T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KARAR DESTEK SİSTEMLERİNİN BİR DURUM ÇALIŞMASINA UYGULANMASI

Nazan DEMİRCİ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

YRD. DOÇ. DR. Rembiye KANDEMİR

EDİRNE, 2012

T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KARAR DESTEK SİSTEMLERİNİN BİR DURUM ÇALIŞMASINA UYGULANMASI

Nazan DEMİRCİ

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Rembiye KANDEMİR

EDİRNE, 2012

T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KARAR DESTEK SİSTEMLERİNİN

BİR DURUM ÇALIŞMASINA UYGULANMASI

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

Nazan DEMİRCİ

Bu tez ...../...../..... tarihinde Aşağıdaki Jüri Tarafından Kabul Edilmiştir.

(İmza)	(İmza)	(İmza)
.....	.....	.....
Yrd. Doç. Dr. Rembiye KANDEMİR	Yrd. Doç. Dr. Altan MESUT	Yrd. Doç. Dr. Hilmi KUŞÇU

Yüksek Lisans Tezi

Karar Destek Sistemlerinin Bir Durum Çalışmasına Uygulanması

T.Ü. Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Sınav çizelgeleme problemi, akademik kurumların her eğitim dönemi içinde karşılaştıkları idari sorunlardan biridir. Fakültelerde her bir bölümün sınavlarının uygulanma sürecinin aynı zaman diliminde yer alması, kısıtlı olan derslik kaynaklarının ve zaman aralıklarının sınavlara atanması problemini daha karmaşık hale getirmektedir. Çalışmada, personel iş yükünü azaltmak ve öğrenciler ile öğretim elemanları açısından en elverişli sınav programını elde etmek için bu problemin çözüme ulaştırılması amaçlanmıştır. Karar Destek Sistemleri'nin, dağınık veri topluluğunu analiz ederek karar vericinin istenilen bilgiye zamanında ulaşmasını ve daha etkin karar vermesini sağlama özelliğinden dolayı çözüm için bu sistemden yararlanılmış ve model olarak sezgisel ve iteratif bir yöntem olan Genetik Algoritma kullanılmıştır. Genetik algoritmaların farklı seçim kriterlerine ve çaprazlama yöntemlerine göre değerlendirilen sistem, zaman açısından %70 kazanç sağlamakta ve birden fazla alternatif çözüm içinden kullanıcının en iyisini, en uygununu seçme olanağı tanımaktadır.

Yıl : 2012

Sayfa Sayısı : 105

Anahtar Kelimeler : Karar Destek Sistemleri, Genetik Algoritmalar, Üniversite Çizelgeleme Problemleri, Sınav Çizelgelemesi, PMX, OX

MS (MASTER) Thesis

Applying Decision Support Systems To A Case Study

Trakya University Institute of Natural Sciences

Computer Engineering Department

ABSTRACT

Exam scheduling problem is one of the administrative problems encountered by academic instituents in each training period. Departments having exams in the same time zone in faculties makes the problem of assigning limited classrooms and time intervals to exams more complicated. In this study, to solve this problem to reduce the workload of academic staff and obtain most suitable exam schedule for students and academic staff is aimed. Due to the feature of providing the access to the needed (intended) data in time and helping to desicion-maker to make more effective decisions by analyzing discrete data collections, Decision Support System is utilizied for solution and as a model, Genetic Algorithm, a heuristic and iterative method, is used. The system which is evaluated by genetic algorithms' different selection criterias and crossover methods, provides a gain of %70 in terms of time and allows user to choose the best solution in alternatives.

Year : 2012

Number of Pages **:** 105

Keywords **:** Decision Support Systems, Genetic Algorithms, University Timetabling Problems, Exam Timetabling, PMX, OX

TEŞEKKÜR

Tezin oluşturulması sırasında bilgi birikimi ve tecrübesiyle yol gösteren danışman hocam Yrd. Doç. Dr. Rembiye KANDEMİR'e, tez jürisinde yer aldıkları ve verdikleri öneriler için Yrd. Doç. Dr. Altan MESUT ve Yrd. Doç. Dr. Hilmi KUŞÇU'ya teşekkür ederim.

Hayatım boyunca attığım her adımla gurur duyan, gerekli özveriden kaçınmadan her daim desteklerini hissettiren ve bunu sabırla, hoşgörüyle devam ettiren, verdikleri emeği hiçbir zaman ödeyemeyeceğim annem, babam ve abime minnettarlığımı iletmeyi bir borç bilirim.

Tezin uygulama kısmında yazılım bilgisiyle yardımcı olan, tez süresince moralimi her zaman üst seviyede tutmaya çalışan ve şikâyet etmeden bana her daim katlanmayı başaran Yük. Bilg. Müh. Emir ÖZTÜRK'e, gergin anlarımda daima hoşgörüyle ve sabırla yaklaşan Arş. Gör. Turgut DOĞAN ile Arş. Gör. Mustafa ARDA'ya ve Edirne'deki en büyük şansımın kendisiyle tanışmam olduğuna inandığım Arş. Gör. Şeyma DERECİ'ye teşekkür ederim.

Tezin tamamlanmasını en az benim kadar isteyen, bir yıl boyunca gerekli sabrı gösteren, bu süre zarfında manevi desteğini, güvenini sürekli olarak hissettiren ve ne zaman gereksinim duysam hep yanımda olan Mimar Emre Murat Zıraman'a teşekkür ederim.

Son olarak, verdiği burs ile yüksek lisans öğrenimim boyunca maddi destek sağlayan TÜBİTAK'a ve proje kapsamında sağladığı donanım desteği için TÜBAP'a teşekkürlerimi sunarım.

İçindekiler

ÖZET i

ABSTRACT ii

TEŞEKKÜR iii

SİMGELER DİZİNİ vii

TABLO LİSTESİ x

ŞEKİL LİSTESİ xi

1. GİRİŞ 1

2. KARAR DESTEK SİSTEMLERİ 3

2.1. Karar Destek Sistemlerinin Yetenekleri ve Özellikleri 8

2.2. Karar Destek Sistemlerinin Temel Bileşenleri 10

2.3. Karar Destek Sistemlerine İhtiyaç Duyulmasının Nedenleri 12

2.4. Karar Destek Sistemlerinin Uygulama Alanları ve Kullanılan Algoritmalar 12

3. GENETİK ALGORİTMALAR 16

3.1. Genetik Algoritmalarda Kullanılan Operatörler 17

3.1.1. Kromozom, Popülasyon ve Gen 17

3.1.2. Kromozomun Kodlanması 17

3.1.2.1. İkili Kodlama 17

3.1.2.2. Permütasyon Kodlama 18

3.1.2.3. Değer kodlama 18

3.1.2.4. Ağaç kodlama 18

3.1.3. Uygunluk Fonksiyonu 19

3.1.4. Seçim 19

3.1.4.1. Rulet Tekeri Seçimi 19

3.1.4.2. Elitizm 20

3.1.4.3. Rank Seçimi 20
3.1.5. Çaprazlama 21
3.1.5.1. Tek Nokta Çaprazlama 21
3.1.5.2. İki Nokta Çaprazlama 22
3.1.5.3. Uniform Çaprazlama 22
3.1.5.4. Aritmetik Çaprazlama 23
3.1.5.5. Kısmi-Haritalanmış Çaprazlama (Partially Mapped Crossover-PMX) 24
3.1.5.6. Sıralı Çaprazlama (Order Crossover- OX1) 25
3.1.5.7. Pozisyon Tabanlı Çaprazlama (Position Based Crossover-POS) 26
3.1.5.8. Sıra Tabanlı Çaprazlama (Order Based Crossover- OX2) 27
3.1.5.9. Döngüsel Çaprazlama (Cycle Crossover- CX) 27
3.1.5.10. Sezgisel Çaprazlama (Heuristic Crossover) 29
3.1.5.11. Sıralı Eşlemeli Çaprazlama (Sorted Match Crossover) 30
3.1.5.12. Maksimal Koruyucu Çaprazlama (Maximal Preservative Crossover-MPX) 31
3.1.6. Mutasyon 32
3.1.6.1. Yerini Alma (Displacement Mutation- DM) 33
3.1.6.2. Yer Değişimi (Exchange Mutation- EM) 33
3.1.6.3. Araya Yerleştirme (Insertion Mutation- ISM) 34
3.1.6.4. Basit Ters Çevirme (Simple Inversion Mutation- SIM) 34
3.1.6.5. Ters Çevirme (Inversion Mutation- IVM) 35
3.1.6.6. Karıştırma (Scramble Mutation- SM) 35
3.1.6.7. Sezgisel (Heuristic Mutation- HM) 35
3.2. Genetik Algoritmaların Parametreleri 36
3.2.1. Çaprazlama Oranı 36

3.2.2. Mutasyon Oranı 37
3.2.3. Popülasyon Büyüklüğü 37
3.3. Genetik Algoritmalarda Kısıtların Değerlendirilmesi 37
3.3.1. Reddeden Yöntem 37
3.3.2. Düzelten Yöntem 38
3.3.3. Probleme Özgü Operatör Geliştirme 38
3.3.4. Cezalandırıcı Yöntem 38
4. ÇİZELGELEME PROBLEMLERİ 39
5. KARAR DESTEK SİSTEMLERİNİN DURUM ÇALIŞMASINA UYGULANMASI (GA_SYP) 49
5.1. Kartezyen Çarpım Menüsü 52
5.2. Tablo Oluşturma Menüsü 53
5.3. Veri Tabanı İşlemleri 55
5.4. Problem Çözümü Menüsü 58
5.5. Derslik Atama Menüsü 60
5.6. Geliştirilen Uygulamada Genetik Algoritmaların İşleme Süreci 63
5.7. GA Parametrelerine Göre Model Sonuçlarının Değerlendirilmesi 74
6. SONUÇLAR 93
KAYNAKLAR 96
ÖZGEÇMİŞ 105

SİMGELER DİZİNİ

C(x):	Ceza fonksiyonu
Ç1:	İlk çocuk
Ç2:	İkinci çocuk
çs:	Çakışma sayısı
D:	Derslik bilgilerinin sonlu kümesi
$dK_{\text{ceza değeri}}$:	Ders kodu çakışması ceza değeri
ds:	Derslik sayısı
E1:	İlk ebeveyn
E2:	İkinci ebeveyn
f(x):	Uygunluk fonksiyonu
GA:	Gen ağırlığı değeri
G_h:	Gen havuzu büyüklüğü
G_i:	Gen değerleri kümesi
gs:	Gün-saat kartezyen çarpım adedi
$gS_{\text{ceza değeri}}$:	Gün-saat çakışması ceza değeri
K:	Kısıt bilgilerinin sonlu kümesi
m:	Esnek kısıt sayısı
n:	Zorunlu kısıt sayısı
P(x):	Kromozomun rulet tekerinde seçilme olasılığı
pb:	Popülasyon büyüklüğü
P_c:	Çaprazlama olasılığı

P_m: Mutasyon olasılığı

S: Sınavı yapılacak ders kodlarının sonlu kümesi

$sC_{\text{ceza değeri:}}$ Şube çakışması ceza değeri

ss: Dönemdeki sınav sayısı

T: Zaman aralıklarının sonlu kümesi

X: Kromozom dizisi

Kısaltmalar

AHP: Analitik Hiyerarşi Prosesi

CM: Cut Mutation

CX: Cycle Crossover

DM: Displacement Mutation

DSS: Decision Support Systems

EM: Exchange Mutation

GA: Genetik Algoritmalar

GA_SYP: Genetik Algoritma ile Sınav Yerleştirme Programı

GSP: Gezgin Satıcı Problemi

HM: Heuristic Mutation

IS: Bilgi Sistemi

ISM: Insertion Mutation

IVM: Inversion Mutation

KDS: Karar Destek Sistemleri

MPX: Maximal Preservative Crossover

ODEP:	Okul Ders Programı
ODYP:	Online Ders Yerleştirme Programı
OLAP:	Online Analitik Bilgi İşleme
OX1:	Order Crossover
OX2:	Order Based Crossover
PMX:	Partially Mapped Crossover
POS:	Position Based Crossover
SIM:	Simple Inversion Mutation
SM:	Scramble Mutation
VTYS:	Veri Tabanı Yönetim Sistemi
YSA:	Yapay Sinir Ağları
Z.K.Ü.:	Zonguldak Karaelmas Üniversitesi

TABLO LİSTESİ

Tablo 3.1 Çaprazlama operatörleri 23
Tablo 3.2 Mutasyon operatörleri 32
Tablo 6.1 Seçim kriterlerine uygun çaprazlamaların belirlenmesi 94
Tablo 6.2 Çaprazlama operatörlerine uygun seçimlerin belirlenmesi 94

ŞEKİL LİSTESİ

Şekil 2.1	Yönetim Seviyelerinde Karar Tipleri	4
Şekil 2.2	İdeal bir Karar Destek Sisteminin özellikleri ve yetenekleri	9
Şekil 2.3	KDS'nin Bileşenleri	10
Şekil 3.1	Değer kodlama için kromozom örneği	18
Şekil 3.2	Ağaç kodlama için kromozom örneği	18
Şekil 3.3	Rulet tekeri	20
Şekil 3.4	Rank seçimi	20
Şekil 3.5	Tek nokta çaprazlama	21
Şekil 3.6	İki nokta çaprazlama	22
Şekil 3.7	Uniform çaprazlama	22
Şekil 3.8	Aritmetik çaprazlama	23
Şekil 3.9	PMX'te rastgele konumların belirlenmesi	24
Şekil 3.10	PMX'te gen değişimi	24
Şekil 3.11	PMX'te gen değerleri arasındaki ilişki	25
Şekil 3.12	PMX sonrası oluşan yeni bireyler	25
Şekil 3.13	OX1'in gerçekleşme şekli	25
Şekil 3.14	OX1 sonrası yeni birey	26
Şekil 3.15	POS gerçekleşme şekli	26
Şekil 3.16	POS sonrası yeni birey	26
Şekil 3.17	OX2 gerçekleşme şekli	27
Şekil 3.18	OX2 sonrası yeni birey	27
Şekil 3.19	CX için döngüyü sağlayan gen değerlerinin tespiti	28
Şekil 3.20	CX'te yeni bireyin oluşumu	28
Şekil 3.21	CX 'te ebeveynlerden gen kopyalanması	28
Şekil 3.22	CX sonrası oluşan yeni birey	28
Şekil 3.23	CX sonrası oluşan diğer birey	29
Şekil 3.24	Sezgisel çaprazlamada permütasyon oluşturan gen değerlerinin belirlenmesi	29
Şekil 3.25	Sezgisel çaprazlama sonrası oluşan bireyler	29
Şekil 3.26	Sıralı eşlemeli çaprazlamada kromozom parçasının belirlenmesi	30
Şekil 3.27	Sıralı eşlemeli çaprazlamada maliyet karşılaştırması	30
Şekil 3.28	Sıralı eşlemeli çaprazlama sonrası yeni birey	30
Şekil 3.29	MPX'in gerçekleşme şekli	31
Şekil 3.30	MPX sonrası oluşan diğer birey	31
Şekil 3.31	DM mutasyonu	33
Şekil 3.32	EM mutasyonu	33
Şekil 3.33	ISM mutasyonu	34
Şekil 3.34	SIM mutasyonu	34
Şekil 3.35	IVM mutasyonu	35
Şekil 3.36	SM mutasyonu	35

Şekil 3.37	HM mutasyonu	36
Şekil 5.1	Ana ekran	52
Şekil 5.2	Kartezyen Çarpım Aşaması	52
Şekil 5.3	Tablo Oluşturma	53
Şekil 5.4	Veri tabanı tabloları ve ilişkileri	55
Şekil 5.5	Veri tabanı işlemleri ders bilgileri arayüzü	56
Şekil 5.6	Veri tabanı işlemleri dersi alan öğrenci sayıları bilgisi arayüzü	57
Şekil 5.7	Problem çözümü	58
Şekil 5.8	Sınavlar için uygun dersliklerin belirlenmesi	61
Şekil 5.9	Problemin akış diyagramı	63
Şekil 5.10	Gen değerinin içerdiği verinin gösterimi	64
Şekil 5.11	Tek bir kromozomun ifadesi	65
Şekil 5.12	Kromozom oluşturma ve başlangıç popülasyonunun elde edilmesi	65
Şekil 5.13	Kromozomların uygunluk değerlerinin hesaplanması	66
Şekil 5.14	Uyarlama fonksiyonu sözde kodu	68
Şekil 5.15	PMX/PMX-s operatörlerinin aşamaları ve Mutasyon işlemi	70
Şekil 5.16	PMX-s/OX-s için kromozom seçimi ve çaprazlanma sonrası popülasyona eklenmesi	71
Şekil 5.17	OX/OX-s operatörlerinin aşamaları ve Mutasyon işlemi	71
Şekil 5.18	Mutasyon sözde kodu	72
Şekil 5.19	Tamir fonksiyonu	72
Şekil 5.20	Tamir fonksiyonu sözde kodu	73
Şekil 5.21	Tek gen tamiri	73
Şekil 5.22	Tek gen tamirinin sözde kodu	74
Şekil 5.23	Uygun çaprazlama, mutasyon ve elitizm oranlarının belirlenmesi	75
Şekil 5.24	Rulet tekeri seçiminde çaprazlama operatörlerinin değerlendirilmesi	78
Şekil 5.25	Rulet tekeri seçiminde çaprazlama operatörlerinin her iterasyona göre değerlendirilmesi	78
Şekil 5.26	Rank seçiminde çaprazlama operatörlerinin değerlendirilmesi	79
Şekil 5.27	PMX çaprazlama operatörünü kullanan Elitizm seçiminde, çaprazlama operatörlerinin değerlendirilmesi	80
Şekil 5.28	OX çaprazlama operatörünü kullanan Elitizm seçiminde, çaprazlama operatörlerinin değerlendirilmesi	80
Şekil 5.29	PMX-s çaprazlama operatörünü kullanan Elitizm seçiminde, çaprazlama operatörlerinin değerlendirilmesi	81
Şekil 5.30	OX-s çaprazlama operatörünü kullanan Elitizm seçiminde çaprazlama operatörlerinin değerlendirilmesi	81
Şekil 5.31	PMX çaprazlama operatöründe seçim kriterlerinin değerlendirilmesi	82
Şekil 5.32	OX çaprazlama operatöründe seçim kriterlerinin değerlendirilmesi	83
Şekil 5.33	PMX-s çaprazlama operatöründe seçim kriterlerinin değerlendirilmesi	84
Şekil 5.34	OX-s çaprazlama operatöründe seçim kriterlerinin değerlendirilmesi	84
Şekil 5.35	İterasyon sayılarının değerlendirilmesi	86
Şekil 5.36	Popülasyon sayılarının değerlendirilmesi	89
Şekil 5.37	Şube çakışması önlenerek hazırlanan sınav programı	90

Şekil 5.38 Toplantı saatlerinde sınav olmaması durumu 91
Şekil 5.39 Belirli bir zaman aralığına sınav yerleştirilmesi durumu 92

BÖLÜM 1

GİRİŞ

Kısıtlı optimizasyon problemlerinden biri olan çizelgeleme probleminin çözümü ve elde edilen alternatiflerin değerlendirilerek en uygun çözüme karar verilmesi karmaşık ve zaman alıcı bir süreçtir. Karmaşık problemlerin çözümü için, çok büyük veri yığınları içerisinden anlamlı bilgi üreten ve bu bilgi sayesinde karar vericinin karar verme sürecini hızlandıran, birden fazla seçenek sunarak en olası durumu seçmesini sağlayan etkileşimli bir bilgi sistemi teknolojisi olan Karar Destek Sistemleri (KDS) bu tür problemlerin çözümünde etkin bir rol almakta ve başarıya ulaşmaktadır.

Yıllardır araştırmacıların sürekli olarak ilgisini çeken çizelgeleme problemleri, NP hard (Non-deterministic polynomial-time hard) olarak bilinen problemlerin giderilmesinde çözüm uzayını en verimli şekilde tarayarak, uygun bir çözüm elde etmeye çalışan kombinatoryal optimizasyon algoritmalarını test etme olanağı sağlamaktadır. Komşu çözümleri araştırarak adım adım iyileştirme yapan ve böylelikle optimum sonuca yakınsamayı amaçlayan bu algoritmalar, daha çok tavlama benzetimi, yapay sinir ağları, genetik algoritmalar, tabu arama, çizge renklendirme ve karınca kolonisi algoritmaları olarak karşımıza çıkmaktadır.

Akademik kurumlarda sıklıkla karşılaşılan sınav çizelgeleme problemi, giderek artan bölüm sayısı ve buna bağlı olarak öğrenci sayısının artması, fakülte bazında yapılan sınavların adedini artırmakta ve derslik kaynaklarını yetersiz kılmaktadır. Aynı şekilde her yıl değişen öğrenci sayısı, derslerle ilgili yapılabilecek güncellemeler, derslik sayısının kısıtlı olması, gözetmen sayısının azalma ihtimali ve öğrenciler ile öğretim elemanlarının istekleri dikkate alındığında sınav programının hazırlanması önemli bir idari soruna dönüşmektedir. Genellikle bir önceki senenin modifiye edilmesi ile elde edilen sınav programları günümüzde hataya sebep olması ve fazla zaman alması nedeni ile tercih edilmemektedir. Bu iş yükünden kurtulmak ve olası çözümler arasından öğrenci ve öğretim elemanı memnuniyetini sağlayacak en verimli sonucun

tespit edilmesi için hazırlanan çalışmada, Karar Destek Sistemleri'nin bir uygulaması olan sezgisel algoritmalardan Genetik Algoritmalar yöntemi kullanılmıştır.

Tezin ikinci bölümünde, "karar verme" durumunun öneminden, KDS'nin tarihsel gelişiminden bahsedilmiş, temel bileşenleri hakkında bilgi verilmiştir. Sistemin genel özellikleri ile yetenekleri belirtilerek Karar Destek Sistemleri'ne ihtiyaç duyulmasının nedenleri sunulmuş ve uygulama alanlarına değinilmiştir.

Üçüncü bölümünde, model olarak kullanılan Genetik Algoritmaların ortaya çıkış süreci anlatılmış ve algoritmadaki tüm işlemlerin adımları anlatılarak, parametrelerinin doğru belirlenmesinin önemi vurgulanmıştır.

Dördüncü bölümde, eğitim kurumlarındaki çizelgeleme problemlerinde (ders/sınav) karşılaşılan zorunlu ve esnek kısıtlar anlatılmış ve konu ile ilgili yapılan literatür çalışmasına yer verilmiştir.

Beşinci bölümde ise Genetik Algoritma'nın seçim kriterlerinden rulet tekeri, rank, elitizm seçimi ve çaprazlama operatörlerinden kısmi haritalanmış çaprazlama ve sıralı çaprazlama yöntemleri kullanılarak hazırlanan sınav çizelgeleme probleminin çözümü için yapılan tüm genetik işlemler anlatılmış ve kullanılan tüm parametrelerin sonuçları alınarak değerlendirilmesi de altıncı bölümde yapılmıştır.

BÖLÜM 2

KARAR DESTEK SİSTEMLERİ

İnsanoğlu yaşadığı sürece sürekli olarak "karar verme" durumundadır. Peki nedir "karar verme"?. Karar verme, istenilen amaçlar için, bu amaçlara ulaştıracak yollar, araçlar ve alternatifler arasında seçim ve tercih yapmakla ilgili zihinsel, bedensel ve duygusal süreçlerin toplamıdır [Eren, 2001]. Amaçlara ulaşmak için ortaya çıkan sorunları bertaraf etme işidir. Kısacası karar verme, sorun çözme ve çevrenin sunduğu fırsatları tanımlama sürecidir [Tekin, 2009]. Önem derecelerine göre değişen bu kararlar, gün içerisinde ne giyeceğini seçme olabileceği gibi, bir fabrikada üretim sürecinde kısıtlı kaynakların nasıl optimum şekilde kullanılacağını belirlemek de olabilir. Kararın kalitesi ise, erişilebilen bilginin yeterliliğine, bilginin kalitesine, seçim sayısına, karar anında uygulanan modelin uygunluğuna bağlıdır. Önemli olan daha fazla bilgi değil, daha uygun bilgi gereksinimidir [Sauter, 1997].

Farklı yönetim seviyelerince alınan farklı karar tipleri bulunmaktadır. Bunlar üç ana grupta ele alındığında, Şekil 2.1'de sınıflandırıldığı gibi yapısal (structured), yarı-yapısal (semi-structured) ve yapısal olmayan (unstructured) kararlardır [TBD Kamu-BİB, 2010].



Şekil 2.1. Yönetim Seviyelerinde Karar Tipleri

Yapısal kararlar, her gün karşılaşılan problemler için sık alınan rutin kararlardır. Belirlilik seviyesi yüksektir ve karar verici sübjektif bir esnekliğe sahip değildir. Bilgisayarlar tarafından otomatik olarak verilebilen kararlardır. Üretim ve satın alma kararları, yenileme kararları, uzun dönem planlama ve yatırım kararları, stok takip sistemi, alacak ve borçların takibi örnek verilebilir.

Yapısal olmayan kararlarda problemler çok iyi tanımlanmamıştır ve doğru cevabı bulmanın mutlak bir yolu bulunmamaktadır. Dolayısıyla hiçbir kriter veya kuralın iyi bir çözümü garanti etmediği bu tür kararlar, karar vericinin yargı ve sezgisini kullandığı ve sübjektif bir esnekliğe sahip olduğu kararlardır. Finansal kararlar, yeni bir pazara açılma veya yeni bir ürün geliştirme gibi geleceğe yönelik kararlardır.

Yarı-yapısal kararlar ise, kısmen tanımlı ve belirli olan, kısmen de yöneticinin yargı ve sezgisine dayanan kararlardır. Geliştirilecek ürün için tesis ve işgücünün planlanması gibi kararlardır [TBD Kamu-BİB, 2010].

Verilecek her bir karar istenilen bir sonuç doğurabileceği gibi karar vericiyi istenmeyen durumlarla da karşı karşıya bırakabilir. Bunun sebebi günümüzde çevresel, sosyal, ekonomik ve diğer amaçların birbiri ile sıkça çelişmesidir (Demir - Gümüşoğlu, 1998). Bu nedenle karar verme süreci doğru bir şekilde gerçekleştirilmelidir.

Geçmiş dönem yöneticilerinin karar verme sürecinde karşılaştıkları en büyük problemlerden biri bilgi toplamak iken, günümüz yöneticilerinin karar verme sürecinde karşılaştıkları en büyük problem, toplanan verilerden anlamlı sonuçlar çıkartılması

olarak değişmiştir. Bu aşamada toplanan veri miktarı da önem arz etmektedir. Veri miktarı arttıkça karar vericilerin bu verileri anlamaları ve yorumlamaları da zorlaşmaktadır. Bu problemi çözmek için karar vericiler ellerindeki veri yığınlarından gerekli olan bilgileri çıkartarak; veriler arasındaki gizli desenleri, örüntüleri gösterecek sistemlere ihtiyaç duymaktadırlar [Arslan ve Yılmaz, 2010].

Son yıllarda karar vericilerin karar verme aşamasında Karar Destek Sistemleri'nin kullanımı oldukça yaygınlaşmıştır. Geliştirilen ve karar vericilerin kullanımına sunulan Karar Destek Sistemleri, organizasyonların/kurumların içinde veya dışında bulunan verilere son kullanıcıların rahatlıkla ulaşabilmelerini sağlamaktadır. Bu sayede ihtiyaç duyulan bilgilere hızlı ve zamanında erişim, kurumlarda kararların zamanında alınmasına yardım ederek verimliliği ve alınan kararların kalitesini yükseltmektedir [Arslan ve Yılmaz, 2010].

Karar Destek Sistemi, karar vermeyi kolaylaştırmak ve daha etkili hale getirmek için tasarlanan model, benzetim ve uygulamaları içinde bulunduran sistemlerdir. Karar desteği, yöneticilerin yönetsel problemleri modellerle çözümleme çabaları sonucu ortaya çıkmıştır ve ilk olarak J.D. Little'ın çalışmasıyla ortaya konmuştur [Marakas, 1999]. Terim olarak KDS'nin kullanıldığı ilk çalışma ise Gorry ve Scott Morton'un "Yönetim Karar Sistemleri" isimli yazısıdır [Kuruüzüm, 1998].

Yaygın bir şekilde kullanılmasına rağmen, Karar Destek Sistemleri'nin kesin bir tanımı yoktur. Bazı araştırmacılar KDS'yi felsefe bilimi olarak değerlendirip, "teknolojik araçlar ile insan yargıları arasındaki ilişkiyi tamamlayan yoldur" şeklinde tanımlarken, Little (1970) Karar Destek Sistemlerini "yöneticilerin karar vermelerine yardımcı olabilmek için yargılarını ve verileri işleyebilecek modele dayalı prosedürler seti" olarak tanımlamaktadır [Little, 1970]. Keen ve Morton 1978'de Karar Destek Sistemlerini, verilen kararların kalitesini arttırmak için bilgisayarların yetenekleri ile bireylerin entelektüel kaynakları arasında ilişki kuran, yarı-yapısal ve yapısal olmayan problemlerin çözümü için karar vericiye yardımcı olan bilgisayar destekli sistemler olarak belirtmiştir [Keen ve Morton, 1978]. Klein ve Hirscheim, KDS'yi kuruluşlar için yararlı, ve doğası gereği müdahale süreci polemik olmayan sistemler olarak görmektedir [Klein vd., 1985]. Ginzberg ve Stohr, KDS'nin tanımına açıklık getirmek için, "KDS hedefini gerçekleştirmek için neler yapar?", "Hangi bileşenler gereklidir?", "Nasıl bir

örüntü kullanımı uygundur?", "Geliştirme süreci gerekliliği nasıl açıklanır?" sorularına cevap aramıştır [Ginzberg vd., 1982].

1970'lerin başında Karar Destek Sistemleri, sadece karar vermede yardımcı olan bilgisayar tabanlı sistemler olarak karşımıza çıkmaktadır. İlk çıkış noktası ise, interaktif bilgisayar teknolojisinin uygulanması ile gerçekleşmiştir. Yönetimsel işlerde daha iyi karar almak için bilgi sistemlerinin kullanılmaya başlanması ile KDS, karar verici olarak bilişsel bir kavram haline gelmiştir. 1970'lerin sonuna doğru büyüyen KDS hareketi/akımı, kötü yapılandırılmış problemleri modellemeyi ve karar vericiye yardım etmek için veri tabanlarını oluşturan interaktif bilgisayar tabanlı sistemleri ön plana çıkarmıştır. Finansal planlamalar için hazır paketlerin ve hızlı uygulama geliştirmek için toolların geliştirilmesi bunlara birer örnektir. 1980'li yılların başında, yönetimsel ve organizasyonel faaliyetlerin etkinliğini artırmak için çeşitli teknolojilerin yeni sistemlerde istihdam edilmesiyle KDS akımı yaygınlaşmıştır. KDS etiketi altında, kullanıcı dostu yazılımlar üretilmiş, bilgi merkezleri ve prototipleme gibi kavramlar da bu etiketin altında yer almış, daha sonra Yöneylem araştırması ve psikoloji gibi disiplinler de bu akıma katılmıştır. 1986'da akıllı iş istasyonlarının ortaya çıkması ile KDS'nin yeni bir yönü keşfedilmiştir. Uzman sistemler ve doküman tabanlı sistemler bu dönemin yeni teknolojileridir. Elam ve arkadaşları KDS'nin gerekliliğini, "önemli kararların yaratıcılığını artırmak için bilgisayar tabanlı teknolojilerin kullanılması" şeklinde vurgulamışlardır [Elam vd., 1986].

Sprague KDS'yi, son kullanıcı, üretici ve araç geliştirici (tool-smith) perspektifinden incelemiştir. Bu ayrımlara uygun olarak, genel yazılım araçları ve KDS arasında köprü görevi gören "DSS generator" kavramını ortaya koymuştur. "DSS generator", kullanıcı ara yüzü ve KDS için model geliştirmek veri tabanı şeması oluşturmak ve kurallar belirlemek için kullanılan yazılım paketidir. Sprague, KDS'nin ana bileşenlerini, veri tabanı, model ve kullanıcı ara yüzü yazılımı olarak belirlemiş ve KDS modelini denklem modeli tipinde öngörmüştür. Ekonometrik modeller, denklem modellerine örnek teşkil etmektedir. Diğer bir kategori (model tipi) ise lineer dinamik ve stokastik özelliklere dayanan optimizasyon modelleridir [Sprague, 1980].

Klein ve Manteau, Bergquist ve McLean çalışmalarında, o yıllarda geliştirilen "DSS generator" ların, veri tabanı ve interaktif olanaklarla (imkân, donanım) beraber,

veri, model ve metin işleme özelliklerini içeren denklem modelleri üzerine odaklandığını belirtmişlerdir [Klein vd., 1983, Bergquist vd., 1983].

Daha kavramsal bir bakış açısı Bonczek ve arkadaşları tarafından ortaya atılmıştır. Bonczek ve arkadaşları, dil sistemi, bilgi sistemi ve problem çözme sistemi gibi bileşenlere değinmiştir. Dil sistemi, KDS tarafından karar vericiye sunulan tüm dilsel özelliklerin toplamıdır. Bilgi sistemi, KDS'nin iskeletidir ve problem uzayı bilgisini içerir. Problem çözme sistemi ise dil sistemindeki problemler ile bilgi sistemindeki bilgiler arasındaki köprü mekanizmasıdır [Bonczek vd., 1981]. Bu çerçevede bakıldığında KDS, yapay zeka ve uzman sistemler ile ilişkilendirilmiştir.

Chang, Melamud ve Seabrok 1983'te, "insanların özel uzmanlık alanlarını kapsayan, düzenli bilgi içeren, olgusal ve sezgisel özelliklere sahip ve kullanıcı için çıkarımlarda bulunabilen bilgisayar sistemi" şeklinde bir uzman sistem tanımlamışlardır. Problem çözme sisteminin bir türü olan çıkarım sistemine ve bilgi sisteminin bir türü olan bilgi tabanı üzerine bakıldığında, uzman sistemler Bonczek çerçevesine uyacaktır [Chang vd., 1983].

Karar destek için bilginin ifade edilmesinde aynı doğrultuda çalışan birçok araştırmacı vardır. Lee (1983) ve Stamper (1985) çalışmalarında, karar alma süreçlerini geliştirmek için bilginin gösterimini epistemoloji ile ilişkilendirmişlerdir. Epistemoloji, bilgi felsefesi olarak da adlandırılan, bilginin doğası, kapsamı, kaynağı, sınırları ve kavramsal bileşenleri ile ilgilenen felsefe dalıdır. Temel kavramları doğruluk, gerçeklik ve temellendirmedir [Lee, 1983; Stamper, 1985].

KDS'nin tasarım süreci ilgi çeken konular arasındadır. Sprague ve Carlson, "KDS'nin üç temel alanlarının (veri tabanı, model, kullanıcı ara yüzü) her birinin gereksinimlerini tanımlamak için sistemler analizini tasarlamışlardır. Bu yaklaşım kullanıcı odaklı dört birime dayandırılmıştır. Bu birimler, temsilcilikler, operasyon, bellek yardımı ve denetim mekanizmaları olarak adlandırılırlar [Sprague vd., 1982].

Long (1989), Karar Destek Sistemleri'ni, "karar verme sürecinde yönetime destek vermek için hedeflenen bilginin üretilmesi ve sunulması için kullanıcı etkileşimli yazılım ve donanım vasıtalarının bütünleşik kümesinden oluşan interaktif bilgi sistemleri" şeklinde betimlemiştir [Long, 1989].

Klein ve Methlie (1990) çalışmalarında Karar Destek Sistemleri'nin teorisi ile ilgili olarak detaylı bilgiler vermektedir. 1990'lı yılların başlarında özellikle online analitik bilgi işleme (OLAP) ve veri ambarları bu sistemlerin yaygınlaşmasında önemli rol oynamıştır. OLAP konusunda detaylı bilgiler Pendse'nin (1999) çalışmasında verilmektedir [Yesevi, 2012].

Turban ve Aronson (1998) ise Karar Destek Sistemleri'ni, yöneticilerin özellikle yapısal olmayan problemlere çözüm üretmek için etkin karar vermelerini destekleyen interaktif, esnek ve adaptif bilgisayar destekli bilişim sistemleri olarak tanımlamaktadırlar [Yesevi, 2012].

Laskey (2003) KDS'yi, bilgiyi organize ederek bir veya daha fazla karar vericiye karar vermelerinde yardım eden ve destek veren bilgisayar destekli sistemler olarak tanımlamaktadır. Karar vericinin özellikle teknolojik gelişmeler ve çok fonksiyonlu işler ile uğraşmasından dolayı çok sayıda bilgiyi değerlendirerek karar vermek zorunda kaldığını belirterek karar vermede bilgisayarları kullanmanın önemine değinmektedir [Yesevi, 2012].

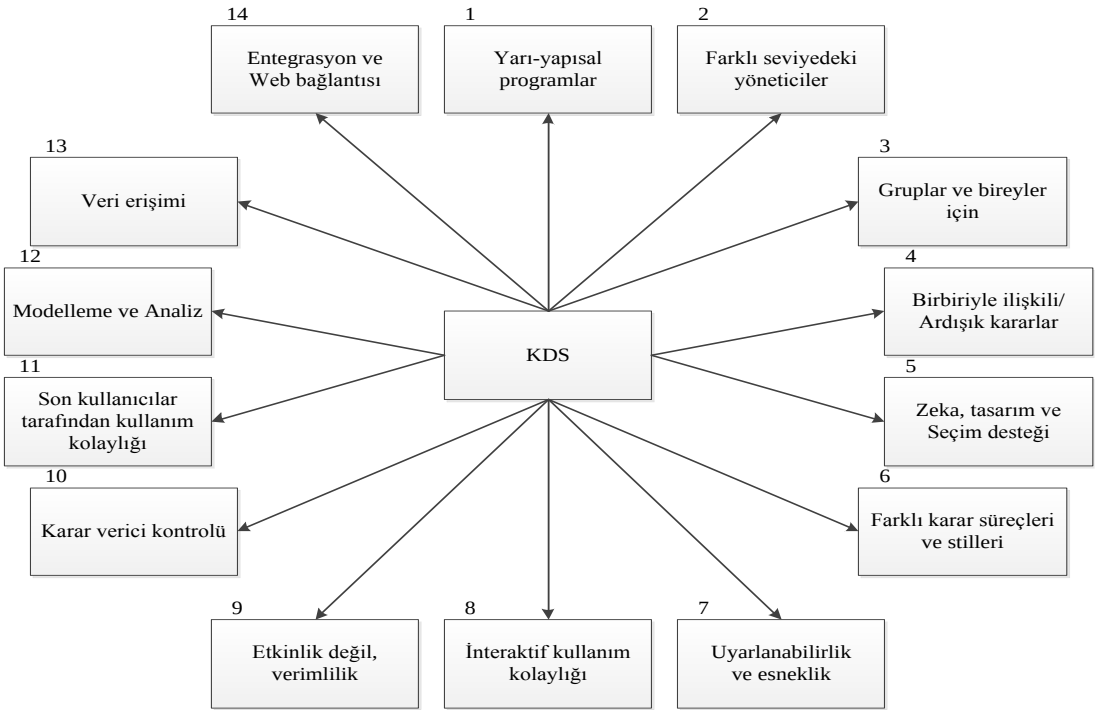
Literatürde, KDS'nin içeriği ve tanımlamalarıyla ilgili yukarıda verilen bilgilere benzer şekilde birçok kaynak bulunabilir. Tüm bu bilgileri özetleyecek şekilde KDS'nin tanımını yapacak olursak, KDS tek başına karar vermeyen, karar vericiye muhakeme yeteneğini de kullandırarak doğru ve etkin karar vermesini sağlayan, yarı-yapısal ve yapısal olmayan kararlarda destek olan, karar vericinin ihtiyaç duyduğu bilgiyi doğru zamanda minimum maliyetle sunan etkileşimli sistemlerdir.

2.1. Karar Destek Sistemlerinin Yetenekleri ve Özellikleri

Karar Destek Sistemleri'nin standart özellikleri ve yeteneklerinin neler olduğuna dair bir belirlilik olmamasından dolayı "KDS nedir?" sorusunun cevabı üzerine net bir bilgi yoktur. Opening Vignette'de yapılan tanımlamalara göre KDS'nin temel özellikleri belirlenmiştir.

Temel KDS özellikleri ve yetenekleri Şekil 2.2'de görüldüğü üzere aşağıdaki gibidir.

1. KDS, bilgisayarın bilgisi ile insan yargısını bir araya getirerek, yarı yapılandırılmış ve yapılandırılmamış durumlarda karar vericiler için destek sağlar. Bu tür sorunlar diğer bilgisayar sistemleri veya standart sayısal yöntemler ve araçlar tarafından çözülemez.
2. Destek, üst düzey yöneticilerden, ilgili bölüm yöneticilerine kadar çeşitli yönetim seviyeleri için sağlanır.
3. Bireysel ve grup tabanlı karar verme desteği sağlar. Daha az yapılandırılmış sorunlarda, genelde farklı departmanlardan ve organizasyonel seviyelerden veya farklı organizasyonlardan bireyler içeren gruplara destek verir.
4. KDS, birbiriyle ilişkili ve/veya ardışık kararlara destek sağlar. Kararlar bir defa, birkaç kez veya art arda verilebilir.
5. KDS, karar verme sürecinin tüm aşamalarına (zeka, tasarım, seçim ve uygulanması) destek sağlar.
6. KDS, farklı karar alma süreçlerini ve stillerini destekler.

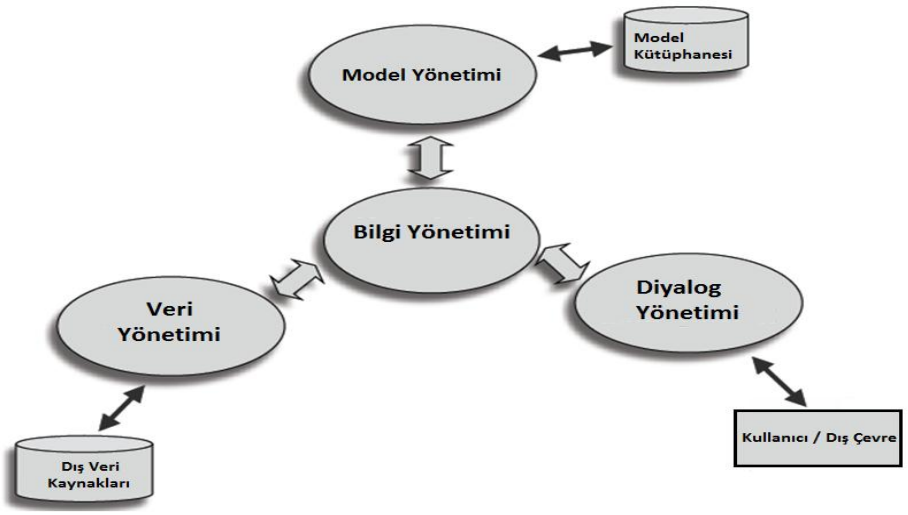


Şekil 2.2. İdeal bir Karar Destek Sisteminin özellikleri ve yetenekleri [Turban ve Aronson, 2001]

7. KDS, zaman içinde uyarlanabilir. Karar verici, reaktif, hızla değişen koşullara uyum sağlayabilen ve bu değişiklikleri karşılamak için KDS'yi uyarlayabilen kişidir.
8. Karar Destek Sistemi'nin esnek olması, kullanıcılara ekleme, silme, birleştirme, değiştirme ve temel unsurları yeniden düzenleme imkânı tanır.
9. KDS, karar almanın etkisinden (karar alma maliyeti) ziyade verimliliğini (doğruluk, güncellik, kalite) arttırmaya çalışır.
10. Karar verici, karar verme sürecinin tüm adımlarında tam kontrole sahiptir. KDS, karar vericinin yerini almayı değil, karar vericiyi desteklemeyi amaçlar.
11. Son kullanıcılar, kendilerine göre basit sistemler oluşturup, modifiye edebilir. Fakat büyük sistemler, bilgi sistemi (IS) uzmanları tarafından inşa edilir.
12. KDS, karar verme durumlarının analizi için modellerden yararlanır. Modelleme yeteneği, farklı yapılandırmalar altında, farklı stratejilerle tecrübe sağlar [Turban ve Aronson, 2001].

2.2. Karar Destek Sistemlerinin Temel Bileşenleri

Tipik bir KDS, Veri Yönetimi, Model Yönetimi, Bilgi Yönetimi ve Diyalog/Arayüzü Yönetimi olmak üzere dört temel bileşenden oluşmaktadır. Bu temel bileşenler ve etkileşimleri şematik olarak Şekil 2.3'te verilmiştir [TBD Kamu-BİB, 2010].



Şekil 2.3. KDS'nin Bileşenleri

Veri Yönetimi, karar vericinin belli bir kararı verebilmesi için, ilgili ve gerekli verinin getirilmesi, saklanması ve organize edilmesiyle ilgili değişik faaliyetlerin yerine getirildiği bir KDS bileşenidir. Veri yönetimi gerekli veriyi barındıran ve Veri tabanı Yönetim Sistemi (VTYS) de denen yazılım tarafından yönetilen veri tabanlarını kapsar. KDS veri tabanı, bir kişisel bilgisayara yerleştirilecek kadar küçük bir veri tabanı ya da veri ambarı şeklinde olabilir [Turban, 1993].

Model yönetimi, KDS için analitik yetenek sağlayan değişik sayısal modellerin getirilmesi, saklanması ve organize edilmesiyle ilgili faaliyetlerin yerine getirildiği bir KDS bileşenidir. Bu bileşenin iki önemli alt sistemi; model tabanı ve model tabanı yönetim sistemidir [Gökçen, 2007]. Model tabanı veri tabanının modelleme bölümüdür. Modellerin çalıştırılması, veri tabanı ile ilişkilendirilmeleri ve çözüm amacı ile entegrasyonlarının sağlanması ile sorumludur. Model tabanı yönetim sistemi ise bütün bunları bir arada çalıştıran, modelleri güncelleyen, işleten ve çözümler üreten gerekli verileri veri tabanlarından alan sistemdir. KDS veri tabanı KDS verilerini tutarken, model tabanı, KDS'nin değişik analizler yapması için kullandığı çeşitli istatistiksel, finansal, matematiksel ve diğer kantitatif modelleri içermektedir [TBD Kamu-BİB, 2010].

Bilgi Yönetimi, bilginin elde edilmesi, gösteriminin sağlanması, muhafaza edilmesi ve bilginin kullanılması ve güncellenmesi ile ilgilenmektedir.

Diyalog Yönetimi/ Kullanıcı Arayüzü, kullanıcı ile KDS'nin iletişimini sağlayan KDS bileşenidir. Kullanıcının sistemle doğrudan temasta bulunduğu tek bileşendir. Arayüzün tasarımı ve hayata geçirilmesi, KDS'nin fonksiyonelliği için önemlidir. Karar verici, verilere, modellere ve KDS'nin işlevsel bileşenlerine kolaylıkla ulaşabilmeli ve kullanabilmelidir. Kullanıcının KDS iletişiminin kolaylığının, KDS'nin kullanım başarısıyla eş anlamlı olduğu unutulmamalıdır [Gökçen, 2007].

2.3. Karar Destek Sistemlerine İhtiyaç Duyulmasının Nedenleri

Hızla gelişen teknoloji ve artan bilgi fazlalığı ile alternatifler çoğalmış dolayısıyla da karar alma durumları daha kompleks bir hal almıştır. Buna insanların, sorunları iyi ifade edememesi, yeniliklerin etkilerine yetişememesi, öncelikleri belirlemede sorun yaşaması, kendilerine olan aşırı özgüven gibi yaptıkları hatalar dahil olunca Karar Destek Sistemleri'ne olan ihtiyaç kaçınılmaz olmuştur. Ayrıca günümüzde yazılımların daha anlaşılır hale gelmesi ve ucuzlaması ile bilgisayarların erişilebilirliği ve kullanılabilirliği artmış, bilgisayar destekli sistemler ön plana çıkmıştır.

Karar Destek Sistemleri, bilgisayarların güçlü özelliklerini insanların güçlü yanlarıyla kullanarak karar verme sürecinin etkinliğini arttırmayı amaçlamaktadır. Bu üstünlüklere bakarsak;

İnsanlar karar verirken, tam tanımlanmamış problemlerdeki eksiklikleri tamamlayabilir, tecrübelerine dayanarak otomatize edilemeyecek değerlendirmeler yapabilir.

Bilgisayarlar ise, farklı kaynaklardan çok miktarda veriyi entegre edebilirler, birden fazla işlemi bütün bir şekilde gerçekleştirebilirler, hafıza kısıtları yoktur ve kompleks işlemleri hızlıca ve hassas bir şekilde yürütebilirler [Tümay, 2010].

2.4. Karar Destek Sistemlerinin Uygulama Alanları ve Kullanılan Algoritmalar

KDS'nin geniş bir kullanım portföyü bulunmaktadır ve kurumun ihtiyaç duyduğu herhangi bir alanda geliştirilebilir. Şu ana kadar, pazarlamayla ilgili karar verme, tedarik zinciri yönetimi, lojistik, finansman-yatırım kararları, operasyon yönetimi, sağlık, güvenlik, orman yönetimi, askeri uygulamalar, ulaştırma, bankacılık, sigorta, personel yönetimi, eğitim, çizelgeleme, yol güvenliği, adli tıp, ulaştırma sistemleri vb. pek çok alanlarda KDS uygulamalarına rastlamak mümkündür.

KDS'lerin uygulanmasında sıklıkla kullanılan algoritmaları şu şekilde sıralayabiliriz.

- Genetik Algoritmalar
- Doğrusal Programlama

- Yapay Sinir Ağları (YSA)
- Uzman Sistemler (US)
- Bulanık Önermeler mantığı
- Destek Vektör Makineleri (DVM)
- Zaman Serileri
- Bayes Algoritması
- Analitik Hiyerarşi Prosesi (AHP)

Genetik algoritmalar, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar [wikipedia, 2012]. Genetik algoritmaların uygulama alanları oldukça geniştir. Optimizasyon problemlerinde (gezgin satıcı problemi, araç rotalama problemi, çinli postacı problemi, atama problemi ve sırt çantası problemi vb.), Otomatik Programlama ve Bilgi Sistemlerinde (bilgisayar çipleri tasarımı, ders veya sınav programı hazırlanması ve ağların çizelgelenmesi vb.), Mekanik öğrenmede (sınıflandırma, proteinin yapısal analizi, sinir ağları vb.), Ekonomik ve Sosyal Sistem Modellerinde (sembolik sistem tanımlama, kara kutu, veri madenciliği ve modelleme problemleri vb.) ve İşletme problemlerinde (finans, pazarlama, üretim/işlemler, montaj hattı dengeleme problemi, sistem güvenirliği problemi ve taşıma problemi vb.) başarılı bir şekilde kullanılmaktadır [Emel ve Taşkın, 2002].

Doğrusal programlama, kaynakların optimal dağılımını elde etmeye, maliyetleri minimize, karı ise maksimize etmeye yarayan bir tekniktir. Kullanım alanları daha çok, üretim ve stok planlama, malzeme kullanımı, petrol ve diğer sanayi sektörlerindeki optimal karışımın tayini, sabit maliyetlerin hesabı, gezgin satıcı problemleri ve kritik yol problemleri şeklindedir.

Yapay sinir ağları (YSA), insan beyninin bilgi işleme teknolojisinden esinlenerek geliştirilmiş bir bilgi işlem teknolojisidir. Bir insanın düşünme ve gözlemleme yeteneklerinin kullanılmasını gerektiren problemlere uygulanır. Robotik uygulamalar, proses kontrol, işlem planlama, kalite kontrol, gerçek zamanlı modelleme, adaptif kontrol, borsa endeksi, enflasyon ve kur tahmini bunlara örnektir.

Uzman sistemler, bir uzmandan alınan bilgilere dayanarak oluşturulan, karmaşık problemleri çözmek için olayları ve deneyimleri kullanan etkileşimli bilgisayar destekli karar aracıdır [Baykal ve Beyan, 2004]. Uzman sistemler kesin ve net algoritmalar yerine deneyime dayalı çıkarım yöntemleri kullanırlar. Bu nedenden ötürü uzman sistemlerin tasarımı karışık ve zaman alan bir işlemdir. Medikal tedavi, kalite kontrol, proses tasarımı ve seçimi, ürün tasarımı, işlem planlama, çizelgeleme, görüntü tanıma, robotik ve ses işleme kullanım alanlarıdır.

Bulanık önermelerde, bir olay ile ilgili karar verebilmek için olayın önce belirsizliği modellenmekte daha sonra belirsizlik altında elde edilen çözüm belirli hale getirilmektedir. Kullanım alanları daha çok süreç planlama, karar verme, elektrikli ev aletleri, oto elektroniği, fren sistemleri ve elektronik denetim sistemleridir.

Destek vektör makineleri, hem doğrusal olarak ayırt edilebilen hem de edilemeyen veri kümesini sınıflandırabilir. Uygun bir dönüşüm ile her zaman veri bir hiper düzlem ile iki sınıfa ayrılabilir. Hiper düzleme en yakın öğrenme verileri destek vektörleri olarak adlandırılır. Ses tanıma, yüz tanıma, patent değerleme, endeks getiri tahmini, arıza teşhisi gibi alanlarda ve çoğunlukla sınıflandırma problemlerinde uygulamaları görülmektedir.

Zaman serisi, eşit aralıklı t = 1, 2, ... , n zaman noktalarında Y değişkeniyle ilgili elde edilen y1, y2, ... , yn gözlem değerlerini zamana göre sıralanmış olarak gösteren seri olarak tanımlanır [Bowerman vd., 1993]. Uygulamalarına daha çok istatistik, sinyal işleme, ekonometri ve matematiksel finans alanlarında rastlanılmaktadır.

Bayes algoritması, hızlı bir biçimde sınıflama ve tahmin etme işlemleri için madencilik modelleri üretmeye yarar. Veri madenciliği uygulamalarında kümeleme işlemleri için sıklıkla kullanılır.

AHP, karar hiyerarşisinin tanımlanabilmesi durumunda kullanılan, kararı etkileyen faktörler açısından karar noktalarının yüzde dağılımlarını veren bir karar verme ve tahminleme yöntemi olarak açıklanabilir [Yılmaz, 1999]. Genel kullanım alanları iş değerlendirmesi, alternatif seçimi, kar/zarar karşılaştırmaları ve olası çıktıların tahmini şeklindedir [Gasimov, 2003].

Günümüzde önemli bir rekabet aracı olan Karar Destek Sistemleri'nde görüldüğü gibi birçok alanda farklı algoritmalar kullanılmaktadır. Problemlerin çözümü için hangi algoritmaların nerelere uygulanılması gerektiği sistemin performansı için oldukça önemlidir. Örneğin başlangıçtaki değerlerden bağımsız sonuca ulaşmamız gerekiyorsa, Genetik Algoritmaları kullanmak Yapay Sinir Ağlarına göre daha başarılı olacaktır.

Genetik Algoritmalar, optimizasyon ve işletme alanındaki problemlerde; Lineer Programlama, optimize edilecek amaç fonksiyonunun lineer olduğu durumlarda; Zaman Serileri, değişkenin zamana bağlı olduğu durumlarda; Yapay Sinir Ağları, öğrenme temelli tahminlerde; Bulanık Önermeler Mantığı, normal, yüksek ve yaklaşık gibi tanımlanan belirsiz durumları belirli hale getirmede; Destek Vektör Makineleri, problemin doğrusal olup olmamasından bağımsız sınıflandırma problemlerinde; Uzman sistemler, uzman tecrübesinin gerektiği alanlarda ve Analitik Hiyerarşi Prosesi ise çok kriterli problemleri çözmek için kullanılmaktadır.

BÖLÜM 3

GENETİK ALGORİTMALAR

Evrimsel programlamanın alt parçası olan Genetik Algoritmalar (GA) Gregor Mendel'in genetik yasalarından ve Darwin'in evrim teorisinden esinlenilerek ortaya çıkmıştır. Evrimsel programlama, 1960'lı yıllarda Ingo Rechenberg'in "Evrim Stratejileri" adlı çalışmasıyla gündeme gelmiştir. Genetik algoritmaların bugünkü biçimi ilk olarak Michigan Üniversitesi'nden, psikolog ve bilgisayar bilimleri uzmanı J. H. Holland tarafından 1975 yılında "Doğal ve Yapay Sistemlerde Adaptasyon" adlı kitabında ifade edilmiştir [Nabiyev, 2010]. 1985 yılında Holland'ın doktora öğrencisi David E. Goldberg'in gaz boru hatlarının döşenmesi ile ilgili çalışmasıyla Genetik Algoritmaların pratikte de uygulanabilen bir araştırma konusu olduğu ortaya konulmuştur [Saraçoğlu, 2009].

Genetik Algoritmalar, en iyinin korunumu ve doğal seçilim ilkesinin benzetim yoluyla bilgisayarlara uygulanması ile elde edilen bir arama yöntemidir [Nabiyev, 2010]. İteratif bir şekilde çözüm arayan, bireylerin yeteneklerini amaç fonksiyonuna göre değerlendiren optimizasyon yaklaşımıdır [Sheikh ve Lanjewar, 2010]. GA'ların genel amaçlı güçlü optimizasyon araçları vardır [Davis, 1991]. Karmaşık arama uzaylarında dahi global optimal çözümü bulma yeteneğine sahiptir [Burke vd., 1994].

Genetik Algoritmaların karakteristik özelliklerini sıralayacak olursak;

1. Değişken grubunun kendisi ile değil, kodlanmış hali ile çalışır.
2. Tek bir çözüm yerine, potansiyel çözümler kümesini (nüfus) geliştirmeye çalışır.
3. Bireylerin çözüm uzayında uygunluklarını belirlemek için amaç fonksiyonu kullanır.
4. Deterministik kurallara göre değil, olasılık tabanlı kurallara göre hareket eder.

3.1. Genetik Algoritmalarda Kullanılan Operatörler

Genetik algoritmalarda kullanılan operatörler, problemin çözüme ulaşmasındaki en önemli etkenlerdir. Bu operatörler için etkin bir seçim yapılması algoritmanın performansını doğrudan etkilemektedir. Yanlış yapılan seçimler, problemi, olası olmayan çözüm uzayında sonsuz bir döngüye itebilir. Dolayısıyla, gen kodlamasının, popülasyon büyüklüğünün, seçim, çaprazlama ve mutasyon operatörlerinin belirlenmesi GA'nın başarısını belirlediği için önemlidir.

3.1.1. Kromozom, Popülasyon ve Gen

Kromozom, genetik algoritmanın uygulandığı problemin olası çözümlerini ifade eder. Genetik algoritmaların çözüm uzayı geniş olduğu için, sonuca çözüm topluluğu ile gidilir. Her bir çözüm kromozom, kromozomların oluşturduğu topluluk da popülasyon olarak adlandırılır. Gen ise kromozomu oluşturan birimdir ve çözümle ilgili veri genlerle temsil edilir.

3.1.2. Kromozomun Kodlanması

Kromozomlar, kodlanmış bilgiyi, çözümü temsil ederler. Kromozomların kodlanması problemin yapısına göre değişkenlik gösterir. Kurulan genetik modelin doğru ve hızlı çalışması için kodlama seçiminin iyi yapılması gerekmektedir. Permütasyon kodlama, ikili kodlama, değer kodlama ve ağaç kodlama sıklıkla kullanılan kodlama türleridir.

3.1.2.1. İkili Kodlama

İkili kodlama, en yaygın olarak kullanılan kodlama yöntemidir. Genetik algoritma ile yapılan ilk çalışmalarda bu kodlama tipi kullanılmıştır. Genler 0 ve 1'lerle ifade edilir. Her bir bit çözümün özellikleri hakkında bir bilgiyi temsil eder. Basit ikili kodlamada gen değerleri tek bir bitle ifade edilirken, bazı problem tiplerinde genler bit kümesi ile de gösterilmektedir.

3.1.2.2. Permütasyon Kodlama

Permütasyon kodlamada genler nümerik değerler ile temsil edilir ve tüm değerler tekildir. Gezgin Satıcı Problemi (GSP) ve iş sıralama problemlerinde ağırlıklı olarak kullanılır. Permütasyon kodlamanın kullanıldığı tüm problemlerde kromozomlar ikili kodlama ile de ifade edilebilir [Nabiyev, 2010]. Ancak büyük nümerik değerler ikili sistemde fazla sayıda bitlerle temsil edileceği için takibi zorlaştırmakta ve bellek kullanımını arttırmaktadır. Örneğin, 13 sayısı ikili sistemde 01101 şeklinde ifade edilir ve tek bir gen 5 bitlik yer kaplar.

3.1.2.3. Değer kodlama

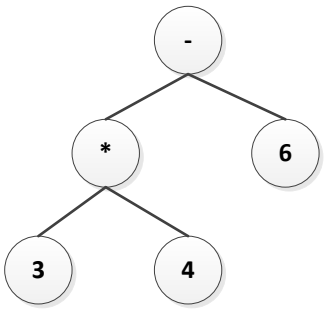
Değer kodlama, kompleks sayıların kullanıldığı problemlerde kullanılır ve genler gerçel sayı, karakter veya karakter dizisi ile gösterilir. Değer kodlamanın kullanıldığı problem tiplerinde ikili kodlamanın kullanımı oldukça zordur [Nabiyev, 2010]. Şekil 3.1'de değer kodlamanın bir örneği verilmiştir.

K1	3.274	6.121	4.345	0.978	1.586	4.876	3.908	0.132	2.836

Şekil 3.1. Değer kodlama için kromozom örneği

3.1.2.4. Ağaç kodlama

Ağaç kodlamada her kromozom, nesneler ve nesneler arası işlemleri içeren bir ağaç yapısından oluşur. Genellikle programlar ve ifadeler oluşturmak için kullanılır. Örneğin, LISP ve Prolog gibi programlama dillerinde ağaç yapısı sıklıkla kullanılan bir kodlama türüdür [Nabiyev, 2010]. Şekil 3.2'de bir örneği gösterilmiştir.



Şekil 3.2. Ağaç kodlama için kromozom örneği

3.1.3. Uygunluk Fonksiyonu

Popülasyondaki her bir aday çözümün, aranan olası çözüme yakınlığını ifade eden fonksiyondur. Uygunluk fonksiyonunun belirlenmesi genetik algoritmanın en temel adımlarından biridir. Probleme bağımlılığı söz konusu olduğundan, her problemin yapısına ve içeriğine göre uygunluk fonksiyonu da değişiklik gösterir. Ayrıca genetik algoritmada olası bir çözümün kısıtları sağlayıp sağlamadığını uygunluk fonksiyonu ile değerlendirmek mümkündür. Herhangi bir kısıtı sağlamayan kromozomun uygunluk değeri cezalandırma yöntemi kullanılarak problemin yapısına göre (maks./min.) arttırılır veya azaltılır. Böylelikle çözüme ulaşmak daha kolay hale gelmektedir. Çoğu zaman algoritmanın başarısı, bu fonksiyonun doğru belirlenmesine bağlıdır.

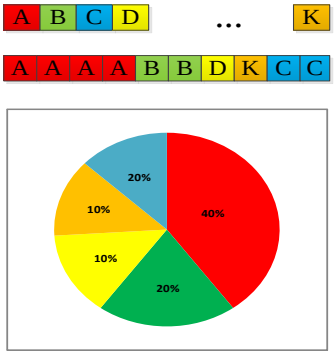
3.1.4. Seçim

Seçim süreci, optimum uygunluk değerine sahip kromozomların hayatta kalmasına yönelik yapılan bir işlemdir. Popülasyondaki her bir bireyin karakteristik özelliği, olası çözüm olup olmamasından bağımsız olarak, amaç fonksiyonundan elde edilen değer ile ölçülür. Ayrıca bu değer, yeni neslin türetilme sürecinde seçilme olasılığını da belirler. Problemin yapısına, maksimizasyon veya minimizasyon olmasına göre maksimum veya minimum uygunluk değerine sahip bireyler seçilir, kötü bireyler elimine edilir.

Darwin'in "iyi olan yaşar" prensibine göre iyi kromozomların seçimi için geliştirilen kriterlerden bazıları Rulet tekeri seçimi, Boltzman seçimi, elitizm, rank seçimi ve sabit durum seçimidir.

3.1.4.1. Rulet Tekeri Seçimi

Rulet tekeri seçiminde bireylerin uygunluk değerleri dikkate alınır. Nesildeki her bir bireyin uygunluk değeri, toplam uygunluk değerine bölünerek normalizasyon işlemi yapılır. [0,1] aralığındaki değerler bireylerin seçilme olasılığını belirler. Bireyler seçilme olasılıklarının büyüklüğüne göre rulet tekerinde bir yüzdeliğe sahip olurlar (Şekil 3.3). Teker üzerinde yapılan rastgele bireylerin seçimi yeni nesli oluşturur. Bu durumda rulet tekerinde daha büyük alana sahip olan bireyin seçilme olasılığı daha fazla olacaktır.



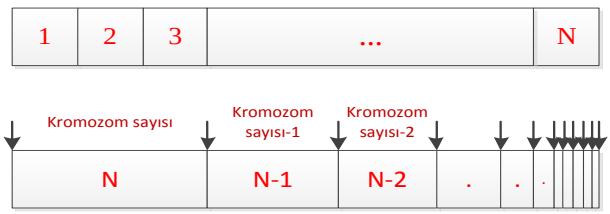
Şekil 3.3. Rulet tekeri

3.1.4.2. Elitizm

Çaprazlama ve mutasyon işlemleri, çalıştıkları oran değerine bağlı olarak, iyi bireylerin kromozom yapısını değiştirdikleri için bu bireylerin kaybolmasına neden olmaktadır. Bu durumu önlemek amacıyla geliştirilen elitizmde, iyi bireyler korunarak bir sonraki nesle aktarılır. Aktarılan bireylerin sayısının belirlenmesi yine çaprazlama ve mutasyonda olduğu gibi oran değerine bağlıdır. Popülasyonun tamamlanması ise kromozomların çaprazlanmasından elde edilen yeni bireylerle gerçekleşir.

3.1.4.3. Rank Seçimi

Rulet tekeri seçiminde, seçilme yüzdeleri arasında çok büyük farklar varsa, bu durum popülasyondaki çeşitliliğin azalmasına neden olur. Rank seçiminde ise öncelikle, kromozomlar optimum uygunluk kriterine (maksimizasyon/minimizasyon) göre sıralanır. En iyi uygunluk değerine sahip bireye N, en kötüsüne ise 1 değeri verilir (Şekil 3.4). N, kromozom sayısını ifade etmektedir. Rulet tekerinde olduğu gibi bu değerler toplanır ve her bir bireyin aldığı değere bölünerek seçilme olasılıkları belirlenir. Fakat bu yöntemde de en iyi kromozomların diğerlerine göre seçilme olasılığı çok farklı olmadığından çözüme yakınsama zaman alacaktır.



Şekil 3.4. Rank seçimi

3.1.5. Çaprazlama

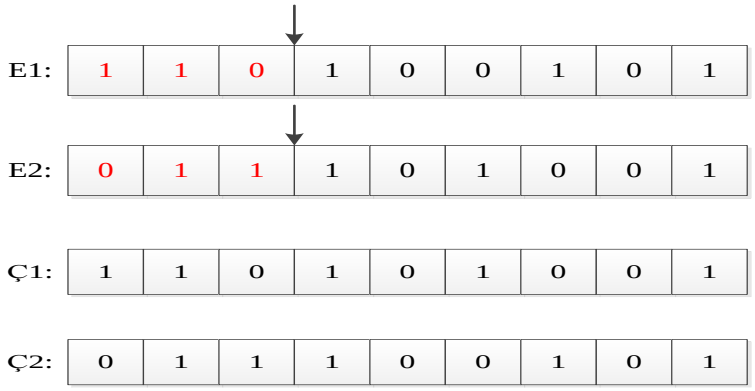
Çaprazlama GA'nın en önemli adımlarından biridir. Karşılıklı iki kromozom arasında, rastgele seçilen genlerin değişimini ifade eder. Amaç, bireylerdeki iyi özelliklerin bir araya gelmesini sağlayarak daha iyi çözümler elde etmek ve nesilde çeşitliliği sağlamaktır.

Çaprazlama yöntemleri kullanılan kodlamanın türüne göre farklılık göstermektedir. İkili kodlamada genel olarak kullanılan çaprazlama yöntemleri 4 çeşittir.

- Tek nokta çaprazlama
- İki nokta çaprazlama
- Uniform çaprazlama
- Aritmetik çaprazlama

3.1.5.1. Tek Nokta Çaprazlama

Çaprazlamanın yapılacağı kromozomlarda rastgele bir çaprazlama noktası belirlenir. Bu noktaya kadar olan gen değerleri (bitler) ilk ebeveynden, noktadan sonraki kısım diğer ebeveynden alınarak çocuk kromozom elde edilir. Şekil 3.5'te çaprazlama sonucu elde edilen yeni bireyler gösterilmiştir.

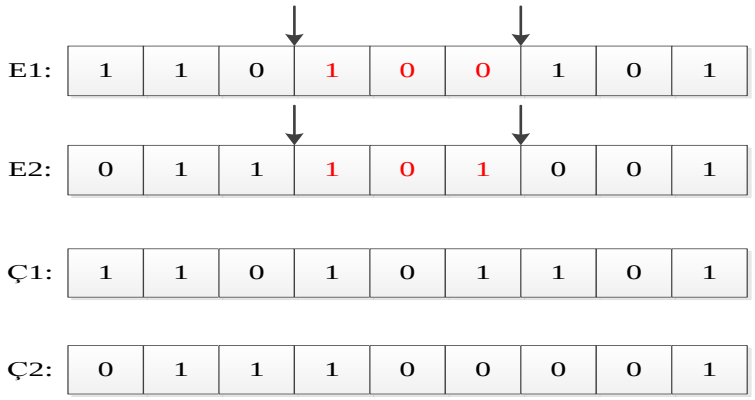


Şekil 3.5. Tek nokta çaprazlama

3.1.5.2. İki Nokta Çaprazlama

Çaprazlamanın yapılacağı kromozomlarda rastgele iki konum belirlenir. İlk konuma kadar olan kısım birinci ebeveynden, iki konum arasındaki kısım ikinci ebeveynden, geri kalan kısım yine birinci ebeveynden alınarak çocuk kromozom elde edilir.

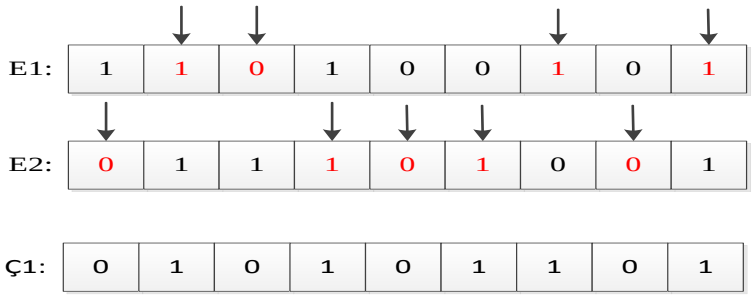
Şekil 3.6'da iki nokta çaprazlamanın gerçeklenişi gösterilmiştir.



Şekil 3.6. İki nokta çaprazlama

3.1.5.3. Uniform Çaprazlama

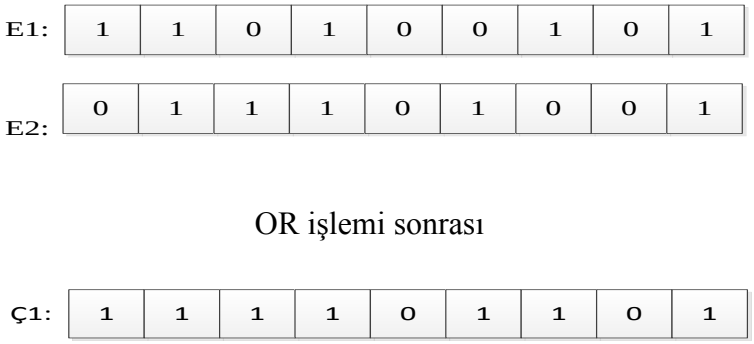
Bu çaprazlama yönteminde her iki ebeveynden de rastgele seçilen gen değerleri (bitler) çocuk kromozoma kopyalanır. Şekil 3.7'de uniform çaprazlamada ebeveynlerden seçilen rastgele gen değerlerinin çocuğa kopyalanması gösterilmiştir.



Şekil 3.7. Uniform çaprazlama

3.1.5.4. Aritmetik Çaprazlama

Yeni bir kromozom oluşturmak için, çaprazlama yapılacak ebeveynler arasında farklı aritmetik işlemler uygulanabilir. And, Or, Xor vb. gibi. Şekil 3.8'de aritmetik işlemlerden Or işlemi uygulanması sonucu elde edilen çocuk kromozom gösterilmiştir.



Şekil 3.8. Aritmetik çaprazlama

Literatürde, permütasyon kodlama ile ilgili birçok çaprazlama operatörü ele alınmıştır. Bu operatörler Tablo 3.1'de verilmiştir.

Tablo 3.1. Çaprazlama operatörleri [Larranaga, 1999].

Çaprazlama Operatörleri	Geliştirenler
Kısmi- haritalanmış çaprazlama (PMX)	Goldberg and Lingle (1985)
Sıralı çaprazlama (OX1)	Davis (1985)
Sıra tabanlı çaprazlama (OX2)	Syswerda (1991)
Pozisyon tabanlı çaprazlama (POS)	Syswerda (1991)
Sezgisel çaprazlama	Grefenstette (1987b)
Sıralı eşlemeli çaprazlama	Brady (1985)
Döngüsel çaprazlama (CX)	Oliver, Smith, Holland (1987)
Maksimal koruyucu çaprazlama (MPX)	Mühlenbein, Gorges- Schleuter vd. (1988)
Genetik kenar birleşmeli çaprazlama (ER)	Whitley vd. (1989, 1991)
Değişen pozisyonlar çaprazlama (AP)	Larranaga vd. (1996a)
Oy(Voting) birleşmeli çaprazlama (VR)	Mühlenbein (1989)

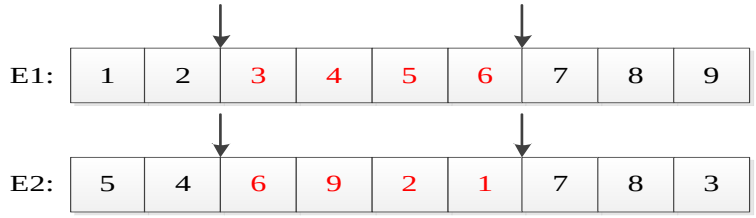
Geçmişteki çalışmalara bakıldığında problemin yapısına göre uygulanacak olan çaprazlama operatörlerinin değişkenlik gösterdiği görülmüştür. Bu sebeple birçok çaprazlama yöntemi geliştirilmiştir. Uygulamalarda sıklıkla kullanılan, kısmi-haritalanmış, sıralı, sıra tabanlı ve pozisyon tabanlı çaprazlama yöntemleri karşımıza çıkmaktadır.

3.1.5.5. Kısmi-Haritalanmış Çaprazlama (Partially Mapped Crossover- PMX)

Goldberg ve Lingle (1985) tarafından önerilen PMX operatöründe çaprazlama işlemi ikili kodlamadaki iki nokta çaprazlaması gibi yapılır ve iki nokta arası "mapping section" olarak adlandırılır. Çaprazlama sonrası oluşacak aykırı durumlar için düzeltme prosedürü uygulanır.

Çaprazlamanın gerçekleşme adımları şu şekildedir;

1. Çaprazlamanın yapılacağı aralığın belirlenmesi için Şekil 3.9'da olduğu gibi rastgele iki konum seçilir.



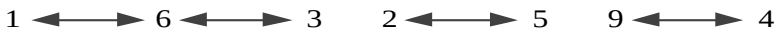
Şekil 3.9. PMX'te rastgele konumların belirlenmesi

2. Seçilen konumlar arasındaki genler ebeveynler arasında birebir yer değiştirilir. Şekil 3.10'da yer değiştirilmiş hali gösterilmiştir.



Şekil 3.10. PMX'te gen değişimi

3. Bu iki alt parçalar arasında ilişki kurulur. Şekil 3.11'de kurulan ilişki görülmektedir.



Şekil 3.11. PMX'te gen değerleri arasındaki ilişki

4. Kurulan ilişki doğrultusunda çocuk kromozomlardaki aykırı durumlar ve gen tekrarları, düzeltilir. Şekil 3.12'de elde edilen çocuk kromozomların son hali gösterilmiştir [Gen ve Cheng, 1997].



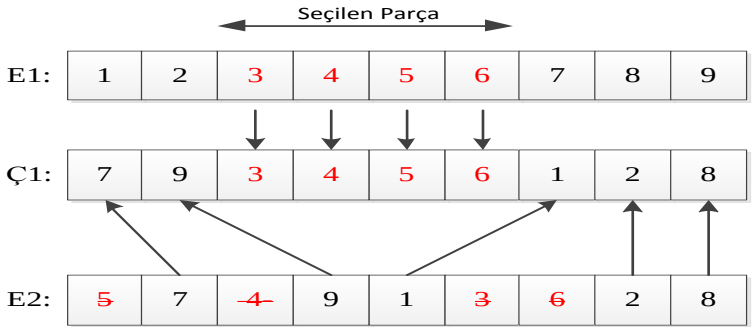
Şekil 3.12. PMX sonrası oluşan yeni bireyler

3.1.5.6. Sıralı Çaprazlama (Order Crossover- OX1)

Davis (1985) tarafından ileri sürülen OX1 operatöründe çaprazlama işlemi algoritması aşağıdaki gibidir.

1. İlk ebeveynden rastgele bir kromozom parçası seçilir.
2. Seçilen parça aynı sırada ve aynı konumda çocuk kromozoma kopyalanır.
3. Seçilen kromozom parçasındaki değerler ikinci ebeveynden silinir.
4. İkinci ebeveynde silme işleminden sonra kalan değerler aynı sırada çocuk kromozomdaki boş konumlara yerleştirilir.

Şekil 3.13'te OX1'in gerçekleşme adımları görülmektedir.



Şekil 3.13. OX1'in gerçekleşme şekli

Aynı işlemler diğer çocuk kromozom için ebeveyn konumları değiştirilerek yapılır. Şekil 3.14'te elde edilen diğer kromozom gösterilmiştir [Gen ve Cheng, 1997].



Şekil 3.14. OX1 sonrası yeni birey

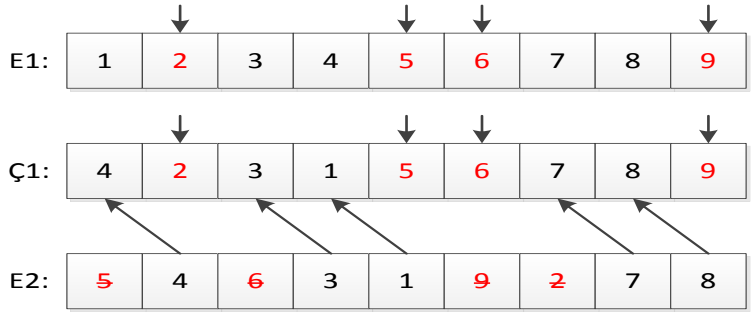
3.1.5.7. Pozisyon Tabanlı Çaprazlama (Position Based Crossover- POS)

Pozisyon temelli çaprazlama, Syswerda'nın 1991'deki çalışmasıyla ortaya konulmuştur.

Algoritması;

1. İlk ebeveynden rastgele seçilen konumlardaki genlerin kümesi oluşturulur.
2. Seçilen gen değerleri aynı konumlarda çocuk kromozoma kopyalanır.
3. İkinci ebeveynden seçilen değerler silinir.
4. İkinci ebeveynde kalan gen değerleri soldan sağa doğru çocuk kromozoma yerleştirilerek kromozom tamamlanır.

   POS'un gerçekleşme adımları Şekil 3.15'te verilmiştir.



Şekil 3.15. POS gerçekleşme şekli

Aynı işlemler diğer çocuk kromozom için ebeveyn konumları değiştirilerek yapılır. Şekil 3.16'da POS sonrası elde edilen diğer birey gösterilmiştir [Gen ve Cheng, 1997].



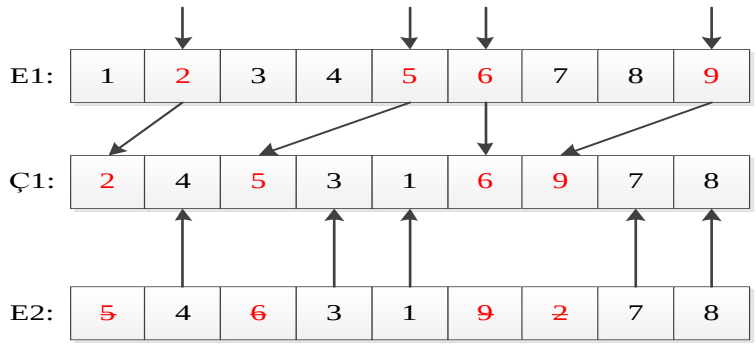
Şekil 3.16. POS sonrası yeni birey

3.1.5.8. Sıra Tabanlı Çaprazlama (Order Based Crossover- OX2)

Syswerda (1991) tarafından geliştirilen OX2 operatörü, kısmen pozisyon temelli çaprazlama ve sıralı çaprazlama operatörlerine benzemektedir. İşleyiş şekli;

1. Sıralı çaprazlamada rastgele kromozom parçası seçilirken, OX2'de rastgele konumların gen değerlerinin oluşturduğu bir küme seçilir.
2. Bu konumlardaki değerler pozisyon tabanlı çaprazlamada olduğu gibi aynı konum bilgilerine göre değil, ikinci ebeveyndeki konumlarına göre çocuk kromozoma yerleştirilir ve bu değerler ikinci ebeveynden silinir.
3. İkinci ebeveynde geriye kalan değerler soldan sağa doğru çocuk kromozomdaki boş alanlara yerleştirilir.

   OX2'nin gerçekleşme adımları Şekil 3.17'de verilmiştir.



Şekil 3.17. OX2 gerçekleşme şekli

Aynı işlemler diğer çocuk kromozom için ebeveyn konumları değiştirilerek yapılır. Şekil 3.18'de OX2 sonrası oluşan yeni birey görülmektedir [Gen ve Cheng, 1997].



Şekil 3.18. OX2 sonrası yeni birey

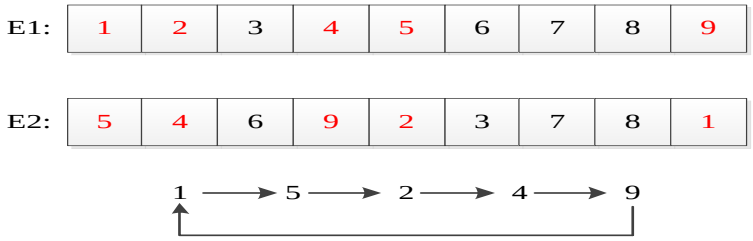
3.1.5.9. Döngüsel Çaprazlama (Cycle Crossover- CX)

Oliver ve arkadaşları (1987) tarafından ortaya konmuştur. Pozisyon tabanlı çaprazlama operatörü ile benzerlik göstermektedir. Farkı, ilk ebeveynden gen değerleri

rastlantısal olarak değil, iki ebeveyn arasında kurulan döngüsel ilişkiye göre seçilir. Çalışma şekli;

1. Ebeveynler arasında karşılıklı döngünün sağlanacağı gen değerleri tespit edilir.

   Şekil 3.19'da döngüyü sağlayan gen değerleri ve aradaki ilişki gösterilmiştir.



Şekil 3.19. CX için döngüyü sağlayan gen değerlerinin tespiti

2. Döngüdeki gen değerlerinin ilk ebeveyndeki konumları baz alınarak çocuk kromozoma kopyalanır. Şekil 3.20'de ilk ebeveynden gelen gen değerlerinin çocuk kromozomlardan birine yerleştirilmesi gösterilmiştir.



Şekil 3.20. CX'te yeni bireyin oluşumu

3. Kopyalanan gen değerleri ikinci ebeveynden silinir. Şekil 3.21'de gen tekrarına neden olmaması için diğer ebeveynden gen değerlerinin silinmesi gösterilmektedir.



Şekil 3.21. CX 'te ebeveynlerden gen kopyalanması

4. İkinci ebeveynde kalan değerler konum sırasına göre çocuk kromozoma yerleştirilir. Şekil 3.22'de ikinci ebeveynden gelen değerlerle çocuk1 kromozomun elde edilmiş hali verilmiştir.



Şekil 3.22. CX sonrası oluşan yeni birey

Aynı işlemler diğer çocuk kromozom için ebeveyn konumları değiştirilerek yapılır. Şekil 3.23'te çocuk2 kromozomun son hali görülmektedir [Gen ve Cheng, 1997].



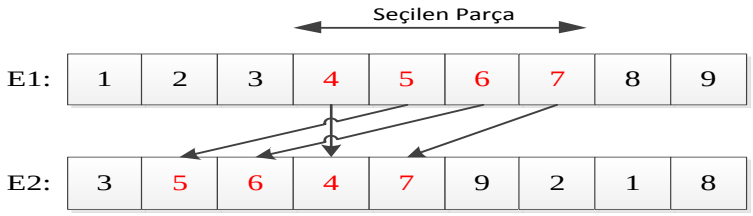
Şekil 3.23. CX sonrası oluşan diğer birey

3.1.5.10. Sezgisel Çaprazlama (Heuristic Crossover)

Grefenstette 1987'de kenar bilgileri üzerinde duran sezgisel çaprazlama operatörleri sınıfını geliştirmiştir. Liepins ve arkadaşları tarafından da kullanılan bu operatörün algoritması;

1. İlk ebeveynden rastlantısal olarak gen değerleri seçilir.
2. Seçilen gen değerlerinin farklı bir permütasyonu ikinci ebeveynde aranır.

İlk ebeveynden seçilen parça ve onun diğer bireyde permütasyonunun elde edilmesi Şekil 3.24'te görülmektedir.



Şekil 3.24. Sezgisel çaprazlamada permütasyon oluşturan gen değerlerinin belirlenmesi

3. Bu kromozom parçasının farklı dizilimleri ebeveynler arasında yer değiştirir.

Elde edilen permütasyonların bireyler arasında yer değiştirmesi Şekil 3.25'te verilmiştir [Gen ve Cheng, 1997].



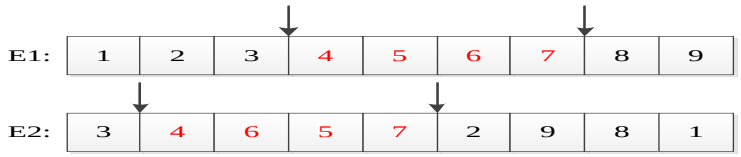
Şekil 3.25. Sezgisel çaprazlama sonrası oluşan bireyler

3.1.5.11. Sıralı Eşlemeli Çaprazlama (Sorted Match Crossover)

Brady (1985) tarafından önerilen bir çaprazlama operatörüdür. Mühlenbein ve arkadaşlarının (1988) çalışmasında da yer almaktadır. Sezgisel çaprazlamanın kısıtlanmış durumu gibi düşünülebilir. Algoritması;

1. Her iki ebeveynde de, aynı uzunlukta, aynı gen değeri ile başlayıp aynı gen değeri ile biten ve bu iki gen değeri arasında aynı değerleri barındıran alt parça(dizilim) aranır.

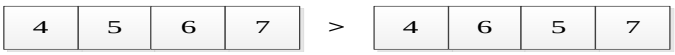
Belirtilen kısıtlar altında bulunan kromozom parçaları Şekil 3.26'da gösterilmiştir.



Şekil 3.26. Sıralı eşlemeli çaprazlamada kromozom parçasının belirlenmesi

2. Bu alt parçaların maliyeti hesaplanır.

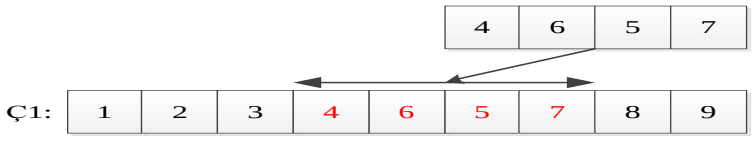
Belirlenen kromozom parçalarının maliyet karşılaştırılması Şekil 3.27'de verilmiştir.



Şekil 3.27. Sıralı eşlemeli çaprazlamada maliyet karşılaştırması

3. Çocuk kromozom, maliyeti yüksek olan parçanın ait olduğu ebeveynden, bu parça çıkarılıp, maliyeti az olan parçanın gelmesi ile elde edilir.

Maliyeti az olan parçanın çocuk kromozoma kopyalanması Şekil 3.28'de gösterilmektedir.



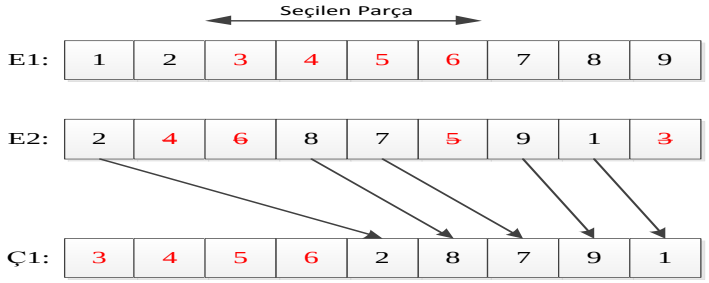
Şekil 3.28. Sıralı eşlemeli çaprazlama sonrası yeni birey

Mühlenbein ve arkadaşları (1988) çalışmalarında, Sıralı Eşlemeli Çaprazlama'nın hesaplama zamanını azalttığını fakat çaprazlama için zayıf bir şema olduğunu belirtmiştir [Larranaga, 1999].

3.1.5.12. Maksimal Koruyucu Çaprazlama (Maximal Preservative Crossover-MPX)

MPX operatörü ilk olarak Mühlenbein ve arkadaşları (1988) tarafından tanıtılmıştır. PMX operatörüne benzer bir şekilde çalışır. Uygulamada çok küçük problem örnekleri göz ardı edilir. Çalışma şekli Şekil 3.29'da görüldüğü üzere;

1. İlk ebeveynden rastgele bir kromozom parçası seçilir. Fakat seçilen parçanın uzunluğu, 10'dan büyük veya eşit, kromozom uzunluğunun yarısından da küçük veya eşit olmalıdır. Bu kısıtlamanın getirilmesinin nedeni, ebeveynlerden çok fazla bilgi kaybetmeden, ebeveynler arasında yeterince bilgi alış verişini sağlamaktır.
2. Seçilen gen değerleri ikinci ebeveynden silinir.
3. İlk ebeveynden seçilen kromozom parçası, çocuk kromozoma başlangıç noktasından itibaren konumlanır.
4. İkinci ebeveyndeki değerler aynı sırada çocuk kromozoma yerleştirilir [Larranaga, 1999].



Şekil 3.29. MPX'in gerçekleşme şekli

Aynı işlemler diğer çocuk kromozom için ebeveyn konumları değiştirilerek yapılır. MPX sonrası elde edilen çocuk2 kromozom Şekil 3.30'daki gibidir.



Şekil 3.30. MPX sonrası oluşan diğer birey

3.1.6. Mutasyon

Mutasyon, biyolojik sistemlerde olduğu gibi, bireylerin genetik yapısında küçük bir değişiklik ile kendini gösterir. Problemin çözümünde, yerel optimuma takılmayı önler ve daha iyi sonuçlar için aramanın devam etmesini sağlar.

İkili kodlamada kullanılan mutasyon operatörü permütasyon kodlamada kullanılanlar kadar çeşitlilik göstermemektedir. 0 ve 1'lerden oluşan gen değerlerinde mutasyon işlemi rastgele seçilen gen değerinin tersinin alınması ile gerçekleşir. 0 gen değeri 1'e, 1 gen değeri 0'a dönüştürülür.

Permütasyon kodlamada, çaprazlama operatörlerinde olduğu gibi çeşitli mutasyon teknikleri geliştirilmiştir. Tablo 3.2'de mutasyon operatörleri ve kim tarafından geliştirildikleri verilmiştir.

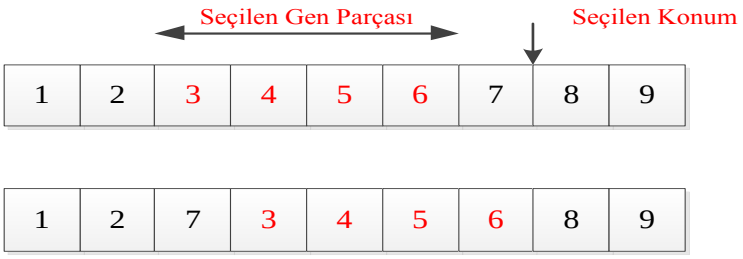
Tablo 3.2. Mutasyon operatörleri [Larranaga, 1999]

Mutasyon operatörleri	**Geliştirenler (Yıl)**
Yerini alma mutasyonu(DM) Kesme mutasyonu (CM)	Michalewicz (1992) Banzhaf (1990)
Yer değişimi mutasyonu (EM) Karşılıklı değişim mutasyonu (REM) Nokta mutasyonu Sıra tabanlı mutasyon	Banzhaf (1990) Michalewicz (1992) Ambati vd. (1991) Syswerda (1991)
Araya yerleştirme mutasyonu (ISM) Pozisyon tabanlı mutasyonu	Fogel (1988) Syswerda (1991)
Basit ters çevirme mutasyonu (SIM)	Holland (1975)
Ters çevirme mutasyonu (IVM) Kesme- ters çevirme mutasyonu	Fogel (1990) Banzhaf (1990)
Karıştırma mutasyonu (SM)	Syswerda (1991)
Sezgisel mutasyon (HM)	Cheng ve Gen (1997)

3.1.6.1. Yerini Alma (Displacement Mutation- DM)

Yerini alma mutasyon operatörü (Michalewicz, 1992) ilk olarak rastgele bir kromozom parçası seçer. Seçilen kromozom parçası ebeveynden çıkartılır ve yine rastgele belirlenen bir konumdan itibaren yerleştirilir.

DM mutasyonunun uygulanması Şekil 3.31'de gösterilmektedir.



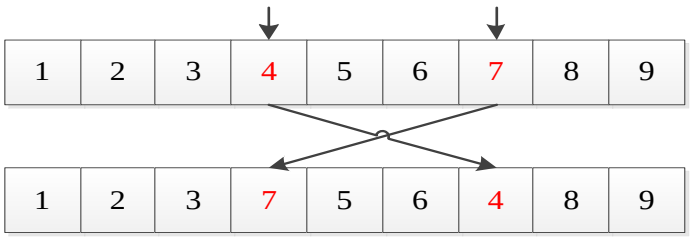
Şekil 3.31. DM mutasyonu

Yerini alma mutasyonu Banzhaf'ın (1990) çalışmasında "Kesme Mutasyon" olarak geçmektedir [Larranaga, 1999].

3.1.6.2. Yer Değişimi (Exchange Mutation- EM)

Banzhaf (1990) tarafından ortaya atılan bu mutasyon operatörü, ebeveyn üzerinde seçtiği rastgele iki konumdaki gen değerlerini değiştirme işlemi yapar.

EM mutasyonunun uygulanması Şekil 3.32'de gösterilmektedir.

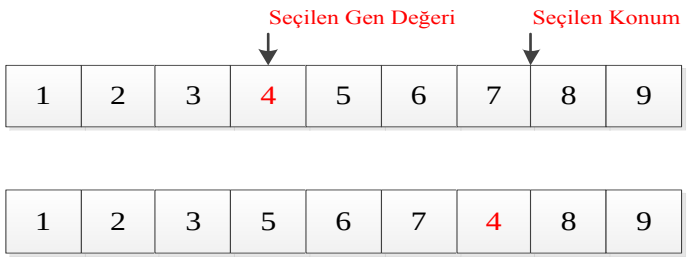


Şekil 3.32. EM mutasyonu

3.1.6.3. Araya Yerleştirme (Insertion Mutation- ISM)

Araya yerleştirme yöntemi (Fogel, 1988; Michalewicz, 1992), rastgele seçtiği gen değerini, ebeveynde bulunduğu konumdan çıkartıp, rastgele belirlediği konuma ekler.

ISM mutasyonunu uygulanması Şekil 3.33'te gösterilmektedir.



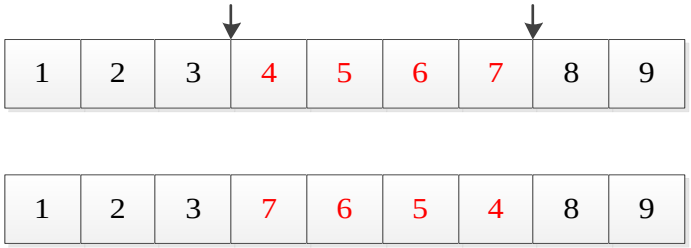
Şekil 3.33. ISM mutasyonu

Syswerda (1991) çalışmasında araya yerleştirme mutasyon operatöründen "Pozisyon Tabanlı Mutasyon" olarak bahsetmektedir [Larranaga, 1999].

3.1.6.4. Basit Ters Çevirme (Simple Inversion Mutation- SIM)

SIM (Holland, 1975; Grefenstette, 1987) ebeveynde rastgele iki tane kesme noktası belirler ve bu iki nokta arasında kalan kromozom parçasını ters çevirir [Larranaga, 1999].

SIM mutasyonunun uygulanması Şekil 3.34'te gösterilmiştir.

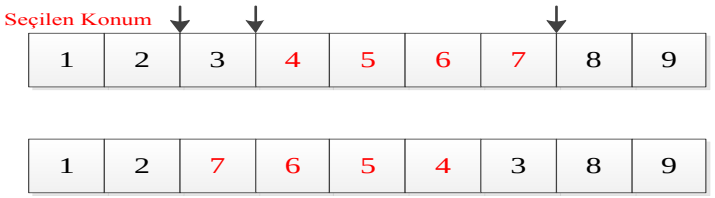


Şekil 3.34. SIM mutasyonu

3.1.6.5. Ters Çevirme (Inversion Mutation- IVM)

Ters çevirme (Fogel, 1990) yöntemi yer değiştirme yöntemi ile benzerlik göstermektedir. Rastgele seçilen kromozom parçası ebeveynden çıkartılır, ters çevrilerek yine rastgele seçilen konuma yerleştirilir.

IVM mutasyonunun uygulanması Şekil 3.35'te gösterilmiştir.



Şekil 3.35. IVM mutasyonu

Banzaf (1990) çalışmasında bu yöntemden "Kesme - Ters Çevirme" olarak söz etmiştir [Larranaga, 1999].

3.1.6.6. Karıştırma (Scramble Mutation- SM)

Karıştırma (Syswerda, 1991) mutasyonu, rastgele seçtiği kromozom parçasında gen değerlerini karıştırma işlemi yapar [Larranaga, 1999].

SM mutasyonunun uygulanması Şekil 3.36'da gösterilmiştir.

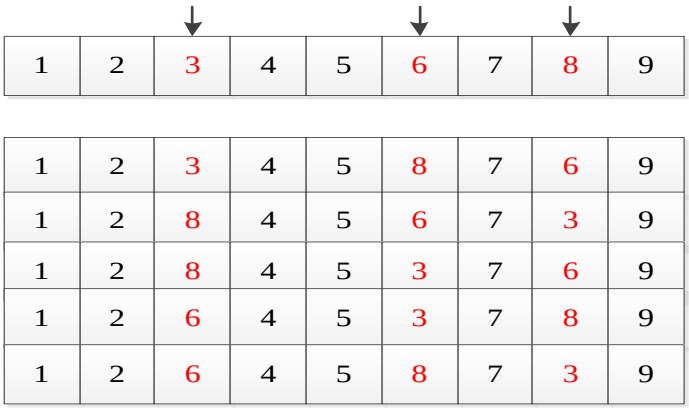


Şekil 3.36. SM mutasyonu

3.1.6.7. Sezgisel (Heuristic Mutation- HM)

Sezgisel mutasyon (Cheng ve Gen, 1997), komşuluk tekniğini kullanarak yeni kromozomları iyileştirmeye çalışır. İlk olarak λ değerinde rastgele gen değerleri seçilir. Seçilen gen değerlerinin tüm permütasyonları mevcut konumlar arasında değiştirilerek elde edilir. Uygunluk değerlerine göre çözüme yakın olan kromozom seçilir.

HM mutasyonunun uygulanması Şekil 3.37'de gösterilmiştir.



Şekil 3.37. HM mutasyonu

3.2. Genetik Algoritmaların Parametreleri

Genetik algoritmalarda kullanılan temel parametreler, çaprazlama oranı, mutasyon oranı ve popülasyon büyüklüğüdür. İterasyon sayısı da dâhil edilebilir fakat belirli bir çözümü olan problemlerde durdurma kriteri istenilen çözüme ulaşma olduğundan iterasyon sayısı parametre olarak kullanılmamaktadır. Parametre değerlerinin problemin yapısına uygun olarak belirlenmesi, çözüme ulaşmayı kolaylaştırdığı için önemlidir.

3.2.1. Çaprazlama Oranı

Çaprazlamaya uğrayacak bireylerin sayısını belirlemek için kullanılır. Çaprazlama olasılığı P_c ile gösterilir ve [0,1] arasında değerler alır. Oranın doğru bir şekilde belirlenmesi önemlidir. Yüksek bir değer seçildiğinde, çözüm uzayı çabuk taranır fakat iyi bireylerin kromozom yapısı değiştiği için çözümden uzaklaşma ihtimali söz konusudur, düşük bir değer seçildiğinde ise, yeni birey sayısı azaldığından, nesilde birey tekrarının meydana gelmesi muhtemeldir. Çaprazlama oranı ile popülasyon büyüklüğünün çarpımı, kaç tane kromozoma çaprazlama uygulanacağı bilgisini verir. Örneğin, P_c =0.5, popülasyon büyüklüğü 50 ise, çaprazlama işlemi 25 kromozoma uygulanacaktır.

3.2.2. Mutasyon Oranı

Mutasyon oranı, kromozom içerisinde mutasyon işleminin uygulanacağı gen değerlerinin sayısını elde etmek için kullanılır. Gen değerlerinin mutasyona uğrama olasılığı ise P_m ile gösterilmektedir. Mutasyon oranının doğru bir şekilde belirlenmesi önemlidir. Küçük bir oran, çözümlerin yerel optimuma düşmesine neden olurken, büyük bir oran bilgi kaybına ve sürecin rastlantısallığa gitmesine neden olur. Mutasyona uğrayacak gen değerlerinin sayısı P_m * kromozom uzunluğu*popülasyon büyüklüğü şeklinde hesaplanır.

3.2.3. Popülasyon Büyüklüğü

Popülasyonun kaç tane kromozomdan oluşacağını ifade eder. Bunun için belirlenmiş bir standart yoktur. Popülasyon büyüklüğünün seçimi, problemin çözümünde yapılan işlemlerin karmaşıklığına ve arama uzayının derinliğine göre farklılık gösterir. Az sayıda kromozomdan oluşan popülasyonlarda, uygun olmayan çözümlere erken yakınsama söz konusu iken, kromozom sayısının fazla olduğu durumlarda da araştırma aralığının büyük olmasından kaynaklanan sürecin uzaması söz konusudur.

3.3. Genetik Algoritmalarda Kısıtların Değerlendirilmesi

Genetik algoritmaların kısıtlı optimizasyon problemlerine uygulanmasındaki temel sorun, kısıtlarla nasıl başa çıkılacağıdır. Prensip olarak rastsal hareket eden algoritmanın her adımında kısıtların sağlanıp sağlanmadığının kontrolü gerekmektedir. Kromozomlar üzerinde uygulanan genetik operatörler (çaprazlama, mutasyon), uygun olmayan çözümler üretilmesine yani kısıtların ihlaline neden olabilmektedir. Söz konusu durum için bazı yöntemler geliştirilmiştir.

3.3.1. Reddeden Yöntem

Reddeden yöntem, evrim süreci boyunca uygun olmayan kromozomların elenmesi mantığına dayanır. Yaygın olarak kullanılan bir yöntemdir. Tüm arama uzayını olası çözümlerin olduğu uzaya indirgediği için çözüme daha kısa sürede ulaşılmasını sağlar. Fakat uygun olmayan bir çözümün istenilen çözüme yakın olması

burada bir anlam ifade etmeyecek ve o kromozoma ait bilgilerin kaybolmasına neden olacaktır.

3.3.2. Düzelten Yöntem

Düzeltme yaklaşımı, uygun olmayan bir kromozomun deterministtik düzeltme prosedürüne dayanarak uygun hale getirilmesi işlemidir. Yöntemin zayıf yönü probleme olan bağımlılıktır. Her bir farklı problem için, ayrı bir düzeltme algoritması tasarlanması gerekmektedir. Ayrıca bazı problem tipleri için uygun olmayan kromozomların uygun hale getirilmesi en az problemin kendisinin çözümü kadar karmaşık olabilmektedir.

3.3.3. Probleme Özgü Operatör Geliştirme

Uygunluk konusu ile ilgilenen bir diğer yaklaşım ise probleme özgü gösterimin bulunması ve kromozomların uygunluğunun sağlanmasında özelleştirilmiş genetik operatörlerin kullanılmasıdır. Birçok uygulayıcı probleme özgü gösterimleri ve özelleştirilmiş operatörleri başarılı genetik algoritmaları oluşturmakta kullanmaktadır. Fakat burada da probleme olan bağımlılık söz konusudur.

3.3.4. Cezalandırıcı Yöntem

Fazla sayıda kısıtın söz konusu olduğu problemlerde, uygun olmayan çözümler göreli olarak popülasyonun büyük bir bölümünde yer almaktadır. Bu gibi durumlarda, genetik arama uygun bölge ile sınırlandırılmazsa olası çözümlerin bulunması oldukça zaman almaktadır. Yöntemde, her nesilde uygun olmayan çözümler reddedilmez. Çünkü bunlardan bazıları optimal çözüm hakkında uygun kromozomlardan daha fazla bilgi vermektedir. Burada temel nokta, ceza değerinin doğru bir şekilde belirlenmesidir. Ceza değerinin belirlenmesi ise problem uzayının boyutuna göre değişiklik göstermektedir.

BÖLÜM 4

ÇİZELGELEME PROBLEMLERİ

Zaman çizelgeleme problemleri 45 yıldan fazladır yöneylem ve yapay zekâ araştırmacılarının dikkatini çekmektedir. Bu problemlerin çözüm uzayının geniş ve kompleks yapıda olması hesaplama karmaşıklığını artırmakta ve çözümünü güçleştirmektedir. NP hard olarak bilinen çizelgeleme problemleri araştırmacılara kombinatoryal çözüm metotlarını test etme imkânı sunmaktadır.

Çizelgeleme problemleri, hemşire nöbetleri, spor programı, uçak çizelgesi, araç çizelgesi, nakliyat çizelgesi ve üniversite çizelgesi gibi birçok farklı biçimde karşımıza çıkmaktadır. Daha farklı uygulama alanları Burke ve Ross (1996), Burke ve Carter (1998), Burke ve Erben (2001), Burke ve Causmaecker (2003) ve Burke ve Trick (2005) çalışmalarında görülmektedir. Bunların arasında gerçek yaşamda en fazla uygulaması yapılan problemler, eğitim bazlı olanlarıdır [Burke vd., 2007]. Eğitimsel çizelgelemenin türevleri, okul çizelgeleme (sınıf- öğretmen çizelgesi), ders programı, sınav programı ve fakülte çizelgesidir [Qu vd., 2009]. Üniversite çizelgeleme, görünürde aynı ama temelde çok farklı olan sınav ve ders programlarını ifade eden idari bir sorundur.

Genel olarak eğitimsel çizelgeleme problemi, kısıtlar kümesine tabi kalarak, olaylar kümesine (dersler, sınavlar) belirli zaman aralıklarının atanması işlemidir.

Kısıtlar genellikle iki tipte tanımlanır;

1. Zorunlu kısıtlar

   Her koşulda sağlanması gereken fiziksel kısıtlardır [Burke vd., 2007]. "Olmazsa olmaz" şeklinde de nitelendirilebilirler. Örneğin; sınav programı için düşünürsek, aynı zaman aralığında, aynı şubeye (öğrencilere) birden fazla sınav

yapılamaz. Zorunlu kısıtların sağlandığı çizelgeler, olası, uygulanabilir çözümler şeklinde ifade edilir [Burke vd., 2007].

2. Esnek kısıtlar

Mümkün olduğunca sağlanması gereken, kritik bir önemi olmayan kısıtlardır [Burke vd., 2007]. İsteğe bağlı özel isteklerin sağlandığı, olası çözümün kalitesini arttırmaya yönelik getirilen kısıtlardır denilebilir. Reel çizelgeleme problemlerinin karmaşıklığından dolayı esnek kısıtların bazıları ihlal edilmeden olası çözüm üretmek mümkün olmuyorsa bu kısıtlar göz ardı edilebilir. Esnek kısıtlar çoğunlukla, çözümlerin ne derece iyi olduğunu değerlendiren maliyet hesaplama fonksiyonu içerisinde kullanılır [Burke vd., 2007]. Örneğin; sınav programında, şubelerin aynı günde birden fazla sınavı olmamasına bunun mümkün olmadığı durumlarda ise en azından art arda zaman aralıklarında olmamasına dikkat edilmelidir.

Reel uygulamalarda tüm esnek kısıtları sağlayan uygulanabilir bir çözüm bulmak imkânsızdır [Qu vd., 2009]. Çünkü istemenin bir sınırı yoktur. Esnek kısıtların tipi ve önem derecesi kurumdan kuruma değişkenlik gösterir ve hatta bazı durumlarda aralarında çelişmektedir. Literatürde, sınav programlarında en çok kullanılan esnek kısıt, sınavların mümkün olduğunca zaman aralıklarına yayılarak yapılması, aradaki boş zamanlarda öğrencilere bir sonraki sınav için gözden geçirme süresi tanınmasıdır [Qu vd., 2009].

Kısıtların çeşitliliği, fazla olması ve bazılarının birbiriyle çelişmesine ek olarak farklı kurumlarda uygulanması, çizelgeleme problemlerinin karmaşıklığını ve zorluğunu arttırmaktadır [Qu vd., 2009].

Literatürde, yıllardır eğitimle ilgili çizelgeleme yöntemleri geniş bir yelpazede yer almıştır. Literatürü özetleyen çalışmalara baktığımızda;

Miles 1975'teki çalışmasıyla, bilgisayar tabanlı çizelgeleme çalışmalarının ilk bibliyografisini çıkartarak literatüre ışık tutmuştur [Miles, 1975]. Yine ilk çalışmalardan biri olan Schmidt ve Strohlein'in 1979'daki yayını, 200'den fazla referansıyla, bu tarihten önce yapılan tüm çalışmaları içermektedir [Schmidt ve Strohlein, 1979]. Kingston (1995), otomatik çizelgeleme üzerine 1000'den fazla referansıyla online bibliyografi hazırlamıştır [Kingston, 1995].

De Werra (1985), matematiksel modeller geliştirmiş, çizge renklendirme ve network akışı metotlarına dayanan ders programını ortaya koymuştur. 1997'de, çizelgelemede kısıtlı çizge renklendirme modelinin gerekliliklerini tanıtmış ve bazı matematiksel programlama formülasyonlarını analiz etmiştir [De Werra, 1985; 1997].

Carter 1986'da, farklı üniversitelerde uygulanabilecek genel bir sınav programı uygulaması üzerine çalışmıştır. Çizge sezgisel çeşitliliğini incelemiş ve hiçbir algoritmanın/paketin birden fazla kurumda uygulanmadığını vurgulamıştır. Sebebini, karşılaştırmaların yapılabileceği standart bir veri olmaması olarak göstermiştir [Carter, 1986].

Burke, Elliman ve Weare (1994), genetik algoritmaları kullanarak, üniversiteler için hem ders hem de sınav programlarının hazırlanmasına yönelik prototip bir sistem sunmuştur. Sınavlar dikkate alınarak yapılan uygulama aynı zamanda ders programı için de uygulanabilirdir. Sınav planlama sürecinin üniversite tarafından merkezi olarak yapılması gerektiğini vurgulamışlardır. Birden fazla alternatif çözüm üreten sistem, birtakım düzeltmelerle Edinburgh üniversitesinde kullanılmaya devam etmektedir [Burke, Elliman vd., 1994].

1996'da Carter ve Laporte, 1986-1996 yılları arasında yapılan çalışmalarda kullanılan algoritmik yaklaşımları özetleyen bir çalışma yapmışlardır. Tartışılan konu, metodun gerçel veriler üzerinde test edilmesi veya gerçek yaşamda uygulanmasıdır. Metotları dört tipte kategorize etmişlerdir; kümeleme yöntemleri, ardışık yöntemler, genelleştirilmiş arama (meta sezgisel) ve kısıt tabanlı teknikler [Carter, Laporte, 1996].

Burke ve diğerleri 1996 yılında sınav çizelgeleme üzerine bir anket hazırlamış, 75 Britanya üniversitesine göndermiş ve 56'sından cevap almıştır. Anket içeriği; problemin yapısını (büyüklüğü, kompleksliği, kısıtları vs.), nasıl çözüldüğünü ve amacını (iyi çözümleri hangi kriterlerin teşkil ettiği) içermektedir. Anket sonucundan elde edilen veriler, sınavlar, öğrenciler, bölümler, zaman aralıkları ve sınıflar için gerekli kısıtları elde etmek için analiz edilmiştir. Ankette listelenen 13 kısıta ek olarak, üniversitelerden gelen cevaplara (anket sonuçlarına) göre 19 kısıt daha eklenmiştir. Bu durum gerçek yaşamda, kurumlar arasında gereksinimlerin farklılaştığını

göstermektedir. Çalışma, reel çizelgeleme sorunlarına ışık tutmuş, otomatik çizelgeleme sisteminin pratikte uygulanabilecek özelliklerini sunmuştur [Burke vd., 1996].

Bardadym 1996'da, çizelgeleme için, bilgisayar destekli yönetim sistemlerinde farklı konuları ele almıştır. Daha çok problemler, gereksinimler, veri gösterimleri ve matematiksel modeller üzerinde durmuştur. 1960'tan 1990'a kadar geliştirilen çözüm yöntemlerinin ağırlıklı olarak sezgisel, meta-sezgisel sistemlere ve Karar Destek Sistemleri'ne entegrasyonu için algoritmik araçları kapsadığını, meta-sezgisel ve interaktif çizelgelemenin, bilgisayar destekli çizelgeleme sistemlerinin yeni bir akımı olduğunu belirtmiştir [Bardadym, 1996].

Wren (1996), Gezgin Satıcı Problemi üzerinde çalışarak, çizelgeleme, zamanlama ve atama arasında ilginç bir bağlantıyı tasvir etmiştir. Çizelgeleme ve personel atama arasındaki benzerliğin, problemlerin farklı türleri arasında başarılı bir çapraz-aşılama (cross-fertilization) gerçekleşmesine öncülük edebileceğini söylemiştir ve önceki çalışmaların da bunu desteklediği yönde açıklama yapmıştır [Wren, 1996].

Schaerf (1999) çalışmasında, okul, ders ve sınav programındaki formülizasyonlara göz atarak ders ve sınav programları arasında ayrım yapmanın zor olduğunu söylemiştir. Bu tür problemlerin varyantlarının tanımına dayanarak, yapay zekâ ile çözüm teknikleri sınıflandırılmıştır. Gelecekteki muhtemel çalışmaların, özel teknikler, standardizasyon, yaklaşımlar, güçlü bir kısıt bazlı dil tasarımı ve kombinasyonu ve farklı teknikleri karşılaştırma olabileceğini belirtmiştir [Schaerf, 1999].

Burke ve Petrovic 2002 ve 2004'te, üniversite (ders ve sınav) çizelgeleme probleminde, hibrid evrimsel algoritmalar, meta-sezgisel yaklaşımlar, çok kriterli yaklaşımlar ve adaptif yaklaşımları içeren bir çalışma yürütmüşlerdir. Böylelikle, sıralı, kümeleme, kısıt tabanlı teknikler ve meta-sezgisel yöntemler üzerine bir taslak sağlanmıştır. Gelecek çalışmaların, çizelgeleme sistemlerini genelleştirmeye yönelik bilgi tabanlı sistemlere ve yaklaşımlara ışık tutacağını vurgulamıştır [Burke ve Petrovic, 2002; 2004].

Burke, Kingston ve Werra (2004) makalesinde, sınıf-öğretmen, ders, sınav ve spor çizelgelerini değerlendirerek, çizge renklendirme metodunu kullanmışlardır.

Çalışma, son 40 yılın çizelgeleme literatüründe çizge renklendirme metodunun rolüne ışık tutmuş ve modern hibrid meta-sezgisel tekniklerin çizge renklendirmeyi içerdiğini göstermiştir [Burke vd., 2004].

Burke ve Landa Silva (2004), planlama ve zamanlama problemlerini çözmek için memetik algoritmaları kullanmıştır. Algoritmanın tasarımına ilişkin dikkate alınması gereken konular tartışılmıştır. Öz uyarlamalı memetik algoritmaların tasarımının çizelgeleme problemlerinin gelecekteki araştırma konusu olabileceğini belirtmiştir [Burke, Landa Silva, 2004].

Landa Silva, Burke ve Petrovic (2004), çok amaçlı (çok aşamalı yaklaşımlar ve çok kriterli evrimsel teknikleri içeren) ve meta-sezgisel tekniklerden bahsetmişlerdir. Bahsettikleri konular, yerel aramada kullanılan problem formülizasyonları, problem alanı bilgisi ve stratejileridir [Landa Silva vd., 2004].

Mc Collum (2007), çizelgeleme programının teorisi ile uygulaması arasındaki boşluğu kapatmak için, sınav ve ders programı ile ilgili birçok makaleyi incelemiştir. Kurumlarda çizelgelerin geliştirilmesinin çok aşamalı bir prosedür olduğunu belirtmiş ve sınav-ders programı için farklı reel model süreçlerini sunmuştur. Yapılan çalışma sınav ve ders programı zorluklarına da ışık tutmaktadır [Mc Collum, 2007].

Schaerf ve Di Gaspero (2007), üniversite çizelgelemede ölçülebilirliğe ve tekrarlanabilirliğe değinmiş, bu konuların önemini vurgulamış ve iki bakış açısının da pratikte çizelgeleme gelişimine katkıda bulunduğunu belirtmişlerdir [Schaerf vd., 2007].

Qu ve arkadaşları (2009), "decomposition", ayrıştırma fikrinin büyük kombinasyonel optimizasyon problemlerin çözümüne başarı ile uygulandığını vurgulamışlar ve çizelgeleme problemlerinde yaygın bir kullanımı olmadığını belirtmişlerdir. Sebebini, her hangi bir alt problemdeki atamanın, ilişkili olduğu diğer alt problemlerin çözümünde çakışmaya yol açması şeklinde açıklamışlardır. Çalışmada, sınav çizelgeleme problemi adaptif ayrışım metodu ile alt problemlere bölünmüş ve çözümleri için tamsayı programlama (IP-Integer programming) modeli geliştirilmiştir. Tüm çözümler içinde alt problemlerin optimal çözüme etkisi incelenmiş ve zor alt problemlerin çözüm kalitesinin, tüm problem için iyi çözümler oluşturmasına katkıda bulunduğu tespit edilmiştir [Qu vd., 2009].

Birbas ve arkadaşları (2009), okul çizelgelemede fizibilite için zorunlu kısıtlara ek olarak, öğretmenlerin tercihlerini karşılamak, her günün başlangıcına temel eğitim derslerini atamak ve öğretim-boş zaman toplamını dengelemeyi amaçlamışlardır [Birbas vd., 2009].

Burke, Qu ve Soghier (2012) çalışmalarında hiper-sezgisel yaklaşımı kullanmışlardır. Esnek kısıtların ihlal edildiği sınavları, daha iyi bir çizelge elde etmek için sıralamakta ve yeniden planlamaktadırlar. Sezgisel hareket yeniden planlanan sınavları kullanarak, üretilen çözümün kalitesini etkileyen çizelgelemeyi geliştirmeyi amaçlar. Hibrid hiper-sezgisel yaklaşımın farklı kombinasyonlarının testinden sonra, Kempe zinciri sezgisel hareketi ve zaman aralığı sezgisel değişiminin, hibridizasyonda kullanılan en iyi sezgisel metot olduğu kanıtlanmıştır. Kempe zinciri, dört renk teoremi çalışmasında ağırlıklı olarak kullanılan matematiksel bir araçtır. Benzer şekilde, doygunluk derecesi kullanılarak sınavların sıralandığını, en büyük ağırlık derecesini kullanarak da ilişkileri kırdığını ve iyi sonuçlar ürettiğini göstermiştir. Bu gözlemlere dayanarak, geliştirilen metodoloji, Kempe zinciri hareketini ve zaman aralığı sezgisel değişimini iki aşamada hibridize eder. İlk aşamada, rastgele sezgisel sıralamalar oluşturulur ve otomatik olarak analiz edilir, iyi sıralamadakiler sabitlenip, geri kalanlar boş tutulana kadar sezgisellik devam ettirilir. İkinci aşamada, en iyi sezgisel sıralamayı bulmak için sıralamalar, boş pozisyonlara sezgisellerin rastgele atanması ile oluşturulur. Son olarak, oluşturulan sıralamalar probleme uygulanır. Düşük seviyedeki sezgisellerin geliştirilmesine yönelik hiper-sezgisel yaklaşım, iki farklı veri setine uygulanmış ve son dönem modern yaklaşımlarla benzer sonuçlar elde etmiştir [Burke vd., 2012].

Uluslararası literatürde yapılan çalışmalar genel olarak yukarıdaki gibidir. Türkiye'de yapılan çalışmalara göz attığımızda ise daha çok çizelgeleme problemlerinden ders programının hazırlanması sorununun üzerinde durulduğu gözlemlenmektedir. Sınav programı üzerine yapılan çalışmalar ise daha kısıtlı kalmıştır. Yapılan çalışmalarda farklı metotlar kullanılmış ve gerektiğinde problemin yapısına göre uyarlanmıştır. Tüm bu çalışmaları özetleyecek olursak;

Bağış (1996), ders çizelgelemenin, önceden belirlenmiş nesnelerin belirli konumlara en uygun şekilde yerleştirme problemi olmasından dolayı ayrık bir optimizasyon problemi olarak sınıflandırmıştır. Bu tip problemlerin çözümünde sezgisel

algoritmaların kullanımına dikkat çekmiştir. Genetik algoritmaları kullanarak geliştirdiği yazılım 4 bilgi dosyasından yararlanmaktadır. Erciyes Üniversitesi Mühendislik Fakültesi Elektronik bölümü için geliştirilen uygulamanın bilgi dosyalarında yapılacak olan değişiklikle her hangi bir eğitim kurumuna uygulanabileceğini belirtmiştir [Bağış, 1996].

Daban ve Özdemir (2004) çalışmasında, Genetik Algoritma kullanarak, öğretim elemanları ve öğrencilerin ihtiyaçlarına göre daha etkin bir eğitim-öğretim sağlamak için ders programı hazırlayan ve optimize eden bir program geliştirmişlerdir. Ders programının uygunluğu, öğretim elemanı ve eğitim-öğretim verimliliği kriterlerine göre belirlenmektedir. Pedagojik esaslara göre, öğrencilerin algısının haftanın günlerine ve gün içerisindeki saatlere göre değişimi göz önünde bulundurularak verimliliklerinin artması amaçlanmıştır. Diğer çalışmalardan farklı olarak, başlangıç popülasyonu, çakışmalar önlenerek hazırlanan, kullanılabilir ders çizelgelerinden oluşturulmuştur. Ayrıca algoritmada standart GA operatörlerinden farklı olarak, probleme özgü çaprazlama ve mutasyon operatörleri geliştirilmiştir. Yapılan simülasyon sonuçlarında, başlangıca göre %40-60 oranlarında iyileşme sağlanmıştır [Daban ve Özdemir, 2004].

Daban ve Özdemir 2005'teki çalışmalarında, 2004'teki çalışmalarına ek olarak uygunluk fonksiyonuna ders ağırlığı kısıtını ekleyerek, öğrenci sayısı fazla olan derslerin, pedagojik esaslara göre verimliliğin fazla olduğu gün ve saatlere yerleştirilmesini amaçlamışlardır. Ders ağırlıkları dikkate alınarak hazırlanan ders programlarında başlangıca göre %35-40 oranında iyileşme sağlanmıştır [Daban ve Özdemir, 2005].

Atanak ve Hocaoğlu (2005), GA'ları kullanarak sınıfların boyutunu, ders sayılarını, derse kaydolması beklenen öğrenci sayılarını ve dersi verecek olan öğretim üyesini girdi olarak alan, verilen sınırlamalar altında dönemlik ders programı hazırlayan bir otomasyon yazılımı geliştirmişlerdir. Geliştirilen yazılım, öğretim üyelerinin derslerini vermek için tercih edecekleri zamanları da girdi olarak alabilmektedir. Yazılım, olası çözümler içinde en iyisini aramakta ve optimale yakın çözümler üretmektedir [Atanak ve Hocaoğlu, 2005].

Çayıroğlu ve Dizdar (2006), uzman sistem yaklaşımıyla gerçekleştirdikleri "Online Ders Yerleştirme Programı" (ODYP) yazılımı ile, haftalık ders programlarının oluşturulması, öğretim elemanı görevlendirmelerinin yapılması, bölüm dersliklerin programlarının hazırlanması ve ek ders formlarının çıkarılması gibi işleri sadece tek bir bölüm için değil, fakültenin tüm bölümleri için gerçekleştirmektedirler. Geliştirilen program Z.K.Ü. Karabük Teknik Eğitim Fakültesi'nde uygulanmış ve iki ayda yapılan işlemleri bir hafta gibi kısa bir süreye indirmiştir. Çalışmanın bundan sonraki aşaması olarak, çizge metotları ya da genetik algoritmaların denenebileceği ve bu uygulamalarında iyi sonuçlar vereceğinin tahmin edildiği söylenmiştir [Çayıroğlu ve Dizdar, 2006].

Gülcü (2006), tez çalışmasında ders çizelgesi problemine genetik algoritma ve tabu arama sezgisel metotlarını kullanarak çözüm aramaya çalışmış ve sonuçlarını karşılaştırmıştır. Çalışmada her iki algoritmanın performansı, her bir algoritma ile tanımlanmış en iyi çözümün kalitesi, başlangıç çözümünden durma kriterlerinin sağlanmasına kadar araştırmanın izlediği seyir ve en iyi çözümü bulmanın maliyeti kriterlerine göre karşılaştırılmıştır. Elde edilen sonuçlar zaman maliyeti açısından GA'ların daha kısa sürede çözüm bulduğunu, tabu aramanın ise aynı çözüme daha uzun sürede ulaştığını göstermiştir. Ancak algoritmaların uygulamaya geçirilmesi açısından genetik algoritmaların daha fazla emek gerektirdiği belirtilmiştir [Gülcü, 2006].

Taç (2006), ders çizelgelemesi problemine genetik algoritmaları kullanarak hazırladığı "Okul Ders Programı" (ODEP) yazılımı ile çözüm bulmaya çalışmıştır. Yazılımın ara yüzü aracılığıyla genetik algoritmaların parametrelerini, ders, dershane, sınıf ve öğretmen bilgileri ile tüm kısıtları tanımlamaktadır. Gen değerinin ifade edilmesinde kötü gen işaretçisi kullanarak, gen değerinin kısıtlara uyup uymadığını kontrol etmektedir. Problemin yapısını maksimizasyon olarak belirlemiş ve hiçbir çakışmanın gözlenmediği durumlarda en iyi uygunluk değerini sabit bir sayıyla ifade etmiştir. Uygulama, Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik-Haberleşme ve Bilgisayar Kontrol bölümleri 2005-2006 yılı bahar dönemi verileri ile test edilmiş ve uygulanabilir ders programları elde edilmiştir [Taç, 2006].

Harmanşah ve arkadaşları (2011), haftalık ders programı hazırlık sürecinin hızlı, verimli ve düşük maliyette gerçekleştirilmesinin eğitim kurumları için önemine

değinmişlerdir. Ders programı hazırlığının web teknolojileri kullanılarak gerçekleştirilmesinin karmaşık ve maliyetli bir sürecin iyileştirilmesini sağlayacağını ve böylece eğitim kurumlarındaki her yarıyılın başlangıcında yaşanan zaman kayıpları ve aşırı dokümantasyondan kaynaklanan maliyetlerin azaltılacağını belirtmişlerdir. Bu çalışmada; Ege Meslek Yüksekokulu haftalık ders programı hazırlığı web tabanlı bir sistem üzerinden gerçekleştirilmiştir. WebDerslik sisteminde, meslek yüksekokulundaki programlar, öğretim elemanları, dersler, derslikler ve laboratuvarlar giriş verisi olarak değerlendirilmektedir. Çıktı olarak her programa ait ders programı üretilmektedir. WebDerslik sistemine tüm kullanıcılar aynı arayüzü kullanarak giriş yapmaktadır. Sistemde "Admin" ve "Koordinatör" olmak üzere iki ayrı tip kullanıcı tanımlanmıştır. Admin paneli kullanılarak Ege Meslek Yüksekokulundaki programlar, öğretim elemanları, derslik ve laboratuvar bilgileri sistem veri tabanına kayıt edilmektedir. Koordinatör olarak adlandırılan kullanıcılar kendileri için hazırlanan giriş panelini kullanarak programlarına ait ders programlarını hazırlamaktadırlar [Harmanşah vd., 2011].

İlkuçar ve arkadaşları (2011), veri tabanı destekli bir yazılım geliştirerek, üniversitede okutulan derslerin vize ve final sınavlarının hangi oturumlarda yapılacağı, salonlara yerleştirilmesi, kimlerin hangi oturumlarda görev alacağı, bazı derslerin özel durumları, gözetmenlerin ve ders yürütücülerinin özel istekleri, salonların durumu, sınav sürelerinin farklı olması, aynı dersi alan öğrencilerin aynı oturumda sınava girmesi vb. gibi onlarca farklı kriter göz önüne alınarak sınavların oturumlara ve salonlara yerleştirilmesini sağlamıştır. Geliştirilen yazılım, Mehmet Akif Ersoy Üniversitesi Meslek Yüksekokulunda 2010-2011 öğretim yılı güz yarıyılı vize ve final sınavlarının yerleşiminde kullanılmıştır. Program, kısıtları dikkate alarak, T-SQL sorguları yardımıyla yerleştirme yapmıştır [İlkuçar vd., 2011].

Kalaycı ve Güngör (2012) çalışmalarında S.A. MirHassani'nin (2006) çalışmasından esinlenmişlerdir. S.A. MirHassani çalışmasında, sınav çizelgeleme problemlerinin çözümlerinde istenen özelliklerden birinin zaman boşluğu (paperspread) maksimizasyonu olduğunu belirtmiştir. Paperspread, her öğrencinin sınavlar arasında sahip olduğu çalışma süresi miktarının bir ölçüsüdür. Çalışma daha önce tanımlanmış sınav programının, sınavlar arasındaki çalışma süresini maksimize etmek için modifiye

edilmesi üzerinedir. Amaca ulaşmak için tamsayı programlamadan yararlanılmış ve model reel bir uygulamada test edilmiştir [MirHassani, 2006]. Kalaycı ve Güngör, bu çalışmayı, sınavların zorluğu ile öğrencilerin sınavlar arasında ihtiyaç duyduğu gerekli çalışma süresinin pozitif ilişkili olduğu fikrini içerecek şekilde genişletmişlerdir. Uygulama, sınav programının sadece sabit kısıtlar baz alınarak değil, öğrenci başarısını da göz önüne alarak, genetik algoritmalar ile programlanmasına dayanır. Model, "öğrenci başarısının, sınavlar arasındaki yeterli hazırlık ve dinlenme süresi ile pozitif ilişkili" olduğu fikrini içermektedir. Bu nedenle çalışmanın temel amacı, sınav zorlukları dikkate alınarak sınavlar arasındaki zaman uzunluğunun maksimize edilmesidir. Optimizasyon için iki farklı genetik algoritma modeli geliştirilmiştir. İlk modelde, kuşaklar boyunca olanaksız çözümleri ortadan kaldırmak için yüksek ceza yaklaşımı kullanılmıştır. İkinci genetik algoritma modeli ise popülasyona katılan her kromozomun kısıtlamaları karşılayıp karşılamadığını kontrol etmektedir. Modelleri değerlendirmek için deney seti Pamukkale Üniversitesi Mühendislik Fakültesi verilerinden elde edilmiştir [Kalaycı ve Güngör, 2012].

Çizelgeleme üzerine yapılan tüm bu spesifik metotlara ilişkin yorumlar, araştırmaların nasıl hızla geliştiğini göstermektedir.

BÖLÜM 5

KARAR DESTEK SİSTEMLERİNİN DURUM ÇALIŞMASINA UYGULANMASI (GA_SYP)

Üniversitelerde zaman çizelgelemesi, dersleri veya sınavları, zaman aralıklarına ve dersliklere atama problemidir. Literatür çalışmasına bakıldığında, her iki durum için de birçok çalışma yapılmıştır. Fakat Türkiye'de daha çok ders programlarının hazırlanması üzerine durulmuş, sınav programı ile yapılan çalışmalar ise daha kısıtlı kalmıştır.

Akademik kurumlarda sınav programı hazırlanması, üniversitelerin sınavlarını yıllık veya dönemsel yapmasına göre değişen sıklıkta karşımıza çıkan bir iş yüküdür. Çoğunlukla bir önceki senenin programının üzerinde küçük değişiklikler yapılmasıyla, manuel olarak ya da basit bir yönetim sisteminin yardımı ile hazırlanır. Giderek artan bölüm sayısı, öğrenci sayısının artmasına ve dersliklerin yetersiz kalmasına neden olmaktadır. Bu durum, çizelgelemenin hesaplama zorluğunu arttırmakta, bir önceki uygulanan programı yetersiz kılmakta ve hazırlayan kişi için zor bir problem halini almaktadır.

Türkiye'deki üniversitelerin birçoğunda merkezi bir sınav sistemi uygulanmamaktadır. Fakültelerin kendi içlerindeki her bir bölümün sınav programları birbirlerinden bağımsız olarak yapılmaktadır. Personelin mevcut iş yükünden sınav çizelgesi problemini azaltmak, hata yapma olasılığını minimuma düşürmek ve alternatif çözümler sunabilmek için "Genetik Algoritma ile Sınav Yerleştirme Programı (GA_SYP)" geliştirilmiş ve Trakya Üniversitesi Mimarlık-Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünün verileri kullanılarak test edilmiştir. Fakültede 2012 Haziran itibariyle Makine Mühendisliği, Mimarlık, Bilgisayar Mühendisliği, Gıda Mühendisliği, Elektrik-Elektronik Mühendisliği ve Peyzaj Mimarlığı bölümleri yer

almaktadır. Mimarlık ve Peyzaj Mimarlığı bölümleri üniversitenin farklı bir kampüsünde bulunmaktadır. Her bir bölümün öğrenci sayıları sırasıyla Makine Mühendisliği normal öğretim 413 kişi, ikinci öğretim 352 kişi, Bilgisayar Mühendisliği normal öğretim 327 kişi, ikinci öğretim 289 kişi, Gıda Mühendisliği normal öğretim 155 kişi olmak üzere normal öğretimdeki toplam öğrenci sayısı 895, ikinci öğretimdeki toplam öğrenci sayısı ise 641'dir. Elektrik-Elektronik Mühendisliğinin henüz öğrencisi yoktur. Mühendislik bölümlerinin eğitim-öğrenim faaliyetlerini yürüttüğü binada toplam 20 derslik, 1 amfi ve 19 tane laboratuvar bulunmaktadır. Bu dersliklerin ve laboratuvarların bir kısmı (4 derslik, 8 laboratuvar) yeterli derslik kapasitesi bulunmadığı için Mimarlık bölümünün farklı bir binaya taşınması sonrasında inşa edilmiştir. Test edilen veriler ise 2012 Bahar dönemi final sınavları için geçerlidir. Laboratuvar derslerinin sınavları uygulamalı olarak değil, yazılı sınav şeklinde yapıldığı için laboratuvarlar sınav salonu olarak kullanılmamaktadır. Her bölüme eğitimlerini yürütebilecekleri belli derslikler tahsis edilmiştir. Sınav çizelgesi hazırlanırken bölümler öncelikle kendilerine tahsis edilen derslikleri kullanmakta, yeterli olmadığı durumlarda diğer derslikleri değerlendirmektedir. Bilgisayar Mühendisliğine tahsis edilen derslikler, 72 kişilik kapasitesi ile D102, D103, D104, D105, D106, 45 kişilik kapasitesi ile L202 ve 120 kişilik kapasitesi ile amfidir. Lakin kopya çekilmesine sebebiyet vermemek için D102, D103, D104, D105, D106 dersliklerinde her üçlü sıraya tek öğrenci yerleştirildiğinde derslik kapasitesi 24 kişiye düşmektedir. Aynı durumda L202 dersliğinin kapasitesi 30'a, amfinin ise 56'ya inmektedir. Derslik yeterliliğinin sağlanamadığı durumlarda ise ön sıralara 2 kişi oturtularak D102-D106 dersliklerinde kapasite 27 kişi olarak belirlenmektedir. Sınavların uygulanması sırasında her derslikte tek gözetmen görevlendirilirken, amfi salonunda fazla sayıda öğrenci olmasından dolayı burada iki gözetmen bulunmaktadır. Sınavlarda görevlendirilebilecek gözetmen sayısı ise 7'dir.

Uygulamada ulaşılması istenen sınav çizelgeleme probleminin çözümü için sağlanması gereken kısıtlar söz konusudur. Kısıt kümeleri zorunlu kısıtlar ve esnek kısıtlar olmak üzere sınıflandırılmıştır. Zorunlu kısıtlar elde edilen çözüm adaylarının olası çözüm olup olmadıklarını belirlerken, esnek kısıtlar olası çözümün kalitesini belirlemektedir. Örneğin, aynı günde bir şubenin 4 sınavının birden olması olası bir durumdur fakat bu durum öğrencinin başarısını olumsuz etkileyeceği için etkin bir

çözüm değildir. Bu durumda amaç, zorunlu kısıtların tümünü sağlayan, esnek kısıtlarınsa sağlanma oranıyla doğrudan ilişkili olan kaliteli bir çözüm bulmak ise, probleme özgü zorunlu ve esnek kısıtları en başta tanımlamak gerekir.

Zorunlu kısıtlar

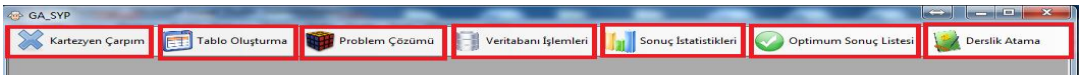
1. Aynı zaman diliminde öğrencilerin birden fazla sınavı olmaması (sınav çakışması).
2. Sınavların yapılması için yeterli kaynakların sağlanması. Derslik kapasitesinin yeterli olması gibi.
3. Tüm sınavların belirli zaman aralıklarına yerleşmiş olması.
4. İkinci öğretim ile normal öğretim sınav programlarının ayrı ayrı yapılması.
5. Bahar, Güz ve Yaz dönemi derslerinin ayrı ayrı sınıflandırılması ve her dönem için o döneme ait sınav programının hazırlanması.
6. Her hangi bir sınava ayrılan derslik sayısının gözetmen sayısından fazla olmaması.

Esnek kısıtlar

1. Aynı gün içerisinde öğrencilerin birden fazla sınavı olmaması, mümkün değilse de arada zaman bırakılarak yapılması.
2. Özellikle dışarıdan gelen öğretim üyeleri için sınavlarının isteğe göre herhangi bir günün herhangi bir saatine yerleştirilmesi.
3. Vize sınav döneminde derslerin devam etmesi yeterli dersliğin sağlanamamasına sebep olabilir. Bu durumda hafta sonunda da sınav yapılmasına olanak tanınması.
4. Sınavların, amfi ve laboratuvar gibi tercih edilmeyen dersliklere zorunlu kalınmadığı sürece yerleştirilmemesi.
5. Toplantı vs. gibi durumlar için belirli bir güne veya belirli bir günün herhangi bir saatine hiçbir sınavın yerleştirilmemesi.
6. Süreklilik açısından her bir gündeki zaman aralıklarının blok olarak tanımlanması.

Sınav programının hazırlanması için geliştirilen programın arayüzünde, Şekil 5.1'e bakıldığında, kullanıcıya 7 farklı menü sunulmaktadır. Kısaca açıklarsak,

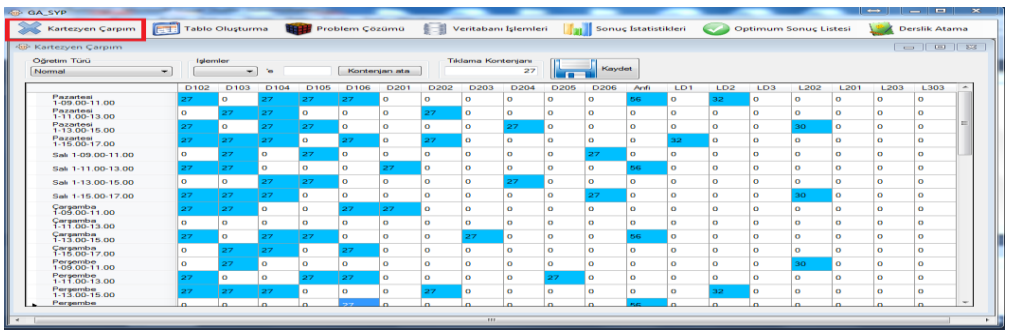
kartezyen çarpımında tüm zaman aralıkları için uygun olan derslikler belirlenmekte, tablo oluşturmada belirli kısıtlar değerlendirilerek gen havuzu oluşturulmakta, problem çözümünde genetik işlemler yürütülmekte ve sonuçları alınmaktadır. Veri tabanı işlemlerinde problemin çözüm sürecinde gereksinim duyulan verilere erişilmesi ile ilgili işlemler bulunmakta, sonuç istatistikleri kısmında ise her bir parametreye göre elde edilen çözümlerin değerlendirilmesi yer almaktadır. Optimum sonuç listesinde, elde edilen sonuç dosyalarının en uygun çözümleri karşılaştırılmak için ayrı bir dosyaya sıralı olarak yazdırılmakta ve aynı zamanda sonuçların gen dizilimi ve uygunluk değerleri görüntülenmektedir. Son olarak, belirli zaman aralıklarına atanan sınavların, hangi dersliklerde yapılacağına karar verme işlemi derslik atama menüsünde yapılmaktadır.



Şekil 5.1. Ana ekran

5.1. Kartezyen Çarpım Menüsü

Fakültede giderek artan bölüm sayısı, öğrenci sayısının artmasına ve haliyle her bir bölüm için sınav sayısının artmasına sebep olmaktadır. Yıllık sınav takvimi fakülte bazlı olduğu için, her bölümün sınavları aynı tarihlerde yapılmaktadır. Bu durum, derslik sayılarını yetersiz kılmakta, tüm dersliklerin bölümler arasında paylaşılmasını gerektirmektedir. Derslik paylaşımı yapılırken, her bölümün kendine ait derslikleri ilgili bölümün sınavı için önceliklidir. Şekil 5.2'de sütun bilgilerine baktığımızda, derslik bilgileri yer almaktadır ve ilk sıradaki derslikler (D102, D103, .., D106) Bilgisayar Mühendisliği Bölümü'ne ait olan dersliklerdir.

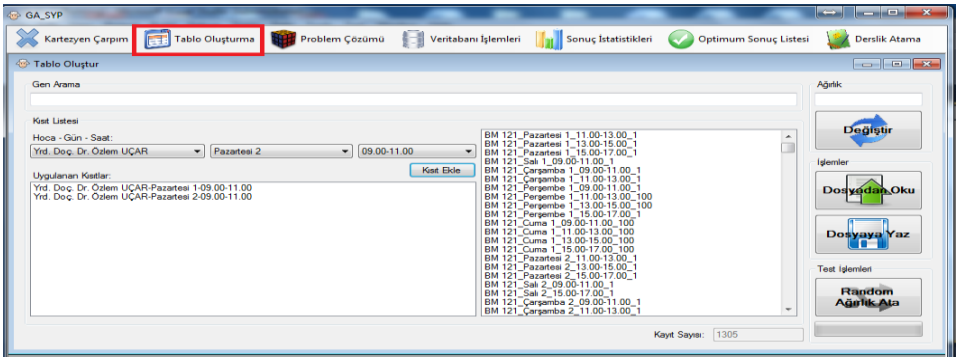


Şekil 5.2. Kartezyen Çarpım Aşaması

Satır değerlerinde ise zaman aralıkları yer almaktadır. Zaman aralıkları, haftanın 7 günü için de düşünülmüştür. İş günlerinde yeterli derslik kapasitesinin sağlandığı bir zaman aralığının bulunamadığı durumlarda, tatil günlerinde de sınav yapılabileceği ihtimali göz önünde bulundurulmuştur. Sınavlar 2 haftalık zaman dilimine yayıldığı için gün bilgileri ilk hafta için Pazartesi1, Salı1, …, Pazar1, ikinci hafta için Pazartesi2, Salı2, …, Pazar2 şeklinde ifade edilmiştir. Zaman aralıkları öğretim türünün tipine göre değişiklik gösterir. Gün içerisinde normal öğretim için düşünüldüğünde sınavlar 2'si gündüz, 2'si öğleden sonra olmak üzere 4 oturumda yapılmaktadır (9.00- 11.00, 11.00- 13.00, 13.00- 15.00, 15.00- 17.00). İkinci öğretim için ise oturumların zaman aralıkları (17.00- 19.00, 19.00- 21.00, 21.00- 23.00) şeklindedir. Kartezyen çarpım tablosu, hangi günün hangi zaman aralığında, hangi dersliklerin kullanılabilir olduğunu ifade eder. Kullanıcı başlangıçta gün içerisindeki zaman aralıklarında hangi derslikler uygun ise, bu dersliklerin kapasite bilgisini o günün o saati ile derslik bilgisinin çakıştığı hücreye girer. Bu verileri tüm tablo için doldurur ve kaydeder. Amaç, herhangi bir sınavın atanacağı zaman aralığında yeterli derslik kapasitesinin olup olmadığını kontrol etmektir.

5.2. Tablo Oluşturma Menüsü

Başlangıç popülasyonunun oluşturulması için rastgele seçilen kromozomlar yine rastgele seçilen genlerden elde edilir. Uygulamada genlerin seçileceği gen havuzunun büyüklüğü bir dersin sınavının 40 farklı zaman aralığına gelme olasılığı ve toplam ders sayısı için düşünüldüğünde oldukça fazladır. Bu nedenle gen havuzunu kısıtlama gereksinimi duyulmuştur. Şekil 5.3'te görüldüğü gibi Tablo Oluştur menüsü bize kısıtlanmış gen havuzunun verilerini ağırlık değerleri ile birlikte getirmektedir.



Şekil 5.3. Tablo Oluşturma

Gen havuzunda, sınavların yapılması için yeterli derslik kapasitesinin olmadığı ve az kontenjanlı sınavların çok fazla derslik alanı olduğu zamanlara atandığı durumlar mevcuttur. Söz konusu durum çözüme ulaşmamızı zorlaştırmakta ve çözüm süresini uzatmaktadır. Bu nedenle, başlangıç popülasyonunu oluştururken, sınavların yeterli dersliğin olmadığı zamana atanması kesinlikle istenmeyen bir durum olduğu için gen değerleri "Asla Olmaz" cezası ile cezalandırılır. Az sayıda öğrencinin girdiği sınavların, fazla sayıda boş dersliğin olduğu zamana atanması ise daha fazla derslik kapasitesine ihtiyaç duyan sınavların açıkta kalmasına sebep olabileceğinden tercih edilmeyen bir durumdur. Ölçüm, derslik kapasitesi ile sınava girecek öğrencilerin sayısı arasındaki fark alınarak yapılmaktadır. Eğer fark 50'den fazla ise cezalandırma işlemi yapılır. Ceza değeri daha düşük tutulmuştur. Çünkü farkın 50'den fazla olması, yeterli dersliğin olması durumuna göre kabul edilebilirdir. (Sorun oluşturmadığı durumlarda bu gen değerlerinin oluşturduğu kromozom da olası çözümü ifade edecektir.).

"Gen arama" eklentisi, isteğe bağlı kısıtlamalar için kullanıcıya kolaylık sağlamaktadır. Örneğin, dışarıdan gelerek ders veren bir öğretim elemanı, sınavının belirli bir günde yapılmasını istediğinde, verdiği dersin kodunu aratarak, o dersin sınavının yerleştirildiği tüm ihtimaller getirilir ve istenen günün dışındaki günler "**Değiştir"** butonu kullanılarak ağırlık değerlerine ceza miktarı (değeri) eklenir. Özel durumlar için cezalandırılan genlerin başlangıç popülasyonunda bulunma olasılığı vardır. Fakat uygunluk fonksiyonundan elde edilen değer yüksek çıkacağı için elenirler. Tablo Oluştur listesindeki tüm bilgiler, Problem Çözümü menüsünde kullanılmak üzere "**Dosyaya Yaz"** butonu ile dosyaya yazdırılır.

"Kısıt listesi" ile esnek kısıtların bir kısmının sağlanması amaçlanmıştır. Belli zaman aralıklarına belirli sınavların yerleştirilmesi veya bu zaman aralıklarında hiçbir sınavın yapılmaması gibi durumlar istenebilir. Örneğin, fakültede düzenli olarak her çarşamba sabahı saat 10.00-12.00 arası idarecilerin katılım gösterdiği bir toplantı yapılmaktadır. Bu durumda bölüm başkanı ve yardımcılarının verdikleri derslerin sınavlarının çarşamba günü belirtilen zaman aralıklarına yerleştirilmemesi gerekir. Uygulamada istenilen kısıtlama için hoca-gün-saat kartezyen çarpımı elde edilmekte ve seçilen öğretim elemanının verdiği tüm derslerin seçilen gün ve saate yerleşme durumları, veri tabanından çekilen "istenmeyen durum" ceza değeri ile

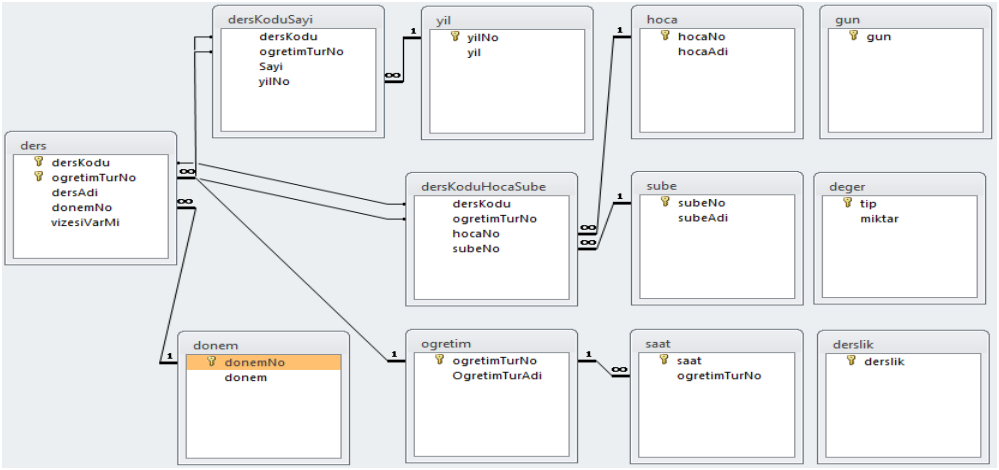
cezalandırılmaktadır. İstenilirse bu durumlar gen havuzuna dâhil edilmeyebilir fakat bu durumda algoritmanın çözüm bulma süreci uzayabilir ya da çok fazla kısıt uygulandığında çözüm bulamama durumu söz konusu olabilir. Gen arama eklentisinde de benzer işlemler yapılabilir. Ancak fazla sayıda kısıt göz önüne alındığında kullanıcının tek tek ağırlık değerlerini değiştirmesi gerekmektedir. Ağırlık değerlerinin belirlenmesi ve yapılması gereken değişim kullanıcının hata yapmasına da neden olabilir.

Elle hazırlanan programlarda, daha çok önceki senelerin programlarından yararlanıldığından söz etmiştik. Çok fazla değişikliğin olmadığı durumlarda önceki taslaktan yararlanılabilir. Burada da, eğer her bir zaman aralığındaki derslik kapasitesi bir sonraki sınav(lar)da değişiklik göstermiyorsa kullanıcının her seferinde Kartezyen Çarpım menüsündeki gridi doldurmasına gerek kalmaksızın, "**Dosyadan Oku**" seçeneği ile bir önceki kaydedilmiş dosyayı kullanmasına imkân tanılır.

Test İşlemleri adı altında "**Random Ağırlık Ata**" eklentisi, elimizde reel verilerin olmadığı durumlarda programı test etmek amacıyla farklı ağırlık değerlerine sahip gen havuzundan seçim yapmamızı sağlamıştır.

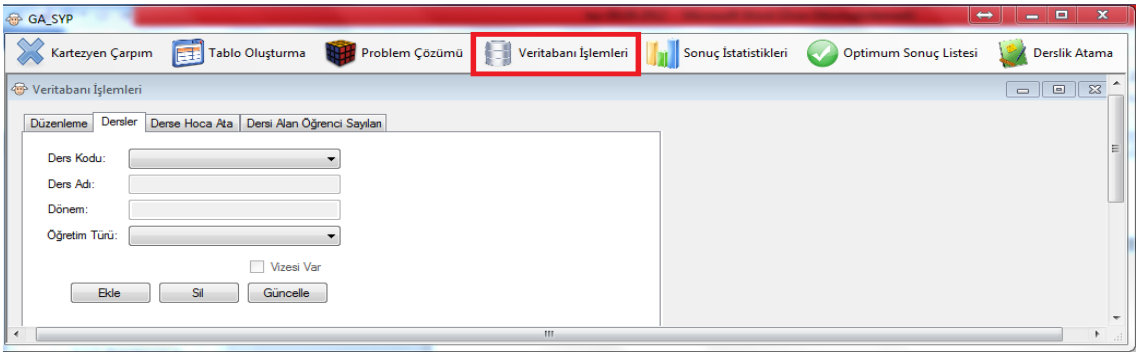
5.3. Veri Tabanı İşlemleri

Geliştirilen uygulamada kullanılan veri tabanı Access veri tabanı yönetim sistemi ile geliştirilmiştir. Kullanılan tüm tablolar ve aralarındaki ilişki Şekil 5.4'te görüldüğü gibidir.



Şekil 5.4. Veri tabanı tabloları ve ilişkileri

ders tablosu: Tabloda dersKodu, ogretimTurNo, dersadi, donemNo ve vizesiVarMi alanları bulunur. dersKodu alanı dersin sistemde geçen kod bilgisini (örn. BM112) saklarken, dersAdi alanı bu kodun karşılığındaki dersin adını, uygulamada takibin kolay hale getirilmesi için saklar. Şekil 5.5'te bu alanların verilerinin girildiği ara yüz verilmiştir.



Şekil 5.5. Veri tabanı işlemleri ders bilgileri arayüzü

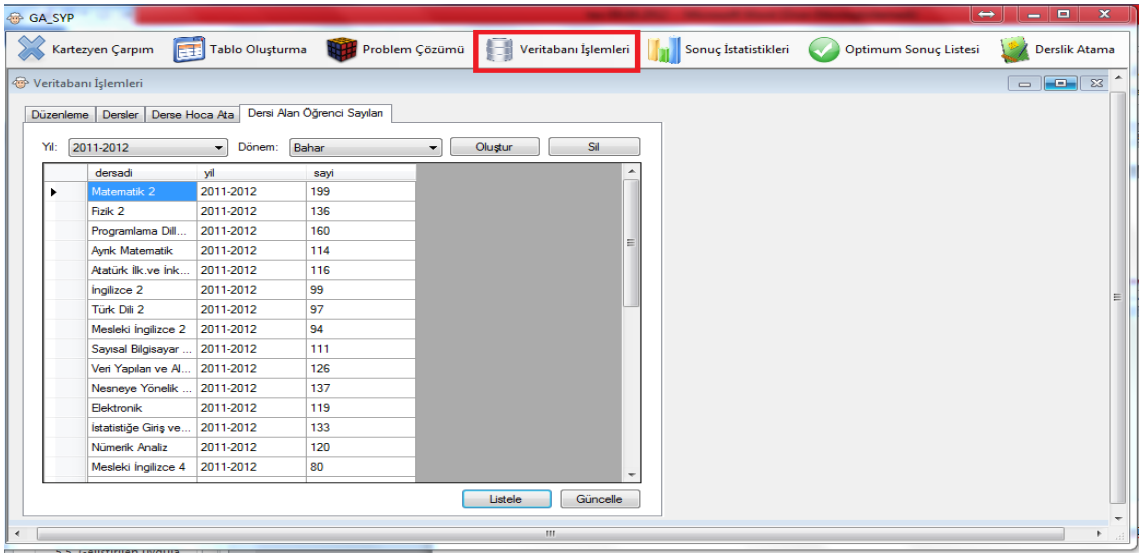
ogretimTurNo, ogretim tablosuyla ilişkili olup, dersin Normal Öğretim veya İkinci Öğretim dersi olduğunu belirtir. Bunun nedeni Normal Öğretim ve İkinci Öğretimde dersi alan öğrenci sayılarının ve zaman aralıklarının farklı olmasıdır. vizesiVarMi alanında ise dersin vizesinin olup olmadığı saklanır. Böylece uygulama vize sınav programı probleminin çözümü sırasında vizesi olmayan dersleri yerleştirme işlemi yapmaz. donemNo alanında ise Güz, Bahar ve Yaz dönemi olmak üzere üç farklı veriye sahip donem tablosu arasında ilişki kurulmuştur.

ogretim tablosu: Tablo, öğretim türünün Normal Öğretim veya İkinci öğretim olduğunu belirten OgretimTurAdi alanını ve bunlara ait id'leri tutan ogretimTurNo alanını içerir. ders tablosuyla ilişkilidir.

donem tablosu: Tabloda donemNo ve donem alanları bulunur. donem alanında her bir dönemin, Güz, Bahar ve Yaz olmak üzere dönem bilgileri yer alırken donemNo dönem isimlerine ait id'leri saklar.

dersKoduSayi tablosu: dersKoduSayi tablosunda, her derse karşılık gelen öğrenci sayıları saklanır. Böylelikle tablo oluşturma aşaması sırasında sınıf kapasitesinin yeterli olup olmaması veya sınıf kapasitesinin öğrenci sayısından çok fazla (>50) olup olmama durumları kontrol edilebilir. Bu durumlar problem çözümü

sırasında elenmek üzere cezaya tabi tutulur. Ayrıca her öğretim türü ve eğitim öğretim yılına göre öğrenci sayıları değişiklik gösterdiği için, Şekil 5.6'da görüldüğü gibi bu alanlar da tabloya eklenmiştir.



Şekil 5.6. Veri tabanı işlemleri dersi alan öğrenci sayıları bilgisi arayüzü

yil, gun ,saat ve derslik tabloları: yil, gun, saat ve derslik tabloları daha önce belirlenmiş olan nitelikte verileri içermektedir. (örn. 12.00-14.00 gibi). yil tablosunda eğitim öğretim yılı (örn. 2011-2012) listeleri bulunurken, gun, derslik ve saat tablolarında sınav dönemi için kullanılan günler saat aralıkları ve derslik isimleri saklanmaktadır.

dersKoduHocaSube tablosu: dersKoduHocaSube tablosunda hangi dersi hangi hocanın verdiği saklanmaktadır. Farklı öğretimlere ve farklı şubelere aynı dersin açılması durumunda bu ders içeriği aynı olduğu halde farklı öğretim üyeleri tarafından verilebilmektedir. Bu yüzden tabloda bu durumları ayırt edebilmek için diğer alanlar eklenmiştir. subeNo, sube tablosundaki şubelere erişmek için kullanılırken, hocaNo ise hocaların isimlerine erişmek için kullanılan hoca tablosuna bağlantı sağlamaktadır. Bu tablonun eklenmesiyle uygulamada spesifik kısıtlar verilmesi mümkündür. Bu kısıtlar herhangi bir öğretim üyesinin herhangi bir zaman aralığına sınavının koyulmamasından, herhangi bir şubenin sınavlarının belirli zaman aralıklarına koyulmamasına kadar farklılık gösterebilir.

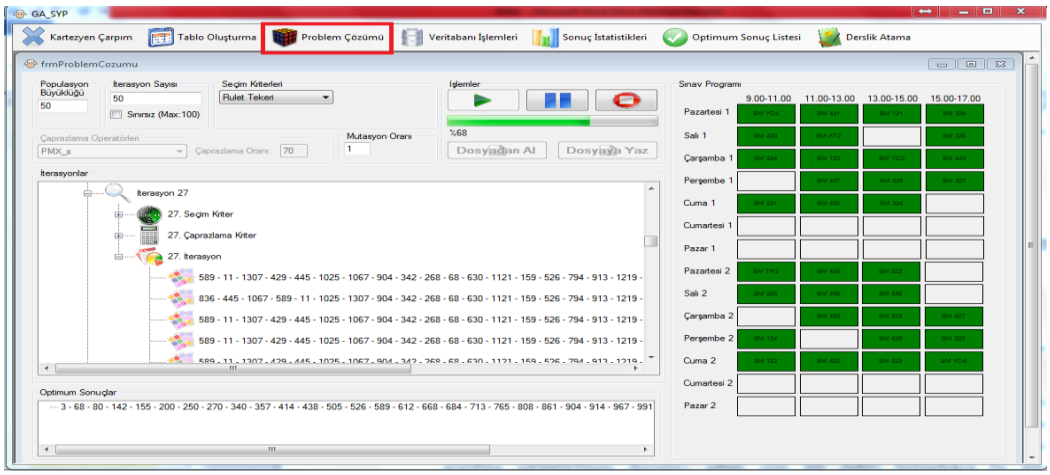
sube tablosu: subeKodu ve subeAdi alanlarından oluşan sube tablosu, her bir subeKoduna karşılık gelen öğrenim sınıfını (1. Sınıf, 2. Sınıf , 3. Sınıf, 4. Sınıf) verir. Esnek kısıtlarda yer alan, aynı günde aynı şubenin birden fazla sınavı olmamasına dikkat edilmesi durumu bu tablonun yardımıyla kontrol edilir. dersKoduHocaSube tablosuyla ilişkili olan sube tablosu dersKoduna ait şube bilgisini bize getirir.

hoca tablosu: hoca tablosu, hocaNo ve hocaAdi alanlarını içerir. İntibak programı içerisinde, farklı şubelere aynı ders koduna sahip dersler verildiği halde programın yoğun olmasından dolayı derslere giren hocaların farklı olması söz konusu olmuştur. Bu karışıklığı önlemek için hoca tablosu ile hangi öğretimin hangi şubesinin hangi dersini kimin verdiği bilgisine kolaylıkla erişilebilecektir.

deger tablosu: deger tablosu ise tablo oluşturma esnasında uygunsuz durumlara ağırlık atayan ceza fonksiyonunun hangi durumda ne kadar ağırlık atayacağını saklar. 40 kişinin aldığı bir dersi 20 kişilik mevcut yer olan bir zaman aralığına koymak çok büyük bir ceza değeri ile cezalandırılırken, yine aynı dersi 120 kişilik kapasiteye sahip bir zaman aralığına koymak mümkün fakat istenmeyen bir durum olacaktır. Bu durumda verilen ceza daha düşük olacaktır. Bu gibi durumların belirlenmesinde bir standart sağlamak amacıyla bu değerler bir kereliğine veri tabanında tanımlanmıştır.

5.4. Problem Çözümü Menüsü

Problem çözümü menüsü, problemin çözümü ile ilgili tüm genetik işlemlerin gerçekleştiği ekrandır. Şekil 5.7'de ekran görüntüsü verilmektedir.



Şekil 5.7. Problem çözümü

Genetik algoritmaların performansını belirleyen önemli etkenlerden biri probleme uygun operatörlerin seçimi ve uygulanma olasılıklarının belirlenmesidir. Aynı şekilde popülasyon büyüklüğünün ve genetik işlemleri durdurma kriterinin doğru bir şekilde belirlenmesi performans açısından önem arz etmektedir. Literatürde bu seçim ve değerlerle ilgili net bir bilgi olmamasıyla birlikte sıklıkla kullanılan operatörlerden ve ortalama olasılık değerlerinden bahsedilmektedir. Ancak probleme özgü olarak değişen bu kriterlerin her birinin etkisini test etmeden yorumlamak bizi yanlış sonuçlara çıkartabilir. Bu nedenle operatör seçimleri, uygulanma olasılıkları, popülasyon büyüklüğü ve durdurma kriterleri kullanıcının seçimine bırakılarak problem çözme ekranından parametre olarak girilmektedir.

"Populasyon büyüklüğü", kullanıcı tarafından programın başında bir kereye mahsus olmak üzere belirlenerek, başlangıç kromozomlarının ve bu kromozomlardan elde edilecek yeni bireylerin sayısını ifade eder. "**İterasyon sayısı"**, kullanılan genetik algoritmadaki tüm işlem adımlarının kaç defa tekrarlanacağını belirler. Aynı zamanda durdurma kriteri olarak da kullanılmaktadır. Özellikle optimum sonuç değerinin kesin olarak bilinmediği durumlarda, kullanıcı işlemlerin belli bir süre sonra bitmesini iterasyon sayısını belirleyerek sağlamaktadır. Geliştirilen uygulama da en iyi sınav programı şeklinde tek bir çözüme ulaşma hedefi yoktur. Verilen kısıtlara bağlı olarak alternatif çözümler de üreten programda durdurma kriteri, iterasyon sayısı ile veya "**sınırsız"** (maksimum 100 iterasyon) seçeneği işaretlenerek sağlanmaktadır.

"Seçim kriteri", başlangıç popülasyonundan elde edilen kromozomlardan rulet tekeri, rank ve elitizm operatörlerinden birinin kullanılmasıyla yeni neslin oluşmasını sağlayan kriterdir. Seçim kriterinin belirlenmesi kullanıcıya bırakılmıştır.

"Çaprazlama operatörleri" olarak kısmi haritalanmış çaprazlama ve sıralı çaprazlama kullanılarak mevcut popülasyondan yeni bireyler türetilmeye çalışılmıştır. Çaprazlamaya katılacak kromozomların seçimi iki farklı şekilde yapıldığı için çalışmanın sonunda dört farklı şekilde karşılaştırma yapılacaktır. Kullanıcının seçtiği çaprazlama operatörüne göre yapılacak olan yeni birey elde etme işlemi yine kullanıcının karar verdiği **"çaprazlama oranı"** içerisinde gerçekleşecektir.

“Mutasyon operatörü” için her hangi bir seçim işlemi söz konusu değildir. “**Mutasyon oranı”**na doğru karar vermek yeterli olacaktır.

“İşlemler” kısmı, kullanıcıya programı çalıştırmaya başlamasına, gerekli olduğu durumlarda durdurmasına ve yeterli gördüğü takdirde iterasyon sayısına erişmeyi beklemeden sonlandırma işlemi yapmasına imkân tanımaktadır

“Progress bar” ile genetik işlemlerin yapılma aşaması kullanıcı tarafından gözlemlenebilir hale gelmektedir. Sürecin ne kadar ilerlediği buradan takip edilebilir.

“İterasyonlar listesi” ise tüm iterasyon adımları ve her birinin içinde gerçekleşen seçim operatörü ve çaprazlama operatörü uygulamalarının sonucunda elde edilen kromozom değerlerini görüntülemektedir. Elde edildikçe ekrana görüntülenen kromozomlar ile programa hareketlilik kazandırılmıştır. Ayrıca kromozomların üstüne gelindiğinde, dışarıdan bakıldığında kullanıcı için nümerik sayıların oluşturduğu bir dizilimden başka bir şey hitap etmeyen kromozom değerlerinde saklanılan bilgiler **“sınav programı”** altında çizelgelenerek anlamlı hale gelmektedir. Programda aynı zaman aralığına yerleştirilen sınavların çakışması söz konusu olabileceği için bu durumlar kırmızı ile gösterilerek kullanıcının dikkatinin çekilmesi sağlanmış ve takibi kolaylaştırmıştır. Çakışma olmadığı durumlar ise yeşil ile gösterilmiştir.

Tüm iterasyonların ilerleme sürecinde optimum sonucun elde edilip edilmediğini tek tek kromozomları kontrol ederek sınav programı çizelgesinde görmek, popülasyon sayısının ve iterasyon sayısının fazla olduğu durumlarda oldukça zordur. Popülasyon büyüklüğü*iterasyon sayısı kadar ihtimali karşımıza getirir. Bunun yerine, her bir iterasyon sonucunda elde edilen optimum sonuçların kontrolü yapılmakta ve “**Optimum Sonuçlar”** listesinde görüntülenmektedir. Şekil 5.7’de sınav programında görüntülenen çizelge optimum sonuç listesindeki kromozom bilgisidir.

5.5. Derslik Atama Menüsü

Problem çözümü menüsünde, sınavların zaman aralıklarına yerleştirilme işlemi tamamlandıktan sonra bu sınavların hangi dersliklerde uygulanacağına karar vermek gerekmektedir. Dersliklerin belirlenmesinde dikkat edilmesi gereken nokta, sınava girecek öğrenci sayısı ile o sınav için atanan dersliklerin kapasitesinin örtüşmesidir. 2

haftalık zaman diliminde yapılan sınavlarda, normal öğretim için gün içerisinde 4, ikinci öğretim içinse 3 ayrı zaman aralığı belirlenmiştir. Her bir zaman aralığında hangi dersliklerin kullanılabilir (diğer sınavlar için atanmamış) olduğu ve mevcut kapasite bilgisi Kartezyen çarpım menüsünde kullanıcı tarafından girilerek bir dosyaya kaydedilmektedir.

Derslik atama menüsü Şekil 5.8'de görüldüğü üzere satır bilgisinde gün-saat, sütun bilgisinde derslik bilgilerinin yer aldığı bir grid içermektedir. Sol tarafta yer alan **"Optimum sonuç listesi"** kullanıcının problem çözümü sonrasında elde ettiği sonuç dosyalarının en iyi (minimum) uygunluk değerlerine sahip (çözüm kalitesi yüksek olan sınav programları) çözümleri sıralı olarak bulundurmaktadır. Sonuç dosyalarından bu çözümlerin tespit edilmesi işlemi ise **"Optimum sonuç listesi"** menüsünde yapılarak, karşılaştırılmak üzere ayrı bir dosyaya yazdırılmaktadır.



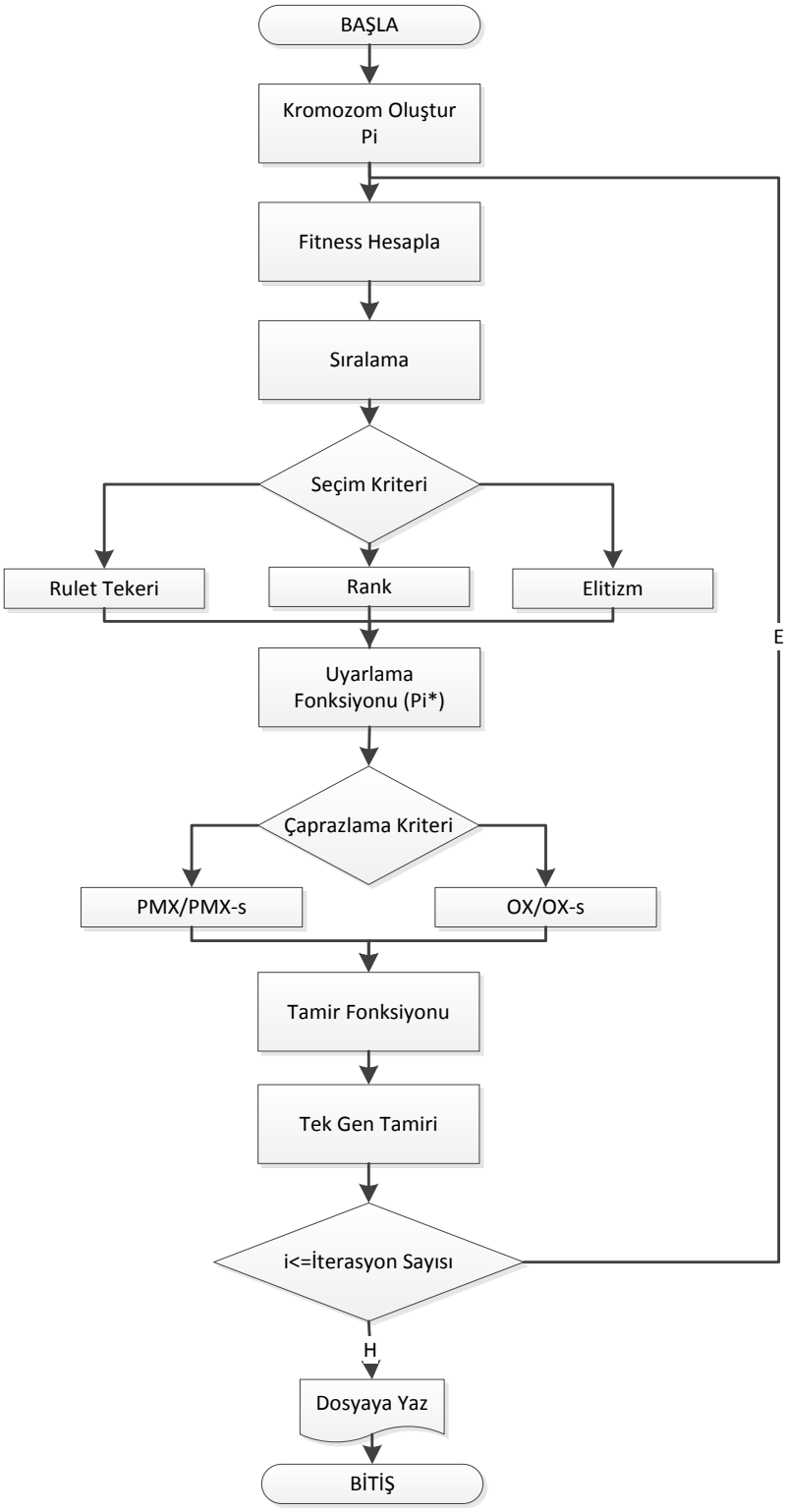
Şekil 5.8. Sınavlar için uygun dersliklerin belirlenmesi

Grid'de yeşil ile gösterilen alanlar gün-saat bilgisine karşılık gelen kullanılabilir derslik bilgilerini, mavi alanlar ise sınavların atandığı derslikleri temsil etmektedir. Kullanıcının tüm ders kodlarının hangi derslere karşılık geldiğini bilmemesi durumu dikkate alınarak veri tabanından, uygulanmasına karar verilen sınav programındaki ders kodlarına karşılık gelen ders isimleri çekilerek, yerleştirildiği gün-saat bilgisi satırındaki tüm yeşil alanlara yazılır. Tüm sınavlar için dersliklerin seçilmesi işlemi kullanıcıya

bırakılmıştır. Çoğu zaman kalabalık olmasından dolayı zorunlu kalınmadıkça amfi salonu sınavlar için tercih edilmemektedir. Seçilen dersin üzerine gelindiğinde o sınava girecek toplam öğrenci sayısı **"Kalan kontenjan"** eklentisinde verilmekte ve derslik ataması yapıldıkça kapasite kadar sayı değeri azaltılmaktadır. Yanlış seçimlerde ise atama işlemi geri alınabilmekte ve yerleştirilmesi gereken öğrenci sayısı artmaktadır. Böylece kullanıcıya, her bir sınav için ne kadar derslik belirlemesi gerektiğinin takibini yapabilme kolaylığı sağlanmıştır.

5.6. Geliştirilen Uygulamada Genetik Algoritmaların İşleme Süreci

GA'ların problemdeki genel işleyişi (akış diyagramı) Şekil 5.9'da gösterilmiştir.



Şekil 5.9. Problemin akış diyagramı

Çözüm uzayının matematiksel olarak ifade edilmesi

- T, zaman aralıklarının sonlu kümesi

  $T = \{ t_1, t_2, \ldots , t_{gs} \}$ *gs= gün-saat kartezyen çarpım adedi*

- S, sınavı yapılacak ders kodlarının sonlu kümesi

  $S = \{ s_1, s_2, \ldots , s_{ss} \}$ *ss= dönemdeki sınav sayısı*

- K, kısıt bilgilerinin sonlu kümesi

  $K = \{ k_1, k_2, \ldots , k_{zorunlu\ kısıt\ +\ esnek\ kısıt} \}$

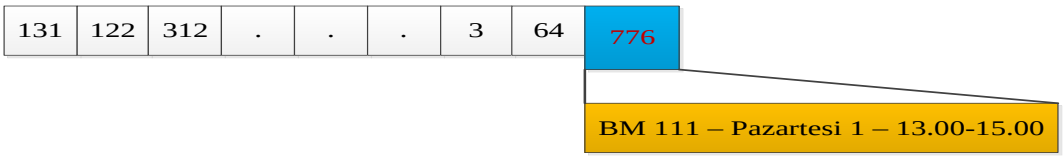
  $K_n = \{ k_1, k_2, \ldots , k_n \}$ $K_m = \{ k_{n+1}, k_{n+2}, \ldots , k_m \}$

  n= zorunlu kısıt sayısı *m= esnek kısıt sayısı*

- D, derslik bilgilerinin sonlu kümesi

  $D = \{ d_1, d_2, \ldots , d_{ds} \}$ *ds= derslik sayısı*

Genler, çözümün karakteristik özelliğini taşıyan, permütasyon kodlama ile kodlanan tekil değerlerdir. Tüm olasılıkların numaralandırıldığı nümerik değerler ile temsil edilir. Tek bir genin ifade ettiği değer (ders kodu-gün-saat), sınavı yapılacak dersin hangi zaman aralığına yerleştirildiğini gösterir (Şekil 5.10).



Şekil 5.10. Gen değerinin içerdiği verinin gösterimi

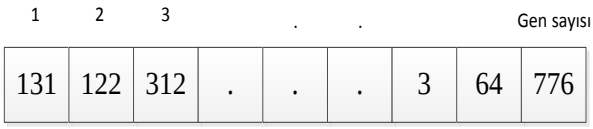
Gen havuzu, sınavların tüm zaman aralıklarına yerleştirilme durumlarını içerir. Tek bir dersin sınavının tüm olasılıkları, sınav günlerinin iş günleri olarak seçildiği düşünülürse, 10 gün ve her günde 4 oturum olmak üzere 40 tanedir. Gen havuzunun büyüklüğü bu durumda;

Gh= gs*ss, Gh= gen havuzu büyüklüğü

gs= gün-saat kartezyen çarpımı, ss= sınavı yapılacak derslerin sayısı şeklindedir.

Kromozom, problemin çözümünü (sınav programını) temsil etmektedir ve sınavı yapılan derslerin sayısı kadar genlerin diziliminden oluşur. Şekil 5.11'de popülasyondaki bir kromozomun nasıl ifade edildiği gösterilmektedir.

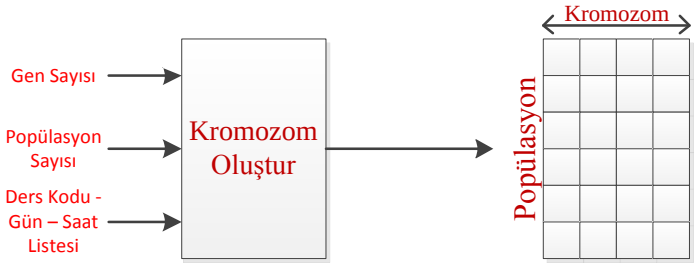
Kromozom= $\sum_{i=1}^{ss} G_i$ = $G_1, G_2, G_3, \ldots , G_{ss}$



Şekil 5.11. Tek bir kromozomun ifadesi

Popülasyon, sınav çizelgelemenin çözümlerinin (kromozomların) oluşturduğu topluluktur. Popülasyon büyüklüğü kullanıcı tarafından parametre olarak girilir.

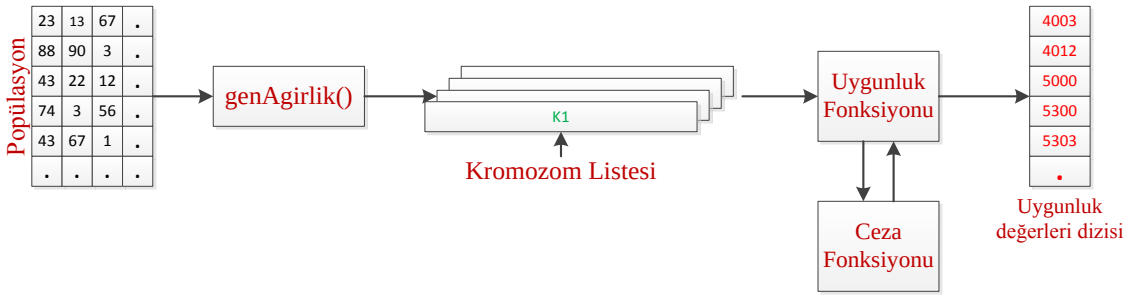
Kromozom Oluştur, (Şekil 5.12) gen sayısı, popülasyon sayısı ve ders kodu-gün-saat listesini parametre olarak almaktadır. Problemin çözümünü kolaylaştırmak ve hesaplama süresini azaltmak için, kromozomları oluştururken ders kodu çakışmasını önleyecek şekilde gen değerleri seçilmiştir. İşlem, seçilen gen değerinin ders kodu bilgisine ait tüm veriler gen havuzundan silinerek, bir sonraki gen seçiminin yapılmasıyla gerçekleşmiştir. (örn. BM 121-Pazartesi 1-09.00-11.00 ifadesini temsil eden gen değeri seçildiyse, BM 121-Salı 1-13.00-15.00 verisi havuzdan silinecektir.)



Şekil 5.12. Kromozom oluşturma ve başlangıç popülasyonunun elde edilmesi

Uygunluk değerlerinin hesaplanması, elde edilen popülasyonun kromozom değerlerinin uygunluk fonksiyonuna verilmesi ile gerçekleşir. Problemimiz kısıtlı kaynakların optimum şekilde kullanılmasını ifade ettiği için, problemin yapısına uygunluğu açısından problem tipi olarak minimizasyon seçilmiştir. Uygunluk değeri, kromozomların olası çözümlere olan uzaklığını belirtir. Giderek azalan uygunluk değerleri çözüme yakınsarken, artan değerler ise çözümden uzaklaşır.

Uygunluk değerlerinin hesaplanması, Şekil 5.13'teki gibi yapılmaktadır. **"genAgirlik()"** fonksiyonu ile popülasyondaki her bir bireyin gen ağırlıkları hesaplanmakta ve kromozom listesi (popülasyon) ile birlikte uygunluk fonksiyonuna parametre olarak gönderilmektedir. Fonksiyon, genlerin ağırlık değerlerini ve ceza fonksiyonundan elde edilen değerleri kullanan matematiksel bir formül ile popülasyondaki tüm kromozomların çözüme yakınlığını hesaplayarak, uygunluk değerleri dizisini döndürmektedir.



Şekil 5.13. Kromozomların uygunluk değerlerinin hesaplanması

$$\boldsymbol{f(x_i)} = \min\left(\sum_{k=1}^{ss} GA_k + C_{\mathrm{dK}}(x_i) + C_{\mathrm{gS}}(x_i) + C_{\mathrm{sC}}(x_i)\right)$$

i= 1,2, … , pb

Kısıt sayısının fazla olduğu problemlerde, uygun olmayan çözümlerin popülasyondaki yüzdeliği daha fazla olmaktadır. Arama uzayında, uygun olmayan bölgeler sınırlandırılmazsa, genetik işlemlerin çözüme ulaşma süresi uzayacaktır. Ceza tekniğinin amacı, uygun olmayan çözümlerin cezalandırılarak, seçim sürecinde elenmesini sağlamak ve böylelikle çözüm uzayını daraltmaktır. **Ceza fonksiyonu**, kromozomları cezalandırma işlemini birçok farklı şekilde yapabilir. Ceza fonksiyonu değeri, kromozomların kısıtları ne derecede ihlal ettiğini gösterir.

Genetik işlemler sırasında, her bir kısıtın ihlali için farklı ceza değerleri belirlenmiştir. Aynı günün aynı saatinde birden fazla sınavın olması, sınav tekrarının yapılması gibi istenmeyen durumlar yüksek ceza değerleri ile cezalandırılarak bu kısıtları ihlal eden kromozomların bir sonraki nesle aktarılması önlenmiştir.

Ders kodu çakışması, yani sınav tekrarı durumunun cezalandırma fonksiyonu aşağıda ifade edilmiştir. Aynı fonksiyon gün-saat çakışması (aynı zaman aralığında birden fazla sınav olması) ve şube çakışması (örn. 1.Sınıf öğrencilerinin gün içerisinde birden fazla sınavı olması) için de geçerlidir. Aralarındaki fark, ceza değerleridir ($gS_{\text{ceza değeri}}$, $sC_{\text{ceza değeri}}$).

$$\boldsymbol{C_{\mathrm{dK}}(x_i)} = \text{ Eğer} \begin{cases} \text{çs}(\mathrm{dK_i}, \mathrm{dK_j}) = 0, & C_{\mathrm{dK}}(x_i) = 0 \\ \text{çs}(\mathrm{dK_i}, \mathrm{dK_j}) \geq 1, & C_{\mathrm{dK}}(x_i) = \text{çs} * \mathrm{dK}_{\text{ceza değeri}} \end{cases}$$

i= 1,2, …, n

j= i+1, i+2, … , n

Sıralama, uygunluk fonksiyonundan elde edilen değerlerin sıralama fonksiyonuna verilme işlemidir. Sıralama minimum değerden maksimum değere göre yapılmaktadır. Amaç, en iyi bireylerin ilk sıralarda kalmasını sağlamak ve konumunu belirlemektir. Konum bilgisi, iyi bireylerin takibini kolaylaştırmaktadır ve elitizm için önemlidir.

Seçim kriteri, rulet tekeri seçimi, elitizm ve rank seçimi kriterlerini içerir.

Rulet tekeri seçiminde, kromozomların her birinin uygunluk değerleri, toplam uygunluk değerine bölünerek [0,1] aralığında seçilme yüzdesi bulunur. Minimizasyon problemi olduğu için, yüzdelik değerleri tersten sıralanarak kromozomlarla eşleştirilmiştir. Böylelikle en iyi birey daha fazla seçilme yüzdeliğine sahiptir.

$$\mathbf{P}(\boldsymbol{x_i}) = \frac{\mathrm{f}(x_i)}{\sum_{\mathrm{k=1}}^{\mathrm{pb}} \mathrm{f}(x_k)} \qquad i= 1, 2, \ldots . , pb \qquad pb= \textit{popülasyon büyüklüğü}$$

Rank seçiminde, ilk olarak kromozomlar uygunluk değerlerine göre küçükten büyüğe sıralanmışlardır. İlk kromozoma (en iyi kromozom), N (popülasyon büyüklüğü) değeri verilir, sonraki kromozomlara bu değer azaltılarak atanır. Rulet tekeri seçiminde olduğu gibi her bir değer toplam değere bölünerek [0,1] aralığında değerler elde edilir. Bu değerler her bir kromozomun seçilme yüzdeliğini belirler.

Elitizm seçiminde, parametre olarak girilen oran değerinde, en iyi kromozomlar bir sonraki popülasyona değişikliğe uğramadan aktarılır. Popülasyonun tamamlanması

için yeni kromozomlar, çaprazlama operatörü ile elde edilir. Hangi çaprazlama operatörünün kullanılacağı, kullanıcı tarafından parametre olarak girilmektedir.

Uyarlama fonksiyonu, kromozomları çaprazlama işlemi öncesi şekillendiren fonksiyondur. Kromozomda, gün-saat çakışmasına (aynı günün aynı saatine birden fazla sınavın atanması) ve ders kodu çakışmasına (aynı dersin sınavının birden fazla zaman aralığına yerleştirilmesi durumu), sebep olan gen değeri bulundukça bu gen kromozomun ilk konumuna yerleştirilir ve diğer genler sağa doğru ötelenir.

Şekil 5.14'te uyarlama fonksiyonunun işlem adımlarının sözde kodu verilmiştir.

```
for (int i = 0'dan popülasyon sayısına kadar, i+=1)
{
      for (j = 0'dan gen sayısına kadar, j+=1)
            for (k = j'den gen sayısına kadar, k+=1)
                  if ((dersKodu1 dersKodu2'ye eşitse veya günSaat1 günSaat2'ye eşitse) ve j!= k ise)
                        Çakışma konumları dizisine k indis değerini ekle;
       while (Çakışma konumları dizisinin eleman sayısı > 0)
       {
                  Çakışma konumundaki elemanı dizinin ilk indisine kaydır
                  Kaydırılan değeri Çakışma konumları dizisinden sil
                  Dizinin eleman sayısını -=1;
        }
}
Aralık değeri = Minimum Çakışma Sayısı
return popülasyon;
```

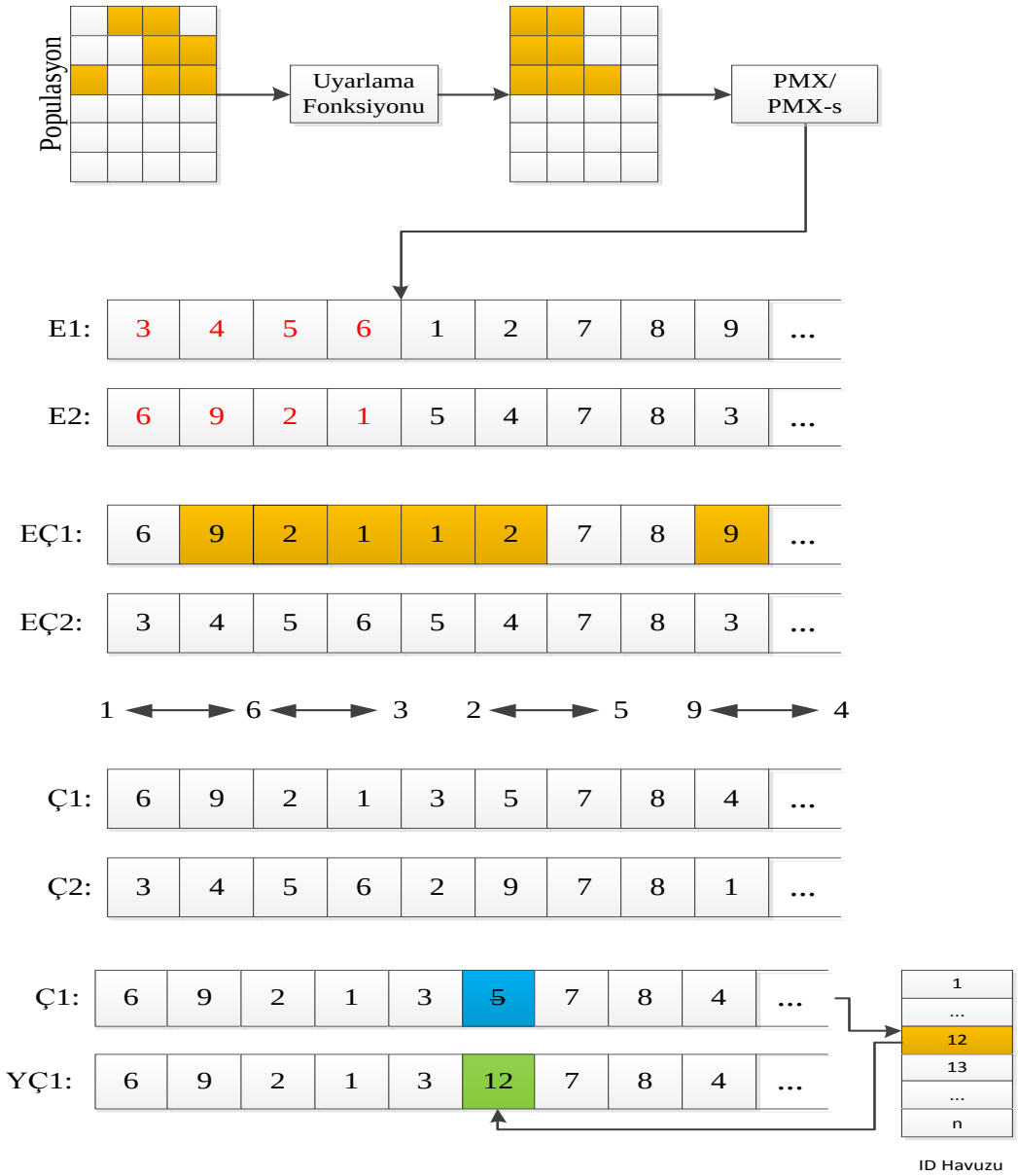
Şekil 5.14. Uyarlama fonksiyonu sözde kodu

Çaprazlama, kullanıcı tarafından girilen çaprazlama oranı (P_c) değerinde kromozomlar arasında gen değişimini sağlar. Amaç, bireylerin iyi özelliklerini bir arada toplayarak popülasyonda çeşitliliği sağlamak ve daha fazla çözümü değerlendirmeye almaktır.

Bir kromozom için çakışmaya neden olan gen değerinin başka bir kromozomda olası bir durumu ifade etmeyeceği bilinemez. Bu sebeple, uygulamada çaprazlama işlemi, rastgele seçilen gen değerleri veya kromozom parçası üzerinde değil, kromozomların ilk konumlarına yerleştirilen sorunlu genler arasında yapılmaktadır. Kodlama olarak permütasyon kodlama kullanılmıştır. Fakat genlerin kromozomda bulunma konumlarının bir önemi yoktur. Bu durum, permütasyon kodlamada kullanılan çaprazlama operatörlerinin birçoğunun probleme uygulanmasını anlamsız kılmaktadır. Geliştirilen uygulamada farklılık yaratan çaprazlama operatörleri, PMX ve OX'tir.

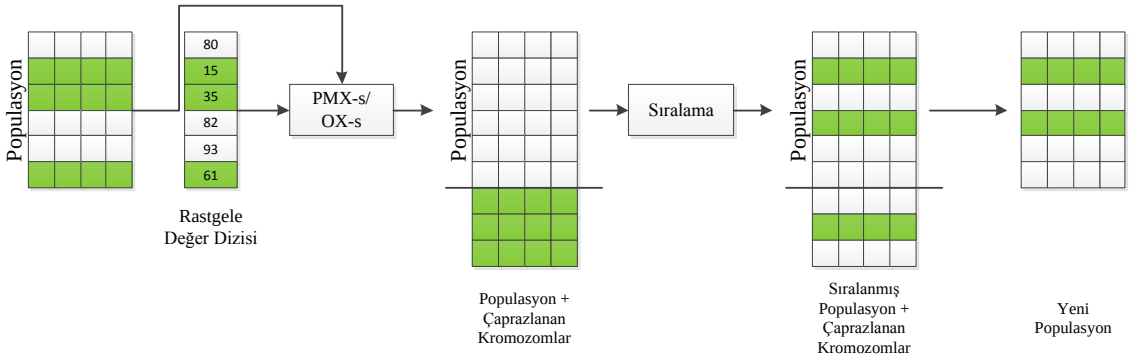
PMX-s ve OX-s şeklinde gösterilen çaprazlama operatörlerinin çalışma mantığı PMX ve OX ile bire bir aynıdır, sadece birbirlerini farklı kılan çaprazlamanın uygulanacağı kromozomların seçim şeklidir. Çalışmada probleme özgü yeni bir çaprazlama operatörü geliştirilmemiştir, çaprazlama havuzuna alınacak kromozomların seçim farklılığını belirtmek için PMX-s ve OX-s şeklinde isimlendirme yapılmıştır. Çaprazlamanın amacı, en iyi kromozomlardan (bireylerden) daha iyi bireyler elde etmektir ancak bunu garanti etmez, aksine iyi bireylerin bilgilerinin kaybolmasına da neden olabilir. Bu nedenle her iki durumunda sağlanması için, PMX, OX, PMX-s ve OX-s operatörleri değerlendirilmiştir. PMX ve OX için bireylerin seçim işlemi, elitizm ile yapılır. Sıralı olan kromozomlarda ilk sıralardaki bireylerin çözüme en yakın bireyler olduğundan söz etmiştik. Çaprazlama işlemi, sıralamanın son kısmından başlayarak çaprazlama oranı değeri kadar ilk kısımlara doğru ilerler. Böylelikle iyi bireyler korunmuş olur.

PMX için çaprazlamanın nasıl yapıldığı Şekil 5.15'te gösterilmiştir. Uyarlama fonksiyonu öncesi dağınık şekilde bulunan sorunlu genlerin, çaprazlama işlemi için uyarlanarak kromozomların ilk konumlarına alındığı görülmektedir.



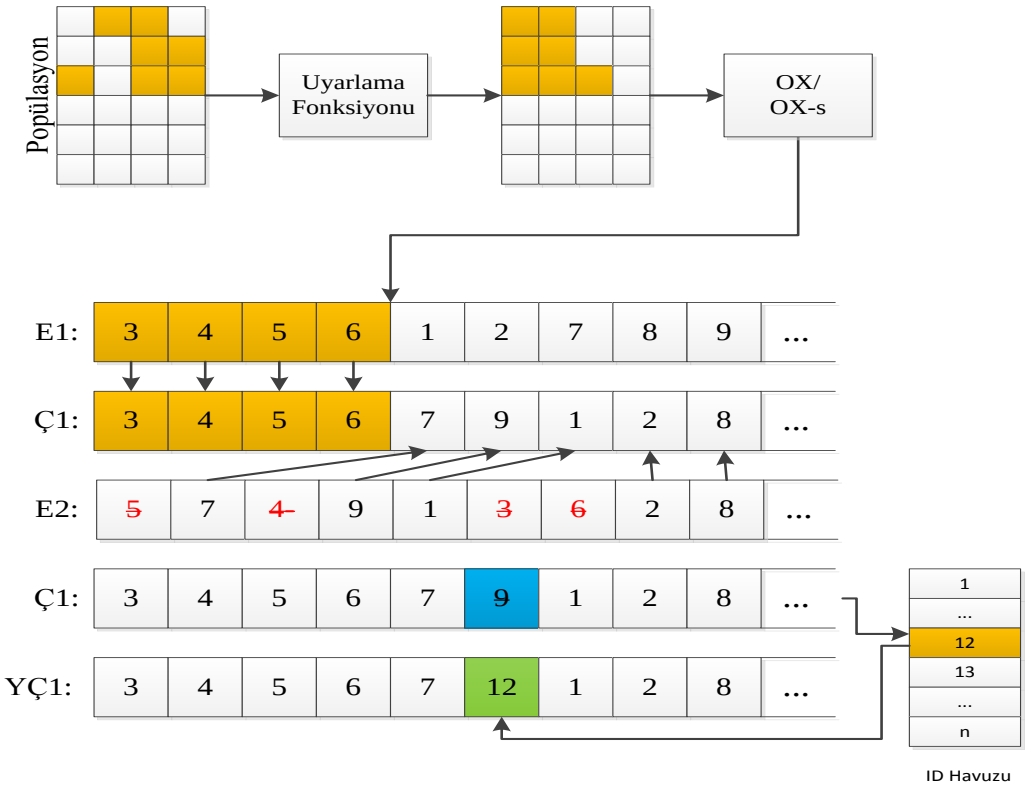
Şekil 5.15. PMX/PMX-s operatörlerinin aşamaları ve Mutasyon işlemi

PMX-s ve OX-s operatörlerinde çaprazlamaya uğrayacak bireylerin seçimi daha çok rastsallığa dayanmaktadır. Çaprazlama oranı burada, rastgele sayılar üretmek için bir üst sınır oluşturur. Popülasyon büyüklüğü kadar üretilen rastgele sayıların orandan küçük olup olmadığı kontrol edilir, her bir kromozom için tutulan sayılar bu şartı sağlıyorsa, bireyler çaprazlama havuzuna aktarılır (Şekil 5.16). Bireylerin seçim işlemi tamamlandıktan sonra çaprazlama havuzunun büyüklüğü kontrol edilir, çiftse çaprazlama operatörü uygulanmaya başlanır, tekse çifte tamamlanacak şekilde havuzun içinde rastgele bir birey kopyalanır.



Şekil 5.16. PMX-s/OX-s için kromozom seçimi ve çaprazlanma sonrası popülasyona eklenmesi

PMX-s ve OX-s operatörlerinden elde edilen yeni bireyler popülasyonda doğrudan, çaprazlama havuzuna seçilen eski bireylerin yerini almazlar. Konumlandırma işleminin aşamaları, yeni bireylerin popülasyonun sonuna eklendikten sonra sıralama fonksiyonuna tabi olmaları ve popülasyon sayısı kadar bireyin ilk sıralardan başlayarak seçilmesi, şeklindedir. Böylelikle, yeni birey ile eski birey arasında karşılaştırma imkânı sunulmuş, çözüme yakın olan bireyler bir sonraki nesle aktarılmıştır.



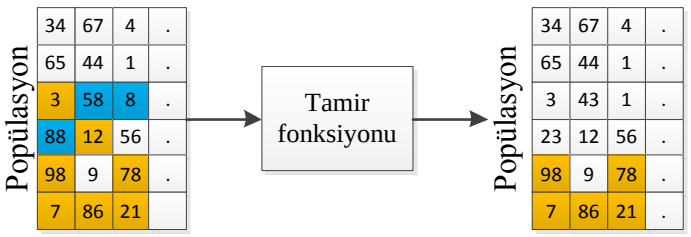
Şekil 5.17. OX/OX-s operatörlerinin aşamaları ve Mutasyon işlemi

Mutasyon, çaprazlama sonucu oluşan yeni bireyler üzerine mutasyon oranı (P_m) kadar uygulanarak, çözüm adaylarının global optimuma ulaşmasını sağlar. Permütasyon kodlamada kullanılan mutasyon operatörleri, konuma dayalı olduğu için farklı bir mutasyon yöntemi uygulanmıştır. Şekil 5.17'de görüldüğü gibi, çaprazlama sonucu oluşan çocukta (yeni birey) rastgele bir konum belirlenmiş, belirlenen konumdaki gen değeri çocuktan çıkartılarak id havuzundan seçilen başka bir değer yerleştirilmiştir. Id havuzundan seçilen gen, kromozoma yerleştirilirken kromozomda her hangi bir çakışmaya neden olup olmadığı kontrol edilir, çakışma sağlamayan bir gen değeri seçilene kadar işlem devam ettirilir (Şekil 5.18).

```
for (j = 1'den havuz büyüklüğüne kadar, j+=1)
      id havuzunu doldur
for (i = 0'dan mutasyona uğrayacak gen sayısına kadar, i+=1)
{
      Rastgele bir x satırı seç
      Rastgele bir y sütunu seç
      0 ile id havuzunun eleman sayısı arasında rastgele bir mutasyon değeri belirle
      if (Belirlenen mutasyon değeri gen dizisinde mevcut değilse)
          Gen dizisinin[satır,sütun] değeri = mutasyon değeri
      else
          Mutasyon değeri belirleme adımına git
}
return popülasyon;
```

Şekil 5.18. Mutasyon sözde kodu

Tamir fonksiyonu, kromozomlar arasında gerçekleşen gen değişiminin sebep olduğu ders kodu tekrarını düzeltmeye yönelik geliştirilmiştir. Başlangıç popülasyonu oluşturulurken ders kodu çakışması önlenmiştir. Fakat çaprazlamada bir kromozom için çakışma oluşturmayan bir gen, diğerinde sınav tekrarına neden olabilmektedir. Şekil 5.19'da ders kodu çakışmaları mavi ile gösterilmiş ve tamir fonksiyonu sonrası istenmeyen bu durum iyileştirilmiştir.



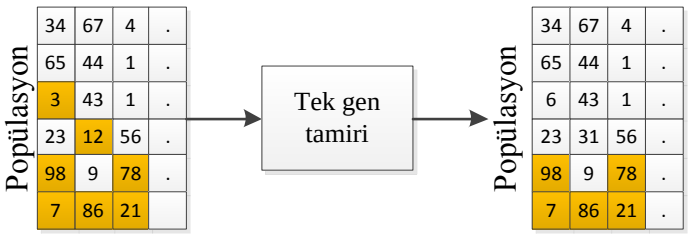
Şekil 5.19. Tamir fonksiyonu

Şekil 5.20'de tamir fonksiyonunun işlem akışının sözde kodu verilmiştir.

```
for ( i = 0'dan popülasyon sayısına kadar, i+=1)
{
      Id listesini oluştur
      for (j = 0'dan gen sayısına kadar, j+=1)
            for (k = 0'dan id listesinin eleman sayısına kadar)
                  if (Gen dizisinin [i,j] elemanının ders kodu = id listesinin[k] elemanının ders kodu)
                        id listesi[k] elemanını sil
      for (j = 0'dan gen sayısına kadar, j+=1)
            for (k = j + 1'den gen sayısına kadar; k+=1)
                  if (Gen dizisinin[i,j] elemanının ders kodu = Gen dizisinin[i,k] elemanının ders kodu)
                        Çakışan listesine k indis değerini ekle
      for (j = 0'dan Çakışan listesinin eleman sayısına kadar, j+=1)
      {
            0 ile id listesinin eleman sayısı arasında rastgele bir R değeri tut
            Gen dizisinde çakışan elemanın yerine id listesinin[R] elemanını koy
            for (k = 0'dan id listesinin eleman sayısına kadar, k+=1)
                  if (id listesinin[k] elemanının ders kodu = id listesinin[R] elemanının ders kodu)
                        id listesi[k] elemanını sil;
      }
}
return popülasyon;
```

Şekil 5.20. Tamir fonksiyonu sözde kodu

Tek gen tamiri, bir iterasyon için tüm genetik işlemler tamamlandıktan sonra, sınav programını ifade eden kromozomlarda sadece tek bir çakışmaya sebep veren geni iyileştirmek için yapılmıştır. Amaç olası alternatif çözümlerin sayısını artırmaktır. Tek gen tamiri mutasyon işlemi olarak da düşünülebilir. Farkı, rastgele konumlardaki gen değerlerini değil, belirli bir konumdaki gen değerini değiştirmesidir. Şekil 5.21'de görüldüğü gibi popülasyon dizisinde 3 ve 12 değerleri gün-saat çakışmasına sebep olduğu için tek gen tamiri ile iyileştirilerek uygulanabilir 2 çözüm daha elde edilmiştir.



Şekil 5.21. Tek gen tamiri

```
for (i = 0'dan popülasyon sayısına kadar, i+=1)
{
    for (j = 0'dan gen sayısına kadar, j+=1)
        for (k = j + 1'den gen sayısına kadar, k+=1)
            if (Gen dizisinin [i,j] elemanının günsaati= Gen dizisinin [i,k] elemanının günsaati)
            {
                Çakışan listesine Gen dizisinin [i,j] elemanını ekle
                konum = k;
            }
    if (çakışan listesinin eleman sayısı= 1)
    {
        foreach (id listesindeki her k elemanı için)
        if (id listesinin[k] elemanının derskodu = çakışan elemanın derskodu)
            id listesine k. elemanı ekle
        for (j = 0'dan gen sayısına kadar, j+=1)
            for (k = 0'dan id listesinin eleman sayısına kadar; k+=1)
                if (Gen dizisinin [i,j] elemanının günsaati = id listesinin[k] elemanının günsaati)
                    id listesi[k] elemanını sil
        0 ile id listesinin eleman sayısı arasında rastgele R değeri belirle
            Gen dizisi[i,konum] = id listesi[R]
    }
}
return popülasyon;
```

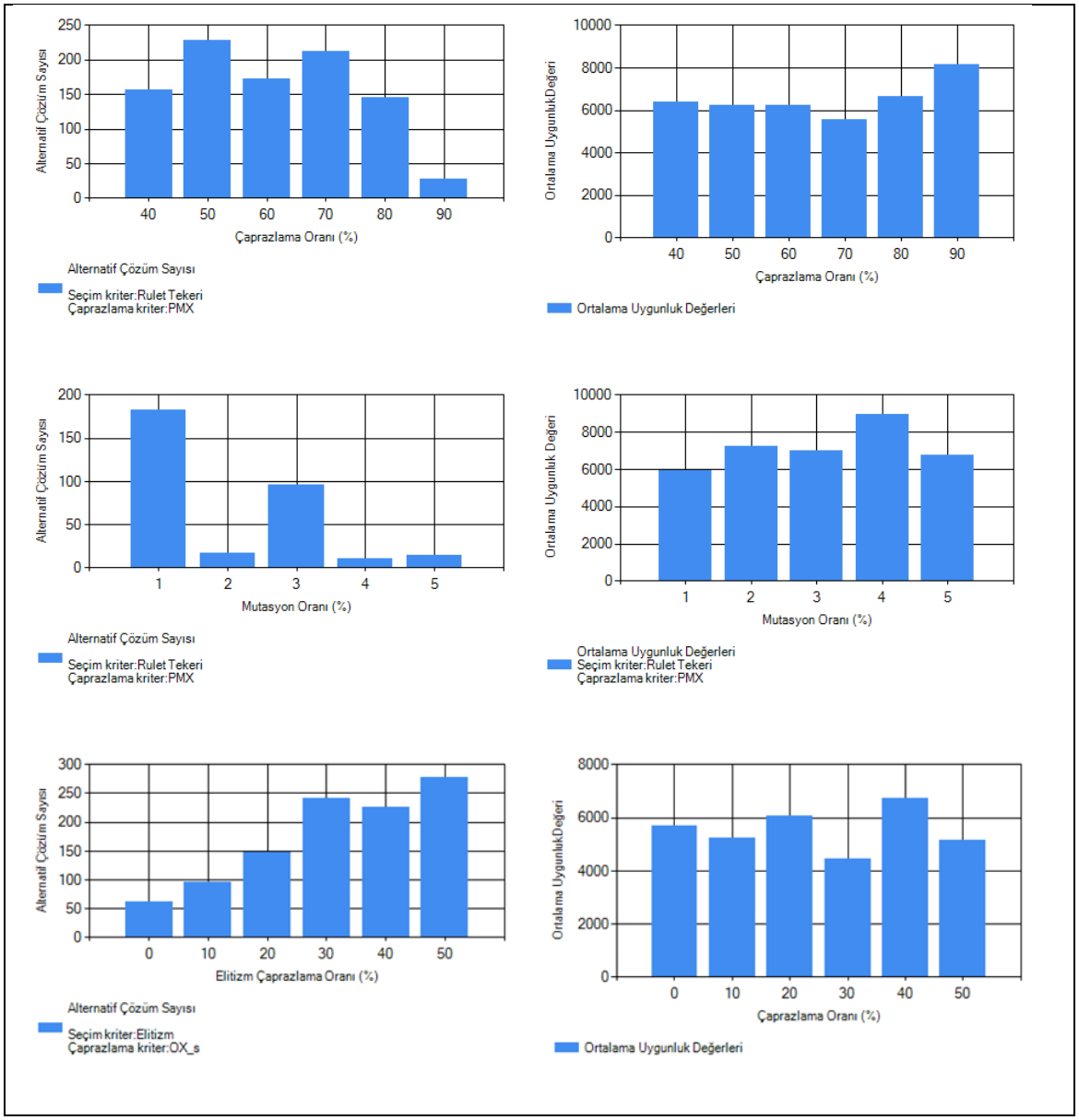
Şekil 5.22. Tek gen tamirinin sözde kodu

5.7. GA Parametrelerine Göre Model Sonuçlarının Değerlendirilmesi

Genetik işlemlerin gerçeklendiği ve uygulanabilir sonuçların alındığı GA_SYP modeli, C# programlama dili ile yazılmış olup, veri tabanı yönetim sistemi olarak Microsoft Access'i kullanmaktadır.Trakya Üniversitesi Mimarlık-Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nün 2011-2012 Bahar dönemi Final sınavları için test edilen sistem, Core i5-2410M CPU @ 2.30 GHz, 6GB RAM ve 64 bit işletim sistemi ile Core i5-M450 CPU @ 2.40 GHz, 4GB RAM, 64 bit işletim sistemi özelliklerini taşıyan iki farklı bilgisayarda denenmiş ve sonuçları alınmıştır.

Problem çözümü menüsünde, kullanıcının karar verdiği her bir genetik algoritma parametre değeri, elde edilen sonucun uygunluk değerini, çözüme ulaşma süresini, sistemin çalışmasını bitirme süresini, alternatif çözüm sayısı vb. şekilde tüm sonuçları olumlu ya da olumsuz yönde etkileme performansına sahiptir. Uygulamada, kullanıcıdan alınan ve doğru tespit edilmesi gereken parametre değerleri ve yöntemler, seçim kriterleri (rulet tekeri, rank, elitizm), populasyon büyüklüğü, iterasyon sayısı, çaprazlama operatörleri (PMX, OX, PMX-s, OX-s), çaprazlama oranı ve mutasyon oranı şeklindedir. Seçim kriterinin elitizm seçildiği durumlarda ise, en iyi bireylerin hangi oranda saklanılacağı ve yeni bireylerin türemesini sağlayan çaprazlama kriterinin elitizm için belirlenmesi gerekmektedir. Bölüm 3'te Genetik Algoritmaların

parametrelerinin seçiminin nasıl yapılması gerektiği, belirlenen değerlere göre sonucun ne yönde seyir edeceği ve öneminden bahsedilmişti. İlk olarak, diğer olası durumların, problemi sonuca ulaştıran çaprazlama, mutasyon ve elitizm oranlarına göre değerlendirilmesi gerektiği düşünüldüğünden kullanıcının bu oranları tespit etmesi gerekmektedir. Bazı oran değerlerinde hiçbir çözüme ulaşılmaması veya az sonuç üretilmesi performans karşılaştırmasını yapmayı zorlaştırmaktadır.



Şekil 5.23. Uygun çaprazlama, mutasyon ve elitizm oranlarının belirlenmesi

Şekil 5.23'te görüldüğü gibi üretilen çözüm sayısı ve sağlanan çözümlerin uygunluğuna göre değerlendirilen oranlarda amaç, mümkün olduğunca fazla sonuç

arasından minimum uygunluğa sahip çözüme erişmektir. Çözümlerin sadece zorunlu kısıtları sağlaması o çözümün elverişli bir durum olduğunu ifade etmediğini belirtmiştik. Çözüm kalitesi, esnek kısıtların belli bir sınıra kadar sağlanması ile doğrudan ilişkilidir. Bu durumun ölçülebilirliğini sağlayan ise uygunluk fonksiyonu değerleridir. Problemimiz minimizasyon problemi olduğu için en düşük uygunluk değerine sahip olan sonuç en iyi durumu sembolize edecektir. Çözüme erken ulaşmak veya fazla sayıda çözüm bulmak bu kriteri sağlamadığında bir anlam ifade etmeyecektir.

Çaprazlama oranı (P_c): %0-100 aralığında değerler alabilen çaprazlama oranında, %40-90 aralığında 10'un katları olan sayılar test edilmiştir. %100 oranında yapılan çaprazlama, bir önceki sınav programları kümesinin(neslin) tamamen farklılaştırılarak en iyi çözümlerin kaybolmasına, %40'ın altındaki oranlar ise her nesilde aynı sınav çizelgelerinin tekrarına neden olacağı için dikkate alınmamıştır. Sonuçlara bakıldığında, ortalama olarak en fazla alternatif çözüme %50, %60 ve %70 oranlarında ulaşılırken minimum ortalama uygunluk değerinin %70'te sağlandığı görülmüştür.

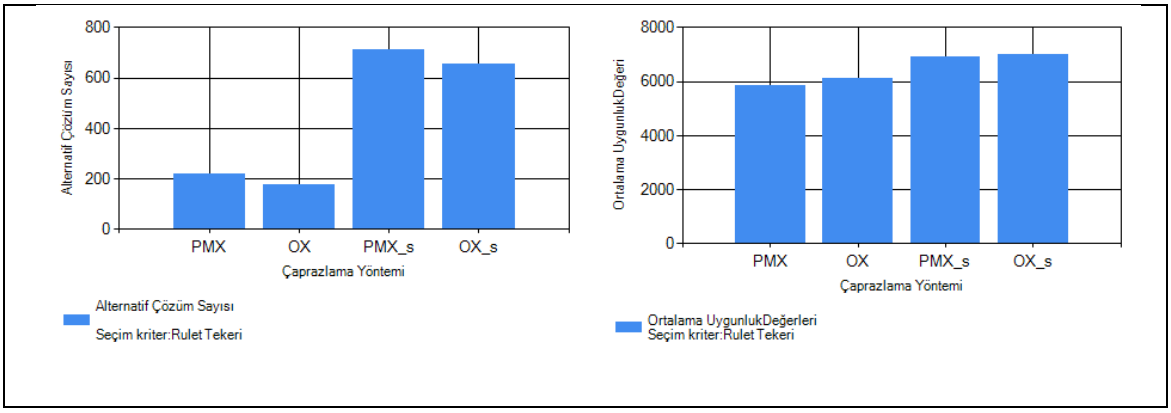
Mutasyon oranı (P_m): Literatürde daha çok %1-5 aralığında değerler alan mutasyon oranı, problemin çözüm sürecinde yerel optimuma takılarak nesil çeşitliliğinin azalmasını önlemektedir. Rastsal olarak uygulanan bir işlem olması nedeniyle oranın büyük seçilmesi optimuma yakın olan programların gen değerlerinin kaybolmasına (veri kaybı) sebep olabileceği için problemin çözüme ulaşmasını engelleyebilir ya da süresini uzatabilir. Yapılan denemelerde ise %5 oranında hiç çözüm elde edilemediği durumlar olmuştur. Fazla sayıda sonuca ortalama %1 ve %3 oranlarında ulaşıldığı fakat en iyi uygunluk değerlerinin %1 mutasyon oranında elde edildiği gözlemlenmiştir.

Elitizm oranı (P_e): Seçim kriterlerinden biri olan elitizmde, belirlenen oran değerinde iyi olan sınav programları bir sonraki nesle aktarılacağı için nesil çeşitliliği oranla ilişkilidir. Oranın büyük bir değer seçilmesi bir önceki nesille cok fazla farklılık göstermeyeceği için işlem tekrarı gibi düşünülebilir. Küçük bir değer seçilmesi ise uygunluk değeri iyi olan çözümlerin korunması mantığı ile hareket eden elitizm yöntemi ile örtüşmemektedir. %0-50 arasında 10'un katı olan sayı değerlerine göre

hesaplanan elitizm yönteminde, en uygun değere sahip sınav programlarına %30 oranında erişilmiştir. %0 oranının test edilmesinin amacı, tüm neslin değişikliğe uğrayarak her iterasyonda yeni sınav programlarının üretilmesi ile çözüme varılıp varılamayacağını göstermektir. Şekil 5.23'te de görüldüğü gibi uygulanabilir çözüme yakın olan sınav programlarının korunmadığı durumlarda, çözüm kümesi elde edilsede ortalama uygunluk değerlerinin 5000'den fazla olması istenmeyen durumları içerdiğini göstermektedir.

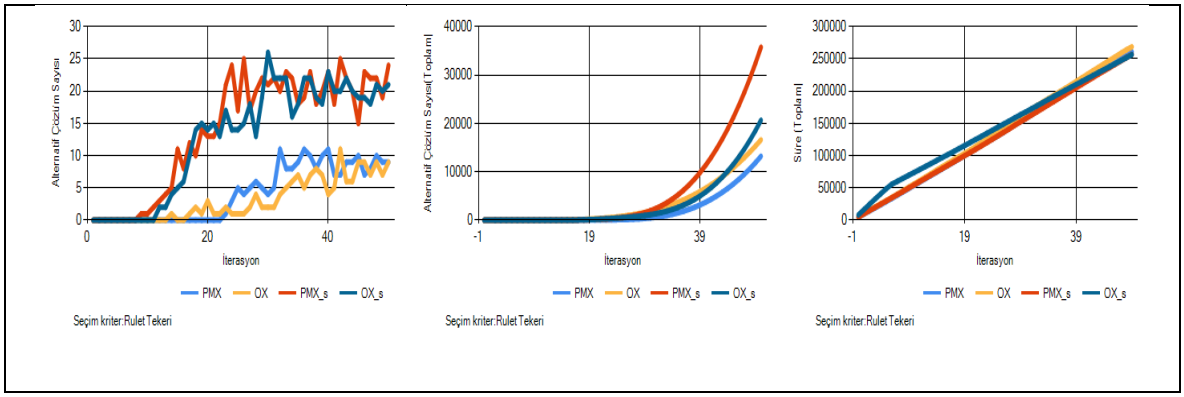
Belirlenen ihtiyaçlara cevap veren sınav programının hazırlanmasında oranlar kadar seçim ve çaprazlama kriterleri de büyük önem arz etmektedir. Sistemde tüm kriterlerin birbiri ile kombinasyonunu oluşturarak birçok sonuç alınması mümkündür. Bu durum kullanıcının en uygun seçim ve çaprazlama kriterlerini seçmesine ve farklı sonuçlar alarak olası çözüm uzayının büyük bir yüzdesine erişmesine imkân tanımaktadır. Örneğin, modelin çalışma performansı çaprazlama operatörlerine göre test edilmek istendiğinde her bir seçim kriteri göz önüne alınarak değerlendirilirse, rulet tekeri seçimi için 1, rank seçimi için 1 fakat elitizm için belirlenen her bir çaprazlama kriterine karşılık gelen çaprazlama operatörlerinden dolayı 4 farklı sonuç grafiği sağlanırken, 24 farklı çözüm kümesi elde edilmektedir.

Çalışmada, PMX ve OX olmak üzere 2 ayrı çaprazlama operatörü kullanılmış ancak çaprazlama havuzuna alınacak sınav programlarının seçimi farklılaştırılmıştır. İlkinde, popülasyon sayısı*çaprazlama oranı kadar kötü birey (çakışma veya istenmeyen durumların yer aldığı sınav programları) havuza dahil edilirken, diğerinde popülasyon sayısı kadar rastgele (0-çaprazlama oranı) arasında değer tutularak, orandan küçük olanlar yeni sınav programları üretmek için havuza eklenir. Aralarında ayrım yapabilmek amacıyla PMX-s ve OX-s şeklinde isimlendirilen çaprazlama yönteminde ikinci seçim yöntemi kullanılmaktadır. Çaprazlama operatörlerinin başarımını, bulunan uygun oran değerleri altında tüm seçim kriterlerine göre karşılaştırdığımızda dikkate alınan sonuçlar yine alternatif çözüm sayısı ve ortalama uygunluk değerleri olmuştur.



Şekil 5.24. Rulet tekeri seçiminde çaprazlama operatörlerinin değerlendirilmesi

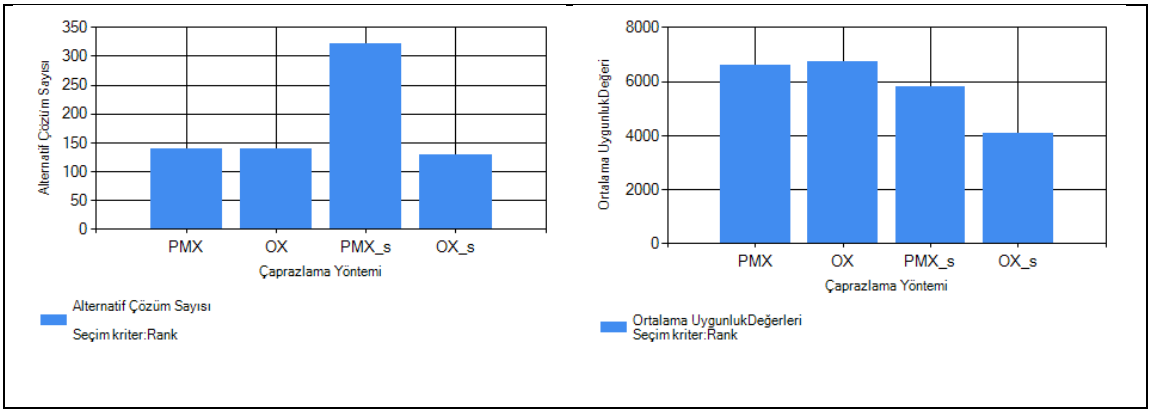
Sırasıyla her bir seçim kriterine göre değerlendirilen çaprazlama yöntemlerinin verdiği sonuçlara bakılırsa, Rulet tekeri seçiminde Şekil 5.24'te PMX-s ve OX-s daha fazla çözüm bulurken, PMX ve OX daha uygun sonuçlar üretmektedir. Net olarak her zaman PMX-s ile fazla sayıda sonuç elde edilmesi mümkündür denilememektedir. Bazı denemelerde OX-s ile de bu sonuca ulaşılmıştır. Alternatif çözüm üretme ve Ortalama uygunluk grafikleri incelendiğinde, aynı seçim kriterinde farklı çaprazlamaların ürettiği sonuçların birbirine yakın olduğu görülmektedir. Alternatif çözüm sayısı ve ortalama uygunluk değerlerinin yanı sıra, ilk optimum sonucu bulma süreleri, ilk olarak hangi iterasyonda çözüme ulaşmaya başladıkları, her iterasyonda buldukları sonuç sayıları ve işlemleri gerçekleştirme süreleri de değerlendirme sırasında göz önünde bulundurulmuştur.



Şekil 5.25. Rulet tekeri seçiminde çaprazlama operatörlerinin her iterasyona göre değerlendirilmesi

Şekil 5.25'te ilk grafiğe bakıldığında PMX-s'nin çözüme erken ulaştığı ve 20. iterasyondan sonra yöntemlerin daha fazla sonuç ürettiği yorumu yapılabilmektedir.

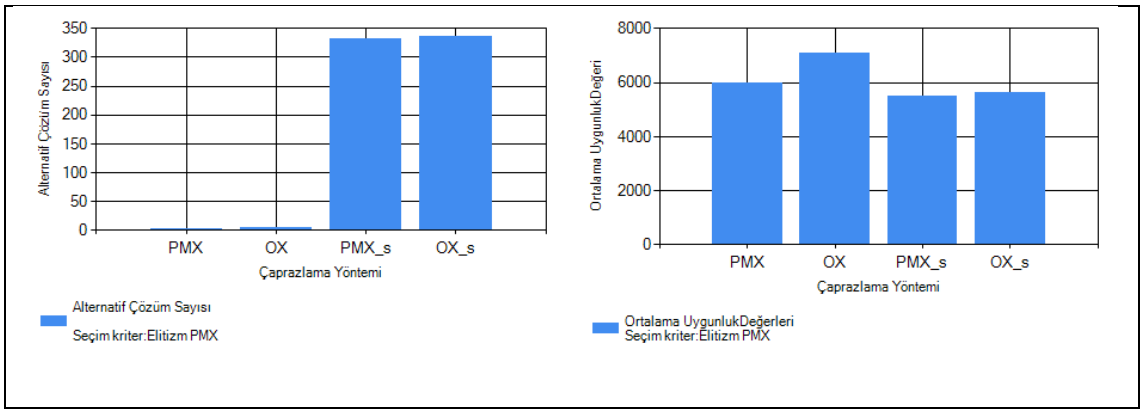
Durdurma kriteri olarak iterasyon sayısı kullanılan programda, çaprazlama operatörlerinin harcadıkları süre dikkate alındığında ortalama olarak aynı sürede işlemlerini tamamladıkları üçüncü grafikte görülmektedir.



Şekil 5.26. Rank seçiminde çaprazlama operatörlerinin değerlendirilmesi

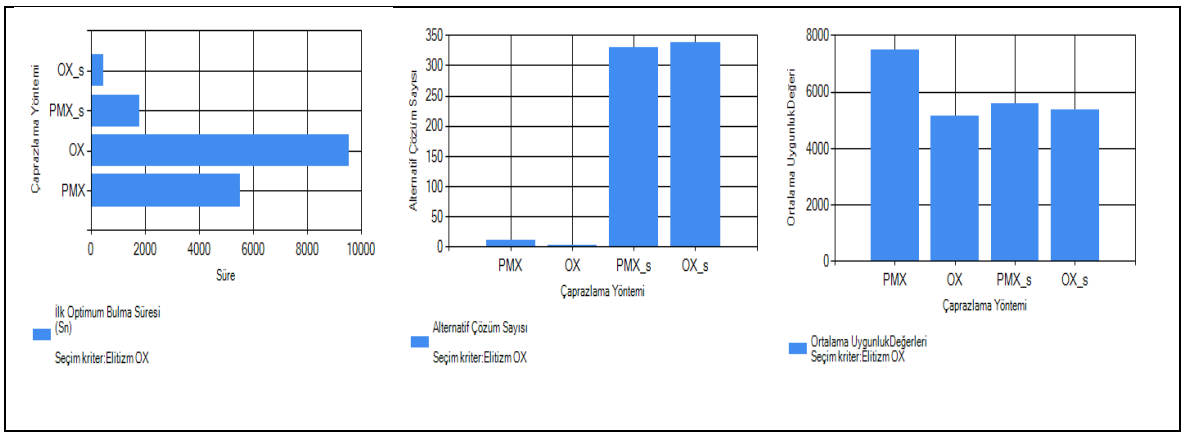
Rank seçim kriteri altında uygulanan farklı çaprazlama yöntemlerinde alternatif çözüm sayılarını karşılaştırdığımızda en fazla çözümün PMX-s ile elde edilirken en az çözümün OX ve PMX ile sağlandığı görülmektedir. Şekil 5.26'nın ikinci grafiğinde ortalama uygunluk değerinin OX-s'de minimum olması, diğerlerine göre bulduğu çözümlerin esnek kısıtları sağlama oranının daha yüksek olduğunu ifade etmektedir. Bu durumda Rank seçimi için, PMX-s'nin ürettiği sonuçların fazla olmasına rağmen OX-s'nin kullanılmasının daha uygun olduğu kanısına varılabilmektedir.

Elitizm seçiminde elitizm oranı*popülasyon sayısı kadar birey (örn. elitizm oranı %30, popülasyon sayısı 100 olduğu durumda 0,3*100 sayısı kadar birey), doğrudan bir sonraki popülasyona aktarılırken, geri kalan bireylerin tamamlanması için çaprazlama yöntemi ile yeni bireyler elde edilerek nesil çeşitliliği sağlanır. Bu durumda çaprazlama işlemi iki defa gerçekleşmektedir. Elitizm seçimi içinde kullanılan çaprazlama operatörünün seçimi kullanıcıya bırakılmıştır. Elitizm seçiminde, çaprazlama operatörlerinin karşılaştırılması, elitizm içerisinde yapılan her bir çaprazlama kriterine göre yapılmıştır.



Şekil 5.27. PMX çaprazlama operatörünü kullanan Elitizm seçiminde, çaprazlama operatörlerinin değerlendirilmesi

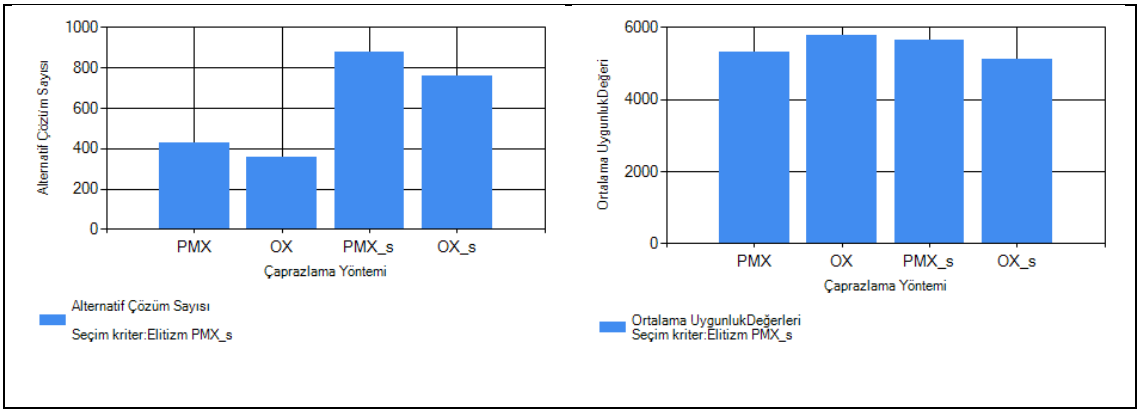
Elitizm içerisinde PMX çaprazlama operatörü kullanıldığında Şekil 5.27'de görüldüğü üzere, PMX ve OX'e göre çok daha iyi sonuçlar veren PMX-s ve OX-s, kendi aralarında kıyaslandığında OX-s'nin daha fazla çözüm ürettiği, aralarında büyük bir fark olmasada PMX-s'nin ulaştığı sonuçların aranılan çözüme daha yakın olduğu söylenebilir. OX operatörünün oldukça elverişsiz çözümler üretmesi, PMX operatöründen sonra uygulanmasının anlamsız olduğu yargısına varmamızı sağlamaktadır. Şekil 5.28'e de bakıldığında, OX operatörünün ardından PMX operatörünün gerçekleşmesinin de elverişsiz bir durum olduğu görülmektedir.



Şekil 5.28. OX çaprazlama operatörünü kullanan Elitizm seçiminde, çaprazlama operatörlerinin değerlendirilmesi

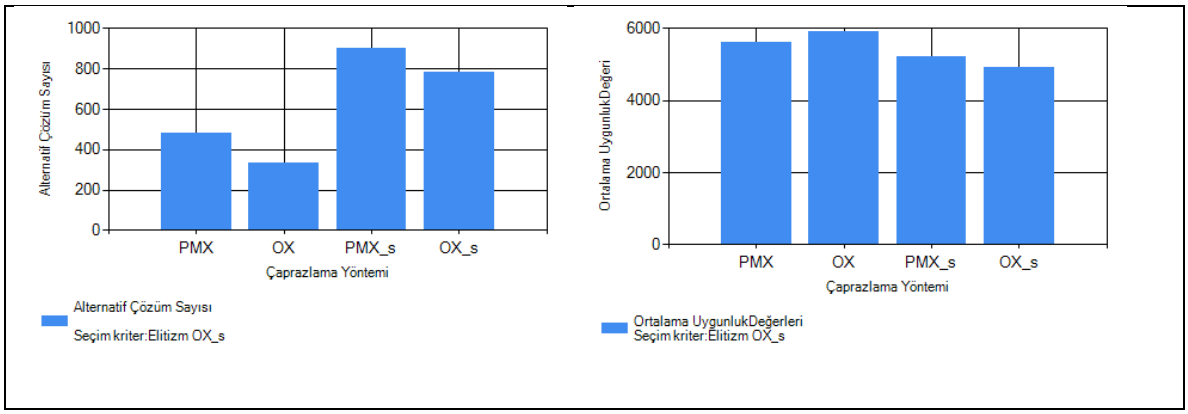
Hesaplama zamanının kısıtlı olduğu durumlarda uygulamanın uygulanabilir bir çözüm bulma süresi önem arz edeceğinden yöntemlerin ilk optimumu bulma süreleri de karşılaştırılmaktadır. Diğer grafiklerde bu kıstasın dikkate alınmamasının nedeni tam olarak tutarlılık göstermemesidir. Lakin burada yapılan tüm denemelerde OX-s'nin çok

kısa sürede çözüme ulaştığı tespit edilmiştir. Elitizm seçiminde kullanılan OX operatörünün, çaprazlama kriteri OX ve OX-s'de diğerlerine göre iyi sonuçlara ulaştığı söylenebilir.



Şekil 5.29. PMX-s çaprazlama operatörünü kullanan Elitizm seçiminde, çaprazlama operatörlerinin değerlendirilmesi

Elitizm seçim kriteri içinde PMX-s operatörünün kullanılarak, çaprazlama kriterlerinin değerlendirilmesinde bulunan çözümlerin ortalama uygunluk değerleri büyük farklılık göstermesede PMX-s'nin fazla sayıda sonuç elde etmesine rağmen Şekil 5.29'un ikinci grafiğine bakıldığında OX-s'nin daha iyi sonuçlara eriştiği söylenebilir. Aralarındaki farkın az olması diğerlerinde şube çakışmasının(örn.2.sınıf sınavlarının gün içerisinde birden fazla olması) daha fazla olduğu anlamına gelebilir.



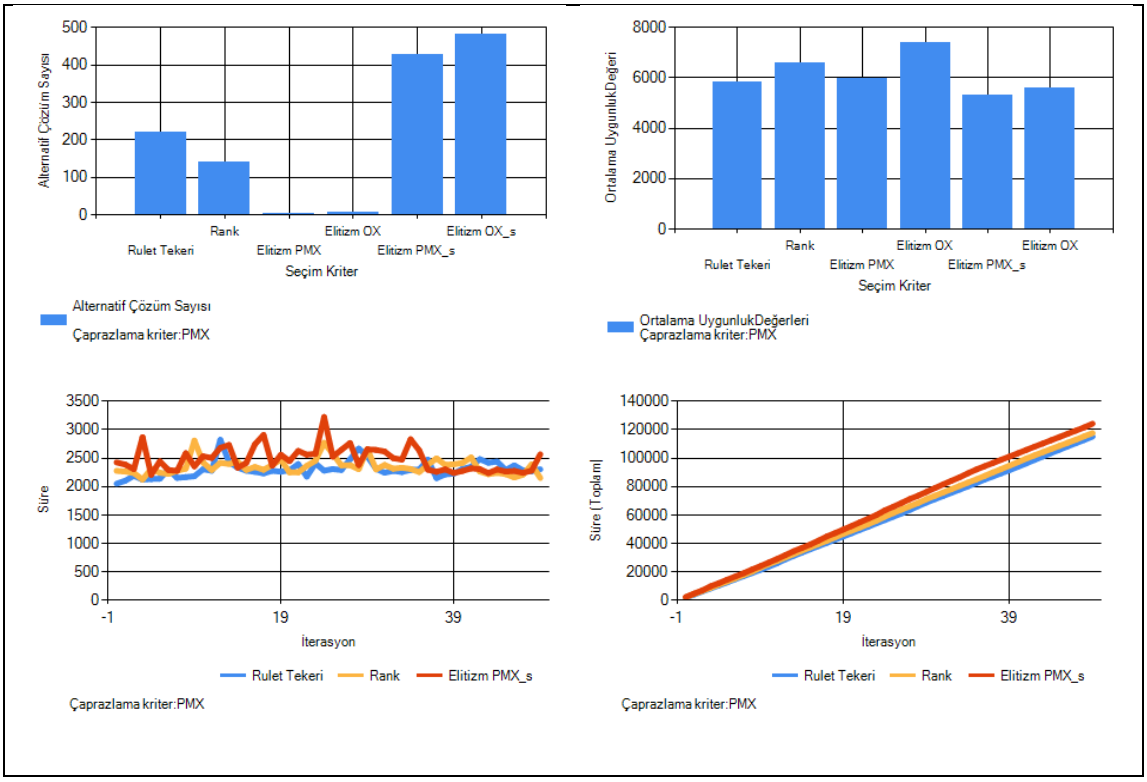
Şekil 5.30. OX-s çaprazlama operatörünü kullanan Elitizm seçiminde, çaprazlama operatörlerinin değerlendirilmesi

Şekil 5.30'da alternetif çözüm sayısı PMX-s'de daha fazla iken OX-s'nin elde ettiği çözümlerin uygulanabilirliği daha yüksektir. Elitizm seçimi işleminde PMX-s ve OX-s'den sonra uygulanan çaprazlama operatörlerinin ayrı ayrı eriştikleri çözüm sayısı

ve buldukları çözümlerin ortalama uygunluk değerleri birbirine oldukça yakın değerlerdir.

Elitizm seçiminde çaprazlama operatörlerini karşılaştırdığımızda, OX operatörünün ikinci seçime göre değerlendirildiği durumun (OX-s), çok sayıda sonuca ulaştığı ve diğerlerine göre daha uygulanabilir çözümler elde ettiği söylenebilir.

Seçim kriterlerine göre çaprazlama operatörlerinden hangisinin tercih edileceğine karar verilmesi kadar, kullanılan çaprazlama operatöründe hangi seçimlerin daha iyi sonuca ulaştığı bilgisi de önemlidir. Seçim kriterleri karşılaştırılarak elde edilen bu bilgi bize aynı zamanda elitizm seçiminde kullanılan çaprazlama operatörlerinin verimliliğini ölçme imkânı da vermektedir.

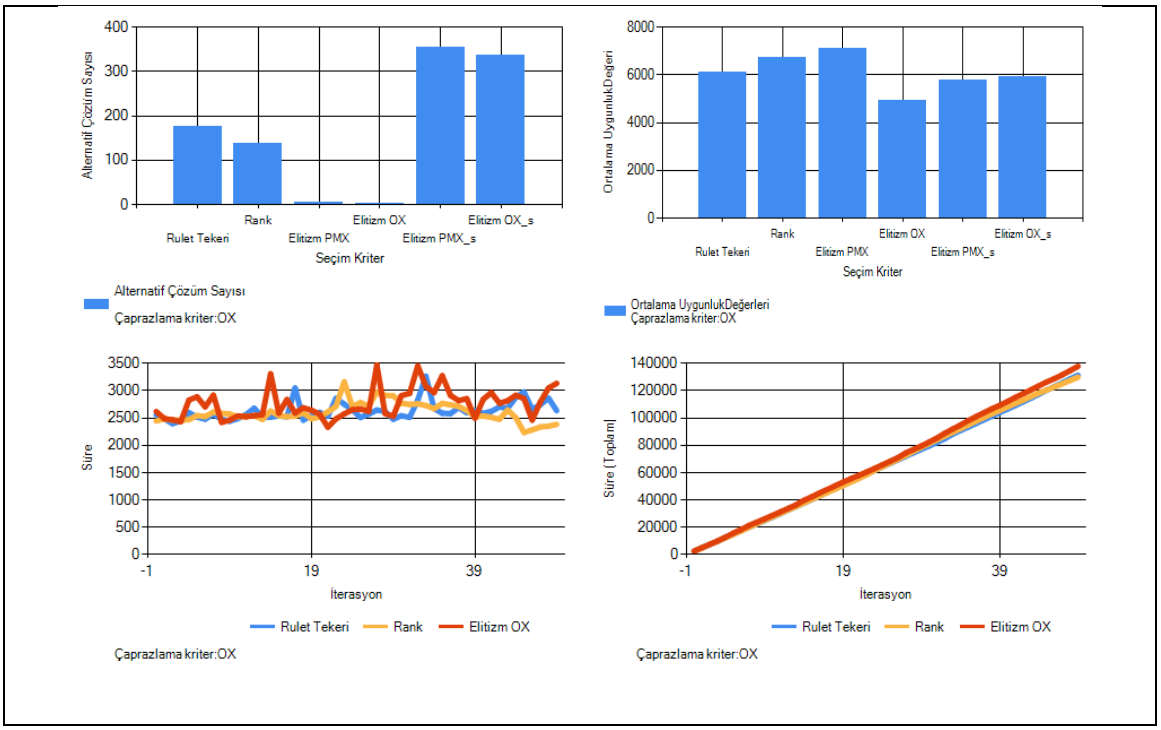


Şekil 5.31. PMX çaprazlama operatöründe seçim kriterlerinin değerlendirilmesi

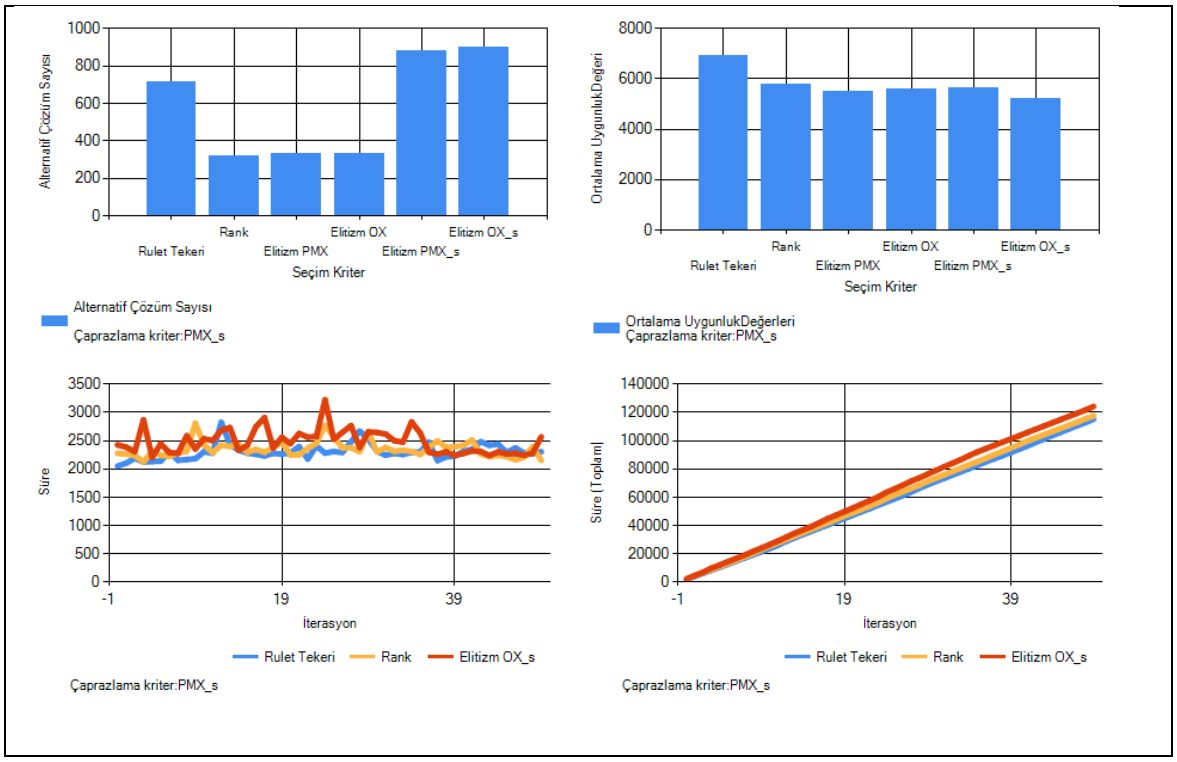
Şekil 5.31'de PMX çaprazlama kriterlerine göre değerlendirilen seçimlerde elitizm OX'in oldukça başarısız olduğu, elitizm PMX'in ise rank seçimine göre az sayıda çözüme erişmesine karşın ortalama uygunluk değerinin, istenilen değere göreli olarak daha yakın olduğu görülmektedir. En iyi sonuçlar, gerek bulduğu alternatif sayısının fazla olması gerekse diğerlerine göre elverişli çözümler elde etmesinden

dolayı elitizm PMX-s ve OX-s'de sağlanmaktadır. İterasyonların her birinde harcanan sürenin tespit edilmesinde rulet tekeri, rank seçimi ve elitizmde optimal sonuca yakınlaşmayı sağlayan operatör dikkate alınmıştır. Grafiklerdeki süre değerleri saniye zaman birimine göre hesaplanmıştır. Rulet tekeri ile rank seçiminin kısmen harcadıkları zaman birbirine yakın iken elitizm seçiminde 2 defa çaprazlama işlemi uygulanmasından dolayı biraz daha fazladır.

Elde ettiği sonuçların fazla olmamasına rağmen diğer kriterlere göre daha iyi çözüm değerlerine erişen elitizm OX'in, Şekil 5.28'e bakıldığında, çaprazlama operatörlerinin kıyaslanmasında OX operatörünün performansı ile örtüştüğü görülmektedir. Şekil 5.33'e göre, çaprazlama kriterinin OX olduğu durumlarda, elitizm seçiminin tercih edilmesi ve elitizm içinde OX veya PMX-s çaprazlamalarının kullanılmasının uygun olacağı görüşüne varılmaktadır.



Şekil 5.32. OX çaprazlama operatöründe seçim kriterlerinin değerlendirilmesi



Şekil 5.33. PMX-s çaprazlama operatöründe seçim kriterlerinin değerlendirilmesi

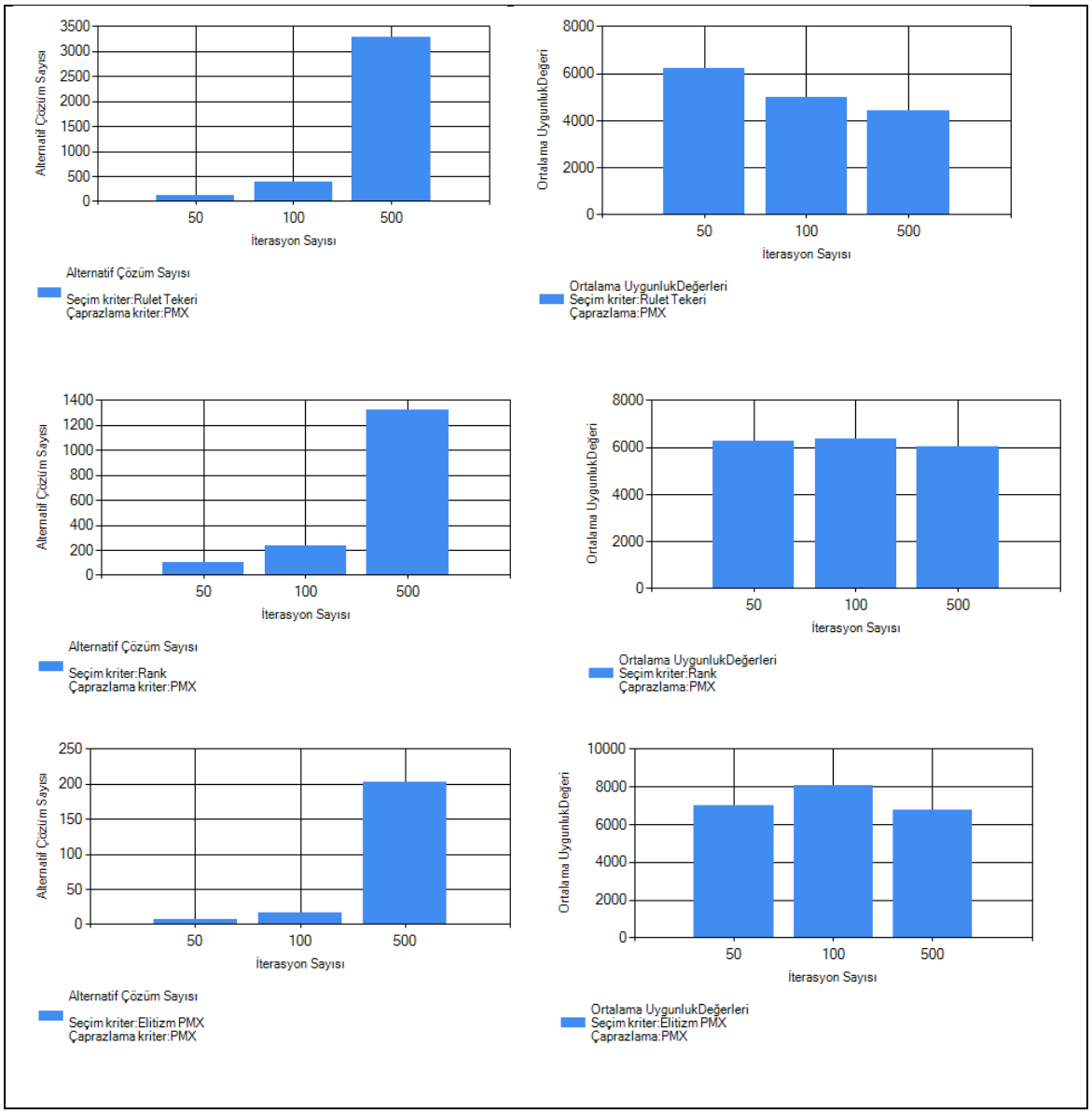
Şekil 5.33'te yer alan grafikler çaprazlama kriteri PMX-s olarak belirlendiğinde, kullanıcıya, elitizm seçimine karar vermesi ile beklenilen ihtiyaçları karşılayabilecek bir sınav programının elde edilmesinin mümkün olduğunu göstermektedir.

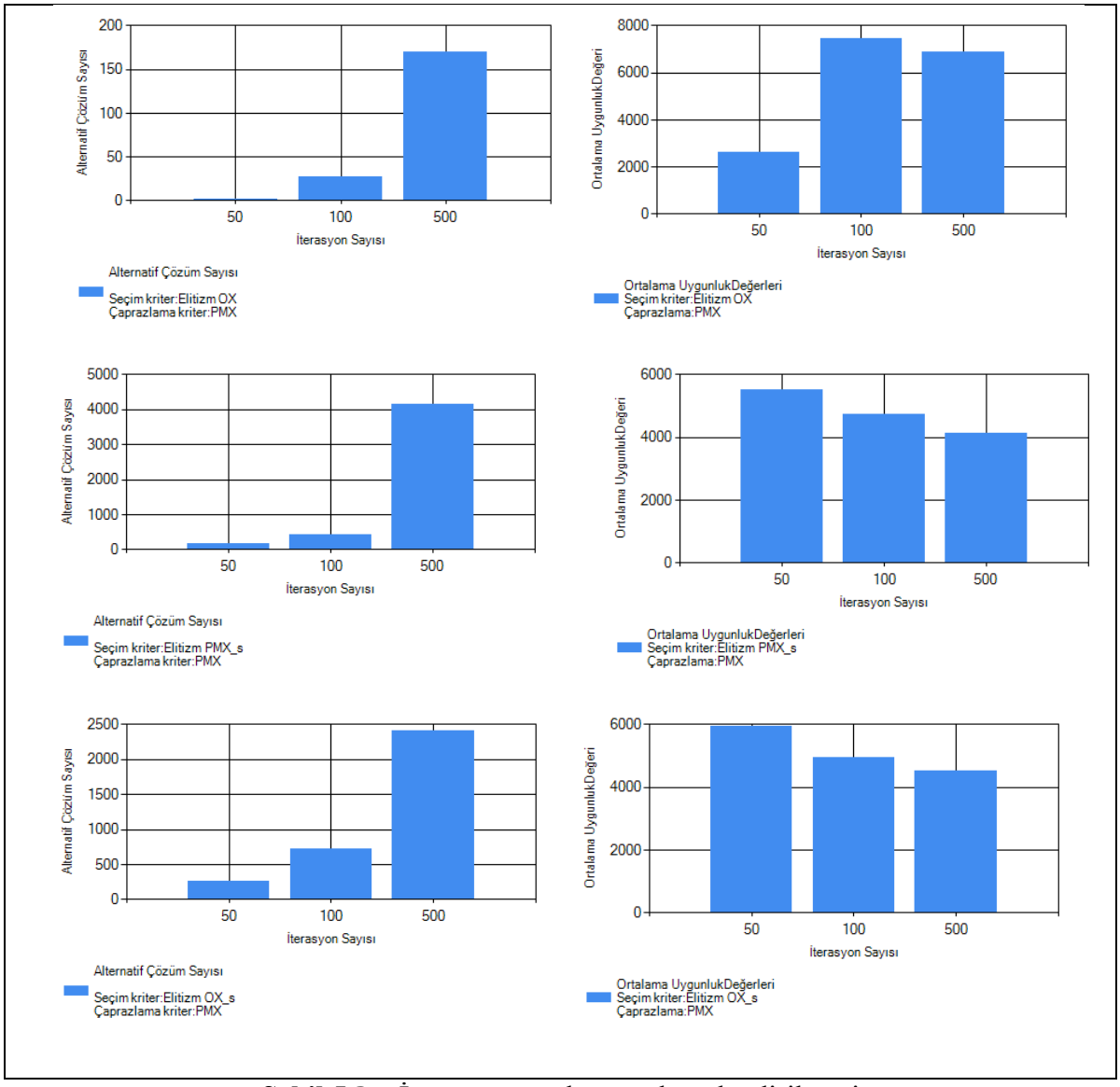


Şekil 5.34. OX-s çaprazlama operatöründe seçim kriterlerinin değerlendirilmesi

Genellikle elitizm seçiminde büyük ve uygun çözüm kümelerine erişilirken OX-s çaprazlamasında(Şekil 5.34) rank seçiminin tercih edilmesi ön plana çıkmaktadır.

PMX, OX, PMX-s ve OX-s operatörlerinin her birine göre yapılan seçim yöntemlerinin değerlendirilmesinde, alternatif çözüm sayısı dikkate alınsa da öncelikli olarak uygunluk değerleri önemsenir. Bulunulan çözüm sayısının fazla olması elbette arzu edilen sınav programının elde edilmesinin olasılığını artırmaktadır fakat bu durum uygunluk değerlerinin minimize olması ile desteklenmelidir.





Şekil 5.35. İterasyon sayılarının değerlendirilmesi

Programın durdurma kriterinin iterasyon sayısı olduğunu belirtmiştik. Bu tür problemlerde, ilk optimum değere veya belli bir sonuç sayısına ulaşıldığında problemi çözme sürecinin bitirilmesi birer seçenek olabilir. Ancak seçilen yöntemler rastsal olarak hareket ettikleri için hangisinde ne kadar çözüme erişelebileceğini tahmin etmek zordur. Örneğin fazla yapılan tahminler uygulamanın sonsuz döngüye girmesine neden olabilir. İlk optimum değerin bulunması ise tüm beklentilerin karşılandığı sınav programının sağlanması için yeterli olmayabilir. Sürecin sonlandırılması için iterasyon sayısının tercih edilmesinin nedeni, parametre olarak girilen bu değerin büyüklüğü fazla tutularak çözüm uzayının büyük bir kısmının taranmasını, küçük değerlerin tercih edilmesi ile de zamandan tasarruf edilmesini sağlamaktır. Her iki duruma da kısmen

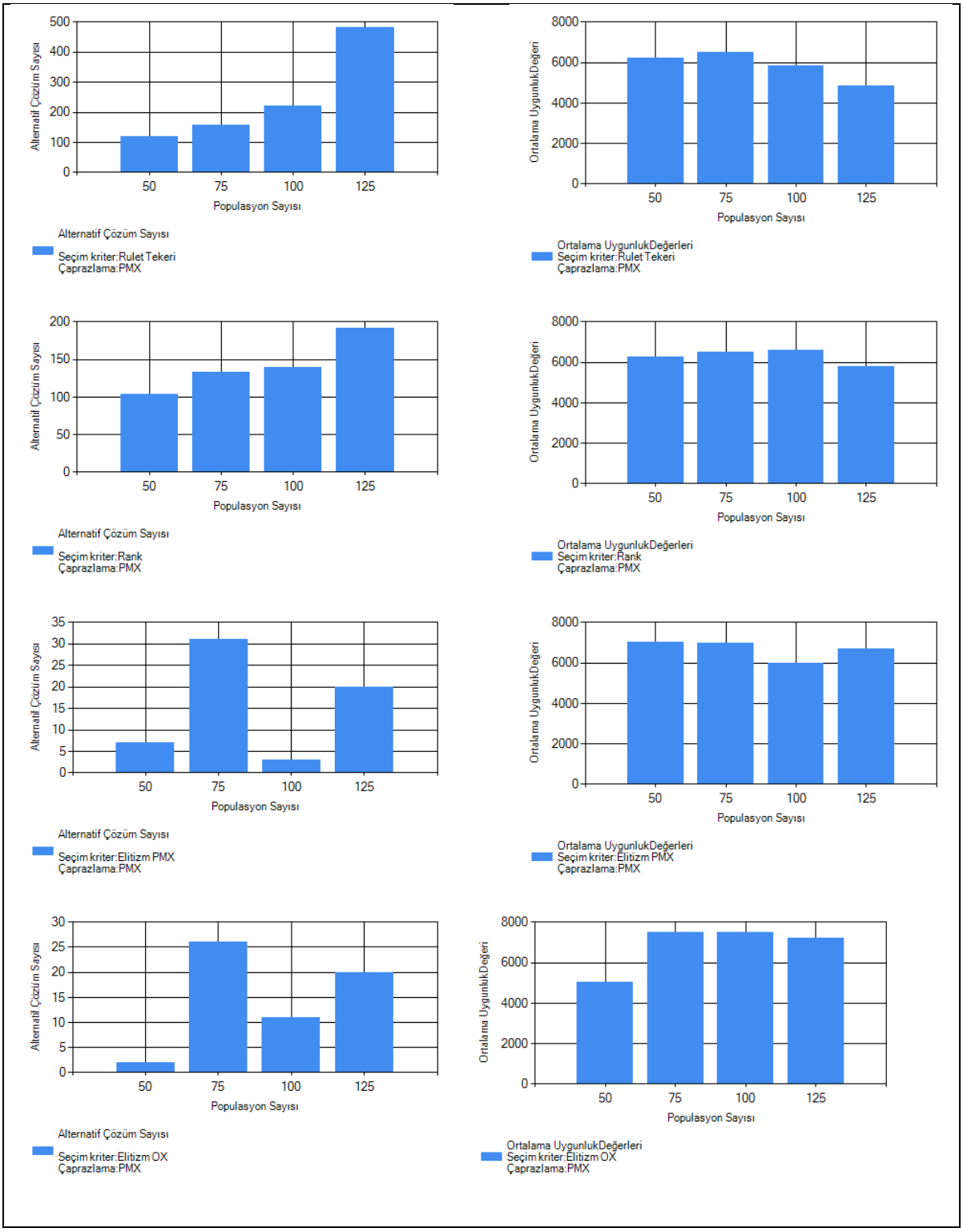
olanak tanıyan iterasyon sayısının belirlenmesi için Şekil 5.35'te görüldüğü gibi 50, 100 ve 500 değerleri test edilmiştir. Tahmin edildiği üzere, iterasyon sayısının fazla olduğu durumlarda çok fazla alternatif sonuç elde edilmiş ve çoğunda uygunluk değerlerinin ortalaması diğerlerine göre istenilene yakın çıkmıştır. Lakin süre kısıtının olduğu durumlarda, hesaplama sürecinin uzun sürdüğü 500 değeri optimal değildir. Nitekim seçim kriter elitizm OX ve çaprazlama kriterinin PMX olduğu durumda 50 iterasyon ile azımsanacak sayıda çözüm üretilmiş fakat diğerlerine göre aranılan çözüme daha yakın sınav programları elde edilmiştir.

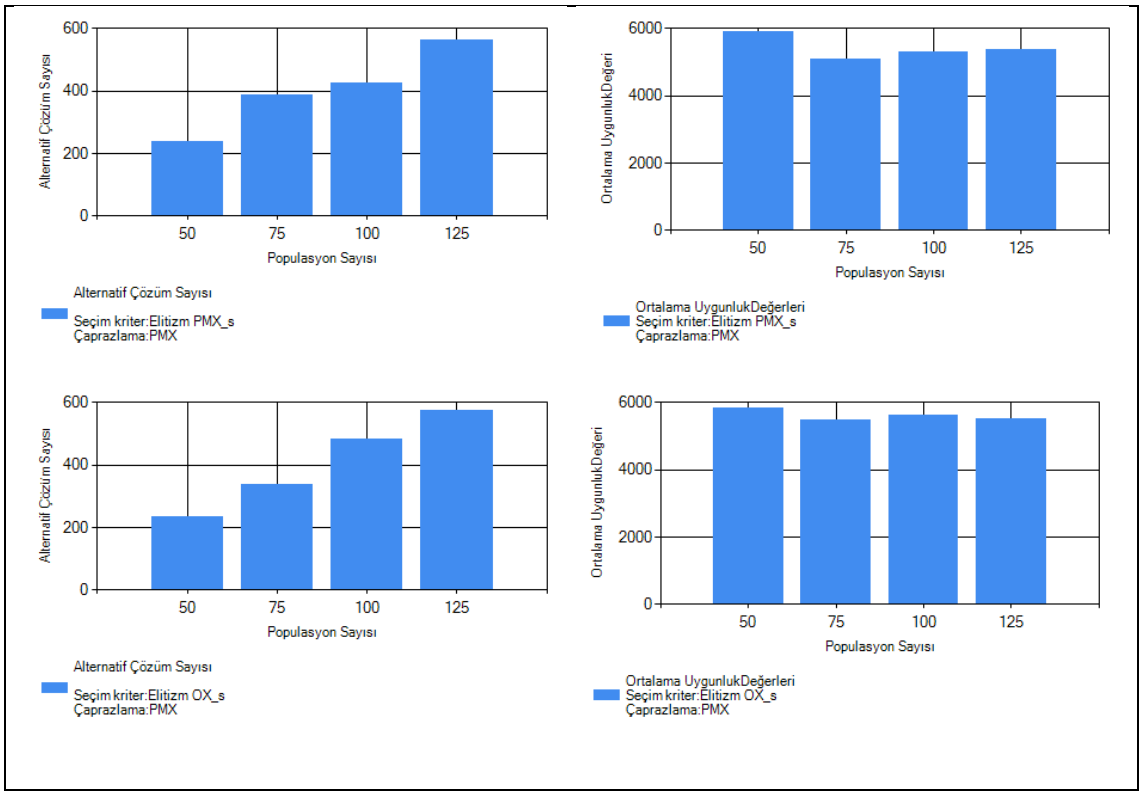
Genetik algoritmaların performansını etkileyen parametrelerden biri olan popülasyon büyüklüğüne doğru bir şekilde karar verilmesi önemlidir. Popülasyonun geniş tutulması çözüm uzayının daha büyük bir çerçevede aranmasını sağlarken, sürecin uzamasına neden olmaktadır. Az sayıda kromozomlardan oluşan popülasyonlarda ise hiçbir çözüme ulaşamadan istenmeyen durumlara yakınsama söz konusudur.

Şekil 5.36'da 50-75-100 ve 125 popülasyon büyüklüğü ile yapılan denemelerin karşılaştırılması verilmektedir. Genellikle kromozom sayısının 125 olduğu durumlarda bulunan çözüm sayısı artış göstermiştir. Elitizm PMX ve elitizm OX'te ise popülasyon büyüklüğünün 75 olduğu durum sonuç bulmada daha etkin gözükmektedir. Ancak elitizm PMX'te ortalama uygunluk değeri minimum olan çözümler (denemeler) 100 popülasyon büyüklüğünde elde edilirken elitizm OX'te 50 değerinde sağlanmaktadır.

Grafik değerlerine eklenmeyen fakat 25 popülasyon büyüklüğünde işlem yapan denemelerde çoğu zaman her hangi bir çözüme varılamamıştır. Popülasyonun her iterasyon sonrası takibi yapıldığında, kromozom çeşitliliğinin kısa zamanda azaldığı ve son olarak tek çözüm adayına indirgendiği gözlemlenmiştir.

Denemelerin yapıldığı popülasyon büyüklüğü değerlerinde kesin olarak bir seçim yapılması mümkün değildir. Seçim ve çaprazlama kriterlerinin türüne göre de değişkenlik gösteren alternatif çözüm sayıları ve ortalama uygunluk değerleri baz alındığında popülasyon büyüklüğü paramatresinin değeri 75 veya 125 olarak belirlenebilir.

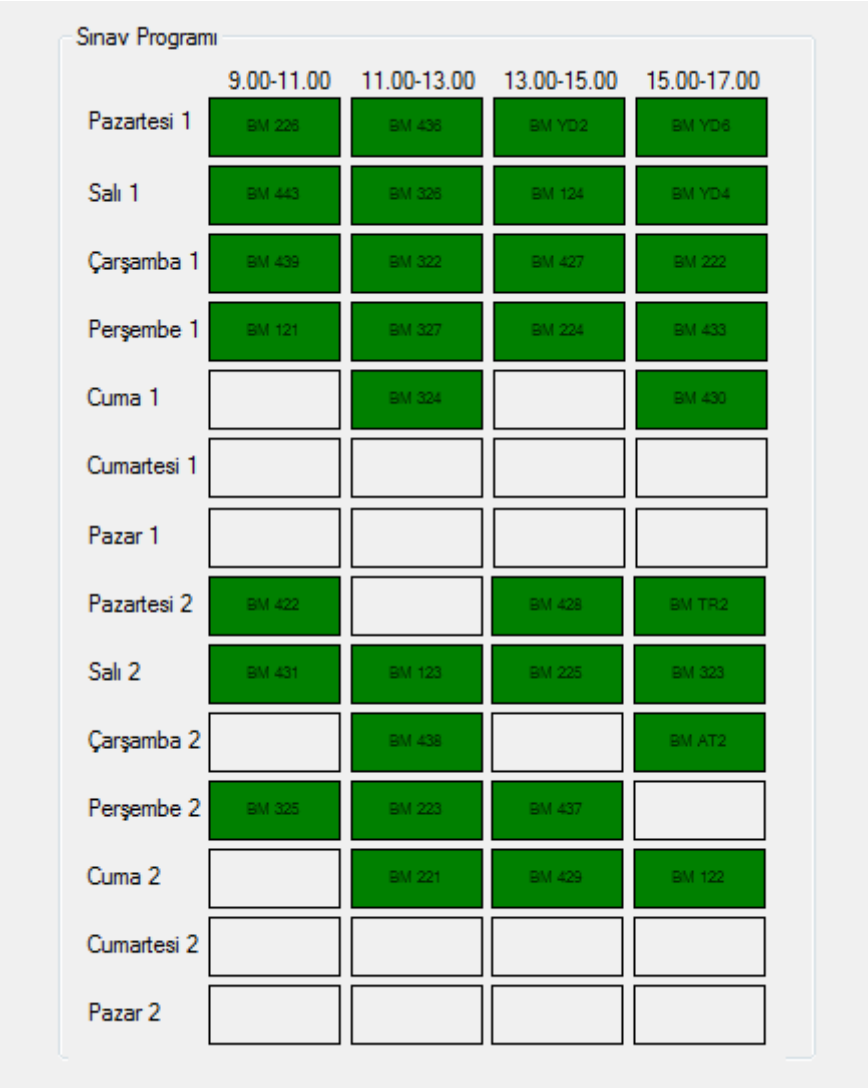

Alternatif Çözüm Sayısı
Populasyon Sayısı
Alternatif Çözüm Sayısı
Seçim kriter:Rulet Tekeri
Çaprazlama:PMX
Ortalama UygunlukDeğeri
Populasyon Sayısı
Ortalama UygunlukDeğerleri
Seçim kriter:Rulet Tekeri
Çaprazlama:PMX
Alternatif Çözüm Sayısı
Populasyon Sayısı
Alternatif Çözüm Sayısı
Seçim kriter:Rank
Çaprazlama:PMX
Ortalama UygunlukDeğeri
Populasyon Sayısı
Ortalama UygunlukDeğerleri
Seçim kriter:Rank
Çaprazlama:PMX
Alternatif Çözüm Sayısı
Populasyon Sayısı
Alternatif Çözüm Sayısı
Seçim kriter:Elitizm PMX
Çaprazlama:PMX
Ortalama UygunlukDeğeri
Populasyon Sayısı
Ortalama UygunlukDeğerleri
Seçim kriter:Elitizm PMX
Çaprazlama:PMX
Alternatif Çözüm Sayısı
Populasyon Sayısı
Alternatif Çözüm Sayısı
Seçim kriter:Elitizm OX
Çaprazlama:PMX
Ortalama UygunlukDeğeri
Populasyon Sayısı
Ortalama UygunlukDeğerleri
Seçim kriter:Elitizm OX
Çaprazlama:PMX



Şekil 5.36. Popülasyon sayılarının değerlendirilmesi

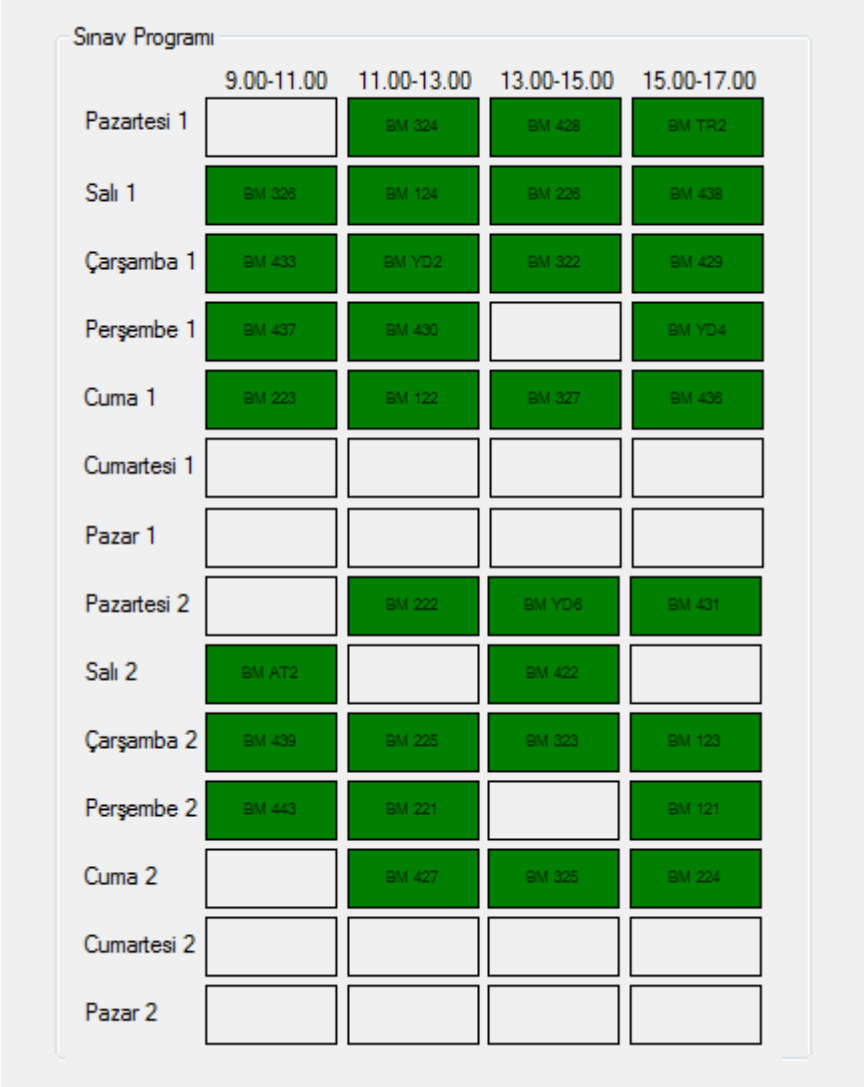
İterasyon ve popülasyon sayılarına göre yapılan değerlendirmeler, her bir operatöre göre test edildiğinde yalnızca çaprazlama operatörünün PMX olduğu durumda elde edilen grafikler gösterilmiştir. Diğer operatörlerde de, belirlenen değerler farklılık göstermemektedir.

Tüm olası durumların değerlendirilmesi yapıldıktan sonra esnek kısıtların sağlanma başarısını gözlemleyecek olursak, Şekil 5.37'de şube çakışmasının önlendiği bir sınav programı örneği gösterilmektedir. 4. Sınıfların dönem içerisinde seçmeli dersler ile birlikte toplam 12 dersi bulunmasından dolayı, 2 gün içerisinde 4 tane sınavları olması (tek günde 2 sınav) kaçınılmaz bir durumdur. Fakat en azından aynı gün içerisinde yapılan birden fazla sınav arasında yeterli boş zaman aralığı bırakılarak öğrenciler için dinlenme ve bir sonraki sınav notlarına bakabilme imkânı sunulmuştur.



Şekil 5.37. Şube çakışması önlenerek hazırlanan sınav programı

Bölümün, dönem içerisinde düzenli olarak tüm öğretim üyeleri ve elemanlarının katılım gösterdiği toplantılar yapması söz konusu olabilir. Bu durumda toplantının yapıldığı zaman aralığına her hangi bir sınav yerleştirilmesi uygun değildir. Örneğin, her pazartesi günü sabahında rutin olarak toplantı yapılması durumunda o günün sabahında her hangi bir sınavın olmaması gerekir. Bu durumun sağlandığı sınav programı Şekil 5.38'de gösterilmiştir.



Şekil 5.38. Toplantı saatlerinde sınav olmaması durumu

Öğretim üyeleri, vermekle yükümlü oldukları derslerin sınavlarının belli bir zaman aralığına yerleştirilmesini talep edebilir. Özellikle dışarıdan gelerek ders veren hocaların bu yöndeki istekleri daha fazla olabilmektedir. Esnek kısıtlar altında öğretim üyesi memnuniyeti olarak yer alan söz konusu durum, uygulamada göz önüne alınarak bu tarz taleplerin karşılanması sağlanmıştır. Örneğin, aynı hocanın verdiği Matematik II ve Fizik II derslerinin pazartesi günü ve 13.00-15.00 saatleri arasında sınavlarının yapılmasını istediği durum Şekil 5.39'da verilmiştir. BM 122 Fizik II'nin ilk hafta pazartesi ve BM 121 Matematik II'nin de yine ilk hafta pazartesi olduğu her iki durum da elde edilmiştir.



Şekil 5.39. Belirli bir zaman aralığına sınav yerleştirilmesi durumu

BÖLÜM 6

SONUÇLAR

Yarı-yapısal bir durum olan sınav çizelgeleme problemi, akademik kurumların her eğitim döneminde vize ve final sınavları olmak üzere en az 2 defa karşılaşılan bir sorundur. Zorunlu ve esnek kısıtlara sahip bu problemin kısıt kümesi büyüdükçe, problemin karmaşıklığı artmakta ve çözümü zorlaşmaktadır. Gerek öğrenci memnuniyetini sağlayan gerekse öğretim elemanlarının özel isteklerini karşılayan bir sınav programı talebi, uygulanabilir ve yüksek kaliteli çözümler üreten bir sistemin geliştirilmesi ihtiyacını doğurmuştur. Çalışmada, belirlenen ihtiyaçlara yanıt veren, karar vericiye birçok alternatif içerisinden en uygununu seçme imkânı tanıyan bir uygulama geliştirilmiştir.

Uygulamada; kesin bir sonuca ulaşmayı amaçlamak yerine elde edilen çözüm kümeleri içinde optimumu aramada etkin bir yöntem olan Genetik Algoritma'lar tercih edilmiştir. GA'lar ile, problemin, çözüm sürecinde istenmeyen durumlara takılmadan, tüm olası çözümlerin oluşturduğu uzayı tarayarak, optimal sonuca yakınsayan değerleri elde etmesi sağlanmış ve kullanıcıya sunulmuştur.

Genetik Algoritmalar'da kullanılan popülasyon büyüklüğü, çaprazlama oranı, mutasyon oranı gibi parametre değerleri ile kullanılan seçim ve çaprazlama kriterleri uygulamanın performansını doğrudan etkilediği için her birinin ayrı ayrı değerlendirilmesi yapılmıştır. İlk olarak mutasyon, çaprazlama ve elitizm oranlarının belirlenerek diğer denemelerin uygun parametre değerlerine göre test edilmesi sağlanmıştır. Çözüm dosyalarının karşılaştırılmasında %1 mutasyon, %70 çaprazlama ve %30 elitizm oranlarında daha uygun çözüm değerlerine erişildiği görülmüştür. Popülasyon büyüklüğü için tam bir yargıya varılamasa da, 75 ve 125 popülasyon büyüklüklerinin tercih edilebileceği belirtilmiştir. Alternatif çözüm sayısı, ulaşılan çözümlerin ortalama uygunluk değerleri, her iterasyon sonucunda bulunan çözüm

sayısı, ilk optimum değerin elde edilme süresi ve harcanan toplam zamana göre kıyaslanan yöntemlerde, öncelikle ortalama uygunluk değerleri dikkate alınmıştır. Hesaplama süresinin çaprazlama operatörlerinde farklılık göstermediği, seçim kriterlerinde ise elitizm seçiminde ardı ardına çaprazlama uygulanmasından kaynaklanan bir süre farkı olduğu tespit edilmiştir. Her iterasyon sonucunda erişilen çözüm sayıları iterasyon sayısının belirlenmesinde kısmen yardımcı olmaktadır. Kullanıcı bu grafiğe bakarak ortalama kaçıncı iterasyondan sonra programın çözüm bulma performansını arttığını gözlemleyebilir. Tabi aynı zamanda bulunan çözümlerin uygunluğunun da test edilmesi gerekir. İlk optimum değeri bulma süresi, rastgele oluşturulan başlangıç popülasyonunun çözüme olan yakınlığı ile doğrudan ilişkili olduğu için yapılan denemelerde bu kriterle ilgili bir kanıya varılamamıştır. Ağırlıklı olarak alternatif çözüm sayıları ve ortalama uygunluk değerleri göz önünde bulundurularak değerlendirilen kombinasyonlar sonucunda;

Tablo 6.1. Seçim kriterlerine uygun çaprazlamaların belirlenmesi

Seçim kriteri	**Elitizmde kullanılan çaprazlama operatörü**	**Çaprazlama operatörü**
Rulet tekeri	-	PMX
Rank seçimi	-	OX-s veya PMX-s
Elitizm	PMX	PMX-s veya OX-s
Elitizm	OX	OX
Elitizm	PMX-s	OX-s
Elitizm	OX-s	OX-s

Tablo 6.2. Çaprazlama operatörlerine uygun seçimlerin belirlenmesi

Çaprazlama operatörü	**Seçim kriteri**	**Elitizmde kullanılan çaprazlama operatörü**
PMX	Elitizm	PMX-s
OX	Elitizm	OX
PMX-s	Elitizm	OX-s
OX-s	Rank	-

Yukarıdaki tablolar oluşturulmuştur. Tablo 6.1 tercih edilen seçim kriterinde, çözümün etkinliğinin artmasını sağlayacak çaprazlama operatörü bilgisini, Tablo 6.2 ise uygulanan çaprazlama operatöründe hangi seçim kriterinin tercih edilmesinin uygun olacağı bilgisini içermektedir.

Birden fazla alternatif çözüm içinden kullanıcının en iyisini, en uygununu seçme olanağı tanıyan sistem, karmaşık ve zor olarak nitelendirilen sınav çizelgesi hazırlama problemine çözüm bulmakta ve sadece Trakya Üniversitesi Bilgisayar Mühendisliği bölümü için değil veri tabanı bilgilerinin güncellenerek diğer bölümler içinde uygulanmasını mümkün kılmaktadır.

KAYNAKLAR

1. Ambati, B.K., Ambati, J. & Mokhtar, M.M., "Heuristic Combinatorial Optimization by Simulated Darwinian Evolution: A Polynomial Time Algorithm for the Traveling Salesman Problem", Biological Cybernetics 65: 31–35, (1991).
2. Arslan, V., Yılmaz, G., "Karar Destek Sistemlerinin Kullanımı İçin Uygun Bir Model Geliştirilmesi", Havacılık ve Uzay Teknolojileri Dergisi, Cilt 4, Sayı 4, 20 Temmuz, (2010).
3. Atanak, M.M., Hocaoğlu, F.O., "Genetik Algoritmalarla Ders Programı Hazırlama Otomasyonu Tasarımı", III. Otomasyon Sempozyumu ve Sergisi Bildirileri, Denizli, (2005).
4. Bağış, A., "Genetik Algoritma Kullanılarak Ders Programının Optimum Şekilde Düzenlenmesi", Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, (1996).
5. Banzhaf, W., "The "Molecular" Traveling Salesman", Biological Cybernetics 64: 7–14, (1990).
6. Bardadym, V.A., "Computer-aided school and university timetabling: The new wave". In: E.K. Burke and P. Ross (eds) (1996). Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference. Lecture Notes in Computer Science, vol. 1153. 22-45, (1996).
7. Baykal, N., Beyan, T., Bulanık Mantık Uzman Sistemler ve Denetleyiciler, Ankara: Bıçaklar Kitapevi, (2004).
8. Bergquist, H.W., McLean, E.R., "Integrated Data Analysis and Management Systems: An APL-Based Decision Support System," in: Sol, H.G. (eds.), Processes and Tools for Decision Support, North-Holland, Amsterdam, (1983).
9. Birbas, T., Daskalaki, S., & Housos, E., "School timetabling for quality student and teacher schedules", Journal of Scheduling, 12, 177-197, (2009).
10. Bonczek, R.H., Holsapple, C.W., and Whinston, A.B., Foundations of Decision Support Systems. Academic Press, (1981).
11. Bowerman, Bruce L, O'Connell: Forecasting and Time Series An Applied Approach, 3rd Edition, Wadsworth Inc., California, (1993).

12. Brady, R.M., “Optimization Strategies Gleaned from Biological Evolution”, Nature 317: 804–806, (1985).
13. Burke, E., Elliman, D., Weare, R., “A Genetic Algorithm Based University Timetabling System”, East-West Conference on Computer Technologies in Education, Crimea, Ukraine, (1994).
14. Burke, E.K. and Carter, M.W. (eds.), “Practice and Theory of Automated Timetabling”, Selected Papers from the 2nd International Conference, Lecture Notes in Computer Science, vol. 1408. ISBN 3-540-64979-4, (1998).
15. Burke, E.K. and De Causmaecker, P. (eds.), “Practice and Theory of Automated Timetabling”, Selected Papers from the 4th International Conference, Lecture Notes in Computer Science, vol. 2740. ISBN 3-540-40699-9, (2003).
16. Burke, E.K. and Erben, W. (eds), “Practice and Theory of Automated Timetabling”, Selected Papers from the 3rd International Conference, Lecture Notes in Computer Science, vol. 2079. ISBN 3-540-42421-0, (2001).
17. Burke, E.K. and Landa Silva, J.D., “The design of memetic algorithms for scheduling and timetabling problems”, in W.E. Hart, N. Krasnogor and J.E. Smith (eds.), Recent Advances in Memetic Algorithms and Related Search Technologies, Studies in Fuzziness and Soft Computing 166, Springer, Berlin, Heidelberg, NewYork. 289-312, (2004).
18. Burke, E.K. and Ross, P. (eds), “Practice and Theory of Automated Timetabling”, Selected Papers from the 1st International Conference, Lecture Notes in Computer Science, vol. 1153. ISBN 3-540-61794-9, (1996).
19. Burke, E.K. and Trick, M. (eds), “Practice and Theory of Automated Timetabling”, Selected Papers from the 5th International Conference, Lecture Notes in Computer Science, vol. 3616. ISBN 3-540-30705-2, (2005).
20. Burke, E.K., & Petrovic, S., “Recent research directions in automated timetabling”, European Journal of Operational Research, 140, 266-280, (2002).
21. Burke, E.K., Bykov, Y., Newall, J.P. and Petrovic, S., “A time-prede ned local search approach to exam timetabling problems”, IIE Transactions, 36(6): 509-528, (2004).
22. Burke, E.K., Elliman, D.G., Ford, P.H. and Weare, R.F., “Examination timetabling in British universities: A survey”, in E.K. Burke and P. Ross (eds).

Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference, Lecture Notes in Computer Science, vol. 1153. 76-90, (1996).

23. Burke, E.K., Kingston, J.H. and DeWerra, D., "Applications to timetabling", in: J. Gross and J. Yellen (eds.) The Handbook of Graph Theory, Chapman Hall/CRC Press, 2004, 445-474, (2004).
24. Burke, E.K., McCollum, B., Meisels, A., Petrovic, S., & Qu, R., "A graph-based hyper-heuristic for educational timetabling problems", European Journal of Operational Research, 176, 177-192, (2007).
25. Burke, E.K., Qu, R. ve Soghier, A., "Adaptive Selection of Heuristics for Improving Constructed Exam Timetables", Annals of Opreration Research, 25 June, (2012).
26. Carter, M.W., "A survey of practical applications of examination timetabling algorithms", Operations Research, 34(2): 193-202, (1986).
27. Carter, M.W., Laporte, G., Lee, S.Y., "Examination timetabling: Algorithmic strategies and applications", The Journal of the Operational Research Society, 47, 373-383, (1996).
28. Chang, C., Melamud, Y., and Seabrook, D., Expert Systems, The Butler Cox Foundation, Report Series no. 37, (1983).
29. Çayıroğlu, İ., Dizdar, E.N., "Uzman Sistem Destekli Online Ders Yerleştirme Programı", Teknoloji, Cilt 9, Sayı 4, (2006).
30. Daban, F., Özdemir, E., "Çok Parametreli Genetik Algoritma Kullanarak Ders Programlarının Hazırlanması", II. Elektrik Elektronik Bilgisayar Mühendislikleri Eğitim Sempozyumu Bildirileri, Samsun, (2005).
31. Daban, F., Özdemir, E., "Eğitimde Verimliliği Arttıran Ders Programlarının Hazırlanması için Genetik Algoritma Kullanımı", Eğitim Bilimleri ve Uygulama,3,(6), 245-257, (2004).
32. Davis L., "Handbook of Genetic Algorithms", Van Nostrand Reinhold, (1991).
33. Davis, L., "Applying Adaptive Algorithms to Epistatic Domains", Proceedings of the International Joint Conference on Artificial Intelligence, 162–164, (1985).
34. De Werra, D., "An introduction to timetabling", European Journal of Operational Research, 19: 151-162, (1985).

35. De Werra, D., “Restricted coloring models for timetabling”, Discrete Mathematics, 165/166: 161-170, (1997).
36. Demir, H., Gümüşoğlu, Ş., Üretim Yönetimi (İşlemler Yönetimi), Beta Basım Yayım Dağıtım, İSTANBUL, (1998).
37. Elam, J.J., Henderson, J. C., Keen, P. G. W., Konsynski, B. , "A Vision for DSS Research," MIS Quarterly, (1986).
38. Emel, G.G., Taşkın, Ç., “Genetik Algoritmalar ve Uygulama Alanları”, Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, Cilt XXI, Sayı 1, 129-152, (2002).
39. Eren, E., “Yönetim ve Organizasyon (Çağdaş ve Küresel Yaklaşımlar)”, Beta Yayınları, 5.basım, İstanbul, (2001).
40. Fogel, D.B., “A Parallel Processing Approach to a Multiple Traveling Salesman Problem Using Evolutionary Programming”, in Canter, L. (eds.) Proceedings on the Fourth Annual Parallel Processing Symposium, 318–326, Fullterton, CA, (1990).
41. Fogel, D.B., “An Evolutionary Approach to the Traveling Salesman Problem”, Biological Cybernetics 60: 139–144, (1988).
42. Gasimov, R., “Karar Analizi”, Başkent Üniversitesi, Ders Notları, Ankara, (2003).
43. Gen, M., Cheng, R., Genetic Algorithms and Engineering Design, A Wiley-Interscience Publication, JOHN WILEY & SONS.INC, (1997).
44. Ginzberg, M.J., Stohr, E.A., "Decision Support Systems: Issues and Perspectives", in: Ginzberg, M.J., Reitman, W., Stohr., E.A. (eds.), Decision Support Systems. North-Holland, Amsterdam, (1982).
45. Goldberg, D.E. & Lingle, Jr., R., “Alleles, Loci and the TSP”, in Grefenstette, J. J. (eds.) Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 154–159. Hillsdale, New Jersey: Lawrence Erlbaum, (1985).
46. Gökçen, H., “Yönetim Bilgi Sistemleri”, Palme Yayıncılık, Ankara, (2007).
47. Grefenstette, J.J., “Incorporating Problem Specific Knowledge into Genetic Algorithms”, in Davis, L. (eds.) Genetic Algorithms and Simulated Annealing, 42–60. Los Altos, CA: Morgan Kaufmann, (1987b).

48. Gülcü, A., “Yapay Zeka Tekniklerinden Genetik Algoritma ve Tabu Arama Yöntemlerinin Eğitim Kurumlarının Haftalık Ders Programlarının Hazırlanmasında Kullanımı”, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, (2006).
49. Harmanşah, C., Seylan, N., Sözeri, V., “Web Tabanlı Derslik Yerleştirme Sistemi”, XVI. Türkiye’de İnternet Konferansı, İzmir, (2011).
50. Holland, J., “Adaptation in Natural and Artificial Systems”, Ann Arbor: University of Michigan Press, (1975).
51. http://ahmet.tumay.etu.edu.tr/wordpress/wp-content/uploads/2010/01/BIL576-Lesson01-Introduction-RLSD-100109.ppt (2012).
52. http://tr.wikipedia.org/wiki/Genetik_algoritma (2012)
53. İlkuçar, M., Aslantaş, A., Kavurur, A., "Sınav Yerleşim ve Gözetmenlik Dağılımı Uygulama Yazılımı", 2. Uluslararası Meslek Yüksekokulları Sempozyumu, 25-27 Mayıs, Aydın, (2011).
54. Kalaycı, C.B., Güngör, A., “A Genetic Algorithm Based Examination Timetabling Model Focussing on Student Success for the Case of the College of Engineering at Pamukkale University, Turkey”, Gazi University Journal of Science GUJ Sci, 25(1), 137-153, (2012).
55. Keen, P.G.W., Morton, S., “Decision Support Systems: An Organizational Perspective.Reading”, MA: Addison-Wesley, (1978).
56. Kingston, J.H., “Bibliography on practice and theory of automated timetabling”, http://liinwww.ira.uka.de/bibliography/Misc/timetabling.html, (1995).
57. Klein, H.K., and Hirschheim, R., "Consequentialist Perspective of Decision Support Systems," Decision Support Systems. Vol. 1, no. 1, (1985).
58. Klein, M., and Manteau, A., "OPTRANS: A Tool for Implementation of Decision Support Centers," in: Sol, H.G. (eds.) Processes and Tools for Decision Support, North-Holland, Amsterdam, (1983).
59. Klein, M., Methlie, L.B., Expert systems: a decision support approach. Reading: Addison-Wesley, (1990).
60. Kuruüzüm, A., “Karar Destek Sistemlerinde Çok Amaçlı Yöntemler”, Akdeniz Üniversitesi Basımevi, Antalya, (1998).

61. Landa Silva, J.D., Burke, E.K. and Petrovic, S., “An introduction to multiobjective meta-heuristics for scheduling and timetabling”, in X. Gandibleux, M. Sevaux, K. Sorensen and V. Tkindt (eds): Multiple Objective Meta-heuristics, Lecture Notes in economics and mathematical systems, vol. 535. 91-129, (2004).
62. Larranaga, P., Kuijpers, C., Murga, R., Inza, I. and Dizdarevic, S., “Genetic Algorithms for the Traveling Salesman Problem: A Review of Representations and Operators.” Artificial Intelligence Review, 13, pp. 129-170, (1999).
63. Lee, R.M., "Epistemological Aspects of Knowledge-Based Decision Support Systems," in: Sol, H.G. (eds.), Processes and Tools for Decision Support, North-Holland, Amsterdam, (1983).
64. Little, J.D.C., "Models and Managers: The Concept of a Decision Calculus", Management Science, 16(8), 35-43, (1970).
65. Long, L., “Management Information Systems”, Prentice Hall, U.S.A., (1989).
66. Marakas, G.M., “Decision Support Systems”, Prentice Hall, New Jersey, (1999).
67. McCollum, B.G.C., “A perspective on bridging the gap between theory and practice in university timetabling”, in E.K. Burke and H. Rudova (eds.) (2007) Practice and Theory of Automated Timetabling: Selected Papers from the 6th International Conference, Lecture Notes in Computer Science, vol. 3867, 3-23, (2007).
68. Michalewicz, Z., “Genetic Algorithms + Data Structures = Evolution Programs”, Berlin Heidelberg: Springer Verlag, (1992).
69. Miles, R., “Computer timetabling: A bibliography”, British Journal of Educational Technology, 6(3): 16-20, (1975).
70. MirHassani, S.A., “Improving paper spread in examination timetables using integer programming”, Applied Mathematics and Computation, 179, 702-706, (2006).
71. Mühlenbein, H., “Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization”, in Schaffer, J. (eds.) Proceedings on the Third International Conference on Genetic Algorithms, 416–421. Los Altos, CA: Morgan Kaufmann Publishers, (1989).

72. Mühlenbein, H., GorgesSchleuter, M. & Kramer, O., "Evolution Algorithms in Combinatorial Optimization", Parallel Computing 7: 65–85, (1988).
73. Nabiyev, V.V., Yapay Zeka (İnsan-Bilgisayar Etkileşimi), Seçkin Yayıncılık, 3.baskı, ANKARA, (2010).
74. Oliver, I.M., Smith, D.J. & Holland, J.R.C., "A Study of Permutation Crossover Operators on the TSP", in Grefenstette, J. J. (eds.) Genetic Algorithms and Their Applications:Proceedings of the Second International Conference, 224–230. Hillsdale, New Jersey: Lawrence Erlbaum, (1987).
75. Pendse, N., The OLAP report: What is OLAP?, http://www.olapreport.com/fashi.htm, http://www.bi-verdict.com/
76. Qu, R., Burke, E., McCollum, B., Merlot, L., & Lee, S., "A survey of search methodologies and automated system development for examination timetabling", Journal of Scheduling, 12, 55-89, (2009).
77. Qu, R., He, F., Burke, E.K., "Hybridizing Integer Programming Models with an Adaptive Decomposition Approach for Exam Timetabling Problems", Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2009), 10-12 August, Dublin, İreland, (2009).
78. Saraçoğlu D., "Web Tabanlı Bir Genetik Algoritma Öğreticisi", (2009).
79. Sauter, V. L., "Decision Support Systems: An Applied Managerial Approach", John Wiley & Sons INC, (1997).
80. Schaerf, A., "A survey of automated timetabling", Articial Intelligence Review, 13(2): 87-127, (1999).
81. Schaerf, A., Di Gaspero, L., "Measurability and reproducibility in university timetabling research: discussion and proposals", in E.K. Burke and H. Rudova (eds) (2007), Practice and Theory of Automated Timetabling: Selected Papers from the 6th International Conference, Lecture Notes in Computer Science, vol. 3867, 40-49, (2007).
82. Schmidt, E.A. and Strohlein, T., "Timetable construction - An annotated bibliography", The Computer Journal, 23: 307-316, (1979).
83. Sheikh, R.A., Lanjewar, U.A., "Decision Support System for Cotton Bales Blending Using Genetic Algorithm", International Journal of Computer Applications (0975 – 8887), Volume 8– No.1, October (2010).

84. Sprague, R.H., "A Framework for Research on Decision Support Systems," in: Fick, G., and Sprague, R.H. (eds.) Decision Support Systems; Issues and Challenges, Pergamon Press, Oxford, (1980).
85. Sprague. R.H.. and Carlson, E.D., Building Effective Decision Support Systems. Prentice Hall, Englewood Cliffs, (1982).
86. Stamper, R., "Management Epistemology: Garbage In, Garbage Out," in: Methlie, L.B., and Sprague, R.H., (eds.), Knowledge Representation for Decision Support, North-Holland, Amsterdam, (1985).
87. Syswerda, G., "Schedule Optimization Using Genetic Algorithms",in Davis, L. (eds.) Handbook of Genetic Algorithms, 332–349. New York: Van Nostrand Reinhold, (1991).
88. Taç, K.C., "Genetik Algoritma Kullanılarak Haftalık Ders Programı Zaman Çizelgeleme Yazılımının Geliştirilmesi", Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, (2006).
89. TBD Kamu-BİB Kamu Bilişim Platformu XII "Kamuda Karar Destek Sistemlerinin Kullanımı ve Bir Model Önerisi" TBD Kamu-BİB Kamu Bilişim Platformu 12, 13-16 Mayıs, Antalya, (2010).
90. Tekin, İ., "İşletmelerin Karar Verme Düzeylerinde Stratejik Planlamanın Yeri ve Ticari Bankalarda Uygulanırlığı Üzerine Bir Araştırma", Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, (2009).
91. Turban, E., "Decision Support and Expert Systems: Management Support Systems", Macmillan Publishing Company , Third Edition, New York, (1993).
92. Turban, E., Aronson, J.E., Decision Support Systems and Intelligent Systems, Prentice Hall, Upper Saddle River, NJ, (2001).
93. Whitley, D., Starkweather, T. & D'Ann Fuquay, "Scheduling Problems and Travelling Salesman: The Genetic Edge Recombination Operator" in Schaffer, J. (eds.) Proceedings on the Third International Conference on Genetic Algorithms, 133–140. Los Altos, CA: Morgan Kaufmann Publishers, (1989).
94. Wren, A., "Scheduling, timetabling and rostering - A special relationship?" in E.K. Burke and P. Ross (eds) (1996). Practice and Theory of Automated Timetabling: Selected Papers from the 1st International Conference, Lecture Notes in Computer Science, vol. 1153. 46-75, (1996).

95. Yesevi, Ramazan, http://www.belgeler.com/blg/2be7/tbil-655-karar-destek-sistemleri-1, (2012).
96. Yılmaz, E, "Analitik Hiyerarşi Süreci Kullanılarak Çok Kriterli Karar Verme Problemlerinin Çözümü", Doğu Akdeniz Ormancılık Araştırma Enstitüsü, Yüksek Lisans Tezi, (1999).

ÖZGEÇMİŞ

Nazan Demirci 1986 yılında Bartın'da doğdu. 2008 yılında Maltepe Üniversitesi Bilgisayar Mühendisliği ve Matematik bölümlerinden (çift lisans) mezun olarak lisans eğitimini tamamladı. Aynı yıl içerisinde Hacettepe Üniversitesi Ekonomi bölümü yüksek lisans eğitimine başladı. Fakat 2010 yılının Ocak ayında Trakya Üniversitesi Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi kadrosuna hak kazanınca Hacettepe Üniversitesi'ndeki eğitimini yarıda bırakıp, burada kendi alanında yüksek lisans eğitimine başladı. Ve halen Trakya Üniversitesi'nde araştırma görevliliğini sürdürmektedir.