önemli problemler aşağıda sıralanmıştır:

- Vaka durum tespitinde yapılan yanlışlıklar
- Çağrı merkezine gelen yanlış çağrılar
- Yanlış adres bildirimleri
- Ambulansların meşgul olma durumları,
- Kurumlar arası bütünleşme bozukluğu,
- Yoğun trafik,
- Personel yetersizliği.

Ambulansların standartlara uygun sürede vaka mahalline ulaşması ve müdahalede bulunması için araç sayısının artırılması veya mevcut araç yerleşim düzeninin geliştirilmesi ile sağlanabilir. Ancak araç sayısının artırılması çok yüksek maliyetli olması bu seçeneği kapsam dışında bırakmaktadır. Bu nedenle acil servis sistemleri planlanırken kısıtlı sayıda ambulansın, talep bölgeleri içinde, optimum veya optimuma yakın yerleşimine ihtiyaç duyulur. Bu yerleşimin daha etkin kurulabilmesi için literatürde çok sayıda çalışma yapıldığı ve çok sayıda model tasarlandığı görülmektedir.

Daha etkin bir yerleşim planı sağlamak üzere geliştirilen bu modeller statik, dinamik ve stokastik modeller olmak üzere üç grupta sınıflandırılabilir. Statik ambulans yerleşim modellerinde genellikle belirli sayıda ambulans ile hizmet verilen nüfusu maksimize etmek amaçlanır. Stokastik modellerde bir ambulansın herhangi bir vakaya gönderildiğinde diğer çağrılara cevap veremeyecek şekilde meşgul olması durumu da göz önüne alınır. Dinamik ambulans yerleşim modellerinde ise ambulansların gün içinde sürekli yer değiştirmesi ile bütün bölgelere hizmet vermesi amaçlanır ve bu tip modellerde, ambulansların sürekli yer değiştirmesi problemin çözümünü zorlaştırır.

Bu çalışmada kurulan probabilistik bir model ile genetik algoritma metodunu kullanarak Trabzon ili şehir merkezindeki acil yardım istasyonları için bir yerleşim modeli bulunması amaçlanmıştır. Bu kapsamda çalışmanın ilk bölümünde genel olarak optimizasyon teorisi ve global optimizasyon konusu incelenmiştir. İkinci kısımda evrimsel algoritma konusunda genel bilgilere yer verilmiştir. Üçüncü bölümde genetik algoritmaların çalışma esasları, terminolojisi, genetik algoritma operatörleri, yakınsama kriterleri (arama sonlandırma), genetik algoritmalarda parametre seçimi, genetik algoritmaların geleneksel optimizasyon yöntemlerinden farkları, avantajları, sınırlılıkları ve dezavantajları, genetik algoritmaların uygulama alanları konuları ve genetik algoritma çeşitleri ayrıntılı biçimde irdelenmiştir.

Dördüncü bölümde acil servis yerleşim modelleri üzerinde durulmuştur. Bu bölümde tesis yerleşimi ile ilgili geliştirilen modeller ile ilgili bilgiler verilmiştir.

Son kısımda ise Trabzon ili şehir merkezindeki acil yardım istasyonlarının yerleşimi ile ilgili probabilistik bir model geliştirilerek ve genetik algoritma metodu ile çözümü gerçekleştirilmiş ve yüzey arama tekniği ile bulunan çözüm test edilmiştir.

BİRİNCİ BÖLÜM

OPTİMİZASYON TEORİSİ ve GLOBAL OPTİMİZASYON

Son kırk-elli yıl içerisinde bilgisayarların etkinliklerinin çok hızlı biçimde sürekli gelişmesi ve nümerik hesaplamalar için geliştirilen yazılımların da buna paralel bir seyir izlemesi ile nümerik optimizasyon teorisi, metotları ve uygulamalarında büyük bir ilerleme kaydedilmiştir. Bu konudaki bilgi ve imkanların artması da tersine, optimizasyon metotlarının uygulamalarının mühendislik, işletme ve fen bilimleri gibi birçok farklı disiplinlere yayılmasını sağlamış ve kısa süre önce zorlu olarak kabul edilen problemlerin kolayca çözümüne imkan tanımıştır.

Optimizasyon prosesi, ‘iyi’ ve ‘kötü’deki değişimin ölçülebildiği durumda ‘en iyi’ye ulaşma sürecidir. Optimizasyon, girdileri minimum veya maksimum çıktı elde etmek üzere tanzim etme veya değiştirme prosesidir. Diğer bir tanımla, optimizasyon, fiziksel bir gerçekliği modelleyen ve matematiksel olarak tanımlanan belirli problemlerin ‘en iyi’ (best) çözümlerini belirleyen bilim olarak tanımlanmaktadır (Fletcher, 1987, s;3).

Gerçek hayatta gelirler veya kâr gibi bazı kalemlerin ‘en fazla’ yada ‘maksimum’ olması amaçlanırken; maliyet yada harcama gibi diğer bazı değerlerin de ‘en az’ yada ‘minimum’ olması istenir. Bu açıdan bakıldığında “optimum” kavramı bu istenen durumların her ikisini de kapsamaktadır. Optimizasyon teorisi ise optimal değer ile ilgili bir sayısal çalışma alanı ve bu değer bulunması ile ilgili metotları kapsamaktadır. Bu teorinin uygulaması da ‘en uygun’ (optima) değer hesaplanmasında kullanılabilecek teknikler, metotlar, prosedürler ve algoritmalar bütünüdür (Antoniou ve Lu, 2007, s;2).

Optimizasyon problemi mühendislik, fizik, matematik, ekonomi, sosyal bilimler gibi çok farklı disiplinlerde ortaya çıkabilmektedir. Bunun yanında elektrik, makine, inşaat ve kimya gibi farklı mühendislik alanları da çok sayıda optimizasyon problemi içermektedir. Farklı mühendislik disiplinleri içerisinde, araçların dizaynı ve modellenmesi, yapıların dizaynı, proses kontrolü, üretim planlama, kalite kontrol, stok kontrol gibi çok farklı konularda optimizasyon uygulamaları yapılmaktadır.

Gerçek hayattaki birçok problem birden çok sayıda çözüme sahiptir. Bu durumda bazı performans kriterleri kullanılarak problem için en iyi çözüm bulunarak

optimizasyon gerçekleştirilebilir. Problemin tek bir parametre değeri setinin kabul edildiği tek bir çözümünün olduğu durumda optimizasyon uygulanamamaktadır.

Optimizasyon konusundaki yaklaşımlar dört farklı başlık altında incelenebilir: Analitik metotlar, grafik metotlar, deneysel metotlar ve nümerik metotlar (Antoniou ve Lu, 2007, s;2).

Analitik metotlar klasik diferansiyel hesaplamalar üzerine kurulmuştur. Çözülecek problemin öncelikle matematiksel terimlerle ifade edilmesi gerekmektedir. Bilgisayar kullanımı gerektirmemekle birlikte yüksek derecede nonlineer olan veya üçten fazla bağımsız değişken bulunan problemlere bu yöntem uygulanamamaktadır.

Grafik metot maksimize veya minimize edilecek fonksiyonun grafiğinin çizilmesi için kullanılır. Çoğu pratik uygulamalarda optimize edilecek fonksiyon dört ve daha fazla değişkene bağımlı olduğundan pratikte grafik metodun kullanımı da sınırlı kalmaktadır.

Bir sistemin optimum performansı bazı durumlarda deneyle elde edilmektedir. Bu metotta değişkenler birer birer ayarlanır ve performans kriterleri her bir durum için yeniden ölçülmektedir. Optimum veya optimuma yakın sonuç vermekle birlikte iki veya daha fazla değişkenin birbiri ile etkileşim halinde olduğu sistemlerde bu metot güvenilir olmayan sonuçlar verebilmektedir.

Optimizasyon konusundaki en önemli yaklaşım nümerik metotları temel almaktadır. Bu yaklaşımda optimizasyon problemine kademe kademe gelişen çözümler serisi sunmak için aşamalı (iterative) nümerik prosedürler kullanılmaktadır. Bu proses bazı yaklaşma kriterleri sağlandığında sonlandırılmaktadır. Nümerik metotlar analitik olarak çözülemeyecek çok karmaşık optimizasyon problemlerinin çözümünde kullanılabilir. Bunun ötesinde bilgisayarlarda kolayca programlanabilmektedirler. Bu özellikleri ile diğer optimizasyon yaklaşımlarının da yerine geçebilmektedir.

Nümerik optimizasyonun teori ve pratiğini kapsayan disiplin matematiksel programlama olarak bilinmektedir. Doğrusal programlama, tamsayılı programlama, kuadratik (Quadratic) programlama, Doğrusal olmayan programlama ve dinamik programlama bu disiplinin son kırk yıl içerisinde gelişen dallarından bazıları olarak sıralanabilir (Antoniou ve Lu, 2007, s;2).

Optimizasyon karar verme ve fiziksel bir sistemin analizi konularında oldukça önemli bir araçtır. Bu aracın kullanılabilmesi için ilk olarak sistemin performansını sayısal olarak ölçebilen bir ‘amaç’ın belirlenmesi gerekmektedir. Bu amaç tek bir rakam ile ifade dileyen kâr, zaman, miktar veya miktar kombinasyonu vb. olabilmektedir. Belirlenen ‘amaç’ sistemin ‘değişken’ yada bilinmeyen olarak tanımlanan belirli karakteristiklerine bağımlıdır. Bu değişkenler sıklıkla sınırlandırılır veya kısıtlanmaktadır.

Çözülmesi istenen problem için amaç, değişkenler ve kısıtların tanımlanması süreci ‘modelleme’ olarak adlandırılmaktadır. Optimizasyon prosesi içinde en önemli aşama ilk adım olan uygun modelin oluşturulmasıdır. Model formüle edildikten sonra bir optimizasyon algoritması çözüm için kullanılabilir. Ünlversal bir optimizasyon algoritması mevcut değildir. Bunun yerine belirli tiplerdeki optimizasyon problemlerine uyarlanmış algoritmalar söz konusudur. Doğru algoritma seçimi problemin doğru ve hızlı çözümünün ilk koşuludur. Önemli bir kısıtın probleme dahil edilmemesi veya uygun bir formülasyon sağlanamaması optimal noktanın ihlal edilmesi sonucunu doğuracaktır.

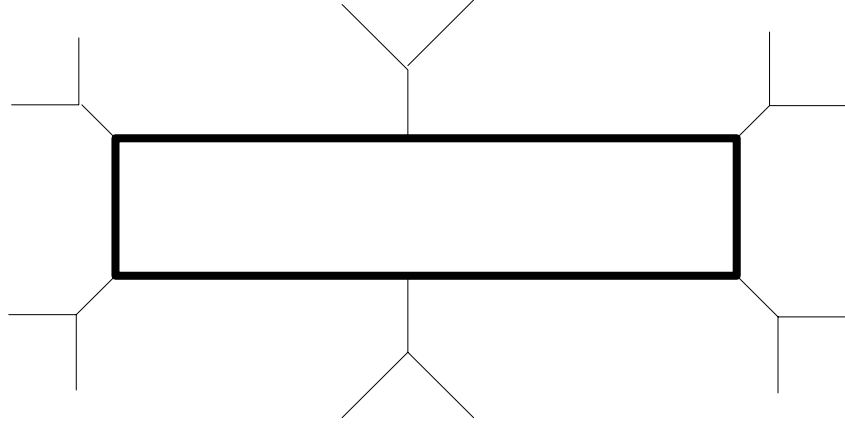
Optimizasyon algoritmasının modele uygulanmasından sonra çözümün elde edilip edilmediğinin de belirlenmesi gerekmektedir. Birçok durumda eldeki değişken setinin problemin çözümü olup olmadığını kontrol etmek için ‘optimallik şartları’ olarak adlandırılan matematik ifadeler kullanılmaktadır. Bu şartların sağlanmadığı durumda da eldeki çözümün nasıl geliştirilebileceği yönünde de bilgi verebilmektedir. Modelde ve verilerdeki değişime çözümün ne kadar hassas olduğu ‘duyarlılık analizi’ teknikleri uygulanarak belirlenebilmektedir (Nocedal ve Wright, 2006, s;2, Arora, 2007, s;2).

1.1 Optimizasyon Problemlerinin Sınıflandırılması

Optimizasyon problemleri ve bu problemlerin çözümüne yönelik geliştirilen metotlar farklı kriterlere göre çok farklı biçimlerde sınıflandırılmaktadır. Problemin yapısı, değişkenlerin özellikleri, modelde kullanılan kısıt ve fonksiyonların karakteristikleri bu sınıflandırmaya temel teşkil etmektedir.

Optimizasyon algoritmaları altı kategoriye ayrılabilir (Şekil 1.1). Bu kategorilerden herhangi birinin veya onların alt dallarının karşılıklı olarak yalnızca ikili

olması gerekmemektedir. Bir dinamik optimizasyon problemi ya sınırlandırılmış ya da sınırlandırılmamış olabilir. Bununla birlikte sınırlandırılmış bir problem seçilmiş ise problemdeki bazı değişkenler sürekli iken bazıları ise kesikli olabilmektedir.



Şekil 1.1 Optimizasyon algoritmaları kategorileri (Haupt ve Haupt, 2004, s;3)

Deneme-yanılma çıktısı ortaya koyan proses hakkında çok fazla bir şey bilinmediği durumda değişkenleri ayarlamanın söz konusu olduğu durumlar için kullanılmaktadır. Birçok büyük buluş bu yöntemle ortaya çıkarılmıştır. Bu yöntemi tercih eden deneycilerden farklı olarak teorisyenlerce, matematiksel olarak formüle edilen amaç fonksiyonunun çok sayıda matematiksel manipülasyonunun optimal sonuca vardığı kabul edilmektedir (Haupt ve Haupt, 2004, s;4).

Tek bir değişkenin söz konusu olduğu durumlarda optimizasyon tek boyutlu olur. Değişken sayısının arttığı durumlarda optimizasyon çok boyutlu hale gelmekte ve doğal olarak çözüm de giderek zorlaşmaktadır.

Fonksiyonun çıktısının zamana bağımlı olduğu durumda dinamik optimizasyon söz konusudur. Ulaştırma problemi çözümünde en iyi rota tespiti için mesafe açısından bakıldığında problem statiktir. En kısa yol seçilir. Uygulamada ise en kısa yol en hızlı rota olmayabilir. Bu durumda en hızlı yolu bulma, çözümü gün içerisindeki zamana, havaya, kazalara vb. koşullara bağlı dinamik bir problem olmaktadır (Haupt ve Haupt, 2004, s;4).

Den
&
yanı

1

fonksiyon

Optimizasyon, deęişkenlerin kesikli veya sürekli olmasına göre de sınıflandırılmaktadır. Kesikli deęişkenler sonlu sayıda muhtemel deęer alırlarken sürekli deęişkenler ise sonsuz sayıda deęer alabilmektedir. Belirli bir listedeki bir dizi görevin hangi sırada yapılacağı kesikli optimizasyon konusu olur. Kesikli deęişken optimizasyonu birleşimsel (combinatorial) optimizasyon olarak da bilinir, çünkü bu durumda optimum çözüm tüm muhtemel deęişkenlerin yer aldığı sonlu bir havuzdan seçilen deęişkenlerin belirli bir bileşiminden oluşmaktadır (Haupt ve Haupt, 2004, s;4, Nocedal ve Wright, 2006, s;5).

Optimizasyon problemleri kısıtların varlığına baęlı olarak da sınıflandırılabilir. Kısıtların olduğu durumda sınırlandırılmış, olmadığı durumda ise sınırlandırılmamış optimizasyon yapılır (Rao, 1996, s;15). Genellikle deęişkenlerin limitleri vardır yada kısıtları bulunur. Sınırlandırılmış optimizasyon deęişken eşitlik yada eşitsizliklerini amaç fonksiyonu ile bütünleştirir. Sınırlandırılmamış optimizasyonda ise deęişkenlerin herhangi bir deęer almasına izin verilmektedir. Çoęu nümerik optimizasyon işlemi sınırlandırılmamış deęişkenler ile daha iyi sonuçlar vermektedir. Sınırlandırılmış yada kısıtlı optimizasyonun deęişkenleri doğrusal bir denklem ve doğrusal kısıtlar ile formüle ettiği durum doğrusal program olarak adlandırılır. Benzer şekilde maliyet denklemleri veya kısıtların nonlinear olduğu durumda da problem doğrusal olmayan (nonlinear) programlama problemi olmaktadır (Haupt ve Haupt, 2004, s;4; Nocedal ve Wright, 2006, s;6).

Bazı algoritmalar deęişkenlerin belirli bir başlangıç seti deęerlerinden hareketle minimum deęeri bulmayı amaçlar. Yerel minimuma takılma riski yüksek olmasına rağmen bu algoritmalar hızlı çalışmaktadır. Geleneksel optimizasyon metotları olarak da adlandırılan bu metotlar genelde cebirsel temellidir. Bir deęişken setinden dięerine hareket bazı belirleyici sıralı adımlarla yapılır. Dięer taraftan olasılıklı modeller ise deęişken setini bulmak için birtakım rassal hesaplamalar yaparlar (Haupt ve Haupt, 2004, s;5). Genetik algoritmalar da rassal olarak yeni deęişken seti arayan modeller içerisinde sınıflandırılmaktadır.

Deęişkenlerin doğasına göre Rao (1996, s;16) tarafından yapılan ayırmda ilk grup, belirli kısıtlar altında ve aynı parametreler kullanılarak oluşturulan önceden tanımlanmış fonksiyonu minimize eden parametre setinin araştırıldığı parametre veya statik optimizasyon problemlerinden oluşmaktadır. İkinci grupta ise dięer bazı

parametrelerin sürekli fonksiyonu olan parametre setinin amaç fonksiyonunu belli kısıtlar altında minimize etmesi söz konusudur. Her bir değişkenin, bir yada daha fazla parametrenin bir fonksiyonu olduğu bu probleme yörünge veya dinamik optimizasyon problemi adı verilmiştir.

Optimizasyon problemlerinin diğer bir sınıflandırması da amaç fonksiyonu ve kısıtlardaki ifadelerin yapısına göre yapılmaktadır. Bu sınıflandırmaya göre optimizasyon problemleri doğrusal, doğrusal olmayan, geometrik ve kuadratik olarak ayrılmaktadır. Hesaplama açısından bakıldığında, belirli bir sınıf problem için etkin çözümler ortaya koyan çok sayıda özel metot geliştirilmiş olduğu için bu sınıflandırma oldukça yararlı olmaktadır (Rao, 1996, s;21).

Karşılaşılan problem sınıfının belirlenmesi birçok durumda kullanılacak çözüm prosedürlerinin biçimini de büyük ölçüde belirlemektedir. Amaç veya kısıt fonksiyonlarından herhangi birinin nonlineer olması problemin doğrusal olmayan bir programlama (NLP) problemi olması sonucunu doğurur. Bu en genel programlama biçimidir. Tüm diğer sınıfların NLP'nin özel bir durumu olduğu düşünülebilir.

Yukarıda belirtilen ayrımlardan farklı olarak da birtakım sınıflandırmalar kullanılmaktadır (Rao,1996, s;15, Sarker ve Newton, 2006, s;5-12, Arora, 2007, s;3-4, Branke ve diğer., 2008, s;2):

Optimizasyon problemi birden çok amaç fonksiyonu içerebilir. Bu durumda problem, çok amaçlı (multi-objective) optimizasyon problemi adını alır. Bu durumda genellikle amaçlar birbiri ile çelişmektedir. Problemin yapısına bağlı olarak modeldeki değişkenler reel, tamsayı (gerçek tamsayı veya ikili (binary)) veya her ikisinin karşımı olabilmektedir.

Birçok karar ve planlama problemi genellikle eş zamanlı olarak çözülmesi gereken ve birbiri ile çelişen çok sayıda amaç içermektedir. Bu açıdan optimizasyon problemi birden çok amaç fonksiyonu içerebilir. Bu durumda problem, çok amaçlı (multi-objective) optimizasyon problemi adını alır. Bu durumda genellikle amaçlar birbiri ile çelişmektedir. Bu tür problemleri karakteristiği yegane (unique) bir çözümünün olmaması ve bunu yerine matematiksel olarak eşit derecede iyi (good) çözümler setinin tanımlanabilmesidir. Problemin yapısına bağlı olarak modeldeki değişkenler reel, tamsayı (gerçek tamsayı veya ikili (binary)) veya her ikisinin karşımı

olabilmektedir. Amaç fonksiyonunun tipine göre de minimizasyon veya maksimizasyon olarak ayırım yapılmaktadır.

Problem ve kısıtlar dikkate alındığında kısıtlı (constrained) ve kısıtsız (unconstrained) ayrımı yapılmaktadır. Kısıtsız (unconstrained) problemlerin çözümü önem arz etmektedir çünkü birçok optimizasyon algoritması sınırlı problemleri sınırsız biçime veya sınırsız problem dizisine dönüştürerek çözmektedir.

Fonksiyonlara göre sınıflandırma ise çözüm metodu açısından oldukça önem arz eder biçimde daha çok fonksiyonun matematiksel özelliklerine göre yapılmaktadır. Amaç veya kısıt fonksiyonları lineer, nonlinear veya bunların bileşimi olabilir. Nonlinear modellerin çözümü lineer modellere göre daha fazla karmaşıklık içermektedir. Bununla birlikte kısıtsız (unconstrained) nonlinear problemler uygulamalarda daha fazla yer almaktadır.

Konvekslik (convexity) tüm optimizasyon teknikleri yada algoritmalarında önemli bir özellik olarak ele alınmaktadır. Birçok algoritma fonksiyonun konveks olduğu varsayımını taşımaktadır. Bu açıdan konveks ve konveks olmayan (nonconvex) ayrımı yapılmaktadır.

Fonksiyonun türevlenebilir olup olmaması da başka bir ayırım konusudur. Türev temelli optimizasyon teknikleri için fonksiyonun türevlenebilir olması gerekmektedir. Bu durum da fonksiyonun sürekliliği ile yakından ilgilidir.

Yukarıdaki ayrımların yanında fonksiyonlar tek yada daha fazla ekstremum noktaya sahip olmasına göre de sınıflandırılmaktadır. Çoklu ekstremum nokta olması durumunda global optimum nokta devreye girmektedir. Amaç fonksiyonu tek optimum nokta içermesi durumunda unimodal, birden fazla veya bilinmeyen sayıda optimum nokta içerdiğinde ise multimodal olarak tanımlanmaktadır.

Uygulamalı optimizasyon genel olarak global optimumun bulunmasını gerektirmektedir. Yerel optimizasyon ise daha çok teorik ve sezgisel kanıtların optimizasyon probleminin tek optimum nokta içerdiğini göstermesi durumunda uygulanabilmektedir. Birden çok yerel minimum olması durumunda yerel (local) optimizasyon algoritmaları en iyiyi bulmak için genellikle çok sayıda başlangıç noktalarından hareket eder. Bununla birlikte uygulamalarda sıklıkla karşılaşılan çoklu optimizasyon içeren problemlerin çözümünde bu metot etkin sonuçlar verememektedir (Zhigljavsky ve Zilinkas, 2008, s;9).

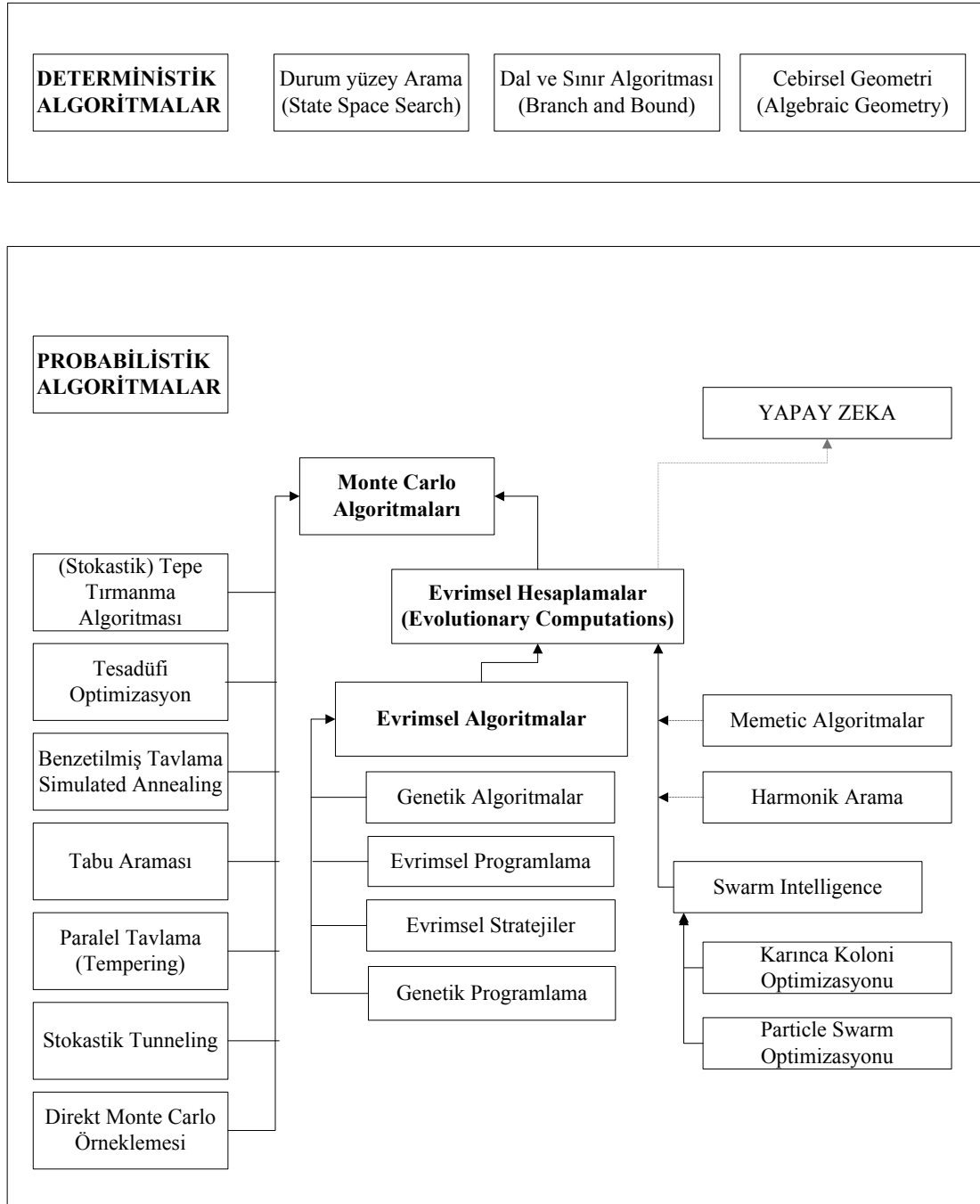
1.2 Global Optimizasyon

Global optimizasyonun amacı çok sayıda yerel optimum arasında modelin global olarak en iyi çözümünün bulunmasıdır. Dizayn, veri analizi, finansal planlama, proses kontrol, bilimsel modelleme gibi çok farklı konularda sıklıkla karşılaşılan nonlinear modellerin çözümü birçok durumda global optimizasyon yaklaşımını gerekli kılmaktadır.

Sistem dizaynında istenen özelliklerin yalnızca belirli bir değer kombinasyonu söz konusudur. Bunun anlamı mümkün çözümün bu özellikler tarafından belirlenen uygun çözüm bölgesi olan bir A alt bölgesi ile sınırlı olmasıdır. Amaç fonksiyonu bu alt bölge içerisinde çok sayıda yerel minimum veya maksimuma sahip olabilmektedir. Global optimizasyonun amacı bu alt bölgede amaç fonksiyonunun aldığı en küçük ya da en büyük değer bulunmasıdır. Bu değer de global minimum veya global maksimum olarak adlandırılmaktadır. Bu açıdan global optimizasyon yalnızca lokal minimumu değil bunun yanı sıra tanımlanan bölge içerisindeki en küçük lokal minimumu bulmaya çalışmaktadır (Törn ve Zilinkas, 1989, s;2).

Global optimizasyonda amaç fonksiyonunun basit ve genel kabul görmüş bir modeli mevcut değildir ve çok değişik biçimlerde çok optimumlu fonksiyonlar söz konusudur ve global optimumu bulmak için kullanılan arama stratejilerinin rasyonelliği güçlüklerle açıklanabilmektedir. Bundan dolayı çok optimumlu fonksiyonlar için genel kabul görmüş bir modelin yanı sıra global arama için universal bir strateji bulmak gerçekçi görünmemektedir. Global optimizasyona yönelik çok sayıda farklı yaklaşımların bulunması da bu durum ile açıklanmaktadır (Zhigljavsky ve Zilinkas, 2008, s;9-10).

Global optimizasyon algoritmaları işlem metoduna göre stokastik ve deterministik olarak ikiye ayrılmaktadır. Deterministik metotlar herhangi bir olasılıklı değişken kavramını içermemektedir ve global çözümün bulunmasını ve bunun doğrulanmasını garanti etmektedirler. Bunu sağlamak için fonksiyona ilave varsayımlar eklenme zorunluluğu söz konusudur (Nemhauser ve diğer., 1998, s;633, Gao ve Sherali, 2009, s;382). Bu ayrıma göre global optimizasyon algoritmalarının sınıflandırılması aşağıda Şekil 1.2’de gösterilmiştir.



Şekil 1.2 Global Optimizasyon Algoritmalarının Sınıflandırılması (Weise, 2008, s;4)

Klasik optimizasyon teorisi direkt olarak global optimizasyon problemlerine uygulanamamasına rağmen, konvekslik analizi gibi geleneksel birçok metot global optimizasyon metotlarının oluşturulmasında yaygın biçimde kullanılmıştır. Bununla birlikte stokastik metotların sadeliği, gürültü tarafından bozulan amaç fonksiyonlarını

değerlendirmeye olan uygunluğu ve boyutun büyütülmesi konusunda daha sağlıklı olması stokastik metotların avantajlarıdır (Zhigljavsky ve Zilinkas, 2008, s;2).

Muhtemel çözümün karakteristikleri ve problemin çözümüne olan katkısı arasında açık bir ilişki olması durumunda daha çok deterministik modeller kullanılmaktadır. Aday bir çözüm ile onun uygunluğu (fitness) arasında açık bir ilişki olmadığında, çok karmaşık olduğunda yada arama yüzeyinin boyutunun çok yüksek olması durumlarında problemin deterministik olarak çözümü zorlaşmaktadır. Bu gibi durumlarda da stokastik çözümler öne çıkmaktadır.

Stokastik modeller nicelendiğinde ise Monte Carlo temelli olasılıklı algoritmalar ailesi öne çıkmaktadır. Oldukça kısa sürede sonuç vermelerine rağmen bu algoritmalar global optimuma ulaşmayı garanti etmemektedirler. Olasılıklı yada Stokastik olarak tanımlanan modelleri yeni tip model içeren problemlere uyarlamak bazı deterministik modellere göre daha zor olmaktadır. Popülasyon temelli arama metotlarının birçoğunda yeni bir geçici çözüm oluşturma aşaması önemli ölçüde model yapısına bağımlı olabilmektedir (Weise, 2008, s;4, Gao ve Sherali, 2009, s;382).

Global algoritmalarda sezgisellik (heuristics), mümkün çözüm setinden bir sonraki adım için hangisinin seçileceğine karar vermede kullanılmaktadır. Deterministik modeller aday çözümlerin işlem sırasını belirlemede sezgisellikten yararlanırken; olasılıklı metotlar ise problem yüzeyinin taranmasında daha sonraki hesaplamalar için de yalnızca bu olasılıklı yaklaşımı kullanabilmektedir.

1.3. Durum Uzayı Araması (State Space Search)

Arama, bilgisayar bilimleri ve yöneylem araştırmasında optimizasyon problemlerinin bilgisayar temelli çözümünde kullanılan temel bir tekniktir. Aramanın amacı, sınırlı yada sınırsız fakat sayılabilir bir çözüm seti arasında optimal veya optimale yakın sonuç bulmaktır.

Durum Uzayı Araması (State-space search) bir optimizasyon algoritması olarak değerlendirilmemektedir. Bu yöntem açık veya kapalı biçimde eldeki problemin çözümlerini sıralamaktadır (Zhang, 1999, s;1, Weise, 2008, s;265).

Bir durum uzayı arama algoritmasının durum uzayındaki tüm noktaları taraması gerekmektedir. Bu açıdan ortalama bir durum için karmaşıklık algoritma için önemli bir performans göstergesi olarak dikkate alınmaktadır (Zhang, 1999, s;34).

Durum uzayı arama algoritmalarında arama uzayı G ile problem uzayı X genellikle aynıdır. durum uzayı aramalarının çoğu arama uzayının sayılabilir (enumerable) olduğu durumda uygulanabilmektedir. Aynı probleme yönelik her çalıştırıldıkları durumda aynı sonucu verdiklerinden dolayı bu algoritmalar deterministik olarak tanımlanmaktadır.

Durum uzayı arama algoritmaları için tanımlanması gereken önemli bir operatör de arama uzayının sıralanmasına (enumerating) yardımcı olan genişleme (expand) fonksiyonudur. Bu operatör G uzayından bir ' g ' elementi ele alır ve onunla ulaşılabilir bir G seti hesaplar. Genişleme (expand) fonksiyonu durum uzayı arama algoritmalarının keşif işlemidir. Tamamen deterministik olup; tek bir birey yerine bir sete dönüşmektedir. Fonksiyonun aynı ' g ' elementi değerlerine uygulanması böylece her zaman aynı G setini elde etmektedir. Bu açıdan genişleme fonksiyonunun mutasyon gibi birli arama operatörleri ile elde edilebilecek tüm mümkün çözüm setlerine ulaşmaktadır.

Genişlemenin gerçekleştirilmesinin arama algoritmasının performansı üzerinde önemli etkisi mevcuttur. Etkin bir uygulama önceki aşamalarda ziyaret edilen durumları (state) tekrar kapsamaz. Aynı elemente dönüldüğünde aynı çözüm adayları ve onların çocukları bir çok defa daha değerlendirileceğinden gereksiz ve zaman harcayıcı bir faaliyet olmaktadır. İki farklı ' g ' elementinin birbirine dönüştüğü durum da diğer bir problemli duruma işaret etmektedir. Bu durumda da arama sonsuz bir döngüye girmektedir. Bu açıdan bakıldığında bir genotipin iki defa ziyaret edilmesinden kaçınılması gerekmektedir. Arama işlemlerinin başlangıçta bunu önleyici yönde tasarlanması ile bu durumun önüne geçmek mümkün olmaktadır. Bunun olmadığı durumda tabu arama tekniğindeki tabu listelerinin kullanılması gerekmektedir (Weise-2008, s.266). Durum uzayı aramasında, arama yöntemi, sezgisellik durumu, sınırların durumu, arama uzayının yapısı, kullanılabilir hafıza durumu ve çözümün optimalliği gibi çeşitli faktörlere göre belirlenmektedir (Floudas ve Pardalos, 2009, s;1906).

Genotip $p.g \in G$ ve fenotip $p.x \in X$ 'i kapsayan bir p birey notasyonu tanımlanır ise genel bir algoritma gösterimi elde etmek için arama uzayının G element setini P birey setine götüren bir genişleme operatörüne gereksinim duyulmaktadır. Bu arama operatörü aşağıdaki algorithmada gösterilmiştir (Weise,2008, s;266)

Genişleme Algoritması

Girdi: g: arama yüzeyinde genişleyecek olan element

Veri: i: hesaplanabilir değişken

Veri: p: bir bireysel kayıt

Çıktı: P: Genişleme sonucunda ulaşılan bireysel kayıt listesi

başla

$G \leftarrow \text{genişlet}(g)$ (problemin çözüm uzayında muhtemel bir çözüm belirle)

$P \leftarrow ()$

for $i \leftarrow 0$ tüm(G)-1'e kadar **do**

$p.g \leftarrow G[i]$

// tamamen (implicitly): $p.x \leftarrow \text{gmp}(p.g)$

$P \leftarrow \text{listeye ekle}(P, p)$

return P

bitir

Tüm durum uzayı arama stratejileri için verilen probleme uygun olup olmadıklarını tespit için dört farklı kriter tanımlanmaktadır:

1. Eksiksizlik: Arama algoritmasının çözüm bulmayı garanti etmesi gerekmektedir.
2. Harcanan süre: Seçilen strateji ile çözüme kısa sürede ulaşılması problem için doğru seçimin yapıldığını göstermektedir.
3. Hafıza kullanımı: Ara adımlar ile ilgili bilgilerin saklanması için algoritmanın düşün bir hafızaya gereksinim duyması etkinliğini diğer bir göstergesi olarak tanımlanmaktadır.
4. Optimallik: Belirlenen problem için birden çok doğru çözümün mevcut olması durumunda algoritmanın optimal çözüme ulaşması diğer bir uygunluk göstergesi olarak tanımlanmaktadır.

İKİNCİ BÖLÜM

EVİRİMSEL ALGORİTMALAR

Evrimsel Algoritmalar, biyoloji, yapay zeka, nümerik optimizasyon ve hemen bütün mühendislik disiplinlerindeki karar destek yapıları ile ilişkili disiplinler arası bir çalışma alanıdır. Organik evrim modeli üzerine geliştirilen bu algoritmalar, bireylerden oluşan popülasyonlardaki kolektif öğrenme prosesini modellemektedirler. Popülasyondaki bireyler yalnızca potansiyel çözüm alanındaki bir noktayı temsil etmekle kalmaz, bunun yanında doğanın “kanunu” ile ilgili mevcuttaki bilginin zamanla nesilden nesile taşınmasını da sağlamaktadır (Bäck,1996, s;7).

Evrimsel algoritmalar Charles Darwin tarafından ortaya atılan doğal, biyolojik evrim modelinden yararlanmıştır. Darwin’in modeli, türlerin adaptif değişimini, çevresel koşullara en iyi uyum sağlayan bireylerin hayatta kalıp evrimi sürdüreceği teorisini içeren “doğal seleksiyon” (seçim) prensibi ile açıklamaktadır. Seçimin yanında Darwin’in öne sürdüğü diğer bir önemli faktör fenotipler arasında gerçekleşen küçük, tesadüfi ve müdahale edilemez varyasyonlardır (değişimlerdir). Mutasyon olarak nitelendirilebilecek bu değişimler, mevcut şartlar ışığında değerlerini ispat ederlerse sürerler, aksi durumda yok olurlar, gerçekleşmezler (Bäck, 1996, s;8). Evrimsel hesaplamalar, doğal seleksiyon ve uyum sağlayanların hayatta kalması gibi evrimsel proseslerin bilgisayar modellerini kullanan bilgisayar tabanlı problem çözme sistemleridir (Engelbrecht, 2007, s;128).

Evrimsel çerçevede ise bir bireyin uygunluğu yalnızca dolaylı olarak başka bireylerin büyüme oranıyla karşılaştırılarak ölçülür. İtici güç de doğal seleksiyon değildir. Farklılaşmış (differential) kalıntılar (survival) ve yeniden üretim (reproduction) popülasyonda seçimi düzenler. Seçim, basit bir tanımla, zorluklara daha dayanıklı bireylerin genetik bilgilerini bir sonraki nesle aktarmaları için hayatta kalma yeteneklerinin adıdır. Biyolojide adaptasyon kavramı, popülasyondaki bazı bireylerin diğerlerine göre ekolojik veya fiziksel etkinlikte daha avantajlı olması ve aynı zamanda bu duruma gelmesine göndermede bulunur. Adaptasyon, genetik ve genetik olmayan (somatic) değişimleri kapsar ve genel anlamda “uygunluk” (fitness) ile birlikte kullanılır (Bäck, 1996, s;9).

Biyoloji ile olan ilişkisinin yanında evrimsel algoritmaların yapay zeka alanı ile de yakın bir ilişkisinden söz edilebilir. Yapay zeka, bilgisayar bilimlerinde başlangıçtan bu güne en çok ilgi çeken, en hızlı gelişen ve en zorlu alanlarından biri durumunda olmuştur. Yapay zeka, mevcut durumda insanların daha iyi yaptığı işleri bilgisayarlara nasıl yaptırılabilceği ile ilgili bir çalışma alanı olarak tanımlanabilir. Bu tanıma göre çok yönlü bir çalışma alanı olan yapay zeka, oyun oynama, teorem ispatlama, tahmin, problem çözme, algılama (örneğin konuşma algılama) ve makine öğrenmesi gibi çok farklı görevleri yerine getirmeye çalışır. Makine öğrenmesi yapay zekanın da en önemli alt çalışma alanıdır. Yapay zeka kavramı üzerine yapılan ilk araştırmalardan bu güne kadar “zeki” olarak adlandırılan sistemin, aynı ya da benzer şeyleri tekrar tekrar yaparak artırdığı deneyimi ile davranışını geliştirmesi, yanlışın ne olduğuna dair bir kavram geliştirmesi ve işlemin tekrarlanması durumunda bundan nasıl uzak durulacağını belirlemesi gerekmektedir. Bu konular makine öğrenmesinin de temel konularını oluşturmaktadır.

Yapay zeka alanında adaptasyon kavramı öğrenme ile yakın ilişkili olup çoğu zaman eş anlamlı olarak kullanılmaktadır. Yapay zeka ile arasındaki ilişkiyi daha açık ifade etmek ve çeşitli makine öğrenme stratejileri içerisinde evrimsel algoritmaların rolünü açıklamak için öğrenme stratejisinin temeli ve çıkarım sürecinin karmaşıklığına dayalı bir sınıflandırma kullanılmaktadır (Bäck, 1996, s;32-33):

- Herhangi bir sonuç çıkarım prosesinin yer almadığı, bunun yerine bilginin direkt olarak uygulandığı ‘mekanik öğrenme’,
- Mevcut bilgi ile bütünleştirmek için öğretici veya diğer bir kaynaktan bilgi gerektiren ‘direktiflerle öğrenme’,
- Çıkarsamalı ve tüm doğru çıkarımların korunduğu ‘tümdengelim ile öğrenme’,
- Eldeki bilgilerin istenen yeni konsept ile ilgili benzerliğinden yararlanarak dönüştürüldüğü ‘benzerliğe dayalı öğrenme’,
- Bütüne yönelik çıkarımların yapıldığı ‘tümevarımla öğrenme’.

Tümevarımla öğrenme mantıksal çıkarımlarla öğrenmedir ve kendi içerisinde ikiye ayrılmaktadır:

1. Örneklerden öğrenme (kavram kazanımı): bir örnek ve karşı örnek setine dayalı olarak tüm olumlu örnekleri açıklayan ve olumsuzları dışarıda bırakan genel bir kavram tanımına ulaşmaktır.
2. Gözlem ve keşiflerden öğrenme (betimsel genelleme, gözetimsiz öğrenme): geri besleme sağlayan bir öğreticinin yokluğunda, tüm yada en azından çoğu gözlemleri açıklayan düzenlilikler (regularities) yada genel kuralların araştırılmasıdır

Kullanılan mantıksal çıkarım mekanizmalarının artan karmaşıklığına göre sıralanan öğrenme stratejileri içerisinde evrimsel algoritmalar gözetimsiz öğrenmeye bir örnek olarak verilmektedir. Evrimsel algoritmaların bir öğrenme algoritması olarak sınıflandırmasının nedenleri aşağıda sıralanan özelliklerinden kaynaklanmaktadır:

- Örnek yada karşı örnekler sunan veya öğrenme sistemi ile ilgili bilgi veren bir öğretici yoktur. Bunun yerine algoritma örnekleri kendisi oluşturur.
- Yeni bir örneğin (arama noktasının) algoritma tarafından oluşturulması mevcut bilgiye dayalı tümevarımsal bir tahmindir. Eğer tahmin başarılı ise bilgi saklanır (popülasyon) değil ise seçim süreci ile silinir (Bäck, 1996, s;33).

Evrimsel algoritmaların öğrenme konusundaki karakteristiklerinden dolayı yapay zeka ile de yakından ilişkilidir. Birçok yapay zeka programında evrimsel algoritmalar heuristic arama prosedürü olarak kullanılmaktadır. Bu noktada evrimsel algoritmaların en büyük avantajı, klasik yapay zeka arama stratejilerinin spesifikliği karşısında hemen her yapıdaki problemlere universal olarak uygulanabilmesidir.

2.1. Evrimsel Algoritmalar ve Gelişimi

1950 ve 60'lı yıllarda çok sayıda bilgisayar bilimcisi, evrimin mühendislik problemleri için bir optimizasyon aracı olarak kullanılması fikri üzerinde çalışmıştır. Bu çalışmalarda yaygın fikir, probleme yönelik aday çözüm grubunun doğal genetik varyasyon ve doğal seleksiyon yapısından esinlenerek oluşturulan operatörler kullanılarak dönüştürülmesidir. Aynı yıllarda makine öğrenmesi ve optimizasyon üzerinde evrimden esinlenen çok sayıda çalışma yapılmıştır. Burada makina öğrenmesi alanı, tecrübe ile otomatik olarak gelişen bilgisayar programlarını nasıl yapabiliriz sorusuyla ilgilenmiştir.

Eş zamanlı olarak farklı yerlerde çalışan bilim adamları problem çözme görevlerinde doğadan esinlenen algoritmalar kullanmışlardır. Başlangıçta farklı kaynakların varlığı üç farklı evrimsel algoritma modelinin ortaya çıkmasını sağlamıştır. (Olariu ve Zomaya, 2006, s;6). ‘Evrimsel strateji’ (evolutionsstrategie) kavramı ilk olarak Rechenberg tarafından 1960’larda ortaya atılmış ve uçak kanatlarının gerçek değerli parametrelerini optimize etmek için kullanılmıştır. Fogel, Owens ve Wash, 1966 yılında sonlu durum makinesinde durum dönüşüm diyagramlarını rastgele dönüştüren ve en iyisini seçen bir ‘evrimsel programlama’ geliştirmişlerdir (Mitchell, 1999, s;7). İlk evrimsel algoritmalar yüzen nokta sayı (floating point number) sunumunun kullanıldığı, mutasyonun yalnızca yeniden birleştirme (recombination) operatörü olarak kullanıldığı evrimsel programlar olarak algılanmıştı. Sürekli değişken içeren çok sayıdaki probleme uygulanagelen bu programlar yakın zamanlarda kesikli problemlerin çözümünde de kullanılmaya başlanmıştır (Michalewicz, 1996, s;158).

İlk evrimsel stratejiler yalnızca tek bir bireyden oluşan bir popülasyona dayalıydı ve evrim prosesinde genetik operatör olarak da yalnızca mutasyon kullanılmıştır. Burada bir birey, yüzen değerli (float-valued) bir çift vektör olarak gösterilmiştir. $v=(x,\sigma)$. İlk vektör olan x , arama uzayında bir noktayı gösterirken, σ standart sapma vektörüdür. X ’in $x^{x+1} = x^t + N(0, \sigma)$ ile değişimi ile mutasyon gerçekleştirilmiştir. Küçük değişikliklerin büyük değişikliklerden daha sık gerçekleştiği şeklindeki biyoloji kuralı ile uyumlu olacak biçimde $N(0, \sigma)$ burada ortalaması sıfır (0) ve standart sapması σ olan tesadüfi Gaussian sayıları ifade etmektedir. Eğer tüm kısıtları sağlar ve daha iyi uygunluk sağlarsa eski bireyin mutasyona uğratılmış hali yeni birey (döl, offspring) olarak atasının yerine kabul edilir (Michalewicz, 1996, s;160).

Popülasyon mutasyona tabi tutulan tek bir bireyden oluşmuş olmasına rağmen bu evrimsel strateji yeni bireyin uygunluk bakımından ebeveyni ile yarışmasından dolayı “iki üyeli evrimsel strateji” olarak adlandırılmıştır (Michalewicz, 1996, s;160).

2.2. Evrimsel Algoritmaların Genel Özellikleri ve İşleyişi

Evrimsel hesaplamalar (EC) evrimsel proseslerin hesaplama modellerini bilgisayar temelli problem çözme sistemlerinin dizaynı ve uygulamasında anahtar elementler olarak kullanır. Evrimsel algoritmalara gönderme yapan çok sayıda evrimsel hesaplama modeli önerilmiş ve üzerinde çalışılmıştır. Tüm bu modeller seçim ve

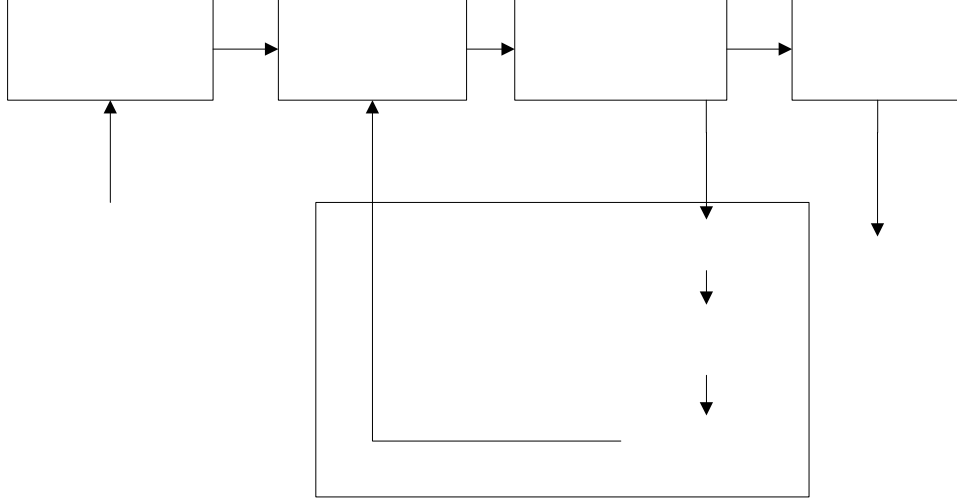
yeniden üretim proseslerini kullanarak bireysel yapıların evrimini simüle etme temeline dayalı ortak kavramlar kullanmaktadır. Tüm evrimsel algoritmalar, seçim (selection) ve yeniden üretme (reproduction) ve mutasyon gibi diğer operatörlerin kurallarına göre evrimleşen bir popülasyon sürdürür. Popülasyondaki her bir birey çevreye olan uygunluğu ölçüsünde bir değer alır. Seçim, mevcut uygunluk bilgisini kullanarak yüksek uygunluğu olan bireyler üzerinde odaklanır. Yeniden birleştirme ve mutasyon, yeni keşifler için sezgisel yöntemler (heuristics) sağlayarak bu bireyleri farklılaştırır.

Tipik bir evrimsel algoritmada bir başlangıç popülasyonu oluşturulur, değerlendirme, seçim, yeniden üretim ve mutasyon gibi tekrarlanan uygulamalarla nesilden nesile dönüştürülür. Popülasyon büyüklüğü aksini gerektirecek bir durum olmadıkça genelde sabit tutulur. Başlangıç popülasyonu da genel olarak tesadüfi olarak belirlenir. Değerlendirme her bir bireyin belirlenen çevre içindeki değerine göre uygunluğunu ölçmektedir. Bu işlem bir uygunluk fonksiyonunu hesaplamak kadar kolay olabileceği gibi çok karmaşık bir simülasyonun çalıştırılması gibi kompleks de olabilmektedir. Seçim ise genel olarak iki aşamalı olarak ele alınmaktadır: Ailenin (ebeveyn) seçimi ve hayatta kalma. Ebeveyn seçimi hangi bireylerin ebeveyn olacağını ve onların da kaç çocuğa sahip olacağını belirlenmesini kapsar. Çocuklar yada yeni bireyler, ebeveynler arasında bilgi değişimini gerçekleştiren yeniden birleştirme (recombination) ve yeni bireyleri daha da dönüştüren mutasyon vasıtasıyla üretilir. Çocuklar ya da yeni bireyler daha sonra yeniden değerlendirilir ve son olarak popülasyonda hangisinin kalacağına karar verilir. Tipik bir evrimsel algoritma işleyişi Şekil 2.1’de gösterilmiştir (Kennedy ve diğer., 2001, s;144, Sumathi ve diğer., 2008, s;32).

Evrimsel, bireylerin tesadüfi olarak seçilen bir popülasyonunun doğal seçimi olarak muhtemel kromozom değerlerinin değer aralığı alanı boyunca bir arama olarak düşünülebilir. Bu açıdan bir evrimsel algoritma verilen problemin optimal çözümünün bulunması için stokastik bir aramadır. Evrimsel arama süreci evrimsel algoritmaların aşağıdaki ana bileşenlerinden etkilenmektedir (Engelbrecht, 2007, s;128):

- Problem çözümünün kromozom olarak kodlanması,
- Uygunluğu değerlendiren bir fonksiyon veya bireylerin yaşamda kalma gücü
- Başlangıç popülasyonu için ilk atamaların yapılması,
- Operatörlerin seçimi,

- Operatörlerin yeniden üretilmesi (reproduction).



Şekil 2.1. Evrimsel algoritma işleyişi

Bir evrimsel algoritma matematiksel ifade ile 8 bileşenli olarak aşağıdaki gibi tanımlanabilir:

$EA = (I, \Phi, \Omega, \Psi, s, \iota, \mu, \lambda)$ Burada $I = A_x \times A_s$ bireyi, A_x ve A_s göreceli seçilim setleri göstermektedir. $\Phi : I \rightarrow R$ bireylere gerçek değerler atayan uygunluk fonksiyonunu;

$\Omega = \{\omega_{\theta_1}, \dots, \omega_{\theta_x} \mid \omega_{\theta_i} : I^\lambda \rightarrow I^\lambda\} \cup \{\omega_{\theta_0} : I^\mu \rightarrow I^\lambda\}$ ise ω_{θ_i} , olasılıklı genetik operatörler setini göstermektedir. Bu operatörlerin her biri $\theta_i \subset R$ setinde gösterilen parametreler tarafından kontrol edilmektedir.

$s_{\theta_s} : (I^\lambda \cup I^{\mu+\lambda}) \rightarrow I^\mu$ seçim operatörünü göstermektedir. Burada birey sayısı λ veya $\lambda + \mu$ ile μ arasında değişmektedir. θ_s , seçim operatörü olarak ilave bir parametre seti olarak kullanılabilir. μ , ailedeki birey sayısını; μ , ise çocuk (döl) sayısını göstermektedir. Son olarak $\iota : I^\mu \rightarrow \{\text{dogru, yanlış}\}$ evrimsel algoritma için sonlandırma kriterini; $\Psi : I^\mu \rightarrow I^\mu$, P popülasyonunun genetik operatörler ve seçim uygulanarak bir sonraki popülasyona dönüştürme prosesinin tamamını tanımlamaktadır.

Bir evrimsel algoritmanın genel işleyişi aşağıdaki algoritmada gösterilmiştir (Bäck, 1996, s;63-66):

$t = 0$;

Başla $P(0) = \{\vec{a}_1(0), \dots, \vec{a}_\mu(0)\} \in I^\mu$

Hesapla $P(0) : \{ \Phi(\vec{a}_1(0)), \dots, \Phi(\vec{a}_\mu(0)) \};$

While ($\iota(P(t)) \neq \text{dogru}$) do

Yeniden birleştir: $P'(t) := r_{\theta_r}(P(t));$

Mutasyon: $P''(t) := m_{\theta_m}(P'(t));$

Değerlendir $P''(t):$

Seç : $P(t+1) := s_{\theta_s}(P''(t) \cup Q);$

$t = t+1;$

son

Evrimsel algoritmaların adımları aşama aşama tekrarlanarak uygulanır ve bu durum durdurma koşulları sağlanıncaya kadar sürer. Her yeni iterasyon yeni bir jenerasyona anlamına gelmektedir (Engelbrecht, 2007, s;128).

Tüm evrimsel algoritmalarda başlangıç popülasyonu algoritmaya bağlı bir metotla belirlenir ve olasılıklı yeniden birleştirme, mutasyon ve seleksiyon (seçim) prosesleri kullanılarak arama uzayının daha iyi bölgelerine doğru adım adım gelişir/evrime uğrar. Yeni bir arama noktası için nitelik bilgisi ya da uygunluk değerini çevre/doğa verir. Yeniden birleştirme mekanizmaları, ana babaya ait bilginin yeni nesillere geçerken birbirine kaynaşmasını (mix) sağlar. Mutasyon ise popülasyona yenilikler kazandırır. Tüm bu mekanizmalar evrimsel algoritmaların farklı metotları altında farklı biçimde uygulanmaktadır (Bäck, 1996, s;7).

Evrimsel algoritmaları diğer arama metotlarından ayıran iki önemli özellik söz konusudur (Sarker ve diğer., 2003, s;29):

- Tüm evrimsel algoritmalar popülasyon temellidir.
- Popülasyondaki bireyler arasında etkileşim ve bilgi değişimi mevcuttur.

Günümüzde biyolojideki evrim kavramından esinlenen farklı metotlar evrimsel hesaplamalar şemsiyesi altında toplanabilir. Evrimsel stratejiler, evrimsel programlama ve genetik algoritmalar evrimsel hesaplamaların üç ana bileşenini oluşturmaktadır (Mitchell, 1999, s;7, Karr ve Freeman, 1999, s;2, Spall, 2003, s;259).

Bu üç yaklaşım arasında temelde nesilden nesile geçiş ve popülasyon bileşenlerinin bilgisayarda gösterimi konularında farklılıklar söz konusudur. Her bir yaklaşım bir diğerinden birtakım kavramları almış ve uyarlamıştır. Evrim modelinden

esinlenildiğinden bu yana evrimsel algoritmalar alanında ortaya konulan tüm çalışmaların bir dizi ortak yanı bulunmaktadır. Bu ortak özellikler bir yandan da evrimsel algoritmaların genel karakteristiklerini yansıtmaktadır (Bäck ve diğer., 2000a, s; 59):

- Evrimsel algoritmalarda her bir birey genellikle problemin potansiyel çözüm alanında bir noktayı temsil eder. Buna ilaveten bireyler strateji parametreleri gibi daha başka bilgileri de kapsayabilir.
- Yeni nesil bireyler mutasyon veya çaprazlama gibi tesadüfi süreçlerle elde edilir. Mutasyon bireyin yanlış bir kendini tekrarına tekabül eder. Çaprazlama ise bir veya daha fazla mevcut birey arasında karşılıklı bilgi değişimidir.
- Bireyleri kendi çevrelerinde değerlendirerek bir kalite ya da uygunluk değeri ölçüsü tanımlanabilir. En azından bireyler hakkında iyi veya kötü kararını verebilmek için uygunlukların karşılaştırılabilir. Bu uygunluk değerine göre seçim prosesi yeniden üretim için iyi bireyleri daha sıklıkla seçer.

Evrimsel algoritmalarda biyolojiden alınan terminoloji ve bunun kullanımı aşağıdaki çizelgede özetlenmiştir (Rutkowski, 2005, s;267, Sumathi ve diğer., 2008, s;4):

Biyoloji Terimi	Evrimsel Algoritma Kullanımı
Kromozom	Sembollerin dizilimi
Popülasyon	Bir kromozom seti
Deme	Yakın ilişkili kromozomların lokal popülasyonu
Gen	Bir özellik, karakter veya detektör
Alel	Özelliğin aldığı değer
Lokus	Kromozom üzerindeki bir yer veya pozisyon
Genotip	Yapı
Fenotip	Bir parametre seti, alternatif bir çözüm/kodlanmış yapı

Çizelge 2.1. Evrimsel algoritma terminolojisi

Evrimsel hesaplamaların teorik analizinde iki temel yaklaşım şema temelli analiz ve Markov zincirleridir. Şema temelli yaklaşım evrimsel hesaplamaları ciddi bir temele yerleştirmeye çalışan ilk önemli yaklaşımdır. Sonraları ise daha başarılı bir

biçimde Markov zincirleri kullanılmıştır. Markov temelli yaklaşım şema temelli yaklaşımda önemli kısaltmalar getirmiş, birçok önemli eksikliğini ortaya koymuş ve şema temelli yaklaşımın anlamı ile (implication) çelişen bir teoriye yol açmıştır.

2.3. Evrimsel Algoritmaların Bileşenleri

Evrimsel algoritmalar çok sayıda bileşen, prosedür veya operatöre sahiptir. Belirli bir evrimsel algoritmanın tanımlanabilmesi için bu yapıların tayin edilmesi, belirlenmesi gerekmektedir. En önemli evrimsel algoritma bileşenleri aşağıda açıklanmıştır:

Gösterim:Gösterim (representation) çözüm yada bireylerin evrimsel algortmada sunulma biçimidir. İçerisinde dış görünüş, davranış, fiziksel nitelikler vb. özellikleri kodlar. Açıklayıcı ve dönüştürülebilir bir gösterimin oluşturulması evrimsel algoritmaların en zor aşamalarından birini oluşturmaktadır. Evrimsel algoritmalarda gösterimde ikili bir bit dizisinin kullanımı standart haline gelmiştir. Genetik gösterimi kullanışlı kılan ana özellik sabit boyutu dolayısıyla parçalarının kolaylıkla sıraya dizilebilmesidir. Bu özellik çaprazlama işlemini de ayrıca kolaylaştırır.

Değerlendirme / Uygunluk Fonksiyonu: Evrimsel algoritmalarda uygunluk fonksiyonu kromozomların optimallliği sağlama derecelerini belirlemeye çalışan amaç fonksiyonunun bir biçimidir. Bu fonksiyon verilerine göre çözümler yada kromozomlar derecelerine göre sıralanmaktadır. Ancak optimale daha yakın kromozomların daha da iyi bir nesil elde etmek üzere eşleşme ve veri değişimi proseslerine katılımına izin verilir. Fonksiyon birçok durumda optimize edilmesi düşünülen problem tarafından belirlenir. Ancak, uygunluk fonksiyonunun tanımı bazı durumlarda açık ve belirli değildir ve birçok durumda aşamalı olarak belirlenmektedir.

Uygunluk fonksiyonunu hem çözüm yada kromozomun ne kadar iyi oluşunu göstermeli; hem de optimal değerin ne kadar yakınına tekabül ettiğine dair de bilgi vermelidir (Sumathi ve diğer., 2008, s;37).

Popülasyon: Evrimsel algoritmalarda popülasyon ile ilgili iki temel konu başlangıç popülasyonunun büyüklüğü ve üyelerinin belirlenme prosedürüdür. Büyük popülasyon seçimi faydalı olabilmektedir ancak hesaplama için büyük oranda hafıza ve zaman gerektirmektedir. Küçük popülasyon seçildiğinde de bireylerin arama uzayına

uygun biçimde yayılmaması riski söz konusu olur. Bu açıdan etkinlik için doğru popülasyon büyüklüğünün seçimi önem taşımaktadır.

Başlangıç popülasyonunu oluşturmak için iki yol kullanılmaktadır: sezgisel yada tesadüfi seçim. Sezgisel seçim çözüm alanının yalnızca küçük bir kısmını tarar ve popülasyonda çeşitlilik olmadığı için optimal çözüme ulaşamaz. Bundan dolayı başlangıç popülasyonu kodlama metotları ile oluşturulur ve tesadüfi seçim yapılır. Bununla birlikte tesadüfi seçim başlangıç popülasyonu için sezgisel tekniklerin dahil edilmesi ile birbirini tamamlayabilir. Evrimsel algoritma, bu algoritmaları kullanarak daha hızlı yakınsama ve daha iyi çözümler elde etme gibi yararlar sağlayabilir (Sumathi ve diğer., 2008, s;37, Olariu ve Zomaya, 2006, s;19).

Ebeveyn Seçim Mekanizması: Seçim, eşleştirme için hangi bireylerin seçileceğini ve seçilen bireylerden kaç tane yeni birey elde edileceğinin belirlenmesini kapsar. Bu süreçte ilk aşama ‘sıralama tabanlı uygunluk tayini’ (rank-based fitness assignment) yada ‘çok amaçlı sıralama’ya (multiple-objective ranking) göre yapılacak belirlemedir. Seçim havuzundaki her bir birey kendi uygunluk değerleri ve diğerlerinin uygunluk değerlerine bağlı olarak bir eşleşme olasılık değeri alır. Bu değer daha sonraki seçim aşamasında seçilecek bireylerin belirlenmesinde kullanılmaktadır.

Evrimsel algoritmalarda seçim sürecinde farklı seçim biçimlerini karşılaştırmada kullanılan bazı terimler aşağıda sıralanmıştır:

- Seçim baskısı: Tüm bireylerin ortalama seçilme olasılığına göre seçilen en iyi bireyin olasılık değeri
- Sapma (Gen havuzu yakınsaması) (bias): Bir bireyin normalleştirilmiş uygunluğu ile beklenen eşleştirme olasılığı arasındaki mutlak fark
- Yayılım (spread): Bir bireyden elde edilen bireylerin sayısı için muhtemel değerler aralığı
- Çeşitlilik kaybı: Seçim aşamasında seçilmeyen popülasyon bireylerinin oranı.
- Seçim yoğunluğu: Normalize edilmiş Gaussian dağılımına bir seçim metodunun uygulanmasından sonra popülasyonun beklenen ortalama uygunluk değeri
- Seçim varyansı: Normalize edilmiş Gaussian dağılımına bir seçim metodunun uygulanmasından sonra popülasyonun uygunluk dağılımının beklenen varyansı

Sıralama Tabanlı Uygunluk Tayini (rank-based fitness assignment): Sıralama tabanlı uygunluk tayininde popülasyon hedef değerlerine göre tasnif edilmektedir. Her bir bireye verilen uygunluk, gerçek amaç fonksiyonu değerine değil yalnızca bireylerin sıralamasındaki yerine göre belirlenmektedir. Sıralama tabanlı uygunluk tayini, oransal uygunluk tayinindeki ölçekleme problemini de ortadan kaldırmaktadır. Herhangi bir bireyin fazla sayıda yeni birey (offspring) üretmesinin önüne geçmek için üretkenliği sınırlanmaktadır. Sıralama, popülasyon için uniform bir ölçekleme ortaya koyar ve seçim baskısının kontrolünde etkinlik sağlar (Sumathi ve diğer., 2008, s;39).

N , popülasyondaki birey sayısı, P , bireyin popülasyondaki yeri ($0 \leq P \leq N$) ve doğrusal sıralamada $[1,2]$ aralığında değer alabilen SP 'nin belirlenen seçim baskısı olduğu durumda doğrusal olarak sıralama uygunluğu aşağıdaki formülle

$$\text{hesaplanmaktadır: } UygunlukP = 2 - SP + 2(SP - 1) \frac{(P - 1)}{(N - 1)}$$

Doğrusal sıralamaya göre daha yüksek seçim baskısına olanak sağlayan doğrusal olmayan sıralamada uygunluk değerleri aşağıdaki formülle hesaplanmaktadır:

$$Uygunluk P = \frac{N * X^{P-1}}{\sum_{i=1}^N X^{i-1}}$$

Bu formülde kullanılan X , $0 = (SP - N) * X^{N-1} + SP * X^{N-2} + ... + SP.X + SP$ polinomunun kökü olarak hesaplanmaktadır. Doğrusal olmayan sıralamada seçim baskısı SP , $[1, N-2]$ aralığında değer alabilmektedir.

Oransal yada doğrusal temelli uygunluk tayininde bireylerin yalnızca bir amaç fonksiyon değeri ortaya koyduğu kabul edilir. Gerçekte ise çok sayıda kriter mevcuttur. Tüm bu kriterlerin karşılaştırılması ile bir bireyin diğerine göre üstünlüğüne karar verilebilir. Bu aşamadan sonra tek amaçlı problemlerde olduğu gibi iki yönlü karşılaştırmalar ile bireyler sıralanır. Bu sıralama yapıldıktan sonra tek amaçlı sıralama metotları bireylerin sırasını dönüştürmek için kullanılabilir (Sumathi ve diğer., 2008, s;39).

Evrimsel algoritmalarda uygunluklarına göre ebeveynler çok farklı algoritmalar kullanılarak eşilmektedir. Rulet tekerleği seçimi, stokastik evrensel örnekleme, lokal seçim, budama (truncation) seçimi, turnuva seçimi bu algoritmalarından bazılarıdır (Sumathi ve diğer., 2008, s;43).

Yeniden Birleştirme (recombination): Yeniden birleştirme eşleştirme havuzundaki ebeveynlerin taşıdığı bilgileri birleştirerek yeni bireyler ortaya koyar. Bireylerin değişkenlerinin sunum biçimine bağlı olarak yeniden birleştirme aşağıdaki biçimlerde uygulanmaktadır:

Kesikli yeniden birleştirme: Kesikli yeniden birleştirmede bireyler arasında değişken değerleri değiştirilir. Değişkenlerini yeni nesle aktaracak ebeveyn eşit olasılıkla tesadüfi olarak seçilir.

$$Var_i^O = Var_i^{P_1} . a_i + Var_i^{P_2} . (1 - a_i) \quad i \in (1, 2, \dots, N_{var}), \quad a_i \in \{0, 1\} \text{ aralığında}$$

tesadüfi dağılmaktadır. Kesikli yeniden birleştirme ikili, tamsayı, reel veya sembol vb. tüm değişken çeşitleri için kullanılabilir.

Intermediate yeniden birleştirme: Intermediate yeniden birleştirme yalnızca gerçek değerli değişkenlere uygulanabilmekte; ikili (binary) değişkenlere uygulanamamaktadır. Bu metotta yeni bireyin alacağı değer ebeveynlerin değerlerinin arasında yada belirlenen oranda etrafından seçilmektedir. Yeni bireylerin elde edilmesinde aşağıdaki kural uygulanmaktadır:

$$Var_i^O = Var_i^P . a_i + Var_i^P . 1 - a_i \quad i \in (1, 2, \dots, N_{var}) \text{ ve } a_i \in [-d, 1 + d] \text{ aralığında}$$

uniform olarak tesadüfi belirlenmektedir. Burada a , ölçekleme faktörüdür. d parametresinin değeri de muhtemel yeni bireylerin değer alacağı alanı belirlemektedir. $d=0$ olduğu durumda yeni neslin değer alacağı alan ebeveynler ile aynı ölçekte demektir. Yeni bireylerin birçok değişkeni mümkün olan alanın sınırına yakın oluşturulmadığı için birkaç nesil sonra bu alanın daralması söz konusu olmaktadır. Bu olumsuz durumdan kaçınmanın yolu d 'nin daha büyük değeri seçilmesidir. Literatürde 0.25 olarak seçilen d değerinin yeni bireylere ebeveynleri ile aynı büyüklükte alanı tarama imkanı verdiği belirtilmektedir (Sumathi ve diğer., 2008, s;54).

Satır (line) yeniden birleştirme: Satır yeniden birleştirmenin intermediate yeniden birleştirmeden farkı tüm değişkenler için tek bir a değeri kullanılmasıdır.

$$Var_i^O = Var_i^P . a_i + Var_i^P . 1 - a_i \quad a_i \in [-d, 1 + d] \text{ aralığında uniform olarak}$$

tesadüfi belirlenmektedir. Yukarıda d için belirtilenler bu metod için de geçerlidir. Satır yeniden birleştirme ile ebeveynler tarafından belirlenen doğru üzerinde herhangi bir noktada yeni bireyler üretilmektedir.

Uzatılmış satır (extended line) yeniden birleştirme: Uzatılmış (extended) satır (line) yeniden birleştirmede ebeveynlerin değişken değerleri tarafından belirlenen bir

doğru üzerinde yeni bireyler oluşturulur. Ancak ebeveynin belirlediği kısım ve bunun yakın komşuluğu ile sınırlı değildir. Ebeveyn yalnızca yeni bireylerin hangi doğru üzerinde olacağını belirler. Muhtemel yeni bireylerin yer alacağı alanın büyüklüğü değişkenlerin alanı (domain) tarafından belirlenir. Bu alan içerisinde yeni bireyler uniform bir dağılım göstermemektedir. Bununla birlikte ebeveynlere yakın bir noktada yeni bireyler üretilmesi olasılığı da yüksektir. Eğer ebeveynler uygun ise yeni bireyler de doğru üzerinde kötüden iyiye giden bölgede üretilecektir.

Uzatılmış (extended) satır (line) yeniden birleştirme yalnızca gerçel değişkenlerle (real variables) uygulanabilmekte, ikili veya tamsayılı değişkenlerle uygulanamamaktadır (Sumathi ve diğer., 2008, s;57).

İkili değerli (binary) yeniden birleştirme: İkili değerli değişkenlerin yeniden birleştirilmesi için ‘çaprazlama’ terimi kullanılmaktadır. Evrimsel algoritmalar için her iki nosyon da (çaprazlama, yeniden birleştirme) aynı anlama gelmektedir. Genetik algoritmalar uygulamalarında daha çok ikili kodlama kullanılmaktadır.

Mutasyon:Yeniden birleştirmeden (recombination) sonra ortaya çıkan her yeni birey mutasyonla tesadüfi olarak değişime uğratılır. Bu değişim için genel olarak düşük olasılık oranları seçilir.

Mutasyon aşamalarının ve oranının belirlenmesinde iki yaklaşım mevcuttur (Sumathi ve diğer., 2008, s;61):

- Bütün parametreler tüm evrim süreci boyunca sabittir. Reel sayılar ve ikili (binary) değişkenlerin mutasyonu bu duruma örnek olarak verilebilir.
- Bir veya tüm parametreler önceki mutasyonlara göre yeniden adapte edilir. Bu duruma örnek olarak da evrimsel stratejilerde mutasyon adımı büyüklüğünün adaptasyonu verilebilir.

Reel değerli mutasyon: Reel değişkenler için mutasyon tesadüfi olarak üretilen değişkenlerin düşük bir olasılık oranı ile mevcut değişkenlere ilave edilmesidir. Bu açıdan değişkeni mutasyona uğratma olasılığının (mutasyon oranı) ve mutasyona uğrayan her bir değişken için değişimin büyüklüğünün (mutasyon aşaması) tanımlanması gerekmektedir.

Bir değişkeni mutasyona uğratma olasılığı değişken sayısı ile ters orantılıdır. Birey ne kadar fazla değişkene sahip ise mutasyona uğrama olasılığı da o derece küçük olmaktadır. Birçok çalışmada $1/n$ (n : bireyin değişken sayısı) mutasyon oranının iyi

sonular verdiđi belirtilmiřtir. Bu her mutasyonda birey bařına bir deđiřkenin deđiřtirilmesi demektir. Bylece mutasyon oranı da poplasyon byklđnden bađımsız olmaktadır.

Mutasyon adımıının byklđnn belirlenmesi de seimi zor bir konudur. Optimal adım byklđ genellikle seilen probleme gre deđiřmektedir. Genellikle kk mutasyon adımlarının seilmesinin daha iyi sonular verdiđi grlmřtir. Bununla birlikte byk mutasyon adımları bazı durumlarda sonuca daha hızlı ulařmayı da sađlayabilmektedir. Sonu olarak iyi bir mutasyon operatrnn kk adım byklklerini yksek bir olasılıkla; byk adım byklklerini dřk bir olasılıkla retmesi daha iyi sonular verebilmektedir (Sumathi ve diđer., 2008, s;62).

İkili Deđerli (binary) Mutasyon: İkili deđerli bireylerde her deđerken yalnızca iki farklı durumda olabileceđi iin mutasyonun da bu deđerken deđerinin diđerine dnřtrlmesidir. Bylece mutasyon adımıının byklđ her durumda 1 olmaktadır.

Hayatta kalanı seme mekanizması (deđerirme, reinsertion): nceki poplasyondan bireylerin seimi, yeniden birleřtirilmesi ve mutasyonu ile yeni birey oluřturulunca ilk olarak bu bireyin de uygunluđunun belirlenmesi gerekmektedir. İlk poplasyon byklđnden daha az sayıda yeni birey retilmiř ise poplasyon byklđn korumak iin bunların eski poplasyona dahil edilmeleri gerekmektedir. Benzer řekilde, retilen tm yeni bireyler poplasyona dahil edilmeyecek ise veya poplasyon byklđndan daha fazla sayıda yeni birey retilmiř ise hangi bireylerin yeni poplasyonda yer alacađının belirlenmesi iin ekleme (reinsertion) planının uygulanması gerekmektedir.

Ekleme planı kullanılan seim algoritmasına gre belirlenmektedir. Rlet tekerleđi seimi, stokastik rnekleme ve budama seimi gibi tm poplaston temelli seim algoritmaları iin global ekleme; lokal seim algoritmaları iin ise lokal ekleme kullanılmaktadır (Sumathi ve diđer., 2008, s;67).

Global Ekleme (reinsertion): Global eklemenin uygulanmasında da kendi arasında farklı uygulamalar vardır:

- Ebeveynler kadar yeni bireyler ret ve tm ebeveynleri yeni bireyler ile deđerir (saf ekleme).
- Ebeveynlerden daha az sayıda yeni birey ret bunları uniform olasılıkla tesadfi olarak ebeveynlerle deđerir (uniform ekleme).

- Ebeveynlerden daha az sayıda yeni birey üret ve bunları en kötü ebeveynler ile değiştir (elitist ekleme).
- Çok sayıda yeni birey üret ve yalnızca en iyilerini popülasyona dahil et (uygunluk temelli ekleme).

Saf eklemede tüm bireyler yalnızca bir jenerasyon boyunca yaşar. Çok iyi bireylerin daha da iyi bireyler üretmeden değiştirilmesi ihtimali yüksek olduğu için bilgi kaybı söz konusu olabilmektedir. Elitist ve uygunluk temelli yaklaşımda bu kaybın önüne geçilebilmekte ve en iyi bireyler nesiller boyunca yaşayabilmektedir.

Lokal Ekleme (reinsertion):Lokal seçimde ekleme tamamen sınırlı bir komşuluk dahilinde gerçekleşmektedir. Böylece lokal bilginin korunması amaçlanmaktadır. Seçim ve değiştirmede kullanılan farklı metotlar aşağıda sıralanmıştır (Sumathi ve diğer., 2008, s;68):

- Belirlenen komşuluk dahilindeki bireyleri uniform olasılıkla yeni bireylerin tümü ile değiştir.
- Her bir yeni bireyi ekle ve bunları komşuluk dahilindeki en zayıf bireylerle değiştir.
- Yeni bireyleri uygunluğuna göre seçerek komşuluk dahilindeki en zayıf bireylerle değiştir.
- Yeni bireyleri uygunluğuna göre seçerek komşuluk dahilindeki ebeveynler ile değiştir.
- Yeni bireyleri uygunluğuna göre seçerek komşuluk dahilinde uniform olasılıkla tesadüfi olarak belirlenen bireyler ile değiştir.
- Ebeveynlerinden daha iyi olan yeni bireyleri ebeveynleri ile değiştir.

2.4. Evrimsel Algoritmaların Avantajları

Evrimsel algoritma kullanımının diğer global optimizasyon tekniklerine göre avantajları aşağıda sıralanmıştır (Sivanandam ve Deepa, 2008, s;9-12, Sumathi ve diğer., 2008, s;72):

- Evrimsel algoritmalar kavramsal olarak basittir. Algoritma başlangıç popülasyonu, aşamalı çeşitlendirme ve seçimi kapsar.

- Evrimsel algoritmaların performansı, yalnızca sürekli veya kısıtlı değişkenler ile uygulanabilen diğer bazı nümerik tekniklerin aksine gösterimden bağımsızdır.
- Evrimsel algoritmaların geniş bir uygulama alanı mevcuttur. Fonksiyon optimizasyon problemi olarak formüle edilebilen herhangi bir probleme uygulanabilmektedirler.
- Evrimsel algoritmalar mukayeseli olarak problem ile ilgili eldeki verilerin daha iyi dahil edilebildiği bir çerçeve sunar. Bu gibi bilgilerin kullanılması evrimsel aramaya odaklanarak mevcut çözüm alanının daha etkin biçimde taranması sonucunu doğurur.
- Evrimsel algoritmalar çok sayıda geleneksel optimizasyon teknikleri ile de bütünleştirilebilir. Bu, ilk aramanın bir evrimsel algoritma ile yapılmasından sonra derece derece değişen (gradient) minimizasyonun uygulanması kadar basit olabildiği gibi diğer algoritmalar ile eş zamanlı yada hibrit bir uygulamayı da kapsayabilmektedir.
- Her bir çözümün değerlendirmesi paralel olarak yapılabilmektedir. Yalnızca seri işlem gerekmektedir. Tavlama benzetimi ve Tabu araması gibi diğer bazı global optimizasyon yöntemlerinde tam anlamı ile paralellik mümkün değildir.
- Geleneksel optimizasyon metotları problem konusundaki dinamik değişikliklere karşı yeterince esnek ve etkili değildir ve çözüme ulaşmak için sıklıkla tamamen baştan başlanmasını gerektirir. Evrimsel algoritmalar ise çözümün değişen koşullara uyarlanmasıyla kullanılabilmektedir.
- Evrimsel algoritmaların en önemli avantajı insanların hiç deneyimi olmayan problemleri adresleyebilmesidir. İnsan deneyimi söz konusu olduğunda kullanılabilmemesine rağmen problem çözme mekanizmasını otomatize etmede etkin olmaktan uzaktır.

2.5. Evrimsel Algoritmaların Uygulama Alanları

Evrimsel algoritmalar birçok farklı alanda etkin biçimde kullanılmaktadır. Bunların arasında en fazla göze çarpanı ise birleşik (combinatorial) optimizasyon problemi uygulamalarıdır. Seyyar satıcı problemi, çoklu sırt çantası problemi ve rakam

bölümlendirme gibi klasik NP-hard problemlerin çözümünün yanı sıra program oluşturma, zaman tablosu oluşturma (timetabling) kuadratik atama, yerleştirme problemi, taşıma problemi ve araç rotalama gibi klasik olmayan diğer birleşik optimizasyon problemlerinin çözümünde de kullanılmaktadır. Diğer bir uygulama sahası ise telekomünikasyondur. Anten yerleşimleri, frekans atamaları, veri network dizaynları, bant genişliği talep tahminleri bu alandaki kullanımlara örnek verilmektedir.

Evrimsel algoritmalar elektronik ve diğer mühendislik uygulamalarında da yaygın olarak kullanılmaktadır. Yapı optimizasyonu, havacılık dizaynları, güç (power) planlama, uzman sistemler, bilgisayar destekli dizayn ve analog-network sentezi konularında da evrimsel algoritmalar çözüme yönelik kullanılmaktadır. Yukarıda sıralanan alanların dışında, evrimsel algoritmaların tıp, matematik, biyoloji gibi diğer birçok alanda da kullanımı söz konusudur (Olariu ve Zomaya, 2006, s;14; Sivanandam ve Deepa, 2008, s;12).

2.6. Evrimsel Strateji

1960'lı yılların başında Berlin Teknik Üniversitesi'nde Bienert, Rechenberg ve Schwefel tarafından ortaklaşa geliştirilen evrimsel strateji kavramı, ilk olarak bükülen bir borunun şekil optimizasyonu, birleştirilen iki plakanın sürtünme minimizasyonu gibi hidrodinamik problemlerin deneysel çözümünde kullanılmıştır. Bu tür optimizasyon problemlerinin analitik olarak veya geleneksel yöntemlerle çözülememesi nedeni ile deneysel ayarlamalara dayanak teşkil edecek biçimde tesadüfi değişimlere dayalı bir algoritmik metot geliştirilmiştir (Bäck, 1996, s;67, Olariu ve Zomaya, 2006, s;6).

Evrimsel stratejilerde sabit uzunluklu gerçek değerli vektör sunumu kullanılmaktadır. Genetik algoritmalarda olduğu gibi bu vektör üzerindeki her bir kısım bireyin belirli bir özelliğine tekabül etmektedir. Bu özellikler yapısal olmaktan ziyade davranışsal nitelik taşımaktadır (Sivanandam ve Deepa, 2008, s;4).

Evrimsel stratejilerin önemli özelliklerinden biri mutasyon uygulamalarını kontrol etmek için kendi kendine adapte olan mekanizmalar kullanmasıdır. Bu mekanizmaların aramanın ilerlemesini optimize etmesi amaçlanır. Bunu gerçekleştirmek için de yalnızca üzerinde çalışılan problemin çözümleri değil bununla birlikte bu çözümleri mutasyona uğratmak için bazı parametreleri de dönüştürülmektedir (Olariu ve Zomaya, 2006, s;6).

Bilgisayarlarda kullanılan organik evrim prensiplerini taklit etmeye dönük ilk çabalar o zamana kadar bilinen, aşamalı (iterative) optimizasyon metodlarına benzemektedir. İki üyeli bir evrimsel stratejide bir ebeveyn normal dağılımlı mutasyonlar uygulanarak her jenerasyonda bir yeni birey (çocuk) meydana getirmektedir. Örneğin çocuk atasından daha iyi performans gösterip onun yerini alıncaya kadar küçük değişikliklerin olması, büyük değişikliklere göre daha muhtemeldir. Bu basit yapıdan dolayı eşik (step) büyüklüğü kontrolü ve yakınsama hızı için teorik sonuçlar türetilir. Başarılı ve tüm mutasyonlar arasındaki oran $1/5$ 'tir. Yalnızca mutasyon kullanan bu ilk algoritma m ebeveynin mevcut bulunması dolayısıyla yeniden birleştirme'nin (recombination) de dahil edildiği bir $(m+1)$ çözümüne yükseltilir. Mutasyon düzeni ve dışardan (exogenous) eşik büyüklüğü kontrolü iki üyeli evrimsel stratejide de aynı kalmıştır. Schwefel daha sonra bu stratejiyi $(m+1)$ ve (m,l) olarak gösterilen iki üyeli evrimsel stratejiye genelleştirmiştir. Bu strateji, organik evrimin aşağıdaki prensiplerini taklit eder (Kenedy ve diğer., 2001, s;140, Rutkowski, 2005, s;289).

Bir popülasyon, tesadüfi eşleşme (mating) ile yeniden birleştirme (recombination) olasılığı, mutasyon, seçim (selection). Bu stratejiler artı (+) stratejisi ve virgül (,) stratejisi olarak adlandırılır. İlkinde ebeveynler seçimde hesaba katılırken ikinci stratejide yalnızca yeni birey (çocuk, offspring) seçime dahil olur. Burada m popülasyon büyüklüğünü gösterirken l de genellikle her bir jenerasyonda oluşturulan yeni birey sayısını göstermektedir.

Evrimsel strateji algoritması, n adet bireyden oluşan bir toplum üzerinde çalışır. Başlangıç toplumu rastgele olarak üretilir. Her bireyin gerçel sayıların oluşturduğu bir vektör olan kromozomu bulunur. Kromozomlar iki bölümden oluşmaktadır: Hedef parametreleri olan genler $(x_1, \dots, x_n)$ ve strateji parametreleri olan mutasyon eşik-adım büyüklüğü olan sapmalar $(\sigma_1, \dots, \sigma_n)$ böylece bireylerin genel gösterimi $(x_1, \dots, x_n, \sigma_1, \dots, \sigma_n)$ şeklindedir. Her kromozom, hedef üyeyi bulmada algoritmanın olası bir çözümünü temsil eder. μ toplumdaki bireylerin sayısını yani ebeveyn sayısını gösterir. λ ise bu toplumdan üretilecek çocukların sayısını temsil eder. λ adet çocuğun üretilmesi için ebeveyn seçimi, yeniden birleştirme ve mutasyon adımları birbirini izleyecek biçimde λ kez tekrarlanır. Yeni bireylerin (çocukların) üretilmesinden sonra hayatta kalanların seçimi gerçekleştirilir. Bu aşamada hayatta kalanlar seçilen stratejiye göre ya

sadece çocukların oluşturduğu kümeden yada ebeveyn ve çocukların oluşturduğu kümeden seçilirler. Bu seçim aşamasından sonra seçilen bireylerle yeni toplum oluşturulur. Evrimsel strateji akış şeması Sekil 2.X'te gösterilmiştir (Rutkowski, 2005, s;291).

Evrimsel stratejilerin uygulama adımları aşağıda sıralanmaktadır (Sarker ve diğer., 2003, s;32):

1. μ bireyden oluşan bir başlangıç popülasyonu oluşturulur. Her bir birey ikili gerçek değerli vektörden oluşturulur (x_i, η_i) ($i=0,1,2,\dots,\mu$)
2. Her bir bireyin uygunluk değeri hesaplanır.
3. Her bir ebeveyn (x_i, η_i) , ortalama olarak λ/μ yeni bir birey (x'_i, η'_i) oluşturur. Bunun için aşağıdaki formül kullanılmaktadır:

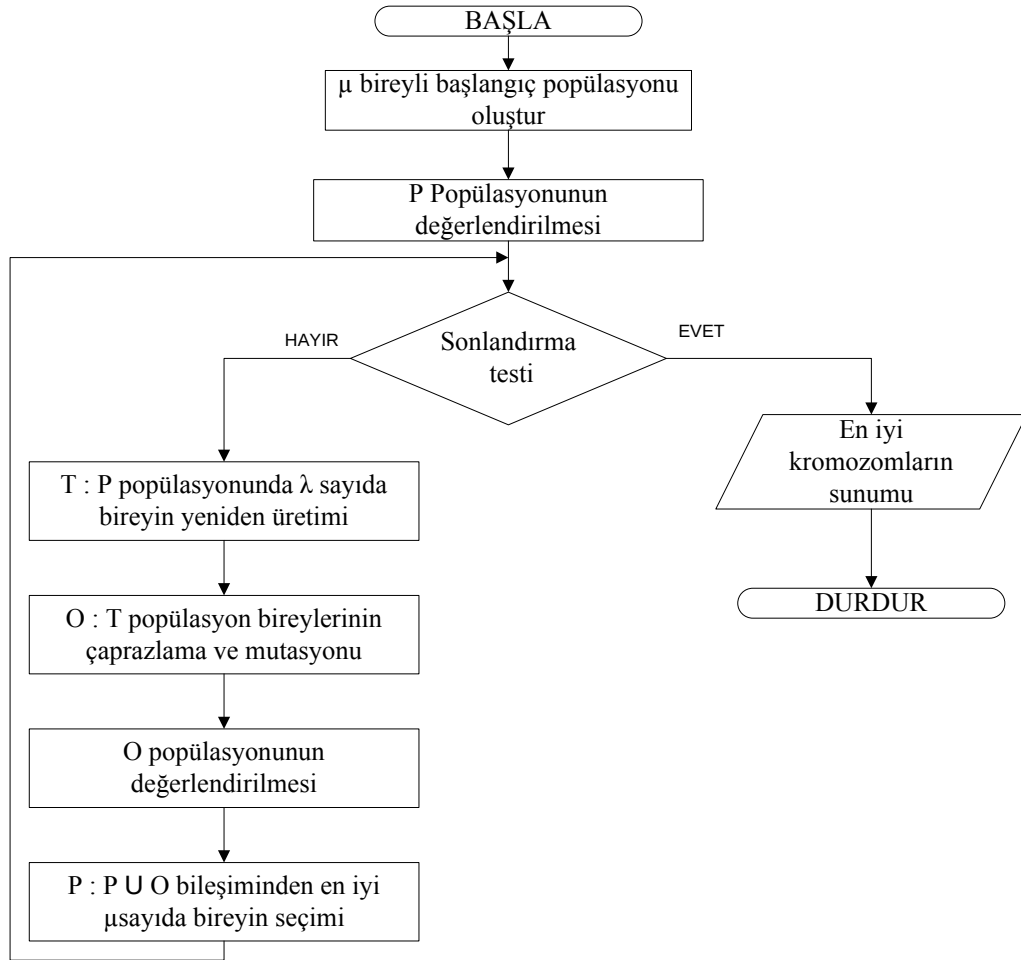
$$\eta'_k(j) = \eta_i(j) \exp(\tau' N(0,1) + \tau N_j(0,1)) \text{ ve}$$

$$x'_k(j) = x_i(j) + \eta'_i(j) N_j(0,1)$$

Burada $N(0,1)$ ortalaması 0 ve standart sapması 1 olarak normal dağılım gösteren tek boyutlu tesadüfi bir sayıyı göstermektedir. τ ve τ' faktörleri ise yaygın olarak sırasıyla $(\sqrt{2\sqrt{n}})^{-1}$ ve $(\sqrt{2n})^{-1}$ değerlerini almaktadır.

4. Her bir yeni bireyin (x'_i, η'_i) uygunluğu hesaplanır.
5. Bireyler (x_i, η_i) veya (x'_i, η'_i) uygunluk değerine göre sıralanır ve en iyi μ sayıda birey bir sonraki nesilde ebeveyn olmak üzere seçilir.
6. Sonlandırma koşulları sağlanıncaya kadar önceki adımlar tekrarlanır.

Evrimsel stratejide mutasyon safi tesadüfi proses olarak modellenmediği için yeni birey ebeveyninden tamamen bağımsızdır. Popülasyonun da doğru sonuca ulaşılabilmesi için geniş olarak tutulması gerekmektedir. Çözülmesi düşünülen problem hakkında çok az bilgiye gereksinim duyduğu için evrimsel stratejiler optimizasyon konusundaki hemen her çeşitteki problemlere uygulanabilmektedirler. Amaç fonksiyonunun da kapalı biçimde olmasına da gerek bulunmamaktadır (Sumathi ve diğer., 2008, s;19).



Şekil 2.2. Evrimsel strateji algoritma işleyişi ($\mu+\lambda$)

Evrimsel hesaplama teknikleri ile ilgili uygulamalar son yirmi yıl içerisinde birbirine benzeme yönünde gelişmeler göstermiştir. Yapılan çalışmalarda her bir teknik daha iyi sonuç vermeyi sağlayacak biçimde diğerinden ona özgü bazı elementleri ödünç almakta ve giderek birbirine benzemektedir. Bununla birlikte bu çatı altında değerlendirilen Genetik algoritmalar ve Evrimsel stratejiler karşılaştırıldığında aşağıdaki hususlar öne çıkmaktadır (Michalewicz, 1996, s;164, Spall, 2003, s;260, Rutkowski, 2005, s;267):

- Evrimsel stratejiler bir nümerik optimizasyon metodu olarak geliştirilmiştir. Genetik algoritmalar ise genel amaçlı bir adaptif arama tekniği olarak formüle edilmiştir.

- Her iki sistem de işleyişi içerisinde bir potansiyel çözüm popülasyonu sürdürür ve daha uygun bireylerin hayatta kalma prensibine göre seçim yapar.
- Evrimsel stratejiler yüzen nokta vektörlerle (floating point vector) işlem yaparken genetik algoritmalarda ikili vektör (binary vector) temelinde çalışır.
- Evrimsel stratejilerde seçim süreci deterministiktir ve bireyler tekrar seçilmez. Genetik algoritmalarda ise seçim süreci tesadüfidir. Güçlü bireyler bir sonraki nesil için tekrar seçilebilirken en zayıf bireylerin de seçilme olasılığı mevcuttur.
- Çaprazlama olasılığı ve mutasyon olasılığı gibi yeniden üretim (reproduction) parametreleri genetik algoritmalarda evrim prosesi süresinde sabit kalır. Ancak, evrimsel stratejilerde bu değerler her defasında değiştirilir. Kontrol parametrelerinin kendi kendine adaptasyonu evrimsel stratejilerde sistemin ince ayar (fine tuning) yapmasını sağlamaktadır.

2.7. Evrimsel Programlama

İlk olarak 1960 yılında Lawrence J. Fogel tarafından tasarlanan evrimsel programlama, genetik algoritmalara benzer bir stokastik optimizasyon stratejisidir. Bireylerin genetik bilgilerinin evriminden ziyade onların adaptasyonuna odaklanır ve bireylerin genlerini değil direkt olarak davranışlarını modifiye eder. Evrimsel programlama tamamen bağımsız olarak geliştirilmesine rağmen evrim stratejilerine de benzerdir (Olariu ve Zomaya, 2006, s;6, Sumathi ve diğer., 2008, s;16). Evrimsel programlama da en uygun olan bireyin seçimine dayanmaktadır ancak bireyleri dönüştürme işlemlerinde yalnızca mutasyon kullanmaktadır.

Sonlu durum makineleri ve makine zekası (machine intelligence) üzerinde çalışan Fogel ve arkadaşları genetik algoritmalar tarafından çözümü zor olan bu tip problemlerin çözümünü evrimsel programlama olarak adlandırdıkları metotla gerçekleştirmişler ve bu yaklaşımın genetik algoritmalar gibi aşağıdan yukarı değil yukarıdan aşağıya bir adaptif davranış prosesi olarak tanımlamışlardır (Kennedy ve diğer., 2001, s;139). Evrimsel programlamaya zamanla ‘kendi kendine adaptasyon’

özelliği eklenmiştir. Bu özellik sayesinde mutasyonu yönlendiren strateji parametrelerin de dönüşmesi sağlanmıştır.

Evrimsel programlama uygulama adımları aşağıdaki gibi sıralanabilir (Sarker ve diğer., 2003, s;33):

1. μ bireyden oluşan bir başlangıç popülasyonu oluşturulur. Her bir birey ikili gerçek değerli vektörden oluşturulur (x_i, η_i) . Burada ilk değer amaç değişkenini ikincisi ise gaussian mutasyon için standart sapmayı göstermektedir.
2. Her bir bireyin uygunluk değeri hesaplanır.
3. Her bir ebeveyn (x_i, η_i) , yeni bir birey (x'_i, η'_i) oluşturur. Bunun için aşağıdaki formül kullanılmaktadır:

$$\eta'_i(j) = \eta_i(j) \exp(\tau' N(0,1) + \tau N_j(0,1)) \text{ ve}$$

$$x'_i(j) = x_i(j) + \eta'_i(j) N_j(0,1)$$

Burada $N(0,1)$ ortalaması 0 ve standart sapması 1 olarak normal dağılım gösteren tek boyutlu tesadüfi bir sayıyı göstermektedir. τ ve τ' faktörleri ise yaygın olarak sırasıyla $(\sqrt{2\sqrt{n}})^{-1}$ ve $(\sqrt{2n})^{-1}$ değerlerini almaktadır.

4. Her bir yeni bireyin (x'_i, η'_i) uygunluğu hesaplanır.
5. Ebeveyn ve onlardan oluşturulan yeni bireyler çiftler olarak sıralanır ve uygunluk değerleri karşılaştırılır. Tesadüfi olarak q adet çift seçilir. Seçilen bu çiftlerde uygunluk değeri büyük olanlar bir sonraki nesil için seçilir.
6. Sonlandırma koşulları sağlanıncaya kadar önceki adımlar tekrarlanır.

Optimizasyon yüzeyinin veya uygunluk alanının çok engebeli, düzensiz olduğu ve çok sayıda yerel optimal çözüme sahip kombinasyonlu (combinatory) ve gerçek değerli fonksiyon optimizasyon problemleri evrimsel programlamaya daha uygundur. Genetik algoritmalarda olduğu gibi evrimsel programlamada da uygunluk alanının değişkenler ile karakterize edilebildiği ve bu değişkenlerin bir optimum değeri ya da değerleri olduğu varsayımı söz konusudur.

Temel bir evrimsel programlama metodu şu üç aşamayı kapsar (Sumathi ve diğer., 2008, s;16-17):

1- Geçici çözümlerden tesadüfi olarak bir başlangıç popülasyonu seç. Popülasyondaki çözüm sayısı optimizasyon hızıyla çok yakın ilişkilidir, ancak kaç çözümün uygun olduğu veya kaç çözümün gereksiz olduğuyla ilgili kesin bir cevap mevcut değildir.

2- Her bir çözüm yeni bir popülasyona kopyalanır/aktarılır. Bu, yeni birey (çocuk, döl) çözümlerin her biri en küçükten en sona kadar sürekli bir mutasyon tipiyle değişim göstererek mutasyon tipinin dağılımına göre mutasyona uğrar. Mutasyonun büyüklüğüne aileye yüklenen fonksiyonel değişimin temeline göre karar verilir.

3- Her bir yeni birey, uygunluğu hesaplanarak değerlendirilir. Tipik olarak çözüm popülasyonu için elde tutulacak N sayıdaki çözümün belirlenmesi için stokastik bir turnuva düzenlenir. Popülasyon büyüklüğünün sabit olması için bir gereksinim söz konusu değildir. Bunun yanında her bir ebeveynden yalnızca bir yeni birey oluşturulması da gerekmemektedir. Buna göre evrimsel programlama genetik operatör olarak çaprazlamayı kullanmaz.

Evrimsel programlamanın beş önemli farklılığı Fogel'e göre aşağıdaki gibi tanımlanabilir (Bäck, 1996, s;92):

- Standart evrimsel programlama, herhangi bir kendi kendine adaptasyon mekanizmasının yokluğu ile karakterize edilir.
- Sürekli standart evrimsel programlama, jenerasyon temelli çalışan mekanizmalardan farklı olarak yeni oluşturulan birey hemen değerlendirilir ve popülasyona dahil edilir.
- Meta-evrimsel programlama, farklılıkları kendi kendine adaptasyonlarına izin vermek için genotipe dahil eder.
- Sürekli meta-evrimsel programlama, yukarıdaki gibi yeni oluşturulan bireyler değerlendirilir ve popülasyona dahil edilir.
- Rmeta-evrimsel programlama, standart sapmalara ilaveten kendi kendine adaptasyon için kovaryansları da genotipe dahil eder.

Evrimsel stratejiler ve evrimsel programlama teknikleri yakın zamanlarda farklı kişiler tarafında birbirinden habersiz olarak geliştirilmiş olmalarına rağmen birçok benzerlikler taşımaktadır. Her iki teknik de yüzen nokta sunumu (floating point representation) kullanmaktadır ve mutasyon her iki teknik için de anahtar operatördür.

Bununla birlikte aralarındaki farklar ise ařağıdaki gibi sıralanabilir (Michalewicz, 1996, s;167, Spall, 2003, s;260, Sarker ve diğeri., 2003, s;32):

- Evrimsel programlama herhangi bir yeniden birleřtirme operatörü kullanmamaktadır.
- Evrimsel programlama tesadüfi seçim uygularken evrimsel strateji bir sonraki seçim için en iyi bireyi seçer ve kullanır.
- Evrimsel programlamada uygunluk deęerleri amaç fonksiyon deęerleri ölçülerek hesaplanmaktadır.
- Evrimsel programlamada her bir bireyin mutasyonundaki standart sapma bireyin kendi uygunluk deęerinin doğrusal dönüşümünün karekökü olarak hesaplanır.

ÜÇÜNCÜ BÖLÜM

GENETİK ALGORİTMALAR

Algoritma, problem çözmek için kullanılan adımlar serisidir. Evrimsel programlamanın (evolutionary computing) bir parçası ve bu alandaki en popüler teknik olan Genetik Algoritmalar (GA) biyolojideki evrim teorisinden esinlenilerek ortaya çıkmıştır. Doğada da popülasyondaki bireyler her biri kısıtlı olan yiyecek, barınak vb. faktörler için birbirleri ile rekabet etmek durumundadır. Bununla birlikte aynı türler içerisinde bireyler üreme için eş seçiminde birbirleri ile de yarışmak durumundadır. Bu seleksiyon nedeni ile zayıf performans gösteren bireyler yaşamlarını sürdürmek için daha az şansa sahiptir ve daha iyi adapte olmuş veya uygun bireyler nispeten daha fazla sayıda döl ortaya koymaktadır (Sivanandam ve Deepa, 2008, s;31). Üreme esnasında her bir ebeveynin iyi niteliklerinin birleşimi, uygunluğu atasından daha fazla olan yeni bir birey meydana getirebilir. Doğal evrim çevrelerine iyi adapte olmuş bireyler ortaya koyar. Benzer bir çevrenin hesaplama yapısını kontrol ederek genetik algoritmalar kromozomları (yapıları) çözümü istenen probleme daha uygun biçime dönüştürebilmektedir (Pelikan ve diğer., 2006, s;40).

3.1. Genetik Algoritmaların Gelişimi

Genetik algoritmalar da ilk olarak 1960’larda Holland tarafından bulunmuş ve daha sonra 60 ve 70’li yıllarda Michigan Üniversitesindeki çalışma arkadaşları ve öğrencileri ile birlikte geliştirilmiştir. Evrimsel strateji ve evrimsel programlamadan farklı olarak Holland’ın ilk amacı spesifik bir problemin çözümü için algoritma ortaya koymaktan ziyade doğada olduğu gibi bir adaptasyonu biçimsel olarak çalışmak ve doğal adaptasyon mekanizmalarının bilgisayar sistemlerine uygulanabileceği metotlar geliştirmektir.

Genetik algoritmaların arama mantığı Holland tarafından formüle edilen şema teoremi üzerine kurulmuştur. Şema, dizilerdeki tanımlanmış belirli kısımların benzerliklerini gösteren bir alt diziyi tanımlayan sabit bir şablondur. Benzer şema içeren diziler bir dereceye kadar benzer bilgi içermektedir (Coley, 1999, s;3). Şema, 0,1 ve * sembollerinden oluşan bir dizidir. Şema teoreminin temel karakteristiğini yansıtan ‘*’ sembolü, uygulamada 0 yada 1 değeri alabilir. Burada ‘*’, belirli uzunluktaki diziler

arasındaki muhtemel tüm benzerlikleri göstermeye imkan veren bir notasyondur. Teorem, en fazla uygun olan şema içeren dizinin bir sonraki toplumda da hayatını devam ettireceği mantığına dayalıdır. Genetik algoritma çözümü en iyi şemanın tatbiki ile en iyi kromozoma doğru yaklaşır (Banzhaf ve diğer., 1998, s;96-97, Şen, 2002, s;99, Spall, 2003, s;263).

Genetik Algoritmalar doğal seçim ve doğal genetik mekanizmalarına dayalı stokastik araştırma teknikleri olarak ortaya atılmıştır (Sakawa, 2002, s;1). 1975 yılında yayınlanan “Doğal ve Yapay Sistemlerde Adaptasyon” (Adaptation in Natural and Artificial Systems) adlı kitabında Holland ilk olarak doğal evrimin optimizasyon problemlerine nasıl uygulanacağını anlatmış ve ilk genetik algoritmayı oluşturmuştur. Holland’ın GA’sı kromozomların bir popülasyonundan yeni bir popülasyona çaprazlama, mutasyon ve inversiyonun genetikten esinlenmiş operatörleri ile bir çeşit ‘doğal seleksiyon’ kullanılarak geçişi sağlayan bir metottur (Mitchell, 1999, s;2). Orijinal genetik algoritmanın iki ana karakteristiği sabit uzunluktaki kromozom sunumu ve oldukça yoğun kullanılan çaprazlamadır (Banzhaf ve diğer., 1998, s;95).

Genetik Algoritmaların popüler kılınması ise Holland’ın doktora öğrencisi olan David Goldberg’in 1989 yılında yayınladığı “Araştırma Optimizasyon ve Makine Öğrenmesinde Genetik Optimizasyon (Genetic Algorithms in Search, Optimization, and Machine Learning) adlı kitapla olmuştur. Goldberg çalışmasında teorik çalışmaları başarılı uygulamalara taşıyarak konuya büyük bir katkı sağlamıştır (İşçi ve Korukoğlu, 2003, s;191, Haupt ve Haupt, 2004, s;22).

Genetik algoritmaların bu alanda çalışan herkes tarafından kabul görmüş kesin bir tanımı olmamasına rağmen, genetik algoritmaları bazı bileşenlerinin diğer evrimsel algoritmalarından ayırdığı ifade edilmiştir. Bu elementler kromozom popülasyonu, uygunluğa göre seçim, yeni bir döl elde etmek için çaprazlama ve rastgele mutasyon olarak sıralanabilir (Mitchell, 1999, s;7).

Genel bir tanımla Genetik Algoritmalar, en iyinin korunumu ve doğal seçim ilkesinin benzetim yoluyla bilgisayarlara uygulanması ile elde edilen bir arama yöntemidir (Nabiyev, 2003, s;606). Doğal genetik kalıtım davranışlarını teorik ve yapay olarak gerçekleştiren stokastik bir tekniktir (Şen, 2004, s;65). Genetik Algoritmalar çok sayıda bireyden oluşan bir popülasyonun tanımlı seçim kuralları

altında “uygunluğu”nu (fitness) maksimize eden bir duruma dönüşümüne imkan verir (Haupt ve Haupt, 2004, s;22).

Genetik algoritmalar stokastik algoritmalar; olasılık genetik algoritmalar da temel bir role sahiptir. Genetik algoritmalar daima bir çözüm popülasyonunu dikkate alır. Algoritma farklı çözümleri bir araya getirerek daha iyilerini araştırır ve popülasyondaki çeşitlilikten yarar sağlar. Genetik algoritmaları uygulamadan önce problemle ilgili özel bir gereksinimi yoktur ve dolayısıyla herhangi bir probleme uygulanabilir.

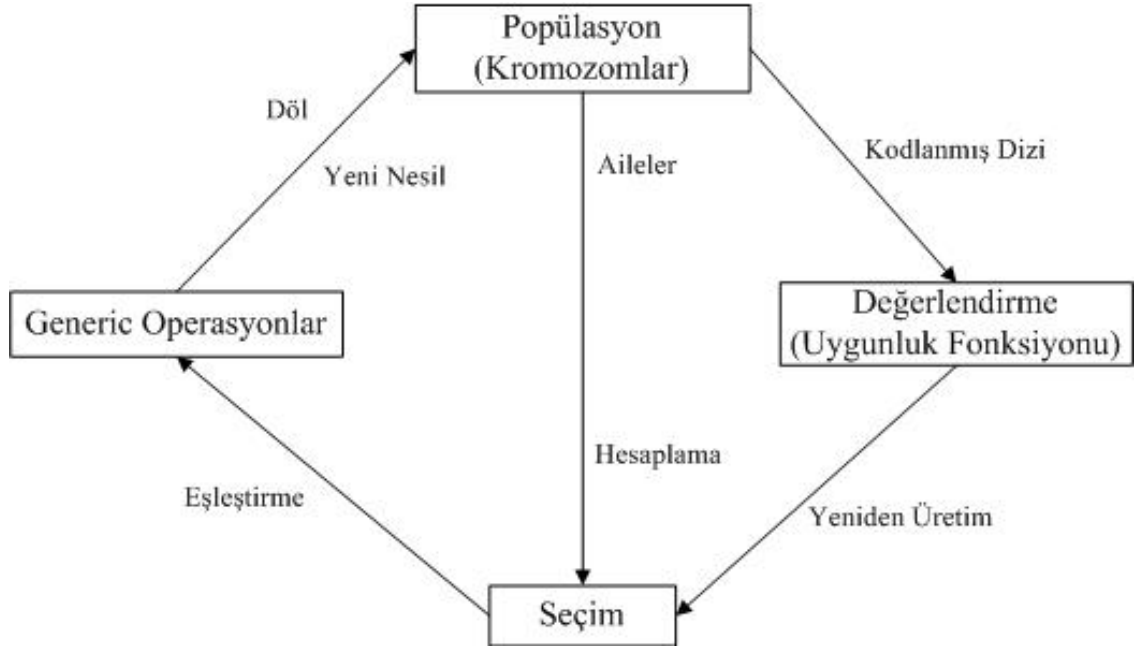
Çoğu stokastik model gibi Genetik algoritmalar da global optimal çözümü garanti etmez. Probleme kabul edilebilir derecede iyi bir çözüm hedefler (Sivanandam ve Deepa, 2008, s;20). Genetik algoritmalar özellikle çözüm uzayının geniş, süreksiz ve karmaşık olduğu problem tiplerinde başarılı sonuçlar vermektedir (Emel ve Taşkın, 2002, s;139).

Belirli bir problem çözümüne yönelik hazırlanan bir genetik algoritma aşağıdaki beş bileşeni kapsamalıdır (Michalewicz, 1996, s;17, Gen ve Cheng, 2000, s;1):

1. Problemin potansiyel çözümü için bir genetik gösterim,
2. Potansiyel çözümün başlangıç popülasyonunu oluşturmak için bir yol,
3. Çözümleri uygunluklarına göre sıralayan bir değerlendirme fonksiyonu,
4. Elde edilen çocukların bileşimini değiştiren genetik operatörler
5. Genetik algoritma tarafından kullanılan (popülasyon büyüklüğü, genetik operatörlerin uygulama olasılıkları gibi) çeşitli parametre değerleri.

3.1. Genetik Algoritmaların Çalışma Esasları

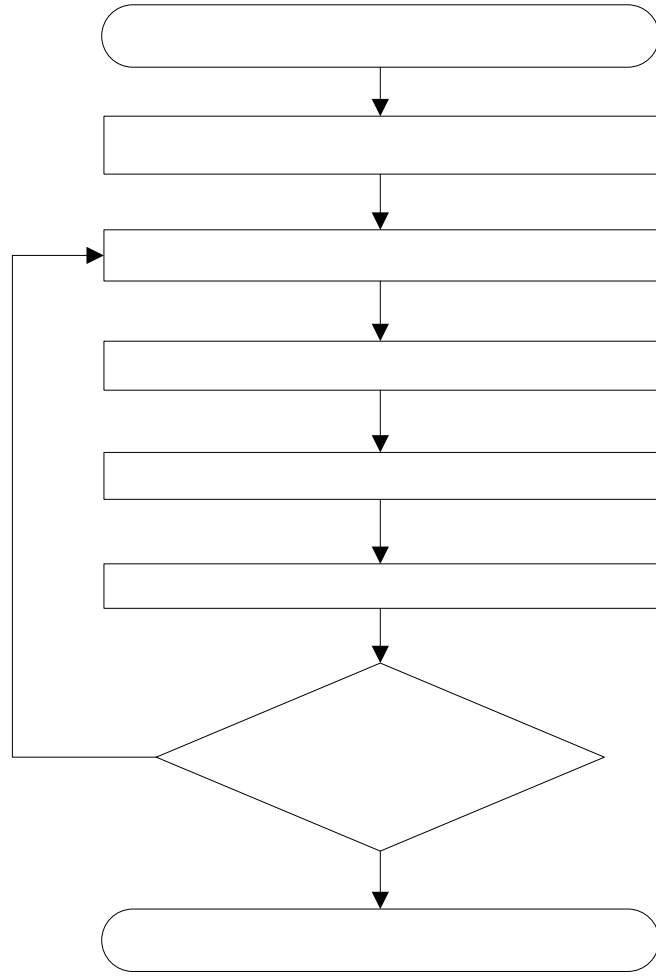
Genetik algoritmalar rastgele seçilen bireylerden oluşan ilk popülasyon ile başlar. Popülasyondaki her bir birey çözülmesi düşünülen problem için potansiyel bir çözüm sunar. Bu bireyler başarılı tekrarlamalar ile (iterations) yavaş yavaş gelişir. Tekrarlamalar ile yeni nesiller ortaya çıkar. Her bir nesilde popülasyondaki her bir birey bazı uygunluk ölçütleri kullanılarak değerlendirilir. Daha sonra da genetik operatörler vasıtasıyla bir sonraki nesil popülasyonu oluşturulur. Bu prosedür sonlandırma şartları sağlanıncaya kadar devam eder (Sakawa, 2002, s;11). Genetik algoritma prosesindeki bu döngü aşağıdaki Şekil 3.1’de gösterilmiştir.



Şekil 3.1. Genetik algoritma döngüsü (Sivanandam ve Deepa, 2008, s;31)

Seçimin (selection) toplumdaki her bireyi karşılaştırabildiği düşünülür. Seçim işlemi uygunluk fonksiyonu kullanılarak yapılır ve her kromozom temsil ettiği çözümün uygunluğunu gösteren bir değere sahiptir. Optimal çözüm uygunluk fonksiyonunu maksimize eden çözümdür. Seçilen kromozomlara genetik operatörlerin uygulanması ile birlikte yeni bireyler elde edilir ve yeni bir popülasyon oluşturulur. Bu döngü sonlandırma koşulları sağlanıncaya kadar sürer (Rutkowski, 2005, s; 271)

Genetik Algoritmaların temel prosedürleri ve işleyişi ise aşağıda Şekil 3.2’de gösterilmektedir (Sakawa, 2002, s;14).



Şekil 3.2. Genetik algoritma işleyişi

Standart bir “Genetik Algoritma İşleyişi” aşağıdaki gibi verilebilir (Jain ve Martin, 1998, s;132, Karr ve Freeman, 1999, s;4, Nabiyev, 2003, s;607, Güden ve diğer., 2005, s;7, Sivanandam ve Deepa, 2008, s;30, Sumathi ve diğer., 2008, s;83):

1. Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Bu çözüm grubu, biyolojideki benzerliği nedeniyle, toplum (population), çözümlerin kodları da kromozom olarak adlandırılır. Toplumda bulunacak birey sayısı için bir standart yoktur. İlk popülasyon genetik materyallerin çeşitliliğini mümkün olduğu kadar geniş yansıtmalıdır. Tipik popülasyon büyüklüğü birkaç düzineden binlere kadar değişebilir. Genel olarak önerilen 100-300 aralığında bir büyüklüktür. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Toplum bu işlemde sonra rastgele oluşturulur.

2. Toplumdaki her kromozomun ne kadar iyi olduđu bulunur. Bu amaçla kullanılan fonksiyona uygunluk fonksiyonu denir. Uygunluk fonksiyonu genetik algoritmanın beynini oluřturmaktadır. Genetik algoritmada probleme özel çalışan tek kısım bu fonksiyondur. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirerek onların bir bakıma şifresini çözmekte (decoding) ve sonra bu parametrelere göre hesaplamayı yaparak kromozomların uygunluđunu bulmaktadır. Çođu zaman genetik algoritmanın başarısı bu fonksiyonun verimli ve hassas olmasına bađlıdır.

3. Seçilen kromozomları eşleyerek yeniden kopyalama ve deđiřtirme operatörleri uygulanır. Sonuçta yeni bir toplum oluřturulur. Bu eşleme kromozomların uygunluk deđerlerine göre yapılır. Bu seçimi yapmak için rulet tekerleđi seçimi (roulette wheel selection), turnuva seçimi (tournament selection) gibi farklı seçme yöntemleri kullanılır. Yeniden kopyalama (recombination) genlerdeki genetik bilginin birinden diđerine geçmesi işlemine benzediđi için çaprazlama (crossover) olarak adlandırılır. Bu işlem, toplumda çeřitliliđi sađlar. Mutasyonun etkisi, yalnızca tek bir çözüm üzerinde olmaktadır.

4. Yeni kromozomlara yer açmak için eski kromozomlar çıkartılarak sabit büyüklükte bir toplum sađlanır.

5. Tüm kromozomların uygunlukları tekrar hesaplanır ve yeni toplumun başarısı bulunur.

6. İşlemler tekrarlanarak verilmiř zaman içerisinde daha iyi olan yeni nesillerin oluřturulması gerçekleştirilir.

7. Sonuçta toplumların hesaplanması sırasında en iyi bireyler bulunduđunda çözüm elde edilmiř olur. Popölasyon optimum çözüme yakınsadıđında algoritma durdurulur.

Yukarıda sıralanan işleyiř kapsamında genetik algoritma en az üç aşamada olasılıđa dayalı adımlar içermektedir (Kinnear, 1994, s;22). Bu adımlar:

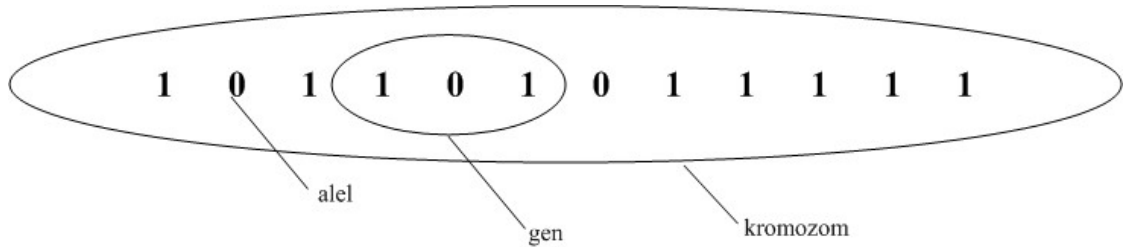
- bařlangıç popölasyonunun oluřturulması
- genetik operasyonlar için popölasyondan bireyler seçilmesi
- çaprazlama veya mutaston gibi genetik operatörleri uygulamak için seçilen birey üzerinde bir nokta belirlenmesidir.

3.2. Genetik Algoritma Terminolojisi

Genetik algoritmalar terminolojisinde biyolojiye ait terimler gerçek anlamları daha basite indirgenerek sıklıkla kullanılmaktadır. Bu açıdan bakıldığında kullanılan terminoloji hakkında bilgi verilmesinin yararlı olacağı düşünülmüştür.

Tüm canlı organizmalar hücrelerden oluşmaktadır ve her bir hücre organizmanın bir ‘plan’ını (blueprint) içeren DNA dizilerinden oluşan aynı kromozom setinin bir veya daha fazlasını taşır. Kromozomlar genlerin dizisidir. Bir kromozom kavramsal olarak genlere bölünebilir. Her bir gen farklı bir proteini kodlar. Çok genel biçimde bir genin göz rengi gibi herhangi bir özelliği kodladığı söylenebilir. Bir özellik için muhtemel farklı niteliklerin her biri (örneğin göz için mavi renk) ‘alel’ olarak adlandırılır. Bir bit dizisinde alel ise 0 veya 1’dir. Her bir gen de kromozom üzerinde ‘lokus’ adı verilen belirli bir yerde konumlanmıştır (Mitchell, 1999, s;5).

Organizmanın sahip olduğu genlerin ve dolayısıyla kromozomların tamamına ‘genom’ denilmektedir. Belirli bir bireyin genlerinin tüm bileşimine ise ‘genotip’ (genotype) denir (Haupt ve Haupt, 2004, s;22). Genotip, başka bir deyişle bir genom içerisindeki belirli bir gen setini ifade eder. Organizmanın çevresel ve genetiksel olarak belirlenen özelliklerine ‘fenotip’ (phenotype) adı verilir. Genel olarak bu özellikler saç rengi, boy veya zeka gibi gözlemlenebilir özelliklerdir (Mitchell, 1999, s;5). Kromozom, gen ve alel ilişkisi aşağıdaki şekilde gösterilmiştir (Rothlauf, 2006, s;11):



Şekil 3.3. Alel, Gen ve Kromozomlar

Genetik algoritmalarda kromozom terimi genelde bir bit dizisi olarak kodlanır ve probleme aday bir çözüme göndermede bulunur. Her bir kromozom yegane bir çözümü tanımlamalıdır, ancak bu durum her bir çözümün belirli bir kromozom tarafından kodlandığı anlamına gelmez. Gerçekte ise tüm çözüm alanının taranabilmesi için problemin tüm aday çözümlerinin en azından muhtemel bir kromozoma uyması gerekir. Her bir genotipi onun fenotipi ile ilişkilendiren fonksiyona morfogenez fonksiyonu

iyi sonuç verirken bir diğeri için kötü ise ne yapılmalıdır? Ancak burada sorun, problemi çözmek için genetik algoritmanın nasıl uygulanacağından ziyade ‘daha iyi’ çözümün tanımından kaynaklanmaktadır. Eğer, farklı kriterlerin basit bir kombinasyonundan elde edilen bir uygunluk fonksiyonu iyi sonuçlar verebiliyorsa bu kriterlerin birbirine uygun bir araya getirildiği düşünülür (Sivanandam ve Deepa, 2008, s;41).

3.2.2. Popülasyon

Popülasyon, test edilen bireyler, bireyleri tanımlayan fenotip parametreleri ve araştırma alanı ile ilgili bazı bilgilerden oluşur. Genetik algoritmalarda kullanılan popülasyonun iki önemli yönü vardır:

1. İlk popülasyon oluşumu
2. Popülasyonun büyüklüğü

Tüm problemler için popülasyonun büyüklüğü problemin karmaşıklığına bağlıdır. İdeal olanı, tüm araştırma alanının araştırılabilmesi için ilk popülasyonun gen havuzunun mümkün olduğu kadar geniş olması gerekmektedir. Popülasyon çeşitliliği yansıtmaz ise algoritma dar bir alanı tarayacak ve muhtemelen global optimuma ulaşamayacaktır. Bundan kaçınmak için de başlangıç popülasyonu birçok durumda random olarak belirlenir. Bazı durumlarda yine de başlangıç popülasyonunu belirlemek için bir çeşit sezgisellik kullanılabilir. Böylece bu popülasyonun ortalama uygunluğu yüksek olur ve çözüme daha hızlı ulaşılabilir.

Seçilen popülasyon büyüklüğü de bazı problemler doğurabilmektedir. Daha geniş popülasyon araştırma alanının daha kolay taranması demektir. Ancak yakınsamak için genetik algoritma için gerekli zaman da popülasyon büyüklüğüne bağlı bir fonksiyonla belirlenir. Tüm bireylerin birbirine çok benzer ve daha ileri iyileştirmelerin yalnızca mutasyon ile mümkün olduğu durumda popülasyon yakınsamış demektir. Goldberg’de çalışmalarında genetik algoritmanın global optimuma ulaşma etkinliğinin büyük oranda popülasyon büyüklüğüne bağlı olduğunu göstermiştir. Sonuç olarak geniş popülasyon oldukça yararlıdır ancak çok daha fazla hesaplama maliyeti, hafıza ve zaman gerektirir. Pratikte genelde 100 civarında bir popülasyon büyüklüğü tercih edilmektedir ancak optimal popülasyon büyüklüğü büyük ölçüde seçilen problemin yapısına bağlıdır (Kennedy ve diğer., 2001, s;153, Sivanandam ve Deepa, 2008, s;42).

3.2.3. Veri Yapıları

Genetik algoritmalarındaki ana veri yapıları kromozomlar, fenotipler, amaç fonksiyon değerleri ve uygunluk değerleridir. Kromozom popülasyonunun tamamı, birey sayısı ve bireylerin genotiplerinin uzunlukları verilerek tek bir dizi içerisinde saklanabilir. Benzer biçimde bazı eşleştirmeler ile elde edilen dizayn değişkenleri veya fenotipler de tek bir dizi içinde saklanabilir. Gerçek eşleme kullanılan tersine kodlama/şifre çözme şemasına bağlıdır. Amaç fonksiyon değerleri skaler veya vektörel olabilir. Bu durum uygunluk değerleri için de aynı olmak durumundadır. Uygunluk değerleri, amaç fonksiyonundan ölçekleme (scaling) veya sıralama (ranking) fonksiyonları kullanılarak elde edilebilir ve vektör olarak saklanabilir (Sivanandam ve Deepa, 2008, s;43).

3.2.4. Arama Uzayı (Search Space)

Optimizasyon problemlerinde genellikle belirli bir çözüm seti arasından en iyi çözüm araştırılır. Tüm mümkün çözümlerin alanına arama uzayı adı verilir. Arama uzayındaki tüm noktalar olası bir çözümü temsil eder (Sivanandam ve Deepa, 2008, s;20). Bundan dolayı her muhtemel çözüm, problemin tanımına bağlı olarak onun uygunluk değeri ile tanımlanır. Uygunluk değeri arama uzayındaki ilgili noktanın niteliği ile ilintilidir (Jain ve Martin, 1998, s;131). Çözümler arasındaki farklar ölçülebildiği durumda arama uzayı kavramı geçerlidir ve her bir çözüme bir başarı yada uygunluk ölçütü tanımlanabilir (Coley, 1999, s;3).

3.2.4. Arama Stratejileri

Arama prosesi ilk popülasyonun belirlenmesi ve sonrasında sonlandırma koşulları sağlanıncaya kadar yeni bireylerin üremesini kapsar. Prosesin farklı amaçları olabilir. Global optimum noktanın bulunması bu amaçlardan biridir. Ancak genetik algoritmaların kullandığı tipteki modeller ile bu asla garanti edilemez. Bir sonraki iterasyonun daha iyi bir çözüm ortaya koyma olasılığı daima vardır.

Aramanın diğer bir amacı daha hızlı yakınsama sağlamaktır. Amaç fonksiyonunu çalıştırmanın zor olduğu durumlarda daha hızlı yakınsama istenebilir

ancak bu durumda da lokal bir yakınsama olasılığının ve düşük değerli bir optimal bulma ihtimali artmaktadır.

Aramanın başka bir amacı da iyi çözümden uzaklaşmadan çeşitliliği artırmaktır. Çözüm uzayının çok sayıda farklı optimum içerdiği durumda, ki bu durum uygunluk için de aynıdır, onların arasından seçim yapabilmek yararlıdır. Çünkü modeldeki faktör değerlerinin bazı kombinasyonları diğerlerinden daha uygun sonuçlar ortaya koyabilmektedir (Sivanandam ve Deepa, 2008, s;43).

3.3. Kodlama

Genetik algoritmaları diğer arama yöntemlerinden ayıran en önemli özelliklerden biri parametrelerin kendileri yerine onları temsi eden serileri kullanmasıdır. Kodlama ayrı ayrı genleri simgeleme prosesidir. Bu proses bitleri, sayıları, ağaçları, listeleri veya başka diğer objeleri kullanarak gerçekleştirilir. Kodlama esasen problemin çözümüne bağlıdır. Genetik algoritma uygulamalarının başarısı bir ölçüde de kodlamanın iyi yapılmasına bağlıdır (Dengiz ve Altıparmak, 1998, s;4, Banzhaf ve diğer., 1998, s;95). Kodlama şemaları bir kodlamadan diğerine büyük ölçüde değişebilir. Hatta bazı durumlarda bir dereceye kadar bu şemalar da evrime uğrayabilmektedir. İlk uygulamalarda kodlama düzeni ardışık ikili alt dizilerden oluşan bir dizisi kromozom oluşturmak için kullanılmıştır. Son dönemde ise farklı çözüm parametreleri değişen noktalı kromozomlarla temsil edilmektedir. Bu durum birçok bilgisayar dilinde kullanılan yapılarla benzerliğinden dolayı programlamada birtakım avantajlar sağlamaktadır (Karr ve Freeman, 1999, s;2).

Literatürde geliştirilen bazı farklı kodlama biçimleri aşağıda sıralanmıştır:

3.3.1. İkili Kodlama

Genetik algoritma uygulamalarında amaç değerlerini kodlamada kullanılan en yaygın kodlama biçimidir. İkili kodlamada her bir kromozom ikili (binary) $\{0,1\}$ bir bit dizisi olarak kodlanır. Dizideki her bir bit, çözüme ait bazı karakteristikleri ortaya koyar. Her bir değişken uzunluğu kullanıcı tarafından belirlenen l_i uzunluğunda bir dizi olarak kodlanır. Daha sonra da bu alt diziler peşi sıra birleştirilir. Her bir bit dizisi bir çözüm demektir ancak bunun optimal çözüm olması gerekmez. Bit dizisinin kodlanma yolu probleminden probleme değişebilmektedir.

Dizinin uzunluğu hassaslığa bağlıdır. Yüksek hassaslık gerektiren değişkenler daha geniş bir dizide kodlanır (Bäck ve diğer., 2000b, s;5). Çözüm için gerekli zaman ve çözümlerin sağlığı değişkenlerin temsili için kullanılacak hanelerin veya genlerin büyüklüklerine de bağlıdır. Ondalık sayılarda her bir hane için dizi üzerinde ortalama 3,3 tane haneye gereksinim olduğu düşünülürse 8 haneli bir hassasiyette 10 değişken için kromozom uzunluğu $8 \times 3,3 \times 10 \cong 270$ haneden oluşur. Bu açıdan problemin özelliğine göre önceden hassasiyet belirlenmelidir (Haupt ve Haupt, 2004, s;107, Şen-2004, s;101).

İkili kodlama dizisi daha az sayıda alel ile çok sayıda muhtemel kromozom ortaya koyabilir. Diğer taraftan bu kodlama türü birçok problem için doğal değildir ve bazı durumlarda genetik operasyonlar bittikten sonra bazı düzeltmelerin yapılması gerekebilmektedir. İkili (binary) kodlama aşağıdaki biçimde gösterilir (Sivanandam ve Deepa, 2008, s;44):

Kromozom 1	1 1 0 1 0 0 0 1 1 0 1 0
Kromozom 2	0 1 1 1 1 1 1 1 1 1 0 0

Şekil 3.4. İkili Kodlama

Bu kodlamada

- Tam sayılar tam olarak (exactly) gösterilir.
- Sonlu sayıda reel sayı gösterilebilir.
- Gerçek sayıların numaralarının sunumu dizi uzunluğunu artırır. (Sivanandam ve Deepa, 2008, s;43).

İkili düzende kodlama çok sık kullanılan bir kodlama tipi olmasına rağmen bazı sakıncaları vardır. Örneğin, çok değişkenli bir fonksiyon optimizasyonu için değişkenlerin alt ve üst sınırlarına bağlı olarak elde edilen dizi çok uzun olabilir. Aynı zamanda gezgin satıcı, çizelgeleme, karesel atama gibi kombinatoriyal optimizasyon problemlerinde ikili düzende kodlama arama uzayını tam olarak temsil edememektedir (Dengiz ve Altıparmak, 1998, s;4)

3.3.2. Sekizli (octal) Kodlama

Sekize kadar olan sayılardan (0-7) oluşturulan dizileri kullanan kodlama türüdür. Kullanım biçimi aşağıda gösterilmiştir (Sivanandam ve Deepa, 2008, s;44).

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

Şekil 3.5. Sekizli (octal) Kodlama

3.3.3. Hekzadesimal Kodlama

Bu kodlama biçiminde de 10'a kadar olan rakam ve ilk altı harften (0-9, A-F) oluşturulan diziler kullanılır. Gösterim biçimi aşağıdaki gibidir (Sivanandam ve Deepa, 2008, s;44).

Kromozom 1	9 C E 7
Kromozom 2	3 D B A

Şekil 3.6. Hekzadesimal Kodlama

3.3.4. Permütasyon Kodlama (Gerçek Sayı Kodlama)

Permütasyon kodlamada diziler ard arda sıralanan rakamlardan oluşmaktadır. Bazı durumlarda genetik operasyonlar tamamlandıktan sonra bazı düzeltmelerin yapılması gerekmektedir. Permütasyon kodlamada her bir kromozom rakamların ard arda sıralandığı, tamsayıli gerçek değerlerin bir dizisidir. Gösterim biçimi aşağıdaki gibidir.

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

Şekil 3.7. Permütasyon Kodlama

Permütasyon kodlama yalnızca problemleri sıralamada yarar sağlar. Bu durumda bile kromozom uyumluluğundan kaçınmak için bazı çaprazlama ve mutasyon düzeltmelerinin yapılması zorunlu olmaktadır. (Sivanandam ve Deepa, 2008, s;45). Bu kodlama fonksiyon optimizasyon problemlerinin çözümünde ikili ve gri kodlamaya göre daha etkin olmaktadır (Gen ve Cheng, 2000, s;3).

3.3.5. Değer Kodlama

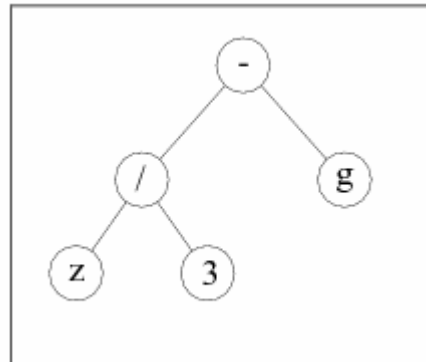
Değer kodlamada her bir kromozom bir değerler dizisinden oluşu ve bu değerler de problem ile bağlantılı rakamlar, reel sayılar veya probleme özgü bazı özel karakterler vb. herhangi bir biçim olabilir. Bazı özel problemler için bu kodlama türü en iyi sonuçları ortaya koymaktadır. Diğer açıdan bakıldığında ise sıklıkla probleme özgü genetik operatörler geliştirilmesini gerekli kılmaktadır. Bu kodlama reel sayılar gibi karmaşık verilerin kullanıldığı problemler için kullanılabilir. Bu tür problemlerde ikili kodlama kullanımı oldukça zordur. Gösterim biçimi aşağıdaki gibidir (Sivanandam ve Deepa, 2008, s;45).

Kromozom A	1.2324 5.3243 0.4556 2.3293 2.4545
Kromozom B	ABDJEIFJDHDIERJFDLDFLFEGT
Kromozom C	(geri), (geri), (sağ), (ileri), (sol)

Şekil 3.8. Değer Kodlama

3.3.6. Ağaç Kodlama

Bu kodlama biçimi çoğunlukla genetik programlama için program ifadelerini dönüştürmede kullanılır. Burada her bir kromozom programlama dilinin komutları veya fonksiyonları gibi bazı objelerden ağacı şeklindedir (Sivanandam ve Deepa-2008, s;45). Ağaç kodlamada her kromozom, adından da anlaşıldığı gibi nesneler ve nesneler arası işlemleri içeren bir ağaç yapısından oluşmaktadır. Ağaç kodlama, program geliştirmek için uygundur. Örneğin, LISP ve Prolog gibi programlama dillerinde ağaçtan yapısal olarak sık bir biçimde kullanılmaktadır. Ağaç kodlamada çaprazlama ve mutasyon çok kolay bir şekilde uygulanabilir. Gösterim biçimi aşağıdaki gibidir (Nabiyev, 2003, s;610).



Şekil 3.9. Ağaç Kodlama

3.3.7. Gri Kodlama

Gri kod dizisi de ikili 0 ve 1 karakterlerinden oluşmaktadır. Ancak amaç gri kodlamada değişkenlerini elde etmek için kullanılan kodlama ve tersine kodlama işleyişi farklıdır. Bir değişkenin gri kodlaması iki aşamalıdır: İlk olarak değişken ikili kodlama ile ifade edilir. Daha sonra bu ikili kod onun yerini tutacak olan gri koda dönüştürülür. $(b_1, b_2, b_3, \dots, b_l)$ $b_i \in \{0,1\}$ ikili dizisinin gri koda dönüştürülmüş biçimi $(a_1, a_2, a_3, \dots, a_l)$ aşağıdaki ifade ile sağlanır (Bäck ve diğer., 2000b, s;6):

$$a_i = \begin{cases} b_i & i = 1 \\ b_{i-1} \oplus b_i & dd \end{cases}$$

Burada $\oplus$ işlemi ikili tabanda toplama işlemini ifade etmektedir. Gri kodlamada dizideki bitişik tamsayılar 1 Hamming mesafesindeki ikili dizi kullanılarak ifade edilir. Hamming mesafesi, iki kromozomun karşılıklı hanelerindeki rakamları arasındaki farkı olarak tanımlanır.

Genetik operatörler kullanılarak elde edilen yeni neslin hedef değeri kodlamanın özelliğinden kaynaklanacak biçimde bazı durumlarda öncekinden oldukça uzaklaşabilmektedir. Bu durum da genetik algoritmanın arama yüzeyinde küçük bir mesafe için çok sayıda gereksiz iterasyon tekrarına neden olmaktadır. Gri kodlama kullanıldığında yeni neslin hedef değerinin önceki iyi nesilden çok da uzak olmaması sağlanmaktadır. Bundan dolayı Gri kodlama ile yapılan genetik algoritma işlemlerinde sonuca %10-20 arasında bir zaman tasarrufu ile daha hızlı ulaşılabilmektedir (Haupt ve Haupt, 2004, s;106, Şen, 2004, s;62, Spall, 2003, s;240).

3.4. Genetik Algoritma Operatörleri

Genetik algoritmalarda bir sonraki nesli elde etmek için üç ana genetik operatör kullanılır: Yeniden üretme (reproduction), çaprazlama (crossover) ve mutasyon (mutation) (Rao, 1996, s;810, Sakawa, 2002, s;13). Genetik algoritma uygulamalarında yapılan işlemlerin %90-95'i seçim ve çaprazlama operatörleri ile ilişkilidir. Mutasyon için genellikle düşük bir olasılık değeri tercih edilmektedir (Banzhaf ve diğer., 1998, s;95).

3.4.1. Yeniden Üretim/Seçim (reproduction)

Genetik algoritma uygulamalarında daha yüksek uygunluğa sahip bireylerin daha fazla çoğalacağı temeline dayalı doğal seleksiyon fikri yeniden üretim veya seleksiyon/seçim olarak adlandırılır. Yeniden üretim veya seçim operatörü yeni bir birey yaratmak için bireylerin popülasyondan nasıl seçileceği ve bunlardan hangi sayıda yeni birey oluşturulacağı konusuyla ilgilidir (Rao, 1996, s;810, Dengiz ve Altıparmak, 1998, s;4, Sakawa, 2002, sf.19).

Üreme prosesi genetik algoritmaların kalbidir. Bu süreçte arama prosesi ile yeni ve muhtemelen daha uygun bireyler ortaya konulmaktadır. Genetik algoritmalarda üreme prosesi üç aşamadan oluşur (Sivanandam ve Deepa, 2008, s;46):

1. Ailelerin seçimi,
2. Yeni bireyler (çocuk veya döl) oluşturulması için ailelerin çiftleştirilmesi ve dönüştürülmesi
3. Popülasyondaki eski bireylerin yenileri ile değiştirilmesi

Üreme prosesinin ilk adımı olan seçim, eşleştirme içim popülasyondan iki aile belirleme işlemidir. Kodlama biçimine karar verdikten sonraki aşama seçimin nasıl yapılacağını belirlenmesidir. Bu seçim, gelecek nesil için yeni bireyler oluşturacak bireylerin popülasyondan nasıl seçileceği; her birinden kaç döl elde edileceği gibi konuları kapsar. Seçimin amacı onlardan daha uygun dölleri elde etmek umudu ile popülasyondaki nispeten daha uygun bireylerin belirlenerek öne çıkarılmasıdır. Kromozomlar yeniden üretim için aile oluşturmak üzere başlangıç popülasyonundan seçilirler. Bu noktada problem kromozomların nasıl seçileceğidir (Bäck ve diğer., 2000a, s;56)

Seçim, değerlendirme fonksiyonu değerlerine göre popülasyondaki kromozomların seçilip ayrılması metodudur. (Sivanandam ve Deepa, 2008, s;46) Daha yüksek uygunluk değeri birey için daha fazla seçilme şansı anlamına gelmektedir. Seçim baskısı, daha iyi bireylerin seçildiği bir seviyeyi ifade eder. Daha yüksek seçim baskısında daha da iyi bireyler belirlenir. Seçim baskısı genetik algoritmayı daha iyi nesiller için popülasyon uygunluğunu artırmaya zorlar.

Genetik algoritmanın yaklaşma oranı büyük ölçüde seçim baskısının büyüklüğü tarafından belirlenir. Daha yüksek seçim baskısı daha yüksek yaklaşma oranı sonucunu doğurur. Genetik algoritmaların akılcı bir seçim baskısı aralığında

optimal veya optimale yakın çözümler ortaya koyması beklenir. Düşük seçilen seçim baskısı ise daha yavaş yakınsama ve daha uzun çözüm süresi demektir. Seçim baskısı çok yüksek ise genetik algorithmadaki artan değişim yanlış bir çözüme erken yakınsamaya yol açar (Chambers, 1995, s;72, Sivanandam ve Deepa, 2008, s;46, Bäck ve diğer., 2000a, s;56). Genel olarak uygulamalarda yüksek seçim baskısı problemleri ile karşılaşılmalıdır (Olariu ve Zomaya, 2006, s;13-196). Seçim metotları karşılaştırıldığında oransal seçimden sıralama ve turnuva seçimlerine geçerken seçim baskısı da artmaktadır. Sonuç olarak belirlenen seçim baskı şemasının popülasyon çeşitliliğini koruması ve erken (premature) yakınsamadan kaçınması gerekmektedir (Bäck, 1994, s;62).

Genel olarak iki tip seçim şeması tanımlanabilir: oransal seçim ve sıra tabanlı seçim. Oran tabanlı seçim, bireyleri, popülasyondaki diğer bireylerin uygunlukları ile göreceli olan uygunluklarına göre seçer. Sıra tabanlı seçim şeması ise bireyleri onların ham uygunluk değerlerine göre değil bu değerlerin popülasyon içerisindeki sırasına göre seçer. Bu durum seçim baskısının popülasyonun uygunluk dağılımından bağımsız olmasını ve yalnızca popülasyonun nispi sıralamasına dayalı olmasını gerektirir.

Seçim baskısını adapte edebilmek için popülasyonun uygunluk alanını yaymada bir ölçekleme fonksiyonu kullanılabilir. Örneğin tüm çözümlerin uygunluk aralığı [999,1000] ise oransal metot kullanılarak daha iyi bireyler elde etme olasılığı önemsiz sayılabilir. Eğer her bir bireyin uygunluğu $[0,1]$ 'e eşit bir aralığa getirilirse kötü birey yerine iyi birey seçme olasılığı önemli olacaktır. Genetik algorithmalarda kullanılan çeşitli seçim metotları aşağıda sıralanmıştır (Sivanandam ve Deepa, 2008, s;47).

3.4.1.1. Rulet Tekerleği Seçimi

Rulet seçiminin ana prensibi, bireylerin uygunluk değerlerine oranlı olarak ağırlıklandırılmış dilimlerden oluşan bir rulet tekerleği ile doğrusal bir arama yapılmasıdır. Hedef değer, popülasyondaki uygunluklar toplamının olasılıklı (random) oranından oluşan bir settir. Hedef değere ulaşıncaya kadar popülasyon ilerletilir. Bu metot ortalama güçlü bir seçim tekniğidir, çünkü uygun bireylerin seçilme garantisi yoktur ancak seçilme olasılıkları yüksektir.

Bir bireyin beklenen değeri popülasyonun gerçek uygunluğuna bölünerek bulunan uygunluk değeridir. Her bir bireye oransal uygunluğu ile orantılı olarak rulet

tekerleğinde bir dilim ayrılır. Tekerlek N defa döndürülür. Burada N popülasyondaki birey sayısıdır. Her bir turda tekerleğin işaretçisi altında kalan birey gelecek nesil için eşleşme havuzuna seçilir.

Bu metot aşağıdaki biçimde uygulanır (Sivanandam ve Deepa, 2008, s;48):

1. Popülasyondaki bireylerin toplam beklenen değerlerini topla. Bu değer T olsun
2. N defa tekrar et:
 - i. 0 ve T arasında rastgele bir 'r' sayısı seç.
 - ii. Popülasyondaki bireyleri döndürmeye başla. Toplam 'r' ye eşit veya onu geçene kadar beklenen değerleri topla. Beklenen değeri bu limiti geçen bireyi seç.

Rulet tekerleği seçiminin uygulaması kolaydır. Evrim oranı popülasyondaki uygunluğun varyansına bağlıdır.

3.4.1.2. Rastgele Seçim

Bu teknikte gelecek nesli oluşturmak için aile popülasyondan rastgele seçilerek belirlenir.

3.4.1.3. Sıralama (Rank) Seçimi

Uygunluk değerlerinin çok fazla farklılık sergilediği durumda rulet tekerleği seçimini uygulamak zor olmaktadır. Eğer en iyi kromozomun uygunluğu %90 ise tekerlekte de %90 yer kaplayacaktır ve bu durumda diğer kromozomların seçilme şansı çok az olmaktadır.

Sıralama seçiminde seçilme olasılığı mevcut popülasyondaki bireylerin uygunluğunun sıralamadaki yerine göre belirlenmektedir. Sıralama seçimi popülasyonu en iyiden en kötü bireye doğru sıralar. Her bir bireyin beklenen değeri onun mutlak uygunluk değeri ile değil sıralamadaki yerine göre belirlenmektedir. Sıralamadaki yerlerine göre bireyler ayrı bir uygunluk değeri alır. Uygunluklar arasındaki mutlak farklılıklar dikkate alınmadığı için sıralama seçiminde uygunlukları derecelendirme gereksinimi de bulunmamaktadır (Sakawa, 2002, s;21).

Bu seçim yakınsamayı yavaşlatır ancak çok hızlı yakınsamayı da önler. Uygunluk varyansının düşük olduğunda da seçim baskısını yüksek tutar. Çeşitliliği

korur ve daha başarılı arama sağlar. Uygulamada potansiyel aileler seçilir ve hangi ailenin bir sonraki nesil seçileceğine yönelik bir turnuva düzenlenir (Mitchell, 1999, s;127, Sivanandam ve Deepa, 2008, s;47).

Sıralama, amaç fonksiyonun tam olarak tanımlanamadığı problemler için doğal bir seçenek olabilir. Bu gibi problemlerde tüm dikkati amaç fonksiyonunu alacağı değere atfetmek mantıklı değildir çünkü bazı durumlarda bu durum mümkün de olmayabilir (Bäck ve diğer., 2000a, s;188).

Farklı şekillerde yapılabilir olmakla birlikte iki yöntem aşağıda sıralanmaktadır (Sivanandam ve Deepa, 2008, s;47):

1. Birey çiftlerini rastgele seç. 0 ile 1 arasında 'R' rassal sayısını üret. Eğer $R < r$ ise ilk bireyi aile olarak seç. $R \geq r$ ise ikinci bireyi seç. Bu ikinci ailenin seçimi için de aynen tekrarlanır. 'r' değeri bu metot için bir parametredir.
2. Rastgele iki farklı birey seç. Daha yüksek değerlendirme değeri olan bireyi aile için seç. İkinci aileyi bulmak için de bu adımı tekrarla.

3.4.1.4. Turnuva Seçimi

Genetik algoritma arama performansının en iyi olması için seçim baskısı ve popülasyonun çeşitliliğinin ayarlanabildiği bir seçim yöntemi idealdir. Turnuva seçimi seçim baskısı açısından sıralama seçimi ile benzerdir ancak ona göre hesaplaması daha etkin ve paralel uygulamalar için daha uygundur (Mitchell, 1999, s;127).

Rulet tekerleri seçiminden farklı olarak turnuva seçimi N_u birey arasında bir turnuva düzenleyerek seçim baskısı sağlar. Kazanan birey uygunluk değerine göre belirlenir. En yüksek uygunluk değerine sahip birey genellikle deterministik olarak seçilir. Turnuva, eşleşme havuzu yeni nesil oluşturmak için doluncaya kadar tekrarlanır. Genelde turnuva iki birey arasında düzenlenir ancak keyfi olarak seçilen bir sayıdaki birey arasında da uygulanabilir. Turnuva gruplardan bağımsız olarak düzenlenir ancak grup sayısı arttıkça seçim baskısı da artacaktır. Uygulamada 6-10 grup önerilmektedir (Bäck ve diğer., 2000a, s; 181, Olariu ve Zomaya, 2006, s;13-196). Turnuva kazananlardan oluşan eşleşme havuzu yüksek bir ortalama popülasyon uygunluğuna sahiptir. Bu uygunluk farkı seçim baskısı sağlar ve bu durum da, genetik algoritmayı başarılı genlerin uygunluğunu artırmaya zorlar (Sivanandam ve Deepa, 2008, s;49).

3.4.1.5. Boltzmann Seçimi

Boltzmann seçimi tavlama benzetimi (simulated annealing) prensiplerini kullanarak ve termodinamik olarak seçim baskısını kontrol eden seçim mekanizmasıdır. Tavlama benzetimi algoritması bir yerel arama algoritmasıdır ve pek çok değişkene sahip fonksiyonların en büyük veya en küçük değerlerinin bulunması ve özellikle pek çok yerel en küçük değere sahip doğrusal olmayan fonksiyonların en küçük değerlerinin bulunması için tasarlanmıştır (Lai ve Chan, 1997, s;115). Boltzmann seçiminde önceden belirlenmiş tabloya göre sürekli değişen sıcaklık seçim oranını kontrol eder. Sıcaklık düşük seçim baskısı anlamına gelecek biçimde yüksek olarak tutulur. Daha sonra seçim baskısı kademeli olarak yükseltilir ve böylelikle uygun bir seviyedeki çeşitlilik korunurken genetik algoritmanın arama uzayını en iyi kısma yakın olacak biçimde küçültmesi sağlanır (Mitchell, 1999, s;126, Sivanandam ve Deepa, 2008, s;49).

Boltzmann seçim mekanizmasının temelinde mevcut i çözümü ve alternatif j çözümü arasındaki rekabet yatar. Burada i 'nin kazanma olasılığı

$$\frac{1}{1 + e^{(f_i - f_j)/T}}$$

formülü ile hesaplanmaktadır (Bäck, Fogel ve Zbigniew-2000b, s; 195). Burada T sıcaklığı, f_i ve f_j mevcut ve alternatif çözüm maliyet, amaç fonksiyon değeri veya uygunluk değerini göstermektedir. $T = T_0(1 - \alpha)^k$ ve $k = (1 + 100 * g/G)$ dir. g ; mevcut jenerasyon sayısı, G ; maksimum g değeridir. α , $[0,1]$ aralığından, T_0 ise $[5,100]$ aralığından seçilmektedir. T , 0 değerine ulaştığında global çözüme ulaşılmış olmaktadır (Sivanandam ve Deepa, 2008, s;49).

Seçim mekanizmasının seçimi diğer yönlerinden bağımsız olarak yapılamaz. Değerlendirme fonksiyonu çok karmaşık ise en fazla artırimsal modeller problemlili olacaktır. Karmaşıklık durumunda seçim baskısı azaltılmalıdır. Karmaşıklık yoksa artırimsal modeller yeni iyi bireyler önemli artışlar sağlayabilir. Bireylerin silinmesi de dikkatli olunmasını gerektirmektedir. En kötü bireylerin ortadan kaldırılması büyüme oranını artırmaktadır ancak bu durum daha düşük kontrol demektir.

Seçkinlik (Elitism)

Seçkinlik, evrimsel algoritmalarda bir popülasyonun maksimum uygunluk değerinin bir diğerine geçildiğinde azalmamasını temin eden özelliktir (Bäck ve diğer.,

2000b, s; xxv). Çaprazlama ve mutasyonla yeni populasyon üretildiği zaman, en iyi kromozom kaybedilecektir. Seçkinlik, önce en iyi kromozomu (yada birkaç en iyi kromozomu) yeni popülasyona kopyalar. Geri kalanı diğer yöntemlerle devam ettirilir. Seçkinlik, bulunan en iyi sonucun kaybını önlediğinden GA'nın performansını oldukça hızlı bir şekilde artırır (Mitchell, 1999, s;126, Nabiye, 2003, s;616).

3.4.2. Çaprazlama

Çaprazlama, genetik algoritmaların benzer metotlar arasında en ayırt edici özelliği ve en önemli operatörüdür. Genetik çözümün performansı büyük ölçüde bu operatöre bağlıdır. Çaprazlama, farklı çözümler arasında bilgi değişimini sağlayarak arama uzayının benzer ancak araştırılmamış bölgelerine ulaşmayı sağlayan bir arama operatörüdür (Gen ve Cheng, 2000, s;8, Kennedy ve diğerleri, 2001, s.157, Dengiz ve Altıparmak, 1998, s;8). Çaprazlama mevcut popülasyondan iki ebeveyni alıp bunlardan bir birey üretme prosesidir. Bu operatörün asıl işlevi popülasyondaki farklı bireylerden gelen parça bilgileri yada genetik materyali bir araya getirmektir (Sakawa, 2002, s;22). Çaprazlama ile seçim prosesinden sonra daha iyi bireyler ile popülasyon zenginleştirilir. Yeniden üretme/seçim (reproduction) ile iyi diziler klonlanır ancak yenileri oluşturulmaz. Çaprazlama operatörü daha iyi bireylerin ortaya konulması beklentisi ile eşleşme havuzuna uygulanır.

Çaprazlama operatörleri çaprazlamaya konu olan ebeveyn sayısına göre üç ana kategoriye bölünebilir (Engelbrecht, 2007, s;144):

- Yeni birey tek bir ebeveyn den üretilecek ise aseksüel,
- Bir veya iki yeni birey iki ebeveyn den üretilecek ise seksüel,
- İki den fazla ebeveyn den bir veya daha fazla yeni birey üretilmesi durumunda da çoklu-birleşim (muti-recombination)

Bunun dışında kodlama biçimine özgü (ikili-tam sayılı gibi) veya operatöre özgü (floating point gibi) sınıflandırmalara da rastlanılmaktadır.

Çaprazlama, üç adımdan oluşan bir yeniden birleştirme operatörüdür (Sivanandam ve Deepa, 2008, s;50):

1. Yeniden üretme operatörü eşleşme için rassal olarak bir çift birey dizisi seçer,
2. Dizi boyunca rassal olarak bir çaprazlama kısmı belirlenir.

3. Son olarak bu kısımdaki değerler karşı dizidekiler ile yer değiştirilir.

Bir nesil içerisinde çaprazlamaya konu olacak ebeveyn dizisinin oranı çaprazlama oranı tarafından belirlenir. $p_c \in [0,1]$ aralığında değer alacak bu oran çaprazlama operatörünün ne kadar sıklıkla kullanılacağını göstermektedir (Sakawa, 2002, s; 22).

Çeşitli çaprazlama teknikleri aşağıda sıralanmaktadır:

3.4.2.1. Tek noktalı çaprazlama

Geleneksel olarak genetik algoritmalar tek noktalı çaprazlamayı kullanır. Bazı kaynaklarda bu yöntem ‘basit çaprazlama’ olarak adlandırılmaktadır. Bu metotta kromozomlar karşılıklı olarak belli bir noktadan kesilir ve kesilen noktadan sonraki kısımlar değiştirilir. Çaprazlama noktası dizi içerisinde rassal olarak belirlenir ve bir sonraki bit’ten başlayarak değişim gerçekleştirilir. Eğer uygun bir nokta seçilmiş ise iyi ebeveynleri eşleyerek daha iyi çocuklar elde edilebilir. Aksi durumda ise dizinin kalitesi oldukça kötü etkilenecektir (Sakawa, 2002, s;24, Sivanandam ve Deepa, 2008, s;51). Tek noktalı çaprazlama aşağıdaki örnekte gösterilmiştir.

Ebeveyn 1	1 0 1 1 0 0 1 0
Ebeveyn 2	1 0 1 0 1 1 1 1
↓	
Çocuk 1	1 0 1 1 0 1 1 1
Çocuk 2	1 0 1 0 1 0 1 0

Şekil 3.10. Tek noktalı çaprazlama

Tek noktalı çaprazlama konumsal eğiliminden dolayı kusurlu olabilmektedir. Bit indisi l 'ye (uzunluk) doğru artarken değişim olasılığı da bire yaklaştığı için değişim olasılığının bit pozisyonuna büyük ölçüde bağlı olduğu söylenebilir (Bäck, 1996, s;114).

3.4.2.2. Çift noktalı çaprazlama

Birden fazla kesim noktası içeren çok farklı çaprazlama algoritmaları geliştirilmiştir. Çaprazlama noktası eklenmesi genetik algoritmanın performansını düşürür. İlave çaprazlama noktaları eklendiğinde ortaya çıkan problem ebeveynler

arasında değiştirilecek blokların birbirine çok fazla benzemesidir. Bununla birlikte daha fazla çaprazlama noktasına sahip olmanın problem yüzeyini tam tarama gibi bir avantajı vardır.

İki noktalı çaprazlamada iki farklı çaprazlama noktası seçilir ve eşleşen iki ebeveynin bu iki nokta arasında kalan kısımları birbiri ile değiştirilir.

Ebeveyn 1	1 0 1 1 1 0 1 0
Ebeveyn 2	0 1 1 0 1 1 0 0
↓	
Çocuk 1	1 0 1 0 1 1 1 0
Çocuk 2	0 1 1 1 1 0 0 0

Şekil 3.11. Çift noktalı çaprazlama

Tek noktalı çaprazlamada dizinin baş ve kuyruk yada son kısmı birlikte döl'e kopyalanamaz. Hem baş hem de kuyruk kısmı iyi genetik bilgi içeriyor ise bu bilgi tek noktalı çaprazlama ile döl'e aktarılamaz. İki noktalı çaprazlama bu problemi ortadan kaldırır ve genellikle tek noktalı çaprazlamaya göre daha iyi sonuçlar verir. (Sivanandam ve Deepa, 2008, s;51). Çaprazlamada kullanılacak nokta sayısı üzerinde yapılan çalışmalar sonucunda standart olarak iki noktalı çaprazlama kullanılmaktadır (Bäck, 1996, s;115).

Kromozom üzerinde birbirine yakın olan genlerin çok noktalı bir çaprazlamada döllere birlikte geçmesi olasılığı daha fazladır. Bu durum da yan yana olan genler arasında istenmeyen bir bağlantı oluşturur. Sonuç olarak N noktalı bir çaprazlamanın etkinliği genlerin kromozomdaki yerlerine bağlıdır. Genetik bir sunumda çözümün birbirine bağımlı karakteristiklerini kodlayan genler birbirine yakın olmalıdır. Genin yerleşimi ile ilgili tüm bu problemlerden kaçınabilmenin yolu uniform çaprazlama kullanılmasıdır (Sivanandam ve Deepa, 2008, s;52).

3.4.2.3. Çok noktalı çaprazlama

Çok noktalı çaprazlamada iki seçenek söz konusudur. Biri çift sayılı, diğeri ise tek sayılı kısım çaprazlamadır. Çift sayılı seçilmesi durumunda çaprazlanacak kısımlar

bir daire etrafında rassal olarak seçilir ve bilgiler değiştirilir. Tek sayılı melez alan seçildiğinde dizinin başında farklı bir çaprazlama noktası farz edilir.

Çaprazlama noktası sayısı tek olduğunda değişim olasılığı asimetrik biçimde bit pozisyonuna bağlı olarak değişir. Bir çok durumda simetri nedeni ile küçük çift sayılı çaprazlama nokta sayısı belirlenmesi önerilmektedir (Bäck, 1996, s;114).

3.4.2.4. Uniform çaprazlama

Uniform çaprazlama n noktalı çaprazlamadan oldukça farklıdır. Döl üzerindeki her bir gen, kromozomlarla aynı uzunlukta ve tesadüfi oluşturulan ikili bir çaprazlama maskesine göre seçilen ebeveynden gelir. Çaprazlama maskesinde 1 olduğu yerde gen ilk ebeveynden, 0 olduğu yerde ise ikinci ebeveynden kopyalanır. Her bir ebeveyn çifti için yeni bir çaprazlama maskesi rassal olarak oluşturulur. Bundan dolayı, dölleri her bir ebeveynden gelen bir gen karışımını taşır. Uniform çaprazlama biçimi aşağıdaki şekilde gösterilmektedir. Örnekte ilk çocuk üretiminde maskede 1 olduğunda gen birinci ebeveynden seçilmiş, ikinci çocukta ise tersi uygulanmıştır (Sakawa, 2002, s;24, Sivanandam ve Deepa, 2008, s;52).

Ebeveyn 1	1 0 1 1 0 0 1 1
Ebeveyn 2	0 0 0 1 1 0 1 0
Maske	1 1 0 1 0 1 1 0
Çocuk 1	1 0 0 1 1 0 1 0
Çocuk 2	0 0 1 1 0 0 1 1

Şekil 3.12. Uniform çaprazlama

3.4.2.5. Üç ebeveynli çaprazlama

Bu çaprazlama tekniğinde tesadüfi olarak üç farklı ebeveyn seçilir. İlk ebeveyndeki her bir bit ikinci ile karşılaştırılır. Eğer her ikisi aynı ise bu gen döl için seçilir; eğer farklı ise üçüncü ebeveynin geni seçilir.

3.4.2.6. Karıştırmalı (shuffle) çaprazlama

Karıştırmalı çaprazlamada ilk olarak tek noktalı çaprazlamada olduğu gibi tek bir çaprazlama noktası seçilir. Ancak değişkenler karşılıklı olarak değiştirilmeden önce

her iki ebeveynde rakamlar tesadüfî olarak karıştırılır. Daha sonra değişim gerçekleştirilir. Her çaprazlama yapıldığında değişkenler random olarak yeniden atandığı için bu metotla konumsal eğilimler engellenmiş olur (Sivanandam ve Deepa, 2008, s;52). Bu teknik herhangi birçok noktalı çaprazlama ile birlikte kullanılabilir. Standart operatörlerin pozisyona bağlı eğilimlerini azaltabilmek için her iki ebeveynin aynı biçimde bit pozisyonları karıştırılır ve daha sonra çok noktalı operatör uygulanır. Son aşamada ise karıştırılan bitler eski yerlerine yerleştirilir. (Bäck, 1996, s;116)

Yukarıda sıralananların dışında literatürde çok sayıda çaprazlama metodu geliştirilmiştir. Bunlardan en fazla kullanılan diğerleri Üstünlük korumalı çaprazlama, Ordered çaprazlama, ve kısmi eşlenmiş çaprazlamadır.

3.4.3. Mutasyon

Çaprazlama mevcut gen potansiyellerini araştırmak üzere kullanılır. Fakat populasyon gerekli tüm kodlanmış bilgiyi içermez ise, çaprazlama tatmin edici bir çözüm üretmez. Bu durum, mevcut kromozomlardan yeni kromozomlar üretme yeteneğine sahip bir operatör gerekmektedir. Bu görevi mutasyon operatörü ile gerçekleştirilir (Emel ve Taşkın, 2002, s;134). Mutasyonun amacı mevcut bir bireyden yeni genetik materyaller ortaya koymak ve popülasyonun karakteristiğine çeşitlilik kazandırmaktır. Mutasyon, her bir gen üzerindeki tüm alellere ulaşılabilmesini sağlamak üzere çaprazlamaya yardımcı olarak kullanılır (Engerbrecht, 2007, s;153).

Mutasyon gendeki bir alel'in tesadüfî olarak bir diğeri ile yer değiştirmesi ile uygulanır. Başka bir ifadeyle önceki potansiyel çözümün sunumunda yapılan küçük tesadüfî değişiklikler ile yeni bir potansiyel çözüm elde edilmektedir (Bäck ve diğer., 2000a, s; 68) Genetik algoritmaların çalışmasında ikinci dereceden rol oynayan mutasyon, küçük bir olasılıkla bir dizideki bir veya birkaç değeri rassal olarak değiştirerek yeni dizilerin yada arama uzayında yeni çözüm noktalarının elde edilmesine çalışır (Dengiz ve Altıparmak, 1998, s;6).

Mutasyon popülasyonu lokal minimuma takılmaktan uzak tutar. Tesadüfî işlemlerle kaybedilen genetik bilgiler gibi kayıp genetik materyalin geri kazanılmasında rol alır. Aynı zamanda genetik materyalin geri dönülemez kaybına karşı da bir sigorta işlevi görür.

Mutasyon geleneksel olarak basit bir arama operatörü olarak düşünülmüştür. Eğer çaprazlamanın daha iyi çözüme ulaşmak için mevcut çözümü kullandığı düşünülürse; mutasyonun da tüm arama uzayının keşfi için yardımcı olacağı düşünülür. Mutasyon popülasyondaki çeşitliliği koruyan temel operatördür.

Farklı sunumlar için çok farklı mutasyon biçimleri mevcuttur. İkili kodlamada basit bir mutasyon, küçük bir olasılıkla her bir genin değerini tersine çevirmekle yapılabilir. Burada bahsedilen olasılık değeri, L kromozom uzunluğu olmak üzere, $1/L$ olarak alınır. Yalnızca çözümün kalitesini artırdığı durumda mutasyon yapılan tepe-tırmanma (hill-climbing) mutasyon operatörlerinin de kullanılması mümkündür. Bu gibi bir operatör aramayı hızlandırabilir ancak popülasyondaki çeşitliliği azaltması ve algoritmayı bazı yerel optimumlara yakınlaştırması konularında da dikkatli olunmalıdır. (Sivanandam ve Deepa, 2008, s;56)

3.4.3.1. Döndürme (flipping)

Bir bit'i döndürme, oluşturulan bir mutasyon kromozomuna göre 0'ı 1'e ve 1'i 0'a çevirme işlemidir. İlk olarak bir ebeveyn seçilir ve tesadüfi olarak bir mutasyon kromozomu oluşturulur. Mutasyon kromozomundaki bir 1'e tekabül eden ebeveyn üzerindeki bit '0' ise '1'e; '1' ise 0'a çevrilerek yeni birey oluşturulur. Mutasyon kromozomunda 0 olan bitlere tekabül eden ebeveyn bitleri yeni bireye dönüştürülmeden aynen kopyalanır. Şekil 3.x te bu işleyiş gösterilmiştir. (Sivanandam ve Deepa, 2008, s;57)

Ebeveyn	1 0 1 1 0 1 0 1
Mutasyon Kromozomu	1 0 0 0 1 0 0 1
Çocuk	0 0 1 1 1 1 0 0

Şekil 3.13. Döndürme (flipping)

3.4.3.2. Değiş-tokuş

Dizi üzerinde tesadüfi olarak iki farklı nokta belirlenir ve bu noktalardaki bitler değiş tokuş edilerek yeni nesil kromozom üretilir. Şekil 3.x te bu işleyiş gösterilmiştir. (Sivanandam ve Deepa, 2008, s;57)

Ebeveyn	1 0 1 1 0 1 0 1
Çocuk	1 1 1 1 0 0 0 1

Şekil 3.14. Değiş-tokuş

3.4.3.3. Gaussian mutasyon

Her bir kromozom için Poisson dağılımına göre mutasyona uğratılacak gen belirlenir. Seçilen bu genler dönüştürülür.

3.4.4. Yenileme/Yer Değiştirme (Replacement)

Yenileme üreme döngüsünün son aşamasıdır. Sabit büyüklükteki popülasyondan iki ebeveyn seçilir ve bunlar da iki yavru yeni birey üretir; ancak bu dört bireyin dördü de popülasyona geri dönmez. Dolayısıyla ikisinin değiştirilmesi gerekmektedir. Hangi bireyin popülasyonda kalacağı ve hangisinin değiştirileceğine karar vermede kullanılan teknik, seçim kadar yakınsama sürecinde etkilidir. Temelde popülasyonun devamı için iki farklı metot vardır: kuşaksal güncelleme ve sabit durum güncellemesi.

Temel kuşaksal güncelleme şeması, gelecek nesli oluşturmak için N büyüklüğündeki bir popülasyondan yine N sayıda çocuk üretimini ve bu yeni çocuk popülasyonunun tamamen ebeveynleri ile yer değiştirmesini kapsamaktadır. Bu metotta bir birey yalnızca aynı nesilden başka bir birey ile yeni bir birey oluşturabilir.

Kuşaksal güncellemenin $(\lambda+\mu)$ güncellemesi ve (λ,μ) güncellemesi gibi türetilmiş biçimleri de kullanılmaktadır. Popülasyon büyüklüğü μ , ve üretilen yeni birey sayısı da $\lambda \geq \mu$ 'dür. Ya oluşturulan yeni nesil popülasyonundan yada bütünleştirilen ebeveyn ve çocuk popülasyonundan gelen μ sayıda en iyi birey gelecek nesli oluşturur.

Sabit durum güncellemesinde ise yeni bireyler, kuşaksal güncellemenin tersine, oluşturulur oluşturulmaz popülasyona dahil edilir. Yeni birey eklenirken popülasyondaki ya en kötü birey ya da en eski birey silinir. Sabit durum güncellemesi metodu hem seçim hem de güncelleştirmede turnuva metodu gibi sıra temelli metotları kullanır. Turnuva yenilemesi turnuva seçimine oldukça benzerdir. (Sivanandam ve Deepa, 2008, s;58)

3.4.4.1 Tesadüfi yenileme (Random replacement)

Bu yöntemde çocuk yada yeni bireyler popülasyonda tesadüfi olarak seçilen bireylerle yer değiştirir. Ebeveynler de bu seçimde aday konumundadır. Küçük popülasyonlarda aramayı sürdürmek için yararlı olabilir çünkü popülasyona zayıf bireyler girebilir.

3.4.4.2. Zayıf ebeveyn yenileme

Bu yenileme/yer değiştirme metodunda zayıf ebeveyn güçlü çocuk ile değiştirilir. Üremeden sonra oluşan dört bireyden yalnızca en uygun ikisi popülasyona geri dönebilir. Bu süreç, çaprazlama için hem uygun hem de zayıf ebeveynleri kullanan bir seçim tekniği ile birlikte kullanıldığında popülasyonun genel uygunluğunu artırır.

3.4.4.3. Her iki ebeveyni yenileme

Bu yöntemde çocuklar ebeveynlerinin yerine geçer. Her birey yalnızca bir kez üremeye katılır. Popülasyon ve genetik materyal sürekli değişir. Ancak ancak uygun ebeveynleri öne çıkaran seçim yöntemleri ile düşünüldüğünde problemler yaşanmaktadır.

3.5. Yakınsama Kriterleri (Arama Sonlandırma)

Yerel bir optimuma ulaşıldığında basit yakın komşuluk arama metotlarından farklı olarak genetik algoritma prensip olarak sonsuza kadar çalışır. Bunun için pratikte bir sonlandırmam kriteri gerekmektedir. Çeşitli durdurma koşulları aşağıda sıralanmıştır (Reeves ve Rowe, 2002, s;30, Sivanandam ve Deepa, 2008, s;59):

- Maksimum Nesil: tanımlanan sayıda nesil üretildiği durumda genetik algoritma durdurulur.
- Geçen Zaman: önceden tanımlanan süre dolduğunda genetik proses sonlandırılır. Nesil sayısı kriteri ile birlikte kullanıldığında önceden tanımlanan nesil sayısına süreden önce ulaşırsa algoritma durur.
- Değişmeyen uygunluk: Tanımlanan sayıdaki nesil için popülasyonun en uygun değeri değişmiyor ise genetik proses sonlanır.

Bazı sonlandırma teknikleri aşağıda sıralanmıştır:

3.5.1 En iyi birey

Bu kriter popülasyondaki en düşük uygunluğu yakınsama değerinin altına düştüğünde arama durdurulur. Bu metot, en azından bir adet iyi çözümü garanti ederek aramayı daha hızlı sonlandırır.

3.5.2. En kötü birey

En kötü birey kriteri kullanıldığında popülasyondaki en az uygun bireyin yakınsama kriterinden daha düşük bir uygunluğa sahip olduğu durumda arama sonlandırılır. Bu durumda tüm popülasyonun belli bir minimum standardın üstünde olması sağlanır. Ancak en iyi bireyin en kötüden önemli ölçüde farklı olmayabilir. Bu durumda, aşılacak belli bir maksimuma göre tanımlanan yüksek bir yakınsama değeri hiçbir zaman sağlanamayabilir.

3.5.3. Uygunluk Toplamı

Bu sonlandırma şemasında, popülasyondaki uygunluğun toplamının popülasyon kaydındaki (record) yakınsama değerine eşit veya daha düşükse aramanın, yakınsamayı sağladığı düşünülür. Bu durum hemen hemen tüm bireylerin belli bir uygunluk aralığında olmasını garanti eder. Ancak bu yaklaşma kriterini en zayıf gen değişimi ile birlikte düşünmek daha iyi sonuç verir. Aksi durumda popülasyondaki yeni birkaç düşük uygunluklu birey toplam uygunluğu bozabilir. Yakınsama değeri belirlenirken popülasyon büyüklüğü dikkate alınmalıdır.

3.5.4. Orta uygunluk

Bu yaklaşımda sonlandırma için bireylerin en azından yarısının belirlenen yakınsama kriterinden eşit ya da daha iyi olması gerekir.

3.6. Genetik Algoritmalarda Parametre Seçimi

Genetik algoritma uygulamasının önemli bir adımı da çeşitli parametrelerin belirlenmesidir. Parametreler genetik algoritma performansı üzerinde çok önemli bir etkiye sahiptir. Kontrol parametreleri olarak da adlandırılan bu parametreler popülasyon

büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, kuşak aralığı, seçim stratejisi, fonksiyon ölçeklemesini kapsamaktadır (Dengiz ve Altıparmak, 1998, s;9, Mitchell, 1999, s;130, Emel ve Taşkın, 2002, s;135)

Parametreler birbirlerini doğrusal olmayan biçimde etkilediği için aynı anda optimize edilemezler. Literatürde parametre seçimi ve adaptasyonu üzerinde oldukça fazla sayıda çalışma yapılmış olmakla birlikte tüm problemlerde kullanılabilecek en iyi parametreler üzerinde bir uzlaşma söz konusu değildir (Mitchell, 1999, s;131)

3.6.1. Popülasyon büyüklüğü

Popülasyon büyüklüğü genetik algoritma kullanıcısı tarafından verilen en önemli kararlardan biridir. Bu değer küçük olarak seçildiğinde algoritmanın yerel optimuma takılması olasılığı vardır. Büyük bir değer ise çözüm uzayının daha iyi tarama olasılığına karşı çözüm için gerekli zaman ve kaynakları artırmaktadır. (Emel ve Taşkın, 2002, s;135, Reeves ve Rowe, 2002, s;25)

Popülasyon büyüklüğü diğer genetik algoritma parametrelerinden tamamen ayrı düşünülemez. Belirlenecek büyüklük, hesaplama etkinliği, yeterli örnekleme sağlayabilme durumu, popülasyonun çeşitliliği, çaprazlama ve seçim baskısı gibi parametrelerin dengelemesine katkı sağlamalıdır (Bäck ve diğer., 2000b, s;134)

3.6.2. Çaprazlama olasılığı

Çaprazlama operatöründe temel parametre çaprazlama olasılığıdır (P_c). Çaprazlama olasılığı, çaprazlamanın ne kadar sıklıkla yapılacağını tanımlayan bir parametredir. Çaprazlama olmadığı durumda döl, ebeveynlerin direkt kopyası olur. Ancak yeni neslin önceki ile aynı olduğu anlamına gelmez. Çaprazlamanın olduğu durumda döl, ebeveynlerin her ikisinin kromozomlarının parçalarından oluşur. Çaprazlama, yeni kromozomların eskilerinin iyi parçalarını taşıyacağı beklentisi ile yapılır (Sivanandam ve Deepa, 2008, s;56).

Çaprazlama olasılığı eşleşme havuzuna girecek kromozomların sayısını belirler. Bu oranın çok yüksek olması durumunda iyi yapı bloklarının tek bir kromozomda birikme olasılığı oldukça azalır. Düşük bir oran belirlenmesi durumunda ise yeterli sayıda yeni çocuk/döl ortaya konulamaz (Haupt ve Haupt, 2004, s;117).

3.6.3. Mutasyon olasılığı

Mutasyon tekniğindeki önemli parametre, kromozomda ne kadar bir değişiklik olacağını gösteren mutasyon olasılığıdır. Eğer mutasyon yok ise yeni birey çaprazlamadan hemen sonra oluşturulur demektir. Mutasyon olasılığının %100 olması ise kromozomun tamamen değiştirildiği anlamına gelir (Sivanandam ve Deepa, 2008, s;57). Mutasyonun temel amacı popülasyondaki çeşitliliği korumaktır. Mutasyon olasılığının artması genetik aramayı raslantısal bir aramaya dönüştürür; ancak diğer taraftan bu durum kayıp genetik materyali tekrar bulmada yardımcı olmaktadır (Emel ve Taşkın, 2002, s;136).

Bazı çalışmalar büyük popülasyonlar için ($\mu > 200$) büyük mutasyon olasılığı ($P_m > 0.05$) kullanıldığında; benzer şekilde küçük popülasyonlar ($\mu < 20$) içinse daha küçük mutasyon oranı ($P_m < 0.002$) kullanıldığında performansın arttığını ortaya koymuştur. Orta büyüklükteki bir popülasyon büyüklüğü için $P_m = 0.001$ oranı seçilebilir (Bäck, 1996, s;123).

3.6.4. Kuşak Aralığı

Her kuşaktaki yeni kromozom oranına kuşak aralığı denir. Yüksek kuşak aralığı çok sayıda kromozomun değiştiği anlamına gelmektedir (Mitchell, 1999, s;128, Emel ve Taşkın, 2002, s;136).

3.6.5. Seçim Stratejisi

Eski kuşağı yenilemenin çeşitli yöntemleri mevcuttur. Kuşaksal stratejide, mevcut popülasyondaki kromozomlar tamamen yavrular ile yer değiştirir ancak popülasyonun en iyi kromozomu da yenilendiğinden dolayı bir sonraki kuşağa aktarılamaz. İyi niteliklerin bir sonraki kuşağa aktarımı için bu strateji en uygun (elitist) stratejisiyle beraber kullanılmaktadır. En uygun stratejisinde, popülasyondaki en iyi kromozomlar hiçbir zaman yenilenmemektedir. Sabit durum (steady state) stratejisinde ise, her kuşakta yalnızca birkaç kromozom yenilenmektedir. Genellikle, yeni kromozomlar popülasyona katıldığında en kötü kromozomlar yenilenir. (Emel ve Taşkın, 2002, s;136)

3.6.6. Fonksiyon Ölçeklemesi

Doğrusal ölçekleme, üstsel ölçekleme gibi yöntemler mevcuttur. Probleme göre en uygun ölçekleme yönteminin seçilmesi genetik algoritmanın etkin işlemesi açısından önem taşımaktadır. (Emel ve Taşkın, 2002, s;136)

3.7. GA'nın Geleneksel Optimizasyon Yöntemleri ile Karşılaştırılması

Genetik algoritmaların temel prensibi basittir: genetik ve doğal seleksiyonu bilgisayar yazılımları kullanarak taklit etmek. Problem parametreleri de DNA'daki gibi bir doğrusal veri yapısı, bir vektör veya bir dizi gibi en doğal biçimi ile kodlanır. Genetik algoritmalar geleneksel optimizasyon tekniklerinden aşağıdaki yönlerden farklıdır (Rao, 1996, s; 808, Sakawa, 2002, s;15, Sivanandam ve Deepa, 2008, s;34):

- Genetik algoritmalar problem parametrelerinin kendilerinden ziyade kodlanmış versiyonları ile çalışırlar.
- Hemen tüm geleneksel optimizasyon teknikleri tek bir noktadan hareket ederek araştırır ancak Genetik algoritmalar ise tek bir çözüm yerine çözümlerin popülasyonunu kullanır. Bu durum GA'nın global optimuma ulaşma olasılığını artırır ve yerel sabit bir noktaya takılmasından kaçınılmasını sağlar.
- Genetik algoritmalar değerlendirme için türevlerden ziyade uygunluk fonksiyonunu kullanır. Bu fonksiyonlar da sürekli yada kesikli her tür optimizasyon problemine uygulanabilir.
- Geleneksel yöntemler sürekli optimizasyonlara deterministik dönüşüm uygularken genetik algoritmalar olasılıklı dönüşüm uygular.

Genetik Algoritmaların geleneksel yaklaşımların başarısız olduğu durumlarda çarpıcı sonuçlar ortaya koymasını sağlayan diğer bazı farkları ve avantajları aşağıdaki gibi sıralanabilir (Haupt ve Haupt, 2004, s;23, Şen, 2004, s;62, ve Sivanandam ve Deepa, 2008, s;34, Sumathi ve diğer., 2008, s;122):

- Sürekli ve kesikli değişkenleri optimize edebilir.
- Türev alma işlemine gereksinim duymaz.
- Çözüm alanında daha geniş örneklemeyi eş zamanlı araştırır. Global optimumu daha kolay belirler. Klasik yöntemlerde ise bu durum tek nokta ve yön olarak karşımıza çıkmaktadır.

- Çok sayıda değişken ile çalışabilir.
- Farklı problemlere kolaylıkla modifiye edilebilir.
- Paralel bilgisayarlarla hesaplamalar için oldukça uygundur.
- Oldukça karmaşık çözüm yüzeylerinde yerel minimumlara takılmadan değişkenleri optimize edebilir.
- Sadece tek bir çözüm yerine optimum değişkenlerin bir listesini verebilir.
- Kodlanmış değişkenler ile optimizasyon sağlayabilmek için değişkenleri kodlayabilir.
- Nümerik olarak oluşturulan verilerle, deneysel verilerle ve analitik fonksiyonlar ile çalışabilir.
- Kendi iç kuralları ile çalıştığı için problemin kurallarını bilmesi gerekmez.
- Problemin doğrusallaştırılmasını gerektirmemektedir.

Bununla birlikte Genetik algoritmaların her tür problemin çözümünde en iyi metot olduğu da söylenemez. Yalnızca birkaç değişkeni olan düzgün bir analitik fonksiyon geleneksel yöntemlerle kolayca çözülebilir. Bu gibi problemlerde Genetik Algoritma başlangıç popülasyonu için maliyetleri hesaplarken geleneksel yöntemler hesaplamayı tamamlamış olabilir. Genetik algoritmaların sınırlılıkları ve dezavantajları ise şu şekilde sıralanabilir (Sivanandam ve Deepa, 2008, s;34):

- Uygunluk fonksiyonunu tanımlama problemi
- Vaktinden önce yakınsama olabilmektedir.
- Popülasyon büyüklüğü, mutasyon oranı, çaprazlama oranı, seçim metodu gibi çeşitli parametrelerin belirlenmesi problem oluşturabilir.
- Probleme özgü bilgiler kolaylıkla dahil edilemez
- Lokal optimumları tanımlamada iyi bir metot değildir.
- İyi bir sonlandırıcı değildir. Harici yardım gerekebilir.
- Kesin global optimumu bulmada sorun yaşayabilir.
- Çok sayıda uygunluk fonksiyonu değerlendirmesine gereksinim duyar.

3.8. Genetik Algoritmaların Uygulama Alanları

Sezgisel bir teknik olarak Genetik algoritmaların Yöneylem Araştırması problemlerinde kullanımı her geçen gün artmaktadır. Genetik algoritmalar temelde iki

farklı amaç için kullanılmaktadır (Chambers, 2001, s;1-2). Bu amaçlardan ilki bir sistem performansını optimize etmek için parametre seçimidir. Bu gibi sistemler muhtemelen kısıtlar ile birlikte büyük ölçüde sistemi kuran ya da işletenlerin belirlediği karar parametrelerine bağlıdır. Genetik algoritma uygulamalarının çoğunluğu sistemleri optimize etmek veya en azından onları optimuma yakınlaştırmaktır.

Genetik algoritmaların ikinci potansiyel ana kullanım alanı ise sayısal modelleri test etmek ve uygunluğunu sağlamaktır. Literatürde karşımıza daha az çıkan bu uygulamalarda öncelikle açıklayıcı ya da tanımlayıcı bir model kurulur, veri toplanır ve model test edilir. Çelişkiler tespit edildiğine model modifiye edilir. Performans optimizasyonunu hedefleyen algoritmaların aksine burada model ile veriler arasındaki uyumsuzluğun minimizasyonu hedeflenir.

Genetik algoritmaların mühendislik problemlerinde optimizasyon amaçlı olarak kullanılmaya başlanmıştır. Özellikle mekanizma tasarımında çok iyi sonuçlar ortaya konulmuştur. Bunların yanında otomatik programlama, öğrenme kabiliyetli makineler, ekonomi, ekoloji, planlama, üretim hattı yerleşimi gibi çok farklı alanlarda da başarılı uygulamalar yapılmaktadır. Ayrıca dijital resim işleme tekniğinde de çokça uygulama alanı bulmuştur. Bu problemlerin hemen hemen hepsi çok geniş bir çözüm havzasının taranmasını gerektirmektedir. Bu çözüm havzasının geleneksel yöntemlerle taranması çok uzun sürmekte, genetik algoritmayla ise kısa bir sürede kabul edilebilir bir sonuç alınabilmektedir (Kurt ve Cumali, 2001, s;1).

Uygulama alanlarının bazıları aşağıdaki gibi sıralanabilir (Coley, 1999, s;3, Chambers, 2001, s;1-2, Kurt ve Cumali, 2001, s;8-15, Emel ve Taşkın, 2002, s;138, Güden ve diğer., 2005, s;7, Sivanandam ve Deepa, 2008, s;35):

- Birleşim optimizasyon (Combinatorial Optimization) problemleri: Gezgin satıcı problemi, araç rotalama problemi, Çinli postacı problemi, iş atölyesi çizelgeleme problemi, atama problemi, yerleşim tasarımı problemi ve sırt çantası problemi gibi problem uygulamaları
- Bilgisayar ağları tasarımı
- Nonlineer dinamik sistem tahmini, veri analizleri
- Robot yörüngesi planlama
- Görüntü işleme
- LISP programlarının geliştirilmesi

- Stratejik planlama
- Protein moleküllerinin şeklinin bulunması
- Gaz boru hattı uygulamaları
- İmalat planlama, yerleşim planlama ve kaynak ataması
- Mekanik öğrenme,
- Yapay sinir ağlarının tasarımı,
- Sinyal işleme, filtre dizaynı
- Portföy optimizasyonu,
- Hisse senedi fiyatlarındaki değişim kalıplarını tahmini
- Montaj hattı dengeleme problemleri
- Taşıma problemi

3.9. Genetik Algoritmaların Sınıflandırılması

Genetik algoritmalar problem çözümünde adım adım (step by step) bir prosedür ortaya koyar. Etkin bir arama tekniği olarak kendini ispatladığı için birçok işletme, fen bilimleri ve mühendislik konularında problem çözümünde kullanılmıştır. Tüm bu özellikleri araştırmacıların genetik algoritmalara farklı yaklaşımlar geliştirmesine de olanak tanımıştır.

Farklı genetik algoritma yaklaşımları aşağıda açıklanmaktadır (Sivanandam ve Deepa, 2008, s;105, Sumathi ve diğer.,-2008, s;109):

3.9.1. Basit genetik algoritma

Basit genetik algoritmanın mekanizması dizileri kopyalamak ve belirli kısımlarını değiştirmek dışında karmaşık işlemler içermemektedir. Birçok problemin çözümünde başarılı sonuçlar elde eden bu yaklaşım üç işlemde oluşmaktadır:

- Yeniden birleştirme (reproduction)
- Çaprazlama
- Mutasyon

Yeniden birleştirme aşamasında birey dizileri uygunluk değerlerine göre kopyalanır. En sık kullanılan yöntem ruht tekerleği seçimidir. Basit genetik algoritma kullanımının faydalı olduğu durumlar aşağıda sıralanmıştır:

- Arama yüzeyi geniş veya karmaşık ise,

- Alan bilgisi veya uzmanlığın arama uzayını kodlamada yetersiz kaldığı durum mevcut ise,
- Herhangi bir matematiksel analiz mevcut değil ise
- Geleneksel metotlar denemesi başarısız olmuş ise

Basit genetik algoritmanın avantajı keyfi kısıt ve amaçlar ile kolayca kullanılabilmesidir. Bununla birlikte kullanımda bazı dezavantajları mevcuttur:

- Genellikle etkin değillerdir,
- Değerlendirmeler çok zaman almaktadır,
- Çok sayıda problemde popülasyonun büyük tutulması gerekmektedir.

3.9.2. Paralel genetik algoritma

Çeşitli basit genetik algoritmaların paralel olarak uygulanması ‘paralel genetik algoritma’ olarak tanımlanmaktadır. Bu yaklaşım basit genetik algoritmaların geniş bir arama yüzeyinde optimale yakın çözümü bulmak için harcadığı sürenin kısaltılmasına yönelik geliştirilmişlerdir. Genetik algoritmaların paralel olarak çalıştırılabilmesi aşağıdaki elementlere bağlıdır:

- Uygunluğun değerlendirme ve mutasyonun uygulanma biçimi,
- Seçimin yerel yada global uygulanması,
- Tek veya çoklu alt popülasyonların kullanılması,
- Çoklu popülasyonlar kullanılıyor ise bireylerin değiştirilme biçimi.

Probleme alternatif çözümler ortaya koyabilmesi, farklı noktalardan hareketle paralel arama yapılmasını sağlaması, daha hızlı sonuç vermesi ve diğer arama prosedürleri ile kolayca bütünleştirilmesi bu yaklaşımın avantajları olarak sıralanabilir. Paralel genetik algoritma modelinin önemli bir diğer avantajı geleneksel ya da basit genetik algoritmaların erken yakınlaşmasını zorlaştırmasıdır. Birçok araştırmacı tarafından paralel genetik algoritmanın daha gerçekçi bir model olduğu ileri sürülmektedir.

Paralel genetik algoritmalar da uygulama metoduna göre kendi içerisinde bağımsız paralel genetik algoritmalar, göçebe paralel genetik algoritması, bölümlendirme paralel genetik algoritması, segmentasyon paralel genetik algoritması gibi kısımlara ayrılmaktadır.

3.9.3. Hibrit genetik algoritma

Hibrit genetik algoritma çaprazlama tarafından üretilen yeni bireylerin gelişimi için sezgiselliğin kullanımını sağlamaya dönük geliştirilmiştir. Bir sezgisellik başlangıç popülasyonu oluşturmada, diğer ikisi de ya çaprazlama veya karıştırmadan gelen bireylere uygulanmaktadır. Son iki sezgisellik ise loksl minimuma takılınmasını önlemede kullanılmaktadır.

Hibrit genetik algoritması adımları aşağıda sıralanmaktadır:

Adım 1:

- BaşlangıçSezgisellik algoritmasını kullanarak ilk popülasyon kısmını oluştur,
- Popülasyonun kalanını tesadüfi olarak belirle

Adım 2:

- Başlangıç popülasyonundaki her bir yerleşim için RemoveSharp algoritmasını uygula
- Başlangıç popülasyonundaki her bir yerleşim için LocalOpt algoritmasını uygula

Adım 3:

- Tesadüfi olarak iki ebeveyn seç,
- Seçilen ebeveynlere çaprazlama uygula ve yeni birey oluştur,
- Yeni bireye RemoveSharp algoritmasını uygula,
- Yeni bireye LocalOpt algoritmasını uygula,
- Yeni bireyin uygunluğu herhangi bir ebeveyninden büyük ise onunla yeni bireyi değiştir.

Adım 4: Popülasyondan seçilen herhangi bir bireyi mutasyona tabi tut,

Adım 5: Adım 3 ve 4'ü belirlenen iterasyon sayısınca tekrar et.

3.9.4. Adaptif genetik algoritma

Popülasyon büyüklüğü, çaprazlama olasılığı veya mutasyon olasılığı genetik algoritmanın çalıştırılması sırasında değişen genetik algoritmalar adaptif genetik algoritmalar olarak adlandırılmaktadır.

Çok optimumlu problemlerin optimizasyonunda genetik algoritmaların iki önemli karakteristiğinin olması zaruridir. Bu özelliklerin ilki optimuma yakınsama

kapasitesi; ikincisi ise çözüm uzayında global optimumu ararken yeni bölgeleri keşfetme kapasitesidir. Bu iki karakteristik arasındaki denge mutasyon olasılığı, çaprazlama olasılığı ve kullanılan çaprazlama yöntemi tarafından belirlenmektedir.

Adaptif genetik algoritma yaklaşımında keşif ve sömürü (exploitation) arasındaki dönüşüm, çözümün uygunluk değerine göre mutasyon olasılığı ve çaprazlama olasılığının değiştirilmesi ile yapılmaktadır. Popülasyonun yerel minimuma takılma eğilimi ortaya çıktığında her iki olasılık da artırılmakta; çözüm uzayına yayılımı artınca da azaltılmaktadır.

3.9.5. Hızlı dağınık (messy) genetik algoritma

Dağınık genetik algoritma, ikili, stokastik, değişken dizi uzunluklu ve popülasyon temelli bir problem çözme yaklaşımıdır. Diğer yaklaşımlardan en önemli farkı, en iyi çözüme ulaşmak için genetik materyalin yapı bloklarını (building blocks) açıkça manipüle edebilmesidir. Dağınık genetik algoritma üç işlem aşaması içermektedir:

- Başlangıç aşaması,
- İlkel (primordial) aşama,
- Bitişiklik durumu (juxtapositional) aşaması.

Başlangıç aşamasında, bir sonraki blok filtreleme aşamasında ortaya çıkabilecek gürültünün (noise) üstesinden gelebilmek için yeter derecede büyük bir popülasyon elde etmek için popülasyon boyutu belirleme denklemi kullanılır. Popülasyon büyüklüğü belirlendikten sonra başlangıç popülasyonu üyeleri tesadüfi olarak belirlenir ve seçilen uygulamaya göre onların uygunluk değerleri hesaplanır. Yapı bloğu filtreleme şeması kullanılarak başlangıç aşamasında belirlenen bu popülasyon üyeleri sistematik olarak, kullanıcı tarafından belirlenen yapı bloğu büyüklüğüne düşürülür. Yapı bloğu filtrelemede her bir popülasyon üyesinin belirli sayıda bit'i silinir. Bir sonraki jenerasyon için en iyi kısmi dizilerin saklanabilmesi için bu silme işlemini turnuva seçimi takip eder.

Son aşamada ise filtreleme aşamasında gelen yapı blokları alınır ve bir 'kes ve dilimle' (cut-and-slice) operatörü ile birleştirilir. Bu prosesi de turnuva seçimi izlemektedir.

DÖRDÜNCÜ BÖLÜM

ACİL SERVİS İSTASYONU YERLEŞİM MODELLERİ

Özel sektör veya devlete ait her işletmeye ait tesisin kuruluşunda çeşitli kriterleri dikkate almayı gerektiren bir yerleşim yeri sorunu ile ilgili bir çalışma yapılması gerekmektedir. Ürünlerin verimli biçimde üretilmesi ve pazarlanması, hizmetlerin etkin biçimde hedef kitleye ulaştırılması tesis yerinin belirlenmesi ile yakından ilgili bir konudur.

Tesis yeri seçimi kararı birçok kamu ve özel sektör kuruluşu için stratejik planlamada kritik bir öneme sahiptir. Bir perakende zinciri yeni bir satış noktası seçerken, bir imalatçı firma depo yeri belirlerken veya bir şehir plancısı itfaiye istasyonu veya acil servis ünitesinin yerini tayin ederken çoğunlukla zor bir uzamsal kaynak tahsis kararı vermek durumunda kalmaktadır. Yerleştirme problemi özellikle Hakimi tarafından 1964 yılında yapılan otoyol sisteminde haberleşme ve polis istasyonlarının yerleştirilmesi çalışması ile öne çıkmıştır. Bu çalışmadan sonra problemin çözümüne yönelik olarak çok sayıda model geliştirilmiştir.

4.1. Ambulans Yerleşim Modelleri

Ambulans yerleşim modelleri yöneylem araştırmasında Taleple Transport yada Taşıma (Transportation on Demand) ana başlığı altında incelenmektedir. Yolcu veya malların kullanıcıların talebi üzerine belirli bir kaynaktan belirli bir hedefe taşınması bu başlığın içeriğini oluşturmaktadır. Bu sistemlerde kullanıcılar tarafından şekillendirilen taşıma talebi donanımlı araçlarla ortak hizmet sağlanarak karşılanmaktadır (Barnhart ve Laporte, 2007, s;429).

Talebe Göre Transport sistemleri statik ve dinamik olabilir. Talebin önceden bilindiği durumlarda statik; dinamik olarak değiştiği ve talebi karşılamak için araç rotasının gerçek zamanlı olarak değiştirilmesinin gerektiği durumlarda da dinamiklik söz konusu olmaktadır. Uygulamada dinamik problemler statik alt-problemler dizisi olarak ele alınmaktadır. Yeni bir talep geldiğinde mevcut çözüm yeniden optimize edilir.

Talebe göre transport problemleri birbiri ile çelişen üç amacın varlığı ile karakterize edilir: karşılanan talep sayısının maksimize edilmesi, işletme maliyetlerinin minimize edilmesi ve kullanıcı memnuniyetsizliğini minimize etmek. Bu amaçlar

arasındaki denge birçok durumda kapasiteye göre kabul edilebilecek talep sayısının maksimize edilmesi, ve belirlenen bir hizmet kalitesi kısıtı altında işletme maliyetinin minimize edilmesi ile sağlanmaktadır. Hizmet kalitesi ise genellikle istenen teslimat sayısı ve teslim süresinde yaşanan sapmalar ve insan taşınması durumunda ulaşım süresinin beklenenden uzun olması ile ölçülmektedir. İşletme maliyetleri de kullanılan araç sayısı, rotanın toplam süresi ve araçlar tarafından katedilen mesafe ile değişmektedir.

Talebe göre transport/taşıma problemlerinin bir diğer özelliği zaman boyutunun bulunmasıdır. Teslim alma ve dağıtım tanımlanan belirli bir zaman dilimi içerisinde sınırlandırılmaktadır. Özellikle insan taşınmasında bu zaman penceresi çok dar olmakta ve yolda geçen sürenin kısalığı bir performans kriteri olarak öne çıkmaktadır. Acil durum araçları için bu durum daha da bir önem arz etmektedir. Birçok durumda yolcular var iken araçların beklemesi önlenmektedir.

Talebe göre transport/taşıma sistemleri üç ana konu ile ilgili karar almayı kapsamaktadır: talep kümeleme, araç rotalama ve araç programlama. Talep kümeleme, taleplerin mesafe ve zamanla ilgili yakınlıklarına göre gruplandırılmasıdır. Belirlenen bu gruplara göre her bir araç tarafından ziyaret edilmesi gereken birbiri ile ilintili teslim alma ve dağıtım yerlerinin sırası ile ilgili kararlar araç rotalamayı oluşturmaktadır. Son olarak da araç programlama ile her bir yerin ziyaret edilmesi gereken kesin zaman belirlenmektedir. Tüm bu kararlar açık ve sıkı biçimde iç içe geçmiş durumdadır ve sistemin doğru yönetimi hepsinin eş zamanlı olarak optimize edilmesini gerektirmektedir.

Yöneylem araştırması literatüründe talebe göre transport/taşıma konusunda çok sayıda çalışma yapılmıştır. Bu çalışmaların çoğu ‘Teslim alma ve dağıtım araç rotalama problemi’nin (vehicle routing problem with pickup and delivery) genelleştirilmesi üzerine yapılmıştır. Acil servis araçları yönlendirimi problemleri çağrıyla dağıtım, kurye dağıtım, çağrı ile uçuş dağıtım yanında bu problemin dört ana uygulama alanından birini oluşturmaktadır. Diğerlerinden farklı olarak bu sistemlerde önem rotalama seçeneklerinden ziyade yerleşim konusundadır (Barnhart ve Laporte, 2007, s.431).

Talebe göre transport/taşıma problemi acil servis ile ilgili üç konuda öne çıkmaktadır: İtfaiye, polis ve ambulans hizmetleri. Yerleşim, personel sağlama ve araç yönlendirimi gibi çok farklı konular bu problemlerin temelini oluşturmaktadır.

Ambulans filo yönetiminde öne çıkan anahtar konulardan biri nüfusu uygun bir oranda kapsayacak şekilde ambulansların nerelere yerleştirileceğine kadar verilmesidir (Barnhart ve Laporte, 2007, s;454)

Ambulans yerleşimi ile ilgili ileri sürülen modeller son otuz-kırk yıl içerisinde önemli bir değişim geçirmiştir. Bu süre zarfında, bilgisayar teknolojisi yanında modelleme ve algoritmik karmaşıklık, matematiksel programlama çözümlerinin ve çeşitli karar verme seviyelerinde bilgisayar yazılımlarının yaygın olarak kullanımında önemli değişimler yaşanmıştır. Ambulans yerleşimi ile ilgili literatür incelendiğinde de bu değişimin açık biçimde görülmektedir. Önerilen ilk modeller basit tamsayılı doğrusal programlama formasyonları iken, zamanla daha gerçekçi özellikler aşama aşama modellere eklenmiş ve çözüm teknikleri daha karmaşıklılaşarak dönüşmüştür (Brotcorne ve diğer., 2003, s;451, Barnhart ve Laporte, 2007, s;454, Owen ve Daskin, 1998, s;425).

Literatürdeki acil servis uygulamaları ile ilgili modeller yapılarına göre iki ana kategori altında incelenmektedir: Deterministik Modeller ve Olasılıklı Modeller. Planlama aşamasında kullanılan deterministik modeller ambulansın hazır bulunma durumu ile ilgili stokastik etkenleri ihmal etmektedir. Olasılıklı modeller ise ambulansın bir kuyruk sisteminde servis sağlayıcı gibi çalıştığını ve her durumda çağrılara cevap veremeyeceğini yansıtmaya çalışır. Bu modellerin dışında gün boyunca ambulansın defalarca yer değiştirmesini yansıtarak daha iyi kapsama sağlamayı amaçlayan dinamik modeller de geliştirilmiştir (Brotcorne ve diğer., 2003, s, 451, Owen ve Daskin, 1998, s;424-425).

Planlamacılara veya sistem yöneticilerine acil yardım araçlarının yerleşimi kararında yardımcı olmak üzere geliştirilmiş olan ve optimum veya optimuma yakın araç yerleşimini sağlayan modeller amaçlarına göre de iki büyük gruba ayrılabilir: Kapsama temelli (covering-based) modeller ve sistem genelinde ortalama ulaşma mesafesini (veya zamanını) enküçüklemeyi amaçlayan p-medyan modelleridir (Daskin ve diğer., 1988, s;19 Selim ve Özkarahan, 2003, s;19). Burada kapsama; bir aracın belirlenmiş bir zaman veya mesafe standardı içerisinde belli bir bölgedeki acil yardım çağrılarına ulaşabilmesi anlamına gelmektedir. Söz konusu standartlar acil servis

hizmetlerinin etkin bir şekilde yerine getirilmesini sağlamak için yaygın olarak kullanılmaktadır. Son yirmi yıl içinde kapsamayı araç yerleşiminde etkinlik ölçütü olarak kullanan çok sayıda acil servis modeli geliştirilmiştir.

4.1.1. Statik Ambulans Yerleşim Modelleri

Ambulans yerleşim modelleri genellikle $G = (V \cup W, A)$ gösterimi ile tanımlanır. Burada V , bir araya toplanan talep noktalarının bir düğüm seti; W ise potansiyel ambulans yerleşim yerlerini göstermektedir. $A = \{(i, j) \in V \cup W\}$ de bir yay setidir. Her bir (i, j) , t_{ij} seyahat süresi ile ilişkilidir. Eğer $t_{ij} \leq r$ ise herhangi bir talep noktası $i \in V$, $j \in W$ bölgesi içerisinde kapsanır. Burada r , önceden belirlenmiş bir kapsama standardıdır. $W_i = \{j \in W : t_{ij} \leq r\}$, i talep noktasını kapsayan yerleşim yeri setini göstermektedir. i noktasından j noktasına olan en kısa seyahat süresini gösteren t_{ij} 'nin de bilindiği kabul edilmektedir.

p-medyan modeli: Tesis yerleşiminin etkinliğini ölçmenin bir yolu kat edilen ortalama mesafenin belirlenmesidir. Bu mesafe arttığında tesisin ulaşılabilirliği azalır ve dolayısıyla etkin bir yer seçiminden söz edilemez. Hakimi tarafından ortaya atılan p-medyan modeli de bu etkinlik ölçüsünü kullanır. Talebin hizmet kalitesine duyarlı olmadığı bu modelde talep merkezleri ve tesisler arasında ağırlıklandırılmış talebe göre mesafe minimize edilir. Basitleştirilmiş modellerde yalnızca düğüm noktaları potansiyel kurulum yeri olarak alınır ve bu durum hesaplamaları önemli ölçüde kolaylaştırır. Bir dağıtım ağı gibi binlerce sayıda düğümün olduğu modellerde potansiyel çözüm P ve düğüm sayısı N olduğunda muhtemel yerleşim konfigürasyonu sayısı $\binom{N}{P}$ kadar olmaktadır. Bu durumun karmaşıklığı daha sofistike algoritmaların geliştirilmesine yol açmıştır (Owen ve Daskin, 1998, s:426). P-medyan modelinde yerleşim için sabit maliyetler ve taleple ilgili limitler dikkate alınmaz.

Toregas ve arkadaşları tarafından 1971 yılında tüm talep noktalarını kapsamak için gerekli olan ambulans sayısını minimize etmeyi amaçlayan Yerleşim Seti Kapsama Yerleşim Modeli (Location Set Coverage Model, LSCM) önerilmiştir. İkili (binary) değişken kullanan bu modelde x_j , ambulansın j 'de konumlandığı durumda 1 değerini alır.

$$(LCSM) \text{ minimize } \sum_{j \in W} x_j$$

Kısıtlar

$$\sum_{j \in W_i} x_j \geq 1 \quad (i \in V)$$

$$x_j \in \{0,1\} \quad (j \in W)$$

Bu model gerçek yaşamdaki birçok durumu dikkate almamaktadır. Bunların en önemlisi ambulansın sevk edilmesinden sonra artık bazı talep noktalarının kapsanmıyor olmasıdır. Bu model ayrıca en fazla arz bölgesi sayısı kadar (W) sayıda ambulans olacağını varsaymaktadır. Ayrıca nüfus ve yol durumu da bu modelde ihmal edilmektedir ancak gerçekte bu durum farklı olabilmektedir. Bununla birlikte bu model tam bir kapsamayı sağlamak için ambulans sayısında daha düşük bir sınır sağlamaktadır (Daskin ve Dean, 2004, s;47, Brotcorne ve diğer., 2003, s;453).

Yerleşim Seti Kapsama Yerleşim Modelinin (LSCM) eksikliklerini gidermek amacı ile önerilen modellerden bir diğeri sınırlı sayıda ambulans ile kapsanacak nüfusun maksimize edilmesidir. Church ve ReVelle tarafından 1974 yılında önerilen Maksimum Kapsama Yerleşim Modelinde (Maximal Covering Location Model, MCLM) d_i , i noktasından gelen talebi, p ise eldeki ambulans sayısını göstermektedir. y_i de ikili (binary) bir değişken olup i noktasının en az bir ambulans tarafından kapsandığı durumda 1 değerini almaktadır. Bu model,

$$(MCLP) \text{ minimize } \sum_{i \in V} d_i y_i$$

Kısıtlar

$$\sum_{j \in W_i} x_j \geq y_i \quad (i \in V),$$

$$\sum_{j \in W} x_j = p,$$

$$x_j \in \{0,1\} \quad (j \in W),$$

$$y_i \in \{0,1\} \quad (i \in V)$$

LCSM ve MCLP modellerinin her birinin kullanımının yararlı olduğu durumlar söz konusudur. LCSM, tüm talebi karşılamak için doğru araç sayısını belirlemek için bir planlama aracı olarak kullanılabilirken, MCLP, eldeki kaynakların mümkün olan en

iyi kullanımını sağlamayı amaçlamaktadır. İlgili literatürde her iki modeli temel alan çok sayıda çalışma yapılmıştır.

4.1.2. Deterministik Statik Ambulans Yerleşim Modelleri

Yerleşim Seti Kapsama Yerleşim Modeli (LSCM) ve Maksimum Kapsama Yerleşim Modeli (MCLP) olay yerine farklı tipteki araçların gönderimini kapsamaz. Birden çok araç tipini kullanmak üzere geliştirilen modellerden ilki tandem araç yerleşim modeli (TEAM) dir. İlk olarak Schilling ve arkadaşları tarafından geliştirilen bu model pompacı ve kurtarma merdivencilerinin gibi iki farklı ekipmanın söz konusu olduğu itfaiye departmanını konu almıştır ancak temel yaşam destek (basic life support, BLS) ve ileri yaşam destek (advanced life support, ALS) birimlerinin kullanıldığı ambulans yerleşimi konusunda da uygulanmıştır. Bu modelde p^A ve p^B , sırasıyla A ve B tipi araç sayılarını göstermektedir. r^A ve r^B de her bir araç tipi için kapsama standardını gösterirken $W_i^A = \{j \in W : t_{ij} \leq r^A\}$ ve $W_i^B = \{j \in W : t_{ij} \leq r^B\}$ olarak tanımlanır. $x_j^A(x_j^B)$ ikili (binary) bir değişken olup i düğümünde A(B) tipi araç olduğu durumda 1 değerini alır. y_i de ikili bir değişken olup $i \in V$ noktasının yalnızca her iki tipteki araç tarafından kapsanması durumunda 1 değerini almaktadır. Bu tanımlara göre TEAM modeli aşağıdaki biçimde gösterilebilir:

$$(\text{TEAM}) \text{ maksimize } \sum_{i \in V} d_i y_i$$

Kısıtlar

$$\sum_{j \in W_i^A} x_j^A \geq y_i \quad (i \in V)$$

$$\sum_{j \in W_i^B} x_j^B \geq y_i \quad (i \in V)$$

$$\sum_{j \in W} x_j^A = p^A ,$$

$$\sum_{j \in W} x_j^B = p^B ,$$

$$x_j^A \leq x_j^B \quad (j \in W)$$

$$x_j^A, x_j^B \in \{0,1\} \quad (j \in W)$$

$$y_i \in \{0,1\} \quad (i \in V)$$

Bu model iki farklı araç tipi arasında bir hiyerarşi sunan son kısıt dışında MCLP modelinin bir uzantısıdır. Ancak aracın meşgul olduğu durumda tüm bu modellerde kapsama eksik kalmaktadır. Tek araç tipi olması durumunda toplam araç sayısını artırmadan daha iyi bir kapsama sağlamak için MCLP modelini modifiye eden çeşitli stratejiler geliştirilmiştir. Daskin ve Stern tarafından (1981) birden çok defa kapsanan talep noktalarının sayısını maksimize etmek için hiyerarşik bir amaç fonksiyonu daha modele eklenmiştir. Hogan ve ReVelle (1986) tarafından önerilen modelde ise iki defa kapsanan toplam talep maksimize edilmiştir.

Hogan ve ReVelle (1986) BACOP1 ve BACOP2 olarak adlandırdıkları yedek kapsama formülasyonlu iki model önermişlerdir. Bu modellerde $i \in V$ talep noktasının bir r kapsama standardında bir ambulans tarafından kapsanması durumunda binary bir değişken olan y_i , 1 değerini almakta; yine ikili bir değişken olan u_i de i talep noktasının iki kez kapsanması durumunda 1 değerini almaktadır. Bu iki model de aşağıda gösterilmiştir:

$$\text{(BACOP1) maksimize } \sum_{i \in V} d_i y_i$$

Kısıtlar

$$\sum_{j \in W_i} x_j - u_i \geq 1 \quad (i \in V),$$

$$\sum_{j \in W} x_j = p,$$

$$0 \leq u_i \leq 1 \quad (i \in V)$$

$$x_j \geq 0 \quad (i \in V), \text{ ve ikinci model}$$

$$\text{(BACOP2) maksimize } \theta \sum_{i \in V} d_i y_i + (1 - \theta) \sum_{i \in V} d_i u_i$$

Kısıtlar

$$\sum_{j \in W_i} x_j - y_i - u_i \geq 0 \quad (i \in V),$$

$$u_i - y_i \leq 0 \quad (i \in V),$$

$$\sum_{j \in W} x_j = p,$$

$$0 \leq u_i \leq 1 \quad (i \in V),$$

$$0 \leq y_i \leq 1 \quad (i \in V),$$

$$x_j \geq 0 \quad (i \in V)$$

Bu modelde kullanılan θ , $[0,1]$ aralığında seçilen bir ağırlıktır.

Gendreau ve arkadaşları (1997) tarafından önerilen modelde $r_1 < r_2$ olmak üzere iki kapsama standardı kullanılmıştır. Modelde tüm talebin r_2 zaman aralığında karşılanması gerekmekte ve belirlenen bir α oranında talep r_1 zaman aralığında karşılanabilmelidir. Bu oran da muhtemelen r_2 aralığına tesadüf etmektedir. 1973 tarihli Amerikan Acil Servis Kanunu r_1 'i $\alpha = 0.95$ aralığında 10 olarak belirlemiştir.

Gendreau, Laporte ve Semet tarafından geliştirilen ikili standart modelde (double Standard model, DSM) ise p sayıda ambulans kullanılarak ikili kapsama kısıtları altında r_1 standardında iki kez kapsanan talebin maksimizasyonu amaçlanmaktadır.

DSM modelinde $W_i^1 = \{j \in W : t_{ij} \leq r_1\}$ ve $W_i^2 = \{j \in W : t_{ij} \leq r_2\}$, y_i de $j \in W$ de yerleşik ambulans sayısını gösteren tamsayıli değişkendir. İkili değişken olan x_i^k , $i \in V$ noktasında talebin r_1 süresi içinde k kez ($k=1$ veya 2) kapsanması durumunda 1 değerini almaktadır. Model (Gendreau ve diğer., 1997, s;77):

$$(DSM) \text{ maksimize } \sum_{i \in V} d_i x_i^2$$

Kısıtlar

$$\sum_{j \in W_i^2} y_j \geq 1 \quad (i \in V),$$

$$\sum_{j \in V} d_j x_j^1 \geq \alpha \sum_{i \in V} d_i,$$

$$\sum_{j \in W_i^1} y_j \geq x_i^1 + x_i^2 \quad (i \in V),$$

$$x_i^2 \leq x_i^1 \quad (i \in V),$$

$$\sum_{j \in W} y_j = p,$$

$$y_j \leq p_j \quad (j \in W),$$

$$x_i^1, x_i^2 \in \{0,1\} \quad (i \in V),$$

$$y_j, \text{ tamsayı } (j \in W)$$

Burada amaç fonksiyonu r_i süresi içerisinde iki defa kapsanan talebi hesaplamaktadır. İlk iki kriter ikili kapsama kriterlerini ifade etmektedir. Üçüncü kriterin sol tarafı r_i süresi içerisinde i noktasını kapsayan ambulans sayısını, sağ tarafı ise i noktası r_i süresinde kapsanmış ise 1, en azından iki kez kapsanmış ise 2 değerini almaktadır. İki ve üçüncü kriterin birleşimi α oranında talebin kapsanmasını ve kapsama standardının da r_i olmasını sağlamaktadır. Dördüncü kriter ise i noktasının bir kez kapsanmadan iki kez kapsanamayacağını ifade etmektedir. Altıncı kriterde $(y_j \leq p_j) p_j, 2$ olarak setlenebilir çünkü bu değeri kullanan optimal bir çözüm daima mevcuttur.

Selim ve Özkarahan (2003) de acil servis sistemlerinin yerleşiminin belirlenmesinde Maksimum yedek kapsama modeli ve sınırlı kapasiteli maksimum kapsama modelini temel alarak doğrusal ve deterministik bir model önermişlerdir.

4.1.3. Probabilistik Statik Ambulans Yerleşim Modelleri

Ambulans yerleşiminde ilk probabilistik modellerinden biri maksimum beklenen kapsama yerleşim problemi formülasyonudur (maximum expected covering location problem formulation, MEXCLP). Bu modelde tüm ambulansların birbirinden bağımsız olduğu ve aynı q meşgul olma oranına sahip olduğu kabul edilmektedir. q meşgul olma oranı tüm talep noktaları için gelen çağrılarının toplam beklenen süresinin eldeki toplam ambulans sayısına bölünmesi ile hesaplanmaktadır. Böylece $i \in V$ noktası k ambulans tarafından kapsanıyorsa ilgili beklenen kapsanan talep $E_k = d_i(1 - q^k)$ formülü ile hesaplanmaktadır. k . Ambulansın marjinal katkısı ise $E_k - E_{k-1} = d_i(1 - q^k)q^{k-1}$ dir. MEXCLP modelinde en fazla p ambulans yerleştirilebilir ancak bir noktaya birden çok ambulans yerleştirilebilmektedir. $i \in V$ noktası en azından k ambulans tarafından kapsandığı durumda y_{ik} değişkeni 1 değerini almaktadır. Model aşağıdaki gibi ifade edilebilir (Daskin, 1983, s;53):

$$(MEXCLP) \text{ Maksimize } \sum_{i \in V} \sum_{k=1}^p d_i (1 - q) q^{k-1} y_{ik}$$

Kısıtlar

$$\sum_{j \in W_i} x_j \geq \sum_{k=1}^p y_{ik}$$

$$\sum_{j \in W} x_j \leq p,$$

x_j , tamsayı ($j \in W$),

$y_{ik} \in \{0,1\}$ ($i \in V, k = 1, \dots, p$)

Modelin geçerliliği amaç fonksiyonun konkav olmasına bağlıdır. Bundan dolayı eğer $y_{ik} = 1$ ise $h \leq k$ için $y_{ih} = 1$ dir. amaç fonksiyonu maksimize etmek gerektiğinden her iki kısıtın da eşitlik olarak sağlanması gerekmektedir. Bunun yanında $i \in V$ noktasını kapsayan ambulans sayısı eşitliğin her iki tarafında aynı olacaktır.

MEXCLP modelinin Fujiwara ve arkadaşları tarafından Bangkok şehri ambulans yerleşim sistemi üzerinde yaptığı uygulamasında ambulans sayısının yakın beklenen kapsama süresi ve cevap süresinde 21'den 15'e düşürülebileceği tespit edilmiştir (Fujiwara ve diğerleri-1987, s.9). Çalışmada talep noktası $V=59$, ambulans yerleşim yeri sayısı $W=46$ ve ambulans sayısı $10 \leq p \leq 30$ olarak alınmıştır.

Repede ve Bernardo tarafından Lousville,Kentucky için yapılan ve TIMEXCLP olarak adlandırılan çalışmada gün boyunca değişen seyahat hızları da modele ilave edilmiştir. Sonuç olarak on dakika içinde ulaşılan vaka oranının % 84'ten % 95'e çıktığı ve cevap süresinin de % 36 azaldığı belirtilmiştir (Repede ve Bernardo, 1994, s;567)

Goldberg ve diğerleri tarafından yapılan farklı bir MEXCLP uygulamasında stokastik seyahat süresi hesaba katılmıştır. Bu modelde amaç 8 dakikalık süre içerisinde beklenen çağrı sayısının maksimize edilmesidir. Potansiyel yerleşim yerleri de bir tercih sırasına göre sıralanmışlardır. Vakaya ulaşma süresi ile ilgili olasılık da aşağıdaki üç olasılığa göre hesaplanmıştır (Goldberg ve diğer., 1990, s;308-312) : 1) k . seçilen bölgede bulunan ambulansın vakaya sekiz dakika içerisinde ulaşabilme olasılığı; 2) bu ambulansın o bölgede bulunuyor olma olasılığı; 3) daha az tercih edilen bir bölge olan $k-1$. bölgede yerleştirilen ambulansın mevcut olmama olasılığı. Çalışma sonucunda belirlenen süre içerisinde % 1 oranında beklenen kapsanan çağrı sayısı artmıştır.

ReVelle ve Hogan tarafından verilen bir α olasılığında kapsanan talebi maksimize etmek için iki ayrı olasılıklı model önerilmiştir. Maksimum hazır bulunma

yerleşim modelinin (maximum availability location problem, MALP I ve MALP II) ilkinde (MALP I) q meşguliyet fonksiyonunun tüm potansiyel yerleşim yerleri için aynı olduğu kabul edilmiştir. Her bir i talep noktası için α güvenlik seviyesinde gerekli minimum ambulans sayısı aşağıdaki kısıt ile belirlenmektedir:

$$1 - q^{\sum_{j \in W_i} x_j} \geq \alpha \quad (i \in V)$$

Bu kısıt ise aşağıdaki gibi doğrusallaştırılmıştır.

$$\sum_{j \in W_i} x_j \geq [\log(1 - \alpha) / \log q] = b \quad (i \in V) \quad \text{MALP I modeli ise aşağıdaki gibi}$$

formüle edilmiştir (ReVelle ve Hogan-1989, s194):

$$\text{(MALP I) maksimize } \sum_{i \in V} d_i y_{ib}$$

Kısıtlar

$$\sum_{k=1}^b y_{ik} \leq \sum_{j \in W_i} x_j \quad (i \in V),$$

$$y_{ik} \leq y_{i,k-1} \quad (i \in V, k = 2, \dots, b),$$

$$\sum_{j \in W} x_j = p,$$

$$x_j \in \{0,1\} \quad (j \in W),$$

$$y_{i,k} \in \{0,1\} \quad (i \in V, k = 1, \dots, p)$$

Burada $y_{ik} \leq y_{i,k-1}$ kriteri MEXCLP modelindeki konkavlık özelliğine artık gereksinim bulunmadığından eklenmiştir.

MALP II modelinde, meşguliyet oranının tüm alanlar için aynı olması kuralı gevşetilmiştir. Bunun yerine her bir $i \in V$ için bir q_i meşguliyet oranının tahminini hesaplamışlardır. Bu meşguliyet oranı da i ile ilgili tüm çağrılarının toplam süresinin W_i 'deki tüm ambulansların toplam mevcudiyetine oranıdır. Bu değer bir alt sınırdır çünkü W_i 'deki bazı ambulanslar i ile ilgili olmayan çağrılar için sevk edilmiş olabilmektedir ancak yine de geçerli aşırıya kaçmayan bir modeldir. MALP II de b 'nin yerine her bir $i \in V$ için b_i hesaplanır. Bu çalışmada ReVelle ve Hogan her bir $j \in W$ 'ye özgü bir q_j meşguliyet oranı ile çalışmanın zorluğunu vurgulamıştır. Çünkü bu değerler modelin çıktılarıdır ve önceden bilinemezler. Bununla birlikte çeşitli çalışmalarda verilen bir ambulans yerleşim planı ve talep seviyesi ile analitik birtakım yöntemler ve

simülasyon kullanılarak olasılıklar da hesaplanabilmektedir. (Brotcorne ve diğer., 2003, s;457)

Ball ve Lin (1993), Rel-P olarak adlandırılan LSCM modelinin bir başka versiyonunu geliştirmişlerdir. Bu modele verilen güvenlik seviyesini sağlamak için gerekli araç sayısı ile ilgili doğrusal bir kısıt ilave edilmiştir. Modele eklenen x_{jk} ikili değişkeni $j \in W$ noktasında k ambulansın yerleşmesi durumunda 1, ve c_{jk} sabiti ise k sayıda aracın j bölgesine yerleştirilmesinin maliyetidir. Üst sınır p_j , j bölgesinde yerleşecek ambulans sayısını sınırlamaktadır. Model aşağıda gösterilmiştir:

$$\text{(Rel-P) Minimize } \sum_{j \in J} \sum_{1 \leq k \leq p_j} c_{jk} x_{jk}$$

Kısıtlar

$$\sum_{1 \leq k \leq p_j} x_{jk} \leq 1 \quad (j \in W)$$

$$\sum_{j \in W_i} \sum_{1 \leq k \leq p_j} a_{jk} x_{jk} \geq b_i \quad (i \in V)$$

$$x_{jk} \in \{0,1\} \quad (j \in W, 1 \leq k \leq p_j)$$

Modelde $\sum_{j \in W_i} \sum_{1 \leq k \leq p_j} a_{jk} x_{jk} \geq b_i$ kısıtında a_{jk} ve b_i sabitleri belirlenen ambulans

sayısının i noktasının talebini kapsamasını temin etmek için çağrıya cevap verememe olasılığı belli bir değeri aşmamaktadır. a_{jk} ve b_i katsayılarının hesaplanması gerçekte bu olasılıkla ilgili bir üst sınır kullanarak yapılmıştır. (Ball ve Lin, 1993, s;24).

Mandell de çalışmasında iki farklı ünitenin (ALS ve BLS) kullanıldığı iki kademeli bir yöntem önermektedir. İleri sürülen modelde ALS ünitelerinin BLS nin işlerini yapabildiği varsayılmıştır. $i \in V$ noktasından çağrı gelmesi ve buna uygun cevap verilmesi olasılığı ALS ünitesinin i noktasına r^A seyahat süresinde h sayısına, i noktasına r^B süresinde ulaşan k sayıdaki ALS sayısına ve i noktasına r^B de ulaşan l sayıdaki BLS ünitesinin sayısına bağlıdır. Burada $r^A \geq r^B$ olarak alınmıştır (Mandell-1998, s.359). kuyruk modeli kullanılarak modelle ilgili θ_{ihkl} olasılığı hesaplanmıştır. Problem p^A sayıda ALS ünitesinin ve p^B sayıda BLS ünitesinin W sayıdaki noktaya yerleştirilmesidir. x_j^A ve x_j^B sırasıyla ALS ve BLS ünite sayıları olarak alınmıştır. Yalnızca h sayıda ALS ünitesinin r^A süresinde i 'de; k sayıda ALS ünitesinin

r^B süresinde i'de; 1 sayıda BLS ünitesinin r^B süresinde i'de yerleşmesi durumunda 1 değeri alan bir y_{ihkl} ikili değişkeni de tanımlanmıştır. $W_i^A = \{j \in W : t_{ij} \leq r^A\}$ ve $W_i^B = \{j \in W : t_{ij} \leq r^B\}$ olarak tanımlanmıştır. y_{ihkl} 'nin de değişim aralığı h için 0 ile $h_i = \min\{p^A, |W_i^A|\}$; k için 0 ile $k_i = \min\{h, p^A, |W_i^B|\}$ ve l için 0 ile $l_i = \min\{p^B, |W_i^B|\}$ arasında değişmektedir. Beklenen kapsanan talebi maksimize etmeyi amaçlayan Mandell tarafından önerilen iki aşamalı model (TTM) aşağıda gösterilmiştir:

$$(TTM) \text{ maksimize } \sum_{i \in V} \sum_{h=1}^{h_i} \sum_{k=0}^{k_i} \sum_{l=0}^{l_i} d_i \theta_{ihkl} y_{ihkl}$$

Kısıtlar

$$\sum_{h=1}^{h_i} h \sum_{k=0}^{k_i} \sum_{l=0}^{l_i} y_{ihkl} \leq \sum_{j \in W_i^A} x_j^A \quad (i \in V)$$

$$\sum_{k=1}^{k_i} k \sum_{h=k}^{h_i} \sum_{l=l_0}^{l_i} y_{ihkl} \leq \sum_{j \in W_i^B} x_j^B \quad (i \in V)$$

$$\sum_{l=1}^{l_i} l \sum_{h=1}^{h_i} \sum_{k=0}^{k_i} y_{ihkl} \leq \sum_{j \in W_i^B} x_j^B \quad (i \in V)$$

$$\sum_{h=1}^{h_i} \sum_{k=0}^{k_i} \sum_{l=0}^{l_i} y_{ihkl} \leq 1 \quad (i \in V)$$

$$\sum_{j \in W} x_j^A \leq p^A$$

$$\sum_{j \in W} x_j^B \leq p^B$$

$$y_{ihkl} \in \{0,1\} \\ (i \in V, 0 \leq h \leq h_i, 0 \leq k \leq k_i, 0 \leq l \leq l_i)$$

$$x_j^A, x_j^B \in \{0,1\} \quad (j \in W)$$

Burada eğer $k = 0$ ise $l_0=1$ ve $k > 0$ ise $l_0=0$ değerini almaktadır. Modelde ilk üç kısıt y_{ihkl} değişkeni tarafından alınan değerlerin yerleştirilen ALS ve BLS ünitesi sayısı ile uyumlu olmasını sağlamaktadır.

4.1.4. Dinamik Ambulans Yerleşim Modelleri

Acil servis istasyonu yerleşiminde bazı alanların kapsanmadan bırakılmasını önlemek için yeniden yerleşim kararlarının periyodik olarak alınması gerekmektedir.

Ambulansların yeniden yerleşimi daha zor olmaktadır çünkü bu problemin çok kısa süre içerisinde defalarca çözülmesi gerekmektedir. Gelişen heuristik teknikler ve bilgisayarlar ile gerçek zamanlı ambulans yerleşim problemlerinin çözümü de kolaylaşmış bulunmaktadır.

Gendreau ve arkadaşları tarafından geliştirilen dinamik model standart kapsama modellerine ilave olarak problemin dinamik doğasından kaynaklanan bazı hususları dikkate almıştır (Gendreau ve diğer., 2001, s.1642, Brotcorne ve diğer., 2003, s.458):

- başarılı yerdeğişimi yapan araçlar daima aynını tekrarlamaz
- aynı iki yerleşim alanı arasında tekrarlanan dairesel seyahatlardan kaçınılmalıdır.
- Başlangıç ve son istasyonlar arasında uzun seyahatlerden kaçınılmalıdır.

Gendreau ve arkadaşları tarafından öne sürülen modelde ambulans yerleşimi probleminin dinamik yönünü t süresinde $j \in W$ deki l ambulans ile yer değiştirmenin maliyetini gösteren ve zamana bağımlı M_{jl}^t sabiti sağlamaktadır. Bu durum ambulansın bulunduğu yerden çağrı gelmesi durumunu ($M_{jl}^t = 0$) da kapsamaktadır. M_{jl}^t sabiti l ambulansın geçmişini de kapsamaktadır. Eğer ambulans t zamanından önce sıklıkla hareket etmiş ise M_{jl}^t de daha büyük bir değer almaktadır. Modelde j bölgesine hareket etmekte olan ambulans için 1 değeri alan y_{jl} ikili değişkeni kullanılmaktadır. Model aşağıda gösterilmiştir:

$$(\text{DDSM}^t) \text{ maksimize } \sum_{i \in V} d_i x_i^2 - \sum_{j \in W} \sum_{l=1}^p M_{jl}^t y_{jl}$$

Kısıtlar

$$\sum_{j \in W_i^2} \sum_{l=1}^p y_{jl} \geq 1 \quad (i \in V)$$

$$\sum_{i \in V} d_i x_i^1 \geq \alpha \sum_{i \in V} d_i$$

$$\sum_{j \in W_i^1} \sum_{l=1}^p y_{jl} \geq x_i^1 + x_i^2 \quad (i \in V)$$

$$x_i^2 \leq x_i^1 \quad (i \in V)$$

$$\sum_{j \in W} y_{jl} = 1 \quad (l = 1, \dots, p)$$

$$\sum_{l=1}^p y_{lj} \leq p_j \quad (j \in W)$$

$$x_i^1, x_i^2 \in \{0,1\} \quad (i \in V)$$

$$y_{jl} \in \{0,1\} \quad (j \in W, l = 1, \dots, p)$$

Modelde y_{jl} dışındaki tüm değişkenler, parametreler ve kısıtlar statik problemde olduğu gibi ifade edilebilir. Amaç fonksiyonu r_1 zaman süresinde iki defa kapsanan talepten t zamanında hareket eden araçla ilgili cezaların toplamının çıkarılması ile elde edilmiştir. Araştırmacılar bu problemi tabu araması yöntemi ile çözmüşlerdir. Montreal şehri verileri ile yapılan çalışmada %95 oranında sistemin gerekli yeniden yerleşimi hesaplayabildiği tespit edilmiştir (Gendreau ve diğer., 2001, s;1642, Brotcorne ve diğer., 2003, s;458).

BEŞİNCİ BÖLÜM

ACİL SERVİS YERLEŞİM PROBLEMİ UYGULAMASI

Ambulans, itfaiye ve güvenlik gibi acil hizmet sistemleri insan sağlığı ve hayatı ile doğrudan ilişkili oldukları için üzerinde önemle durulan sistemlerdir. Bu sistemlerin etkinliğini ölçmek, verimliliklerini ve etkinliklerini artırmak, hizmet kalitesini geliştirmek amaçlı literatürde çok sayıda çalışma yapılmıştır. Bu duruma paralel olarak acil sağlık hizmetlerinin dizayn ve uygulamalarına da bilim adamları tarafından verilen önem giderek artmış bulunmaktadır. Acil durumlarda zaman kritik öneme sahip olduğundan yeterli kapsama ve hızlı müdahalenin sağlanması için optimal bir istasyon yerleşim planı yapılması gerekmektedir. Yapılan çok yönlü çalışmalarla bu hizmetlerin daha kolay ulaşılır kılınmış olması, ihtiyaç olduğunda daha hızlı verilebilmesi ve zamanla daha iyi ekipmanlar kullanılır olması sağlanmasına rağmen bütçe sorunları, hazırlık maliyetlerinin azaltılması ve işgücü planlaması gibi problemler konuya olan ilgiyi canlı tutmuş ve daha ileri araştırmaların yapılmasına yol açmıştır.

Acil sağlık hizmeti üzerinde yapılan bazı çalışmalar stratejik planlama üzerine yoğunlaşırken bazıları acil olmayan çağrılarda ambulans hizmeti gibi konuları da kapsayan taktiksel alana yönelmiş; bazıları da ambulans sevki ve yerleşimi gibi operasyonel konulara odaklanmıştır (Beraldi ve Bruni, 2008, s:1).

Çalışma, Trabzon İl Sağlık Müdürlüğü bünyesinde faaliyet göstermekte olan 112 acil yardım acil yardım istasyonlarının şehir merkezindeki yerleşiminin genetik algoritma yöntemi kullanılarak belirlenmesi üzerine yapılmıştır.

112 acil yardım istasyonları acil sağlık hizmeti sunmak ve tıbbi müdahalelerde bulunmak amacıyla kurulan birimlerdir. İstasyonlarda acil sağlık hizmetleri konusunda eğitim görmüş sağlık ekibi ile tıbbi donanımlı ambulanslar görev yapar. İstasyonda ambulans ve ambulanda görev yapan ekibe lojistik destek sağlamak amacıyla, en az üç oda, eğitim salonu, tuvalet, banyo, mutfak, malzeme deposu, ambulans garajı ile telefon, sabit telsiz ve gereken diğer malzeme bulunur. İstasyonların kurulacağı yerin belirlenmesinde aşağıdaki kriterler göz önüne alınmaktadır (RG 24/3/2004- 25412):

- Hizmet sunulması planlanan hedef nüfusun azami elli bin kişi olması
- Ulaşım imkanlarının güçlüğü
- Acil yardım gerektiren olayların sıklığı

- Trafik ve iş kazaları sayısı ve benzeri olayların sıklığı

Sağlık Bakanlığı tarafından yayınlanan Acil Sağlık Hizmetleri Yönetmeliği'ne göre üç tip istasyon bulunmaktadır (RG-15/03/2007-26463) :

A Tipi İstasyon: 24 saat kesintisiz sadece ambulans hizmeti verilen, ihtiyaca göre birden fazla ekip ve ambulans bulundurulan, idari ve özlük hakları bakımından başhekimliğe bağlı ve kadrolu personeli olan istasyonlardır. Bu istasyonlar da kendi içerisinde iki kısma ayrılmaktadır:

1. A1 tipi istasyon: İçerisinde hekim bulunan A tipi istasyon,
2. A2 tipi istasyon: İçerisinde hekim bulunmayan A tipi istasyonlardır.

B Tipi İstasyon: Birinci, ikinci ve üçüncü basamak resmi sağlık kurum ve kuruluşları ile entegre olarak kesintisiz ambulans ve acil servis hizmeti verilen, kadrosu ve özlük hakları bakımından bünyesinde bulunduğu kuruma, ambulans hizmeti bakımından merkeze bağlı olan, ekip içerisinde hekim bulunan istasyonlardır. B tipi istasyonların da kendi içerisinde iki farklı tipi söz konusudur:

1. B1 tipi istasyon: Hastane acil servisi ile entegre olan istasyonlar,
2. B2 tipi istasyon: Birinci basamak sağlık kuruluşları ile entegre olan istasyonlardır.

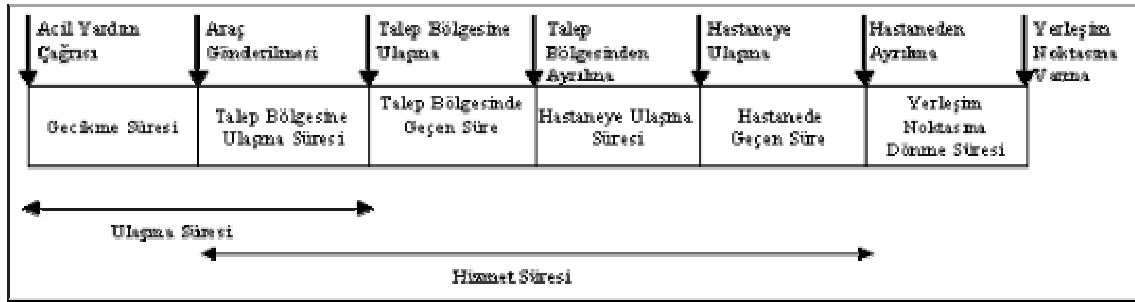
C Tipi İstasyon: İhtiyaca göre günün belirlenen saatlerinde sadece ambulans hizmeti verilen, idari ve özlük hakları bakımından İl Ambulans Servisi Başhekimliğine bağlı acil sağlık istasyonlarıdır.

Bir acil tıbbi hizmet sistemi aşağıdaki alt sistemlerden oluşur (Selim ve Özkarhan, 2003, s;19)

- Haberleşme
- Ulaştırma
- Tıbbi müdahale

Haberleşme ve taşıtırma alt sistemleri birlikte ambulans sistemi olarak ele alınır. Bir ambulans sistemi, acil yardım isteğinin bildirildiğı çağrı ile başlar ve hastanın en uygun tıbbi hizmet birimine taşıtırılmasıyla sona eren bir dizi olaydan oluşmaktadır. Bu olaylar dizisi Şekil 5.1'de gösterilmiştir.

Bir ambulans sistemindeki olaylar dizisi acil sađlık hizmetinin verildiđi bölgedeki herhangi bir noktada meydana gelen ve insan sađlıđının tehlikeye düřtüđü olaylar veya sađlık problemlerinin tespiti ile bařlar. Tespit edilen durum 112 Acil Sađlık Komuta Kontrol Merkezi aranarak bildirilir. Acil çağrı geldiđi anda kontrol merkezindeki acil servis çalışanları için önemli olan olayın řiddetini ve aciliyet derecesini belirlemek ve gönderilecek ambulans tipi, sayısı ve çıkış noktasına karar vermektir. Görüşmede alınan bilgiler ışığında Merkez tarafından gerekli yönlendirme yapılır. İstasyon yakınlıđı, istasyondaki ambulansın vakanın durumuna uygun ekipman ve personeli olması ve herhangi bařka bir vakaya yönlendirilmemiř olması bu aşamada dikkat edilen hususları oluřturmaktadır. Ambulans olay yerine ulařtıđında ilk müdahale yapılır. Vakanın ilk müdahale dıřında hastanede birtakım müdahaleler gerektirmesi durumunda hasta en yakın hastaneye ulařtırılır. Hastayı ilgili acil servise ulařtıran ambulans bađlı olduđu merkeze geri döner.



Şekil 5.1: Bir ambulans sistemindeki olaylar dizisi

Standartlara göre řehir merkezlerinden gelen çağrıların %95'ine 10 dakikada, kırsaldan gelen çağrıları ise 30 dakikada müdahalede bulunulması gerekmektedir (Brotcorne ve diđer., 2003, s;452)

5.1. Genetik Algoritma Metodu Kullanılarak Trabzon İl Merkezi Ambulans Yerleřim Planı Çıkarılması

Çalışmada Trabzon il merkezindeki acil yardım istasyonları yerleřim planı irdelenmiřtir. Trabzon'un il merkez nüfusu 2007 yılı adrese dayalı nüfus sayımına göre 220.000'dir. İl merkezinde 1993 yılından bu yana acil yardım hizmeti verilmektedir. Ülke genelinde acil yardım hizmetlerinin yaygınlařması ile çağrı merkezine gelen

aramaların sayısı da yıllar itibarı ile sürekli artış göstermiştir. İl Merkezinde 6 adet hastane faaliyet göstermektedir: Numune Araştırma Hastanesi, Fatih Devlet Hastanesi, Farabi Hastanesi, Yavuz Selim Kemik Hastalıkları Hastanesi, Göğüs Hastalıkları Hastanesi ve Doğum Hastanesi.

Mevcut durumda acil çağrılarının tamamına yakını üç hastanedeki acil servislere yönlendirilmektedir:

- Numune Araştırma Hastanesi Acil Servisi,
- Fatih Devlet Hastanesi Acil Servisi,
- Farabi Hastanesi Acil Servisi.

Trabzon 112 Acil Komuta ve Kontrol Merkezine gün boyunca çok sayıda çağrı gelmektedir. Çağrılarının önemli bir kısmı muhatabı olmadan sonlandırılmaktadır. Diğer bir kısım çağrıda ise konuşmalı görüşme yapılmaktadır. Bu görüşmelerin sonucunda ise günde ortalama 50 vakaya hizmet verilmesi söz konusudur. 2006 yılında toplam 15.866 vaka söz konusu iken bu rakam 2007 yılında 19.024 olmuştur. İl merkezindeki vaka sayıları ise sırasıyla 9355 ve 9537'dir.

112 acil Komuta Merkezi yetkilileri il genelindeki vakalarde şehir merkezleri için ilk 10 dakikada vakaya müdahale etme oranlarının %87 olduğunu ifade etmişlerdir. Standartlara göre bu oranın %90'ın üzerinde olması gerekmektedir.

Çağrılara daha iyi hizmet verebilmek için bağımsız bir komuta kontrol merkezi oluşturulmuş, çağrı karşılama ve ambulans yönlendirme işlemlerinde otomasyona geçilmiş, çağrı karşılama yazılımı, ses kayıt sistemi ve araç takip ve yönlendirme sistemi kullanılmaya başlanmıştır.

İl merkezinde üç farklı noktada acil yardım istasyonları kurulu bulunmaktadır. İlgili yönetmelikte de belirtildiği üzere A tipi istasyonlarda kadrolu personeller çalışırken B tipi istasyonlarda ise istasyonun bünyesinde kurulu bulunduğu hastane personeli çalıştırılmaktadır. Maliyet açısından bakıldığında B tipi istasyonların operasyon maliyeti A tipine göre oldukça düşüktür. Trabzon'da da bu özellikler dikkate alınarak istasyonlar daha çok B tipinde kurulmuştur. Şehir merkezinde kurulu bulunan istasyonlar aşağıda Çizelge 5.1'de gösterilmektedir.

Çalışmada Coğrafi Bilgi Sistemleri uygulamalarından faydalanılmıştır. Şehir merkezinde potansiyel yerleşim yerlerinin belirlenmesi için mahalle idari bölümlendirmesi kullanılmıştır. Şehir merkezinde 32 mahalle bulunmaktadır. Şehirle

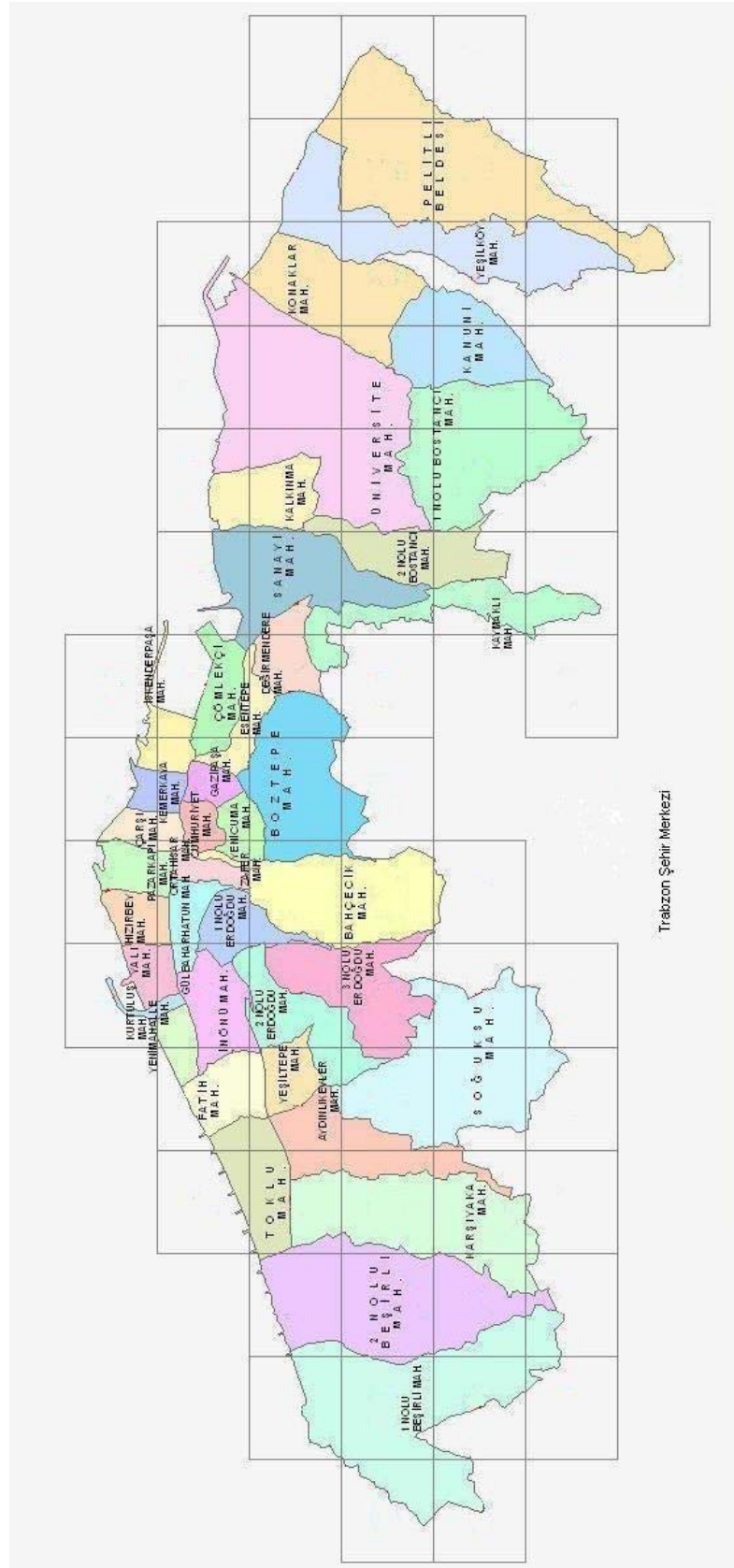
her bakımdan bütünleşik olan iki belde de bu mahalleler ile birlikte düşünülmüş ve toplamda 34 yer tespit edilmiştir. Şehir merkezindeki mahalle yerleşimleri Şekil 5.2' de gösterilmiştir.

İstasyonun Adı	Tipi	Konuşlandığı Yer
Merkez 1 Nolu Acil Yardım İstasyonu	A	Numune Araştırma Hastanesi Kampüsü
Merkez 1A Nolu Acil Yardım İstasyonu	C	Numune Araştırma Hastanesi Kampüsü
Merkez 2 Nolu Acil Yardım İstasyonu	B	Fatih Devlet Hastanesi Kampüsü
Merkez 3 Nolu Acil Yardım İstasyonu	B	Göğüs Hastalıkları Hastanesi Kampüsü
Merkez 4 Nolu Acil Yardım İstasyonu	B	Yavuz Selim Kemik Hastalıkları Hastanesi Kampüsü

Çizelge 5.1. Trabzon İl Merkezi Acil Servis İstasyonlarının durumu

Potansiyel yerleşim yerlerinin tespiti için ilgili yönetmelikte belirtilen kısıtlar, ulaşım durumu, mahalle içerisindeki konumu, önemli noktalara uzaklığı, istasyon için yer bulma durumu gibi faktörler dikkate alınarak her bir mahalle için birer nokta potansiyel istasyon merkezi olarak belirlenmiştir. Mevcuttaki istasyonların özellikle kampüsü içerisinde kurulu bulundukları hastaneler de dikkate alınarak yerleri değiştirilmemiş ve içinde bulundukları mahalle için merkez kabul edilmişlerdir.

iGO 8 yazılımı kullanılarak daha önce belirlenen merkezler arasındaki uzaklıklar hesaplanmıştır. Mesafe hesaplamalarında gerçekte kullanılan yollar ve gidiş yönleri dikkate alınarak merkezler arasındaki en kısa karayolu uzaklıkları hesaplanmıştır. Potansiyel istasyon merkezleri arasındaki mesafeler Ek1'de gösterilmiştir.



Şekil 5.2. Trabzon Şehir Merkezi Haritası

5.2. Varsayımlar ve Oluşturulan Model

Çalışmanın sonuçlandırılabilmesi için gerek duyulan varsayımlar aşağıda sıralanmaktadır:

- İstasyon yerleşimi için her bir aday bölge aynı zamanda ambulans hizmeti alacak bölgedir.
- Her ambulans yaklaşık 50000 kişiye hizmet verebilmelidir.
- Günlük vaka sayısı toplam vaka sayısının gün sayısına bölünerek hesaplanmış ve bölgeler arasında uniform dağıldığı varsayılmıştır.
- Tüm alan acil sağlık hizmeti tarafından kapsanacaktır.

Bu varsayımlar ışığında minimum mesafeyi bulmaya yönelik olarak oluşturulan model aşağıda gösterilmiştir:

$$\min Z = \sum_{i=1}^m \sum_{j=1}^n d_{ij} (1 - q_i) q_i^{k-1} x_{ij}$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, m$$

$$x_{ij} \leq x_{jj} \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n$$

$$\sum_{j=1}^n x_{jj} = p$$

$$x_{ij} \in \{0, 1\} \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n$$

Notasyon;

i: talep noktası indeksi

j: potansiyel istasyon yerleşim noktaları

m: toplam talep noktaları sayısı

n: toplam potansiyel istasyon yerleşim yeri sayısı

k: yerleşimi düşünülen istasyon sayısı

d_{ij} : i talep alanı ve j istasyonu arasındaki uzaklık

x_{ij} : i talep noktası j istasyonundaki araç tarafından kapsanıyorsa 1, değilse 0.

Herhangi bir j alanındaki ambulansın çağrı geldiğinde meşgul olup olmama olasılığı q olarak alınmıştır. bu açıdan model Daskin (1983) tarafından geliştirilen MEXCLP modeli temel alınarak oluşturulmuştur. Modelde Bernoulli olasılık dağılımı esas alınmış ve temel alınan modelden farklı olarak her alanın farklı q değerine sahip olduğundan hareket edilmiştir. k sayıda aracın hizmet verdiği bölge için meşgul olma olasılığı da q^k olmaktadır. Bu durumda en az bir aracın meşgul olmama olasılığı da $1 - q^k$ olarak hesaplanmaktadır. Verilen araçlardan en az birinin meşgul olmaması durumunda artırımsal olasılık ise

$$(1 - q^k) - (1 - q^{k-1}) = q^{k-1}(1 - q)$$

formülü ile hesaplanmıştır.

Genetik algoritma ile çözümde ilk olarak başlangıç popülasyonu oluşturulmuştur. Başlangıç popülasyonu için öncelikle her bir potansiyel yer için uniform olasılıkla rassal sayı üretilmiş ve matris içerisinde her bir nokta potansiyel yeri temsil edecek şekilde konumlandırılmıştır.

Oluşturulan genetik algoritma doğrusal modeldeki kısıtları kullanmıştır. Çözüm alanının daha iyi taranabilmesi ve erken çözüm elde edilebilmesi için literatürde belirtilen değerler doğrultusunda başlangıç popülasyon büyüklüğü belirlenmiştir.

Şehir nüfusuna göre standartlardaki değerleri kapsayacak biçimde ilk olarak dört adet istasyon kurulumunu öngörerek oluşturulan başlangıç popülasyonunun ilk kısmı Çizelge 5.2’de gösterilmektedir.

Başlangıç popülasyonu için 16 kromozom seçilmiştir. Oluşturulan matrisin boyutu 16×136 olmuştur. Hesaplama istasyon tipleri arasında herhangi bir ayırım gözetilmemiştir. Bununla birlikte potansiyel merkezlerin seçiminde mevcutta bulunan hastaneler merkez olarak tanımlanmış olduğundan dolayı genetik algoritma tarafından bu merkezlere istasyon atanması durumunda istasyon tipinin doğal olarak B tipi olacağı düşünülmüştür.

Kromozomların istenilen sonuca ulaşip ulaşmadığını denetleyen uygunluk fonksiyonu olarak ise kat edilen mesafe fonksiyonu kullanılmıştır. Vakaya müdahale süresinin de kat edilen mesafe ile doğru orantılı olduğu varsayımından hareket edilmiştir. Bu açıdan mesafenin minimize edilmesi vakaya en erken sürede müdahale edilmesi anlamını taşımaktadır. Değerlendirmede daha kısa mesafeyi veren kromozom tercih edilmiştir.

1	0	1	0	0	1	0	1	0	0	1	0	1	1	1	1	1	0	0	1	0	1	0	0	1	0	1	0	1	1	0	1	0	1	
1	1	0	1	1	0	1	1	1	1	0	1	1	0	0	0	1	1	1	0	0	1	1	0	1	1	1	0	1	1	1	0	1	1	
0	1	0	0	1	0	0	1	1	0	1	0	0	0	1	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	0	1	
1	0	1	1	0	1	1	0	0	0	1	1	0	1	1	0	1	0	0	0	0	1	0	0	1	1	1	0	1	1	0	1	1	0	
0	1	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	1	1	0	1	1	1	1	0	0	0	0	1	0	0	0
0	1	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	0	0	0	0	1	0	1	0	0	1	0	0	0	0	1	0	0	0
1	0	0	1	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	0	0	1	1	
1	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	
0	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	1	0
1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	1	1	1	1	0	0	1	0	1	0	0	1	0	0	1	1	1	1	1
0	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	1	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0
0	0	1	1	0	1	1	0	0	0	0	1	0	1	1	1	1	1	0	1	0	0	1	0	1	0	1	0	0	0	0	1	1	0	
1	0	1	0	0	1	0	1	0	1	0	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0	1	1	0	1	0	1	0	1	
1	0	1	1	0	1	1	0	1	1	0	0	1	1	0	0	0	1	1	0	1	0	1	0	0	0	1	1	1	1	0	1	1	0	
0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	1	0	1	1	0	1	1	1	1	0	0	0	0	0	
1	1	1	0	1	1	0	1	1	1	1	0	0	0	1	1	0	1	0	0	1	0	0	1	0	0	0	1	0	1	1	1	1	0	1

Çizelge 5.2. İlk popülasyon matrisi

Trabzon'a özgü coğrafi koşullar nedeni ile potansiyel yerleşim yerleri arasındaki ulaşım imkanları literatürde öngörülenden farklılıklar arz etmektedir. Gerek ulaşım altyapısının iyi planlanmamış olması ve gerekse yolların vakaların ağırlık taşıdığı şehir merkezinde dar olması ambulansların ortalama hızlarını oldukça düşürmektedir. Çalışmada eldeki verilere dayalı olarak ortalama ambulans hızı 30 km/saat olarak alınmıştır.

Çaprazlama için kromozom seçiminde rulet tekerleri yöntemi kullanılmıştır. Bu yöntemler popülasyondaki her bir kromozomun hesaplanan uygunluk değerleri sıraya dizilmiş ve ağırlıklandırılmıştır. Belirlenen bu değerler ile ilk %50'lik dilim seçilmiştir. Bu şekilde daha iyi kromozomların çaprazlama ve mutasyona uğrama olasılığı artırılmış ve daha da iyi bireylerin oluşturulması amaçlanmıştır.

Çalışmada çaprazlama oranı 0.25 olarak alınmıştır. Bu büyüklükteki popülasyonlar için literatürde ileri sürülen uygun oranlar ile uyumlu bir oran seçilmiştir.

Çaprazlama metodu olarak ise iki noktalı çaprazlama tercih edilmiştir. Tek noktalı çaprazlama uygulamada oldukça kolay olmasına rağmen pratikte sonuca daha hızlı ulaşmayı garanti etmemektedir. Literatürde yapılan çalışmalar incelendiğinde iki noktalı çaprazlamanın daha iyi sonuç verdiği tespit edilmiştir.

Çaprazlamadan sonra elde edilen kromozomlar mutasyona tabi tutulmuştur. Çalışmada 0,01'lik bir mutasyon oranı tercih edilmiştir.

Çalışmada sonlandırma kriteri olarak ise iterasyon sayısı alınmıştır. Yapılan çok sayıdaki denemeden sonra 700 rakamının gerekli yakınsamayı sağladığı tespit edilmiştir.

BULGULAR ve ÖNERİLER

Türkiye İstatistik Kurumu Trabzon bölge müdürlüğü'nden Trabzon merkez ve mahalle bazında nüfus bilgileri alınarak dijital şehir haritasında bu veriler yerine konulmuştur. Mahallelerin konumu ve diğer veriler ışığında bazı mahalleler birlikte ele alınarak şehir merkezi 34 alana bölünmüştür.

Genetik algoritma oluşturulan statik probabilistik model varsayımları ile hareket ettirilmiştir. Mevcut durumda 6,11,12, 21 nolu bölgelerde istasyon kurulu bulunmaktadır.

Yapılan denemelerde önerilen model için:

- Rulet tekerleği seçimi,
- iki noktalı çaprazlama,
- 0,25 çaprazlama oranı ve
- 0,01 mutasyon oranının daha iyi sonuçlar verdiği tespit edilmiştir.

700 iterasyon sonucunda elde edilen minimum taşıma mesafesini içeren yerleşim modeli aşağıda Çizelge 5.3'de gösterilmektedir.

İstasyonun Adı	Tipi	Konuşlandığı Bölge
Merkez 1 Nolu Acil Yardım İstasyonu	B	Fatih Devlet Hastanesi Kampüsü
Merkez 2 Nolu Acil Yardım İstasyonu	A	14
Merkez 3 Nolu Acil Yardım İstasyonu	A	18
Merkez 4 Nolu Acil Yardım İstasyonu	B	Yavuz Selim Kemik Hastalıkları Hastanesi Kampüsü

Çizelge 5.3. Trabzon İl Merkezi Acil Servis İstasyonlarının yerleşimi

Elde edilen sonuca göre minimum mesafe 11,14,18, 21 nolu bölgelere istasyon kurulması ile elde edilebilmektedir.

Mevcut yerleşim ile yapılan hesaplamalara göre toplam taşıma mesafesinde önerilen yerleşimde %19,44'lük bir iyileşme sağlanmıştır (131284,105759).

Elde edilen sonuçların geçerliliğini sınamak için yüzey arama (space search) tekniği kullanılmıştır. Bu teknik temel alınarak oluşturulan yazılım ile model tekrar çalıştırılmış ve tüm yüzeyin taranması sağlanmıştır. Bu işlem için geliştirilen yazılım kodları ekte sunulmuştur.

Yüzey arama tekniği ile yapılan uygulama sonuçları da genetik algoritma ile 700 iterasyon sonucunda elde edilen en iyi çözümü doğrulamış olup, en kısa mesafenin 11,14,18, 21 nolu bölgelere istasyon yerleşimi ile elde edilebileceği sonucunu vermiştir. Genetik algoritma tekniği kullanımında kısa sürede sonuç elde edilebilmiş iken yüzey arama tekniği çözümünün elde edilmesi yaklaşık altı saat sürmüştür.

Tesis yerleşim problemleri üzerinde uzunca bir süredir çalışılan konulardan biridir. Farklı koşullar ve sistemler üzerinde daha iyi sonuçlar veren modeller üzerindeki çalışmalar halen sürdürülmektedir. Problemin çözümüne yönelik olarak çok sayıda deterministik, dinamik ve stokastik modeller ortaya konulmuştur. Koşullardaki belirsizliklerin artması sezgisel ve metasezgisel metotların kullanımını yaygınlaştırmıştır. Uygulamada statik probabilistik bir model kullanılmıştır. Elde edilen sonuçlar ve tecrübeye göre ileride aşağıdaki çalışmaların yapılması mümkün görülmektedir:

- Kullanılan probabilistik model ambulansların meşgul olma durumlarını dikkate almakla birlikte gün içinde ambulansların hizmet sürelerinin de değişebildiği dikkate alınarak yeni bir model geliştirilebilir.
- Bir simülasyon modeli ile dinamik ve belirsiz koşullarda sistemin işleyişi irdelenebilir.
- Modele maliyet bileşeni ilave edilebilir. Bu durumda A tipi istasyon kurulumunun maliyetinin yüksekliği elde edilen sonucu etkileyecektir.

KAYNAKLAR

Kitap ve Yayınlar

- Antoniou, A., Lu, Wu-Sheng, Practical Optimization: Algorithms and Engineering Applications, Springer Science+Business Media, New York, 2007
- Arora, J.S.(Eds), Optimization of Structural and Mechanical Systems, World Scientific Publishing, Singapore, 2007
- Ashlock, Daniel, Evolutionary Computation For Modeling and Optimization, Springer Science – Business Media Inc., 2006
- Bäck, Thomas, Evolutionary Algorithms in Theory and Practice, Oxford University Press, New York, First Edition, 1996
- Bäck, Thomas., Fogel, David B., Michalewicz, Zbigniew. 2000a, Evolutionary Computation I: Basic Algorithms and Operators, Institute of Physics Publishing,
- Bäck, Thomas., Fogel, David B., Michalewicz, Zbigniew. 2000b, Evolutionary Computation II: Advanced Algorithms and Operators, Institute of Physics Publishing, Bristol and Philadelphia
- Banzhaf, W., Nordin, P., Keller, R.E., Francone, F.D., Genetic Programming: An Introduction, Morgan Kaufman Publishers, Inc, San Francisco, California, 1998
- Barnhart, C., Laporte, G., Handbooks in Operations Research & Management Science: Transportation, North-Holland, First Edition, Amsterdam 2007
- Bertsekas, Dimitri P., Network Optimization: Continuous and discrete Models, Athena Scientific, Massachusetts, 1998
- Bramel, Julien, Simchi-Levi, David, The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management, springer-Verlag, New York, 1997
- Brandeau, M.L., Sainfort, F., Pierskalla, W.P. (Eds), Operations Research in Health Care, Kluwer Academic Publisher, Boston, 2004
- Branke, J., Deb, K., Miettinen, K., Slowinsky, R., (Eds.), Multiobjective Optimization, Springer-Verlag, Berlin, 2008
- Chambers, Lance, The Practical Handbook of Genetic Algorithms, Vol.1 Applications, Chapman&Hall/CRC, Second Edition 2001
- Chambers, Lance, The Practical Handbook of Genetic Algorithms, Vol.2 New Frontiers, Chapman&Hall/CRC, 1995

- Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A., *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, Second Edition, 2007
- Coley, David A., *An Introduction to Genetic Algorithms for Scientists and Engineers*, World Scientific Publishing Co., 1999
- Dantzig, George B., Thapa, Mukund N., *Linear Programming 1: Introduction*, Springer, 1st edition, New York, 1997
- Engelbrecht, Andries P., *Computational Intelligence*, John Wiley & Sons Inc., Second Edition, 2007
- Fletcher, R., *Practical Methods of Optimization*, John Wiley & Sons, Second Edition, 1987
- Floudas, C.A., Pardalos, P.M., *Encyclopedia of Optimization*, Springer Science+Business Media, Second Edition, New York, 2009
- Gao, D.Y., Sherali, H.D., (Eds), *Advances in Applied, Mathematics, and Global Optimization*, Springer Science+Business Media, LLC, 2009
- Gen, M., Cheng, R., *Genetic Algorithms & Engineering Optimization*, John Wiley & Sons, Inc., 2000
- Haupt, Randy L., Haupt, Sue Allen. *Practical Genetic Algorithms*, John Wiley & Sons Inc., Second Edition, 2004
- Jain, Lakhmi C., Martin, N.M. *Fusion on Neural Networks, Fuzzy Systems and Genetic Algorithms: Industrial Applications*, CRC Press LLC, 1998
- Karr, Charles., Freeman, L. Michael., *Industrial Applications of Genetic Algorithms*, CRC Press LLC, 1999
- Kennedy, James, Eberhart, Russel C., Shi, Yuhui, *Swarm Intelligence*, Morgan Kaufman Publishers, San Francisco, CA, 2001
- Kinnear, Kenneth E., *Advances in Genetic Programming, Volume 1*, MIT Press, 1994
- Michalewicz, Zbigniew, *Genetic Algorithm + Data Structure = Evolution Programs*, Springer-Verlag, 3rd Edition, New York, 1996
- Mitchell, Melanie, *An Introduction to Genetic Algorithms*, MIT Press, Fifth Printing, 1999
- Nabiyev, Vasif, *Yapay Zeka, Seçkin Yayıncılık, 1. Baskı.*, 2003

- Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. (Eds.), *Handbooks in Operations Research and Management Science: Optimization*, Elsevier Science B.V., Third Impression, Amsterdam, 1998
- Nocedal, J., Wright, S.J., *Numerical Optimization*, Springer Science+Business Media, LLC, Second Edition, 2006
- Olariu, Stephan. Zomaya, Albert Y., *Handbook of Bioinspired Algorithms and Applications*, Chapman & Hall/CRC, 2006
- Pelikan, M., Sastry, K., Cantu-Paz, E., *Scalable Optimization via Probabilistic Modeling*, Springer-Verlag, Berlin, 2006
- Rao, Singiresu S., *Engineering Optimization: Theory and Practice*, John Wiley&Sons Inc., Third Edition, New York, 1996
- Reeves, C.R., Rowe, J.E., *Genetic Algorithms: Principles and Perspectives*, Kluwer Academic Publishers, Dordrecht, 2003
- Rothlauf, Franz. *Representation for Genetic and Evolutionary Algorithms*, Springer-Heidelberg, 2006
- Rutkowski, Leszek, *Computational Intelligence: Methods and Techniques*, Springer-Verlag, Heidelberg, 2005
- Sakawa, Masatoshi, *Genetic Algorithms and Fuzzy Multiobjective Optimization*, Kluwer Academic Publisher, Massachusetts, 2002
- Salvendy, Gavriel, *Handbook of Industrial Engineering: Technology and Operations Management*, John Wiley & Sons, Third Edition, 2001
- Sarker, R., Mohammadian, M., Yao, X., (Eds), *Evolutionary Optimization*, Kluwer Academic Publishers, Dordrecht, 2003
- Sivanandam, S.N., Deepa, S.N., *Introduction to Genetic Algorithms*, Springer, 1st Edition, 2008
- Spall, James C., *Introduction to Stochastic Search and Optimization*, John Wiley & Sons Inc., New Jersey, 2003
- Sule, Dileep R., *Logistics of Facility Location and Allocation*, Dekker Inc., First Edition, New York, 2001
- Sumathi, S., Hamsapriya, T., Surekha, P., *Evolutionary Intelligence*, Springer-Verlag, Coimbatore, India, 2008

- Şen, Zekai, Genetik Algoritmalar ve En İyileme Yöntemleri, Su Vakfı Yayınları, İstanbul 2004
- Törn, A., Zilinkas, A., Global Optimization, Springer-Verlag, Heidelberg, 1989
- Weise, T., Global Optimization Algorithms: Theory and Application, www.it-weise.de, Second Edition, 2008
- Zhang, W., State Space Search: Algorithms, Complexity, and Applications, Springer-Verlag, New York, 1999
- Zhigljavsky, A., Zilinskas, A., Stochastic Global Optimization, Springer Science+Business Media, New York, 2008

Makaleler

- Bäck, Thomas. 1994, “Selective pressure in evolutionary algorithms: A characterization of selection mechanisms”, Proceedings of the first IEEE Conference on Evolutionary Computation, IEEE Press, Orlando, pp.57-62
- Ball, M.O., Lin, L.F., 1993, “A reliability model applied to emergency service location”, Operations Research, 41, ss.18-36
- Beraldi, P., Bruni, M.E., 2009, “A probabilistic model applied to emergency service vehicle location”, European Journal of Operational Research, 196, ss.323-331
- Brotcorne, L., Laporte, G., Semet, F., 2003, “ambulance location and relocation models”, European Journal of Operational Research, 147, ss.451-463
- Daskin, M.S., Hogan, K., ReVelle, C., 1988, “Integration of multiple, excess, backup, and expected covering models”, Environment and Planning, 15, ss.15-35
- Daskin, Mark S., 1983, “A maximum expected covering location model: Formulation, properties, and heuristic solution”, Transportation Science, Vol.17, No.1, February 1983, ss.48-70
- Dengiz, Berna, Altıparmak, Fulya, 1998, “Genetik Algoritmalar Genel Bir Giriş”, Endüstri mühendisliği Dergisi, Cilt:9, Sayı:3, pp.3-14)
- Emel, Gül Gökay., Taşkın, Çağatay., 2002, “Genetik Algoritmalar ve Uygulama Alanları”, Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi, Cilt.XXI, Sayı:1, s;129-152

- Fujiwara, O., Makjamroen, T., Gupta, K.K., 1987, "Ambulance deployment analysis: A case study of Bangkok", *European Journal of Operational Research*, 31, ss.9-18.
- Gendreau, M., Laporte, G., Semet, F., 2001, "A dynamic model and paralel tabu search heuristic for real-time ambulance relocation", *Parallel Computing*, 27, s.1641-1653
- Gendreau, M., Laporte, G., Semet, F., 1997, "Solving an ambulance location model by tabu search", *Location Science*, Vol.5, No.2, ss.75-88
- Goldberg, J., Dietrich, R., Chem, J.M., Mitwasi, M.G., 1990, "Validating and applying a model for locating emergency medical services in Tucson, AZ", *European Journal of Operational Research*, 49, ss.308-324
- Güden, H., Vakvak, B., Özkan, B.E., Altıparmak, F., Dengiz, B., 2005, "*Genel Amaçlı Arama Algoritmaları ile Benzetim Eniyilemesi: En iyi kanban sayısının bulunması*", *Endüstri mühendisliği Dergisi*, Cilt:16, Sayı:1, pp.2-15
- İşçi, Öznur, Korukoğlu, Serdar, 2003, "Genetik algoritma Yaklaşımı ve Yöneylem Araştırmasında bir Uygulama", *Yönetim ve Ekonomi*, Cilt:10, Sayı:2, s;191-208
- Kurt, Mustafa.-Cumali, Sametay. "Genetik Algoritma ve Uygulama Alanları", *Mühendis ve Makina*, Sayı:501, Ekim 2001
- Lai, K.K. ve Chan, J.W.M., 1997. *Developing A Simulated Annealing Algorithm for The Cutting Stock Problem*. *Computers and Industrial Engineering*, Vol. 32, pp. 115-127
- Mandell, M.B., 1998, "Covering models for two-tiered emergency medical services systems", *Locations Science*, 6, ss.355-368
- Owen, S.H., Daskin, M.S., 1998, "Strategic facility location: A review", *European Journal of Operational Research*, 111, ss.423-447
- Repede, J.F., Bernardo, J.J., 1994, "Developing and validating a decision support system for locating emergency medical vehicles in Luisville, Kentucky", *European Journal of Operational Research*, 75, ss.567-581
- ReVelle, C., Hogan, K., 1989, "The maximum availability location problem", *Transportation Science*, Vol.23, No.3, August 1989, ss.192-200
- Selim; H., Özkarahan, İ., 2003, "Acil Servis Araçları Yerleşiminin Belirlenmesinde Yeni bir Model", *Endüstri Mühendisliği*, Sayı 14, No:1, 18-27

EKLER

EK-1 Mesafe Matrisi

[illegible]

EK-2**Genetik Algoritma MATLAB Programı**

```

clear;
gridsize=34;
yerlesimsayisi=4;
N=gridsize*yerlesimsayisi; %Bir kromozomdaki bit sayısı
M=16; %Kromozom sayısı
last=16; %Jenerasyon Sayısı
sel=0.5; %Seçme Oranı
M2=2*ceil(sel*M/2);
mutrate=0.01; %Mutasyon oranı
nmuts=mutrate*N*(M-1); %Mutasyon sayısı
pop=[];
for jj=1:M
    test=0;
    while test==0
        [poptekli1 poptekli2 poptekli3 poptekli4]=popteklifunc6(gridsize);
        poptek=[poptekli1 poptekli2 poptekli3 poptekli4];
        poptek1=sum(poptekli1);
        poptek2=sum(poptekli2);
        poptek3=sum(poptekli3);
        poptek4=sum(poptekli4);
        if (poptek1~=poptek2) && (poptek2~=poptek3) &&...
            (poptek3~=poptek4) && (poptek1~=poptek3) &&...
            (poptek1~=poptek4) && (poptek2~=poptek4)
            test=1;
        end
    end
    pop=[pop ; poptek];
end

for ii=1:M
    if isempty(find(pop(ii,:)==1))
        pop(ii,round(rand*N))=1;
    end
end
%Her bölgenin meşguliyet oranlarını yukler
load('Qular.txt');
Qu=Qular;
%Mesafe Matrisini yukler
load('MesafeMatrisi.txt');
Mesafe=MesafeMatrisi;
for ib=1:last
    cost=costfunctioncok1MEX(pop,Qu,Mesafe,gridsize);
    [cost,ind]=sort(cost);
    pop=pop(ind(1:M2),:);
end

```

```

cross=ceil((N-1)*rand(M2,1));
popkopya=pop;
caprazlamatesti=0;
kromozomtest=zeros(1,M2);
while caprazlamatesti==0
    for ic=1:2:M2
        popkopya(ceil(M2*rand),1:cross)=popkopya(ic,1:cross);
        popkopya(ceil(M2*rand),cross+1:N)=popkopya(ic+1,cross+1:N);
        popkopya(ceil(M2*rand),1:cross)=popkopya(ic+1,1:cross);
        popkopya(ceil(M2*rand),cross+1:N)=popkopya(ic,cross+1:N);
    end
    for kk=1:M2
        popkopya1=sum(popkopya(kk,1:gridsize));
        popkopya2=sum(popkopya(kk,gridsize+1:2*gridsize));
        popkopya3=sum(popkopya(kk,2*gridsize+1:3*gridsize));
        popkopya4=sum(popkopya(kk,3*gridsize+1:4*gridsize));
        if (popkopya1~=popkopya2) && (popkopya2~=popkopya3) &&...
            (popkopya3~=popkopya4) && (popkopya1~=popkopya3) &&...
            (popkopya1~=popkopya4) && (popkopya2~=popkopya4)
            kromozomtest(1,kk)=1;
        end
    end
    if all(kromozomtest)==1
        caprazlamatesti=1;
    end
end
pop=[pop ; popkopya];
mutasyontesti=0;
while mutasyontesti==0
    for ic=1:nmut
        ix=ceil(M*rand);
        iy=ceil(N*rand);
        pop(ix,iy)=1-pop(ix,iy);
    end
    pop1=sum(pop(1,1:gridsize));
    pop2=sum(pop(1,gridsize+1:2*gridsize));
    pop3=sum(pop(1,2*gridsize+1:3*gridsize));
    pop4=sum(pop(1,3*gridsize+1:4*gridsize));
    if (pop1~=pop2) && (pop2~=pop3) &&...
        (pop3~=pop4) && (pop1~=pop3) &&...
        (pop1~=pop4) && (pop2~=pop4)
        mutasyontesti=1;
    end
end
end
sonuclar=[];
for ii=1:M

```

```

sonuclar=[sonuclar ;...
    ii cost(ii) sum(pop(ii,1:gridsize))...
    sum(pop(ii,gridsize+1:2*gridsize))...
    sum(pop(ii,2*gridsize+1:3*gridsize))...
    sum(pop(ii,3*gridsize+1:4*gridsize)) ];
end
sonuclar

```

```

function sonuc=costfunctioncoklMEX(pop,Qu,Mesafe,gridsize)
yuzeymatrisi=[];
k=4;
for jj=1:size(pop,1)
    yuzey1=0;
    yuzey2=0;
    yuzey3=0;
    yuzey4=0;
    for ii=1:34
        yuzey1=yuzey1+Mesafe(ii,sum(pop(jj,1:gridsize)))*((1-Qu(ii))*Qu(ii)^(k-1));
        yuzey2=yuzey2+Mesafe(ii,sum(pop(jj,gridsize+1:2*gridsize)))*((1-Qu(ii))*Qu(ii)^(k-1));
        yuzey3=yuzey3+Mesafe(ii,sum(pop(jj,2*gridsize+1:3*gridsize)))*((1-Qu(ii))*Qu(ii)^(k-1));
        yuzey4=yuzey4+Mesafe(ii,sum(pop(jj,3*gridsize+1:4*gridsize)))*((1-Qu(ii))*Qu(ii)^(k-1));
    end
    yuzey=yuzey1+yuzey2+yuzey3+yuzey4;
    yuzeymatrisi=[yuzeymatrisi ; yuzey];
end
sonuc=yuzeymatrisi;

```

```

function [final1 final2 final3 final4]=popteklifunc6(gridsize)
sonuc=zeros(1,4);
secim=1:gridsize;
kk=1;
while kk<5
    rands=ceil(gridsize*rand);
    if (secim(rands)~=0) && (rands~=0)
        sonuc(1,kk)=secim(rands);
        secim(rands)=0;
        kk=kk+1;
    end
end
kk=1;
final1=zeros(1,gridsize);
while kk<sonuc(1,1)+1
    rands=ceil(gridsize*rand);
    if (final1(rands)~=1) && (rands~=0)

```

```

        final1(1,rands)=1;
        kk=kk+1;
    end
end
kk=1;
final2=zeros(1,gridsize);
while kk<sonuc(1,2)+1
    rands=ceil(gridsize*rand);
    if (final2(rands)~=1) && (rands~=0)
        final2(1,rands)=1;
        kk=kk+1;
    end
end
kk=1;
final3=zeros(1,gridsize);
while kk<sonuc(1,3)+1
    rands=ceil(gridsize*rand);
    if (final3(rands)~=1) && (rands~=0)
        final3(1,rands)=1;
        kk=kk+1;
    end
end
kk=1;
final4=zeros(1,gridsize);
while kk<sonuc(1,4)+1
    rands=ceil(gridsize*rand);
    if (final4(rands)~=1) && (rands~=0)
        final4(1,rands)=1;
        kk=kk+1;
    end
end
end

```

ÖZGEÇMİŞ

1974 yılında Trabzon'un Çaykara ilçesinde doğdu. İlk ve orta öğrenimini Çaykara'da, liseyi ise Trabzon'da Trabzon Lisesi'nde tamamladı. 1998 yılında Gazi Üniversitesi Endüstri Mühendisliği bölümünden mezun oldu. Yüksek Lisansını Karadeniz Teknik Üniversitesi İşletme bölümünde tamamladıktan sonra 2002 yılında Küçük ve Orta Ölçekli Sanayi Geliştirme ve Destekleme İdaresi Başkanlığı-Karadeniz Teknik Üniversitesi Teknoloji Geliştirme Merkezinde (KOSGEB-KTÜ TEKMER) çalışmaya başladı. Halen bu kurumdaki görevini 'KOBİ Uzmanı' olarak sürdürmektedir.