

**ÖĐRENCİ BAŐARISINA ODAKLI SINAV ÇİZELGELEME  
MODELİ VE YAZILIM UYGULAMASI**

**Pamukkale Üniversitesi  
Fen Bilimleri Enstitüsü  
Yüksek Lisans Tezi  
Endüstri MühendisliĐi Anabilim Dalı**

---

**Can Berk KALAYCI**

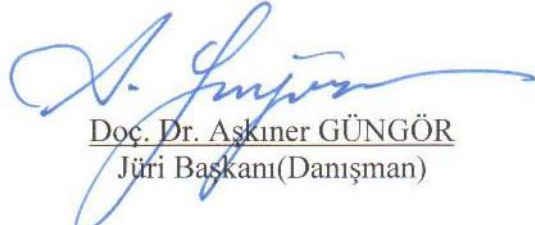
**Danışman: Doç. Dr. Aőkıner GÜNGÖR**

**Mayıs, 2008**


**DENİZLİ**

## YÜKSEK LİSANS TEZİ ONAY FORMU

Can Berk KALAYCI tarafından Doç. Dr. Aşkİner GÜNGÖR yönetiminde hazırlanan “**Öğrenci Başarısına Odaklı Sınav Çizelgeleme Modeli ve Yazılım Uygulaması**” başlıklı tez tarafımızdan okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans Tezi olarak kabul edilmiştir.



Doç. Dr. Aşkİner GÜNGÖR  
Jüri Başkanı(Danışman)



Doç. Dr. Osman KULAK  
Jüri Üyesi



Doç. Dr. Halim CEYLAN  
Jüri Üyesi

Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ...../...../..... tarih ve.....sayılı kararıyla onaylanmıştır.

**Prof. Dr. Mehmet Ali SARIGÖL**  
**Müdür**

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu çalıřmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan çalıřmalara atfedildiđini beyan ederim.

İmza :  
Öğrenci Adı Soyadı : Can Berk KALAYCI

## TEŞEKKÜR

Endüstri mühendisliğinin disiplinler arası vizyonununu ısrarla anlatarak beni bölümümüz akademik kadrosuna dâhil olmaya ikna eden, tez çalışmam süresince bana özgür bir çalışma takvimi sunarak hep bir yol gösterici olan, doktora eğitimi araştırmalarım nedeniyle tezime yoğunlaşmadığım dönemlerde gerekli anlayışı fazlasıyla göstererek bu süreçte bana her konuda destek ve referans olan, akademik dünyanın kapılarını aralamaya çalışırken benim için her zaman ilk başvuru ve destek ünitesi olan, titizliği ve nezaketini örnek almaya çalıştığım, kendisinden çok şey öğrendiğim ve her zaman da öğrenmeye devam edeceğime inandığım değerli danışman hocam Doç. Dr. Aşkîner Güngör'e,

Tez çalışmamdaki yoğun dönemde, genetik algoritmalar konusundaki bilgi ve deneyimi ile çalışmalarımda yön gösteren, resmi danışmanım olmamasına rağmen gerektiğinde benim için değerli vaktini hiç çekinmeden ayıran, doktora eğitimi arayışlarımda samimiyetle destek ve referans olan, aynı zamanda gençliği ve samimiyetiyle bir arkadaş olan, bitmek tükenmek bilmeyen enerjisine hayran olduğum değerli hocam Doç. Dr. Osman Kulak'a,

Bölümümüz akademik kadrosuna dâhil olmamı teşvik ederek bana destek olan, bana her zaman güvenerek referans olan, her zaman en yüksek anlayışı gösteren, ciddi ve sert duruşunun altındaki sevecen kişiliği ve tatlı sohbetiyle örnek insan olan değerli hocam Yrd. Doç. Dr. Özcan Mutlu'ya,

Endüstri Mühendisliği ile tanışmamda büyük pay sahibi olan, tez çalışmamdaki deneysel tasarım çalışmaları sırasında bana ihtiyacım olan her türlü teknik donanımı sağlayan, ihtiyaç duyduğum her anda yakınlık ve ilgi göstererek hiçbir zaman bir yabancı gibi hissetmememi sağlayan, çok konuda yardımcı olan, bana her zaman güvenerek referans olan değerli hocam Yrd. Doç. Dr. Abdülkadir Yaldır'a,

Yardıma ihtiyacım olduğu her zaman manevi desteğini benden esirgemeyen, yaptığım esprilere anlayış ve yansıma göstererek sıcak karakteriyle çalışma ortamımıza renk katan sevgili çalışma arkadaşım Yusuf Şahin'e,

Tezimde kapsam dışına çıkardığımız bölümdeki sanal sınıf önerisiyle bilimde çığır açan :p, bana kefil olmakla büyük bir hata yaptığını ilerde anlayacak olan ☺, birlikte bir teşekkürle sığdıramayacak kadar çok şey paylaştığımız ve paylaşacağımız (kolay kolay kurtulamayacak olan!) sevgili oda arkadaşım, dostum Olcay Polat'a,

Bu tezin gerçekleştirilmesinde 2007FBE014 numaralı proje ile destek sağlayan Pamukkale Üniversitesi Bilimsel Araştırma Projeleri Birimine,

Ailemiz için dünden bugüne hep çalışan, maddi manevi her konuda destek olarak beni bugünlere taşıyan, zaman zaman yüksek disiplin katsayısına tahamül edememiş olsam da değerini her dakika anladığım, her geçen gün birbirimizi daha iyi anladığımız, sağlığına daha çok dikkat ederek daha mutlu günler geçirmemizi sağlayacak olan canım babam Mustafa Kalaycı'ya,

Ailemiz için büyük bir gurur kaynağı olan, çok çabuk olgunlaşarak daha az yardıma ihtiyaç duysa da her zaman yanında olacağım canım kardeşim Ayberk Kalaycı'ya,

“Bir anne daha ne yapabilir!” dedirten, çocuklarının mutluluğu için yaşayan, asıl mutluluk kaynağı olan canım annem Öznur Kalaycı'ya,

Son olarak; hayatın üzerimdeki hakkını ödemeye fırsat vermediği, Alzheimer hastalığına yenik düşen, fotoğraf kareleriyle tesellisini bulduğum canım babaannem Ayşe Kalaycı'ya sonsuz teşekkürlerimi sunarım.

**Can Berk KALAYCI**

## ÖZET

### ÖĞRENCİ BAŞARISINA ODAKLI SINAV ÇİZELGELEME MODELİ VE YAZILIM UYGULAMASI

Kalaycı, Can Berk  
Yüksek Lisans Tezi, Endüstri Mühendisliği ABD  
Tez Yöneticisi: Doç. Dr. Aşkıner GÜNGÖR

Mayıs 2008, 144 Sayfa

Üniversite birimlerinde, eğitim öğretim faaliyetleri kapsamında yer alan sınavların çizelgelenmesi önemli ve zor bir görevdir. Sınavların öğrenme sürecinin bir parçası olması nedeniyle, sınavların çakışmadan ve belirlenen sınav döneminde yapılabilmesi öğrenciler açısından yeterli değildir. Sınavların çizelgelenmesi çalışmalarında daha fazla unsura dikkat etmek gerekir. Bu çalışmada, sınav çizelgelemenin öğrencilerin başarısını açığa çıkaracak şekilde düzenlenmesi gerektiği fikri üzerine odaklanılmıştır. Öğrencilerin sınav döneminde başarılı olması, sınavları arasında yeterli hazırlanma ve dinlenme zamanı olmasıyla mutlak bağlantılıdır. Öğrencilerin sınavları arasında ihtiyaç duydukları hazırlanma ve dinlenme zamanı, sınavların zorluk dereceleri ile doğru orantılı olarak artmaktadır. Bu nedenle sınavların sınav dönemine yayılımı gerçekleştirilirken derslerin zorluk derecelerinin de değerlendirilmesi gerekmektedir. Bu çalışmada sınavların çizelgelenmesi için, zorunlu kısıtların sağlanması ön koşulu ile zor olan sınavların birbirinden daha uzak zaman dilimlerine atanmasıyla en uygun sınav dönemi yayılımını sağlamaya yönelik iki farklı genetik algoritma modeli geliştirilmiştir. Bu şekilde, öğrencilere sıradaki sınava en uygun şekilde hazırlanabilmeleri için mümkün olan en uzun hazırlanma ve dinlenme süresini tanıyarak öğrenci başarısının artırılacağı düşünülmektedir. Bir minimizasyon problemi haline getirilen bu sınav çizelgeleme problemini çözmek için geliştirilen ilk genetik algoritma modelinde; zorunlu kısıtların sağlanabilmesi için, istenmeyen çözümler içeren kromozomların uygunluk değerine yüksek bir ceza puanı eklenerek seçim olasılıklarının düşürülmesi ve toplumdaki zamanla silinmesi amaçlanmıştır. Geliştirilen ikinci genetik algoritma modelinde ise; topluma yeni katılan her bireyin zorunlu kısıtları sağlayıp sağlamadığı kontrol edilmektedir. Eğer zorunlu kısıtları sağlamıyorsa bir tamir fonksiyonu aracılığı ile çakışmalar ortadan kaldırılmakta, kaldırılmadığında ise kromozom yok edilmektedir. Bu yöntemle, geliştirilen genetik algoritma sadece kabul edilebilir bireylerden oluşan bir toplumda arama yapabilmektedir. Farklı parametre setleri ile yapılan deneyler sonucunda tamir fonksiyonu modelinin daha yavaş olmasına rağmen daha yüksek performans sergilediği görülmüştür.

**Anahtar kelimeler:** Genetik Algoritma, Sınav Çizelgeleme, Öğrenci Başarısı

Doç. Dr. Aşkıner GÜNGÖR  
Doç. Dr. Osman KULAK  
Doç. Dr. Halim CEYLAN

**ABSTRACT****EXAMINATION SCHEDULING MODEL AND ITS SOFTWARE  
IMPLEMENTATION FOCUSING ON STUDENTS' SUCCESS**

Kalaycı, Can Berk  
M.Sc. Thesis in Industrial Engineering  
Supervisor: Assoc. Prof. Dr. Aşkiner GÜNGÖR

May 2008, 144 Pages

As it is known, exam scheduling is a difficult and lengthy task for which universities devote a large amount of human and material resources every year. Scheduling exams without any conflicts in the examination period is not enough for students because it is accepted that exams are the part of learning process. It is required to pay attention to much more components during timetabling in order to reach a “good” schedule. In this study, we focus on students’ success when scheduling exams. Student success is positively related to the adequate preparation and resting time among examinations. The amount of study time that students need among examinations (i.e. paper spread) is directly proportional to the difficulty of exams. Therefore, the main objective of this study is to maximize paper spread.

In this study, two different genetic algorithm models are developed to optimize paper spread under several hard constraints and the difficulties of exams. When a student finishes a difficult examination at a time period, s/he shall have enough time to prepare him(her)self for the other difficult one. In the first genetic algorithm model, a high penalty value is added to the fitness value of the chromosome which generates an infeasible solution. Thus, the selection probabilities of this type of chromosomes for the next generation are decreased. The second genetic algorithm model controls whether each chromosome joining the population satisfies the hard constraints or not. If the hard constraints are not satisfied, conflicts are removed from the chromosome with an embedded repair function. If repair fails, the chromosome is destroyed. This method forces the genetic algorithm to search only in the feasible solution space. A set of experiments have been designed and studied. The results showed that the performance of the repair function model was better than the high penalty value model.

**Keywords: Genetic Algorithms, Examination Scheduling, Timetabling, Student Success**

Assoc. Prof. Dr. Aşkiner GÜNGÖR  
Assoc. Prof. Dr. Osman KULAK  
Assoc. Prof. Dr. Halim CEYLAN

## İÇİNDEKİLER

	Sayfa
YÜKSEK LİSANS TEZİ ONAY FORMU .....	i
BİLİMSEL ETİK SAYFASI .....	ii
TEŞEKKÜR.....	iii
ÖZET .....	iv
ABSTRACT.....	v
İÇİNDEKİLER .....	vi
ŞEKİLLER DİZİNİ.....	viii
TABLolar DİZİNİ .....	x
SİMGE VE KISALTMALAR DİZİNİ .....	xii
1. GİRİŞ .....	1
2. ÜNİVERSİTE SINAV ÇİZELGELEME PROBLEMİ .....	4
2.1 Tanım .....	4
2.2 Çözüm Yöntemleri ve Literatür Taraması .....	5
2.2.1 Yöneylem araştırması yaklaşımları.....	6
2.2.1.1 Sıralı (Enumerative) arama yöntemleri.....	6
2.2.1.2 Sezgisel yöntemler .....	8
2.2.2 Yapay zekâ yaklaşımları .....	12
3. GENETİK ALGORİTMALAR .....	15
3.1 Genetik Algoritmalara Giriş.....	15
3.2 Genetik Algoritma Nedir?.....	15
3.3 Genetik Algoritmalar Nasıl Çalışır? .....	17
3.4 Genetik Algoritmaların Ayrıntılı Adımları .....	17
3.4.1 Başlatma.....	17
3.4.2 Yeniden üretim.....	17
3.4.2.1 Nesilsel yeniden üretim.....	17
3.4.2.2 Kararlı hal üretimi .....	18
3.4.3 Seçim.....	18
3.4.3.1 Uygunluk değeri tabanlı seçim (Rulet tekeri seçimi).....	18
3.4.3.2 Sıralama tabanlı seçim .....	18
3.4.3.3 Turnuva tabanlı seçim .....	18
3.4.3.4 Uzaya dayalı seçim .....	19
3.5 Genetik Operatörler.....	19
3.5.1 Çaprazlama operatörü .....	19
3.5.2 Tersine çevirme operatörü .....	22
3.5.3 Mutasyon operatörü .....	22
3.5.4 Göç operatörü.....	24
4. ÖĞRENCİ BAŞARISINA ODAKLI SINAV ÇİZELGELEME MODELİ .....	26
4.1 Temsil.....	26
4.2 İlk Nüfusun Oluşturulması.....	29
4.3. Uygunluk.....	31
4.4 Kromozomların Eşleştirme Havuzuna Alınması .....	37
4.5 Çaprazlanacak Bireylerin Seçilmesi .....	39
4.6 Çaprazlama.....	41
4.7 Mutasyon.....	45
4.8 En iyi kromozomların seçimi .....	49
4.9 Entegre Tamir Fonksiyonu.....	51
5. DENEYSEL ÇALIŞMALAR .....	62

6. SONUÇ VE ÖNERİLER.....	116
KAYNAKLAR .....	118
EKLER.....	123
ÖZGEÇMİŞ .....	132



## ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1 Tek noktali çaprazlama işlemi .....	20
Şekil 3.2 İki noktali çaprazlama işlemi .....	20
Şekil 3.3 Çok noktali çaprazlama işlemi.....	21
Şekil 3.4 Uniform çaprazlama işlemi.....	22
Şekil 3.5 Tersine çevirme işlemi.....	22
Şekil 3.6 Bit çevirme operatörü .....	23
Şekil 3.7 Değiş tokuş operatörü .....	24
Şekil 3.8 Yerine koyma operatörü .....	24
Şekil 4.1 Genetik Algoritmada sınav çizelgeleme problemine aday çözümün kromozom olarak temsili.....	28
Şekil 4.2 $N$ sayıda gün ve her günde $T$ sayıda zaman dilimi için sembolik gösterim.....	28
Şekil 4.3 Kromozomdaki genlerin çizelge üzerindeki yayılımı.....	28
Şekil 4.4 Şekil 4.1'deki genlerin çizelge üzerindeki anlamı.....	29
Şekil 4.5 Bir genin çizelge üzerindeki uygun yerleşimin belirlenmesi.....	29
Şekil 4.6 Bir zincirdeki genlerin değerlerinin çakışması .....	31
Şekil 4.7 Bir sınıfa ait tüm sınavların yerleşim dizisi .....	32
Şekil 4.8 Sınavların zaman dilimi ve zorluk derecesi gösterimi.....	35
Şekil 4.9 Yakınlık katsayıları gösterimi.....	36
Şekil 4.10 Kromozomların eşleştirme havuzuna alınması.....	38
Şekil 4.11 Rulet tekeri seçimi .....	40
Şekil 4.12 Normal uniform çaprazlama sonucu oluşan çakışmalar .....	42
Şekil 4.13 Uyarlanmış uniform çaprazlama ile 1. çocuk oluşumu .....	43
Şekil 4.14 Uyarlanmış uniform çaprazlama ile 2. çocuk oluşumu .....	44
Şekil 4.15 Herhangi bir zincirdeki mutasyon işlemi.....	46
Şekil 4.16 Yeni nesil seçimi.....	49
Şekil 4.17 Ortak sınav kontrolü ve tamir .....	53
Şekil 4.18 Üst zincirlerin taranması ve kromozom tamiri .....	54
Şekil 4.19 Alt zincirlerin taranması ve kromozom tamiri.....	55
Şekil 4.20 Geri besleme dizisi.....	56
Şekil 5.1 1. parametre seti deney sonuçları.....	93
Şekil 5.2 2. parametre seti deney sonuçları.....	94
Şekil 5.3 3. parametre seti deney sonuçları.....	94
Şekil 5.4 4. parametre seti deney sonuçları.....	95
Şekil 5.5 5. parametre seti deney sonuçları.....	95
Şekil 5.6 6. parametre seti deney sonuçları.....	96
Şekil 5.7 7. parametre seti deney sonuçları.....	96
Şekil 5.8 8. parametre seti deney sonuçları.....	97
Şekil 5.9 9. parametre seti deney sonuçları.....	97
Şekil 5.10 10. parametre seti deney sonuçları.....	98
Şekil 5.11 11. parametre seti deney sonuçları.....	98
Şekil 5.12 12. parametre seti deney sonuçları.....	99
Şekil 5.13 13. parametre seti deney sonuçları.....	99
Şekil 5.14 14. parametre seti deney sonuçları.....	100
Şekil 5.15 15. parametre seti deney sonuçları.....	100
Şekil 5.16 16. parametre seti deney sonuçları.....	101
Şekil 5.17 17. parametre seti deney sonuçları.....	101
Şekil 5.18 18. parametre seti deney sonuçları.....	102
Şekil 5.19 19. parametre seti deney sonuçları.....	102

Şekil 5.20	20. parametre seti deney sonuçları.....	103
Şekil 5.21	21. parametre seti deney sonuçları.....	103
Şekil 5.22	22. parametre seti deney sonuçları.....	104
Şekil 5.23	23. parametre seti deney sonuçları.....	104
Şekil 5.24	24. parametre seti deney sonuçları.....	105
Şekil 5.25	25. parametre seti deney sonuçları.....	105
Şekil 5.26	26. parametre seti deney sonuçları.....	106
Şekil 5.27	27. parametre seti deney sonuçları.....	106
Şekil 5.28	Tamir fonksiyonu modeli ile yeni nesil seçimi.....	113

## TABLOLAR DİZİNİ

	Sayfa
Tablo 4.1 İlk popülasyonun yaratılması fonksiyonu.....	30
Tablo 4.2 Rastgele kromozom oluşturulması fonksiyonu .....	31
Tablo 4.3 Ortak sınavlar zincirine bağlı olan zincirler matrisi .....	33
Tablo 4.4 Ortak sınavlar zincirine bağlı olan tüm zincirler matrisi .....	34
Tablo 4.5 Uygunluğu hesaplanacak dizinin belirlenmesi .....	35
Tablo 4.6 Uygunluk değerinin hesaplanması.....	37
Tablo 4.7 Kümülatif toplamın bulunması .....	37
Tablo 4.8 Kromozomların eşleştirme havuzuna alınması fonksiyonu.....	39
Tablo 4.9 Çaprazlanacak bireylerin seçilmesi fonksiyonu .....	41
Tablo 4.10 Çaprazlama işlemi fonksiyonu .....	44
Tablo 4.11 Her kromozomda tek geni mutasyona uğratan fonksiyon .....	47
Tablo 4.12 Her kromozomda iki geni mutasyona uğratan fonksiyon.....	48
Tablo 4.13 Her kromozomda dört geni mutasyona uğratan fonksiyon.....	48
Tablo 4.14 En iyi kromozomların seçimi fonksiyonu.....	50
Tablo 4.15 Sınav bilgi formu .....	52
Tablo 4.16 Tamir fonksiyonu.....	56
Tablo 4.17 Uygun olmayan zaman dilimlerini silen fonksiyon .....	61
Tablo 5.1 Modelde kullanılan parametreler ve seviyeleri.....	62
Tablo 5.2 Taguchi L27 .....	64
Tablo 5.3 İndirgenmiş L27.....	64
Tablo 5.4 Sahte işleyiş (Dummy treatment) uygulanması sonrası indirgenmiş L27 .....	65
Tablo 5.5 Parametre setleri tablosu .....	66
Tablo 5.6 Yüksek ceza katsayısı modeli ile 1. parametre seti deney sonuçları .....	67
Tablo 5.7 Tamir fonksiyonu modeli ile 1. parametre seti deney sonuçları.....	67
Tablo 5.8 Yüksek ceza katsayısı modeli ile 2. parametre seti deney sonuçları .....	68
Tablo 5.9 Tamir fonksiyonu modeli ile 2. parametre seti deney sonuçları.....	68
Tablo 5.10 Yüksek ceza katsayısı modeli ile 3. parametre seti deney sonuçları .....	69
Tablo 5.11 Tamir fonksiyonu modeli ile 3. parametre seti deney sonuçları.....	69
Tablo 5.12 Yüksek ceza katsayısı modeli ile 4. parametre seti deney sonuçları .....	70
Tablo 5.13 Tamir fonksiyonu modeli ile 4. parametre seti deney sonuçları.....	70
Tablo 5.14 Yüksek ceza katsayısı modeli ile 5. parametre seti deney sonuçları .....	71
Tablo 5.15 Tamir fonksiyonu modeli ile 5. parametre seti deney sonuçları.....	71
Tablo 5.16 Yüksek ceza katsayısı modeli ile 6. parametre seti deney sonuçları .....	72
Tablo 5.17 Tamir fonksiyonu modeli ile 6. parametre seti deney sonuçları.....	72
Tablo 5.18 Yüksek ceza katsayısı modeli ile 7. parametre seti deney sonuçları .....	73
Tablo 5.19 Tamir fonksiyonu modeli ile 7. parametre seti deney sonuçları.....	73
Tablo 5.20 Yüksek ceza katsayısı modeli ile 8. parametre seti deney sonuçları .....	74
Tablo 5.21 Tamir fonksiyonu modeli ile 8. parametre seti deney sonuçları.....	74
Tablo 5.22 Yüksek ceza katsayısı modeli ile 9. parametre seti deney sonuçları .....	74
Tablo 5.23 Tamir fonksiyonu modeli ile 9. parametre seti deney sonuçları.....	75
Tablo 5.24 Yüksek ceza katsayısı modeli ile 10. parametre seti deney sonuçları .....	75
Tablo 5.25 Tamir fonksiyonu modeli ile 10. parametre seti deney sonuçları.....	76
Tablo 5.26 Yüksek ceza katsayısı modeli ile 11. parametre seti deney sonuçları .....	76
Tablo 5.27 Tamir fonksiyonu modeli ile 11. parametre seti deney sonuçları.....	77
Tablo 5.28 Yüksek ceza katsayısı modeli ile 12. parametre seti deney sonuçları .....	77
Tablo 5.29 Tamir fonksiyonu modeli ile 12. parametre seti deney sonuçları.....	78
Tablo 5.30 Yüksek ceza katsayısı modeli ile 13. parametre seti deney sonuçları .....	78
Tablo 5.31 Tamir fonksiyonu modeli ile 13. parametre seti deney sonuçları.....	79

Tablo 5.32 Yüksek ceza katsayısı modeli ile 14. parametre seti deney sonuçları .....	79
Tablo 5.33 Tamir fonksiyonu modeli ile 14. parametre seti deney sonuçları.....	80
Tablo 5.34 Yüksek ceza katsayısı modeli ile 15. parametre seti deney sonuçları .....	80
Tablo 5.35 Tamir fonksiyonu modeli ile 15. parametre seti deney sonuçları.....	81
Tablo 5.36 Yüksek ceza katsayısı modeli ile 16. parametre seti deney sonuçları .....	81
Tablo 5.37 Tamir fonksiyonu modeli ile 16. parametre seti deney sonuçları.....	82
Tablo 5.38 Yüksek ceza katsayısı modeli ile 17. parametre seti deney sonuçları .....	82
Tablo 5.39 Tamir fonksiyonu modeli ile 17. parametre seti deney sonuçları.....	83
Tablo 5.40 Yüksek ceza katsayısı modeli ile 18. parametre seti deney sonuçları .....	83
Tablo 5.41 Tamir fonksiyonu modeli ile 18. parametre seti deney sonuçları.....	84
Tablo 5.42 Yüksek ceza katsayısı modeli ile 19. parametre seti deney sonuçları .....	84
Tablo 5.43 Tamir fonksiyonu modeli ile 19. parametre seti deney sonuçları.....	85
Tablo 5.44 Yüksek ceza katsayısı modeli ile 20. parametre seti deney sonuçları .....	85
Tablo 5.45 Tamir fonksiyonu modeli ile 20. parametre seti deney sonuçları.....	86
Tablo 5.46 Yüksek ceza katsayısı modeli ile 21. parametre seti deney sonuçları .....	86
Tablo 5.47 Tamir fonksiyonu modeli ile 21. parametre seti deney sonuçları.....	87
Tablo 5.48 Yüksek ceza katsayısı modeli ile 22. parametre seti deney sonuçları .....	87
Tablo 5.49 Tamir fonksiyonu modeli ile 22. parametre seti deney sonuçları.....	88
Tablo 5.50 Yüksek ceza katsayısı modeli ile 23. parametre seti deney sonuçları .....	88
Tablo 5.51 Tamir fonksiyonu modeli ile 23. parametre seti deney sonuçları.....	89
Tablo 5.52 Yüksek ceza katsayısı modeli ile 24. parametre seti deney sonuçları .....	89
Tablo 5.53 Tamir fonksiyonu modeli ile 24. parametre seti deney sonuçları.....	90
Tablo 5.54 Yüksek ceza katsayısı modeli ile 25. parametre seti deney sonuçları .....	90
Tablo 5.55 Tamir fonksiyonu modeli ile 25. parametre seti deney sonuçları.....	91
Tablo 5.56 Yüksek ceza katsayısı modeli ile 26. parametre seti deney sonuçları .....	91
Tablo 5.57 Tamir fonksiyonu modeli ile 26. parametre seti deney sonuçları.....	92
Tablo 5.58 Yüksek ceza katsayısı modeli ile 27. parametre seti deney sonuçları .....	92
Tablo 5.59 Tamir fonksiyonu modeli ile 27. parametre seti deney sonuçları.....	93
Tablo 5.60 Yüksek ceza katsayısı modeli ile parametre setleri performansları sıralaması	108
Tablo 5.61 Tamir fonksiyonu modeli ile parametre setleri performansları sıralaması .....	109
Tablo 5.62 Yüksek ceza katsayısı modeli ile tamir fonksiyonu modelinin ürettikleri uygunluk değerlerine göre karşılaştırılması .....	110
Tablo 5.63 Genetik algoritmanın genel akış kodlaması.....	112
Tablo 5.64 10. parametre setine göre GA değişkenlerinin aldığı değerler .....	114
Tablo 5.65 Yüksek ceza katsayısı modeli ile tamir fonksiyonu modelinin sözel olarak karşılaştırılması .....	114
Tablo 5.66 Zincirlere göre sınavların yerleştiği zaman dilimleri.....	115

## SİMGE VE KISALTMALAR DİZİNİ

<i>AL</i>	: Uygunluk değeri hesaplanacak dizinin uzunluğu
<i>bps</i>	: Başlangıç popülasyonu sayısı
<i>c</i>	: Standart sapma katsayısı
<i>cbsf</i>	: Çaprazlanacak bireylerin seçilmesi fonksiyonundaki toplam işlem sayısı
<i>ciyf</i>	: Çaprazlama işlemi yapan fonksiyondaki toplam işlem sayısı
<i>co</i>	: Çaprazlama oranı
<i>DF</i>	: Zorluk derecesi katsayısı
<i>ebzd</i>	: En büyük zaman dilimi
<i>eiksf</i>	: En iyi kromozomların seçimi fonksiyonundaki toplam işlem sayısı
<i>eo</i>	: Elit oranı
<i>es</i>	: Elit sayısı
<i>GA</i>	: Genetik Algoritma
<i>gbdu</i>	: Geri besleme dizisi uzunluğu
<i>ics</i>	: İstenen çocuk sayısı
<i>ims</i>	: İstenen mutant sayısı
<i>ipyf</i>	: İlk popülasyonun yaratılması fonksiyonundaki toplam işlem sayısı
<i>kehaf</i>	: Kromozomların eşleştirme havuzuna alınması fonksiyonundaki toplam işlem sayısı
<i>ktbf</i>	: Kümülatif toplamın bulunması fonksiyonundaki toplam işlem sayısı
<i>ku</i>	: Kromozom uzunluğu
<i>m1</i>	: Her kromozomda tek geni mutasyona uğratan fonksiyonundaki toplam işlem sayısı
<i>m2</i>	: Her kromozomda iki geni mutasyona uğratan fonksiyondaki toplam işlem sayısı
<i>m4</i>	: Her kromozomda dört geni mutasyona uğratan fonksiyonundaki toplam işlem sayısı
<i>mo</i>	: Mutasyon oranı
<i>mt</i>	: Mutasyon taktiği
<i>N</i>	: Sınav dönemindeki toplam gün sayısı
<i>ns</i>	: Nesil sayısı,
<i>oz1</i>	: Ortak zincirin satır sayısı
<i>oz2</i>	: Ortak zincirin sütun sayısı
<i>ps</i>	: Popülasyon sayısı
<i>rkof</i>	: Rastgele kromozom oluşturulması fonksiyonundaki toplam işlem sayısı
<i>S</i>	: Standart sapma
<i>sdu</i>	: Şüpheli dizi uzunluğu
<i>t</i>	: Sınavın atandığı zaman dilimi
<i>tf</i>	: Tamir fonksiyonu için gereken toplam işlem sayısı
<i>TN</i>	: Zaman dilimleri yakınlık katsayısı
<i>TS</i>	: Atanabilecek toplam zaman dilimi sayısı
<i>udhf</i>	: Uygunluk değerinin hesaplanması fonksiyonundaki toplam işlem sayısı
<i>udu</i>	: Uygunluk dizisi uzunluğu
<i>uhdbf</i>	: Uygunluğu hesaplanacak dizinin belirlenmesi fonksiyonundaki toplam işlem sayısı
<i>uozdsf</i>	: Uygun olmayan zaman dilimlerini silen fonksiyondaki toplam işlem sayısı
<i>zs</i>	: Zincir sayısı
<i>zu</i>	: Zincir uzunluğu

## 1. GİRİŞ

Üniversite birimlerinde, eğitim öğretim faaliyetleri kapsamında yer alan sınavların çizelgelenmesi önemli ve zor bir faaliyettir. Bu amaçla oldukça fazla zaman ve emek harcanmasına rağmen, elle yapılan çizelgelerin büyük bir bölümü tatmin edici sonuçlar ortaya koyamamaktadır. Genelde elde edilen çizelge için başarı ölçütü olarak sınavların sorunsuz yapılabilmesi düşüncesi kullanılır. Sınavların çakışmadan ve belirlenen sınav döneminde yapılabilmesinin, yürütülen sınav çizelgeleme çalışmasını başarılı kıldığı kabul edilebilir.

Sınavların öğrenme sürecinin bir parçası olduğu herkes tarafından kabul edilmektedir. Sınavların bu işlevi yerine getirebilmesi için, sınavların çizelgelenmesi çalışmalarında daha fazla unsura dikkat etmek gerekir. Bu çalışmada, sınav çizelgelemenin öğrencilerin başarısını açığa çıkaracak şekilde düzenlenmesi gerektiği fikri üzerine odaklanılmıştır. Üniversitelerde, uygun sınav programlarının hazırlanması, eğitim dönemleri boyunca karşılaşılan önemli ve sürekli problemlerden biridir. Öğrenci sayısının gün geçtikçe artması, alttan ve üstten ders alımlarının varlığı, fakülte bazında genel kısıtlar ve bölümden bölüme değişen özel kısıtların bulunması, problemin büyüklüğünde belirleyicidir. Kısıt sayısının fazla olmasının yanında, derslerin zorluk derecelerinin değerlendirilmesi de dikkate alındığında problemin karmaşıklığı daha da artmaktadır. Genetik algoritmalarla, çizelgeleme problemlerine benzer çok kısıtlı ve büyük çözüm uzaylarına sahip problemlerin çözümünde etkin sonuçlar üretilmiştir. Bu çalışmada, üniversite çizelgeleme probleminin çözümünde farklı parametreleri ve farklı arama modellerini uygulayarak genetik algoritmaların etkinliğinin araştırılması düşünülmüştür. Genetik algoritmaların farklı modelleme seçenekleriyle ortaya çıkaracağına inanılan farklı sonuçlara duyulan merak nedeni ile bu problemin çözümünde genetik algoritmalar tercih edilmiştir.

Genetik algoritmaların üniversite çizelgeleme problemine uygulanabilmesi için uygun kromozom yapısının geliştirilmesi, problemin çözümünde en önemli kısımlardan birisini oluşturmaktadır. Bu yapının basit olması genetik algoritmaların performansı

üzerinde doğrudan etkilidir. Genetik algoritmaların performansında, kromozom yapısı, farklı parametreler ve algoritmanın yapısal modelinin yanında, hedeflenen sonuca göre uygun kısıtların seçilmesi de mutlak etkilidir. Kısıt sayısının çok fazla olması çözüm uzayını daraltacağı için, oluşan çözümlerde performans beklemek bir yana, kabul edilebilir çözümlere ulaşılması da zorlaşmaktadır. Bu yüzden, bazı kısıtların problemin kapsamı dışında tutulması kaçınılmazdır.

Bu araştırmada, üniversite çizelgeleme probleminin en kritik bölümü olan sınavların yapılacağı zaman dilimlerinin belirlenmesi, genetik algoritmalar ile farklı parametre ve modellerin sonuçların karşılaştırılması problemin kapsamı içine alınırken; zaman dilimlerine atanmış olan sınavların salonlara yerleştirilmesi, salon kısıtlarının üniversiteden üniversiteye değişmesi (evrensel olmaması) ve kritik olmaması nedeniyle problemin kapsamı dışında tutulmuştur.

Bilinen bir problem olan üniversite çizelgeleme probleminin çözümü için literatürde üretilmiş çok sayıda yöntem bulunmaktadır. Bu çalışmada literatürden farklı olarak, genetik algoritmaları yönlendiren amaç fonksiyonuna sınavların zorluk dereceleri de eklenerek, zor olan derslerin birbirinden daha uzak zaman dilimlerine atanmasını sağlamaya yönelik bir arama yapılmaktadır. Burada amaç, öğrencilere zor olan sınavları arasında, sıradaki sınava en uygun şekilde hazırlanabilmeleri için mümkün olan en uzun hazırlanma ve dinlenme süresini tanımadır. Sınav döneminde öğrenci başarısının bu şekilde artırılacağı düşünülmektedir. Ayrıca, geliştirilen farklı iki genetik algoritma modelinin karşılaştırmalı sonuçlarıyla literatüre katkı sağladığı düşünülmektedir. Modelin anlatım sürecinde algoritma karmaşıklıklarına da değinilerek daha zengin karşılaştırmalı bir inceleme sağlanmıştır.

Geliştirilen algoritmanın uygulanması ve deneysel simülasyon sonuçlarının elde edilmesi çalışmalarında, algoritma geliştirmeye ve simülasyona uygun olan MATLAB ortamı tercih edilmiştir. Geliştirilen özgün algoritma ve karşılaştırmalı deney sonuçları öncelikli hedef olarak ele alındığı için veri tabanı modellemesi yerine dosya giriş-çıkış işlemleri tercih edilmiştir. Kullanıcı dostu bir yazılım elde etmeyi hedeflemeyen bu uygulama için herhangi bir arayüz çalışması da yapılmamıştır. Değişkenlerin ve fonksiyonların parametrik hale getirilmesiyle istenen değişiklikler rahatlıkla uygulanabilmektedir. Zaten, farklı parametrelerin simülasyon sonuçlarının alınabilmesi gerekliliği parametrik tasarımı zorunlu kılmaktadır.

Gerçekleştirilen çalışma ile ilgili genel bilgiler verilen bu bölümü, ikinci bölümde üniversite çizelgeleme probleminin tanımı, incelenen çözüm yöntemleri ve literatür taraması izlemektedir. Üçüncü bölümde genetik algoritmalar hakkında genel bilgiler verilmiştir. Dördüncü bölümde geliştirilen sınav çizelgeleme modeli anlatılmış ve izleyen beşinci bölümde geliştirilen modelin deneysel sonuçlarına yer verilmiştir. Son bölümde ise, elde edilen sonuçlar ve öneriler yer almaktadır.



## 2. ÜNİVERSİTE SINAV ÇİZELGELEME PROBLEMİ

Bu bölümde, yaygın bir problem olan üniversite çizelgeleme problemi tanıtılarak problemin literatürde bulunan evrensel kısıtlarına ve bu çalışmada ele alınan kısıtlara değinilmiştir.

### 2.1 Tanım

Bir çizelgeleme problemi, zorunlu olan birincil kısıtların (hard constraints) ve isteğe bağlı olan ikincil kısıtların (soft constraints) sağlanarak, gerçekleştirilecek olan görevler kümesinin bir kaynaklar kümesine atanmasıdır. Bir zaman çizelgeleme problemi olan sınav programı hazırlamada hedef, her ders için yapılacak sınavın zaman dilimlerinin istenen kısıtlar sağlanarak belirlenmesidir. Bu problemde, atanacak zaman dilimleri kaynakları, görevler ise ilgili zaman dilimlerinde yapılacak olan sınavları temsil etmektedir.

Üniversitelerdeki sınav çizelgeleme problemi, bir fakülte için bütün olarak ele alındığında, fakültedeki her bölümün kendine özgü kısıtları ve bölümlerin ortak kısıtları bir araya geldiğinde çoğalan kısıt sayısı ile çözülmesi daha zor olan bir problem haline dönüşmektedir. Sadece zorunlu kısıtlar ele alındığında problemin çözümü kolaylaşmakta, ancak tatmin edici sonuçlar ortaya çıkmamaktadır. Zorunlu kısıtların sağlanması ile sadece her öğrencinin girmek zorunda olduğu sınavlara girmesini sağlanmaktadır. Ancak bu durum beklentiyi karşılayamamaktadır. Öğrencilerin ardına sınavlara girmek zorunda kalması veya sınavların bazı öğrenciler için belli sınav günlerinde kümelenmesi sorunu ile karşılaşılabilmektedir. Bu sıkıntıların da ortadan kalkmasını sağlayan ikincil kısıtları kullanarak sonuçların daha kabul edilebilir olması sağlanabilmektedir.

Bir sınav çizelgeleme probleminde birincil (zorunlu) kısıtlar aşağıdaki gibi özetlenebilir:

- Bir öğrencinin girmek zorunda olduğu sınavların tümü farklı zaman dilimlerine atanmak zorundadır:

- Her öğrencinin, hangi sınavlara girmesi gerektiği ve hangi bölüm ve sınıfın öğrencisi olduğu önceden tanımlanmıştır.
- Bir sınıfa ait sınavların tümü farklı zaman dilimlerine atanmak zorundadır.
- Bir öğrencinin alttan veya üstten aldığı sınavlar ile normal dönemdeki sınavların tümü farklı zaman dilimlerine atanmak zorundadır.
- Bölümlerin ortak yapmak zorunda olduğu sınavlar, aynı zaman dilimine atanmak zorundadır.

Sınav çizelgeleme probleminde ikincil (isteğe bağlı) kısıtlardan bazıları aşağıda sunulmuştur:

- Öğrencilerin ardı ardına iki farklı sınava girmek zorunda kalmamaları için zaman dilimlerinin ayarlanması gerekmektedir (Burke vd 2007).
- Uygun sınav çizelgelerinin kalitesinin artırılabilmesi için sınavların sınav dönemindeki zaman dilimlerine maksimum uzaklıktaki yayılımını gereklidir (MirHassani 2006). Bu yayılım sayesinde, öğrencilere sınavlar arasında çalışabilecekleri maksimum sürenin tanınabilecektir. Öyleyse, bir öğrencinin girmesi gereken sınavlar, resmi olan sınav tarihleri içinde kalmak şartıyla, birbirine en uzak olan zaman dilimlerine atanmalıdır.
- Bu çalışmada, bir öğrencinin sınav döneminde, kendini sınava en iyi şekilde hazırlayabilmesi için gerekli çalışma süresinin tanınması gerekliliğine ek olarak; zor olan sınavların kolay olan sınavlara göre daha uzun bir çalışma süresi gerektirdiği düşünülmektedir. Öyleyse her öğrenciye, bir sınavdan çıktığında sıradaki sınavına hazırlanabilmesi için yeterli süre tanınmalı ve bir öğrencinin girmesi gereken sınavların zaman dilimlerine yayılımında, sınavların zorluk dereceleri de dikkate alınmalıdır. Zor olan sınavlar birbirinden daha uzak olan zaman dilimlerine atanmalıdır.

## 2.2 Çözüm Yöntemleri ve Literatür Taraması

Bu bölümde üniversite çizelgeleme probleminin çözümünde kullanılan çözüm yöntemleri ve literatürde yapılan çalışmalar incelenerek özetlenmiştir.

## 2.2.1 Yöneylem araştırması yaklaşımları

### 2.2.1.1 Sıralı (Enumerative) arama yöntemleri

Dinamik programlama, dal-sınır arama yöntemi, lagrange gevşemesi, doğrusal, tamsayı programlama teknikleri ve genel olarak matematiksel programlama teknikleri sıralı arama yöntemleri arasında yer almaktadır.

Büyük bir problemin, çözülmeye önce daha küçük alt problemlere bölünerek, her alt problemin ayrı ayrı çözülmesi ve daha sonraki kullanım için saklanması mantığına dayanan dinamik programlama yönteminin büyük çaptaki çizelgeleme problemlerinde etkisiz olduğu Held ve Karp (1962) tarafından gösterilmiştir.

İlk aşamada problemin iki veya daha fazla problemlere dallandırılarak, ikinci aşamada ise ilgili alt problemin uygun değer çözümleri üzerinde alt sınır hesaplaması yapılması mantığına dayanan dal-sınır arama yönteminde yapılan çalışmalar, dallanma stratejilerine ve sınırların yaratılmasına göre farklılık kazanmaktadır. Balas (1969), McMahan ve Florian (1975), Barker ve McMahan (1985), Carlier ve Pinson (1989) çalışmalarında bu yaklaşımın örneklerini göstermişlerdir.

Tripathy (1980) çizelgeleme problemi için yaptıkları matematiksel programlama yaklaşımları çalışması sırasında, dal-sınır işlemi içerisine gömülü, Lagrange gevşemesi (Lagrangian Relaxation) tabanlı bir algoritma geliştirmişlerdir. Küçük çaptaki çizelgeleme problemi için uyguladıkları bu yöntemin, daha büyük çaptaki çizelgeleme problemleri için hala geliştirme aşamasında olduğunu belirtmişlerdir. Arani vd (1988) çalışmalarında, sınav çizelgeleme problemini üç aşamaya bölerek sınav bloklarının sınav günlerine atanması aşamasını Lagrange gevşemesi tabanlı bir yöntem ile çözmüşlerdir.

Tripathy (1984) çalışmasında, çizelgeleme problemini 0 ve 1 değişkenleri ile büyük bir tamsayı doğrusal programlama problemi olarak formüle etmiştir. Billionnet (1999) çalışmasında, klasik personel çizelgeleme problemini çözmek için tamsayı programlamanın etkin bir teknik olduğunu göstermiştir. Maksimum ardışık izin günü sayılarının programlanmasında da formül seçiminin kritik önemi olduğunu belirtmişlerdir. Hesaplama süresinin ise küçük bir mikrobilgisayarda sadece birkaç saniye olduğunu söylemektedirler. Dimopoulou ve Miliotis (2001) çalışmalarında,

üniversite ders ve sınav çizelgeleme sistemi geliştirmişlerdir. Bu sistem, dersleri zaman dilimleri ve sınıflara atayan tam sayılı programlama modeli kullanmaktadır. Model, öğrencilerin boş zamanlarını azaltmaya, sınıfların daha etkin kullanımına ve oturumların farklılığına yönelik çalışmaktadır. Öğretim üyeleri istekleri ise amaç fonksiyonuna eklenen katsayılar ile karşılanmaktadır. Sınav çizelgeleme çalışmasının başlangıç noktası için ders çizelgeleme çalışmasının sonuçları kullanılmaktadır. Atina Üniversitesi Ekonomi ve İşletme bölümünün sistemi başarıyla kullandıklarını söylemektedirler. Daskalaki vd (2004) çalışmalarında, üniversite çizelgeleme problemi için 0-1 değişkenli tamsayı programlama modeli geliştirmişlerdir. Belirlenen her kısıtın sağlanmadığı durumlarda maliyet fonksiyonuna değer eklenmekte ve en düşük maliyet fonksiyonuna karşılık gelen çözüm, uygun değer çözüm olarak belirlenmektedir. Ardı ardına gelen zaman dilimlerinde yapılan oturumlar ve aynı derslerin farklı oturumlardaki tekrarlarının engellenebilmesi için, modelde bu kısıtlar zorunlu kısıtlar olarak belirlenmişlerdir. Modellerini, üniversitelerinin elektrik ve bilgisayar mühendisliği bölümlerinin çizelgelerinin oluşturulmasında test etmişler ve başarılı sonuçlar elde etmişlerdir. Daskalaki ve Birbas (2005) çalışmalarında, üniversite çizelgeleme problemi için yeni bir yaklaşım geliştirmişlerdir. İki evreli tamsayı programlama modellerinin, birinci evresinde, belli derslerin çoklu oturumlarının arka arkaya gelmesinin garanti edilmesi ile ilgilenilmekte, ikinci evrede ise, yerel optimuma ulaşmak için belirlenen alt kısıtlar dikkate alınmaktadır. Tek aşamada çözen yöntemlerle karşılaştırıldığında, çözüm kalitesinde bir kayıp olmadan hesaplama zamanının önemli derecede azaldığını söylemektedirler. MirHassani (2006) çalışmasında, üniversite çizelgeleme problemi için yeni bir 0-1 tamsayı programlama yöntemi geliştirmiştir. Öğrenci gereksinimlerini ve sınıf kapasitelerini dikkate alan, diğer kısıtları göz ardı eden bir amaç fonksiyonu kullanmıştır. Penaltı fonksiyonlarını değiştirerek hesaplama süresini büyük oranda düşürmenin mümkün olduğunu, yani optimum çözüme ulaşmanın çok daha hızlı olabileceğini belirtmiştir. Al-Yakoob ve Sherali (2006) çalışmalarında, Kuveyt Üniversitesinde bölüm öğretim üyelerinin belli zaman dilimlerinde belli sınıflara atanması gerçekleştirmek için matematiksel programlama modeli geliştirmişlerdir. Atamalar öncelikle karışık-doğrusal programlama ile öğretim üyelerinin özelliklerine göre yapılmakta, daha sınıf çakışmaları ve etkinliği gözeterek, toplam memnuniyeti artırmak öğretim üyelerinin istekleri göz önüne alınmaktadır. MirHassani (2006) çalışmasında, tamsayı programlama kullanarak sınavların mümkün olan en uzun aralıklarla yapılmasını sağlama amacını güden sınav çizelgeleme modeli geliştirmiştir.

Bu model, önceden tanımlanmış bir sınav çizelgesini bütün öğrenciler için, göz önüne alan kısıtlara göre en uygun hale getirmeye çalışmaktadır. Al-Yakoob ve Sherali (2007) çalışmalarında, Kuveyt Üniversitesi'nin çizelgeleme problemini çözmek için karışık bir doğrusal programlama modeli geliştirmişlerdir. Bu modelde, sınıf çakışmalarını minimize etmeye yönelik, giriş ve çıkışlarda trafik akışının hızlandırılmasını sağlayacak çözümler üretmeye çalışılmaktadır. Yaptıkları çalışmanın sonuçlarını, elle yapılan eski atamaların sonuçları ile karşılaştırdıklarında, hem zaman hem de trafik açısından kaynakların etkin kullanımını sağlayan model sayesinde büyük kazanç sağladıklarını belirtmektedirler. Boland vd (2008) çalışmalarında, ders çizelgeleme problemi için yeni bir doğrusal programlama yaklaşımı geliştirmişlerdir. Bu yaklaşımda, sınıflar kümesini alt-kümelere ayırarak, blok olarak adlandırılan her alt-küme kendi içinde paralel olarak çizelgelenmektedir. Sundukları modelle, yerel bir lisenin derslerinin doğrusal programlama ile çizelgelenebileceğini kanıtladıklarını söylemektedirler. İlerideki çalışmalarında, bloklama yaklaşımlarını üniversite ders çizelgeleme problemleri için denemeyi düşünmekte olduklarını belirtmişlerdir.

Sıralı arama yöntemleri içerisinde en çok çalışma yapılmış olan matematiksel programlama, bağımsız değişkenler tarafından kısıtlanan bir fonksiyonu optimize etmek için kullanılan bir tekniktir. Küçük çaptaki çizelgeleme problemleri için en uygun çözümlere ulaşabilse de, problemin büyüklüğü arttıkça iyi çözümlere ulaşması zorlaşmaktadır. Problemin karmaşıklığının ve boyutunun artmasıyla çözüm uzayının büyümesi farklı çözüm yöntemlerinin aranmasını zorunlu kılmıştır.

### **2.2.1.2 Sezgisel yöntemler**

Çoğu çizelgeleme problemi polinom zamanda çözülemeyen tam (NP-complete) problemler olduğu için, Garey ve Johnson (1990) tarafından çalışmalarının ekinde işaret edildiği gibi sıralı arama yöntemleri ile polinom zamanda sonuç elde edebilmek neredeyse imkânsızdır. Öyleyse bir kaynağın ne zaman ve nasıl kullanılacağına karar verirken her ne kadar optimum sonuca ulaşmayı garanti etmese de bazı kurallar uygulanmak zorundadır. Gerçek çizelgeleme problemlerinde performansın önemli olmadığı durumlarda basit modeller kullanılabilir. Basit sezgiseller içinde rastgele arama yapılarak uygun bir çözüm elde etmek mümkün olsa da, tamamen rastgele yapılan aramalarda performans beklenemez. Örneğin, birçok sınavın kısa bir zaman bloğuna fazla sayıda kısıtları göze alarak rastgele çizelgelenmesi samanlıkta iğne

aramaya benzer. Bu yüzden klasik sezgisel tabanlı arama yöntemleri daha çok küçük çaptaki kısıtı az olan çizelgeleme problemlerinde ancak yeterli olabilir. Çoğunlukla bir yerel optimuma takılması beklenir.

Carter (1986) çalışmasında, pratik çizelgeleme problemlerine uygulanan grafik renklendirme sezgisellerini incelemiştir. Ergül (1995) yüksek lisans çalışmasında, Fang (1994) doktora çalışmasında üniversite çizelgeleme problemine uygulanan grafik renklendirme sezgisellerini incelemiştir. Weitz ve Lakshminarayanan (1997) çalışmalarında, çeşitli grupların maksimizasyonu, VLSI tasarımı ve sınav çizelgeleme problemlerin çözümü için geliştirilen literatürdeki yöntemleri deneysel olarak karşılaştırmışlardır. Gerçek veri setlerine uyguladıklarında, üretilen sonuçların kalitesi ve CPU işlem zamanı göz önüne alındığında, 'Lotfi-Cerveney Weitz' (LCW) yöntemini seçmişlerdir. Drexl ve Salewski (1997) çalışmalarında, farklı uzunluktaki derslerin gereksinimlerinin, farklı tipteki kısıtların sağlanabilmesi için yeni bir model geliştirerek optimuma yakın çözümler elde ettiklerini söylemişlerdir. de Werra ve Mahadev (1997) çalışmalarında, önceden atama yapılması gereken küçük çapta sınıf-öğretmen çizelgeleme problemlerinin, kurulan modeller ile polinom zamanda çözülebileceğini matematiksel olarak göstermişlerdir. de Werra (1997) çalışmasında, basit sınav çizelgeleme algoritmalarının karmaşıklıklarını incelemiştir. Çok kapsamlı inceleme içermeyen ve araştırmaya katkı iddiasında bulunmayan bir çalışmadır. Sınav çizelgeleme gibi, klasik yöntemlerle polinom zamanda çözülemeyen karmaşık problemlerin çözümünde, yeni sezgisel algoritmaların geliştirilmesi gerekliliğinin altını çizmiştir. de Werra (1999) çalışmasında, yerel ve genel değişkenlerin aynı zamanda olması durumunda da, kromatik çizelgeleme problemlerin çözülebileceğini göstermişlerdir. Ancak uygulamada eksiklikler olduğunu, kombinasyonel optimizasyon problemlerinde etkili çözüm yöntemlerinin geliştirilmesi için daha fazla araştırma gerektiğini belirtmiştir. Çeşitli tipteki gereksinimleri aynı zamanda karşılayabilmek için genel bir yaklaşıma ihtiyaç olduğunu altını çizmiştir. Bu noktada da, problemi daha kolay problemlere ayırıp, çözdükten sonra tekrar birleştirmeyi önermiştir. Dammak vd (2006) çalışmalarında, sınav çizelgeleme ile ilişkili olan sınav-sınıf atamalarını gerçekleştirmek için basit bir sezgisel yöntem geliştirmişlerdir. Bu yöntem, sınıf-sınav atamasını, her sınıfta bir sınav olacak şekilde yapmaya çalışmakta, eğer bunu sağlayamazsa en fazla iki sınıfa atama yapmaktadır. Her sınıfta, bir ya da iki sınav olabileceği düşünüldüğünde, geliştirdikleri modelin, en uygun çözümü bulamadığı için,

dezavantaja sahip olduğunu belirtmektedirler. Head ve Shaban (2007) çalışmalarında, Birleşik Arap Emirlikleri Üniversitesi'nin Genel Gereksinimler Ünitesinde, üniversitenin kaynaklarının optimuma yakın kullanımına yönelik, sezgisel bir yaklaşımla Visual Basic ve Sql programla dillerini kullanarak çizelgeleme sistemi geliştirmişlerdir. Oluşturulan çizelgeler, yönetim ve öğrenciler tarafından iyi olarak kabul edilmiştir. Yeni sistemin geliştirdiği çizelge, elle hazırlanan çizelge ile karşılaştırıldığında, 66 tane daha az sınıf kullanıldığı, sınıflara öğrenci sayısı dağılımının çok daha düzenli olduğunun görüldüğünü söylemektedirler.

Doğada bulunan bir kristal yapısının oluşması için elektronların soğutma işinin bir optimizasyon yöntemi olarak uyarlanması olan benzetilmiş tavlama yöntemine ilgi Kirkpatrick vd (1988) tarafından yapılan çalışma ile başlamıştır. En düşük enerji seviyesindeki elektronların oluşturduğu mükemmel bir kristal, soğutmanın uygun bir sıcaklıktan başlayarak uygun bir orana gelmesiyle elde edilmektedir. Optimizasyon terimlerinde, mükemmel kristal yapısı global minimuma, başlangıç sıcaklığı bir başlangıç çözümüne, soğutma oranı ise minimize edilecek olan amaç fonksiyonunda bir artışla sonuçlanan değişimi kabul etme oranına karşılık gelmektedir. Yöneylem araştırması literatüründe birçok benzetilmiş tavlama uygulamaları bulunmaktadır. Hem Eglese ve Rand (1987), hem de Dowsland (1990) çalışmalarında, geçerli çözümler üretilmeyen çizelgeleme problemlerinde bazı kısıtların göz ardı edilmesinin gerekliliği üzerinde durmuşlardır. Abramson (1991) çalışmasında, hem sıralı hem de paralel makineler kullanarak okul çizelgeleme problemine benzetilmiş tavlama modelini uygulamıştır. Benzetilmiş tavlama tekniği Johnson (1990) tarafından sınavların her günde 2 zaman dilimi olan 10 günlük sınav dönemine atanması gereken South Pacific Üniversitesi'ndeki sınavların çizelgelenmesi için kullanılmıştır. Başlangıç çizelgesinin oluşturulması, başlangıç çizelgesinin iyileştirilmesi ve sınav günlerinin yeniden ayarlanması şeklinde problemi üç aşamada çözmüşlerdir. Birinci aşamada grafik renklendirme yöntemi kullanılmıştır. Her iki oturumdaki sınavları alan öğrencilerin sayısını içeren çakışma matrisi kullanarak benzetilmiş tavlama yöntemi ile daha iyi çizelgeler aranmıştır. Son aşamada arka arkaya gelen sınavların minimize edilmesine ve uygun sınav salonlarının bulunmasına yönelik bir çalışma yapılmıştır.

Yasaklı arama yönteminin modern şekli Fred (1986) tarafından türetilmiştir. Glover ve Laguna (1993) yazdıkları kitap bölümünde, yasaklı arama yönteminin

uygulanabilirlik sınırlarına geçmek, yerel optimum engellerini ortadan kaldırmak ve diğer yasak bölgelerin taranmasına izin vermek için sistematik olarak kısıtları serbest bırakmak veya uygulamaya koymak mantığı üzerine kurulmuş olduğunu belirtmişlerdir. Yasaklı arama yönteminde zor bölgelere ulaşabilmek için bir arama yöntemi sağlayan kısıtlamalar uygulamaya konulmaktadır. Hertz (1991) çizelgeme problemlerine sadece ikincil kısıtlarla uğraşan yasaklı arama yöntemini uyarlamışlardır. Hem Widmer ve Hertz (1989), hem de Taillard (1990) çalışmalarında, akış tipi çizelgeme problemlerine yasaklı arama yöntemlerini uygulamışlardır. Taillard (1993) çalışmasında, Beasley (1990) tarafından verilen çizelgeme problemlerinin karşılaştırması üzerine koydukları sonuçları göstermişlerdir. Yasaklı aramayı kullanarak en iyi çözüme ulaşmayı hedeflediği için çalışma süresinin çok uzun olduğunu belirtmiştir. Alvarez-Valdes vd (2002) çalışmalarında, yasaklı arama yöntemini kullanarak ders çizelgeme sisteminin tasarımı ve uygulamasını geliştirmişlerdir. Zorunlu ve diğer kısıtlara ağırlık değerleri vererek, iterasyonların sonunda en küçük toplamı veren çözümü çizelge olarak belirlemişlerdir. Kullanıcı dostu bilgisayar programlarının oluşturduğu çizelgeyi elle hazırlanan çizelge ile karşılaştırdıklarında daha tatmin edici sonuçlar aldıklarını söylemektedirler. Burke vd (2007) çalışmalarında, eğitimsel çizelgeme problemleri için, yasaklı arama ve dik iniş metotlarını içeren, başlangıç popülasyonuna ihtiyaç duymayan bir sezgisel bir yaklaşım geliştirmişlerdir. Yaklaşımlarını literatürdeki problemlere uyguladıklarında iyi sonuçlar almışlar, ilerideki çalışmalarda farklı amaç fonksiyonu olan, farklı kısıtlardaki sınav çizelgeme problemlerine uygulanarak, hipotezlerinin test edilmeye ihtiyacı olduğunu belirtmişlerdir.

Burke ve Petrovic (2002) çalışmalarında, otomatik çizelgeme konusundaki mevcut araştırmaları incelemişlerdir. Yaklaşımların genellikle, çizelgeme probleminin belli sürümlerini çözmeye yönelik olduğunu belirlemişlerdir. Özel problemlere özgü algoritma geliştirilmesinden önce, problem için uygun algoritmayı belirleyen yöntemler geliştirilmesi gerektiğini söylemektedirler. Genetik algoritmaların iyi bir algoritmanın aranmasında kullanılmasını tavsiye etmektedirler. Asratian ve de Werra (2002) çalışmalarında, temel çizelgeme problemi için ortak olarak alınmak zorunda olan dersleri dikkate alan teorik bir model geliştirmişlerdir. Bu modelin belirledikleri kabullere göre optimuma yakın sonuç verdiğini söylemektedirler. Santiago-Mozos vd (2005) çalışmalarında, bir İspanya üniversitesindeki bazı derslerde, kişiselleştirilmiş



çizelgeler elde etmek için, sezgisel bir algoritma geliştirmişlerdir. Bu algoritmada, öğrenciler verilen bir konuda tercih ettikleri grupları ve derslerde önceliklerini belirleyebilmektedirler. İki aşamadan oluşan bu sezgiselde, ilk aşamada, problem kısıtları çözülmekte, ikinci aşamada ise bulunan çözümler iyileştirilmeye çalışılmaktadır. 1301 öğrencinin kişisel çizelgeleri belirlenen kriterlerine göre %99 başarı ile gerçekleştirildiğini söylemektedirler. Najî Azimi (2005) çalışmasında, sınav çizelgeleme problemi için tavlama benzetimi, yasaklı arama, genetik algoritma ve karınca koloni sistemi yaklaşımlarını incelemiştir. Belirlediği karşılaştırma ölçütlerine göre, en iyi çözümü karınca kolonisi sistemi vermiş ve onu sırasıyla yasaklı arama, genetik algoritma, tavlama benzetimi iyi çözüm olarak takip etmişlerdir. En fazla bozulma üreten algoritma ise yasaklı arama olmuş ve onu da sırasıyla, karınca kolonisi sistemi, tavlama benzetimi ve genetik algoritma bozulma oranlarıyla takip etmişlerdir. Saf metotların davranışlarını inceledikten sonra, karınca kolonisi sistemi ile yasaklı arama yönteminin farklı şekillerde birleşiminden oluşan melez bir algoritma geliştirmiştir. Bu melez algoritma, her iki yöntemin de iyi sonuç veren özelliklerinden yararlanarak daha iyi çözüm üreten bir yöntem haline gelmiştir. Azimi, elde edilen sonuçların, geliştirilen algoritmanın tasarımına göre farklılık gösterebileceğinin altını çizmiştir. Projemizde de Azimi'nin elde ettiği bu bulguya dayanarak melez bir yapı ortaya çıkarılmaya çalışılacaktır. Beligiannis vd (2008) çalışmalarında, Yunanistandaki liselerde, uygun ve etkili çizelgeler oluşturabilmek için, evrimsel hesaplama teknikleri tabanlı uyarlamalı bir algoritma tasarlamışlardır. Bu algoritma birçok Yunan lisesinden gelen gerçek verilerle etkinliğini kanıtlamak için test edilmiştir. Simülasyon sonuçlarına göre, bu algoritma kısa sürede çok etkili çizelgeler oluşturarak, öğretmenlerin şikâyetleri en aza indirilmiş ve dönem boyunca kaynaklar en etkili şekilde kullanılmıştır.

### **2.2.2 Yapay zekâ yaklaşımları**

Kısıt tatmin problemi yaklaşımları, bilgi tabanlı sistemler, dağıtık yapay zekâ yöntemleri, kural tabanlı uzman sistemler ve yapay sinir ağları çözüm yöntemleri genel olarak yapay zekâ yaklaşımları içinde gruplandırılabilir.

Kısıt tatmin problemi tekniğinde; ayrık ve sonlu olası değerler kümesine sahip olan değişkenler kümesi ve bu değişkenler arasında kısıtlar kümesi bulunmaktadır. Bütün kısıtların sağlandığı değişken-değer atamalar kümesi kısıt tatmin probleminde çözüme

karşılık gelmektedir. Deris vd (2000) çalışmalarında, ders çizelgeleme problemini kısıt-tatminine yönelik çalışan nesne tabanlı bir algoritma geliştirmişlerdir. Bir kolejden aldıkları gerçek veriler üzerinde çalışarak, 18 haftalık çizelgeye uygun yerleşimi sağlamışlardır. Algoritmanın etkinliğini, kısıtların kontrol edilerek yayılmasıyla, esnekliği ve adaptasyonu ise nesne tabanlı tasarımı ile başarmıştır. Bu yaklaşımın farklı problemlere de uygulanabileceğini söylemektedirler. Valouxis ve Housos (2003) çalışmalarında, öğretmenlerin tercih kriterlerine ve özelliklerine göre atama yapan kısıt tabanlı bir çizelgeleme sistemi geliştirmişlerdir. Öğretmenlerin gün içinde fazla boşta beklememesini önemli kriter olarak belirlemişlerdir. Abbas ve Tsang (2004) projelerinde, genel bir çizelgeleme problemini temel alarak, yazılım mühendisliği bakış açısıyla kısıtların tatminine yönelik bir çalışma yapmışlardır. Tatmin her öğrenci için ayrı ayrı hesaplanmakta ve toplanarak maliyet fonksiyonu belirlenmektedir. Bir üniversite ve bir okulda test edilen proje, her zaman optimum sonucu verememesine rağmen etkili sonuçlar alınmıştır.

Foulds ve Johnson (2000) çalışmalarında, üniversite çizelgeleri oluşturmak için Microsoft Windows ortamında Access ve bileşenlerini kullanarak, kullanıcı dostu menüleri olan bir karar destek sistemi geliştirmişlerdir. Bu sistemi kullananların tecrübelerine göre, çizelgenin oluşturulmasına vazgeçilmez yardımda bulunmakta olduğu ve tamamen olumlu olduğu kabul edilmiştir. Dimopoulou ve Miliotis (2004) çalışmalarında, üniversitedeki derslerin çizelgelenmesi için bilgisayar ağı tabanlı bir sistem geliştirmişlerdir. Bu sistem, gerekli tüm bilgilerin bulunduğu merkezi bir veritabanı kullanmaktadır. Sistem, her bölümün kısıtlarına göre, ayrı ayrı çizelgeler oluşturmak için tamsayılı programlama yöntemini kullanmaktadır. Bölümlerden gelen verilere göre hesaplamalar yapıldıktan sonra, merkezde oluşan çizelge bölümlere dağıtılmaktadır. Bu sistemi Atina Üniversitesi'nden sağlanan veriler ile test etmişlerdir. Bu problemin farklı optimizasyon teknikleri ile çözümlerinin araştırılması gerekliliğinin altını çizmişlerdir. Petrovic vd (2007) çalışmalarında, sınav çizelgeleme için CBR (case based reasoning) tabanlı bir sistem geliştirmişlerdir. Bu sistem, yeni bir problemi sıfırdan çözmeye başlamak yerine, daha önce çözülmüş olan problemlerin çözümlerinin sakladığı veri tabanına bağlanarak, yeni probleme en benzer olan problemin çözümünü başlangıç çözümü olarak kullanmaktadır. Yasaklı arama yöntemi ile bu çözümdeki gerekli düzeltmeler yapılmakta ve GDA (great deluge algorithm) ile de nihai çözüm bulunmaktadır. Dikkate alınan zorunlu ve isteğe bağlı kısıtlara göre bulunan bu çözüm

de CBR veritabanına, başka bir problemin çözümünde kullanılabilirlik üzere kaydedilmektedir.

Kombinasyonel optimizasyon problemlerinde ileri beslemeli yapay sinir ağları yöntemi yaygın olarak kullanılmaktadır. Smith vd (2003) çalışmalarında, çizelgeleme problemini çözmek için, ayrık Hopfield sinir ağı yaklaşımı geliştirmişlerdir. Mevcut sezgisel yöntemlerle karşılaştırdıklarında, daha büyük problemlerde daha etkili sonuçlar aldıklarını söylemektedirler. Modelde kullanılan formülün performans üzerindeki etkisinin altını çizmişlerdir. Carrasco ve Pato (2004) çalışmalarında, çizelgeleme probleminin çözümü için yapay sinir ağı tabanlı iki sezgiselin karşılaştırmasını yapmışlardır. Problemin karakteristikleri yapay sinir ağına özgü enerji fonksiyonuna dönüştürülerek her iki sezgisel yaklaşım ile problem çözülmüştür. Sadece zorunlu kısıtlar göz önüne alındığında, her iki sezgisel de iyi sonuçlar vermiştir. Ancak ayrık sinir ağı, devamlı sinir ağına göre, çok daha hızlı bir şekilde kabul edilebilir çözümler üretmiştir. Uygulama tecrübelerine göre, ayrık sinir ağı algoritmasının çakışmayan çizelgeler üretebildiğini vurgulamışlardır.

Sonuç olarak popüler bir problem olan üniversite çizelgeleme problemlerinin çözümünde kullanılması gereken yöntem, problemin büyüklüğüne, hangi kısıtlardan ne kadar fedakârlık edilebileceğine ve hedeflenen amaç fonksiyonuna göre farklılık göstermektedir. Bu çalışmada, öğretim üyesi istekleri ve kapasite durumları gibi kısıtlar göz ardı edilerek tamamen zaman dilimlerinin öğrenciler için en uygun şekilde ayarlanması üzerine odaklanılmıştır. Çözüm yöntemi olarak seçilen genetik algoritmalar ile geliştirilen özgün modelin, üniversite çizelgeleme problemlerinin çözümünde genetik algoritmaların etkinliğinin araştırılmalarına katkı sağlayacağı düşünülmektedir. Sınavların birbirinden mümkün olan en uzak yayılıma sahip olması fikri tamsayı programlama modeli ile daha önce gerçekleştirilmiştir. Temel çakışmaları önleme fikri de genetik algoritma modeli ile çözülmüştür. Bu çalışmada ise; sınavların zorluk dereceleri de hesaba katılmış, zor olan sınavların kolay olan sınavlara göre birbirinden daha uzak zaman dilimlerine atanmaları fikri üzerine odaklanılarak amaç fonksiyonu zorlaştırılmıştır. Zorluk derecelerini de kapsayan amaç fonksiyonunu minimize etmeye yönelik genetik algoritma modeli ile çözüm yaklaşımı geliştirilmiştir. Geliştirilen klasik genetik algoritma modeline alternatif yaklaşım sunularak karşılaştırılmış ve sonuçları beşinci bölümde verilmiştir.

### 3. GENETİK ALGORİTMALAR

Bu bölümde genetik algoritmaların çalışma mantığı ve operatörleri hakkında genel bilgiler verilmiştir. Bu bölümün amacı, okuyuculara GA konusunda bir altyapı oluşturmaktır.

#### 3.1 Genetik Algoritmalara Giriş

Genetik algoritmalar evrim teorisinin doğal seçim fikrine ve genetiğin temelleri üzerine oturmaktadır. Her bireyin özelliklerini çocuklarına geçirmesi, doğanın farklı özellikteki bireyler üretmesi, avantajlı özellikleri olan bireylerin diğer bireylerden daha çok çocuk sahibi olma eğiliminde olmaları ve uzun zaman içinde değişimin belli çevre şartlarına uyum gösteren özelliklere sahip olan tamamen yeni bireyler üreteceğini Darwin (1936) 'Türlerin Kökeni' aldı eserinde belirtmiştir.

Daha iyi uyum gösteren bireyler çocuk üretimi için seçilmeye daha çok eğilimlidir. Bu daha iyi olan bireylerin çocukları genetik özelliklerini kalıtsal olarak almaktadırlar. Kalıtımdaki genetik özellikleri taşıyan genlerden de yeni nesilde daha iyi uyum gösterenler eşleşme ve yeniden üretme için daha sık seçilme eğilimindedirler. Bu evrimsel döngü nesilden nesile devam eder. Böylece uyum gösteremeyen kromozomlar ve genler çevreden zamanla yok olarak yerlerini daha uyumlu genler içeren kromozomlara bırakırlar. En uyumlunun hayatta kalması sonucunda yeni nesiller kademeli olarak çevreye optimuma yakın bir uyum göstereceklerdir.

#### 3.2 Genetik Algoritma Nedir?

Bir GA olağandışı bir arama stratejisi olarak görülebilir. Bir GA' da, bir problem için genelde ilk neslini rastgele çözümlerin oluşturduğu aday çözümler kümesi vardır. Her aday genellikle genlerin oluşturduğu belli uzunluktaki dizilerden oluşur. Belli bir problem için genetik algoritma yapısını oluşturmadaki ilk iş olası çözümlerin nasıl temsil edileceğine karar vermektir. Bir GA genellikle aşağıdaki gibi ilerler:

- Bir aday çözümler kümesi rastgele üretilir.

- Uygun durdurma kriteri (belli bir zaman sonra en iyi çözümde bir gelişme olmaması gibi) sağlanana kadar aşağıdaki adımlar icra edilir:
  - Değerlendirme: Probleme göre önceden tanımlanmış uygunluk ölçütleri kullanılarak toplumdaki her bireyin ne kadar iyi çözümler olduğu hesaplanır. Daha iyi uygunluk değerine sahip olan adaylar aranan çözümlerdir. Ancak uygunluk değeri her zaman iyi bir kalite ölçütü olmayabilir, çünkü iyi uygunluk değerine sahip bir kromozomda kötü genler olabileceği gibi, kötü uygunluk değerine sahip bir kromozomda da iyi genler bulunuyor olabilir.
  - Seçim: Toplumdaki aday çözümlerden eşleşme için çiftler seçilir. Bu seçim işlemi tamamen rastgele, uygunluk değerleri temel alınarak veya başka bir şekilde yapılabilir.
  - Doğurma: Seçilen bireylerden genetik operatörler kullanılarak yeni bireyler üretilir. İki ana çeşit genetik operatör bulunmaktadır.
    - ✓ Çaprazlama: Bir çift kromozomun genlerinin yeniden birleşmesi ile yeni bir birey üretilir.
    - ✓ Mutasyon: Bir birey çok az değiştirilerek yeni bir birey elde edilir.
 Çaprazlama işlemiyle, çaprazlanan bireylerin parçalarının birleşiminden daha iyi bireyler üretilmesi hedeflenmektedir. Daha kötü bir birey oluşması durumunda ise bu bireyin seçilme olasılığı azalmaktadır. Her durumda da, çaprazlanan bireylerin özellikleri farklı kombinasyonlar ile çocuğa iletilmektedir. Mutasyon ise çaprazlama ile elde edilemeyebilen küçük değişikliklerle daha iyi bireyler oluşturulmasına hizmet etmektedir.
  - Toplum güncelleme: Nesildeki kötü bireyler, elde edilen çocukların ve mutasyona uğrayan bireylerin en iyileri ile değiştirilerek sıradaki adımda kullanılacak olan yeni nesil elde edilir.

GAların diğer optimizasyon ve arama yöntemlerinden nasıl ayrıldığını Goldberg (1989) dört ana farkla açıklamıştır:

1. GAlar parametrelerin kendisi ile değil, parametreler kümesinin kodlanmasıyla çalışır.
2. GAlar tek bir noktadan değil, noktalar popülasyonundan arama yapar.
3. Türev ya da diğer yardımcı bilgileri değil, amaç fonksiyonu bilgisini kullanır.
4. GAlar kesin kuralları değil, olasılık geçiş kurallarını kullanır.

### 3.3 Genetik Algoritmalar Nasıl Çalışır?

GA'lar bir problemi aday çözümleri üreterek, değiştirerek ve değerlendirerek çözerler. Bir aday çözüme, çözümün gösterimi ya da kodlaması olan bir kromozom adı verilir.

Başlangıçta, rastgele kromozomlardan oluşan bir popülasyon üretilir. Kromozomlar, çaprazlama veya mutasyon işlemleri ile değiştirilir.

Temel bir GAnın aşamaları aşağıdaki şekildedir:

- Rastgele kromozomlardan oluşan bir popülasyonu başlat.
- Popülasyondaki her kromozomun uygunluk değerini değerlendir.
- Popülasyondaki kromozomları olasılıklarına göre çiftleştirmek için seç.
- Seçilen kromozomlara çaprazlama ve mutasyon operatörlerini uygula.
- Yeni popülasyonu belirle.
- Yukarıdaki adımları yeterli bir çözüm bulunana kadar tekrar et.

Bir GA küçük bilgi alanına ihtiyaç duyan bir arama metodu olduğu için fazla sayıda probleme uygulanabilmektedir.

### 3.4 Genetik Algoritmaların Ayrıntılı Adımları

#### 3.4.1 Başlatma

Başlangıç popülasyonu ikili veya ikili olmayan genlerden, sabit ya da değişken uzunlukta kromozomlardan oluşabilir. Holland (1992)'ın kitabındaki orijinal kodlama biçimi ikili genlerden ve sabit uzunlukta kromozomlardan oluşur. Başlangıç aşamasında sistem, rastgele geçerli kromozomlar oluşturur ve her birini değerlendirir.

#### 3.4.2 Yeniden üretim

##### 3.4.2.1 Nesilsel yeniden üretim

Bu üretim yönteminde, popülasyonun tamamı her yeni nesilde değişir (Holland 1992). En sık uygulanan işlem popülasyon sayısının yarısı kadar nesil üretim yapmaktır.

### **3.4.2.2 Kararlı hal üretimi**

Bu üretim yönteminde ise, seçilen kromozomlara çaprazlama ve mutasyon operatörleri uygulandıktan sonra oluşan yeni kromozomlar popülasyona eklenir ve uygunluk değeri en kötü olan kromozomlar yok edilir (Whitley 1989).

### **3.4.3 Seçim**

Çaprazlama ve mutasyon işlemleri için kromozomlar uygunluk değerlerinin popülasyon içindeki olasılık değerlerine göre seçilir.

#### **3.4.3.1 Uygunluk değeri tabanlı seçim (Rulet tekeri seçimi)**

Kromozom seçimi için standart ve orijinal olan yöntem uygunluk değeri tabanlı seçimdir (Rulet tekeri seçimi). Bu seçimde, her kromozom tamamen uygunluk değerinin oranına göre seçilme şansına sahiptir. Bu etki tamamen popülasyondaki uygunluk değerlerinin dağılımına bağlıdır. Popülasyondaki en yüksek uygunluk değeri ile en düşük uygunluk değeri arasındaki farkın oransal olarak azalması seçilme olasılıklarını birbirine çok yakın yaparak aramada hareketsizliğe neden olacağı için istenmeyen bir durumdur. Bu problemi çözmek için seçim aşamasına geçmeden önce uygunluk değerlerini ölçülendirmek gerekir (Goldberg 1989).

#### **3.4.3.2 Sıralama tabanlı seçim**

Sıralama tabanlı seçimde, seçim olasılıkları mutlak uygunluk değerlerinden ziyade, kromozomun popülasyondaki pozisyonuna ya da göreceli sırasına göre belirlenmektedir (Barker ve McMahon 1985). Sıralama tabanlı seçimin, sıraların nasıl amaç uygunluğuna çevrileceğine bağlı olan farklı şekilleri bulunmaktadır (Whitley 1989). Whitley'in eğilim seçiminde, daha iyi uygunluk değerine sahip olan kromozomlara daha büyük ağırlık verilerek seçilme olasılıkları artırılmaktadır.

#### **3.4.3.3 Turnuva tabanlı seçim**

Orijinal turnuva seçiminde, belli sayıda rastgele seçilen kromozom arasından en iyi uygunluk değerine sahip olan kromozom seçilir (Brindle 1981). Boltzmann turnuva seçiminde popülasyondaki olasılıklara göre kabul veya red mekanizması çalışmaktadır. Rastgele seçilen bir kromozom komşu kromozom ile değerlendirilerek olasılık hesabına

göre kabul veya reddedilmektedir (Goldberg 1990). Evlilik turnuva seçimine göre rastgele bir kromozom seçilir. Daha iyi bir kromozom bulmak için belli sayıda deneme yapılır ve sonuçta en iyi kromozom seçilir (Ross ve Ballinger 1993).

#### **3.4.3.4 Uzaya dayalı seçim**

Tüm popülasyon içinden seçim yapmak yerine farklı bölgelerdeki kromozomlar arasından seçim yapıldığı için global değil, yerel bir seçim yöntemidir. Rastgele kromozomlardan başlayarak belli sayıda rastgele iki farklı yürüyüş ile bulunan en iyi iki kromozom seçilir. Bu eşleşmeden elde edilen çocuklar daha iyi kromozomlar ise, ilk kromozomların yerlerine geçerler (Ross ve Ballinger 1993).

### **3.5 Genetik Operatörler**

#### **3.5.1 Çaprazlama operatörü**

Çaprazlama operatörü genetik algoritmalarındaki en önemli operatördür. Kromozom çiftlerinin parçaları arasında farklı kombinasyonlarda değişim işlemidir. Birçok çaprazlama çeşidi bulunmaktadır:

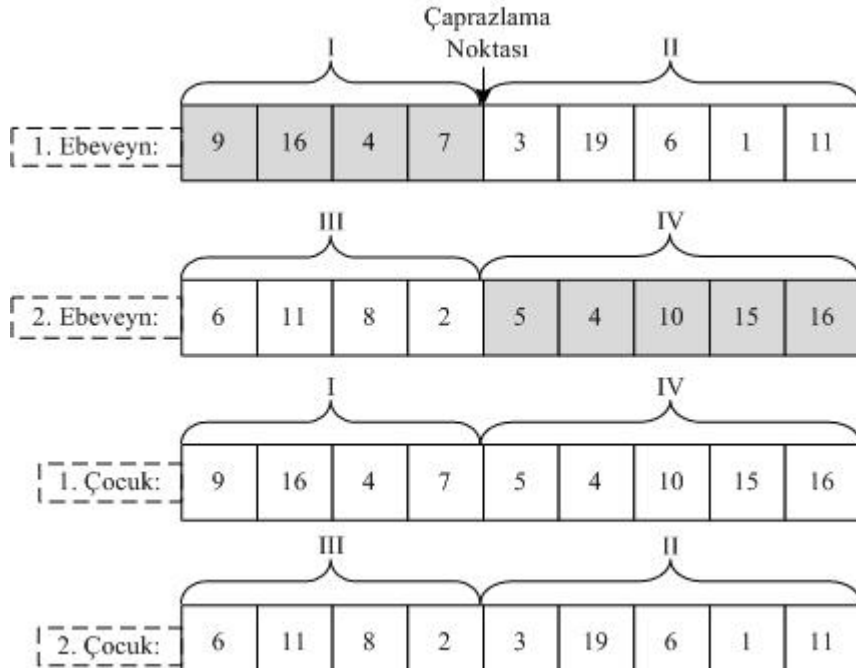
- Tek Noktalı Çaprazlama: Şekil 3.1’de görüldüğü gibi kromozom uzunluğundan küçük ya da eşit bir nokta çaprazlama noktası olarak rastgele seçilir. Çaprazlanmak için seçilen iki kromozomda, bu noktadan önceki genler sabit bırakılarak, bu noktadan sonraki genler ise iki kromozom arasında değiştirilir.

Şekil 3.1’de verilen örnek çaprazlamada, genlerin bir ve yirmi arasındaki tamsayı değerlerden oluştuğu varsayılarak, dokuz gen uzunluğundaki kromozomlar üzerinde işlem gösterilmiştir.

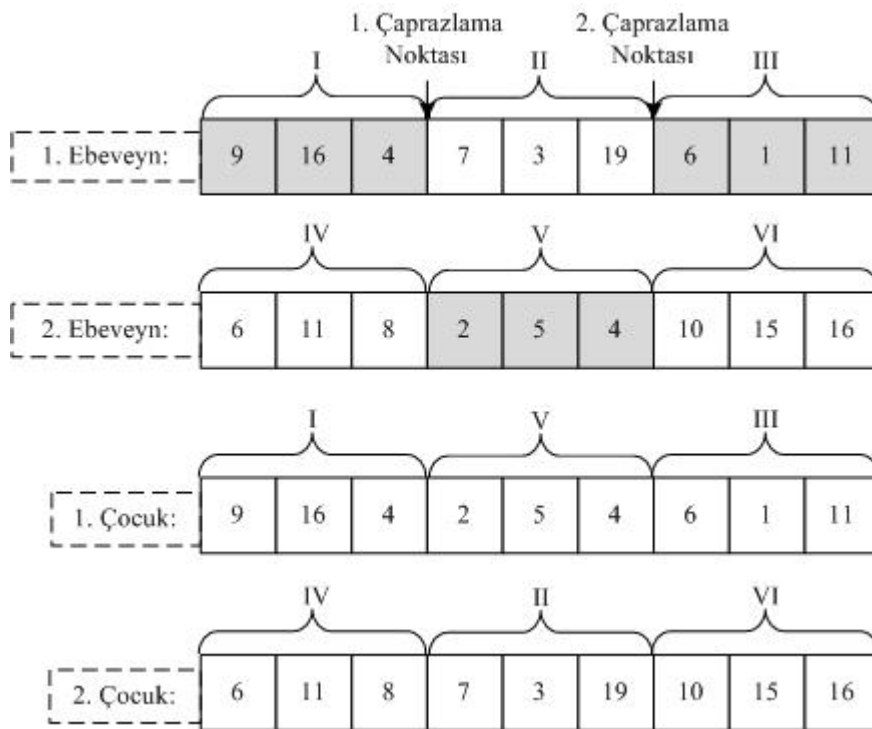
- İki Noktalı Çaprazlama: Şekil 3.2’de görüldüğü gibi kromozom uzunluğundan küçük ya da eşit iki nokta çaprazlama noktaları olarak rastgele seçilir. Bu iki nokta arasındaki genler kromozomlar arasında değiştirilirken, diğer genler ise sabit bırakılır.

Şekil 3.2’de verilen örnek çaprazlamada, Şekil 3.1’de kullanılan örnek ebeveynlerin aynısı kullanılmıştır.





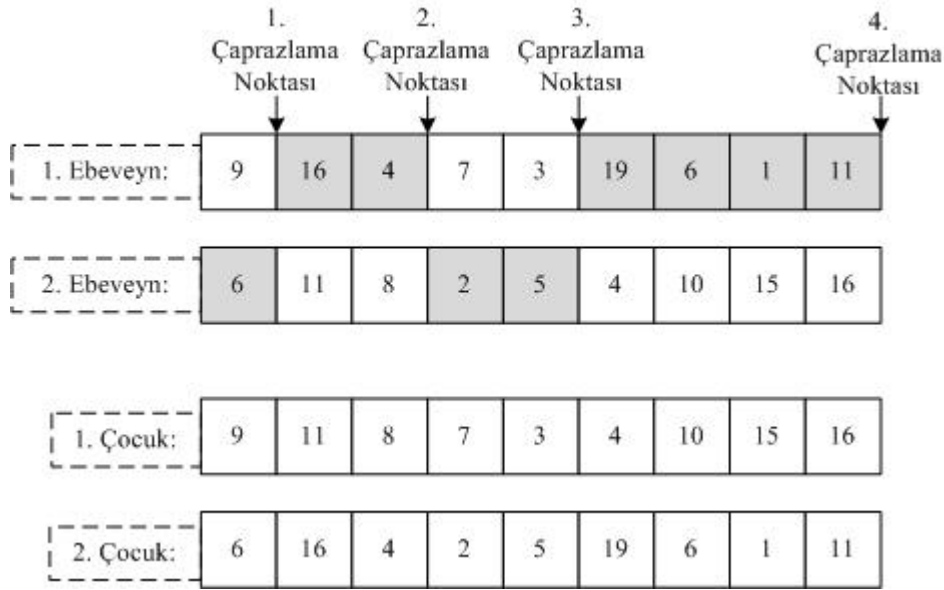
Şekil 3.1 Tek noktalı çaprazlama işlemi



Şekil 3.2 İki noktalı çaprazlama işlemi

- Çok Noktalı Çaprazlama: Şekil 3.3'de görüldüğü gibi kromozom uzunluğundan küçük ya da eşit çok sayıda nokta çaprazlama noktası olarak rastgele seçilir. Sırasıyla, tek ve çift çaprazlama noktaları arasındaki genler kromozomlar

arasında değiştirilirken, çift ve tek çaprazlama noktaları arasındaki genler sabit bırakılır.

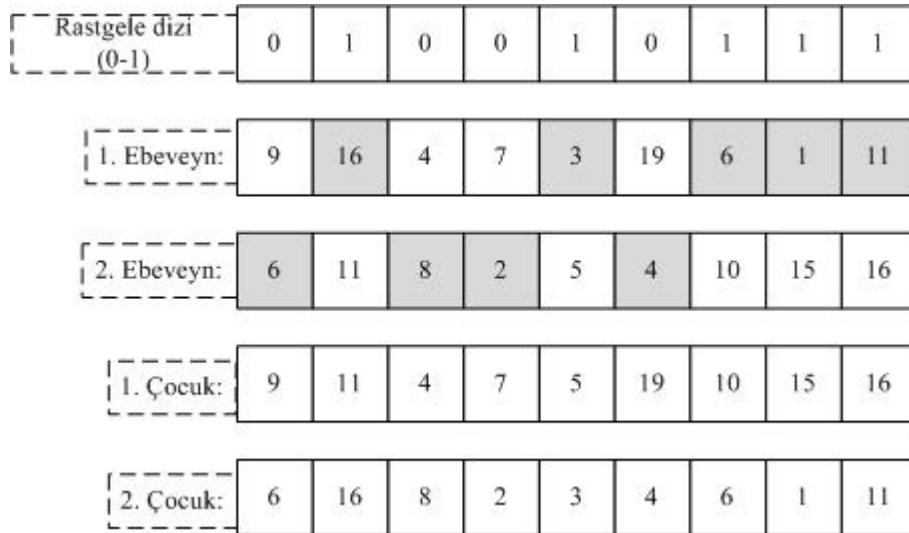


**Şekil 3.3** Çok noktalı çaprazlama işlemi

Şekil 3.3’de Şekil 3.1’de kullanılan örnek ebeveynlerin aynısı kullanılmış, çok noktalı çaprazlamaya örnek olarak dört adet çaprazlama noktası seçilerek işlem gösterilmiştir.

- Uniform Çaprazlama: Çaprazlanmak için seçilen iki kromozomda, bir kromozomun her bir geninin diğer kromozomdaki karşılığı olan gen ile değiştirilmesi olasılığı 0,5 belirlenmiştir (Syswerda 1989). Şekil 3.4’de gösterildiği gibi kromozomlardaki gen sayısı uzunluğunda rastgele sıfır veya bir değerlerinden oluşan bir dizi oluşturularak, sıfır değerine karşılık gelen genler sabit tutularak bir değerine karşılık gelen genler kromozomlar arasında değiştirilir.

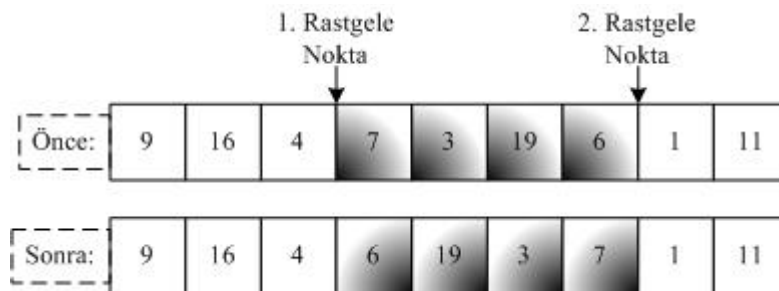
Şekil 3.4’de Şekil 3.1’de kullanılan ebeveynlerin aynısı kullanılmıştır.



Şekil 3.4 Uniform çaprazlama işlemi

### 3.5.2 Tersine çevirme operatörü

Tersine çevirme bir çeşit yeniden sıralama tekniğidir. Şekil 3.5’de görüldüğü gibi bir kromozomdaki rastgele seçilen iki nokta arasındaki genlerin sıralanışını tersine çevrilmesi işlemidir (Holland 1975). Biyolojik işlemden esinlenen bu operatör ile ilgili yapılan araştırmalarda genel olarak genetik algoritmalarda iyi sonuçlar verdiği rapor edilmemiştir.



Şekil 3.5 Tersine çevirme işlemi

### 3.5.3 Mutasyon operatörü

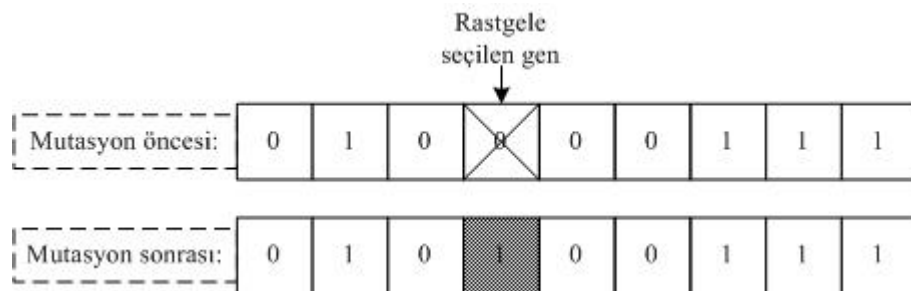
Mutasyon operatörü olası tüm kromozomların ulaşılabilir olmasını sağlamaktadır. Örneğin bir kromozomun ilk geninin 1 ve 20 arasındaki her tamsayı değer olabileceğini varsaydığımızda, rastgele üretilen başlangıç popülasyonundaki hiçbir kromozomun ilk geninin 6 değerine sahip olmadığı düşünülürken, çaprazlama veya tersine çevirme operatörleri ile hangi kombinasyon uygulanırsa uygulansın, arama popülasyondaki

kromozomlarla kısıtlanacağı için 6 değerine hiçbir zaman ulaşamaz. İşte mutasyon operatörü bu noktaya ulaşmada çeşitliliği sağlamak için gereklidir.

Mutasyon operatörünün tanımı farklı yazarlar tarafından farklı yapılmıştır. Goldberg'e göre mutasyon operatörünü uygulamaya bir kere karar verildiğinde o gen mutlaka mutasyona uğramalıdır (Goldberg 1989). Davis'e göre ise mutasyon uygulamak için seçilen gende değişiklik yapıp yapılmayacağına rastgele karar verilebilir (Davis 1991).

Mutasyon operatörlerinden bazıları aşağıda örnekleriyle açıklanmıştır:

- Bit çevirme (Flip Bit) operatörü: Şekil 3.6'da gösterildiği gibi kromozomdan rastgele seçilen bir veya daha fazla genin bit değeri sıfır ise bir, bir ise sıfıra dönüştürülmektedir. Bu mutasyon operatörü ancak ikili kodlanmış gen sistemine uygulanabilmektedir.

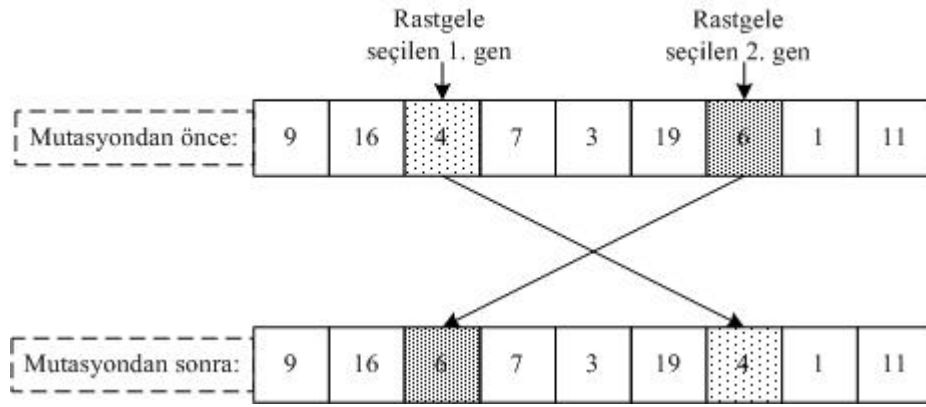


**Şekil 3.6** Bit çevirme operatörü

Şekil 3.6'da gösterilen örnekte sıfır ve bir bit değerlerinden oluşan gen sistemi üzerinde yalnızca bir genin mutasyona uğradığı varsayılmıştır.

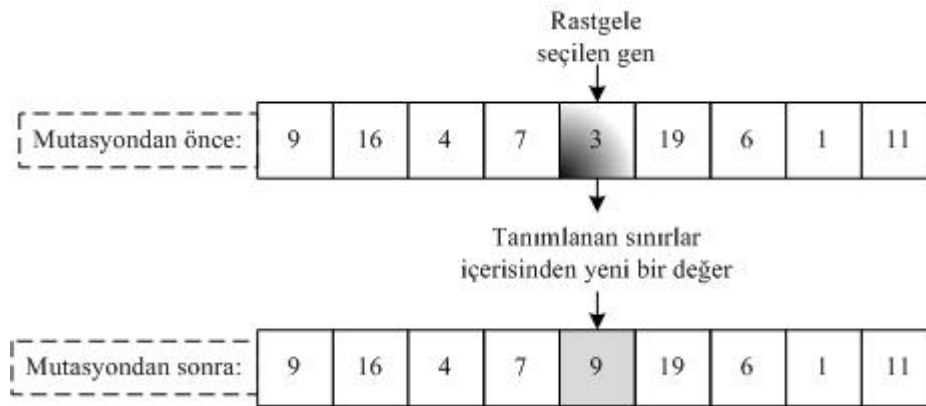
- Değiş tokuş (Swap) operatörü: Şekil 3.7'de gösterildiği gibi mutasyon uygulanmak için belirlenen kromozomda rastgele seçilen iki gen yerlerini değiş tokuş etmektedirler.

Şekil 3.7'de kromozomlardaki gen sisteminin tamsayı değerlerden oluştuğu kabul edilmiştir. Aynı operatör sıfır ve bir bit değerlerinden oluşan gen sistemine de uygulanabilmektedir.



**Şekil 3.7** Değiş tokuş operatörü

- Yerine koyma (Replacement) operatörü: Şekil 3.8’de görüldüğü gibi mutasyona uğratılmak için belirlenen kromozomda rastgele seçilen bir gen, tanımlanan sınırlar içerisindeki değerlerden herhangi bir değer ile değiştirilmektedir. Şekil 3.8’deki örnek için 1 ile 20 arasındaki tamsayı değerler tanımlanan sınırları belirlemiştir. Bu operatör ikili bitlerden oluşan gen sistemine uygulanamamaktadır.



**Şekil 3.8** Yerine koyma operatörü

### 3.5.4 Göç operatörü

Genetik algoritma ile aynı anda farklı popülasyonlarda çalıştırarak bir popülasyondan diğerlerine kromozom transfer edilmesine göç işlemi denir. Göç alan popülasyondaki en kötü uygunluk değerine sahip kromozom yok edilerek popülasyon büyüklüğünün sabit kalması sağlanır.

Doğanın optimizasyon konusunda sunduğu bir örnek olan genetik algoritmalarda amaç fonksiyonunun büyük önem arz etmektedir. En iyiye ulaşması için doğru kontrol

edilmesi gerekmektedir. Kullanılan parametreler ve kurulan model genetik algoritmanın performansı üzerinde doğrudan etkilidir. Dördüncü bölümde geliştirilen öğrenci başarısına odaklı sınav çizelgeleme modeli anlatılmıştır.

#### 4. ÖĞRENCİ BAŞARISINA ODAKLI SINAV ÇİZELGELEME MODELİ

Genetik algoritma modelinde, içindeki her kromozomun bir aday çözüm olduğu nüfus, çaprazlama ve mutasyon gibi genetik operatörlerin etkisi altında sürekli evrim geçirir. Her bireyin hayatta kalması bir uygunluk fonksiyonuna göre belirlenerek, daha iyi gen kombinasyonlarından oluşan kromozomların sonraki nesillerde varlığını sürdürebilme şansları daha fazla iken, daha kötü gen kombinasyonlarına sahip kromozomlar ise seçim olasılıklarına bağlı olarak evrim sürecinde kaybolurlar. Belirlenen sonlandırma koşulu sağlanıncaya kadar devam eden evrim sürecindeki en son nesilde en iyi gen kombinasyonuna sahip kromozom ulaşılan en iyi çözümü temsil eder.

Genetik algoritmalar etkili arama araçları olmakla beraber fazla sayıda parametreye karşı oldukça duyarlıdır. Parametrelerin etkisini küçümsemek uzun süreli denemelere rağmen çözümlerde iyileşme gerçekleşmemesine neden olabilir. Bazen programlama hatalarına veya yanlış seçilen parametrelere rağmen genetik algoritmalar etkili çözümler üretebilse de, bazı durumlarda da durum tam tersine dönüşebilmektedir. Bu noktada, GA geliştiricisinin evrimsel işlemleri yöneten parametre tasarımlarına gereken özeni göstermesi fazlasıyla önemli olmaktadır.

Bu bölümde, mevcut çalışmada bir üniversitedeki sınav çizelgeleme problemlerinin çözümünde kullanılmak üzere geliştirilen genetik algoritma modeli anlatılmaktadır.

##### 4.1 Temsil

Genetik algoritmalar, optimizasyon problemlerini çözmek için kullanıldığında bir çözüme karşılık gelen kromozomun nasıl gösterileceği en önemli adım olarak öne çıkmaktadır. Çaprazlama ve mutasyon operatörlerinin başarısı kromozom yapısı ile doğrudan ilgilidir.

Geliştirilen öğrenci başarısına odaklı sınav çizelgeleme modelinin gerçek verilerle test edilebilmesi için Pamukkale Üniversitesi Mühendislik Fakültesi 2007-2008 Güz

dönemi verileri evrensel kısıtlar gözetilerek kullanılmıştır. Pamukkale Üniversitesi'nde farklı bölümlerin aynı anda yapmak zorunda oldukları ortak ders kavramı, diğer üniversitelerde de bulunan genel bir kısıt olarak öne çıkmaktadır. Bu çalışmada kurulan modelde, tüm bölümlerin birlikte aldığı ortak zorunlu dersler ile herhangi iki bölümün ortak aldığı dersler de bu kavram içinde değerlendirilmektedir.

Bir fakültede  $D$  sayıda bölüm olduğu ve her bölüme ait 4 farklı müfredat dönemi olduğunu kabul edelim. Fakülte'deki bir sınav döneminde yapılacak olan toplam sınav sayısı  $E$  ve sınavların yerleşebileceği boş bir zaman çizelgesinde  $TS$  adet slot olduğu varsayıldığında; arama uzayının çok büyük olduğu görülmektedir. Sınavların paralel olarak yapılması olasılığı olduğundan, arama uzayının büyüklüğü  $TS^E$ 'ye kadar çıkabilir. Bu büyüklükteki bir problemin karmaşıklığı kurulan modele göre değişiklik kazanacaktır.

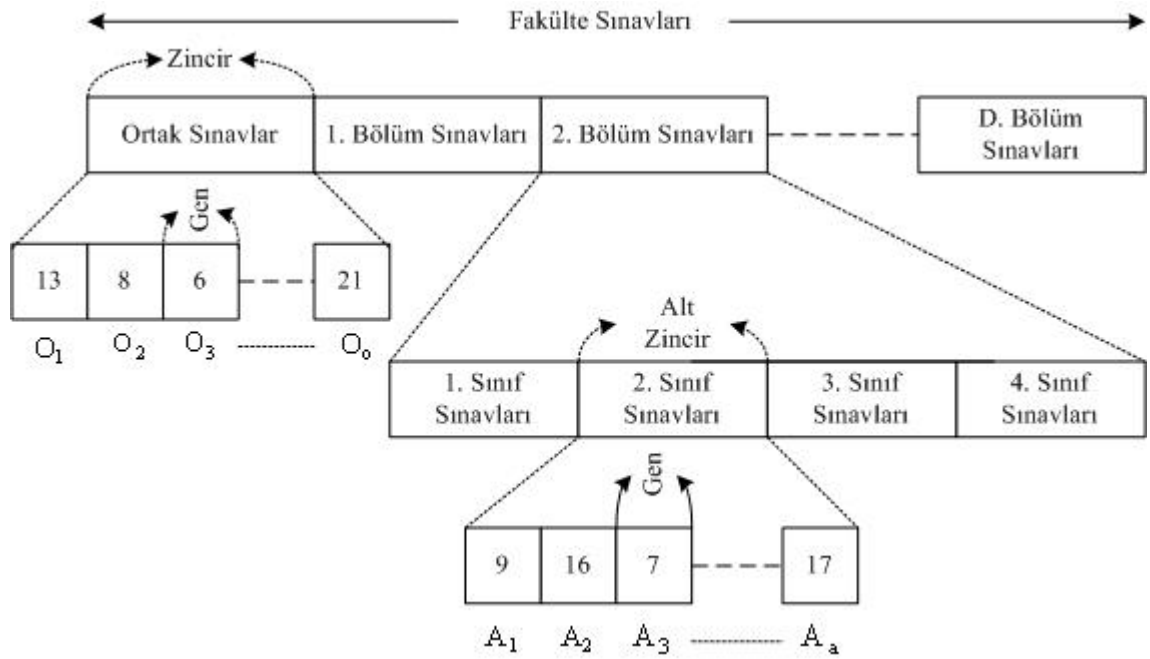
Şekil 4.1'de görüldüğü gibi, aday çözüm olan kromozomlardaki her gen, ilgili sınavın yerleştirildiği zaman dilimine karşılık gelmektedir. Fakültede yapılması gereken ortak sınavlar ilk zincir olarak kromozom yapısına yerleştirilmiştir. Fakülte'deki her bölüme ait olan özel sınavlar sırasıyla ikinci zincirden başlayarak yapıya eklenmiştir. Fakülte'deki ortak sınavlar ile bölümlere ait her sınıfın gen grupları birleşerek bir bireyi oluşturur.

Sınav periyodundaki gün sayısı ve her günde sınav için kullanılacak zaman dilimi sayısı, kromozomdaki genlerin alabileceği tamsayı değer aralığını belirlemektedir. 9 günlük bir sınav periyodundaki her günde 4 farklı zaman dilimi olduğu varsayıldığında, atanabilecek toplam zaman dilimi sayısı 36 olmaktadır. Öyleyse, kromozomdaki genler ancak  $[1, 36]$  aralığındaki tamsayı değerlerden oluşabilir.

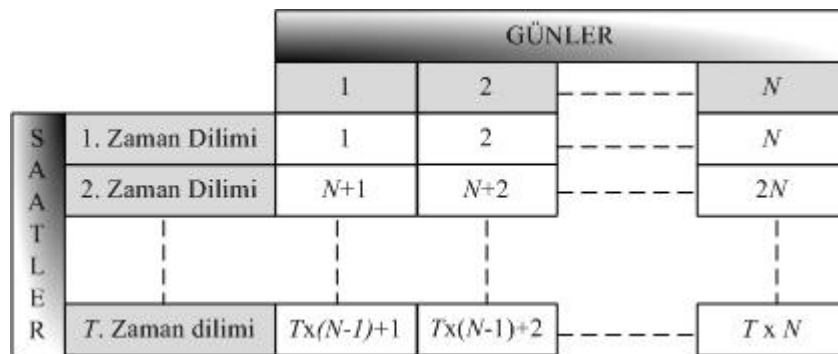
Şekil 4.2'de sınav için ayrılan  $N$  sayıda gün ve her günde ayrılan  $T$  sayıda zaman dilimi bulunduğu durumda zaman dilimlerinin çizelge üzerindeki yerleşimi gösterilmiştir.

Şekil 4.2'de verilen sembolik gösterim bu çalışmada kullanılan gerçek verilere uyarlandığında, belirlenen 9 günlük sınav dilimindeki her günün 4 farklı zaman dilimine bölündüğü durumda oluşan genlerin çizelge üzerine yayılımı Şekil 4.3'de gösterilmiştir.

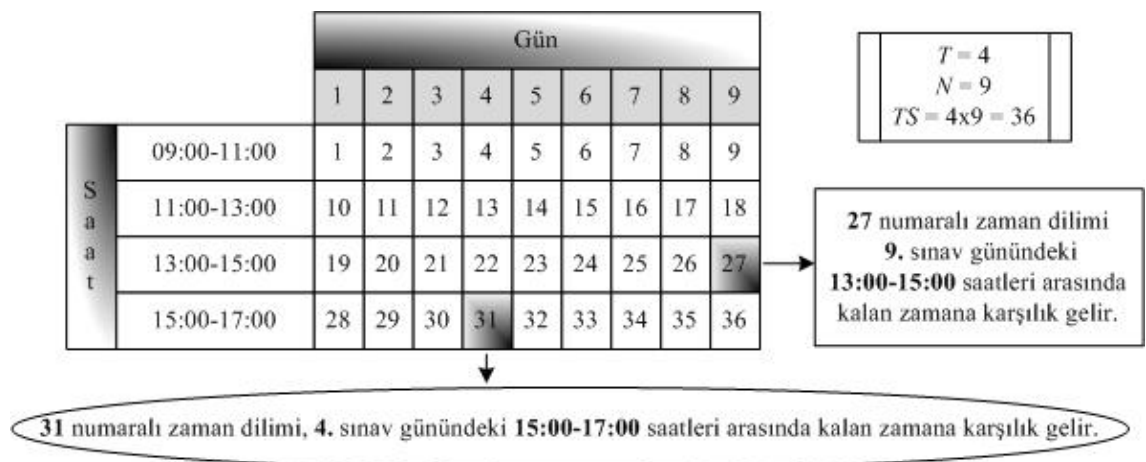




Şekil 4.1 Genetik Algoritmada sınav çizelgeleme problemine aday çözümün kromozom olarak temsili



Şekil 4.2  $N$  sayıda gün ve her günde  $T$  sayıda zaman dilimi için sembolik gösterim



Şekil 4.3 Kromozomdaki genlerin çizelge üzerindeki yayılımı

Şekil 4.1’de örnek olarak gösterilen bir kromozoma ait kısmi genlerin çizelge üzerindeki anlamı Şekil 4.4’de gösterilmiştir.

		Gün								
		1	2	3	4	5	6	7	8	9
S a a t	09:00-11:00						O <sub>3</sub>	A <sub>3</sub>	O <sub>2</sub>	A <sub>1</sub>
	11:00-13:00				O <sub>1</sub>			A <sub>2</sub>	A <sub>3</sub>	
	13:00-15:00			O <sub>0</sub>						
	15:00-17:00									

Şekil-9’daki A<sub>1</sub> dersinin yerleştiği zaman dilimi

**Şekil 4.4** Şekil 4.1’deki genlerin çizelge üzerindeki anlamı

ZD ilgili genin karşılık geldiği zaman dilimi,  $N$  toplam gün sayısı,  $B$  bölüm,  $K$  ise kalan olmak üzere, bir genin çizelge üzerindeki uygun yerleşiminin belirlenmesinde kullanılan ifade Şekil 4.5’de açıklanmıştır. Bu ifadede kalan hangi güne karşılık geldiğini gösterirken, bölüme bir eklendiğinde de o gündeki kaçınıcı zaman diliminin kullanılacağı gösterilmektedir.

$$\begin{array}{r} ZD \\ \hline N \\ \hline B \\ \hline K \end{array} \begin{array}{l} \nearrow \text{toplam gün sayısı} \\ \rightarrow +1 = (B+1) \text{ . zaman dilimi} \\ \text{gün} \end{array}$$

$$\begin{array}{r} 13 \\ \hline 9 \\ \hline 1 \\ \hline 4 \end{array} \begin{array}{l} \nearrow \text{toplam gün sayısı} \\ \rightarrow +1 = (2) \text{ . zaman dilimi} \\ \text{gün} \end{array}$$

**Şekil 4.5** Bir genin çizelge üzerindeki uygun yerleşimin belirlenmesi

## 4.2 İlk Nüfusun Oluşturulması

Tablo 4.1’de verilen fonksiyon ile başlangıç popülasyon sayısı kadar kromozom rastgele belirlenerek ilk nüfus oluşturulur. Kromozomlara rastgele değerler atanması Tablo 4.2’de pseudo kodu verilen fonksiyon ile yapılmaktadır. Rastgele kromozom oluşturulması fonksiyonu ilk popülasyonun yaratılması fonksiyonu tarafından başlangıç popülasyon sayısı kadar çağırılmaktadır.

Başlangıç popülasyonu sayısı, genel popülasyon sayısının iki katı büyüklüğünde olacak şekilde ayarlanmıştır. Bu şekilde, rastgele oluşturulan ilk popülasyonun en iyi değerlere sahip olan yarısı genel popülasyona dâhil edilerek evrim süreci başlatılır (Kulak vd 2007).

Tablo 4.1 ve Tablo 4.2’de belirtildiği gibi; *bps* başlangıç popülasyonu sayısı, *zs* zincir sayısı, *zu* zincir uzunluğu olmak üzere, ilk popülasyonu yaratmak için gereken işlem sayısı (*ipyf*) 4.1 nolu denklem ile hesaplanabilmektedir.

$$ipyf = 4 \times bps \times zu \times zs + bps \times zs + 2 \times bps + 1 \quad 4.1$$

İşlem sayısı hesaplanırken, bir satırda yapılan dört işlem ya da atama işlemleri toplam bir işlem olarak kabul edilmiştir. Toplam işlem sayısı döngülerin ve alt fonksiyonların çağırılma sayısı ile doğru orantılı olmaktadır. Tablo 4.1 ve Tablo 4.2’de her işlemin, döngünün ve fonksiyonun yanında işlem sayısı belirtilmiştir. Tablo 4.2’deki rastgele kromozom oluşturulması fonksiyonu, Tablo 4.1’deki ilk popülasyonu tarafından döngü içinde başlangıç popülasyon sayısı kadar çağırıldığı için alt fonksiyonun toplam işlem sayısı başlangıç popülasyon sayısı ile çarpılarak ana fonksiyonun işlem sayısına eklenmektedir. Tablo 4.2’de verilen rastgele kromozom oluşturulması fonksiyonundaki toplam işlem sayısı (*rkof*) ise 4.2 nolu denklem ile hesaplanabilir.

Geliştirilen algoritmaya ait kodlar, tez boyunca evrensel pseudo kod standartlarına uygun olarak verilmiştir. Bu kodlama standardı ile fonksiyonların okuyan herkes tarafından rahatlıkla anlaşılabilmesi hedeflenmiştir.

$$rkof = 4 \times zu \times zs + zs + 1 \quad 4.2$$

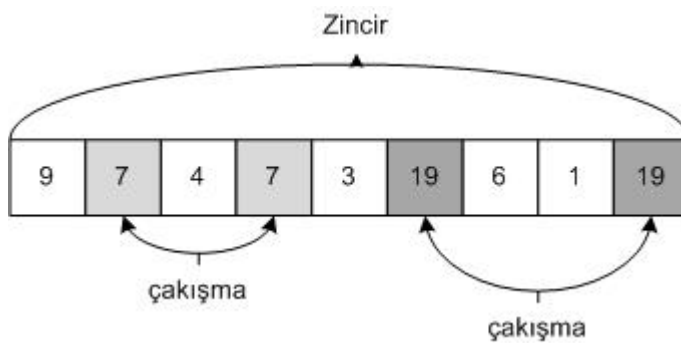
**Tablo 4.1** İlk popülasyonun yaratılması fonksiyonu

Pseudo Kod	İşlem Sayısı
<b>FUNCTION generateIniPop</b> SET numberOfChoromosomes to 0. REPEAT CALL <b>createRandomChorosome</b> with chainsArray, maxTimeSlot RETURNING Chorosome INCREMENT numberOfChoromosomes UNTIL numberOfChoromosomes = initialPopulationSize RETURN IniPop <b>END FUNCTION</b>	$bps \times ((4 \times zu + 1) \times zs + 2) + 1$ 1 $bps \times ((4 \times zu + 1) \times zs + 2)$ $(4 \times zu + 1) \times zs + 1$ 1

**Tablo 4.2** Rastgele kromozom oluşturulması fonksiyonu

Pseudo Kod	İşlem Sayısı
<b>FUNCTION createRandomChromosome</b> SET gene to zero. FOR each chain in the chainsArray SET randomTimeSlotsArray to maxTimeSlot FOR each gene in the chain SELECT a timeSlot randomly from randomTimeSlotsArray INCREMENT gene SET gene of chromosome to selected timeSlot DELETE selected timeSlot from randomTimeSlotsArray ENDFOR ENDFOR RETURN	$(4 \times zu + 1) \times zs + 1$ 1 $(4 \times zu + 1) \times zs$ 1 $4 \times zu$ 1 1 1 1 1

Bir sınıfa ait bütün sınavların farklı zaman dilimlerinde yapılması gerekliliği nedeniyle, ilk nüfusu oluşturan bütün kromozomlardaki her zincir kendi içinde farklı genlere sahip olacak şekilde ayarlanmak zorundadır. Şekil 4.6'da görüldüğü gibi, eğer bir zincire ait her gen birbirinden farklı değerlere sahip olmazsa bariz çakışmalar meydana gelecektir. Bu sebeple oluşturulan yapıda, kromozomdaki zincirlerin her zaman birbirinden farklı genlerden oluşması garanti edilerek bu bariz çakışmaya fırsat bırakılmamıştır. Tablo 4.2'de de görüldüğü gibi rastgele oluşan bir zincirde kullanılan bir gen değerinin bir başka gen değerinde kullanılmaması için değer dizisinden silinmektedir.

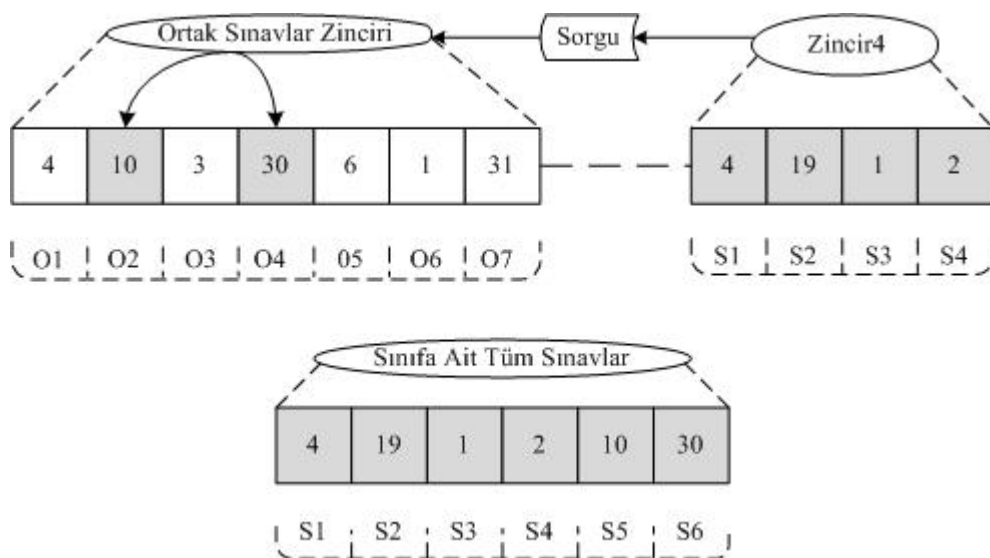
**Şekil 4.6** Bir zincirdeki genlerin değerlerinin çakışması

### 4.3. Uygunluk

Uygunluk fonksiyonu, herhangi bir kısıta uymayan bir atamanın cezalandırılması yöntemiyle yapılır. Oluşturulan kromozom yapısında farklı bölümlerin farklı sınavları olması nedeniyle cezalandırılması gereken atamalar zincir bazında değerlendirilmeli ve ayrıca ortak yapılma zorunluluğu olan sınavlar da dikkate alınmalıdır. Bu çalışmada

kurulan model ile öğrenci başarısını maksimize etmeye yönelik bir yapı ortaya koyulduğu için her öğrencinin durumu dikkate alınmalıdır. Veritabanındaki her öğrencinin durumunu ayrı ayrı değerlendirmek yerine; bir zincir, bir bölümdeki alt sınıfa ait sınavları temsil ettiğine göre, ortak sınavlar zincirindeki o sınıfa ait ortak sınavları da bu zincire dâhil ederek sınıftaki tüm öğrencilerin durumunun birlikte değerlendirilmesi sağlanmıştır.

Şekil 4.7’de gösterildiği gibi, kromozom yapısındaki dördüncü zincir, bir alt sınıfa ait özel sınavları içermektedir. Ancak bu sınıfın diğer bölüm ya da sınıflarla ortak olarak alması gereken sınavlar bulunabilir. Bu sınavların hepsi ortak sınavlar zincirinde bulunmaktadır. Uygun sorgu çalıştırılarak; O2 ve O4 sınavlarının dördüncü zincirin ortağı olduğu ortaya çıkarılmıştır. Bu şekilde O2 ve O4 sınavları da işgal ettikleri zaman dilimleri ile birlikte dördüncü zincire eklenmiştir.



**Şekil 4.7** Bir sınıfa ait tüm sınavların yerleşim dizisi

Hangi zincirin, hangi ortak sınavlara dâhil olduğunu Tablo 4.3’de verilen doğrudan ortaklar matrisinden yararlanarak bulunmaktadır. Örneğin, 15 optik kodlu “Olasılık Teorisi” sınavı üçüncü ve onikinci zincirlerin birlikte aldığı ortak bir sınav olduğu anlaşılmaktadır. Matristeki sıfır değerleri matrisi ikinci boyutta eşitlemek için eklenmiştir. Bu sayede hücre yapısı kullanılmasına gerek kalmadan iki boyutlu matris ile değerler tutulabilmektedir.

Tablo 4.3’de ortak sınavlar zincirine bağlı olan zincirler matrisi verilmiştir. Bu matriste aynı ortaklar aynı optik kodu ile bir arada tutulmuş, farklı ortaklar ise farklı

optik kodları ile birbirinden ayrılmıştır. Endüstri Mühendisliği, Elektrik-Elektronik Mühendisliği ve Tekstil Mühendisliği bölümleri 13 optik kodlu “Genel Kimya” sınavına aynı anda girecekler, Gıda Mühendisliği, İnşaat Mühendisliği ve Jeoloji Mühendisliği bölümleri ise 14 optik kodlu “Genel Kimya” sınavını aynı anda olacaktır. Bu bölümlerin aynı sınava girmeyecek olmaları nedeniyle, aynı anda sınava tabi tutulmalarına gerek yoktur. Bu sebeple, farklı optik kodları ile birbirinden ayrılmış ve aynı zaman dilimine yığılmamaları sağlanmıştır. Kısaca, sadece aynı anda yapılması zorunlu olan sınavlar aynı optik kod içerisinde kümelendirilmiştir.

**Tablo 4.3** Ortak sınavlar zincirine bağlı olan zincirler matrisi

Sınavlar	Optik Kod	Doğrudan Ortaklar (Sınavı alan Zincirler)								
		2	7	11	14	18	22	26	30	
AIIT – I	1	2	7	11	14	18	22	26	30	
Türk Dili – I (En-B-T-J)	2	2	6	14	30	0	0	0	0	
Türk Dili – I (El-G-M-İ)	3	10	18	22	26	0	0	0	0	
Yabancı Dil – I	4	2	6	10	14	18	22	26	30	
Beden Eğitimi (En-B-J)	5	2	6	30	0	0	0	0	0	
Beden Eğitimi (M-İ)	6	22	26	0	0	0	0	0	0	
Beden Eğitimi (El-T-G)	7	10	14	18	0	0	0	0	0	
Genel Matematik –I (En-J)	8	2	30	0	0	0	0	0	0	
Genel Matematik –I (B-El)	9	6	10	0	0	0	0	0	0	
Genel Matematik –I (G-M)	10	18	22	0	0	0	0	0	0	
Diferansiyel Denklemler (G-M)	11	19	23	0	0	0	0	0	0	
Genel Fizik – I (T-İ-J)	12	14	26	30	0	0	0	0	0	
Genel Kimya (En-El-T)	13	2	10	14	0	0	0	0	0	
Genel Kimya (G-İ-J)	14	18	26	30	0	0	0	0	0	
Olasılık Teorisi	15	3	12	0	0	0	0	0	0	
Mekanizma Tekniği	16	16	24	0	0	0	0	0	0	

Tablo 4.3’de verilen doğrudan ortaklar matrisi yalnızca bu sınavlara ortak olan zincirleri tutmaktadır. Ancak alttan ve üstten ders alımları da işin içine girdiğinde bu yapı yetersiz kalmaktadır. Bu yüzden, alttan ve üstten ders alımlarının da kontrol mekanizmasına dâhil edilmiş hali Tablo 4.4’de görülmektedir. Tablo 4.4’deki matris, ortak sınavları alan zincirlere, alttan ve üstten alan zincirlerin de eklenmesiyle ortaya çıkmaktadır. Bu sayede alt ve üst sınıflardan ders alan öğrencilerin durumları da dikkate alınabilmektedir. Örneğin 15 optik kodlu dersi 3 ve 12 numaralı zincirler alıyor olmasına rağmen, bu derslerden kalan öğrenciler olduğu için sırasıyla 4,5 ve 13 numaralı zincirler de ortaklar arasına katılmıştır.

**Tablo 4.4** Ortak sınavlar zincirine bağlı olan tüm zincirler matrisi

Optik Kod	Ortaklar (Sınavı alan Tüm Zincirler)																						
	2	3	4	6	7	8	11	12	13	14	18	19	20	22	23	24	26	27	28	30	31	32	
1	2	3	4	6	7	8	11	12	13	14	18	19	20	22	23	24	26	27	28	30	31	32	
2	2	6	7	14	15	30	31	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	10	11	18	19	20	22	23	24	26	27	28	0	0	0	0	0	0	0	0	0	0	0	
4	2	6	7	10	11	12	14	15	18	19	20	22	23	24	26	27	28	30	31	32	0	0	
5	2	6	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	22	23	24	26	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	10	11	12	14	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	2	3	30	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	6	7	10	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	18	19	22	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	19	20	21	22	23	24	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	14	15	26	27	30	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	2	3	10	11	14	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	18	19	26	27	30	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	3	4	5	12	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	15	16	17	23	24	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Uygunluk hesaplanmadan önce,Tablo 4.5’de verilen fonksiyon ile uygunluğu hesaplanacak olan dizi her zincir için tek tek belirlenmektedir. Altan ve üstten ders alımlarının ve ortak derslerin de dikkate alınarak oluşturulan her dizinin uygunluk değeri ve çakışma matrisi Tablo 4.6’da verilen fonksiyon ile hesaplanmaktadır.

*oz1*: Ortak zincirin satır sayısı,

*oz2*: Ortak zincirin sütun sayısı,

*udu*: Uygunluk dizisinin uzunluğu,

*zu*: Zincir uzunluğu, olmak üzere;

Tablo 4.7’de verilen kümülatif toplamın bulunması fonksiyonunda toplam işlem sayısı (*ktbf*) 4.3 nolu denklem ile,Tablo 4.8’de verilen uygunluk değerinin hesaplanması fonksiyondaki toplam işlem sayısı (*udhf*) ise 4.4 nolu denklem ile hesaplanabilmektedir. Bu her iki fonksiyonu da kendi içinden çağıran Tablo 4.5’deki uygunluğu hesaplanacak diziyi belirleyen fonksiyon ile yapılan toplam işlem sayısı (*uhdbf*) ise 4.5 nolu denklem ile hesaplanabilmektedir.

$$ktbf = 2 \times zu^2 + zu \quad 4.3$$

$$udhf = 8 \times udu^2 - 8 \times udu + 1 \quad 4.4$$

$$uhdbf = 8 \times udu^2 + 2 \times zu^2 + 3 \times oz1 \times oz2 - 8 \times udu + 3 \times zu + 3 \quad 4.5$$

**Tablo 4.5** Uygunluğu hesaplanacak dizinin belirlenmesi

Pseudo Kod
<pre> <b>FUNCTION determineFitnessArray</b>  SET fitvalue to zero. SET confM to NULL  CALL <b>computeCumulative</b> with chainsArray RETURNING qChainsArray  FOR each chain in the chainsArray   FOR each timeSlot in the current chain     Append timeSlot to FitnessArray     Append difficulty of the related exam to FitnessArray   ENDFOR    FOR each row of SharedArray     FOR each column of SharedArray       IF SharedArray position is occupied THEN         Append timeslot to FitnessArray         Append difficulty of the related exam to FitnessArray       ENDIF     ENDFOR   ENDFOR  CALL <b>computeFitness</b> with FitnessArray, nodays, confM RETURNING fitness AND confM ADD fitness to fitvalue SET fitnessARRAY to NULL  <b>ENDFOR</b> </pre>

Tablo 4.5’de verilen fonksiyon ile uygunluk değeri hesaplanacak olan sınavların zaman dilimleri ve zorluk dereceleri okunarak Şekil 4.8’de gösterildiği gibi Tablo 4.6’da verilen fonksiyona gönderilmektedir.

Zaman Dilimi	4	10	3	30	6	1	31
Sınav	E1	E2	E3	E4	E5	E6	E7
Zorluk Derecesi	1	6	10	10	4	2	8

**Şekil 4.8** Sınavların zaman dilimi ve zorluk derecesi gösterimi

Şekil 4.8’de gösterildiği gibi her sınavın atandığı bir zaman dilimi ve her sınavın da bir zorluk derecesi bulunmaktadır. Öğrenci başarısını maksimize etme iddiasında olan modelde, zorluk derecesi yüksek olan derslerin birbirinden mümkün olduğu kadar uzak zaman dilimlerine atanmasına çalışılmaktadır. Bunu sağlamak için zaman dilimleri farkı katsayılarına zorluk dereceleri katsayıları da çarpan olarak amaç fonksiyonunda kullanılmak üzere eklenmiştir.



$AL$ : Uygunluk değeri hesaplanacak olan dizi uzunluğu,

$DF$ : Zorluk derecesi katsayısı,

$N$ : Sınav dönemindeki toplam gün sayısı,

$t$ : Sınavın atandığı zaman dilimi,

$TN$ : Zaman dilimleri yakınlık katsayısı, olmak üzere;

geliştirilen GA modelinde kullanılan amaç fonksiyonu 4.6 nolu denklemde gösterilmiştir.

$$\begin{aligned} \text{Min } F &= \sum_{i=1}^{AL} \sum_{j=i+1}^{AL} (DF_i + DF_j) \times TN(i, j) \\ TN(i, j) &= \begin{cases} 1000 & \text{eger } t_i = t_j. \\ 2^{4-|(t_i-t_j)/N|} & \text{eger } \text{Mod}(t_i, N) = \text{Mod}(t_j, N). \\ 1 & \text{eger } |\text{Mod}(t_i, N) - \text{Mod}(t_j, N)| = 1. \\ 0 & \text{diğer durumlarda.} \end{cases} \end{aligned} \quad 4.6$$

$i$ . ve  $j$ . sınavların birbirine yakınlıklarını ifade eden  $TN(i, j)$  katsayısı, bu iki zaman diliminin aynı olması durumunda zorunlu bir kısıtın bozulmasını engellemek için 1000 ceza değerini alır. Aynı günde olması durumunda ise, aralarındaki zaman farkı göz önüne alınır. Zaman farkı azaldıkça katsayı derecesi logaritmik olarak artmaktadır. Sınavlar arasındaki gün farkının bir olması durumunda katsayı bire indirgenmiştir. Diğer durumlardaki sınavların birbirlerini etkilemeyeceği varsayılarak sıfır katsayısı verilmiştir. Bir günde, sınav için ayrılan dört zaman dilimi olduğu varsayıldığında, kurulan matematiksel modele göre yakınlık katsayılarının oluşumu Şekil 4.9'da gösterilmiştir.

	Gün farkı = 0				Gün farkı = 1	Gün farkı > 1
Zaman dilimi uzaklığı: $T(i, j)$	0	1	2	3	Bakılmaz.	Bakılmaz.
Yakınlık Katsayısı ( $TN$ )	1000	8	4	2	1	0

**Şekil 4.9** Yakınlık katsayıları gösterimi

Şekil 4.9'da gösterildiği gibi yakınlık katsayılarının uygun koşullara göre eklenerek toplam uygunluk değerini üreten fonksiyon Tablo 4.6'da verilmiştir.

**Tablo 4.6** Uygunluk deęerinin hesaplanması

Pseudo Kod
Function <b>computeFitness</b>  SET fitness to zero.  FOR each timeSlot in the FitnessArray FOR each following timeSlot in the FitnessArray IF timeslots are in the consecutive days THEN ADD both of exam difficulties to fitness INCREMENT related element of confM ELSEIF timeslots are in the same day THEN IF timeslots clash THEN ADD 1000 to fitness INCREMENT related element of confM ELSEIF timeslots are consecutive ADD 8 times of sum of both of exam difficulties to fitness INCREMENT related element of confM ELSEIF there is one time slot break between exams THEN ADD 4 times of sum of both of exam difficulties to fitness INCREMENT related element of confM ELSEIF there is two timeSlot breaks between exams THEN ADD 2 times of sum of both of exam difficulties to fitness INCREMENT related element of confM ENDIF ENDIF ENDFOR ENDFOR  RETURN fitness AND confM

**Tablo 4.7** Kümülatif toplamın bulunması

Pseudo Kod
Function <b>computeCumulative</b>  FOR each element of the Array SET k to zero. WHILE k does not reach to next element of the Array ADD all elements of Array to current element of qArray INCREMENT k ENDWHILE ENDFOR  RETURN qArray ENDFUNCTION

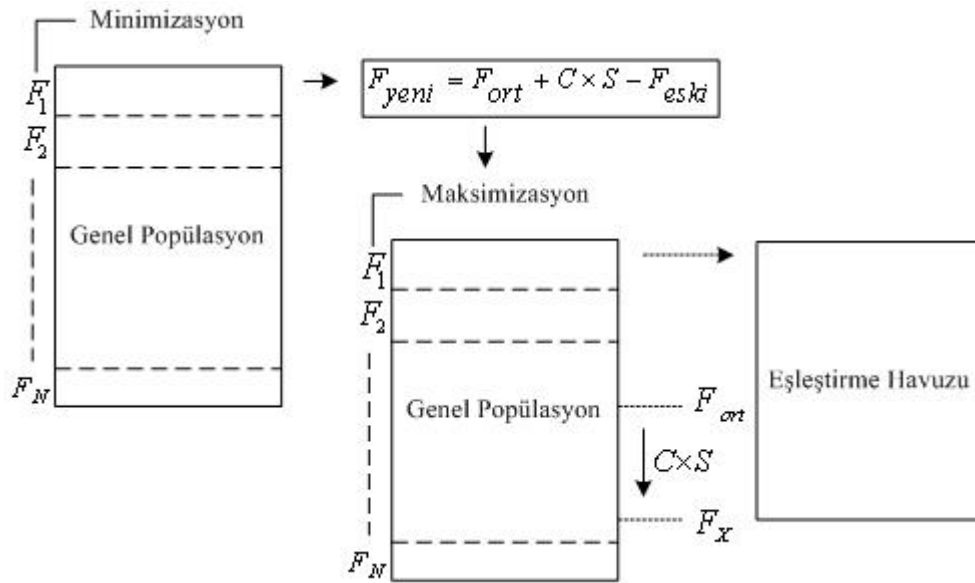
#### 4.4 Kromozomların Eşleştirme Havuzuna Alınması

Her genel popülasyondaki bireyler, çaprazlama işlemine sokulmadan önce uygunluk deęerlerine göre sıralanarak, belli noktadan daha iyi uygunluk deęerine sahip olan

bireyler eşleştirme havuzuna alınır.  $C = 0,1,2$  veya  $3$  olmak üzere, 4.7 nolu denkleme göre uygunluk değerleri yeniden hesaplanarak pozitif uygunluk değerine sahip olan bireylerin eşleştirme havuzuna alınması sağlanır.  $C$  katsayısının sıfıra eşit olması, ortalamanın altındaki her bireyin eşleştirme havuzu dışında tutulması anlamına gelirken, bu katsayı arttıkça ortalamanın altındaki bireyler de eşleştirme havuzuna alınmaktadır.

$$Uygunluk_{yeni} = Uygunluk_{ort} + C \times S \text{ standart Sapma} - Uygunluk_{eski} \quad 4.7$$

Şekil 4.10’da görüldüğü gibi genel popülasyondaki en küçük uygunluk değerine sahip olan kromozom en iyi çözümü temsil ederken,  $F_{yeni} = F_{ort} + C \times S - F_{eski}$  formülü ile en iyi kromozom en büyük uygunluk değerine sahip olmaktadır. Bu şekilde problem ölçeklendirme yapılabilmesi için minimizasyon probleminden maksimizasyon problemine dönüşmektedir. Artık bireyler olasılık kurallarının uygulanmasına hazır hale gelmiştir. Uygunluk değerleri yeniden düzenlenen genel popülasyonda, hangi bireylerin eşleştirme havuzuna alınacağı ortalamanın ne kadar altına inileceği ile bağlantılıdır.  $C$  katsayısının her artışı ortalamanın standart sapma kadar altına inilmesi anlamına gelmektedir. Kullanılan formüle göre, bu noktanın altındaki kromozomların uygunluk değeri negatif olduğu için eşleştirme havuzunun dışına itilmektedirler. Sonuçta, sadece eşleştirme havuzuna alınan bireylere çaprazlama uygulanır. Bu noktada amaç, iyi bireylerin birbirleriyle çaprazlama olasılıkların artırılmasıdır.



Şekil 4.10 Kromozomların eşleştirme havuzuna alınması

- oz1*: Ortak zincirin satır sayısı,  
*oz2*: Ortak zincirin sütun sayısı,  
*ps*: Popülasyon sayısı,  
*udu*: Uygunluk dizisinin uzunluğu,  
*zu*: Zincir uzunluğu; olmak üzere,

Tablo 4.8’de verilen kromozomların eşleştirme havuzuna alınması fonksiyonu ile gerçekleştirilen toplam işlem sayısı (*kehaf*) 4.8 nolu denkleme göre hesaplanabilmektedir.

$$kehaf = (8 \times udu^2 + 2 \times zu^2 + 3 \times oz1 \times oz2 - 8 \times udu + 3 \times zu + 4) \times ps + (ps + 2)^2 \quad 4.8$$

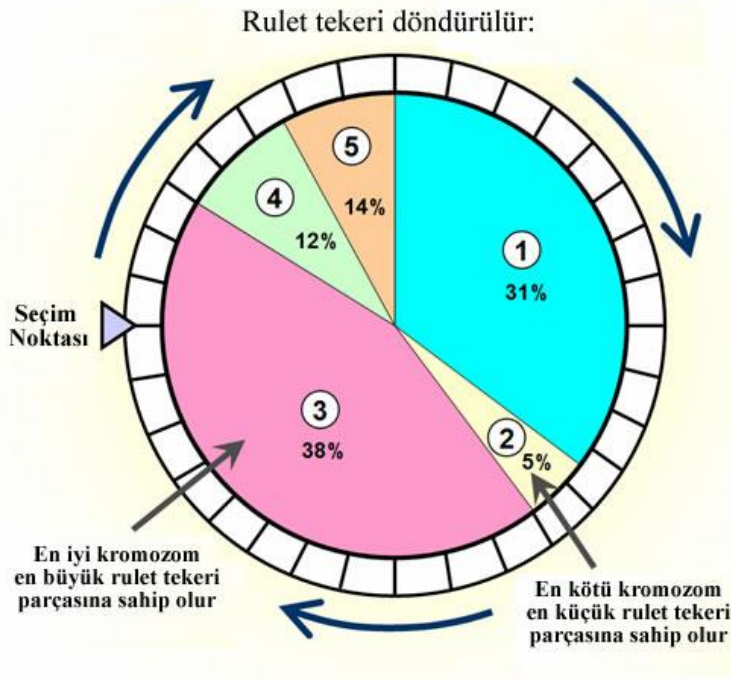
**Tablo 4.8** Kromozomların eşleştirme havuzuna alınması fonksiyonu

Pseudo Kod
<pre> <b>FUNCTION matingPool</b>  SET fsum to zero.  FOR each chromosome in the population     CALL <b>determineFitnessArray</b> with chainsArray, Choromosome, Shared, SharedInfo,     CoursesInfo, nodays RETURNING fitvalue and confM     ADD fitvalue to fsum ENDFOR  SET fave to fsum / popsize SET sd to standard deviation of the fitness values of each choromosome  SORT fitvalue of each chromosome in Pop in ascending order. SET Pool to NULL  FOR each chromosome in the population     SET newfit to fave + sdcoef * sd - fitvalue     IF newfit &gt; 0         APPEND choromosome to Pool     ELSE         BREAK     ENDIF ENDFOR  RETURN Pool <b>ENDFUNCTION</b> </pre>

#### 4.5 Çaprazlanacak Bireylerin Seçilmesi

Eşleştirme havuzuna alınan kromozomlar hesaplanan yeni uygunluk değerleri ile Şekil 4.11’de görülen rulet tekeri seçimine tabi tutulurlar. Rulet tekeri seçimi diğer

seçim yöntemlerine göre daha iyi sonuç verdiği için tercih edilmiştir. Her kromozom, uygunluk değeri ne kadar iyi ise rulet tekerinden o kadar pay almaktadır. Bu şekilde daha iyi olan kromozomların çaprazlama olasılığı artırılmaktadır. Ancak bu, yine de daha kötü olan kromozomların seçilemeyeceği anlamına gelmez. Piyango en kötü kromozoma da vurabilir.



**Şekil 4.11** Rulet tekeri seçimi

İstenilen çocuk sayısının yarısı kadar çift eşleşme için seçilir. Hiçbir kromozom kendisi ile eşleşemez, çünkü bu durumda ne tür bir çaprazlama olursa olsun iyi ya da kötü herhangi bir değişim beklenemez. Her ikisi de farklı kromozom olmak kaydı ile aynı çiftlerin tekrar seçilmesi olasılık dâhilindedir. Tablo 4.9'da verilen çaprazlanacak bireylerin seçilmesi fonksiyonu ile öncelikle eşleştirme havuzundaki her bireyin uygunluk değerlerine göre Şekil 4.11'de görüldüğü gibi ne kadar rulet tekeri parçasına sahip olacağı belirlenir. Rulet tekeri birinci ebeveyn için çevrilerek bir kromozom seçilir. İkinci ebeveyn için ise ilk ebeveyn'den farklı bir kromozom seçilene kadar rulet tekerinin çevrilmesine devam edilir. Bu fonksiyona parametre olarak gönderilen elde edilmesi gereken çocuk sayısını sağlayacak kadar kromozom çiftleri eşleştirilmesi sağlanır.

Şekil 4.11'deki rulet tekerleği beş adet kromozoma sahip bir popülasyon için temsili olarak verilmiştir. Popülasyon büyüklüğü kadar rulet tekeri parçalara ayrılmaktadır.

Tablo 4.9’da verilen çaprazlanacak bireylerin seçilmesi fonksiyonunda; *ics* istenen çocuk sayısı, *ps* popülasyon sayısı olmak üzere, toplam işlem sayısı (*cbsf*) 4.9 nolu denkleme göre hesaplanabilmektedir.

$$cbsf = 2 \times ps^2 + 2 \times ps \times ics + 3 \times ps + \frac{5 \times ics}{2} + 2 \quad 4.9$$

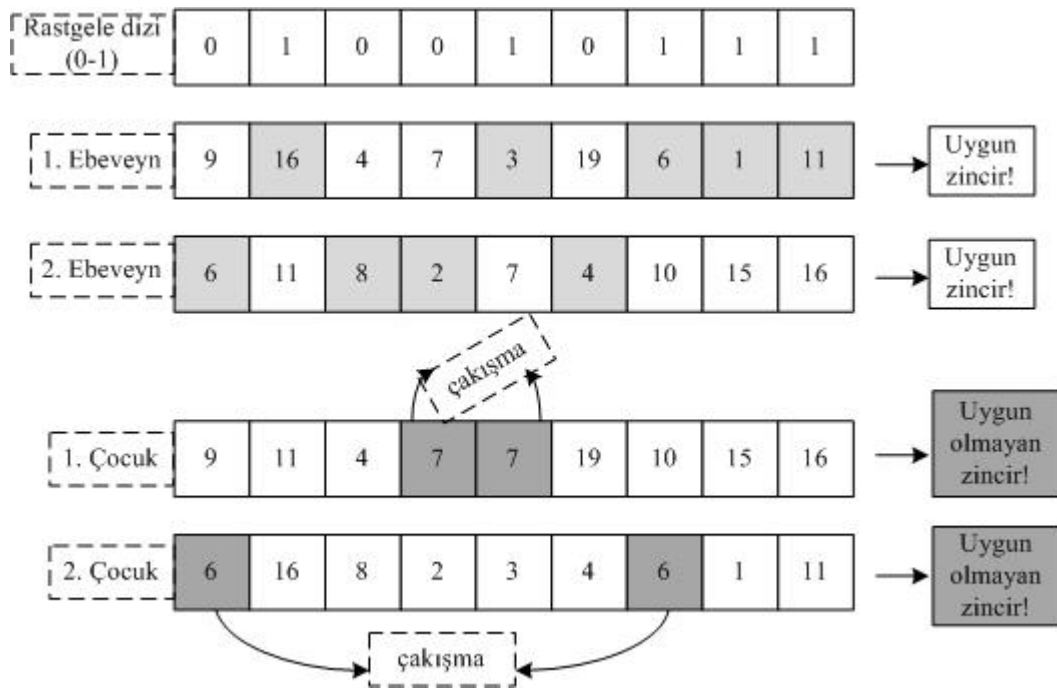
**Tablo 4.9** Çaprazlanacak bireylerin seçilmesi fonksiyonu

Pseudo Kod
<pre> <b>FUNCTION crossWhom</b>  SET fitsum to sum of the fitness values of each chorosome in the Pool.  FOR each chorosome in the Pool     SET Probability of chorosome to fitvalue / fitsum     APPEND Probability to ProbabalityArray ENDFOR  CALL <b>computeCumulative</b> with ProbabilityArray RETURNING qProbabilityArray  SET mating to zero REPEAT     SET randx to a random number between 0 and 1     FOR each probability in the qProbabilityArray         IF randx IS LESS THAN the element of qProbabilityArray             SELECT current element as parent1             BREAK         ENDIF     ENDFOR     SET parent2 to parent1      WHILE parent2 IS EQUAL parent1         SET randx to a random number between 0 and 1         FOR each probability in the qProbabilityArray             IF randx IS LESS THAN the element of qProbabilityArray                 SELECT current element as parent2                 BREAK             ENDIF         ENDFOR     ENDWHILE      APPEND parent1 and parent2 to MatingSet     INCREMENT mating  UNTIL mating IS LESS THAN nooff / 2 RETURN MatingSet <b>END FUNCTION</b> </pre>

#### 4.6 Çaprazlama

Çaprazlama işlemi için üçüncü bölümde anlatılan tek noktali çaprazlama yöntemi seçildiğinde kromozomlar ancak tek noktadan itibaren karıştırılabildiği için gen

çeşitliliği diğer yöntemlere göre zayıf kalmaktadır. Çok noktalı çaprazlama yönteminde ise, bu problemde üretilen çocukların istenmeyen çözümler olması ile karşı karşıya kalındığında kromozomların tamiri uniform çaprazlamaya göre daha zor olmaktadır. Bu sebeple çaprazlama işlemi için uniform çaprazlama yöntemi tercih edilmiştir. Ancak bu yöntemin aynen kullanılması durumunda bazı istenmeyen gen birleşimleri gerçekleşebilmektedir. Bu yüzden bu yöntemin istenmeyen durumlara imkân tanınmayacak şekilde uyarlanması ihtiyacı doğmuştur. Şekil 4.12’de iki uygun zinciri gösteren ebeveynlerin normal uniform çaprazlama sonucu her iki çocukta da istenmeyen gen bileşimlerinin meydana geldiği görülmüştür. Bu şekilde çakışan genlerden oluşan zincirler, zorunlu bir kısıt olan bir öğrencinin aynı anda iki sınava girmemesi kısıtını sağlamadığı için kabul edilemez.

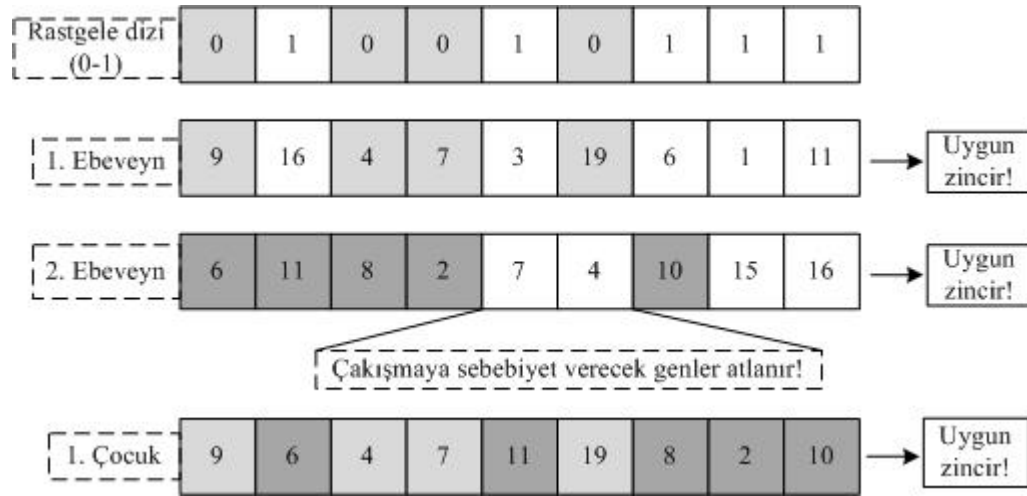


**Şekil 4.12** Normal uniform çaprazlama sonucu oluşan çakışmalar

Şekil 4.12’de görülen çakışmaları engelleyebilmek için uniform çaprazlama işleminin problemde kurulan kromozom yapısına göre uyarlanması gerekmektedir. Geliştirilen kromozom yapısındaki her zincir kendi içinde farklı gen değerlerine sahip olmalıdır. Aksi takdirde aynı sınıfa ait olan sınavların aynı anda yapılması anlamına gelen zorunlu kısıt delinmiş olur. Bu aşamada geliştirilen uyarlanmış uniform çaprazlama ile çocuklardaki gen çakışmaları engellenmiştir. Şekil 4.12’de normal uniform çaprazlama işlemine tutulan ebeveynlerin aynısı, Şekil 4.13 ve Şekil 4.14’de

geliştirilen uyarlanmış uniform çaprazlama yöntemiyle çaprazlanarak Şekil 4.12'deki çakışmaların nasıl önlendiği gösterilmiştir.

Uyarlanmış uniform çaprazlamada, birinci çocuk üretimi için rastgele oluşturulan dizide sıfıra karşılık gelen genler aynen birinci ebeveynden alınır. Birinci ebeveynden seçilen bu genlerin aynısı ikinci ebeveynde bulunuyor ise ikinci ebeveynden alınmasına müsaade edilmeyerek çakışma olasılığı önlenir. İkinci ebeveynde kalan genler, sırasıyla birinci çocuktaki boşluklara eklenerek birinci çocuk oluşumu tamamlanır.



**Şekil 4.13** Uyarlanmış uniform çaprazlama ile 1. çocuk oluşumu

İkinci çocuk üretimi için ise, rastgele oluşturulan dizideki sıfıra karşılık gelen ikinci ebeveyndeki genler aynen alınır. Bu genlerin aynısı birinci ebeveynden temizlenir ve ikinci çocuktaki boşluklar birinci ebeveynden doldurulur. Bu şekilde kendi içinde çakışmaya sahip olmayan zincirler üretilmiş olur.

Tablo 4.10'da çaprazlama işlemini yapan fonksiyon (*ciyf*) verilmiştir. Her zincir için çaprazlama işlemi ayrı ayrı yapılarak oluşan çocuklar birbirine eklenmekte ve sonuçta oluşan kromozomlar çocuk havuzuna eklenmektedir.

Tablo 4.10'da verilen çaprazlama fonksiyonunda;

*ics*: İstenen çocuk sayısı,

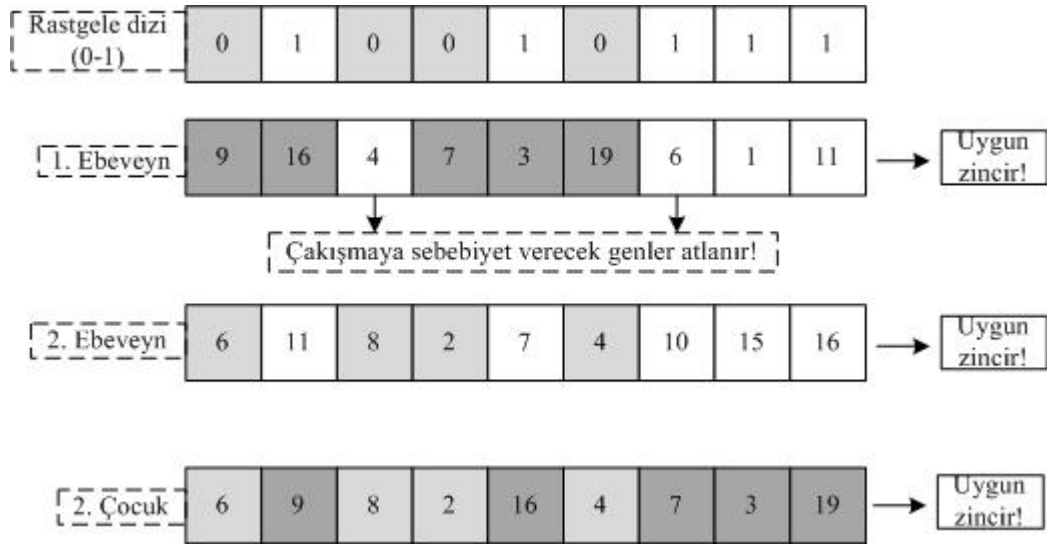
*ku*: Kromozom uzunluğu,

*zs*: Zincir sayısı,

*zu*: Zincir uzunluğu, olmak üzere;

toplam işlem sayısı 4.10 nolu denkleme göre hesaplanabilmektedir.





**Şekil 4.14** Uyarlanmış uniform çaprazlama ile 2. çocuk oluşumu

$$ciyf = (8 \times ku^2 + 6 \times ku + 2 \times zu + 5) \times zs \times ics / 2 + 2 \times zu^2 + zu$$

4.10

**Tablo 4.10** Çaprazlama işlemi fonksiyonu

Pseudo Kod
<pre> <b>FUNCTION crossover</b> CALL <b>computeCumulative</b> with chainsArray RETURNING qChainsArray FOR each mating in the MatingSet   FOR each chain in qChainsArray     FOR each gene in the chains       SET Parent1 to first mating individual of population       SET Parent2 to second mating individual of population     END FOR     SET P1 to Parent1     SET P2 to Parent2     SET randArray to random numbers composed of 0 or 1 with size of each parent     FOR each element of randArray       IF the element of randArray IS EQUAL 0 THEN         SET the element of Offspring1 to the element of Parent1         FOR each gene in P2           IF the element of Offspring1 IS EQUAL the gene of P2             DELETE the gene of P2             BREAK           ENDIF         ENDFOR       ENDIF     ENDFOR   ENDFOR   FOR each element of randArray     IF the element of randArray IS EQUAL 1 THEN       FOR each gene in P2         SET the element of Offspring1 to the gene of P2         DELETE the gene of P2         BREAK       END FOR     END IF   END FOR </pre>

(Devamı arkada)

```

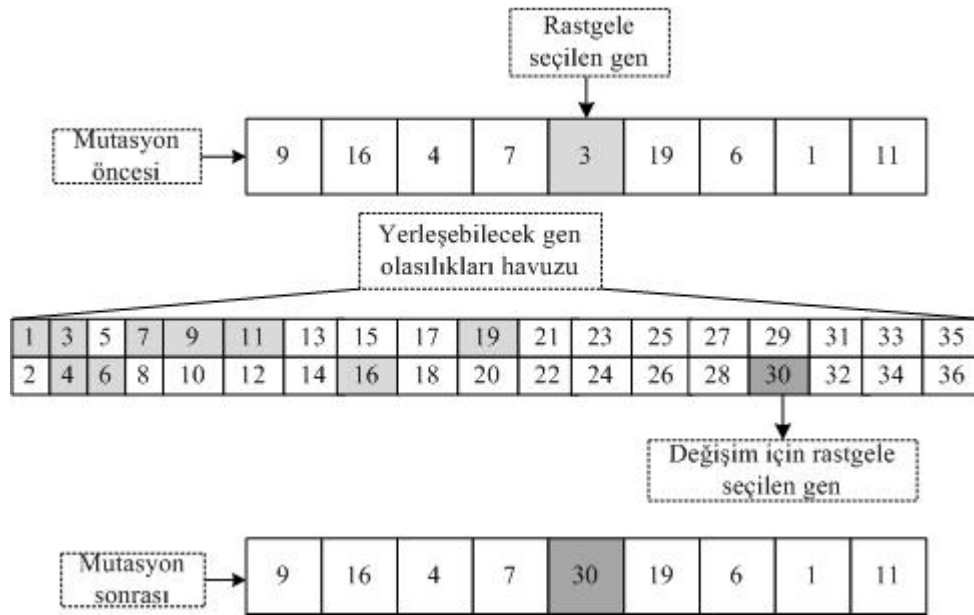
                                ENDFOR
                            ENDIF
                    ENDFOR
                FOR each element of randArray
                    IF the element of randArray IS EQUAL 0 THEN
                        SET the element of Offspring2 to the element of Parent2
                        FOR each gene in P1
IF the element of Offspring2 IS EQUAL the gene of P1
                                DELETE the gene of P1
                                BREAK
                            ENDIF
                        ENDFOR
                    ENDIF
                ENDFOR
                FOR each element of randArray
                    IF the element of randArray IS EQUAL 1 THEN
                        FOR each gene in P1
                            SET the element of Offspring2 to the gene of P1
                            DELETE the gene of P1
                            BREAK
                        ENDFOR
                    ENDIF
                ENDFOR
                APPEND Offspring1 to Offsprings
                APPEND Offspring2 to Offsprings
            ENDFOR
        ENDFOR
    RETURN Offsprings
END FUNCTION

```

#### 4.7 Mutasyon

Mutasyon işlemi için, üçüncü bölümde anlatılan bit çevirme (flip bit) operatörü kromozom yapısında ikili genler bulunmadığı için bu probleme uygulanamamaktadır. Değiş tokuş (swap) operatörü ise, kromozomda zaten bulunan genlerin yerlerini değiştirdiği için çeşitlilik yaratmamaktadır. Oysaki bu problemde mutasyonun çeşitlilik yaratması kesinlikle gereklidir. Bir zincirde hiç bulunmayan bir genin mutasyon ile oluşturulabilmesi için yerine koyma (replacement) operatörüne ihtiyaç vardır. Bu sebeple, yerine koyma (replacement) operatörünün kullanılması tercih edilmiştir. Bu operatör sayesinde çaprazlama ile elde edilemeyen gen çeşitliliği sağlanmaktadır.

Şekil 4.15'te görüldüğü gibi mutasyona uğratmak için seçilen bir kromozomda rastgele bir gen mutasyon işlemi için seçilir. Bu genin değeri kendi zinciri içindeki diğer genlerden farklı olması gerektiği için yerleşebilecek gen olasılıkları içerisinde zincirdeki arkadaş genler çıkarılır ve geriye kalan genlerden bir tanesi rastgele seçilerek ilgili genin yerine geçer. Bu şekilde mutasyon işlemi tamamlanmış olur.



**Şekil 4.15** Herhangi bir zincirdeki mutasyon işlemi

Tablo 4.11’de de verilen her kromozomda tek geni mutasyona uğratan fonksiyonda (m1);

*ims*: İstenen mutant sayısı,

*TS*: Atanabilecek toplam zaman dilimi sayısı,

*zu*: Zincir uzunluğu, olmak üzere;

toplam işlem sayısı 4.11 nolu denklem ile hesaplanabilir.

$$m1 = (2 \times TS \times zu + 8) \times ims + 2 \times zu^2 + zu + 1 \quad 4.11$$

Tablo 4.12’de verilen her kromozomda iki geni mutasyona uğratan fonksiyonda (m2);

*ims*: İstenen mutant sayısı,

*TS*: Atanabilecek toplam zaman dilimi sayısı

*zu*: Zincir uzunluğu, olmak üzere;

toplam işlem sayısı 4.12 nolu denklem ile hesaplanabilmektedir.

$$m2 = (2 \times TS \times zu + 8) \times ims / 2 + 2 \times zu^2 + zu + 1 \quad 4.12$$

Tablo 4.13’de verilen her kromozomda dört geni mutasyona uğratan fonksiyonda ( $m4$ );

$ims$ : İstenen mutant sayısı,

$TS$ : Atanabilecek toplam zaman dilimi sayısı,

$zu$ : Zincir uzunluğu, olmak üzere;

toplam işlem sayısı 4.13 nolu denklem ile hesaplanabilmektedir.

$$m4 = (2 \times TS \times zu + 8) \times ims / 4 + 2 \times zu^2 + zu + 1$$

4.13

**Tablo 4.11** Her kromozomda tek geni mutasyona uğratan fonksiyon

Pseudo Kod
<pre> <b>FUNCTION mutation1</b>  CALL <b>computeCumulative</b> with chainsArray RETURNING qChainsArray  SET mutant to zero <b>WHILE</b> mutant <b>IS LESS THAN</b> nomutant     <b>INCREMENT</b> mutant     <b>SELECT</b> a choromosome randomly in the population     <b>SELECT</b> a chain randomly in the selected choromosome     <b>SELECT</b> a gene randomly in the selected chain     <b>SET</b> randomTimeSlotsArray to maxTimeSlot      <b>FOR</b> each gene in the selected chain         <b>FOR</b> each timeSlot in randomTimeSlotsArray             <b>IF</b> gene <b>IS EQUAL</b> timeSlot <b>THEN</b>                 <b>DELETE</b> timeSlot                 <b>BREAK</b>             <b>END IF</b>         <b>ENDFOR</b>     <b>ENDFOR</b>      <b>SELECT</b> a timeSlot randomly from randomTimeSlotsArray     <b>REPLACE</b> selected gene with selected timeSlot to create a Mutant     <b>APPEND</b> Mutant to Mutants <b>END WHILE</b>  <b>RETURN</b> Mutants <b>END FUNCTION</b> </pre>

Her kromozomda bir, iki veya dört genin mutasyona uğratılacağı sisteme parametre olarak eklenmiştir. Bu şekilde farklı mutasyon taktiklerinin sisteme sonuç olarak nasıl yansıtacağı görülebilecektir.

**Tablo 4.12** Her kromozomda iki geni mutasyona uğratan fonksiyon

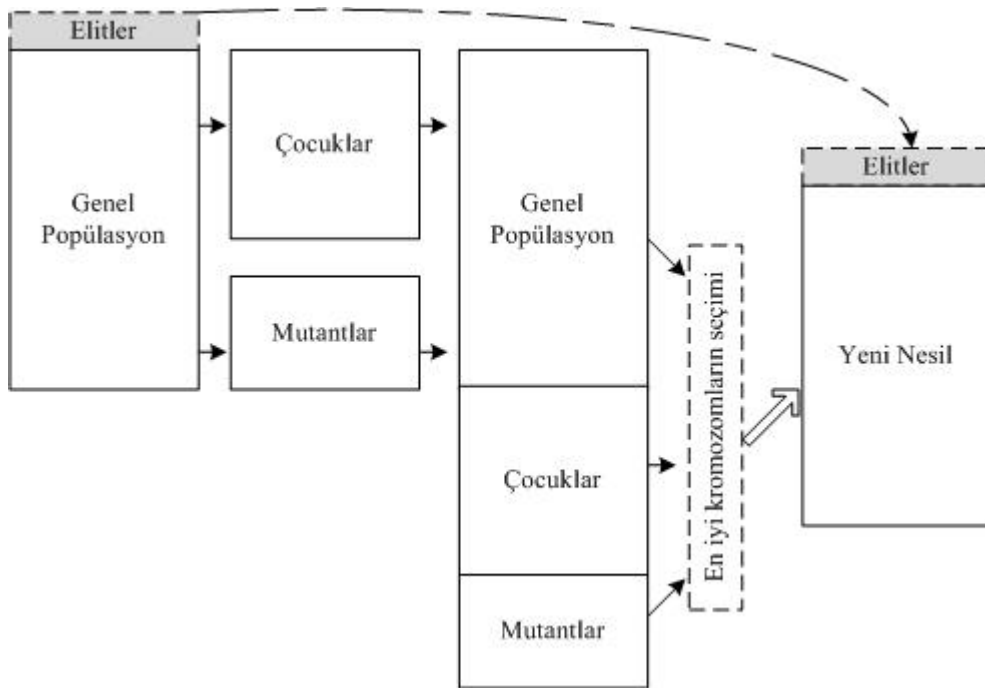
Pseudo Kod
<pre> <b>FUNCTION mutation2</b> CALL <b>computeCumulative</b> with chainsArray RETURNING qChainsArray SET mutant to zero <b>WHILE</b> mutant <b>IS LESS THAN</b> half of nomutant   <b>INCREMENT</b> mutant   <b>SELECT</b> a chorosome randomly in the population   <b>SELECT</b> two chains randomly in the selected chorosome   <b>SELECT</b> two genes randomly in the selected chains   <b>FOR</b> each chain in the selected chains     <b>SET</b> randomTimeSlotsArray to maxTimeSlot     <b>FOR</b> each gene in the chain       <b>FOR</b> each timeSlot in randomTimeSlotsArray         <b>IF</b> gene <b>IS EQUAL</b> timeSlot <b>THEN</b>           <b>DELETE</b> timeSlot from randomTimeSlotArray           <b>BREAK</b>         <b>END IF</b>       <b>ENDFOR</b>     <b>ENDFOR</b>     <b>SELECT</b> a timeSlot randomly from randomTimeSlotsArray     <b>REPLACE</b> selected gene with selected timeSlot to create Mutant   <b>ENDFOR</b>   <b>APPEND</b> Mutant to Mutants <b>END WHILE</b> <b>RETURN</b> Mutants <b>END FUNCTION</b> </pre>

**Tablo 4.13** Her kromozomda dört geni mutasyona uğratan fonksiyon

Pseudo Kod
<pre> <b>FUNCTION mutation4</b> CALL <b>computeCumulative</b> with chainsArray RETURNING qChainsArray SET mutant to zero <b>WHILE</b> mutant <b>IS LESS THAN</b> quarter of nomutant   <b>INCREMENT</b> mutant   <b>SELECT</b> a chorosome randomly in the population   <b>SELECT</b> four chains randomly in the selected chorosome   <b>SELECT</b> four genes randomly in the selected chains   <b>FOR</b> each chain in the selected chains     <b>SET</b> randomTimeSlotsArray to maxTimeSlot     <b>FOR</b> each gene in the chain       <b>FOR</b> each timeSlot in randomTimeSlotsArray         <b>IF</b> gene <b>IS EQUAL</b> timeSlot <b>THEN</b>           <b>DELETE</b> timeSlot from randomTimeSlotArray           <b>BREAK</b>         <b>END IF</b>       <b>ENDFOR</b>     <b>ENDFOR</b>     <b>SELECT</b> a timeSlot randomly from randomTimeSlotsArray     <b>REPLACE</b> selected gene with selected timeSlot to create Mutant   <b>ENDFOR</b>   <b>APPEND</b> Mutant to Mutants <b>END WHILE</b> <b>RETURN</b> Mutants <b>END FUNCTION</b> </pre>

#### 4.8 En iyi kromozomların seçimi

Şekil 4.16'da görüldüğü gibi, genel popülasyondan seçilen bireyler belirlenen çaprazlama ve mutasyon oranlarına ve evrim kurallarına göre çaprazlama işlemine tutulduklarında çocuklar, mutasyon işlemine tutulduklarında ise mutantlar meydana gelir. Bu aşamada, elde edilen çocuk ve mutantlar hemen popülasyona dâhil edilmez. Genel popülasyondaki kromozomlar, elde edilen çocuklar ve mutantlar uygunluk değerlerine göre en iyiden en kötüye doğru sıralanır ve genel popülasyon sayısını tamamlayacak şekilde en iyiler seçilir. İlk popülasyondaki en iyiler olan elitler ise yeni nesile aynen aktarılır. Elitlerin aktarılması bir popülasyondaki iyi bireylerin korunarak kaybolmasını önlemek için yapılır. Bu şekilde yeni nesil oluşturulmuş olur. En iyi kromozomların seçimi her nesilde bu şekilde devam eder.



Şekil 4.16 Yeni nesil seçimi

Tablo 4.14'de verilen en iyi kromozomların seçimi fonksiyonunda (*eiksf*);

*es*: Elit sayısı,

*ics*: İstenen çocuk sayısı,

*ims*: İstenen mutant sayısı,

*oz1*: Ortak zincirin satır sayısı,

*oz2*: Ortak zincirin sütun sayısı,

*ps*: Popülasyon sayısı,

*udu*: Uygunluk dizisi uzunluğu,

*zu*: Zincir uzunluğu, olmak üzere;

toplam işlem sayısı 4.14 nolu denklem ile hesaplanabilmektedir.

$$eiksf = (ps - es + ics + ims) \times (uhdbf + \frac{ps - es + ics + ims + 1}{2}) + \frac{ps^2 + 9 \times ps}{2} + ics + ims + 1 \quad 4.14$$

**Tablo 4.14** En iyi kromozomların seçimi fonksiyonu

Pseudo Kod
<pre> <b>FUNCTION bestof</b>  FOR each chorosome in the population     IF chorosome &gt; elits THEN         APPEND chorosome to CandidatesArray     ENDIF END FOR  FOR each offspring in Offsprings     APPEND offspring to CandidatesArray END FOR  FOR each mutant in Mutants     APPEND mutant to CandidatesArray END FOR  FOR each chorosome in the CandidatesArray     CALL determineFitnessArray with chainsArray, Chorosome, Shared, SharedInfo,     CoursesInfo, nodays RETURNING fitvalue and confM ENDFOR  SORT fitvalue of each chorosome of the CandidatesArray  DETERMINE Pop from elits and best of CandidatesArray  SORT fitvalue of each chorosome in the general population again.  FOR each chorosome in Pop     IF chorosome IS EQUAL 1 THEN         SELECT confM and bestfitvalue         BREAK     ENDIF ENDFOR  RETURN bestfitvalue, confM and Pop </pre>

#### 4.9 Entegre Tamir Fonksiyonu

Öğrenci başarısına odaklı sınav çizelgeleme modelinin geliştirilmesi aşamasında yapılan testlerde, zorunlu kısıtların sağlanmasına yönelik kullanılan yüksek ceza değerinin genetik algoritmanın performansını olumsuz yönde etkilemiş olabileceği düşünülmüştür. Bu sebeple zorunlu olan birincil kısıtların sağlanmasını garanti etmek amacıyla, yüksek ceza değerine alternatif bir model geliştirilebileceği fikri ortaya çıkmıştır. Bu fikir üzerine geliştirilen bir tamir fonksiyonu ile çakışmaların oluşması engellenmektedir.

Her sınavın alttan veya üstten alınan bir ders olup olmadığı bilgisi ile ortak bir sınav olup olmadığı bilgisine ihtiyaç vardır. Tablo 4.15’de örnek olarak verilen endüstri mühendisliği bölümüne ait sınav bilgi formundaki ‘Alt’ sütunu, ilgili sınavın alttan alınan bir ders olup olmadığını, alttan alınıyorsa hangi sınıfların alttan aldığı bilgisini içermektedir. Söz konusu sınavı, üst sınıflardan ders alan hiçbir öğrenci almıyorsa ‘0’ değeri formdaki ilgili sütuna girilmiştir. Sadece bir üst sınıftan ders alan en az bir öğrenci alıyorsa ‘1’ değeri, sadece iki üst sınıftan ders alan en az bir öğrenci alıyorsa ‘2’ değeri, hem bir hem de iki üst sınıftan ders alan en az birer öğrenci alıyorsa ‘3’ değeri form bilgilerine girilerek kaydı tutulur. Bu sayede formdaki ilgili sütundaki değerlere göre hangi zincirlerin kontrol edilmesi gerektiği ortaya çıkar. 0 değeri okunursa, herhangi bir alt zincir kontrolü yapmaya gerek yoktur. 1 değeri okunduğunda, sadece bir alt zincir kontrole dâhil edilmeli, 2 değeri okunduğunda sadece iki alt zincir kontrole dâhil edilmeli, 3 değeri okunduğunda ise hem bir hem de iki alt zincirler kontrole dâhil edilmelidir. Aynı şekilde ‘Üst’ sütunu da ilgili sınavın üstten alınan bir sınav olup olmadığını, eğer üstten alıyorsa hangi sınıfların üstten aldığı bilgisini içermektedir. Söz konusu sınavı, alt sınıflardan ders alan hiçbir öğrenci almıyorsa ‘0’ değeri formdaki ilgili sütuna girilmiştir. Sadece bir alt sınıftan ders alan en az bir öğrenci alıyorsa ‘1’ değeri, sadece iki alt sınıftan ders alan en az bir öğrenci alıyorsa ‘2’ değeri, hem bir hem de iki alt sınıftan ders alan en az birer öğrenci alıyorsa ‘3’ değeri form bilgilerine girilerek kaydı tutulur. Bu sayede formdaki ilgili sütundaki değerlere göre hangi zincirlerin kontrol edilmesi gerektiği ortaya çıkar. 0 değeri okunursa, herhangi bir üst zincir kontrolü yapmaya gerek yoktur. 1 değeri okunduğunda, sadece bir üst zincir kontrole dâhil edilmeli, 2 değeri okunduğunda sadece iki üst zincir kontrole dâhil edilmeli, 3 değeri okunduğunda ise hem bir hem de iki üst zincirler kontrole dâhil

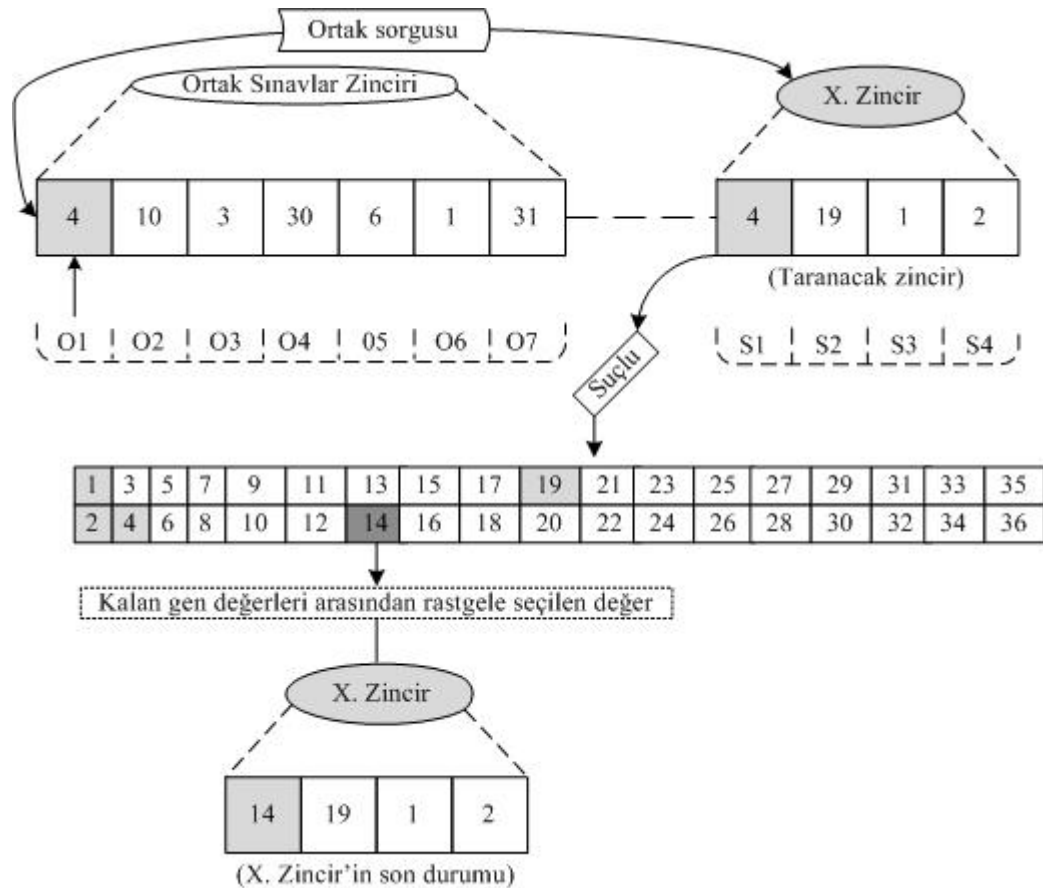


edilmelidir. ‘Ortak’ sütununda ise ilgili sınavın ortak yapılması gereken bir sınav olup olmadığı bilgisi ve ortak bir sınav ise hangi bölümlerin hangi sınıfları ile ortak yapılması gerektiği bilgisi bulunmaktadır. Bu sütundaki bilgilerden yararlanılarak ortak sınavlar matrisi elde edilmektedir. Bu matris kromozomun ilk zincirini oluşturan ortak sınavların diğer zincirlerle bağlantı bilgilerini içermektedir. Hangi sınavların aynı anda yapılması ve hangi sınavların aynı zamanda yapılmaması gerektiği bilgisi ile kromozomların bağlantısal olarak tamir edilebilmesi sağlanmaktadır.

**Tablo 4.15** Sınav bilgi formu

Endüstri1					
Ders	Optik Kod	Alt	Üst	Zorluk	Ortak
TÜRK DİLİ - I	11101	0	0	4	En1,B1,T1,J1
ATATÜRK İLKELERİ VE İNKILÂP TARİHİ - I	12101	3	0	4	Tüm bölümler
YABANCI DİL - I (İNGİLİZCE)	13101	0	0	1	Tüm bölümler
TEMEL BİLGİ TEKNOLOJİSİ KULLANIMI	20101	1	0	3	
GENEL FİZİK - I	151111	0	0	6	
GENEL KİMYA	153115	1	0	7	En1,E1,T1
GENEL MATEMATİK - I	155113	1	0	10	EN1,J1
ENDÜSTRİ MÜHENDİSLİĞİNE GİRİŞ	235101	1	0	6	
TEKNİK RESİM	235103	1	0	6	
BEDEN EĞİTİMİ - I	16101	0	0	1	En1,B1,J1
Endüstri2					
Ders	Optik Kod	Alt	Üst	Zorluk	Ortak
TEMEL BİLGİSAYAR BİLİMLERİ	20201	1	0	7	
DİFERANSİYEL DENKLEMLER	155219	3	0	8	
OLASILIK	235201	3	0	10	Elektronik3
DOĞRUSAL CEBİR	235203	3	0	10	
MÜHENDİSLİKTE MALZEME BİLGİSİ	235205	3	0	8	
MALİYET MUHASEBESİ	235209	3	1	6	
ENDÜSTRİYEL PSİKOLOJİ	235233	3	1	5	
Endüstri3					
Ders	Optik Kod	Alt	Üst	Zorluk	Ortak
TESİS TASARIMI VE PLANLAMASI	235301	1	0	8	
YÖNEYİM ARAŞTIRMASI	235303	1	0	10	
İŞLETME YÖNETİMİ	235305	1	0	6	
SİMÜLASYON	235307	1	0	9	
MESLEKİ İNGİLİZCE	235309	0	0	5	
ARAŞTIRMA YÖNTEMLERİ	235331	1	0	6	
Endüstri4					
Ders	Optik Kod	Alt	Üst	Zorluk	Ortak
KALİTE PLANLAMA VE KONTROLÜ	235401	0	0	9	
BAKIM PLANLAMASI VE GÜVENİRLİK	235403	0	0	6	
YATIRIM DEĞERLENDİRMESİ	235405	0	1	8	
KALİTE YÖNETİM SİSTEMLERİ	235431	0	0	5	
BİLGİSAYARLI BÜTÜNLEŞİK ÜRETİM	235437	0	0	10	
İNSAN KAYNAKLARI YÖNETİMİ	235439	0	0	7	

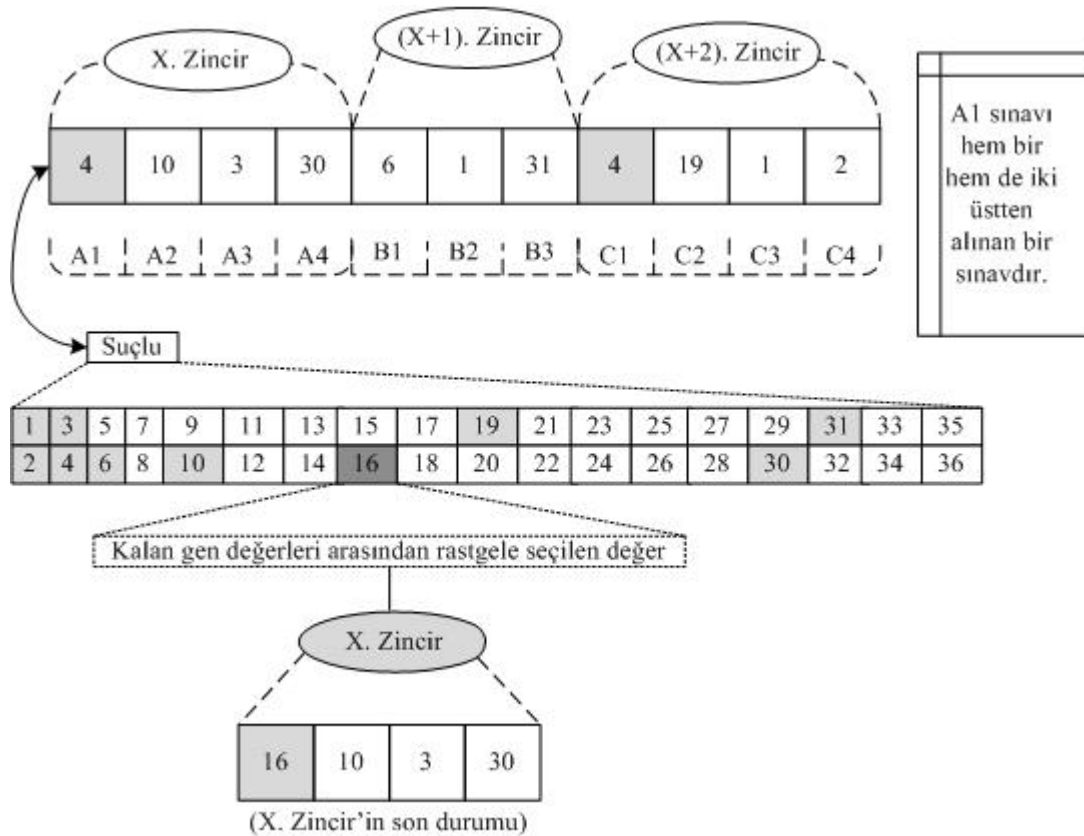
Şekil 4.17’de verilen örnekte, O1 sınavının X. zincire ait olan ortak sınavlardan biri olduğu varsayılmıştır. Tamir öncesi, 4,19,1 ve 2 değerlerinden oluşan genlere sahip olan X zinciri, O1 sınavının da bu zincire dâhil olması gerekliliği yüzünden, O1 sınavının 4. zaman diliminde yapılması nedeniyle bu zincirdeki 4. zaman dilimi ortak sınav için boşaltılacaktır. Bu sayede ortak olan sınav ile zincir içi çakışmanın önüne geçilmiştir. X. zincirdeki boşaltılan gen değerine atanması mümkün olmayan değerler olan zincirdeki diğer sınavların zaman dilimleri de seçim olasılıkları içerisinde çıkarılır. Kalan gen değerleri arasından bir tanesi rastgele seçilerek boşaltılan gen değerinin yerine yerleştirilir. Bu tamir işleminde yerine koyma operatörü ile bir çeşit zorunlu mutasyon uygulanıyor olduğu düşünülebilir. Aynı şekilde, ortak sınavlar matrisindeki bilgilerden yararlanarak, her ortak dersin sorgusu sırayla çalıştırılır ve ilgili zincirlerdeki düzeltmeler sırasıyla yapılır.



Şekil 4.17 Ortak sınav kontrolü ve tamir

Şekil 4.18’deki örnekte X. Zincire ait olan A1 sınavının hem bir, hem de iki alttan alınan bir sınav olduğu varsayılmıştır. Dolayısıyla, A1 sınavı X. Zincire ait olduğu için, (X+1). Zincir ve (X+2). Zincirdeki gen değerlerinden farklı bir gen değerine sahip

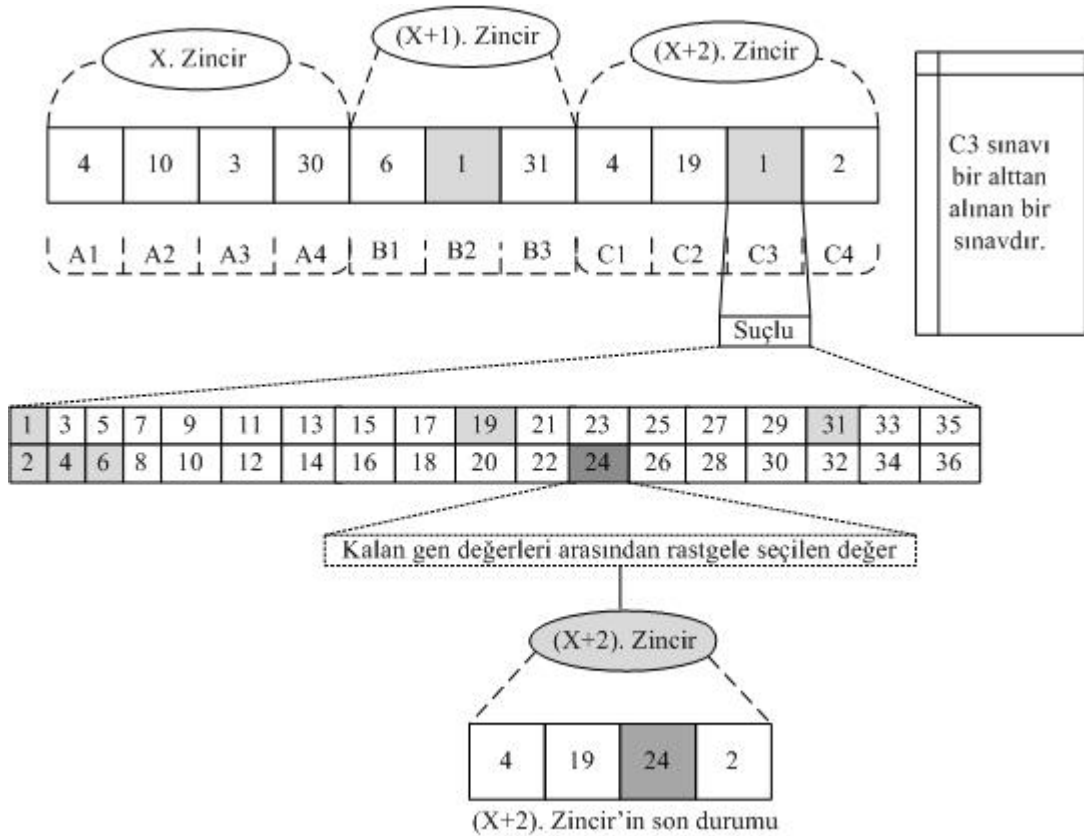
olmalıdır.  $(X+1)$ . Zincir tarandığında her hangi bir çakışma olmadığı görülmektedir, ancak  $(X+2)$ . Zincirdeki C1 sınavı, A1 sınavı ile aynı zaman dilime atanmış ve bir çakışma olduğu gözükmemektedir. Bu yüzden A1 sınavının atanmış olduğu zaman dilimi değiştirilmelidir. A1 sınavına atanacak yeni gen değeri  $X$ ,  $(X+1)$ . Ve  $(X+2)$ . Zincirdeki tüm gen değerlerinden farklı bir gen değerine sahip olmak zorundadır. Öyleyse suçlu ilan edilen A1 sınavının atanabileceği gen değerleri kümesinden bu zincirlerdeki gen değerlerinin tümü çıkarılır. Kalan gen değerleri arasından bir tanesi rastgele seçilerek suçlu gen değerinin yerine yerleştirilir. Ortak sınav kontrolü örneğinde belirtildiği gibi, bu sınavın atanacağı zaman diliminin değiştirilmesi, yerine koyma operatörü ile zorunlu mutasyon işlemi yapıldığı düşünülebilir.



**Şekil 4.18** Üst zincirlerin taranması ve kromozom tamiri

Altan alınan bir sınav olabileceği gibi, üstten alınan bir sınav da olabilir. Dolayısıyla, sınav bilgi formlarında, her sınav için girilen 'Üst' sütunundaki bilgilere dayanılarak alt zincir kontrolü yapılır. Şekil 4.19'deki örnekte C3 sınavının bir alttan alınan bir sınav olduğu varsayılmıştır. Dolayısıyla C3 sınavının atandığı zaman diliminin bir alt zincirdeki sınavların atandığı zaman dilimlerinden herhangi birisi ile aynı olmaması gerekmektedir. Bir alt zincir tarandığında B2 sınavının da C3 sınavı ile

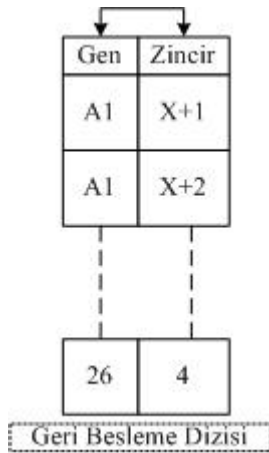
aynı zaman dilimine atanmış olduğu görülmektedir. Bu çakışmayı önlemek için C3 sınavının gen değeri değiştirilecektir. Suçlu ilan edilen C3 geni bir alt zincirdeki ve kendi zincirindeki gen değerlerinden herhangi birine sahip olamayacağı için, bu gen değerleri atanabilecek değerler arasından çıkarılır. Kalan gen değerleri arasından bir tanesi rastgele seçilerek C3 gen değerine yerleştirilir. Zorunlu mutasyon işlemi bu şekilde tamamlanmış olur.



**Şekil 4.19** Alt zincirlerin taranması ve kromozom tamiri

Kromozomdaki her gen için sırayla önce ortak kontrolünde, sonra alt ve üst kontrolünden geçirilerek değer değişikliğine gerek olup olmadığına karar verilir. Eğer bir çakışma tespit edilirse, o gen için tamir işlemi uygulanır. Kromozomdaki ilk genden son gene doğru tamir işlemleri yapılırken, sonradan yapılan değişiklikler önceden yapılan işlemleri etkileyebilir ve çakışmalar meydana gelebilir. Bu yüzden kromozomdaki zincirlerde ve gen değerlerinde yapılan her değişikliğin kaydı tutulması gerekir. Yapılan her değişikliğin kaydı tutulduğunda, hangi genin değerinin hangi zincir temel alınarak yapıldığı saklanmış olur. Böylece temel alınan zincirde bir değişiklik yapılmak istendiğinde, o genin değeri de dikkate alınarak zincirde değişikliğe gidilir.

Şekil 4.20’de verilen örnekte, A1 geni (X+1). ve (X+2). zincirlerdeki genlerin değerleri göz önüne alınarak tamir edildiği için geri besleme dizisine kaydedilmiştir. Bu geri besleme dizisinden yararlanılarak, ilerideki adımlarda (X+1). ve (X+2). zincirlerdeki herhangi bir gende değişiklik yapılacağı zaman A1 geninin sahip olduğu değer de göz önüne alınır. A1 geninin sahip olduğu değer de atanamayacak değerler arasına alınarak çakışmanın önüne geçilmiştir.



**Şekil 4.20** Geri besleme dizisi

Tablo 4.16’da verilen tamir fonksiyonunda( $tf$ ),

$ebzd$ : En büyük zaman dilimi,

$gbdu$  Geri besleme dizisi uzunluğu,

$oz1$ : Ortak zincirin satır sayısı,

$oz2$ : Ortak zincirin sütun sayısı,

$sdu$ : Şüpheli dizi uzunluğu,

$zs$ : Zincir sayısı,

$zu$ : Zincir uzunluğu, olmak üzere;

toplam işlem sayısı 4.15 nolu denklem ile hesaplanabilmektedir.

$$tf = (5 \times oz1 \times oz2 + (2 \times ebzd + 3) \times sdu + 2 \times gbdu + 4 \times zu + 20) \times zs \times zu + 2 \times zu^2 + zu + 1 \quad 4.15$$

**Tablo 4.16** Tamir fonksiyonu

Pseudo Kod
FUNCTION <b>repair</b>
CALL <b>computeCumulative</b> with chainsArray RETURNING qChainsArray
SET feedbackArray to NULL

(Devamı arkada)

```

FOR each chain in qChainsArray
  FOR each gene in the chain

    SET suspiciousArray to NULL
    FOR each row of SharedArray
      FOR each column of SharedArray
        IF SharedArray position is occupied with the chain THEN
          APPEND row numbered element of Chorosome to
            suspiciousArray
          BREAK
        ENDIF
      ENDFOR
    ENDFOR

  IF there is nobody who takes the course in coursesInfo from upper and lower classes
  THEN
    FOR each gene in the current chain except itself
      APPEND gene numbered element of Chorosome to suspiciousArray
    ENDFOR

  ELSEIF one lower class takes the course in coursesInfo THEN
    FOR each gene in the current and one lower chain except itself
      APPEND gene numbered element of Chorosome to suspiciousArray
    ENDFOR
    FOR each row of directSharedArray
      FOR each column of directSharedArray
        IF directSharedArray position is occupied with one lower chain
        THEN
          APPEND row numbered element of Chorosome to
            suspiciousArray
        ENDIF
      ENDFOR
    ENDFOR
    APPEND scanned chain numbers and affected gene to feedbackArray

  ELSEIF two lower class takes the course in coursesInfo THEN
    FOR each gene in the current and two lower chain except itself
      APPEND gene numbered element of Chorosome to suspiciousArray
    ENDFOR
    FOR each row of directSharedArray
      FOR each column of directSharedArray
        IF directSharedArray position is occupied with two lower chain
        THEN
          APPEND row numbered element of Chorosome to
            suspiciousArray
        ENDIF
      ENDFOR
    ENDFOR
    APPEND scanned chain numbers and affected gene to feedbackArray
  ELSEIF one lower and two lower classes takes the course in coursesInfo THEN
    FOR each gene in the current, one lower and two lower chains except itself
      APPEND gene numbered element of Chorosome to
        suspiciousArray
    ENDFOR
    FOR each row of directSharedArray
      FOR each column of directSharedArray
        IF directSharedArray position is occupied with one lower or
        two lower chain THEN

```

```

                APPEND row numbered element of Chorosome to
                suspiciousArray
            ENDIF
        ENDFOR
    ENDFOR
    APPEND scanned chain numbers and affected gene to feedbackArray

ELSEIF one upper class takes the course in coursesInfo THEN
    FOR each gene in the current and one upper chains except itself
        APPEND gene numbered element of Chorosome to suspiciousArray
    ENDFOR
    FOR each row of directSharedArray
        FOR each column of directSharedArray
            IF directSharedArray position is occupied with one upper chain
            THEN
                APPEND row numbered element of Chorosome to
                suspiciousArray
            ENDIF
        ENDFOR
    ENDFOR
    APPEND scanned chain numbers and affected gene to feedbackArray

ELSEIF one lower and one upper class takes the course in coursesInfo THEN
    FOR each gene in the current, one lower and one upper chains except itself
        APPEND gene numbered element of Chorosome to suspiciousArray
    ENDFOR
    FOR each row of directSharedArray
        FOR each column of directSharedArray
            IF directSharedArray position is occupied with one lower or one
            upper chain THEN
                APPEND row numbered element of Chorosome to
                suspiciousArray
            ENDIF
        ENDFOR
    ENDFOR
    APPEND scanned chain numbers and affected gene to feedbackArray
ELSEIF two lower and one upper class takes the course in coursesInfo THEN
    FOR each gene in the current, two lower and one upper chains except itself
        APPEND gene numbered element of Chorosome to suspiciousArray
    ENDFOR
    FOR each row of directSharedArray
        FOR each column of directSharedArray
            IF directSharedArray position is occupied with two lower or one
            upper chains THEN
                APPEND row numbered element of Chorosome to
                suspiciousArray
            ENDIF
        ENDFOR
    ENDFOR
    APPEND scanned chain numbers and affected gene to feedbackArray
ELSEIF two lower, one lower and one upper class takes the course in coursesInfo
THEN
    FOR each gene in the current, two lower, one lower and one upper chains
    except itself
        APPEND gene numbered element of Chorosome to suspiciousArray
    ENDFOR

```

```

FOR each row of directSharedArray
  FOR each column of directSharedArray
    IF directSharedArray position is occupied with two lower or one
    lower or one upper chains THEN
      APPEND row numbered element of Chorosome to
      suspiciousArray
    ENDIF
  ENDFOR
ENDFOR
APPEND scanned chain numbers and affected gene to feedbackArray

ELSEIF two upper class takes the course in coursesInfo THEN
  FOR each gene in the current and two upper chains except itself
    APPEND gene numbered element of Chorosome to
    suspiciousArray
  ENDFOR
  FOR each row of directSharedArray
    FOR each column of directSharedArray
      IF directSharedArray position is occupied with two upper chain
      THEN
        APPEND row numbered element of Chorosome to
        suspiciousArray
      ENDIF
    ENDFOR
  ENDFOR
  APPEND scanned chain numbers and affected gene to feedbackArray

ELSEIF one lower and two upper class takes the course in coursesInfo THEN
  FOR each gene in the current, one lower and two upper chains except itself
    APPEND gene numbered element of Chorosome to
    suspiciousArray
  ENDFOR
  FOR each row of directSharedArray
    FOR each column of directSharedArray
      IF directSharedArray position is occupied with one lower or two
      upper chain THEN
        APPEND row numbered element of Chorosome to
        suspiciousArray
      ENDIF
    ENDFOR
  ENDFOR
  APPEND scanned chain numbers and affected gene to feedbackArray

ELSEIF one upper and two upper class takes the course in coursesInfo THEN
  FOR each gene in the current, one upper and two upper chains except itself
    APPEND gene numbered element of Chorosome to
    suspiciousArray
  ENDFOR
  FOR each row of directSharedArray
    FOR each column of directSharedArray
      IF directSharedArray position is occupied with one upper or two
      upper chains THEN
        APPEND row numbered element of Chorosome to
        suspiciousArray
      ENDIF
    ENDFOR
  ENDFOR
  APPEND scanned chain numbers and affected gene to feedbackArray

```

(Devami arkada)



```

ELSEIF one lower, one upper and two upper class takes the course in coursesInfo
THEN
  FOR each gene in the current, one lower, one upper and two upper chains except
  itself
    APPEND gene numbered element of Chorosome to suspiciousArray
  ENDFOR
  FOR each row of directSharedArray
    FOR each column of directSharedArray
      IF directSharedArray position is occupied with one lower or one
      upper or two upper chain THEN
        APPEND row numbered element of Chorosome to
        suspiciousArray
      ENDIF
    ENDFOR
  ENDFOR
  APPEND scanned chain numbers and affected gene to feedbackArray
ENDIF
IF feedbackArray IS NOT EQUAL TO NULL
  FOR each element of feedbackArray
    IF feedbackArray is occupied with the current chain
      APPEND the affected gene to suspiciousArray
    ENDIF
  ENDFOR
ENDIF

IF suspiciousArray IS NOT EQUAL TO NULL
  SET criminal to FALSE
  FOR each element of suspiciousArray
    IF element IS EQUAL to gene of Chorosome
      SET criminal to TRUE
      BREAK
    ENDIF
  ENDFOR
  IF criminal IS EQUAL TO TRUE
    CALL dropRandTimeSlot with suspiciousArray and maxTimeSlot
    RETURNING availableTimeSlots
    IF availableTimeSlots IS EQUAL TO NULL
      SET Chorosome to DAMAGED.
      RETURN
    ELSE
      SELECT a timeSlot randomly from availableTimeSlots
      REPLACE the gene of Chorosome with selected
      TimeSlot
    ENDIF
  ENDIF
ENDIF
ENDFOR
ENDFOR

```

Tablo 4.17’de verilen uygun olmayan zaman dilimlerini silen fonksiyon Tablo 4.16’da verilen tamir fonksiyonu içinden çağrılarak atanması gerçekleştiğinde çakışmaya neden olacak zaman dilimlerinin atanması olanaksızlaştırılmaktadır. Tablo 4.17’de verilen fonksiyonda (*uozdsf*);

*ebzd*: En büyük zaman dilimi,

*sdu*: Şüpheli dizi uzunluğu, olmak üzere;  
toplam işlem sayısı 4.16 nolu denklem ile hesaplanılabilmektedir.

$$uozdsf = 2 \times ebzd \times sdu + sdu + 1$$

4.16

**Tablo 4.17** Uygun olmayan zaman dilimlerini silen fonksiyon

Pseudo Kod
<pre> <b>FUNCTION dropRandTimeSlot</b> SET availableTimeSlots to maxTimeSlot FOR each element of suspiciousArray   IF suspiciousArray IS NOT NULL     FOR each element of availableTimeSlots       IF element of suspiciousArray IS EQUAL TO element of availableTimeSlots         DELETE the element of availableTimeSlots         BREAK       ENDIF     ENDFOR   ENDIF ENDFOR  RETURN availableTimeSlots  <b>END FUNCTION</b> </pre>

## 5. DENEYSEL ÇALIŞMALAR

Bu bölümde, geliştirilen öğrenci başarısına odaklı genetik algoritma modelinin Pamukkale Üniversitesi Mühendislik Fakültesi 2007–2008 güz dönemi verileri üzerindeki etkileri araştırılmıştır. Farklı parametreler ve alternatif modeller karşılaştırılmalı olarak incelenmiştir.

Parametre testleri iki alternatif test yaklaşımı ile yapılabilmektedir. Birinci yaklaşımda, bir parametre sürekli değiştirilirken, diğer parametreler sabit tutulmaktadır. Her parametre diğer parametrelerin sabitliğinde değiştirilerek denenmektedir. Bu yöntemle, her parametrenin en iyi değerine ulaşılması amaçlanmaktadır. İkinci yaklaşım ise, parametrelerin farklı değerlerinin birbirleriyle farklı kombinasyonlarda birleşmesiyle ortaya çıkan sonuçların karşılaştırılmasına dayalıdır. Genetik algoritmalarda bir parametredeki değişikliğin diğer parametrelerin performansı üzerinde de değişiklik yapma ihtimali yüksek olduğu için, birinci yaklaşımın daha az gerçekçi sonuçlar ortaya çıkaracağı düşünülmüştür. Bu sebeple ikinci yaklaşımın kullanımı tercih edilmiştir.

Geliştirilen genetik algoritma modelinde; popülasyon büyüklüğü, elit oranı, çaprazlama oranı, mutasyon oranı, mutasyon taktiği ve standart sapma katsayısı olmak üzere toplam 6 farklı parametre kullanılmıştır. Tablo 5.1’de görüldüğü gibi her parametre için farklı seviyeler belirlenmiştir.

**Tablo 5.1** Modelde kullanılan parametreler ve seviyeleri

Parametreler	No	İsim	Seviyeler		
			1. Seviye	2. Seviye	3. Seviye
	1	Popülasyon Büyüklüğü (ps)	20	40	80
	2	Elit Oranı (eo)	0.1	0.2	-
	3	Çaprazlama Oranı (co)	0.3	0.6	0.9
	4	Mutasyon Oranı (mo)	0.01	0.05	0.1
	5	Mutasyon Taktiği (mt)	1	2	4
	6	Standart Sapma Katsayısı (c)	0	3	-

Tablo 5.1’de görülen parametrelerin alt seviyeleriyle birlikte tüm kombinasyonları birer kere yapabilmek için  $3 \times 2 \times 3 \times 3 \times 3 \times 2 = 2^2 \times 3^4 = 324$  adet deneye gerek vardır.

Her parametre setinin rassallık içeren deneylerde farklı performans sergileyebilmesi nedeniyle, parametre setleri birden fazla sayıda denenmeli ve ortaya çıkan sonuçların ortalaması dikkate alınmalıdır. Her parametre setinin beşer kere deneneceği düşünülürse, toplam  $5 \times 324 = 1620$  adet deney yapılması gerekmektedir. Dördüncü bölümde anlatılan yüksek ceza puanı katsayısı yöntemi Pentium IV 1.80 Ghz hızındaki bir işlemcide uygulandığında; Tablo 5.1’de verilen parametreler için, bir parametre setinin sonuç üretme zamanı ortalama yaklaşık 10 dakika olarak ortaya çıkmaktadır. Bu nedenle 1620 adet deneme  $16200 \text{ dakika} = 270 \text{ saat} = 11.25 \text{ gün}$  sürecektir. Tamir fonksiyonu içeren yöntem Pentium IV 1.80 Ghz hızındaki bir işlemcide uygulandığında ise, Tablo 5.1’de verilen parametreler için, bir parametre setinin sonuç üretme zamanı ortalama yaklaşık 25 dakika olarak ortaya çıkmaktadır. Bu nedenle 1620 adet deneme  $40500 \text{ dakika} = 675 \text{ saat} = 28.125 \text{ gün}$  alacaktır. Her iki yöntemle tüm deney sonuçlarının elde edilmesi  $11.25 + 28.125 \cong 40 \text{ gün}$  alacaktır. Bu süreç işletilmeye çalışılmış, ancak fakültemizdeki sık ve uzun süreli elektrik kesintileri nedeniyle kullanmış olduğumuz kesintisiz güç kaynağı üniteleri de çaresiz kalmıştır. Deneylerde aksama meydana gelmesi nedeniyle, parametre setlerinden daha kısa sürede sonuç elde edebilmek için Taguchi’nin deneysel tasarım kataloğundan yararlanılmıştır. Taguchi’nin L27 tablosu ile yirmi yedi adet parametre seti beşer kez denenerek toplam  $27 \times 5 = 135$  deney yapılarak deneysel tasarım süreci her iki yöntem için de sonuçlandırılmıştır.

Tablo 5.2’de verilen Taguchi’nin L27 tablosu her birinde üç farklı seviye olan on üç farklı parametre içindir. Modelimizde altı adet parametre bulunduğu için L27 tablosundaki fazla parametreler Tablo 5.3’de görüldüğü gibi kaldırılmıştır. Fazla olan sütunların göz ardı edilmesiyle uygulanacak parametre sayısına göre uyarlanmıştır.

Tablo 5.3’de verilen İndirgenmiş L27 tablosunda her parametre için üçer tane seviye bulunmaktadır. Ancak Tablo 5.1’de verilen parametrelerden, elit oranı ve standart sapma katsayısı parametrelerinde ikişer adet seviye bulunduğu için İndirgenmiş L27 tablosuna sahte işleyiş (dummy treatment) uygulanması gerekmektedir.

Tablo 5.3’deki İndirgenmiş L27 tablosundaki elit oranının 3. seviyesi bulunmadığı için, 2. seviyedeki daha yüksek olan elit oranı kullanılmıştır. Aynı şekilde standart sapma katsayısında da 3. seviye bulunmadığı için, popülasyondaki daha fazla bireyin taranmasını sağlayan 2. seviyedeki katsayı kullanılmıştır. Sahte işleyiş (dummy treatment) uygulandıktan sonraki değerler Tablo 5.4’de görülmektedir.

**Tablo 5.2** Taguchi L27

Deney	Parametreler												
	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3	3	3	3	3
4	1	2	2	2	1	1	1	2	2	2	3	3	3
5	1	2	2	2	2	2	2	3	3	3	1	1	1
6	1	2	2	2	3	3	3	1	1	1	2	2	2
7	1	3	3	3	1	1	1	3	3	3	2	2	2
8	1	3	3	3	2	2	2	1	1	1	3	3	3
9	1	3	3	3	3	3	3	2	2	2	1	1	1
10	2	1	2	3	1	2	3	1	2	3	1	2	3
11	2	1	2	3	2	3	1	2	3	1	2	3	1
12	2	1	2	3	3	1	2	3	1	2	3	1	2
13	2	2	3	1	1	2	3	2	3	1	3	1	2
14	2	2	3	1	2	3	1	3	1	2	1	2	3
15	2	2	3	1	3	1	2	1	2	3	2	3	1
16	2	3	1	2	1	2	3	3	1	2	2	3	1
17	2	3	1	2	2	3	1	1	2	3	3	1	2
18	2	3	1	2	3	1	2	2	3	1	1	2	3
19	3	1	3	2	1	3	2	1	3	2	1	3	2
20	3	1	3	2	2	1	3	2	1	3	2	1	3
21	3	1	3	2	3	2	1	3	2	1	3	2	1
22	3	2	1	3	1	3	2	2	1	3	3	2	1
23	3	2	1	3	2	1	3	3	2	1	1	3	2
24	3	2	1	3	3	2	1	1	3	2	2	1	3
25	3	3	2	1	1	3	2	3	2	1	2	1	3
26	3	3	2	1	2	1	3	1	3	2	3	2	1
27	3	3	2	1	3	2	1	2	1	3	1	3	2

**Tablo 5.3** İndirgenmiş L27

Deney	Popülasyon Büyüklüğü	Elit Oranı	Çaprazlama Oranı	Mutasyon Oranı	Mutasyon Taktiği	Standart Sapma Katsayısı
1	1	1	1	1	1	1
2	1	1	1	1	2	2
3	1	1	1	1	3	3
4	1	2	2	2	1	1
5	1	2	2	2	2	2
6	1	2	2	2	3	3
7	1	3	3	3	1	1
8	1	3	3	3	2	2
9	1	3	3	3	3	3
10	2	1	2	3	1	2
11	2	1	2	3	2	3
12	2	1	2	3	3	1
13	2	2	3	1	1	2
14	2	2	3	1	2	3
15	2	2	3	1	3	1
16	2	3	1	2	1	2
17	2	3	1	2	2	3
18	2	3	1	2	3	1
19	3	1	3	2	1	3
20	3	1	3	2	2	1
21	3	1	3	2	3	2
22	3	2	1	3	1	3
23	3	2	1	3	2	1
24	3	2	1	3	3	2
25	3	3	2	1	1	3
26	3	3	2	1	2	1
27	3	3	2	1	3	2

**Tablo 5.4** Sahte işleyiş (Dummy treatment) uygulanması sonrası indirgenmiş L27

Deney	Popülasyon Büyüklüğü	Elit Oranı	Çaprazlama Oranı	Mutasyon Oranı	Mutasyon Taktiği	Standart Sapma Katsayısı
1	1	1	1	1	1	1
2	1	1	1	1	2	2
3	1	1	1	1	3	2
4	1	2	2	2	1	1
5	1	2	2	2	2	2
6	1	2	2	2	3	2
7	1	2	3	3	1	1
8	1	2	3	3	2	2
9	1	2	3	3	3	2
10	2	1	2	3	1	2
11	2	1	2	3	2	2
12	2	1	2	3	3	1
13	2	2	3	1	1	2
14	2	2	3	1	2	2
15	2	2	3	1	3	1
16	2	2	1	2	1	2
17	2	2	1	2	2	2
18	2	2	1	2	3	1
19	3	1	3	2	1	2
20	3	1	3	2	2	1
21	3	1	3	2	3	2
22	3	2	1	3	1	2
23	3	2	1	3	2	1
24	3	2	1	3	3	2
25	3	2	2	1	1	2
26	3	2	2	1	2	1
27	3	2	2	1	3	2

Tablo 5.4'deki sahte işleyiş uygulanmış olan İndirgenmiş L27 tablosundaki parametre numaralarının yerine Tablo 5.1'de verilen modeldeki parametre ve seviyeleri koyulduğunda Tablo 5.5'deki parametre setleri tablosu elde edilir.

**Tablo 5.5** Parametre setleri tablosu

Deney	Popülasyon Büyüklüğü	Elit Oranı	Çaprazlama Oranı	Mutasyon Oranı	Mutasyon Taktiği	Standart Sapma Katsayısı
1	20	0.1	0.3	0.01	1	0
2	20	0.1	0.3	0.01	2	3
3	20	0.1	0.3	0.01	4	3
4	20	0.2	0.6	0.05	1	0
5	20	0.2	0.6	0.05	2	3
6	20	0.2	0.6	0.05	4	3
7	20	0.2	0.9	0.1	1	0
8	20	0.2	0.9	0.1	2	3
9	20	0.2	0.9	0.1	4	3
10	40	0.1	0.6	0.1	1	3
11	40	0.1	0.6	0.1	2	3
12	40	0.1	0.6	0.1	4	0
13	40	0.2	0.9	0.01	1	3
14	40	0.2	0.9	0.01	2	3
15	40	0.2	0.9	0.01	4	0
16	40	0.2	0.3	0.05	1	3
17	40	0.2	0.3	0.05	2	3
18	40	0.2	0.3	0.05	4	0
19	80	0.1	0.9	0.05	1	3
20	80	0.1	0.9	0.05	2	0
21	80	0.1	0.9	0.05	4	3
22	80	0.2	0.3	0.1	1	3
23	80	0.2	0.3	0.1	2	0
24	80	0.2	0.3	0.1	4	3
25	80	0.2	0.6	0.01	1	3
26	80	0.2	0.6	0.01	2	0
27	80	0.2	0.6	0.01	4	3

*c*: Standart sapma katsayısı,

*co*: Çaprazlama oranı,

*eo*: Elit oranı,

*mo*: Mutasyon oranı,

*mt*: Mutasyon taktiği,

*ps*: Popülasyon büyüklüğü, olmak üzere;

1. Parametre Seti:  $ps = 20$ ,  $eo = 0.1$ ,  $co = 0.3$ ,  $mo = 0.01$ ,  $mt = 1$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.6'da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.7'de verilmiştir.

**Tablo 5.6** Yüksek ceza katsayısı modeli ile 1. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
22079	4988	106	230	0	0	16	32	304
22873	5133	85	186	0	1	21	37	281
23306	4778	121	264	0	0	12	41	298
21839	5119	92	201	0	4	15	28	297
23775	4850	110	239	0	3	19	37	269

**Tablo 5.6**'da verilen deney sonuçlarına göre, ortalama 103 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4974, bu uygunluk değerlerinin standart sapması ise 158 olmuştur.

**Tablo 5.7** Tamir fonksiyonu modeli ile 1. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6491	2516	156	146	0	0	4	14	176
6174	2377	180	170	0	2	4	14	164
6538	2590	152	143	0	0	8	7	174
6566	2434	145	137	0	0	3	16	172
6424	2536	131	123	0	0	1	18	184

**Tablo 5.7**'de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 153 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2491, bu uygunluk değerlerinin standart sapması ise 85 olmuştur.

Şekil 5.1'de 1. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**2. Parametre Seti:**  $ps = 20$ ,  $eo = 0.1$ ,  $co = 0.3$ ,  $mo = 0.01$ ,  $mt = 2$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar **Tablo 5.8**'de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise **Tablo 5.9**'da verilmiştir.



Tablo 5.8’de verilen deney sonuçlarına göre, ortalama 171 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4859, bu uygunluk değerlerinin standart sapması ise 134 olmuştur.

**Tablo 5.8** Yüksek ceza katsayısı modeli ile 2. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
18025	4876	149	396	0	1	13	35	304
22535	5040	160	424	0	5	11	36	296
22064	4678	180	478	0	0	10	41	291
22387	4903	178	474	0	9	9	33	291
24662	4798	187	497	0	0	13	43	292

**Tablo 5.9** Tamir fonksiyonu modeli ile 2. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6339	2600	254	361	0	2	8	6	177
6402	2359	295	421	0	0	3	15	168
6304	2301	270	385	0	0	3	17	164
6561	2404	245	349	0	0	3	13	176
6488	2349	384	548	0	0	3	12	179

Tablo 5.9’da verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 290 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2403, bu uygunluk değerlerinin standart sapması ise 116 olmuştur.

Şekil 5.2’de 2. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**3. Parametre Seti:**  $ps = 20$ ,  $eo = 0.1$ ,  $co = 0.3$ ,  $mo = 0.01$ ,  $mt = 4$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.10’da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.11’de verilmiştir.

Tablo 5.10’da verilen deney sonuçlarına göre, ortalama 117 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 6257, bu uygunluk değerlerinin standart sapması ise 605 olmuştur.

**Tablo 5.10** Yüksek ceza katsayısı modeli ile 3. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
23157	6194	90	270	0	7	30	35	298
21825	7288	123	368	0	28	21	13	299
21423	5872	106	317	0	5	34	31	278
23198	6163	146	437	0	18	16	21	310
24044	5766	119	355	0	7	33	18	293

**Tablo 5.11** Tamir fonksiyonu modeli ile 3. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6416	2990	114	216	0	5	10	10	175
6248	2902	121	227	0	4	6	16	174
6720	2874	123	233	0	2	8	14	171
6580	3238	104	196	0	8	8	7	180
6582	2916	178	342	0	5	8	14	175

Tablo 5.11’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 128 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2984, bu uygunluk değerlerinin standart sapması ise 148 olmuştur.

Şekil 5.3’de 3. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

4. Parametre Seti:  $ps = 20$ ,  $eo = 0.2$ ,  $co = 0.6$ ,  $mo = 0.05$ ,  $mt = 1$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.12’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.13’de verilmiştir.

Tablo 5.12’de verilen deney sonuçlarına göre, ortalama 196 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4471, bu uygunluk değerlerinin standart sapması ise 65 olmuştur.

**Tablo 5.12** Yüksek ceza katsayısı modeli ile 4. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
23259	4552	179	145	0	0	5	43	300
22698	4418	221	179	0	0	1	48	291
23008	4531	178	144	0	0	2	47	298
21040	4414	216	175	0	0	0	54	276
20387	4442	188	152	0	0	1	51	277

**Tablo 5.13** Tamir fonksiyonu modeli ile 4. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
5870	2323	384	92	0	0	1	15	175
6166	2338	472	113	0	0	1	14	175
6050	2307	448	107	0	0	2	17	168
6648	2252	524	125	0	0	1	14	171
6259	2392	419	100	0	0	5	13	171

Tablo 5.13’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 449 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2322, bu uygunluk değerlerinin standart sapması ise 51 olmuştur.

Şekil 5.4’de 4. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**5. Parametre Seti:**  $ps = 20$ ,  $eo = 0.2$ ,  $co = 0.6$ ,  $mo = 0.05$ ,  $mt = 2$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.14’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.15’de verilmiştir.

Tablo 5.14’de verilen deney sonuçlarına göre, ortalama 271 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4563, bu uygunluk değerlerinin standart sapması ise 68 olmuştur.

Tablo 5.15’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 555 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2292, bu uygunluk değerlerinin standart sapması ise 35 olmuştur.

**Tablo 5.14** Yüksek ceza katsayısı modeli ile 5. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
22148	4541	255	318	0	0	3	50	279
24035	4498	260	326	0	0	6	42	289
20318	4509	277	347	0	0	6	43	287
22163	4618	319	400	0	0	8	43	294
22753	4651	244	306	0	0	5	48	287

**Tablo 5.15** Tamir fonksiyonu modeli ile 5. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6468	2314	534	224	0	0	1	16	174
6421	2293	453	191	0	0	2	13	175
6235	2302	506	213	0	0	4	18	160
6023	2233	676	285	0	0	2	16	167
5810	2320	605	256	0	0	6	14	168

Şekil 5.5’de 5. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**6. Parametre Seti:**  $ps = 20$ ,  $eo = 0.2$ ,  $co = 0.6$ ,  $mo = 0.05$ ,  $mt = 4$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.16’da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.17’de verilmiştir.

Tablo 5.16’da verilen deney sonuçlarına göre, ortalama 117 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 5964, bu uygunluk değerlerinin standart sapması ise 376 olmuştur.

Tablo 5.17’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 304 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2792, bu uygunluk değerlerinin standart sapması ise 135 olmuştur.

Şekil 5.6’da 6. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.16** Yüksek ceza katsayısı modeli ile 6. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
22729	6470	77	131	0	16	28	27	282
20527	6150	148	252	0	17	24	25	274
16942	5682	150	256	0	9	20	25	300
23249	5995	118	200	0	9	23	30	294
22009	5522	93	159	0	6	21	32	287

**Tablo 5.17** Tamir fonksiyonu modeli ile 6. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6235	2959	237	160	0	2	11	15	174
6310	2858	311	210	0	4	5	16	166
5701	2649	366	248	0	4	4	14	173
6281	2837	318	216	0	3	10	7	175
6423	2659	289	196	0	4	7	10	177

**7. Parametre Seti:**  $ps = 20$ ,  $eo = 0.2$ ,  $co = 0.9$ ,  $mo = 0.1$ ,  $mt = 1$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.18'de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.19'da verilmiştir.

Tablo 5.18'de verilen deney sonuçlarına göre, ortalama 304 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4455, bu uygunluk değerlerinin standart sapması ise 68 olmuştur.

Tablo 5.19'da verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 849 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2291, bu uygunluk değerlerinin standart sapması ise 56 olmuştur.

Şekil 5.7'de 7. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.18** Yüksek ceza katsayısı modeli ile 7. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
22872	4560	251	107	0	0	2	49	298
21217	4420	348	148	0	0	2	51	270
22928	4446	320	136	0	0	6	44	290
21640	4379	296	126	0	0	0	51	282
21306	4472	306	130	0	0	3	46	296

**Tablo 5.19** Tamir fonksiyonu modeli ile 7. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6112	2284	955	114	0	0	1	14	175
5521	2331	777	93	0	1	4	11	171
6251	2363	711	85	0	0	4	14	167
6178	2257	973	116	0	0	1	16	167
6246	2222	828	99	0	0	1	19	160

**8. Parametre Seti:**  $ps = 20$ ,  $eo = 0.2$ ,  $co = 0.9$ ,  $mo = 0.1$ ,  $mt = 2$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.20’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.21’de verilmiştir.

Tablo 5.20’de verilen deney sonuçlarına göre, ortalama 333 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4526, bu uygunluk değerlerinin standart sapması ise 54 olmuştur.

Tablo 5.21’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 941 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2276, bu uygunluk değerlerinin standart sapması ise 59 olmuştur.

Şekil 5.8’de 8. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.20** Yüksek ceza katsayısı modeli ile 8. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21921	4464	373	278	0	0	1	45	301
20173	4572	289	216	0	0	8	43	288
20634	4592	302	225	0	0	3	44	306
20959	4496	299	223	0	0	2	48	292
21193	4505	401	298	0	0	1	48	294

**Tablo 5.21** Tamir fonksiyonu modeli ile 8. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
5861	2279	1086	243	0	0	0	22	156
6477	2304	960	214	0	0	3	15	164
6206	2357	855	191	0	0	7	13	160
6290	2212	913	204	0	0	0	16	169
6128	2227	890	199	0	0	0	16	169

9. Parametre Seti:  $ps = 20$ ,  $eo = 0.2$ ,  $co = 0.9$ ,  $mo = 0.1$ ,  $mt = 4$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.22'de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.23'de verilmiştir.

**Tablo 5.22** Yüksek ceza katsayısı modeli ile 9. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
22473	5695	152	167	0	11	22	29	277
19648	5650	136	150	0	8	22	27	305
20026	5922	177	195	0	12	20	18	303
21269	5488	185	205	0	8	22	28	284
18106	5411	135	149	0	4	13	34	310

Tablo 5.22'de verilen deney sonuçlarına göre, ortalama 157 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 5633, bu uygunluk değerlerinin standart sapması ise 199 olmuştur.

**Tablo 5.23** Tamir fonksiyonu modeli ile 9. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6183	2651	438	166	0	2	5	7	183
6234	2699	240	90	0	1	8	16	162
6121	3130	211	79	0	7	10	10	176
6183	2537	411	155	0	0	9	9	174
6038	2554	400	150	0	0	5	15	177

Tablo 5.23’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 340 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2714, bu uygunluk değerlerinin standart sapması ise 242 olmuştur.

Şekil 5.9’da 9. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

10. Parametre Seti:  $ps = 40$ ,  $eo = 0.1$ ,  $co = 0.6$ ,  $mo = 0.1$ ,  $mt = 1$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.24’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.25’de verilmiştir.

**Tablo 5.24** Yüksek ceza katsayısı modeli ile 10. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
19878	4388	824	152	0	0	5	50	265
19488	4414	1057	195	0	0	0	57	267
21636	4324	1226	227	0	0	0	49	290
20358	4327	1059	196	0	0	1	49	278
19891	4267	1256	232	0	0	5	49	276

Tablo 5.24’de verilen deney sonuçlarına göre, ortalama 1084 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4344, bu uygunluk değerlerinin standart sapması ise 58 olmuştur.

Tablo 5.25’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 2529 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2217, bu uygunluk değerlerinin standart sapması ise 16 olmuştur.



**Tablo 5.25** Tamir fonksiyonu modeli ile 10. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6136	2202	2127	118	0	0	0	17	171
6160	2230	2218	123	0	0	1	17	163
6023	2215	2758	153	0	0	2	20	155
5910	2236	2744	152	0	0	0	17	169
6053	2201	2797	155	0	0	1	19	161

Şekil 5.10'da 10. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

11. Parametre Seti:  $ps = 40$ ,  $eo = 0.1$ ,  $co = 0.6$ ,  $mo = 0.1$ ,  $mt = 2$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.26'da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.27'de verilmiştir.

**Tablo 5.26** Yüksek ceza katsayısı modeli ile 11. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
20410	4446	1217	443	0	0	4	49	276
22828	4413	1020	371	0	0	0	45	303
19510	4465	853	310	0	0	8	45	279
19023	4379	1250	453	0	0	2	50	272
21818	4393	985	358	0	0	1	46	305

Tablo 5.26'da verilen deney sonuçlarına göre, ortalama 1065 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4419, bu uygunluk değerlerinin standart sapması ise 36 olmuştur.

Tablo 5.27'de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 2937 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2246, bu uygunluk değerlerinin standart sapması ise 42 olmuştur.

Şekil 5.11’de 11. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.27** Tamir fonksiyonu modeli ile 11. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6330	2289	2966	329	0	0	3	15	167
6171	2248	2707	300	0	0	2	19	158
5911	2285	2821	313	0	0	0	18	169
6145	2219	2845	315	0	0	0	19	168
6216	2191	3346	371	0	0	0	18	168

12. Parametre Seti:  $ps = 40$ ,  $eo = 0.1$ ,  $co = 0.6$ ,  $mo = 0.1$ ,  $mt = 4$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.28’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.29’da verilmiştir.

**Tablo 5.28** Yüksek ceza katsayısı modeli ile 12. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21095	6554	146	84	0	15	19	28	298
19783	7000	165	95	0	22	21	23	295
22394	5898	193	111	0	9	29	23	291
21920	5982	167	97	0	13	18	30	291
21535	6728	157	91	0	20	28	19	277

Tablo 5.28’de verilen deney sonuçlarına göre, ortalama 166 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 6432, bu uygunluk değerlerinin standart sapması ise 478 olmuştur.

Tablo 5.29’da verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 374 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2752, bu uygunluk değerlerinin standart sapması ise 163 olmuştur.

**Tablo 5.29** Tamir fonksiyonu modeli ile 12. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6006	2795	307	60	0	3	8	10	176
6166	2495	414	81	0	0	3	16	180
5753	2946	446	88	0	5	7	8	178
6243	2778	404	79	0	1	12	12	167
6084	2746	298	58	0	0	13	8	175

Şekil 5.12’de 12. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

13. Parametre Seti:  $ps = 40$ ,  $eo = 0.2$ ,  $co = 0.9$ ,  $mo = 0.01$ ,  $mt = 1$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.30’da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.31’de verilmiştir.

**Tablo 5.30** Yüksek ceza katsayısı modeli ile 13. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21268	4377	334	302	0	0	0	49	290
22684	4407	359	324	0	0	1	49	293
21206	4446	355	321	0	0	7	55	253
21562	4492	386	349	0	1	6	50	265
22338	4339	358	324	0	0	2	48	277

Tablo 5.30’da verilen deney sonuçlarına göre, ortalama 358 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4412, bu uygunluk değerlerinin standart sapması ise 59 olmuştur.

Tablo 5.31’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 728 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2246, bu uygunluk değerlerinin standart sapması ise 35 olmuştur.

Şekil 5.13'de 13. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.31** Tamir fonksiyonu modeli ile 13. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6046	2297	588	224	0	0	5	14	165
6374	2249	697	266	0	0	0	21	162
6361	2208	732	280	0	0	1	20	165
5593	2220	760	291	0	0	0	17	166
5794	2257	863	331	0	0	2	19	164

14. Parametre Seti:  $ps = 40$ ,  $eo = 0.2$ ,  $co = 0.9$ ,  $mo = 0.01$ ,  $mt = 2$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.32'de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.33'de verilmiştir.

**Tablo 5.32** Yüksek ceza katsayısı modeli ile 14. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21166	4635	290	308	0	0	8	49	283
21688	4738	280	297	0	0	16	36	291
20362	4827	261	277	0	6	5	36	292
21146	4975	216	229	0	6	11	42	277
22843	4697	316	336	0	0	16	32	292

Tablo 5.32'de verilen deney sonuçlarına göre, ortalama 273 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4774, bu uygunluk değerlerinin standart sapması ise 132 olmuştur.

Tablo 5.33'de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 403 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2429, bu uygunluk değerlerinin standart sapması ise 43 olmuştur.

**Tablo 5.33** Tamir fonksiyonu modeli ile 14. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6173	2487	351	180	0	0	4	14	179
6102	2410	391	203	0	0	4	14	176
6055	2421	421	218	0	0	1	16	175
5987	2452	450	232	0	0	7	13	171
5764	2374	404	210	0	0	2	15	177

Şekil 5.14’de 14. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

15. Parametre Seti:  $ps = 40$ ,  $eo = 0.2$ ,  $co = 0.9$ ,  $mo = 0.01$ ,  $mt = 4$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.34’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.35’de verilmiştir.

**Tablo 5.34** Yüksek ceza katsayısı modeli ile 15. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21247	7641	95	110	0	26	30	26	308
19401	8300	47	55	0	36	37	9	295
21841	6222	75	87	0	11	29	29	284
21859	8326	84	97	1	22	24	25	312
21829	7109	56	65	0	18	29	29	332

Tablo 5.34’da verilen deney sonuçlarına göre, ortalama 71 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 7520, bu uygunluk değerlerinin standart sapması ise 884 olmuştur.

Tablo 5.35’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 105 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 3191, bu uygunluk değerlerinin standart sapması ise 188 olmuştur.

Şekil 5.15’de 15. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.35** Tamir fonksiyonu modeli ile 15. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6272	3048	133	82	0	6	9	14	174
5539	3086	76	46	0	2	16	11	178
5577	3191	131	82	0	7	13	8	178
6249	3116	106	63	0	4	16	10	177
6184	3515	80	47	0	7	10	15	174

16. Parametre Seti:  $ps = 40$ ,  $eo = 0.2$ ,  $co = 0.3$ ,  $mo = 0.05$ ,  $mt = 1$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.36’da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.37’de verilmiştir.

**Tablo 5.36** Yüksek ceza katsayısı modeli ile 16. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
20965	4347	674	267	0	0	5	51	271
22958	4441	551	219	0	0	1	51	285
22435	4354	746	296	0	0	0	52	277
22228	4501	741	294	0	1	3	51	277
22642	4354	685	272	0	0	0	50	287

Tablo 5.36’da verilen deney sonuçlarına göre, ortalama 679 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4399, bu uygunluk değerlerinin standart sapması ise 69 olmuştur.

Tablo 5.37’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 1503 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2268, bu uygunluk değerlerinin standart sapması ise 32 olmuştur.

Şekil 5.16'da 16. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.37** Tamir fonksiyonu modeli ile 16. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6321	2246	1592	189	0	0	1	15	171
5631	2305	1374	163	0	0	3	16	166
6235	2234	1605	190	0	0	4	20	151
6159	2257	1704	202	0	0	0	18	172
6124	2298	1241	147	0	0	4	12	170

17. Parametre Seti:  $ps = 40$ ,  $eo = 0.2$ ,  $co = 0.3$ ,  $mo = 0.05$ ,  $mt = 2$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.38'de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.39'da verilmiştir.

**Tablo 5.38** Yüksek ceza katsayısı modeli ile 17. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21371	4319	1114	738	0	0	1	51	281
18461	4471	753	500	0	0	0	48	290
19358	4463	1018	673	0	0	3	51	269
20760	4461	1015	674	0	0	3	51	275
20389	4362	917	609	0	0	1	52	277

Tablo 5.38'de verilen deney sonuçlarına göre, ortalama 963 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4415, bu uygunluk değerlerinin standart sapması ise 70 olmuştur.

Tablo 5.39'da verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 2687 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2231, bu uygunluk değerlerinin standart sapması ise 27 olmuştur.

Şekil 5.17’de 17. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.39** Tamir fonksiyonu modeli ile 17. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6254	2220	2597	576	0	0	2	16	165
5758	2241	2191	485	0	0	0	22	157
6444	2242	3142	699	0	0	1	16	168
6148	2262	2596	575	0	0	0	21	161
6014	2190	2907	646	0	0	2	16	164

18. Parametre Seti:  $ps = 40$ ,  $eo = 0.2$ ,  $co = 0.3$ ,  $mo = 0.05$ ,  $mt = 4$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.40’da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.41’de verilmiştir.

**Tablo 5.40** Yüksek ceza katsayısı modeli ile 18. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
22231	5921	186	175	0	16	18	27	276
20080	5899	170	158	0	11	27	29	284
20751	5957	159	148	0	10	29	19	297
20571	6753	100	94	0	15	28	24	313
20944	6141	142	133	0	12	25	31	272

Tablo 5.40’da verilen deney sonuçlarına göre, ortalama 151 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 6134, bu uygunluk değerlerinin standart sapması ise 359 olmuştur.

Tablo 5.41’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 324 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2957, bu uygunluk değerlerinin standart sapması ise 214 olmuştur.



Şekil 5.18’de 18. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.41** Tamir fonksiyonu modeli ile 18. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6468	2958	428	159	0	4	7	10	181
6064	3314	287	106	0	7	14	11	159
5592	2811	279	103	0	4	8	13	169
6168	2776	394	146	0	4	10	9	174
5925	2925	230	85	0	5	4	14	174

19. Parametre Seti:  $ps = 80$ ,  $eo = 0.1$ ,  $co = 0.9$ ,  $mo = 0.05$ ,  $mt = 1$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.42’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.43’de verilmiştir.

**Tablo 5.42** Yüksek ceza katsayısı modeli ile 19. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21528	4375	1962	312	0	0	2	53	280
20789	4362	1589	253	0	0	0	58	267
20361	4375	1489	237	0	0	0	50	283
20605	4364	1684	268	0	0	0	55	272
19445	4292	1453	231	0	0	0	53	272

Tablo 5.42’de verilen deney sonuçlarına göre, ortalama 1635 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4354, bu uygunluk değerlerinin standart sapması ise 35 olmuştur.

Tablo 5.43’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 3683 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2236, bu uygunluk değerlerinin standart sapması ise 15 olmuştur.

Şekil 5.19’da 19. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.43** Tamir fonksiyonu modeli ile 19. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6241	2254	3737	191	0	0	2	19	161
5828	2218	3166	162	0	0	0	17	167
6102	2249	3460	177	0	0	1	18	165
5592	2227	4065	208	0	0	1	19	166
5882	2231	3987	204	0	0	1	18	161

20. Parametre Seti:  $ps = 80$ ,  $eo = 0.1$ ,  $co = 0.9$ ,  $mo = 0.05$ ,  $mt = 2$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.44’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.45’de verilmiştir.

**Tablo 5.44** Yüksek ceza katsayısı modeli ile 20. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
19629	4443	572	158	0	0	6	43	296
20409	4722	463	128	0	2	5	44	286
20442	4409	625	173	0	0	2	48	287
19832	4482	536	148	0	1	10	40	282
19194	4581	489	135	0	1	6	41	289

Tablo 5.44’de verilen deney sonuçlarına göre, ortalama 537 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4527, bu uygunluk değerlerinin standart sapması ise 126 olmuştur.

Tablo 5.45’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 1261 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2310, bu uygunluk değerlerinin standart sapması ise 58 olmuştur.

Şekil 5.20’de 20. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.45** Tamir fonksiyonu modeli ile 20. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6329	2376	902	86	0	0	5	13	171
6149	2230	1658	158	0	0	0	16	169
6113	2279	1313	125	0	0	3	12	172
5986	2312	1336	128	0	0	7	11	168
5622	2352	1098	105	0	0	3	14	176

21. Parametre Seti:  $ps = 80$ ,  $eo = 0.1$ ,  $co = 0.9$ ,  $mo = 0.05$ ,  $mt = 4$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.46’da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.47’de verilmiştir.

**Tablo 5.46** Yüksek ceza katsayısı modeli ile 21. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21159	5007	609	233	0	2	20	30	289
21726	5113	527	202	0	6	11	40	277
20188	5251	491	188	0	9	18	24	286
21648	5712	522	200	0	11	20	27	287
19596	5218	525	201	0	9	16	27	290

Tablo 5.46’da verilen deney sonuçlarına göre, ortalama 535 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 5260, bu uygunluk değerlerinin standart sapması ise 270 olmuştur.

Tablo 5.47’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 1262 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2570, bu uygunluk değerlerinin standart sapması ise 81 olmuştur.

Şekil 5.21’de 21. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.47** Tamir fonksiyonu modeli ile 21. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6055	2657	1030	157	0	2	7	11	170
6097	2635	1421	216	0	3	3	17	169
5971	2584	1055	160	0	0	6	14	172
6017	2469	1436	219	0	2	4	14	174
5926	2504	1366	208	0	2	9	8	169

22. Parametre Seti:  $ps = 80$ ,  $eo = 0.2$ ,  $co = 0.3$ ,  $mo = 0.1$ ,  $mt = 1$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.48’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.49’da verilmiştir.

**Tablo 5.48** Yüksek ceza katsayısı modeli ile 22. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21889	4331	2838	212	0	0	0	61	267
19784	4330	2354	176	0	0	0	61	255
17544	4316	3064	229	0	0	0	49	281
21914	4390	2972	221	0	0	2	50	273
21538	4359	2518	188	0	0	0	53	276

Tablo 5.48’de verilen deney sonuçlarına göre, ortalama 2749 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4345, bu uygunluk değerlerinin standart sapması ise 30 olmuştur.

Tablo 5.49’da verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 6353 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2218, bu uygunluk değerlerinin standart sapması ise 20 olmuştur.

Şekil 5.22’de 22. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.49** Tamir fonksiyonu modeli ile 22. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
5941	2249	6520	159	0	0	0	20	164
6068	2208	5612	137	0	0	1	19	156
6248	2199	8254	201	0	0	1	21	156
6186	2229	5904	144	0	0	1	17	166
5697	2207	6527	159	0	0	1	19	158

23. Parametre Seti:  $ps = 80$ ,  $eo = 0.2$ ,  $co = 0.3$ ,  $mo = 0.1$ ,  $mt = 2$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.50’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.51’de verilmiştir.

**Tablo 5.50** Yüksek ceza katsayısı modeli ile 23. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
19040	4468	1468	242	0	0	3	55	267
17997	4343	1731	287	0	0	0	56	260
20078	4491	1628	269	0	1	1	44	291
21924	4515	2132	352	0	0	5	45	278
20726	4360	1631	269	0	0	2	59	257

Tablo 5.50’de verilen deney sonuçlarına göre, ortalama 1718 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4435, bu uygunluk değerlerinin standart sapması ise 79 olmuştur.

Tablo 5.51’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 4264 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2236, bu uygunluk değerlerinin standart sapması ise 30 olmuştur.

Şekil 5.23’de 23. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.51** Tamir fonksiyonu modeli ile 23. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6197	2200	4138	217	0	0	0	20	160
6163	2249	4098	215	0	0	0	18	162
6111	2280	3929	206	0	0	2	14	176
5847	2221	4770	250	0	0	0	18	163
6057	2232	4387	229	0	0	1	17	170

24. Parametre Seti:  $ps = 80$ ,  $eo = 0.2$ ,  $co = 0.3$ ,  $mo = 0.1$ ,  $mt = 4$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.52’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.53’de verilmiştir.

**Tablo 5.52** Yüksek ceza katsayısı modeli ile 24. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
19163	5210	1261	371	0	6	14	29	305
19970	4812	2001	588	0	1	18	30	290
21572	4984	1488	439	0	2	12	39	288
20966	5031	1550	456	0	4	14	32	295
19885	5618	1730	510	0	10	15	25	303

Tablo 5.52’de verilen deney sonuçlarına göre, ortalama 1606 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 5131, bu uygunluk değerlerinin standart sapması ise 307 olmuştur.

Tablo 5.53’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 4633 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2578, bu uygunluk değerlerinin standart sapması ise 140 olmuştur.

Şekil 5.24’de 24. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.53** Tamir fonksiyonu modeli ile 24. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
5825	2696	4649	467	0	3	4	13	173
6034	2565	3743	375	0	1	5	14	170
6050	2490	5865	588	0	2	2	18	168
5919	2402	3452	345	0	2	2	13	174
5758	2738	5454	548	0	3	9	8	170

25. Parametre Seti:  $ps = 80$ ,  $eo = 0.2$ ,  $co = 0.6$ ,  $mo = 0.01$ ,  $mt = 1$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.54’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.55’de verilmiştir.

**Tablo 5.54** Yüksek ceza katsayısı modeli ile 25. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
21025	4278	765	379	0	0	0	56	261
18012	4391	788	392	0	0	0	54	278
21553	4433	967	481	0	0	4	44	295
22029	4390	872	432	0	0	0	59	268
22103	4443	899	445	0	0	3	52	281

Tablo 5.54’de verilen deney sonuçlarına göre, ortalama 858 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4387, bu uygunluk değerlerinin standart sapması ise 65 olmuştur.

Tablo 5.55’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 1625 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2238, bu uygunluk değerlerinin standart sapması ise 16 olmuştur.

Şekil 5.25’de 25. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.55** Tamir fonksiyonu modeli ile 25. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6017	2224	1501	317	0	0	0	20	166
5992	2228	1381	291	0	0	0	16	171
6209	2247	1851	389	0	0	2	19	157
6179	2262	1398	294	0	0	6	12	166
5914	2231	1992	422	0	0	7	12	160

26. Parametre Seti:  $ps = 80$ ,  $eo = 0.2$ ,  $co = 0.6$ ,  $mo = 0.01$ ,  $mt = 2$ ,  $c = 0$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.56’da, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.57’de verilmiştir.

**Tablo 5.56** Yüksek ceza katsayısı modeli ile 26. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
18875	4790	293	176	0	1	8	40	302
18928	4976	226	135	0	0	25	36	281
20770	5189	232	139	0	3	18	31	296
19925	4767	335	200	0	0	8	44	300
19850	4812	271	163	0	3	8	38	306

Tablo 5.56’da verilen deney sonuçlarına göre, ortalama 271 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 4907, bu uygunluk değerlerinin standart sapması ise 178 olmuştur.

Tablo 5.57’de verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 348 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2493, bu uygunluk değerlerinin standart sapması ise 53 olmuştur.



Şekil 5.26’de 26. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.57** Tamir fonksiyonu modeli ile 26. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6351	2467	349	104	0	0	4	16	171
6177	2495	348	104	0	0	7	12	166
6140	2578	304	90	0	0	8	11	175
6163	2434	397	119	0	0	5	13	175
6140	2491	344	103	0	0	9	7	178

27. Parametre Seti:  $ps = 80$ ,  $eo = 0.2$ ,  $co = 0.6$ ,  $mo = 0.01$ ,  $mt = 4$ ,  $c = 3$ ; olmak üzere, yüksek ceza katsayısı modeli ile yapılan beş ayrı denemede elde edilen sonuçlar Tablo 5.58’de, gömülü tamir fonksiyonu modeli ile yapılan beş farklı denemede elde edilen sonuçlar ise Tablo 5.59’da verilmiştir.

**Tablo 5.58** Yüksek ceza katsayısı modeli ile 27. parametre seti deney sonuçları

İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
20409	5175	474	314	0	9	17	30	293
20384	6004	409	271	0	11	23	29	295
21060	6212	427	283	0	14	26	19	288
18305	5599	469	311	0	5	19	36	303
19375	5565	465	309	0	8	26	16	313

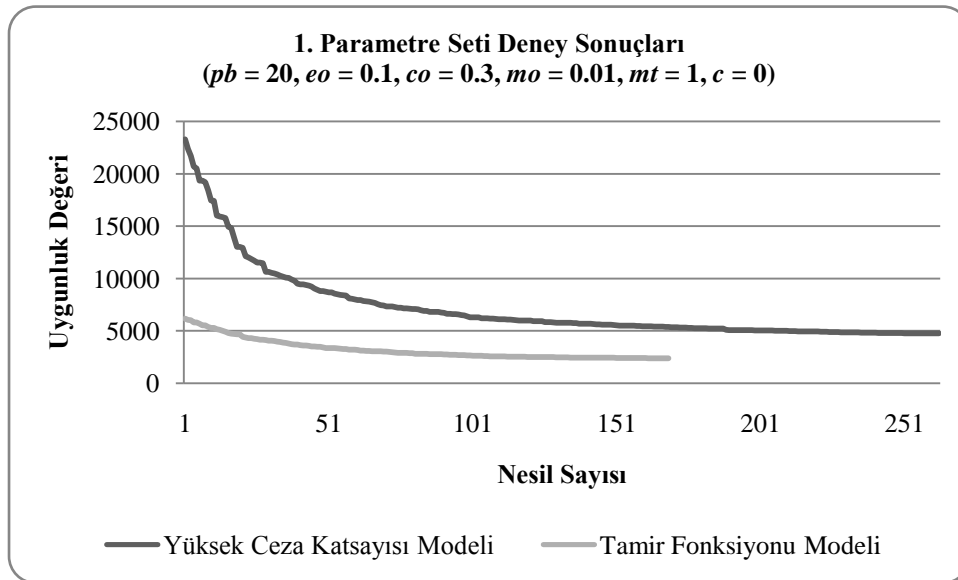
Tablo 5.58’de verilen deney sonuçlarına göre, ortalama 449 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 5711, bu uygunluk değerlerinin standart sapması ise 406 olmuştur.

Tablo 5.59’da verilen deney sonuçlarına göre, tamir fonksiyonu modeli ile ortalama 616 saniye süren beş denemede son nesildeki en iyi uygunluk değerleri ortalaması 2582, bu uygunluk değerlerinin standart sapması ise 106 olmuştur.

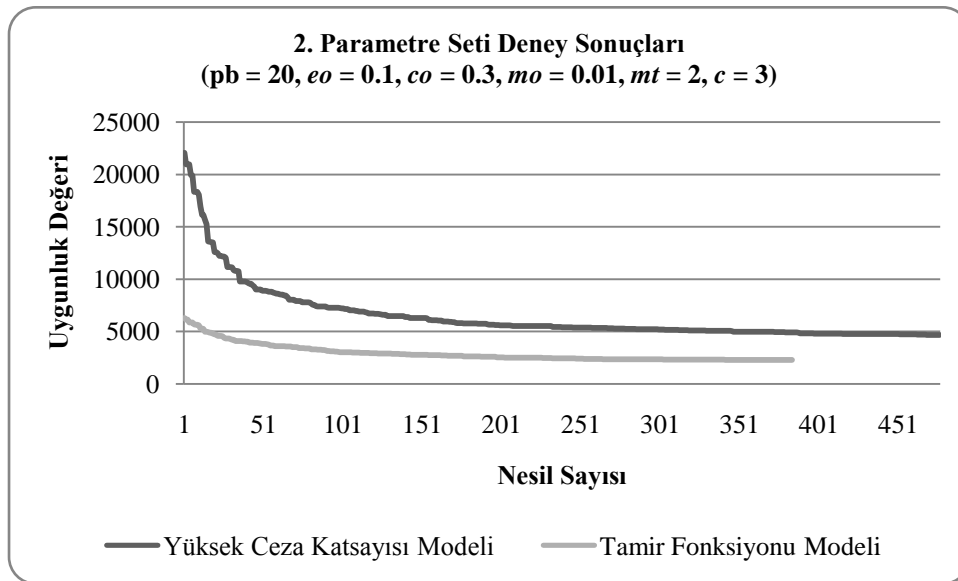
Şekil 5.27’de 27. parametre setinin yüksek ceza katsayısı modeli ile tamir fonksiyonu modeline uygulanması sonucu ortaya çıkan uygunluk değerleri değişimi verilmiştir.

**Tablo 5.59** Tamir fonksiyonu modeli ile 27. parametre seti deney sonuçları

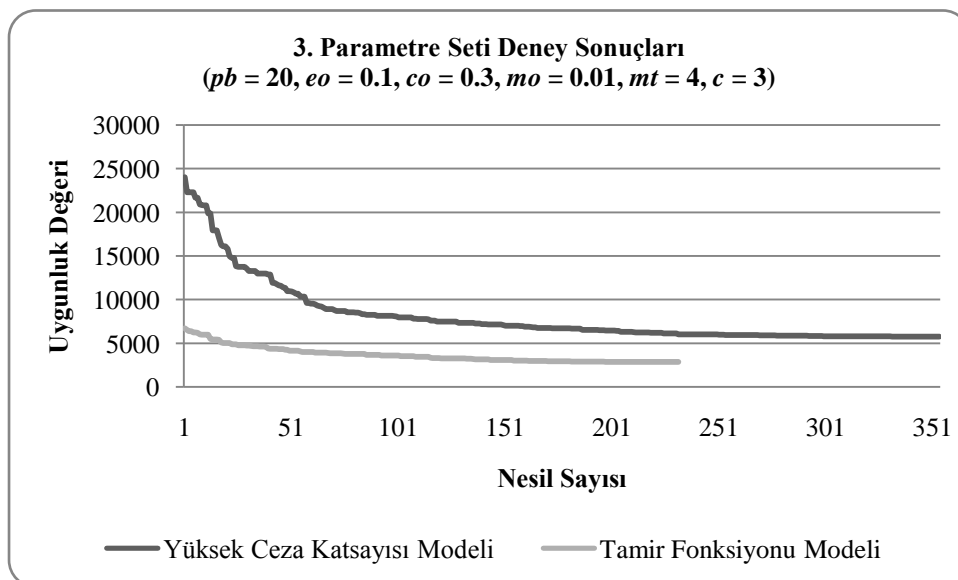
İlk nesildeki en iyi uygunluk değeri	Son nesildeki en iyi uygunluk değeri	Toplam Süre (saniye)	Nesil Sayısı	Çakışan Toplam Sınav Sayısı	Arka-arkaya yapılan sınav sayısı	Bir dinlenme dilimi ile yapılan sınav sayısı	İki dinlenme dilimi ile yapılan sınav sayısı	Bir gün ara ile yapılan toplam sınav sayısı
6357	2697	518	195	0	5	5	9	174
5978	2520	617	233	0	2	7	10	178
5775	2562	599	226	0	1	10	11	173
5616	2449	651	247	0	1	6	11	175
5647	2681	695	266	0	4	7	10	167



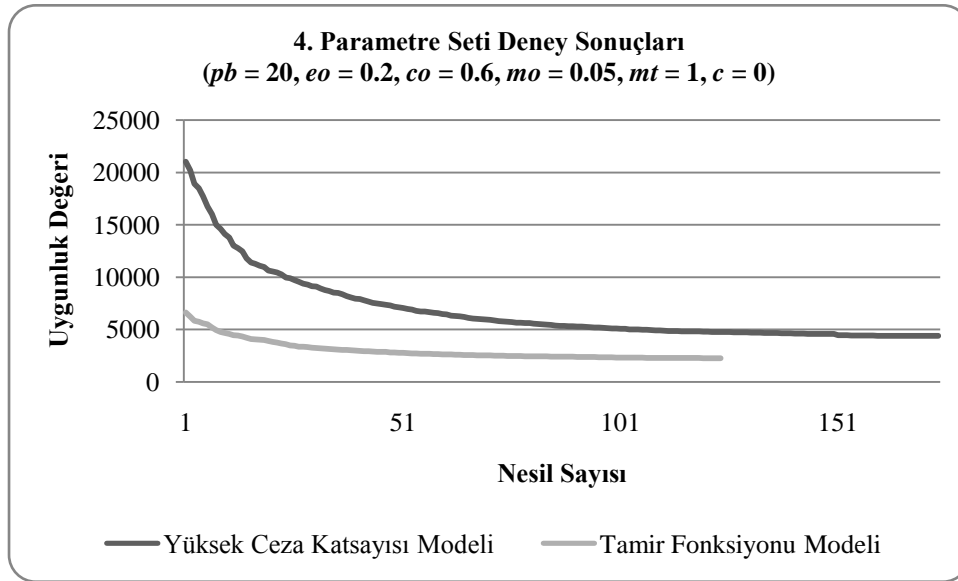
**Şekil 5.1** 1. parametre seti deney sonuçları



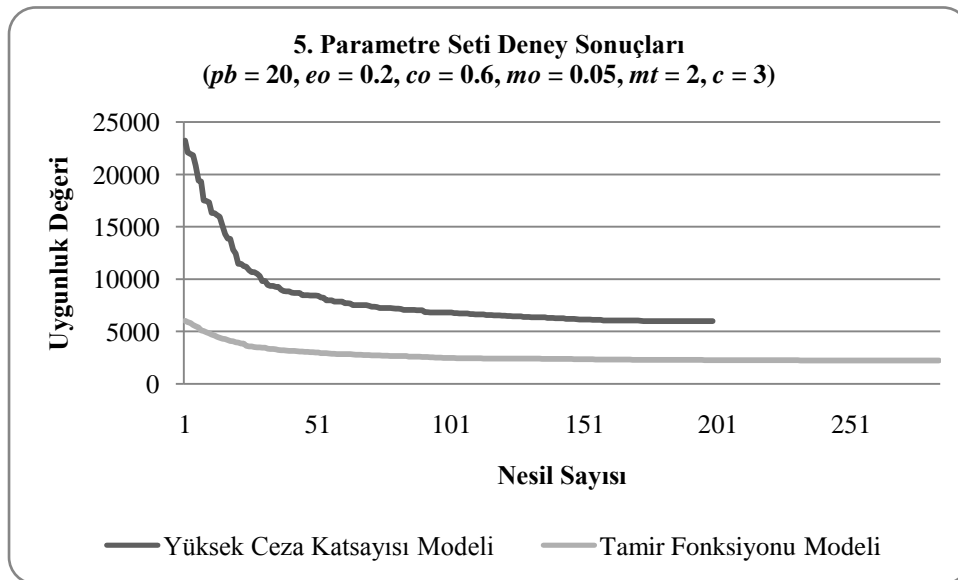
Şekil 5.2 2. parametre seti deney sonuçları



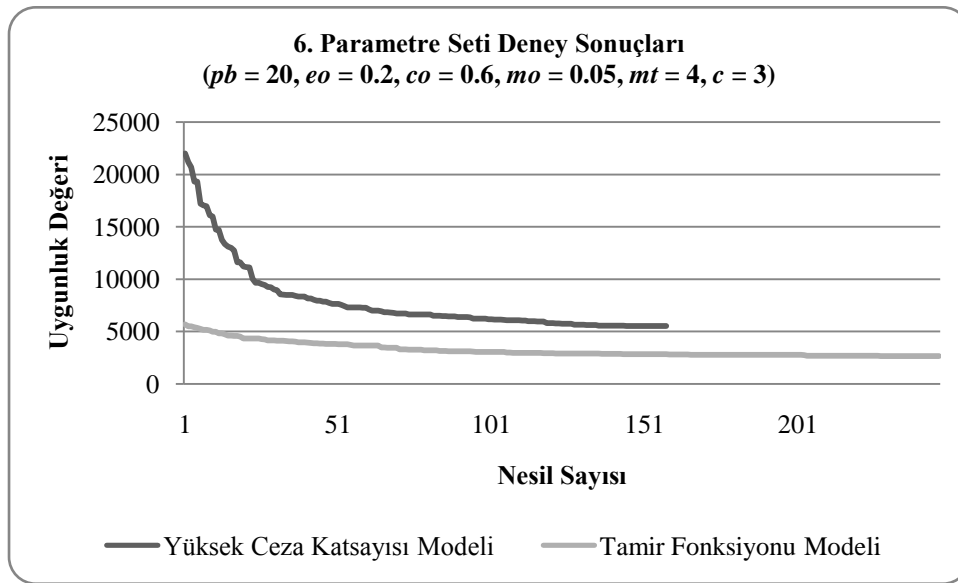
Şekil 5.3 3. parametre seti deney sonuçları



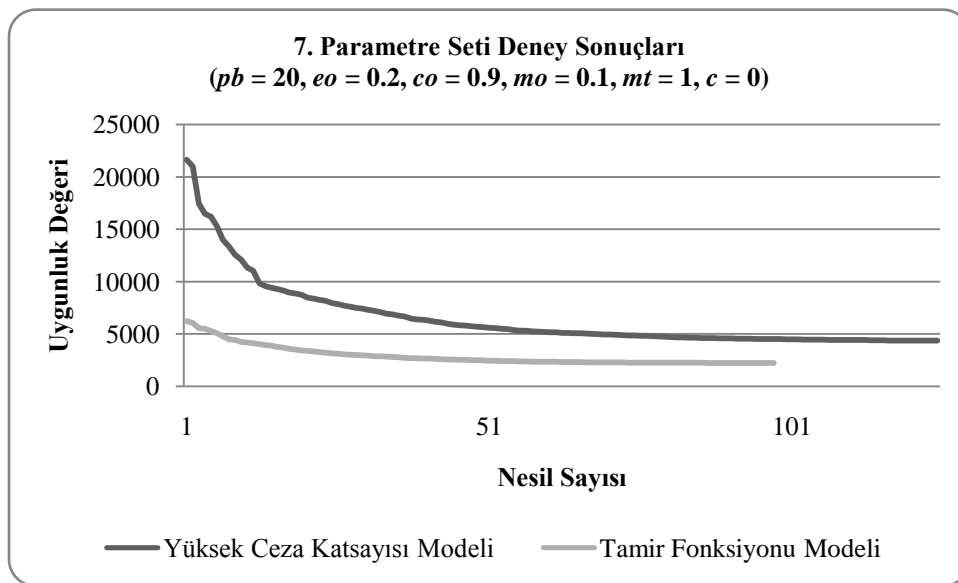
Şekil 5.4 4. parametre seti deney sonuçları



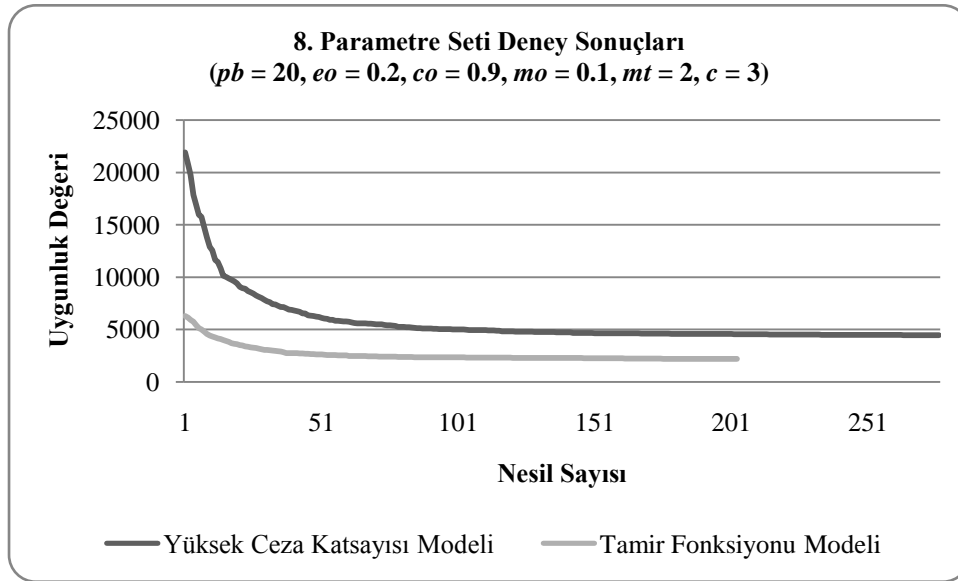
Şekil 5.5 5. parametre seti deney sonuçları



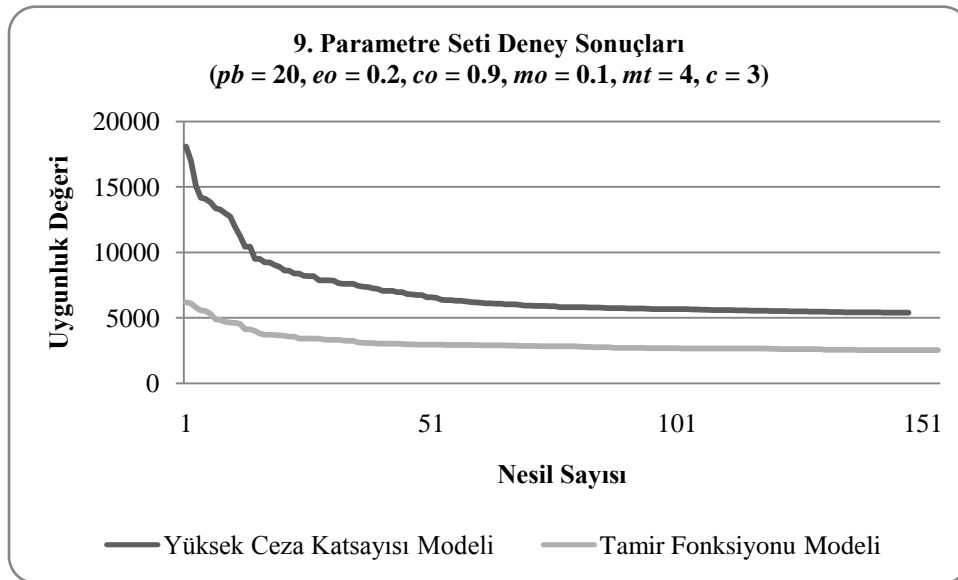
Şekil 5.6 6. parametre seti deney sonuçları



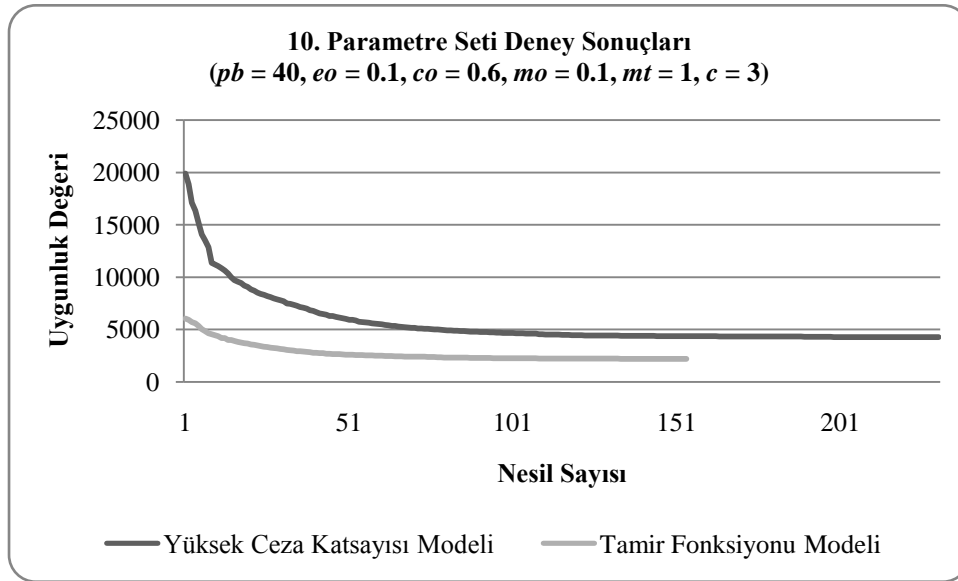
Şekil 5.7 7. parametre seti deney sonuçları



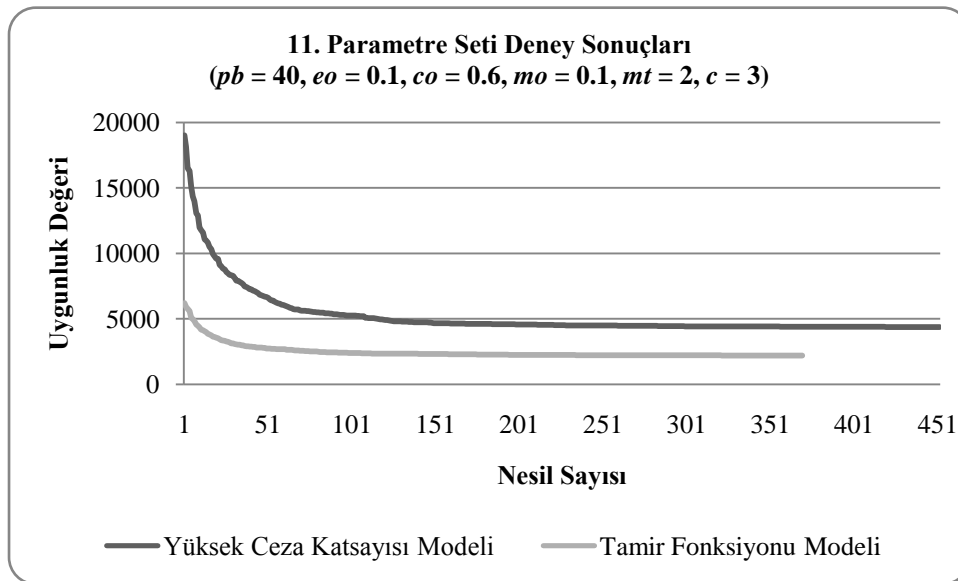
Şekil 5.8 8. parametre seti deney sonuçları



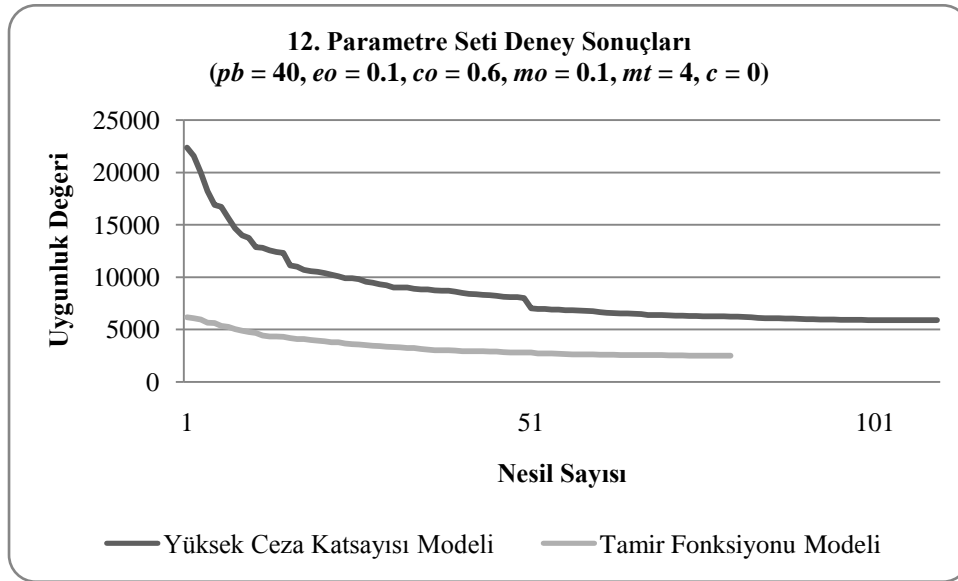
Şekil 5.9 9. parametre seti deney sonuçları



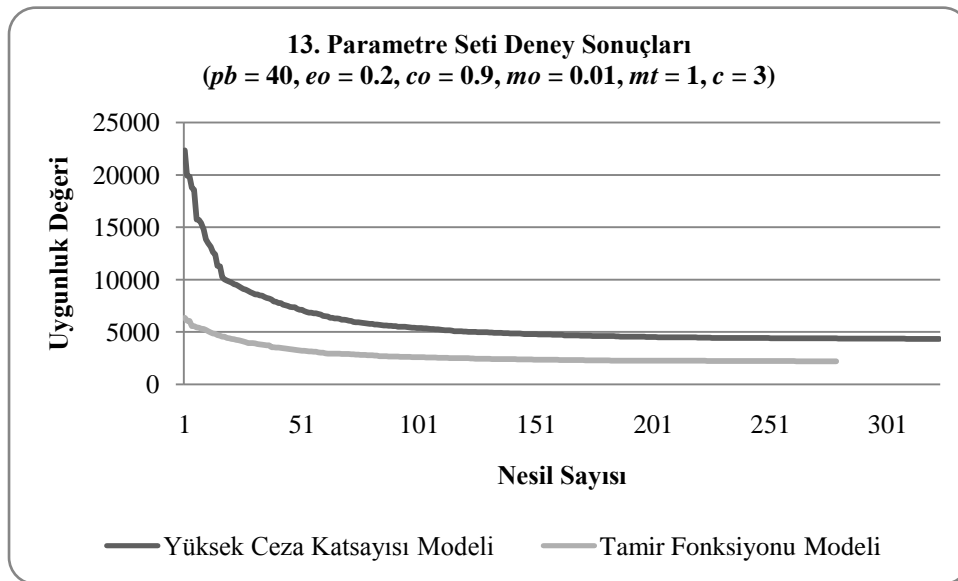
**Şekil 5.10** 10. parametre seti deney sonuçları



**Şekil 5.11** 11. parametre seti deney sonuçları

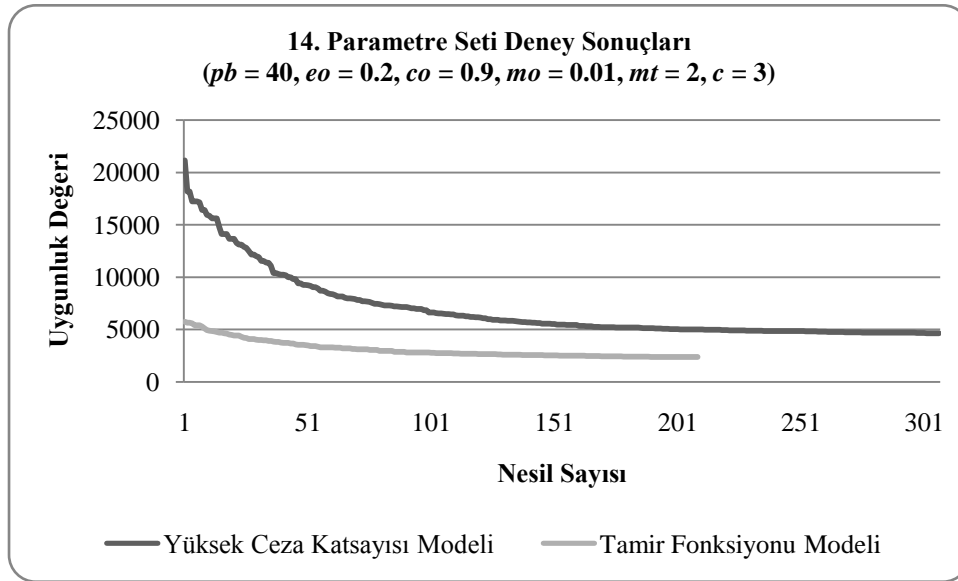


Şekil 5.12 12. parametre seti deney sonuçları

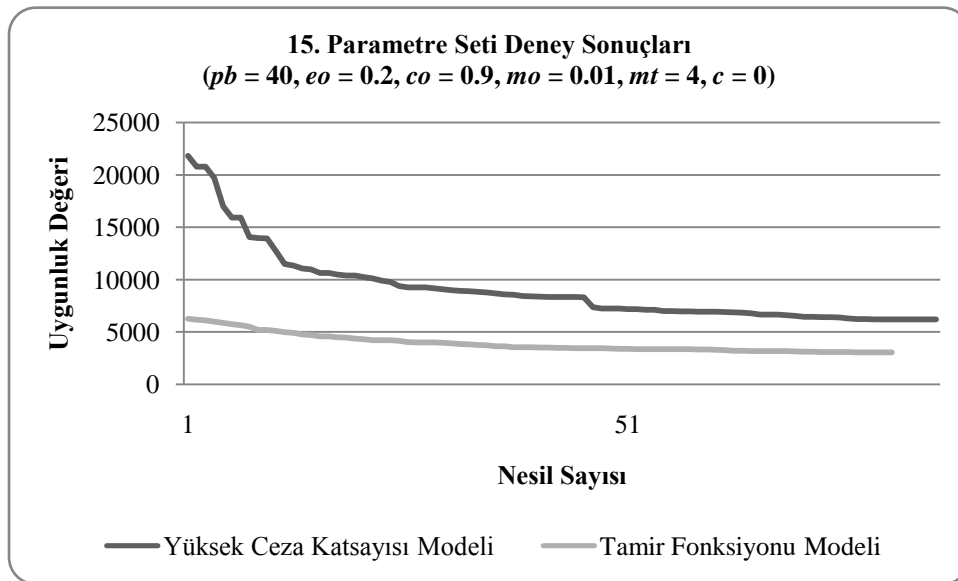


Şekil 5.13 13. parametre seti deney sonuçları

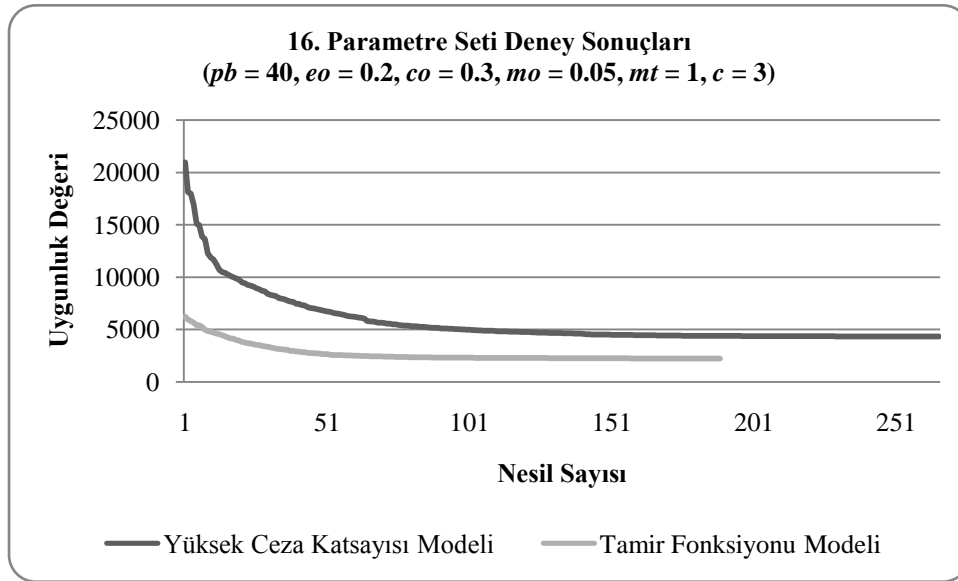




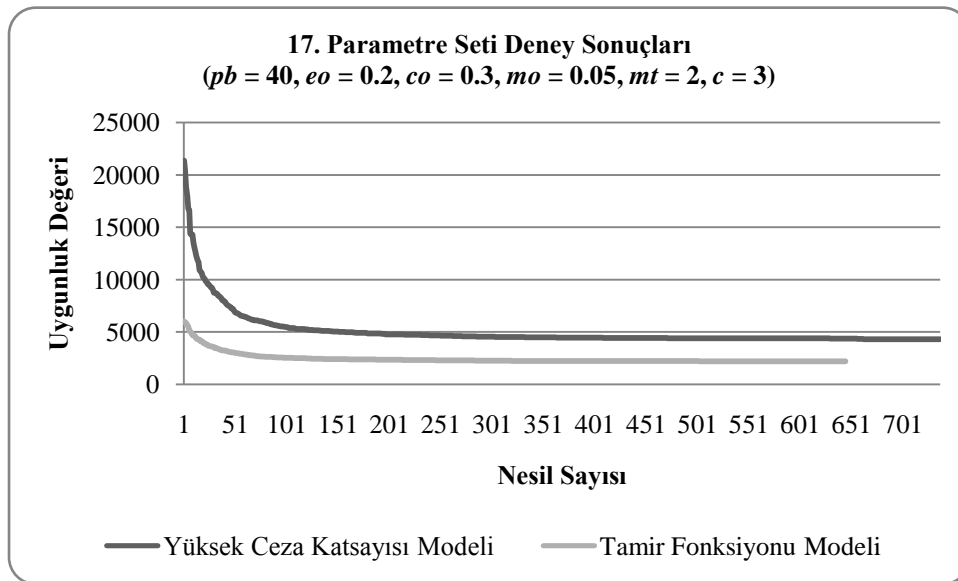
**Şekil 5.14** 14. parametre seti deney sonuçları



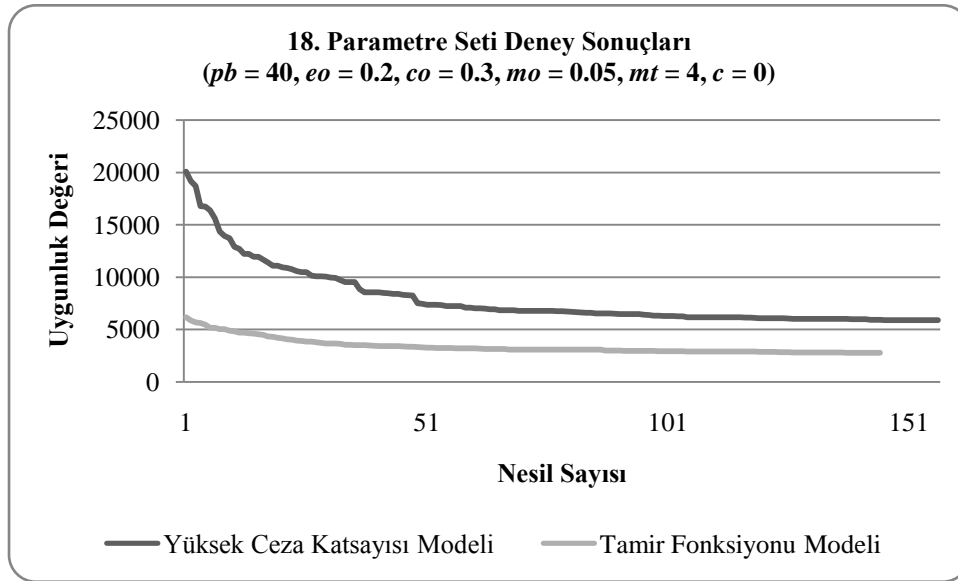
**Şekil 5.15** 15. parametre seti deney sonuçları



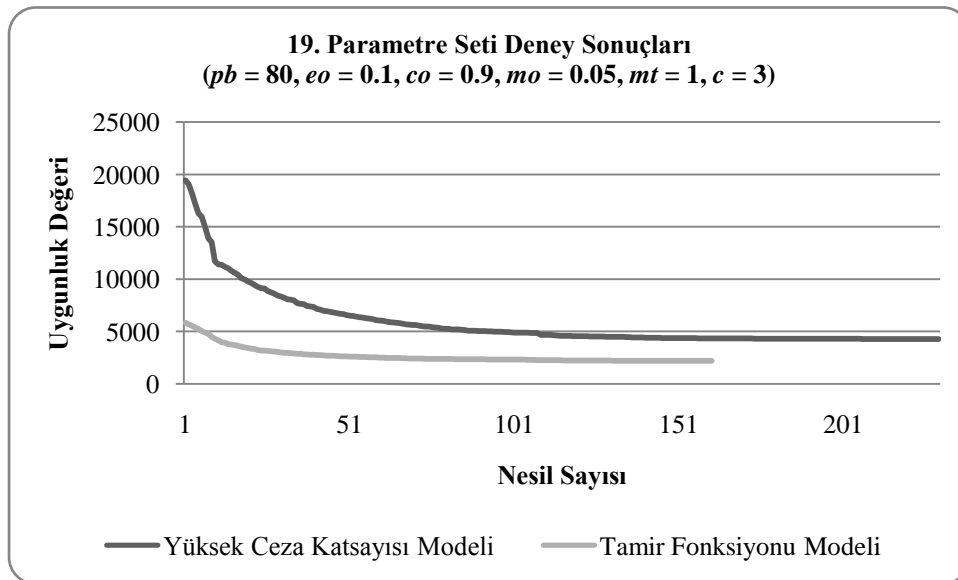
Şekil 5.16 16. parametre seti deney sonuçları



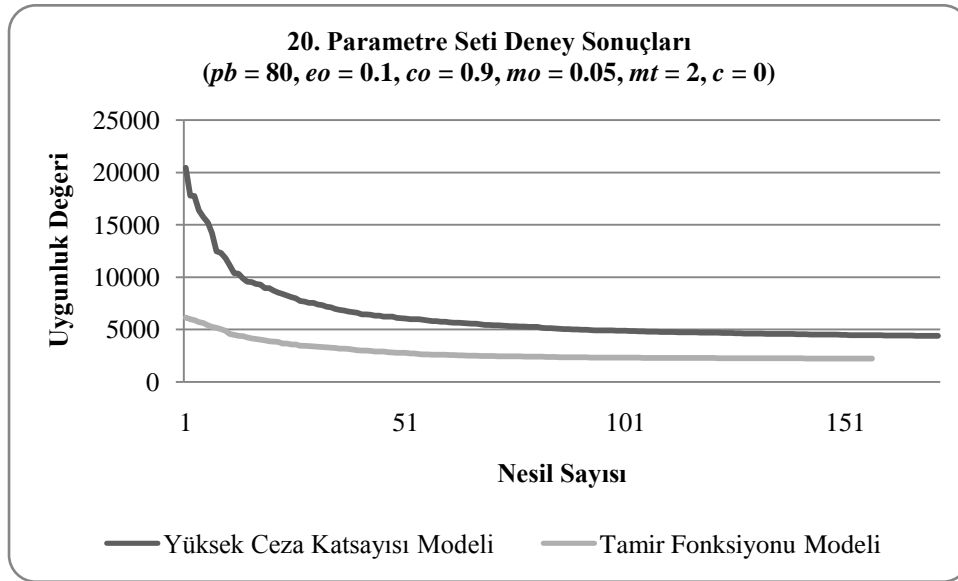
Şekil 5.17 17. parametre seti deney sonuçları



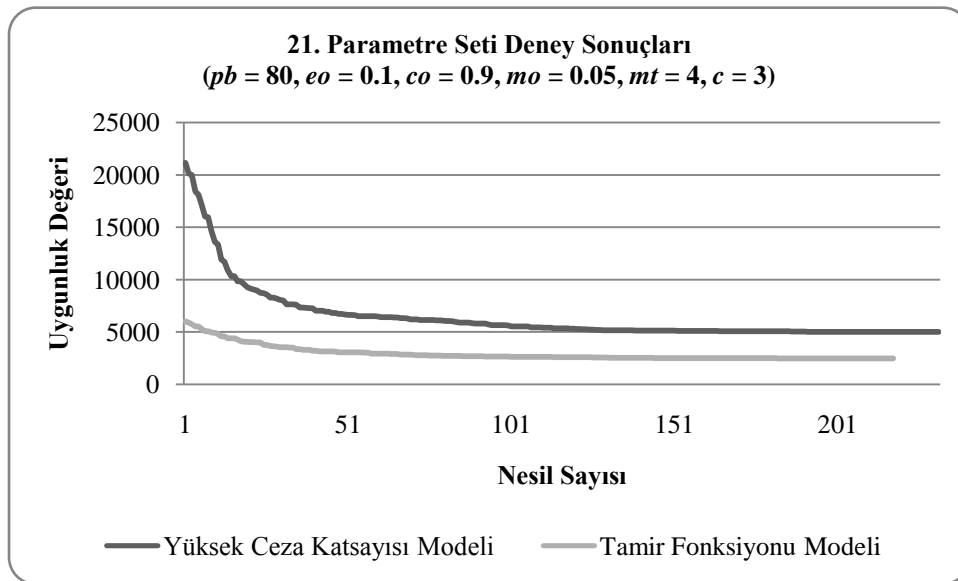
Şekil 5.18 18. parametre seti deney sonuçları



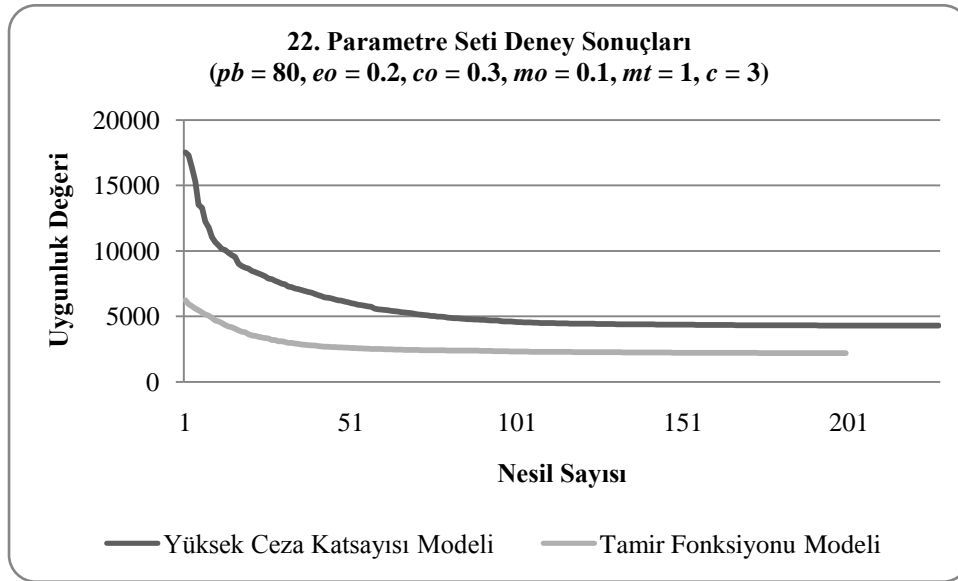
Şekil 5.19 19. parametre seti deney sonuçları



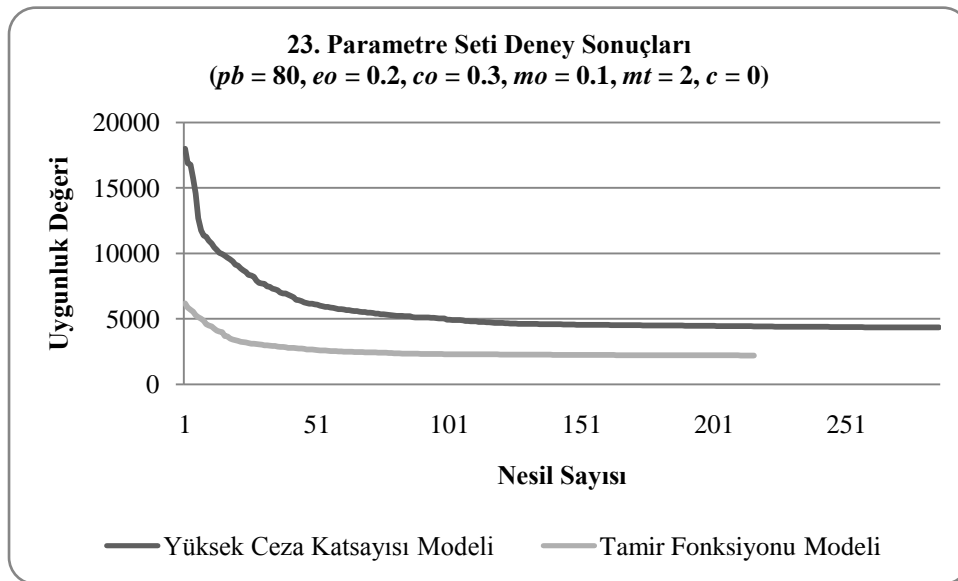
Şekil 5.20 20. parametre seti deney sonuçları



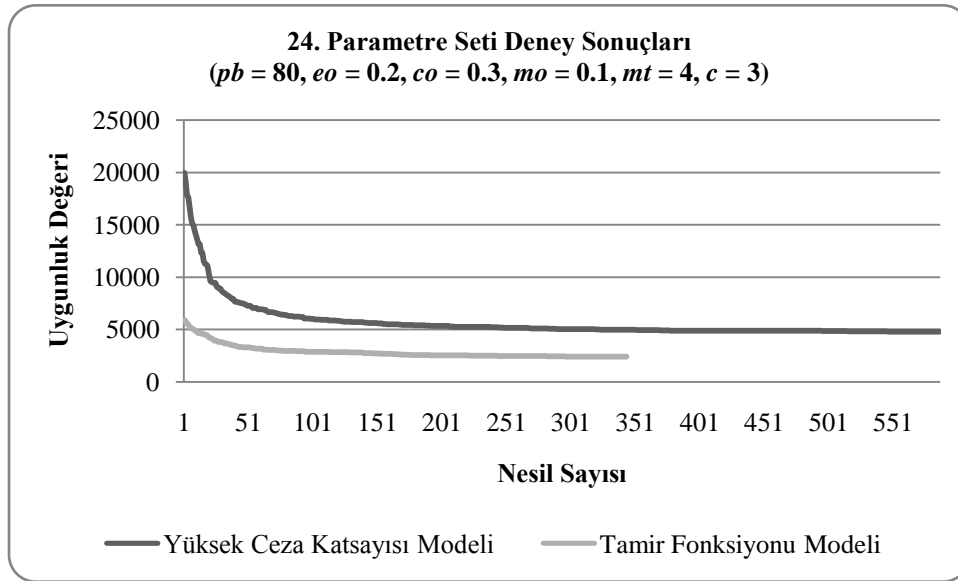
Şekil 5.21 21. parametre seti deney sonuçları



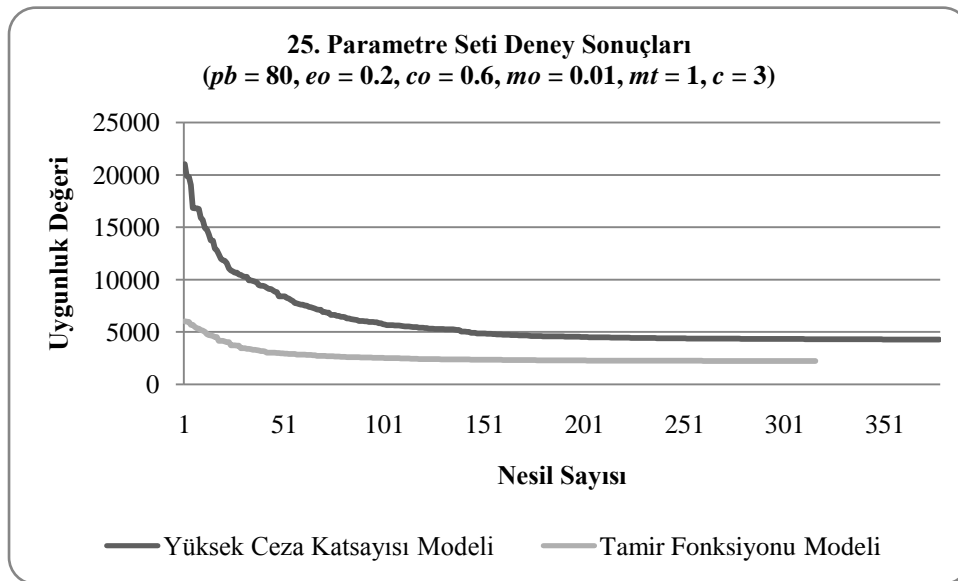
Şekil 5.22 22. parametre seti deney sonuçları



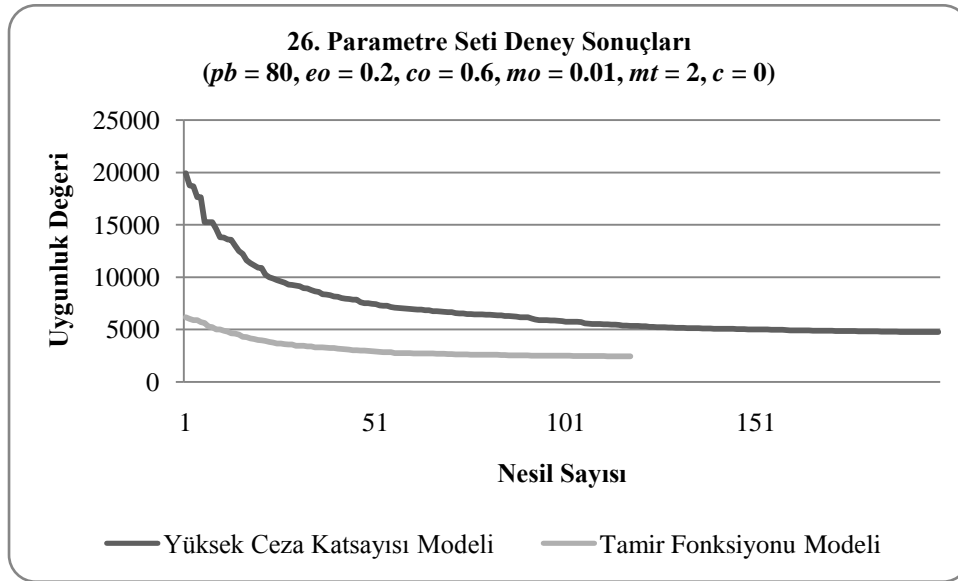
Şekil 5.23 23. parametre seti deney sonuçları



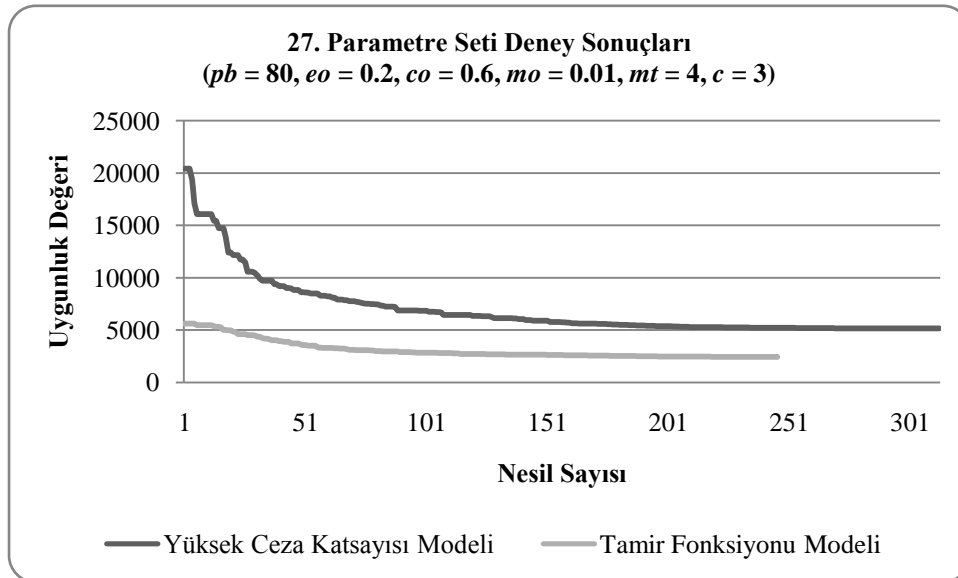
Şekil 5.24 24. parametre seti deney sonuçları



Şekil 5.25 25. parametre seti deney sonuçları



**Şekil 5.26** 26. parametre seti deney sonuçları



**Şekil 5.27** 27. parametre seti deney sonuçları

Tablo 5.60'da yüksek ceza katsayısı modeli ile parametre seti sonuçları son nesildeki uygunluk değerleri ortalamalarına göre küçükten büyüğe doğru sıralanarak gösterilmiştir. Bu sonuçlara göre; yüksek ceza katsayısı modeli ile en iyi uygunluk değeri ortalamasını 10. parametre seti vermiştir. 22. Parametre seti ile yaklaşık aynı uygunluk değeri ortalamasına ulaşılması ve daha düşük uygunluk değeri standart sapması elde edilmiştir. Ancak 22. Parametre seti 10. Parametre setine göre çok daha fazla sürede sonuç vermesi nedeniyle daha fazla karmaşıklık düzeyine sahip olduğu anlaşılmaktadır. Bu nedenle, yüksek ceza katsayısı modeli için popülasyon

büyükliğünün 40, elit oranının 0.1, çaprazlama oranının 0.6, mutasyon oranının 0.1, mutasyon taktiğinin 1 ve standart sapma katsayısının 3 olduğu, 10. Parametre seti en iyi parametre seti olarak değerlendirilmiştir.

Tablo 5.61’de tamir fonksiyonu modeli ile parametre seti sonuçları son nesildeki uygunluk değerleri ortalamalarına göre küçükten büyüğe doğru sıralanarak gösterilmiştir. Bu sonuçlara göre; tamir fonksiyonu modeli ile en iyi uygunluk değeri ortalamasını 10. parametre seti vermiştir. 22. Parametre seti ile yaklaşık aynı uygunluk değeri ortalamasına ulaşılmasına rağmen, elde edilen uygunluk değerlerinin daha büyük standart sapma değerleri ve daha büyük karmaşıklık düzeyi nedeniyle daha düşük bir performans sergilemiştir. Bu nedenle, tamir fonksiyonu modeli için de popülasyon büyüklüğünün 40, elit oranının 0.1, çaprazlama oranının 0.6, mutasyon oranının 0.1, mutasyon taktiğinin 1 ve standart sapma katsayısının 3 olduğu, 10. Parametre seti en iyi parametre seti olarak değerlendirilmiştir.

Tablo 5.62’de yüksek ceza katsayısı modeli ile tamir fonksiyonu modeli ürettikleri uygunluk değerlerine göre karşılaştırılmıştır. Bu sonuçlara göre her parametre seti için tamir fonksiyonu modeli ile daha düşük standart sapma değerleri ile daha iyi uygunluk değerleri ortalamalarına ulaşıldığı görülmektedir. Hatta tamir fonksiyonun en kötü parametre seti ile bile yüksek ceza katsayısını modelinin en iyi parametre seti ile elde edilen uygunluk değeri ortalamasından çok daha iyi uygunluk değeri ortalamasına ulaşıldığı görülmektedir.



**Tablo 5.60** Yüksek ceza katsayısı modeli ile parametre setleri performansları sıralaması

Yüksek Ceza Katsayısı Modeli				
Parametre seti	Son Nesildeki Uygunluk Değerleri Ortalaması	Son Nesildeki Uygunluk Değerleri Standart Sapması	Ortalama Süre (Saniye)	Ortalama Nesil Sayısı
10	4344	58	1084	200
22	4345	30	2749	205
19	4354	35	1635	260
25	4387	65	858	426
16	4399	69	680	270
13	4412	59	359	324
17	4415	70	963	639
11	4419	36	1065	387
23	4435	79	1718	284
7	4455	68	304	129
4	4471	65	196	159
8	4526	54	333	248
20	4527	126	537	148
5	4563	68	271	339
14	4774	132	273	289
2	4859	134	171	454
26	4907	178	271	163
1	4974	158	103	224
24	5131	307	1606	473
21	5260	270	535	205
9	5633	199	157	173
27	5711	406	449	298
6	5964	376	117	200
18	6134	359	152	142
3	6257	605	117	349
12	6432	478	165	96
15	7520	884	71	83

**Tablo 5.61** Tamir fonksiyonu modeli ile parametre setleri performansları sıralaması

Tamir Fonksiyonu Modeli				
Parametre seti	Son Nesildeki Uygunluk Değerleri Ortalaması	Son Nesildeki Uygunluk Değerleri Standart Sapması	Ortalama Süre (Saniye)	Ortalama Nesil Sayısı
10	2217	16	2529	140
22	2218	20	6564	160
17	2231	27	2687	596
19	2236	15	3683	188
23	2236	30	4264	223
25	2238	16	1625	343
13	2246	35	728	278
11	2246	42	2937	326
16	2268	32	1503	178
8	2276	59	941	210
7	2291	56	849	101
5	2292	35	555	234
20	2310	58	1261	120
4	2322	51	449	107
2	2403	116	289	413
14	2429	43	403	209
1	2491	85	153	144
26	2493	53	349	104
21	2570	81	1262	192
24	2578	140	4633	465
27	2582	106	616	233
9	2714	242	340	128
12	2752	163	374	73
6	2792	135	304	206
18	2957	214	324	120
3	2984	148	128	243
15	3191	188	105	64

**Tablo 5.62** Yüksek ceza katsayısı modeli ile tamir fonksiyonu modelinin ürettikleri uygunluk değerlerine göre karşılaştırılması

Yüksek Ceza Katsayısı Modeli			Tamir Fonksiyonu Modeli		
Parametre Seti	Son Nesildeki Uygunluk Değerlerinin Ortalaması	Son Nesildeki Uygunluk Değerleri Standart Sapması	Parametre Seti	Son Nesildeki Uygunluk Değerlerinin Ortalaması	Son Nesildeki Uygunluk Değerleri Standart Sapması
10	4344	58	10	2217	16
22	4345	30	22	2218	20
19	4354	35	17	2231	27
25	4387	65	19	2236	15
16	4399	69	23	2236	30
13	4412	59	25	2238	16
17	4415	70	13	2246	35
11	4419	36	11	2246	42
23	4435	79	16	2268	32
7	4455	68	8	2276	59
4	4471	65	7	2291	56
8	4526	54	5	2292	35
20	4527	126	20	2310	58
5	4563	68	4	2322	51
14	4774	132	2	2403	116
2	4859	134	14	2429	43
26	4907	178	1	2491	85
1	4974	158	26	2493	53
24	5131	307	21	2570	81
21	5260	270	24	2578	140
9	5633	199	27	2582	106
27	5711	406	9	2714	242
6	5964	376	12	2752	163
18	6134	359	6	2792	135
3	6257	605	18	2957	214
12	6432	478	3	2984	148
15	7520	884	15	3191	188

Tablo 5.63' de genetik algoritmanın genel akış kodlaması (*gak*) verilmiştir. Bu akış, yüksek ceza katsayısı modeli ile çalıştırıldığında;

*bps*: Başlangıç popülasyonu sayısı

*cbsf*: Çaprazlanacak bireylerin seçilmesi fonksiyonunda yapılan toplam işlem sayısı

*ciyf*: Çaprazlama işleminin yapılması fonksiyonunda yapılan toplam işlem sayısı

- co*: Çaprazlama oranı
- eiksf*: En iyi kromozomların seçilmesi fonksiyonu
- ipyf*: İlk popülasyonun yaratılması fonksiyonunda yapılan toplam işlem sayısı
- kehaf*: Kromozomların eşleştirme havuzuna alınması fonksiyonunda yapılan toplam işlem sayısı
- m1*: Her kromozomda tek geni mutasyona uğratan fonksiyonda yapılan toplam işlem sayısı
- ns*: Nesil sayısı
- ps*: Popülasyon sayısı
- uhdbf*: Uygunluğu hesaplanacak dizinin belirlenmesi fonksiyonunda yapılan toplam işlem sayısı, olmak üzere;

toplam işlem sayısı 5.1 nolu denklem ile hesaplanabilmektedir. Bu denklem, 4.1, 4.5, 4.8, 4.9, 4.10, 4.11 ve 4.14 nolu denklemlerden elde edilmiştir. GA akışı içerisinde her fonksiyon çağrıldığı kadar işlem sayısına etki etmiştir. Ancak bu denklemde gösterilen işlem sayısı yüksek ceza katsayısı modeli için geçerlidir. Çünkü tamir fonksiyonu bu hesaba katılmamıştır.

$$gak = bps \times uhdbf + ipyf + 4 \times ps^2 - 2 \times ps + 6 + ns \times (kehaf + cbsf + ciyf + m1 + eiksf + 4) \quad 5.1$$

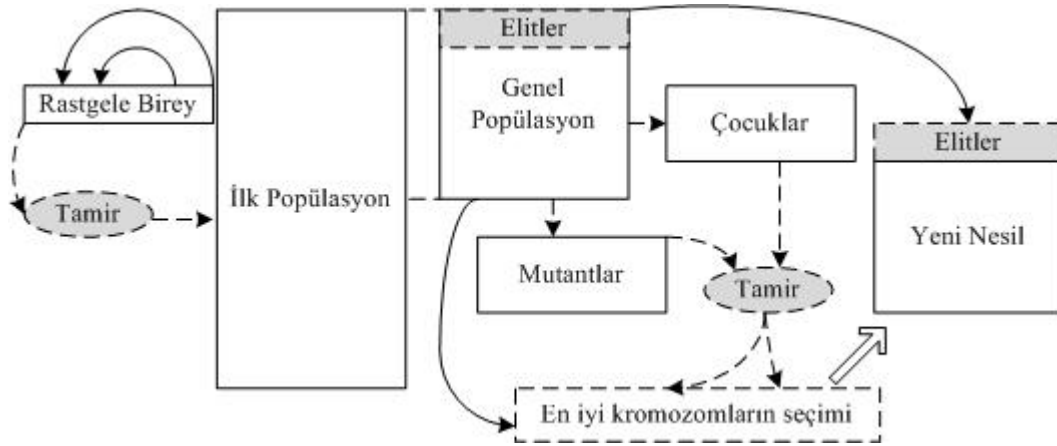
Tablo 5.63' de verilen GA genel akış kodlaması bütünleşmiş tamir fonksiyonu modeli ile çalıştırılırsa, Tablo 4.16'da verilen tamir fonksiyonu hem Tablo 4.1'de verilen ilk popülasyonun yaratılması fonksiyonundan, hem de Tablo 4.14'de verilen en iyi kromozomların seçilmesi fonksiyonu içerisinde çağırılmaktadır. İlk popülasyonun yaratılması fonksiyonu içinden çağırılması sayesinde, yaratılan ilk popülasyondaki bireylerin hepsinin bire-bir çakışmaya izin vermeyen kromozomlar olması sağlanmaktadır. Böylece GA, kabul edilebilir bireyler ile çalışmaya başlayacaktır. Çaprazlama ve mutasyon işlemlerinden elde edilen çocukların ve mutantların bütünleşmiş tamir fonksiyonu modeli mantığına göre kabul edilebilir kromozomlar olması gerekir. Bire-bir çakışma içeren kromozomların bu yapı içerisinde yeri yoktur. Kabul edilemez gen bileşimleri içeren kromozomlar bozuk olarak nitelendirilerek tamir edilmekte, tamir edilemediğinde ise yok edilmektedir. Bu mantık çerçevesinde, çaprazlama ve mutasyon işleminden elde edilen çocuk ve mutantlar, eğer gerekliyse tamir edildikten sonra seçim işlemine tabi tutulurlar. Şekil 5.28'de tamir fonksiyonu modeli ile yeni nesil seçim işlemi gösterilmiştir. Yüksek ceza katsayısı modelinden

farkı, ilk popülasyona alınan her bireyin, çaprazlama ve mutasyon işlemlerinden elde edilen çocukların ve mutantların gerekliyse tamir edilmesidir. Bu tamir fonksiyonu sayesinde hiçbir kromozom 1000 ceza katsayısı ile cezalandırılmamaktadır.

**Tablo 5.63** Genetik algoritmanın genel akış kodlaması

Pseudo Kod
<pre> SET inisize to popsize * 2 SET nooff to popsize * crossrate SET elits to popsize * elitrate  CALL <b>generateIniPop</b> with inisize RETURNING IniPop  SET nomutant to popsize * mutrate * lchoro  FOR each chromosome in the IniPop     CALL <b>determineFitnessArray</b> with chainsArray, Choromosome, Shared, SharedInfo,     CoursesInfo, nodays RETURNING fitvalue and confM ENDFOR  SORT fitvalue of each choromosome in IniPop in ascending order. DETERMINE general population from initial population SET trial to zero.  REPEAT     CALL <b>matingPool</b> with sdcoef, RETURNING Pool     IF Pool has not enough choromosomes THEN         RETURN     ELSE         CONTINUE     ENDIF     CALL <b>crosswhom</b> with Pool and nooff RETURNING MatingSet     CALL <b>crossover</b> with Pop, MatingSet and chainsArray RETURNING Offsprings      CASE muttactic OF         1: CALL <b>mutation1</b> with Pop, chainsArray, nomutants, mts RETURNING Mutants         2: CALL <b>mutation2</b> with Pop, chainsArray, nomutants, mts RETURNING Mutants         4: CALL <b>mutation4</b> with Pop, chainsArray, nomutants, mts RETURNING Mutants     ENDCASE      CALL bestof with Pop, Offsprings, Mutants, Elits RETURNING Pop  UNTIL trial &lt; nogeneration </pre>

GA genel akış kodlamasının tamir fonksiyonu modeli ile uygulanması durumunda, toplam işlem sayısının hesaplanabilmesi için, 5.1 nolu denkleme tamir fonksiyonundan kaynaklanan işlem sayıları da ilave edilmelidir. 5.2 nolu denklem tamir fonksiyonu modeli ile yapılan fazla işlem sayısını göstermektedir.



**Şekil 5.28** Tamir fonksiyonu modeli ile yeni nesil seçimi

$$gak2 = gak + tf \times (bps + (ics + ims) \times ns)$$

5.2

2007-2008 Güz Dönemi Pamukkale Üniversitesi Mühendislik Fakültesi verileri göz önüne alınarak,  $ps = 40$ ,  $eo = 0.1$ ,  $co = 0.6$ ,  $mo=0.1$ ,  $mt=1$ ,  $c=3$  olan 10. parametre seti (en iyi performans!) uygulandığında 5.1 ve 5.2 nolu denklemlerdeki değişkenlerin aldığı değerler Tablo 5.64'de gösterilmiştir. Seçilen parametre setine ve verilere göre bu değişkenlerin değerleri doğal olarak değişecektir.

Tablo 5.64'deki değişken değerlerine göre yüksek ceza katsayısı modelinde toplam 20.294.251.207 adet işlem yapılırken, bütünleşik tamir fonksiyonu modelinde ise en karmaşık durumda 94.381.970.167 adet işlem yapılmaktadır. Bu parametre seti ve verilere göre tamir fonksiyonu modelinin yüksek ceza katsayısı modeline göre en karmaşık durumda 4,65 kat daha yavaş sonuca ulaşacağı anlaşılmaktadır. İşlem sayısı karmaşıklığı hesaplanırken en kötü durum düşünülerek değerlendirildiği için bu sonuca ulaşılmıştır. Gün sayısının artması veya kısıtların azalmasıyla problem kolaylaştıkça bu fark azalmaktadır. Nitekim 10. Parametre seti ile yüksek ceza katsayısı modeli 200 nesille sonuca 1084 saniyede ulaşırken, aynı parametre seti ile tamir fonksiyonu modeli 140 nesille 2529 saniyede sonuca ulaşmıştır. Bu sonuca göre tamir fonksiyonu modeli yüksek ceza katsayısı modeline göre 3,3 kat daha yavaştır.

Tablo 5.65'de belirtildiği gibi yüksek ceza katsayısı modeli daha düşük karmaşıklık düzeyine sahip olduğu için daha hızlı sonuç üretmekte ancak daha düşük bir performans sergilemektedir.

**Tablo 5.64** 10. parametre setine göre GA değişkenlerinin aldığı değerler

$bps = 2 \times ps = 80$ ;	(Başlangıç popülasyonu sayısı popülasyon büyüklüğünün iki katı olacak şekilde sabitleştirilmiştir.)
$zu = 7$ ;	(Zincir uzunluğu değişken olduğu için, ortalama değer alınmıştır.)
$zs = 33$ ;	(Zincir sayısı 33'tür.)
$oz1 = 16$ ;	(Ortak zincirin satır sayısı 16'dır.)
$oz2 = 9$ ;	(Ortak zincirin sütun sayısı 9'dur.)
$udu = 10$ ;	(Uygunluk dizisi uzunluğu değişken olduğu için ortalama değer alınmıştır.)
$ics = ps \times co = 24$ ;	(İstenen çocuk sayısı popülasyon sayısının çaprazlama oranı ile çarpılmasıyla elde edilmektedir.)
$ku = 212$ ;	(Kromozom uzunluğu 212'dir.)
$TS = N \times 4$ ;	(Atanabilecek zaman dilimi sayısı toplam gün sayısının 4 katıdır.)
$ims = ps \times mo \times ku = 848$ ;	(İstenen mutant sayısı, popülasyon sayısının mutasyon oranı ve kromozom uzunluğu ile çarpılması ile elde edilir. )
$es = ps \times eo = 4$ ;	(Elit sayısı, popülasyon sayısının elit oranı katıdır.)
$gbdu = 216$ ;	(Geri besleme dizisi uzunluğunun yapılan deneylerde ortalama 216 olduğu görülmüştür.)
$sdu = 19$ ;	(Şüpheli dizi uzunluğunun yapılan deneylerde 19 olduğu görülmüştür.)
$ebzd = 36$ ;	(En büyük zaman dilimi 36'dır.)
$ns = 200$ ;	(Nesil sayısı ortalama 200 olarak alınmıştır.)

**Tablo 5.65** Yüksek ceza katsayısı modeli ile tamir fonksiyonu modelinin sözel olarak karşılaştırılması

Yüksek Ceza Katsayısı Modeli	Tamir Fonksiyonu Modeli
Daha Düşük Karmaşıklık Düzeyi	Daha Yüksek Karmaşıklık Düzeyi
Daha Hızlı	Daha Yavaş
Düşük Performans	Yüksek Performans

Tamir fonksiyonu modeli ile 10. Parametre seti çalıştırıldığında elde edilen zaman dilimi sonuçları zincirlere göre Tablo 5.66'da verilmiştir. İlk zincir ortak sınavların yerleştiği zaman dilimlerini gösterirken, diğer 32 zincir her bölümün sadece kendine ait olan sınavların yerleştiği zaman dilimlerini göstermektedir. Her bölümün sınavlarının yerleştiği zaman dilimleri ise EK-1.A, EK-1.B, EK-1.C, EK-1.D, EK-1.E, EK-1.F, EK-1.G ve EK-1.H'da ayrı ayrı gösterilmiştir.

**Tablo 5.66** Zincirlere göre sınavların yerleştiği zaman dilimleri

Zincirler	Bölüm - Sınıf	Sınavların Yerleştiği Zaman Dilimleri															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Ortak	3	36	4	16	5	23	30	6	9	15	18	11	26	19	14	35
2	Endüstri	1	9	19	20	31											
3	Endüstri	2	36	25	21	10	11	17									
4	Endüstri	3	32	27	22	20	8	16									
5	Endüstri	4	12	5	15	2	36	35									
6	Bilgisayar	1	29	22	21	6	26										
7	Bilgisayar	2	2	15	18	7	35	31									
8	Bilgisayar	3	10	24	12	36	8										
9	Bilgisayar	4	20	23	6	30	4	35	28	27							
10	Elektronik	1	11	32	33												
11	Elektronik	2	19	6	8	13	36	29									
12	Elektronik	3	12	35	9	34	11	10									
13	Elektronik	4	26	18	2	33	21	32	31								
14	Tekstil	1	13	15	5	9	10										
15	Tekstil	2	25	23	22	29	18	17									
16	Tekstil	3	32	36	30	34	28	11									
17	Tekstil	4	31	2	8	6	9	19									
18	Gıda	1	36	29	23	9	8										
19	Gıda	2	17	11	14	22	6										
20	Gıda	3	20	8	27	30	5	33									
21	Gıda	4	32	16	29	31	36	35	19	24							
22	Makine	1	9	20	36	8	31										
23	Makine	2	22	32	17	33	11	30									
24	Makine	3	24	36	14	7	2	13									
25	Makine	4	34	31	29	8	5	19	27	15	21						
26	İnşaat	1	24	36	8	35	31										
27	İnşaat	2	26	25	27	20	14	21	10								
28	İnşaat	3	2	13	17	18	15										
29	İnşaat	4	5	16	8	22	19	11	33	36	3	9	35	30	6		
30	Jeoloji	1	4	31	9	26											
31	Jeoloji	2	25	20	22	17	33	27									
32	Jeoloji	3	14	1	26	7	21	24	9								
33	Jeoloji	4	4	15	8	2	35	18	3	10							



## 6. SONUÇ VE ÖNERİLER

Bu çalışmada öğrenci başarısına odaklı bir sınav çizelgeleme modeli geliştirilmiştir. Bu modelin geliştirilmesi aşamasında; sınav dönemlerinde öğrencilerin, sınavları arasında hazırlanmak ve dinlenmek için en fazla süreye ihtiyaç duydukları fikri üzerine çalışmalara başlanmıştır. İhtiyaç duyulan çalışma ve dinlenme süreleri üzerinde sınavların zorluk derecelerinin doğrudan etkili olduğu anlaşılmıştır. Bu nedenle, zor olan sınavlar arasındaki ihtiyaç duyulan sürenin, kolay olan sınavlar arasındaki ihtiyaç duyulan süreye göre daha fazla olması gerektiği ortaya çıkmıştır. Bir eğitim öğretim döneminde fakültelerdeki her bölümün öğrencilerinin girmek zorunda olduğu sınavlar farklı olabileceği için, her bölüm için bu çalışma ve dinlenme zamanları ayrı ayrı sağlanmalıdır. Bir fakültede yapılan sınavların bazıları belli bölümler arasında ortak olarak aynı anda yapılmak zorunda olduğu için, ortak sınav durumu zorunlu bir kısıt olarak değerlendirilmiştir. Ayrıca bazı öğrencilerin derslerden kalma veya üst sınıflardaki derslere kayıt yaptırma durumlarının da değerlendirilebilmesi için alttan ve üstten sınav alımlarının çakışmaları önleyecek şekilde ayarlanması da zorunlu kısıt olarak öne çıkmıştır. Bu zorunlu kısıtları sağlayan ve öğrenci başarısını gözetecek şekilde sınavların sınav dönemine en uygun yayılımını gerçekleştiren matematiksel bir çizelgeleme modeli geliştirilmiştir.

Sınavların atandıkları zaman dilimlerini tüm öğrenciler için en uygun şekilde ayarlamak üzerine tasarlanan matematiksel çizelgeleme modeli, geliştirilen bir genetik algoritma modeli içerisine amaç fonksiyonu olarak gömülerek, genetik algoritmaların sınav çizelgeleme problemlerinin çözümünde etkili bir yaklaşım olduğu gösterilmiştir. Sınav çizelgeleme problemini çözmek için genetik algoritmalar ile iki farklı yöntem geliştirilmiştir. İlk yöntemde, bir bölümün almak zorunda olduğu ortak sınavlarda, ya da alttan veya üstten sınav alımlarında meydana gelen bir zaman dilimi çakışması bir yüksek ceza puanı ile cezalandırılarak önlenmeye çalışılmaktadır. Normal yayılımı sağlamak için küçük ceza katsayıları ile ilerleme sağlanırken, bir çakışma meydana geldiğinde yüksek bir ceza puanı ile bu çakışma içeren kromozomun aday çözümler arasından uzaklaştırılması amaçlanmıştır. Bu yüksek ceza puanı bire-bir çakışmaların önlenmesinin garanti altına alınması amacıyla zorunlu görülmüştür. Ancak, bire-bir

çakışma içeren bu kromozomların popülasyon içinde yer almasının genetik algoritmanın performansı açısından olumsuz olabileceği düşünülmüştür. Bu fikir üzerine, bire-bir çakışmaları her zaman engelleyen ikinci bir yöntem olan tamir fonksiyonu modeli geliştirilmiştir. Bu tamir fonksiyonu ile bire-bir çakışmalar içeren kromozomlara yüksek ceza puanı eklemek yerine, bire-bir çakışmaların mümkünse düzeltilmesi, mümkün değilse kromozomun yok edilmesi sağlanmaktadır. Tamir fonksiyonu, ilk popülasyonun yaratılması aşamasında, yeni çocukların ve mutantların elde edilmesinden sonra devreye girmektedir. Böylece kabul edilemeyen çözümler içeren kromozomlar hiçbir zaman popülasyon içerisinde yer alamamakta ve genetik algoritma her zaman kabul edilebilir aday çözümler ile çalışmaktadır.

Geliştirilen genetik algoritma modeli ile elle hazırlanan çizelgelere göre çok daha kısa sürede ve öğrencilerin başarısına odaklı bir çizelge hazırlanabilmesi mümkün kılınmıştır. Genetik algoritmaların farklı parametreler ile farklı sonuçlar üretebileceği bilindiği için, problemin yapısına göre en uygun parametreler test için seçilmiştir. Belirlenen parametrelerin tüm kombinasyonların uygulanabilmesi zaman açısından mümkün olamadığı için Taguchi'nin deneysel tasarım tablolarından yararlanılarak geliştirilen her iki yöntem farklı parametre setleri ile teste tutulmuştur. Her iki yöntemde de, popülasyon büyüklüğünün 40, elit oranının 0.1, çaprazlama oranının 0.6, mutasyon oranının 0.1, mutasyon taktiğinin 1 ve standart sapma katsayısının 3 olduğu parametre seti en iyi sonuçları üretmiştir. Elde edilen parametre sonuçlarına göre bu iki yöntem karşılaştırıldığında; yüksek ceza katsayısı modelinin tamir fonksiyonu modeline göre, daha az karmaşıklık düzeyinden dolayı daha hızlı çalışmasına rağmen daha düşük performans sergilediği görülmüştür. En iyi sonuç veren parametre seti tamir fonksiyonu modeli ile tekrar çalıştırılarak sınav çizelgesi ortaya çıkarılmıştır.

Bu çalışma ileride bir arayüz çalışması ve veritabanı modellemesi ile desteklenebilir. Arayüz ile belli sınavların belli zaman dilimlerine sabitlenmesiyle genetik algoritmanın sadece kalan sınavlar için çalışması sağlanabilir. Bu şekilde her kromozomda yeri sabitlenen sınavlar ve yeri sürekli değişen sınavlar olacaktır. Ayrıca sınavların yerleşeceği salon kısıtları da dikkate alınabilir. Salon kısıtlarının da dikkate alınmasıyla doğal olarak belli oranda öğrenci başarısından fedakârlık edilmesi gerekecektir.

## KAYNAKLAR

- Abbas, A. and Tsang, E. P. K., (2004) Software engineering aspects of constraint-based timetabling - a case study. **Information and Software Technology**, 46: 6 359-372.
- Abramson, D., (1991) Constructing school timetables using simulated annealing: sequential and parallel algorithms. **Management Science**, 37: 1 98-113.
- Al-Yakoob, S. M. and Sherali, H. D., (2006) Mathematical programming models and algorithms for a class-faculty assignment problem. **European Journal of Operational Research**, 173: 2 488-507.
- Al-Yakoob, S. M. and Sherali, H. D., (2007) A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations. **European Journal of Operational Research**, 180: 3 1028-1044.
- Alvarez-Valdes, R., Crespo, E. and Tamarit, J. M., (2002) Design and implementation of a course scheduling system using tabu search. **European Journal of Operational Research**, 137: 3 512-523.
- Arani, T., Karwan, M. and Lofti, V., (1988) A lagrangian relaxation approach to solve the second phase of the exam scheduling problem. **European Journal of Operational Research**, 34: 3 372-383.
- Asratian, A. S. and de Werra, D., (2002) A generalized class-teacher model for some timetabling problems. **European Journal of Operational Research**, 143: 3 531-542.
- Balas, E., (1969) Machine sequencing via disjunctive graphs: An implicit enumeration algorithm. **Operations Research**, 17: 6 941-957.
- Barker, J. R. and McMahon, G. B., (1985) Scheduling the general job-shop. **Management Science**, 31: 5 594-598.
- Beasley, J. E., (1990) OR-Library: Distributing test problems by electronic mail. **The Journal of the Operational Research Society**, 41: 11 1069-1072.
- Beligiannis, G. N., Moschopoulos, C. N., Kaperonis, G. P. and Likothanassis, S. D., (2008) Applying evolutionary computation to the school timetabling problem: The Greek case. **Computers & Operations Research**, 35: 4 1265-1280.
- Billionnet, A., (1999) Integer programming to schedule a hierarchical workforce with variable demands. **European Journal of Operational Research**, 114: 1 105-114.
- Boland, N., Hughes, B. D., Merlot, L. T. G. and Stuckey, P. J., (2008) New integer linear programming approaches for course timetabling. **Computers & Operations Research**, 35: 7 2209-2233.
- Brindle, A. (1981) Genetic algorithms for function optimization, Ph.D. Thesis, **University of Alberta**, Edmonton,

- Burke, E. K. and Petrovic, S., (2002) Recent research directions in automated timetabling. **European Journal of Operational Research**, 140: 2 266-280.
- Burke, E. K., McCollum, B., Meisels, A., Petrovic, S. and Qu, R., (2007) A graph-based hyper-heuristic for educational timetabling problems. **European Journal of Operational Research**, 176: 1 177-192.
- Carlier, J. and Pinson, E., (1989) An algorithm for solving the job-shop problem. **Management Science**, 35: 2 164-176.
- Carrasco, M. P. and Pato, M. V., (2004) A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem. **European Journal of Operational Research**, 153: 1 65-79.
- Carter, M. W., (1986) A survey of practical applications of examination timetabling algorithms. **Operations Research**, 34: 2 193-202.
- Dammak, A., Elloumi, A. and Kamoun, H., (2006) Classroom assignment for exam timetabling. **Advances in engineering software**, 37: 10 659-666.
- Darwin, C. (1936) *The origin of species by means of natural selection; or the preservation of favored races in the struggle for life and the descent of man and selection in relation to sex*, Editor, **Modern library**, New York,
- Daskalaki, S. and Birbas, T., (2005) Efficient solutions for a university timetabling problem through integer programming. **European Journal of Operational Research**, 160: 1 106-120.
- Daskalaki, S., Birbas, T. and Housos, E., (2004) An integer programming formulation for a case study in university timetabling. **European Journal of Operational Research**, 153: 1 117-135.
- de Werra, D., (1997) The combinatorics of timetabling. **European Journal of Operational Research**, 96: 3 504-513.
- de Werra, D., (1999) On a multiconstrained model for chromatic scheduling. **Discrete Applied Mathematics**, 94: 1-3 171-180.
- de Werra, D. and Mahadev, N. V. R., (1997) Preassignment requirements in chromatic scheduling. **Discrete Applied Mathematics**, 76: 1-3 93-101.
- Deris, S., Omatu, S. and Ohta, H., (2000) Timetable planning using the constraint-based reasoning. **Computers & Operations Research**, 27: 9 819-840.
- Dimopoulou, M. and Miliotis, P., (2001) Implementation of a university course and examination timetabling system. **European Journal of Operational Research**, 130: 1 202-213.

- Dimopoulou, M. and Miliotis, P., (2004) An automated university course timetabling system developed in a distributed environment: A case study. **European Journal of Operational Research**, 153: 1 136-147.
- Dowland, K. A., (1990) A timetabling problem in which clashes are inevitable. **The Journal of the Operational Research Society**, 41: 10 907-918.
- Drexl, A. and Salewski, F., (1997) Distribution requirements and compactness constraints in school timetabling. **European Journal of Operational Research**, 102: 1 193-214.
- Eglese, R. W. and Rand, G. K., (1987) Conference seminar timetabling. **The Journal of the Operational Research Society**, 38: 7 591-598.
- Ergül, A. (1995) A genetic algorithm for university examination scheduling, Yüksek lisans tezi, **Ortadoğu Teknik Üniversitesi**, Ankara, 150
- Fang, H.-L. (1994) Genetic algorithms in timetabling and scheduling, **University of Edinburgh**, Edinburgh, 243
- Foulds, L. R. and Johnson, D. G., (2000) SlotManager: a microcomputer-based decision support system for university timetabling. **Decision Support Systems**, 27: 4 367-381.
- Fred, G., (1986) Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, 13: 5 533-549.
- Garey, M. R. and Johnson, D. S. (1990) Computers and intractability; a guide to the theory of np-completeness, Editor, **W. H. Freeman**, 338
- Glover, F. and Laguna, M., (1993). Tabu search", Modern heuristic techniques for combinatorial problems,**John Wiley Sons, Inc.**, 70-150.
- Goldberg, D. E. (1989) Genetic algorithms in search, optimization and machine learning, Editor, **Addison-Wesley Longman Publishing Co., Inc.**, 372
- Head, C. and Shaban, S., (2007) A heuristic approach to simultaneous course/student timetabling. **Computers & Operations Research**, 34: 4 919-933.
- Held, M. and Karp, R. M., (1962) A dynamic programming approach to sequencing problems. **Journal of the Society for Industrial and Applied Mathematics**, 10: 1 196-210.
- Hertz, A., (1991) Tabu search for large scale timetabling problems. **European Journal of Operational Research**, 54: 1 39-47.
- Holland, J. H. (1992) Adaptation in natural and artificial systems, Editor, **MIT Press**, 211
- Johnson, D., (1990) Timetabling university examinations. **The Journal of the Operational Research Society**, 41: 1 39-47.

- Kirkpatrick, S., C. D. Gelatt, Jr. and Vecchi, M. P., (1988). Optimization by simulated annealing", *Neurocomputing: Foundations of Research*, **MIT Press**, 551-567.
- Kulak, O., Yilmaz, I. and Günther, H.-O., (2007) A GA-based solution approach for balancing printed circuit board assembly lines. **OR Spectrum**,
- McMahon, G. and Florian, M., (1975) On scheduling with ready times and due dates to minimize maximum lateness. **Operations Research**, 23: 3 475-482.
- MirHassani, S. A., (2006) Improving paper spread in examination timetables using integer programming. **Applied Mathematics and Computation**, 179: 2 702-706.
- MirHassani, S. A., (2006) A computational approach to enhancing course timetabling with integer programming. **Applied Mathematics and Computation**, 175: 1 814-822.
- Naji Azimi, Z., (2005) Hybrid heuristics for examination timetabling problem. **Applied Mathematics and Computation**, 163: 2 705-733.
- Petrovic, S., Yang, Y. and Dror, M., (2007) Case-based selection of initialisation heuristics for metaheuristic examination timetabling. **Expert Systems with Applications**, 33: 3 772-785.
- Santiago-Mozos, R., Salcedo-Sanz, S., DePrado-Cumplido, M. and Bousoño-Calzon, C., (2005) A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a Spanish university. **Computers & Operations Research**, 32: 7 1761-1776.
- Smith, K. A., Abramson, D. and Duke, D., (2003) Hopfield neural networks for timetabling: formulations, methods, and comparative results. **Computers & Industrial Engineering**, 44: 2 283-305.
- Taillard, E., (1990) Some efficient heuristic methods for the flow shop sequencing problem. **European Journal of Operational Research**, 47: 1 65-74.
- Taillard, E., (1993) Benchmarks for basic scheduling problems. **European Journal of Operational Research**, 64: 2 278-285.
- Tripathy, A., (1980) A lagrangean relaxation approach to course timetabling. **The Journal of the Operational Research Society**, 31: 7 599-603.
- Tripathy, A., (1984) School timetabling - A case in large binary integer linear programming. **Management Science**, 30: 12 1473-1489.
- Valouxis, C. and Housos, E., (2003) Constraint programming approach for school timetabling. **Computers & Operations Research**, 30: 10 1555-1572.
- Weitz, R. R. and Lakshminarayanan, S., (1997) An empirical comparison of heuristic and graph theoretic methods for creating maximally diverse groups, VLSI design, and exam scheduling. **Omega**, 25: 4 473-482.

Widmer, M. and Hertz, A., (1989) A new heuristic method for the flow shop sequencing problem. **European Journal of Operational Research**, 41: 2 186-193.

**EKLER**



**EK-1.A. Endüstri Mühendisliği Bölümü Sınav Bilgi Formu ve Sınav Çizelgesi**

<b>Endüstri1</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TÜRK DİLİ - I	11101	0	0	4	36	ozd
ATATÜRK İLKELERİ VE İNKILÂP TARİHİ - I	12101	3	0	4	3	ozd
YABANCI DİL - I (İNGİLİZCE)	13101	0	0	1	16	ozd
TEMEL BİLGİ TEKNOLOJİSİ KULLANIMI	20101	1	0	3	9	
GENEL FİZİK - I	151111	0	0	6	19	
GENEL KİMYA	153115	1	0	7	26	ozd
GENEL MATEMATİK - I	155113	1	0	10	6	ozd
ENDÜSTRİ MÜHENDİSLİĞİNE GİRİŞ	235101	1	0	6	20	
TEKNİK RESİM	235103	1	0	6	31	
BEDEN EĞİTİMİ - I	16101	0	0	1	5	ozd
<b>Endüstri2</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TEMEL BİLGİSAYAR BİLİMLERİ	20201	1	0	7	36	
DİFERANSİYEL DENKLEMLER	155219	3	0	8	25	
OLASILIK	235201	3	0	10	14	Elektronik3
DOĞRUSAL CEBİR	235203	3	0	10	21	
MÜHENDİSLİKTE MALZEME BİLGİSİ	235205	3	0	8	10	
MALİYET MUHASEBESİ	235209	3	1	6	11	
ENDÜSTRİYEL PSİKOLOJİ	235233	3	1	5	17	
<b>Endüstri3</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TESİS TASARIMI VE PLANLAMASI	235301	1	0	8	32	
YÖNEYLEM ARAŞTIRMASI	235303	1	0	10	27	
İŞLETME YÖNETİMİ	235305	1	0	6	22	
SİMÜLASYON	235307	1	0	9	20	
MESLEKİ İNGİLİZCE	235309	0	0	5	8	
ARAŞTIRMA YÖNTEMLERİ	235331	1	0	6	16	
<b>Endüstri4</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
KALİTE PLANLAMA VE KONTROLÜ	235401	0	0	9	12	
BAKIM PLANLAMASI VE GÜVENİRLİK	235403	0	0	6	5	
YATIRIM DEĞERLENDİRMESİ	235405	0	1	8	15	
KALİTE YÖNETİM SİSTEMLERİ	235431	0	0	5	2	
BİLGİSAYARLI BÜTÜNLEŞİK ÜRETİM	235437	0	0	10	36	
İNSAN KAYNAKLARI YÖNETİMİ	235439	0	0	7	35	

**EK-1.B. Bilgisayar Mühendisliği Bölümü Sınav Bilgi Formu ve Sınav Çizelgesi**

<b>Bilgisayar1</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TÜRK DİLİ - I	11101	1	0	1	36	ozd
YABANCI DİL - I (İNGİLİZCE)	13101	1	0	1	16	ozd
GENEL FİZİK - I	151111	1	0	8	29	
GENEL MATEMATİK - I	155113	1	0	9	9	ozd
BİLGİSAYAR PROGRAMLAMA - I	233101	1	0	10	22	
BİLGİSAYAR PROGRAMLAMA LABORATUARI - I	233103	1	0	10	21	
BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ - I	233105	1	0	10	6	
BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ SEMİNERİ	233107	1	0	7	26	
BEDEN EĞİTİMİ - I	16101	0	0	1	5	ozd
<b>Bilgisayar2</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
ATATÜRK İLKELERİ VE İNKILÂP TARİHİ - I	12201	1	1	4	3	ozd
VERİ YAPILARI	233201	3	0	10	2	
DOĞRUSAL CEBİR	233203	3	0	7	15	
SAYISAL SİSTEMLER	233205	1	0	8	18	
SAYISAL SİSTEMLER LABORATUARI	233207	1	0	7	7	
ELEKTRİK DEVRELERİ	233209	1	0	8	35	
DİFERANSİYEL DENKLEMLER	233211	3	0	9	31	
<b>Bilgisayar3</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
VERİ TABANI YÖNETİMİ VE MODELLEMESİ	233301	1	0	10	10	
BİLGİSAYAR MİMARİSİ	233303	1	0	10	24	
İŞLETİM SİSTEMLERİ	233305	1	0	10	12	
WEB TABANLI TEKNOLOJİLER	233337	1	0	9	36	
FELSEFE	233343	1	0	5	8	
<b>Bilgisayar4</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
BİLGİSAYAR MÜHENDİSLİĞİ PROJESİ	233401	0	0	10	20	
MESLEK SEMİNERİ - I	233403	0	0	8	23	
VERY İLETİŞİMİ VE BİLGİSAYAR AĞLARI	233405	0	0	10	6	
HABERLEŞME TEKNOLOJİLERİ	233439	0	0	9	30	
EVİRİMSEL HESAPLAMA	233443	0	0	9	4	
YAPAY ZEKA	233445	0	0	10	35	
GÖRÜNTÜ İŞLEME	233447	0	0	10	28	
BİLGİSAYAR HUKUKU	233453	0	1	7	27	

### EK-1.C. Elektrik-Elektronik Mühendisliği Bölümü Sınav Bilgi Formu ve Sınav Çizelgesi

<b>Elektronik1</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TÜRK DİLİ - I	11101	1	0	1	4	ozd
YABANCI DİL - I (İNGİLZCE)	13101	3	0	1	16	ozd
GENEL FİZİK - I	151111	1	0	8	11	
GENEL KİMYA	153115	1	0	10	26	ozd
GENEL MATEMATİK - I	155113	1	0	10	9	ozd
BİLGİSAYAR - I	227109	1	0	7	32	
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİNE GİRİŞ	227111	1	0	5	33	
BEDEN EĞİTİMİ - I	16101	3	0	1	30	ozd
<b>Elektronik2</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>		<i>Ortak?</i>
ATATÜRK İLKELERİ VE İNKILÂP TARİHİ - I	12201	3	0	3	3	ozd
DİFERANSİYEL DENKLEMLER	155225	3	0	10	19	
MÜHENDİSLİK MATEMATİĞİ	227211	3	0	10	6	
DEVRE ANALİZİ - I	227213	3	0	10	8	
SAYISAL SİSTEMLER - I	227215	3	0	10	13	
ELEKTRONİK - II	227217	3	0	7	36	
DEVRE LABORATUVARI	227219	3	0	7	29	
<b>Elektronik3</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>		<i>Ortak?</i>
ELEKTROMANYETİK DALGALAR	227323	1	0	8	12	
SİNYALLER VE SİSTEMLER	227325	1	0	8	35	
ELEKTROMEKANİK ENERJİ DÖNÜŞÜM SİSTEMLERİ	227327	1	0	7	9	
OLASILIK TEORİSİ	227329	1	0	7	14	Endustri2
ELEKTRONİK LABORATUVARI	227333	1	0	7	34	
DEVRE SENTEZİ	227343	1	0	5	11	
ELEKTRİK TESİSLER	227351	1	0	6	10	
<b>Elektronik4</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>		<i>Ortak?</i>
ELEKTRİK-ELEKTRONİK MÜH.ÖZEL KONULAR - I	227701	0	0	8	26	
MİKRO-ELEKTRONİK DEVRELER	227707	0	0	8	18	
YÜKSEK YOĞUNLUKLU ENTEGRE(CMOS VLSI)TAS.	227721	0	0	9	2	
MİKRODALGA DEVRELER GİRİŞ	227725	0	0	9	33	
ELEKTRİK TESİSLERİ PROJE	227751	0	0	7	21	
AYDINLATMA TEKNİĞİ	227757	0	0	7	32	
MÜHENDİSLİKTE YÖN. VE KAYNAKLARIN DEĞ.	227789	0	0	6	31	

**EK-1.D. Tekstil Mühendisliği Bölümü Sınav Bilgi Formu ve Sınav Çizelgesi**

<b>Tekstil1</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TÜRK DİLİ - I	11101	1	0	1	36	ozd
ATATÜRK İLKELERİ VE İNKILÂP TARİHİ - I	12101	0	0	1	3	ozd
YABANCI DİL - I (İNGİLİZCE)	13101	1	0	1	16	ozd
TEMEL BİLGİ TEKNOLOJİSİ KULLANIMI	20101	1	0	1	13	
GENEL FİZİK - I	151111	1	0	8	11	ozd
GENEL KİMYA	153115	1	0	8	26	ozd
GENEL MATEMATİK - I	155113	1	0	8	15	
TEKSTİL MÜHENDİSLİĞİNE GİRİŞ	231101	1	0	5	5	
TEKNİK RESİM	231109	1	0	1	9	
DOĞAL LİFLER	231151	1	0	5	10	
RESİM - I	17101	0	0	1	30	ozd
<b>Tekstil2</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TEMEL BİLGİSAYAR BİLİMLERİ	20201	1	1	5	25	
DİFERANSİYEL DENKLEMLER	155219	3	1	9	23	ozd
TEKSTİL KİMYASI	231201	2	1	5	22	
MUKAVEMET	231205	3	1	9	29	
AKIŞKANLAR MEKANİĞİ	231207	3	1	9	18	
DOKUMA HAZIRLIK	231209	3	1	6	17	
<b>Tekstil3</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
MAKİNE ELEMANLARI	231301	1	0	9	32	
MEKANİZMA TEKNİĞİ	231303	1	1	8	35	Makine3
TERMODİNAMİK	231305	1	1	9	36	
İMAL USULLERİ	231309	1	1	7	30	
PAMUK İPLİK VE TEKN. - II	231353	1	1	6	34	
ÖN TERBİYE	231355	1	1	6	28	
DOKUMA TEKNİĞİ VE ANALİZİ	231357	1	1	6	11	
<b>Tekstil4</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
OTOMATİK KONTROL	231401	0	1	7	31	
DOKUMA MAKİNALARI	231403	0	0	7	2	
BİTİM İŞLEMLERİ	231451	0	0	7	8	
TEKSTİL MALZEME MUAYENESİ	231461	0	1	7	6	
TEKNİK İNGİLİZCE - I	231439	0	0	6	9	
İPLİKÇİLİKTE YENİ TEK. VE TEKSTÜRE P.	231453	0	1	7	19	

**EK-1.E. Gıda Mühendisliği Bölümü Sınav Bilgi Formu ve Sınav Çizelgesi**

<b>Gıda1</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TÜRK DİLİ - I	11101	3	0	1	4	ozd
ATATÜRK İLKELERİ VE İNKILÂP TARİHİ - I	12101	3	0	1	3	ozd
YABANCI DİL - I (İNGİLİZCE)	13101	3	0	3	16	ozd
TEMEL BİLGİ TEKNOLOJİSİ KULLANIMI	20101	3	0	2	36	
GENEL FİZİK - I	151111	1	0	5	29	
GENEL KİMYA	153115	1	0	3	19	ozd
ORGANİK KİMYA	153123	1	0	3	23	
GENEL MATEMATİK - I	155113	1	0	8	15	ozd
TEKNİK RESİM	229109	1	0	2	9	
GIDA MÜHENDİSLİĞİNE GİRİŞ	229115	1	0	9	8	
BEDEN EĞİTİMİ - I	16101	0	0	1	30	ozd
<b>Gıda2</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
DİFERANSİYEL DENKLEMLER	155219	3	0	8	18	ozd
TERMODİNAMİK	229207	3	0	6	17	
MÜHENDİSLİK MEKANIĞI	229215	3	0	5	11	
GIDA MİKROBİYOLOJİSİ	229217	3	0	3	14	
GIDA BİYOKİMYASI	229219	3	0	5	22	
REAKSİYON KİNETİĞİ	229221	3	0	5	6	
<b>Gıda3</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
GIDA MÜHENDİSLİĞİ TEMEL İŞLEMLERİ - I	229315	1	0	3	20	
ET TEKNOLOJİSİ	229317	1	0	7	8	
SÜT TEKNOLOJİSİ	229319	1	0	6	27	
YAĞ TEKNOLOJİSİ	229321	1	1	5	30	
PROSES KONTROL	229323	1	0	3	5	
GIDA MÜHENDİSLİĞİ UYGULAMALARI - I	229325	1	0	2	33	
<b>Gıda4</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
İŞLETME SANİTASYONU	229409	0	0	2	32	
PROJE TEKNİĞİ	229415	0	0	5	16	
GIDA KATKI MADDELERİ	229417	0	1	4	29	
BESLENME	229419	0	1	4	31	
SOĞUK TEKNİĞİ	229437	0	0	4	36	
TEKNİK İNGİLİZCE	229449	0	0	4	35	
ASİDİK FERMENTE GIDALAR TEKNOLOJİSİ	229459	0	1	3	19	
GIDA MÜHENDİSLİĞİ VE MESLEKİ ETİK	229469	0	1	1	24	

## EK-1.F. Makine Mühendisliği Bölümü Sınav Bilgi Formu ve Sınav Çizelgesi

<b>Makine1</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TÜRK DİLİ - I	11101	3	0	2	4	ozd
ATATÜRK İLKELERİ VE İNKILÂP TARİHİ - I	12101	3	0	1	3	ozd
YABANCI DİL - I (İNGİLİZCE)	13101	3	0	4	16	ozd
TEMEL BİLGİ TEKNOLOJİSİ KULLANIMI	20101	1	0	6	9	
GENEL FİZİK - I	151111	1	0	8	20	
GENEL KİMYA	153115	1	0	7	36	
GENEL MATEMATİK - I	155113	1	0	10	15	ozd
TEKNİK RESİM - I	223151	1	0	9	8	
MAKİNE MÜHENDİSLİĞİNE GİRİŞ	223161	1	0	2	31	
BEDEN EĞİTİMİ - I	16101	3	0	1	23	ozd
<b>Makine2</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TEMEL BİLGİSAYAR BİLİMLERİ	20201	3	1	7	22	
DİFERANSİYEL DENKLEMLER	155219	3	1	9	18	ozd
ELEKTROTEKNİK	223241	3	1	7	32	
MALZEME - I	223251	3	1	8	17	
TERMODİNAMİK - I	223253	3	1	9	33	
MUKAVEMET-I	223255	3	1	8	11	
LABORATUVAR & ATÖLYE	223257	3	1	2	30	
<b>Makine3</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
ELEKTRİK MAKİNALARI	223341	1	1	7	24	
MAKİNA ELEMANLARI - I	223351	1	1	9	36	
TAKIM TEZGAHLARI	223353	1	1	6	14	
ÖLÇME TEKNİĞİ	223355	1	1	6	7	
AKIŞKANLAR MEKANİĞİ - I	223357	1	1	9	2	
OTOMATİK KONTROL	223359	1	1	8	13	
MEKANİZMA TEKNİĞİ	223361	1	1	8	35	Tekstil3
<b>Makine4</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
BUHAR KAZANLARI	223451	0	1	6	34	
HİDROLİK MAKİNALAR	223453	0	1	8	31	
İMAL USULLERİ - II	223455	0	1	4	29	
MAKİNA PROJE - I	223457	0	1	6	8	
MAKİNA LABORATUVARI - I	223459	0	1	3	5	
KALIPÇILIK TEKNİĞİ	223443	0	1	5	19	
HİDROLİK VE PNÖMATİK DEVRELER	223463	0	1	5	27	
İSITMA HAVALANDIRMA	223471	0	1	5	15	
BİLGİSAYAR DESTEKLİ ÇİZİM	223481	0	1	5	21	

**EK-1.G. İnşaat Mühendisliği Bölümü Sınav Bilgi Formu ve Sınav Çizelgesi**

<b>İnşaat1</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TÜRK DİLİ - I	11101	3	0	2	4	ozd
ATATÜRK İLKELER VE İNKILÂP TARİHİ - I	12101	3	0	2	3	ozd
YABANCI DİL - I (İNGİLİZCE)	13101	3	0	4	16	ozd
TEMEL BİLGİ TEKNOLOJİSİ KULLANIMI	20101	3	0	3	24	
GENEL FİZİK - I	151111	1	0	6	11	ozd
GENEL KİMYA	153115	1	0	6	19	ozd
GENEL MATEMATİK - I	155113	1	0	7	36	
İNŞ. MÜH. GİRİŞ	221107	1	0	5	8	
TEKNİK RESİM	221109	1	0	7	35	
GENEL JEOLJİ	221111	1	0	7	31	
BEDEN EĞİTİMİ - I	16101	1	0	1	23	ozd
<b>İnşaat2</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TEMEL BİLGİSAYAR BİLİMLERİ	20201	3	0	8	26	
DİFERANSİYEL DENKLEMLER	155219	3	0	8	25	
YAPI TEKNİK RESİM	221207	3	0	7	27	
MÜH. MEKANİĞİ - II	221209	3	0	8	20	
MUKAVEMET - I	221211	3	0	10	14	
İSTATİSTİK	221213	3	0	8	21	
MÜHENDİSLİKTE SAYISAL YÖNTEMLER	221215	3	0	8	10	
<b>İnşaat3</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
ZEMİN MEKANİĞİ - I	221311	1	0	7	2	
HİDROLOJİ	221313	1	0	8	13	
HİDROLİK - II	221315	1	0	7	17	
YAPI STATİĞİ - I	221317	1	0	9	18	
ULAŞTIRMA - I	221319	1	0	8	15	
<b>İnşaat4</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Y. Zaman Dilimi</i>	<i>Ortak?</i>
TEMEL İNŞAAT	221401	0	0	7	5	
SU YAPILARI - II	221403	0	0	7	16	
BETONARME - II	221405	0	0	8	8	
ÇELİK YAPILAR - II	221407	0	0	8	22	
ULAŞTIRMA - II	221409	0	0	7	19	
YAPISAL HASARLAR	221417	0	0	6	11	
YÜKSEK YAPILAR	221421	0	0	6	33	
YAPI ELEMANLARININ DAVRANIŞI	221423	0	0	6	36	
ZEMİN İYİLEŞTİRME YÖNTEMLERİ	221429	0	0	6	3	
TRAFİK MÜHENDİSLİĞİ	221431	0	0	6	9	
KENTSEL ALTYAPI TASARIMI - I	221445	0	0	6	35	
DOLGU BARAJLAR	221457	0	0	6	30	
İŞ MAKİNALARI	221465	0	0	6	6	

**EK-1.H. Jeoloji Mühendisliği Bölümü Sınav Bilgi Formu ve Sınav Çizelgesi**

<b>Jeoloji1</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>	<i>Yerleştiği Zaman Dilimi</i>	<i>Ortak?</i>
TÜRK DİLİ - I	11101	3	0	2	36	ozd
ATATÜRK İLKELERİ VE İNKILÂP TARİHİ - I	12101	3	0	2	3	ozd
YABANCI DİL - I (İNGİLİZCE)	13101	3	0	4	16	ozd
TEMEL BİLGİ TEKNOLOJİSİ KULLANIMI	20101	1	0	3	4	
GENEL FİZİK - I	151111	1	0	7	11	ozd
GENEL KİMYA	153115	1	0	6	19	ozd
GENEL MATEMATİK - I	155113	1	0	8	6	ozd
GENEL JEOLJİ	225111	1	0	7	31	
JEOLJİ MÜH. GİRİŞ	225113	1	0	5	9	
TEKNİK RESİM	225121	1	0	8	26	
BEDEN EĞİTİMİ - I	16101	0	0	1	5	ozd
<b>Jeoloji2</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>		<i>Ortak?</i>
TEMEL BİLGİSAYAR BİLİMLERİ	20201	1	0	3	25	
DİFERANSİYEL DENKLEMLER	155219	3	0	8	20	
YAPISAL JEOLJİ - I	225213	3	0	8	22	
ÖLÇME BİLGİSİ	225221	3	0	5	17	
MÜHENDİSLİK MEKANİĞİ - II	225251	3	0	7	33	
MUKAVEMET-I	225253	3	0	7	27	
<b>Jeoloji3</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>		<i>Ortak?</i>
DOĞAL MALZEME BİLGİSİ	225303	1	0	6	14	
ZEMİN MEKANİĞİ - I	225311	1	0	6	1	
JEOLJİK HARİTA BİLGİSİ	225319	1	0	8	26	
STRATİGRAFİ - SEDİMANTOLOJİ	225321	1	0	6	7	
METAMORFİK KAYAÇ PETROGRAFİSİ	225323	1	0	7	21	
MÜHENDİSLİK JEOLJİSİ - I	225343	1	0	6	24	
JEOLJİ UYGULAMALARI - I	225345	1	0	6	9	
<b>Jeoloji4</b>						
<i>Ders</i>	<i>Optik Kod</i>	<i>Alt?</i>	<i>Ust?</i>	<i>Zorluk</i>		<i>Ortak?</i>
MADEN YATAKLARI	225431	0	0	7	4	
KAYA MEKANİĞİ - I	225439	0	0	8	15	
HİDROJEOLJİ-I	225445	0	0	6	8	
YAKITLAR JEOLJİSİ	225451	0	0	6	2	
ÇEVRE JEOLJİSİ	225467	0	0	4	35	
MALZEME JEOLJİSİ	225469	0	0	4	18	
DEPREM JEOLJİSİ	225479	0	0	4	3	
MERMER JEOLJİSİ	225485	0	0	4	10	



## ÖZGEÇMİŞ

Can Berk Kalaycı 13 Haziran 1982 tarihinde Denizli’de doğdu. İlkokul eğitimini Denizli Emsan İlköğretim okulunda tamamladı. Ortaokul ve lise eğitimini Denizli Anadolu Lisesi’nde tamamladıktan sonra 2001 yılında Sakarya Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünde lisans eğitimine başladı. 2005 yılında bu bölümden bilgisayar mühendisi olarak mezun olduktan sonra Pamukkale Üniversitesi Fen Bilimleri Enstitüsü’nde Bilgisayar Mühendisliği anabilim dalında yüksek lisans eğitimine başladı. Ders aşamasını tamamladıktan sonra, tez aşamasında aynı enstitünün Endüstri Mühendisliği anabilim dalına yatay geçiş yaptı ve yüksek lisans eğitimini bu anabilimdalında tamamladı. 2005 yılından beri Pamukkale Üniversitesi Endüstri Mühendisliği Bölümü’nde araştırma görevlisi olarak çalışmaktadır.