

**TABAKALI RASTGELE ÖRNEKLEMEDE ÖRNEKLEM  
BÜYÜKLÜKLERİNİN GENETİK ALGORİTMA İLE  
BELİRLENMESİ**

**DETERMINATION OF SAMPLE SIZES IN STRATIFIED  
RANDOM SAMPLING WITH GENETIC ALGORITHM**

**DERYA TURFAN**

**DOÇ. DR. ÖZGÜR YENİAY**  
**Tez Danışmanı**

Hacettepe Üniversitesi  
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin  
İstatistik Anabilim Dalı için Öngördüğü  
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2014

**DERYA TURFAN**'ın hazırladığı “**Tabakalı Rastgele Örneklemede Örneklem Büyüklüklerinin Genetik Algoritma İle Belirlenmesi**” adlı bu çalışma aşağıdaki jüri tarafından **İSTATİSTİK ANABİLİM DALI**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Doç. Dr. Mehtap AKÇİL OK  
Başkan

.....

Doç. Dr. Özgür YENİAY  
Danışman

.....

Doç. Dr. Çağdaş Hakan ALADAĞ  
Üye

.....

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma SEVİN DÜZ  
Fen Bilimleri Enstitüsü Müdürü

## ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

29/ 05 /2014

DERYA TURFAN

## ÖZET

# TABAKALI RASTGELE ÖRNEKLEMEDE ÖRNEKLEM BÜYÜKLÜKLERİNİN GENETİK ALGORİTMA İLE BELİRLENMESİ

**Derya TURFAN**

**Yüksek Lisans, İstatistik Bölümü**

**Tez Danışmanı: Doç. Dr. Özgür Yeniay**

**Mayıs 2014, 64 sayfa**

Araştırmacıların optimizasyon problemlerinin çözümüyle ilgili çalışmaları oldukça eskiye dayanmaktadır. Özellikle son yıllarda, optimizasyon sahasında çalışan bazı bilim adamları esnek ve performansı yüksek yöntemler geliştirmek için doğada var olan sistemlere yönelmişlerdir. Çünkü, doğada var olan karmaşık optimizasyon problemleri yine doğada işlemekte olan sistemlerle mükemmel bir şekilde çözülmektedir. Doğada kusursuz işleyen sistemlerden ilham alan yöntemler “Sezgisel Yöntemler” olarak bilinirler. Son yıllarda sezgisel yöntemler büyük boyutlu gerçek hayat optimizasyon problemlerinin çözümlenmesinde başarıyla kullanılmaktadır. Meta sezgisel yöntemlerin en bilinenlerinden biri “Genetik Algoritmalar” yöntemidir. Bu algoritmalar biyolojik organizmaların genetik sürecine ve popülasyonların doğadaki evrimleşme ilkelerine dayanır.

Yapılan tez çalışmasında, sezgisel ve sezgisel algoritmaların bir sınıfı olan meta sezgisel algoritmaların mantığı anlatılmış ve genetik algoritma yöntemi ayrıntılı olarak incelenmiştir. Günümüzde birçok alanda ihtiyaç duyulan örnekleme konusuna kısaca değinilmiş ve örnekleme yöntemlerinden biri olan tabakalı rastgele örnekleme yöntemi tanıtılmıştır. Tabakalı rastgele örneklemede amaç, tahmin varyansını minimum yapmak ve basit rastgele örneklemeyle kıyasla istatistiksel doğruluğu arttırmaktır. Dolayısıyla, tabaka sınırlarının belirlenmesi ve örneklem büyüklüklerinin tabakalara dağıtılması problemlerinin çözümü oldukça önemlidir. Bu problemlerin çözümü için bir genetik algoritma yaklaşımı geliştirilmiştir. İki gerçek, yedi tanesi simülasyon sonucu farklı dağılımlardan elde edilen veri setleri, hem geliştirilen genetik algoritma ile hem de literatürde yer alan eşit, orantılı ve Neyman yöntemleriyle çözümlenmiş ve elde edilen sonuçlar karşılaştırılmıştır. Sonuç olarak, geliştirilen genetik algoritmanın tahmin varyansını küçültmede daha iyi performansa sahip olduğu görülmüştür.

**Anahtar Kelimeler:** Sezgisel yöntemler, genetik algoritma, tabakalı rastgele örnekleme, örneklem dağıtımı

## **ABSTRACT**

### **DETERMINATION OF SAMPLE SIZES IN STRATIFIED RANDOM SAMPLING WITH GENETIC ALGORITHM**

**Derya TURFAN**

**Master of Science, Department of Statistics**

**Supervisor: Assoc. Prof. Dr. Özgür Yeniay**

**May 2014, 64 pages**

The studies done by researchers about solving optimization problems date back to a long time ago. Especially, in recent years, some scientists working on the field of optimization have tended to systems exist in nature to develop methods which are flexible and have high performance. The reason of this is that, the complicated natural optimization problems existed in nature are solved with the natural methods in a perfect balance. Optimization methods inspired from nature are known as “heuristic methods”. In recently years, meta heuristic methods have been successfully used to solve real world large scale optimization problems. One of the most known heuristic methods is “genetic algorithm”. These algorithms are based on the underlying genetic process in biological organisms and on the natural evolution principles of populations.

In this study, logic of heuristics and meta heuristics, a subclass of heuristics, is described and genetic algorithm is examined in detail. The topic of sampling is briefly mentioned and the stratified random sampling, one of sampling methods, is defined. One of the main objectives of stratified sampling is to reduce the variance of the estimator and to get more statistical precision than the simple random sampling. Therefore, the determination of stratum boundaries and sample allocation are the main problems in the application of the stratified sampling. In this study, a new genetic algorithm approach proposed in order to deal with these problems. Two real data sets and seven simulation data sets are analyzed with both the proposed genetic algorithm and equal, proportional, Neyman methods available in the literature used for sample allocation problem. The results obtained from the proposed algorithm are compared with the results of other classical

methods mentioned above. Consequently, it is observed that the proposed genetic algorithm has better performance than the other methods.

**Keywords:** Heuristic methods, genetic algorithm, stratified random sampling, sample allocation

## TEŞEKKÜR

Tez çalışmam boyunca yanımda olan, her zaman güler yüzüyle yardımlarını ve desteğini esirgemeyen, vakit ayıran kıymetli hocam ve danışmanım Sayın Doç. Dr. Özgür Yeniay'a,

Akademik hayatım boyunca her zaman bana yol gösterici ve örnek olan, her türlü konuda bilgi ve birikimlerini benimle paylaşan, büyük bir sabırla kıymetli zamanından ayırarak her türlü sorunumu dinleyen ve çözmek için çaba harcayan değerli hocam Sayın Doç. Dr. Çağdaş Hakan Aladağ'a,

Lisans hayatım boyunca ve hemen sonrasında iş hayatımda da aralarında bulunmaktan gurur duyduğum Hacettepe Üniversitesi İstatistik Bölümü'nün saygıdeğer hocalarına ve sevgili çalışma arkadaşlarıma,

Her zaman bana destek olan, sabır gösteren, sıkıntılı günlerimde manevi desteklerini hep yanımda hissettiğim ve birlikte çalışmaktan mutluluk duyduğum oda arkadaşlarım Deniz Taşçı ve Onur Toka'ya,

Ve beni bugünlere getiren, her zaman yanımda olup maddi ve manevi desteğini hiçbir zaman esirgemeyen, hayatta sahip olduğum her değeri borçlu olduğum çok kıymetli aileme,

teşekkürlerimi sunarım.



# İÇİNDEKİLER

	<b><u>Sayfa</u></b>
ÖZET .....	i
ABSTRACT .....	iii
TEŞEKKÜR .....	v
İÇİNDEKİLER.....	vi
ÇİZELGELER.....	viii
ŞEKİLLER .....	ix
SİMGELER VE KISALTMALAR .....	x
1.GİRİŞ.....	1
1.1. Sezgisel Yöntemler .....	3
1.1.1. Sezgisel Yöntem Nedir?.....	3
1.1.2. Sezgisel Yöntemlere Gerek Duyulmasının Sebepleri.....	3
1.1.3. Sezgisel Algoritmaların Değerlendirilmesi İçin Kriterler.....	4
1.1.4. Meta Sezgisel Algoritmalar .....	6
1.2. Genetik Algoritmalara Giriş.....	7
1.2.1. Evrimsel Hesaplama .....	7
1.2.2. Genetik algoritmalarla ilgili temel bilgiler .....	8
2. GENETİK ALGORİTMA.....	10
2.1. Genetik Algoritmanın Tanımı ve Önemi .....	10
2.2. Genetik Algoritmanın Tarihçesi .....	10
2.3. Genetik Algoritmanın Temel Kavramları .....	11
2.3.1. Gen.....	12
2.3.2. Kromozom (Çözüm).....	12
2.3.3. Popülasyon .....	13
2.3.4. Uygunluk Fonksiyonu .....	13
2.4. Genetik Algoritmada Kullanılan Seçim Yöntemleri .....	13
2.4.1. Elitizm Yöntemi .....	14
2.4.2. Rulet Çarkı Yöntemi.....	14
2.4.3. Turnuva Yöntemi .....	14
2.5. Genetik İşlemler .....	15
2.5.1. Üreme.....	15
2.5.2. Çaprazlama.....	16
2.5.2.1. Tek Noktalı Çaprazlama.....	16
2.5.2.2. Çok Noktalı Çaprazlama .....	17

2.5.2.3. Pozisyona Dayalı Çaprazlama .....	17
2.5.2.4. Sıraya Dayalı Çaprazlama .....	18
2.5.2.5. Kısmi Planlı Çaprazlama.....	18
2.5.3. Mutasyon .....	19
2.6. Genetik Algoritmanın Parametreleri.....	20
2.6.1. Popülasyon Büyüklüğü .....	21
2.6.2. Çaprazlama Oranı .....	21
2.6.3. Mutasyon Oranı.....	22
2.6.4. Durdurma Kriteri .....	23
2.7. Genetik Algoritmanın Aşamaları ve Akış Şeması.....	24
2.7.1. Genetik Algoritmanın Aşamaları .....	25
2.7.2. Genetik Algoritmanın Akış Şeması .....	26
2.8. Genetik Algoritmanın Özellikleri .....	28
2.8.1. Genetik Algoritmanın Avantajları.....	28
2.8.2. Genetik Algoritmanın Dezavantajları .....	30
2.9. Genetik Algoritmanın Uygulama Alanları.....	30
2.10. Genetik Algoritmanın Kullanılma Nedenleri.....	33
3. TABAKALI RASTGELE ÖRNEKLEME .....	34
3.1. Örneklem Nedir? .....	34
3.2. Tabakalı Rastgele Örneklem .....	35
3.2.1. Problemin Tanımı .....	36
3.2.2. Problemin Formülasyonu .....	38
4. GELİŞTİRİLEN GENETİK ALGORİTMA.....	41
4.1. Çözümlerin Kodlanması.....	41
4.2. Başlangıç Popülasyonu Oluşturma .....	42
4.3. Uygunluk Fonksiyonu .....	43
4.4. Seçim Yöntemi.....	43
4.5. Çaprazlama.....	43
4.6. Mutasyon. ....	44
5. UYGULAMA .....	46
6. SONUÇLAR.....	57
KAYNAKLAR .....	60
ÖZGEÇMİŞ .....	64

# ÇİZELGELER

	<u>Sayfa</u>
Çizelge 1.1. Meta sezgisel sınıflandırması.....	7
Çizelge 3.1. Tabakalı örneklemenin genel gösterimi.....	39
Çizelge 5.1. Uygulamada kullanılan veri setlerinin özellikleri.....	47
Çizelge 5.2. Geliştirilen genetik algoritmanın parametre değerleri.....	48
Çizelge 5.3. Simülasyon ile elde edilen veri setleri için bulunan tahmin varyansı değerleri .....	48
Çizelge 5.4. Gerçek veri setleri için bulunan tahmin varyansı değerleri.....	49
Çizelge 5.5. İSO 483 verileri için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	50
Çizelge 5.6. İSO 469 verileri için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	51
Çizelge 5.7. Normal dağılımdan ( $\mu = 100, \sigma = 10$ ) elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	51
Çizelge 5.8. $\chi_1^2$ dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	52
Çizelge 5.9. $\chi_5^2$ dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	53
Çizelge 5.10. $\chi_{10}^2$ dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	53
Çizelge 5.11. $\chi_{15}^2$ dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	54
Çizelge 5.12. Beta dağılımından ( $a=10, b=3$ ) elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	55
Çizelge 5.13. Tekdüze dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları.....	55

## ŞEKİLLER

	<u>Sayfa</u>
Şekil 1.1. Evrimsel algoritmanın genel yapısı.....	8
Şekil 2.1. Tek noktalı çaprazlamaya bir örnek.....	16
Şekil 2.2. Çok noktalı çaprazlamaya bir örnek.....	17
Şekil 2.3. Pozisyona dayalı çaprazlamaya bir örnek.....	17
Şekil 2.4. Sıraya dayalı çaprazlamaya bir örnek.....	18
Şekil 2.5. Kısmi planlı çaprazlamaya bir örnek(1. Adım).....	18
Şekil 2.6. Kısmi planlı çaprazlamaya bir örnek(2. Adım).....	18
Şekil 2.7. Mutasyon işlemi için genel bir örnek.....	19
Şekil 2.8. Farklı mutasyon türleri için örnekler.....	20
Şekil 2.9. Genetik algoritmanın genel işleyişi.....	27
Şekil 3.1. Tabakalı örnekleme.....	38
Şekil 4.1. Basit bir çözüm kodlama örneği.....	40
Şekil 4.2. Tek nokta çaprazlamaya basit bir örnek (şartlı).....	42
Şekil 4.3. Karşılıklı değişim mutasyonuna basit bir örnek.....	43

## SİMGELER VE KISALTMALAR

K	Popülasyon Büyüklüğü
$\ell$	Kromozom Uzunluğu
N	Kitle Büyüklüğü
n	Örneklem Büyüklüğü
$N_h$	h-ıncı Tabaka Büyüklüğü ( $h=1, \dots, L$ )
$n_h$	h-ıncı Tabakadan Seçilen Örneklem Büyüklüğü
L	Tabaka Sayısı
$y_{hi}$	h-ıncı Tabakada i-inci Birim Değeri
İSO	İstanbul Sanayi Odası

# 1. GİRİŞ

Optimizasyon, eniyileme anlamına gelmektedir. Bir problemde belirli kısıtlar altında mümkün olan alternatifler içinden en iyisini seçme işlemidir. Belirli kısıtları sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem, optimizasyon problemi olarak adlandırılabilir [1]. Optimizasyon problemlerinin çözümü için literatürde birçok farklı yöntem önerilmiştir. Son yıllarda gerçek hayat optimizasyon problemlerinin çözümü için en çok kullanılan yaklaşımlardan biri sezgisel (heuristic) algoritmalarıdır. Sezgisel algoritmalar, çözümü zor, büyük boyutlu optimizasyon problemleri için, kabul edilebilir sürede optimuma yakın çözümler verebilen algoritmalarıdır.

Günümüz optimizasyon problemlerinin karmaşık yapısını ele alabilecek sezgisel yöntemlerin geliştirilmesi konusu bir çok araştırmacı tarafından çalışılmaktadır. Her yöntemin özel yapısını oluşturan fikirler, bazen hiç umulmadık bağlantılardan ortaya çıkabilir. Örneğin Tavlama Benzetimi (Simulated Annealing), metalürjideki fiziksel süreçler üzerine dayanır. Karınca Algoritması (Ant Algorithm), ortak bir problemi çözmek için işbirliği yapan bir karınca kolonisi gibi davranır. Tabu Arama'nın (Tabu Search) felsefesi, öğrenme ve hafıza kullanımı gibi zeki problem çözme prensiplerinin bir derlemesini kullanmaktır. Yapay Arı Kolonisi (Artificial Bee Colony) algoritması, arıların yiyecek arama davranışlarını taklit eder. Genetik Algoritmalar (Genetic Algorithms) ise evrimsel yeniden üretimin biyolojik olgusunu taklit eder [2]. Bir başka ifadeyle, genetik algoritmalar, biyolojik evrimin işleyiş biçimini örnek alır.

Genetik algoritmaların taklit ettiği süreç şu şekilde açıklanabilir [3]:

Doğal popülasyonlar, Charles Darwin'in, Türlerin Kökeni (The Origin of Species) adlı kitabında belirttiği, doğal seçme (natural selection) ve en iyi uyum yapanın yaşaması (survival of the fittest) prensibine göre nesiller (generation) boyunca gelişmektedir [4]. Doğada, bireyler arasındaki yiyecek, su ve barınak gibi kıt kaynaklar ve eşler için yapılan mücadeleler, yüksek uyumlu (highly adapted) ya da uygun (fit) bireylerin daha zayıf olanlara üstünlüğü ile sonuçlanır. İyi uyum sağlayan bireyler yaşar ve daha fazla yavru sahibi olur. Düşük uyumlu bireyler az sayıda, belki de hiç yavru sahibi olamayacaktır [5]. Bu, uygun bireylerin genlerinin sonraki her nesilde daha fazla bireye dağılması demektir. Farklı ebeveynlerden gelen iyi özelliklerin birleşimi, bazen her bir ebeveyninden daha büyük uygunluğa

sahip üstün yavrular verebilir. Böylece türler, bulunduğu çevre için giderek daha uygun olurlar [6]. Örnek olarak bir tavşan popülasyonu düşünölsün. Bazı tavşanlar diğçerlerine göre daha hızlı ve daha kurnazdır. Bu tavşanların tilkiler tarafından yenilme olasılığı daha düşöktür. Çoğu yaşamını sürdürür. Kuşkusuz yavaş ve durgun olan tavşanların bazıları da şanslı oldukları için yaşayacaklardır. Bu yaşayan tavşan popülasyonu üremeye başlayacaktır. Popölasyonun çoğunluğu, tilkilerden kurtulan daha hızlı ve kurnaz tavşanlar olduğundan, yavru tavşanlar önceki popölasyondakilerden ortalama olarak daha hızlı ve kurnaz olacaktır [4].

Genetik algoritmalar, Bilgisayar Bilimi, Mühendislik, Yöneylem Araştırması, Sosyal Bilimler, Tıp, Matematik, İstatistik vb. alanlarda karşılaşılan çeşitli problemlerin çözümünde kullanılmaktadır. Diğçer sezgisel yöntemlerde de olduğü gibi, genetik algoritmalar optimal çözümlü bulmayı garanti etmezler, ancak kabul edilebilir hızla, kabul edilebilir ölçüde iyi (optimale yakın) çözümler bulunmasında genel olarak başarılıdırlar.

Tezin giriş bölümünde genel bilgiler verilmiştir. Sezgisel yöntem, bu yöntemlerin bir sınıfını oluşturan meta sezgisel yöntemler ve genetik algoritmalar hakkında genel bilgiler sunulmuştur. Tezin ikinci bölümünde genetik algoritmaların tanımı, genel işleyişi ve yöntemin çalışmasını etkileyen faktörler incelenmiştir. Tezin üçüncü bölümünde, ilk önce örnekleme konusundan genel olarak bahsedilmiş ve problemin temelini oluşturan, örneklemenin bir alt başlığı olan tabakalı rastgele örnelemeye yer verilmiştir. Tabakalı rastgele örnelemede, her tabakanın örnekleme büyüklüğünün genetik algoritma ile elde edilmesi problemi tanıtılmıştır. Tezin dördüncü bölümünde, tanımlanan problemi çözmek için geliştirilen genetik algoritma anlatılmış, geliştirilen genetik algoritmayı kullanan ve kullanıcının belirleyeceği bazı parametreleri göz önünde bulundurarak tabaka sınırlarını ve tabakalardaki örnekleme büyüklüklerinin bulunması amacıyla yazılan bilgisayar programı tanıtılmıştır. Uygulama kısmında ise tezde sunulan çalışma özetlenmiş, yazılan programla elde edilen sonuçlar ile literatürde kullanılan diğçer yöntemlerin sonuçları karşılaştırılmış ve çizelgeler halinde sunulmuştur. Son olarak altıncı bölümde de sonuçlar yorumlanmıştır.

## 1.1. Sezgisel Yöntemler

### 1.1.1. Sezgisel Yöntem Nedir?

Sezgisel kelimesi, eski Yunanca'da "problemleri çözmek için yeni yöntemler geliştirme" ya da "problem çözüme sanatı" anlamına gelen *heuriskein* kelimesinden gelmektedir. Bilgisayar bilimlerinde ve yapay zekâda, "Sezgisel" terimi zeki aramada (intelligent search) yer alan yöntemler için kullanılır. Bu anlamda "sezgisel arama", en az umut vaat eden yolları atlar ve en çok umut vaat eden yol boyunca ilerleyerek bir çözüme ulaşmak için tüm mevcut bilgileri kullanır. Burada amaç, önceden toplanmış verilerin içerdiği bilgiye bağlı olarak arama sürecini, başarısız çözümleri incelemekten uzak tutmaktır [2].

Yöneylem araştırmasında "sezgisel" terimi, tam çözümü bulunması zor olan problemler için makul çözümler bulmada kullanılan yöntemler için kullanılır. Özel olarak optimizasyonda bir sezgisel yöntem, optimale yakın ya da yaklaşık olarak optimal bir çözüme götüren (fakat bunu garanti etmeyen) stratejilere dayanan, hızlı ve pratik bir yöntem anlamına gelir. Buna göre, bu sezgisel yöntemlerden söz edildiğinde, "çözmek" fiili, "optimale, tatmin edici bir yakınlık bulmak" yan anlamına sahiptir. İyi sezgisel yöntemler, prensipte, izin verilen zamanda elde edilebilecek en iyi çözümü belirleyebilmelerine rağmen optimal bir çözüm bulmayı garanti etmezler [2].

Reeves'in sezgisel yöntemler için tanımı şu şekildedir [7]: "Bir sezgisel yöntem, kabul edilebilir hesaplama maliyetleriyle optimalliği garanti etmeksizin optimale yakın çözümü arayan yöntemdir."

Sezgisel algoritmalar, büyük boyutlu optimizasyon problemleri için, kabul edilebilir sürede optimuma yakın çözümler verebilen algoritmalarlardır. Bu algoritmalar, çözüm uzayında optimum çözüme yakınsaması ispat edilemeyen algoritmalar olarak da bilinir. Bu tür algoritmalar yakınsama özelliğine sahiptir, ama kesin çözümü garanti edemezler ve sadece kesin çözüm yakınında bir çözümü garanti edebilirler [8].

### 1.1.2. Sezgisel Yöntemlere Gerek Duyulmasının Sebepleri

Optimizasyon problemlerinde tam (exact) yöntemler, problemin varsa optimal çözümünü makul bir zamanda bulmayı garanti eden yöntemlerdir. Buna karşın, tam yöntemlerin çözüm uzayının çok büyük veya karmaşık olduğu problemlerde uygulanabilirliği oldukça azdır [9].



Gerçek yaşam problemlerinin çoğunda problemin çözüm uzayı sonsuz olabilir veya tüm çözümlerin makul bir sürede değerlendirilmesi mümkün olmayabilir. Makul bir sürede çözümlerin değerlendirilerek iyi bir çözümün bulunması gerekmektedir. Böyle problemlerin çözümünde, kabul edilebilir bir sürede çözümlerin değerlendirilmesi, aslında tüm çözüm uzayında “bazı çözümlerin” değerlendirilmesi anlamına gelmektedir. Bazı çözümlerin neye göre ve ne şekilde değerlendirileceği, yani nasıl bir strateji izleneceği ise kullanılacak sezgisel yönetime göre değişiklik gösterir.

Problemlerin çözümünde sezgisel yöntemlere gerek duyulmasının başlıca sebepleri şu şekilde açıklanabilir [8]:

- ✓ Optimizasyon problemi, kesin çözümü bulma işleminin tanımlanamadığı bir yapıya sahip olabilir.
- ✓ Anlaşılabilirlik açısından sezgisel algoritmalar karar verici için çok daha basit olabilir.
- ✓ Sezgisel algoritmalar, öğrenme amaçlı ve kesin çözümü bulma işleminin bir parçası olarak kullanılabilir.
- ✓ Matematik formülleriyle yapılan tanımlamalarda genellikle gerçek hayat problemlerinin en zor tarafları (hangi amaçlar ve hangi kısıtlar kullanılmalı, hangi alternatifler test edilmeli, problem verisi nasıl toplanmalı) ihmal edilir. Model parametrelerini belirleme aşamasında kullanılan verinin hatalı olması, sezgisel yaklaşımın üretebileceği alt optimal çözümden daha büyük hatalara sebep olabilir.

Yukarıda yapılan tanımlamalardan ve açıklamalardan da anlaşılacağı üzere, bir problem için optimal çözümü bulmak çok zor veya imkansız olabilir. Tam yöntemler, matematiksel olarak tanımlanamayan ya da çözüm uzayı çok büyük olan gerçek hayat problemlerinde etkili olamazlar. Literatürde yapılan birçok çalışmada, sezgisel yöntemlerin bu tür problemler için çok kullanışlı oldukları kanıtlanmıştır [10].

### **1.1.3. Sezgisel Algoritmaların Değerlendirilmesi İçin Kriterler**

Sezgisel algoritmaların değerlendirilmesinde kullanılan kriterler genel olarak şu şekildedir [8]:

- ✓ Çözüm kalitesi ve hesaplama zamanı (Quality of solution and computation time)

Çözüm kalitesi ve hesaplama zamanı herhangi bir algoritmanın etkinliğinin değerlendirilmesi için önemli kriterlerdir. Bundan dolayı iyi bir sezgisel algoritma, ayarlanabilir parametreler setine sahip olmalı ve bu parametreler kullanıcıya önemlilik açısından hesaplama maliyeti ile çözüm kalitesi arasında bir vurgulamanın yapılabilmesine imkan vermelidir. Yani, çözüm kalitesi ve hesap zamanı arasındaki ilişki kontrol edilebilmelidir.

- ✓ Algoritma basitliği ve gerçekleştirilebilirlik (Ease of algorithm and realizability)

Sezgisel algoritma prensipleri basit olmalı ve genel olarak uygulanabilir olmalıdır. Bu durum, problem yapısı ile ilgili başlangıçta çok az bilgiye sahip olunması halinde bile sezgisel algoritmanın yeni alanlara kolaylıkla uygulanabilmesini sağlar.

- ✓ Esneklik (Flexibility)

Sezgisel algoritmalar modelde, kısıtlarda ve amaç fonksiyonlarında yapılacak değişiklikleri kolayca karşılayabilmelidir.

- ✓ Sağlamlık (Robustness)

Sezgisel algoritma, başlangıç çözümünün seçimine bağlı olmaksızın her zaman yüksek kaliteli, kabul edilebilir çözümleri üretebilme kabiliyetine sahip olmalıdır.

- ✓ Basitlik ve analiz edilebilirlik (Simplicity and analyzability)

Karmaşık algoritmalar, esneklik ve çözüm kalitesi açısından basit algoritmalarından daha zor analiz edilebilmektedir. Sezgisel algoritma kolayca analiz edilebilir olmalıdır.

- ✓ Etkileşimli hesaplama ve teknoloji değişimleri

Sezgisel algoritma içinde insan-makine etkileşimini kullanma fikri çoğu sistemde yaygın olarak gerçekleştirilmektedir. Herkes tarafından bilindiği gibi iyi bir kullanıcı ara-yüzü herhangi bir bilgisayar sistemini veya algoritmayı daha çekici yapmaktadır. Bunun en önemli avantajı çözümlerin grafiksel olarak sergilenebilmesidir.

#### 1.1.4. Meta Sezgisel Algoritmalar

Aladağ [2] tez çalışmasında meta sezgisel yöntemleri şu şekilde açıklamıştır:

“Sezgisel algoritmaların bir sınıfı meta sezgisel algoritmalarıdır. Meta sezgisel terimi, tabu aramadan ilk kez bahsedilen makalede [11] geçmiştir ve daha sonra literatürde yaygın olarak kullanılmaya başlanmıştır [12,13,14,15,16].

Meta sezgisel algoritmaların en bilinen örnekleri arasında Genetik Algoritmalar, Tabu Arama, Tavlama Benzetimi ve Yapay Sinir Ağları yer almaktadır.

Meta sezgisel algoritmalar, bazı yerel arama yöntemlerini ve ileri düzey farklı stratejileri kullanan yapılara sahip algoritmalarıdır. Bir meta sezgisel yöntem, ileri düzey işlemler ya da ilgili bir değerlendirme kuralı ile beraber, bir çözümü başka bir çözüme dönüştürmek için mümkün hareketlerin (move) bir tanımından da ibaret olabilir [17].

Meta sezgisel yöntemler kendi aralarında sınıflandırılabilirler. Glover ve Laguna'ya göre meta sezgisel yöntemlerin sınıflandırılmasında dikkate alınan üç temel dizayn seçeneği, (1) Uyarlamalı (adaptive) hafıza kullanımı (2) Kullanılan komşuluk (neighborhood) keşfinin çeşidi (3) Bir iterasyondan bir sonraki iterasyona taşınan mevcut çözümlerin sayısıdır [18].

Bu seçenekler bir sınıflama planı içine yerleştirilir. Bu planda, (1)'in seçenekleri; eğer meta sezgisel yöntem uyarlamalı hafıza kullanıyorsa 'A' ve eğer yöntem hafızasız ise 'M' dir. (2)'nin seçenekleri; bir sonraki hareketi seçmek ya da verilen bir çözümü geliştirmek amaçlarından biri için sistematik komşuluk aramayı kullanan yöntemler için 'N' ve rastgele örnekleme güvenen yöntemler için 'S' dir. Son olarak (3)'ün seçenekleri ise eğer her iterasyondan sonra yöntem mevcut bir çözümden bir sonraki çözüme hareket ediyorsa '1' ve eğer yöntem, her bir iterasyonda tek bir çözüm yerine çözümlerin bir derlemine kullanan popülasyon temelli bir strateji kullanıyorsa 'P' dir. Bu basit üç boyutlu plan, meta sezgisellerin sınıflandırılması için bir temel teşkil eder. İyi bilinen meta sezgisel yöntemlerin bazıları için yapılan sınıflandırma Çizelge 1.1' de gösterilmektedir [18]:

**Çizelge 1.1.** Meta sezgisel sınıflandırması

<b>Meta-sezgisel</b>	<b>1. Sınıflama</b>	<b>2. Sınıflama</b>
Genetik Algoritmalar	M / S / P	M / N / P
Tavlama Benzetimi	M / S / 1	M / N / 1
Tabu Arama	A / N / 1	A / N / P

Görüldüğü gibi meta sezgisellerin sınıflandırılması iki farklı şekilde yapılmıştır. Bu sınıflandırmalar arasındaki farklılıklar, yöntemle ilgili olan değişik nedenlerden ileri gelir. Bazı farklılıklar yöntemler üzerinde çalışan araştırmacıların bazıları tarafından öne sürülen yeni değişimlerden kaynaklanırken, bazı farklılıklar ise yöntem ilk önerildiği zaman ortaya çıkmıştır [18].”

## **1.2. Genetik Algoritmalar Giriş**

Genetik algoritmalar evrime dayalı algoritmalarından biri olduğu için, bu bölümde önce evrimsel algoritmalar hakkında kısa bir giriş yapılacak, daha sonra genetik algoritmalarla ilgili genel bilgilerden kısaca bahsedilecektir.

### **1.2.1. Evrimsel Hesaplama**

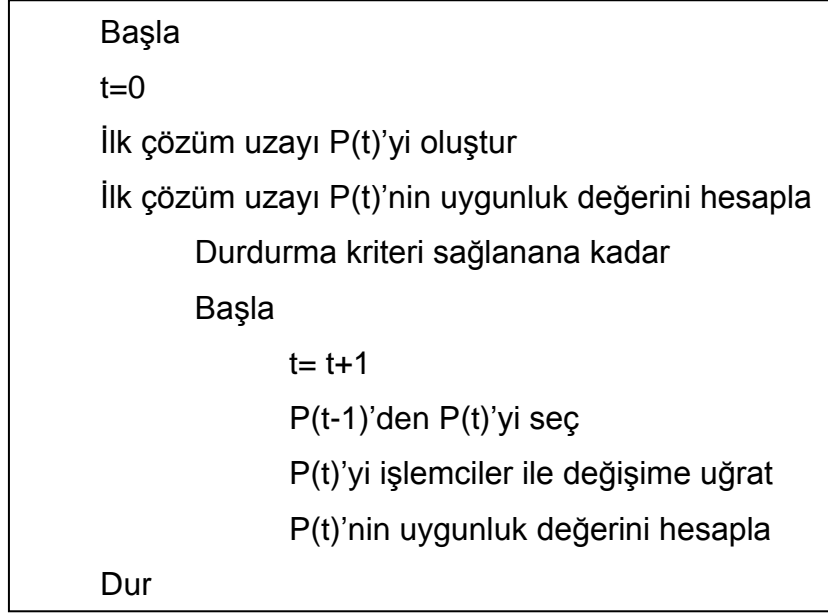
Evrimin temel ilkesi kısaca, çevre şartlarına en iyi uyum sağlayan bireylerin hayatta kalması olarak ifade edilebilir. Evrimin bu temel ilkesinden yola çıkılarak oluşturulmuş arama yöntemlerinin tümü, evrimsel algoritmalar (evolutionary algorithms) veya evrimsel hesaplama yöntemleri olarak adlandırılmaktadır. Son 30 yıldır, evrim ilkesine dayanan çözüm yöntemleri ile ilgili araştırmacıların ve literatürdeki çalışmaların sayısı hızla artış göstermektedir [19].

Evrimsel hesaplama yöntemleri sınıfına giren algoritmalara örnek olarak genetik algoritmalar, evrimsel programlama (evolutionary programming), evrimsel stratejiler (evolution strategies) genetik programlama (genetic programming) ve diferansiyel gelişim algoritmaları sayılabilir. Bir problemi çözmeye kullanılacak herhangi bir evrimsel algoritma genel olarak aşağıdaki elemanlara ihtiyaç duymaktadır [8]:

- ✓ Problem için çözümlerin genetik temsili (representation),
- ✓ Çözümlerin başlangıç popülasyonunu oluşturacak bir yöntem,
- ✓ Çözümlerin uygunluk açısından değerlendirilmesini sağlayacak değerlendirme fonksiyonu (evaluation function),
- ✓ Genetik yapıyı değiştirecek işlemciler (operatörler),

- ✓ Kontrol parametrelerinin değerleri (popülasyon büyüklüğü, işlemcileri uygulama olasılıkları vb.).

Evrimsel algoritmaların genel işleyişi Şekil 1.1' de gösterilmektedir:



**Şekil 1.1.** Evrimsel algoritmanın genel yapısı [19]

Evrimsel algoritmalar tek bir bireyle (çözümle) değil, bireylerin popülasyonun (çözüm kümesi) ile ilgilenir. Yukarıdaki tabloda da görüldüğü gibi, bu algoritma her t. iterasyonda bireylerden oluşan bir P(t) popülasyonu içerir. Her bir birey söz konusu problem için potansiyel bir çözümü ifade eder. Her çözümün uygunluk (fitness) değeri hesaplanır ve bir sonraki iterasyonda uygunluk değeri yüksek bireyler seçilerek yeni bir popülasyon oluşturulur. Yeni popülasyonun bazı bireyleri yeni çözümler oluşturmak için bazı işlemciler aracılığıyla değişime uğratılır. Yeni popülasyonun uygunluk değeri hesaplanır ve tüm bu iterasyonlar durdurma kriteri sağlanana kadar devam eder.

### 1.2.2. Genetik algoritmalarla ilgili temel bilgiler

Genetik algoritmalar biyolojik evrimin işleyişini taklit eden süreçlerdir. Bu algoritmalar genetik ve doğal seçilimi temel alan evrimsel algoritmaların başını çekmektedir. Genetik algoritmalar ilk kez Holland tarafından ortaya konmuştur [20]. Holland, arkadaşları ve öğrencileri ile birlikte bu yaklaşımı geliştirmiş ve ilk çalışmaları sonucunda "Adaptation in Naturel and Artificial Systems" isimli kitabını çıkartmıştır [20]. Popülasyon temelli bir sezgisel algoritma olan genetik algoritmalar, doğal seçme (natural selection) ile canlılarda bulunan genetik gelişimin taklidini gerçekleştirmektedir. Bahsedilen algoritma, diğer evrimsel

algoritmalar gibi arama uzayında bulunan çözümlerin bazılarının oluşturduğu bir başlangıç popülasyonunu (initial population) kullanmaktadır. Başlangıç popülasyonu her nesilde, doğal seçim ve tekrar üreme (reproduction) işlemleri yardımıyla art arda geliştirilir. En son neslin en uygun (fittest) yani en kaliteli bireyi, problem için optimal çözüm olmaktadır. Bu çözüm her zaman optimal olmayabilir ama kesinlikle optimale yakın bir çözüm olacaktır [8].

Holland [20], basit bit dizileri (bit strings) kullanarak karmaşık yapıların kodlanabileceğini göstermiştir. Bu yapılar, çözülecek problem için çözümleri temsil etmektedir. Bunlar, muhtemel tüm çözümleri içine alan arama uzayından alınır ve bu dizilerin veya çözümlerin belirli bir miktarı genetik algoritmanın kullanacağı popülasyonu oluşturur. Daha sonra temel genetik işlemcilerin belirli bir seti, art arda gelen nesillerde çözümleri geliştirmek amacıyla kullanılır. Eğer bu işlem uygun şekilde kontrol edilirse, çözüm popülasyonunun ortalama kalitesi çok hızlı olarak gelişme gösterir. Yani, çözülecek probleme çok iyi uyarlanmış yapıları içeren çözüm kümesinin ortaya çıkması sağlanır [8].

Herhangi bir problem genetik algoritma ile çözülecekse, genel olarak izlenecek adımlar şu şekilde özetlenebilir [3]:

- ✓ Genetik algoritmayı kullanmadan önce, problem için uygun bir kodlama (gösterim) tasarlanır.
- ✓ Başlangıç popülasyonu, genellikle rastgele yaratılır. Ancak sezgisel seçimin uygun olduğu bazı durumlar da vardır.
- ✓ Başlangıç popülasyonundaki her bir kromozomun uygunluk değeri hesaplanır.
- ✓ Popülasyonu değiştirmek için seçme, çaprazlama (crossover) ve mutasyon (mutation) işlemcileri kullanılır.
- ✓ Yeni popülasyona, uygunluk değerleri atanarak test edilir ve bu işlemlere durdurma kriteri sağlanıncaya kadar devam edilir.

## 2. GENETİK ALGORİTMA

### 2.1. Genetik Algoritmanın Tanımı ve Önemi

Genetik algoritmalar, doğadaki canlıların geçirdiği süreci örnek alır ve iyi nesillerin kendi yaşamlarını korurken, kötü nesillerin yok olması ilkesine dayanır. Matematiksel modellemenin yapılamadığı veya kesin çözümün olmadığı problemlerde genetik algoritmadan yararlanılır [21].

Goldberg'e göre genetik algoritma, rastgele arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre esasına dayanan sezgisel bir yöntemdir [21].

Genetik algoritmalar, karmaşık düzenli problemlerin çözümünü gerçekleştirmek amacıyla, kromozomların yeniden üreme esasını temel alan, sezgisel bir arama yöntemidir [22,23].

Genetik algoritmaların çalışma prensibi, genetik aşamaları kullanıp ve bu aşamalara benzetim yaparak arama yapmasıdır. Genel olarak, algoritma şu şekilde çalışmaktadır: Kromozomlardan oluşan bir popülasyon ile aramaya başlayıp, bu popülasyon içerisinde yeni ve daha iyi bireyler üreterek, kromozomları sürekli olarak değişime uğratarak, problem için alternatif çözümleri artırarak en iyi çözüme ulaşmaktadır [24].

Genetik algoritmaların karmaşık arama uzaylarında iyi sonuçlar elde edebildiği teorik ve görsel olarak kanıtlanmıştır [25]. Genetik algoritmanın hızlı bir şekilde yaygınlaşmasının nedeni hesaplama olarak basit ve en iyiyi arama mekanizması konusunda güçlü bir yapıya sahip olmasıdır [26].

### 2.2. Genetik Algoritmanın Tarihçesi

Genetik algoritma, Darwin ve onu izleyen bilim adamlarının çalışmalarının en iyileme problemleri üzerine uygulanmasıdır. "Genetik Algoritma" kalıbı, ilk olarak 1967 yılında Bagley tarafından bir oyun programının yenmek üzere tasarlanmasında kullanılmıştır. Bagley ile aynı tarihte Rosenberg, biyolojik ve simülasyon esaslı bir çalışma yapmıştır [25]. Bagley'in kullandığı genetik algoritma yöntemleri bugünkünden çok farklı olmakla birlikte günümüz genetik algoritmaları için temel olarak kullanılmıştır. Rosenberg de aynı dönemde bu algoritmaya biyolojik ve benzetimsel etmenleri eklemiştir. De Jong, 1975 yılında matematiksel fonksiyonları genetik algoritma ile çözmeye çalışmıştır [27].

Bagley, Rosenberg, De Jong gibi öncülerinin bulunmasına rağmen, genetik algoritmanın babası Holland olarak kabul edilmektedir. Çünkü Holland, "Genetik Algoritma" kavramını "Cellular Automata" çalışmaları ve "Doğal ve Yapay Sistemlerde Uyarılma" kitabının yayınlanması ile literatüre kazandırmıştır. Holland'ın bu kitabında ana düşünce şudur: "Verilen bir popülasyonun genetik havuzu potansiyel olarak istenen en iyi çözümü içerir veya uyarılan probleme ilişkin iyi bir çözüm vardır" [28]. Holland, genetik algoritmanın temel ilkeleri olan yeniden üreme ve çaprazlama işlemcilerini tanımlamıştır. Holland'dan sonra genetik algoritmaların gelişimi doktora öğrencisi olan Goldberg ile devam etmiştir. David E. Goldberg'in 1985 yılındaki çalışmaları genetik algoritmanın gelişimini sağlamıştır. Bu çalışmada Goldberg'in amacı, doğalgaz borularındaki kayıpları, basınç oranını değiştirmek suretiyle minimize etmektir. Dinamik programlama yardımı ile Wong ve Larson tarafından basınç oranları hesaplanan bu problem, Goldberg tarafından genetik algoritma ile çözülmeye çalışılmıştır [27].

Genetik algoritma, kısıtlı optimizasyon problemleri kapsamına giren çizelgeleme yöntemlerinden biri olan atölye çizelgeleme problemlerinde etkili olarak ilk defa Davis tarafından 1985 yılında kullanılmıştır. 1987 yılında Liepis ise ilk defa genetik algoritma yapısını iki makineli çizelgelemeye uygulamıştır. Biegal ve Daven 1990 yılındaki çalışmalarında atölye çizelgelemede genetik algoritmayı bütünleşmiş imalat çevrimi içinde kullanmışlar ve bu yapıyı tek, iki ve çok makineli sistemlere uygulamışlardır. Nakano ise 1996'da genetik algoritmaların ikili kod sisteminde gösterimini atölye çizelgeleme probleminde kullanmıştır. Spears ve De Jong 1991 yılındaki çalışmalarında iki noktalı çaprazlama işlemcisinin her zaman bir noktalı çaprazlama işlemcisinden daha etkili olduğunu belirtmiştir. Chen ve arkadaşları genetik algoritmaların literatürdeki diğer sezgisel yaklaşımlardan daha iyi sonuç verdiğini 1995 yılındaki çalışmalarında göstermişlerdir [21].

### **2.3. Genetik Algoritmanın Temel Kavramları**

Genetik algoritmalar, isminden de anlaşılacağı üzere genetik bilimi ile benzerlik gösterdiğinden terminolojileri birbirlerine oldukça yakındır. Bu başlık altında, genetik algoritmanın temel kavramlarının açıklanmasına yer verilmiştir.



### **2.3.1. Gen**

Bir canlıya ait genetik özelliklerden herhangi birini taşıyan en küçük yapı birimine gen denir. Kısmi bilgiler taşıyan bu küçük yapıların bir araya gelmesiyle kromozomlar meydana gelir.

Kromozom üzerinde, belli bir konumda bulunan genler, temsil edilen değişkenin değerini ifade etmektedir [29]. Dolayısıyla, problemin sahip olduğu değişken sayısı aynı zamanda bir kromozomun sahip olduğu gen sayısını göstermektedir.

Başlangıçta oluşturulmuş olan kromozomlardaki genlerin sırasının, genetik algoritma işlemleri boyunca kesinlikle değiştirilmemesi gerekmektedir.

Genetik algoritmaların kullanıldığı programlama yapısında bu gen yapıları programcının tanımlamasına bağlıdır. Bir genin içerdiği bilgi sadece ikili tabandaki sayıları içerebileceği gibi onluk taban ve onaltılık tabandaki sayı değerlerini de içerebilir. Dolayısıyla yazılan programa göre gen içeriği çok önem kazanmaktadır [21].

### **2.3.2. Kromozom (Çözüm)**

Genlerin bir dizi halinde sıralanması ile ortaya çıkan genler dizisine “kromozom” adı verilmektedir. Böylece bir kromozomda, problemdeki karar değişkenlerinin her birinin bir arada bir dizi halinde bulunduğu anlaşılmaktadır [30].

Genetik algoritma, problemin çözüm (kromozom) alternatiflerini içeren bir rastgele çözüm kümesi ile başlatılmaktadır. Her kromozom, temsil ettiği çözüme ait bilgileri içermektedir ve bilgiler bir dizi halinde kodlanmaktadır.

Kromozom üzerinde yer alacak bilgiler; genetik algoritmanın, çözüm uzayında en iyi çözümü araması için gerekli olan bilgilerin tümünü içerecek şekilde tasarlanmalıdır [31]. Bu şekilde tasarlanan kromozomlar ile daha etkili sonuçları üretmek mümkün olmaktadır [32].

Kromozomlar, genetik algoritma yaklaşımında üzerinde durulan en temel birim olduğundan, bilgisayar ortamında iyi ifade edilmeleri gerekmektedir. Kromozomun hangi kısmının ne anlam taşıyacağı, ne tür bilgi içereceği kullanıcının olaya bakışını değiştirmektedir [21].

Kromozomların kodlanması esnasında, temel olarak ikili sayı (binary) sistemi kullanılsa da, tam sayı ve reel sayılar da kromozom kodlamalarında

kullanılabilmektedir. Problemlerin çözümünde kullanılan karar değişkenlerinin karşılığının ikili sayı sisteminde uzun olduğu durumlarda, tam sayı ve reel sayıların kullanıldığı kodlamalara başvurulabilir [24].

### **2.3.3. Popülasyon**

Popülasyon, çözüm bilgilerini içeren kromozomların bir araya gelmesiyle oluşan olası çözüm kümesine denir. Çözüm kümesindeki kromozom sayısı sabit olup problemin özelliğine göre programlayıcı tarafından belirlenir. Genetik algoritmanın işleyişi esnasında bu çözüm kümesinden bir takım kromozomlar yok olmakta ve yerlerine yeni kromozom yapıları eklenerek popülasyon büyüklüğü sabitlenmektedir [21].

Popülasyon büyüklüğünün ne olacağını belirlemek, tüm genetik algoritma kullanıcıları açısından oldukça önemli bir aşamadır. Popülasyon büyüklüğü, başka bir deyişle kromozom ya da çözüm sayısı, genetik algoritmanın başarısını ve optimum sonuca erişim süresini etkileyen önemli unsurlardan biridir [32].

### **2.3.4. Uygunluk Fonksiyonu**

Mevcut popülasyonda bulunan kromozomlardan hangilerinin bir sonraki nesle aktarılacağı ve hangilerinin yok olacağına karar verilmesini sağlayan bir değerlendirme kriteridir.

Problem için en uygun çözümü belirleme kriteri olan uygunluk fonksiyonunun, üzerinde durulan konuyla ilgili kar veya verimliliği maksimum yapacak, maliyet veya kaybı minimum yapacak değişkenlerin ölçülmesini sağlayacak bir fonksiyon olması gerekmektedir [27].

Problemin yapısına göre başlangıçta belirlenen uygunluk fonksiyonu ile popülasyondaki tüm kromozomların uygunluk değerleri hesaplanır. Böylece K elemanlı popülasyonda her bir kromozom için;  $f_1, \dots, f_K$  olmak üzere K adet uygunluk değeri hesaplanmaktadır.

Uygunluk değeri en iyi olan kromozomların seçilerek, yeni oluşturulacak popülasyona aktarılması, genetik algoritmanın gücünü oluşturan başlıca faktörlerden biridir [33].

## **2.4. Genetik Algoritmada Kullanılan Seçim Yöntemleri**

Genetik algoritmalarda, uygun olanların seçilmesi işlemi “en iyi olanın hayatta kalması” prensibi temel alınarak yapılmaktadır. İlk etapta oluşturulan başlangıç

popülasyonundaki kromozomlar arasından, en iyi uygunluk değerine sahip olanlardan bir kısmı oluşturulacak yeni popülasyona aktarılırken, bir kısmı da ebeveyn olarak seçilmektedir. Seçilen bu ebeveynlere genetik işlemler uygulanarak yeni popülasyonun diğer kromozomları elde edilmektedir.

Genetik algoritmalarda seçim yöntemi olarak geliştirilmiş birçok yöntem bulunmaktadır. Ancak, elitizm, rulet çarkı ve turnuva seçim yöntemleri en yaygın kullanılanlardandır [32,34].

#### **2.4.1. Elitizm Yöntemi**

Elitizm, en iyinin saklanması yöntemidir. Bu yöntemde, popülasyon içindeki en iyi kromozomlar ya da belli bir genişlikteki yüzdeliğe sahip kromozomlar, o popülasyondan alınarak hiçbir değişikliğe uğratılmadan yeni popülasyona aktarılır. Genetik işlemlerin(üreme, çaprazlama, mutasyon) kullanımı sonrası en iyi bireyin yok olması söz konusu olduğu için popülasyon içindeki, çözümü en iyi temsil eden kromozom bir sonraki popülasyona kopyalanır [21].

#### **2.4.2. Rulet Çarkı Yöntemi**

Holland'ın orijinal genetik algoritma yaklaşımında kullanmış olduğu bu yöntemde, her kromozom kendi uygunluk değeri ( $f_i$ ) ile orantılı bir olasılık değeriyle seçilmektedir. Her kromozomun uygunluk değeri, popülasyonun uygunluk değerine bölünerek, o kromozomun seçilme olasılığı elde edilir. Dolayısıyla,  $i$ . kromozomun seçilme olasılığı,

$$f(i) / \sum_{i=1}^K f(i) \text{ 'dir.}$$

Burada  $K$  popülasyon büyüklüğüdür. Rulet çarkının yüzeyi kromozomların uygunluk değerleriyle orantılı olarak işaretlenir. Rulet çarkı  $K$  defa döndürülür ve her seferinde bir kromozom ebeveyn olarak seçilerek, eşleme havuzuna atılır. Daha iyi uygunluk değerine sahip kromozomlar, diğerlerine oranla çarkın daha büyük bölümünü kaplarlar ve bundan dolayı seçilme şansları daha yüksektir [19,35].

#### **2.4.3. Turnuva Yöntemi**

Kolay uygulanabilirliği açısından oldukça tercih edilen bir seçim yöntemidir. En basit biçimiyle, ikili turnuva seçiminde, popülasyondan rastgele olarak iki kromozom seçilir ve turnuvaya katılır. Bu iki kromozom uyum değerlerine bakılarak kıyaslanır ve daha iyi uyum değerine sahip olan kromozom turnuvayı kazanır. Her

kromozom iki defa turnuvaya girer. Popülasyondaki en iyi kromozom her iki turnuvayı da kazanır. En kötü uygunluk değerine sahip olan kromozom ise iki turnuvayı da kaybeder ve böylece seçilme şansı ortadan kalkmış olur. Ortalama uyum değerine sahip olan kromozomlar ise en iyi kromozomların yarısı kadar bir sıklıkta seçilmiş olur. Bu işlemler sistematik olarak yapılarak, popülasyondaki en iyi kromozomların bir sonraki popülasyonda ikişer kopya ile temsil edilmesi sağlanmış olur. Popülasyonun sahip olduğu kromozom sayısına ulaşıncaya kadar işlemler devam ettirilir. Bu sayede, popülasyondaki kromozom sayısının sabit kaldığı gibi, popülasyonun ortalama uyum değeri de artar [19,36].

## **2.5. Genetik İşlemler**

Genetik algoritmada çözüm kümesi incelenirken belirli noktalardan sonra nesil çeşitliliği olmadığı için çözüme gidilememektedir. Problemin istenen kısıtlarını sağlayacak olan çözüm kümesine nesil çeşitliliği sağlanarak ulaşılabilir. Bu amaçla, kromozomlara bazı genetik işlemler uygulanarak, nesil çeşitliliği sağlanmış ve sistemin belirli noktalara gelip takılması önlenmiş olur. Kullanılan en temel genetik işlemler; üreme, çaprazlama ve mutasyondur.

Genetik işlemler, yeni popülasyon elde etmede, popülasyonla uyumlu bireylerin uygun ve iyi olan özelliklerini yeni popülasyona aktarmada, popülasyonun daha önceki popülasyondan ayrıklaştırılmasında ve de çözüm kümesinde daha farklı bölgelere ulaşmada kullanılmaktadır. Genetik işlemciler, belirtilen bu amaçlara ulaşmak için belirli işlemleri popülasyondan seçilen ebeveyn bireylere ve bu ebeveynlerden üretilen yeni bireylere uygular. Genetik işlemcilerin uygulanacağı kromozomlar, algoritmanın çalışması esnasında yeni popülasyon oluşturma basamağının başında seçilmektedir. Bu nedenden dolayı, GA uygulamalarında seçme işlemi çok önemli bir adımdır [24].

### **2.5.1. Üreme**

Üreme, uygunluk kriterlerine uyan bir kromozomun özelliklerinin, yeni nesle aktarılmasını sağlayan işlemdir. Uygun olarak seçilen kromozom çifti, yeni popülasyonun bir ya da daha fazla kromozomunun oluşumuna katkıda bulunabilmektedir. Ebeveyn olarak adlandırılan bu kromozom çiftinin özellikleri, üreme sonucunda yeni bireylere aktarılmaya çalışılmaktadır. Üreme işlemi sonrasında yeni kromozomlar üzerinde tekrarlı bir şekilde işlem yapılmaması ve

yeni toplumda yeni bireylerin ebeveynlerinin birer kopyası olmasını engellemek amacıyla çaprazlama ve gerekirse mutasyon işlemleri uygulanır [32].

### 2.5.2. Çaprazlama

Çaprazlama, iki kromozomun bir araya gelerek karşılıklı gen yapılarının değişimi ile yeni kromozomların oluşumunu sağlayan işlemidir.

Mevcut gen havuzunun potansiyelini araştırmak üzere, bir önceki nesilden daha iyi nitelikler içeren yeni kromozomlar yaratmak amacı ile çaprazlama işlemi kullanılmaktadır. Çaprazlama, belirlenen bir çaprazlama oranına eşit olasılıkla seçilen ebeveyn kromozomlarına uygulanmaktadır [19,37].

Genetik algoritmanın performansını etkileyen önemli parametrelerden biri olan çaprazlama işlemi, doğal popülasyonlardaki çaprazlamaya karşılık gelmektedir. Seçim işlemleri sonucunda elde edilen yeni popülasyondan rastgele olarak iki dizi seçilmekte ve karşılıklı çaprazlama işlemine tabi tutulmaktadır. Çaprazlama işleminde kromozom uzunluğu  $\ell$  olmak üzere, 1 ile  $(\ell-1)$  aralığından bir tamsayı rastgele olarak seçilmektedir. Bu tamsayı değeri çaprazlama noktasını belirtir [19,38].

Çaprazlama işleminde, en önemli unsurlardan biri ne tür çaprazlamanın yapılacağıdır. Literatürde yer alan çeşitli çaprazlama yöntemleri aşağıda açıklanmaktadır.

#### 2.5.2.1. Tek Noktalı Çaprazlama

Seçim sürecinden gelen kromozomlar rastgele olarak eşlenirler.  $\ell$  kromozom uzunluğu olmak üzere, seçilen her kromozom çifti için,  $[1, \dots, \ell-1]$  aralığında rastgele olarak bir tamsayı üretilir. Bu tamsayı çaprazlama noktasını göstermektedir. Eşlenen iki kromozomda bu çaprazlama noktasından sonraki bölümler yer değiştirerek iki tane yeni kromozom elde edilir [21].

1.Ebeveyn	10110 01001	1. Çocuk	10110 11010
2.Ebeveyn	11000 11010	2. Çocuk	11000 01001

**Şekil 2.1.** Tek noktalı çaprazlamaya bir örnek [21]

### 2.5.2.2. Çok Noktalı Çaprazlama

Bu işlemde çaprazlama noktası 1 ile  $l-1$  arasında rastgele çoklu bölge seçilir. Eşlenen iki kromozomda bu çaprazlama noktaları arasında kalan bölümler yer değiştirilerek iki tane yeni kromozom elde edilir [21].

1.Ebeveyn	10 110 01 001	1. Çocuk	10 000 01 010
2.Ebeveyn	11 000 11 010	2. Çocuk	11 110 11 001

**Şekil 2.2.** Çok noktalı çaprazlamaya bir örnek [21]

Tek noktalı ve çok noktalı çaprazlama işlemi genetik algorithmada ilk akla gelen çaprazlama yöntemleridir. Ancak problemin özelliğine göre farklı tiplerde çaprazlama yapmakta mümkündür. Bu çaprazlama yöntemlerinden birkaç tanesi kısıtlı eniyileme problemleri için incelenecek olunursa aşağıda belirtilen tiplerde çaprazlama yapmakta mümkündür. Atölye çizelgeleme gibi kısıtlı eniyileme problemlerinde, gen kodlamanın farklı olmasından dolayı probleme uygun çeşitli çaprazlama türleri üzerinde çalışmalar yapılmıştır. Bunlardan bazıları şu şekildedir [21]:

- Pozisyona dayalı çaprazlama
- Sıraya dayalı çaprazlama
- Kısmi planlı çaprazlama

### 2.5.2.3. Pozisyona Dayalı Çaprazlama

Bu çaprazlamada kalıp olarak, sabit kalacak olan gen yapılarını belirlemede kullanılan yapı bulunur. Kalıbın gösterdiği noktalar dizide sabit kalırken diğer noktalar iki birey arasında yer değiştirilerek yeni bireylerin oluşumu sağlanır. Şekil 2.3' teki kalıp dizisinde, 1'lerin gösterdiği değerler sabit kalacak değerleri göstermektedir [21].

1.Ebeveyn	3 4 7 1 1 0 4 8 9 2 3 3
2.Ebeveyn	0 0 1 4 7 2 8 9 2 1 0 0
Kalıp	1 1 1 0 0 0 1 1 0 0 1 0
1. Çocuk	3 4 7 4 7 2 4 8 2 1 3 0
2. Çocuk	0 0 1 1 1 0 8 9 9 2 0 3

**Şekil 2.3.** Pozisyona dayalı çaprazlamaya bir örnek

#### 2.5.2.4. Sıraya Dayalı Çaprazlama

Aşağıdaki şekildeki örnekten de görüldüğü gibi; kalıp üzerindeki 1'lerin gösterdiği değerler çaprazlamada kullanılacak olan değerleri belirtir. İkinci ebeveynden sırasıyla 7, 2, 3 değerleri çaprazlanacak olan genlerdir. Birinci ebeveynde bulunan 2, 3, 7 değerleriyle aynı sıralı olacak şekilde değiştirilir. Aynı işlem 1'lerin birinci ebeveynde gösterdiği değerlerin ikinci ebeveyne aktarılmasıyla tamamlanır [21].

1.Ebeveyn	1 2 3 4 5 6 7 8 9 0 4 5
2.Ebeveyn	7 4 6 1 2 8 3 5 3 1 9 6
Kalıp	1 0 0 0 1 0 1 0 0 0 0 0
1. Çocuk	1 7 2 4 5 6 3 8 9 0 4 5
2. Çocuk	1 4 6 5 2 8 3 7 3 1 9 6

**Şekil 2.4.** Sıraya dayalı çaprazlamaya bir örnek [21]

#### 2.5.2.5. Kısmi Planlı Çaprazlama

İki bireyden rastgele bir aralık belirlenir. Bu aralıktaki değerler yer değiştirilir. Şekil 2.5 ve 2.6' da (iki adımda) bu çaprazlamaya bir örnek gösterilmiştir [21].

1.Ebeveyn	28 645 713
2.Ebeveyn	87 213 456
	↓ ↓
1. Çocuk	28 213 713
2. Çocuk	87 645 456

**Şekil 2.5.** Kısmi planlı çaprazlamaya bir örnek (1. Adım)

Yer değiştirme sonunda dizide aynı olan değerler değiştirilen değerlerle tamamlanır (2. adım).

1. Çocuk	68 213 745
2. Çocuk	87 645 123

**Şekil 2.6.** Kısmi planlı çaprazlamaya bir örnek (2. Adım)

Daha önce de bahsedildiği gibi problemin özelliğine göre farklı yapılardaki çaprazlama yöntemleri kullanılabilir. Ancak temel olarak tek ve çok noktalı çaprazlama yöntemleri kullanılmaktadır [21].

### 2.5.3. Mutasyon

Çaprazlama işlemi, popülasyondaki mevcut gen potansiyellerini araştırır. Eğer popülasyon problemi çözmek için ihtiyaç duyulan tüm şifrelenmiş bilgiyi içermiyorsa, mevcut gen havuzundan yeni kromozomlar üretme yeteneğine sahip bir işleme gereksinim duyulacaktır [19,39].

Genetik algoritmada kullanılan mutasyon, başkalaştırma anlamına gelen doğal genetik mutasyon kavramından esinlenerek ortaya çıkmıştır. Kromozomların başkalaştırılması ya da farklılaştırılması için kullanılan bir işlemdir [40].

Mutasyon ile mevcut popülasyona yeni genetik bilgi eklenmiş olur. Bu durum, popülasyonun çeşitliliğini artırarak, çaprazlamanın aksine, algoritmanın çalışma sürecini rastgele bir şekilde çözüm uzayının yeni alanlarına yönlendirir. Mutasyon işlemi, genetik algoritmalara daha önceden farkedilmemiş veya yeni çözümlere ulaşabilme ve yerel alt optimumlara takılmama özelliklerini kazandırmaktadır [41].

Mutasyonun amacı, genetik çeşitliliği sağlamak ve korumaktır. Çaprazlama işleminden sonra yapılan mutasyon, bireylerde rastgele olarak küçük değişimler gerçekleştirerek yeni kromozomların oluşmasını sağlar. Bu değişim belirlenen bir mutasyon oranı ile gerçekleştirilir. Kromozomun sahip olduğu mutasyon oranına göre, hangi kromozomun hangi gen veya genlerinin değiştirileceğine karar verilir.

Şekil 2.7' de mutasyon işlemi için genel bir örnek verilmektedir [42]:

Mutasyon Noktası	↓
Kromozom	1 0 1 0 0 1 0 0 1 0
Mutasyona	1 0 1 0 1 1 0 0 1 0
Uğramış Kromozom	

**Şekil 2.7.** Mutasyon işlemi için genel bir örnek

Literatürde, problemin yapısına bağlı olarak uygulanan farklı mutasyon çeşitleri mevcuttur. Yaygın olarak kullanılan mutasyon çeşitlerinden bazıları şunlardır [32]:

- ✓ Ters çevirme
- ✓ Yer değişikliği
- ✓ Ekleme
- ✓ Karşılıklı değişim



Yukarıda belirtilen mutasyon türleri dışında, tümleyen mutasyonu olarak bilenen ve kromozomlardaki genlerin, onu tümleyen genlerle değiştirilmesi esasına dayanan bir mutasyon çeşidi daha vardır. Bu mutasyon türü, özellikle ikili sayı sisteminde kodlanan kromozomlar için daha çok kullanılmaktadır. Bu sistemde, rastgele seçilen kromozom dizisinde 0'ların yerine tümleyeni olan 1'lerin; 1'lerin yerine de tümleyeni olan 0'ların geçmesidir [24].

Ters çevirme mutasyonunda, çözüm kümesinden bir kromozom seçilir ve kromozom boyunca rastgele iki nokta seçilir, seçilen iki nokta arasında kalan genlerin sıraları tamamen ters çevrilerek bulunduğu yere yeniden yerleştirilir.

Yer değişikliği mutasyonunda, çözüm kümesinden bir kromozom seçilir ve kromozom boyunca rastgele iki nokta seçilir, seçilen iki nokta arasında kalan genler kromozom üzerinde rastgele bir yere yerleştirilir.

Karşılıklı değişim mutasyonunda, çözüm kümesinden seçilen bir kromozomdan rastgele alınan iki genin yerleri değiştirilir.

Eklemede ise çözüm kümesinden seçilen bir kromozomdan rastgele bir parça seçilir, yine rastgele seçilen bir konuma yerleştirilir [32].

Şekil 2.8' de, ikili sayı sistemi kullanılarak kodlanmış bir kromozom üzerinde, belirtilen dört mutasyon türünün uygulanışını gösteren bir örnek verilmektedir [32].

Mutasyon öncesi seçilen kromozom: [0 1 1 1 0 0 0 1 0 1]			
Mutasyon sonrası oluşan kromozom:			
Ters çevirme	Yer değişikliği	Ekleme	Karşılıklı değişim
[0 1 <b>0 0 1 1</b> 0 1 0 1]	[0 1 0 1 0 1 <b>1 1 0 0</b> ]	[0 1 <b>0</b> 1 0 0 0 1 0 1]	[0 <b>0</b> 1 1 0 <b>1</b> 0 1 0 1]

**Şekil 2.8.** Farklı mutasyon türleri için örnekler

Herhangi bir konumda yapılan bir genetik değişim asla geri alınamaz ya da değiştirilemez [32]. Mutasyonun bu özelliğinden dolayı, mutasyon yapılırken iyi genlerin zarar görmemesine dikkat edilmelidir.

## 2.6. Genetik Algoritmanın Parametreleri

Genetik işlemler, genetik parametreler olarak adlandırılan ve genetik algoritmanın performansı üzerinde oldukça büyük etkiye sahip olan parametrelerden fazlasıyla

etkilenmektedir. Genetik işlemler için bir sınır oluşturma özelliğine sahip olan genetik parametrelerin kontrolü ve uygun parametre değerlerinin belirlenmesi, ele alınan problemin uygun çözümünün elde edilmesinde önemli rol oynar. Bir genetik algoritmanın temel parametreleri; popülasyon büyüklüğü, çaprazlama oranı, mutasyon oranı ve durdurma kriteri olarak sayılabilir.

Problem yapısına uygun parametreleri bulmak için birçok çalışma yapılmıştır fakat tüm problemler için genel olarak kullanılabilir parametreler bulunamamıştır [27,43]. Bir problem için uygun olarak belirlenen genetik parametre değerleri başka bir problem için etkin sonuç vermeyebilir [32].

### **2.6.1. Popülasyon Büyüklüğü**

Popülasyon büyüklüğünün belirlenmesi, genetik algoritmada karar verici tarafından alınması gereken en önemli kararlardan biridir. Popülasyon büyüklüğünün çok büyük olması, farklı potansiyel çözümlerin artışına neden olurken, algoritmanın yerel optimum noktalara takılma olasılığını da azaltır. Ancak, bu olumlu özelliklere karşın arama uzayında optimal bölgeye yakınsamak için gereksinim duyulan zaman artar [19]. Bu durum özellikle gerçek zamanlı (real-time) problem uygulamalarında istenmeyen bir durumdur. Popülasyon büyüklüğünün çok küçük olması ise, arama uzayının yetersiz örneklenmesine sebep olacağı için genetik algoritmanın yerel bir optimuma takılma olasılığını arttırmaktadır. Bu yüzden popülasyon büyüklüğü için uygun bir değer belirlenmelidir.

Goldberg 1985 yılında, yalnızca kromozom uzunluğuna bağlı olan bir popülasyon büyüklüğü hesaplama yöntemi önermiştir.  $K$ , popülasyon büyüklüğü ve  $\ell$  kromozom uzunluğu iken [19];

$$K = 1,65 * 2^{0,21 * \ell}$$

Genel olarak yapılan çalışmalar sonucunda, kısıt içeren problemlerde popülasyon büyüklüğünün kromozom uzunluğuna yakın bir sayı olması gerektiği ortaya konmuştur. Örneğin kromozom uzunluğu 30 ise popülasyon büyüklüğünün 30-50 arası olması yeterlidir veya kromozom uzunluğu 200 olan bir problem için popülasyon büyüklüğünün 100 ile 200 arası olması yeterlidir [19].

### **2.6.2. Çaprazlama Oranı**

Çaprazlama işlemi ile mevcut olan uygunluk değeri yüksek kromozomların özelliklerini birleştirerek daha iyi kromozomlar oluşturmak amaçlanmıştır.

Çaprazlama yapılırken tüm popülasyona bu işlem uygulanmaz, sadece belirli bir kısmına yapılır. Çaprazlama oranı, çaprazlama işlemi uygulanacak kromozom sayısını, bir başka deyişle çaprazlama işlemi sıklığını belirlemek için kullanılan parametredir.

Çaprazlama oranının yüksek olması, popülasyonda değişime uğrayan kromozom sayısının fazla olacağını yani, yeni kromozom sayısının artacağını göstermektedir [44]. Gereğinden yüksek olarak belirlendiği durumda, mevcut popülasyonda bulunan iyi kromozomların bir sonraki popülasyona taşınamama riskini doğurması beklenmektedir [32].

Yüksek çaprazlama oranı, çözüm uzayını hızlı bir şekilde aramayı sağlarken diğer yandan da iyi sonuçlar verecek kromozomların atlanmasına ya da genetik algoritmanın performansının düşmesine neden olabilecektir [45]. Çaprazlama oranın düşük olarak belirlenmesi halinde ise, değişime uğrayacak kromozom sayısı azalacak ve dolayısıyla algoritma yavaşlayarak sonuca geç ulaşılabilecektir [32].

Popülasyonda K kromozom varsa,  $P_c$  çaprazlama oranı olmak üzere,  $P_c * K$  kadar kromozom çaprazlama için seçilmelidir [30]. Bu kromozomların seçimleri, tüm kromozomlar arasından rastgele bir şekilde yapılabildiği gibi, uyumluların seçilmesi yöntemleri (elitizm, rulet çarkı gibi) kullanılarak da yapılabilmektedir [32]. Çaprazlama olasılığının literatürde genellikle 0.25 ile 1 arasında seçildiği görülmektedir.

### **2.6.3. Mutasyon Oranı**

Çaprazlama işlemi takip eden mutasyon işlemi, bir kromozomda rastgele değişimi sağlayarak yeni kromozom oluşmasını sağlar. Tıpkı çaprazlamada olduğu gibi, mutasyon işlemi uygulanacak kromozom sayısı da belli bir mutasyon oranı ile belirlenir. Belirlenen mutasyon oranları ile kromozom üzerindeki bazı genlerin yerleri değiştirilerek ya da kromozomun genleri değiştirilerek mutasyon uygulanır.

Mutasyon işleminin temel amacı popülasyondaki genetik çeşitliliği sürdürmektir. Mutasyon belirli bir olasılıkla, bir kromozomdaki her bitte meydana gelebilir. Eğer mutasyon oranı artarsa, genetik arama rastgele bir aramaya dönüşür. Arama uzayındaki aşırı rastgelelik çözümden uzaklaşmaya, yani popülasyonun gelişmesine değil zarar görmesine neden olur. Ancak, yüksek bir mutasyon oranı

aynı zamanda kayıp genetik bilginin tekrar ortaya çıkarılmasına da yardımcı olur [46]. Mutasyon oranının çok düşük olması ise araştırma uzayının tamamen araştırılmasını engelleyerek, algoritmanın yerel optimuma takılmasına neden olur.

Pratik çalışmalarda, algoritmanın tamamen rastgele olmasını engellemek amacıyla mutasyon oranının 0,01 ile 0,001 arasında alındığı görülmektedir [30].

Muhlenbein ve Bäck'in yaptığı çalışmalar, mutasyonun olasılığının optimal oranının dizinin uzunluğu ve problemin çözüm uzayı ile orantılı olduğunu ortaya koymuştur. Çaprazlama olasılığı, genellikle 0.25 ile 1 arasında, mutasyon olasılığı 0.01 ile 0.001 arasında seçilmektedir. Özellikle küçük popülasyonlarda sistemin performansını çaprazlama olasılığından çok mutasyon olasılığı belirler [47].

Mutasyon ve çaprazlama ile ilgili parametrelerin kullanım amacı her yeni popülasyonda daha iyi kromozomlar oluşturmaktır. Ancak seçim, çaprazlama veya mutasyon sırasında kötü bireyler de oluşabilmektedir ve uygunluk değeri yüksek olan bir kromozomun bozulmasına neden olabilmektedir. Bu nedenle özellikle mutasyon oranı düşük tutulmaktadır. Böylelikle bir kromozom içinde birden fazla genin değişmesi ile kromozomun tamamen farklılaşması engellenmiş olacaktır [26].

#### **2.6.4. Durdurma Kriteri**

Genetik işlemler olarak adlandırılan üreme, çaprazlama ve mutasyon işlemlerinden sonra yeni bir nesil oluşmaktadır. Yeni neslin uygunluk değeri hesaplanır ve tüm bu işlemler bir döngü içinde yapılır. Gelecek döngüler için de bu işlemler yinelenenektir. Eğer sonlandırma koşulu oluşturacak bir durdurma kriteri belirlenmezse bahsi geçen evrimsel süreç sonsuza dek devam eder.

Bir genetik algoritmanın kaç kez yinlendiği, yani nesil sayısının ne olacağı konusunda kesin bir yaklaşım mevcut değildir [48].

Literatürde genetik algoritmaların durdurma kriterleriyle ilgili çeşitli yöntemler yer almaktadır. Literatürde yer alan bu durdurma kriterlerinden en yaygın olanları aşağıda başlıklar altında açıklanmaktadır [19].

##### **✓ Hesaplama Zamanı Kriteri**

Bu yöntemde göre, genetik algoritma önceden belirlenen bir hesaplama zamanı ya da döngü sayısına göre çalıştırılmakta, bu zaman dolduğunda veya belirlenen döngü sayısına ulaşıldığında da durdurulmaktadır. Böyle bir yöntemin bazı

sakıncaları mevcuttur. Çünkü belirlenen döngü sayısı erken bir duruşa neden olabilmektedir. Döngü devam ettirildiğinde iyileşmenin görülebileceği nesiller olabilir. Ayrıca tersi bir durum da söz konusu olabilir. Belirlenen döngü sayısı gereğinden fazla olabilir ve bu da hesaplama zamanını arttıracaktır [49].

✓ Optimizasyon Hedefi Kriteri

Hesaplama zamanı kriterinde olduğu gibi bu yöntemde de önceden belirlenen bir değere göre algoritma çalıştırılır. Burada önceden belirlenen değer, ulaşılması istenen amaç fonksiyonu değeridir. Genetik algoritma sürecinde uygunluk değeri belirlenen bu değere ulaştığında algoritma sonlandırılmaktadır [37].

✓ Minimum İyileşme Kriteri

Bu yöntem, nesiller boyunca iyileşme miktarları arasındaki farkların alınarak iyileşmenin giderek azaldığı bir döngüde algoritmanın durdurulmasını içermektedir. Uygunluk fonksiyonunun çözüm sırasındaki seyri izlenerek algoritmanın ne zaman sonlandırılacağı tespit edilmektedir. Genetik algoritma problemlerinde genellikle eldeki en iyi çözüm önce hızlı sonra da yavaş yavaş artış göstermektedir. Değerdeki iyileşme hızının giderek azalması ve sifıra yaklaşması, artık daha fazla iyileşme beklenmemesi gerektiğini gösterebilir. Algoritma, çözüme harcanacak zaman ve çözümden beklenen kalite arasında bir denge kurularak durdurulur [50].

## 2.7. Genetik Algoritmanın Aşamaları ve Akış Şeması

Genetik algoritmalarda ilk olarak kromozomları yani çözümleri temsil eden kodlama işlemi yapılır. Burada, genler çözümlerin özelliklerini, genlerden oluşan kromozom da bir çözümü ifade etmektedir.

Tüm problemler için uygun tek bir gösterim (kodlama) şekli yoktur. Çoğu zaman çözümler, Holland tarafından ilk olarak kullanıldığı gibi, çözümün her bir elemanının 0 veya 1 değeri alabildiği ikili değerlerin kullanıldığı sabit uzunluklu diziler olarak kodlanır. Ancak, kullanılan gösterim için uygun genetik işlemler tanımlandığı sürece çözümlerin gösterimi için bir çözümün sınırlı bir dizi olarak kodlanmasına imkan veren herhangi bir gösterim de kullanılabilir. Örneğin, uygulamaya bağlı olarak, bazı problemler için çözüm parametrelerinin daha uygun ifade edilebilmesi nedeniyle tamsayı veya gerçel sayıların kullanılması da mümkündür. Genel olarak ise, gösterim ve çözümlerin kodlanması için diziler, ağaç yapıları, listeler veya herhangi bir nesne kullanılabilir [41].

Genetik algoritmalar, kodlama işlemi yapıldıktan sonra kodlanmış kromozomlardan meydana gelen bir başlangıç popülasyonu oluşturarak başlarlar. Daha önce de bahsedildiği gibi belirlenen popülasyon büyüklüğü problemin yapısına göre belirlenmelidir. Daha sonra mevcut popülasyona temel genetik işlemler (üreme, çaprazlama, mutasyon) uygulanarak oluşturulan her nesil için uygunluk değeri hesaplanır. Bu durum belirlenen durdurma kriteri sağlanıncaya kadar devam eder.

### **2.7.1. Genetik Algoritmanın Aşamaları**

Standart bir genetik algoritma, genetik algoritmalara özel süreçleri içeren bazı adımlardan oluşur. Tipik bir genetik algoritmanın çalışma adımları temel olarak aşağıda verilmiştir [41].

1. Adım: Gösterim (kodlama) yönteminin belirlenmesi.
2. Adım: Başlangıç popülasyonunun (ilk nesil) oluşturulması.
3. Adım: Başlangıç popülasyonundaki her bireyin (kromozomun) performansının amaç fonksiyonuna göre hesaplanması (uygunluk değeri).
4. Adım: Yeni neslin oluşturulmasında kullanılacak bireylerin seçilmesi.
5. Adım: Seçilmiş bireylere genetik işlemlerin uygulanarak yeni neslin elde edilmesi.
6. Adım: Yeni neslin bireylerinin performanslarının uygunluk fonksiyonuna göre hesaplanması.
7. Adım: Durdurma kriteri sağlanmamışsa 4. adıma dönülmesi
8. Adım: Durdurma kriteri sağlanmışsa en iyi bireyin sonuç olarak belirlenmesi ve sürecin durdurulması.

Basit bir genetik algoritmanın temel adımları şu şekilde de ifade edilebilir [8]:

1. Adım: Çözümlerin bir başlangıç popülasyonunu oluştur (Genellikle rastgele oluşturulur).
2. Adım: Popülasyondaki her çözümün uygunluk değerini hesapla.
3. Adım: Durdurma kriteri sağlanıyorsa araştırmayı durdur.

Yoksa, aşağıdaki adımları gerçekleştir:

- 3.1. Seçim yöntemi uygula (Uygunluk değeri daha yüksek olan çözümler yeni popülasyonda daha fazla temsil edilirler).
- 3.2. Çaprazlama işlemi uygula (Mevcut iki çözümden yeni iki çözüm üretilir).

3.3. Mutasyon işlemini uygula (Çözümlerde rastgele deęişim meydana getirilir).

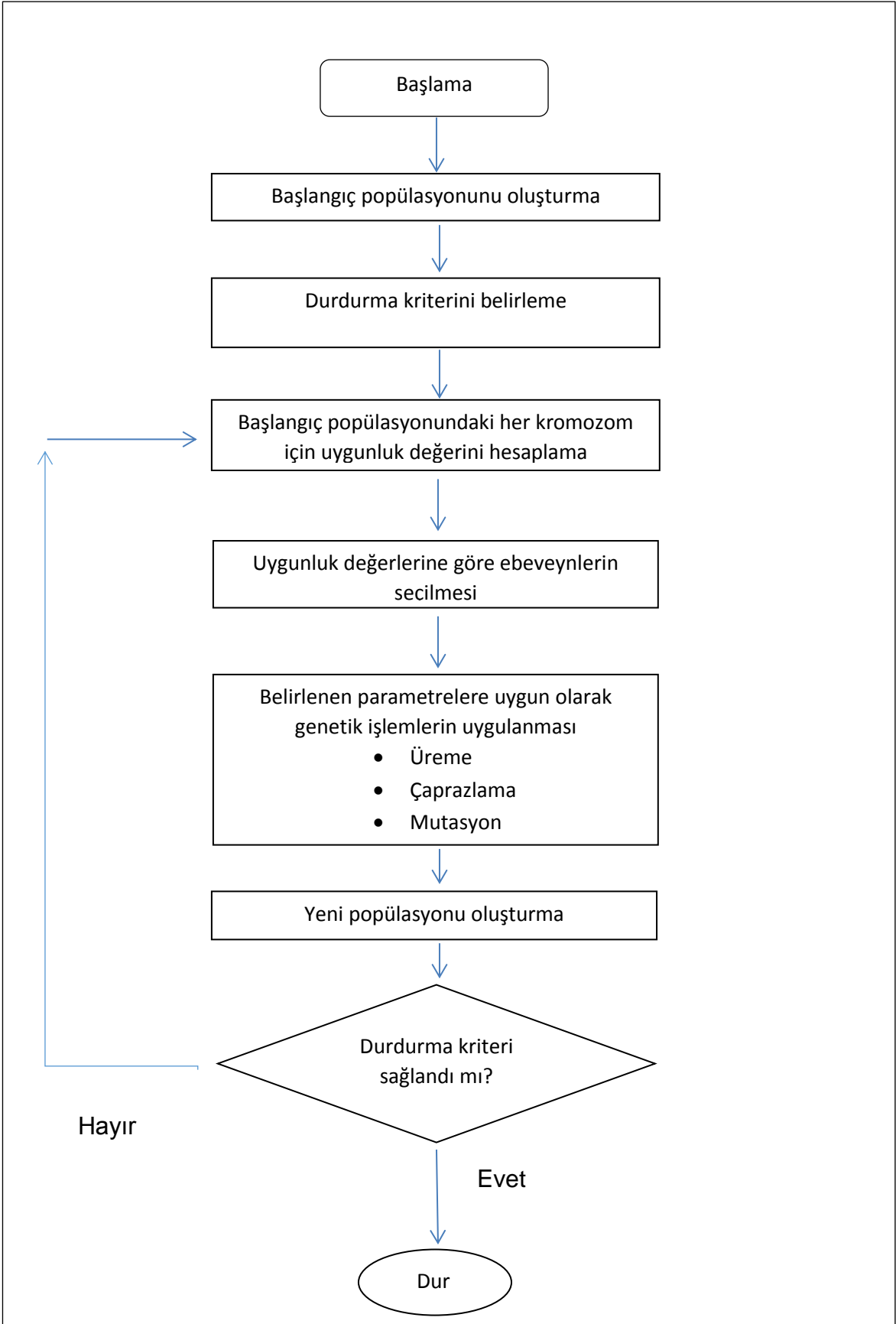
4. Adım: Mevcut popülasyonu oluşturulan yeni popülasyon ile deęiştir.

5. Adım: 2. Adım'a git.

Yukarıda verilen algoritmadaki adımlarda, 2-5. adımlar arasında gerçekleştirilen her bir döngü, bir nesil ya da yeni popülasyonu oluşturan bir üreme sürecidir. Her bir üreme sürecinde bir ya da daha fazla uygun kromozom oluşacak ve bu sayede her yeni popülasyon bir öncekinden daha iyi olacaktır.

### **2.7.2. Genetik Algoritmanın Akış Şeması**

Genetik algoritmanın temel işleyişini gösteren akış diyagramı Şekil 2.9' da verilmiştir:



**Şekil 2.9.** Genetik algoritmanın genel işleyişi



## **2.8. Genetik Algoritmanın Özellikleri**

Günümüzde sezgisel yöntemlerin kullanımı oldukça artmıştır. Genetik algoritma, güçlü bir çözüm aracı olması ve kolayca bilgisayar programına dönüşebilmesi açısından gerçek problemlerin çözümünde oldukça sık başvurulan bir sezgisel yöntemdir. Genetik algoritma literatüründe çok sayıda başarılı uygulamalar olduğu gibi, bazı uygulamalarda da yetersiz ya da zayıf sonuçlar verdiği gözlenmiştir. İzleyen bu başlık altında genetik algoritmanın avantajları ve dezavantajları ele alınmıştır.

### **2.8.1. Genetik Algoritmanın Avantajları**

Genetik algoritmalar, klasik yöntemlerle çözümü elde edilemeyen karmaşık problemlerin çözümünde etkin olarak kullanılabilen hızlı, kolay uygulanabilen ve güçlü bir yöntemdir. Günümüze kadar yapılmış olan uygulamalarda, genetik algoritmanın hangi koşullar altında ve hangi uygulamalarda daha başarılı sonuçlar verdiği tek bir ölçüte bağlanamamış, bunu etkileyen birden fazla ölçüt olduğu görülmüştür. Bunların başlıcaları; çözümlerin kodlanma şekli, genetik işlemler ve bunların bağlı olduğu genetik parametrelerdir.

Klasik optimizasyon yöntemleri başarısızlığa uğradığında, genetik algoritmanın getirdiği yararlar merak uyandırmış ve şaşırtıcı sonuçlar vermiştir. Genetik algoritma kullanılmasının başlıca yararları aşağıdaki gibi ifade edilebilir [32];

1. Genetik algoritmanın sürekli ya da kesikli değişkenler içeren optimizasyon problemlerinde optimum sonuçlar vermesi.
2. Türev alma işlemlerine gereksinim duyulmaması.
3. Çok sayıdaki değişkenleri kapsayan modellerde uygulanabilmesi.
4. Çözümleri, geniş çözüm uzaylarında eş zamanlı bir şekilde taraması.
5. Oldukça karmaşık maliyet/kar fonksiyonlarının değişkenlerini optimize edebilmesi.
6. Kodlanan değişkenlerde optimizasyonun yapılması.
7. Üretilen, deneysel verilerle ya da analitik fonksiyonlar ile çalışması.

Genetik algoritmaların yukarıdaki avantajlarını desteklemek ve arttırmak mümkündür [41]:

- ✓ Genetik algoritmaların optimal ya da optimele yakın sonuçlara ulaşabilme özelliği ve gücü seçim, üreme, çaprazlama ve mutasyon işlemleri ile doğrudan ilişkilidir. Genetik algoritmaların diğer optimizasyon yöntemlerinin aksine yerel minimumlarda takılıp kalma olasılığı çok daha azdır. Bunun sebebi ise bu genetik işlemlerin rastgele olarak gerçekleştiriliyor olmasıdır.
- ✓ Seçim işleminde performansı daha iyi olan bireylerin seçilme şansı yüksek olsa da bütün bireylerin seçilme şansı bulunmaktadır. Bu da çözüm uzayının araştırılması sürecinin daha geniş bir alanda sürdürülmesiyle ilk bakışta çözüm için uygun görülmeyen çözüm alanlarının daha detaylı araştırılması sonucunda çok daha iyi sonuçlara ulaşılabilmesine olanak sağlamaktadır.
- ✓ Çaprazlama işlemi ile mevcut çözüm kümesinde olmayan yeni kromozomların elde edilmesi mümkün olmaktadır. Bu şekilde mevcut genetik bilginin kombinasyonlarıyla daha iyi çözümlere ulaşmaya çalışılmakta, çözüm uzayını araştırma süreci dahi iyi sonuçlar elde edilmesi olası olan alanlara doğru yönlendirilmektedir.
- ✓ Mutasyon işlemi ise arama sürecini rasgele olarak farklı yönlerde geliştirerek çözüm uzayının farklı bölgelerinin keşfedilmesine olanak sağlamaktadır. Ancak, çözüm uzayındaki arama sürecinin genel gidişini çok fazla bozmamak amacıyla mutasyon oranı genelde düşük tutulur. Böylece çözüm uzayında sürdürülen genel arama süreci devam ederken, mutasyon ile çözüm uzayının hiç araştırılmamış bölümleri de arama sürecine katılmış olur.
- ✓ Genetik algoritmaların çözüme oldukça hızlı ulaşabilmesi çalışmalarının doğrudan (explicite) ve dolaylı (implicit) paralellik içermesi sayesinde. Doğrudan paralellik genetik işlemlerin aynı anda bütün bir popülasyona uygulanması nedeniyle çözüm uzayının paralel bir şekilde araştırılması demektir. Dolaylı paralellik ise bir çözümün performansının değerlendirilmesinde amaç fonksiyonu kullanılırken çözümün tüm özelliklerinin (genlerinin) aynı anda değerlendiriliyor olması demektir. Bu paralellikler çözüm uzayının farklı bölgelerinin aynı anda araştırılmasını sağlayarak büyük bir verimlilik sağlamış olurlar. Bu verimliliği sayesinde genetik algoritmalar optimal sonuca diğer tekniklere göre çok daha hızlı yakınsarlar.
- ✓ Ancak burada hız ve ulaşılan sonucun doğruluğu (accuracy) arasında bir tercihin söz konusu olduğunu belirtmek gereklidir. Genetik işlemler anlatılırken değinildiği gibi çaprazlama için farklı yöntemler kullanılabilir.

- ✓ Ayrıca çaprazlama ve mutasyon önceden belirlenmiş oranlara göre rasgele olarak gerçekleştirilmektedir. Kullanılan çaprazlama yöntemi ve çaprazlama ile mutasyon oranlarının seçimi, hız ve doğruluk arasındaki tercihi etkilemektedir. Ayrıca, küçük değişikliklerle farklı problemlere kolaylıkla uygulanabilen esnek bir yöntem olması genetik algoritmaların cazibesini arttırmaktadır.
- ✓ Hazırlanmış olan bir genetik algoritma programının yeni bir probleme uyarlanabilmesi için çoğu zaman problemin uygun şekilde kodlanması ve uygunluk fonksiyonunun doğru bir şekilde formüle edilmesi yeterlidir. Ayrıca, çözülecek problemle ilgili özel bilgiler genetik algoritma tarafından kolaylıkla kullanılabilir ve genetik algoritmalar başka yöntemlerle birleştirilerek kullanılabilirler.

### **2.8.2. Genetik Algoritmanın Dezavantajları**

Genetik algoritmaların avantajlarının yanı sıra bazı dezavantajları da mevcuttur. Bunlardan başlıcası, diğer sezgisel yöntemlerde de olduğu gibi optimal çözümü garanti etmeyişiştir. Klasik yöntemlerle çözümü olmayan problemlerde optimal sonuca ulaşp ulaşılmadığını belirlemek zordur.

Çözümüne ulaşmadan önce yerel optimumlara takılmayı önlemek ve yakınsamayı en iyi düzeyde gerçekleştirebilmek için karar verici tarafından problemin yapısına uygun parametrelerin seçimi çok büyük önem taşımaktadır. Bu parametrelerin belirlenmesinde genel kurallar olmadığı için problemin iyi tanınması ve buna bağlı parametrelerin doğru belirlenmesi oldukça zordur. Bu yüzden bu seçimlerin de optimize edilmesi ve problem yapısına uygun olarak seçilmesi gerekmektedir. Parametrelerin farklı değerlerle tekrar denenmeleri ve buna ek olarak daha önce genetik algoritmalarla çözülmüş problemlerle karşılaştırılmaları bu dezavantajı aşmanın yolu olabilir.

Genetik algoritmanın en iyi çözümün mevcut çözümler arasından seçilmesi nedeniyle göreceli olması da bir dezavantajdır. Bu durum, ulaşılan çözümün en iyi çözüm olup olmadığının kontrol edilmesine imkân vermeyebilir. Bu nedenle genetik algoritmalar en iyi çözümün ne olacağını bilinememesi durumunda kullanılmaktadır [30,32].

### **2.9. Genetik Algoritmanın Uygulama Alanları**

Karmaşık problemleri hızlı ve optimale yakın olarak çözebilen genetik algoritmalar, çeşitli problem tiplerine uygulanabilmektedir. Büyük çözüm uzaylarının geleneksel

yöntemlerle taranması hesaplama zamanını arttırmaktadır. Ancak bu tip problemlere, genetik algoritmalar ile kısa sürede, kabul edilebilir çözümler bulunabilmektedir [51]. Genetik algoritmalar özellikle çözüm uzayının geniş, süreksiz ve karmaşık olduğu problem tiplerinde başarılı sonuçlar vermektedir [52].

Temel ilkelerinin ortaya atılmasından sonra, genetik algoritmalar hakkında birçok bilimsel araştırma yayınlanmış ve genetik algoritmaların aşağıda verilen alanlarda başarılı uygulamalarının olduğu görülmüştür [21].

#### ✓ Optimizasyon

Genetik algoritma araştırmalarının önemli bir bölümü fonksiyon optimizasyonu ile ilgilidir. Genetik algoritmalar, geleneksel optimizasyon tekniklerine göre zor, süreksiz ve gürültü (noisy) içeren fonksiyonları çözmeye daha etkindirler. Genetik algoritmaların uygulandığı diğer bir optimizasyon problemi ise, istenen amaçlara ulaşmak üzere, sınırlı kaynakların etkin tahsis edilmesi ile ilgili kombinatoriyal optimizasyon problemleridir. Gezgin satıcı problemi, araç rotalama problemi, Çinli postacı problemi, iş atölyesi çizelgeleme problemi, atama problemi, yerleşim tasarımı problemi ve sırt çantası problemi kombinatoriyal optimizasyon problemlerine örnektir.

#### ✓ Otomatik Programlama ve Bilgi Sistemleri

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

#### ✓ Mekanik Öğrenme

Sınıflama sistemi, genetik algoritmaların mekanik öğrenme alanında bir uygulamasıdır. Basit dizi kurallarını öğrenen bir mekanik öğrenme sistemi olan sınıflama sisteminin kural ve mesaj sistemi, özel bir üretim sistemi olarak adlandırılabilir. Bu üretim sistemi, “eğer-sonra” kural yapısını kullanır. Bir üretim kuralı, “eğer” yapısından sonra belirtilen durum için, “sonra” yapısından sonra gelen faaliyetin gerçekleştirilmesini içerir. Genetik algoritmalar, sınıflama sistemlerinde kural-bulma mekanizması olarak kullanılmaktadırlar. Genetik

algoritmalar ayrıca, sinir ağlarında ve proteinin yapısal analizinde de kullanılmaktadır.

#### ✓ Finans ve Pazarlama

Genetik algoritmalar, finansal modelleme uygulamaları için son derece uygundur. Özellikle hisse senedi fiyatlarındaki değişim kalıplarını tahmin etmede ve bulmada, kaynak tahsisi ve uluslararası sermaye tahsisi stratejilerini belirlemede genetik algoritmalar kullanılabilir. Pazarı ve tüketiciyi tanımada son derece önemli rol oynayan veri madenciliği, veriyi bilgiye bilgiyi de güvenli kararlara dönüştürür. Veri madenciliğinin verimlilik, karlılık, müşteri tatmini ve rekabet edebilme yeteneği gibi yaşamsal konularda işletme üzerinde çok önemli etkileri bulunmaktadır. Veri madenciliğinde kullanılan tekniklerden birisi de genetik algoritmadır. Genetik algoritma tabanlı yaklaşım kullanılarak veri yığınlarından modeller elde edilmektedir.

#### ✓ Ekonomik ve Sosyal Sistem Modelleri

Genetik algoritmalar yenilik sürecinin modellenmesi amacıyla kullanılmaktadır. Ayrıca genetik algoritmaların, fiyat verme stratejilerinin gelişim süreçlerini ve kazanç getiren pazarların ortaya çıkış süreçlerini modelleme alanlarında da kullanımları oldukça yaygındır. Genetik algoritmalar sosyal sistemlerin evrimsel yönlerini anlamak amacıyla kullanılmaktadır. Bunlara örnek olarak işbirliğinin evrimi, iletişimin evrimi ve karıncalardaki iz takibi davranışının evrimi verilmektedir.

Yukarıda anlatılan alanların dışında daha birçok alanda özellikle de işletmelerdeki uygulama alanlarında genetik algoritmalar başarı sonuçlar vermektedir. Bunlardan en yaygın olanları; üretim/işlemler, montaj hattı dengeleme problemi, çizelgeleme problemi, tesis yerleşim problemi, atama problemi, hücreli üretim problemi, sistem güvenilirliği problemi, taşıma problemi, gezgin satıcı problemi, araç rotalama problemi, minimum yayılan ağaç problemi olarak sayılabilir.

Genetik algoritmaların bu kadar geniş alanda uygulanabilir olması ve başarılı sonuçlar vermesi, araştırmacıların konuya ilgisini arttırmıştır. Özellikle optimizasyon problemlerinde, daha etkin sonuçlar elde etmek amacıyla genetik algoritma başka yöntemler ile bir arada kullanılmıştır.

## 2.10. Genetik Algoritmanın Kullanılma Nedenleri

Genetik algoritmaların etkili ve faydalı olarak kullanılabileceği problemler şu şekilde belirtilebilir [24]:

- ✓ Problem karmaşık, anlaşılması açısından zor ve çözüm için araştırılacak alan çok geniş olduğunda,
- ✓ Probleme ilgili mevcut bilgiler yetersiz veya sahip olunan bilgiler çözüm için gerekli araştırma alanını daraltmada başarısız olduğunda,
- ✓ Deterministik yöntemlerle çözüm zor olduğunda ve uzun hesaplamalar sonunda elde edildiğinde,
- ✓ Klasik yöntemlerden yararlanarak hazırlanan paket programlar yetersiz kaldığında,
- ✓ Problemin modelini kurmak için sahip olunması gereken bilgiler elde edilemediği durumda genetik algoritma kullanmak avantajlı olabilir.

### 3. TABAKALI RASTGELE ÖRNEKLEME

Bu bölümde, ilk olarak problemimizin temelini oluşturacak olan örnekleme konusuna kısaca değinilmiştir. Daha sonra sık kullanılan örnekleme yöntemlerinden biri olan tabakalı rastgele örnekleme yöntemi genel hatlarıyla verilmiş ve bu yöntemdeki iki ana problem olan tabaka sınırlarının belirlenmesi ve örnekleme dağıtımı konusu işlenmiştir.

#### 3.1. Örneklem Nedir?

İnsanların mantıkları yardımıyla bir karar vermek amacıyla kullandıkları yöntemlerden biri örnekleme yöntemidir. Günümüzde, fizik, kimya, biyoloji gibi fen dallarında, çeşitli mühendislik dallarında, tıp, ecza, diş gibi sağlık bilimleri ve sosyal bilimlerde yapılan pek çok araştırmada; kamuoyu yoklamalarında ve pazarlama araştırmalarında, örnekleme yönteminden yararlanır. Günlük yaşamda da örnekleme yöntemi kullanılır. Örneğin, bir ev hanımı pişirmekte olduğu yemeğin tadına bakarak yemek hakkında karar verirken; satın aldığı bir mal bozuk ya da kusurlu çıktığı için o satın aldığı yerden bir kez daha alışveriş yapmayan müşteri gerçekte örnekleme yönteminden yararlanmaktadır. Kalite kontrol problemlerinde de örneklemeden yararlanır. Bu tür problemler, daha çok fabrikalarda üretilen mallar satışa sunulurken ya da çeşitli kuruluşlar tarafından alım yapılırken ortaya çıkar. Burada, üretilen ya da alımı yapılacak olan malların tek tek ele alınması, çoğu zaman olanaksızdır. O nedenle, malları simgeleyebilen, bir diğer deyişle, kitlenin özelliklerini taşıyan bir alt grup incelenerek karar verilir [53].

Kitleyi simgeleyebilecek nitelikte bir miktar birimin oluşturduğu alt gruba “örneklem” (sample), kitleden örneklem seçme işine de “örneklem” (sampling) adı verilir. Örneklemden yararlanarak kitle hakkında tahminler yapılır. Örneklem seçmek için ya da örnekleme yapmak için kullanılan yöntemlere “örneklem yöntemleri” (sampling techniques) denir [53].

Örneklem üzerinden çalışmak, araştırmacıya zaman, iş gücü ve para açısından tasarruf sağlar. Kitle üzerinde çalışmanın bir güçlüğü de, araştırma için gerekli kontrollerin sağlanmasındaki engellerin artmasıdır. Örneklem üzerinde denetim kurmak daha kolaydır. Araştırmada amaç, çok veri toplamak değil, sağlam, geçerli güvenilir veriler toplamaktır. Bu nedenlerle araştırmacı, kitle yerine, örneklem üzerinde çalışmayı tercih eder [54].

Örnekleme planının iyi oluşturulması, istatistiksel çalışmalardan elde edilen sonuçların kullanılabilir ve gerçeği yansıtır nitelikte olmasında çok önemli rol oynar. Örnekleme planı ne kadar iyi oluşturulur, seçilen örneklerle kitle ne kadar iyi temsil edilirse, elde edilen istatistiksel sonuçlar o kadar güvenilir olur. Uygulamalarda kullanılacak çok sayıda örnekleme yöntemi mevcuttur. Örneğin, kitle homojen yapıda ise basit rastgele örnekleme, kitle heterojen yapıda ise tabakalı rastgele örnekleme, en sık kullanılan örnekleme yöntemleridir [55].

Örnekleme kuramı, sonlu  $N$  sayıda kitle birimi içeren kitleden,  $n$  büyüklüğünde rastgele örneklem seçme ve seçilen örneklemden tahminler yapma yöntemlerini inceler. Bir başka ifadeyle, örnekleme kuramı konusu, kitleden, kitlenin yapısına en uygun örnekleme yöntemiyle, örneklem seçme süreci ve örneklemden kitlenin özelliklerinin tahmin edilmesi sürecidir. Seçim sürecinde kullanılan yöntemlere göre kitle parametreleri tahmin edilir [54]. Kitleye en uygun örnekleme yönteminin belirlenmesi, parametreye ilişkin örnekleme varyansının en küçük kılınmasıyla mümkündür. Bu nedenle, uygun örnekleme yönteminin seçimi istatistikte oldukça önemlidir.

### **3.2. Tabakalı Rastgele Örnekleme**

Kitleden örneklemelerin seçildiği en temel örnekleme yöntemi, “basit rasgele örnekleme” yöntemidir. Bu yöntemde, sonlu büyüklükteki homojen bir kitleden seçilebilecek tüm mümkün örneklemelere, dolayısıyla her bir örneklem birimine, eşit seçilme şansı verilerek ve seçilen örneklem birimi yerine konulmaksızın ya da konularak  $n$  büyüklüğünde örneklem seçilir. Bu örneklem ile parametre tahminleri yapılır [54].

Örnekleme birimlerinin herhangi bir ölçüsüne ilişkin birimden birime değişim büyük ise, bu durumda kitle, değişkenliği daha küçük alt gruplara ayrılabilir. Böylece, kitle varyansı büyük iken, alt grupların varyansı daha küçük olacaktır. Bu ise, duyarlılıkta önemli bir kazanç sağlar.

İşte, kitle her bir kitle birimi bir ve yalnız bir tabakaya ait olacak ve hiçbir kitle birimi açıkta kalmayacak; tabaka içi değişim olabildiğince küçük, tabakalar arası değişim oldukça büyük kalacak şekilde alt gruplara bölünüp örneklemin her bir tabakadan ayrı ayrı ve birbirinden bağımsız olarak çekildiği örnekleme yöntemine tabakalı örnekleme (stratified sampling) adı verilir [53].



Kitle heterojen yapıda iken basit rastgele örnekleme kullanıldığında, seçilen örneklerle kitlenin iyi temsil edilememesi gibi sorunlar ortaya çıkar. Dolayısıyla, kitle heterojen yapıda iken, her biri homojen olan ve birbiri ile kesişmeyen tabakalar oluşturulup, oluşturulan bu tabakadan basit rastgele örneklem seçmek hem kitlenin daha iyi temsil edilmesini hem de elde edilen tahminlerin hassaslığının artmasını sağlar. Burada her bir tabaka bir kitle olarak düşünülebilir. Her bir tabakaya basit rastgele örnekleme yönteminin uygulandığı tabakalı örnekleme “tabakalı rastgele örnekleme (stratified random sampling)” adı verilmektedir. Ayrıca, tabakalı rastgele örnekleme kullanılarak, kitleyi oluşturan her bir alt gruba yani tabakaya ait istatistikler de elde edilebilir. Tabakalı rastgele örnekleme kullanılmasının nedenleri aşağıdaki gibi özetlenebilir [55]:

- ✓ Belli şartlar altında istatistiğin duyarlılığını arttırabilir olmasıdır. Bu şart, tabaka içi yapının homojen olması yani tabakalara ayrılan birimlerin birbirine yakın olmasıdır. Başka bir ifadeyle, kitle heterojen bir yapıya sahipse yani kitleyi oluşturan birimler arasında uzaklık fazla ise tabakalı rastgele örnekleme kullanılması duyarlılığı arttıracaktır.
- ✓ Kitleyi oluşturan alt grupları yani her bir tabakayı ilgilendiren bilgiler elde edilebilir.
- ✓ Hem idari hem de fiziksel nedenlerden dolayı bilginin toplanmasını kolaylaştırabilir.

### **3.2.1. Problemin Tanımı**

Tabakalı rastgele örnekleme kullanıldığında, araştırmacının örnekleme planını yürütebilmesi için dikkat etmesi gereken en önemli iki husus tabaka sınırlarının belirlenmesi ve tabakalardan seçilecek örneklem büyüklüğünün belirlenmesidir. Aslında bunların her biri birer optimizasyon problemidir.

Tabakalı rastgele örneklemede amaçlanan en önemli şey tahmin varyansını minimize ederek, basit rastgele örneklemeyle kıyasla istatistiksel doğruluğu arttırmaktır [56]. Bu amaca ulaşmak için her bir tabakanın kendi içindeki değişkenliğinin minimum olması gerekmektedir. Buradan da anlaşılacağı üzere, tabaka sınırlarının belirlenmesi tabakalı örneklemin uygulanması aşamasında karşılaşılan en önemli problemlerin başında gelmektedir [57].

Literatürde tabaka sınırlarının belirlenmesi konusunda ilk önerilenlerden ve basit bir yöntem olan Dalenius ve Hodges (1959)'in frekansların kümülatif karekökleri

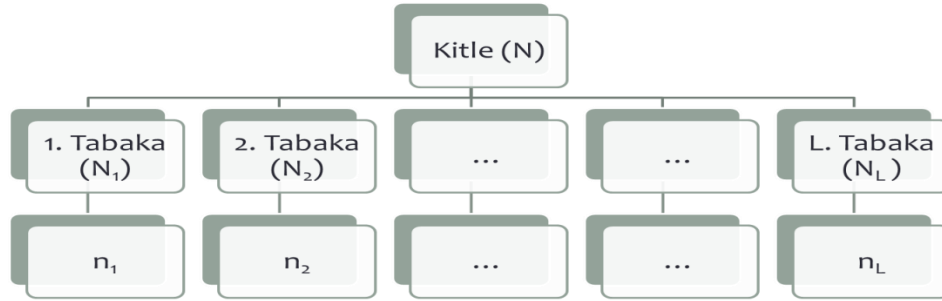
yöntemi (cum $\sqrt{f}$ ) mevcuttur. Lavallée ve Hidiroglou (1988)' nun algoritması ile Gunning ve Horgan (2004)'in geometrik yöntemi çarpıklığı yüksek olan kitleler için önerilirken, Kozak (2004)'in rassal arama yöntemi ile Keskinürk ve Er (2007)'in genetik algoritma yöntemi çarpık olmayan kitleler için de önerilmiştir. Nicolini (2001)'nin NCM'si (Natural Classes Method-Doğal Sınıflar Yöntemi), Ekman (1959)'nin kuralı, Sethi (1959)'nin kuralı gibi birçok farklı yaklaşım da bulunmaktadır [58,59].

Literatürdeki yöntemler genel olarak incelendiğinde, tabaka sınırlarının belirlenmesi konusunda kullanılan klasik yöntemlerin belirli varsayım ve kısıtlar altında gerçekleştirildiği görülmektedir. Örneğin, geometrik yöntem normal dağılıma sahip veri için uygun olmazken, tabakalardaki dağılımın tekdüze olması varsayımına dayanmaktadır. Ayrıca, bu yöntem pozitif çarpıklığın yüksek olduğu durumlarda daha iyi sonuçlar vermektedir [60]. Geometrik yöntemdeki ve diğer klasik yöntemlerdeki buna benzer kısıtlar, herhangi bir varsayıma bağlı olmayan genetik algoritmanın kullanımının önemini ve etkinliğini göstermektedir [58].

Tabakalardan seçilecek örneklem büyüklüğünün belirlenmesi, literatürde en çok ele alınan problemlerden biridir. Araştırma için ayrılan sabit bir bütçe ile her tabakadan bir birim seçmenin ya da her tabakaya ulaşmanın maliyeti bir birim arttırdığı kabul edilen sabit maliyet kısıtı altında, tabakalardan seçilecek örneklem büyüklüğünün belirlenmesi için pek çok çalışma yapılmıştır. Bu çalışmaların ilki Neyman (1934) tarafından yapılmıştır. Neyman, tek değişkenli tabakalı rastgele örneklemede, doğrusal maliyet kısıtı altında örneklem ortalaması istatistiğinin varyansını minimum yapacak örneklem büyüklüğünü elde etmeye çalışmıştır. Bunu yaparken, eşitlik kısıtlı optimizasyon problemlerinin çözümünde kullanılan Lagrange çarpanları yöntemini kullanmış ve tabakalı rastgele örneklemede çok sık kullanılan paylaşırma formülünü elde etmiştir. Tek değişkenli tabakalı rastgele örneklemede, tabakalardan seçilecek örneklem büyüklüğünün belirlenmesi problemi için, son yıllarda yapılan çalışmalardan birkaçı hariç, hemen hemen hepsi doğrusal maliyet kısıtı altında yapılmıştır [55]. Ayrıca örneklemin tabakalara dağıtım problemi için Neyman dağıtımının yanı sıra eşit ve orantılı dağıtım da literatürde [61] yaygın olarak kullanılan yöntemlerin başında gelmektedir.

Bu çalışma, sabit maliyet kısıtı ile örneklem büyüklüğünün ve tabaka sayısının bilindiği varsayımı altında gerçekleştirilmiştir. Öncelikle, tahmin varyansını minimize edecek tabaka sınırları ve belirlenen bu tabakalardan çekilecek örneklem büyüklükleri genetik algoritma yaklaşımı ile belirlenmiştir. Ardından, elde edilen aynı kitle için eşit, orantılı ve Neyman dağıtımı yardımı ile her tabakanın örneklem büyüklüğü elde edilerek, her bir dağıtım sonucundaki tahmin varyansı hesaplanmıştır. Klasik yöntemler olan bu üç yaklaşımdan (eşit, orantılı, Neyman) elde edilen sonuçlar ile sezgisel bir yöntem olan genetik algoritma yaklaşımından elde edilen sonuçlar karşılaştırılmıştır.

### 3.2.2. Problemin Formülasyonu



**Şekil 3.1.** Tabakalı örnekleme

Şekil 3.1’ de görüldüğü gibi tabakalı örneklemede, N büyüklüğündeki kitle  $N_1, N_2, \dots, N_L$  büyüklüklerinde birbiriyle kesişmeyen ve tüm kitleyi oluşturan L tabakadan oluşsun. h- indisi ile tabaka, i-indisi ile birim gösterilsin. Her biri ayrı ayrı bir kitle gibi düşünülen bu tabakalardan  $n_1, n_2, \dots, n_L$  büyüklüğünde örneklem çekilsin. Buna göre h-inci tabaka için gösterimler aşağıdaki gibidir [53]:

$y_{hi}$  h-inci tabakada i-inci birim değeri

$N_h$  h-inci tabaka büyüklüğü

$n_h$  h-inci tabakadan seçilen örneklem büyüklüğü

$W_h = N_h / N$  h-inci tabaka ağırlığı

$f_h = n_h / N_h$  h-inci tabakanın örnekleme oranı

$\bar{Y}_h = \left( \sum_{i=1}^{N_h} y_{hi} \right) / N_h$  h-inci tabakada gerçek ortalama

$$\bar{y}_h = \left( \sum_{i=1}^{n_h} y_{hi} \right) / n_h \quad \text{h-inci tabaka örneklem ortalaması}$$

$$S_h^2 = \frac{\sum_{i=1}^{N_h} (y_{hi} - \bar{Y}_h)^2}{N_h - 1} \quad \text{h-inci tabakada birim başına düşen gerçek varyans}$$

$$s_h^2 = \frac{\sum_{i=1}^{n_h} (y_{hi} - \bar{y}_h)^2}{n_h - 1} \quad \text{h-inci tabakada birim başına düşen örneklem varyansı}$$

**Çizelge 3.1.** Tabakalı örneklemin genel gösterimi

	1. Tabaka N <sub>1</sub> (h=1)	2. Tabaka N <sub>2</sub> (h=2)	...	L. Tabaka N <sub>L</sub> (h=L)
i=1	y <sub>11</sub>	y <sub>21</sub>	...	y <sub>L1</sub>
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
i=N <sub>h</sub>	.	.	.	.
	Y <sub>1</sub>	Y <sub>2</sub>	...	Y <sub>L</sub>

$$N = \sum_{h=1}^L N_h$$

$$n = \sum_{h=1}^L n_h$$

Tabakalanmış bir kitlede kitle ortalamasının gerçek değeri,  $\bar{Y}_b$ , tahmini  $\bar{y}_{tb}$  değeri ile gösterilsin [53].

- ✓ Tabakalı örneklemede ortalama için tahmin varyansı;

$$V(\bar{y}_{tb}) = \sum_{h=1}^L \frac{N_h^2}{N^2} V(\bar{y}_h) = \sum_{h=1}^L W_h^2 V(\bar{y}_h)$$

- ✓ Tabakalara BRÖ uygulandığında h-inci tabaka için varyans;

$$V(\bar{y}_h) = \left(1 - \frac{n_h}{N_h}\right) \frac{S_h^2}{n_h}$$

- ✓ Tabakalı rastgele örnekleme varyansı;

$$V(\bar{y}_{tb}) = \sum_{h=1}^L \frac{N_h^2}{N^2} \left(1 - \frac{n_h}{N_h}\right) \frac{S_h^2}{n_h} \quad (3.1)$$

$$V(\bar{y}_{tb}) = \frac{1}{N^2} \sum_{h=1}^L N_h (N_h - n_h) \frac{S_h^2}{n_h}$$

Tabakalı rastgele örnekleme varyansı olarak Eşitlik 3.1' de verilen formül, bu çalışmanın hedefi olan, minimize edilmeye çalışılan tahmin varyansını göstermektedir.

Tabakalardan seçilecek örneklem büyüklüğünün belirlenmesi ile ilgili literatürde bulunan ve bu çalışmada yer alan yöntemlere ilişkin formüller (sabit maliyet kısıtı altında) aşağıda verilmiştir:

- ✓ Eşit Dağıtım:

$$n_h = \frac{n}{L} \quad n_1 = n_2 = \dots = n_L \quad h = 1, 2, \dots, L \quad (3.2)$$

- ✓ Orantılı Dağıtım:

$$n_h = n * \frac{N_h}{N} \quad h = 1, 2, \dots, L \quad (3.3)$$

- ✓ Neyman Dağıtımı:

$$n_h = n * \frac{N_h * S_h}{\sum_{h=1}^L N_h * S_h} \quad h = 1, 2, \dots, L \quad (3.4)$$

Bu çalışmada yukarıda verilen üç örneklem dağıtım yöntemine ek olarak, her tabakadan çekilecek olan birim sayısına ilişkin herhangi bir kısıt gerektirmeyen bir genetik algoritma yöntemi önerilmiş ve bu üç yöntemle karşılaştırılmıştır.

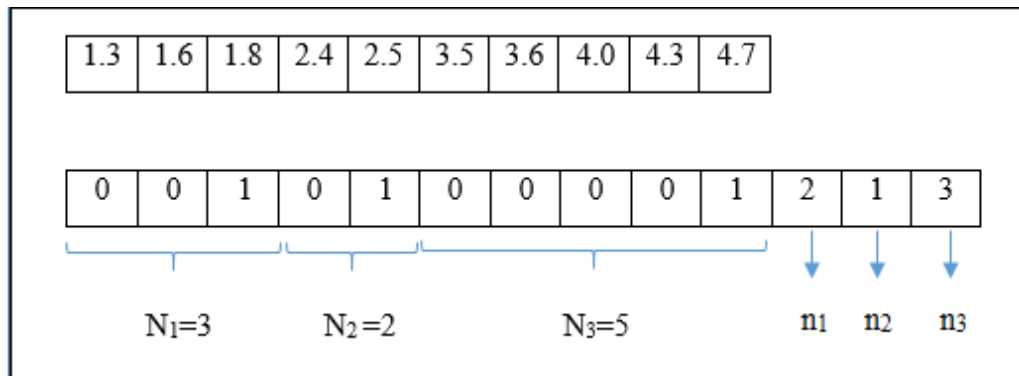
## 4. GELİŞTİRİLEN GENETİK ALGORİTMA

Verilen problemin çözümü için geliştirilen genetik algoritma yaklaşımının adımları ve uygulamada kullanılan genetik algoritma parametreleri bu bölümde açıklanmıştır.

### 4.1. Çözümlerin Kodlanması

Genetik algoritmanın ilk adımı olarak, tabaka sınırlarının belirlenmesi ve örneklem dağıtımı problemlerinin çözümünü temsil edecek olan, kromozom ya da birey olarak adlandırılan sonlu dizilerin oluşturulması yani kromozomların kodlanması gerekir. Tabakalama probleminin genetik algoritma ile çözümü için, değerler kromozomlar içerisinde kodlanır. Kitledeki veriler küçükten büyüğe sıralandığında, yani artan bir değer aralığında veri setindeki değerler her parça bir tabaka sınırını gösterecek şekilde  $Y_1 < Y_2 < \dots < Y_{L-1}$  noktalarından, L parçaya ayrılırlar. Genetik algoritmada, bu yapıyı temsil edebilecek farklı kodlama yöntemleri mevcuttur. En yaygın olarak kullanılan yöntem ikili (binary) kodlamadır. Bunun yanı sıra gerçek-değerli (real-valued), tam sayılı ve üçlü kodlama yöntemleri de kullanılabilir [58].

Bu çalışmada ikili ve gerçek-değerli kodlama yöntemleri kullanılmıştır. Kitlenin oluşturduğu verilerin karşılık geldiği ve değerlerin küçükten büyüğe doğru sıralandığı, kitle büyüklüğü sayısı kadar genden oluşan bir kromozom düşünülmüştür. Bu kromozoma karşılık gelen değerleri tutan, 0 ve 1'lerden oluşan ikili kodlama yöntemi yardımı ile tabaka sınırlarını belirleyecek olan kromozom oluşturulmuştur. Ayrıca, belirlenen tabaka sınırları yardımı ile tabakalara dağıtılacak örneklem büyüklüğünü belirlemek için de gerçek-değerli kodlama yöntemi kullanılmıştır. Her iki durumu basitçe ifade eden örnek bir kodlama biçimi Şekil 3.2' de belirtildiği gibidir.



Şekil 4.1. Basit bir çözüm kodlama örneği

Şekil 4.1' de verilen ilk kromozom veri setindeki yani kitledeki değerleri göstermektedir. Problemdaki verilerin sayısı (N) kromozomdaki genlerin toplam sayısına eşittir. Değerlerin küçükten büyüğe sıralı olduğu bu kromozoma karşılık gelen 0 ve 1'lerden oluşan kromozom ise tabaka sınırlarını ifade etmektedir. Her kromozom için, ilk sıfırdan ilk 1 değerine kadar olan genlerin sayısı 1. tabakanın büyüklüğünü ( $N_1$ ) gösterirken, ilk 1'den sonra gelen 0'dan ikinci 1'e kadar olan genlerin sayısı 2. tabakanın büyüklüğünü gösterir ve böyle devam eder. İkili kodlama yapılan kromozomda 1'lerin karşılık geldiği yerler tabaka sınırlarını göstermektedir. Şekil 4.1' de verilen örnekte,  $N=10$  olarak kitle büyüklüğü ve kitledeki değerler verilmiştir. Buna göre karşılık gelen 1'lerin sayısı  $L=3$  tabaka sayısını gösterirken, bu tabakalara düşen değerler de belirtilmiştir:

1. Tabaka  $\rightarrow$  1.3, 1.6, 1.8  $(N_1=3)$
2. Tabaka  $\rightarrow$  2.4, 2.5  $(N_2=2)$
3. Tabaka  $\rightarrow$  3.5, 3.6, 4.0, 4.3, 4.7  $(N_3=5)$

Bu çalışmada ikili kodlama ile oluşturulan kitle büyüklüğündeki kromozomun sonuna eklenmiş gerçek değerli kodlamayı gösteren sayılar ise toplamları örneklem büyüklüğünü ( $n$ ) veren, her tabakadan çekilecek örneklem büyüklüklerini gösterir. Her kromozomda, o kromozomdaki tabaka sayısı kadar bu değerleri gösteren genler mevcuttur. Bu değerler belirlenen tabaka sınırlarına göre her tabaka için kendi büyüklüğünü aşmayacak şekilde belirlenmiştir.

#### 4.2. Başlangıç Popülasyonu Oluşturma

Kodlanan kromozomların oluşturduğu çözümler kümesini gösteren popülasyonun rastgele oluşturulmasıyla algoritma başlatılır. "1" olarak gösterilen genlerin sayısı, tabaka sayısına ( $L$ ) eşittir. Kromozomlardaki son gen mutlaka "1" olmalıdır, çünkü bu gen son tabakanın üst sınırını ifade etmektedir.

Genetik algoritmanın popülasyon büyüklüğü, algoritmanın hesaplama zamanını doğrudan etkilemektedir. Tercih edilen büyük bir popülasyon büyüklüğü, algoritmanın çözüm uzayında daha etkin bir arama yapmasını sağlar. Ancak, öte yandan algoritmanın çalışma zamanını uzatabilir [58].

Bu çalışmada önerilen genetik algoritma yaklaşımında, her veri seti için farklı tabaka sayılarının yer aldığı ( $L=2,3,4,5$ ) tüm denemelerde popülasyon büyüklüğü

60 olarak alınmıştır. Belirlenen bu parametre ile makul hesaplama zamanında iyi sonuçlar elde edildiği görülmüştür.

### **4.3. Uygunluk Fonksiyonu**

Başlangıç popülasyonu üretildikten sonra, popülasyondaki her kromozom için uygunluk fonksiyonu olarak adlandırılan bir amaç fonksiyonu kullanılarak, bir uygunluk değeri hesaplanır. Uygunluk değeri bireylerin (kromozomların) bir sonraki nesilde hayatta kalma olasılığını belirleyen bir çözüm değeridir.

Bu çalışmadaki problem için önerilen algoritmada, tabakalı rastgele örneklemedeki tahmin varyansını gösteren Eşitlik 3.1' de verilen formülasyon amaç fonksiyonunu yani uygunluk fonksiyonunu ifade etmektedir. Burada amaç, algoritmanın iterasyon süreci boyunca tahmin varyansını en küçük yapmaya çalışmaktır.

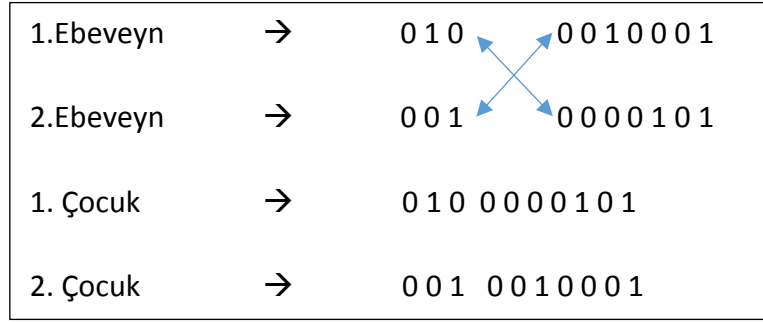
### **4.4. Seçim Yöntemi**

Uygunluk değerlerinin belirlenmesinden sonraki aşama seçim sürecidir. Seçim, kromozomların hesaplanan uygunluk değerlerine göre bir sonraki nesilde hayatta olup olmayacaklarını belirleyen süreçtir. Çalışmanın ikinci bölümünde anlatılan seçim yöntemlerinden elitizm yöntemi bu çalışmada uygulanmıştır. Uygunluk değerleri küçükten büyüğü doğru sıralanan kromozomların ilk %50'lik kısmı bir sonraki nesle aktarılmak üzere korunmuştur. Burada amaç fonksiyonu minimum yapılmak istendiğinden uygunluk değeri küçük olarak elde edilen kromozomlar genetik işlemcilerin uygulanması için seçilmiştir.

### **4.5. Çaprazlama**

Çaprazlama, iki kromozomun kendi aralarında bireysel karakterlerinin değişimini sağlayan önemli bir genetik algoritma işlemcisidir. Çaprazlamanın birçok farklı türünün olduğundan daha önce bahsedilmişti. Bu çalışmada, rastgele bir nokta seçilerek o nokta üzerinden bireyler arasında değişim sağlanan tek noktalı çaprazlama yöntemi kullanılmıştır. Ancak burada tabaka sayısının sabit kalabilmesini sağlamak amacıyla şartlı bir çaprazlama yöntemi geliştirilmiştir.





**Şekil 4.2.** Tek nokta çaprazlamaya basit bir örnek (şartlı)

Şekil 4.2' deki örnek incelendiğinde, algoritmanın başında belirlenen bir değer olan tabaka sayısının, çaprazlama işleminden sonra ebeveyn ve çocuk kromozomlarda farklı olmadığı görülmektedir. Çaprazlama noktasının 3. genden sonra olduğu durum yani çaprazlama noktasından önceki ve sonraki toplam "1" sayısının eşit olduğu durum için bu koşul değişmemiştir. Ancak, çaprazlama noktasının 6. genden sonra olduğu düşünülürse ebeveynlerdeki ve çocuklardaki tabaka sayılarının sabit kalmadığı görülmektedir. Bu istenmeyen durumu gidermek için öncelikle rastgele bir çaprazlama noktası seçilmiş, çaprazlama uygulanmış ve sonrasında oluşan yeni kromozomlardaki "1" değerlerinin toplamının başlangıçta seçilen tabaka sayısına eşit olup olmadığına bakılmıştır. Eşit olduğu durumda yeni çocuklar oluşmuş, eşit olmadığı durumda ise eşit oluncaya kadar basit bir mutasyon yöntemi uygulanmıştır. Burada uygulanan ve şartlı olarak adlandırılan mutasyon yöntemine göre, oluşan çocuk kromozomdaki 1'lerin toplam sayısı tabaka sayısından büyük ise rastgele seçilen 1'ler 0'a dönüştürülmüş, küçük ise yine rastgele seçilen 0'lar 1'e dönüştürülmüştür.

Yüksek çaprazlama oranı, çözüm uzayını hızlı bir şekilde aramayı sağlarken diğer yandan da iyi sonuçlar verecek kromozomların atlanmasına ya da genetik algoritmanın performansının düşmesine neden olabileceğinden daha önce de bahsedilmişti. Çaprazlama oranının düşük olarak belirlenmesi halinde ise, değişime uğrayacak kromozom sayısı azalacak ve dolayısıyla algoritma yavaşlayarak sonuca geç ulaşılacaktır. Bu çalışmada çaprazlama oranı 0,50 olarak kullanılmıştır.

#### 4.6. Mutasyon

Çaprazlama işleminden sonra yapılan mutasyon, bireylerde rastgele değişimler gerçekleştirerek yeni kromozomların oluşmasını sağlar. Bu değişim belirlenen bir

mutasyon oranı ile gerçekleştirilir. Kromozomun sahip olduğu mutasyon oranına göre, hangi kromozomun hangi gen veya genlerinin değiştirileceğine karar verilir.

Literatürdeki uygulamalarda, algoritmanın tamamen rastgele olmasını engellemek amacıyla mutasyon oranının 0,01 ile 0,001 arasında alındığı görülmektedir. Yapılan uygulamada mutasyon oranı 0,01 olarak alınmıştır. Mevcut bulunan farklı mutasyon çeşitlerinden ise “karşılıklı değişim (takas)” yöntemi kullanılmıştır.

Mutasyon Noktası			
Kromozom	→	1 0 1 0 0 1 0 0 1 0	
Mutasyona Uğramış Kromozom	→	1 0 1 0 1 1 0 0 0 0	

**Şekil 4.3.** Karşılıklı değişim mutasyonuna basit bir örnek

Bu mutasyon çeşidinde, rastgele iki nokta seçilmiş ve bu noktalardaki genler birbirleriyle yer değiştirmişlerdir. Uygulanan mutasyon olasılığı ile değişkenliğin sağlanabilmesi için, seçilen iki noktanın birinin “1” değerinin “0” olan gen olması algoritmaya işlenmiştir.

Yapılan bu çalışmada algoritma; çözümleri kodlanması, başlangıç popülasyonunun oluşturulması, uygunluk fonksiyonunun belirlenmesi ve buna göre seçim, çaprazlama ve mutasyon işlemlerinin yapılması ile algoritma hesaplama zamanını esas alan bir iterasyon sınırına kadar çalıştırılmıştır. Durdurma kriterini oluşturan iterasyon sınırı 500 olarak belirlenmiştir.

Tabaka sayısının (L) ve toplam örneklem büyüklüğünün (n) önceden belirlenmiş olduğu bu algorithmada, öncelikli olarak tabaka sınırları genetik algoritma yardımı ile belirlenmiş ve aynı zamanda her bir tabakadan çekilecek örneklem büyüklükleri de genetik algoritma ile hesaplanmıştır. Sınırları genetik algoritma ile belirlenmiş olan aynı kitle için, sabit maliyet kısıtı altında, en yaygın kullanılan üç örneklem dağıtım yöntemi olan eşit, orantılı ve Neyman dağıtım yöntemleri yardımı ile de her bir tabakanın örneklem büyüklüğü hesaplanmıştır ve karşılaştırılmıştır.

## 5. UYGULAMA

Geliştirilen algoritmada, varyans tahminini minimize eden uygunluk fonksiyonu kullanılarak tabaka sınırları genetik algoritma ile elde edilmiş ve yine genetik algoritma ile örneklem dağıtımları da bulunmuştur. Algoritmaya eşit, orantılı ve Neyman dağıtımları da dahil edilerek aynı kitle için bu yöntemlerle de tabakalardaki örneklem büyüklükleri bulunmuştur. Tüm bu işlemleri içeren algoritma MATLAB R2013a programı kullanılarak geliştirilmiştir.

Klasik yöntemler ile genetik algoritmayı karşılaştırmayı hedefleyen bu algoritmanın uygulanması için ikisi gerçek, yedi tanesi ise simülasyonla elde edilen toplam dokuz ayrı veri kümesinde program çalıştırılmıştır.

Kullanılan gerçek veri kümeleri, bugüne kadar akademik düzeyde birçok tez ve araştırmaya konu olan, 1968 yılında 100 Büyük Sanayi Kuruluşu olarak başlatılan ve her yıl geliştirilerek sürdürülen Türkiye'nin 500 Büyük Sanayi Kuruluşu (İSO 500) çalışmasından alınmıştır.

Yurtiçi ve yurtdışında yoğun bir ilgi ile karşılanan Türkiye'nin 500 Büyük Sanayi Kuruluşu çalışması, Türk sanayisinin gelişiminin görülmesine ve geleceğe yönelik yol haritasının belirlenmesine önemli katkılar sağlamaktadır. Sıralamada temel kriter, üretimden satışlardır. Yani sonuçlar, kuruluşların kendi üretmiş olduğu ürünlerden yaptığı satışları kapsamaktadır [62].

Birinci gerçek veri kümesi Türkiye'nin Birinci 500 Büyük Sanayi Kuruluşu, ikinci veri kümesi ise Türkiye'nin İkinci 500 Büyük Sanayi Kuruluşu'nun 2012 yılı verilerini kapsamaktadır. Veriler üretimden satışları (net) TL cinsinden ifade etmektedir. Değerler çok büyük olduğu için ( $*10^9$ ) olacak şekilde sadeleştirilerek işleme alınmıştır.

Türkiye'nin 2012 yılındaki Birinci 500 Büyük Sanayi Kuruluşu'nun satışlarını gösteren ilk veri kümesinde, 17 şirketin bilgilerine ulaşılamamış dolayısıyla bu şirketler hariç tutulmuştur. İşlemler birinci veri kümesi için 483 değer üzerinden (İSO 483) yürütülmüştür.

Türkiye'nin 2012 yılındaki İkinci 500 Büyük Sanayi Kuruluşu'nun satışlarını gösteren ikinci veri kümesinde, 31 şirketin bilgilerine ulaşılamamış dolayısıyla bu şirketler hariç tutulmuştur. İşlemler ikinci veri kümesi için 469 değer üzerinden (İSO 469) yürütülmüştür.

Geriye kalan yedi veri kümesi farklı dağılımlardan farklı parametrelerle türetilerek kullanılmıştır.  $\mu=100$  ve  $\sigma=10$  parametreleri ile normal dağılımdan, 1, 5, 10 ve 15 serbestlik dereceleri ile ki-kare dağılımından,  $a=10$  ve  $b=3$  parametreleri ile beta dağılımından ve son olarak da tekdüze dağılımdan rastgele olarak  $N=1000$  olacak şekilde farklı veriler üretilmiştir.

Çalışmada yapılacak uygulamada tabaka sayılarının (L) ve toplam örneklem büyüklüğünün önceden belirlenen değerler olacağından bahsedilmiştir. Örneklemelerin her biri için 2, 3, 4 ve 5 olmak üzere farklı tabaka sayılarında çözümler elde edilmiştir. Örneklem büyüklüğü ise İSO 483 ve İSO 469 veri kümeleri için 80 ( $n=80$ ), rastgele üretilen veri kümeleri için ise 100 ( $n=100$ ) olarak alınmıştır. Kullanılan veriler için bahsedilen durumlar Çizelge 5.1’ de özetlenmiştir:

**Çizelge 5.1.** Uygulamada kullanılan veri kümelerinin özellikleri

Veri Kümeleri	N (kitle büyüklüğü)	n (örneklem büyüklüğü)	L (tabaka sayısı)
İSO 483	483	80	2, 3, 4, 5
İSO 469	469	80	2, 3, 4, 5
Normal ( $\mu=100, \sigma=10$ )	1000	100	2, 3, 4, 5
$\chi_1^2$	1000	100	2, 3, 4, 5
$\chi_5^2$	1000	100	2, 3, 4, 5
$\chi_{10}^2$	1000	100	2, 3, 4, 5
$\chi_{15}^2$	1000	100	2, 3, 4, 5
Beta ( $a=10, b=3$ )	1000	100	2, 3, 4, 5
Tekdüze	1000	100	2, 3, 4, 5

Genetik algoritmanın klasik dağıtım yöntemlerine göre daha etkili olduğunu göstermek için yazılan algordmada kullanılan parametrelerin değerleri aşağıda verildiği gibidir:

**Çizelge 5.2.** Geliştirilen genetik algoritmanın parametre değerleri

Parametre	Değeri
Popülasyon büyüklüğü	60
Çaprazlama oranı	0,50
Mutasyon oranı	0,01
İterasyon sayısı	500

**Çizelge 5.3.** Simülasyon ile elde edilen veri kümeleri için bulunan tahmin varyansı değerleri

Veri Kümesi	Tabaka Sayısı (L)	Örneklem Dağıtım Yöntemi			
		Genetik Algoritma	Eşit Dağıtım	Orantılı Dağıtım	Neyman Dağıtım
Normal ( $\mu=100$ , $\sigma=10$ )	2	$1,38*10^{-5}$	$9,73*10^{-5}$	$7,52*10^{-5}$	$8,19*10^{-5}$
	3	$1,31*10^{-5}$	-	$8,42*10^{-5}$	$6,59*10^{-5}$
	4	$5,50*10^{-6}$	$5,49*10^{-5}$	$5,58*10^{-5}$	$4,83*10^{-5}$
	5	$6,95*10^{-6}$	$3,92*10^{-5}$	$5,58*10^{-5}$	$5,74*10^{-5}$
$\chi_1^2$	2	$3,40*10^{-11}$	$3,28*10^{-6}$	$2,14*10^{-6}$	$2,24*10^{-6}$
	3	$2,13*10^{-10}$	-	$2,01*10^{-6}$	$2,15*10^{-6}$
	4	$9,68*10^{-11}$	-	$1,29*10^{-6}$	$1,95*10^{-6}$
	5	$1,33*10^{-11}$	$1,38*10^{-6}$	$3,36*10^{-6}$	$1,50*10^{-6}$
$\chi_5^2$	2	$5,84*10^{-6}$	0,06596	0,06360	0,05286
	3	$7,68*10^{-5}$	0,03700	0,04135	0,03541
	4	$7,68*10^{-6}$	-	0,03318	0,03737
	5	$9,24*10^{-5}$	0,02533	0,02381	0,02438
$\chi_{10}^2$	2	$8,61*10^{-6}$	0,05571	0,04867	0,05491
	3	$2,65*10^{-5}$	0,03631	0,02827	0,03940
	4	$3,99*10^{-5}$	0,01019	0,01371	0,01509
	5	$1,13*10^{-5}$	0,00831	0,00890	0,00958
$\chi_{15}^2$	2	$6,68*10^{-5}$	0,03854	0,03792	0,04170
	3	$1,26*10^{-5}$	0,05122	0,04921	0,04182
	4	$5,04*10^{-5}$	0,02155	0,02344	0,01718
	5	$3,22*10^{-5}$	0,01757	0,01968	0,02286
Beta (a=10, b=3)	2	$7,90*10^{-6}$	0,05377	0,04788	0,04300
	3	$1,03*10^{-5}$	0,04007	0,04475	0,03839
	4	$2,07*10^{-6}$	0,04554	0,05094	0,04652
	5	$2,39*10^{-7}$	0,00393	0,00327	0,00357
Tekdüze	2	$2,57*10^{-5}$	0,05322	0,04906	0,04919
	3	$5,09*10^{-5}$	0,00482	0,00623	0,00564
	4	$8,00*10^{-6}$	0,03894	0,03616	0,03216
	5	$1,23*10^{-6}$	0,05329	0,04658	0,04935

Simülasyon ile farklı dağılımlardan elde edilen yedi ayrı kitleye geliştirilen genetik algoritma uygulanmış, tabaka sınırları ve örneklem büyüklükleri yine önerilen genetik algoritma ile hesaplanmıştır. Örneklem dağıtımını ayrıca eşit (3.2), orantılı (3.3) ve Neyman (3.4) yöntemleri ile de elde edilmiştir. Bahsi geçen tüm işlemler sonucunda elde edilen tahmin varyansı değerleri Çizelge 5.3' te verilmiştir.

Türetilen kitleler için kitle büyüklüğü 1000, örneklem büyüklüğü 100 olarak alınmıştır. Eşit dağıtım yapılırken, örneklem büyüklüğünün ( $n=100$ ) tabaka sayılarına tam olarak bölünemediği durumlar için  $n_h$  değerlerinin toplamının yuvarlamadan dolayı tam olarak 80 olmadığı durumlar oluşmuştur. Ayrıca, önceden sınırları genetik algoritma ile belirlenen bir kitleden örneklem dağıtımını yaparken, eşit dağıtımda  $n_h$  değerlerinin  $N_h$  değerlerinden büyük olduğu durumlar oluşmuş, bu değerler için tahmin varyansı değeri hesaplanamamıştır. Çizelge 5.3' te böyle durumlar için “-” ifadesi kullanılmıştır.

Çizelge 5.3' te, örneklem dağıtımını amaçlı uygulanan genetik algoritma, eşit, orantılı ve Neyman dağıtım yöntemlerinden elde edilen tahmin varyansı değerleri farklı tabaka büyüklükleri için verilmiştir. Sonuçlar incelendiğinde, tabaka büyüklüklerine bakılmaksızın genetik algorithmadan elde edilen varyans değerlerinin, diğer yöntemlerle elde edilen varyans değerlerinden daha küçük olduğu açıkça görülmektedir. Özellikle farklı parametrelerle farklı dağılımlardan elde edilen örnek kitlelerden hepsinde de, toplam örneklem büyüklüğünü hiçbir kısıta veya varsayıma bağlı olmadan dağıtan genetik algoritma yöntemi daha iyi sonuçlar vermiştir. Normal dağılımlı kitlede genetik algorithmadan elde edilen sonuçlarla diğer yöntemlerden elde edilen sonuçlar arasındaki fark az iken, ki kare, beta ve tekdüze dağılımdan elde edilen kitlelerde bu fark oldukça fazladır.

**Çizelge 5.4.** Gerçek veri kümeleri için bulunan tahmin varyansı değerleri

Veri Kümesi	Tabaka Sayısı (L)	Örneklem Dağıtım Yöntemi			
		Genetik Algoritma	Eşit Dağıtım	Orantılı Dağıtım	Neyman Dağıtım
İSO 483 ( $*10^9$ )	2	$1,55*10^{-5}$	0,43016	1,37691	11,53570
	3	$1,50*10^{-7}$	0,16499	0,40722	0,38126
	4	$6,93*10^{-6}$	0,56561	0,87712	0,22191
	5	$8,54*10^{-8}$	-	0,05267	0,22189
İSO 469 ( $*10^9$ )	2	$5,66*10^{-7}$	0,00052	0,00038	0,00041
	3	$4,10*10^{-7}$	0,00044	0,00046	0,00049
	4	$1,35*10^{-7}$	0,00032	0,00032	0,00041
	5	$3,23*10^{-7}$	0,00025	0,00032	0,00035

Simülasyon ile elde edilen veri setleri haricinde, algoritma iki tane gerçek veri kümesine de uygulanmış ve sonuçlar Çizelge 5.4' te verilmiştir. Önceden sınırları genetik algoritma ile belirlenen bir kitleden örneklem dağıtımı yaparken, eşit dağıtımda  $n_h$  değerlerinin  $N_h$  değerlerinden büyük olduğu durumlar oluşmuş, bu değerler için tahmin varyansı değeri hesaplanamamıştır. Çizelge 5.4' te böyle durumlar için "-" ifadesi kullanılmıştır. Çizelge 5.4 incelendiğinde, gerçek verilerden elde edilen sonuçlarda da genetik algoritmanın performansının diğer yaklaşımlara göre daha iyi olduğu açıkça görülmektedir. Farklı tabaka büyüklükleri için sonuçların değişmediği açıktır.

Hem simülasyon verileri için hem de gerçek veri kümeleri için bulunan tahmin varyansı değerlerinin hangi tabaka sınırları ve hangi örneklem büyüklükleri kullanıldığında elde edildiği dokuz farklı kitle için aşağıdaki çizelgelerde özetlenmiştir. Çizelgelerde eşit dağıtım yöntemi sütunlarında koyu olarak gösterilen gözeler daha önce açıklanan "-" ifadelerine karşılık gelmektedir.

**Çizelge 5.5.** ISO 483 verileri için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

ISO 483		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$
2	1	359	1	359	40	359	59	359	79
	2	124	79	124	40	124	21	124	1
3	1	421	23	421	27	421	69	421	78
	2	34	34	34	27	34	6	34	1
	3	28	23	28	27	28	5	28	1
4	1	306	1	306	20	306	51	306	77
	2	37	13	37	20	37	6	37	1
	3	42	7	42	20	42	7	42	1
	4	98	59	98	20	98	16	98	1
5	1	11	1	<b>11</b>	<b>16</b>	11	2	11	71
	2	282	2	282	16	282	47	282	6
	3	74	21	74	16	74	12	74	1
	4	60	8	60	16	60	10	60	1
	5	56	48	56	16	56	9	56	1

Geliştirilen algoritma verilere uygulandığında, ISO 483 veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.5' te verilmiştir.

**Çizelge 5.6.** ISO 469 verileri için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

ISO 469		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>
2	1	377	1	377	40	377	64	377	79
	2	92	79	92	40	92	16	92	1
3	1	72	40	72	27	72	12	72	1
	2	59	39	59	27	59	10	59	1
	3	338	1	338	27	338	58	338	78
4	1	357	1	357	20	357	61	357	77
	2	59	41	59	20	59	10	59	1
	3	20	8	20	20	20	3	20	1
	4	33	30	33	20	33	6	33	1
5	1	46	15	46	16	46	8	46	1
	2	37	20	37	16	37	6	37	1
	3	26	4	26	16	26	4	26	1
	4	285	1	285	16	285	49	285	76
	5	75	40	75	16	75	13	75	1

Geliştirilen algoritma verilere uygulandığında, ISO 469 veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.6' da verilmiştir.

**Çizelge 5.7.** Normal dağılımdan ( $\mu = 100, \sigma = 10$ ) elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

Normal ( $\mu = 100, \sigma = 10$ )		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>
2	1	106	98	106	50	106	11	106	2
	2	894	1	894	50	894	89	894	98
3	1	12	12	12	33	12	1	12	1
	2	93	87	93	33	93	9	93	1
	3	895	1	895	33	895	90	895	98
4	1	379	1	379	25	379	38	379	64
	2	291	96	291	25	291	29	291	9
	3	300	1	300	25	300	30	300	25
	4	30	1	30	25	30	3	30	2
5	1	427	6	427	20	427	43	427	66
	2	115	5	115	20	115	12	115	1
	3	59	10	59	20	59	6	59	1
	4	137	77	137	20	137	14	137	1
	5	262	2	262	20	262	25	262	31



Geliştirilen algoritma verilere uygulandığında, Normal dağılımdan ( $\mu=100, \sigma=10$ ) elde edilen veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.7' de verilmiştir.

**Çizelge 5.8.**  $\chi_1^2$  dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

$\chi_1^2$		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$
2	1	123	99	123	50	123	12	123	1
	2	877	1	877	50	877	88	877	99
3	1	23	12	<b>23</b>	<b>33</b>	23	2	23	1
	2	172	87	172	33	172	17	172	1
	3	805	1	805	33	805	81	805	98
4	1	13	13	<b>13</b>	<b>25</b>	13	1	13	1
	2	103	80	103	25	103	10	103	1
	3	67	6	67	25	67	7	67	1
	4	817	1	817	25	817	82	817	97
5	1	83	16	83	20	83	8	83	1
	2	53	19	53	20	53	5	53	1
	3	298	43	298	20	298	30	298	1
	4	131	21	131	20	131	13	131	1
	5	435	1	435	20	435	44	435	96

Geliştirilen algoritma verilere uygulandığında,  $\chi_1^2$  dağılımından elde edilen veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.8' de verilmiştir.

**Çizelge 5.9.**  $\chi^2_5$  dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

$\chi^2_5$		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$
2	1	910	10	910	50	910	91	910	99
	2	90	90	90	50	90	9	90	1
3	1	824	2	824	33	824	82	824	98
	2	60	25	60	33	60	6	60	1
	3	116	73	116	33	116	12	116	1
4	1	25	5	25	25	25	2	25	1
	2	73	34	73	25	73	7	73	1
	3	79	60	79	25	79	8	79	1
	4	823	1	823	25	823	83	823	97
5	1	177	72	177	20	177	18	177	2
	2	34	5	34	20	34	3	34	1
	3	46	11	46	20	46	5	46	1
	4	723	1	723	20	723	72	723	95
	5	20	11	20	20	20	2	20	1

Geliştirilen algoritma verilere uygulandığında,  $\chi^2_5$  dağılımından elde edilen veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.9' da verilmiştir.

**Çizelge 5.10.**  $\chi^2_{10}$  dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

$\chi^2_{10}$		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$
2	1	901	1	901	50	901	90	901	99
	2	99	99	99	50	99	10	99	1
3	1	127	65	127	33	127	13	127	1
	2	819	1	819	33	819	82	819	97
	3	54	34	54	33	54	5	54	1
4	1	638	1	638	25	638	64	638	96
	2	55	38	55	25	55	6	55	1
	3	163	2	163	25	163	16	163	2
	4	144	59	144	25	144	14	144	1
5	1	585	1	585	20	585	59	585	94
	2	183	58	183	20	183	18	183	3
	3	25	3	25	20	25	3	25	1
	4	115	6	115	20	115	11	115	1
	5	92	32	92	20	92	9	92	1

Geliştirilen algoritma verilere uygulandığında,  $\chi_{10}^2$  dağılımından elde edilen veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.10' da verilmiştir.

**Çizelge 5.11.**  $\chi_{15}^2$  dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

$\chi_{15}^2$		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$	$N_h$	$n_h$
2	1	837	1	837	50	837	84	837	99
	2	163	99	163	50	163	16	163	1
3	1	115	91	115	33	115	12	115	1
	2	39	8	39	33	39	4	39	1
	3	846	1	846	33	846	84	846	98
4	1	111	66	111	25	111	11	111	1
	2	156	31	156	25	156	16	156	1
	3	707	1	707	25	707	70	707	97
	4	26	2	26	25	26	3	26	1
5	1	150	31	150	20	150	15	150	1
	2	62	30	62	20	62	6	62	1
	3	55	31	55	20	55	6	55	1
	4	27	6	27	20	27	3	27	1
	5	706	2	706	20	706	70	706	96

Geliştirilen algoritma verilere uygulandığında  $\chi_{15}^2$  dağılımından elde edilen veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.11' de verilmiştir.

**Çizelge 5.12.** Beta dağılımından (a=10,b=3) elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

Beta (a=10, b=3)		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>
2	1	901	1	901	50	901	90	901	99
	2	99	99	99	50	99	10	99	1
3	1	827	1	827	33	827	83	827	98
	2	104	32	104	33	104	10	104	1
	3	69	67	69	33	69	7	69	1
4	1	850	1	850	25	850	85	850	97
	2	11	2	11	25	11	1	11	1
	3	67	33	67	25	67	7	67	1
	4	72	64	72	25	72	7	72	1
5	1	452	1	452	20	452	45	452	83
	2	226	17	226	20	226	23	226	11
	3	141	20	141	20	141	13	141	3
	4	65	54	65	20	65	7	65	1
	5	116	8	116	20	116	12	116	2

Geliştirilen algoritma verilere uygulandığında, Beta dağılımından (a=10,b=3) elde edilen veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.12' de verilmiştir.

**Çizelge 5.13.** Tekdüze dağılımından elde edilen veriler için genetik algoritma ile elde edilen tabaka sınırları ve dört yöntem için örneklem dağıtımları

Tekdüze		Örneklem Dağıtım Yöntemi							
L	h	Genetik Algoritma		Eşit Dağıtım		Orantılı Dağıtım		Neyman Dağıtım	
		N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>	N <sub>h</sub>	n <sub>h</sub>
2	1	869	1	869	50	869	87	869	99
	2	131	99	131	50	131	13	131	1
3	1	166	98	166	33	166	17	166	3
	2	336	1	336	33	336	33	336	28
	3	498	1	498	33	498	50	498	69
4	1	85	69	85	25	85	9	85	1
	2	80	23	80	25	80	8	80	1
	3	792	2	792	25	792	79	792	97
	4	43	6	43	25	43	4	43	1
5	1	44	1	44	20	44	4	44	1
	2	803	1	803	20	803	85	803	94
	3	65	17	65	20	65	7	65	1
	4	32	25	32	20	32	3	32	1
	5	56	56	56	20	56	1	56	3

Geliştirilen algoritma verilere uygulandığında, Tekdüze dağılımından elde edilen veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri Çizelge 5.13' te verilmiştir.

## 6. SONUÇLAR

Son yıllarda, büyük boyutlu ve karmaşık yapıli optimizasyon problemlerinin çözümünde bilinen tam yöntemlerin yetersiz kalmasından dolayı, bazı araştırmacılar makul zamanda çözüme ulaşabilen alternatif yöntemler üzerine yoğunlaşmışlardır. Doğada kusursuz işleyen sistemlerden esinlenerek geliştirilen optimizasyon yöntemleri “sezgisel yöntemler” olarak adlandırılmaktadır. En iyi bilinen sezgisel yöntemlerden biri, evrim teorisini esas alan “genetik algoritma” yaklaşımıdır. Özellikle, matematiksel modellemenin yapılamadığı veya zor olduğu, kesin çözümün tam yöntemlerle bulunamadığı, büyük boyutlu optimizasyon problemlerinin çözümünde genetik algoritmalarla yararlanılabilmektedir. Mühendislik, Yöneylem Araştırması, Tıp, Sosyal Bilimler, Matematik, İstatistik gibi geniş bir yelpazede bu sezgisel yönteme başvurulur.

Bir araştırma için gerekli verilerin elde edilmesi, bu verilerin çözümlenmesi ve çözümlene sonucu elde edilen bulguların araştırma amaçları için yorumlanmasıyla ilgili en önemli araçlardan biri örneklemedir. Günümüzde birçok alanda örnekleme yöntemlerinden yararlanılmaktadır. Örnekleme amacı kitleyi en iyi temsil edecek örnekleme oluşturmadır. Yani kitle parametresinin tahminine ilişkin varyansın olabildiğince küçük olmasını sağlamaktır. Tabakalı rastgele örneklemede amaçlanan en önemli şey tahmin varyansını minimize ederek, basit rastgele örneklemeyle kıyasla istatistiksel doğruluğu arttırmaktır. Bu amaca ulaşmak için her bir tabakanın kendi içindeki değişkenliğin minimum, tabakalar arası değişkenliğin ise maksimum olması gerekmektedir. Yani burada, tabaka sınırlarının belirlenmesi oldukça önemli bir optimizasyon problemidir.

Tezin uygulama kısmında, tahmin varyansını minimize edecek biçimde bir genetik algoritma geliştirilerek tabaka sınırları elde edilmiş ve her bir tabakaya düşecek olan örneklem büyüklüğü bulunmuştur. Algoritmaya eşit, orantılı ve Neyman dağıtımları da dahil edilerek aynı kitle için bu yöntemlerle de tabakalardaki örneklem büyüklükleri bulunmuştur. Geliştirilen genetik algoritmaya dayalı yaklaşımın performansını değerlendirmek amacıyla ikisi gerçek veri seti, yedi tanesi ise simülasyonla elde edilen veri seti olmak üzere dokuz ayrı veri setine geliştirilen algoritma uygulanmıştır. Bu amaçla karşılaştırma yapabilmek için belirtilen veri setlerine bahsedilen klasik yaklaşımlar da uygulanarak sonuçlar elde edilmiştir.

Geliştirilen algoritma verilere uygulandığında, her bir veri seti için genetik algoritma kullanılarak elde edilen tabaka sınırları ile genetik algoritma, eşit, orantılı, Neyman dağıtım yöntemleriyle elde edilen tabakalardaki örneklem büyüklükleri çizelge 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13' te verilmiştir. Bu çizelgeler incelendiğinde, her bir veri seti için genetik algoritma ile yapılan örneklem dağıtımında aykırı değerlerden kaynaklı olarak, tahmin varyansını minimize edecek dağıtımlar olduğu görülmektedir. Örneğin Çizelge 5.5 incelenecek olursa, buradaki İSO 483 veri seti oldukça heterojen bir yapıya sahip olduğundan genetik algoritma ile yapılan tabakalara örneklem dağıtımının buradaki aykırı değerleri gözeterek tahmin varyansını küçültmeye gittiği açıktır. Aynı şekilde diğer veri setleri için de aynı durum söz konusudur. Bu da genetik algoritmanın aykırı değerlerden etkilenmeden, sadece optimizasyon problemindeki amaç fonksiyonuna odaklı olarak çalıştığının göstergesidir.

Çizelge 5.3 incelendiğinde, tabaka büyüklüklerine bakılmaksızın genetik algoritmadan elde edilen varyans değerlerinin, diğer yöntemlerle elde edilen varyans değerlerinden daha küçük olduğu açıkça görülmektedir. Özellikle farklı parametrelerle farklı dağılımlardan elde edilen kitlelerin hepsinde de, toplam örneklem büyüklüğünü hiçbir kısıta veya varsayıma bağlı olmadan dağıtan genetik algoritma yöntemi daha iyi sonuçlar vermiştir. Normal dağılımlı kitlede genetik algoritmadan elde edilen sonuçlarla diğer yöntemlerden elde edilen sonuçlar arasındaki fark azken, ki-kare, beta ve tekdüze dağılımdan elde edilen kitlelerde bu fark oldukça fazladır. Bunun nedeni ise, tabakalı rastgele örnekleme özellikle aykırı değere sahip ve heterojen kitlelerde kullanılmalıdır.

Çizelge 5.4' te verilen, gerçek verilerden elde edilen sonuçlarda da genetik algoritmanın performansının diğer yöntemlere göre daha iyi olduğu açıkça görülmektedir.

Sabit maliyet kısıtı altında yapılan uygulama sonucunda, tabaka sayısı her ne olursa olsun herhangi bir kısıta bağlı olmadan çalışabilen ve tez çalışmasında geliştirilen genetik algoritmanın, örneklem dağıtımı için literatürde kullanılan eşit, orantılı ve Neyman dağıtım yöntemlerine göre tahmin varyansını elde etmede daha başarılı olduğu, farklı veri setleri için değişmeksizin daha iyi performans gösterdiği görülmüştür. Geliştirilen genetik algoritmanın, heterojen kitleleri

tabakalandırmada başarılı olduđu uygulamada kullanılan veri setleri üzerinden elde edilen tahmin varyansı deęerleri ile doęrulanmıřtır.



## KAYNAKLAR

- [1] Murty, K.G., *Optimization Models for Decision Making, Chapter 1: Models for Decision Making*, Internet Edition, 1, 1-18, **2003**.
- [2] Aladağ, Ç.H., *Tabu Arama Algoritması İle Bir Ders Zaman Çizelgeleme Probleminin Çözümü*, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara, **2004**.
- [3] Yeniay, Ö., *Taguchi Deney Tasarımı Problemlerine Genetik Algoritma Yaklaşımı*, Doktora Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara, **1999**.
- [4] Michalewicz, Z., *Genetic Algorithms + Data Structures = Evolution Program*, 2nd Edition, Springer-Verlag, New York, 340, **1994**.
- [5] Srinivas, M., Patnatik, L.M., Genetic algorithms: A survey , *IEEE Computer Society*, 27(6), 17-26, **1994**.
- [6] Beasley, D., Bull, D.R., Martin, R.R., An overview of genetic algorithms: Part 1 – Fundamentals, *University of Computing*, 15(2), 258-269, **1993**.
- [7] Reeves, C.R., *Modern Heuristic Techniques for Combinatorial Problems*, McGraw-Hill, London, **1995**.
- [8] Karaboğa, D., *Yapay Zeka Optimizasyon Algoritmaları*, 2. Baskı, Nobel Yayın Dağıtım, Ankara, **2011**.
- [9] Talbi, E.G., *Metaheuristics from Design to Implementation*, John Wiley and Sons, **2009**.
- [10] Osman, I.H., *Metastrategy Simulated Annealing and Tabu Search Algorithms for Combinatorial Optimization Problems*, PhD. Thesis, Imperial College University of London, **1991**.
- [11] Glover, F., Future paths for integer programming and links to artificial intelligence, *Computers & Operations Research*. 13(5), 533-549, **1986**.
- [12] Laporte, G., Osman, I.H., *Annals of Operations Research: Metaheuristics in Combinatorial Optimization*, J.C. Baltzer Science Publishers, Switzerland, 60, **1995**.
- [13] Gendreau, M., Laporte, G., Potvin, J.Y., *Local Search Algorithms: Metaheuristic for The Vehicle Routing Problem*, John Wiley and Sons, Chichester, **1995**.
- [14] Al-Mahmeed, A.S., *Meta-Heuristics: Theory and Applications: Tabu Search Combination and Integration*, (eds: Osman, I.H. and Kelly, J.P.), Kluwer Academic Publishers, 319-330, **1996**.
- [15] Charon, I., Hurdy, O., *Meta-Heuristics: Theory and Applications: Mixing Different Components of Metaheuristics*, (eds: Osman, I.H. and Kelly, J.P.), Kluwer Academic Publishers, 589-604, **1996**.

- [16] Osman, I.H., Kelly, J.P., *Meta-Heuristics: Theory and Applications*, Kluwer Academic Publishers, **1996**.
- [17] Reeves, C.R., *Modern Heuristic Techniques for Combinatorial Problems*, Halsted Press, John Wiley and Sons., New York, **1993**.
- [18] Glover, F., Laguna, M., *Tabu Search*, Kluwer Academic Publishers, **1997**.
- [19] Taşkın, Ç., Emel, G.G., *Sayısal Yöntemlerde Genetik Algoritmalar*, 1. Baskı, Alfa Aktüel Yayınları, **2009**.
- [20] Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, **1975**.
- [21] Elmas, Ç., *Yapay Zeka Uygulamaları*, 2. Baskı, Seçkin Yayıncılık, **2011**.
- [22] Özçakar, N., Genetik algoritmalar, *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 27(1), 69-82, **1998**.
- [23] Lee, C.Y., Kim, Seok J., Parallel genetic algorithm for the earliness-tardiness job scheduling problem with general penalty weights, *Computers & Industrial Engineering*, 28(2), 231-243, **1995**.
- [24] Okkalı, A., *Genetik Algoritmalar İle Aydınlatma Hesabı Optimizasyonu*, Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Kocaeli, **2013**.
- [25] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, **1989**.
- [26] Tabak, O., *Genetik Algoritma İle Kapasiteli Servis Güzergahı Belirlenmesi ve Bir Uygulama*, Yüksek Lisans Tezi, Anadolu Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, **2008**.
- [27] Candan, G., *Esnek Üretim Çizelgeleme Probleminin Genetik Algoritma Ve Bulanık Mantık Yöntemleri İle Çözülmesi*, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Sakarya, **2010**.
- [28] Özkan, R., *Tek Modelli Deterministik Montaj Hattı Dengeleme Problemlerine Genetik Algoritma İle Çözüm Yaklaşımı*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, **2003**.
- [29] Koza, J.R., Two ways of discovering the size and shape of a computer program to solve a problem, *Proceedings of the Sixth International Conference on Genetic Algorithm*, 287-294, **1995**.
- [30] Şen, Z., *Genetik Algoritmalar ve En İyileme Yöntemleri*, Su Vakfı, İstanbul, **2004**.
- [31] Masao, M., Tseng, C.C., A genetic algorithm for multi-model resource constrained project scheduling problem, *European Journal of Operational Research*, 100(1), 134-141, **1997**.

- [32] Paksoy, S., *Genetik Algoritmayla Proje Çizelgeleme*, Doktora Tezi, Çukurova Üniversitesi Sosyal Bilimler Enstitüsü, Adana, **2007**.
- [33] Buckles, B.P., Frederick E.P., Genetic Algorithms, *IEEE Computer Society Press, Technology Series*, **1992**.
- [34] Yeo, M.F., Agyei, E.O., Optimising engineering problems using genetic algorithms, *Engineering Computations*, 15(2), 268-280, **1998**.
- [35] Chambers, L.D., *Practical Handbook of Genetic Algorithms: Complex Coding Systems Volume III*, CRC Press, USA, **1999**.
- [36] Chambers, L.D., *Practical Handbook of Genetic Algorithms: Complex Coding Systems Volume II*, CRC Press, USA, **1999**.
- [37] Jang, J.S.R., *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, USA, **1997**.
- [38] Fırlalı, A., Orhan, E., Genetik Algoritmalarla Akış Tipi Çizelgelemede Üretim Yöntemi Optimizasyonu, *İstanbul Teknik Üniversitesi Dergisi*, 1(1), **2002**.
- [39] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Company, Inc., USA, **1989**.
- [40] Chan, F.T.S., Chung, S.H., Wadhwa, S., A hybrid genetic algorithm for production and distribution, *The International Journal of Management Science, Omega*, 33, 345-555, **2005**.
- [41] Genel, H., *Genetik Algoritmalarla Portföy Optimizasyonu*, Yüksek Lisans Tezi, Ankara Üniversitesi Sosyal Bilimler Enstitüsü, Ankara, 2004.
- [42] İşçi, Ö., Korukoğlu, S., Genetik algoritma yaklaşımı ve yöneylem araştırmasında bir uygulama, *Yönetim ve Ekonomi*, 10(2), **2003**.
- [43] Altıparmak, F., Dengiz, B., An evolutionary approach for reliability optimization in fixed topology computer networks, *Transactions on Operational Research*, 12(1-2), 57-75, **2000**.
- [44] Kahvecioğlu, A., Onarılabilir elemanlara önleyici bakımın etkisi ve optimizasyonu, *Mühendis ve Makine*, 45(531), 43-51, **2004**.
- [45] Nearchou, A.C., Path planning of mobile robot using genetic heuristics, *Robotica*, 16, 575-588, **1998**.
- [46] Yeniay, Ö., An overview of genetic algorithms, *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*, 2(1), 40, **2001**.
- [47] Işık, Y., *Genetik Algoritma Tabanlı Bulanık Kontrolün Uçuş Kontrol Sistem Tasarımına Uygulanması*, Anadolu Üniversitesi Fen Bilimleri Enstitüsü, Eskişehir, Doktora Tezi, **2006**.
- [48] Mitchell, M., Charles, E.T., Evolutionary computations: an overview, *Annual Review of Ecology and Systematics*, 30, 593-616, **1999**.

- [49] Hulin, M., An optimal stop criterion for genetic algorithms: A bayesian approach, *Proceedings of the 7th International Coference on Genetic Algorithms*, Michigan State University, 135, **1997**.
- [50] İşlier, A.A., Üretim hücrelerinin bir genetik algoritma kullanılarak oluşturulması, *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*, 2(1), 147, **2001**.
- [51] Gonzales E.L., Fernandez M.A.R., Genetic optimisation of a fuzzy distribution model, *International Journal of Physical Distribution and Logistics Management*, 30(7-8), 681-696, **2000**.
- [52] Emel, G.G., Taşkın, Ç., Genetik algoritmalar ve uygulama alanları, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21(1), 129-152, **2002**.
- [53] Çıngı, H., *Örnekleme Kuramı*, 3. Baskı, Bizim Büro Basımevi, Ankara, **2009**.
- [54] Koyuncu, N., *Tabakalı Rasgele Örneklemede Yardımcı Değişken Bilgisi Kullanılarak Kitle Ortalaması ve Varyansının Tahmin Edilmesi*, Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara, **2007**.
- [55] Şahin, S.T., *Tabakalı Tesadüfi Örneklemede Doğrusal Olmayan Maliyet Kısıtları Altında Örnek Çapının Tabakalara Paylaştırılması*, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, **2010**.
- [56] Cochran, W.G., *Sampling Techniques*. 2nd Edition, John Wiley and Sons, USA, **1963**.
- [57] Er, Ş., Keskindürk, T., Zümrelere göre örneklemede zümre sınırlarının ve örnek büyüklüklerinin genetik algoritma kullanılarak belirlenmesi, *Yöneylem Araştırması/ Endüstri Mühendisliği-XXVI. Ulusal Kongresi*, 3-5 Temmuz, Kocaeli, Türkiye, **2006**,
- [58] Er, Ş., Keskindürk, T., A genetic algorithm approach to determine stratum boundaries and sample sizes of each stratum in stratified sampling, *Computational Statistics & Data Analysis*, 52, 53-67, **2007**.
- [59] Er, Ş., Comparison of the efficiency of the various algorithms in stratified sampling when the initial solutions are determined with geometric method, *International Journal of Statistics and Applications*, 2(1), 1-10, **2012**.
- [60] Gunning, P., Horgan, J.M., A new algorithm for the construction of stratum boundaries in skewed populations, *Survey Methodology*, 30(2), 159-166, **2004**.
- [61] Hess, I., Sethi, V.K., Balakrishnan, T.R., Stratification: a practical investigation, *Journal of the American Statistical Association*, 61(313), 74-90, **1966**.
- [62] İstanbul Sanayi Odası, <http://www.iso.org.tr/projeler/arastirmalar/turkiyenin-500-buyuk-sanayi-kurulusu/> (Mart, 2014)

## ÖZGEÇMİŞ

### Kimlik Bilgileri

Adı Soyadı : Derya Turfan  
Doğum Yeri : Samsun  
Medeni Hali : Bekar  
E-posta : deryaturfan@hacettepe.edu.tr  
Adres : Hacettepe Üniversitesi İstatistik Bölümü / Ankara

### Eğitim

Lise : 2001-2005 Samsun Cumhuriyet Lisesi (YDA)  
Lisans : 2006-2011 Hacettepe Üniversitesi İstatistik Bölümü  
Yüksek Lisans : 2011-2014 Hacettepe Üniversitesi İstatistik Bölümü

### Yabancı Dil ve Düzeyi

İngilizce, İyi

### İş Deneyimi

2011-... Hacettepe Üniversitesi İstatistik Bölümü Araştırma Görevlisi

### Deneyim Alanları

### Tezden Üretilmiş Projeler ve Bütçesi

### Tezden Üretilmiş Yayınlar

### Tezden Üretilmiş Tebliğ ve/veya Poster Sunumu ile Katıldığı Toplantılar