

Eđitimsel Zaman izelgeleme Problemleri İin özüm Yaklařımları ve Web Tabanlı  
Bir Karar Destek Sistemi Önerisi

Zehra Kamıřlı Öztürk

**DOKTORA TEZİ**

Endüstri Mühendisliđi Anabilim Dalı

Haziran 2010

Solution Approaches for Educational Timetabling Problems and a Web Based Decision  
Support System Proposal

Zehra Kamışlı Öztürk

**DOCTORAL DISSERTATION**

Department of Industrial Engineering

June 2010

Eđitimsel Zaman izelgeleme Problemleri İin özüm Yaklařımları ve Web Tabanlı  
Bir Karar Destek Sistemi Önerisi

Zehra Kamıřlı Öztürk

Eskiřehir Osmangazi Üniversitesi  
Fen Bilimleri Enstitüsü  
Lisansüstü Yönetmeliđi Uyarınca  
Endüstri Mühendisliđi Anabilim Dalı  
Yöneylem Arařtırması Bilim Dalında  
DOKTORA TEZİ  
Olarak Hazırlanmıřtır

Danıřman: Do. Dr. Müjgan Sađır Özdemir

Haziran 2010

## ONAY

Endüstri Mühendisliği Anabilim Dalı Doktora öğrencisi Zehra Kamışlı Öztürk'ün DOKTORA tezi olarak hazırladığı “Eğitimsel Zaman Çizelgeleme Problemleri için Çözüm Yaklaşımları ve Web Tabanlı Bir Karar Destek Sistemi Önerisi” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

**Danışman** : Doç. Dr. Müjgan Sağır Özdemir

**İkinci Danışman** : -

**Doktora Tez Savunma Jürisi:**

**Üye** : Doç. Dr. Müjgan Sağır Özdemir

**Üye** : Prof. Dr. İmdat Kara

**Üye** : Doç. Dr. Refail Kasımbeyli

**Üye** : Doç. Dr. Esra Karasakal

**Üye** : Doç. Dr. Muzaffer Kapanoğlu

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

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

# EĞİTİMSEL ZAMAN ÇİZELGELEME PROBLEMLERİ İÇİN ÇÖZÜM YAKLAŞIMLARI VE WEB TABANLI BİR KARAR DESTEK SİSTEMİ ÖNERİSİ

ZEHRA KAMIŞLI ÖZTÜRK

## ÖZET

Bu çalışmada ilk evrede; üniversitelerde karşılaşılan çok amaçlı eğitimsel zaman çizelgeleme problemlerinin önemli ve olabildiğince çok bileşeni içeren üç türü için kaliteli çözümler veren yöntemlerin geliştirilmesi amacıyla, üç yeni çok amaçlı karma tamsayılı matematiksel model ve iki yeni genetik algoritma sunulmuştur. Faaliyetlerin (ders) kaynaklara (derslik-zaman) atandığı ders-derslik-zaman çizelgeleme problemi (birinci problem) için geliştirilen karma tam sayılı matematiksel model ile literatürde yer alan test problemleri çözdürülmüş ve bir kısmı için en iyi çözüm elde edilmiştir. Büyük boyutlu problemlerin çözümüne hızlı bir şekilde ulaşma ihtiyacı ile bu problem için rassal anahtar temelli bir genetik algoritma da (RKGA) geliştirilmiştir. Ayrıca, problemin çok amaçlı yapısı nedeniyle RKGA'nın uyum değerinin hesaplanması aşamasında çok amaçlı bir GA olan NSGA-II algoritması RKGA'ya entegre edilmiştir. Çalışmada faaliyetlerin kaynaklara atandığı bir diğer problem olan sınav-derslik-zaman çizelgeleme problemi (ikinci problem) için geliştirilen karma tamsayılı matematiksel model gerçek bir problem için en iyi çözümü vermiştir. Bu problemlerin yanı sıra, literatürde diğer eğitimsel zaman çizelgeleme problemlerine nazaran çok az sayıda yer almış olan, bireylerin faaliyetlere atandığı gözetmen-sınav atama probleminin (üçüncü problem) çözümü için çok amaçlı bir karma tamsayılı matematiksel model geliştirilmiştir. Ayrıca, problemin daha sistematik ve gerçekçi bir şekilde ele alınabilmesi için problem parametreleri ve amaçların ağırlıklarını belirlemeye yönelik iki farklı çok ölçütlü model kurulmuştur. Çalışmanın son evresinde, ele alınan tüm problemlerin çözümü için web tabanlı bir karar destek sistemi (KDS) tasarlanmıştır. Hem kullanıcıların tercihlerinin toplandığı hem de ilgili problemler için veritabanına veri girişlerinin yapıldığı arayüzler ile yine ilgili problemin çözümü için gerekli model ve algoritmaların bulunduğu model tabanlı KDS'de yer almaktadır.

**Anahtar Kelimeler:** ders-derslik-zaman çizelgeleme problemi, sınav-derslik-zaman çizelgeleme problemi, gözetmen-sınav atama problemi, matematiksel modelleme, çok amaçlı karar verme, karar destek sistemi, rassal anahtarlı genetik algoritma

## **SOLUTION APPROACHES FOR EDUCATIONAL TIMETABLING PROBLEMS AND A WEB BASED DECISION SUPPORT SYSTEM PROPOSAL**

**ZEHRA KAMIŞLI ÖZTÜRK**

### **SUMMARY**

In the first phase of this study, three new multiobjective mixed-integer mathematical models and two new genetic algorithms are proposed for the purpose of obtaining high-quality solutions to three types of multiobjective educational timetabling problems faced in the universities that containing important and as many as possible components. A mathematical model (first problem) is developed for course-room-timeslot scheduling problem that assigns events (course) to resources (room-timeslot). Optimum results are gained for some types of test problems. Also a random key based genetic algorithm is developed for the purpose of reaching big dimensional problems' solutions in a rapid way. Besides, a multi objective genetic algorithm NSGA-II is integrated to RKGGA because of the problem' multiobjective structure. An another event-resource assignment problem handled in the study is exam-room-timeslot scheduling problem (second problem). A mixed integer mathematical model developed and the optimum solution is obtained for a real life problem. Invigilator assignment problem (third problem) that assigns individuals to events, relatively has not been well described and intensively worked yet as much as the other educational timetabling problems. A multi objective mixed integer model is developed for the solution of this problem. Also, to handle the problem in a more systematic and realistic way, two different multi criteria models are developed to determine the parameters and the objective weights of the problem. A web based decision support system (DSS) is designed to solve all above problems. The user interfaces where the user preference and restrictions are collected and where the related data entries are done, and the model base where the related models and algorithms for solving the problems are located in the DSS.

**Keywords:** course-room-timeslot scheduling problem, exam-room-timeslot scheduling problem, invigilator-exam assignment problem, mathematical modeling, multiobjective decision making, decision support system, random key based genetic algorithm

## TEŞEKKÜR

Lisans eğitimimden bu yana akademik çalışmalarını ve kişiliğini ile örnek aldığım sayın hocam Doç. Dr. Müjgan Sağır'a, doktora tez çalışmalarım sırasında vermiş olduğu bilimsel katkılardan ve zaman ve mekan konusunda hiçbir sınır koymadan vermiş olduğu destekten ötürü teşekkürlerimi sunuyorum.

Bugüne kadar hem akademik birikimi hem de hayat tecrübesinden faydalandığım sayın Doç. Dr. Refail Kasımbeyli'ye tez izleme sürecindeki katkılarından dolayı teşekkür ederim. Ayrıca, tez izleme jürimde yer alan sayın Doç. Dr. Esra Karasakal'a da katkılarından dolayı teşekkür ederim.

Çalışmalarında bilgi ve deneyimlerini paylaşan, her koşulda desteğini esirgemeyen sevgili eşim Yard. Doç. Dr. Gürkan Öztürk'e teşekkürü bir borç bilirim.

Aynı zamanda, 200815009 nolu tez projemi destekleyen ESOGÜ Bilimsel Araştırma Projeleri Komisyonu'na ve doktora eğitimim süresince maddi destek veren TÜBİTAK Bilim İnsanı Destekleme Daire Başkanlığı, Yurtiçi Doktora Burs Programı'na teşekkür ederim.

Bugüne kadar her zaman yanımda olan sevgili aileme ve özellikle ailemizin en yeni üyesi biricik oğlum Onur Ata Öztürk'e tez çalışmalarımın en yoğun olduğu dönemde bana verdiği destek ve moralden ötürü sonsuz teşekkürlerimi sunuyorum.

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> .....	Hata! Yer işareti tanımlanmamış.
<b>SUMMARY</b> .....	Hata! Yer işareti tanımlanmamış.
<b>TEŞEKKÜR</b> .....	vii
<b>İÇİNDEKİLER</b> .....	viii
<b>ŞEKİLLER DİZİNİ</b> .....	x
<b>ÇİZELGELER DİZİNİ</b> .....	xii
<b>1 GİRİŞ</b> .....	1
<b>2 EĞİTİMSEL ZAMAN ÇİZELGELEME PROBLEMLERİ</b> .....	5
2.1 Zaman Çizelgeleme Problemleri .....	5
2.2 Eğitimsel Zaman Çizelgeleme Problemleri ve Literatür Taraması .....	6
2.2.1 Ders çizelgeleme problemleri .....	13
2.2.2 Sınav çizelgeleme problemleri.....	15
2.3 Çözüm Yaklaşımları .....	20
2.3.1 Matematiksel programlama temelli yaklaşımlar .....	24
2.3.2 Ağ renklendirme temelli yaklaşımlar.....	27
2.3.3 Yerel arama temelli yaklaşımlar .....	30
2.3.4 Metasezgisel yöntemler .....	31
2.3.5 Olay-temelli çıkarsama yaklaşımları .....	44
2.3.6 Hiper-sezgisel yöntemler .....	44
2.3.7 Kısıt temelli teknikler .....	46
2.3.8 Çok-amaçlı yaklaşımlar .....	47
2.4 Karşılaşılan Güçlükler ve Gelineyen Nokta .....	49
<b>3 DERS DERSLİK ZAMAN ÇİZELGELEME PROBLEMİ İÇİN YENİ ÇÖZÜM YAKLAŞIMLARI</b> .....	<b>52</b>
3.1 Ders Derslik Zaman Çizelgeleme Problemi ve Matematiksel Model ile Çözümü .....	52

3.2 Ders Derslik Zaman Çizelgeleme Problemi için Rassal Anahtar Temelli bir Genetik Algoritma (RKGA).....	65
3.2.1 Genetik algoritmalar, eğitimsel zaman çizelgeleme ve genetik algoritma uygulamaları.....	66
3.2.2 Rassal anahtar temelli genetik algoritmanın (RKGA) adımları.....	76
3.2.3 Çok amaçlı genetik algoritmalar .....	84
3.2.4 NSGA-II temelli çok amaçlı genetik algoritma .....	87
3.2.5 Test problemlerinin denenmesi.....	88
<b>4 SINAV DERSLİK ZAMAN ÇİZELGELEME PROBLEMİ İÇİN ÇÖZÜM YAKLAŞIMLARI .....</b>	<b>92</b>
4.1 Sınav-Derslik-Zaman Çizelgeleme Problemi ve Matematiksel Model ile Çözümü .....	92
4.2 Gözetmen-Sınav Atama Problemi .....	103
4.2.1 Çok amaçlı gözetmen-sınav atama problemi için matematiksel model ve çözümü.....	105
4.1.1 Problem parametrelerinin tahmini için bir AHP modeli.....	114
4.1.2 Problem parametrelerinin tahmini için bir ANP modeli.....	123
<b>5 ÇOK AMAÇLI EĞİTİMSEL ZAMAN ÇİZELGELEME PROBLEMLERİ İÇİN WEB TABANLI BİR KARAR DESTEK SİSTEMİ TASARIMI.....</b>	<b>135</b>
5.1 Karar Destek Sistemleri.....	136
5.2 Web Tabanlı Karar Destek Sistemleri.....	138
5.3 Eğitimsel Zaman Çizelgeleme Problemleri için Web Tabanlı Karar Destek Sistemi Önerisi (EZÇ-KDS) .....	139
5.3.1 Veri yönetimi .....	145
5.3.2 Diyalog yönetimi .....	147
5.3.3 Model yönetimi .....	155
<b>6 SONUÇ VE ÖNERİLER.....</b>	<b>156</b>
<b>KAYNAKLAR DİZİNİ .....</b>	<b>161</b>
<b>EKLER .....</b>	<b>184</b>
<b>ÖZGEÇMİŞ</b>	

## ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
2.1. Eğitimsel zaman çizelgeleme problemleri .....	9
2.2 Eğitimsel zaman çizelgeleme problemlerinin uygulama alanı başlığına göre dağılımı.....	10
2.3 Çalışmaların gerçekleştiği yıllar.....	12
2.4 Eğitimsel zaman çizelgeleme problemlerinin çözümünde kullanılan çözüm yaklaşımlarının dağılımı.....	21
2.5 Ağ renklendirme ve basit bir zaman çizelgeleme problemi ile ilişkisinin gösterimi (Lewis, 2008'den) .....	28
2.6 Kullanılan metasezgisel yöntemler .....	39
3.1 Small01 probleminin her ardıştırmadaki amaç fonksiyonu değerleri .....	63
3.2 GA'da kromozom yapısı .....	68
3.3 Bir zaman çizelgesinin matris gösterimi (Colorni et al., 1992).....	70
3.4 Bir zaman çizelgesinin matris gösterimi (Yu and Sung, 2002).....	70
3.5 Üç satırlı bir kromozom örneği (Vorac et al., 2002).....	72
3.6 5 ders ve 5 zaman dilimi için tek nokta çaprazlama sonucu elde edilen uygun olmayan çözüm.....	74
3.7 RKGA'da kullanılan kromozomun yapısı.....	76
3.8 Rassal anahtar kullanılarak zaman dilimi atamasının yapılması.....	77
3.9 Parametrik çaprazlama örneği .....	78
3.10 RKGA'nın sınıf diyagramı .....	80
3.11 NSGA-II algoritması .....	88
4.1 Dört seviyeli hiyerarşi .....	118
4.2 Tüm ölçütlere ait ağırlıklar.....	120
4.3 Serim modeli .....	125
4.4 Gözetmen sınav atama problemi için geliştirilen ANP modeli.....	128
5.1 KDS bileşenleri .....	137
5.2 EZÇ-KDS için içerik diyagramı.....	141

**ŞEKİLLER DİZİNİ (devam)**

<b><u>Sekil</u></b>	<b><u>Sayfa</u></b>
5.3 EZÇ-KDS'nin genel yapısı .....	143
5.4 Dersler tablosu.....	145
5.5 Veritabanında yer alan ilişkiler .....	146
5.5 Sistem giriş ekranı .....	147
5.6 EZÇ-KDS yönetici ekranı .....	148
5.7 Sınav-derslik-zaman çizelgeleme yönetici arayüzü .....	149
5.8 Gözetmen-sınav atama problemi için ana yönetici ekranı .....	150
5.9 Gözetmen seçimi ekranı .....	151
5.10 Sınav-derslik-zaman çizelgesinin giriş ekranı.....	151
5.11 Ön atamaların yapıldığı arayüz .....	152
5.12 Gözetmenler için tercih ve kısıt giriş ekranı .....	153
5.13 Veritabanından modele veri gönderimi .....	154
5.14 Ders-derslik-zaman çizelgeleme problemi için GA çözüm ekranı.....	155

## ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
2.1 Sezgisellerin kullanıldığı çalışmalar .....	23
2.2 Matematiksel programlama temelli çözüm yöntemlerinin kullanıldığı çalışmalar.....	26
2.3 Yerel arama tekniklerinin kullanıldığı çalışmalar .....	30
2.4 Yerel arama tekniklerinin kullanıldığı çalışmalar .....	31
2.5 Yıllar temelinde eğitimsel zaman çizelgeleme problemlerinin çözümünde kullanılan yaklaşımlar .....	32
2.6 Tavlama benzetiminin kullanıldığı çalışmalar .....	35
2.8 Yasaklı aramanın kullanıldığı çalışmalar .....	37
2.8 Evrimsel algoritmaların kullanıldığı çalışmalar .....	39
2.9 Hiper-sezgisellerin kullanıldığı çalışmalar .....	46
3.1 Test problemlerinin özellikleri.....	54
3.2 Öğrenci ders matrisi.....	55
3.3 Derslik nitelik matrisi .....	56
3.4 Ders nitelik matrisi.....	56
3.5 Small tipi problemler için elde edilen amaç fonksiyonu değerleri .....	64
3.6 Ders-derslik-zaman çizelgeleme problemi için geliştirilen matematiksel model için boyut analizi .....	65
3.7 small, medium ve large problem türleri için literatürde elde edilen sonuçlar .....	89
3.8 RKGA ile elde edilen sonuçlar .....	90
3.9 NSGA-II temelli RKGA ile elde edilen sonuçlar .....	91
4.1 Gün temelinde sınav yapılabilir zaman dilimleri kümesi .....	94
4.2 Sınıf temelinde standart dersler kümesi.....	99
4.3 Geliştirilen model ile elde edilen sınav-derslik-zaman çizelgesi.....	100
4.4 Sınav-derslik-zaman çizelgeleme problemi için geliştirilen matematiksel model için boyut analizi .....	102
4.5 Kısıt ve tercih girişinde kullanılan simgeler .....	108

4.6 Zaman dilimleri.....	109
4.7 2004–2005 yılı bahar dönemi birinci arasınava gözetmen yük oranları .....	110
4.8 2004-2005 Güz dönemi final sınavları gözetmen atamaları.....	111
4.9 2004-2005 bahar dönemi I. arasınava sınav gözetmen atamaları.....	112
4.10 Gözetmen-sınav atama problemi için geliştirilen matematiksel model için boyut analizi .....	113
4.11 Saaty ölçeği.....	116
4.12 Bazı uzmanlar ve grup kararına göre sınav parametrelerinin göreceli ağırlıkları .....	119
4.13 AHP'den elde edilen yeni sınav ağırlıkları .....	122
4.14 Mevcut ve AHP ağırlıkları kullanılarak elde edilen çözümler .....	123
4.15 ANP modeline göre elde edilen ölçüt ve seçenek ağırlıkları.....	129
4.16 Amaç fonksiyonlarının göreceli önem değerleri .....	130
4.17 ANP'den elde edilen yeni sınav ağırlıkları .....	131
4.18 ANP ağırlıkları kullanılarak elde edilen çözüm .....	133
5.1 Gözetmen-sınav atama modülünde gerçekleştirilen olaylar tablosu .....	144
5.2 Kısıt ve tercih girişinde kullanılan simgeler .....	153

## BÖLÜM 1

### GİRİŞ

Zaman çizelgeleme problemleri ile ilköğretim kurumlarında, liselerde ve üniversitelerde haftalık ders/sınav zaman çizelgelerinin hazırlanmasından, havayolu taşımacılığında ulaşım faaliyetlerinin çizelgelenmesine, sağlık sektöründe personelin nöbet çizelgelerinin hazırlanmasından, üretim ve/veya servis işletmelerinde vardiya çizelgelenmesine kadar yaygın bir alanda karşılaşılmaktadır. Eğitimsel zaman çizelgeleme ise bu alanda üzerinde sürekli çalışılan bir problem türüdür.

Bir eğitimsel zaman çizelgeleme probleminde bireyler faaliyet ve/veya kaynaklara yanı sıra faaliyetler de kaynaklara atanabilmektedir. Öğreticilerin derslere veya gözetmenlerin sınavlara atanması bireylerin faaliyetlere atanmasına; gözetmen ya da öğreticilerin zaman dilimleri ya da dersliklere atanması ise bireylerin kaynaklara atanmasına örnek olarak verilebilir. Öte yandan, derslerin/sınavların dersliklere ve/veya zaman dilimlerine atanması ise faaliyetlerin kaynaklara atanmasıdır.

Bu sınıflama içerisinde çok farklı kombinasyonlarda alt problemler ile karşılaşılabilir. Bu çalışma kapsamında her bir problem sınıfına örnek verebilmek açısından; bireylerin faaliyetlere atanması problemi için *Gözetmen Sınav Atama*, faaliyetlerin kaynaklara atanması ise *Ders Derslik Zaman Çizelgeleme*, yanı sıra farklı tür bir faaliyet kaynak atama problemi tipi olduğu için de *Sınav Derslik Zaman Çizelgeleme* problemleri ele alınmıştır. Çalışmada ders ve sınav çizelgeleme problemleri birden fazla boyutlarıyla ele alınmış ve bunlara yönelik çözüm yaklaşımları geliştirilmiştir.

Bu çalışma kapsamında öncelikle, literatürde yer alan 196 eğitimsel zaman çizelgeleme çalışması incelenmiştir. Bu çalışmalarda ele alınan bileşenler belirlenmiş, yanı sıra kullanılan çözüm yaklaşımları da irdelenmiştir. Problemlerin farklı bileşenleri içermelerine rağmen aynı başlık altında ele alınmış olmaları ise terminolojide henüz bir

standartlaşmanın olmadığını göstermiştir.

Çalışma kapsamında ele alınan ders-derslik-zaman çizelgeleme probleminde, literatürde yer alan problemlerin birçoğuna ek olarak, derslik kapasitesi ve derslerin ihtiyaç duyduğu nitelikleri içeren dersliklere atanması kısıtları da ele alınmıştır. Ders-derslik-zaman çizelgeleme problemi için kurulacak modelin çözümdeki başarısını sınamak amacıyla literatürde yer alan çeşitli test problemleri incelenmiştir. Çalışmada çözümü aranan problemler Ben Paechter<sup>1</sup> tarafından tanımlanan ve Metasezgiseller Ağı<sup>2</sup>'nda çalışılan problemlerdir. Ele alınan üç amaçlı ders-derslik-zaman çizelgeleme probleminin öncelikle matematiksel modeli geliştirilmiştir. Bu problem yapısına ait bazı test problemleri (*small1*, ..., *small5*) için en iyi çözüm elde edilmiştir. Ancak, boyut analizi ve uzun çözüm süreleri ders-derslik-zaman çizelgeleme probleminin çözümünde sezgisel bir yonteme olan ihtiyacı ortaya çıkmıştır. Bu kapsamda, henüz eğitimsel zaman çizelgeleme problemlerinin çözümünde kullanılmamış olan *rassal anahtar* yaklaşımı ile ders-derslik-zaman çizelgeleme probleminin çözümü için bir genetik algoritma (RKGA-*random key based genetic algorithm*) geliştirilmiştir. Burada her bir çözüm, ders sayısı kadar uzunlukta bir reel sayı vektörü ile gösterilmiştir. Her rassal sayı, hem ilgili dersin atandığı sınıfın, hem de zaman diliminin belirlenmesinde kullanılmıştır. Rassal anahtarlar sayesinde, bir kromozomdan her zaman uygun çözüm oluşturmayı sağlayan bir çözümleme işlemi tanımlanmıştır. Yanısıra, problemin çok amaçlı yapısından dolayı bir çok amaçlı genetik algoritma olan NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) algoritması ile RKGA algoritması birleştirilerek problemler çözdürülmüştür.

Ders zaman çizelgeleme hem üniversitelerde hem de orta öğretim kurumlarında karşılaşılan bir problemdir. Ancak, orta öğretim kurumlarında her sınıfın sabit bir dersliği olduğundan bir derslik atama problemi ile genellikle karşılaşılmaz. Yanısıra, ardışık dersler arasında boş zaman dilimleri olmadığından orta öğretimlerde karşılaşılan ders çizelgeleme problemi üniversite ders zaman çizelgeleme probleminin bir alt problemi gibi düşünülebilir. Çalışma kapsamında ele alınan ders zaman çizelgeleme

<sup>1</sup> <http://www.dcs.napier.ac.uk/~benp/>

<sup>2</sup> Metaheuristic network

problemi üniversitelerdeki ders-derslik-zaman çizelgeleme problemi için ele alınmış olmakla birlikte başka çizelgeleme problemlerine de benzetilerek çözümlerine temel olması söz konusudur.

Çalışma kapsamında ele alınan sınav-derslik-zaman çizelgeleme probleminde ise bir sınavın birden fazla derslikte yapılması gibi çözümleri zorlaştıran kısıtlar da göz önünde bulundurularak, belirlenen dört amaç için en iyi çözümü veren bir matematiksel model geliştirilmiştir. Bu çalışmada çok amaçlı olarak ele alınan bir diğer alt problem de kullanıcı tercihlerinin göz önünde bulundurulduğu gözetmen-sınav atama problemi olmuştur. Bu problem türü için en iyi çözümü veren bir matematiksel model geliştirilmiştir (Öztürk vd., 2010). Ayrıca, bu problemin problem parametrelerinin ve amaçlarının problem parametrelerinin ve amaçların ağırlıklarını belirlemeye yönelik olarak çok amaçlı karar verme teknikleri (AHP ve ANP) uygulanmıştır. Geliştirilen matematiksel modelin, model amaçlarının AHP ve ANP ile ağırlıklandırılması sonrasında yeniden en iyi çözümler bulunmuş ve çözümler karşılaştırılmıştır.

Ele alınan tüm problemlerin çözümü için web tabanlı bir karar destek sistemi (KDS) tasarlanmıştır. Hem kullanıcıların tercihlerinin toplandığı hem de ilgili problemler için veritabanına veri girişlerinin yapıldığı arayüzler ile yine ilgili problemin çözümü için gerekli model ve algoritmaların bulunduğu model tabanlı KDS'de yer almaktadır. Problem çözümleri web üzerinden yetkili kişiler tarafından yaptırılmakta, sonuçlar da yine tüm kullanıcılara arayüzlerle rapor şeklinde sunulmaktadır.

Bir eğitim döneminde en az iki defa karşılaşılan eğitimsel zaman çizelgeleme problemlerinin çözümü yoğun iş gücü ve kaynak gerektirmektedir. Dolayısıyla, eğitim kurumları açısından önemli olduğu kadar da zor problemlerdir. Ayrıca, kısıt ve amaçları kurumdan kuruma farklılık gösterebilir. Bu durum, bu alandaki çalışmaların sürekliliğine neden olmaktadır.

İzleyen bölümlerde sırasıyla; eğitimsel zaman çizelgeleme problemleri ayrıntılı literatür taraması desteğiyle açıklanmakta, çözüm yaklaşımları verilmektedir. Karşılaşılan güçlükler ve geline nokta tartışılarak; ders-derslik-zaman çizelgeleme,

sınav-derslik-zaman çizelgeleme, yanı sıra gözetmen-sınav atama problemleri ve çözüm yaklaşımları verilmektedir.

Son bölümde çok amaçlı eğitimsel zaman çizelgeleme problemi için geliştirilen web tabanlı bir KDS tasarımı tanıtılmaktadır.

## BÖLÜM 2

### EĞİTİMSEL ZAMAN ÇİZELGELEME PROBLEMLERİ

Eğitimsel zaman çizelgeleme problemleri ve çözüm yaklaşımlarının ele alındığı bu bölümde, öncelikle zaman çizelgeleme problemleri tanıtılmıştır. Literatür taraması ile eğitimsel zaman çizelgeleme problemlerinin temel yönleri Bölüm 2.2’de incelenmiş, Bölüm 2.3’te ise bu problemler için önerilen farklı çözüm yaklaşımları tanıtılmıştır. Son olarak Bölüm 2.4’te, karşılaşılan güçlükler ve gelinen nokta sunulmuştur.

#### 2.1 Zaman Çizelgeleme Problemleri

Bir tür çizelgeleme (*scheduling*) problemi olan zaman çizelgeleme (*timetabling*) problemleri genel olarak, belirli sayıda *faaliyetin* sınırlı sayıda *zaman dilimine* belirli koşulları sağlayacak şekilde atanması ile ilgilenir. Her zaman çizelgeleme problemi kendine özgü kısıtlar içerir. Genellikle bu kısıtlar *sıkı* (hard) ve *esnek* (soft) olmak üzere iki sınıfa ayrılır. Sıkı kısıtlar, bir olayın aynı zaman diliminde birden fazla kaynağa atanmaması gibi hiçbir şekilde ihlal edilemeyen kısıtlardır. Diğer yandan, iyi bir çizelge için zorunlu olmayan ancak faydalı olan koşullar da vardır. Bu kısıtlar ne kadar çok sağlanırsa çizelge de o kadar iyi kabul edilir. *Esnek kısıt* olarak tanımlanan bu kısıtlar, genellikle ortak kişilere ve/veya kaynak ihtiyacına sahip olayların arka arkaya gerçekleşmesini tercih etmeme gibi zorunlu olmayan ama mümkün olduğunca sağlanması istenen kısıtlardır. Bu tür esnek kısıtların yerine getirilemeyenlerinin enküçüklenmesi, bazı durumlarda bu problemlerin amaç fonksiyonunu oluşturur.

Zaman çizelgeleme problemlerinde

- ders, sınav vb. çizelgelenmesi gereken olaylar *faaliyet*,
- faaliyetlerin çizelgeleneceği zaman aralıkları *zaman dilimi*,
- derslik, projektör ve İnternet bağlantısı gibi faaliyetlerin ihtiyaç duyduğu olanaklar *kaynak*,

- derslik kapasitesi, özel zaman dilimlerine atanma vb. ile ilgili sınırlılıklar *kısıt*,
- öğrenci, öğretici ve gözetmen gibi faaliyetlere katılması gerekenler *kişi*,
- en az bir tane ortak kişisi veya kaynak ihtiyacı olan iki faaliyetin aynı zaman dilimine çizelgelenmesi *çakışma*

olarak tanımlanmaktadır

Karmaşık yapısı, farklı uygulama alanları, elle çözümün zaman ve kaynak açısından çok maliyetli olması ve bu nedenle çözüm için bilgisayar desteğinin kaçınılmazlığı, zaman çizelgeleme problemlerinin 1950'li yıllardan beri sürekli gündemde olmasını sağlamıştır. Farklı uygulama alanlarında yüzlerceyle karşılaşılan zaman çizelgeleme problemlerine bir fikir vermesi açısından izleyen birkaç örnek verilebilir:

- Ders, sınav ve/veya sunum zamanlarının çizelgelenmesi gibi eğitimsel zaman çizelgeleme (Badri, 1996; Hilton et al., 2001)
- Nöbet çizelgeleme (Azmat and Widmer, 2004; Cappanera and Gallo, 2004)
- Tren, otobüs ve uçak seferleri ile ilgili ulaşım zaman çizelgeleme (Lampkin and Saalmans, 1967; Liebchen and Möhring, 2002; Chierici et al., 2004)
- Spor karşılaşmalarının zaman çizelgelenmesi (Wright, 1991; Schönberger et al., 2004; Russell and Urban, 2006)
- Avukatların davalara çizelgelenmesi (Wilson, 1981)

## 2.2 Eğitimsel Zaman Çizelgeleme Problemleri ve Literatür Taraması

Son 25 yıl içerisinde çeşitli zaman çizelgeleme problemleri üzerine modeller ve çözüm yaklaşımları önerilmiştir. Ancak, bunların içinde en çok ilgiyi eğitimsel zaman çizelgeleme problemleri görmektedir (Johnson, 1993; Foulds and Johnson, 2000). Eğitim kurumlarındaki zaman çizelgelerinde ders, derslik gibi kaynaklar veya

faaliyetler ile öğrenci, öğretici, gözetmen gibi kişiler yer almaktadır. Öğreticiler de bir anlamda, aynı zamanda kaynak olarak da sayılabilirler.

Eğitimsel zaman çizelgeleme ile ilgili çalışmalar incelendiğinde farklı sınıflandırmalar ile karşılaşılmaktadır. Burke vd. (2004) bu problemleri “sınıf-öğretmen (*class-teacher*)”, “üniversite-ders (*university-course*)” ve “üniversite-sınav (*university-examination*)” olmak üzere üç ana başlık altında; Petrovic ve Burke (2004) ise üniversite zaman çizelgeleme problemlerini “ders” ve “sınav” olarak iki ana başlık altında incelemişlerdir.

Bir başka sınıflandırma Schaerf (1999) tarafından verilmiştir. Buna göre, eğitimsel zaman çizelgeleme problemleri üç temel grupta ele alınabilir:

- i. Okul zaman çizelgeleme (*School timetabling*).** Derslerin haftalık çizelgelerinin oluşturulması.
- ii. Ders zaman çizelgeleme (*Course timetabling*).** Üniversitelerdeki derslerin haftalık çizelgelerinin oluşturulması.
- iii. Sınav zaman çizelgeleme (*Exam timetabling*).** Üniversitelerdeki derslerin sınavlarının çizelgelenmesi.

Carter ve Laporte (1998) ise eğitimsel zaman çizelgeleme problemlerini izleyen beş alt probleme ayırmıştır:

- i.** Derslerin ne zaman ve hangi derslikte yapılacağını dikkate almadan sadece öğretmenlerin derslere atanması (örn. McClure ve Wells, 1984).
- ii.** Bir öğreticinin aynı zaman diliminde birden fazla dersi veremeyeceği koşulunu dikkate alarak sadece öğretmenlerin ders verme zamanlarının çizelgelenmesi. Öğreticilerin derslere, derslerin de zaman dilimlerine atandığı örnek çalışmalardan biri Gunavan ve diğerlerine (2007) aittir.
- iii.** Derslerin zaman dilimleri ve dersliklere atanmasını ele alan ders çizelgeleme

problemi (örn. Wright, 1996).

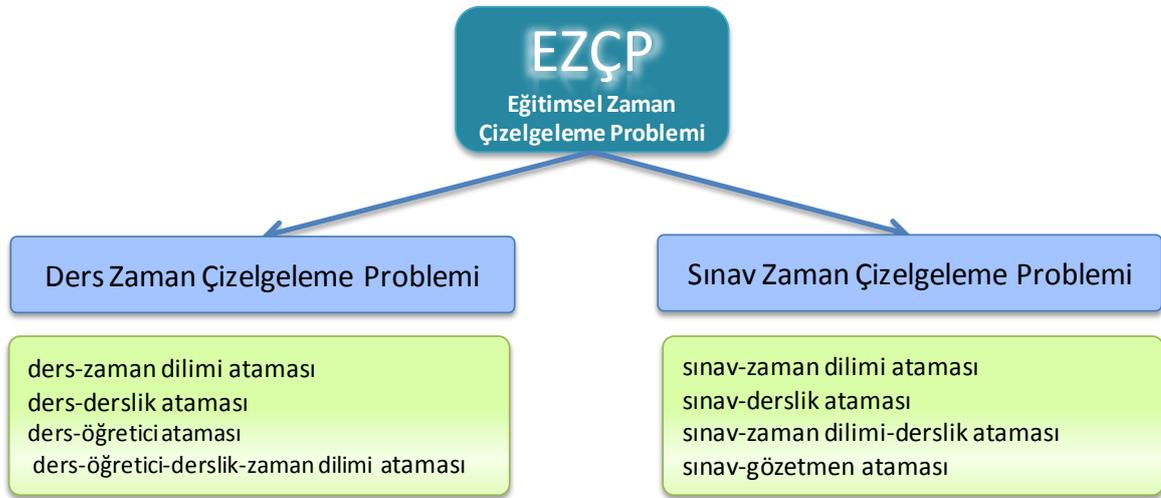
- iv. Ortak öğrencisi olan derslerin aynı zaman dilimlerine atanması önlenecek şekilde derslerin zaman dilimlerine çizelgelenmesini ele alan öğrenci çizelgeleme problemi (örn. Alvarez-Valdes vd. 2000).
- v. Oturma kapasitesi, gerekli ekipmanlar gibi özel gereksinimleri dikkate alarak derslerin zaman dilimlerine atanmasını ele alan derslik atama problemi (Glassey ve Mizrach, 1986).

Yukarıda tanımlanmış problemlerden de anlaşılacağı üzere, eğitimsel zaman çizelgeleme problemlerinde temel bileşenler,

- Ders
- Derslik
- Zaman dilimi
- Öğretici
- Öğrenci

şeklinde dir. Literatürde veya gerçek hayatta karşılaşılan herhangi bir probleme yukarıdaki bileşenlerin tümünü veya herhangi bir veya birkaç anlamlı kombinasyonunu farklı sıralarda çözmekle çözüm aranabilir. Genellikle tüm bileşenleri aynı anda ele almak, çözümü olanaksız yapılar ortaya çıkarmaktadır. Dinkel vd. (1989), 1970 ve 80'lerde yapılan çalışmaların birçoğunda öğretici, ders ve derslik kombinasyonlarının zaman dilimlerine atanmasının ele alınmadığını, açılan ders çeşitliliği arttıkça problem boyutunun ve çizelgelerin oluşturulması için gerekli sürenin arttığını belirtmiştir. Schaerf (1999) de 1963-1999 yılları arasındaki çalışmaları derlediği yayınında, problemin NP-tam olduğunu ve en iyi çözümün sadece küçük boyutlu problemler (örneğin en fazla 10 ders için) için elde edilebileceğini göstermiştir.

Bu çalışma kapsamında *eğitimsel zaman çizelgeleme problemleri*, *ders ve sınav çizelgeleme* şeklinde iki ana gruba ayrılmıştır (Şekil 2.1). Yukarıda bahsedilen, karşılaşılabılır alt problemler, tanımlanan iki ana başlığın birisinde yer almaktadır. Şekil 2.1'de en çok karşılaşılan alt problemler listelenmiştir.



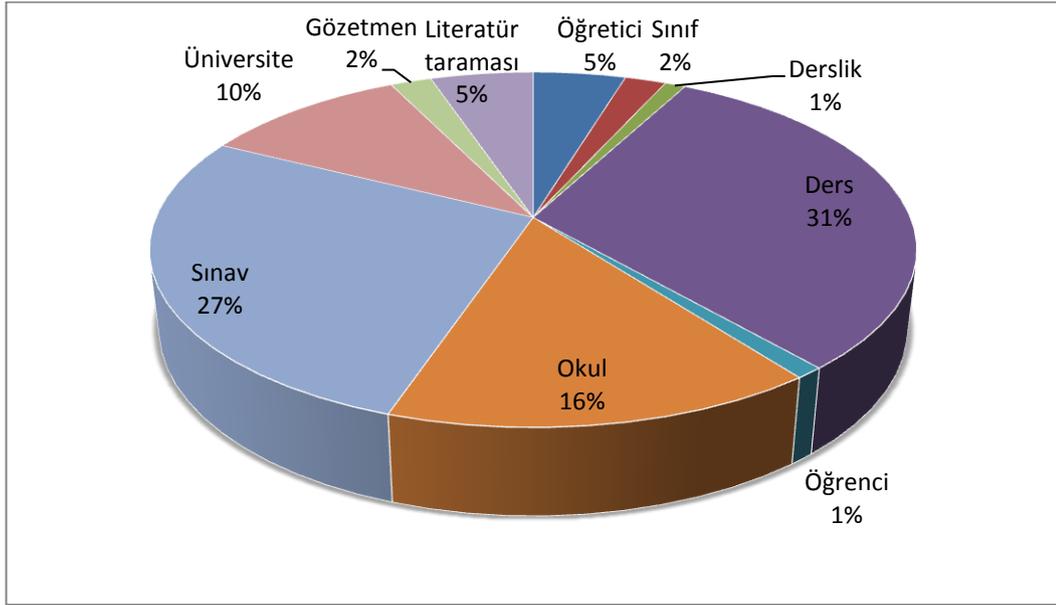
**Şekil 2.1.** Eğitimsel zaman çizelgeleme problemleri

Literatürde; yukarıda belirtilenlere ek olarak, çok sayıda olmasa da öğretici-derslik, öğretici-zaman dilimi atama problemleri de yer almaktadır. Problemlerde temel bileşen; tanımlanan iki ana grupta da olduğu gibi ders çizelgeleme için ders, sınav çizelgeleme içinse sınavdır.

Şekil 2.1’den de görüldüğü gibi, bazı alt problemlerde zaman boyutu yer almadığı halde problem bir tür zaman çizelgeleme (*timetabling*) problemi gibi tanımlanabilmektedir. Örneğin sadece ders-derslik ataması yapıldığı halde ders zaman çizelgeleme (*course timetabling*) olarak tanımlanan çalışmalar olduğu gibi (Ferland, 1985). Bu çalışmada bu anlamda literatüre önemli ölçüde bağlı kalınmış, *timetabling* kelimesi için *zaman çizelgeleme* karşılığı kullanılmıştır. İzleyen paragraflarda bu konudaki bir sınıflamaya yer verilmektedir.

Bu çalışma kapsamında, 1964 ile 2010 yılları arasında yayınlanmış ve eğitimsel zaman çizelgeleme ile ilgili olan 196 çalışma incelenmiş ve uygulama alanlarına göre sınıflandırılmıştır. Sınıflandırma sonucunda farklı başlıklar ortaya çıkmıştır. Öğretici (*faculty*), sınıf (*class*), derslik (*classroom*), ders (*course*), öğrenci (*student*), okul (*school*), sınav (*exam*), gözetmen (*proctor/invigilator*) ve üniversite (*university*) olarak

belirlenen bu başlıklara ait çalışmaların yüzdeleri Şekil 2.2’de verilmiştir. Ders ve sınav çizelgeleme problemlerinin sırasıyla %31 ve %27’lik pay ile üzerinde en çok çalışılan konular olduğu belirlenmiştir.



**Şekil 2.2** Eğitimsel zaman çizelgeleme problemlerinin uygulama alanı başlığına göre dağılımı

Çalışmalarda göz önüne alınan bileşenler incelendiğinde ise aynı başlık altında farklı atamaların yapıldığı görülmüştür. Buna göre, çalışmalarla ilgili olarak şunlar söylenebilir:

- Çalışmaların %2’si sınıf (*class*) başlığı altında ele alınmıştır. Bu çalışmaların %33’ünde ders-derslik-zaman dilimi, %22’sinde ders-zaman dilimi, %22’sinde ders-öğretici-zaman dilimi, %11’inde ders-öğretici ve yine %11’inde ders-derslik öğrenci ataması yapılmıştır.
- Çalışmaların %9’u ders (*course*) başlığı altında ele alınmıştır. Ders başlığı altındaki çalışmaların %16’sında ders-öğretici, %16’sında ders-zaman dilimi ve

bir diğerk %16'sında ders-öğretici-zaman dilimi ataması yapılırken, %40'ında ders-derslik-zaman dilimi, %6'sında ders-derslik-öğrenci-zaman dilimi ve diğerk %6'sında da ders-öğrenci ve öğretici ataması yapılmıştır.

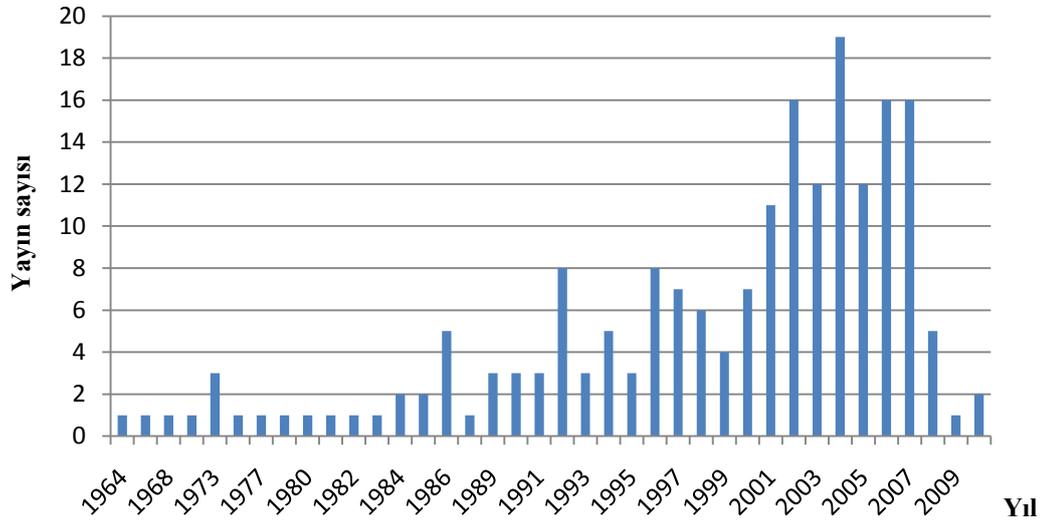
- Çalışmaların %16'sı okul (*school*) başlığı altında ele alınmıştır. Okul başlığı altındaki çalışmaların %22'sinde ders-öğretici-zaman dilimi, %19'unda ders-zaman dilimi ve bir diğerk %19'unda ders-derslik-öğretici-zaman dilimi, %16'sında ders-derslik-zaman dilimi, %12'sinde ders-öğretici, %6'sında öğretici-zaman dilimi, %3'ünde derslik-zaman dilimi ve diğerk %3'ünde de ders-öğrenci-zaman dilimi ataması yapılmıştır.
- Çalışmaların %22'si üniversite-ders (*University-Course*) başlığı altında ele alınmıştır. Bu çalışmaların %63'ünde ders-derslik-zaman dilimi, %12'sinde ders-zaman dilimi ve %10'unda ders-öğretici-zaman dilimi, %5'inde ders-derslik-öğretici-zaman dilimi ataması, %2,5'inde ders-derslik, %2,5'inde ders-öğretici-öğrenci, %2,5'inde derslik-zaman dilimi ve bir diğerk %2,5'inde de ders-öğrenci ataması yapılmıştır.
- Çalışmaların %27'si sınav (*exam*) başlığı altında ele alınmıştır. Bu başlık altındaki çalışmaların %48'inde sınav-zaman dilimi ataması yapılırken, %8'inde sınav-derslik, %21'inde sınav-derslik-zaman dilimi ataması yapılmıştır.
- Çalışmaların %2'si gözetmen (*proctor/invigilator*) başlığı altında ele alınmıştır. Bu çalışmalarda gözetmenlerin ilgili sınav dönemlerinde sınavlara atamaları yapılmıştır.
- Çalışmaların %10'u üniversite zaman çizelgeleme (*university timetabling*) başlığı altında ele alınmıştır. Bu çalışmaların %30'unda ders-derslik-zaman dilimi, %33'ünde ders-öğretici ve %5'inde sınav-derslik-zaman dilimi ataması yapılmıştır.
- Çalışmaların %5'i öğretici (*faculty-course* ve *faculty-teacher*) başlığı altında ele alınmıştır. Bu çalışmalarda öğretmenler, derslere atanmıştır.
- Çalışmaların %1'i öğrenci (*student*) başlığı altında ele alınmıştır. Bu

çalışmalarda öğrenciler, derslere atanmıştır.

- Çalışmaların %5'i de literatür taraması çalışmasıdır.

Bu çalışmalarda ele alınan bileşenler Çizelge EK-1'de özetlenmiştir.

1960'lı yıllarda başlamış ve günümüze kadar devam etmekte olan eğitimsel zaman çizelgeleme çalışmalarının özellikle 2000 yılından sonra artan bir ivmeyle seyrettiği Şekil 2.3'ten de gözlenmektedir.



**Şekil 2.3** Çalışmaların gerçekleştiği yıllar

İzleyen bölümlerde eğitimsel zaman çizelgeleme problemlerinin iki grubu olarak tanımladığımız ders ve sınav çizelgeleme problemleri; alt problemleri ile birlikte tanımlanmaktadır.

### 2.2.1 Ders çizelgeleme problemleri

Ders çizelgeleme problemi temel olarak, hiçbir öğrenci ve öğreticinin aynı anda birden fazla dersi olmayacak şekilde derslerin, derslik ve zaman dilimlerine atanması problemidir.

İlköğretim ve liselerdeki derslerin çizelgelenmesinden farklı olarak üniversitelerde, ortak öğrencisi olan derslerin aynı zaman dilimlerine atanmaması gerekmektedir. Ayrıca üniversitelerde, her dersin öğrenci sayısı farklı olduğundan, dersliklerin kapasiteleri de problem için önemli bir parametredir. Bu çalışma kapsamında ders zaman çizelgeleme problemleri üniversiteler için ele alınmıştır.

Ders zaman çizelgeleme problemleri belirli sıkı ve esnek kısıtlar altında çözülür. Sağlanması istenen ancak zorunlu olmayan esnek kısıtlar ise problemin ele alındığı kuruma göre değişiklik gösterir. Temel olarak sağlanması zorunlu olan sıkı kısıtlar,

1. Hiçbir kaynak (öğrenci, öğretici) aynı zaman diliminde birden fazla faaliyete atanamaz,
2. Her zaman dilimi için, o zaman dilimine atanan tüm faaliyetlere yeterli kaynak ayrılmalıdır,

olarak tanımlanabilir.

Literatürde yeralan çeşitli esnek kısıtlar ise Burke ve Petrovic (2002) tarafından izleyen şekilde derlenmiştir:

1. Bir ders önceden belirlenmiş bir zaman dilimine atanabilir.
2. Öğrencilerin derslerinin gün içinde ardışık zaman dilimlerine atanması engellenebilir.
3. Öğreticilerin dersleri belirli günlere atanabilir. Böylece öğretici, haftanın diğer günlerini boş bırakmayı tercih edebilir.
4. Öğreticinin isteğine bağlı olarak, dersler belirli dersliklere atanabilir.

Ayrıca, Sarin vd.'nin (2010) çalışmalarında olduğu gibi, öğretmenlerin ofislerinden

dersliklere kadar katettikleri toplam mesafenin olabildiğince az olmasını amaçlayan uygulama yerine özgü çalışmalarla da karşılaşmıştır.

Bloomfield ve McSharry (1979), öğrencilerin de tercihlerini göz önünde bulundurarak, dersleri derslik ve zaman dilimlerine iki aşamada atamıştır. Glassey ve Mizrach (1986) ise oturma kapasitesi, gerekli ekipmanlar gibi özel gereksinimleri dikkate alarak derslerin derslik ve zaman dilimlerine atanmasını ele almıştır. Tripathy (1984), Hertz (1991), Goltz vd. (1999), Schaefer ve di Gaspero (2001), Dimopoulou ve Miliotis (2001), Petrovic ve Qu (2002), Socha vd. (2002), Daskalaki vd. (2004) ve Murray (2007) ders-derslik-zaman çizelgesini oluşturan çalışmalara diğer örnekler olarak verilebilir.

Literatürde yeralan ders zaman çizelgeleme problemlerinin; problem boyutunun ve çözüm için gerekli sürenin artması nedeniyle, genellikle birden fazla aşamada, ders-derslik atama, ders-öğretici atama gibi alt problemler şeklinde çözüldüğü bir önceki bölümde belirtilmişti. Bu çalışmaların büyük çoğunluğu, öğrencilerin vereceği derslerin önceden bilindiği varsayımı altında derslerin çizelgenmesiyle ilgilenmektedir.

Hangi dersin hangi zaman diliminde yapılacağına yanısıra derslerin hangi dersliklerde yapılacağına belirlenmesi de bir alt problem olarak ortaya çıkmaktadır. Bu konuda derslik kapasitelerinin aşılması, bir derslikte yapılacak derslerin gereksinimlerinin karşılanması gibi zorunluluklar söz konusudur. Problem boyutu artacağı için genellikle ders-derslik-zaman dilimi ataması aynı anda yapılamamaktadır.

Ders-derslik atamasını ele alan ilk çalışmalara örnek olarak Ferland ve Roy (1985), Gosselin ve Truchon (1986) verilebilir. Ferland ve Roy (1985) ders çizelgeleme problemini iki aşamalı olarak ele almıştır. Birinci aşamada ders saatlerini zaman dilimlerine, ikinci aşamada dersliklere atamaktadır. Gosselin ve Truchon (1986)'un geliştirdikleri otomatikleştirilmiş derslik atama sistemi ile önceleri elle hazırlanan atamalar bilgisayar ortamında yapılmaya başlanmıştır. 1980'lere kadar yapılmış olan çalışmalarda derslik uygunlukları üzerinde çok durulmamıştır (Dinkel vd., 1989). Dinkel vd. (1989), derslik kullanım oranlarını arttırmak amacıyla model tabanlı bir

karar destek sistemi geliřtirmiřtir. Derslik ataması ile ilgili bir bařka alıřma da Carter ve Tovey (1992) tarafından gerekleřtirilmiřtir.

Öğreticilerin derslere atanması problemi de ders-derslik atama problemi gibi zaman boyutu içermeyen bir alt problemdir. Bu alt problemde amaç, her derse bir öğretici atamasının yapılmasıdır. Badri (1996) iki aşamalı olarak ele aldığı ders çizelgeleme probleminin ilk aşamasında öğretmenlerin tercihlerini en yüksek düzeyde karşılayacak şekilde öğretmenleri derslere atamaktadır. Kara ve Sağır Özdemir (1996), Sağır Özdemir ve Kara (1997) ile MirHassani (2006) de öğretici tercihlerinin dikkate alındığı bir başka çalışmadır. Öğretici tercihlerinin göz önüne alındığı çalışmalar derlendiğinde (Harwood, 1975; Hiton vd., 2004; Ozdemir ve Gasimov, 2004) bu tercihlerin genel olarak

- Öğreticinin tercih ettiğı derse atanması,
- Derslerin belirli zaman dilimlerine atanması,
- Bazı derslerin belirli dersliklere atanması,
- Bir günde ilk ve son ders arasındaki sürenin mümkün olduğunca kısa olması,
- Derslerin mümkün olduğunca en az sayıda güne atanması,
- Gece derslerinin mümkün olduğunca az olması

şeklinde belirlendiğı görülmüřtür.

Aubin ve Ferland (1989) ise öğrencilerin de derslere atandığı çalışmalara örnek olarak verilebilir. Ders akıřmalarının en küçüklenmeye alıřıldığı bir lise için ele alınan problemde, önce dersler zaman dilimlerine, ardında da öğrenciler derslere atanmaktadır.

### **2.2.2 Sınav çizelgeleme problemleri**

Sınav çizelgeleme problemi, ortak öğrencisi olan sınavların akıřması önlenecek şekilde, sınav salonlarının (derslik) kapasiteleri de dikkate alınarak sınavların zaman

dilimlerine atanması problemidir. Sınav zaman çizelgeleme ve ders zaman çizelgeleme problemleri arasında kesin bir ayırım yapmak çoğu zaman zordur. Birbirine oldukça benzer bu iki problemden, sınav zaman çizelgeleme probleminin izleyen özellikleri ayırıcı niteliktedir:

- Bir öğrenci için genellikle sınav çakışması olmamalıdır. Derslerin çakışması durumunda öğrenci başka dersler alabilirken, çakışan sınavlardan herhangi biri bırakılamaz.
- Kapasite yeterli ise bir derslikte birden fazla sınav yapılabilir. Ancak, bir derslikte aynı zaman diliminde iki farklı dersin yapılması söz konusu olamaz.
- Bir ders birden fazla derslikte yapılamazken, bir sınav birden fazla derslikte yapılabilir.

Sınav zaman çizelgeleme problemi, NP-tam alt problemler içeren oldukça karmaşık kombinatorik bir problemdir ve ele alınan kısıt ve amaçlar temelinde literatürde farklı şekillerde yer almaktadır (Carter ve Laporte,1998; Schaerf, 1999). Sıkı kısıtların tümünü sağlayan bir sınav zaman çizelgesi uygun bir çözüm olarak kabul edilir. En temel sıkı ve esnek kısıtlar aşağıda sıralanmıştır:

#### Sıkı kısıtlar

1. Bütün sınavlar birer zaman dilimine atanmalıdır.
2. Bir öğrencinin aynı zaman dilimine birden fazla sınavı atanmamalıdır.
3. Sınav(lar)ın yapılacağı derslik kapasitesi aşılmamalıdır.
4. Gözetmenler aynı zaman diliminde birden fazla sınava atanmamalıdır (gözetmen atama problemi genellikle sınav çizelgelemeden ayrı ele alınmaktadır).

Ders çizelgeleme problemlerinde olduğu gibi, sınav çizelgeleme problemlerinde de esnek kısıtlar her eğitim kurumunun yapısına göre farklılık gösterebilir. Burke ve Petrovic (2002), literatürde sıklıkla karşılaşılan esnek kısıtları izleyen şekilde belirlemiştir:

1. Bir sınav belirli bir zaman dilimine atanabilir. Özellikle, sınava girecek öğrenci sayısı çok olan sınavlar mümkün olduğunca çizelgenin başlangıcına

atanmaya çalışılır.

2. Bir sınavın başka bir sınavdan önce ya da sonra atanması gerekebilir (bu kısıt bazı kurumlarda sıkı kısıt olarak da ele alınabilir).
3. Bir öğrencinin sınavları aynı gün içinde mümkün olduğunca ardışık zaman dilimlerine atanmamalıdır.
4. Bir sınav belirli bir sınav salonuna atanabilir.

Bir başka çalışmada Burke vd. (1997)'nin belirlemiş olduğu esnek kısıtlar ise şunlardır:

1. Hiç bir öğrenci aynı günde iki sınava girmemelidir.
2. İki sınav arasında en az belirli bir süre bulunmalıdır.
3. Herhangi iki sınav aynı veya benzer gereksinimler sebebiyle aynı oturumda yapılmalıdır.

Tüm bu kısıtlar dışında başka sıkı ve esnek kısıtları da ele almak mümkündür. Örneğin tek bir sınav salonuna birden fazla sınavın atanması ya da bir sınavın birden fazla sınav salonuna atanması gibi.

Sınavların derslik ve zaman dilimlerine atanması aynı anda yapıldığında problemin boyutu oldukça büyümektedir. Bu nedenle geliştirilen çözüm yaklaşımları, makul sürelerde çözebilmek amacıyla problemi ayrıştırmaktadır (Eley, 2007). Bu süreçte çizelgeleme problemi, sırasıyla çözülmek üzere iki alt probleme bölünür. İlk alt problemde ilgili zaman dilimi için her dersliğin kapasitesi yerine toplam derslik kapasitesi dikkate alınarak sınavlar zaman dilimlerine atanır. İkinci aşamada da zaman dilimlerine atanmış sınavlar, dersliklere, kapasiteler ve ihtiyaçlar doğrultusunda atanmaktadır.

Herhangi bir zaman diliminde bir derslikte sadece bir sınavın yapılabildiği problemlerde, sınavların dersliklere atanması polinom zamanda çözülebilirken, bir derslikte birden fazla sınavın yapılması durumunda problem NP-tam olmaktadır (Schaerf, 1999).

1995 yılında Burke vd. (1996a) İngiltere'deki üniversitelerde sınav çizelgelerinin nasıl oluşturulduğunu ve farklı problem yapılarını belirlemek üzere bir anket düzenlemiştir. Anket sonuçlarına göre, ankete katılan üniversitelerin çoğu, sınavları dersliklere atarken problem yaşamaktadır. Ancak, Dammak v.d.'nin (2006) de belirttiği üzere, sınav zaman çizelgeleme problemlerinde sınavların dersliklere atanması literatürde henüz geniş bir yer almamıştır. Merlot vd. (2003) sadece toplam oturma kapasitesi kısıtının sağlanmasının yeterli olduğunu ve sınavların dersliklere atanmasının ayrı olarak ele alınması gerektiğini belirtmiştir. Azimi (2005) derslik kapasitelerine ilişkin kısıtları ele almayan çalışmalara örnek olarak verilebilir. Bullheimer (1998), derslik kapasitelerinin yeterli olduğu varsayımı ile sınavları sadece zaman dilimlerine atarken, Merlot vd. (2003), Erben ve Song (2004), Burke ve Newall (2004) ile Cheong vd. (2009) de ilgili zaman diliminde toplam oturma kapasitesi aşılmayacak şekilde sınavları zaman dilimine atamıştır.

Cole (1964), dersliklerin kapasitesine sınırlama getirirken, sınav zaman çizelgeleme üzerine yapılan çalışmaların çoğunda bir sınav salonunda sadece bir sınavın yapılmasına izin verilmektedir. Dammak vd. (2006), sınav-zaman ile sınav-derslik atama problemlerini birlikte çözmüştür. Bir dersliğe birden fazla sınavın atanabilmesi durumunda, problem bir ulaştırma problemi olarak modellenmiştir. Bir derslikte sadece bir, sadece iki ve ikiden fazla sınavın yapılabileceği durumlar için geliştirilen sezgisel algoritma, bir derslikte birden fazla sınavın yapılabileceği durumlarda uygun bir çözüm bulamamaktadır.

Carter vd. (1996) sınavları derslik temelinde atarken; Di Gaspero ve Schaerf (2000), Burke ve Newall (1999), Burke vd. (2004b) ile Caramia vd. (2007) sınavların dersliklere atanmasını derslik temelinde ele almamıştır. Bunun yerine, ilgili zaman diliminde sınavlara giren toplam öğrenci sayısı hesaplanarak boş olan tüm oturma kapasitesi aşılmayacak şekilde sınavlar zaman dilimlerine atanmıştır. Sınavların hem zaman dilimlerine hem de dersliklere atandığı bir diğer çalışmada da (Eley, 2007) farklı karınca kolonileri algoritmaları geliştirilmiştir.

Laporte ve Desroches (1984) ve Burke vd. (1993) ise bazı sınavların gerekli

özellikleri içeren özel dersliklere atanmasını ele almanın yanı sıra bir sınavın birden fazla derslikte yapılması durumunu da göz önüne almıştır.

Bir derslikte birden fazla sınavın yapılmasına izin veren, sınav-derslik-zaman çizelgeleme problemi için Johnson (1990) tarafından 0–1 tamsayılı model yanısıra, toplam ardışık sınav sayısının en küçüklenmesini amaçlayan bu modelin çözümü için de bir tavlama benzetimi algoritması geliştirilmiştir.

Lotfi ve Cervený (1991) bir günde ikiden fazla ve arka arkaya olan sınavlara giren öğrenci sayısının en küçüklenmesi gibi birden fazla amacı en iyilemeye çalışan dört aşamalı bir sezgisel algoritma geliştirmiştir. Algoritmanın son aşamasında sınavlar dersliklere atanmaktadır.

Sınav çizelgeleme probleminin genellikle sınavların zaman dilimi ve dersliklere atanmasından ayrı olarak ele alınan, bir alt problemi de *gözetmen-sınav atama* problemidir. Bir sınav döneminde, bir gözetmen genellikle birden fazla sınava atanır ve bir sınav için de birden fazla gözetmene ihtiyaç vardır. Yanı sıra gözetmenlerin, atandıkları sınavlar ve zaman dilimleri konusundaki tercihleri de önemlidir.

Literatürde eğitimsel zaman çizelgeleme problemine ilişkin çok sayıda çalışma bulunmasına rağmen, gözetmen atama problemini ele alan çalışması sayısı oldukça azdır. Bu durumun birkaç nedeni olabilir. Birincisi, sınavların zaman dilimi ve dersliklere atanmasının gözetmen atama problemine göre daha yaygın ve dah baskın olmasıdır. Göreceli olarak gözetmen ataması daha az önceliğe ve öneme sahip bir problem olarak görülmektedir. İkinci neden de hemen hemen bütün eğitim sistemlerinde sınav uygulaması olmakla birlikte sınavlarda her zaman gözetmen bulunmayabilmesidir. Ölçme ve değerlendirme sisteminde eğrilerin kullanılması ve onur kodları özellikle yurt dışı eğitim sistemlerinde bazen bu gerekliliği elimine etmektedir. Bir diğer neden öğrencilerin sınavlarına web servisleri üzerinden girmesini sağlayan üniversite politikalarıdır. Awad ve Chinneck (1998), Marti vd. (2000) ve Al-Yakoob vd. (2010) literatürde bu problemi ele alan çalışmalardır.

Awad ve Chinneck (1998) Carleton Üniversitesi için bir gözetmen atama modeli geliştirmiştir. Problemin çözümü için bir genetik algoritma çatısı altında probleme özgü sezgiseller ve basit bir kullanıcı arayüzü oluşturulmuştur. Ancak en iyi çözüm elde edilememiştir.

Marti vd. (2000) çok amaçlı gözetmen-sınav atama problemini bir üniversitede final sınavı için modellemiştir. Modele gözetmenlerin zaman dilimi tercihlerinin de eklenmesiyle, problemin karmaşıklığı artmıştır. Her bir gözetmen için saat cinsinden bir üst sınır vardır. Bu sınır, gözetmenlerin yönetim ile yapmış olduğu anlaşmalara bağlıdır. Tamsayılı programlama ile modellenen problemin çözümü için dağınık arama algoritması (*scatter search*) temelli sezgisel bir algoritma geliştirilmiş sonuçları da fayda fonksiyonu ile elde edilen sonuçlar ile karşılaştırılmıştır. Önerilen dağınık arama yönteminin tatmin edici sonuçlara ulaştığı görülmüştür.

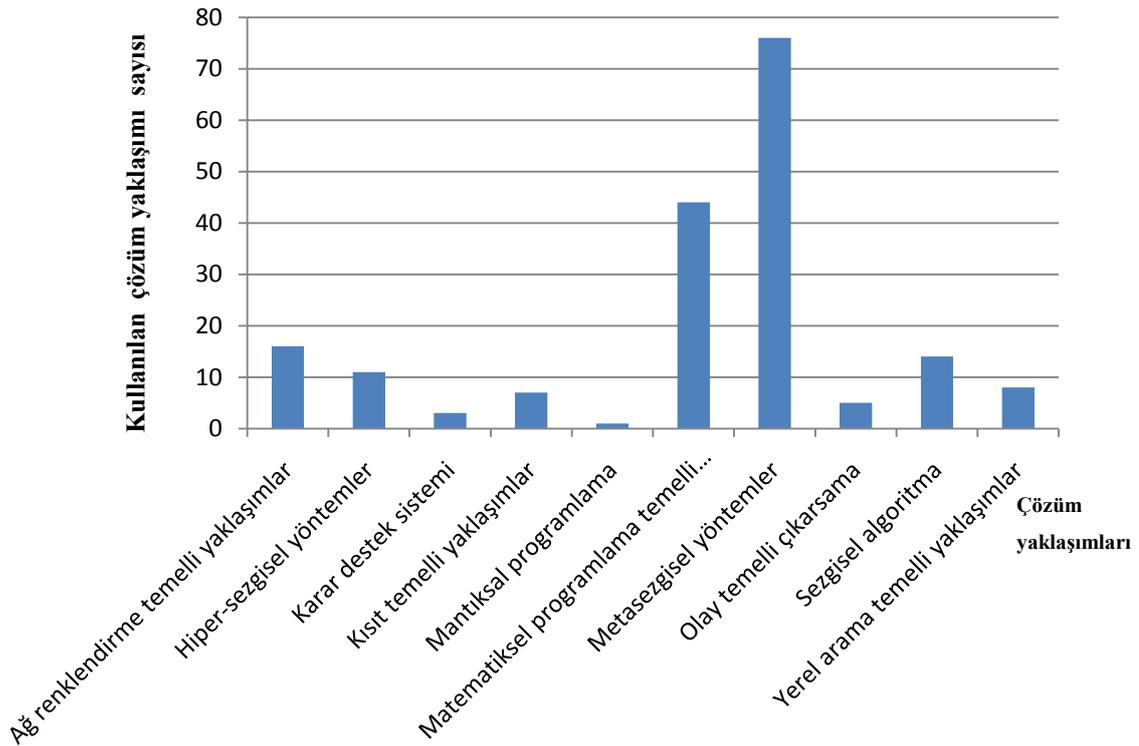
Al-Yakoob vd. (2010) ise sınav zaman çizelgeleme ve gözetmen atama alt problemlerini ele almıştır. Kuwait Üniversitesi gözetmen atama sistemi için bir karma tamsayılı programlama modeli geliştirmişlerdir. Bu çalışmada, gözetmenlerin gün ve zaman dilimi tercihleri de modele dahil edilmiştir.

Öztürk vd. (2006) çalışmalarında gözetmen tercih ve kısıtlarını dikkate alarak, istenmeyen zaman dilimlerine yapılan atamaları en küçüklerken, gözetmenlere dengeli yük atamasını gerçekleştirmek için çok amaçlı bir karma tamsayılı model geliştirilmiştir. Öztürk vd. (2010) ise çalışmalarında gözetmen tercih ve kısıtlarını web tabanlı bir karar destek sistemi aracılığı ile almıştır. Bu çalışmada Öztürk vd.'nin (2006) önerdikleri modeli geliştirerek, Eskişehir Osmangazi Üniversitesi'nde seçilen bir uygulama problemi için en iyi çözüm elde edilmiştir.

### 2.3 Çözüm Yaklaşımları

Eğitimsel zaman çizelgeleme problemlerinin çözümünde matematiksel programlama, sezgiseller, metasezgiseller ve kısıt programlama (constraint

programming) gibi farklı yaklaşımlar kullanılmaktadır. Bölüm 2.2’de ele alınan çalışmalar incelendiğinde, metasezgisellerin ve matematiksel programlama temelli yaklaşımların bu problemlerin çözümünde en sık kullanılan yaklaşımlar olduğu görülmüştür (Şekil 2.4).



**Şekil 2.4** Eğitimsel zaman çizelgeleme problemlerinin çözümünde kullanılan çözüm yaklaşımlarının dağılımı

Zaman çizelgeleme problemlerinin ele alındığı ilk yıllarda, çözümler elle yapılmaktaydı. Yaygın yaklaşım, en fazla kısıtı olan dersin önce çizelgelenmesi ve ardından tüm dersler çizelgelenene kadar bu yaklaşımın devam etmesiydi. Kısmi çözüm üretmek zaman çizelgeleri oluşturan bu teknikler, sezgiseller olarak adlandırılmaktadır (Schaerf, 1999).

Sezgisellik (sezgisel kurallar, sezgisel yöntem), problemin durum uzayı çok büyük olduğunda çözümün aranmasını kesin biçimde sınırlayan herhangi kural, strateji, hile, sadeleştirme ve diğer etmenlerin kullanımınıdır. Dolayısıyla sezgisellik, problem karmaşıklık içerdiğinde, çözüm yolunun bulunmasındaki yardımcı anahtardır (Nabiyev, 2003).

Bu sezgisel yaklaşımlar, sonraki yıllarda geliştirilen metasezgiseller öncesinde bir programlama dili kullanarak, probleme özgü olarak geliştirilen ve yasaklı arama, tavlama benzetimi gibi bilinen bir sezgisel yaklaşımı temel almayan yaklaşımlardır.

Aubin ve Ferland (1989), liselerde atama ve öğrencilerin gruplanması problemlerinin aynı anda ele alındığı bir problemin çözümü için sezgisel bir yaklaşım önermişlerdir. Sınav zaman çizelgelerinin oluşturulmasında bir sezgiselin kullanıldığı ilk çalışmalara Cole (1964) ve Broder (1964) örnek verilebilir (Johnson, 1990). Bu çalışmaların ardından, öncelikli olarak zor olan sınavların çizelgelenmesi mantığı ile çalışan ağ renklendirme temelli farklı sezgiseller de geliştirilmiştir (de Werra, 1997). Eğitimsel zaman çizelgeleme problemlerinin çözümünde sezgisel yöntemlerin kullanıldığı çalışmalar Çizelge 2.1’de verilmiştir.

Werra (1985), Carter (1986), Balakrishnan vd. (1992), Carter ve Laporte (1996), Bardadym (1996), Burke et al. (1997), Schaerf (1999) Burke and Petrovic (2002) ile Petrovic ve Burke (2004) tarafından eğitimsel zaman çizelgeleme problemleri üzerine geliştirilen çözüm yaklaşımları verilmiştir.

Carter ve Laporte (1996) bu yaklaşımları aşağıda verilen dört grupta incelemiştir:

- Küme (*cluster*) yöntemleri

Bu yöntemde problem, olaylardan oluşan belli sayıda kümeye bölünür. Bu kümelerin her biri tüm sıkı kısıtları sağlayacak şekilde tanımlandıktan sonra, esnek kısıtlar da sağlanacak şekilde kümeler zaman dilimlerine atanır. Bu yöntemin en büyük sakıncası, olaylar kümesinin algoritmanın en başında oluşturularak sabitlenmesi ve böylece düşük kaliteli çizelgelerin oluşturulması olarak belirlenmiştir (Burke and Petrovic, 2002). Bu yöntemin kullanıldığı

çalışmalara örnek olarak White and Chan (1979), Fisher and Shier (1983) ve Balakrishnan vd. (1992) verilebilir.

- Sıralı (*sequential*) yöntemler  
Zaman çizelgeleme problemlerini ağ problemleri olarak ele alan yöntemler sıralı yöntemlerdir. Genellikle, sezgisel bir yaklaşımla olaylar sıralanır ve ardından olaylar sıralı olarak geçerli zaman dilimlerine atanır (Carter, 1986; de Werra, 1985).
- Metasezgiseller (*meta-heuristics*)
- Kısıt temelli yöntemler (*constraint-based approaches*)

**Çizelge 2.1** Sezgisellerin kullanıldığı çalışmalar

Yıl	Yazar	Gözönüne alınan bileşenler						Yayında tanımlanan başlık
		Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
1964	Cole, A.J.		+			+	+	Exam (University)
1968	Wood, D.C.		+			+	+	Exam (University)
1973	Appleton, D.R.		+				+	University
1979	Bloomfield, S.D. and McSharry, M.M.	+	+				+	Course
1981	de Gans, O.B.	+	+	+			+	Secondary Schools
1986	Sabin, G.C.W. and Winter, G.K.	+			+		+	University
1986	Glassey, C.R. and Mizrach, M.	+	+					University
1990	Johnson, D.		+			+	+	Exam (University)
1991	Lotfi, V., and Cerveny, R.		+			+	+	Exam (University)
1992	Cangalovic, M. and Schreuder, J.A.M.	+					+	Class-Teacher
1995	Sampson, S.E., Freeland, J.R. and Weiss, E.N.	+	+		+			Class
1996	Wright, M.	+	+				+	School
2006	Dammak A., Elloumi A. and Kamoun H		+			+		Exam

Petrovic ve Burke (2004) bu kategoriye çok ölçütlü yaklaşımlar (*multi-criteria approaches*), durum-temelli çıkarsama teknikleri (*case-based reasoning techniques*) ve hiper-sezgiselleri (*hyper-heuristics*) eklemiştir.

Tüm bu tanımlama ve sınıflama kapsamı içinde problemin çok amaçlı ele alınması halinde çözüm yöntemleri değişmekte ve özel bir önem kazanmaktadır. Genel çözüm başlıkları yine eniyileme veya sezgisellere dönük olmakla birlikte problem çok amaçlı ele alındığında başka bir düzleme geçilmektedir. İzleyen bölümlerde ders ve sınav zaman çizelgeleme problemlerinin çözümü için kullanılan teknik ve yaklaşımlara ait literatür taraması verilmiştir. Çok amaçlı çözüm yaklaşımlarını kullanan çalışmalar da en son alt bölümde ayrıca ele alınmıştır.

### 2.3.1 Matematiksel programlama temelli yaklaşımlar

Çözüm tekniği olarak sezgisellerin kullanımının ardından tamsayılı programlama, ağ renklendirme gibi daha genel yöntemlerin kullanımına geçilmiştir. Akkoyunlu (1973) ve Csima ve Gotlieb (1961), derslerin zaman dilimine atanması problemini bir doğrusal programlama modeli ile ele almıştır. Akkoyunlu (1973), üniversitede sadece bir bölüm için, atamalar için uygun dersliklerin bulunduğu varsayımı altında, dersleri zaman dilimine atayan bir doğrusal program geliştirmiş ve bütünsel en iyi çözümü elde etmiştir. Lawrie (1969), zaman çizelgeleme problemini amaç fonksiyonu olmayan ve doğrusal kısıtları olan bir tamsayılı programlama problemi olarak modellemiştir. McClure ve Wells (1984) ile Tripathy (1980) zaman çizelgeleme problemlerini doğrusal bir tamsayılı programlama problemi olarak tanımlamıştır. Ancak tüm bu teknikler, sadece küçük boyutlu problemlerde uygulanabilmektedir. 0–1 tamsayılı en iyileme tekniklerinin kullanımı problem boyutu büyüdükçe güçleşmektedir.

Sherman (1963), üniversitelerde ders çizelgelerinin oluşturulması için bir tamsayılı doğrusal programlama modeli önermiştir (Cole, 1964). Lotfi (1991), bir gezgin satıcı problemi olarak modellediği çalışmalarda (Colijn, 1979; White ve Chan, 1979) ardışık sınavlara giren öğrenci sayısında azalma olmasına rağmen, bir günde iki

veya daha fazla sınava giren öğrenci sayısında artış olduğunu belirtmiştir.

Harwood ve Lawless (1975), 7 öğretim elemanı ve 12 dersten oluşan küçük boyutlu bir sistemde, öğretim elemanlarının tercihlerini göz önünde bulundurarak, karma tamsayılı hedef programlama ile öğretim elemanlarını derslere ve dersleri de zaman dilimlerine atamıştır.

Hinchliffe (1973), öğrenci ve öğretmenler açısından derslikler arasındaki gidış gelişlerde kaybolan zamanın en küçüklenmesi için hangi dersin hangi zaman diliminde yapılacağı bilgisi doğrultusunda, derslikleri zaman dilimlerine atamıştır. Shih ve Sullivan (1977) ise, öğretmenleri derslere dersleri de zaman dilimlerine atamak için iki aşamadan oluşan bir 0-1 tamsayılı programlama modeli geliştirmiştir. Brunetta ve DePoli (1997), tamsayılı programlama temelli bir karar destek sistemi ve Internet üzerinden kullanıcının 39 tane dersin zaman çizelgesini oluşturabileceği bir platform geliştirmiştir.

Birbas vd. (1997) Yunanistan'daki liselerde kullanılan haftalık ders çizelgelerini oluşturmak için bir 0-1 tamsayılı programlama modeli geliştirmiştir. Daskalaki vd. (2004) de Yunanistan'da bir üniversitenin ders-derslik-zaman çizelgeleme problemi için 0-1 tamsayılı programlama modeli geliştirmiştir. Bu çalışmada ayrıca çizelgelenecek derslerin haftada bir saatten fazla yapılması nedeniyle ilgili dersler için ardışıklık kısıtları da dikkate alınmıştır.

Dimopoulou ve Miliotis (2001), dersleri derslik ve zaman dilimlerine atadıkları çalışmalarında, problemi bir kısıt sağlama modeli ile ele almıştır. Ozdemir ve Gasimov (2004) ise, ders-öğretim elemanı ataması problemini yönetim ve öğretim elemanlarının tercihlerini göz önünde bulunduran, çok amaçlı bir yapıyla ele almış ve bir tamsayılı model ile çözmüştür. Bu çalışmadaki kısıtlar, her bir öğretim elemanının haftalık yükünün belirli sınırlar arasında olması ve her bir dersin sadece bir öğretim elemanına atanması şeklindedir. Ismayilova vd., 2005'teki çalışmalarında ders-öğretim elemanı-zaman dilimi ataması problemini yönetim ve öğretim elemanlarının tercihlerini göz önünde bulundurarak çok amaçlı olarak ele almıştır.

Al-Yakoob ve Sherali (2006) Kuwait Üniversitesi'nde ders çizelgeleme problemi çözümü için, Ozturk vd. (2010) de gözetmen-sınav atama probleminin çözümü için karma tamsayı model geliştirmiştir.

Eğitimsel zaman çizelgeleme problemlerinin çözümünde matematiksel programlama temelli yöntemlerin kullanıldığı çalışmalar Çizelge 2.2'de verilmiştir.

**Çizelge 2.2** Matematiksel programlama temelli çözüm yöntemlerinin kullanıldığı çalışmalar

Gözönüne alınan bileşenler								
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayında tanımlanan başlık
1966	Carlson, R.C. and Nemhauser, G.L.	+					+	Class-Teacher
1969	Lawrie, N.	+					+	School
1973	Akkoyunlu, E.A.	+					+	University
1973	Hinchliffe, P.D.		+				+	School
1975	Harwood, G.B. and Lawless, R.W.	+		+			+	Course
1977	Shih, W. and Sullivan, J.A.	+		+			+	Course
1980	Tripathy, A.	+					+	Course
1983	Lee, S.M. and Schniederjans, M.J.			+				School
1984	McClure, R.H. and Wells, C.E.							Faculty-Course
1984	Tripathy, A.	+	+				+	School
1985	Ferland, J.A. and Roy, S.	+	+					Course (university)
1986	Gosselin, K. and Truchon, M.		+				+	University classroom
1987	Schniederjans, M.J. and Kim, G.C.	+		+				Course
1989	Aubin, J. and Ferland, J.A.	+		+			+	Timetabling
1992	Carter, M.W. and Tovey, C.A.	+	+					Classroom assignment
1992	Tripathy, A.	+	+				+	Class
1993	Johnson, D.	+		+			+	Course
1994	Ferland, J.A. and Fleurent, C.	+			+			Course
1996	Badri, M.A.	+		+			+	University
1997	Birbas, T., Daskalaki, S. and Housos, E.	+		+			+	School
1998	Wood, J. and Whitaker, D.	+			+		+	School

Çizelge 2.2 devam

Gözönüne alınan bileşenler								
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayında tanımlanan başlık
1998	Badri, M.A., Davis, D.L., Davis, F.D., and Hollingsworth, J.	+		+			+	Course (university)
2001	Dimopoulou, M. and Miliotis, P.	+	+				+	Course (University)
2002	Baker, K.R., Magazine, M.J., and Polak, G.G.	+					+	Course
2004	Dimopoulou, M. and Miliotis, P.	+	+				+	University
2004	Daskalaki, S., Birbas, T., and Housos, E.	+	+				+	Course (university)
2005	Avella, P., and Vasil'Ev, I.	+	+				+	Course (university)
2005	S. Daskalaki T. Birbas	+	+	+	+		+	University
2006	Al-Yakoob S.M. and Sherali H.D.	+		+			+	Class-Faculty
2006	MirHassani S.A.	+		+			+	Course (university)
2006	Günalay, Y. and Şahin, T.	+		+			+	Course (university)
2007	S. M. Al-Yakoob and H.D.Sherali	+	+				+	Class
2007	Ismayilova, N.A., Sagir, M. and Gasimov, R.N.	+		+			+	Faculty-Course
2007	Schimmelpfeng, K. and Helber, S.	+	+				+	Course (university)
2007	Al-Yakoob, S.M., Sherali, H.D. and Al-Jazzaf, M.			+		+		Proctor
2008	Boland, N., Hughes, B.D., Merlot, L.T.G., and Stuckey, P.J.	+		+	+			Course (high school)
2008	Burke, E.K., Marecek, J., Parkes, A.J., and Rudová, H.	+	+				+	Course (university)
2010	Sarin, S.C., Wang, Y. Varadarajan, A.	+	+				+	Course (university)
2010	Ozturk, Z.K., Ozturk, G. ve Sagir, M.			+		+		Invigilator

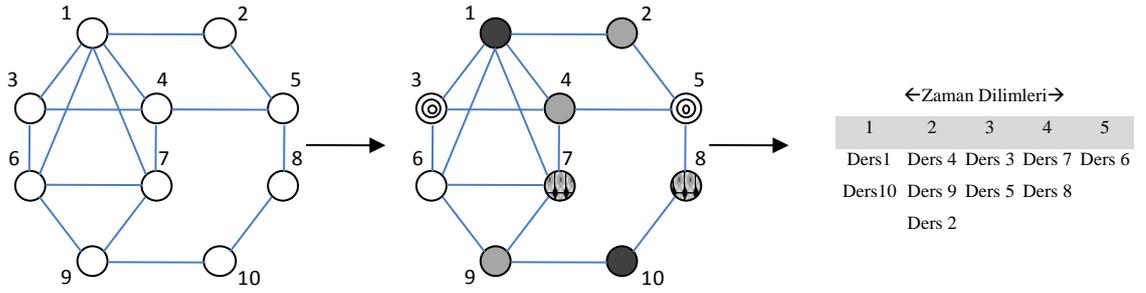
### 2.3.2 Ağ renklendirme temelli yaklaşımlar

Temel zaman çizelgeleme problemleri, ağ teorisindeki köşe renklendirme (vertex coloring) problemine eşdeğerdir. Bir eğitimsel zaman çizelgeleme probleminde ağ, izleyen şekilde oluşturulur:

- Her ders/sınav bir köşe ile temsil edilir.

- ii. Herhangi iki dersin/sınavın en az bir ortak öğrencisi varsa ilgili iki köşe bir kenar ile birleştirilir. Bir kenar ile birleştirilen iki köşe (ders/sınav) ortak kaynak kullanıyor demektir. Dolayısıyla, iki ders/sınav aynı anda olamaz.

Ağ renklendirme problemi olarak modellenen zaman çizelgesinde her zaman dilimine karşı bir renk kullanılır. Böylece, köşelerden herhangi komşu (bir kenar ile birleştirilmiş) iki tanesi aynı renge sahip olmayacak şekilde çizelge oluşturulur. Ağ renklendirme ve basit bir zaman çizelgeleme problemi ile arasındaki ilişki Şekil 2.5'te verilmiştir.



(1) Basit bir zaman çizelgeleme problemi önce boyanmış serime dönüştürülür. (Bu örnekte 10 olay / 10 renk köşesine çizelgenmeye çalışılmıştır.)

(2) Daha sonra bu örnek için herhangi bir yolla bir çözüm bulunur. ( Bu özel çözüm en iyi sayıda renk kullanıyor)

(3) Renklendirilmiş çözüm daha sonra geçerli her rengin bir zaman dilimini gösterdiği bir zaman çizelgesine dönüştürülür. Bu sayede komşu köşeler aynı zaman dilimine atanmaz.

**Şekil 2.5** Ağ renklendirme ve basit bir zaman çizelgeleme problemi ile arasındaki ilişkinin gösterimi (Lewis, 2008'den)

Ders çizelgeleme problemlerinde olduğu gibi sınav zaman çizelgeleme problemlerinde de ağ renklendirme sezgiselleri sıklıkla kullanılmaktadır (Cole, 1964; Peck ve Williams, 1966; Welsh ve Powell, 1967; Laporte ve Desroches, 1984). Cole (1964)'un çalışmasında sınavlar için ardışıklık, ön atama ve derslik kapasiteleri ile bazı sınavlar için de sınav sırası koşulları göz önüne alınmıştır. Bu çalışmanın önemli katkılarından biri çakışma matrisinde dersleri çakışan öğrenci sayıları yerine sadece

çakışan derslerin ele alınması ile problemin boyutunun azalmasıdır.

Sınav zaman çizelgeleme konusunda yayınlanan ilk çalışmalardan biri 1964 yılında Broder tarafından yapılmıştır (Carter, 1986). Aynı zaman diliminde birden fazla sınavı olan öğrenci sayısının en küçüklenmeye çalışıldığı bu çalışmada bir ağ renklendirme algoritması geliştirilerek algoritmanın birçok defa çalıştırıldığı bir Monte Carlo benzetimi yaklaşımı önerilmiştir. En iyi çözümün elde edildiği tekrar, problemin çözümü olarak seçilmektedir.

Sınav zaman çizelgeleme problemi; derslik kapasiteleri, ön atamalar, ardışıklık gibi ek kısıtların da probleme dahil olmasıyla ağ renklendirme probleminden farklılaşmaktadır. Dolayısıyla ağ renklendirme algoritmalarının çeşitli farklı versiyonları geliştirilmiştir (Peck ve Williams, 1966; Welsh ve Powell, 1967; Laporte ve Desroches, 1984; ) ve “en büyük dereceli önce (*largest degree first*)” gibi farklı ağ renklendirme sezgiselleri geliştirilmiştir. Tüm bu algoritmaların temel çalışma prensibi, öncelikle çizelgenmesi en zor olayların ele alınmasıdır.

Eğitimsel zaman çizelgeleme problemlerinin çözümünde ağ renklendirme temelli yöntemlerin kullanıldığı çalışmalar Çizelge 2.3’te verilmiştir.

Lai vd.’nin (2008) belirttiği üzere, ağ renklendirme temelli teknikler küçük ölçekli problemler için başarılı sonuçlar verse de, büyük ölçekli problemlerde başarılı değildir ve değişiklikler de modele kolaylıkla yansıtılamamaktadır.

**Çizelge 2.3** Yerel arama tekniklerinin kullanıldığı çalışmalar

Yıl	Yazar	Gözönüne alınan bileşenler						Yayımda tanımlanan başlık
		Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
1986	Carter, M.W.							Literatür taraması
1991	Hertz, A.	+	+				+	Course (university)
1992	Kiarer, L. and Yellen, J.	+					+	Course (university)
1993	Burke, E.K., Elliman, D.G., and Weare, R.		+			+	+	Exam (University)
1996	Carter, M.W., Laporte, G. and Lee, S.Y.					+	+	Exam
1997	de Werra, D.	+					+	Course
1997	Weitz, R.R. and Lakshminarayanan, S.				+	+	+	Exam
1997	de Werra, D.							Survey
2000	de Werra, D., Hertz, A., Kobler, D., and Mahadev, N.V.R.	+	+				+	Class-Teacher
2001	Hilton, A.J.W., Slivnik, T., and Stirling, D.S.G.	+	+				+	School/conference
2002	Burke, E.K. and Newall, J.P.					+	+	Exam
2002	Asratian, A.S, and de Werra, D.	+	+				+	University
2004	Burke, E.K. and Newall, J.P.					+	+	Exam
2004	Burke, E., de Werra, D. and Kingston, J.							Survey
2005	Munoz, S., Ortuno, M.T., Ramirez, J., and Yanez, J.					+	+	Exam

### 2.3.3 Yerel arama temelli yaklaşımlar

Yerel arama (*Local Search*) yöntemleri, bir çözümden o çözümün komşuluğundaki başka bir çözüme geçerek çözüm uzayını araştırma fikrini temel alan yöntemlerdir. Bir durma ölçüti sağlanana kadar çözüm uzayını arayan bu algoritmalar ile elde edilen çözüm yerel en iyi çözümdür. Bu algoritmalar bütünsel en iyi çözümü garanti etmezler (Shaer vd., 2000).

Temel yerel arama yöntemlerinden biri olan Tepe Tırmanma algoritması, bir yerel

en iyi noktaya ulaşıncaya kadar bir çözümden daha iyi bir çözüme geçer ve ardıştırmalı bir iyileştirme yapar. Ancak bu yöntem ile genel en iyiden daha kötü bir noktaya ulaşabilir. Bu durumdan kaçınmak için, mevcut çözümden kötü olan çözümleri de kabul ederek daha iyi sonuçlar elde edebilen metasezgiseller geliştirilmiştir. Arama uzayındaki farklı komşuluk yapıları ve hareket işlemleri ile Yasaklı Arama, Tavlama Benzetimi ve Evrimsel Algoritmalar gibi yerel arama temelli teknikler genellikle metasezgiseller olarak adlandırılmaktadır. Tanımlanmış bir amaç fonksiyonu ile arama yönlendirilir ve elde edilen çizelgelerin kalitesi de değerlendirilir. Bir başlangıç çözümü ile başlar ve yerel değişimler ile mevcut çözümü iyileştirmeye çalışırlar.

Çizelge 2.4'te eğitimsel zaman çizelgeleme problemlerinin çözümünde yerel arama tekniklerinin kullanıldığı çalışmalar verilmiştir.

**Çizelge 2.4** Yerel arama tekniklerinin kullanıldığı çalışmalar

Yıl	Yazar	Gözönüne alınan bileşenler					Yayımda tanımlanan başlık	
		Ders	Derslik	Öğretici	Öğrenci	Sınav		Zaman
2001	Burke, E., Bykov, Y., and Petrovic, S.					+	+	Exam
2001	Caramia, M., Dell'Olmo, P. and Italiano, G.F.					+	+	Exam
2001	Muller, T. and Bartak, R.	+	+	+			+	School
2001	Schaerf, A., and di Gaspero, L.					+	+	Exam (University)
2001	Schaerf, A., and di Gaspero, L.	+	+				+	Course (university)
2002	Rossi-Doria, O., Blum, C., Knowles, J., Sampels, M., Socha, K., and Paechter,	+	+				+	Course (university)
2003	Casey, S. and Thompson, J.					+	+	Exam
2007	Caramia, M., and Dell'Olmo, P.					+	+	Exam (University)

### 2.3.4 Metasezgisel yöntemler

Yıllar temelinde eğitimsel zaman çizelgeleme problemlerinin çözümünde

kullanılan yaklaşımlar incelendiğinde özellikle 2000’li yıllardan itibaren metasezgisel çözüm yaklaşımlarının sıklıkla kullanıldığı görülmüştür (Çizelge 2.5).

Çok sayıda değişkeni ve kısıtı olması nedeniyle, özellikle büyük ölçekli zaman çizelgeleme problemlerini bir tamsayılı matematiksel model yardımıyla çözmek konusunda tam bir başarı sağlanamamaktadır (Gunawan vd., 2007; Burke vd., 2008). Farklı yönetsel kararların da etkisiyle eğitimsel zaman çizelgeleme problemlerinin boyutu büyümekte ve yapısı karmaşıklaşmaktadır. Dolayısıyla, çözüm için gerekli süre üstel olarak arttığından, son yıllarda problemlerin çözümü için farklı metasezgiseller geliştirilmiştir. Örnek olarak tavlama benzetimi metasezgiselinin bu alandaki ilk uygulamalarından biri Abramson (1991) tarafından yapılmıştır.

**Çizelge 2.5** Yıllar temelinde eğitimsel zaman çizelgeleme problemlerinin çözümünde kullanılan yaklaşımlar

	Ağ renklendirme temelli yaklaşımlar	Hiper-sezgisel yöntemler	Karar destek sistemi	Kısıt temelli yaklaşımlar	Mantıksal programlama	Matematiksel programlama temelli yaklaşımlar	Metasezgisel yöntemler	Olay temelli çıkarsama	Sezgisel algoritma	Yerel arama temelli yaklaşımlar
1964									1	
1966						1				
1968									1	
1969						1				
1973						2			1	
1975						1				
1977						1				
1979									1	
1980						1				
1981									1	
1982						1				

Çizelge 2.5 devam

	Ağ renklendirme temelli yaklaşımlar	Hiper-sezgisel yöntemler	Karar destek sistemi	Kısıt temelli yaklaşımlar	Mantıksal programlama	Matematiksel programlama temelli yaklaşımlar	Metasezgisel yöntemler	Olay temelli çıkarsama	Sezgisel algoritma	Yerel arama temelli yaklaşımlar
1983						1				
1984						2				
1985						2				
1986	1		1			1		2		
1987						1				
1989	2					1				
1990							2		1	
1991							2		1	
1992	1				1	2	3		1	
1993	1					1	1			
1994						1	4			
1995							1		1	
1996	1					1	4		1	
1997	3					1	1			
1998				1		2	3			
1999							3			
2000	1		1	1			2	1		
2001	1					1	7			3
2002	2	2				1	7			1
2003		3		2			6			1
2004	2	2	1	1		3	10			
2005	1	2		1		2	6			
2006		1				4	6	3	1	1
2007		1				5	6	1	1	2
2008				1		2	1			
2009							1			
2010						2				

Kombinatorik en iyileme problemlerinin çözümünde sıklıkla kullanılan

metasezgiseller

- Nokta temelli metasezgiseller
  - Tavlama benzetimi
  - Yasaklı arama
  - Yönlendirilmiş yerel arama (*Guided local search-GLS*)
  - Değişken komşuluk araması (*Variable neighbourhood search-VNS*)
- Topluluk temelli metasezgiseller
  - Genetik algoritmalar
  - Karınca kolonileri
  - Evrimsel stratejiler
  - Genetik programlama
  - Memetik algoritmalar

başlıklarında toplanabilir.

Yerel eniyi çözümlerden kaçınan arama stratejilerini kullanan metasezgisel yöntemlerin bir üstünlüğü de yüksek kaliteli çözümler üretmeleridir. Ancak, birçok metasezgiselin de başlangıç parametrelerine oldukça bağımlı olması durumu göz ardı edilmemelidir.

### Tavlama benzetimi

Tavlama Benzetimi algoritması, metallerin ısı işleme ile bir en iyileme problemine çözüm araştırma olayları arasındaki benzerlikten esinlenerek ortaya konulmuştur (Karaboğa, 2004).

Derslik kapasitelerinin yeterli olduğu varsayımı altında, öğrencilerin çalışmaları için, sınavları arasındaki gerekli zamanı en büyüklemeyi ve yönetsel açıdan da zaman dilimleri sayısını en küçüklemeyi amaçlayan Bullnheimer (1998), sınav çizelgeleme problemini karesel atama problemi olarak modellemiştir. Gerçek bir problemin çözüldüğü bu çalışmada, problem boyutları çok küçük (15 iş günü-15 sınav-391

öğrenci) olmasına rağmen uygun çözüm için sezgisel bir algoritmanın gerekliliği nedeniyle bir tavlama benzetimi algoritması geliştirilmiştir.

500'den fazla öğrencinin gireceği sınavların sabah oturumunda olması gibi, uygulama yerine özgü sıkı kısıtların belirlendiği Merlot vd. (2003)'nin çalışmasında ise tavlama benzetimi algoritması için başlangıç çözüm, kısıt temelli bir başka algoritma ile bulunmuştur.

Tavlama benzetimi algoritmalarında da arama sürecinde farklı hareket stratejileri uygulanmaktadır. Örneğin Bullheimer (1998) hem zaman dilimi hem de sınavlar arasında değişiklik yaparken Burke vd. (2004b) sadece sınavlar arasında hareketleri gerçekleştirmiştir.

Dowland (1990) ve Kostuch (2005) eğitimsel zaman çizelgeleme problemlerini tavlama benzetimi yaklaşımı ile ele alan diğer çalışmalardır. Eğitimsel zaman çizelgeleme problemlerinin çözümünde tavlama benzetiminin kullanıldığı çalışmalar Çizelge 2.6'da verilmiştir.

**Çizelge 2.6** Tavlama benzetiminin kullanıldığı çalışmalar

Yıl	Yazar	Çözününe alınan bileşenler					Yayında tanımlanan başlık	
		Ders	Derslik	Öğretici	Öğrenci	Sınav		Zaman
1990	Dowland, K.A.	+	+				+	University
1991	Abramson, D.	+	+	+			+	School
1998	Thompson, J.M. and Dowland, K.A.					+	+	Exam
1998	Bullheimer, B.					+	+	Exam (University)
2002	Chiarandini, M. and Stütze, T.	+	+				+	Course (university)
2003	Merlot, L.T.G., Boland, N., Hughes, B.D., and Stuckey, P.J.					+	+	Exam
2004	Burke, E., Bykov, Y., Newall, W., and Petrovic, S.					+	+	Exam (University)
2005	Kostuch, P.	+	+				+	Course (university)
2007	Avella, P., D'Auria, B., Salerno, S., and Vasil'ev, I.	+		+			+	School

### Yasaklı arama

Yerel arama yöntemlerinin karşılaştığı *yerel en iyiye takılma problemi*, elde edilen çözümü iyileştirmeyen hareketlere (*move*) de izin vererek önleyen Yasaklı Arama, Glover tarafından 1986'da kombinatoriyel problemlerin çözümü için önerilmiştir (Gendreau, 2002). Yasaklı Arama algoritmasının yerel en iyiyi aşmak amacıyla kullandığı temel ilke, değerlendirme fonksiyonu tarafından her ardıştırmada en yüksek değerlendirme değerine sahip hareketin bir sonraki çözümü oluşturmak amacıyla seçilmesine dayanmaktadır (Karaboğa, 2004).

Yasaklı arama sezgiselinin sınav zaman çizelgeleme problemlerinde kullanımının örnekleri Hertz (1991), Costa (1994), Alvarez-Valdes vd. (2002) ile Arntzen ve Lookkelangen (2005) tarafından verilmiştir. di Gaspero ve Schaerf (2000), ağ renklendirme temelli sezgiselleri kullanan iki yasaklı arama algoritması geliştirmişlerdir. Bazı sınavlar için ön atamaların yapıldığı bu çalışmada, sınavların dersliklere atanması için geliştirilen kısıt, herhangi bir zaman diliminde yapılacak sınavların toplam öğrenci sayısının, o anda uygun olan toplam kapasiteyi aşmaması üzerinedir. Geliştirilen algoritmalarla elde edilen sonuçların karşılaştırılması için hem gerçek problemler hem de rassal olarak geliştirilen problemler çözdürülmüş, ancak sadece rassal problemlerde ön atama ve tercihler dikkate alınmıştır. Elde edilen sonuçlar literatürdeki sonuçlarla rekabet edecek düzeydedir.

Doğrudan bir çizelgeleme problemi olmasa da, Alvaraez-Valdes vd. (2000) ders çizelgeleme probleminin bir alt problemi olan öğrenci-ders atanması problemi için iki aşamalı bir Yasaklı Arama algoritması geliştirmiştir. Birinci aşamada öğrenci tercihleri karşılanacak şekilde, her öğrenci için dengeli bir atanmanın yapıldığı eniyi çözümler elde edilir. İkinci aşamada da tüm çözümler bir araya getirilerek dersler arasında tatmin edici bir denge oluşturulmaktadır.

Eğitimsel zaman çizelgeleme problemlerinin çözümünde yasaklı aramanın kullanıldığı çalışmalar Çizelge 2.8'de verilmiştir.

Çizelge 2.7 Yasaklı aramanın kullanıldığı çalışmalar

Gözönüne alınan bileşenler								
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayımda tanımlanan başlık
1992	Hertz, A.	+	+				+	Course
1994	Costa, D.	+					+	School
1996	Valdes, R.A., Martin, G. and Tamarit, J.M.	+					+	School
2000	Alvarez-Valdes, R., Crespo, E. and Tamarit, J.M.	+			+			Student (university)
2000	di Gaspero, L., and Schaerf, A.					+	+	Exam (University)
2002	Alvarez-Valdes, R., Crespo, E. and Tamarit, J.M.	+	+				+	University
2004	White, G.M., Xie, B.S., and Zonjic, S.					+	+	Exam
2005	H. Arntzen A.Lokketangen	+	+				+	University
2006	Andrea Zampieri and Andrea Schaerf			+		+	+	Exam (University)
2006	Beyrouthy, Camille / Burke, Edmund K / Landa-Silva, Dario / McCollum, Barry / McMullan, Paul / Parkes, Andrew J	+	+				+	Course (university)

### Evrimsel Algoritmalar

Evrimsel programlama, 1960'lı yıllarda I. Rechenberg'in *Evrimsel stratejileri* adlı çalışmasıyla gündeme gelmiştir (Nabiyev, 2005). Son yıllarda kullanımı yaygınlaşan evrimsel algoritmalar, doğal seçim ve doğal genetik mekanizmasına dayanan araştırma algoritmalarıdır. Evrimsel algoritmalar formüle edilmişlere göre genetik algoritmalar, evrimsel programlama, evrim stratejileri ve genetik programlama gibi değişik isimlerle anılmaktadırlar. Genetik algoritmalar (GA) ile evrim stratejileri arasındaki en önemli fark; genetik algoritmaların çaprazlama ve mutasyon işlemcilerini, evrim stratejisinin ise sadece mutasyon işlemi kullanmasıdır. Evrim stratejileri kullanıldığında problem verilerinin kodlanmasına gerek yoktur. Evrim stratejisi sadece Monte Carlo yöntemine benzer olan sayısal eniyileme problemlerinde kullanılır. Genetik programlamada ise her çözüm, ağaç kodlamayla ifade edilmektedir.

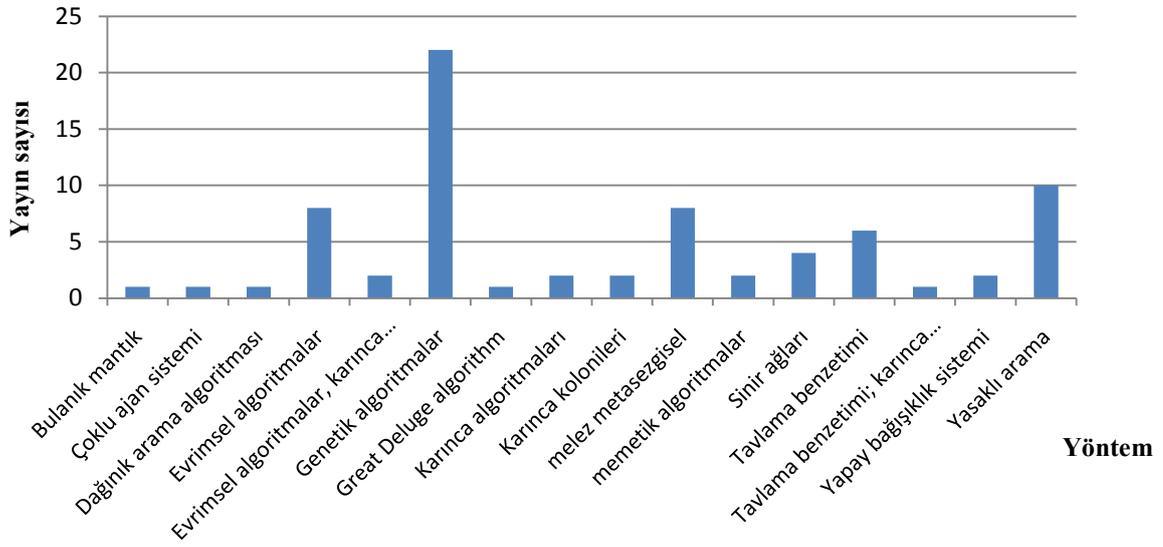
Geleneksel arama yöntemleri, probleme bir çözüm adayı önerir ve onu

değiştirerek daha iyi çözümler elde etmeye çalışır. Tavlama benzetimi ve yasaklı arama gibi metasezgiseller de tek bir çözümün iyileştirilmesi temeline dayanır. Evrimsel algoritmalar ise bir çözüm adayları nesili oluşturur ve bu nesil zamanla evrimleşir. Birden çok çözümü eş zamanlı olarak değerlendirebilmeleri nedeniyle, son yıllarda çok amaçlı en iyileme problemlerinin çözümünde evrimsel algoritmalara olan ilgi artmıştır (Coello ve Lamont, 2005). Corne vd. (1994), Paquete ve Fonseca (2001), Gani vd. (2004), Santiago-Mozos (2005) ve Beligiannisa vd. (2008)' nın çalışmaları, evrimsel algoritmaların eğitimsel zaman çizelgeleme problemlerinde kullanımına örnek olarak verilebilir.

Rossi-Doria ve Sampels (2002), derslerin derslik ve zaman dilimine atandığı ders zaman çizelgeleme probleminde evrimsel algoritma, karınca kolonileri, tavlama benzetimi, yasaklı arama ve ardıştırmalı yerel arama (*Iterated Local Search*) gibi beş farklı metasezgiselin kullanımını karşılaştırmıştır.

Eğitimsel zaman çizelgeleme problemlerinin çözümünde en çok kullanılan evrimsel algoritma, genetik algoritmalar olarak karşımıza çıkmaktadır. Yanısıra, diğer metasezgiseller arasında da en çok genetik algoritmaların kullanıldığı belirlenmiştir (Şekil 2.6).

Burke vd. (1994), bir sınav çizelgesi oluşturdukları GA'da başlangıç çözümü elde etmek için bir ağ renklendirme algoritması kullanmışlardır. Yine Burke vd. (1996b), bir yerel arama prosedürü ile evrimsel algoritmayı bir arada kullanarak sınav çizelgeleme probleminin çözümünü araştırmıştır. Ders çizelgeleme probleminin çözümü için Abramson ve Abela'nın (1992) geliştirdiği paralel GA ile de problemin çözüm süresinde dikkate değer bir düşüş olduğu belirlenmiştir. GA'ya ilişkin ayrıntılı açıklamalar ve eğitimsel zaman çizelgeleme problemleri için geliştirilmiş farklı kromozom yapılarına sahip uygulamalar Bölüm 3.2'de verilmiştir.



**Şekil 2.6** Kullanılan metasezgisel yöntemler

Eğitimsel zaman çizelgeleme problemlerinin çözümünde farklı evrimsel algoritmaların kullanıldığı çalışmalar Çizelge 2.8’de özetlenmiştir.

**Çizelge 2.8** Evrimsel algoritmaların kullanıldığı çalışmalar

Yıl	Yazar	Gözönüne alınan bileşenler						Yayında tanımlanan başlık
		Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
1990	Colomi, A., Dorigo, M. and Maniezzo, V.	+		+			+	School
1992	Abramson, D. and Abela, J.	+	+	+			+	School
1992	Colomi, A., and Morimo, D.	+					+	School
1994	Corne, D., Ross, P., and Fang, H.							survey
1994	Burke, E., Elliman, D. and Weare, R.		+			+	+	University
1994	Ross, P., Corne, D., and Fang, H.	+	+				+	Lecture (university)
1995	Erben, W. and Keppler, J.	+	+	+			+	Course (University)

Çizelge 2.8. devam

		Gözönüne alınan bileşenler						
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayında tanımlanan başlık
1996	Monfroglio, A.	+	+				+	School
1997	Drexl, A. and Salewski, F.	+	+				+	School
1998	Awad, R.M. and Chinneck, J.W			+		+		Proctor
1999	Marin, H.T., Ross, P., and Rendon, M.V.					+	+	Exam
1999	Deris, S., Omatu, S., Ohta, H., and Saad, P.	+	+				+	Course (university)
2001	Erben, W.					+	+	Exam
2001	Paquete, L.F., and Fonseca, C.M.					+	+	Exam (University)
2001	Filho, G.R., and Antonio, L.	+		+			+	School
2001	di Stefano, C. and Tettamanzi, A.G.B.	+					+	School
2002	Rossi-Doria, O. and Sampels, M.	+	+				+	Course (university)
2002	Blum, C., Correia, S., Dorigo, M., Paechter, B., Rossi-Doria, O. and Snoek, M.	+	+				+	University
2002	Voráč, J., Vondrák, I. and Vlček, K.	+	+				+	School
2002	Yu, E. and Sung, K.S.	+	+				+	Course (university)
2003	Yanez, J. and Ramirez, J.					+	+	Exam (University)
2003	Rossi-Doria, O., Sampels, M., Birattari, M., Chiarandini, M., Dorigo, M., Gambardella, L.M., Knowles, J., Manfrin, M., Mastrolilli, Paetcher, B., Paquete, L. and Stützle, T.	+	+				+	Course (university)
2004	Gani, T.A., Khader, A.T. and Budiarto, R.					+	+	Exam
2004	Erben, W. and Song, P.Y.					+	+	Exam
2004	Landa Silva, J.D., Burke, E.K. and Petrovic, S.							survey
2004	Kanoh, H. and Sakamoto, Y.	+					+	Course (university)
2005	Santiago-Mozos, R., Salcedo-Sanz, S., DePrado-Cumplido, M. and Bousoño-Calzón, C.	+			+			Course (university)
2005	Kazarlis, S.	+	+	+			+	Course (university)
2007	Beligiannis, G.N., Moschopoulos, C. and Likothanassis, S.D.	+		+			+	School
2007	Dilip Datta and Kalyanmoy Deb and Carlos M. Fonseca	+					+	School

Çizelge 2.8. devam

Yıl	Yazar	Gözönüne alınan bileşenler						Yayında tanımlanan başlık
		Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
2008	Beligiannisa, G.N. Moschopouloa, C.N., Kaperonisa, G.P., and Likothanassisa, S.D.	+		+			+	School
2009	Cheong, C.Y., Tan, K.C., and Veeravalli, B.					+	+	Exam (University)

### Karınca Kolonileri

Doğal karıncaların yuvaları ile besin kaynakları arasında izledikleri yolların gözlenmesine dayanan karınca kolonileri meta sezgiseli, son yıllarda eğitimsel zaman çizelgeleme problemlerinin çözümünde kullanılan bir başka çözüm yaklaşımıdır. Karıncalar hareket ederken geçtikleri yollara *feromen* adı verilen bir kimyasal madde bırakırlar. Böylece, hem diğer karıncaların da aynı yiyecek kaynağını bulmalarına yardımcı olurlar hem de koloninin yiyecek toplama süresi en iyilenmiş olur.

Dowland ve Tompson (2005), Azimi (2005) ve Eley (2007) sınav zaman çizelgeleme, Socha vd. (2002) de ders zaman çizelgeleme problemlerinde karınca kolonileri algoritmalarını kullanan çalışmalardır. Dowland ve Tompson (2005), ağ renklendirme sezgisellerini temel alarak geliştirdikleri karınca kolonileri algoritması ile sınav çizelgeleme problemini de çözmüşlerdir. Azimi (2005), yasaklı arama ve karınca kolonileri algoritmalarından oluşan melez bir algoritma ile problemlerin çözümünü araştırmıştır. Elde edilen sonuçlardan, melez algoritmanın bireysel algoritmalara göre daha başarılı olduğu gözlenmiştir. Eley (2007) ise, öğrencilerin çakışmalarını en küçükmeye çalışan bir sınav çizelgeleme probleminin çözümünde farklı stratejileri test etmiş ve karınca kolonileri yaklaşımlarının büyük ölçekli gerçek problemlerin çözümünde yüksek başarı elde ettiğini göstermiştir.

### Diğer metasezgisel yaklaşımlar

Kovacic (1993), Smith vd. (2003), Carrasco ve Pato (2004) ve Corr vd. (2006), *sinir ağları*'nı ders zaman çizelgeleme probleminde kullanan çalışmalardır.

Yapay sinir ağları ve genetik algoritmalarda olduğu gibi, doğadan taklit edilerek ortaya çıkarılmış genel amaçlı bir sezgisel yöntem olan, *yapay bağışıklık sistemi* (Karaboğa, 2004), ders-derslik-zaman ve sınav-derslik-zaman çizelgelerinin oluşturulmasında Malim vd. (2006) tarafından kullanılmıştır.

Petrovic ve Burke (2004)'un farklı eğitimsel zaman çizelgeleme problemleri için geliştirilen çözüm yaklaşımlarını verdikleri çalışmada, sınav zaman çizelgeleme problemlerinin çözümü için başlangıç parametrelerine fazla bağımlı olmayan ve uzman olmayan kişiler tarafından da kolaylıkla kullanılabilen *“great deluge”* algoritması da tanıtılmıştır. Tavlama benzetiminin bir türü olan bu algoritma, başlangıç çözüm kalitesinde belirlenmiş bir seviyeden daha düşük çözümler elde edildiği sürece daha kötü hareketleri de belirli bir olasılığa bağlı olarak kabul etmektedir. Bu algoritmada kötü sonuçların kabul olma olasılığı her adımda azalmaktadır. Qu vd. (2009) bu algoritma ile 13 test probleminin 8'i için en iyi çözümü elde etmiştir. Petrovic ve Bykov (2003) dokuz amaçlı sınav zaman çizelgeleme probleminin çözümünde bu algoritmayı kullanırken, Burke vd. (2004b) de sınav-derslik-zaman çizelgeleme probleminin çözümü için great deluge algoritmasını yasaklı arama algoritmasına bir seçenek olarak sunmuştur.

Oldukça yeni bir yaklaşım olan *Çoklu-Ajan Sistemi*'ni ders-derslik-zaman çizelgelemede kullanan ilk çalışma Strnad ve Guid (2007) tarafından yapılmıştır.

Bazı çalışmalarda ise problemlerin çözümünde, farklı metasezgisellerin meta sezgisel ve/veya diğer çözüm yaklaşımları ile birleştirilmesiyle geliştirilen, melez algoritmalar kullanılmıştır. Literatürde yer alan çalışmalar incelendiğinde, özellikle GA ile yerel arama yöntemlerinin birleştirilmesiyle oluşan bu tür melez algoritmaların eğitimsel zaman çizelgeleme problemlerinin çözümünde başarılı olduğu görülmüştür.

Hem evrimsel süreçleri hem de yerel arama işlemcilerini birlikte kullanan melez algoritmalar *memetik algoritmalar* olarak adlandırılmaktadır. Memetik algoritmalarda elde edilen her çözüm adayı, kendi komşuluğundaki en iyi yerel çözümdür.

Burke vd. (1996b) sınavları derslik ve zaman dilimlerine atadıkları çalışmalarında, çözüm için tek başına kullanıldığında önemli iyileşmeler sağlamayan mutasyon işlemcisi ile bir tepe tırmanma algoritmasını birleştirerek bir memetik algoritma geliştirmişlerdir. Bu memetik algoritma ile çözüm için uzun süre gerekmesine rağmen elde edilen zaman çizelgesinin kalitesi artmıştır. Burke ve Newall (1999)'ın önerdiği memetik algoritma ise, mutasyon ve yerel aramanın bir birleşimi ile sınavları zaman dilimlerine atamaktadır.

Gani vd. (2004) evrimsel algoritmalar ile bir yerel arama algoritmasını bir arada kullandıkları melez bir algoritma geliştirmiştir. Bir günde, birbirine yakın zaman dilimlerinde iki sınava giren toplam öğrenci sayısının en küçüklenmeye çalışıldığı bu çalışmada, sınavların zaman dilimlerine ataması yapılmış ve birçok veri seti için en iyi çözüm elde edilmiştir. Yanı sıra, sınavların dersliklere atanması bu çalışmada ele alınmamıştır. Eğitimsel zaman çizelgeleme problemlerinde memetik algoritmaları kullanan diğer çalışmalara örnek olarak Alkan ve Özcan (2003) ile Burke ve Silva (2004) verilebilir.

Chiarandini vd. (2006) tarafından geliştirilen melez meta sezgisel ile ders zaman çizelgeleme probleminin çözümünde başarılı sonuçlar elde edilmiştir. Bu çalışmada yasaklı arama, tavlama benzetimi ve Değişken Komşuluk Azalması (VND) algoritmalarının birleştirilmesinden oluşan melez algoritmalar, ders çizelgeleme probleminin çözümünde kullanılmıştır. Tavlama benzetimi algoritmasının VND ile değiştirilmesi ile elde edilen melez algoritmanın oldukça başarılı olduğu görülmüştür. Gunawan vd. (2007) de tamsayı programlama, aç gözlü bir sezgisel ve tavlama benzetiminden oluşan melez meta sezgisel ile öğrencilerin ders ve zaman dilimi tercihlerini en üst düzeyde karşılamaya çalışmıştır.

### 2.3.5 Olay-temelli çıkarsama yaklaşımları

**Olay temelli çıkarsama** (case-based reasoning) benzer problemlerin çözümlerini kullanarak yeni problemleri çözmeyi amaçlayan bir yapay zeka olgusudur. Eğitimsel zaman çizelgeleme problemlerinin çözümünde göreceli olarak yeni olan bu yaklaşımı Burke vd. (2006a), hem ders hem de sınav zaman çizelgelerinin oluşturulmasında kullanılacak olan sezgisellerin seçiminde kullanmışlardır. Sınavların çizelgelenmesinde olay temelli çıkarsama yaklaşımını kullanan bir diğer çalışma da Petrovic vd. (2007a) tarafından yapılmıştır. Petrovic ve Qu (2002) ders-derslik-zaman çizelgelemede olay temelli çıkarsama yaklaşımlarını temel alan bir hiper-sezgisel geliştirmiştir. Burke vd. (2003c) ve Burke vd. (2006b) de yine bu alanda ders zaman çizelgelemesi problemini ele alan çalışmalardır.

### 2.3.6 Hiper-sezgisel yöntemler

Tavlama benzetimi, genetik algoritmalar ve yasaklı arama gibi meta sezgiseller, eğitimsel zaman çizelgeleme problemlerinin çözümünde başarılı sonuçlar elde etmelerine karşın, parametre bağımlı yöntemlerdir. Örneğin, yasaklı arama için yasak listesinin uzunluğu, genetik algoritmalar için kromozom uzunluğu birer parametredir. Ayrıca bu yaklaşımlar genellikle uygulamanın yapıldığı yere özgü geliştirilmiş olup yeni bir problem için güncellenmeleri de oldukça zordur. Burke ve diğerlerinin (2003a) belirttiği gibi, günümüzde genel görüş, önce belirli problemlere özgü çözüm algoritmalarının geliştirilmesi daha sonra eldeki problem için bu algoritmaları sistematik olarak inceleyerek, akıllıca uygun bir algoritmanın seçilmesi yönündedir. Örneğin, Ross vd. (1998) GA ile belirli bir problem için çözüm bulmak yerine, bu problem için uygun algoritmanın GA ile bulunmasının daha etkili olacağını belirtmiştir. Bu kavram, hiper-sezgisel (*hyper-heuristic*) yaklaşım olarak isimlendirilmektedir.

Hiper-sezgisel yaklaşım Burke ve diğerleri tarafından (2003a) “sezgiselleri seçen sezgisel” olarak açıklanmıştır. Amacı, en iyileme yaklaşımlarının genellik seviyesini yükseltmek olan hiper sezgisellere olan ilgi gün geçtikçe artmaktadır.

Hiper-sezgisel, bir yüksek seviye (hiper) sezgisel ile düşük seviyeli sezgiseller kümesinden oluşur. Düşük seviyeli sezgiseller genellikle uygulamaya koymak için tam, kesin ve hızlıdır. Problem üzerinde doğrudan uygulanabilirler ve probleme özgüdürler. Diğer taraftan yüksek seviyeli sezgisel son derece kuramsaldır ve problem hakkında çok az bilgi içerir. Yüksek seviyeli sezgisel herhangi bir anda hangi düşük seviyeli sezgiselin kullanılacağına karar verir. Aldığı tek geri bildirim, düşük seviyeli sezgiselin uygulama süresi ve eğer gerçekleştiyse problemin amaç fonksiyonu değerindeki değişiklidir. Düşük seviyeli sezgiseller bir kere tasarlandıktan sonra birçok yüksek seviyeli sezgisel, aynı düşük seviyeli sezgiselleri kullanarak test edilebilirler.

Bir hiper-sezgisel, aday çözümlerden oluşan arama uzayı yerine sezgisellerden oluşan arama uzayında işlem görür. Burke ve diğerleri (2003b), geliştirdikleri yasaklı arama temelli hiper-sezgiseli ders zaman çizelgeleme ve hemşire nöbet çizelgeleme problemlerine uygulamıştır. Bu hiper sezgiselde tüm düşük seviyeli sezgiseller içinde en yüksek dereceye sahip olan bir sonraki ardıştırmada uygulanır. Eğer bu düşük seviyeli sezgisel çözümde bir iyileşme sağlamazsa yasak listesine alınır. Böylece yasak listesi, çözümde iyileşme sağlamayan sezgisellerin tekrar seçilmesini önlemiş olur. Burke ve diğerleri tarafından (2005b) geliştirilen bir diğer yasaklı arama temelli hiper-sezgiselde, sınav zaman çizelgeleme problemlerini çözmek için, düşük seviyeli sezgisel olarak farklı ağ renklendirme sezgiselleri permütasyonları kullanılmıştır. Literatürde yeralan hiper-sezgisel temelli çalışmalar Çizelge 2.9'da verilmiştir.

Çizelge 2.9 Hiper-sezgisellerin kullanıldığı çalışmalar

Yıl	Yazar	Gözönüne alınan bileşenler						Yayımlanan TT Türü
		Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
2002	Petrovic, S. and Qu, R.	+	+				+	Course
2002	Ross, P. and Hart, E.		+			+	+	Exam
2003	Rossi-Doria, O. and Paechter, B.	+	+				+	course
2003	Burke, E.K., Kendall, G. and Soubeiga, E.	+	+				+	Course (University)
2003	Burke, E.K., MacCarthy, B.L., Petrovic, S., and Qu, R.	+	+				+	Course (university)
2004	Burke, E.K. and Newall, J.P.					+	+	Exam
2005a	Burke, E.K., Landa Silva, J.D. and Soubeiga, E.	+	+				+	Course (university)
2005b	E. Burke M.Dror S.Petrovic R.Qu					+	+	Exam
2006	Bilgin, B., Özcan, E.,and Korkmaz, E.E.					+	+	Exam (University)
2007	E.K. Burke B.McCollum A.Meisels S. Petrovic R.Qu	+	+				+	Course

### 2.3.7 Kısıt temelli teknikler

Kısıt temelli teknikler (*constraint-based techniques*), belirli sayıda kısıtı sağlayacak şekilde olayların kaynaklara atanması şeklinde zaman çizelgeleme problemlerinin modellendiği çözüm teknikleridir. Mantıksal kısıt programlama (*constraint logic programming*) ve kısıt sağlama (*constraint satisfaction*) tekniklerinin kullanıldığı çalışmalarda zaman dilimleri sırayla faaliyetlere atanır. Tüm zaman dilimlerinin atanmasının ardından, uygun bir çizelge oluşturulana kadar geri dönüşlü bir süreç ile zaman dilimleri yeniden faaliyetlere atanır. Cheong vd. (2009)'nin de belirttiği üzere, bu teknikler ağ temelli sıralı teknikler gibi tek başlarına uygulandıklarında yüksek kalitede çözümler üretememektedir. Dolayısıyla, melez algoritmalarda önce bu teknikler kullanılarak bir başlangıç çözüm bulunur, ardından da arama yöntemleri ile uygun çözüm kalitesi artırılır.

Rudova ve Murray (2003) ve Goltz vd. (1998) mantıksal kısıt programlama; Abdennadher vd. (2000), Abbas ve Tsang (2004), Deris vd. (2000) ile Murray vd.

(2007) de kısıt sağlama tekniklerini kullanarak ders çizelgesi oluşturan çalışmalara örnek verilebilir.

### 2.3.8 Çok-amaçlı yaklaşımlar

Birçok gerçek hayat probleminde genellikle birden fazla amaç aynı anda eniyilenmek istenir. Çok amaçlı eniyileme problemlerinde amaç, kimi zaman birbiriyle çelişen ölçütler arasında bir uzlaşmayı temsil eden sonuçlar bulmaktır. Genel bir çok amaçlı eniyileme problemi, izleyen şekilde formüle edilir:

$x \in S$  kısıtları altında

$$\text{enk veya enb } F(x) = (f_1(x), f_2(x), \dots, f_k(x))$$

Zaman çizelgeleme problemlerinin çözümü sonucunda sağlanmayan esnek kısıtların sayısı, çözüm kalitesinin bir göstergesidir. Dolayısıyla, uygulanan çok amaçlı yaklaşımlarda, sağlanmayan esnek kısıtların her biri en küçüklenecek şekilde amaç fonksiyonları oluşturulabilir. Bu esnek kısıtlar mümkün olduğunca; kullanıcı tercihlerinin sağlanması, gün içinde girilen sınav sayısının ve süresinin belirlenen limitlerde olması, günün son zaman diliminde öğrencilere ders atanmaması, öğretici ve/veya öğrenci tercihlerinin karşılanması gibi farklı kurum ve/veya durumlara göre değişebilen kısıtlar olarak karşımıza çıkmaktadır. Genellikle tüm esnek kısıtları sağlayan bir uygun çözüm elde etmek mümkün değildir. Yanı sıra, toplam çakışmaların en küçüklenecek, toplam derslik kullanım oranının en büyüklenmesi ve/veya toplam günlük dersler arası boş zamanın en küçüklenecek gibi esnek kısıt haricinde problemin doğrudan birden fazla amacı olabilir.

Literatürde yer alan çalışmalar incelendiğinde, çok amaçlı olarak ele alınan problemlerin, genellikle birden fazla amacın tek bir skaler değere dönüştürülerek çözüldüğü görülmüştür. Son yıllarda geliştirilen metasezgisel algoritmalar ile problem, tek amaçlı bir yapıya dönüştürülmeden ele alınabilmektedir. Farklı veri kümeleri ve gerçek uygulama problemleri üzerinde denenen bu meta sezgisel algoritmalar ile

başarılı sonuçlar elde edilmiştir.

Çok amaçlı eğitimsel zaman çizelgeleme problemlerinde kullanılan genel stratejiler tek ve çok aşamalı yaklaşımlar olarak sınıflandırılabilir. Çok aşamalı yaklaşımlarda ilk aşamada uygun bir zaman çizelgesi bulunduktan sonra ikinci aşamada esnek kısıtların en iyilenmesi ele alınır. (Thompson and Dowsland; 1998; Dowsland, 1996; Dowsland, 1997; El Moudani et al., 2001).

Arani ve Lotfi (1989) sınav zaman çizelgeleme probleminin çözümü için üç aşamalı bir model geliştirmiştir. Lotfi ve Cervený (1991) ise bu modeli sınavların dersliklere de atanmasının eklenmesiyle her aşamada farklı esnek kısıtlardan sağlanmayanların sayısını en küçükleyen, dört aşamalı bir model olarak geliştirmiştir. Algoritmanın son aşamasında sınavlar dersliklere atanmaktadır. Birinci aşamada final sınavları, önceden tanımlanmış olan sınav bloklarına atanmaktadır. İlk aşamanın amacı herhangi bir zaman diliminde birden fazla sınavı olan öğrenci sayısını en küçükmektir. İkinci aşamada bir günde iki veya daha fazla sayıda sınavı olan öğrenci sayısını en küçükmek amacıyla sınav blokları günlere atanmaktadır. Üçüncü aşamada arka arkaya sınavı olan öğrenci sayısını en küçükmek için sınav günleri, sınav günlerinde sınav blokları olacak şekilde ayarlanmaktadır. Son aşamada da birden fazla derslikte yapılan sınav sayısının en küçükmek amacıyla sınavlar dersliklere atanmaktadır. Burada amaç, bir sınavın birden fazla derslikte yapılması durumunda kullanılan ekstra derslik sayısının en küçükmek olarak belirlenmiştir.

Özdemir (1998), ders zaman çizelgeleme probleminin çözümü için iki aşamadan oluşan çok amaçlı bir yaklaşım geliştirmiştir. Birinci aşamada ders-zaman dilimi ataması yapılmaktadır. İkinci aşamada ise öğreticilerin zaman dilimi ve derslik yönündeki tercihleri göz önüne alınarak öğretici ders ataması yapılmaktadır. Ayrıca Kara ve Ozdemir (1996, 1997) ile Özdemir ve Kara (1997) öğreticilerin tutum ve tercihlerini amaç olarak göz önüne alan başlıca çalışmaları inceleyerek uygulamalardaki sakıncaları belirtmiştir. Yanı sıra, öğretici tercihlerinin dikkate alındığı yeni bir modelleme yaklaşımı geliştirmiştir.

Burke vd. (2001), sınav zaman çizelgeleme problemlerinin çözümü için iki aşamadan oluşan çok amaçlı bir yaklaşım geliştirmiştir. Birinci aşamada her ölçüt için ayrı bir uygun çözüm elde edildikten sonra, ikinci aşamada tüm dokuz ölçüt (derslik kapasitesi, sınavların zamanı ve sırası vb.) aynı anda ele alınarak elde edilen çözüm kalitesi artırılmaya çalışılmaktadır.

Thompson ve Dowsland (1998) sınavların zaman dilimlerine atanması için, birinci aşamada uygun bir çözümün oluşturulduğu, ikinci aşamada da komşuluklara bakılarak uygun olmayan çözümlerin çözüm uzayından çıkarıldığı iki aşamalı bir tavlama benzetimi algoritması geliştirmiştir. Tüm zorunlu kısıtlar birinci aşamada sağlandıktan sonra, ikinci aşamada belirlenen esnek kısıtlara yönelik en iyi çözüm araştırılmaktadır.

Paquete ve Fonseca (2001) sınavların zaman dilimine atandığı sınav çizelgeleme problemi için çok amaçlı bir evrimsel algoritma uygulamıştır.

Literatürde zaman çizelgeleme başlığı altında yüzlerce çalışma vardır. Ancak, tez kapsamında eğitimsel, aynı zamanda çok amaçlı zaman çizelgeleme problemleri ele alındığı için, bu iki özelliği de içeren çalışma sayısı doğal olarak daha küçük bir kümeye karşılık gelmektedir. Petrovic ve Burke (2004) eğitimsel zaman çizelgeleme problemleri ele alındığında metasezgisellerin ve çok amaçlılık yaklaşımlarının birbirinden bağımsız konular olmadığını ve aralarında dikkate değer bir örtüşme olduğunu belirtmiştir. Zaten kombinatorik yapıya sahip olan bu problemlerin çok amaçlı olarak ele alınması, çözümü daha da zorlaştırdığından metasezgisellere olan ihtiyaç daha belirgin hale gelmiştir. Ek.2'de eğitimsel zaman çizelgeleme problemlerini çok amaçlı olarak ele alan çalışmalar, kullandıkları çözüm yaklaşımları ile birlikte verilmiştir.

## **2.4 Karşılaşılan Güçlükler ve Geline Nokta**

Çalışmanın bu bölümünde ayrıntılı olarak incelenen 196 eğitimsel zaman çizelgeleme çalışmasında görülmüştür ki, çalışmalarda problemlerin farklı bileşenleri

içermelerine rağmen aynı başlık altında ele alınmıştır. Dolayısıyla, eğitimsel zaman çizelgeleme terminolojisinde henüz bir standartlaşmanın olmadığı belirlenmiştir.

Eğitimsel zaman çizelgeleme problemleri, tüm eğitim kurumlarında, genellikle her dönem en az iki kez çözülmesi gereken, boyutundaki büyüklük nedeniyle çözüm süresi polinom zamanlı olmayan bir fonksiyona bağlı olarak değişen problemlerdir. Ders ve sınav programlarının hazırlanması bu sınıfa giren iki ana problem grubudur. Problemin çok farklı versiyonları örnek olarak verilebilir. Ders-derslik, ders-derslik-zaman dilimi, ders-zaman dilimi, ders-öğretici, ders-öğretici-zaman dilimi, sınav-zaman dilimi, sınav-derslik, sınav-derslik-zaman dilimi, sınav-gözetmen ataması olarak sayılan alt problemlerin her biri eğitimsel zaman çizelgeleme sınıfında yer almaktadır. Bir eğitim sisteminde genellikle yukarıda sayılan atamaların hepsini gerçekleştirmek gerekmektedir. Bazen sadece iki bileşenle (örn: ders-derslik veya ders-zaman dilimi ataması) ilgili atama yapıp, diğerleri ayrıca ve bu çözümle ilişkilendirerek ele alınırken, bazı durumlarda ikiden fazla (örn: ders-derslik-zaman dilimi) bileşene karşı gelen bir problem çözülmeye çalışılmaktadır. Hangi alt problem olursa olsun, özellikle karar değişkenlerinin tamsayı (çoğunlukla da 0-1) olması gerekliliği yanı sıra kısıtların ve problem boyutunun sistemden sisteme değişebilmesi, problemin çözümünü yanısıra genel bir çözüm yöntemi geliştirilebilmesini oldukça zorlaştırmaktadır.

Problemin yaygın etkisi oldukça fazladır. İlköğretimden yüksek öğrenime hemen her kurumda bu tür planlama problemleri vardır. Örnek olarak senede iki eğitim dönemi olan bir üniversitede, en azından bir ara sınav, bir de final sınavı dönemi bulunmaktadır. Bir dönem için olabilecek ikinci ara sınav dönemi, yanı sıra son yıllarda başlayan yaz okulu uygulamaları ve beraberinde gelen sınav çizelgeleri, dönem içerisindeki telafi ve mazeret vb. sınav çizelgeleri sayılmadan bile genellikle en az üç kez ilgili programlar hazırlanmaktadır.

Öte yandan son yıllarda, sistem içinde yer alan kişilerin (öğretici, öğrenci ve yönetim) tercih ve eğilimlerinin çizelgeleme sürecinde yer alması, üzerinde önemle durulan bir konu olmaya başlamıştır. İsteklerin dikkate alınması, doğrudan performansa yansıtacaktır (Kara and Özdemir, 1997). Fakat tercihlerin ne şekilde alınacağı ve nasıl

değerlendirileceği de ayrı bir problemdir. Problemin bu boyutunda da hem sayısal olmayan faktörler, hem de çok sayıda ve çoğu birbiriyle çelişen amaçlar ortaya çıkmaktadır. Gerçek hayat problemleri bu tür sayısal olmayan faktörlerle iç içedir. Kararları etkileyen ama göz önüne alınması zor olduğu için çoğu zaman sezgisel olarak problem sonuçlarına yansıtılan bu faktörlerin ele alınabileceği teknikler (örn: Analitik Hiyerarşi Süreci ve Analitik Serim Süreci) konusunda gelişmeler olmaktadır. Bu çalışma kapsamında bu gibi teknikler, çalışmaya esas olan konu için parametre tahmin yöntemleri olarak kullanılacaktır.

Son yıllarda yazılım ve bilgisayar donanımındaki hızlı gelişim, web tabanlı arayüzleri karar vericilere sunmakta, ayrıca çok büyük boyutlu problemlerin çözümü için de ümit vermektedir.

Bu bölümde incelenen çalışmalardan da görüldüğü üzere, eğitimsel zaman çizelgeleme problemleri literatürde oldukça yoğun bir ilgi görmektedir. Yukarıda belirtilen nedenlerle problem ya belirli bileşenleriyle çözülmüştür ya da ardışık yaklaşımlarla bir aşamanın sonuçlarını başka bir aşamada kullanan ve toplamda olabildiğince çok bileşeni çözmeye yönelik olan yöntemler denenmiştir. Bununla birlikte çoğu çalışma belirli bir sistemin problemini çözmeye dönüktür. Bu tespitlerin yanı sıra, tüm çalışmalar içerisinde gözetmen boyutunun ele alındığı çalışma sayısının çok az olduğu (196 çalışmada 4 çalışma) belirlenmiştir.

Bu tez çalışması ile hedeflenen, çok amaçlı eğitimsel zaman çizelgeleme problemlerini olabildiğince çok bileşeni ile ele almak ve çözümü için olabildiğince genel bir çözüm yaklaşımı geliştirmektir. Sistemin web ortamında kullanıcı arayüzleri ile bireysel tercihleri alacağı, arka planda matematiksel modelin oluşturulup önce eniyi çözümün, bu mümkün değilse uygun bir algoritma ile en iyiye yakın bir çözümün araştırılacağı bir yapı oluşturmak amaçlanmaktadır.

## BÖLÜM 3

### DERS DERSLİK ZAMAN ÇİZELGELEME PROBLEMİ İÇİN YENİ ÇÖZÜM YAKLAŞIMLARI

Bu bölümde, Bölüm 2.2.1’de tanımlanmış olan ders çizelgeleme problemi çok amaçlı olarak ele alınmış ve öncelikle matematiksel olarak modellenerek çözülmüştür. Test problemlerinin bir kısmında eniyi çözüme ulaşılmasına rağmen daha büyük boyutlu problemlerin çözüm aşamasında yaşanan güçlüklerin, problemin çözümünde sezgisel bir yöneme olan ihtiyacı ortaya çıkarması nedeniyle rassal anahtar temelli bir genetik algoritma (RKGA-*random key based genetic algorithm*) ile çok amaçlı bir genetik algoritma olan NSGA-II (*non-dominated sorting genetic algorithm*) temelli RKGA geliştirilmiştir.

#### 3.1 Ders Derslik Zaman Çizelgeleme Problemi ve Matematiksel Model ile Çözümü

Bölüm 2.3’te de belirtildiği üzere, eğitimsel zaman çizelgeleme problemlerinin çözümünü için çok çeşitli yaklaşımlar bulunmaktadır.

Bugüne kadar yapılan çalışmaların bir kısmında, zaman çizelgeleme problemi doğrusal ve tamsayı programlama gibi matematiksel programlama yöntemleri ile ele alınmıştır. Bu konu ile ilgili yapılmış çalışmalara Bölüm 2.3.1’de değinilmiştir. Çalışmanın bu boyutunda ele alınan ders-derslik-zaman çizelgeleme probleminde, literatürde yer alan problemlerin birçoğuna ek olarak, derslik kapasitesi ve derslerin ihtiyaç duyduğu nitelikleri içeren dersliklere atanması kısıtları da ele alınmakta, dolayısıyla problemin boyutu daha da artmaktadır.

Ders-derslik-zaman çizelgeleme probleminin çözümü için tasarlanacak yaklaşımın test edilmesi için literatürde yer alan çeşitli test problemleri incelenmiştir. Çalışmada

çözümü aranan problemler Ben Paechter tarafından tanımlanan ve Metasezgiseller Ağı'nda çalışılan problemlerdir. Karar vericilerin, derslerin yapılacağı dersliklere ilişkin özel nitelik gerekliliklerinin dikkate alındığı, ders programının haftalık olarak hazırlandığı ve her dersin haftalık ders saatinin 1 saat olduğu problemin kısıt ve amaçları izleyen şekilde belirlenmiştir.

Sıkı kısıtlar:

1. Her ders, projeksiyon aleti, Internet erişimi gibi bazı niteliklere gereksinim duymaktadır. Dersin atandığı derslikte gerekli nitelikler bulunmalıdır.
2. Bir öğrenci aynı zaman diliminde birden fazla dersi alamaz.
3. Bir derslikte aynı zaman diliminde birden fazla ders yapılamaz.
4. Bir dersin atandığı dersliğin kapasitesi, dersi alan öğrenci sayısı için yeterli olmalıdır.

Esnek kısıtlar:

1. Bir öğrencinin mümkün olduğunca günün son zaman diliminde dersi olmamalıdır.
2. Bir öğrenci, gün içinde mümkün olduğunca ardışık olarak en fazla iki derse girmelidir.
3. Bir öğrencinin, mümkün olduğunca bir günde sadece bir dersi olmamalıdır.

Tüm sıkı kısıtların sağlandığı bir zaman çizelgesi uygun çözüm olarak kabul edilir. Tüm esnek kısıtların sağlanması ise ya çok zordur ya da genellikle mümkün değildir. Çözüm sonucunda sağlanmayan esnek kısıtların sayısı, çözüm kalitesinin bir göstergesidir. Ele alınan problemin amacı, sağlanmayan toplam esnek kısıt sayısını enküçükmektir.

Çalışmada ele alınan problemler boyutlarına göre *small*, *medium* ve *large*<sup>3</sup> olmak üzere üç sınıfa ayrılmıştır. Small ve medium sınıfında da beş ayrı problem

---

<sup>3</sup> Literatürde bu konudaki test problemlerinde small, medium, large tipi problemlerin tanımları ve etiketleri konuyla özdeşleşmiştir. Bu sebeple kelimeler özel isim olarak kabul edilmiş, çalışmada Türkçeleştirilmemiştir.

yeralmaktadır. Problem sınıfları Çizelge 3.1’de verilen özelliklere sahiptir:

**Çizelge 3.1** Test problemlerinin özellikleri

	<b>Small</b>	<b>Medium</b>	<b>Large</b>
<b>Ders sayısı</b>	100	400	400
<b>Derslik sayısı</b>	5	10	10
<b>Nitelik sayısı</b>	5	5	10
<b>Öğrenci sayısı</b>	80	200	400
<b>Toplam zaman dilimi</b> (9 zaman dilimi * 5 gün)	45	45	45

Problemlere ait parametreler ise izleyen şekilde verilmiştir:

- Öğrenci Ders Matrisi (OD): öğrencinin aldığı dersleri gösteren matris
- Derslik Nitelik Matrisi (SN): derslikte bulunan nitelikleri gösteren matris
- Ders Nitelik Matrisi (DN): ders için gerekli olan nitelikleri belirten matris
- Derslik kapasiteleri

Çalışmanın bu bölümünde açıklayıcı bir örnek olarak Small01\* problemi ele alınmıştır. Probleme ait matrisler, açıklayıcı olması için ilgili çizelgelerde özetlenerek verilmiştir. Örneğin, Çizelge 3.2’de verilen öğrenci ders matrisinin birinci satırında yer alan birinci öğrenci 11., 88. ve 98. dersleri almaktadır.

---

\* <http://iridia.ulb.ac.be/supp/IridiaSupp2002-001/index.html>

Çizelge 3.2 Öğrenci ders matrisi

O/D	1	2	3	4	5	6	7	8	9	10	11	17	18	...	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1											1																	
2									1						1											1		
3									1		1				1											1		
4									1			1			1											1		
5												1			1	1										1		
6						1						1			1													
7						1						1			1									1				
8						1						1			1									1				
9						1						1			1									1				
10						1						1			1									1				
11						1						1			1									1				
12						1						1			1									1				
13						1						1			1									1				
14						1						1			1									1				
15						1						1	1		1	1								1	1			
16						1						1	1		1	1								1	1			
17						1						1	1		1	1								1	1			
18						1						1	1		1	1								1	1			
⋮																												
54			1	1				1							1													
55	1			1				1							1													
56				1				1							1													
57				1				1			1				1	1												
58				1				1			1				1	1										1		
59	1			1			1				1				1	1										1		
60				1			1				1				1	1										1		
79												1														1		
80																										1		

Çizelge 3.3'te verilen derslik nitelik matrisine göre, birinci derslik tüm beş niteliği de bulundururken, dördüncü derslikte sadece beşinci nitelik bulunmaktadır.

**Çizelge 3.3** Derslik nitelik matrisi

S/N	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1

Ders nitelik matrisine göre, birinci dersin yapılabilmesi için ikinci ve dördüncü niteliklere ihtiyaç vardır. Dolayısıyla, birinci ders bu iki niteliğin bulunduğu dersliğe atanmalıdır.

**Çizelge 3.4** Ders nitelik matrisi

D/N	1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1
6	1	1	1	1	1
7	1	1	1	1	1
8	1	1	1	1	1
9	1	1	1	1	1
10	1	1	1	1	1
11	1	1	1	1	1
12	1	1	1	1	1
13	1	1	1	1	1
...	1	1	1	1	1
96	1	1	1	1	1
97	1	1	1	1	1
98	1	1	1	1	1
99	1	1	1	1	1
100	1	1	1	1	1

Bu bölümde üç amaçlı ders-derslik-zaman çizelgeleme probleminin karma tamsayı matematiksel modeli, açıklayıcı örnek olarak seçilen Small01 probleminin parametreleri ile, tanımlanan kısıtlar ve amaçlar doğrultusunda kurulmuştur. İndis kümeleri, parametreler ve karar değişkenleri izleyen şekilde verilmiştir.

### *İndis kümeleri*

Ders indisleri kümesi:	$J = \{j \mid j=1, \dots, 100\}$
Öğrenci indisleri kümesi:	$I = \{i \mid i=1, \dots, 80\}$
Derslik indisleri kümesi:	$K = \{k \mid k = 1, \dots, 5\}$
Nitelik indisleri kümesi:	$L = \{l \mid l = 1, \dots, 5\}$
Zaman dilimleri kümesi:	$T = \{t \mid t=1, \dots, 45\}$
Gün indisleri kümesi:	$D = \{d \mid d=1, \dots, 5\}$
Günün son zaman dilimleri kümesi:	$T_{son} = \{t \mid t = 9, 18, 27, 36, 45\}$

### *Parametreler*

$k$ . dersliğin kapasitesi (derslik kapasiteleri sırasıyla 9, 2, 3, 11 ve 15'tir.):  $r_k$

$i$ . öğrenci  $j$ . dersi alıyor ise 1 diğer durumda 0 olan öğrenci-ders matrisi

elemanları:  $OD_{ij}$

$i$ . öğrencinin dersleri =  $\{j \in J \mid OD_{ij}=1\}$ :  $ds_i$

$k$ . derslikte  $l$ . nitelik bulunuyorsa 1 diğer durumda 0 olan derslik-nitelik matrisi

elemanları:  $SN_{kl}$

$j$ . ders  $l$ . niteliğe ihtiyaç duyuyorsa 1 diğer durumda 0 olan ders-nitelik matrisi

elemanları:  $DN_{jl}$

*Karar Değişkenleri*

$$y_{jk} = \begin{cases} 1, & j. \text{ ders } k. \text{ derslikte yapılıyorsa} \\ 0, & d. d. \end{cases}$$

$$z_{jt} = \begin{cases} 1, & j. \text{ ders } t. \text{ zaman diliminde yapılıyorsa} \\ 0, & d. d. \end{cases}$$

$$x_{jkt} = \begin{cases} 1, & j. \text{ ders } k. \text{ derslikte } t. \text{ zaman diliminde yapılıyorsa} \\ 0, & d. d. \end{cases}$$

$$rr_{it} = \begin{cases} 1, & i. \text{ öğrencinin } t. \text{ zaman diliminde atanmış dersi varsa} \\ 0, & d. d. \end{cases}$$

$$y1_{id} \in \{0,1\}$$

$$y2_{id} \in \{0,1\}$$

$$y3_{id} = \begin{cases} 1, & i. \text{ öğrencinin } d. \text{ günde sadece bir dersi varsa} \\ 0, & d. d. \end{cases}$$

$$ya2_{id} \in \{0,1\}$$

*d1*: tüm öğrencilerin tüm günlerde son zaman dilimlerine atanan ders sayısı

*d2*: tüm öğrencilerin tüm günlerde, gün içinde ardışık ikiden fazla dersi olma durumu sayısı

*d3*: tüm günlerde, öğrencilerin günde sadece bir dersi olma durumu sayısı

**Model<sup>4</sup>**

$$y_{jk} DN_{jl} \leq SN_{kl} \quad \forall(j, k, l) \quad (1)$$

$$\sum_t x_{jkt} \geq y_{jk} \quad \forall(j, k) \quad (2)$$

$$\sum_t x_{jkt} \leq M * y_{jk} \quad \forall(j, k) \quad (3)$$

$$\sum_k \sum_t x_{jkt} = 1 \quad \forall j \quad (4)$$

$$\sum_j x_{jkt} \leq 1 \quad \forall(k, t) \quad (5)$$

---

<sup>4</sup> Aksi belirtilmedikçe,  $t \in T$ 'dir.

$$\sum_i OD_{ij} y_{jk} \leq r_k \quad \forall (j, k) \quad (6)$$

$$\sum_j OD_{ij} z_{jt} \leq 1 \quad \forall (i, t) \quad (7)$$

$$\sum_k x_{jkt} \geq z_{jt} \quad \forall (j, t) \quad (8)$$

$$\sum_k x_{jkt} \leq M * z_{jt} \quad \forall (j, t) \quad (9)$$

$$rr_{it} = \sum_j OD_{ij} z_{jt} \quad \forall (i, t) \quad (10)$$

$$\sum_i \sum_{t \in T_{son}} rr_{it} = d1 \quad (11)$$

$$\sum_{(d-1)*9+1 \leq t \leq (d-1)*9+3} rr_{it} \geq 3 * ya2_{id} \quad \forall (i, d) \quad (12)$$

$$\sum_{(d-1)*9+2 \leq t \leq (d-1)*9+4} rr_{it} \geq 3 * ya2_{id} \quad \forall (i, d) \quad (13)$$

$$\sum_{(d-1)*9+3 \leq t \leq (d-1)*9+5} rr_{it} \geq 3 * ya2_{id} \quad \forall (i, d) \quad (14)$$

$$\sum_{(d-1)*9+4 \leq t \leq (d-1)*9+6} rr_{it} \geq 3 * ya2_{id} \quad \forall (i, d) \quad (15)$$

$$\sum_{(d-1)*9+5 \leq t \leq (d-1)*9+7} rr_{it} \geq 3 * ya2_{id} \quad \forall (i, d) \quad (16)$$

$$\sum_{(d-1)*9+6 \leq t \leq (d-1)*9+8} rr_{it} \geq 3 * ya2_{id} \quad \forall (i, d) \quad (17)$$

$$\sum_{(d-1)*9+7 \leq t \leq (d-1)*9+9} rr_{it} \geq 3 * ya2_{id} \quad \forall (i, d) \quad (18)$$

$$\sum_{(d-1)*9+1 \leq t \leq (d-1)*9+3} rr_{it} \leq 3 * ya2_{id} + 2 * (1 - ya2_{id}) \quad \forall (i, d) \quad (19)$$

$$\sum_{(d-1)*9+2 \leq t \leq (d-1)*9+4} rr_{it} \leq 3 * ya2_{id} + 2 * (1 - ya2_{id}) \quad \forall (i, d) \quad (20)$$

$$\sum_{(d-1)*9+3 \leq t \leq (d-1)*9+5} rr_{it} \leq 3 * ya2_{id} + 2 * (1 - ya2_{id}) \quad \forall (i, d) \quad (21)$$

$$\sum_{(d-1)*9+4 \leq t \leq (d-1)*9+6} rr_{it} \leq 3 * ya2_{id} + 2 * (1 - ya2_{id}) \quad \forall (i, d) \quad (22)$$

$$\sum_{(d-1)*9+5 \leq t \leq (d-1)*9+7} rr_{it} \leq 3 * ya2_{id} + 2 * (1 - ya2_{id}) \quad \forall (i, d) \quad (23)$$

$$\sum_{(d-1)*9+6 \leq t \leq (d-1)*9+8} rr_{it} \leq 3 * ya2_{id} + 2 * (1 - ya2_{id}) \quad \forall (i, d) \quad (24)$$

$$\sum_{(d-1)*9+7 \leq t \leq (d-1)*9+9} rr_{it} \leq 3 * ya2_{id} + 2 * (1 - ya2_{id}) \quad \forall (i, d) \quad (25)$$

$$\sum_i \sum_d ya2_{id} = d2 \quad (26)$$

$$\sum_{(d-1)*9 \leq t \leq d*9} rr_{it} \geq 2 * y2_{id} + y3_{id} \quad \forall (i, d) \quad (27)$$

$$\sum_{(d-1)*9 \leq t \leq d*9} rr_{it} \leq y3_{id} + 9 * y2_{id} \quad \forall (i, d) \quad (28)$$

$$y1_{id} + y2_{id} + y3_{id} = 1 \quad \forall (i, d) \quad (29)$$

$$\sum_i \sum_d y3_{id} = d3 \quad (30)$$

$$x_{jkt} \in \{0,1\} \quad \forall (j, k, t) \quad (31)$$

$$y_{jk} \in \{0,1\} \quad \forall (j, k) \quad (32)$$

$$z_{jt} \in \{0,1\} \quad \forall (j, t) \quad (33)$$

$$rr_{it} \in \{0,1\} \quad \forall (i, t) \quad (34)$$

$$y1_{id} \in \{0,1\} \quad \forall (i, d) \quad (35)$$

$$y2_{id} \in \{0,1\} \quad \forall (i, d) \quad (36)$$

$$y3_{id} \in \{0,1\} \quad \forall (i, d) \quad (37)$$

$$ya2_{id} \in \{0,1\} \quad \forall (i, d) \quad (38)$$

$$d1, d2, d3 \geq 0 \quad (39)$$

*kısıtları altında*

$$enk z = d1 + d2 + d3 \quad (40)$$

Kısıt kümesi (1) ile, derslerin ihtiyaç duydukları nitelikleri içeren dersliklere atanmaları sağlanmaktadır. Herhangi ders-derslik  $(j,k)$  ikilisi bir zaman dilimine atanırsa, kısıt kümesi (3)  $y_{jk}$ 'nin "1" değerini almasını zorlar.  $(j,k)$  ikilisinin bir zaman dilimine atanmaması durumunda ise (2) ve (3) kısıt kümelerindeki eşitsizliklerin sol tarafları sıfır değerini alır. Böylece kısıt kümesi (2),  $y_{jk}$ 'yi "0" değerini almaya zorlamaktadır.

Bir dersin mutlaka bir zaman diliminde bir dersliğe atanması ve herhangi bir dersliğe herhangi bir zaman diliminde en fazla bir dersin atanması, sırasıyla (4). ve (5). kısıt kümeleri ile sağlanmaktadır. Kısıt kümesi (6) ile bir dersin atandığı dersliğin kapasitesinin, dersi alan öğrenci sayısı için yeterli olması, (7) ile bir öğrencinin aynı zaman diliminde birden fazla ders almaması sağlanmaktadır.

Herhangi ders-zaman dilimi  $(j,t)$  ikilisi bir dersliğe atanırsa, kısıt kümesi (9)  $z_{jt}$ 'nin "1" değerini almasını zorlar.  $(j,t)$  ikilisinin bir dersliğe atanmaması durumunda ise (8) ve (9) kısıt kümelerindeki eşitsizliklerin sol tarafları sıfır değerini alır. Böylece kısıt kümesi (9),  $z_{jt}$ 'nin "0" değerini almasını zorlamaktadır.

Kısıt kümesi (10) ile, bir öğrencinin bir  $t$  zaman diliminde dersinin olup olmadığı bilgisi tutulmaktadır. Bu bilgi kullanılarak, kısıt kümesi (11) ile de tüm öğrencilerin tüm

günlerdeki son zaman dilimlerinde derse atanma sayısı ( $d1$ ) hesaplanmaktadır.

Bir öğrencinin gün içinde ikiden fazla ardışık dersi olması durumunu kontrol etmek için bir gündeki tüm ardışık üçlü zaman dilimleri dikkate alınmıştır. Belirli bir öğrencinin bir günde ardışık ikiden fazla dersinin olup olmadığı  $ya2_{id}$  değişkeni ile kontrol edilmektedir. Bir günde sadece bir defa ardışık üç ders ile karşılaşıldığında ilgili öğrenci için  $ya2_{id}$  değişkeni 1 değerini almaktadır. Tüm gün ve öğrenciler üzerinden bu değişkenin toplamı, tüm günlerde öğrencilerin ikiden fazla ardışık dersi olma durumu sayısını gösterir ve bu toplam ( $d2$ ), amaç fonksiyonuna yazılarak en küçüklenmeye çalışılır. Gün içinde 9 zaman dilimi olduğundan 7 farklı ardışıklık durumunun kontrol edilmesi gerekmektedir.

Kısıt kümeleri (12)-(25) ardışıklık kısıtları olarak modele eklenmiştir. Örneğin, birinci öğrencinin pazartesi günü ilk üç saatte ardışık dersinin olması durumunda  $ya2_{11}$  değişkeni 1 değerini alır, öğrencinin bu üç zaman diliminde aldığı ders sayısı 2 veya daha küçük ise değişken sıfır değeri alır. Bu durum, kısıt (12) ve kısıt (19) ile gerçekleştirilir. Birinci öğrenci için  $i$  indisi 1, pazartesi günü için de  $d$  indisi 1 değerini alarak ilk üç zaman dilimi için kısıt (12)

$$rr_{11} + rr_{12} + rr_{13} \geq 3ya2_{11}$$

olarak, kısıt (19) ise

$$rr_{11} + rr_{12} + rr_{13} \leq 3ya2_{11} + 2(1 - ya2_{11})$$

olarak elde edilir. Eğer bu öğrencinin pazartesi gününün ilk ardışık üç zaman diliminde üç dersi varsa  $rr_{11} + rr_{12} + rr_{13}$  toplamı 3 olacaktır. Bu durumda (12) ve (19) nolu kısıtların sağlanması için  $ya2_{11}$  değişkeninin 1 değerini alması zorunludur. Eğer toplam, 2 veya daha küçük ise değişken zorunlu olarak sıfır değerini alacaktır. Kısıt kümesi (26) ile de tüm öğrencilerin tüm günlerde, gün içinde ardışık ikiden fazla dersi olma durumu sayısı ( $d2$ ) hesaplanmaktadır.

Bir öğrencinin bir gün içinde hiç dersi olmayabilir. Bu durum,  $y1_{id}$  değişkeninin 1 değerini alması ve dolayısıyla  $y2_{id}$  ve  $y3_{id}$  değişkenlerinin 0 değerini alması ile sağlanır. Bir öğrencinin  $d$ . günde sadece bir dersinin olup olmadığı  $y3_{id}$  değişkeni ile

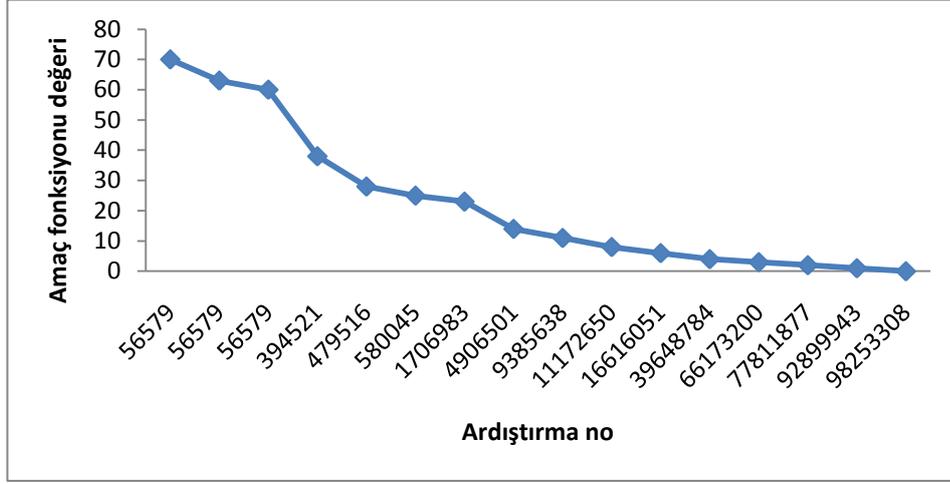
kontrol edilmektedir.  $d$ . günde sadece bir dersi olan öğrenci için  $y_{3id}$  değişkeni 1,  $y_{2id}$  ve  $y_{1id}$  değişkenleri de 0 değerini alacaktır. Bu her iki durumda, (27), (28) ve (29) nolu kısıt kümeleri ile sağlanmaktadır. Tüm gün ve öğrenciler üzerinden  $y_{3id}$  değişkeninin toplamı, tüm günlerde öğrencilerin günde sadece bir dersi olma durumu sayısını ( $d3$ ) gösterir. Bu toplam (30) nolu kısıt kümesi ile hesaplanmakta ve amaç fonksiyonuna yazılarak en küçüklenmeye çalışılmaktadır.

Çalışmada ele alınan ders-derslik-zaman çizelgeleme probleminin amacı, uygun bir çözümde sağlanmayan esnek kısıt sayısını en küçüklemektir. Sağlanmayan esnek kısıtlar izleyen şekilde tanımlanmıştır:

1. Tüm öğrencilerin tüm günlerde son zaman dilimlerine atanan ders sayısı:  $d1$
2. Tüm öğrencilerin tüm günlerde, gün içinde ardışık ikiden fazla dersi olma durumu sayısı:  $d2$
3. Tüm günlerde, öğrencilerin günde sadece bir dersi olma durumu sayısı:  $d3$

(40) nolu amaç fonksiyonu ile toplam sağlanmayan esnek kısıt sayısı en küçüklenmektedir.

Deris vd. (1999) ile Kanoh ve Sakamoto (2004)'nin da belirttiği gibi, eğitimsel zaman çizelgeleme problemlerinin geleneksel eniyileme teknikleri ile çözümü oldukça zordur. Ele alınan problem için geliştirilen modelin, seçilen test problemi Small01 için tüm esnek kısıtların sağlandığı en iyi çözümü GAMS yazılımında 2 GHz işlemcili, 2 GB ram'li bir bilgisayarda 101 saat çalışmanın ardından 98.253.308 inci ardıştırmada elde edilmiştir (Şekil 3.1). Elde edilen çözümde hiçbir öğrencinin günün son zaman dilimine dersi atanmamış, tüm öğrencilerin tüm günlerde gün içinde en fazla iki ardışık dersi atanmış ve tüm öğrencilerin gün içinde ya hiç dersi atanmamış ya da en az iki dersi atanmıştır. Geliştirilen modelin çözüm çıktıları EK 1'de verilmiştir.



**Şekil 3.1** Small01 probleminin her ardıştırmadaki amaç fonksiyonu değerleri

2.13 GHz, 8 çekirdekli ve 6 GB Ram’li MacPro bilgisayarda ise diğer dört small tipi test problemleri için eniyi çözümler ortalama 3,5 saatte elde edilmiştir Small tipi test problemleri için literatürde elde edilmiş çözümler ile çalışmada elde edilen çözümler (toplam sağlanmayan esnek kısıt sayısı) Çizelge 3.5 ile verilmiştir.

100 dersin çizelgelendiği *small* tipi problemlerin çözümünün ardından 400 dersin çizelgelendiği *medium* tipi problemlerin çözümüne geçilmiştir. Ancak, 96 saat çalışmanın ardından GAMS programı uygun bir çözüm bulamamıştır. Schaerf (1999) de 1963-1999 yılları arasındaki çalışmaları derlediği yayınında, ders-derslik-zaman çizelgeleme problemin NP-tam olduğunu ve en iyi çözümün sadece küçük boyutlu problemler (örneğin en fazla 10 ders için) için elde edilebileceğini göstermiştir.

**Çizelge 3.5** Small tipi problemler için elde edilen amaç fonksiyonu değerleri

	Kullanılan algoritmalar			Geliştirilen matematiksel model ile elde edilen sonuçlar
	HH (Burke et al., 2003b)	RRLS (Socha et al., 2002)	ANT (Socha et al., 2002)	
Small01	1	8	1	<b>0</b>
Small02	2	11	3	<b>0</b>
Small03	<b>0</b>	8	1	<b>0</b>
Small04	1	7	1	<b>0</b>
Small05	<b>0</b>	5	<b>0</b>	<b>0</b>

### Boyut Analizi

Ders-derslik-zaman çizelgeleme problemi için kurulan matematiksel modelin karmaşıklığını göstermek için, modelin boyut analizi Çizelge 3.6’da verilmiştir.

Açıklayıcı örnek Small01 için ( $j=100$ ,  $i=80$ ,  $k=5$ ,  $l=5$ ,  $t=45$ ,  $d=5$ ) bu modelin toplam kısıt sayısı 27322, toplam değişken sayısı da 32703 olarak belirlenmiştir. Medium tipi problemler için toplam kısıt sayısı 67852, toplam değişken sayısı da 215003 olmaktadır. Görüldüğü üzere, test problemleri içinde 100 dersin çizelgelendiği en küçük örnek için bile model boyutu oldukça büyüktür. Değişken ve kısıt sayıları arttıkça, boyutun artacağı görülmektedir.

**Çizelge 3.6** Ders-derslik-zaman çizelgeleme problemi için geliştirilen matematiksel model için boyut analizi

Kısıt no	İndisler	Toplam kısıt sayısı
1	$j,k,l$	$j \times k \times l$
2	$j,k$	$j \times k$
3	$j,k$	$j \times k$
4	$j$	$j$
5	$k,t$	$k \times t$
6	$j,k$	$j \times k$
7	$i, t$	$i \times t$
8	$j, t$	$j \times t$
9	$j, t$	$j \times t$
10	$i, t$	$i \times t$
11	-	1
12-25	$i, d$	$i \times d \times 14$
26	-	1
27-29	$i, d$	$i \times d \times 13$
Değişken	İndisler	Toplam değişken sayısı
$x_{jkt}$	$j,k,t$	$j \times k \times t$
$y_{jk}$	$j,k$	$j \times k$
$z_{jt}$	$j,t$	$j \times t$
$rr_{it}$	$i, t$	$i \times t$
$y1_{id}$	$i,d$	$i \times d$
$y2_{id}$	$i,d$	$i \times d$
$y3_{id}$	$i,d$	$i \times d$
$ya2_{id}$	$i,d$	$i \times d$
$d1$	-	1
$d2$	-	1
$d3$	-	1
Toplam kısıt sayısı		$3jk + 2jt + jkl + j + kt + 2it + 17id + 2$
Toplam değişken sayısı		$jkt + jk + jt + it + 4id + 3$

### 3.2 Ders Derslik Zaman Çizelgeleme Problemi için Rassal Anahtar Temelli bir Genetik Algoritma (RKGA)

Ele alınan bir problemin çözümüne yönelik bir algoritma araştırılmadan önce, bu problemin sonlu sayıda aşamada çözümlenip çözülemeyeceğini bilmek gerekir. Çok fazla seçeneğin olduğu durumlarda, olası tüm durumların ayrıntılı bir şekilde aranması, mümkün bile olsa, uygulama açısından pratik değildir. Bir problemi bir tamsayılı

programlama problemi olarak modellemek de çok fazla deęişken ve kısıt olduęu için yine pratik deęildir (Abramson, 1991). Bölüm 3.1’de ele alınan test problemi için yapılan boyut analizi Abramson’u destekler niteliktedir. Bu ve benzeri problemleri çözmek için kullanılan ve yerel en iyilerden kaçınma yeteneęine sahip sezgisel yöntemlerin gereklilięi, Bölüm 2.3’te de belirtildięi üzere, bu noktada kendini göstermektedir.

Evrimsel algoritmaların ve evrimsel bir algoritma olan genetik algoritmaların (GA), kombinatorik en iyileme problemlerinin çözümündeki başarıları göz önünde bulundurularak, çalışmanın bu bölümünde, ders-derslik-zaman çizelgeleme problemlerinin çözümü için rassal anahtar temelli bir genetik algoritma (RKGA) geliştirilmiştir.

Geliştirilen GA tanıtılmadan önce, GA’lar hakkında kısa bir bilgi verilmiş ve literatürde eğitimsel zaman çizelgeleme problemlerinin çözümü için gelitirilmiş olan GA’lar incelenmiştir.

### **3.2.1 Genetik algoritmalar, eğitimsel zaman çizelgeleme ve genetik algoritma uygulamaları**

Genetik algoritmalar, seçim, üreme, mutasyon ve çaprazlama gibi doğal biyolojik evrim sürecini taklit eden evrimsel algoritmalar başlığı altında yer almaktadır. Bu konudaki ilk çalışma Bremermann ve dięerleri (1966) tarafından gerçekleştirilmiştir (Bagchi, 1999). Geçen son 20-30 yılda evrim prensibine dayalı algoritmalara ilgi oldukça artmış ve bu konularla ilgili araştırmacıların ve literatürdeki çalışmaların sayısı hızla çoęalmıştır. Sezgisel algoritmalarından farklı olarak, yerel en iyi noktalardan kaçma mekanizmaları olan ve bu sebeple metasezgisel olarak adlandırılan evrimsel algoritmalar topluluk temelli yaklaşımlardır. Evrim prensibine dayanan tekniklerin hepsini temsilen ortak bir terim olarak *evrimsel hesaplama* terimi yaygın olarak kullanılmaya başlanmıştır (Karaboęa, 2004). Yapay zekanın gittikçe genişleyen bir kolu olan evrimsel algoritmaların alt dalları olarak genetik algoritmalar, genetik

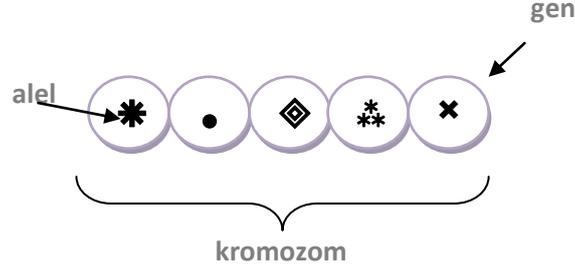
programlama, yapay sinir ağıları, tavlama benzetimi, yasaklı arama ve bunlarla birlikte bulanık mantık işletme, temel bilimler ve mühendislik problemlerinde tek başına veya karma sistemler olarak kullanılabilir (Back vd., 2000).

Holland tarafından 1975 yılında geliştirilen GA, Darwin'in *en iyi olan yaşar* prensibine dayalı olarak biyolojik sistemlerin gelişim sürecini modellemektedir. Holland araştırmalarını, arama ve en iyiyi bulmak için doğal seçme ve genetik evrim ilkesinden yola çıkarak yapmıştır. Süreç boyunca, biyolojik sistemde bireyin, bulunduğu çevreye uyum sağlayıp daha uygun hale gelmesi örnek alınmış, en iyiyi bulmak için bilgisayar yazılımları geliştirilmiştir. GA rassal arama tekniklerini kullanarak çözüm bulmaya çalışan, parametre kodlama esasına dayalı bir arama tekniğidir. GA, birçok kombinatoriyel en iyileme problemi üzerinde başarılı bir şekilde uygulanmıştır (Goldberg, 1989).

GA sahip olduğu, izleyen üç özellik nedeniyle, en iyileme problemlerinin çözümünde tercih edilen bir tekniktir (Gen and Cheng, 1997):

1. Özel matematiksel yapılar içermez.
2. Bütünsel en iyiyi araştırmada etkin bir yöntemdir.
3. Özel problemler için daha güçlü araştırma yapılarının geliştirilebileceği, melez yöntemlerin kullanımına uygun olan esnek bir yapıya sahiptir.

Herhangi bir probleme uygulanan GA'da, *gen* olarak adlandırılan, GA'nın en küçük birimi, en iyilenecek değişkene aittir. Biyolojideki benzerlik kullanılarak, her bir genin aldığı değer de "allel" olarak adlandırılır. Şekil 3.2'de de gösterildiği gibi, genler bir seri oluşturacak şekilde kodlanır ve *kromozomu* oluştururlar. Her komozom, ilgili problemin bir çözümüne karşı gelmektedir.



**Şekil 3.2** GA’da kromozom yapısı

Genetik algoritma, en iyinin korunumu ve doğal seçim ilkesini esas alan bir arama yöntemidir. Standart bir GA’da, aday sonuçlar eşit boyutlu vektörler olarak ifade edilir. Başlangıçta, bu vektörlerden bir grup, rassal olarak seçilerek belirli büyüklükte bir *topluluk* oluşturulur. Kromozom adı verilen bu vektörler, yeni *nesiller* (jenerasyon) oluşturarak değişikliklere uğrar. Her nesilde kromozomların iyiliği ölçülür, yani her vektör, amaç fonksiyonuna yerleştirilerek vermiş olduğu sonuç hesaplanır (Nabiyev, 2005).

GA, çözüm uzayının tamamı yerine belirli bir kısmını tarar. Dolayısıyla, etkin bir arama sonucunda çözüme çok daha kısa bir sürede ulaşır. Genetik arama nesildeki her bir bireyin amaç fonksiyonu (uyum değeri) değeri ile yönlendirilir. Problem türüne göre iyi uyum değerine sahip bireylere üremeleri için daha çok şans verilir.

Araştırma uzayında yüksek uyum değerine sahip çözüm noktalarına ulaşmada genetik işlemciler kullanılır. Yaygın olarak kullanılan genetik işlemciler seçim, yeniden üreme, çaprazlama ve mutasyon işlemcileridir. Farklı problemler türleri için bu işlemciler temel alınarak çeşitli yeni işlemciler de geliştirilmiştir (örn. Parçalı çaprazlama mantığını kullanan Kanoh ve Sakamoto (2004) vb.).

Standart bir GA yordamı izleyen adımlardan oluşmaktadır (İşlier, 1995):

0. Uyum fonksiyonunu kur, kromozomdaki bilgiyi kodla, parametreleri belirle.
1. İlk nesli üret.
2. Kromozomların uyum değerlerini hesapla.
3. Uyum ile orantılı seçilme şansı ver.
4. Bu şansa bağlı olarak bir çift birey seç (kopyalama).
5. Bunları rassal kırılma noktasına göre seç (çaprazlama).
6. Mutasyon sırası gelmediyse 8'e atla.
7. Rassal bir noktadan mutasyon uygula.
8. Yeterli sayıda nesil yetiştirildiyse DUR. Aksi halde 2'ye dön.

### Eğitimsel Zaman Çizelgeleme ve GA Uygulamaları

Eğitimsel zaman çizelgeleme çalışmaları ile ilgili literatür incelendiğinde, problemi metasezgisel algoritmalar ile çözen çalışmaların çoğunda matris ve vektör tabanlı kodlama kullanıldığı görülmüştür. İkili kodlamanın kullanıldığı çalışmalarda ise kromozom uzunluğu, “(ders sayısı) \* (atanan kaynak sayısı)” kadar olmaktadır (örn. Ross vd. 1994).

Genel bir eğitimsel zaman çizelgesi, sütunlar günleri, satırlar da zaman dilimlerini gösterecek şekilde düzenlenir. İlgili hücrelerde de ders bilgisi yer alır. Bunu genetik algorithmada kodlamak genetik işlemciler açısından zor olduğu için farklı dönüşümler uygulanmaktadır. Colorni vd. (1992) çalışmalarında kromozomu, her elemanı bir gen olan bir matris ile temsil etmiştir. Şekil 3.3'te verilen, her satırı öğreticisi belirli olan bir derse ve her sütunu da bir güne karşı gelen çizelgede, derslerin zaman dilimine ataması yapılmaktadır.

Teacher-Subject	Mon	Tue	Wed	Thu	Fri	Sat
Literature - 1	*11*1	112**	***11	2212*	11111	-----
Literature - 2	*6**6	7777*	***77	66**7	-----	7777*
Literature - 3	***2*	666*6	2**22	-----	6266*	6622*
Literature - 4	*8***	44***	-----	**4*8	84888	88444
Literature - 5	-----	*5555	**355	**353	3*33*	33***
Literature - 6	000*0	-----	0*999	0099*	9****	*9***
English	152*5	32411	53***	****5	43422	-----
German	77997	98800	607*6	-----	*6***	**08*
History and Philosophy - 1	5*33*	*3343	-----	55*44	***4*	4555*
History and Philosophy - 2	9**8*	-----	*88*0	990**	0*009	908*8
Math and Physics - 1	-----	5****	45434	4453*	5*55*	*4333
Math and Physics - 2	*9*09	09999	***09	88800	*9**0	-----
Math - 1	SSSS2	2S1AA	112**	***1*	-----	22*1*
Math - 2	6S66S	SS**A	*7S6*	-----	7777*	***6*
Natural sciences	33444	80022	-----	7378*	*8995	5*9**
Art	84518	-----	96643	37279	2****	01*05
Experimental Physics	22778	S**67	-----	1166*	**2SS	SS1**
Gymnastic - 1	SSS**	***34	34SSS	***SS	SS***	-----
Gymnastic - 2	SSS**	***89	890SS	***SS	S0***	-----
Religion	4S853	*****	721S*	SS***	-----	SS690

Şekil 3.3 Bir zaman çizelgesinin matris gösterimi (Colorni et al., 1992)

Yu ve Sung (2002) kromozomu, Şekil 3.4'te verilen bir matris şeklinde tanımlamıştır. Sütunların her biri bir zaman dilimini (gün-saat temelinde), satırlar da derslerin yapıldığı derslikleri temsil etmektedir. Önceliği olan derslerin önceden çizelgelendiği bu çalışmada, çaprazlama işlemcisi olarak PMX (kısmi uyumlu çaprazlama) yöntemi kullanılmıştır.

	Pzt.1	Pzt.2	...	Cuma 5	Cuma 6
derslik 1	<i>Ders 1-1</i>	<i>Ders 2-2</i>	...	...	...
...	...	...	...	...	...
derslik i	...	<i>Ders i-2</i>	...	<i>Ders i-5</i>	...
Lab 1	...	...	...	...	...
...	...	...	...	...	...
Lab j	...	...	...	...	<i>Ders j-6</i>

Şekil 3.4 Bir zaman çizelgesinin matris gösterimi (Yu and Sung, 2002)

Burke vd. (1994)'nin sınavları derslik ve zaman dilimlerine atadığı çalışmada, kromozomlar yine bir zaman çizelgesi yani bir matris olarak tasarlanmıştır. Bir gen, doğrudan hangi sınavın ne zaman ve nerede yapılacağı bilgisini vermektedir. Geliştirilen çaprazlama işlemcisinde, seçilen iki çizelgeden birinde erken zaman dilimde yer alan sınavlar diğer çizelgede geç zaman diliminde yer alan sınavlarla yer değiştirmektedir.

Filho ve Loreno (2001), “öğretim elemanı, ders” çiftlerini ikili sütunlar ile temsil etmiştir. Bu ikili sınıflar, her bir zaman dilimi için çakışmayan kümeleri oluşturmada kullanılmıştır. Bu çalışmadaki zaman çizelgeleme problemi, genetik algoritmalar kullanarak bir kümeleme problemi olarak ele alınmıştır. Geliştirilen GA, her sütun için zaman dilimi sayısı kadar çekirdek içeren şemalardan oluşan bir popülasyonda çalışmaktadır. Sütunlar, her bir zaman dilimine karşı gelecek şekilde, benzerlik katsayısı hesaplanarak kümelenebilir.

Vorac vd. (2002), okul zaman çizelgeleme problemi için Java programlama dilini kullanarak üç satırlı bir kromozom tanımlamıştır (Şekil 3.5). Birinci satırda dersler, dersi veren öğretim elemanı ve dersi alan grup numarası tanımlanmakta, ikinci satırda derslerin yapıldığı derslikler ve üçüncü satırda da zaman dilimleri tanımlanmaktadır. Mutasyon işlemcisi sonucunda, uygunluğu bozulan bireylere özel bir onarım işlemcisi çalışmaktadır. Ayrıca genel bir GA'da mutasyon oranı 0.01 ile 0.2 arasında olacak şekilde çok düşük tutulurken, bu çalışmada bu oran yüksek tutulmuştur.

Erben ve Kepler (1995) kromozomu 5\*5'lik bir matris olarak tasarlamışlardır. Ders, derslik, öğrenci ve zaman dilimi atamasının yapıldığı bu çalışmada problem, PROLOG dilinde yazılmış bir program ile çözdürülmüştür.

UDPJ, ben64, L101, size=18, type=uc, length=2, time-period 1.	UDPJ, ben64, L101, size=18, type=uc, length=2, time-period 2.	UDPJ, ben64, L102, size=16, type=uc, length=2, time-period 1.	UDPJ, ben64, L102, size=16, type=uc, length=2, time-period 2.	UDPJ, ben64, L103, size=15, type=uc, length=2, time-period 1.	UDPJ, ben64, L103, size=15, type=uc, length=2, time-period 2.
G317, type=poc, number of students=25	NK320, type=poc, number of students=80	E320, type=poc, number of students=25	E322, type=poc, number of students=16	D312, type=poc, number of students=20	D312, type=poc, number of students=20
5	4	45	12	7	21

**Şekil 3.5** Üç satırlı bir kromozom örneği (Vorac et al., 2002)

Corne vd. (1994), her biri üç genden oluşan ve dolayısıyla toplam uzunluğu “*ders sayısı\*3*” kadar olan bir kromozom yapısı tasarlamıştır. Brinci gen zamanı, ikinci gen dersliği ve üçüncü gen de öğretim elemanını temsil etmektedir. Kromozomda dersler sırasıyla temsil edilmektedir. Bu çalışmada hem ders hem de sınav zaman çizelgeleme problemi bir arada ele alınmıştır.

Deris vd. (1999) kromozom uzunluğu ders sayısı kadar olan ve allellerde zaman dilimi ve derslik bilgisinin yer aldığı bir kromozom tasarlamıştır. Geçmiş yıllardaki çizelgelerin de kullanıldığı bu çalışmada genetik algoritmalar ve kısıt temelli çıkarsama yönteminden melez bir algoritma geliştirilmiştir.

### Rassal Anahtar Temelli Genetik Algoritmalar

Temel bir GA, beş temel elemandan oluşmakta ve bunların her biri algoritmanın performansını önemli derecede etkilemektedir (Michalewicz vd., 1991; Karaboğa, 2004). Bu parametreler:

1. Problem için çözümlerin genetik gösterimi
2. Çözümlerin başlangıç popülasyonunu oluşturacak bir yöntem
3. Çözümleri uygunluk açısından değerlendirecek değerlendirme fonksiyonu

4. Genetik kompozisyonu deęiřtirecek iřlemciler
  5. Kontrol parametrelerinin deęerleri
- 'dir.

GA'larda karřılařılan ilk soru kromozomların oluřturulması ve problemin cözümünü göstermek için seçilen genetik gösterim türünün seçimi ile ilgilidir. Genetik algoritmalarda problem cözümleri kromozom olarak kodlanır. Bir bireyin GA'da nasıl tanımlandığı büyük önem taşır. Bu tanımlama, cözüm uzayının sınırlarını etkiler. Deęiřkenlerin olası deęerleri belirli aralıklara ayrılmalı veya her bir koda karřılık gelecek deęiřken deęeri belirlenmelidir.

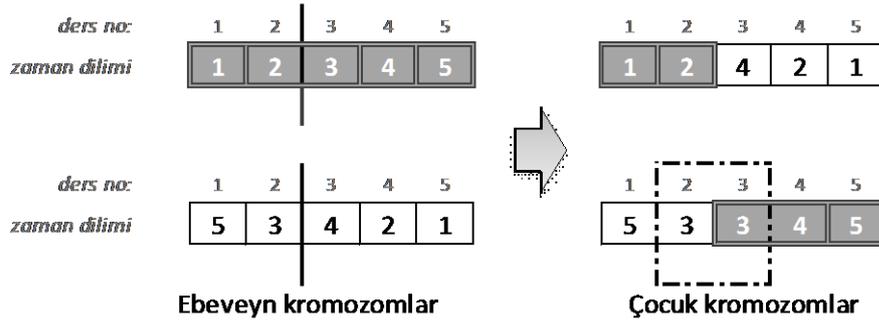
Gen ve Cheng (2000)'in de belirttiđi gibi, en sık kullanılan gösterim tipi ikili düzende kodlama olmasına rađmen gezgin satıcı, çizelgeleme, araç rotalama, tesis yerleřimi gibi bazı kombinatorik en iyileme problemleri için permütasyon kodlama, matris ve vektör tabanlı kodlama ve rassal anahtarlar gibi farklı kodlama yapıları kullanılmaktadır.

Ele alınan problemin yapısına bađlı olan kodlama türüne, problemin cözümüne başlamadan önce karar verilmelidir. Yanlıř bir kodlama türü seçimi geliştirilen algoritmanın etkinliđini ve problemin cözümünü dođrudan etkileyecektir (Rothlauf vd., 2002; Mendes vd., 2005; Eclund vd., 2006). "0" ve "1" deęerlerini alabilen genlerin kullanıldıđı ve bu deęerlerin bir vektör řeklinde yazılarak kromozomun oluřturulduđu *ikili kodlama*, kromozomun nesnelere ve nesnelere arası iřlemleri içeren bir ađaç yapısından oluřtuđu *ađaç kodlama*, her kromozomun bir sayı dizisinden oluřtuđu *permütasyon kodlama* yaygın řekilde kullanılan kodlama türlerindedir.

Ancak ikili kodlama, birçok problem için uygulanması zor bir türdür. Örneđin çizelgeleme problemleri gibi permütasyon ve kombinasyon içeren problemlerde, genetik iřlemcilerin uygulanmasının ardından onarım (repair) iřlemlerinin ya da farklı genetik iřlemcilerin kullanılması gerekmektedir. Ayrıca, her gen sadece bir özelliđi temsil ettiđinden, küçük boyutlu problemlerde bile uzun kromozomlar oluřmaktadır.

Ağaç kodlama ise program geliştirmek için kullanılan en ideal kodlama biçimlerinden birisidir (Tunalıoğlu ve Öcalan, 2007). Ancak ağaç yapısı, kontrol edilemeyecek bir biçimde büyüyebilir. Büyük ağaçların anlaşılması ve basitleştirilmesi de oldukça zor olabilir.

Kombinatorik en iyileme problemlerinin çözümünde GA'ların kullanılmasında birçok problem ortaya çıkar. Bu problemlerden en sık karşılaşılanı, çaprazlama ve mutasyon işlemcilerinin uygun olmayan çözümler üretmesidir. Örneğin Şekil 3.6'da da gösterildiği gibi, çaprazlama işlemi sonucunda elde edilecek yeni çözümde tek derslik olması varsayımı altında aynı zaman dilimine birden fazla ders atanabilir.



**Şekil 3.6** 5 ders ve 5 zaman dilimi için tek nokta çaprazlama sonucu elde edilen uygun olmayan çözüm

Kromozom boyu arttıkça işlem yükü artacaktır. Bundan kaçınmak için, kromozomların matris şeklinde tanımlanmasının ise farklı sakıncaları bulunmaktadır. Örneğin, matris ve vektör tabanlı kodlamada kromozomlar  $|N|*|M|$  ( $|N|$ : tedarikçi sayısı,  $|M|$ : müşteri sayısı) boyutunda matris ve vektör, direkt kodlamada ise  $2(|N| + |M| - 1)$  boyutunda bir vektör olarak tanımlanmaktadır (Karaoğlu ve Altıparmak, 2005). Ancak bu kodlamalarda en büyük sakınca, bir çözümü göstermek için gerekli olan hafıza ihtiyacının büyük olmasıdır. Ayrıca, çaprazlama ve mutasyon işlemcileri ile elde edilen yeni bireyler uygun çözüm vermeyebilir. Dolayısıyla, bu tür kodlamalarda karşılaşılan sorunları gidermek için çözüm gösterimini değiştirmek, yeni çaprazlama ya da

mutasyon işlemcileri tanımlamak ya da klasik işlemcileri uyguladıktan sonra genetik onarma işlemleri yapmak gerekir (Colorni et al., 1992). Bean (1994) tarafından önerilen *rassal anahtar temelli GA*'lar ise, bu gibi işlemleri gerektirmeden tüm bireyler için uygun çözümü garantilemektedir.

İkili ya da permütasyon gösterim ile kodlanan zaman çizelgeleme problemlerinde karşılaşılan bir diğer sorun da gende birden fazla bilginin saklanamamasıdır. Örneğin, ders sayısı kadar gen içeren bir kromozomda gendeki bilgi, dersin yapılacağı zaman dilimi, dersi veren öğretim elemanı bilgisi ya da dersin yapılacağı derslik bilgilerinden sadece biri olabilir. Bu sorunun üstesinden gelmek için kullanılan matris gösteriminde, “ders sayısı \* kaynak sayısı” boyutunda bir kromozom kullanmanın da daha önce bahsedilen farklı sakıncaları bulunmaktadır.

Bean (1994) tarafından önerilen rassal anahtar gösterimi ile bir çözüm rassal sayılar ile kodlanmaktadır. Bu rassal değerler, çözümü deşifre etmek için sıralama anahtarları olarak kullanılmaktadır. Bu çalışmada kullanılacak olan rassal anahtarlı gösterim ve literatürdeki diğer gösterimler arasındaki temel fark rassal sayıların çözümlerin gösteriminde etiket olarak kullanılmasıdır. Rassal sayılar tipik olarak  $[0,1]^n$  uzayından seçilir. Genetik algoritma, gerçek problemi temsil eden uzayın yerine geçecek olan bu uzayı arar. Rassal anahtarlar uzayındaki noktalar değerlendirme için problemin gerçek uzayındaki noktalara dönüştürülür.

Bu kodlama türü, problem yapısına bağlı olmadan iyi çözümler vermesinin yanı sıra, bir gende birden fazla bilginin saklanabilmesine olanak sağlayarak diğer kodlama türlerinden farklılaşmaktadır.

Literatürde rassal anahtar yaklaşımını kullanan çeşitli çalışmalar yer almaktadır. Örneğin, Kim et al. (2002) çoklu-robot kaynak görevi sıralama problem, Gonçaves ve Resende (2005) makine hücreleri oluşturma ve ürün ailelerini belirleme problem, Mendes et al. (2005) kaynak kısıtlı proje çizelgeleme problem, Daniel ve Rajendran (2006) tedarik zinciri yönetiminde taban-stok miktarlarının belirlenmesi, Eklund (2006) koşullu konumsal benzetim, Eukland et al. (2006) deterministik olmayan polinom

zamanlı tesis yerleşimi problem, Norman ve Smith (1997) ve Norman et al. (1998) tesis yerleşimi problem, Snyder ve Daskin (2006) genelleştirilmiş gezgin satıcı problemi için rassal anahtar temelli genetik algoritmalar geliştiren çalışmalarıdır.

Bu çalışmada, eğitimsel çizelgeleme problemlerinin GA'lar ile çözümünde karşılaşılan büyük hafıza, özel işlemci geliştirme ve genetik onarım işmecileri kullanma ihtiyacı gibi sakıncaları ortadan kaldırmak amacıyla rassal anahtar temelli bir GA (RKGA) önerilmektedir. Bölüm 3.2.2'de, geliştirilen RKGA tanıtılmaktadır. Bölüm 3.2.4'te ise ele alınan problemin çok amaçlı yapısından dolayı, geliştirilen RKGA bir çok amaçlı GA olan NSGA-II algoritması ile entegre edilmiştir. Test problemlerinin çözümünden elde edilen sonuçlar Bölüm 3.2.5'te karşılaştırılmaktadır.

### 3.2.2 Rassal anahtar temelli genetik algoritmanın (RKGA) adımları

Rassal anahtar yöntemi, topluluktaki her bir bireyi temsil eden kromozomun rassal sayılar kullanılarak kodlanması temeline dayanır. Bir kromozom, çizelgelenecek toplam ders sayısı kadar genden oluşmaktadır. Şekil 3.7'de 100 ders için bir kromozomun yapısı gösterilmiştir.

<i>Ders no</i>	1	2	3	4	5	6	.	.	99	100
	1335	2357	1190	7809	1010	2369	.	.	.	9813

**Şekil 3.7** RKGA'da kullanılan kromozom yapısı

Kromozom çözümlenirken rassal sayının ilk iki basamağı derslik, son iki basamağı ise zaman dilimi atamasında kullanılacaktır. Örneğin, Şekil 3.8'de verilen kromozomda ilk dersin atanabileceği zaman dilimleri 13, 19 ve 44 olarak belirlenmiştir.

Dersin bu zaman dilimlerinden hangisine atanacağını belirlemek üzere birinci gende yeralan rassal anahtarın son iki basamağı kullanılmaktadır. Yapılan hesaplama sonucu, birinci ders üç aday zaman diliminden ikinci sıradaki zaman dilimine ( $t=19$ ) atanacaktır. Kromozomun çözümlenme süreci algoritmanın adımlarında ayrıntılı olarak verilmiştir.

Ders no	1	2	3	4	5	6	.	.	99	100
Derslik & zaman dilimi	1335	2357	1190	7809	1010	2369	.	.	.	9813

13
19
44

$\lfloor (35 \cdot 3) / 100 \rfloor + 1 = 2$

**Şekil 3.8** Rassal anahtar kullanılarak zaman dilimi atamasının yapılması

Başlangıç popülasyonunun oluşturulması sırasında derslerin dersliklere atanmasında, zorunlu atamaların dışında (bazı dersler sadece bir derslikte yapılabildiği için ilgili dersliğe atanması zorunludur) her dersliğe yaklaşık eşit sayıda dersin atanması ile uygun çözümlerin daha kolay elde edileceği düşünülmüştür. Bu amaçla, geliştirilen bir matematiksel model sayesinde dersliklere en dengeli atama gerçekleştirilmiş ve bu atamanın sonucunun genetik algoritmanın popülasyonunun içine dahil edilmesi sağlanmıştır. Bu işlemler geliştirilen programda tamamen otomatik olarak gerçekleştirilmektedir. Yani, ilgili problem için bir GAMS modeli oluşturulur, GAMS programı çağrılarak bu model çözdürülür ve modelin çıktılarının yazıldığı metin dosyası program tarafından okunarak matematiksel modelin çözümü genetik algoritmanın bir bireyine dönüştürülür.

Rassal anahtar temelli GA'da her zaman uygun çözümü garanti eden bir çözümlenme prosedürü geliştirilmiş ve elitizm stratejisi, çaprazlama için parametrik düzgün çaprazlama ve mutasyon için göç işlemcileri kullanılmıştır.

### Çaprazlama işlemcisi

Algoritmada geleneksel tek/çok nokta çaprazlama işlemcisi yerine *parametrik çaprazlama işlemcisi* kullanılmıştır. Parametrik çaprazlama, RKGA'nın parametresi olarak tanımlanan *tura olasılığına* göre uygulanır. Bu işlemci ile rasgele seçilen iki kromozomdan yeni bir kromozom oluşturulur. Yeni kromozomun her geni için rasgele bir sayı türetilir, eğer bu sayı belirlenen tura olasılığından küçük ise yeni kromozomun ilgili geni birinci kromozomdan, büyük ise ikinci kromozomdan kopyalanır. Tura olasılığının 0.6 olduğu örnek bir parametrik çaprazlama Şekil 3.9'da verilmiştir.

Kromozom1	1214	2241	6173	...	1829
Kromozom2	1132	4342	3324	...	4453
Rassal sayı	.17	.77	.89	...	.54
Yeni birey	1214	4342	3324	...	1829

**Şekil 3.9** Parametrik çaprazlama örneği

### Göç işlemcisi

RKGA'da mutasyon işlemcisi olarak da Bean (1994) tarafından önerilen *göç işlemcisi* kullanılmıştır. Mutasyon ile uygun olmayan çözümler elde edilebileceğinden, çok düşük bir olasılıkla her bir gene mutasyon uygulamak yerine, göç işlemcisi kullanılmıştır. Göç işlemcisi ile her nesilden belirlenen göç oranında birey, bir sonraki nesile aktarılmaktadır.

### Elitizm stratejisi

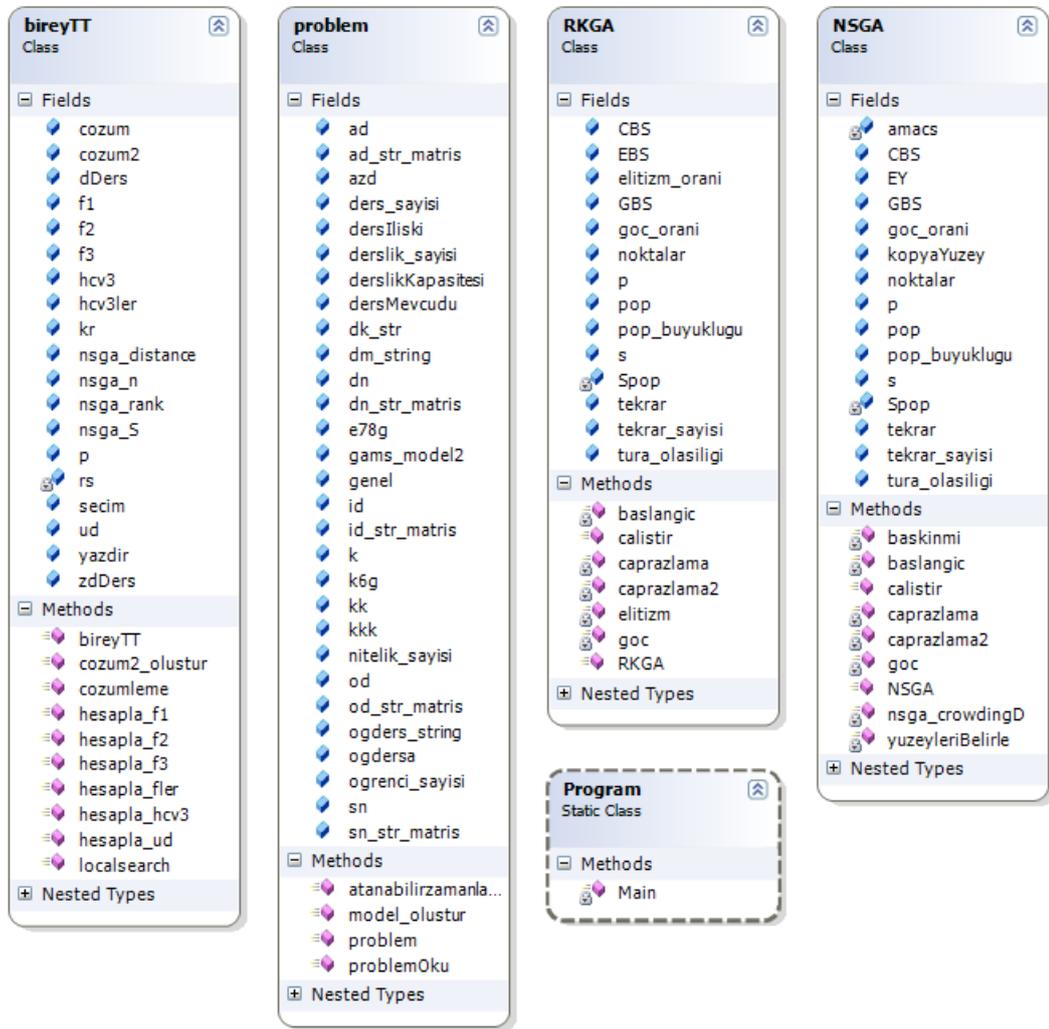
Seçme, çaprazlama ve mutasyon işlemleri sonrasında mevcut topluluğun en iyi uygunluk değerine sahip bireyi bir sonraki nesle aktarılamayabilir. Bunu önlemek için bu işlemlerden sonra, bir önceki topluluğun en iyi (seçkin, elit) bir veya daha çok bireyi, yeni oluşturulan topluluğa doğrudan aktarılır. Elitizm adı verilen bu yaklaşım genetik algoritmalarda yaygın olarak kullanılmakta ve daha kaliteli çözümler için olanak sağlamaktadır (Özkan vd., 2008). RKGGA'da da belirlenen elitizm oranında birey bir sonraki nesle aktarılmaktadır.

Ders zaman çizelgeleme probleminin çözümü için C#'da geliştirilen rassal anahtarlı genetik algoritma temelli program literatürde verilen ders çizelgeleme problemlerini okuyacak şekilde tasarlanmıştır. Program üç temel sınıftan (class) oluşmaktadır: problem, bireyTT, RKGGA.

Problem sınıfı yardımıyla herhangi bir problem, hazırlanan arayüzden seçilir. Bu seçim sonrasında, problem ile ilgili derslik sayısı, öğrenci sayısı, nitelik sayısı, ders sayısı, derslik kapasiteleri, öğrenci-ders matrisi, ders-nitelik matrisi, derslik nitelik matrisi gibi parametreler elde edilmiş olur. Bu parametreler yardımıyla probleme özel veri yapısı dinamik olarak oluşturulmaktadır.

Genetik algoritmadaki popülasyonda yer alan bireyler bireyTT sınıfı yardımı ile oluşturulur. BireyTT sınıfı çözümü ifade eden rassal sayılardan oluşan kromozomu, rassal kromozomun çözümlenerek problemin bir çözümünün elde edilmesini sağlayan çözümlenme metodunu ve elde edilen çözüm için uyum değerini hesaplayan hesapla\_ud metodunu içermektedir.

Algoritmanın ilk aşamasında başlangıç nesli oluşturulmaktadır. Nesilde bireyler zorunlu kısıtlar sağlanacak şekilde oluşturulmaktadır. Rassal anahtarlı genetik algoritmanın temel adımları ise RKGGA sınıfı altında toplanmıştır. Geliştirilen programın sınıf diyagramı Şekil 3.10'da verilmiştir.



Şekil 3.10 RKGA'nın sınıf diyagramı

RKGA'ya ait kodlar EK-2'de verilmiştir.

..... problem sınıfı .....

**Problem okuma metodu** - `problemOku()`

Problem parametrelerini ilgili metin dosyasından oku (sırasıyla ders, derslik, nitelik ve öğrenci sayıları; derslik kapasiteleri; öğrenci-ders matrisi; derslik-nitelik matrisi ve ders-nitelik matrisi).

Ders mevcutlarını hesapla.

Öğrencilerin aldığı dersler doğrultusunda, ilişkili dersler matrisini oluştur.

Her ders için mevcuduna uygun kapasiteye ve aynı zamanda dersin gerektirdiği niteliklere sahip olan **atanabilir derslikler** listesini oluştur.

Her derslik için **atanabilir zaman dilimleri** listesini oluştur.

#### **Model oluşturma metodu - model\_olustur()**

Problem için dersliklere yaklaşık eşit sayıda ders atayacak şekilde uygun çözüm veren GAMS modelini oluşturur.

..... bireyTT sınıfı .....

#### **Yapıcı Metod - bireyTT()**

Toplam ders sayısı kadar genden oluşan ve her bir gende dört basamaklı Rassal Anahtar olan ilk bireyi oluştur.

#### **Çözümleme Metodu - cozumleme()**

- i. Her derslik için atanabilir zamanları sıfırla.
- ii. Her bir zaman dilimi için, **zaman diliminde yapılan dersler** {zdDers[i]} listesini oluştur.
- iii. Her derslik için, **dersliğe atanan dersler** {dDers[i]} listesini oluştur.
- iv. Derslikleri atanabilir derslikler listesine göre derslere ata.
  - a. Dersleri toplam atanabilir derslik sayılarına göre sırala.
  - b. Toplam atanabilir derslik sayısı bir olan dersleri o dersliğe ata.
  - c. Sıradaki dersin atanmasında kullanılacak olasılıkları dersliklere o ana kadar atanmış olan ders sayısına göre hesapla.

Her ders için **ilgili dersliđe atanma olasılıkları** ( $o_{jk}$ ) izleyen şekilde belirlenmektedir:

$ad_{jk} = j$ . dersin atanabileceđi derslikler matrisi

$nd_k = k$ . dersliđe o ana kadar atanmış toplam ders sayısı

$$toplamlam = \sum_j \sum_k ad_{jk} * (45 - nd_k)$$

$$o_{jk} = \frac{ad_{jk} * (45 - nd_k)}{toplamlam}$$

- d. Kromozomdaki rassal anahtarın ilk iki basamađı ile dersliđe atanma olasılıkları kullanılarak derslik ataması gerçekleştirilir. Bir başka ifadeyle  $rs_{1j} < o_{jk} * 100$  ise  $j$ . ders  $k$ . dersliđe atanır.
- v. Ders-derslik çiftine zaman dilimlerini ata.
- Günlerin en son zaman dilimlerine atamayı mümkünse en sona bırak.
  - İlişkili dersler aynı zaman dilimine atanmayacak.
  - Kromozomdaki Rassal Anahtar'ın son iki basamađı ( $rs_2$ ) kullanılarak zaman dilimi ataması yapılır.
- $j$ . dersin ilişkili olduđu dersleri kontrol et. İlişkili ders yoksa,  $j$ . dersin atandıđı dersliđin atanabilir zaman dilimlerinden, rassal anahtarın ikinci kısmını kullanarak uygun zaman dilimini hesapla. Atanan zaman dilimini, ilgili dersliđin atanabilir zaman dilimleri listesinden çıkar.

“*Toplam atanabilir zaman dilimi sayısı \* rs<sub>2</sub>*” ile ders-derslik çiftine kaçınıcı atanabilir zaman diliminin atanacađı belirlenir.

$j$ . dersin ilişkili olduđu ders varsa, tüm ilişkili derslerin

atandığı zaman dilimlerini, ilgili dersliklerin atanabilir zaman dilimleri listesinden çıkar.  $j$ . dersin atandığı derslik için, kalan zaman dilimlerinden yine rassal anahtarın ikinci kısmını kullanarak zaman dilimi ataması yap.

Zaman diliminde yapılan dersler listesine {zdDers[cozum[ders,1]}atanan yeni dersi ekle.

**Uyum değeri hesaplama metodu - hesapla\_ud() :**

Her bireyin uyum değerini ( $ud$ ) hesapla,  $ud = f_1 + f_2 + f_3$

**f1 hesaplama metodu- hesapla\_f1() :**

Tüm öğrencilerin tüm günlerde son zaman dilimlerine atanan ders sayısını hesapla ( $f_1$ ).

**f2 hesaplama metodu- hesapla\_f2() :**

Tüm öğrencilerin tüm günlerde, gün içinde ardışık ikiden fazla dersi olma durumu sayısını hesapla ( $f_2$ ).

**f3 hesaplama metodu- hesapla\_f3() :**

Tüm günlerde, öğrencilerin günde sadece bir dersi olma durumu sayısını hesapla ( $f_3$ ).

..... RKGA sınıfı .....

**Başlangıç topluluğunu oluşturma metodu - baslangic() :**

Topluluk büyüklüğü kadar her biri bireyTT sınıfından türetilmiş bir nesne olan kromozomları oluşturur.

**Elitizm metodu - elitizm() :**

En iyi bireyleri bir sonraki nesile aktarmak için elitist stratejisini kullan.

**Çaprazlama metodu - caprazlama()**

Parametrik çaprazlama işlemcisini çalıştır.

**Göç metodu - goc()**

Göç işlemcisini çalıştır. Göç edecek birey sayısı kadar birey bir sonraki nesile aktar.

**Çalıştır metodu - calistir()**

Başlangıç metodunu çağır.

Tekrar sayısı kadar yap {

    Elitizm metodunu çağır

    Çaprazlama metodunu çağır

    Göç metodunu çağır

}

Çalışmada ele alınan test problemlerinin çözüm değerlerinin karşılaştırılmasında sağlanmayan toplam esnek kısıt sayısı temel alındığından, her bireyin uyum değeri ( $ud = f_1 + f_2 + f_3$ ) sağlanmayan üç esnek kısıt sayısının toplanması ile elde edilmektedir.

**3.2.3 Çok amaçlı genetik algoritmalar**

Bir problemde birden fazla amacın varlığı, tek bir eniyi çözüm yerine Pareto optimal olarak da adlandırılan bir eniyi çözümler kümesinin oluşmasına yol açmaktadır. Pareto-Optimal çözüm; amaçların herhangi biri için en kötü olmayan ve en azından bir amaç için diğerlerinden daha iyi olan çözümdür. Diğer bir ifadeyle çözüm kümesindeki diğer herhangi bir çözüm tarafından bastırılmamış olan çözümdür. Bu durum, mümkün

olduğunca çok Pareto-optimal çözümler bulunmasını teşvik eder (Deb et al., 2002). Geleneksel eniyileme yöntemleri, çok amaçlı eniyileme problemini bir tek amaçlı eniyileme problemine dönüştürürler. Ancak bu yöntemlerin, her problemin çözümü için yeterli olmadığı görülmüştür. Son zamanlarda, çok amaçlı optimizasyon problemlerinin çözümünde genetik ve evrimsel algoritmalar yaygın olarak kullanılmaya başlanmıştır. Çok amaçlı genetik algoritmaların performanslarını değerlendirmede Pareto yüzeyin bulunması önemli bir göstergedir (Sağ ve Çunkaş, 2009). Son yıllarda çeşitli çok amaçlı evrimsel algoritmalar (MOEA) geliştirilmiştir. Bunun en önemli nedeni, bu algoritmaların tek bir çalıştırmada çok sayıda Pareto-optimal çözümler bulma yetenekleridir.

Çok amaçlı eniyileme problemlerinin çözümü için geliştirilmiş çeşitli MOEA'lar; Schaffer (1985) tarafından önerilen Vektör Değerlendirmeli Genetik Algoritma (VEGA), Fonseca ve Fleming (1993) tarafından önerilen Çok Amaçlı Genetik Algoritma (MOGA), Horn ve Nafpliotis (1993) tarafından önerilen Hücrelendirilmiş Pareto Genetik Algoritması (NPGA), Srinivas ve Deb (1994) tarafından önerilen Bastırılmamış Sınıflandırmalı Genetik Algoritma (NSGA), Bastırılmamış Sınıflandırmalı Genetik Algoritma II (NSGA-II), Zitzler ve Thiele (1999) tarafından önerilen Kuvvet Pareto Evrimsel Algoritma (SPEA), Kuvvet Pareto Evrimsel Algoritma 2 (SPEA2) ve Pareto Zarflama-Temelli Seçim Algoritması (PESA) olarak verilebilir.

Çok amaçlı genetik algoritmalar Sağ ve Çunkaş (2009) tarafından kısaca izleyen şekilde açıklanmıştır:

- VEGA'nın bilinen genetik algoritmadan ayrılan tek yanı birden fazla olan amaçlar için amaç fonksiyonu sayısınca arama uzayının alt bölgelere ayrılmasıdır. Shaffer'ın önermiş olduğu VEGA, çok amaçlı optimizasyon problemlerinin çözümünde genetik algoritmaların kullanılması bakımından öncü bir çalışma niteliği taşımaktadır. Ancak topluluk temelli ve ilkel bir seçim stratejisine sahip olmasından dolayı az sayıda ve yalnızca uç noktalardaki çözümlere odaklanan VEGA'nın, verimsiz bir algoritma olduğu açıktır.

- MOGA topluluk temelli bir evrimsel eniyileme algoritmasıdır. Bu yaklaşımda çoklu amaçlar, geleneksel yöntemlere paralellik göstererek değişken ağırlıklı katsayılarla sayısal bir fonksiyon içinde birleştirilir. Rasgele üretilen bu katsayılar sayesinde MOGA, VEGA'ya göre daha iyi bir dağılım gösterir. Ancak kötü bir yakınsamaya sahiptir.
- NPGA pareto üstünlüğe dayalı turnuva seçimi ve uygunluk paylaşım tekniği kullanarak popülasyondaki çeşitliliğin korunmasını sağlar.
- NSGA'da topluluk bastırılmamışlık ilkesiyle derecelendirilir. Tüm bastırılmamış bireyler sahte uygunluk değeri ile bir kategori içinde sınıflandırılır. Kategori sayısı topluluk boyutuyla orantılıdır. Aynı kategorideki bireylerin hepsine aynı uygunluk değeri atanarak eşit seviyede yeniden üretilme potansiyeli sağlanır. Daha sonra popülasyonda çeşitliliği sağlamak için paylaşım metodu uygulanır. İlk yüzeydeki bireyler geçici olarak ihmal edilir ve kalan topluluk aynı süreçten geçirilerek ikinci bastırılmamış yüzey tanımlanır. Önceki yüzeyin minimum paylaşılan sahte uygunluk değerinden daha küçük tutulan yeni bir sahte uygunluk değeri yeni yüzeydeki tüm bireylere atanır. Bu işlem popülasyondaki tüm bireyler sınıflandırılıncaya kadar devam eder.
- NSGA-II'de ilk olarak rasgele ebeveyn topluluğu oluşturulur. Topluluk, Pareto üstünlüklerine dayalı olarak sıralanır. Her çözümün uygunluğu, kendi bastırılmamışlık seviyesine eşit olarak atanır. Sonra, bilinen ikili turnuva seçimi, çaprazlama ve mutasyon işlemcileri, N boyutlu popülasyonun bireylerini oluşturmak için ebeveyn popülasyona uygulanır. Seçkinlik işlemi ise, önceki en iyi bastırılmamış çözümlerle mevcut topluluk karşılaştırılarak gerçekleştirilecektir. Bu yüzden, başlangıç jenerasyonundan sonra prosedür farklı bir şekilde işler.
- SPEA2, genellikle başarılı sonuçlar üreten SPEA algoritmasının; uygunluk ataması, yoğunluk tahmini ve arşiv küçültme gibi konulardaki bazı eksiklikleri

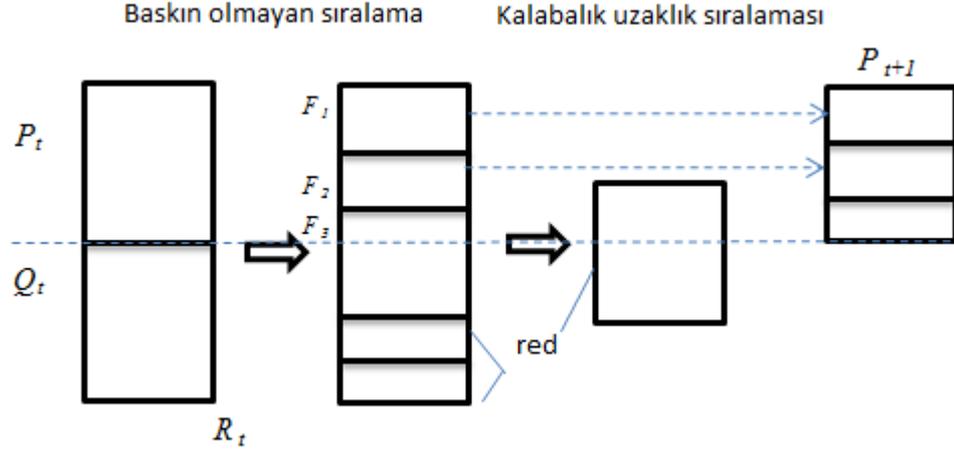
göz önüne alınarak geliştirilmiştir. SPEA2 yoğunluk bilgisini kullanan ince-taneli (fine-grained) uygunluk atama stratejisini kullanır. Ayrıca bastırılmamış bireyleri harici olarak depolayan arşivin boyutu da sabittir. Bastırılmamış bireylerin sayısı önceden tanımlanmış arşiv boyutundan daha küçükse, arşiv bastırılmamış bireylerle doldurulur. Ek olarak; bastırılmamış yüzey arşiv boyutunu aştığı zaman kullanılan kümeleme tekniği, benzer özellikleri olan ama sınır noktalarını kaybetmeyen seçenек bir küçültme yöntemiyle değiştirilmiştir. Son olarak, SPEA'dan bir başka farkı seçim işleminde sadece arşiv üyelerinin kullanılmasıdır.

- PESA algoritmasında seçim ve çeşitliliğin korunması hiper-grid aracılığıyla kontrol edilir. PESA'nın seçim yöntemi, bastırılan çözümlerin derecelendirilmesine bağlı olan sıkıştırma faktörü (squeeze factor) adı verilen bir katsayı kullanır. Çeşitliliğin sağlanması için kullanılan yöntem ise, diğer algoritmaların da kullandıkları yöntemlere benzemektedir.

### 3.2.4 NSGA-II temelli çok amaçlı genetik algoritma

Bir çok test probleminin çözümünden elde edilen sonuçlara göre, *NSGA-II (hızlı ve elitist çok amaçlı evrimsel algoritma)* algoritması Pareto-Optimal çözümler üretmede en iyi performansları göstermektedir (Zitzler et al., 2001; Deb et.al., 2002; Sağ ve Çunkaş, 2009).

Ele alınan ders-derslik-zaman çizelgeleme test problemlerine ait çözümlerin literatürde yer alan çözümlerle karşılaştırılabilmesi için, amaç fonksiyonlarının sağlanmayan toplam esnek kısıt sayısını temel aldığı belirtilmiştir. Dolayısıyla, RKGA algoritmasında kromozomların uyum değerleri bu şekilde hesaplanmıştır. Ayrıca problemin çok amaçlı yapısından dolayı, bu bölümde RKGA algoritmasında uyum değerleri bir çok amaçlı GA olan NSGA-II algoritması kullanılarak hesaplanmıştır. Geliştirilen algoritmanın kodu EK-3'de verilmiştir. NSGA-II algoritmasının çalışma prosedürü Şekil 3.11 ile gösterilmiştir.



Şekil 3.11 NSGA-II algoritması (Deb et al.,2002)

Algoritmaya göre  $N$  tane birey içeren mevcut  $P_t$  popülasyonu, göç ve çaprazlama işlemcileri kullanılarak oluşturulan  $Q_t$  popülasyonu ile birleştirilerek  $R_t$  isimli  $2N$  bireyden oluşan topluluk oluşturulur. Bu popülasyonda bulunan bireylere, birbirine baskın olmayan bireyler aynı pareto yüzeyde olacak şekilde sıra numaraları atanır. Sıra numarası en küçük olan pareto etkin yüzeyden başlanarak  $R_t$  popülasyonunda bulunan bireyler bir sonraki topluluk  $P_{t+1}$ 'e aktarılır.  $N$ . bireyin içinde bulunduğu pareto yüzeyde bulunan bireylerden hangilerinin bir sonraki nesile aktarılacağı özel olarak tanımlanmış olan uzaklık formülü (*kalabalık uzaklık-crowding distance*) kullanılarak belirlenir.

### 3.2.5 Test problemlerinin denenmesi

Burke ve diğerlerinin (2003b) geliştirdikleri Hiper Sezgisel (HH) algoritması ile RRLS (Socha vd., 2002) ve ANT (Socha vd., 2002) meta sezgisel algoritmalarından *small*, *medium* ve *large* problem türleri için elde edilen problem sonuçları (toplam sağlanmayan esnek kısıt sayıları) Çizelge 3.7'de verilmiştir.

Çizelge 3.7 small, medium ve large problem türleri için literatürde elde edilen sonuçlar

<b>Problem</b>	<b>HH<sup>5</sup></b>	<b>RRLS<sup>6</sup></b>	<b>ANT<sup>7</sup></b>
S1	1 / 2.2 / <b>1</b>	8	1
S2	1 / 3 / <b>2</b>	11	3
S3	1 / 1.4 / <b>0</b>	8	1
S4	1 / 1.8 / <b>1</b>	7	1
S5	1 / 0.2 / <b>0</b>	5	<b>0</b>
M1	1 / 179 / <b>146</b>	199	195
M2	1 / 197.6 / <b>173</b>	202.5	184
M3	1 / 295.4 / 267	77.5% Inf	<b>248</b>
M4	1 / 180 / 169	177.5	<b>164.5</b>
M5	0.8 / 388.5 / 303	100% Inf	<b>219.5</b>
L	0.2 / 1166 / 1166	100% Inf	<b>851.5</b>

HH sütununda yer alan veriler şu şekilde açıklanabilir:

“5 çalıştırmada elde edilen uygun çözüm oranı / uygun çözümler için ortalama sağlanmayan esnek kısıt sayısı / tüm çalıştırmalar içindeki en iyi sağlanmayan zorunlu kısıt sayısı”

RRLS algoritması için verilen sonuçlar da, 40 çalıştırım sonucunda elde edilen sağlanmayan esnek kısıt sayılarının ortanca değeridir. M3, M5 ve large problem türleri için bu algoritma ile uygun çözüm elde edilememektedir. Benzer şekilde, ANT algoritması için verilen sonuçlar elde edilen sağlanmayan esnek kısıt sayılarının ortanca değeridir.

Bu çalışmada geliştirilen rassal anahtar temelli GA ile 20 çalıştırmada elitizm ile göç oranlarını 0.1, tura olasılığının 0.8 ve topluluk büyüklüğünün 20 olduğu durumda

<sup>5</sup> (Burke et al., 2003b)

<sup>6</sup> (Socha et al., 2002)

<sup>7</sup> (Socha et al., 2002)

elde edilen en iyi sonuçlar Çizelge 3.8’de verilmiştir. Burke vd. (2003b)’nin geliştirmiş olduğu hipersezgisel yaklaşımın çözüm aşamasında zorunlu kısıt ihlaline izin verilmektedir. Geliştirilen RKGA algoritmasının diğer algoritmalara olan üstünlüğü, başlangıç popülasyonunun her zaman uygun bir çözüm olması ve göç işlemcisi kullanılarak uygun çözümlerin bozulmaması yanısıra her çalıştırmada sağlanmayan zorunlu kısıt sayısının sıfır olmasıdır.

**Çizelge 3.8** RKGA ile elde edilen sonuçlar

<b>Problem</b>	<b>RKGA ile en iyi sağlanmayan esnek kısıt sayısı</b>
S1	20
S2	45
S3	32
S4	28
S5	32
M1	530
M2	523
M3	347
M4	511
M5	542
L	1132

HH ve ANT algoritmaları ile elde edilen sonuçlar RKGA’dan elde edilen sonuçlara göre daha iyi olduğu görülmektedir. Ancak, RKGA ile tüm problemler için zorunlu kısıtların sağlandığı uygun çözümler garanti edilmektedir. Bu özellik RKGA’nın bir üstünlüğüdür.

NSGA-II temelli RKGA algoritması ile problemlerin çözdürülmesi sonucunda ise RKGA algoritması ile elde edilen sonuçlar kadar iyi çözümlere ulaşamamıştır.

NSGA-II temelli RKGA algoritması ile elde edilen en iyi deęerler izelge 3.9 ile verilmiřtir.

**izelge 3.9** NSGA-II temelli RKGA ile elde edilen sonular

<b>Problem</b>	<b>NSGA-II ile en iyi saęlanmayan esnek kısıt sayısı</b>
S1	69
S2	80
S3	87
S4	79
S5	93
M1	657
M2	633
M3	448
M4	636
M5	697
L	1259

alıřmanın bu blmnde ders zaman izelgeleme problemi ders, derslik ve zaman dilimai boyutlarıyla ele alınmıřtır. Byk boyutlu problemlerin zm iin geliřtirilen GA'lar her alıřtırmıda tm sıkı kısıtları saęlayan zmler retmektedir. Geliřtirilen algoritmaların, saęlanmayan esnek kısıt sayılarını da olabildięince enkklemeleri iin farklı metasezgisel algoritmalar ile birleřtirilerek geliřtirilmesi mmkndr.

İzleyen blmde farklı yapısından dolayı, bir bařka faaliyetlerin kaynaklara atandıęı eęitimsel zaman izelgeleme problemi olan sınav-derslik-zaman izelgeleme problemi ele alınmaktadır.

## BÖLÜM 4

### SINAV DERSLİK ZAMAN ÇİZELGELEME PROBLEMİ İÇİN ÇÖZÜM YAKLAŞIMLARI

Bu bölümde, Bölüm 2.2.2’de belirtilen sınav çizelgeleme problemlerinden *sınav-derslik-zaman çizelgeleme* ve *gözetmen-sınav atama* problemleri çok amaçlı olarak ele alınmış ve her bir problemin çözümü için farklı modeller geliştirilmiştir. Geliştirilen modeller ve elde edilen çözümler izleyen alt bölümlerde verilmiştir.

#### 4.1 Sınav-Derslik-Zaman Çizelgeleme Problemi ve Matematiksel Model ile Çözümü

Çalışmada ele alınan sınav-derslik-zaman çizelgeleme problemi için sınavları derslik ve zaman dilimlerine, gerektiğinde bir dersin sınavını birden fazla dersliğe yerleştirerek atayan, karma tamsayılı, üç amaçlı bir matematiksel model kurulmuştur. Geliştirilen model, Eskişehir Osmangazi Üniversitesi Endüstri Mühendisliği Bölümü’nde kullanılmakta olan sınav çizelgeleme sisteminin parametreleri kullanılarak çözdürülmüştür.

Problem, üniversite sınav çizelgesi için ele alınmıştır. Her sınıfın kendi standart derslerinin bulunduğu kabul edilmektedir. Modelde *düzenli öğrenciler* olarak tanımlanan öğrenciler, her sınıfa ait bu derslerin tümünden başarılı olarak bir üst sınıfa geçen ve sadece bulunduğu sınıfın derslerini alan öğrencilerdir. Sınav çizelgeleri de düzenli öğrencilerin sınavlarında çakışma olmayacak şekilde oluşturulmaktadır.

*Düzensiz öğrenciler* ise bulunduğu sınıfın dersleri dışında, alt sınıflara ait başarısız olduğu ders(ler)i de alabilen öğrencilerdir. Dolayısıyla, farklı sınıflara ait dersleri aynı anda alan öğrencilerin yer aldığı bir sistemde sınav çakışmalarının olmadığı bir çizelge elde etmek güçleşmektedir.

Çalışmada problemin sıkı ve esnek kısıtları izleyen şekilde belirlenmiştir.

Sıkı kısıtlar:

1. Her sınav bir zaman dilimine atanmalı..
2. Aynı zaman dilimine düzenli bir öğrencinin birden fazla sınavı atanmamalı..
3. Sınav(lar)ın yapılacağı derslik kapasitesi aşılmamalı.

Esnek kısıtlar:

1. Düzenli bir öğrencinin bir günde girdiği sınavların mümkün olduğunca en fazla iki olması.
2. Düzenli bir öğrencinin bir günde girdiği sınavların toplam süresinin mümkün olduğunca belirli bir üst sınırı (*A* dakika) aşmaması.
3. Düzensiz öğrencilerin sınav çakışmalarının mümkün olduğunca engellenmesi.
4. Sınav haftası boyunca kullanılan toplam derslik sayısının mümkün olduğunca belirli bir üst sınırı (*B* adet derslik) aşmaması

Uygun bir sınav-derslik-zaman çizelgesi elde etmek için tüm sıkı kısıtların sağlanması gerekmektedir. Belirlenen farklı esnek kısıtlar ile de çizelgenin kalitesi arttırılmaya çalışılır.

Geliştirilen modelde her esnek kısıt için bir hedef fonksiyonu oluşturulmuştur. Bu fonksiyonlar ile ifade edilen amaçlar için de hedef değerleri belirlenmiştir. Hedef değerlerinden olabilecek artı yöndeki sapmaları en küçükleyecek şekilde amaç fonksiyonları oluşturulmuştur.

Geliştirilen modele ait varsayımlar izleyen şekilde belirlenmiştir:

- Sınavların başlama saatleri, 09:00, 11:00, 14:00 ve 16:00'dır.

- Zaman blokları ardışık ikişer saattir. Bu durumda, ilk sınav gününden başlamak üzere, 9:00, 11:00, 14:00 ve 16:00 saatlerinde başlayan zaman dilimlerine *1. zaman dilimi*, *2. zaman dilimi*, ..., *24. zaman dilimi* numerasyonu atanmıştır (Çizelge 4.1).
- Bir dersin sınav süresi iki saatten fazla olamaz.
- Öğrenci sayılarından dolayı bazı dersler birden fazla gruba bölünebilir. Bu derslerin sınav soruları aynı olacağından, sınavları da aynı zaman diliminde yapılmalıdır. Modelde bu gruplar ayrı birer ders gibi düşünülmüş ve aynı zaman dilimine atanmaları sağlanmıştır.
- Herhangi bir zaman diliminde, bir sınavın tek bir derslikte yapılabilmesi mümkün değilse, sınav birden fazla derslikte yapılabilir (*room splitting*).

**Çizelge 4.1** Gün temelinde sınav yapılabilir zaman dilimleri kümesi

<i>Gün</i>	<i>t (zaman dilimleri)</i>
1	1, 2, 3, 4
2	5, 6, 7, 8
3	9, 10, 11, 12
4	13, 14, 15, 16
5	17, 18, 19, 20
6	21, 22, 23, 24

Yukarıda açıklanan sınav zaman çizelgeleme problemi için geliştirilen matematiksel model ile ilgili karar değişkenlerinin ve parametrelerin tanımları aşağıda verilmiştir:

*İndis kümeleri*

Sınav indisleri kümesi:  $I = \{1, \dots, m\}$

Derslik indisleri kümesi:  $D = \{1, \dots, n\}$

Zaman dilimleri kümesi:  $T = \{1, \dots, k\}$

Sınıf indisleri kümesi:  $CL = \{1, \dots, f\}$

Sınav günleri kümesi:  $G = \{1, \dots, h\}$

$cl.$  sınıfın standart dersleri kümesi:  $CC = \{(cl, i) \mid cl \in CL, i \in I\}$

Günlere ait zaman blokları kümesi:  $DZ = \{(g, t) \mid g \in G, t \in T\}$

Farklı sınıflara ait olup ortak öğrencisi olan ve çakışması istenmeyen ders ikilileri kümesi :  $OG = \{(i, j) \mid \exists i \in I, \exists j \in I, i \neq j\}$

#### *Parametreler*

$a_i$ :  $i.$  sınava ait öğrenci sayısı

$c_d$ :  $d.$  dersliğin sınav için uygun kapasitesi

$s_i$ :  $i.$  sınavın süresi (dk.)

#### *0-1 Tamsayıli karar değişkenleri*

$y_{it} = \begin{cases} 1 & i. \text{ sınav } t.\text{zaman dilimine atanırsa} \\ 0 & \text{diğer durumda} \end{cases}$

$x_{idt} = \begin{cases} 1 & i.\text{sınav } d. \text{ dersliğe } t.\text{zaman diliminde atanırsa} \\ 0 & \text{diğer durumda} \end{cases}$

#### *Sapma değişkenleri*

$dtp_{cl,g}$ : Toplam sınav süresi hedefi olarak belirlenen üst sınırdan (A) artı yönde sapma

$dtm_{cl,g}$ : Toplam sınav süresi hedefi olarak belirlenen üst sınırdan (A) eksi yönde sapma

$dnp_{cl,g}$ : Düzenli bir öğrencinin bir günde gireceği toplam sınav sayısı hedefi olarak belirlenen 2 sınavdan artı yönde sapma

$dnm_{cl,g}$ : Düzenli bir öğrencinin bir günde gireceği toplam sınav sayısı hedefi olarak belirlenen 2 sınavdan eksi yönde sapma

$dirp_{ijt}$ :  $t$  anında düzensiz öğrencilerin sınav çakışmasının olabildiğince engellenmesinde kullanılan artı yönde sapma

$dirm_{ijt}$ :  $t$  anında düzensiz öğrencilerin sınav çakışmasının olabildiğince engellenmesinde kullanılan eksi yönde sapma

$dxp$ : Sınav haftası boyunca kullanılan toplam derslik sayısı hedefi olarak belirlenen üst sınırdan ( $B$ ) artı yönde sapma.

$dxm$ : Sınav haftası boyunca kullanılan toplam derslik sayısı hedefi olarak belirlenen üst sınırdan ( $B$ ) eksi yönde sapma.

Problem, matematiksel olarak izleyen şekilde modellenmiştir:

$$\sum_t y_{it} = 1 \quad \forall i \quad (1)$$

$$\sum_d x_{idt} \geq y_{it} \quad \forall (i, t) \quad (2)$$

$$\sum_d x_{idt} \leq M * y_{it} \quad \forall i \quad (3)$$

$$\sum_i y_{it} \leq 1 \quad \forall (t, cl), (cl, i) \in CC \quad (4)$$

$$\sum_d x_{idt} c_d \leq \alpha_i y_{it} \quad \forall (i, t) \quad (5)$$

$$(\sum_i \sum_t y_{it}) - dnp_{cl,g} + dnm_{cl,g} = 2 \quad \forall (cl, i) \in CC, \forall (g, t) \in DZ \quad (6)$$

$$(\sum_i \sum_t y_{it} * s_i) - dtp_{cl,g} + dtm_{cl,g} = A \quad \forall (cl, i) \in CC, \forall (g, t) \in DZ \quad (7)$$

$$y_{it} + y_{jt} - dirp_{ijt} + dirm_{ijt} = 1 \quad \forall t, \forall (i, j) \in OG, i \neq j \quad (8)$$

$$\sum_i \sum_d \sum_t x_{idt} - dxp + dxm = B \quad (9)$$

$$y_{it} \in \{0,1\} \quad \forall (i, t) \quad (10)$$

$$x_{idt} \in \{0,1\} \quad \forall (i, d, t) \quad (11)$$

$$dtp_{cl,g}, dtm_{cl,g} \geq 0 \quad \forall (cl, g), (cl, i) \in CC, (g, t) \in DZ \quad (12)$$

$$dnp_{cl,g}, dnm_{cl,g} \geq 0 \quad \forall (cl, g), (cl, i) \in CC, (g, t) \in DZ \quad (13)$$

$$dirp_{ijt}, dirm_{ijt} \geq 0 \quad \forall t, \forall (i, j) \in OG, i \neq j \quad (14)$$

$$d xp, dxm \geq 0 \quad (15)$$

kısıtları altında

$$\text{enk} \left\{ \sum_{cl} \sum_g \left( \frac{dtp_{cl,g}}{4} + \frac{dnp_{cl,g}}{480} \right) + \left( \sum_i \sum_j \sum_t \frac{dirp_{ij,t}}{3} \right) + \frac{d xp}{264} \right\} \quad (16)$$

Geliştirilen matematiksel model, Eskişehir Osmangazi Üniversitesi Endüstri Mühendisliği Bölümü 2005-2006 Güz Dönemi 2. arasınav parametreleri kullanılarak çözdürülmüştür.

Kısıt kümesi (1) her sınavın bir zaman dilimine atanmasını sağlamaktadır. (2) ve (3) kısıt kümeleri, bir sınav bir veya birden fazla dersliğe bir  $t$  anında atandığında ilgili sınav-zaman dilimi eşleşmesini sağlayan kısıtlardır. Herhangi sınav-zaman dilimi  $(i,t)$  ikilisi bir dersliğe atanırsa, kısıt kümesi (2)  $y_{it}$ ' yi "1" değerini almaya zorlar.  $(i,t)$  ikilisinin bir dersliğe atanmaması durumunda ise (2) ve (3) kısıt kümelerindeki eşitsizliklerin sol tarafları sıfır değerini alır. Böylece kısıt kümesi (3)  $y_{it}$ ' nin "0" değerini almasını sağlar.

Kısıt kümesi (4) bir sınıfın herhangi bir zaman diliminde standart derslerinden en fazla birinin sınavının olmasını garanti eder. Derslik kapasitelerinin aşılması da kısıt kümesi (5) ile sağlanır.

Birinci hedef kısıt kümesi (6) ile düzenli bir öğrencinin bir günde ikiden fazla sınava mümkün olduğunca girmemesi, ikinci hedef kısıt kümesi (7) ile düzenli bir öğrencinin bir günde girdiği sınavların toplam süresinin 240 dakikayı ( $A$  dakika) mümkün olduğunca aşmaması, üçüncü hedef kısıt kümesi (8) ile herhangi bir anda düzensiz öğrencilerin sınav çakışmasının olabildiğince engellenmesi ve dördüncü hedef kısıt kümesi (9) ile de sınav haftası boyunca kullanılan toplam derslik sayısının 60'ı ( $B$  adet derslik) olabildiğince aşmaması sağlanmaktadır.

Amaç, hepsi eşit öneme sahip hedeflerden pozitif yöndeki sapmaların toplamının en küçüklenmesidir (16). Her hedefe karşı gelen sapma değişkeni süre ve adet gibi farklı ölçekte olduğundan amaç fonksiyonunda normalleştirme işlemi yapılmıştır.

$$enk \left\{ \sum_{cl} \sum_g \left( \frac{dtpls_{cl,g}}{4} + \frac{dnpls_{cl,g}}{480} \right) + \left( \sum_i \sum_j \sum_t \frac{dirpls_{ijt}}{3} \right) + \frac{dxpls}{264} \right\} \quad (16)$$

Ele alınan sistemde bir günde bir öğrencinin girebileceği sınav sayısı en fazla 4 olabileceğinden birinci hedef için yazılan  $dtp_{cl,g}$  değişkeni normalleştirilerken 4'e bölünmüştür.

Tüm sınav bloklarında sınav olması ve her sınavın 120 dakika olması durumunda bir öğrencinin bir günde girebileceği toplam sınav süresi teorik olarak en fazla 480 dakika olabilir. Dolayısıyla, ikinci hedef için amaç fonksiyonuna yansıtılan  $dnp_{cl,g}$  değişkeni 480'e bölünerek normalleştirilmiştir.

Örnek problem için bulunduğu sınıf dışındaki diğer sınıflardan da ders alan düzensiz bir öğrencinin en fazla 3 sınavı çakışabileceğinden, üçüncü hedef için amaca yansıtılan  $dirp_{ijt}$  değişkeni 3'e bölünerek normalleştirilmiştir.

Sınavların yapılabileceği 12 derslik ve pazartesi-cuma günleri için günde dört oturum, cumartesi günü ise iki oturum bulunduğundan toplam 22 oturum bulunmaktadır. Bir sınav döneminde kullanılacak en fazla derslik sayısı ise  $12 \cdot 22 = 264$  olacaktır. Bu sebeple, son hedef için yazılan  $dxp$  değişkeni 264'e bölünerek normalleştirilmiştir.

Mevcut durumda, farklı kapasitelere sahip 12 derslik kullanılarak, 64 dersin sınavı 6 gün boyunca toplam 22 zaman diliminde yapılmaktadır. İlgili döneme ait mevcut sınav-derslik-zaman çizelgesi EK 4'te verilmiştir. Çizelge 4.2'de 2005-2006 Güz Dönemi 2. Arasınavı haftasında çizelgelenen düzenli öğrencilere ait standart dersler kümesi verilmiştir. Bazı derslerin öğrenci sayısına bağlı olarak birden fazla grubu (örn: Benzetim A ve Benzetim B) ve ayrıca bazı derslerin hem birinci hem de ikinci öğretimde açılmış grupları bulunabilir. Bu grupların sınavları aynı zaman diliminde yapıldığından modelde tek bir ders olarak ele alındığından (örn: Benzetim) ilgili dönem

için toplam çizelgelenen sınav sayısı 27 olmuştur.

**Çizelge 4.2** Sınıf temelinde standart dersler kümesi

Sınıf	<i>i (ders no)</i>
1	1, 2, 3, 4, 5, 6, 7
2	8, 9, 10, 11, 12
3	13, 14, 15, 16,17, 18
4	19, 20, 21, 22, 23, 24, 25, 26, 27

GAMS CPLEX çözücüsü ile 13 saniyede tüm esnek kısıtlar sağlanarak en iyi çözüm elde edilmiştir. Elde edilen sınav-derslik-zaman çizelgesi Çizelge 4.3'te verilmiştir. Elde edilen çözüme karşı gelen atamalar ve bölümde yapılmış mevcut atamalar karşılaştırıldığında izleyen sonuçlara ulaşılmıştır:

- Mevcut durumda toplam **80** derslik kullanılırken, önerilen durumda **60** derslik yeterli olmaktadır.
- Özellikle **5** derslikte yapılan sınavlar için, **3** dersliğin yeterli olduğu görülmüştür. Örneğin Mühendislik Mekaniği sınavı mevcut durumda End 1,2,6, ve 7 derslikleri ile FL dersliğine atanmışken önerilen durumda bu sınav sadece 3 dersliğe atanmıştır.
- *Diferansiyel Denklemler (I. Öğr., II. Öğr.)*, *Differential Equations*, *Benzetim (I.Öğr., II.Öğr.)* ile *Teknik İngilizce V (I.Öğr., II.Öğr.)* derslerinin sınavları mevcut durumda 4'er dersliğe atanmış iken, önerilen durumda bu sınavlar 2 dersliğe atanmıştır. Bir diğer 4 derslikte yapılan *İş Etüdü (I. Öğr. A, II. Öğr. A, I. Öğr. B)* sınavı da 3 dersliğe atanmıştır.
- Ortak öğrencisi olan 3. ve 4. sınıf derslerinden Yöneylem Araştırması (I. Öğr., II. Öğr.), Operations Research, Üretim Planlama (I. Öğr., II. Öğr.), Mühendislik Ekonomisi (I. Öğr., II. Öğr.) ve ESKA'nın aynı zaman dilimine atanması

modelde engellenmiştir. Böylece, herhangi bir anda düzensiz öğrencilerin sınav çakışmasının olabildiğince engellenmesi esnek kısıtı sağlanmıştır.

- Mevcut durumda toplam 22 zaman dilimine sınav çizelgelemesi yapılırken, önerilen durumda kullanılan zaman dilimi sayısı 18 olmuştur.

**Çizelge 4.3** Geliştirilen model ile elde edilen sınav-derslik-zaman çizelgesi

Sınav	Derslik	Zaman dilimi
Mühendislik Malzemeleri (I. Öğr.) & Mühendislik Malzemeleri (II.Öğr.)	11, 12	2
TBTK (I. Öğr.) & TBTK (II. Öğr.)	5, 12	4
Teknik İngilizce V (I. Öğr.) & Teknik İngilizce V (II. Öğr.)	10, 12	4
Üretim Yöntemleri (I. Öğr.) & Üretim Yöntemleri (II. Öğr.)	11, 12	5
NKV (I. Öğr. + II. Öğr.)	12	6
Benzetim (I. Öğr.) & Benzetim (II. Öğr.)	11, 12	7
İstatistik II ( I. + II. Öğr.A) & İstatistik II ( I. + II. Öğr.B)	7, 9, 12	8
Fizik I (I. Öğr.) & Fizik I (II. Öğr.) & Physics I	6, 7, 12	8
Diferansiyel Denk. (I. Öğr.) & Diferansiyel Denk. (II. Öğr.) & Differential Equ.	11, 12	8
Olasılık (I. Öğr.) & Olasılık (II. Öğr.) & Probability (I. Öğr.)	8, 11, 12	9
İnsan Kaynakları Yön.	4	10
Tesis Planlaması (I. Öğr.) & Tesis Planlaması (II. Öğr.)	11, 12	12
ESKA	1	13
Matematik I (I. Öğr.) & Matematik I (II. Öğr.) & Calculus I	6, 11, 12	13
Kimya (I. Öğr.) & Kimya (II. Öğr.) & Chemistry	10, 11, 12	15
Üretim Planlama (I. Öğr.) & Üretim Planlama (II. Öğr.)	11, 12	16
İngilizce I (I. Öğr. + II. Öğr.)	11	17
KKİT	1	17
Mühendislik Ekonomisi (I. Öğr.) & Mühendislik Ekonomisi (II. Öğr.)	5, 11, 12	17
Pazarlama Yön. (I. Öğr.) & Pazarlama Yön. (II. Öğr.)	11, 12	18
Teknik İngilizce III (I. Öğr.) & Teknik İngilizce III (II. Öğr.)	11, 12	19
Mühendislik Mekaniği (I.Öğr. A) & Mühendislik Mekaniği (II.Öğr. A) & Mühendislik Mekaniği (I.Öğr. B)	6, 11, 12	19
Teknik Resim (I. Öğr.) & Teknik Resim (II. Öğr.)	11, 12	19
İş Etüdü(I.Öğr. A) & İş Etüdü(II.Öğr. A) & İş Etüdü(I.Öğr. B)	7, 11, 12	21
Bilgisayar Programlama II (I. Öğr.) & Bilgisayar Programlama II (II. Öğr.)	6, 7, 11, 12	23
Türk Dili I (I. Öğr.) & Türk Dili I (II. Öğr.)	11, 12	23
Yöneylem Araş. I (I. Öğr.) & Yöneylem Araş. I (II. Öğr.) & Operations Research I	4, 11, 12	24

Önerilen model ile elde edilen sonuçlara göre, kullanılan kaynak sayısında azalma olmaktadır. Kullanılan derslik sayısının azalması ile sınavlara atanacak gözetmen sayısında da azalma olacaktır. Ayrıca, düzensiz öğrencilerin sınav çakışmalarının çizelge oluşturulduktan sonra önlenmeye çalışılması ve/veya bu öğrenciler için sınav haftası sonunda tekrar yeni sınavların yapılması gibi oluşabilecek ek iş yükü de bu sayede olabildiğince engellenmiştir.

Yukarıda tanımlanan problem, dolaylı olarak öğrenci boyutunu da dikkate aldığından bir öğrencinin aldığı derslerin sınavlarının aynı zaman dilimine atanmaması –düzenli öğrenci- veya olabildiğince az atanması –düzensiz öğrenci- sınav çizelgeleme problemi için oldukça genel bir yapıdadır. Merkezi sistemle gerçekleştirilen KPSS, ÜDS vb. sınavların çizelgelenmesi bu kapsamda, zaman diliminin bilindiği duruma karşılık gelerek sınav-derslik atama probleminin çözümüne dönüşmektedir. Burada tanımlanan hedef kısıtları bu tür problemler için anlamlı olmayacaktır. Öte yandan, bir öğrencinin atanacağı dersliğin belirttiği adrese olabildiğince yakınlığı gibi yeni hedefler söz konusu olup önerilen modelin benzer problemlere uyarlanması söz konusu olabilir.

Sınav-derslik-zaman çizelgeleme problemi için kurulan matematiksel modelin boyut analizi Çizelge 4.4'te verilmiştir. Bir bölüm için ele alınan örnek problemde, 64 ders için ders kümeleri belirlenmiş ve böylece 27 dersin sınavının çizelgelendiği ( $i=27$ ,  $d=12$ ,  $g=6$ ,  $cl=4$ ,  $t=22$ ) bu modelin toplam kısıt sayısı 1489, 7722'si 0-1 tamsayı değişken olmak üzere toplam değişken sayısı da 8040 olarak belirlenmiştir.

Her bölümde çizelgelenecek 27 sınav ve 12 derslik olması varsayımıyla, sınav-derslik-zaman çizelgelemesinin on bölüm içeren bir fakülte temelinde yapılması durumunda ( $i=270$ ,  $d=120$ ,  $g=6$ ,  $cl=4$ ,  $t=22$ ), modelin kısıt sayısı 13657 toplam değişken sayısı da 721038 olacaktır. Bölüm temelinde yapılan çizelgelemenin fakülte temelinde yapılması durumunda modelin boyutu oldukça büyümektedir. Çizelgelemenin tüm üniversite kapsamında yapılması durumunda ise sınav ve derslik sayılarının artmasıyla model boyutunun çok daha fazla artacağı görülmektedir.

Geliştirilen modelin daha büyük boyutlu problemlere uygulanabilirliği, tüm

illerde aynı zaman diliminde gerçekleştirilen Açıköğretim Fakültesi sınavları örneği için irdelenebilir. Sınav zamanları belirli olan ve bir sınav-derslik atama problemi olan AÖF örneğinde her sınav oturumunda birden fazla dersin sınavı gerçekleştirilmektedir. Çalışmada ele alınan örnek problemde olduğu gibi bu problemde de ders kümeleri oluşturulmakta ve ortalama olarak 3 dersin sınavı bir sınav oturumunda gerçekleştirilmektedir. 282 dersin sınavının çizelgelendiği 2010 final dönemi için yaklaşık 94 sınav kümesi çizelgelenmiştir. Bir milyonun üzerinde öğrencinin girdiği sınavların çizelgelenmesi il temelinde yapılmaktadır. 282 dersin ildeki hangi okul ve dersliklere atanacağını belirlemek problemde öğrenci sayısının çok fazla olması nedeniyle, atamaların yapılacağı okul, dolayısıyla derslik sayısı da fazla olacağından problem boyutu oldukça artacaktır. Örneğin, 10.000 öğrenci için ortalama 25'er kişilik derslikler kullanılması durumunda 400 dersliğe atama yapılacaktır. Öte yandan, problemde zaman dilimi belirli olduğundan, zaman dilimine ilişkin karar değişkeni ve kısıtlar modelde yer almayacaktır. Geliştirilen modelin AÖF problemine uygulanabilirliği tüm ülke temelinde boyut nedeniyle güç olabilir. Ancak, sınav-derslik atamalarının bölge ya da mahalle temelinde yapılması geliştirilen model ile uygun olacaktır.

**Çizelge 4.4** Sınav-derslik-zaman çizelgeleme problemi için geliştirilen matematiksel model için boyut analizi

Kısıt no	İndisler	Toplam kısıt sayısı
1	$i$	$i$
2	$i, t$	$i \times t$
3	$i$	$i$
4	$t, cl$	$cl \times t$
5	$i, t$	$i \times t$
6	$cl, g$	$cl \times g$
7	$cl, g$	$cl \times g$
8	$t, i$	$i^* \times t$
9	-	1
Değişken	İndisler	Toplam değişken sayısı
$x_{idt}$	$i, d, t$	$i \times d \times t$

$y_{it}$	$i, t$	$i \times t$
$dtp_{cl,g}$	$cl, g$	$cl \times g$
$dtm_{cl,g}$	$cl, g$	$cl \times g$
$dnp_{cl,g}$	$cl, g$	$cl \times g$
$dnm_{cl,g}$	$cl, g$	$cl \times g$
$dirp_{ijt}$	$i, t$	$i^{\diamond} \times t$
$dirm_{ijt}$	$i, t$	$i^{\diamond} \times t$
$dxp$	-	1
$dxm$	-	1
Toplam kısıt sayısı		$2i + 2it + (2 \times cl \times g) + (cl \times t) + i^{\diamond}t + 1$
Toplam deęişken sayısı		$it + idt + (4 \times cl \times g) + 2 + 2 i^{\diamond} t$

#### 4.2 Gözetmen-Sınav Atama Problemi

Bu bölümde, Bölüm 2.2.2’de belirtilen gözetmen-sınav atama problemi çok amaçlı olarak ele alınmış ve çözümünü için karma tamsayılı bir matematiksel model geliştirilmiştir.

Literatürde gözetmen atama problemini dolaylı olarak dikkate alan sınav zaman çizelgeleme çalışmaları bulunmaktadır (Reis and Oliveira, 2000). Bu çalışmalarda doğrudan gözetmen ataması yapılmamış, bunun yerine ilgili parametreler problemin çözüm sürecine dahil edilmiştir. Örneğin Cowling vd. (2002), öğrenci ve gözetmenlerin mevcut sınav ve gözetmenlik çizelgeleri hakkında düşüncelerini ve yeni bir sınav zaman çizelgeleme yazılımından beklentilerini öğrenmek için bir anket düzenlemiştir. Anket sonuçlarına göre, gözetmenlerin mevcut çizelgelerinden memnun olmadığı ve gözetmenlik yüklerinin dengeli dağılımı için çalışmaların yapılması yönünde isteklerinin olduğu belirlenmiştir.

Gözetmen-sınav atama probleminin elle yapılan çözümünü de diğer zaman

---

$\diamond$  Çakışması mümkün olduğunca engellenmek istenen ders sayısı,  $i \neq j, \forall (i, j) \in OG$

çizelgeleme problemlerinin çözümünde olduğu gibi oldukça zaman alıcıdır. Üstelik elde edilen çözüm çoğu zaman gözetmenler açısından tatmin edici olmamaktadır. Atama sonuçlarının memnun edici düzeyde olması için, gözetmenlerin gün ve zaman dilimlerine ilişkin tercihlerinin alınması durumunda ise problemin zorluğu ve buna bağlı olarak çözüm için harcanan işgücü artmaktadır. Bu nedenlerle, diğer eğitimsel zaman çizelgeleme problemlerinde olduğu gibi gözetmen-sınav atama problemi için de otomatikleştirilmiş bir sistemin gerekliliği ortaya çıkmaktadır.

Eniyileme teknikleri ve Internet'in birlikte kullanılması ile kullanıcılar bir çözüm sistemini web arayüzleri kullanarak istediği zaman istediği yerden çalıştırabilir (Eom ve Kim, 2005). Ancak Schaerf (1999) çalışmasında birçok araştırmacının, zaman çizelgeleme probleminin iki nedenden dolayı tam olarak otomatikleştirilemeyeceği konusunda hem fikir olduğunu belirtmiştir. İlk neden, otomatikleştirilmiş bir sistemde bir çizelgenin diğerinden daha iyi olduğunun belirlenememesidir. Diğer yandan, çözüm arama uzayı genellikle çok büyük olduğundan kullanıcılar sistemin arama yönünü yönlendirilebilir. Otomatikleştirilmiş bir sistem çakışmaları önleyebilir ancak kullanıcı istek ve tercihlerini, uygun olmayan durumlarını vb.ni dikkate alamayabilir. Tüm bu nedenlerden dolayı birçok sistem, kullanıcının, en azından son çıktıyı manüel olarak düzenlemesine izin vermektedir.

Bölüm 4.2.1'de gözetmen atama problemi için gözetmenlerin tercihlerini dikkate alacak ve iş yüklerini dengeleyecek karma tamsayılı, üç amaçlı bir matematiksel model geliştirilmiştir. Kullanıcıların kendi arayüzlerini kullanarak kısıt ve tercihlerini girdiği ve bu bilgilerin matematiksel modele aktarıldığı web tabanlı sistem 5. Bölüm'de ayrıntılı olarak verilmiştir. Ayrıca, atama problemini daha gerçekçi bir şekilde ele alabilmek için, sınav sistemine ait parametrelerin tahmini için iki farklı çok amaçlı model sırasıyla bölüm 4.2.2 ve 4.2.3'te verilmiştir.

#### 4.2.1 Çok amaçlı gözetmen-sınav atama problemi için matematiksel model ve çözümü

Bir sınav sürecinde gözetmenlere eşit sayıda sınav ataması yapmak, tatmin edici bir atama elde edebilmek için yeterli değildir. Sınavların özelliklerine de bağlı olarak yükleri dengeleyici bir atamanın yapılması gereklidir. Bu gerekliliklerden yola çıkılarak, hem kullanıcı tercihlerini olabildiğince karşılamaya çalışan hem de gözetmenlerin yüklerini dengeleyen bir sistem geliştirilmiştir (Öztürk vd, 2010). Sistemin bu üstünlükleri ile Schaerf (1999) tarafından otomatikleştirilmiş sistemler için belirtilen, sonuç üzerinde elle düzeltmeler gibi sakıncaların olabildiğince üstesinden gelinmektedir.

Hangi sınavın hangi zaman dilimi ve derslikte yapılacağı bilindiği varsayımı altında, gözetmenlerin tercihlerini dikkate alan ve iş yüklerini dengeleyen karma tamsayı, üç amaçlı matematiksel modele ilişkin değişkenler kümesi, parametreler ve karar değişkenleri aşağıda verilmiştir.

##### *İndis kümeleri*

Gözetmen indisleri kümesi:  $I = \{1, \dots, m\}$

Sınav indisleri kümesi:  $J = \{1, \dots, n\}$

Zaman dilimleri kümesi:  $T = \{1, \dots, k\}$

Gözetmenler için uygun bulunmayan zaman

dilimleri kümesi:  $U = \{(i, t) | \exists i \in I, \exists t \in T\}$

Gözetmenler için sınava ön atamalar kümesi:  $V = \{(i, j) | \exists i \in I, \exists j \in J\}$

##### *Parametreler*

$g_j$ :  $j$  inci sınav için gerekli gözetmen sayısı

$a_j$ :  $j$  inci sınavın ağırlığı

$s_{jt}$ :  $j$  inci sınav  $t$  inci zaman dilimine çizelgelendiye 1, diğere durumda 0

$m_{it}$ :  $i$  inci gözetmen  $t$  inci zaman dilimine atanamıyorsa 1, diğere durumda 0,  $(i, t) \in U$

$p_{ij}$ :  $i$  inci gözetmenin  $j$  inci sınava ön ataması yapıldıysa 1, diğere durumda 0,  
 $(i, j) \in V$

$c_{it}$ :  $i$  inci gözetmeni  $t$  inci zaman dilimine atama maliyeti

$w_i$ :  $i$  inci gözetmenin yük oranı

*Karar değışkenleri*

$$y_{ij} = \begin{cases} 1 & i. \text{gözetmen } j. \text{sınava atanırsa} \\ 0 & d. d. \end{cases}$$

$$r_1, r_2 \geq 0$$

$$r_1 = \text{enk}\{\text{enb}_i(\sum_j y_{ij} a_j)\}$$

$$r_2 = \text{enk}\{\text{enb}_i(\sum_j y_{ij} s_{jt})\}, \exists(i, t)$$

Pozitif karar değışkenleri  $r_1$  ve  $r_2$ , modelde ileride açıklanacak olan *enk-enb* yapısının doğrusallaştırılmasında kullanılmaktadır. Belirlenen karar değışkenleri ve parametreler doğrultusunda gözetmen-sınav atama problemi izleyen şekilde modellenmiştir.

$$\sum_j y_{ij} s_{jt} \leq 1 \quad \forall(i, t) \quad (1)$$

$$\sum_i y_{ij} = g_j \quad \forall j \quad (2)$$

$$y_{ij} = p_{ij} \quad \forall(i, j) \in V \quad (3)$$

$$\sum_j y_{ij} s_{jt} = m_{it} - 1 \quad \forall(i, t) \in U \quad (4)$$

$$\sum_j y_{ij} a_j \leq w_i r_1 \quad \forall(i, t) \quad (5)$$

$$\sum_j y_{ij} s_{jt} \leq r_2 \quad \forall i \in I, \exists t \in T \quad (6)$$

$$y_{ij} \in \{0, 1\} \quad (7)$$

$$r_1, r_2 \geq 0 \quad (8)$$

*kısıtları altında*

$$\text{enk } \sum_i \sum_t \sum_j c_{it} y_{ij} s_{ij} + r_1 + r_2 \quad (9)$$

Amaç, gözetmenlerin istek ve tercihlerini dikkate alarak sınav yüklerinin dengeli bir şekilde dağıtılmasıdır. (1) nolu kısıt kümesi bir gözetmenin  $t$  zaman diliminde en fazla bir sınava atanmasını, (2) nolu kısıt kümesi de her sınava gerekli sayıda gözetmenin atanmasını sağlamaktadır. Bazı sınavların gözetmenliği için ön atamalar (gözetmenliğin araştırma görevlileri tarafından yapıldığı sistemlerde, bir dersin sınavına, o derse destek veren araştırma görevlisinin atanması gibi) olabilmektedir. Bu sınavlar için gerekli olan ön atamalar (3) nolu kısıt kümesi ile yapılmaktadır. Gözetmenlerin bazı özel durumlarda sınavlara giremeyeceği düşünülmüş, (4) nolu kısıt kümesi ile bu durumun modele yansıtılması sağlanmıştır. Dördüncü kısıt kümesinde yer alan  $m_{it}$  parametresi  $i$ . gözetmenin  $t$  anında kesinlikle uygun olmadığı durumda 1 değerini almaktadır ( $(i,t) \in U$ ). “ $m_{it}-1$ ” ifadesi ile (4) nolu kısıt kümesinde ilgili gözetmenin istenmeyen  $t$ . zaman dilimine atanması izleyen şekilde engellenir:

$$\sum_j y_{ij} s_{jt} = 0, \quad (i,t) \in U$$

Kısıt kümesi sadece  $m_{it} = 1$ 'e karşı gelen  $(i, t)$  ikilileri için oluşacağından ( $(i, t) \in U$ ),  $m_{it}=0$  olup,  $\sum_j y_{ij} s_{jt} = 1 - 1 = 0$  durumu oluşacaktır.

Bölüm 5'te tanıtılan web tabanlı karar destek sistemi ile gözetmenler, kısıtlarını ve tercihlerini kendileri için hazırlanan arayüzler yardımıyla sisteme girerler. Çizelge 4.5, kısıt ve tercih girişinde kullanılan simgeleri göstermekte olup, son satırında yer alan simge,  $i$ . gözetmenin  $t$  anında kesinlikle gözetmenlik yapamayacağı bilgisine ( $m_{it} = 1$ ) dönüştürülmektedir. Diğer tercih parametreleri,  $c_{it}$ , ise doğrudan Internet üzerinden girildikleri şekilde (1 veya 2) modele eklenirler. Toplam tercih etmeme düzeyi en küçüklenecek maliyet yapısında olduğundan, “1” tercih etmiyorum, “2” daha çok tercih etmiyorum anlamındadır.

**Çizelge 4.5** Kısıt ve tercih girişinde kullanılan simgeler

Simge	Anlamı
	Bu zaman diliminde sınava girmeyi 1 düzeyinde tercih etmiyorum
	Bu zaman diliminde sınava girmeyi 2 düzeyinde tercih etmiyorum
	Bu zaman diliminde kesinlikle sınava giremem

Kurulan matematiksel model çok amaçlı bir yapıya sahiptir. Eş zamanlı olarak toplam atama maliyetinin ve gözetmen yükleri ile istenmeyen zaman dilimlerine yapılan atamaların en büyüklerinin en küçüklenmesi istenmektedir. Gözetmen yüklerinin ve istenmeyen zaman dilimlerine yapılan atamaların en büyüklerinin en küçüklenmesi ile ilgili amaçlar enk-enb şeklinde doğrusal olmayan birer yapıya sahiptir. Bu amaçların doğrusallaştırılması için aşağıda sırasıyla  $r_1$  ve  $r_2$  pozitif karar değişkenlerinden yararlanarak (5) ve (6) nolu kısıt kümeleri kullanılmıştır. Amaçların eşit öneme sahip olduğu varsayılmış ve doğrusal toplamları alınmıştır.

Gözetmen-sınav atama sistemi için yukarıda önerilen matematiksel model, Eskişehir Osmangazi Üniversitesi (ESOGÜ) Endüstri Mühendisliği Bölümü'nde iki farklı sınav dönemine uygulanmıştır. ESOGÜ'nün bir akademik yılı, güz ve bahar olmak üzere iki dönemden oluşmaktadır. Her dönemde iki ara ve bir final olmak üzere toplam üç sınav dönemi bulunmaktadır. Sınav dönemlerinde, sınavlar haftanın altı günü, yapılmakta ve 9, 11, 14 ve 16 olmak üzere dört seçenek saatte başlayabilmektedir. Bir sınavın süresi 90 ile 120 dakika arasında değişebilir. Bu nedenle, sınavların yapılabileceği zaman dilimleri Çizelge 4.6'da verildiği gibi dört blok olarak tanımlanmış ve bir günün belirli bir bloğuna sabit bir numara verilmiştir.

Sınav çizelgeleri kısmi olarak dekanlıklar tarafından yapılmaktadır. Tüm bölümlerde ortak olan bazı derslerin sınavlarının yer ve zaman dilimi atamaları

dekanlıklar tarafından, diğer derslerinkiler ise ilgili bölümlerde görevlendirilmiş kişiler tarafından yapılmaktadır. Bu diğer grup için, atamalar haftanın sınav atanmamış zaman dilimleri kullanılarak gerçekleştirilir. Bu bölümde, sınav-derslik-zaman dilimi atamasının önceden yapıldığı varsayılmaktadır.

**Çizelge 4.6 Zaman dilimleri**

Zaman dilimleri	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi
9:00 – 11:00	1	5	9	13	17	21
11:00 – 13:00	2	6	10	14	18	22
14:00 – 16:00	3	7	11	15	19	23
16:00 – 18:00	4	8	12	16	20	24

Tüm araştırma görevlilerinden gözetmenlik yapmak istemedikleri ve gözetmenlik yapamayacakları zaman dilimleri bilgisi alınmıştır. Genellikle hafta sonu yapılan sınav saatleri istenmeyen zaman dilimleri olarak karşımıza çıkmıştır. Çalışmada önerilen atama süreci, gözetmenlerin sınav yükleri arasındaki farklılığı olabildiğince ortadan kaldırmaya yöneliktir. Her gözetmenin hem istenmeyen zaman dilimlerindeki toplam görev sayısını en küçükleyerek hem de görev yüklerini kıdemlerini de dikkate alınarak dengeli bir şekilde dağıtan atamalar gerçekleştirilmektedir.

Önerilen gözetmen-sınav atama modeli, mevcut sistem parametreleri kullanılarak, 2004–2005 güz dönemi final ve 2004–2005 bahar dönemi birinci ara sınavları için uygulanmıştır. Uygulamanın yapıldığı Eskişehir Osmangazi Üniversitesi Endüstri Mühendisliği Bölümü'nde gözetmenlikler bölümde görevli araştırma görevlileri tarafından yapılmaktadır. Bu araştırma görevlileri de kendi içlerinde doktora aşamasında ya da tamamlamış olanlar deneyimli, diğerleri de yeni olacak şekilde sınıflandırılmışlardır. Deneyimli görevlilere diğer işleri için de vakit ayırmaları

açısından daha az yük yani yüksek önem atanmaktadır. Gözetmenlerin yük oranlarında ( $w_i$ ) bu durum göz önüne alınarak deneyimli olanlara “0,5” diğerlerine de “1” değeri verilmiştir. Final sınavı atamalarındaki dokuzuncu ve onuncu araştırma görevlileri ile vize sınavı atamalarındaki yedinci ve sekizinci gözetmenler doktoralarını bitirmiş olanlardır. Arasınav döneminde görevli dokuz gözetmene ait yük oranları Çizelge 4.7’de verilmiştir. Ayrıca, her sınava bölüm yönetimi tarafından bir ağırlık değeri ( $a_j$ ) 2, 3 veya 4 olarak atanmıştır.

Her iki problem için de GAMS CPLEX çözücüsü ile en iyi çözümler elde edilmiştir. Elde edilen çözüme karşı gelen atamalar ve bölümde yapılmış mevcut atamalar sırasıyla Çizelge 4.8 ve Çizelge 4.9’da verilmiştir.

**Çizelge 4.7** 2004–2005 yılı bahar dönemi birinci arasınav gözetmen yük oranları

<i>i. gözetmen</i>	$w_i$
1	1
2	1
3	1
4	1
5	1
6	1
7	0,5
8	0,5
9	1

Çizelge 4.8’deki sonuçlara göre, mevcut durumda deneyimli olan dokuzuncu ve onuncu gözetmenler eşit sınav yüklerine sahip değildir. Önerilen çözümde ise, her iki gözetmen de eşit sayıda sınava atanmış, ayrıca eşit sınav yüküne de sahiptir. Diğer atamalar da irdelenirse, mevcut durumda gözetmenlerin sınav sayıları ve yükleri arasında bir dengesizlik göze çarpmaktadır. Birinci gözetmen toplam 10 sınava atanmış ve sınav yükü 28’dir. Beşinci ve on üçüncü gözetmenler de 10’ar sınava atanmış ancak, sınav yükleri sırasıyla 31 ve 30 birimdir. Önerilen çözüm ile her iki sınıf gözetmen için eşit sınav yüklerine sahip olacakları en iyi atama elde edilmiştir.

Çizelge 4.8 2004-2005 Güz dönemi final sınavları gözetmen atamaları

Gözetmenler	Önerilen Çözüm																												Mevcut									
	1	2	3	4	5	6	7	8	9	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3	Sınav sayısı	Toplam yük	Sınav sayısı	Toplam yük				
1	0	0	0	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	1	11	29	10	28	
2	1	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	1	0	1	0	1	0	0	11	29	11	28
3	1	0	0	1	0	1	1	0	1	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	11	29	12	30	
4	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1	0	0	0	1	0	0	1	11	29	11	27		
5	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	1	0	1	1	0	0	0	11	29	10	31		
6	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	0	0	1	0	0	10	29	11	29		
7	1	0	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1	1	1	10	29	10	28	
8	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	1	0	0	1	10	29	10	30		
9	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	5	14	5	11	
10	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	5	14	6	16		
11	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0	0	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	11	29	10	28		
12	0	0	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	0	0	10	29	10	31	
13	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	10	29	10	30		

2004-2005 bahar dönemi birinci ara sınav mevcut gözetmen atama sonuçlarına göre ise, sınav yükleri açısından gözetmenler arasında yine dengeli bir dağılım olmadığı görülmektedir. Önerilen çözüm ile sınav yükleri daha dengeli bir şekilde dağıtılmıştır. Birinci gözetmenin yükünün diğerlerinden farklı olması, bu kişinin ilgili sınav dönemindeki zaman dilimi kısıtlarından kaynaklanmaktadır.

**Çizelge 4.9** 2004-2005 bahar dönemi I.arasınav sınav gözetmen atamaları

Gözetmenler	Önerilen Çözüm																												Mevcut				
	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	Sınav sayısı	Toplam yük	Sınav sayısı	Toplam yük
1	0	1	1	0	0	0	1	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0	0	1	1	0	0	0	12	32	14	35	
2	1	1	1	0	0	1	0	1	0	0	1	0	1	0	1	0	0	0	1	1	0	0	1	1	1	1	0	0	14	35	14	37	
3	1	1	1	0	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1	0	0	1	0	1	1	1	0	0	14	36	14	38	
4	1	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	1	1	1	14	36	13	35	
5	0	1	1	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	1	0	0	14	36	14	35	
6	0	0	0	1	0	1	0	1	1	1	0	1	0	0	1	0	0	0	0	1	1	0	1	0	1	1	1	1	14	36	12	32	
7	1	0	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	7	18	7	16	
8	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	7	18	7	18
9	0	1	0	0	1	0	0	1	1	0	1	0	1	1	1	0	0	0	1	1	0	0	0	1	0	0	1	0	1	13	36	14	37

### Boyut analizi

Gözetmen-sınav atama problemi için kurulan matematiksel modelin boyut analizi Çizelge 4.10’da verilmiştir. Örnek problemde, 9 gözetmen toplam 22 zaman diliminde yapılan 28 sınava atanmıştır.

Bir bölüm temelinde ele alınan örnek problem fakülte temelinde irdelendiğinde, zaman dilimi sayısı sabit kalırken, gözetmen ve sınav sayısı artacaktır. Örneğin 10 bölüm içeren bir fakülte her bölümde 9 gözetmen, 28 sınav eşit sayıda istek ve tercih kısıtlarının olması durumunda ( $i=90$ ,  $j=280$ ) modelin toplam kısıt sayısı 4760, değişken sayısı da 25202 olacaktır. Bir bölüm temelinde yapılan gözetmen atamalarının, fakülte temelinde yapılması durumunda model boyutu artmaktadır. Atamaların tüm üniversite kapsamında yapılması durumunda ise sınav ve gözetmen sayılarının artmasıyla model boyutunun çok daha fazla artacağı görülmektedir.

**Çizelge 4.10** Gözetmen-sınav atama problemi için geliştirilen matematiksel model için boyut analizi

Kısıt no	İndisler	Toplam kısıt sayısı
1	$i, t$	$i \times t$
2	$j$	$j$
3	$i, j$	$i \times j^8$
4	$i, t$	$i \times t^9$
5	$i, t$	$i \times t$
6	$i, t$	$i \times t^{10}$
Değişken	İndisler	Toplam değişken sayısı
$y_{ij}$	$i, j$	$i \times j$
$r_1$	-	1
$r_2$	-	1
Toplam kısıt sayısı		$2it + j + i \times j^8 + i \times t^9 + i \times t^{10}$
Toplam değişken sayısı		$ij + 2$

Yukarıda tanımlanan problem, yine merkezi sistemle gerçekleştirilen sınav sürecinde görev alacak gözetmen, salon başkanı vb. kişilerin sınav bina ve salonlarına atanması problemi için oldukça genel bir yapıdadır. Zaman diliminin bilindiği ve tek bir sınavın yapıldığı bu problem, gözetmen-derslik atama problemine dönüşmektedir. Gözetmen-sınav atama problemi için tanımlanan amaç fonksiyonları ve ön atama kısıtları ise bu tür problemler için anlamlı olmayacaktır. Öte yandan, bir gözetmenin atanacağı dersliğin belirttiği adrese olabildiğince yakınlığı gibi yeni hedefler söz konusu olup önerilen modelin benzer problemlere uyarlanması söz konusu olabilir.

Geliştirilen modelin daha büyük boyutlu problemlere uygulanabilirliği merkezi bir sınav için irdelendiğinde, bir ilde 2000 derslik kullanılması durumunda yaklaşık 4000 gözetmen atamasının yapılacağı görülmektedir. Bu örnek problem yapısı, il temelinde atamaların yapılmasına izin verdiğinden, geliştirilen matematiksel model ile her il için gözetmen-derslik atama problemi çözdürülebilir.

<sup>8</sup>  $\forall (i, j) \in V$

<sup>9</sup>  $\forall (i, t) \in U$

<sup>10</sup>  $\forall i, \exists t \in T$

#### 4.1.1 Problem parametrelerinin tahmini için bir AHP modeli

Bir önceki bölümde geliştirilen gözetmen-sınav atama modeli ile gerçek problemler için en iyi çözüm elde edilmiştir. Ancak, problem parametrelerinden olan sınav ağırlığı ( $a_j$ ) parametresi yönetim tarafından önceden belirlenmiş olan 2, 3 ve 4 değerlerini almaktadır. Bir karar probleminde sonucun parametrelere bağımlı olması, parametrelerin sağlıklı tahmin edilmesini gerektirir.

Çalışmanın bu bölümünde sınav ağırlıklarının daha gerçekçi ve sistematik bir şekilde bulunması ve ayrıca gözetmen gereksinimlerine ilişkin ağırlıkların elde edilmesi için bir Analitik Hiyerarşi Süreci (AHP) modeli geliştirilmiştir (Öztürk ve Sağır, 2008).

Gözetmen-sınav atama problemi, farklı eğitim kurumları için farklı kısıtları ve amaçları olan bir problemidir. Her kurum için genellikle ortak olan kısıtlar ise şunlardır:

- Bir gözetmen herhangi bir zaman diliminde sadece bir sınava atanabilir.
- Her sınava gerekli sayıda gözetmen atanmalıdır.

Bazı durumlarda ise dersin asistanının sınava gözetmen olarak atanması gibi ön koşullar istenebilmektedir.

Gözetmenler bir sınavın kurallara uygun gerçekleşmesini sağlamak üzere sınavlara atanırlar. Dolayısıyla, sınav süresi boyunca bir çaba harcarlar. Gözetmenleri sınavlara atama süreci yukarıda belirtilen temel nicel gerekliliklerin yanı sıra, bazı nitel faktörler ve tercihler gibi nitel faktörlerden de etkilenmektedir. Katılımcıların bu gibi beklenti ve tercihleri tüm sistemin başarısını etkilemektedir. Ancak özellikle nitel faktörlerin sisteme yansıtılması oldukça zordur. Gözetmen-sınav atama sisteminde bu faktör ve tercihler genellikle sınavın yapılacağı *gün, zaman dilimi, sınav tipi* vb.'ne bağlı olmaktadır. Gözetmenlerin ihtiyaçları karşılanacak şekilde eşit sınav yükünün dağıtılacağı sınav çizelgesinin parametre ağırlıklarının tahmin edilmesi yönetsel bir problemidir.

Saaty (1986, 1980) tarafından geliştirilen Analitik Hiyerarşi Süreci (AHP), çok ölçütlü karar verme problemleri için kullanılan bir tekniktir. Aynı zamanda AHP, bir seçim ve önceliklendirme aracı olarak da kullanılmaktadır. Çalışmanın bu kısmında gözetmen-sınav atama probleminin nitel ve nicel nitelikte olan parametre ve amaç fonksiyonları AHP yaklaşımı ile ağırlıklandırılmıştır. Burada geliştirilen AHP modeli, seçenekler olmaksızın sadece problemin önemli ölçütlerini ağırlıklandırmak amacıyla kullanılacak bir araç olarak geliştirilmiştir. Bu kapsamda örneğin sınav süresi gözetmen-sınav atama sisteminde önemli bir parametre olup, uzun olması ilgili sınavın daha az tercih edilmesi ve çizelgelemede daha çok darboğaz yaratması demektir. Bu yüzden ilgili sınav yük dengesinde daha ağırlıklı olarak değerlendirilmelidir.

AHP ile, ele alınan karar probleminin amacı belirlenir ve ardından bu doğrultuda amacı etkileyen ölçüt ve ölçütlere ait alt ölçütler bir hiyerarşi formunda hazırlanır.

Üçüncü adımda, ölçüt ve alt ölçütlerin kendi aralarındaki önem derecelerinin belirlenmesi için ikili karşılaştırma matrisleri oluşturulur. AHP ile bulunan ağırlıklar sayesinde, farklı birimlerle de ifade edilseler ölçütler, aynı birime dönüştürülmüş olurlar. Bu hesaplamalarda kullanılan birkaç yöntem vardır. Tüm yöntemler “özdeğer” kavramından yola çıkarlar. Özdeğer,  $|A - \lambda I| = 0$  yapan  $\lambda$  değeridir.

İkili karşılaştırmalar matrisi  $A$ 'nın genel gösterimi (1) nolu denklem ile gösterilmiştir.  $a_{mn}$  değeri,  $m$  ölçütünün  $n$  ölçüti üzerindeki göreceli değerini temsil etmektedir.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & & \vdots \\ a_{m1} & \cdots & & a_{mn} \end{bmatrix} \quad (1)$$

$w$  ise değeri bulunmak istenen ağırlık vektörüdür.  $A$  matrisinin sağdan ağırlık vektörü ile çarpımı sonucu  $Aw = nw$  (2) nolu denklem elde edilir. Bu sistemin çözümü ancak matris boyutu  $n$ 'nin  $A$  matrisinin bir özdeğeri olması halinde vardır. Matrisin

enbüyük özdeğeri  $\lambda_{\max}$ 'ın  $n$ 'e eşit olması durumunda ise matris tutarlıdır (Özdemir, 2004).

$$Aw = \begin{matrix} A_1 & & & A_n \\ \vdots & & & \\ A_n & & & \end{matrix} \begin{bmatrix} w_1/w_1 & \cdots & w_1/w_n \\ \vdots & \ddots & \vdots \\ w_n/w_1 & \cdots & w_n/w_n \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = n \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix} = nw \quad (2)$$

İkili karşılaştırmalar hiyerarşideki tüm seviyelerde yapılır. Karşılaştırmaların yapılmasında Saaty (1994) tarafından önerilen 1-9 ölçeği kullanılır. Bu ölçekte kullanılan değerler ve anlamları Çizelge 4.11''de verilmiştir. Tüm seviyelerde yapılan karşılaştırmaların tamamlanmasının ardından, her ölçütün ağırlığı ve son olarak hiyerarşik sentez ile de tüm seçeneklerin ağırlıkları elde edilir.

**Çizelge 4.11** Saaty ölçeği

Değer	Açıklama
1	Eşit önemli
3	Orta derecede önemli
5	Kuvvetli derecede önemli
7	Çok kuverli derecede önemli
9	Kesin önemli
2, 4, 6, 8	Ara değerler

Bu çalışmada gözetmen gereksinimleri ve sınav parametreleri ile ilgili iki ana grup ölçüt belirlenmiştir. Bu ölçütler açısından değerlendirilecek seçenek söz konusu değildir. Amaç, problem parametrelerinin sadece ağırlıklarını bulmaktır. Gözetmen gereksinimleri yüklerin dengeli atandığı bir sınav çizelgesi ve istenmeyen zaman dilimlerine olabildiğince dengeli atamalarla ilgilidir. İstenmeyen zaman dilimleri ise genellikle sınav günlerinin ilk zaman dilimleri ile hafta sonlarında yer alan zaman dilimleri olmaktadır. Probleme etki eden ve ağırlıkları araştırılan diğer parametreler de;

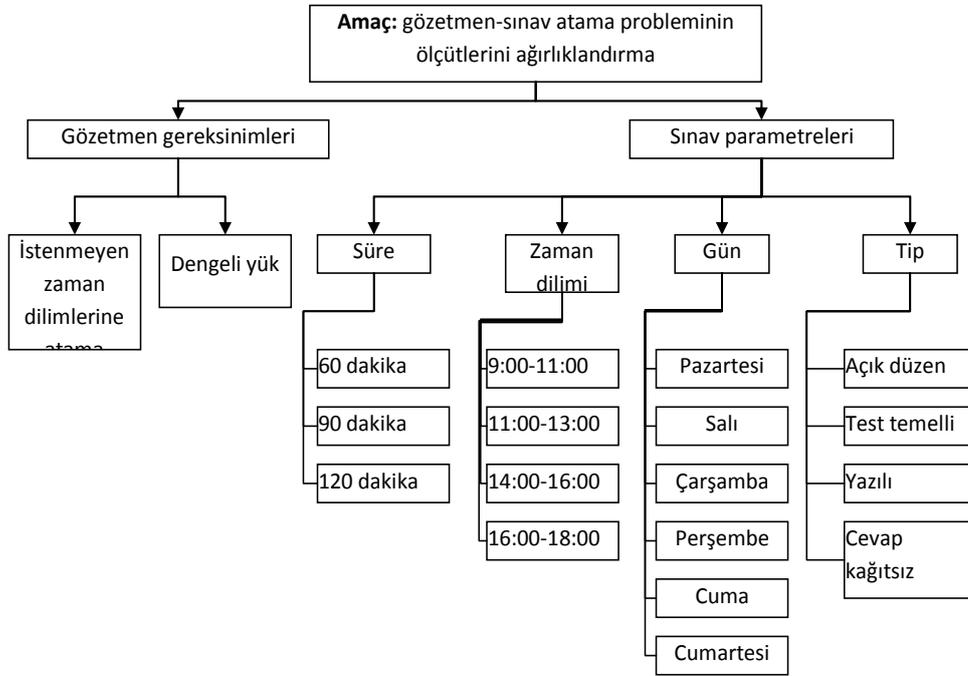
*sınav süresi*, sınavın atandığı *zaman dilimi*, *sınavın günü* ve *sınav tipi* olarak belirlenmiştir.

Sınav süresi 60, 90 ve 120 dakika olarak belirlenmiştir. Sınav süresi uzadıkça, gözetmenlik daha yorucu olup fazla dikkat gerektireceğinden gözetmenlerin memnuniyetsizlik düzeyleri artmaktadır. Bir günde dört sınav zaman dilimi bulunmaktadır. Dolayısıyla, zaman dilimi ölçüti de 9:00-11:00., 11:00-13:00, 14:00-16:00 ve 16:00-18:00 olarak dört alt ölçüte ayrılmıştır. Sınav dönemlerinde haftanın altı günü (Pazartesi, Salı, Çarşamba, Perşembe, Cuma ve Cumartesi) sınav yapılmaktadır ve bu altı gün sınav ölçütünün alt ölçütleridir. Son ölçüt *sınav tipi* ise açık düzen, yazılı, test ve cevap kağıtsız alt ölçütlerine sahiptir. Burada tasarım yönlü bir açıklama yapılmasında yarar vardır. AHP uygulamalarında normalde saat dilimlerinin saat ölçütünün, farklı günlerin ise gün ölçütünün alt ölçütleri gibi tanımlanmaları alışılmış bir durum değildir. Çünkü bunlar gün ve saat parametrelerinin alabileceği değerlerdir. Seçeneklerin olduğu bir problem olsaydı örneğin bir seçeneğin 9-11 alt ölçüti açısından veya “Salı” günü alt ölçüti açısından değerlendirilmesi bir anlam karışıklığı yaratabilirdi. Bu çalışmada ise durum bu değildir. Alt ölçütler birbirleriyle kıyaslanmış ve saat olarak 9-11 dilimi mi 14-16 dilimi mi gözetmenlik için daha zordur, veya Salı günü mü Cuma günü mü daha az tercih edilir sorguları sözkonusu olmuştur. Bu da problemin amacı ile uyumludur.

Gözetmenler, sınav boyunca dikkatli olmak zorundadırlar. Ancak, yukarıda belirtilen ölçütler gözetmenlerin dikkat dolayısıyla memnuniyetsizlik düzeylerini de etkilemektedir. Gözetmenler tercihleri dikkate alındığında ve sınav yükleri dengeli dağıtıldığında atamalardan memnun olmaktadır. Bu beklentileri karşılamak için Expert Choice yazılımında oluşturulan dört seviyeli hiyerarşi Şekil 4.1’deki gibi verilmiştir.

Hiyerarşinin oluşturulmasının ardından ikili karşılaştırmalar Eskişehir Osmangazi Üniversitesi ve Anadolu Üniversitesi’nde görevli on araştırma görevlisi tarafından yapılmıştır. Tutarlı karşılaştırmaların ardından, her bir ölçütün ağırlığı geometrik ortalamalar alınarak elde edilmiştir. AHP uygulamalarında herhangi bir görüş değil uzman görüşü önemlidir. Gözetmenlik sürecini tüm sıkıntılarıyla yaşayan araştırma

görevlileri bu süreçte uzman kabul edilmiştir.



Şekil 4.1 Dört seviyeli hiyerarşi

Ölçütlerin tercih edilme düzeyleri gözetmenlere atanan göreve bağlı olarak değişmektedir. Eğer bir sınav parametresi gözetmen açısından *istenmeyen* ya da *zor* ise, ağırlığı daha fazla olacaktır. Örnek bir ikili karşılaştırma sorusu, gözetmen gereksinimi ölçütünün iki alt ölçütü için “*eşit yük dağılımına sahip bir çizelgeye mi yoksa istenmeyen zaman dilimlerine atama yapılmamış bir çizelgeye mi sahip olmak daha az tercih edilir?*” olarak verilebilir. Bu soru için üç uzman, istenmeyen zaman dilimlerine atama yapılmamış bir çizelgenin daha az tercih edildiğini belirtmiştir.

Sınav parametreleri ana ölçütüne göre de bir karşılaştırma örneği verilecek olursa; uzun süreli bir sınavın ağırlığı hem süre yönünden gözetmenlerin tercihinin olumsuz etkilemesi hem de planlama açısından yönetimin çizelgeleme zorlukları ile karşılaşması sebebi ile kısa süreli sınavlara göre fazladır. Bu nedenle soru, 60 dakikalık mı 120 dakikalık sınav mı daha az tercih edilir olup, örneğin 120 dakikalık sınav 3 kat daha az

tercih edilir yanıtı 120 dakikalık sürenin daha fazla ağırlık almasına yol açacaktır. Daha önce de bahsedildiği üzere, sınav süresi uzadıkça gözetmenlerin memnuniyetsizlik düzeyi de artmaktadır. Bu durum da, kısa süreli sınavların daha çok tercih edileceği ve dolayısıyla daha az yüke sahip olacağı anlamına gelmektedir. İlk sekiz uzmana göre en tercih edilmeyen sınavlar 120 dakikalık sınavlardır. 90 ve 60 dakikalık sınavlar da sırasıyla ikinci ve üçüncü sırada tercih edilmeyen sınavlardır. Sadece son iki uzman 120 dakikalık sınavların en az tercih edilen sınavlar olduğunu belirtirken, 90 ve 60 dakikalık sınavlar arasında bir fark olmadığını belirtmiştir.

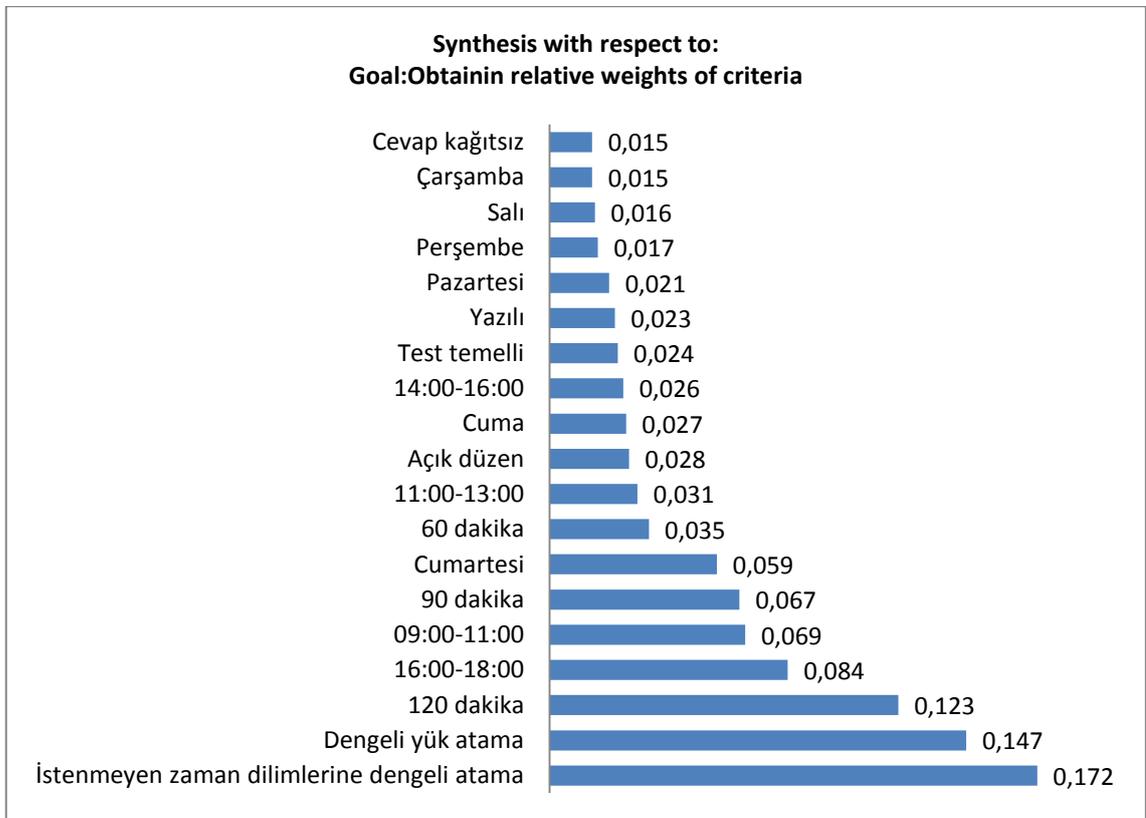
Sınav parametreleri için farklı uzman görüşleri ve nihai grup kararı Çizelge 4.12’de verilmiştir. Grup kararına göre sınav süresi sınav parametreleri arasında atama sürecini olumsuz yönde en çok etkileyen parametre olarak belirlenmiştir. Grup kararının yanı sıra, uzmanların bireysel kararları arasında da farklılıklar olduğu görülmektedir. Örneğin, 1. ve 2. uzman için gün parametresi, 6. ve 10. uzman için süre, 8. uzman için de zaman dilimi parametresi sınav sürecini olumsuz yönde en fazla etkileyen ölçüt olmuştur.

Yukarıda verilmiş olan karşılaştırma örneklerinden de anlaşılacağı üzere, istenmeyen bir sınav atamasına sebep olan herhangi bir ölçüt daha yüksek ağırlığa sahip olmaktadır. Tüm ikili karşılaştırmaların yapılmasının ardından elde edilen ölçüt ağırlıkları Şekil 4.2’de verilmiştir.

**Çizelge 4.12** Bazı uzmanlar ve grup kararına göre sınav parametrelerinin göreceli ağırlıkları

Sınav parametreleri	Bireysel Sonuçlar					Grup kararı
	Uzman 1	Uzman 2	Uzman 6	Uzman 8	Uzman 10	
Süre	0.052 (3)	0.200 (2)	0.222 (1)	0.117 (2)	0.201 (1)	0.226 (1)
Zaman Dilimi	0.073 (2)	0.154 (4)	0.183 (2)	0.234 (1)	0.167 (2)	0.210 (2)
Tip	0.031 (4)	0.164 (3)	0.148 (3)	0.078 (3)	0.105 (4)	0.090 (4)
Gün	0.094 (1)	0.220 (1)	0.183 (2)	0.117 (2)	0.134 (3)	0.155 (3)

Şekil 4.2’den de görüldüğü üzere, gözetmenlerin sınav yükünü en çok artıracak olan ölçüt, *120 dakikalık* sınavdır. Bu tahmin edilmesi zor bir durum değildir. Vurgulanmak istenen konu, bu gibi faktörlerin ağırlıklarının sistematik bir yolla bulunmasıdır. Sınav günü alt ölçütlerinden en az tercih edilen *Cumartesi* günü, sınav tipleri içinden ise *açık düzen* olanlardır. Yanısıra, cevap kağıtsız ve Çarşamba günü yapılan sınavlar sınav yükünü en az artıran sınavlar olmuştur.



**Şekil 4.2** Tüm ölçütlere ait ağırlıklar

Bölüm 4.2.1’de gerçek sisteme ait mevcut parametreler ile çözdürülen model, AHP modelinden elde edilen ağırlıklar ile tekrar çözdürülmüştür. Bir sınavın ağırlığını etkileyen süre, zaman dilimi, gün ve tip ölçütlerinin ağırlıkları toplanarak yeni sınav ağırlıkları ( $a_j$ ) elde edilmiştir. Çizelge 4.13’te 2004–2005 bahar dönemi birinci ara sınavları için ilgili sınavların mevcut ve önerilen yeni ağırlıkları kullanılarak bulunan

gerekli gözetmen sayıları ( $g_j$ ) verilmiştir. Elde edilen bu ağırlıklar kullanılarak her sınavın ağırlığı, dolayısıyla gözetmenlerin toplam sınav yükü hesaplanabilir. Örneğin, Pazartesi günü (0.021), saat 11:00'de (0.031), 120 dakikalık (0.123) ve yazılı (0.023) olan Yöneylem Araştırması I (A,B) sınavının ağırlığı 0.256 olacaktır. Benzer şekilde, mevcut durumda sadece 2, 3 ve 4 değerleri verilmiş olan sınavlara, sahip oldukları özelliklere göre yeni ağırlıklar atanmıştır. Öte yandan, mevcut duruma göre daha fazla ağırlığa sahip bazı sınavlar, önerilen durumda daha düşük ağırlık değerine sahip olmuştur. Örneğin Sosyal Seçmeli ve Mühendislik Mekaniği derslerinin sınavlarının ağırlıkları sırasıyla 2 ve 3 iken, yeni durumda Mühendislik Mekaniği dersi sınavı Sosyal Seçmeli dersi sınavına göre daha fazla ağırlığa sahip olmuştur.

AHP ile elde edilen yeni ağırlıklar ile her sınavın kendine ait ağırlığı hesaplanmış ve Bölüm 4.2.1'de geliştirilmiş olan matematiksel model yeniden çözdürüldüğünde elde edilen sonuç Çizelge 4.14'te verilmiştir. AHP ağırlıkları ile elde edilen yeni çözüme göre gözetmenler, aynı gruptaki diğer gözetmenlerle birlikte dengeli yüklere sahip olmuşlardır. Kıdemli gözetmenlerin toplam yükü 1.387 ile 1.391 olurken, diğer gözetmenlerin yükleri 1.9 ile 2.7 arasında değişmiştir.

Çizelge 4.13 AHP'den elde edilen yeni sınav ağırlıkları

Gün	Zaman dilimi	Sınav	$g_j$	$a_j$	yeni $a_j$	zaman dilimi	tip	süre	gün
12.12.2005 Pazartesi	09:00	Fizik I (A,B) <sup>11</sup>	2	2	0,180	0,069	0,023	0,067	0,021
	09:00	Physics I	1	2	0,180	0,069	0,023	0,067	0,021
	11:00	Yöneylem Araş. I (A,B)	2	3	0,198	0,031	0,023	0,123	0,021
	11:00	Operations Research I	1	3	0,198	0,031	0,023	0,123	0,021
	14:00	Mühendislik Malzemeleri (A,B)	2	2	0,105	0,026	0,023	0,035	0,021
	16:00	İnsan Kaynakları Yön.	2	2	0,187	0,084	0,015	0,067	0,021
	16:00	Pazarlama Yön. (A,B)	1	2	0,164	0,084	0,024	0,035	0,021
13.12.2005 Salı	09:00	Matematik I (A,B)	3	3	0,231	0,069	0,023	0,123	0,016
	09:00	Calculus I	1	3	0,231	0,069	0,023	0,123	0,016
	11:00	İngilizce I (A and B)	2	2	0,106	0,031	0,024	0,035	0,016
	14:00	Benzetim (A,B)	3	3	0,180	0,026	0,015	0,123	0,016
	16:00	Diferansiyel Denk. (A,B)	2	3	0,246	0,084	0,023	0,123	0,016
	16:00	Differential Equ.	1	3	0,246	0,084	0,023	0,123	0,016
14.12.2005 Çarşamba	09:00	Türk Dili I (A,B)	2	2	0,143	0,069	0,024	0,035	0,015
	11:00	Bilg. Prog. II (A,B)	3	3	0,128	0,031	0,015	0,067	0,015
	14:00	Teknik Resim (A,B)	2	3	0,187	0,026	0,023	0,123	0,015
	14:00	Mühendislik Ekonomisi (A,B)	2	3	0,131	0,026	0,023	0,067	0,015
	18:00	Üretim Planlama (A,B)	2	4	0,245	0,084	0,023	0,123	0,015
15.12.2005 Perşembe	09:00	Kimya (A, B)	2	2	0,176	0,069	0,023	0,067	0,017
	09:00	Chemistry	1	2	0,176	0,069	0,023	0,067	0,017
	09:00	KKİT	1	2	0,176	0,069	0,023	0,067	0,017
	09:00	ESKA	1	2	0,176	0,069	0,023	0,067	0,017
	09:00	NKV (A, B)	2	2	0,176	0,069	0,023	0,067	0,017
	11:00	Olasılık (A,B)	2	3	0,194	0,031	0,023	0,123	0,017
	11:00	Probability (A)	1	3	0,194	0,031	0,023	0,123	0,017
	14:00	Üretim Yöntemleri (A, B)	3	2	0,133	0,026	0,023	0,067	0,017
	16:00	Teknik İngilizce III (A, B)	2	2	0,151	0,084	0,015	0,035	0,017
16.12.2005 Cuma	09:00	TBTK (A,B)	1	2	0,146	0,069	0,015	0,035	0,027
	09:00	Teknik İngilizce V (A, B)	2	2	0,155	0,069	0,024	0,035	0,027
	11:00	Tesis Planlaması (A, B)	2	2	0,136	0,031	0,043	0,035	0,027
	14:00	İstatistik II ( A1, A2, B1, B2)	2	3	0,183	0,026	0,007	0,123	0,027
	16:00	Mühendislik Mek. (A1-2, B1)	2	2	0,189	0,084	0,043	0,035	0,027
17.12.2005 Cumartesi	11:00	Sosyal Seçmeli I	2	3	0,148	0,031	0,023	0,035	0,059
	14:00	İş Etüdü(A1-2, B1)	2	4	0,231	0,026	0,023	0,123	0,059

Farklı sınav ağırlıkları kullanıldığından, mevcut ağırlıklar ile elde edilen çözümden farklı bir atama sonucu elde etmek şartı olmalıdır. Sonuç olarak diyebiliriz ki, sistemde yer alan farklı uzmanların da görüşleri alınarak elde edilen ağırlıkların gözetmen-sınav atama probleminin parametrelerine yansıtılması ile

<sup>11</sup> Bir derse ait farklı gruplar

geliştirilmiş olan matematiksel model daha gerçekçi olarak ele alınmıştır.

**Çizelge 4.14** Mevcut ve AHP ağırlıkları kullanılarak elde edilen çözümler

Sınavlar	Ağırlıklar		Mevcut ağırlıklar kullanılarak elde edilen çözüm									AHP ağırlıkları kullanılarak elde edilen çözüm								
	Mevcut	Yeni	Gözetmenler									Gözetmenler								
			1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
Fizik I (I. Öğr.), Fizik I (II. Öğr.), Physics I	2	0,180		1	1	1			1	1			1	1	1			1	1	
Yöneylem Araş. I (I.-II Öğr.), OR	3	0,198	1	1	1		1					1			1	1	1	1	1	
Mühendislik Malzemeleri (I.-II Öğr.)	2	0,105	1	1	1		1								1				1	
İnsan Kaynakları Yön.	2	0,187							1										1	
Pazarlama Yön. (I.-II. Öğr.)	2	0,164				1	1								1	1	1			
Matematik I (I.-II. Öğr.)-Calculus	3	0,231		1	1	1		1	1	1				1	1	1	1	1	1	
İngilizce I (I. Öğr. + II. Öğr.)	2	0,106	1				1						1	1						
Benzetim (I. -II.Öğr.)	3	0,180	1	1		1		1					1	1	1					
Diferansiyel Denk. , Differential Equ.	3	0,246				1	1	1			1	1			1	1	1	1	1	
Türk Dili I	2	0,143			1	1		1	1					1	1	1	1			
Bilgisayar Programlama II	3	0,128	1	1	1	1	1						1					1		
Teknik Resim	3	0,187				1	1	1	1					1	1				1	
Mühendislik Ekonomisi	3	0,131	1	1	1									1	1	1	1			
Üretim Planlama	4	0,245	1		1		1						1	1	1				1	
Kimya , Chemistry	2	0,176		1					1	1	1	1			1		1	1	1	
KKİT	2	0,176	1											1						
ESKA	2	0,176				1								1						
NKV (I. Öğr. + II. Öğr.)	2	0,176			1		1							1		1				
Olasılık, Probability	3	0,194	1	1	1		1						1	1	1	1	1			
Üretim Yöntemleri	2	0,133	1	1			1	1					1	1	1	1				
Teknik İngilizce III	2	0,151				1	1	1						1	1				1	
TBTK	2	0,146			1					1								1	1	
Teknik İngilizce V	2	0,155		1		1		1		1				1	1	1	1			
Tesis Planlaması	2	0,136	1	1	1								1	1	1		1			
İstatistik II	3	0,183	1	1	1		1	1						1	1				1	
Mühendislik Mekaniği	2	0,189		1	1	1	1	1		1				1	1	1	1	1	1	
Sosyal Seçmeli I	3	0,148				1		1						1	1					
İş Etüdü	4	0,231				1		1	1	1	1	1			1	1	1	1	1	

#### 4.1.2 Problem parametrelerinin tahmini için bir ANP modeli

Çalışmanın bu bölümünde ise, gözetmen-sınav atama probleminin parametreleri

arasındaki ilişkilerin de dikkate alındığı bir Analitik Serim Süreci (ANP) modeli geliştirilmiştir. Bölüm 4.2.2’de geliştirilen AHP modeli ile sadece sınavlara ilişkin ağırlıklar elde edilirken, bu bölümde önerilen ANP modeli ile hem sınavların ağırlıkları hem de amaç fonksiyonunun skalerleştirilmesinde kullanılacak olan ağırlıklar elde edilmiştir. Bir başka ifadeyle amaç fonksiyonları da ağırlıklandırılmıştır.

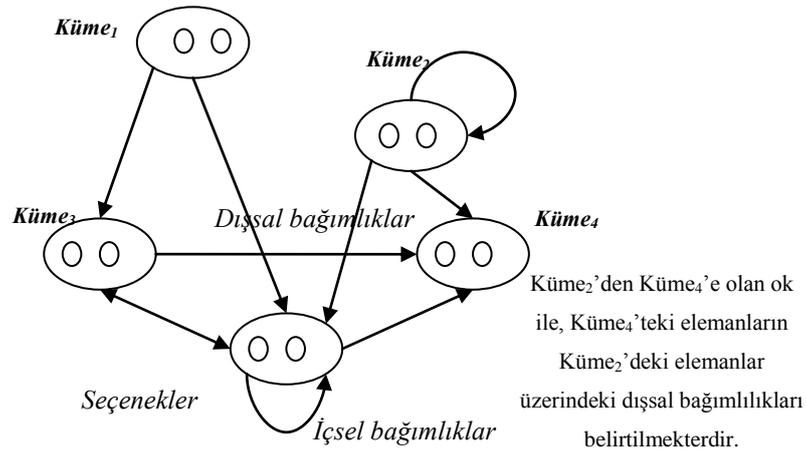
Birçok karar problemi, ölçütler ve seçenekler arasında etkileşim ve bağımlılık içerir (Carlsson and Fuller, 1995; Karwan et al., 1995; Saaty, 1996). Bir problemde yeralan bileşenler arasındaki ilişkiler tek yönlü değil karşılıklı olduğu zaman, hiyerarşik tanımlamalar yeterli olmaz. Bu durumda seviyeler ortadan kalkar ve bileşenlerin ağırlıklarını bulmak daha karmaşık bir sürecin analizini gerektirir (Özdemir, M.S., 2004). ANP, karar problemlerini kümelerde gruplandırılmış elemanlar ve seçeneklerden oluşan bir serim şeklinde ifade eder (Şekil 4.3). Bu yapı sayesinde, doğrudan ilişkilendirilmemiş bileşenler arasında olabilecek dolaylı etkileşimler ve geribildirimler de dikkate alınmaktadır.

Serim modelinde, AHP yönteminde olduğu gibi seçeneklerin önemini sadece ölçütler belirlemez, problemde yer alan ölçütler ve seçenekler birbirleriyle karşılıklı etkileşim halinde olabileceğinden aynı zamanda seçenekler de ölçütlerin önemini belirler (Saaty, 1996). ANP’de, karar verme problemine ait tüm bileşenler ve ilişkilerin tanımlanmasının ardından karşılıklı ilişkiler de ifade edilir. Aynı kümeyle ait olsun veya olmasınlar, herhangi iki ölçüt arasındaki bağımlılık dikkate alınmaktadır.

ANP de AHP’de olduğu gibi ikili karşılaştırma esasına dayanır ve karşılaştırmalarda Saaty’nin 1-9 ölçeği kullanılır. Ele alınan karar probleminde birbiriyle etkileşim halinde olan tüm ölçütler ve seçenekler sistematik bir biçimde değerlendirilirler. ANP’de temel olarak üç tip sorgulama vardır (Özdemir, 2003):

1. Bir ölçüt ve iki bileşen verildiğinde, bileşenlerden hangisinin ilgili ölçüte göre etkisinin daha çok olduğu,
2. Bir seçenek ve iki bileşen verildiğinde bileşenlerden hangisinin ilgili seçenek üzerinde etkisinin daha çok olduğu,

3. Bir ölçüt ile A, B ve C bileşenleri verildiğinde, B ve C'den hangisinin verilen ölçüt açısından A bileşeni üzerinde daha çok etkisinin olduğu.



Şekil 4.3 Serim modeli

ANP'de ağırlıklandırılmamış matris (unweighted matrix-UM), ağırlıklandırılmış matris (weighted matrix- WM) ve limit matrisi (limit matrix-LM) olmak üzere üç tür matris kullanılarak analizler yapılır. Ağırlıklandırılmamış matris, ikili karşılaştırmalar sonucu bulunan ve her bileşenin göreceli önem vektörünü veren matristir. Ağırlıklandırılmış matris, bu değerlerin, ilgili bileşenin içinde yer aldığı kümenin ağırlığı ile çarpılması sonucu elde edilen değerlerin yer aldığı matristir. Limit matris ise, problemin geri bildirim içermesi nedeniyle ağırlıklandırılmış matrisin limiti alınarak bileşenlerin göreceli önem değerlerinin yakınsadıkları değerlerin elde edildiği matristir. Problemin sonuç değerleri bu limit matristen okunur (Üstün vd., 2005).

Yönetmelik açıdan bakıldığında, sorunsuz geçen sınav dönemleri ve zamanında gerçekleştirilen görevler büyük önem taşımaktadır. Sınav dönemlerinde sürecin performansını etkileyen diğer katılımcılar da öğrenciler ve öğretmenler olmaktadır. Öğretmenlerin de sınavların yapıldığı zaman dilimleri ve derslikler konusunda çeşitli istekleri olabilir. ANP, özellikle sonlu sayıda seçeneğin bulunduğu problemlerde eniyi

seçeneğin belirlenmesi için kullanılmaktadır. Seçenekler elde edilen değerler, bazen bir bütçenin elde edilen ağırlıklar oranında seçenek yatırımlara dağıtılması gibi kararlarda kullanılabilir.

Çalışmada, gözetmen-sınav atama probleminin amaçlarını önceliklendirmek için tek serimli fakat ayrıntılı bir ANP modeli oluşturulmuştur. Problemin amaçları olan *gözetmenlerin en büyük yükünün en küçüklenmesi, istenmeyen zaman dilimlerine yapılan atamaların en büyüğünün en küçüklenmesi* ve *toplam atama maliyetinin en küçüklenmesi* seçenekler kümesinin elemanları olarak belirlenmiştir. Bu amaçları etkileyen ölçütler özelliklerine göre gruplandırılarak kümeler oluşturulmuştur. Bu kümeler: *öğrenciler, öğretmenler, gözetmenler, yönetim, ders, sınav tipi, sınav süresi, sınav zaman dilimi, sınav günü* ve *seçenekler* olarak belirlenmiştir.

Sınavlarla ilgili belirlenen kümelerin elemanları izleyen şekildedir:

<b><u>Küme</u></b>	<b><u>Eleman</u></b>
• Sınav süresi	<i>60, 90 ve 120 dakika</i>
• Sınav günü	<i>Pazartesi, Salı, Çarşamba, Perşembe, Cuma, Cumartesi</i>
• Sınav zaman dilimi	<i>9:00-11:00, 11:00-13:00, 14:00-16:00, 16:00-18:00</i>
• Sınav tipi	<i>açık düzen, yazılı, test ve cevap kağıtsız</i>

Serimde yer alan diğer küme ve elemanları ise aşağıda verilmiştir:

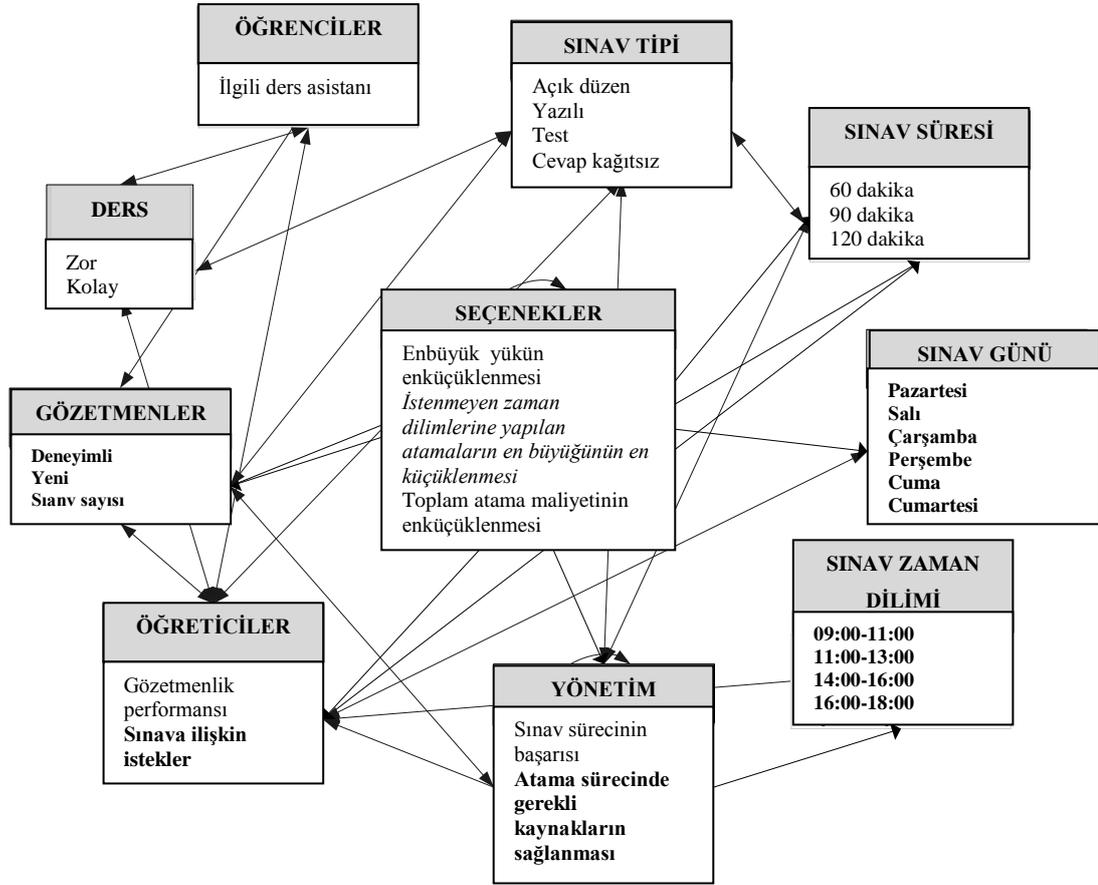
• Gözetmenler	<i>Deneyimli, yeni, sınav sayısı</i>
• Öğreticiler	<i>Gözetmenlik performansı, sınava ilişkin istekler</i>
• Yönetim	<i>Sınav sürecinin başarısı, atama sürecinde gerekli kaynakların sağlanması</i>
• Ders	<i>Zor, kolay</i>
• Öğrenciler	<i>İlgili ders asistanı</i>

Küme ve elemanlarının belirlenmesinin ardından, içsel ve dışsal bağlantılar

belirlenmiştir. Şekil 4.4'te örnek olarak seçenekler kümesinde yer alan, eğik yazılan, *istenmeyen zaman dilimlerine yapılan atamaların en büyüğünün en küçüklenmesi* amacını etkileyen ölçütler gösterilmiştir. Diğer kümelerde kalın harflerle yazılmış olan ölçütler, seçilen ölçüti etkileyen ölçütler olarak tanımlanmıştır. *İstenmeyen zaman dilimlerine yapılan atamaların en büyüğünün en küçüklenmesi* amacı; sınav zaman dilimi, sınav günü ve gözetmenler kümelerindeki tüm ölçütler tarafından etkilenmektedir. Yanısıra, öğretim elemanının ilgili sınava ilişkin istekleri ve atamalar konusunda yönetimin gerekli kaynakları sağlaması da, seçili amacı etkileyen diğer ölçütlerdir.

Küme ve elemanlar arasındaki bağlantıların belirlenmesinin ardından, ölçütlerin birbirleri üzerindeki etkileri göz önünde bulundurularak, görece önemlerin belirlenmesi için ikili karşılaştırmalar yapılmıştır. Örneğin, “istenmeyen zaman dilimlerine yapılan atamaların en büyüğünün en küçüklenmesi amacı üzerinde hangi faktör ne kadar daha fazla etkiye sahiptir? Öğretim üyelerinin ilgili sınava ait ilgili istekleri mi yoksa deneyimli bir gözetmen olmak mı?” şeklinde sorular sorulmuştur. Öğretim üyeleri kendi derslerine yardımcı olan araştırma görevlilerinin dersin sınavında gözetmenlik yapmalarını genellikle tercih ederler. Bu noktada deneyimli de olsa bir gözetmenin istenmeyen zaman dilimine atanmama eğilimi karşısında öğretim üyesinin ilgili araştırma görevlisini talep etmesi, bu süreci daha fazla etkileyebilir. Yukarıda örneği verilen bu gibi tutumlar bu sayede atama sürecinde yer almaktadır.

Gözetmen-sınav atama probleminin farklı katılımcıları olduğundan ikili karşılaştırmalar büyük çoğunluğu gözetmenlerden oluşan bir grup tarafından yapılmıştır. AHP yönteminde olduğu gibi burada da farklı yargıları bir araya getirirken geometrik ortalama alınmıştır.



Şekil 4.4 Gözetmen sınav atama problemi için geliştirilen ANP modeli

Elde edilen ağırlıklar Çizelge 4.15’te verilmiştir. Bu sonuçlara göre, cumartesi günü yapılan, açık düzen olan, sabah saat 9’daki sınavlar yüksek ağırlık elde etmiştir. Amaçlara bakıldığında da istenmeyen zaman dilimlerine yapılan atamaların en küçüklenmesinin diğer amaçlara göre daha fazla öneme sahip olduğu görülmektedir. Bu ağırlıklar, gözetmenlerin istenmeyen zaman dilimlerine kaçınılmaz olarak atandığı durumlarda bu tür sınavların varlığının gözetmenlere eşit olarak atanmasında yönlendirici olacaktır.

Ayrıca, deneyimli gözetmenlerin ve yönetimin atama süreci için gerekli kaynakları sağlamanın da atama sürecini etkileyen önemli faktörler olduğu görülmektedir.

**Çizelge 4.15** ANP modeline göre elde edilen ölçüt ve seçenek ağırlıkları

Küme	Eleman	Önem	Küme	Eleman	Önem	
Yönetim	Sınav sürecinin başarısı	0.081	Öğretim elemanları	Gözetmenlik performansı	0.065	
	Atama sürecinde gerekli kaynakların sağlanması	<b>0.128</b>		Sınava ilişkin istekler	<b>0.129</b>	
Gözetmenler	Deneyimli	<b>0.051</b>	Sınav süresi	60 dakika	0.007	
	Yeni	0.025		90 dakika	0.012	
	Sınav sayısı	0.011		120 dakika	<b>0.021</b>	
Sınav dilimi	zaman	9-11	Sınav günü	Pazartesi	0.003	
		11-1		Salı	0.002	
		2-4		Çarşamba	0.002	
		4-6		Perşembe	0.002	
				Cuma	0.003	
				Cumartesi	<b>0.006</b>	
Sınav tipi	Açık düzen	<b>0.011</b>	Öğrenciler	İlgili ders asistanı	0.086	
	Test	0.010				
	Yazılı	0.007				
	Cevap kağıtsız	0.005				
Seçenekler	İstenmeyen zaman dilimlerine yapılan en büyük atamanın en küçüklenmesi	<b>0.014</b>	Ders	Kolay Zor	0.099 <b>0.173</b>	
		En büyük gözetmen yükünün en küçüklenmesi				0.011
		Toplam atama maliyetinin en küçüklenmesi				0.010

Çizelge 4.15'te görüldüğü üzere, 120 dakikalık ya da cumartesi günlerine çizelgelenmiş sınavlar beklendiği üzere daha fazla öneme sahiptir. Daha fazla önem, gözetmenlik sırasında daha fazla emek, dikkat ve daha fazla süre harcamak anlamına gelmektedir. Yine benzer olarak, sabah ya da akşamüzeri saatlerinde ya da hafta sonlarındaki sınavlar, uzun süreli sınavlar ya da haftanın son gününe çizelgelenmiş sınavlar daha fazla öneme sahiptir.

Limit matrizen elde edilen sonuçlara göre istenmeyen zaman dilimlerine yapılan atamaların en büyüğünün en küçüklenmesi amacının ağırlığı 0.014, gözetmenlerin en büyük yükünün en küçüklenmesi amacının ağırlığı 0.011 ve toplam atama maliyetinin en küçüklenmesi amacının ağırlığı 0.010 olarak elde edilmiştir. Bu değerlerin normalleştirilmesi ile de amaç fonksiyonlarının göreceli önemleri sırasıyla 0.4, 0.31 ve 0.29 olarak elde edilmiştir (Çizelge 4.16).

**Çizelge 4.16** Amaç fonksiyonlarının görelî önem değerleri

Seçenekler	Limit matristen elde edilen önem değerleri	Normalleştirilmiş önem değerleri
Istenmeyen zaman dilimlerine yapılan en büyük atamanın en küçüklenmesi	0.014	0.4
En büyük gözetmen yükünün en küçüklenmesi	0.011	0.31
Toplam atama maliyetinin en küçüklenmesi	0.010	0.29

Bölüm 4.2.1’de gerçek sisteme ait mevcut parametreler ile çözdürülen model, bu bölümde ANP modelinden elde edilen ağırlıklar ile tekrar çözdürülmüştür. Bir sınavın ağırlığını etkileyen süre, zaman dilimi, gün, tip ve zorluk ölçütlerinin ağırlıkları toplanarak yeni sınav ağırlıkları ( $a_j$ ) elde edilmiştir.

Çizelge 4.17’de 2004–2005 bahar dönemi birinci ara sınavları için ilgili sınavların mevcut ve önerilen yeni ağırlıkları ile sınav için gerekli gözetmen sayıları ( $g_j$ ) verilmiştir. Yeni ağırlıkları bir örnek ile açıklayacak olursak, Pazartesi günü sabah 9’da çizelgelenmiş 90 dakikalık Fizik dersi sınavının ağırlığı ilgili sınav parametrelerine karşı gelen ağırlıkların toplanması ile 0.131 olarak elde edilmiştir. Benzer şekilde, mevcut durumda sadece 2, 3 ve 4 değerleri verilmiş olan sınavlara, sahip oldukları özelliklere göre farklı ağırlıklar atanmıştır.

Ayrıca, gözetmen-sınav atama probleminin çözümü için kurulan çok amaçlı modelin amaç fonksiyonunun skalerleştirilmesi için de elde edilen bu ağırlıklar kullanılmıştır: Gözetmenlerin en büyük yükünün en küçüklenmesi 0.31, istenmeyen zaman dilimlerine yapılan atamaların en büyüğünün en küçüklenmesi 0.4 ve toplam atama maliyetinin en küçüklenmesi amacı 0.29 ağırlıklarını almıştır.

Çizelge 4.17 ANP'den elde edilen yeni sınav ağırlıkları

Gün	Zaman dilimi	Sınav	$g_j$	$a_j$	yeni $a_j$	zaman dilimi	tip	süre	gün	zorluk
12.12.2005 Pazartesi	09:00	Fizik I (A,B) <sup>12</sup>	2	2	0.131	0.01	0.007	0.012	0.003	0.099
	09:00	Physics I	1	2	0.131	0.01	0.007	0.012	0.003	0.099
	11:00	Yöneylem Araş. I (A,B)	2	3	0.209	0.005	0.007	0.021	0.003	0.173
	11:00	Operations Research I	1	3	0.209	0.005	0.007	0.021	0.003	0.173
	14:00	Mühendislik Malzemeleri (A,B)	2	2	0.120	0.004	0.007	0.007	0.003	0.099
	16:00	İnsan Kaynakları Yön.	1	2	0.125	0.006	0.005	0.012	0.003	0.099
	16:00	Pazarlama Yön. (A,B)	1	2	0.125	0.006	0.01	0.007	0.003	0.099
13.12.2005 Salı	09:00	Matematik I (A,B)	3	3	0.213	0.01	0.007	0.021	0.002	0.173
	09:00	Calculus I	1	3	0.213	0.01	0.007	0.021	0.002	0.173
	11:00	İngilizce I (A and B)	2	2	0.123	0.005	0.01	0.007	0.002	0.099
	14:00	Benzetim (A,B)	3	3	0.205	0.004	0.005	0.021	0.002	0.173
	16:00	Diferansiyel Denk. (A,B)	2	3	0.209	0.006	0.007	0.021	0.002	0.173
	16:00	Differential Equ.	1	3	0.209	0.006	0.007	0.021	0.002	0.173
14.12.2005 Çarşamba	09:00	Türk Dili I (A,B)	2	2	0.128	0.01	0.01	0.007	0.002	0.099
	11:00	Bilg. Prog. II (A,B)	3	3	0.197	0.005	0.005	0.012	0.002	0.173
	14:00	Teknik Resim (A,B)	2	3	0.207	0.004	0.007	0.021	0.002	0.173
	14:00	Mühendislik Ekonomisi (A,B)	2	3	0.198	0.004	0.007	0.012	0.002	0.173
	18:00	Üretim Planlama (A,B)	2	4	0.209	0.006	0.007	0.021	0.002	0.173
15.12.2005 Perşembe	09:00	Kimya (A, B)	2	2	0.130	0.01	0.007	0.012	0.002	0.099
	09:00	Chemistry	1	2	0.130	0.01	0.007	0.012	0.002	0.099
	09:00	KKİT	1	2	0.130	0.01	0.007	0.012	0.002	0.099
	09:00	ESKA	1	2	0.130	0.01	0.007	0.012	0.002	0.099
	09:00	NKV (A, B)	2	2	0.130	0.01	0.007	0.012	0.002	0.099
	11:00	Olasılık (A,B)	2	3	0.208	0.005	0.007	0.021	0.002	0.173
	11:00	Probability (A)	1	3	0.208	0.005	0.007	0.021	0.002	0.173
	14:00	Üretim Yöntemleri (A, B)	3	2	0.124	0.004	0.007	0.012	0.002	0.099
	16:00	Teknik İngilizce III (A, B)	2	2	0.129	0.006	0.015	0.007	0.002	0.099
16.12.2005 Cuma	09:00	TBTK (A,B)	1	2	0.134	0.01	0.015	0.007	0.003	0.099
	09:00	Teknik İngilizce V (A, B)	2	2	0.129	0.01	0.01	0.007	0.003	0.099
	11:00	Tesis Planlaması (A, B)	2	2	0.130	0.005	0.016	0.007	0.003	0.099
	14:00	İstatistik II ( A1, A2, B1, B2)	2	3	0.208	0.004	0.007	0.021	0.003	0.173
	16:00	Mühendislik Mek. (A1-2, B1)	2	2	0.131	0.006	0.016	0.007	0.003	0.099
17.12.2005 Cumartesi	11:00	Sosyal Seçmeli I	2	3	0.198	0.005	0.007	0.007	0.006	0.173
	14:00	İş Etüdü(A1-2, B1)	2	4	0.211	0.004	0.007	0.021	0.006	0.173

Ayrıca, her amaç farklı bir ölçekte olduğundan, bu amaçların normalleştirilmesi de yapılmıştır. Bir sınav döneminde gözetmenlerin zaman dilimlerini tercih etmeme

<sup>12</sup> Bir derse ait farklı gruplar

düzeylelerine ilişkin bilgi web tabanlı sistem ile toplanmaktadır. Buna göre, toplam atama maliyeti, en fazla tüm gözetmenlerin  $t$ . zaman dilimlerini tercih etmeme düzeylerinin toplamı ( $ctoplam$ ) kadar olacaktır. Dolayısıyla bu amaç fonksiyonu  $ctoplam$  değerine bölünerek normalleştirilmiştir.

Bir gözetmen sınav dönemi boyunca tüm sınavlara atanması durumunda en büyük yüke sahip olacaktır. İkinci amaç fonksiyonu da tüm sınav ağırlıkları toplamına ( $atoplam$ ) bölünerek normalleştirilmiştir.

İstenmeyen zaman dilimlerine yapılan atamaların en büyüğünün en küçüklenmesi amacı da istenmeyen zaman dilimlerinde yapılan sınavlarda görevlendirilen toplam gözetmen sayısına ( $btoplam$ ) bölünerek normalleştirilmiştir.

Yapılan normalleştirme ve skalerleştirme işlemlerinin ardından, Bölüm 4.2.1’de geliştirilmiş olan modelin amaç fonksiyonu izleyen (1) nolu fonksiyona dönüştürülmüştür. Çalışmanın bu bölümünde, ANP’den elde edilen sınav ve amaç fonksiyonu ağırlıkları (sırasıyla 0.29, 0.31 ve 0.4) ile oluşturulan yeni amaç fonksiyonu ile çözdürülen model bir en iyi çözüm vermiştir.

$$enk \left\{ \frac{0.29}{ctoplam} * \left( \sum_i \sum_t \sum_j c_{it} y_{ij} s_{ij} \right) \right\} + \frac{0.31}{atoplam} r_1 + \frac{0.4}{btoplam} r_2$$

ANP ile elde edilen yeni ağırlıklar ile Bölüm 4.2.1’de geliştirilmiş olan matematiksel model yeniden çözdürüldüğünde yeni bir atama sonucu elde edilmiştir (Çizelge 4.18). Bölüm 4.2.2’de olduğu gibi çalışmanın bu bölümünde farklı sınav ağırlıkları kullanıldığından, mevcut ağırlıklar ile elde edilen çözümden farklı bir atama sonucu elde edilmiştir. Sistemde yer alan farklı uzmanların görüşlerinin alınmasının yanısıra, problem ölçütleri arasındaki ilişkilerin de ele alınması ile geliştirilmiş olan matematiksel model bu sayede daha gerçekçi olarak ele alınmıştır. Karşılıklı ilişkilerin

ele alınabilmesi ve sistematik bir yaklaşımın kullanılması kullanıcıların sisteme olan güvenleri ve örgütsel performansları açısından önemlidir.

**Çizelge 4.18** ANP ağırlıkları kullanılarak elde edilen çözüm

Sınav	Ağırlıklar		İlk çözüm									ANP ağırlıkları kullanılarak elde edilen çözüm								
	Mevcut	Yeni (ANP)	Gözetmenler									Gözetmenler								
			1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
Fizik I (A, B), Physics I	2	0.131		1	1	1				1	1		1	1	1				1	1
Yöneş. Araş. I (A, B), OR	3	0.209	1	1	1		1				1	1	1	1		1				1
Müh. Malz. , (A,B)	2	0.120	1	1	1		1					1			1		1			1
İnsan Kaynakları Yön.	2	0.125							1							1				
Pazarlama Yön. (A,B)	2	0.125				1	1				1				1	1				1
Math. I (A,B), Calculus	3	0.213		1	1	1		1	1	1		1	1	1		1	1	1		
İngilizce I (A, B)	2	0.123	1				1						1			1				
Benzetim (A,B)	3	0.205	1	1		1		1			1	1	1	1		1				1
Dif. Denk. , Differential Equ.	3	0.209				1	1	1		1	1				1	1	1	1		1
Türk Dili I	2	0.128			1	1		1	1			1	1	1		1				
Bilgis. Prog. II	3	0.197	1	1	1	1	1				1	1	1	1	1	1	1			
Teknik Resim	3	0.207				1	1	1	1			1		1						1
Mühendislik Ekonomisi	3	0.198	1	1	1						1	1		1		1		1		
Üretim Planlama	4	0.209	1		1		1				1	1		1						1
Kimya , Chemistry	2	0.130		1				1	1	1	1		1			1	1	1	1	1
KKIT	2	0.130	1									1								
ESKA	2	0.130				1									1					
NKV (A,B)	2	0.130			1	1						1			1					
Olasılık, Probability (A)	3	0.208	1	1	1		1				1	1	1	1		1				1
Üretim Yöntemleri (A,B)	2	0.208	1	1			1	1			1	1	1	1	1	1				
Teknik İngilizce III (A,B)	2	0.124				1	1	1							1	1				1
TBTK (A,B)	2	0.129			1					1		1					1			
Teknik İngilizce V	2	0.129		1		1		1		1			1	1		1		1		
Tesis Planlaması (A,B)	2	0.130	1	1	1						1	1	1	1						1
İstatistik II (A,B)	3	0.208	1	1	1		1	1				1	1		1	1				1
Mühendislik Mekaniği (A,B)	2	0.131		1	1	1	1	1	1		1	1	1	1	1	1		1	1	1
Sosyal Seçmeli I	3	0.198				1		1							1		1			
İs Etüdü (A,B)	4	0.211			1		1	1	1	1	1			1		1	1	1	1	1

Mevcut ağırlıklar kullanılarak elde edilen çözümde, dördüncü gözetmen Fizik I, Pazarlama, Math. I, Benzetim, Dif.Denk. Bilg. Prog., Türk Dili, Teknik Resim, ESKA,

Tek. Ing. III ve V, Mühendislik Mekaniği, Sosyal Seçmeli ve İş Etüdü derslerinin sınavlarına atanmıştır. Önerilen yeni sınav ağırlıkları kullanıldığında ise, Fizik I, Pazarlama, Math. I, Müh. Malz. Dif.Denk. Bilg. Prog., Türk Dili, Müh. Eko., ESKA, Üretim Yönt., Teknik Ing. V, Tesis Planlaması, Mühendislik Mekaniği, Sosyal Seçmeli ve İş Etüdü derslerinin sınavlarına atanmıştır. İlk çözümde ise dördüncü gözetmen Benzetim, Teknik Resim, Tek. Ing. III sınavlarına atanmışken, yeni çözümde bu sınavların yerine Muh. Malz., Muh. Eko., Uretim Yont. ve Tesis Planlaması derslerinin sınavlarına atanmıştır. Bu durumda gözetmenin toplam yükü ilk çözümde 0.407 iken, ikinci çözüm ile 0.445 değerini almıştır. ANP ağırlıkları ile elde edilen yeni çözümde  $r_1=2.340$ ,  $r_2=2$  değerleri elde edilirken, ilk çözümde karşı gelen değerler 0.417 ve 2 olarak elde edilmiştir.

Uygulamaların yapıldığı sistemde, AHP ve ANP modellerinden elde edilen ağırlıklar karşılaştırıldığında, parametrelerin aldığı değerlerin farklı olduğu görülmektedir. Ancak, *120 dakikalık, sabah 9, cumartesi günü ve açık düzen* sınavlarının her iki modelde de kendi ana ölçütleri arasında en yüksek ağırlığı aldığı belirlenmiştir.

Literatürde yer alan gözetmen-sınav atama çalışmalarında gözetmenlerin zaman dilimlerine ilişkin tercihleri alınırken, bu çalışmada tercihlerin olabildiğince sağlanmasının yanı sıra gözetmenlerin sınav yüklerinin de olabildiğince dengeli olması amaçlanmıştır. Sınav yüklerinin belirlenmesi aşamasında da sınav ağırlıklarının gerçekçi bir şekilde belirlenmesi önemlidir. Çalışma kapsamında, olabildiğince farklı kurumlardan katılımcıların görüşleri alınarak kurulan her iki çok ölçütlü model ile gözetmen-sınav atama süreci daha sistematik bir şekilde ele alınmıştır. Gözetmenlere sınav süresi boyunca daha fazla dikkat gerektiren ve dolayısıyla gözetmenlerin sınav yükünü artıran faktörler ile sınav ağırlıkları belirlenmiştir.

## BÖLÜM 5

### ÇOK AMAÇLI EĞİTİMSEL ZAMAN ÇİZELGELEME PROBLEMLERİ İÇİN WEB TABANLI BİR KARAR DESTEK SİSTEMİ TASARIMI

Bu bölümde, çalışma kapsamında ele alınan ders-derslik-zaman, sınav-derslik-zaman çizelgeleme ve gözetmen-sınav atama problemlerini web tabanlı, kullanıcı etkileşimli ve esnek bir karar destek ortamında çözecek otomatikleştirilmiş bir sistemin (EZÇ-KDS) tasarımı gerçekleştirilmiştir. Tasarlanan sistemde, problemler için gerekli veritabanı ile gözetmenlerin tercihlerini girebileceği ve yöneticinin çeşitli görevleri yerine getireceği arayüzler oluşturulmuştur. Yanısıra, 3. ve 4. bölümlerde geliştirilen matematiksel modeller, model tabanını oluşturmaktadır.

Ele alınan problemlerin boyutu; öğrenci, ders ve sınav sayıları ile derslik ve gözetmen kısıtlarına bağlı olarak arttıkça, problemlerin zorluk dereceleri de artmaktadır. Bu durum tatmin edici, uygun ve kaliteli çizelgeler üretecek otomatikleştirilmiş sistemlere olan ihtiyacı artırmaktadır. Otomatikleştirilmiş çizelgeleme sisteminin, kullanıcı etkileşimli olması, özel kısıt ve tercihlerin girilebilmesine ve farklı kısıtlara ağırlıklar verilmesine olanak tanınması önemlidir.

Eniyileme teknikleri ve Internet'in birlikte kullanılması ile kullanıcılar çözüm sistemini web arayüzlerini de kullanarak istediği zaman istediği yerden çalıştırabilir. Yanı sıra, tasarlanan sistem ile kullanıcılar istek ve tercihleri ile uygun olmadıkları zaman dilimlerini ilgili modele aktarabilmektedir.

Bu bölümde, öncelikle karar destek sistemleri ve web tabanlı karar destek sistemleri genel özellikleri ile ele alınmış, Bölüm 5.3'te ise eğitimsel zaman çizelgeleme problemlerinin çözümü için tasarlanan ve yukarıda açıklanan Web Tabanlı Eğitimsel Zaman Çizelgeleme Karar Destek Sistemi (EZÇ-KDS) tanıtılmıştır.

## 5.1 Karar Destek Sistemleri

Karar aşamasında karar vericilere yardımcı olması için bilgisayar kullanma fikri ilk olarak 1963'te ortaya atılmıştır (Eom ve Kim, 2005). En genel anlamda Karar Destek Sistemi (KDS), yönetici konumundaki karar vericilerin karar vermelerinde yardımcı olan sistemlerdir. Temel olarak KDS, karar vericinin karar vermesini gerektiren durumla ilgili olarak istediği, ihtiyaç duyduğu bilgileri derleyip, dilediğince değerlendirdiği ve daha bilgili olarak karar vermesi imkanının ortaya çıktığı bir ortam oluşturur (Gökçen, 2002).

Kuruüzüm (1998) KDS'ne ait izleyen üç temel özelliği tanımlamıştır:

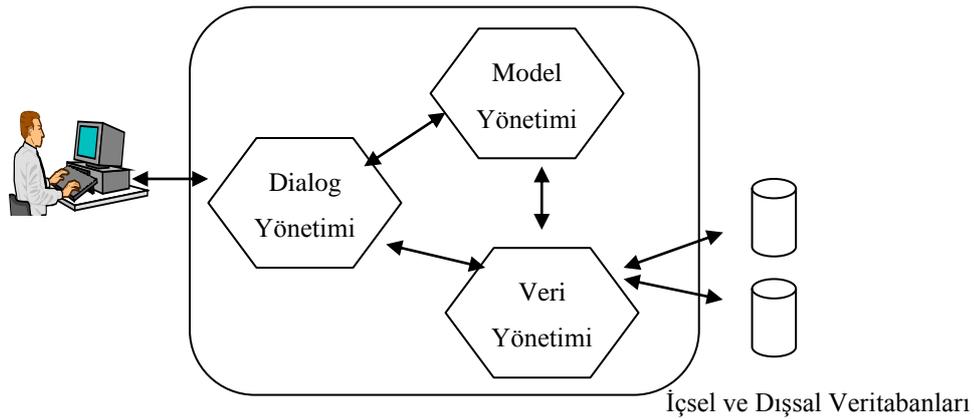
1. KDS, geleneksel veri erişim ve güncelleştirme fonksiyonları ile model ya da analitik tekniklerin bütünleşikliğini sağlamaya yönelmektedir.
2. KDS, bilgisayar ile yakından ilgisi olmayanların onu kolaylıkla kullanabileceği ve etkileşebileceği ortamı sağlamaya odaklanmaktadır.
3. KDS, karar verici kişi ya da grubun karar verme yaklaşımındaki değişiklikleri göz önüne alarak buna uyumu sağlayacak nitelikte esneklik özelliğine sahip olabilmektedir.

Bu özelliklerden yola çıkarak bir KDS'ni, “model ve verileri kullanan, karar vericileri destekleyen, bilgisayar temelli, etkileşimli bir sistem” olarak tanımlayabiliriz.

Eom ve Kim'in (2005) 1995 ile 2001 yılları arasında yayınlanmış KDS ile ilgili çalışmaları derledikleri makalelerinde, incelenen çalışmalar içinde eğitim alanındaki çalışmaların en büyük ikinci paya sahip olduğu belirtilmiştir. Eğitim alanındaki KDS; ders çizelgeleme (Kassicieh vd., 1986; Piechowiak ve Kolski, 2004), sınav çizelgeleme, gözetmen atama (Ozturk vd., 2010), üniversite zaman çizelgeleme (Foulds ve Johnson, 2000; Dimopoulou ve Miliotis, 2001) ve kayıtlara ilişkin tahminler ile yeni öğrenim kurumlarını yerleştirme (Taylor vd., 1999) gibi problemlerin çözümünde kullanılmıştır.

Temel bir KDS, kullanıcı ile KDS arasında iletişim için *diyalog sistemi*, problemle ilgili verilerden oluşan alan bilgisini içeren *bilgi sistemi* ve karar problemini

çözebilecek programlardan oluşan *problem işleme sistemi* bileşenlerinden oluşmaktadır (Kersten, 2002). Gökçen (2002) de KDS'nin üç bileşenini Şekil 5.1'de verildiği gibi göstermiştir.



**Şekil 5.1** KDS bileşenleri

### *Veri yönetimi*

Veri yönetimi, karar vericinin belli bir kararı verebilmesi için, ilgili verinin getirilmesi, saklanması ve organize edilmesiyle ilgili değişik faaliyetlerin yerine getirildiği KDS bileşenidir. Verilere erişimi tek bir yerden sağlamak ve yeni veri girişi yapmak vb. nedenlerle veritabanı kullanılmaktadır. Bir problemin sonuçları bir başka problem için girdi olabileceğinden sonuçlar da veritabanında saklanabilir. Örneğin, ders zaman çizelgeleme probleminin çıktısı olan hangi dersin hangi zaman diliminde yapılacağı bilgisi ders-öğretici atama problemi için girdi olabilmektedir. Ayrıca, veritabanları ile tüm kullanıcıların sonuçlara erişimi de sağlanmış olur.

### *Model yönetimi*

KDS, veri inceleme ve çözüm üretmede analitik modeller kullanır. Bu bileşen model tabanı ile model tabanı yönetim sistemi bileşenlerinden oluşmaktadır. Model tabanı; çeşitli istatistiksel, finansal, matematiksel vb. modelleri içermektedir. Model tabanı yönetim sistemi de, modellere veri ve parametre girişini yerine getirir.

### *Diyalog yönetimi*

Arayüzler, kullanıcıların sistemle doğrudan temasta bulunduğu tek bileşendir. Kullanıcıdan gelen işlem komutlarını konuşma dili işleyicisinden geçirerek bilgisayar programlarına dönüştürür. Bu programlar doğrultusunda model yönetimi ve veri yönetimi ile ilişki kurar ve kullanıcının istediği bilgileri oluşturur. Bu bilgiler, yine konuşma dili işleyicisinden geçirilerek görüntü-çıkıtı oluşturma komutlarına dönüştürülür. Dolayısıyla yazıcı ve grafik çıkıtı araçları ile elde edilmiş sonuçlar kullanıcıya iletilir. Böylece, karar verici daha bilgili olarak karar verebileceği ortama sahip olur.

## **5.2 Web Tabanlı Karar Destek Sistemleri**

Kullanımı, 1990'ların başından günümüze hızla artan Web (World Wide Web – www), en önemli iletişim platformu ve bilgi kaynaklarından biri durumuna gelmiştir (Lee ve Chen, 2002). Web teknolojilerinin gelişmesi sonucunda, sadece uzman kişilerin kullanabileceği programları, herhangi bir uzmanlığı olmayan kişiler de kullanabilir duruma gelmiştir. Web tabanlı uygulamalara Web tarayıcıları ile herhangi bir anda herhangi bir bilgisayardan erişilebilir. Böylece, eniyileme problemlerinin çözümü gibi karmaşık işlemler de Web üzerinden ilgili kullanıcılar tarafından herhangi bir uzmanlığa, özel bir yazılım ve/veya donanıma gereksinim duyulmadan yapılabilir. Ayrıca, tüm bilgisayarlara gerekli yazılım ve programları yüklemek yerine, bir sunucu üzerinde kurulu yazılım ve programlar yetkili bir kişi tarafından güncellenebilir ve diğer

tüm kullanıcıların da erişimi sağlanmış olur.

İnternetteki gelişime bağlı olarak, Web-tabanlı Karar Destek Sistemleri de KDS çalışmalarında yeni bir eğilim yaratılmıştır (Zhang ve Goddard, 2007). Bhargava vd. (2007) son yıllarda karar verme problemlerinin çözümünde web ortamında çalışan karar destek sistemlerinin yaygınlaştığını belirtmiştir. Rangaswamy ve Shell (1997), Kersten ve Noronha (1999), Mustajoki ve Hamalainen (2000) ile Eom ve Kim (2005) Web tabanlı karar destek sistemi ile ilgili çalışmalara örnek olarak verilebilir.

Eğitim sistemlerinde yer alan kullanıcıların (öğreticiler, öğrenciler, vb.) memnuniyet düzeylerini artırmak, yöneticilerin sistemi bir bütün olarak ele almalarına olanak verip sistem performansını yükseltmek amaçlarıyla, zaman çizelgelerinin oluşturulmasında da bilgisayar ve diğer teknolojik olanakların kullanılması yaygınlaşmaktadır.

### **5.3 Eğitimsel Zaman Çizelgeleme Problemleri için Web Tabanlı Karar Destek Sistemi Önerisi (EZÇ-KDS)**

Yetmiş bölümün olduğu bir üniversitede senede iki eğitim dönemi olduğu ve her dönemde üç sınav yapıldığı düşünülürse, sınav-derslik-zaman çizelgeleme ile gözetmen sınav atama problemlerinin bir yılda toplam  $70 \times 2 \times 3 = 420$  kez çözülmesi gerekir. Her sınav dönemi için en az bir kişinin (programı hazırlayan kişi) 10 saat harcadığı varsayılırsa, sadece sınavlara gözetmen atama işi için böyle bir üniversitede 4200 adam-saatlik bir iş gücü gerekmektedir.

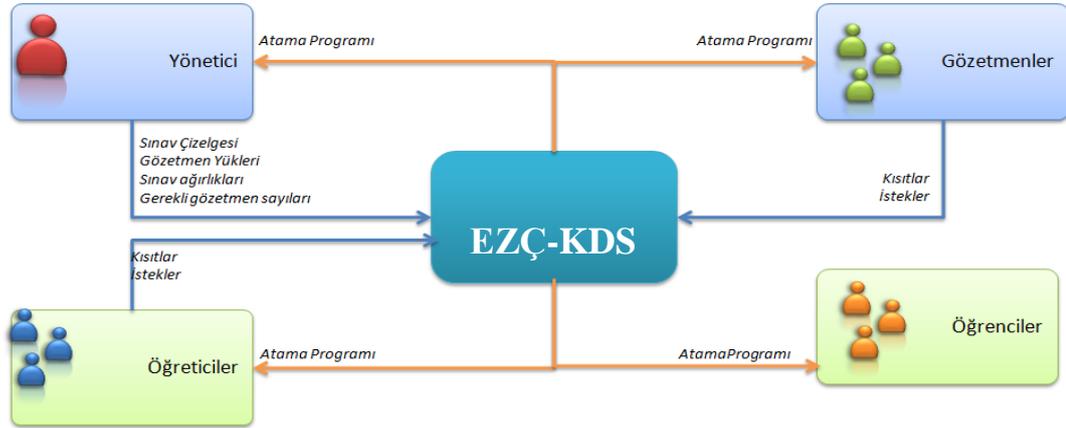
Çalışma kapsamında ele alınan eğitimsel zaman çizelgeleme problemleri için tasarlanan web tabanlı karar destek sistemi (EZÇ-KDS) ile gerekli verilerin sisteme girilmesi durumunda, ilgili eğitimsel zaman çizelgeleme probleminin çözümü, kişileri sadece bir butonun tıklanma süresi kadar meşgul edecektir. Bu yüzden önemli oranda zaman kazanılacaktır. Bunun yanında, gözetmen ve öğretmenler kendi istek ve

tercihlerini de web aracılığı ile bildireceğinden, bu bilgilerin kişilerden alınması ve istenen formata getirilmesi için de ayrı bir zaman kaybı olmayacaktır. ,

EZÇ-KDS'nin web tabanlı olarak geliştirilmesi, yanı sıra izleyen kazanımları da beraberinde getirmektedir:

- Birebir etkileşimli kullanıcı arayüzleri sayesinde yapılan veri girişleri ile yazılı belge aktarımı gibi işlemleri yavaşlatan unsurlar ortadan kalkar.
- Sistemin web tabanlı olması, kullanıcıların istek ve kısıtlarının 7 gün 24 saat boyunca girilmesine olanak sağlar.
- Yöneticiler gerek duydukları tüm bilgiye web ortamında en ayrıntılı şekilde erişebilir. Örneğin, gözetmenler ile tek tek görüşmek yerine her birinin kısıt ve isteklerine tek bir noktadan hızlı bir şekilde erişebilirler.
- Problemin çözümü için her bilgisayarda ilgili yazılımın kurulu olması gerekliliği yoktur. Sunucu üzerinden yetkisi olan kişiler herhangi bir uzmanlık gerektirmeden problemi çözdürebilir.

EZÇ-KDS'nin içerik diyagramı 5.2'de verilmiştir. Sistemin kullanıcıları *yönetici*, *gözetmenler*, *öğreticiler* ve *öğrencilerden* oluşmaktadır. Web tabanlı uygulamalarda karar verilmesi gereken konulardan biri de hangi kullanıcılara hangi yetkilerin verileceğidir. Sisteme birden çok bilgisayardan Internet üzerinden erişim sağlandığı için farklı tipteki kullanıcılar için farklı erişim izinleri sağlanmıştır. Geliştirilen kullanıcı dostu arayüzler ile kullanıcılar web üzerinden veri girişi yapabilmekte ve yetkisi olan kullanıcılar da ilgili problemleri çözmek üzere matematiksel model ile C#' da geliştirilen programları çalıştırabilmektedir. Yönetici, sistemin genel kurallarını belirleyen parametreleri kendisi için hazırlanan ara yüzler yardımıyla sisteme girmektedir.



Şekil 5.2 EZÇ-KDS için içerik diyagramı

Yöneticinin sisteme girdiği bilgiler izleyen şekilde verilmiştir:

*Ders-derslik-zaman çizelgeleme problemi için;*

- Ders dönemi (güz/bahar/yaz)
- Öğrenci ders matrisi
- Derslik verileri (kapasite ve diğer nitelikler)
- Ders verileri (dersin gerektirdiği nitelikler)
- Zaman dilimi seçimi

*Sınav-derslik-zaman çizelgeleme problemi için;*

- Sınav dönemi (güz/bahar/yaz) ve türü (arasınav/final/küçük sınav)
- Sınavı yapılacak dersler
- Çakışması istenmeyen dersler
- Derslik verileri
- Sınav takvimi verileri (hangi gün hangi zaman dilimlerinde sınav yapılacak)

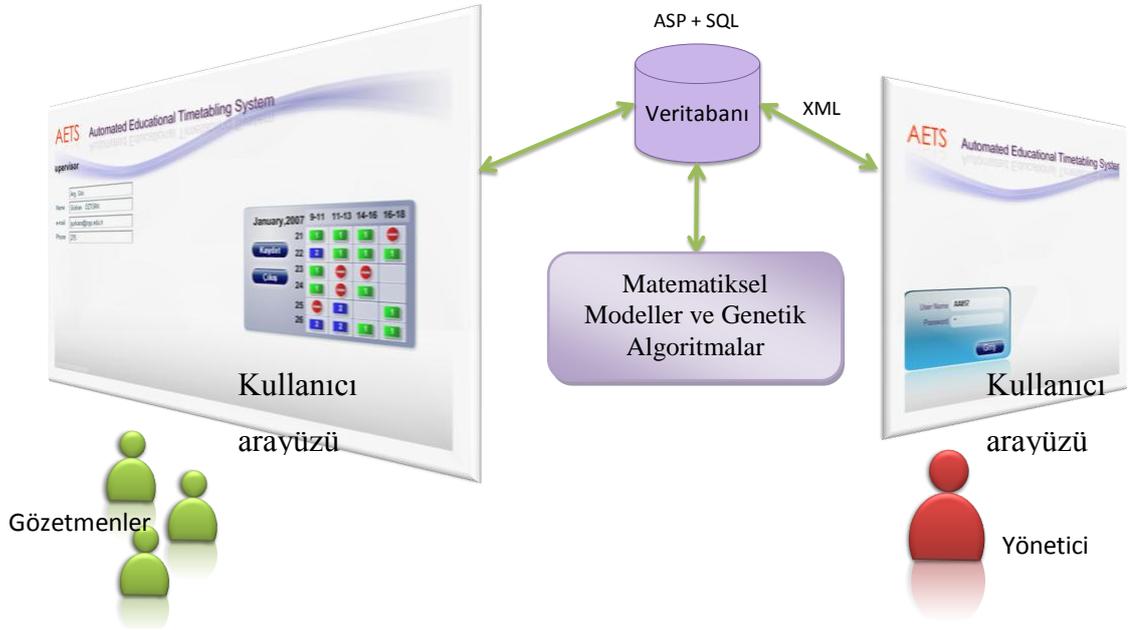
*Gözetmen-sınav atama problemi için;*

- Hangi sınavın hangi zaman dilimi ve hangi derslikte yapıldığını gösteren sınav çizelgesi

- Gözetmenlerin deneyimlerine göre deęişen yük oranları
- Sınav aęırlıkları
- Sınava girecek öęrenci sayısına ve derslik koşullarına göre belirlenen gerekli gözetmen sayıları

EZÇ-KDS'nin model tabanında, çalışmada ders-derslik-zaman çizelgeleme, sınav-derslik-zaman çizelgeleme ve gözetmen-sınav atama problemlerinin çözümü için geliştirilen matematiksel modeller ile genetik algoritmalar yer almaktadır. Web tabanlı olarak tasarlanan sistemin web arayüzleri Flash programı ile oluşturulmuştur. Veritabanı ile web ortamı arasındaki bilgi alış verişi için XML ve ASP teknolojilerinden yararlanılmıştır. Ayrıca, veritabanı işlemleri için de ASP ve SQL kullanılmaktadır.

Gözetmen-sınav atama probleminin çözümü için gözetmenler tercih ve kısıtlarını yine kendi kullanıcı adı ve şifrelerini kullanarak ilgili ara yüzler aracılığı ile veritabanına kaydetmektedirler. Sistem tarafından, gözetmenlerin kısıt ve tercih girişi için bir süre ayrılmaktadır. Probleme ait modelin oluşturulabilmesi için gerekli olan tüm parametreler elde edildikten sonra yönetici, matematiksel modelin otomatik olarak oluşturularak çözdürülmesi için gerekli olan komutu arayüz aracılığı ile verir. Bu model GAMS aracılığı ile çözdürülerek elde edilen sonuçlar veritabanına yazılır. Çözüm yayınlandığında ise yine veritabanında bulunan bu bilgiler ilgili kişilere istenilen biçimlerde aktarılır. Modelin çözümü sonucu elde edilen atama programı ilgili tüm kullanıcılara yayınlanır. Sistemin genel işleyişi Şekil 5.3 ile gösterilmiştir.



Şekil 5.3 EZÇ-KDS'nin genel yapısı

Karar destek sistemlerinde her olay bir tetikleyici ile başlar ve bir kaynak tarafından yürütülür. Olay ile ilgili bir etkinliğin gerçekleşmesinin ardından çıktısı (tepki) kullanıcılara yani hedefe sunulur. Otomatik olarak yapılan etkinlikler için sistemin kendisi kaynak olarak kabul edilir. EZÇ-KDS'nin gözetmen-sınav atama modülünde gerçekleştirilen tüm olayların kapsamlı bir açıklaması Çizelge 5.1'de verilmiştir.

**Çizelge 5.1** Gözetmen-sınav atama modülünde gerçekleştirilen olaylar tablosu

Olay	Tetikleyici	Kaynak	Etkinlik	Çıktı	Hedef
1. Yönetici sınav-derslik zaman çizelgesi ve sınav ağırlıklarının girişini yapar.	Sınav-derslik-zaman çizelgesinin yayınlanması	Yönetici	Veri tabanında “Sınavlar” tablosunun oluşturulması	Sınav ağırlıkları Sınavların zaman dilimleri	
2. Yönetici sınavlarda görev alacak gözetmenle ile bu gözetmenlerin yük oranlarını belirler.	Yeni sınav döneminin yayınlanması	Yönetici	Gözetmenlerin seçilmesi	Gözetmenler Yük oranları	
3. Yönetici her sınav için gerekli gözetmen sayısını belirler.	Yeni sınav döneminin yayınlanması	Yönetici	Gözetmen sayısının belirlenmesi	Gerekli gözetmen sayısı	
4. Sistem, kısıt ve tercih giriş ekranlarını oluşturur.	Sınav-derslik-zaman çizelgesinin girilmesi	Sistem	Sınav-derslik-zaman çizelgesine bağlı olarak arayüzün oluşturulması		
5. Gözetmenler, kısıt ve tercihlerinin girişini yaparlar.	Kısıt ve tercih girişlerinin bildirilmesi	Gözetmen	Kullanıcı adı ve şifrelerini kullanarak gözetmenlerin kendi bilgilerini girmesi	Ön-atamalar Tercihler Kısıtlar	
6. Sistem uygunluk kontrolünü yapar.	Veri girişinin tamamlanması	Sistem	Gerekli olan ve görevlendirilmiş olan gözetmen sayılarının karşılaştırılması	Uygun olmayan zaman dilimleri	Yönetici
7. Yönetici, kaynakların yeterliliğini kontrol eder.	Kaynakların sorgulanması	Yönetici		Kaynakların detayları	
8. Yönetici, ilave kaynaklar sağlar.	Uygun olmama durumunun tespit edilmesi.	Yönetici	Yeni kaynakların ilave edilmesi	Gözetmen listesi	
9. Sistem, matematiksel modeli oluşturur.	Uygunluk durumunun tespit edilmesi	Sistem	Elde edilen parametrelere göre matematiksel modelin oluşturulması	Model	
10.Yönetici modeli çözdürür.	Modelin oluşturulması	Yönetici	En iyilleme paketini kullanarak modelin çözülmesi.	Atama programı.	
11.Yönetici, atama programını yayınlar.	Sonucu elde etmek	Yönetici	Atama programının ilgili kullanıcılara yayınlanması	Program	Gözetmenler Öğreticiler Öğrenciler

### 5.3.1 Veri yönetimi

EZÇ-KDS'nin veritabanı bileşeni MsAccess'te geliştirilmiştir. Sistemde ele alınan tüm problemlere ilişkin izleyen tablolar oluşturulmuştur:

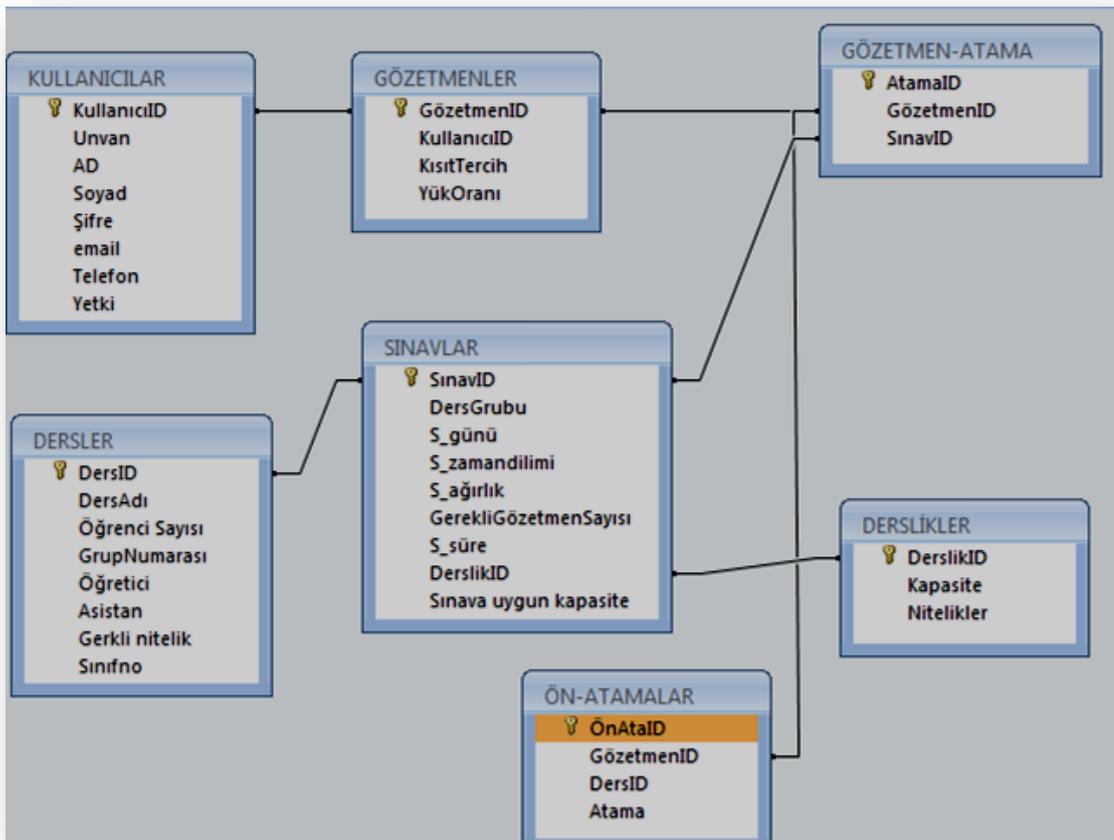
- Ders-derslik-zaman çizelgeleme problemi için geliştirilen matematiksel modelin parametreleri; derse kayıtlı öğrenci sayısı, derse veren öğretici, dersin yapıldığı derslik, dersin gerektirdiği nitelikler ile derslik kapasiteleri ve derslikte bulunan nitelik verileridir. Bu bilgiler bir metin dosyasında ve ayrıca veritabanında yer alan DERSLER ve DERSLİK tablolarında tutulmaktadır.
- Sınav-derslik-zaman çizelgeleme problemi için geliştirilen matematiksel modelin parametreleri; veritabanında saklanan sınav, gözetmen ve diğer gerekli bilgilerden oluşmaktadır. Bu bilgiler, DERSLER, DERSLİKLER, GÖZETMENLER ve SINAVLAR tablolarında tutulmaktadır. Şekil 5.4'te DERSLER tablosu ve ilişkili olduğu SINAVLAR tablosuna ait veriler verilmiştir.

DersID	DersAdı	Öğrenci	GrupNum	Öğre
151312123A	END.MÜH.GİRİŞ	45	5	Yrd.Doç.Dr. İNCİ SARIÇ
71	SınavID	S_günü	S_zamandı	Derslik
71	31.05.2007	11:00	End11	1
*	GereklGözet	S_ağırlık	Grup	
*	1	2	28	24
151312123B	END.MÜH.GİRİŞ B	42	5	Öğr.Gör.Dr. ŞERAFETTİ
151312131A	INTRODUCTION TO IND.ENG.	7	5	Yrd.Doç.Dr. AYKUT AR
151312132A	EKONOMİ	96	4	Yrd.Doç.Dr. FÜSUN YEİ
151312133A	ECONOMICS	12	4	Yrd.Doç.Dr. ERDAL GÜİ
151312182A	TÜRK DİLİ II	80	33	Öğr.Gör. OKTAY YİVLİ
15	SınavID	S_günü	S_zamandı	Derslik
15	24.05.2007	11:00	Fizik Lab.	2
*	GereklGözet	S_ağırlık	Grup	
*	2	2	6	5
151312186A	Seminer II ( Yab. Uyr. Öğr. )	1	25	Doç.Dr. EMİN KAHYA
151312196A	FİZİK II	70	8	Yrd.Doç.Dr. ÖMER ÖZE
151312197A	FİZİK ILLAR	66	8	Yrd.Doç.Dr. ÖMER ÖZE

Şekil 5.4 Dersler tablosu

- Gözetmen-sınav atama problemi için geliştirilen matematiksel modelin parametreleri ise; veritabanında saklanan sınav ağırlığı, gerekli gözetmen sayısı, gözetmen yük oranı, kısıt ve tercihler ile diğer gerekli bilgilerden oluşmaktadır. Bu bilgiler KULLANICILAR, GÖZETMENLER, SINAVLAR, DERSLİKLER, GÖZETMEN-ATAMA ve ÖNATAMALAR tablolarında tutulmaktadır.

Veritabanında yer alan tüm tablolar ve aralarındaki ilişkiler Şekil 5.5'te verilmiştir.



Şekil 5.5 Veritabanında yer alan ilişkiler

### 5.3.2 Diyalog yönetimi

Macromedia Flash yazılımı kullanılarak geliştirilen kullanıcı arayüzleri, kullanıcıların sistem parametrelerini tanımlamasını, ilgili problemin modelinin kurularak çözülmesini ve çözüm raporlarını almasını sağlar. Sınavlar, dersler, derslikler, zaman dilimleri, öğretici ve gözetmenler ile ilgili uygun veri dosyaları MsAccess'te tablolarda oluşturulmuştur. Dolayısıyla veri, arayüzler ile toplanır ve veritabanına XML ve SQL sorguları kullanılarak gönderilir.

Kullanıcılar veri giriş, düzenleme ve görüntüleme işlemlerini yetkileri dahilinde pop-up menüler kullanarak yapmaktadır. Kullanıcının yönetici ya da gözetmen olmasına bağlı olarak ilgili sayfa ekrana gelmektedir. Kullanıcılar, kullanıcı adı ve şifrelerini Şekil 5.6'da verilen ekrandan sisteme giriş yapmaktadır.

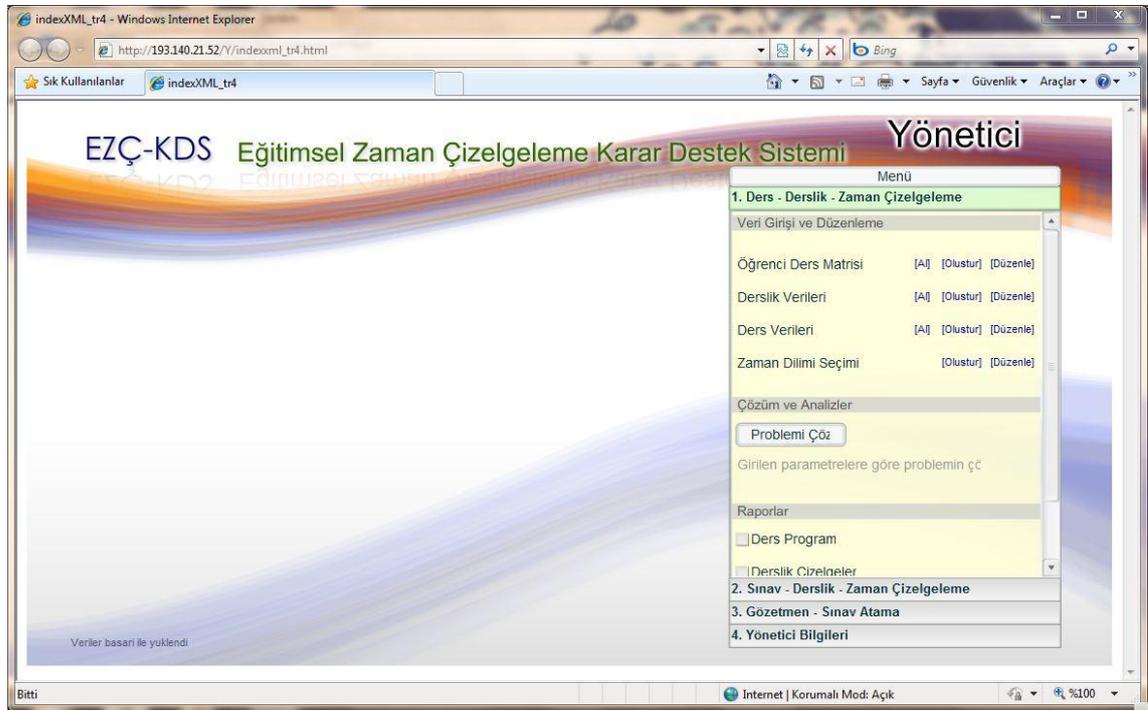


**Şekil 5.6** Sistem giriş ekranı

EZÇ-KDS ile yönetici; ders-derslik-zaman ve sınav-derslik-zaman çizelgeleme ile gözetmen-sınav atama problemlerine ilişkin gerekli veri girişi, düzenlenmesi ve

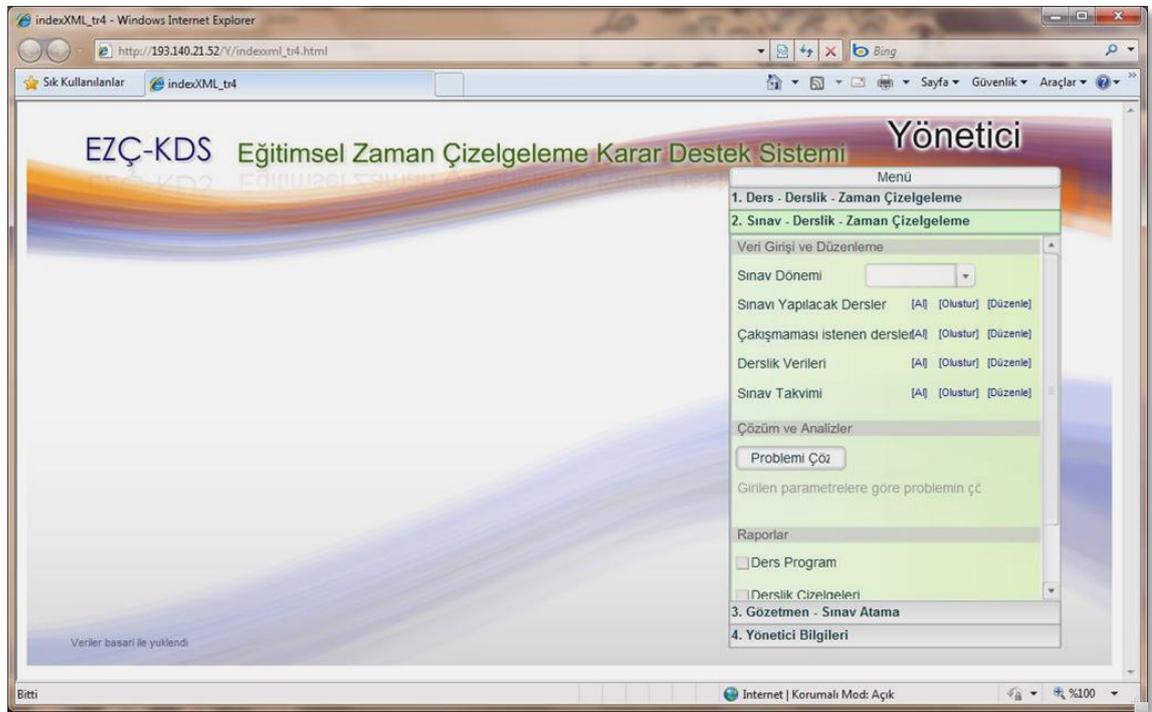
görüntülenmesini yapabilmektedir.

Ders-derslik-zaman çizelgeleme problemi için geliştirilen matematiksel model ve GA'nın parametrelerinden Öğrenci-Ders matrisi, Derslik-Nitelik matrisi ve Ders-Nitelik matrisi bir metin dosyasında tutulmaktadır. Yönetici bu verileri Şekil 5.7'de verilen arayüz aracılığı ile ilgili metin dosyasından alabilir, yenilerini oluşturabilir ya da mevcut dosyalar üzerinde düzenlemeler yapabilir. Probleme ait derslik bilgileri ayrıca veritabanında yer alan DERSLER ve DERSLİK tablolarında tutulmaktadır. Bu parametreler, derse kayıtlı öğrenci sayısı, dersi veren öğretici, dersin yapıldığı derslik, dersin gerektirdiği nitelikler ile derslik kapasiteleri ve derslikte bulunan nitelik verileridir. Yönetici *problemi çöz düğmesine* tıklayarak ilgili parametreleri modele gönderir ve modeli çözdürür. Çözümüne ilişkin raporları da farklı şekillerde yayınlatabilir.



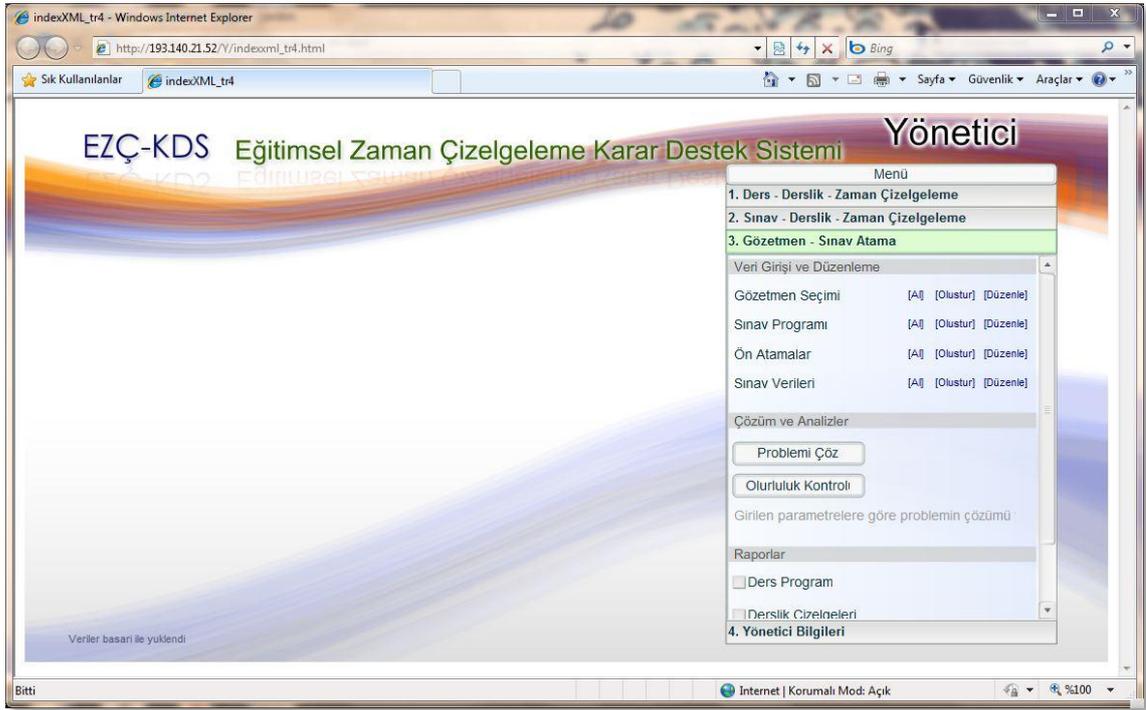
Şekil 5.7 EZÇ-KDS yönetici ekranı

Yönetici sınav-derslik-zaman çizelgesinin oluşturulması için Şekil 5.8’de verilen arayüzden ilgili sınav dönemini seçerek sınavı yapılacak dersleri ve derslik kapasitelerini belirler. Yanısıra, sınavlarının çakışması istenmeyen ders ikililerinin seçimini yapabilir. Probleme ait veriler SINAVLAR tablosuna aktarılır, yönetici ilgili ekranda problemi çöz düğmesine tıklayarak ilgili parametreleri modele gönderir ve modeli çözdürür. Çözümüne ilişkin raporları da diğer kullanıcılara yayımlayabilir.



Şekil 5.8 Sınav-derslik-zaman çizelgeleme yönetici arayüzü

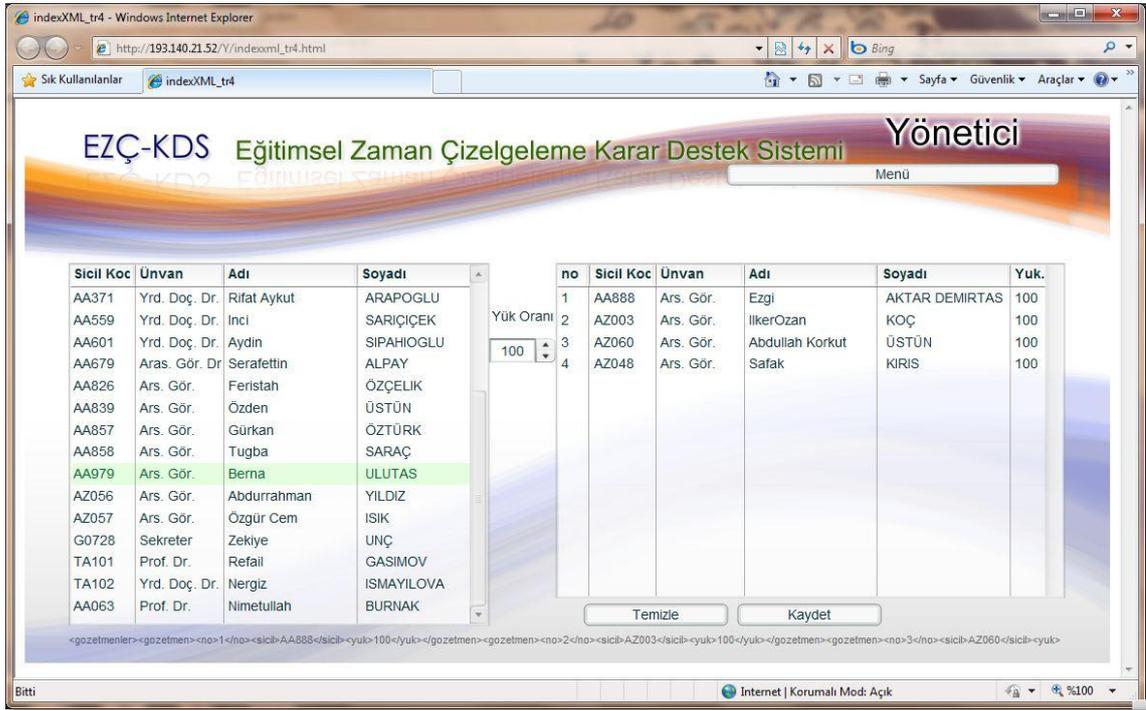
Gözetmen-sınav atama problemine ilişkin yönetici arayüzü ise Şekil 5.9’da verilmiştir. Gözetmen ve sınavlara ait veriler yönetici tarafından sisteme girilmekte veya sistemde yer alan mevcut veriler düzenlenmektedir.



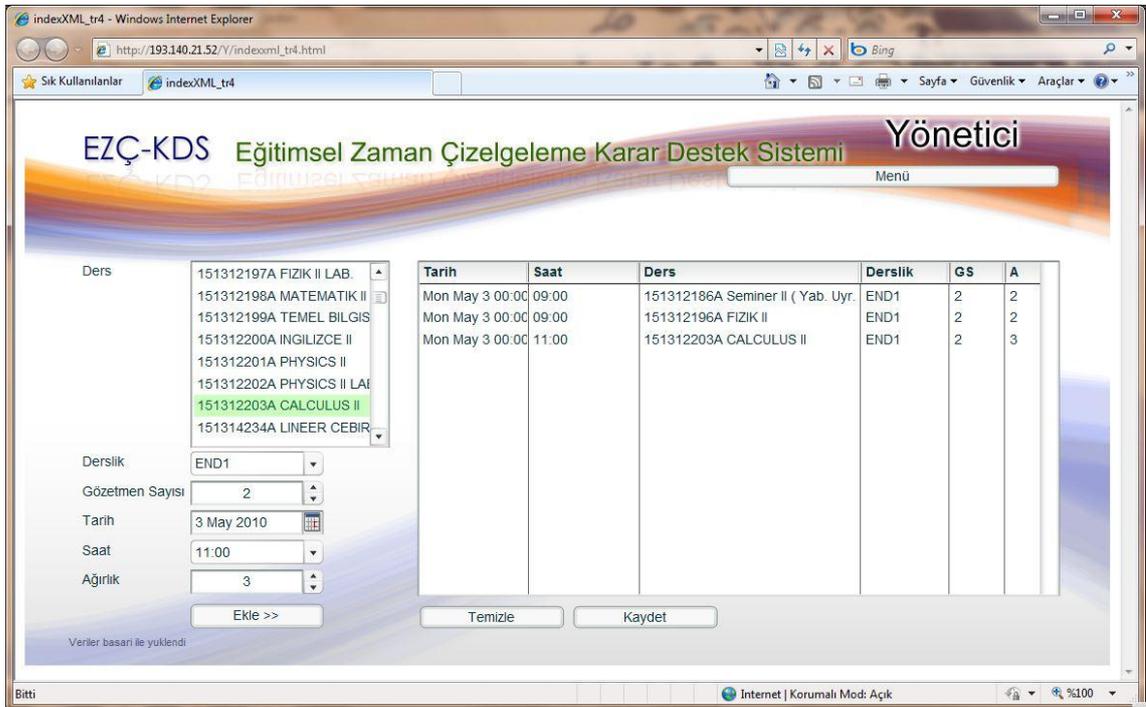
**Şekil 5.9** Gözetmen-sınav atama problemi için ana yönetici ekranı

Sınavlarda görev alacak gözetmenlerin listesi Şekil 5.10’da verilen arayüzde oluşturulmaktadır. Sistemde yer alan tüm kullanıcılar arasından ilgili sınav döneminde gözetmenlik yapacak olanlar, deneyimlerine bağlı olarak, yük oranları ile birlikte listeye alınırlar. Yük oranı seçimi ile bir başka sınav döneminde gözetmen yük oranlarının değiştirilmesi de mümkün kılınmıştır.

Şekil 5.11’de verilen ekranda yönetici hangi dersin sınavının hangi derslikte ve hangi gün hangi zaman diliminde yapılacağı bilgisinin girişini yapmaktadır. Yanısıra, sınav için gerekli gözetmen sayısı ve sınavların ağırlıkları da bu ekranda çizelgeye girilmektedir. Şekilde verilen örnekte, Calculus dersinin sınavı 3 Mayıs 2010 pazartesi günü END1 dersliğinde ve 2 gözetmen desteği ile yapılacaktır.

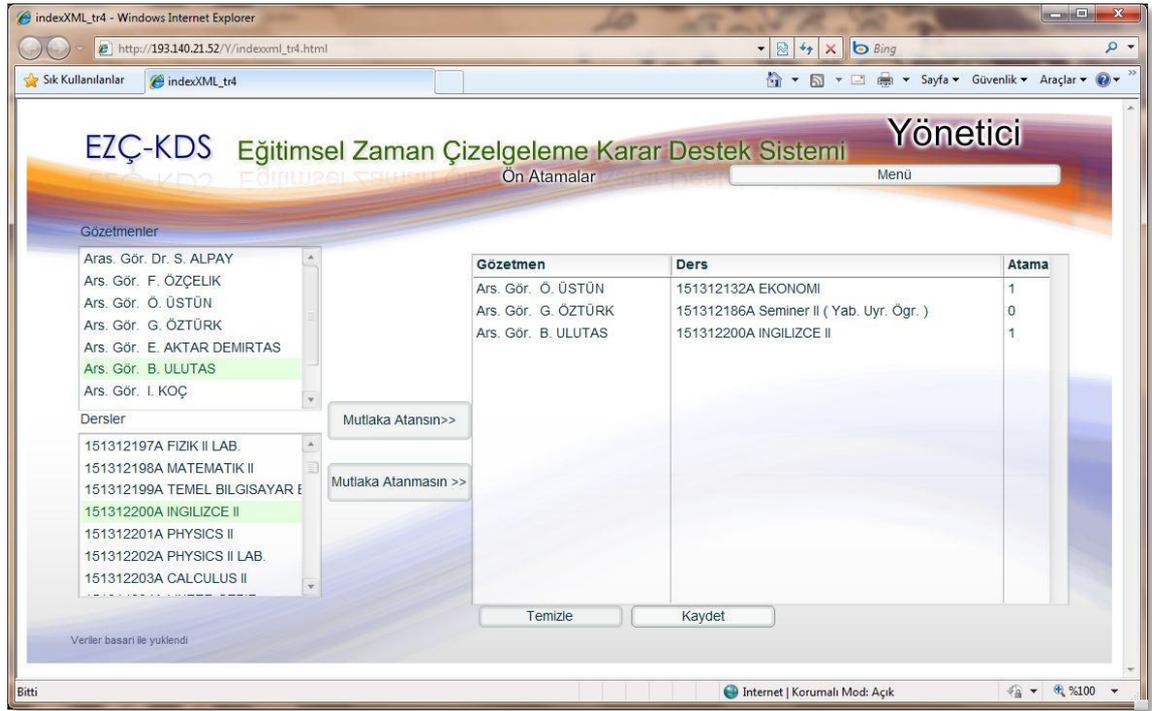


Şekil 5.10 Gözetmen seçimi ekranı



Şekil 5.11 Sınav-derslik-zaman çizelgesinin giriş ekranı

Şekil 5.12’de verilen arayüzde ise yönetici, bazı gözetmenleri atama modelini çözdürmeden önce belirli derslerin sınavlarına atayabilir. Bu durumda gözetmen ve dersler listesinden seçimler yapılarak ön atamalar gerçekleştirilmektedir.



Şekil 5.12 Ön atamaların yapıldığı arayüz

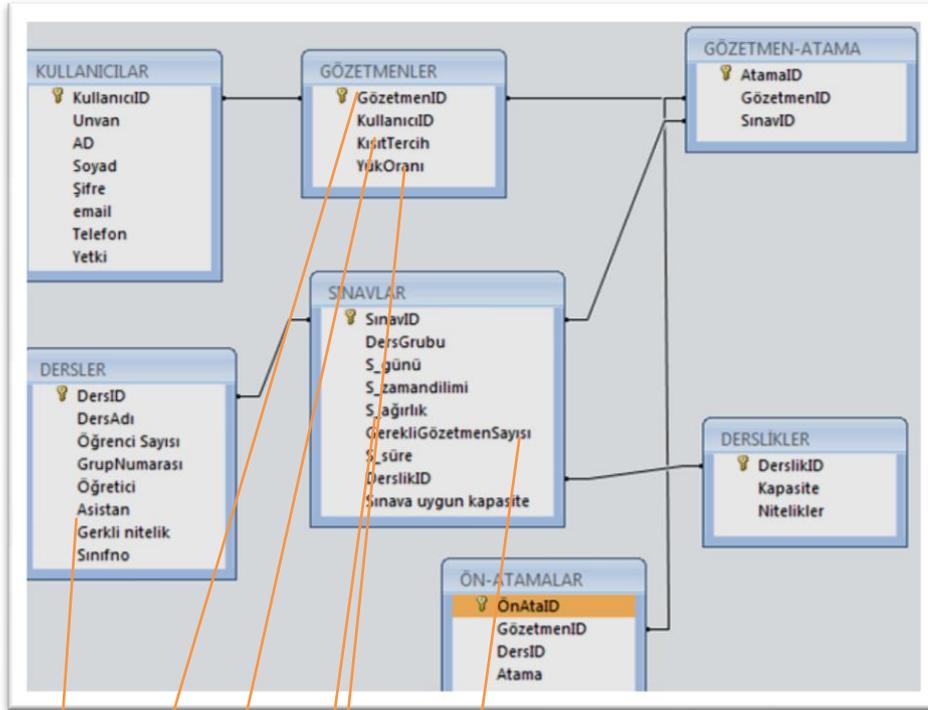
Gözetmen-sınav atama sürecinde, yöneticinin yanısıra gözetmenler de zaman dilimlerine ilişkin kısıt ve tercihlerini kendileri için hazırlanan ve Şekil 5.13 ile verilen arayüz yardımıyla sisteme girmektedir. Kısıt ve tercih girişinde kullanılan simgeler ve anlamları Çizelge 5.2’de verilmiştir.

Şekil 5.13 Gözetmenler için tercih ve kısıt giriş ekranı

Çizelge 5.2 Kısıt ve tercih girişinde kullanılan simgeler

Simge	Anlamı
	Bu zaman diliminde sınava girmeyi tercih etmiyorum
	Bu zaman diliminde sınava girmek istemiyorum
	Bu zaman diliminde kesinlikle sınava giremem

Gözetmenlerin kısıt ve tercih girişlerinin, tercihler ve kısıtlar girildiğinde bir sınavın atanacağı saate yeterince gözetmen kalıp kalmadığının kontrolü vb. olurluluk kontrolü sistem yöneticisi tarafından yapılabilmektedir. Probleme ilişkin tüm veri girişinin tamamlanmasının ardından yönetici *problemi çöz* düğmesine tıklayarak veritabanındaki verinin GAMS'te yazılmış modele Şekil 5.14'te gösterildiği gibi aktarılmasını sağlamaktadır.



```
sets
i   gozetmen      /1*13/
j   sinav         /1*32/
t   zamanDilimi  /1*27;/

parameters
g(j) j. sinavin ihtiyac duydugu gozetmen sayisi
/1 5, 2 1,3 5, 4 5, 5 4, 6 5, 7 6, 8 4, 9 5, 10 2, 11 1, 12 1, 13 1, 14 4
15 6, 16 5, 17 5, 18 5, 19 4, 20 5, 21 5, 22 3, 23 5, 24 2, 25 4, 26 4, 27 5
28 5, 29 5, 30 6, 31 1, 32 2/

a(j) j. sinavin agirligi
/1 3, 2 2,3 3, 4 3, 5 3, 6 2, 7 3, 8 2, 9 2, 10 3, 11 3, 12 3, 13 3, 14 4
15 4, 16 3, 17 2, 18 3, 19 3, 20 3, 21 3, 22 2, 23 2, 24 2, 25 2, 26 3, 27 3
28 2, 29 3, 30 3, 31 3, 32 2/

ss(j) j.sinavin atandigi zaman dilimi
/1 1, 2 2,3 3, 4 4, 5 4, 6 5, 7 6, 8 7, 9 8, 10 9, 11 9, 12 9, 13 9, 14 10
15 11, 16 12, 17 13, 18 14, 19 15, 20 16, 21 17, 22 18, 23 19, 24 20, 25 21, 26 22, 27 23
28 24, 29 25, 30 26, 31 27, 32 27/

s(j,t) j.sinav t. zaman diliminde yapiliyorsa 1 dd 0
w(i) i. gozetmenin agirligi

p(i,j) /1.29 =1, 2.1 = 1, 7.1 = 1, 7.6 = 1, 3.7 = 1, 1.8 = 1, 6.22 = 1, 8.23 = 1
10.24 = 1, 4.25 = 1, 10.31 = 1, 9.30 = 1 /

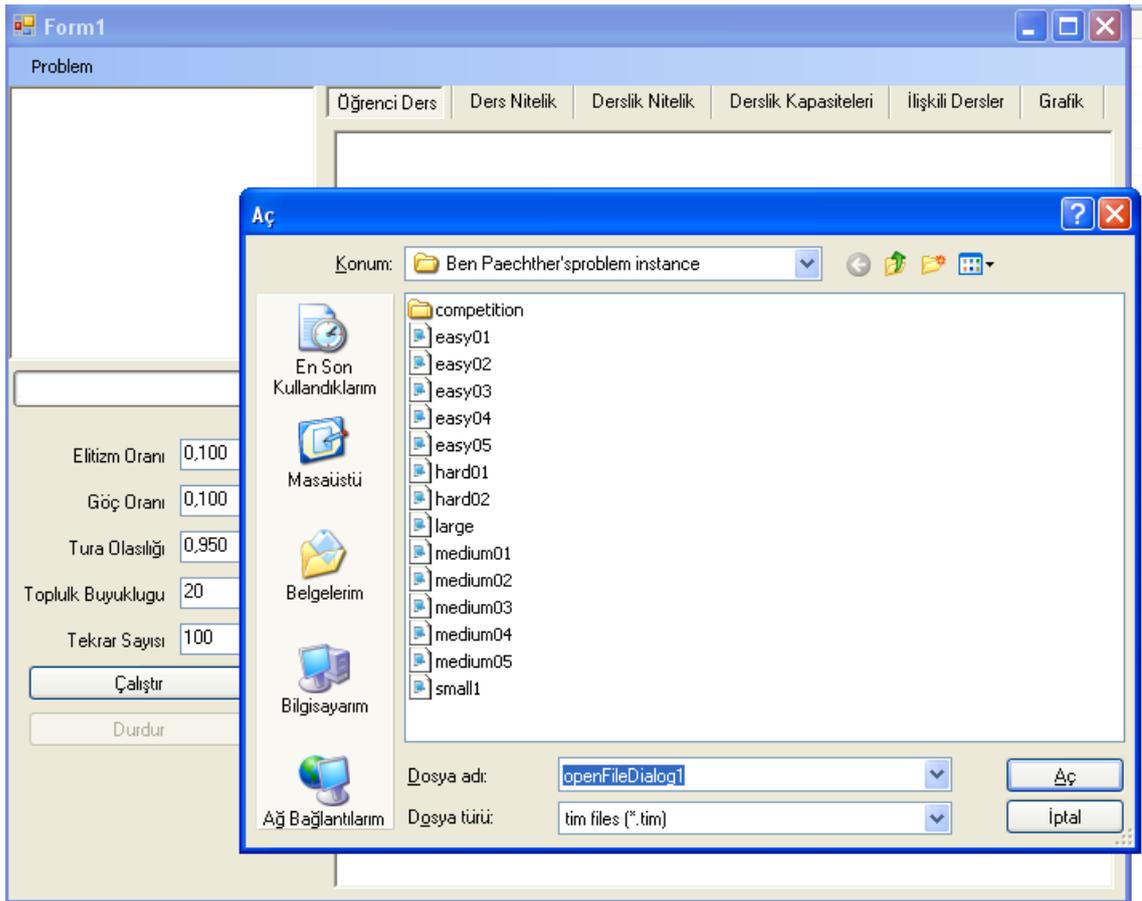
m(i,t) j. gozetmen t. zaman dilimine atanamaz
/12.6=1, 12.13=1, 6.13=1/

c(i,t) i. gozetmenin t. zaman dilimini tercih etmeme duzeyi
/12.2=3, 12.5 = 3, 12.6=3, 12.16=3, 12.13=3, 12.18=3, 12.20=3, 12.22=3, 12.24=3
5.1=2, 5.7=2, 5.10=2, 5.12=2, 5.17=2, 5.21=2, 5.26=2
6.5=3, 6.9=3, 6.16=3, 6.20=3, 6.24=3 /
```

Şekil 5.14 Veritabanından modele veri gönderimi

### 5.3.3 Model yönetimi

EZÇ-KDS'nin model tabanını ders-derslik-zaman çizelgeleme problemi için 3. Bölüm'de geliştirilen matematiksel model ve genetik algoritmalar; sınav-derslik-zaman çizelgeleme problemi için Bölüm 4.1'de geliştirilen matematiksel model ve sınav-gözetmen atama probleminin çözümü için Bölüm 4.2'de geliştirilen matematiksel model oluşturmaktadır. Her problemin çözümünde yöneticinin problemi çözdür komutunu vermesi ile birlikte ilgili model çalışmaktadır. Model yönetimine örnek olarak Şekil 5.15 ile verilen ekranda ders-derslik-zaman çizelgesinin oluşturulmasında çalıştırılan GA'nın program ekranı verilebilir.



Şekil 5.15 Ders-derslik-zaman çizelgeleme problemi için GA çözüm ekranı

## SONUÇ VE ÖNERİLER

Çalışma kapsamında, üniversitelerde karşılaşılan eğitimsel zaman çizelgeleme problemleri genel anlamda bireylerin faaliyet veya kaynaklara, faaliyetlerin ise yine kaynaklara atandığı alt problemler olarak sınıflanarak, her grubun önemli ama olabildiğince çok bileşenli bir örnek problemi için eniyi çözüm yöntemleri geliştirilmiştir. Bu amaçla, ders-derslik-zaman ve sınav-derslik-zaman çizelgeleme ile gözetmen-sınav atama problemleri ele alınmıştır. Her problem ayrı ayrı çözülerek, literatürde yer alan varsa mevcut eniyi çözümü ile karşılaştırılmıştır. Tanımlanan bu problemlerin kullanıcı tarafından arayüzler aracılığıyla kolayca çözümüne olanak verecek bir KDS tasarlanmış, kullanımı bir problem için gösterilmiştir. Ele alınan problemlerin hangi başka gerçek hayat problemlerine benzetilerek çözümlerine esas olacağı ve yaygın etkisi açıklanmıştır. İzleyen kesimlerde bahsedilen kazanımlar ayrıntılandırılmaktadır.

Çalışmada öncelikle literatürde yer alan 196 eğitimsel zaman çizelgeleme çalışması ele alınan bileşenleri ve çözüm yaklaşımlarına göre ayrıntılı olarak incelenmiştir. Bu çalışmalarda problemlerin farklı bileşenleri içermelerine rağmen aynı başlık altında ele alınmış olmaları, terminolojide henüz bir standartlaşmanın olmadığını göstermiştir.

Literatürde yer alan problemlerin birçoğuna ek olarak, ders-derslik-zaman çizelgeleme probleminde derslik kapasitesi ve derslerin ihtiyaç duyduğu nitelikleri içeren dersliklere atanması kısıtları da ele alınarak problemin boyutu daha da artırılmıştır. Parametre ile sıkı ve esnek kısıtları [Ben Paechter](#) tarafından tanımlanan ders-derslik-zaman çizelgeleme problemi için bir karma tamsayıli matematiksel model geliştirilmiştir. Ele alınan problem, çok fazla çalışılmasına rağmen özellikle belirlenmiş esnek kısıtları ile birlikte bir matematiksel modeline literatürde rastlanmamıştır. Geliştirilen model ile test problemleri çözdürülmüş, GAMS CPLEX çözücüsü ile 100 ders ve 80 öğrencinin 45 zaman dilimine atandığı literatürde yer alan *small* tipi beş problem için tüm esnek kısıtların sağlandığı en iyi çözümler elde edilmiştir. Literatürde

*small* sınıfında yer alan bazı test problemleri için en iyi çözümü veren metasezgisel algoritmalar bulunmakla birlikte, bu metasezgiseller ile *medium* ve *hard* sınıfındaki problemlerin en iyi çözümleri elde edilememiştir.

400 ders ve 400 öğrencinin 45 zaman dilimine atandığı daha büyük boyutlu problemler için, önerilen model ile 96 saat çalışmanın ardından GAMS programı uygun bir çözüm bulamamıştır. Bu nedenle, literatürde yer alan büyük boyutlu problemlerin çözümünde daha hızlı bir şekilde çözümlere ulaşma ihtiyacı ile yeni bir çözüm yaklaşımı geliştirilmiştir. Eğitimsel zaman çizelgeleme problemlerinin genetik algoritmalar ile çözümünde karşılaşılan sakıncaları ortadan kaldırmak amacıyla özel genetik işlemciler ile onarma işlemcilerine ihtiyaç duymayan, tüm kaynakları aynı anda ele alabilen ve her çalıştırmada uygun çözümü garantileyen rassal anahtar temelli bir GA olan RKGA algoritması geliştirilmiştir. RKGA, her çalıştırmada tüm sıkı kısıtları sağlayarak ve esnek kısıtları da olabildiğince sağlayarak literatürde yer alan diğer yaklaşımlarla rekabet edebilir düzeye getirilmiştir.

Ayrıca, problemin çok amaçlı yapısından dolayı RKGA algoritması, bir çok amaçlı GA olan NSGA-II algoritması ile yeniden kodlanmıştır. Ancak tüm çalıştırmalarda sıkı kısıtların sağlanmasının yanında, RKGA algoritmasında elde edilen sonuçlar kadar iyi sonuçlar elde edilememiştir.

Çalışmada ele alınan ikinci problem, sınav-derslik-zaman çizelgeleme problemi olmuştur. Literatürde yer alan çalışmaların birçoğuna ek olarak bir sınavın birden fazla derslikte yapılabilmesini sağlayacak ve özellikle düzensiz öğrenciler için sınav çakışmalarını mümkün olduğunca engelleyecek kısıtlar ile mümkün olduğunca sağlanması istenen dört esnek kısıtın eklenmesi ile problem için daha genel bir model geliştirilmiştir. Geliştirilen model ile bir gerçek problem için en iyi çözüm GAMS CPLEX çözücüsü ile elde edilmiştir. Elde edilen sonuçlara göre, kullanılan kaynak sayısında azalma olmaktadır. Kullanılan derslik sayısının azalması ile sınavlara atanacak gözetmen sayısında da azalma olacaktır. Ayrıca, oluşabilecek sınav çakışmalarının sebep olduğu ek iş yükü de olabildiğince engellenmiştir.

Çalışmada ele alınan bir diğer eğitimsel zaman çizelgeleme problemi de literatürde henüz çok fazla ele alınmamış gözetmen-sınav atama problemi olmuştur. Önceki çalışmalarda gözetmenlerin zaman dilimi ve gün tercihleri olabildiğince karşılanmaya çalışılmış, geliştirilen çözüm yaklaşımları ile tüm problemlerde en iyi çözüm elde edilememiştir. Bu çalışmada ise gözetmenlerin zaman dilimlerine ilişkin istek ve kısıtlarını dikkate alan ve yanı sıra sınav özelliklerine bağlı olarak gözetmen yüklerinin dengeli atandığı çok amaçlı problem için bir matematiksel model geliştirilmiştir (Öztürk vd., 2010). Modelin çözümü gerçek iki problem üzerinde araştırılmış ve GAMS CPLEX çözücüsü ile en iyi çözümler elde edilmiştir. Mevcut durum ile karşılaştırmalar ve elde edilen iyileşmeler irdelenmiştir.

Bir karar probleminde sonucun parametrelere bağımlı olması, sözkonusu parametrelerin sağlıklı tahmin edilmesini gerektirir. Gözetmen-sınav atama probleminin daha gerçekçi ve sistematik bir şekilde ele alınabilmesi için, problem parametrelerinin belirlenmesine yönelik farklı uzman görüşleri alınarak AHP (Ozturk and Sagir, 2008) ve ANP modelleri geliştirilmiştir. Geliştirilen çok ölçütlü modeller ile problemi etkileyen faktörlerin neler olduğunun belirlenmesinin yanı sıra, aynı zamanda bu faktörler ve problem amaçları ağırlıklandırılmıştır. Bu ağırlıklar, gözetmen-sınav atama problemi için geliştirilen çok amaçlı karma tamsayı matematiksel modelin parametrelerini ağırlıklandırmak için kullanılmıştır. Elde edilen yeni ağırlıklar ile, geliştirilen gözetmen-sınav atama problemi yeniden çözdürülmüş ve en iyi çözümler elde edilmiştir.

Son olarak, çalışmada ele alınan tüm problemleri web tabanlı, kullanıcı etkileşimli ve esnek bir karar destek ortamında çözecek otomatikleştirilmiş bir sistemin (EZÇ-KDS) tasarımı gerçekleştirilmiştir. Tasarlanan sistemde, problemler için gerekli veritabanı ile kullanıcıların tercihlerini girebileceği ve yöneticinin problemler ile ilgili çeşitli görevleri yerine getireceği arayüzler oluşturulmuştur.

EZÇ-KDS ile gözetmenler kendi arayüzlerinden kısıt ve tercihlerini veritabanı ve matematiksel modele aktarabilmektedirler. Böylece, hem kullanıcı tercihlerini olabildiğince karşılamaya çalışan hem de gözetmenlerin yüklerini dengeleyen bir sistem

geliştirilmiştir. Yöneticinin girdiği sınav programına ve gözetmenlerin kendi girdikleri kısıt ve tercihlerine bağlı olarak otomatik bir şekilde oluşturulan matematiksel model ile problemin en iyi çözümüne ulaşılmaktadır. EZÇ-KDS'nin web tabanlı olarak geliştirilmesi izleyen kazanımları da beraberinde getirmektedir:

- Birebir etkileşimli kullanıcı arayüzleri ile yapılan veri girişleri ile, yazılı belge aktarımı gibi işlemleri yavaşlatan unsurlar ortadan kalkar.
- Sistemin web tabanlı olması, kullanıcıların istek ve kısıtlarının 7 gün 24 saat boyunca girilmesine olanak sağlar.
- Yöneticiler gerek duydukları tüm bilgiye web ortamında en ayrıntılı şekilde erişebilir. Örneğin, gözetmenler ile tek tek görüşmek yerine her birinin kısıt ve isteklerine tek bir noktadan hızlı bir şekilde erişebilirler.
- Problemin çözümü için her bilgisayarda ilgili yazılımın kurulu olması gerekliliği yoktur. Sunucu üzerinden yetkisi olan kişiler herhangi bir uzmanlık gerektirmeden problemi çözdürebilir.

Çalışmada ele alınan problemlerin, her eğitim kurumunda bir dönemde en az bir defa karşılaşılmaları sebebiyle yaygın etkisi vardır. Bu problemlerin çözümü için önerilen modeller, aşağıda belirtilen gerekçelerle, gerçek hayat problemlerine uygulanabilir olmaları nedeniyle önemlidir.

Orta öğretim kurumlarında karşılaşılan ders zaman çizelgeleme problemlerinde her sınıfın dersliğinin belirli (sabit) olduğu durumda, sınıfların derslerinin zaman dilimlerine atanması sözkonusudur. Dolayısıyla, orta öğretimlerde karşılaşılan bu problem üniversite ders-derslik-zaman çizelgeleme probleminin bir alt problemi gibi düşünülebilir.

Zaman diliminin belirli olduğu, merkezi sistemle yapılan sınavlar da çalışma kapsamında ele alınan sınav zaman çizelgeleme probleminin özel bir halidir. Çalışmada geliştirilen sınav-derslik-zaman çizelgeleme modeli, ÜDS, KPDS, AÖF vb. sınavlar için ülke genelinde öğrenci sayısı ve derslik kapasitelerini de dikkate alarak zorunlu ve

esnek kısıtlar çerçevesinde çözdürülebilir.

Zaman diliminin belirli olduğu yukarıda değinilen merkezi sınavlarda karşılaşılan gözetmen, salon başkanı, bina sınav sorumlusu vb. kişilerin, derslik ve binalara atanması problemi de çalışma kapsamında ele alınan gözetmen-sınav atama probleminin özel bir durumudur.

Gelecek çalışmalarda, ders-derslik-zaman çizelgeleme problemi için geliştirilen RKGGA'nın, performansını artırmaya yönelik olarak diğer metasezgiseller ile birlikte kullanımı araştırılabilir. Rassal anahtar düşüncesi uygun çözümler üzerinde çalışmaya olanak verdiğiinden, benzer yaklaşım, tek bir çözüm üzerinde iyileştirmeler yapan yasadaki arama ve tavlama benzetimi gibi metasezgiseller ile de kullanılabilir. Ayrıca, RKGGA, çözüm sırasında karar vericinin tercihlerini göz önüne alan etkileşimli bir yöntem ile birleştirilebilir. Son olarak, terminolojide standartlaşmaya katkı çalışmasının tamamlanması planlanmaktadır.

Sonuç olarak, bir eğitim döneminde en az iki defa karşılaşılan eğitimsel zaman çizelgeleme problemlerinin çözümü, yoğun iş gücü ve kaynak gerektirmektedir. Dolayısıyla, eğitim kurumları açısından önemli olduğu kadar da zor problemlerdir. Ayrıca, kısıt ve amaçları kurumdan kuruma farklılık gösterebilir. Bu durum, bu alandaki çalışmaların sürekliliğine neden olmaktadır. Tüm eğitim kurumlarının tüm farklı eğitimsel zaman çizelgeleme problemlerini çözebilecek genel bir sistem geliştirmek son derece zordur. Fakat bu çalışmada eğitimsel zaman çizelgeleme problemlerinde karşılaşılan üç temel problem için, pek çok kurum için anlamlı olabilecek esnek kısıtları da dikkate alan çözüm yaklaşımlarının bir KDS ortamında uzman bilgisinden bağımsız uygulanmasına olanak veren bir yapı tasarlanmıştır.

Gelinen noktanın literatürde bir eksikliği gidermeye destek ve ümit verici olduğu düşünülmektedir.

## KAYNAKLAR DİZİNİ

- Abdennadher, S., Saft, M. and Will, S., 2000, Classroom assignment using constraint logic programming, In: Proceedings of Second International Conference and Exhibition on The Practical Application of Constraint Technology and Logic Programming (PACLP 2000), Manchester, United Kingdom (10th - 12th April 2000).
- Abramson, D., 1991, Constructing school timetables using simulated annealing: sequential and parallel algorithms, *Management Science*, 37, 1, 98-113.
- Akkoyunlu, E.A., 1973, A linear algorithm for computing the optimum university timetable, *The Computer Journal*, 16, 4, 347-350.
- Alkan, A., and Özcan, E., 2003, Memetic algorithms for timetabling, *Proceedings of 2003 IEEE Congress on Evolutionary Computation*, 1796-1802.
- Alvarez-Valdes, R., Martin, G., and Tamarit, J.M., 1996, Constructing good solutions for the spanish school timetabling problem, *Journal of the Operational Research Society*, 47, 1203-1215.
- Alvarez-Valdes, R., Crespo, E. and Tamarit, J.M., 2000, Assigning students to course sections using tabu search, *Annals of Operations Research*, 96, 1-16.
- Alvarez-Valdes, R., Crespo, E., and Tamarit, J.M., 2002, Design and implementation of a course scheduling system using Tabu Search, *European Journal of Operational Research*, 137, 512-523.
- Al-Yakoob, S.M. and Sherali, H.D., 2006, Mathematical programming models and algorithms for a class-faculty assignment problem, *European Journal of Operational Research*, 173, 2, 488-507.
- Al-Yakoob, S.M. and Sherali, H.D., 2007, A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations, *European Journal of Operational Research*, 180, 3, 1028-1044.
- Al-Yakoob, S.M., Sherali, H.D. and Al-Jazzaf, M., 2010, A mixed-integer mathematical modeling approach to exam timetabling, *Computational Management Science*, 7, 1, 19-46.

### KAYNAKLAR DİZİNİ (devam)

- Asratian, A.S, and de Werra, D., 2002, A generalized class-teacher model for some timetabling problems, *European Journal of Operational Research*, 143, 531–542.
- Arntzen, H., and Lokketangen, A., 2005, A Tabu search heuristic for a university timetabling problem, In: Ibaraki T., Nonobe K., Yagiura M. eds., *Meta-heuristics: Progress as Real Problem Solvers, Selected Papers from the 5<sup>th</sup> Metaheuristics International Conference MIC 2003*, Chapter 3, 65-85.
- Aubin, J., and Ferland, J.A., 1989, A large scale timetabling problem, *Computers and Operations Research*, 16, 1, 67–77.
- Avella, P., D’Auria, B., Salerno, S., and Vasil’ev, I., 2007, A computational study of local search algorithms for Italian high-school timetabling, *Journal of Heuristics*, 13, 543–556, DOI: 10.1007/s10732-007-9025-3.
- Awad, R.M. and Chinneck, J.W., 1998, Proctor assignment at Carleton University. *INTERFACES*, 28, 2, pp.58-71.
- Azmat, C.S., and Widmer, M., 2004, A case study of single shift planning and scheduling under annualized hours: a simple three-step approach, *European Journal of Operational Research*, 153, 148–175.
- Azimi, Z.N., 2004, Comparison of metaheuristic algorithms for examination timetabling problem, *Applied Mathematics and Computation*, 161, 2, 337-354.
- Azimi, Z.N., 2005, Hybrid heuristics for examination timetabling problem, *Applied Mathematics and Computation*, 163, 705-733.
- Badri, M.A., 1996, A two-stage multiobjective scheduling model for faculty-course-time assignments, *European Journal of Operational Research*, 94,16–28.
- Badri, M.A., Davis, D.L., Davis, F.D., and Hollingsworth, J., 1998, A multi-objective course scheduling model: Combining faculty preferences for courses and times 1998, *Computers & Operations Research*, 25, 4, 303-316.
- Balakrishnan, N., Lucena, A., and Wong, R.T., 1992, Scheduling examinations to reduce second order conflicts, *Computers and Operations Research*, 19, 353-361.

### KAYNAKLAR DİZİNİ (devam)

- Bardadym, V.A., 1996, Computer-aided school and university timetabling: a new wave, The Practice and Theory of Automated Timetabling I: Selected Papers from 1<sup>st</sup> International Conference on the Practice and Theory of Automated Timetabling (PATAT I), Edinburgh, UK, Lecture Notes in Computer Science, 1153, Springer-Verlag. (Editors: E.K. Burke and P. Ross), 22-45.
- Bean, J.C., 1994, Genetic algorithms and random keys for sequencing and optimization ORSA, Journal on Computing, 6, 154–160.
- Beligiannis, G.N., Moschopoulos, C. and Likothanassis, S.D., 2007, A genetic algorithm approach to school timetabling, Journal of the Operational Research Society, 60, 23-42.
- Beligiannis, G.N. Moschopouloa, C.N., Kaperonisa, G.P., and Likothanassisa, S.D., 2008, Applying evolutionary computation to the school timetabling problem: The Greek case, Computers and Operations Research, 35, 4, 1265-1280.
- Beyrouthy, C., Burke, E.K., Landa-Silva, D., McCollum, B., McMullan, P., and Parkes, A.J., 2006, The Teaching Space Allocation Problem with Splitting, Lecture Notes in Computer Science, 3867, 228-247.
- Bilgin, B., Özcan, E., and Korkmaz, E.E., 2006, An experimental study on hyper-heuristics and exam timetabling, Lecture Notes in Computer Science, 3867, 394-412.
- Birbas, T., Daskalaki, S., and Housos, E., 1997, Timetabling for Greek high schools, Journal of the Operational Research Society, 48, 1191-1200.
- Bloomfield, S.D., and McSharry, M.M., 1979, Preferential course scheduling, Interfaces, 9, 4, 24-31.
- Blum, C., Correia, S., Dorigo, M., Paechter, B., Rossi-Doria, O., and Snoek, M., 2002, A GA evolving instructions for a timetable builder, PATAT 2002 Proceedings of the 4<sup>th</sup> international conference on the Practice And Theory of Automated Timetabling, Gent, Belgium, 120-123.
- Boland, N., Hughes, B.D., Merlot, L.T.G., and Stuckey, P.J., 2008, New integer linear programming approaches for course timetabling, Computers and Operations Research, 35, 7, 2209-2233.

### KAYNAKLAR DİZİNİ (devam)

- Bullnheimer, B., 1998, An examination scheduling model to maximize students' study time, *Lecture Notes in Computer Science* , 1408, 78-91.
- Burke, E.K., Elliman, D.G., and Weare, R., 1993, Extensions to a university exam timetabling system. In *IJCAI-93 Workshop on Knowledge-Based Production, Planning, Scheduling and Control*, Chambery, France, 42-48.
- Burke, E., Elliman, D. and Weare, R., 1994, A genetic algorithm based university timetabling system, *Proceedings of the 2<sup>nd</sup> East-West International Conference on Computer Technologies in Education (Crimea, Ukraine, 19th-23rd Sept 1994)*.
- Burke, E. K., Elliman, D.G., Ford, P. H. and Weare, R. F., 1996a, 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 first international conference*, Volume 1153 of *Lecture notes in computer science*. Springer-Verlag, Berlin, Heidelberg, 76-90.
- Burke, E.K., Newall, J.P. and Weare, R.F., (1996b), A memetic algorithm for university exam timetabling, *Lecture Notes in Computer Science*, 1153. *The Practice and Theory of Automated Timetabling: Selected Papers (ICPTAT '95)*, Burke, E.K., Ross, P. (eds), Springer-Verlag, Berlin, Heidelberg, New York, 241-250.
- Burke, E.K., Jackson, K., Kingston, J.H. and Weare, R.F, 1997, Automated university timetabling: the state of the art, *The Computer Journal*, 40, 9, 565-571.
- Burke, E.K., and Newall, J.P., 1999, A multi-stage evolutionary algorithm for the timetable problem, *IEEE Transactions on Evolutionary Computation*, 3, 1, 63-74.
- Burke, E., Bykov, Y., and Petrovic, S., 2001, A multicriteria approach to examination timetabling, In: E.K. Burke and W. Erben (eds), *Practice and Theory of Automated Timetabling: Selected Papers from the 3<sup>rd</sup> International Conference*, *Lecture Notes In Computer Science*, 2079, Springer-Verlag, Berlin, Heidelberg, 118–131.
- Burke, E.K., and Newall, J.P., 2002, A new adaptive heuristic framework for examination timetabling problems, *Computer Science Technical Report No.NOTTCS-TR-2002-1*.
- Burke, E.K. and Petrovic, S., 2002, Recent research directions in automated timetabling. *European Journal of Operational Research*, 140, 2, 266-280.

**KAYNAKLAR DİZİNİ (devam)**

- Burke, E.K., Kendall, G., Newall, J., Hart, E., Ross, P., and Schulenburg, S., 2003a, Hyper-heuristics: An emerging direction in modern search technology, In: Glover, F., Kochenberger, G. (Eds.), *Handbook of Meta-Heuristics*. Kluwer, 457–474.
- Burke, E. K., Kendall, G. and Soubeiga, E. (2003b) A tabu-search hyperheuristic for timetabling and rostering, *Journal of Heuristics*, 9, 451–470.
- Burke, E.K., MacCarthy, B.L., Petrovic, S., and Qu, R., 2003c, Knowledge discovery in a hyper- heuristic for course timetabling using case-based reasoning, *Lecture Notes in Computer Science*, 2740, 276-287.
- Burke, E.K, de Werra, D., and Kingston, J., 2004a, Applications to timetabling, In: *Handbook of Graph Theory*, Section 5.6, 445-474, CRC Press London.
- Burke, E., Bykov, Y., Newall, W., and Petrovic,S., 2004b, A time-predefined local search approach to exam timetabling problems, *IIE Transactions*, 36, 6, 509-528.
- Burke, E.K., and Newall, J.P., 2004, Solving examination timetabling problems through adaptation of heuristic orderings, *Annals of Operations Research*, 129, 107-134.
- Burke, E., and Silva, J.D.L, 2004, The design of memetic algorithms for scheduling and timetabling problems, *Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing*, 166, 289-312.
- Burke, E.K., Silva, J.D.L., and Soubeiga, E., 2005a, Multi-objective hyper-heuristic approaches for space allocation and timetabling, meta-heuristics: Progress as real problem solvers, *Selected papers from the 5<sup>th</sup> Metaheuristics International Conference MIC 2003*, Chapter 6, 129-158.
- Burke, E., Dror, M., Petrovic, S., and Qu, R., 2005b, Hybrid graph heuristics within a hyper-heuristic approach to exam timetabling problems, *Conference Volume of the 9th Informs Computing Society Conference*, 79-91.
- Burke, E.K., Petrovic, S., and Qu, R., 2006a, Case-based heuristic selection for timetabling problems, *Journal of Scheduling*, 9, 2, 115–132.
- Burke, E.K., MacCarthy, B. L., Petrovic, S., and Qu, R., 2006b, Multiple-Retrieval case-based reasoning for course timetabling, *Journal of Operations Research Society*, 57,148-162.

### KAYNAKLAR DİZİNİ (devam)

- Burke, E. K., Meisels, A., Petrovic, S., Qu, R. and McCollum, B., 2007, A graph-based hyper-heuristic for educational timetabling problems. *European Journal of Operational Research*, 176, 177–192.
- Burke, E.K., Marecek, J., Parkes, A.J., and Rudová, H., 2008, Penalising patterns in timetables: Novel integer programming formulations, *Operations Research Proceedings*, Berlin, Springer, Germany, ISSN 0721-5924, 2007, 409-414.
- Cappanera, P., and Gallo, G., 2004, A multicommodity flow approach to the crew rostering problem, *Operations Research*, 52, 4, 583–596.
- Caramia, M., Dell’Olmo, P., and Italiano, G.F., 2001, New algorithms for examination timetabling, *Proceedings of WAE 2000, Lecture Notes in Computer Science*, 1982, 230-241.
- Caramia, M., and Dell’Olmo, P., 2007, Coupling stochastic and deterministic local search in examination timetabling, *Operations Research*, 55, 2, 351–366.
- Carlson, R.C., and Nemhauser, G.L., 1966, Scheduling to minimize interaction cost, *Operations Research*, 14, 1, 52-58.
- Carlsson, C., and Fuller, R., 1995, Multiple criteria decision making: the case for interdependence, *Computers Operations Research*, 22, 3, 251-260.
- Carter, M.W., 1986, A survey of practical applications of examination timetabling algorithms, *Operations Research*, 34, 2, 193-202.
- Carter, M.W., and Tovey, C.A., 1992, When is the classroom assignment problem hard? *Operations Research*, 40, 1, 28-38.
- Carter, M. W. and Laporte, G., 1998, Recent developments in practical course timetabling. In E. Burke and M. Carter (eds.), *Practice and Theory of Automated Timetabling II*, *Lecture Notes in Computer Science*, Springer-Verlag New York, 1408, 3-19.
- Carter, M., Laporte, G., and Lee, S. Y., 1996, Examination timetabling: Algorithmic strategies and applications, *Journal of the Operational Research Society*, 74, 373-383.

### KAYNAKLAR DİZİNİ (devam)

- Casey, S., and Thompson, J., 2003, GRASPing the examination scheduling problem, Practice and Theory of Automated Timetabling: Selected Papers from the 4<sup>th</sup> International Conference, Lecture Notes in Computer Science, 2740, Springer-Verlag, Berlin, Heidelberg, 232-244.
- Cheong, C.Y., Tan, K.C., and Veeravalli, B., 2009, A multi-objective evolutionary algorithm for examination timetabling, Journal of Scheduling, 12, 2, 121 – 146. I
- Chierici, A., Cordone, R., and Maja, R., 2004, The demand-dependent optimization of regular train timetables, Electronic Notes in Discrete Mathematics, 17, 99–104.
- Chiarandini, M. Birattari, M., Socha, K., Rossi-Doria, O., 2006, An effective hybrid algorithm for university course timetabling, Journal of Scheduling 9, 403-432.
- Coello Coello, C.A., and Lamont, G.B., 2005, An introduction to multi-objective evolutionary algorithms and their applications, Coello Coello, Carlos A. Editor. Applications of Multi-Objective Evolutionary Algorithms. Chapter 1. River Edge, NJ, USA: World Scientific Publishing Company, Incorporated.
- Cole, A.J., 1964, The preparation of examination time-tables using a small-store computer, The Computer Journal, 7, 2, 117-121.
- Coloni, A., Dorigo, M., and Maniezzo, V., 1990, Genetic algorithms: a new approach to the timetable problem, NATO ASI Series, Vol.F82, Combinatorial Optimization Edited by M. Akgül et al. Springer-Verlag Berlin Heidelberg.
- Coloni, A., Dorigo, M., and Maniezzo, V., 1992, A genetic algorithm, to solve the timetable problem, Technical report 90.060, Politecnico di Milano, Italy.
- Cooper, T. B. and Kingston, J. H., 1996, The complexity of timetable construction problems, In the Practice and Theory of Automated Timetabling, ed. E. K. Burke and P. Ross, 283-295, Springer-Verlag.
- Corne, D., Ross, P., and Fang, H., 1994, Evolutionary timetabling: practice, prospects and work in progress, Presented at the UK Planning and Scheduling SIG Workshop, Strathclyde.
- Corne, D., Dorigo, M., and Glover, F., 1999, New ideas in optimization. Maidenhead, UK, England: McGraw-Hill Ltd.

**KAYNAKLAR DİZİNİ (devam)**

- Carrasco, M.P., and Pato, M.V., 2004, A comparison of discrete and continuous neural network approaches to solve the class/teacher timetabling problem, *European Journal of Operational Research*, 153, 65-79.
- Corr, P, McCollum, B., McGreevy, M., McMullan, P, 2006, A new neural network based construction heuristic for the examination timetabling problem, *PPSN IX, Lecture Notes In Computer Science*, 4193, 392-401.
- Costa, D., 1994, A tabu search algorithm for computing an operational timetable, *European Journal of Operational Research*, 79, 98-110.
- Cowling, P., Kendall, G. and Hussin, N.M., 2002, A survey and case study of practical examination timetabling problems, in *Proceedings of 4<sup>th</sup> International Conference of Practice and Theory of Automated Timetabling (PATAT2002)*, 258-261.
- Daniel, J.S.R. and Rajendran, C., 2006, Heuristic approaches to determine base-stock levels in a serial supply chain with a single objective and with multiple objectives, *European Journal of Operational Research*, 175, 566–592.
- Daskalaki, S., Birbas, T., and Housos, E., 2004, An integer programming formulation for a case study in university timetabling, *European Journal of Operational Research*, 153, 117–135.
- Daskalaki, S., and Birbas, T., 2005, Efficient solutions for a university timetabling problem through integer programming, *European Journal of Operational Research*, 160, 1, 106-120.
- Datta, D., Deb, K., and Fonseca, C.M., 2007, Multi-Objective evolutionary algorithm for university class timetabling problem, in *Evolutionary Scheduling* (K. P. Dahal, K. C. Tan, and P. I. Cowling, eds.), 49, ch. 8, 197-236.
- Dammak, A., Elloumi, A., and Kamoun, H., 2006, Classroom assignment for exam timetabling. *Advances in Engineering Software*, 37, 659-666.
- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., 2002, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6, 2, 182-197.
- de Gans, O.B., 1981, A computer timetabling system for secondary schools in the Netherlands, *European Journal of Operational Research*, 7, 175-182.

**KAYNAKLAR DİZİNİ (devam)**

- Deris, S., Omatu, S., Ohta, H., and Saad, P., 1999, Incorporating constraint propagation in genetic algorithm for university timetable planning, *Engineering Applications of Artificial Intelligence*, 12, 241-253.
- Deris, S., Omatu, S., and Ohta, H., 2000, Timetable planning using the constraint-based reasoning, *Computers & Operations Research*, 27, 819-840.
- de Werra, D., 1985, An introduction to timetabling, *European Journal of Operations Research*, 19, 151-162.
- de Werra, D., 1997, Restricted coloring models for timetabling, *Discrete Mathematics*, 165/166, 161-170.
- de Werra, D., 1997, The combinatorics of timetabling, *European Journal of Operational Research*, 96, 504-513.
- de Werra, D., Hertz, A., Kobler, D., and Mahadev, N.V.R., 2000, Feasible edge colorings of trees with cardinality constraints, *Discrete Mathematics*, 222, 61-72.
- di Gaspero, L., and Schaerf, A., 2000, Tabu search techniques for examination timetabling, *Lecture Notes In Computer Science*, 2079, 104-117.
- di Stefano, C. and Tettamanzi, A. G. B., 2001, An evolutionary algorithm for solving the school time-tabling problem, *Lecture Notes in Computer Science*, 2037, 452-462.
- Dimopoulou, M. and Miliotis, P., 2001, Implementation of a university course and examination timetabling system, *European Journal of Operational Research*, 130, 202-213.
- Dinkel, J.J., Mote, J., and Venkataramanan, 1989, An efficient decision support system for academic course scheduling, *Operations research*, 37, 6, 853-864.
- Drex1, A., and Salewski, F., 1997, Distribution requirements and compactness constraints in school timetabling, *European Journal of Operational Research*, 102, 193-214.
- Dowland, K.A., 1990, A timetabling problem in which clashes are inevitable, *Journal of Operations Research*, 41, 10, 907-918.

### KAYNAKLAR DİZİNİ (devam)

- Dowland, K.A., and Thompson, J., 2005, Ant colony optimization for the examination scheduling problem, *Journal of the Operational Research Society*, 56, 4, 426-438.
- Eley, M., 2007, Ant algorithms for the exam timetabling problem. *Lecture Notes in Computer Science*, 3867, 364-382.
- Eom, S. and Kim, E., 2006, A survey of decision support system applications (1995-2001), *Journal of the Operational Research Society*, 57, 11, 1264-1278.
- Erben, W. and Keppler, J., 1995, A Genetic algorithm solving a weekly course-timetabling problem, *Lecture Notes in Computer Science*, 1153, 198-211.
- Erben, W., 2001, A grouping genetic algorithm for graph colouring and exam timetabling, *Practice and Theory of Automated Timetabling: Selected Papers from the 3<sup>rd</sup> International Conference*, *Lecture Notes in Computer Science*, 2079, 132-156.
- Erben, W., and Song, P.Y., 2004, A hybrid grouping genetic algorithm for examination timetabling, In: E.K. Burke and M. Trick eds. *Proceedings of The 5<sup>th</sup> International Conference on the Practice and Theory of Automated Timetabling*. 18th-20th Aug, Pittsburgh, PA USA. 487-490.
- Eklund, N. H. W., Embrechts, M.J., and Goetschalckx, M., 2006, Efficient chromosome encoding and problem-specific mutation methods for the flexible bay facility layout problem, *IEEE Transactions on Systems, Man, and Cybernetics—PART C: Applications and Reviews*, 36, 4, 495-502.
- Eklund, N. H. W., 2006, Using genetic algorithms to estimate confidence intervals for missing spatial data, *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 36, 4, 519-524.
- Ferland, J.A., Roy, S. and Loc, T.G., 1986, The timetabling problem, In J. D. Coelho and L. V. Tavares, editors, *O.R. Models on Microcomputers*, 97-103, North-Holland.
- Ferland, J.A., and Fleurent, C., 1994, SAPHIR: A decision support system for course scheduling, *Interfaces* 24, 105-115.
- Ferland, J.A., and Roy, S., 1985, Timetabling problem for university as assignment of activities to resources, *Computers and Operations Research*, 12, 2, 207-218.

### KAYNAKLAR DİZİNİ (devam)

- Filho, G.R., and Antonio, L., 2001, A constructive evolutionary approach to school timetabling, *Lecture Notes In Computer Science*, 2037, 130-139.
- Fisher, J.G. and Shier, D.R., 1983, A heuristic procedure for large-scale examination scheduling problems, Technical Report 417, Department of Mathematical Sciences, Clemson University.
- Fonseca, C.M. and Fleming, P.J., 1993, Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, In S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, California, 416–423. Morgan Kaufmann.
- Foulds, L.R., and Johnson, D.G., 2000, SlotManager: a microcomputer-based decision support system for university timetabling, *Decision Support Systems*, 27, 367-381.
- Gani, T. A., Khader, A. T., and Budiarto, R., 2004, Optimizing examination timetabling using a hybrid evolution strategies, 2<sup>nd</sup> International Conference on Autonomous Robots and Agents, Palmerston North, New Zealand, 345-349.
- Gendreau, M., 2002, An introduction to tabu search, in: *Handbook of Metaheuristic*, (Eds: Glover, F. and Kochenberger, G.A.), Kluwer Academic Publishers, 37.
- Glassey, C.R., and Mizrach, M., 1986, A decision support system for assigning classes to rooms, *Interfaces*, 16, 92-100.
- Goldberg, D. E., 1989, *Genetic algorithms in search optimization and machine learning*, Addison-Wesley, Reading, MA.
- Goltz, H.J., K uchler, G. and Matzke, D., 1998, Constraint-Based timetabling for universities, In: *Proceedings of 11<sup>th</sup> International Conference on Applications of Prolog (INAP-98)*, Tokyo, Japan.
- Gosselin, K., and Truchon, M., 1986, Allocation of classrooms by linear programming, *The Journal of the Operational Research Society*, 37, 6, 561-569.
- Gunawan, A., Ng, K.M., and Poh, K.L., 2007, Solving the teacher assignment-course scheduling problem by a hybrid algorithm, *International Journal of Computer, Information, and Systems Science, and Engineering*, 1, 2, 136-141.

### KAYNAKLAR DİZİNİ (devam)

- Günelay, Y. and Şahin, T., 2006, A decision support system for the university timetabling problem with instructor preferences, *Asian Journal of Information Technology*, 5, 12, 1479-1484.
- Harwood, G.B. and Lawless, R.W., 1975, Optimizing organizational goals in assigning faculty teaching schedules, *Decision Sciences*, 6, 3, 513 – 524.
- Hertz, A., 1991, Tabu search for large scale timetabling problems, *European Journal of Operational Research*, 54, 39-47.
- Hertz, A., 1992, Finding a feasible course schedule using tabu search, *Discrete Applied Mathematics*, 35, 3, 255-270.
- Hilton, A.J.W., Slivnik, T., and Stirling, D.S.G., 2001, Aspects of edge list-colourings, *Discrete Mathematics*, 231, 253-264.
- Hinchliffe, P.D., 1973, Movement problems in comprehensive schools, *Operational Research Quarterly (1970-1977)*, 24, 2, 235-240.
- Horn, J. and Nafpliotis, N., 1993, Multiobjective optimization using theniched pareto genetic algorithm, *IlliGAL Report 93005*, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Champaign.
- Ismayilova, N.A., Sagir, M. and Gasimov, R.N., 2007, A multiobjective faculty-course-time slot assignment problem with preferences, *Mathematical and Computer Modelling*, 46, 7-8, 1017 – 1029.
- Johnson, D., 1990, Timetabling university examinations, *Journal of Operations Research Society*, 41, 1, 39-47.
- Johnson, D., 1993, A database approach to course timetabling, *Journal of the Operational Research Society*, 49, 1146-1152.
- Kanoh, H. and Sakamoto, Y., 2004, Interactive timetabling system using knowledge-based genetic algorithms, *IEEE International Conference on System, Man & Cybernetics (SMC'2004)*, 5852-5857.
- Kara, İ. ve Sağır Özdemir, M., 1996, Öğretici-Ders Atama Problemine EnkEnb Yaklaşım, XVIII. Yöneylem Araştırması ve Endüstri Mühendisliği Ulusal Kongresi, İstanbul.

**KAYNAKLAR DİZİNİ (devam)**

- Kara, İ. and Ozdemir, M.S., 1997, Minmax approaches to the faculty-course assignment problem, Proceedings of the Second International Conference on the Practice and Theory of Automated Timetabling, 167-181.
- Karaboğa, D., 2004, Yapay Zeka Optimizasyon Algoritmaları, Atlas yayın-dağıtım.
- Karwan, M.H., Lotfi, V., Zelgen, J., and S. Zionts (Eds.), 1995, Redundancy in mathematical programming: A state of the art survey, Springer, New York.
- Kazarlis, S., 2005, Solving university timetabling problems using advanced genetic algorithms, 5<sup>th</sup> International Conference on Technology and Automation (ICTA'05), <http://icta05.teithe.gr/papers/24.pdf>.
- Kersten, E., 2002, Decision making and decision support systems, In book: Decision Support Systems for Sustainable Development, Eds. Mikolajuk, Z., Gar-On Yeh, A., Kluwer Academic Publishers, New York.
- Kiarer, L., and Yellen, J., 1992, Weighted graphs and university course timetabling, Computers and Operations Research, 19, 1, 59-67.
- Kostuch, P., 2005, The university course timetabling problem with a three-phase approach, Practice and Theory of Automated Timetabling V, Lecture Notes in Computer Science, 3616, 109-125.
- Kovacic, M., 1993, Timetable construction with Markovian neural network, European Journal of Operational Research, 69, 92-96.
- Lai, L., Hsueh, N., Huang, L., and Chen, T., 2008, An artificial intelligence approach to course timetabling, International Journal On Artificial Intelligence Tools, 17, 1, 223-240.
- Lampkin, W., and Saalmans, P.D., 1967, The design fo routes, service frequencies, and schedules for a municipal bus undertaking, Operational Research Quarterly, 18, 4, 375-397.
- Laporte, G., and Desroches, S., 1984, Examination timetabling by computer. Computers and Operational Research, 11, 4, 351-360.
- Lawrie, N., 1969, An integer linear programming model of a school timetabling problem, The Computer Journal, 12, 307-316.

**KAYNAKLAR DİZİNİ (devam)**

- Lee, S.M., and Schniederjans, M.J., 1983, Multicriteria assignment problem: a goal programming approach, *Interfaces*, 13, 75-81.
- Lee, Y.M., and Chen, E.J. (2002) BASF Uses a framework for developing web-based production-planning-optimization tools, *Interfaces*, 32, 6, 15-24.
- Le Huede, F., Grabisch, M., Labreuche, C., and Saveant, P., 2006, Integration and propagation of a multi-criteria decision making model in constraint programming, *Journal of Heuristics*, 12, 4-5, 329-346.
- Lewis, R., 2008, A survey of metaheuristic-based techniques for university timetabling problems, *OR Spectrum*, 30, 167–190.
- Liebchen, C., and Möhring, R.H., 2002, A case study in periodic timetabling, *Electronic Notes in Theoretical Computer Science*, 66, 2, 1–14.
- Lotfi, V., and Cerveny, R., 1991, A final-exam-scheduling package. *The Journal of the Operational Research Society*, 42, 3, 205-216.
- Malim, M.R., Khader, A.T., and Mustafa, A., 2006, Artificial immune algorithms for university timetabling, E.K. Burke, H. Rudová (Eds.): *PATAT 2006*, 234–245.
- Marin, H.T., Ross, P., and Rendon, M.V., 1999, Evolution of constraint satisfaction strategies in examination timetabling, In *Genetic and Evolutionary Computation Conference, GECCO'99*, 635-642.
- Marti, R., Lourenço, H., and Laguna, M., 2000, Assigning proctors to exams with scatter search in computing tools for modeling, *Optimization and Simulation: Interfaces in Computer Science and Operations Research*, M. Laguna and J. L. González-Velarde, Eds. Kluwer Academic Publishers, Boston, MA, 215--227.
- McClure, R.H., and Wells, C.E., 1984, A mathematical programming model for faculty course assignment, *Decision Sciences*, 153, 409-420.
- Meade, L.M., and Sarkis, J., 1999, Analyzing organizational project alternatives for agile manufacturing processes: An analytical network approach, *International Journal of Production Research*, 37, 2, 241–261

### KAYNAKLAR DİZİNİ (devam)

- Mendes, J. J. M., Gonçalves, J.F., and Resende, M. G.C., 2005, A random key based genetic algorithm for the resource constrained project scheduling problem, AT&T Labs Research Technical Report TD-6DUK2C, June 30, 2005.
- Merlot, L.T.G., Boland, N., Hughes, B.D., and Stuckey, P.J.A., 2003, Hybrid algorithm for the examination timetabling problem, Lecture Notes in Computer Science, 2740, 207-231.
- Michalewicz, Z., Vignaux, G.A., and Hobbs, M., 1991, A nonstandard genetic algorithm for the nonlinear transportation problem, ORSA Journal of Computing, 3, 4, 307-316.
- MirHassani, S.A., 2006, A computational approach to enhancing course timetabling with integer programming, Applied Mathematics and Computation, 175, 1, 814-822.
- Monfroglio, A., 1996, Timetabling through constrained heuristic search and genetic algorithms, Software-Practice and Experience, 263, 251-279.
- Muller, T. and Bartak, R., 2001, Interactive Timetabling, CoRR: Programming Languages, PL/0109022.
- Mulvey, J.M. , 1982, A classroom/time assignment model, European Journal of Operational Research, 9, 64-70.
- Munoz, S., Ortuno, M.T., Ramirez, J., and Yanez, J., 2005, Coloring fuzzy graphs, Omega, 33, 211-221.
- Murray, K., Muller, T., and Rudova, H., 2007, Modeling and solution of a complex university course timetabling problem, In Edmund Burke and Hana Rudova, editors, Practice and Theory of Automated Timetabling, Selected Revised Papers, Springer-Verlag, LNCS 3867, 189–209.
- Norman, B. and Bean, J., 1995, Random keys genetic algorithm for scheduling, Technical Report 94-5, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, 48109-2117.
- Norman, B.A., and Smith, A.E., 1997, Random keys genetic algorithm with adaptive penalty function for optimization of constrained facility layout problems. In Proceedings of the Forth International Conference on Evolutionary Computation, 407–411, IEEE, Piscataway, New Jersey.

### KAYNAKLAR DİZİNİ (devam)

- Norman, B.A., Smith, A.E., and Arapoglu, R.A., 1998, Integrated facility design using an evolutionary approach with a subordinate network algorithm. In Eiben, A. E. et al., editors, *Parallel Problem Solving from Nature, PPSN V*, 937–946, Springer-Verlag, Berlin, Germany.
- Ozdemir, M.S. and Gasimov, R.N., 2004, The analytic hierarchy process and multiobjective 0-1 faculty course assignment, *European Journal of Operational Research*, 157, 398-408.
- Ozturk, Z.K., Ozturk, G. and Sagir, M., 2006, Multiobjective exam-supervisor assignment problem, *Multi Criteria Decision Making Conference, MCDM 2006*, Chania, Greece.
- Ozturk, Z.K. and Sagir, M., 2008, Exam scheduling problem: An AHP model for parameter estimation, *EURO Mini Conference, Continuous Optimization and Knowledge-Based Technologies*, Neringa, Lithuania.
- Ozturk, Z.K., Ozturk, G. and Sagir, M., 2010, An automated multi-objective invigilator-exam assignment system, *International Journal of Information Technology & Decision Making*, 10, 2, 1-16.
- Özdemir, M.S. and Kara, İ., 1997, Faculty-course assignment problem, *Annual Meeting of Mathematical Association of America*, 18, Wayne State University.
- Özdemir, M.S., 1998, Haftalık ders programlarının hazırlanmasında iki kademeli bir yaklaşım, *Doktora tezi, Osmangazi Üniversitesi, (yayımlanmamış)*.
- Özdemir, M.S., 2003, Karmaşık Ortamlarda Karar Verme: Analitik Hiyerarşi ve Serim Süreçleri, *IV. Endüstri İşletme Mühendisliği Kurultayı Bildiriler Kitabı*, 205-213.
- Özdemir, M.S., 2004, Analitik Serim Süreci ve EM'deki Uygulamaları, *Yöneylem Araştırması Ulusal Kongresi*, Adana, 2005.
- Özkan, B., Cevre, U. ve Uğur, A., 2008, Melez bir eniyileme yöntemi ile rota planlama, *Akademik Bilişim 2008, Çanakkale Onsekiz Mart Üniversitesi, Çanakkale*, 30 Ocak - 01 Şubat 2008. [http://ab.org.tr/ab08/kitap/Bildiriler/47\\_44\\_AB08.pdf](http://ab.org.tr/ab08/kitap/Bildiriler/47_44_AB08.pdf)
- Paquete, L.F., and Fonseca, C.M., 2001, A study of examination timetabling with multiobjective evolutionary algorithms, *MIC'2001 - 4<sup>th</sup> Metaheuristics International Conference*, Porto, Portugal, July 16-20, 2001, 149-153.

**KAYNAKLAR DİZİNİ (devam)**

- Peck, J.E.L. and Williams, M.R., 1966, Algorithm 286: Examination scheduling. Communications of the ACM, 9, 433–434.
- Petrovic, S., and Qu, R., 2002, Case-based reasoning as a heuristic selector in a hyperheuristic for course timetabling problems, Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies, Volume 82. Proceedings of KES'02, pp.336-340.
- Petrovic, S., and Bykov, Y., 2003, A multiobjective optimisation technique for exam timetabling based on trajectories, Practice and Theory of Automated Timetabling: Selected Papers from the 4<sup>th</sup> International Conference. Lecture Notes in Computer Science, 2740, Springer-Verlag, Berlin, Heidelberg, 179-192.
- Petrovic, S., and Burke, E.K., 2004, University timetabling, In J. Leung, editor, Handbook of Scheduling: Algorithms, Models, and Performance Analysis, Chapter 45. Chapman Hall/CRC Press.
- Petrovic S., Yang Y. and Dror M., 2007a, Case-based selection of initialisation heuristics for metaheuristic examination timetabling, Expert Systems with Applications: An International Journal, 33, 3, 772-785.
- Petrovic S., Yang Y. and Dror, M., 2007b, Investigating Ahuja-Orlin's large neighbourhood search approach for examination timetabling, OR Spectrum, 29, 2, 351-372.
- Qu, R., Burke, E.K., McCollum, B., Merlot, L.T.G., and Lee, S.Y., 2009, A survey of search methodologies and automated system development for examination timetabling, Journal of Scheduling, 12, 55–89.
- Ross, P., Corne, D., and Fang, H., 1994, Successful lecture timetabling with evolutionary algorithms, In: Proceedings of the workshop on applied genetic and other evolutionary algorithms (ACAI'94).
- Ross, P., and Hart, E., 2002, Using evolutionary algorithms to solve problems by combining choices of heuristics, Sarker, Ruhul Editor. Evolutionary Optimization. Secaucus, NJ, USA: Kluwer Academic Publishers, 229 p.
- Rossi-Doria, O., Blum, C., Knowles, J., Sampels, M., Socha, K., and Paechter, 2002, A local search for the timetabling problem, In Proceedings of the 4<sup>th</sup> International Conference on the Practice And Theory of Automated Timetabling, PATAT 2002, Aug. 2002, 124-127.

### KAYNAKLAR DİZİNİ (devam)

- Rossi-Doria, O., and Sampels, M., 2002, A comparison of the performance of different metaheuristics on the timetabling problem, In E. Burke and P. De Causmaecker, editors, Proceedings of Fourth International the Conference on Practice and Theory of Automated Timetabling, 115-119.
- Rossi-Doria, O., and Paechter, B., 2003, An hyperheuristic approach to course timetabling problem using an evolutionary algorithm, Technical report, Napier Kaufmann. University, Edinburgh, United Kingdom.
- Rossi-Doria, O., Sampels, M., Birattari, M., Chiarandini, M., Dorigo, M., Gambardella, L.M., Knowles, J., Manfrin, M., Mastrolilli, Paetcher, B., Paquete, L. and Stützle, T.A., 2003, Comparison of the performance of different metaheuristics on the timetabling problem, Lecture Notes in Computer Science, 2740, 330-351.
- Rothlauf, F., Goldberg, D.E., and Heinzl, A., 2002, Network random keys – a tree representation scheme for genetic and evolutionary algorithms, Evolutionary Computation, 10, 1, 75-97.
- Rudova, H., and Murray, K., 2003, University course timetabling with soft constraints, in: Burke, De Causmaecker (Eds.), Practice and Theory of Automated Timetabling IV. KaHo St.-Lieven, Gent (Belgium) : Springer, 2003. ISBN 3-540-40699-9, 310-328.
- Russell, R.A., and Urban, T.L., 2006, A constraint programming approach to the multiplevenue, sport-scheduling problem, Computers & Operations Research, 33, 7, 1895–1906.
- Saaty, T.L., 1994, Fundamentals of Decision making and priority theory with the Analytic Hierarchy Process, RWS Publications, Pittsburgh, PA.
- Saaty, T.L., 1996, Decision making with dependence and feedback: The Analytic Network Process, RWS Publications, Pittsburgh
- Sabin, G.C.W., and Winter, G.K., 1986, The impact of automated timetabling on universities- a case study, Journal of Operations Research Society, 37, 7, 689-693.
- Sağ, T. ve Çunkaş, M., 2009, Çok amaçlı genetik algoritmalar için bir çevrimdışı performans değerlendirmesi, 5. Uluslararası İleri Teknolojiler Sempozyumu (IATS'09), 13-15 Mayıs 2009, Karabük, Türkiye.

### KAYNAKLAR DİZİNİ (devam)

- Sağır Özdemir, M., Kara, İ., 1997, Faculty-course assignment problem, Annual Meeting of Mathematical Association of America, 18, Wayne State University, Detroit, USA.
- Sampson, S.E., Freeland, and J.R. Weiss, E.N., 1995, Class scheduling to maximize participant satisfaction, *Interfaces*, 25. 30-41.
- Santiago-Mozos, R., Salcedo-Sanz, S., dePrado-Cumplido, M., and Bousoño-Calzón, C., 2005, A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a Spanish university, *Computers and Operations Research*, 32, 1761-1776.
- Schaerf, A., 1999, A survey of automated timetabling, *Artificial Intelligence Review*, 13 2, 87-127.
- Schaerf, A., Cadoli, M., and Lenzerini, M., 2000, LOCAL++: A C++ framework for local search algorithms, *Software- Practice and Experience*, 30, 3, 233-257.
- Schaerf, A., and Di Gaspero, L., 2001, Local search techniques for educational timetabling problems (invited paper), In L. Lenart, L. Zadnik Stirn, and S. Drobne, editors, *Proc. of the 6<sup>th</sup> International Symposium on Operations Research in Slovenia (SOR-01)*, 13–23.
- Schaffer, J.D., 1985, Multi-objective optimization with vector evaluated genetic algorithms, In: *Proceedings of an international conference on genetic algorithms and their applications*, 93–100.
- Schimmelpfeng, K., and Helber, S., 2007, Application of a real-world university-course timetabling model solved by integer programming, *OR Spectrum*, 29, 783–803.
- Schönberger, J., Mattfeld, D.C., and Kopfer, H., 2004, Memetic algorithm timetabling for non-commercial sport leagues, *European Journal of Operational Research*, 153, 102–116.
- Schniederjans, M.J., and Kim, G.C., 1987, A goal programming model to optimize departmental preference in course assignments, *Computers and Operations Research*, 14, 2, 87–96.
- Shih, W., and Sullivan, J.A., 1977, Dynamic course scheduling for college faculty via zero-one programming, *Decision Sciences*, 8, 711-721.

**KAYNAKLAR DİZİNİ (devam)**

- Silva, J.D.L., Burke, E.K., and Petrovic, S., 2004, An introduction to multiobjective metaheuristics for scheduling and timetabling, *Metaheuristic for Multiobjective Optimisation, Lecture Notes in Economics and Mathematical Systems*, 535, 91–129.
- Smith, K.A., Abramson, D., and Duke, D., 2003, Hopfield neural networks for timetabling: formulations, methods, and comparative results, *Computers & Industrial Engineering*, 44, 2, 283–305.
- Socha, K., Knowles, J., and Sampels, M., 2002, A MAX-MIN ant system for the university course timetabling problem, *Lecture Notes In Computer Science*, 2463, *Proceedings of the Third International Workshop on Ant Algorithms*, Springer-Verlag, London, UK.
- Snyder, L.V., and Daskin, M.S., 2006, A random-key genetic algorithm for the generalized traveling salesman problem, *European Journal of Operational Research*, 174, 38-953.
- Srinivas, N. and Deb, K., 1994, Multiobjective optimization using non-dominated sorting in genetic algorithms, *Evolutionary Computation*, 2, 3, 221–248.
- Strnad, D., and Guid, N., 2007, A multi-agent system for university course timetabling, *Applied Artificial Intelligence*, 21, 137–153.
- Paquete, L.F., and Fonseca, C.M., 2001, A study of examination timetabling with multiobjective evolutionary algorithms, *MIC'2001 - 4<sup>th</sup> Metaheuristics International Conference*, Porto, Portugal, July 16-20, 2001, 149-153.
- Piechowiak, S. and Kolski, C., 2004, Towards a generic object oriented decision support system for university timetabling: an interactive approach, *International Journal of Information Technology & Decision Making*, 3, 1, 179–208.
- Reis, L.P. and Oliveira, E., 2000, A language for specifying complete timetabling problems, *Lecture Notes Computer Science*, 2079, 322-341.
- Thompson, J.M., and Dowsland, K.A., 1998, A robust simulated annealing based examination timetabling system, *Computers & Operations Research*, 25, 7/8, 637-648.
- Tripathy, A., 1980, A lagrangean relaxation approach to course timetabling, *Journal of Operations Research Society*, 31, 599–603.

### KAYNAKLAR DİZİNİ (devam)

- Tripathy, A., 1984, School timetabling-a case in large binary integer linear programming, *Management Science*, 30, 12, 1473-1489.
- Üstün, Ö., Özdemir, M.S., Demirtaş, E.A., 2005, Kıbrıs Sorunu Çözüm Önerilerini Değerlendirmede Analitik Serim Süreci Yaklaşımı, *Endüstri Mühendisliği Dergisi*, 16, 4, 2-13.
- Tunalıoğlu, N. ve Öcalan, T., 2007, Üç boyutlu karayolu güzargah optimizasyounda karar destek sistemi olarak genetik algoritmaların kullanımı, 11. Türkiye Harita Bilimsel ve Teknik Kurultayı, 2-6 Nisan 2007, Ankara.
- Valouxis, C., and Housos, E., 2003, Constraint programming approach for school timetabling, *Computers & Operations Research*, 39, 1555-1572.
- Voráč, J., Vondrák, I. and Vlček, K., 2002, School timetable generating using genetic algorithm, 3<sup>rd</sup> International PhD Workshop: Advances in Supervision and Control Systems Young Generation Viewpoint. <http://www-e2.ijs.si/PhDWorkshop/2002/Papers.htm>.
- Yanez, J., and Ramirez, J., 2003, The robust coloring problem, *European Journal of Operational Research*, 148, 546–558.
- Yu, E. and Sung, K.S. , 2002, A genetic algorithm for a university weekly courses timetabling problem, *Intl. Trans. in Op. Res.*, 9, 703-717.
- Welsh, D.J.A. and Powell, M.B., 1967, An upper bound for the chromatic number of a graph and its application to timetabling problems, *The Computer Journal*, 10, 1, 85-86.
- White, G.M., Chan, P.W., 1979, Towards the Construction of Optimal Examination Timetables, *INFORMS*, 17, 219-229.
- White, G.M., Xie, B.S., and Zonjic, S., 2004, Using tabu search with longer-term memory and relaxation to create examination timetables, *European Journal of Operational Research*, 153, 80–91.
- Wilson, J.M., 1981, The scheduling of magistrates to courts, *The Journal of the Operational Research Society*, 32, 2, 121-124.

**KAYNAKLAR DİZİNİ (devam)**

- Wood, D.C., 1968, A system for computing university examination timetables, *Computer Journal*, 11, 1, 41-47.
- Wood, J., and Whitaker, D., 1998, Student centred school timetabling, *Journal of the Operational Research Society*, 44, 5, 425-433.
- Wren, A., 1996, Scheduling, timetabling and rostering - a special relationship? In the *Practice and Theory of Automated Timetabling*, ed. E.K. Burke and P. Ross, 46-75.
- Wright, M.B., 1991, Scheduling English cricket umpires, *Journal of the Operational Research Society*, 42, 42, 447-452.
- Wright, M., 1996, School timetabling using heuristic search, *Journal of the Operational Research Society*, 47, 347-357.
- Zampieri, A., and Schaerf, A., 2006, Modelling and solving the Italian examination timetabling problem using tabu search, E. K. Burke, H. Rudová (Eds.): *PATAT 2006*, 487-491.
- Zhang, S., and Goddard, S., 2007, A software architecture and framework for Web-based distributed decision support systems, *Decision Support Systems*, 43, 1133-1150.
- Zitzler, E., and Thiele, L., 1999, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.*, 3, 257-271.
- Zitzler, E., Laumanns, M., and Thiele, L., 2001, SPEA2: Improving the strength pareto evolutionary algorithm, *TIK-Report 103*.

## ÖZGEÇMİŞ

1979 yılında Eskişehir’de doğan Zehra Kamışlı Öztürk, ilk, orta ve lise eğitimini sırasıyla Porsuk İlkokulu ve Eskişehir Anadolu Lisesi’nde tamamladı. Lisans derecesini bölüm üçüncüsü olarak 2001 yılında Osmangazi Üniversitesi Endüstri Mühendisliği Bölümü’nden aldı. Aynı üniversitede 2001 yılında Endüstri Mühendisliği Bölümü Yöneylem Araştırması Bilim Dalı’nda yüksek lisans eğitimine başladı. 2002 yılında ise Anadolu Üniversitesi Açıköğretim Fakültesi’nde araştırma görevlisi olarak akademik göreve başladı. 2004 yılında Yard. Doç. Dr. Servet Hasgöl danışmanlığında yürüttüğü “Açıköğretim Fakültesinde Öğrenci Sayısı Tahmini için Sistem Dinamiği Yaklaşımı ve Benzetim Uygulamaları” başlıklı tezini savunarak yüksek lisans derecesini aldı. Aynı yıl, Osmangazi Üniversitesi Endüstri Mühendisliği Bölümü, Yöneylem Araştırması Bilim Dalı’nda doktora başladı.

Zehra Kamışlı Öztürk, 2000 yılı Uluslararası Endüstri Mühendisliği Öğrencileri Sempozyumu Proje Yarışması üçüncülük (ekip olarak); 2007 yılı 4. Lojistik Vak’a yarışması ikincilik (ekip olarak) ve 2007 yılı UNESCO ile COL tarafından 21. Asya Açık Üniversiteler Birliği konferansında mali destek ödülleri ile 2005 yılında 5 yıl süre boyunca TÜBİTAK yurt içi doktora bursuna layık görüldü. 15 Temmuz- 17 Eylül 2008 tarihleri arasında ise Ballarat Üniversitesi, Avustralya’da misafir araştırmacı olarak çalıştı.

Anadolu Üniversitesi Proje Birimi Yönetim Kurulu üyesi, İleri Teknolojiler Araştırma Birimi proje bilimci, Açıköğretim Fakültesi araştırma görevlisi temsilciliği ve Açıköğretim Fakültesi Okulöncesi Öğretmenliği Lisans Programı Topluma Hizmet Uygulamaları dersi uygulama koordinatörlüğü görevlerinde bulunan Zehra Kamışlı Öztürk halen Anadolu Üniversitesi Açıköğretim Fakültesi’nde görev yapmaktadır.

Kamışlı Öztürk, evli ve bir çocuk annesidir.

**EKLER**

**Ek.1:** İncelenen çalışmalarda ele alınan bileşenler

**Ek.2:** Eğitimsel zaman çizelgeleme problemlerini çok amaçlı olarak ele alan çalışmalar

**Ek.3:**Ders-derslik-zaman çizelgeleme small01 problemi için sonuç dosyası

**Ek.4:** Ders-Derslik-zaman çizelgeleme problemi için RKGA C# kodu

**Ek.5:** Ders-derslik-zaman çizelgeleme problemi için NSGA-II temelli RKGA C# kodu

**Ek.6:** Sınav-derslik-zaman çizelgeleme problemi için 2005-2006 Güz Dönemi 2. arasınav çizelgesi



Ek.1 devam

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler					Yayımlanan başlık	
			Ders	Derslik	Öğretici	Öğrenci	Sınav		Zaman
19	1986	Carter, M.W.							Survey
20	1986	Sabin, G.C.W. and Winter, G.K.	+			+		+	University
21	1986	Glasse, C.R. and Mizrach, M.	+	+					University
22	1986	Kassiech, S.K., Burleson, D.K. and Lievano, R.J.	+	+	+			+	University
23	1986	Gosselin, K. and Truchon, M.		+				+	University classroom
24	1987	Schniederjans, M.J. and Kim, G.C.	+		+				Course
25	1989	Dinkel, J.J., Mote, J. and Venkataramanan, M.A.	+	+				+	Course (university)
26	1989	Chahal, N. and de Werra, D.	+	+	+			+	School
27	1989	Aubin, J. and Ferland, J.A.	+		+			+	Timetabling
28	1990	Johnson, D.		+			+	+	Exam (University)
29	1990	Colomi, A., Dorigo, M. and Maniezzo, V.	+		+			+	School
30	1990	Dowland, K.A.	+	+				+	University
31	1991	Hertz, A.	+	+				+	Course (university)
32	1991	Abramson, D.	+	+	+			+	School
33	1991	Lotfi, V., and Cerveny, R.		+			+	+	Exam (University)
34	1992	Hertz, A.	+	+				+	Course
35	1992	Kiarer, L. and Yellen, J.	+					+	Course (university)
36	1992	Abramson, D. and Abela, J.	+	+	+			+	School
37	1992	Fahrion, R. and Dollansky, G.	+	+				+	University
38	1992	Carter, M.W. and Tovey, C.A.	+	+					Classroom assignment

Ek.1 devam

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler					Yayımda tanımlanan başlık	
			Ders	Derslik	Öğretici	Öğrenci	Sınav		Zaman
39	1992	Tripathy, A.	+	+				+	Class
40	1992	Cangalovic, M. and Schreuder, J.A.M.	+					+	Class-Teacher
41	1992	Colorni, A., and Morimo, D.	+					+	School
42	1993	Johnson, D.	+		+			+	Course
43	1993	Kovacic, M.			+				School
44	1993	Burke, E.K., Elliman, D.G., and Weare, R.		+			+	+	Exam (University)
45	1994	Ferland, J.A. and Fleurent, C.	+			+			Course
46	1994	Costa, D.	+					+	School
47	1994	Corne, D., Ross, P., and Fang, H.							survey
48	1994	Burke, E., Elliman, D. and Weare, R.		+			+	+	University
49	1994	Ross, P., Corne, D., and Fang, H.	+	+				+	Lecture (university)
50	1995	Sampson, S.E., Freeland, J.R. and Weiss, E.N.	+	+		+			Class
51	1995	Wren, A.							Survey
52	1995	Erben, W. and Keppler, J.	+	+	+			+	Course (University)
53	1996	Euler, R. and Verge, H.L.	+		+				Class-Teacher
54	1996b	Burke, E. K., Newall, J. P., and Weare, R. F.,		+			+		Exam
55	1996a	Burke, E., Elliman, D., Ford, P. and Weare, R.							Exam
56	1996	Carter, M.W., Laporte, G. and Lee, S.Y.					+	+	Exam
57	1996	Wright, M.	+	+				+	School
58	1996	Valdes, R.A., Martin, G. and Tamarit, J.M.	+					+	School



Ek.1 devam

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler						Yayımda tanımlanan başlık
			Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
79	1999	Deris, S., Omatu, S., Ohta, H., and Saad, P.	+	+				+	Course (university)
80	2000	Deris, S., Omatu, S., and Ohta, H.	+	+				+	College
81	2000	Foulds, L.R. and Johnson, D.G.	+	+				+	University
82	2000	Abdennadher, S. Saft, M. and Will, S.		+				+	Course (University)
83	2000	Alvarez-Valdes, R., Crespo, E. and Tamarit, J.M.	+			+			Student (university)
84	2000	de Werra, D., Hertz, A., Kobler, D., and Mahadev, N.V.R.	+		+			+	Class-Teacher
85	2000	di Gaspero, L., and Schaerf, A.					+	+	Exam (University)
86	2000	Marti, R., Lourenço, H., and Laguna, M.			+		+		Proctor
87	2001	Dimopoulou, M. and Miliotis, P.	+	+				+	Course (University)
88	2001	Burke, E., Bykov, Y., and Petrovic, S.					+	+	Exam
89	2001	Caramia, M., Dell'Olmo, P. and Italiano, G.F.					+	+	Exam
90	2001	Erben, W.					+	+	Exam
91	2001	Hilton, A.J.W., Slivnik, T., and Stirling, D.S.G.	+		+			+	School/conference
92	2001	Muller, T. and Bartak, R.	+	+	+			+	School
93	2001	Paquete, L.F., and Fonseca, C.M.					+	+	Exam (University)
94	2001	Schaerf, A., and di Gaspero, L.					+	+	Exam (University)
95	2001	Schaerf, A., and di Gaspero, L.	+	+				+	Course (university)
96	2001	Filho, G.R., and Antonio, L.	+		+			+	School
97	2001	di Stefano, C. and Tettamanzi, A.G.B.	+					+	School
98	2002	Baker, K.R., Magazine, M.J., and Polak, G.G.	+					+	Course

Ek.1 devam

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler						Yayımda tanımlanan başlık
			Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
99	2002	Petrovic, S. and Qu, R.	+	+				+	Course
100	2002	Socha, K., Knowles, J., and Sampels, M.	+	+				+	Course (University)
101	2002	Burke, E.K. and Newall, J.P.					+	+	Exam
102	2002	Burke, E.K. and Petrovic, S.							survey
103	2002	Nepal, T. and Ally, M.I.	+	+	+			+	Tertiary
104	2002	Alvarez-Valdes, R., Crespo, E. and Tamarit, J.M	+	+				+	University
105	2002	Asratian, A.S						+	University
106	2002	Rossi-Doria, O. and Sampels, M.	+	+				+	Course (university)
107	2002	Blum, C., Correia, S., Dorigo, M., Paechter, B., Rossi-Doria, O. and Snoek, M.	+	+				+	University
108	2002	Ross, P. and Hart, E.		+			+	+	Exam
109	2002	Cowling, P., Kendall, G. and Hussin, N.M.							Exam (University)
110	2002	Chiarandini, M. and Stützle, T.	+	+				+	Course (university)
111	2002	Rossi-Doria, O., Blum, C., Knowles, J., Sampels, M., Socha, K., and Paechter,	+	+				+	Course (university)
112	2002	Voráč, J., Vondrák, I. and Vlček, K.	+	+				+	School
113	2002	Yu, E. and Sung, K.S.	+	+				+	Course (university)
114	2003	Rossi-Doria, O. and Paechter, B.	+	+				+	Course
115	2003	Alkan, A. and Özcan, E.	+					+	Course (university)
116	2003a	Burke, E.K., Kendall, G. and Soubeiga, E.	+	+				+	Course (University)
117	2003	Petrovic, S. and Bykov, Y.					+	+	Exam
118	2003	Casey, S. and Thompson, J.					+	+	Exam

Ek.1 devam

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler						Yayımda tanımlanan başlık
			Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
119	2003	Yanez, J. and Ramirez, J.					+	+	Exam (University)
120	2003	Smith, K.A., Abramson, D. and Duke, D.	+	+	+			+	School
121	2003	Valouxis, C. and Housos, E.	+		+			+	School
122	2003	Rudova,H., and Murray, K.	+	+				+	Course (university)
123	2003b	Burke, E.K., MacCarthy, B.L., Petrovic, S., and Qu, R.	+	+				+	Course (university)
124	2003	Merlot, L.T.G., Boland, N., Hughes, B.D., and Stuckey, P.J..					+	+	Exam
125	2003	Rossi-Doria, O., Sampels, M., Birattari, M., Chiarandini, M., Dorigo, M., Gambardella, L.M., Knowles, J., Manfrin, M., Mastrolilli, Paetcher, B., Paquete, L. and Stützle, T.	+	+				+	Course (university)
126	2004	Carrasco, M.P. and Pato, M.V.	+	+				+	Class-Teacher
127	2004	Burke, E.K. and Newall, J.P.					+	+	Exam
128	2004	Gani, T.A., Khader, A.T. and Budiarto, R.					+	+	Exam
129	2004	Erben, W. and Song, P.Y.					+	+	Exam
130	2004	Azimi, Z.N.					+	+	Exam
131	2004	Burke, E.K. and Newall, J.P.					+	+	Exam
132	2004	Asmuni, H., Burke, E. K., and Garibaldi, J.M.					+	+	Exam
133	2004	Burke, E. and Landa Silva, J.D.							genel
134	2004a	Burke, E., de Werra, D.and Kingston, J.							survey
135	2004	Dimopoulou, M. and Miliotis, P.	+	+				+	University
136	2004	Petrovic, S. and Burke, E.K.	+					+	University (Course and Exam)

Ek.1 devam

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler						Yayımda tanımlanan başlık
			Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
137	2004	Landa Silva, J.D., Burke, E.K. and Petrovic, S.							survey
138	2004	White, G.M., Xie, B.S., and Zonjic, S.					+	+	Exam
139	2004	Ozdemir, M.S. and Gasimov, R.N.	+		+				Faculty-Course
140	2004	Abbas, A., and Tsang, E.P.K.	+					+	Course (university)
141	2004	Daskalaki, S., Birbas, T., and Housos, E.	+	+				+	Course (university)
142	2004b	Burke, E., Bykov, Y., Newall, W., and Petrovic, S.					+	+	Exam (University)
143	2004	Piechowiak, S. and Kolski, C.	+	+				+	Course (university)
144	2004	Kanoh, H. and Sakamoto, Y.	+					+	Course (university)
145	2005	Muller, T., Rudova, H., Bartak, R.	+	+				+	Class (university)
146	2005a	Burke, E.K., Landa Silva, J.D. and Soubeiga, E.	+	+				+	Course (university)
147	2005	Santiago-Mozos, R., Salcedo-Sanz, S., DePrado-Cumplido, M. and Bousoño-Calzón, C.	+			+			Course (university)
148	2005	Avella, P., and Vasil'Ev, I.	+	+				+	Course (university)
149	2005	Kostuch, P.	+	+				+	Course (university)
150	2005	Dowland, K.A. and Thompson, J.					+	+	Exam
151	2005	Azimi, Z.N.					+	+	Exam
152	2005	Munoz, S., Ortuno, M.T., Ramirez, J., and Yanez, J.					+	+	Exam
153	2005b	E. Burke M.Dror S.Petrovic R.Qu					+	+	Exam
154	2005	S. Daskalaki T. Birbas	+	+	+	+		+	University
155	2005	H. Arntzen A.Lokketangen	+	+				+	University
156	2005	Kazarlis, S.	+	+	+			+	Course (university)

Ek.1 devam

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler						Yayımlanan başlık
			Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
157	2006	Al-Yakoob S.M. and Sherali H.D.	+		+			+	Class-Faculty
158	2006b	E. K. Burke, B. L. MacCarthy, S.Petrovic R. Qu	+	+				+	Course
159	2006	MirHassani S.A.	+		+			+	Course (university)
160	2006a	E.K. Burke S.Petrovic R.Qu	+	+				+	Course (university)
161	2006a	E.K. Burke S.Petrovic R.Qu		+			+	+	Exam (University)
162	2006	Le Huede F, Grabisch M, Labreuche C, Saveant P		+			+	+	Exam
163	2006	Dammak A., Elloumi A. and Kamoun H		+			+		Exam
164	2007	Petrovic S., Yang Y. and Dror M.					+	+	Exam
165	2007	S. Petrovic Y.Yang M.Dror		+			+	+	Exam
166	2006	Edmund K. Burke, Yuri Bykov					+	+	Exam
167	2006	Muhammad Rozi Malim, Ahamad Tajudin Khader2, and Adli Mustafa	+	+				+	Course (University)
168	2006	Muhammad Rozi Malim, Ahamad Tajudin Khader2, and Adli Mustafa		+			+	+	Exam (University)
169	2006	Andrea Zampieri and Andrea Schaerf		+			+	+	Exam (University)
170	2006	Chiarandini, M. Birattari, M., Socha, K., Rossi-Doria, O.	+	+				+	Course (University)
171	2006	Corr, P, McCollum, B., McGreevy, M., McMullan, P		+			+	+	Exam (University)
172	2006	Bilgin, B., Özcan, E.,and Korkmaz, E.E.					+	+	Exam (University)
173	2006	Beyrouthy, Camille / Burke, Edmund K / Landa-Silva, Dario / McCollum, Barry / McMullan, Paul / Parkes, Andrew J	+	+				+	Course (university)
174	2006	Günalay, Y. and Şahin, T.	+		+			+	Course (university)
175	2007	S. M. Al-Yakoob H.D.Sherali	+	+				+	Class

Ek.1 devam

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler						Yayımlanan başlık
			Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	
176	2007	E.K. Burke B.McCollum A.Meisels S. Petrovic R.Qu	+	+				+	course
177	2007	C.Head S.Shaban	+	+		+		+	Course/ student
178	2007	Beligiannis, G.N., Moschopoulos, C. and Likothanassis, S.D.	+		+			+	School
179	2007	Ismayilova, N.A., Sagir, M. and Gasimov, R.N.	+		+			+	Faculty-Course
180	2007	Dilip Datta and Kalyanmoy Deb and Carlos M. Fonseca	+					+	School
181	2007	Murray, K., Muller, T., and Rudova, H.	+	+				+	Course (university)
182	2007	Schimmelpfeng,K. and Helber, S.	+	+				+	Course (university)
183	2007	Gunawan, A., Ng, K.M., and Poh, K.L.	+		+			+	Course (university)
184	2007	Strnad, D., and Guid, N.	+	+				+	Course (university)
185	2007	Avella, P., D'Auria, B., Salerno, S., and Vasil'ev, I.	+		+			+	School
186	2007	Caramia, M., and Dell'Olmo, P.					+	+	Exam (University)
187	2007	Eley, M.		+			+	+	Exam (university)
188	2010	Al-Yakoob, S.M., Sherali, H.D. and Al-Jazzaf, M.			+		+		Proctor
189	2008	Boland, N., Hughes, B.D., Merlot, L.T.G., and Stuckey, P.J.	+		+	+			Course (high school)
190	2008	Burke, E.K., Marecek, J., Parkes,A.J., and Rudová, H.	+	+				+	Course (university)
191	2008	Lai, L., Hsueh, N., Huang, L., and Chen, T.	+	+				+	Course (university)
192	2008	Beligiannisa, G.N. Moschopouloa, C.N., Kaperonisa, G.P., and Likothanassisa, S.D.	+		+			+	School
193	2008	Lewis, R.							University timetabling (survey)
194	2009	Cheong, C.Y., Tan, K.C., and Veeravalli, B.					+	+	Exam (University)
195	2010	Sarin, S.C., Wang, Y. Varadarajan, A.	+	+				+	Course (university)

**Ek.1 devam**

Makale no	Yıl	Yazar	Gözönüne alınan bileşenler					Yayımda tanımlanan başlık	
			Ders	Derslik	Öğretici	Öğrenci	Sınav		Zaman
196	2010	Ozturk, Z.K., Ozturk, G. ve Sagir, M.			+		+		Invigilator

**Ek.2** Eğitimsel zaman çizelgeleme problemlerini çok amaçlı olarak ele alan çalışmalar

Gözönüne alınan bileşenler									
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayımlanan TT Türü	Kullanılan Yöntem
1975	Harwood, G.B. and Lawless, R.W.	+		+			+	Course	Hedef programlama
1983	Lee, S.M. and Schniederjans, M.J.			+				School	Hedef programlama
1987	Schniederjans, M.J. and Kim, G.C.	+		+				Course	Hedef programlama
1989	Aubin, J. and Ferland, J.A.	+		+			+	Timetabling	Tamsayılı programlama ve sezgisel bir algoritma
1990	Dowland, K.A.	+	+				+	University	Tavlama benzetimi
1991	Lotfi, V., and Cerveny, R.		+			+	+	Exam (University)	4 aşamalı sezgisel
1992	Cangalovic, M. and Schreuder, J.A.M.	+					+	Class-Teacher	Sezgisel algoritma
1992	Colorni, A., and Morimo, D.	+					+	School	Genetik Algoritmalar
1993	Johnson, D.	+		+			+	Course	Hedef programlama
1994	Ferland, J.A. and Fleurent, C.	+			+			Course	Tamsayılı programlama
1995	Erben, W. and Keppler, J.	+	+	+			+	Course (University)	Genetik algoritmalar
1996	Wright, M.	+	+				+	School	Sezgisel algoritma
1996	Badri, M.A.	+		+			+	University	Hedef programlama
1997	Ozdemir, M.S. and Kara, İ.	+		+				Course	Tamsayılı programlama
1997	Kara, İ. and Ozdemir, M.S.	+		+				Course	Tamsayılı programlama
1998	Thompson, J.M. and Dowland, K.A.					+	+	Exam	Tavlama benzetimi
1998	Wood, J. and Whitaker, D.	+			+		+	School	Hedef programlama
1998	Badri, M.A., Davis, D.L., Davis, F.D., and Hollingsworth, J.	+		+			+	Course (university)	Hedef programlama

## EK.2 devam

Gözönüne alınan bileşenler									
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayımda tanımlanan TT Türü	Kullanılan Yöntem
1999	Burke, E.K. And Newall, J.P.					+	+	Exam (University)	Memetik algoritmalar
2000	di Gaspero, L., and Schaerf, A.					+	+	Exam (University)	Yasaklı arama
2000	Marti, R., Lourenço, H., and Laguna, M.			+		+		Proctor	Dağınık arama algoritması
2001	Burke, E., Bykov, Y., and Petrovic, S.					+	+	Exam	Tepe tırmanma ve Ağ renklendirme sezgiselleri
2001	Muller, T. and Bartak, R.	+	+	+				School	Yerel arama algoritması
2001	Paquete, L.F., and Fonseca, C.M.					+	+	Exam (University)	Evrimsel algoritmalar
2001	Filho, G.R., and Antonio, L.	+		+				School	Genetik algoritmalar
2002	Burke, E.K. and Petrovic, S.							survey	Literatür taraması
2002	Rossi-Doria, O. and Sampels, M.	+	+				+	Course (university)	Evrimsel algoritmalar, karınca kolonileri, tavlama benzetimi, yasaklı arama
2002	Blum, C., Correia, S., Dorigo, M., Paechter, B., Rossi-Doria, O. and Snoek, M.	+	+				+	University	Genetik algoritmalar
2002	Ross, P. and Hart, E.			+		+	+	Exam	Hiper sezgiseller
2002	Rossi-Doria, O., Blum, C., Knowles, J., Sampels, M., Socha, K., and Paechter,	+	+				+	Course (university)	Yerel arama
2002	Yu, E. and Sung, K.S.	+	+				+	Course (university)	Genetik algoritmalar
2003	Rossi-Doria, O. and Paechter, B.	+	+				+	course	Hiper sezgiseller

## EK.2 devam

Gözönüne alınan bileşenler									
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayımda tanımlanan TT Türü	Kullanılan Yöntem
2003	Petrovic, S. and Bykov, Y.					+	+	Exam	Great Deluge algorithm
2003	Valouxis, C. and Housos, E.	+		+			+	School	Kısıt programlama
2003	Rudova,H., and Murray, K.	+	+				+	Course (university)	Mantıksal kısıt programlama
2003	Burke, E.K., MacCarthy, B.L., Petrovic, S., and Qu, R.	+	+				+	Course (university)	Hiper sezgiseller
2003	Rossi-Doria, O., Sampels, M., Birattari, M., Chiarandini, M., Dorigo, M., Gambardella, L.M., Knowles, J., Manfrin, M., Mastrolilli, Paetcher, B., Paquete, L. and Stützle, T.	+	+				+	Course (university)	Evrimsel algoritmalar, karınca kolonileri, tavlama benzetimi, yasaklı arama
2004	Carrasco, M.P. and Pato, M.V.	+	+				+	Class-Teacher	Sinir ağları
2004	Burke, E.K. and Newall, J.P.					+	+	Exam	Hiper sezgiseller
2004	Petrovic, S. and Burke, E.K.	+					+	University (Course and Exam)	Hiper sezgisellerle kısıt temelli çıkarsama
2004	Landa Silva, J.D., Burke, E.K. and Petrovic, S.							Literatür taraması	Genetik algoritmalar
2004	White, G.M., Xie, B.S., and Zonjic, S.					+	+	Exam	Yasaklı arama
2004	Ozdemir, M.S. and Gasimov, R.N.	+		+				Faculty-Course	Modified subgradient method
2004	Daskalaki, S., Birbas, T., and Housos, E.	+	+				+	Course (university)	Tamsayılı programlama
2004	Burke, E., Bykov, Y., Newall, W., and Petrovic, S.					+	+	Exam (University)	Tavlama benzetimi ve Great Deluge algoritması
2005	Burke, E.K., Landa Silva, J.D. and Soubeiga, E.	+	+				+	Course (university)	Hiper sezgiseller

## EK.2 devam

Gözönüne alınan bileşenler									
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayımda tanımlanan TT Türü	Kullanılan Yöntem
2005	Santiago-Mozos, R., Salcedo-Sanz, S., DePrado-Cumplido, M. and Bousoño-Calzón, C.	+			+			Course (university)	Evrimsel algoritmalar
2005	Kostuch, P.	+	+				+	Course (university)	Tavlama benzetimi
2005	E. Burke M.Dror S.Petrovic R.Qu					+	+	Exam	Hiper sezgiseller
2005	Kazarlis, S.	+	+	+			+	Course (university)	Genetik algoritmalar
2006	Al-Yakoob S.M. and Sherali H.D.	+		+			+	Class- Faculty	Karma tamsayılı programlama
2006	E.K. Burke S.Petrovic R.Qu	+	+				+	Course (university)	Olay temelli çıkarsama
2006	Le Huede F, Grabisch M, Labreuche C, Saveant P			+		+	+	Exam	Multi-Attribute Utility Theory
2006	Andrea Zampieri and Andrea Schaerf			+		+	+	Exam (University)	Yasaklı arama
2006	Chiarandini, M. Birattari, M., Socha, K., Rossi-Doria, O.	+	+				+	Course (University)	Melez metasezgisel
2006	Beyrouthy, Camille / Burke, Edmund K / Landa-Silva, Dario / McCollum, Barry / McMullan, Paul / Parkes, Andrew J	+	+				+	Course (university)	Yasaklı arama
2006	Günelay, Y. and Şahin, T.	+		+			+	Course (university)	Hedef programlama
2007	E.K. Burke B.McCollum A.Meisels S. Petrovic R.Qu	+	+				+	Course	Hiper sezgiseller
2007	Beligiannis, G.N., Moschopoulos, C. and Likothanassis, S.D.	+		+			+	School	Genetik algoritmalar
2007	Ismayilova, N.A., Sagir, M. and Gasimov, R.N.	+		+			+	Faculty- Course	Konik skalerleştirme

## EK.2 devam

Gözönüne alınan bileşenler									
Yıl	Yazar	Ders	Derslik	Öğretici	Öğrenci	Sınav	Zaman	Yayımda tanımlanan TT Türü	Kullanılan Yöntem
2007	Dilip Datta and Kalyanmoy Deb and Carlos M. Fonseca	+					+	School	Genetik algoritmalar
2007	Murray, K., Muller, T., and Rudova, H.	+	+				+	Course (university)	Kısıt sağlama
2007	Schimmelpfeng, K. and Helber, S.	+	+				+	Course (university)	Tamsayılı programlama
2007	Gunawan, A., Ng, K.M., and Poh, K.L.	+		+			+	Course (university)	Melez metasezgisel
2007	Avella, P., D'Auria, B., Salerno, S., and Vasil'ev, I.	+		+			+	School	Tavlama benzetimi ve özel bir yerel arama algoritması
2008	Burke, E.K., Marecek, J., Parkes, A.J., and Rudová, H.	+	+				+	Course (university)	Tamsayılı programlama
2008	Lai, L., Hsueh, N., Huang, L., and Chen, T.	+	+				+	Course (university)	Kısıt programlama ve uzman sistemlerin birleştirildiği bir yapay zeka
2008	Beligiannisa, G.N. Moschopouloa, C.N., Kaperonisa, G.P., and Likothanassisa, S.D.	+		+			+	School	Evrimsel algoritmalar
2009	Cheong, C.Y., Tan, K.C., and Veeravalli, B.					+	+	Exam (University)	Evrimsel algoritmalar
2010	Ozturk, Z.K., Ozturk, G. ve Sagir, M.			+		+		Invigilator	Tamsayılı programlama



#### Ek.4: Ders-Derslik-Zaman Çizelgeleme Problemi için RKGGA C# kodu

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace RKGATT
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Problem.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;

namespace RKGATT_FMSG
{
    class problem
    {
        public class adC {
            public int ders;
            public List<int> derslik;
        }
        public class azdC{
            public int derslik;
            public List<int> zd;
        }
        public class dersIliskiC
        {
            public int ders;
            public List<int> iliskilid;
        }

        public int ders_sayisi;
        public int k;
        public int kk;
        public int kkk;
        public int derslik_sayisi;
        public int nitelik_sayisi;
        public int ogrenci_sayisi;
        public int[,] od; // ogrenci ders matrisi
        public string od_str_matris;
    }
}
```

```

        public int[,] sn; // her bir derslikte bulunan nitelik matrisi
        public string sn_str_matris;
        public int[,] dn; //her bir dersin gereksinim duydugu nitelikler
matrisi
        public string dn_str_matris;
        public int[,] id; //ortak ogrencisi olan dersler matrisi
        public string id_str_matris;
        public string genel;
        public string ad_str_matris;
        public string gams_model2;

        public int[] dersMevcudu; //her bir dersi alan toplam öğrenci
sayıları vektörü
        public int[] derslikKapasitesi; //her bir dersliğin kapasitesi
vektörü
        public string dk_str;
        //public int[,] atanabilirDerslikler;
        public adC[] ad; //atanabilir derslikler, class
        public azdC[] azd;
        public dersIliskiC[] dersIliski;
        public int[] ogdersa; //ogrencinin aldigi toplam ders sayisi
        public string ogders_string;
        public int[] k6g; //haftalik ders sayisi 6 dan az olan ogrenci indisi
grubu
        public int[] e78g; //haftalik ders sayisi 7 ve 8 olan ogrenci indisi
grubu

        //public int[,] atanabilirZamanlar;
        //public int[] azds; //atanabilir zaman dilimi sayilari.
        //public int[,] dersIliski; // bir dersin iliskili oldugu tum dersler
        //public int[] dit; // her dersin iliskili oldugu toplam ders sayisi
        public string dm_string ;

        public problem(){ }
        public void problemOku(string dosya)
        {
            StreamReader oku = new StreamReader(dosya);
            string allFile = oku.ReadToEnd();
            oku.Close();
            string[] satirlar = allFile.Split('\n');

            //problem parametrelerinin, ders, derslik, ozellik, ogrenci
sayilarinin ilk satirdan okunmasi
            genel = "";
            string[] degerler = satirlar[0].Split(' ');
            ders_sayisi = Convert.ToInt32(degerler[0]);
            genel += "ders sayisi \t :" + ders_sayisi.ToString() + "\r";
            derslik_sayisi = Convert.ToInt32(degerler[1]);
            genel += "derslik sayisi \t :" + derslik_sayisi.ToString() + "\r";
            nitelik_sayisi = Convert.ToInt32(degerler[2]);
            genel += "nitelik sayisi \t :" + nitelik_sayisi.ToString() + "\r";
            ogrenci_sayisi = Convert.ToInt32(degerler[3]);
            genel += "ogrenci sayisi \t :" + ogrenci_sayisi.ToString() + "\r";

            //atanabilirDerslikler = new int[ders_sayisi, derslik_sayisi];
            ad = new adC[ders_sayisi];
            // sinif kapasitelerinin okunmasi
            derslikKapasitesi = new int[derslik_sayisi];
            dk_str = "";
            int sno=0;
            for (int i = 0; i < derslik_sayisi; i++)
            {
                sno++;
            }
        }
    }

```

```

        derslikKapasitesi[i] = Convert.ToInt32(satirlar[sno]);
    }

    //ogrenci ders matrisini oku
    od_str_matris = "";
    od = new int[ogrenci_sayisi, ders_sayisi];
    for (int i = 0; i < ogrenci_sayisi; i++)
    {
        for (int j = 0; j < ders_sayisi; j++)
        {
            sno++;
            od[i, j] = Convert.ToInt32(satirlar[sno]);
        }
    }

    //derslik nitelik matrisini oku
    sn_str_matris = "";
    sn = new int[derslik_sayisi, nitelik_sayisi];
    for (int i = 0; i < derslik_sayisi; i++)
    {
        for (int j = 0; j < nitelik_sayisi; j++)
        {
            sno++;
            sn[i, j] = Convert.ToInt32(satirlar[sno]);
        }
    }

    //ders nitelik matrisinin okunması
    dn = new int[ders_sayisi, nitelik_sayisi];
    dn_str_matris = "";
    for (int i = 0; i < ders_sayisi; i++)
    {
        for (int j = 0; j < nitelik_sayisi; j++)
        {
            sno++;
            dn[i, j] = Convert.ToInt32(satirlar[sno]);
        }
    }

    //----ders mevcutlerini hesapla
    dm_string = "";
    dersMevcudu = new int[ders_sayisi];
    for (int i = 0; i < ders_sayisi; i++)
    {
        int toplamAlinanDers = 0;
        for (int j = 0; j < ogrenci_sayisi; j++)
        {
            toplamAlinanDers += od[j, i];
        }
        dersMevcudu[i] = toplamAlinanDers;
        dm_string += dersMevcudu[i].ToString() + "\r";
    }

    //----ogrencinin aldigi ders sayilarini hesapla
    ogders_string = "";

    ogdersa = new int[ogrenci_sayisi];
    k6g=new int[ogrenci_sayisi];
    e78g=new int[ogrenci_sayisi];

    for (int i = 0; i < ogrenci_sayisi; i++)
    {

```

```

int toplamDers = 0;
for (int j = 0; j < ders_sayisi; j++)
{
    toplamDers += od[i, j];
}
ogdersa[i] = toplamDers;

if (ogdersa[i] <= 6)
{
    k6g[k] = i;
    k++;
}

if (ogdersa[i] == 7)
{
    e78g[kk] = i;
    kk++;
}

if (ogdersa[i] == 8)
{
    e78g[kk]=i;
    kk++;
}

ogders_string += ogdersa[i].ToString() + "\r";
}

//----iliskili dersler matrisini olustur
dersIliski = new dersIliskiC[ders_sayisi];
id = new int[ders_sayisi, ders_sayisi];
for (int i = 0; i < ders_sayisi; i++)
{
    dersIliski[i] = new dersIliskiC();
    dersIliski[i].ders = i;
    dersIliski[i].iliskiliID = new List<int>();
    for (int j = 0; j < ders_sayisi; j++)
    {
        id[i, j] = 0;
        if (i == j) continue;
        for (int k = 0; k < ogrenci_sayisi; k++)
        {
            if ((od[k, i] == 1) && (od[k, j] == 1))
            {
                id[i, j] = 1;
                dersIliski[i].iliskiliID.Add(j);
                break;
            }
        }
    }
}

id_str_matris = "table id(i,ii) \n \t";
string[] id_str = new string[ders_sayisi];
for (int i = 0; i < ders_sayisi; i++) id_str_matris +=
i.ToString() + "\t";
id_str_matris += "\n";
for (int i = 0; i < ders_sayisi; i++)
{
    for (int j = 0; j < ders_sayisi; j++)
        id_str[j]= id[i, j].ToString()+"\t";
}

```

```

        id_str_matris += i.ToString()+"\t"+ string.Concat(id_str)+"\n"
;
    }

    od_str_matris = "table od(k,i) \n \t";
    string[] od_str = new string[ders_sayisi];
    for (int i = 0; i < ders_sayisi; i++) od_str_matris +=
i.ToString() + "\t";
    od_str_matris += "\n";
    for (int i = 0; i < ogrenci_sayisi; i++)
    {
        for (int j = 0; j < ders_sayisi; j++)
            od_str[j] =od[i, j].ToString() + "\t";
        od_str_matris += i.ToString() + "\t" + string.Concat(od_str) +
"\n";
    }

    // ders mevcutlari ve gerekli niteliklere gore derslerin ilgili
siniflarin atanmasi
    string[] ad_str = new string[derslik_sayisi];
    ad_str_matris = "set i /0*" + (ders_sayisi - 1).ToString() +
"/,ia(i),ib(i), j /0*" + (derslik_sayisi - 1).ToString() + "/, t /0*44/, k
/0*" + (ogrenci_sayisi - 1).ToString() + "/; \n";
    ad_str_matris += "alias(j,jj); \n";
    ad_str_matris += "alias(i,ii); \n";
    ad_str_matris += "table ad(i,j) \n";
    ad_str_matris += "\t";
    for (int j = 0; j < derslik_sayisi; j++) ad_str_matris +=
j.ToString() + "\t";
    ad_str_matris += "\n";
    int m;
    for (int i = 0; i < ders_sayisi; i++)
    {
        ad[i] = new adC();
        ad[i].ders = i;
        ad[i].derslik = new List<int>();
        for (int j = 0; j < derslik_sayisi; j++)
        {
            ad_str[j] = "0\t" ;
            if ((derslikKapasitesi[j] >= dersMevcudu[i]))
            {
                for (m = 0; m < nitelik_sayisi; m++)
                {
                    if (dn[i,m] == 1 && sn[j,m] == 0)
                        break;
                }
                if (m == nitelik_sayisi)
                {
                    ad[i].derslik.Add(j);
                    ad_str[j] = "1\t";
                }
            }
        }
        ad_str_matris += i.ToString() + "\t" + string.Concat(ad_str) +
"\n";
    }
    StreamWriter yaz = new StreamWriter("c:\\derslikA\\ad.inc");

    ad_str_matris += ";\n\n";
    id_str_matris += ";\n\n";
    od_str_matris += ";\n\n";
    yaz.Write(ad_str_matris + id_str_matris + od_str_matris);
    yaz.Close();

```

```

}

//her ders icin atanabilir Zamanlar mapini olustur
public void atanabilirzamanlari_sifirla()
{
    azd = new azdC[derslik_sayisi];
    for (int i = 0; i < derslik_sayisi; i++)
    {
        azd[i] = new azdC();
        azd[i].derslik = i;
        azd[i].zd = new List<int>();
        for (int j = 0; j < 45; j++)
            azd[i].zd.Add(j);
    }
}

public void model_olustur() { //gams'e parametre gonderimi...

    gams_model2 = "Sets \n";
    gams_model2 += "i number of students /000*" + (ogrenci_sayisi-
1).ToString("d3") + "\n";
    gams_model2 += "j number of events /000*" + (ders_sayisi-
1).ToString("d3") + "\n";
    gams_model2 += "k number of rooms /000*" + (derslik_sayisi-
1).ToString("d3") + "\n";
    gams_model2 += "l number of features /000*" + (nitelik_sayisi-
1).ToString("d3") + "\n";
    gams_model2 += "t /000*044/\n";
    gams_model2 += "d number of days /000*004/;\n";
    gams_model2 += "Parameters\n";
    string sss="/";

    for (int i = 0; i < derslik_sayisi - 1; i++) sss +=
i.ToString("d3") + " " + derslikKapasitesi[i] + ",";
    sss += (derslik_sayisi - 1).ToString("D3") + " " +
derslikKapasitesi[derslik_sayisi - 1] + "\n";
    gams_model2 += "r(k) room capacities " + sss;
    gams_model2 += "SE(i,j) student event matrix\n";
    gams_model2 += "RF(k,l) room feature matrix\n";
    gams_model2 += "EF(j,l) event feature matrix\n";
    gams_model2 += "M /10000000/;\n\n";
    gams_model2 += " table SE(i,j)\n\n";
    List<string> satirlar = new List<string>();
    string satir=" ";
    for (int j = 0; j < ders_sayisi; j++)
        satir += j.ToString("D3") + " ";
    satir += "\n";
    satirlar.Add(satir);
    for (int i = 0; i < ogrenci_sayisi; i++) {
        satir = string.Format(i.ToString("D3")) + " ";
        for (int j = 0; j < ders_sayisi; j++)
            satir += od[i, j].ToString("D3") + " ";
        satir += "\n";
        satirlar.Add(satir);
    }

    satirlar.Add(" table RF(k,l)\n\n");
    satir = " ";
    for (int j = 0; j < nitelik_sayisi; j++)
        satir += j.ToString("D3") + " ";
    satir += "\n";
    satirlar.Add(satir);
    for (int i = 0; i < derslik_sayisi; i++)

```

```

        {
            satir = string.Format(i.ToString("D3")) + " ";
            for (int j = 0; j < nitelik_sayisi; j++)
                satir += sn[i, j].ToString("D3") + " ";
            satir += "\n";
            satirlar.Add(satir);
        }

        satirlar.Add(" table EF(j,l)\n\n");
        satir = " ";
        for (int j = 0; j < nitelik_sayisi; j++)
            satir += j.ToString("D3") + " ";
        satir += "\n";
        satirlar.Add(satir);
        for (int i = 0; i < ders_sayisi; i++)
        {
            satir = string.Format(i.ToString("D3")) + " ";
            for (int j = 0; j < nitelik_sayisi; j++)
                satir += dn[i, j].ToString("D3") + " ";
            satir += "\n";
            satirlar.Add(satir);
        }

        string[] ss = satirlar.ToArray();
        string s = string.Concat(ss);

        gams_model2 += s;

    }
}

```

bireyTT.cs

```

using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace RKGATT
{
    class bireyTT
    {
        public class zdDersC {
            public int zd;
            public List<int> dersler;
        }
        public class secC {
            public int ders;
            public int deger;
        }
        public class dDersC
        {
            public class derslerC{
                public int ders;
                public int mevcut;
                public int rs;
                public int zd;
            }
        }
    }
}

```

```

        public int derslik;
        public List<derslerC> dersler;
    }
    public class hcv3C {
        public int zd;
        public int ders1;
        public int ders2;
        public int derslik1;
        public int derslik2;
    }

    public List<hcv3C> hcv3ler;
    public int[] kr; //kromozom
    public int f1, f2, f3; //saglanmayan esnek kisit sayilari
    public int ud; // uyum fonksiyonu degeri
    public List<int> nsga_S;
    public int nsga_n;
    public int nsga_rank;
    public double nsga_distance;
    public int[,] cozum;
    public int[,] cozum2;
    public zdDersC[] zdDers;
    public dDersC[] dDers;
    public string yazdir;
    public int hcv3;
    public problem p;
    public List<secC> secim;
    private static Random rs = new Random();
    public bireyTT(problem pb)
    {
        p = pb;
        kr = new int[p.ders_sayisi];
        for (int i = 0; i < p.ders_sayisi; i++)
            kr[i] = rs.Next(10000);
    }
    public void cozumleme()
    {
        cozum = new int[p.ders_sayisi, 3];
        int azdNo, i;
        for (i = 0; i < p.ders_sayisi; i++) cozum[i, 1] = -1;
        p.atanabilirzamanlari_sifirla();

        zdDers = new zdDersC[45];
        for (i = 0; i < 45; i++)
        {
            zdDers[i]=new zdDersC();
            zdDers[i].zd = i;
            zdDers[i].dersler = new List<int>();
        }

        dDers = new dDersC[p.derslik_sayisi];
        for (i = 0; i < p.derslik_sayisi; i++)
        {
            dDers[i] = new dDersC();
            dDers[i].derslik = i;
            dDers[i].dersler = new List<dDersC.derslerC>();
        }
        for (int ddd = 1; ddd <= p.derslik_sayisi; ddd++)
        {
            for (i = 0; i < p.ders_sayisi; i++)
            {
                dDersC.derslerC eklenecekD = new dDersC.derslerC();
                if (ddd == p.ad[i].derslik.Count)
                {

```

```

        if (ddd == 1)
        {
            cozum[i, 0] = p.ad[i].derslik[0]; //ilk atanabilir
derslik ataniyor...
            eklenecekD.ders = i;
            eklenecekD.mevcut = p.dersMevcudu[i];
            eklenecekD.rs = kr[i] % 100;
            dDers[cozum[i, 0]].dersler.Add(eklenecekD);
        }
        else
        {
            int rs1 = ((kr[i] - kr[i] % 100) / 100); //
gendeki degerin ilk iki basamagi--derslik atamasi icin
            double toplam = 0.0;
            for (int ttt = 0; ttt < p.ad[i].derslik.Count;
ttt++)
                toplam += Math.Pow((45-
dDers[p.ad[i].derslik[ttt]].dersler.Count),1);
            int sinifNo = 0; double olasilik = 0.0;
            for (int ttt = 0; ttt < p.ad[i].derslik.Count;
ttt++)
                {
                    olasilik += Convert.ToDouble(Math.Pow((45 -
dDers[p.ad[i].derslik[ttt]].dersler.Count),1) / toplam);
                    if (rs1 < olasilik*100)
                    {
                        sinifNo = ttt;
                        break;
                    }
                }
            cozum[i, 0] = p.ad[i].derslik[sinifNo];
            eklenecekD.ders=i;
            eklenecekD.mevcut=p.dersMevcudu[i];
            eklenecekD.rs=kr[i]%100;
            dDers[cozum[i, 0]].dersler.Add(eklenecekD);
        }
    }
}

for (i = 0; i < p.derslik_sayisi; i++) {
    IEnumerable<dDersC.derslerC> sorgu=
dDers[i].dersler.OrderBy(d1 => d1.rs);
    int sayi = dDers[i].dersler.Count;
    dDers[i].dersler.AddRange(sorgu);
    dDers[i].dersler.RemoveRange(0, sayi);
}
}
//..... zaman dilimi atamasi .....//
List<string> yaz = new List<string>();
for (i = 0; i < p.derslik_sayisi; i++)
{
    yaz.Add(i.ToString() + "\n\n");
    for (int j = 0; j < dDers[i].dersler.Count; j++)
    {
        string ssss = dDers[i].dersler[j].ders.ToString() + "\t" +
dDers[i].dersler[j].mevcut.ToString() + "\t" +
dDers[i].dersler[j].rs.ToString() + "\t" + dDers[i].dersler[j].zd.ToString() +
"\n";
        yaz.Add(ssss);
        int rs2 = dDers[i].dersler[j].rs;           // gendeki degerin
ikinci iki basamagi--zaman dilimi atamasi icin
        List<int> listel = new List<int>();
        int ders = dDers[i].dersler[j].ders;

```

```

        int[] geciciarray = p.azd[cozum[ders, 0]].zd.ToArray();
        listel = geciciarray.ToList();
        foreach (int d in p.dersIliski[ders].iliskiliD)
            if(listel.Count()>1)
                listel.Remove(cozum[d, 1]);
            else
                break;
        for (int d = 0; d < 5; d++)
            if(listel.Count()>1)
                listel.Remove(d*9+8);
            else
                break;

        azdNo = (int)(listel.Count * rs2 / 100);
        cozum[ders, 1] = listel[azdNo];
        zdDers[cozum[ders, 1]].dersler.Add(ders);
        p.azd[cozum[ders, 0]].zd.Remove(cozum[ders, 1]);
    }
}
yazdir = string.Concat(yaz.ToArray());
}

public void hesapla_f1()
{
    int gks1;
    gks1 = 0; // set soft constraint violations to zero to start with
    for (int i = 0; i < p.ders_sayisi; i++)
    {
        if (cozum[i, 1] % 9 == 8)
        {
            gks1 += p.dersMevcudu[i]; // gunun son zaman diliminde
            dersleri olan herbir ogrenciye bir ceza ver
            cozum[i, 2] = 1;
        }
    }
    f1 = gks1;
}

public void hesapla_f2()
{
    int ardisikDersler, gks2;
    gks2 = 0;
    bool attendsTimeslot;

    for (int j = 0; j < p.ogrenci_sayisi; j++)
    {
        ardisikDersler = 0;
        for (int i = 0; i < 45; i++)
        { // bir gun icindeki ardisik derslerin sayilmasi
            if ((i % 9) == 0) ardisikDersler = 0;
            attendsTimeslot = false;
            for (int k = 0; k < zdDers[i].dersler.Count; k++)
            {
                if (p.od[j, zdDers[i].dersler[k]]==1)
                {
                    attendsTimeslot = true;
                    ardisikDersler ++;
                    if (ardisikDersler > 2)
                    {
                        gks2++;
                        cozum[k, 2] += 10;
                    }
                }
            }
        }
    }
}

```

```

        break;
    }
    }
    if (!attendsTimeslot) ardisikDersler = 0;
}
}
f2 = gks2;
}
public void hesapla_f3()
{
    int gundekiDers, gks3, d, t, k, j;
    gks3 = 0;
    for (j = 0; j < p.ogrenci_sayisi; j++)
    {
        int ders = 0;
        gundekiDers = 0;
        for (d = 0; d < 5; d++)
        {
            // her gun icin
            gundekiDers = 0; // gundeki ders sayisi
            for (t = 0; t < 9; t++)
            {
                // gunun her zaman dilimi icin
                for (k = 0; k < zdDers[9 * d + t].dersler.Count; k++)
                {
                    if (p.od[j,zdDers[9*d+t].dersler[k]]==1)
                    {
                        gundekiDers = gundekiDers + 1;
                        ders = k;
                        break;
                    }
                }
                if (gundekiDers > 1) // ogrencinin gunde birden fazla
dersi varsa
                    break; // diger güne geç
            }
            if (gundekiDers == 1) { gks3++; cozum[ders, 2] += 100; }
        }
    }
    f3 = gks3;
}
public void hesapla_fler(){
    cozumleme();
    hesapla_f1();
    hesapla_f2();
    hesapla_f3();
}
public void hesapla_ud(){
    cozumleme();
    hesapla_f1();
    hesapla_f2();
    hesapla_f3();
    hesapla_hcv3();
    ud = (f1 + f2 + f3);

    //----- nsga yonteminde kullanılan degiskenler
sifirlaniyor....
// NSGA classında bu degerler tekrar hesaplanacak..

    nsga_S = new List<int>();
    nsga_n = 0;
    nsga_rank = 0;
    nsga_distance = 0;
    //-----
    secim = new List<secC>();
}

```

```

        for (int i = 0; i < p.ders_sayisi; i++) {
            if (cozum[i, 2] > 0) {
                secC s= new secC();
                s.ders = i;
                s.deger = cozum[i, 2];
                secim.Add(s);
            }
        }
        IEnumerable<secC> sorgu = secim.OrderByDescending(sec =>
sec.deger);
        int buyukluk = secim.Count;
        secim.AddRange(sorgu);
        secim.RemoveRange(0, buyukluk);
    }
    public void hesapla_hcv3()
    {
        hcv3 = new int();
        hcv3ler = new List<hcv3C>();
        for (int i = 0; i < p.ders_sayisi-1; i++) {
            for (int j= i+1; j<p.ders_sayisi;j++)
                if ((p.id[i, j] == 1) && (cozum[i, 1] == cozum[j, 1]))
                {
                    hcv3++;
                    hcv3C h = new hcv3C();
                    h.ders1 = i;
                    h.ders2 = j;
                    cozum[i, 2] += 10000;
                    cozum[j, 2] += 10000;
                    h.derslik1 = cozum[i,0];
                    h.derslik2 = cozum[j,0];
                    h.zd= cozum[i,1];
                    hcv3ler.Add(h);
                }
        }
    }

    public void cozum2_olustur() {
        cozum2 = new int[p.derslik_sayisi, 45];
        for (int i = 0; i < p.derslik_sayisi; i++)
            for (int j=0;j<45;j++)
                cozum2[i,j] = -1;
        for (int i = 0; i < p.ders_sayisi; i++) cozum2[cozum[i, 0],
cozum[i, 1]] = i;
    }

    public void localsearch() {
        cozum2_olustur();
        for (int i = 0; i < hcv3ler.Count; i++)
        {
            int zd = hcv3ler[i].zd;
            int ders1 = hcv3ler[i].ders1;
            int ders2 = hcv3ler[i].ders2;
            int derslik1 = hcv3ler[i].derslik1;
            int derslik2 = hcv3ler[i].derslik2;
            int zd_degis=-1;
            for (int zdi = 0; zdi < 45; zdi++)
            {
                if (zdi == zd) continue;
                bool degistir = true;
                for (int di = 0; di < p.derslik_sayisi; di++)
                {
                    if (di == derslik1) continue;
                    if (cozum2[di, zdi] != -1)

```

```

        {
            if (p.id[ders1, cozum2[di, zdi]] == 1)
            {
                degistir = false;
                break;
            }
        }
    }
    if (degistir)
        zd_degis = zdi;
    else
        continue;

    if (zd_degis != -1)
    {
        if (cozum2[derslik1, zd_degis] == -1)
        {
            cozum2[derslik1, zd_degis] = cozum2[derslik1, zd];
            cozum[ders1, 1] = zd_degis;
            break;
        }
        else
        {
            degistir = true;
            int ders = cozum2[derslik1, zd_degis];
            for (int di = 0; di < p.derslik_sayisi; di++)
            {
                if (di == derslik1) continue;
                if (cozum2[di, zd] != -1)
                {
                    if (p.id[ders, cozum2[di, zd]] == 1)
                    {
                        degistir = false;
                        break;
                    }
                }
            }
            if (degistir)
            {
                int gecici = cozum2[derslik1, zd_degis];
                cozum2[derslik1, zd_degis] = cozum2[derslik1,

zd];

                cozum2[derslik1, zd] = gecici;
                cozum[ders1, 1] = zd_degis;
                cozum[ders, 1] = zd;
                break;
            }
        }
    }
}
}
}
hesapla_f1();
hesapla_f2();
hesapla_f3();
hesapla_hcv3();
ud = f1 + f2 + f3 ;
}

} //class
} //namespace RKGATT

RKA.cs
sing System;

```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.IO;

namespace RKGATT_FMSG
{
    class RKGA
    {
        public double goc_orani;
        public double elitizm_orani;
        public double tura_olasiligi;
        public int tekrar_sayisi;
        public int pop_buyuklugu;
        public int EBS; //Elitizm ile olusturulacak birey sayisi
        public int GBS; //Goc ile olusturulacak birey sayisi
        public int CBS; //Caprazlama ile olusturulacak birey sayisi
        public List<bireyTT> pop;
        private List<bireyTT> Spop; // //sonraki populasyon
        public problem p;
        public string s;
        public int tekrar;

        public class doubint
        {
            public int x;
            public double y;
        }
        public List<doubint> noktalar = new List<doubint>();

        public RKGA(problem prob, double eo, double go, double to, int ts, int
pb)
        {
            p = prob;
            elitizm_orani = eo;
            goc_orani = go;
            tura_olasiligi = to;
            tekrar_sayisi = ts;
            pop_buyuklugu = pb;

            EBS = Convert.ToInt32(pop_buyuklugu * elitizm_orani);
            if (EBS < 1) EBS = 1;
            GBS = Convert.ToInt32(pop_buyuklugu * goc_orani);
            if (GBS < 1) GBS = 1;
            CBS = pop_buyuklugu - EBS - GBS;
            baslangic();
        }

        //-----
        void baslangic()
        {
            pop = new List<bireyTT>();
            for (int i = 0; i < pop_buyuklugu; i++)
            {
                pop.Add(new bireyTT(p));
                pop[i].hesapla_ud();
            }
        }
        void elitizm(){
            //populasyon en iyi EBS kadar bireyi ilk siralara aliniyor.
            Spop = new List<bireyTT>();
            IEnumerable<bireyTT> sorgu = pop.OrderBy(pop1 => pop1.ud);

```

```

        Spop.AddRange(sorgu);
        Spop.RemoveRange(EBS, pop_buyuklugu - EBS);
    }

    void caprazlama()
    {
        Random rs = new Random();
        int i, j, b1, b2; // caprazlanacak bireyleri sec birey1, birey2
        bireyTT yedekb;

        for (i = EBS; i < pop_buyuklugu - GBS; i++)
        {
            int faktor = 1;
            if (rs.Next(100) < 60)
                b1 = rs.Next(Convert.ToInt16(pop_buyuklugu / 2));
            else
            {
                b1 = 0; // pop_buyuklugu - 1;
                faktor = 0;
            }
            b2 = rs.Next(pop_buyuklugu);

            yedekb = new bireyTT(p);
            for (j = 0; j < p.ders_sayisi; j++)
            {
                if (rs.Next(100) < faktor*tura_olasiligi*100)
                    yedekb.kr[j] = pop[b1].kr[j];
                else
                    yedekb.kr[j] = pop[b1].kr[j] - pop[b1].kr[j] % 100 +
                    pop[b2].kr[j] % 100;
            }
            yedekb.hesapla_ud();
            Spop.Add(yedekb);
        }
    }

    void goc()
    {
        bireyTT yedekb;
        for (int i = 0; i < GBS; i++)
        {
            yedekb = new bireyTT(p);
            yedekb.hesapla_ud();
            Spop.Add(yedekb);
        }
    }

    public void calistir()
    {
        Process proc = new Process();
        proc.StartInfo.FileName = "C:\\Program Files\\GAMS23.3\\gams.exe";
        proc.StartInfo.Arguments = "C:\\derslikA\\a.gms
gdx=C:\\derslikA\\sonuc";
        proc.StartInfo.Domain = "C:\\derslikA\\";
        proc.StartInfo.CreateNoWindow = true;
        proc.Start();

        proc.WaitForExit();
        StreamReader oku = new StreamReader("C:\\derslikA\\p.txt");
        string tumveriler = oku.ReadToEnd();
        oku.Close();
        string[] satirlar = tumveriler.Split(new char[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        bireyTT b = new bireyTT(p);
        for (int i = 0; i < p.ders_sayisi; i++) {

```

```

        b.kr[i] = Convert.ToInt16(satirlar[i].Trim('\r'));
    }
    b.hesapla_ud();
    b.cozum2_olustur();
    s = "";
    int k = 0;
    double onceki;
    Random rs = new Random();
    for (tekrar=0;tekrar<tekrar_sayisi;tekrar++)
    {
        onceki = pop[0].ud;
        elitizm();
        caprazlama();
    }
    goc();
    pop = Spop;
    //pop[0].localsearch();
    doubint yeniNokta = new doubint();
    if (onceki != pop[0].ud)
    {
        yeniNokta.x = k++;
        yeniNokta.y = pop[0].ud;
        noktalar.Add(yeniNokta);
        //s += tekrar.ToString() + ".\t" + pop[0].ud + "\n";
        s += tekrar.ToString() + "\t" + pop[0].ud + ":\t" + pop[0].f1
+ "\t" + pop[0].f2 + "\t" + pop[0].f3 + "\t" + "\n";
    }
}
}
}
}

```

## Ek.5: Ders-Derslik-Zaman Çizelgeleme Problemi için NSGA-II temelli RKGGA C#

### kodu

NSGA.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Diagnostics;
using System.IO;

namespace RKGATT_FMSG
{
    class NSGA
    {
        public double goc_orani;
        public int tekrar_sayisi;
        public int GBS;
        public int CBS;
        public double tura_olasiligi;
        public int pop_buyuklugu;
        public List<bireyTT> pop;
        private List<bireyTT> Spop; // //sonraki populasyon
        public problem p;
        public string s;
        public int tekrar;
        private int amacs;
        public class etkinYuzeylerC
        {
            public int yuzey;
            public List<int> birey;
        }
        public List<etkinYuzeylerC> EY;
        public etkinYuzeylerC kopyaYuzey;

        public class doubint
        {
            public int x;
            public double y;
        }
        public List<doubint> noktalar = new List<doubint>();

        public NSGA(problem prob, double go, int ts, int pb)
        {
            p = prob;
            goc_orani = go;
            tekrar_sayisi = ts;
            pop_buyuklugu = pb;

            GBS = Convert.ToInt32(pop_buyuklugu * goc_orani);
            if (GBS < 1) GBS = 1;
            CBS = pop_buyuklugu - GBS;

            baslangic();
        }

        void baslangic()
        {
```

```

    pop = new List<bireyTT>();
    for (int i = 0; i < pop_buyuklugu; i++)
    {
        pop.Add(new bireyTT(p));
        pop[i].hesapla_fler();
    }
    amacs=3;
}
void caprazlama()
{
    Random rs = new Random();
    int i, j, b1, b2; // caprazlanacak bireyleri sec birey1, birey2
    bireyTT yedekb;

    for (i = 0; i < pop_buyuklugu - GBS; i++)
    {
        b1 = rs.Next(pop_buyuklugu);
        b2 = rs.Next(pop_buyuklugu);

        /* if (pop[b1].ud > pop[b2].ud)
        {
            int gecici = b1;
            b1 = b2;
            b2 = gecici;
        }*/
        yedekb = new bireyTT(p);
        for (j = 0; j < p.ders_sayisi; j++)
        {
            if (rs.Next(100) < tura_olasiligi * 100 &&
pop[b1].cozum[j, 2] == 0)
                yedekb.kr[j] = pop[b1].kr[j];
            else
                yedekb.kr[j] = pop[b2].kr[j];
        }
        yedekb.hesapla_fler();
        Spop.Add(yedekb);
    }
}
void caprazlama2()
{
    Random rs = new Random();
    int i, j, b1, b2; // caprazlanacak bireyleri sec birey1, birey2
    bireyTT yedekb;

    for (i = 0; i < pop_buyuklugu - GBS; i++)
    {
        int faktor = 1;
        if (rs.Next(100) < 60)
            b1 = rs.Next(Convert.ToInt16(pop_buyuklugu / 2));
        else
        {
            b1 = 0; // pop_buyuklugu - 1;
            faktor = 0;
        }
        b2 = rs.Next(pop_buyuklugu);

        yedekb = new bireyTT(p);
        for (j = 0; j < p.ders_sayisi; j++)
        {
            if (rs.Next(100) < faktor * tura_olasiligi * 100)
                yedekb.kr[j] = pop[b1].kr[j];
            else

```

```

        yedekb.kr[j] = pop[b1].kr[j] - pop[b1].kr[j] % 100 +
pop[b2].kr[j] % 100;
    }
    yedekb.hesapla_fler();
    Spop.Add(yedekb);
}
}
void goc()
{
    Spop = new List<bireyTT>();
    bireyTT yedekb;
    for (int i = 0; i < GBS; i++)
    {
        yedekb = new bireyTT(p);
        yedekb.hesapla_fler();
        Spop.Add(yedekb);
    }
}
void yuzeyleriBelirle() {
    EY=new List<etkinYuzeylerC>();
    Spop = new List<bireyTT>(); //yuzey belirleme isleminde sonraki
populasyonu belirlemek icin spop ikinci kez tekrar baslatiliyor..

    kopyaYuzey = new etkinYuzeylerC();
    kopyaYuzey.birey = new List<int>();
    for (int i = 0; i < pop.Count; i++) { //populasyondaki birey
sayisi ikiye
        pop[i].nsga_n = 0;
        pop[i].nsga_S = new List<int>();
        for (int j = 0; j < pop.Count; j++) {
            if (i != j){
                if (baskinmi(i,j)){
                    pop[i].nsga_S.Add(j);
                }else if(baskinmi(j,i)) {
                    pop[i].nsga_n++;
                }
            }
        }

        if (pop[i].nsga_n == 0) {
            pop[i].nsga_rank = 1;
            kopyaYuzey.birey.Add(i);
        }
    }
    EY.Add(kopyaYuzey);

    int yuzeyNo = 1;
    while (kopyaYuzey.birey.Count != 0) {
        etkinYuzeylerC yeniYuzey = new etkinYuzeylerC();
        yeniYuzey.birey = new List<int>();
        for (int j = 0; j < kopyaYuzey.birey.Count; j++) {
            int pp = kopyaYuzey.birey[j];
            for (int k = 0; k < pop[pp].nsga_S.Count; k++) {
                int qq = pop[pp].nsga_S[k];
                pop[qq].nsga_n--;
                pop[qq].nsga_n--;
                if (pop[qq].nsga_n == 0) {
                    pop[qq].nsga_rank = yuzeyNo + 1;
                    yeniYuzey.birey.Add(qq);
                }
            }
        }
        yuzeyNo++;
        kopyaYuzey = yeniYuzey;
    }
}

```

```

        EY.Add(kopyaYuzey);
    }
    int toplam = 0;
    for (int i = 0; i < EY.Count; i++) {
        toplam += EY[i].birey.Count;
        if (toplam < pop_buyuklugu)
        {
            for (int j = 0; j < EY[i].birey.Count; j++)
                Spop.Add(pop[EY[i].birey[j]]);
        }
        else {
            nsga_crowdingD(EY[i]);
            toplam -= EY[i].birey.Count;
            int fark = pop_buyuklugu - toplam;
            int j = 0;
            // fark sadece sonraki pop'u N'e tamamlayacak sekilde
            uzakliga gore siralaniyor..
            // o yuzden asagidaki ii dongusu fark'a kadar
            donduruluyor..
            for (int ii = 0; ii < fark; ii++) {
                for (int jj = ii+1; jj < EY[i].birey.Count; jj++) {
                    if (pop[EY[i].birey[jj]].nsga_distance >
                        pop[EY[i].birey[ii]].nsga_distance) {
                        int gecici = EY[i].birey[ii];
                        EY[i].birey[ii] = EY[i].birey[jj];
                        EY[i].birey[jj] = gecici;
                    }
                }
            }
            while (toplam < pop_buyuklugu)
            {
                Spop.Add(pop[EY[i].birey[j]]);
                j++; toplam++;
            }
            break;
        }
    }
}

void nsga_crowdingD(etkinYuzeylerC yuzey)
{
    int[,] objs = new int[yuzey.birey.Count, 4];
    int[,] Ikumesi = new int[yuzey.birey.Count, 4];
    for (int i = 0; i < yuzey.birey.Count; i++) {
        Ikumesi[i, 1] = yuzey.birey[i];
        Ikumesi[i, 2] = yuzey.birey[i];
        Ikumesi[i, 3] = yuzey.birey[i];

        objs[i, 1] = pop[yuzey.birey[i]].f1;
        objs[i, 2] = pop[yuzey.birey[i]].f2;
        objs[i, 3] = pop[yuzey.birey[i]].f3;

        pop[yuzey.birey[i]].nsga_distance = 0;
    }

    for (int j = 1; j <= 3; j++) {
        for (int i = 0; i < yuzey.birey.Count-1; i++) {
            for (int k = 0; k < yuzey.birey.Count; k++) {
                if (objs[k, j] < objs[i, j]) {
                    int gecici = objs[k, j];
                    objs[k, j] = objs[i, j];
                }
            }
        }
    }
}

```

```

        objs[i, j] = gecici;

        int geciciIndis = Ikumesi[k, j];
        Ikumesi[k, j] = Ikumesi[i, j];
        Ikumesi[i, j] = geciciIndis;
    }
}
int ilk = Ikumesi[0, j];
int son = Ikumesi[yuzey.birey.Count-1, j];

pop[ilk].nsga_distance = 10000;
pop[son].nsga_distance = 10000;

for (int i = 1; i < yuzey.birey.Count-1; i++)
    if (objs[yuzey.birey.Count - 1, j] > objs[1, j])
        pop[Ikumesi[i, j]].nsga_distance += (objs[i+1, j]-
objs[i-1, j])/(objs[yuzey.birey.Count-1, j]-objs[1, j]);

}

}

public void calistir()
{
    Process proc = new Process();
    proc.StartInfo.FileName = "C:\\Program Files\\GAMS23.3\\gams.exe";
    proc.StartInfo.Arguments = "C:\\derslikA\\a.gms
gdx=C:\\derslikA\\sonuc";
    proc.StartInfo.Domain = "C:\\derslikA\\";
    proc.StartInfo.CreateNoWindow = true;
    proc.Start();

    proc.WaitForExit();
    StreamReader oku = new StreamReader("C:\\derslikA\\p.txt");
    string tumveriler = oku.ReadToEnd();
    oku.Close();
    string[] satirlar = tumveriler.Split(new char[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
    bireyTT b = new bireyTT(p);
    for (int i = 0; i < p.ders_sayisi; i++)
    {
        b.kr[i] = Convert.ToInt16(satirlar[i].Trim('\r'));
    }
    b.hesapla_fler();
    b.cozum2_olustur();
    s = "";
    int k = 0;
    double onceki;

    Random rs = new Random();
    for (tekrar = 0; tekrar < tekrar_sayisi; tekrar++)
    {
        onceki = pop[0].f1 + pop[0].f2 + pop[0].f3;
        goc(); // spop burada tariflendiginde goc ilk siraya alindi
        if (rs.Next(100) < 25) caprazlama();
        else
        {
            pop[pop_buyuklugu - 1] = b;
        }
    }
}

```

```

        caprazlama2();
    }
    pop.AddRange(Spop);
    yuzeyleriBelirle(); //nsga mantigina gore amacfonksiyonu
degerlerini hesaplar.. /fastNonDominatedSort-NSGA
//nsga_crowdingD() //crowding distance sadece n. siraya denk
gelen etkin yuzey icin hesaplanacak...
    pop = Spop; // burada pop'a atanan Spop yuzeybelirle alt
programinda tekrar olusturuluyor...
    double toplam = pop[0].f1 + pop[0].f2 + pop[0].f3;
    if (onceki != toplam )
    {
        //yeniNokta.x = k++;
        //yeniNokta.y = pop[0].ud;
        //noktalar.Add(yeniNokta);
        s += tekrar.ToString() + "\t" + toplam + ":\t" + pop[0].f1 +
"\t" + pop[0].f2 + "\t" + pop[0].f3 + "\t" + "\n";
    }
}

bool baskinmi(int a, int b) {
    if (pop[a].f1 > pop[b].f1) return false;
    if (pop[a].f2 > pop[b].f2) return false;
    if (pop[a].f3 > pop[b].f3) return false;
    return true;
}

}
}

```

**Ek.6: Sınav-Derslik-Zaman Çizelgeleme Problemi için 2005-2006 Güz Dönemi 2. Arasınava Çizelgesi**

**2005-2006 Güz dönemi 2. arasınava çizelgesi<sup>13</sup>**

Tarih	Saat	Sınav	Derslik	Mevcut
12.12.2005	09:00	Fizik I (I. Öğr.)	End 6 + End 10	61
	09:00	Fizik I (II. Öğr.)	FL	72
	09:00	Physics I	End 7	34
	11:00	Yöneylem Araş. I (I. Öğr.)	End 6 + End 7	72
	11:00	Yöneylem Araş. I (II. Öğr.)	FL	72
	11:00	Operations Research I	End 10	27
	14:00	Mühendislik Malzemeleri (I. Öğr.)	End 6 + End 7	79
	14:00	Mühendislik Malzemeleri (II. Öğr.)	FL	71
	16:00	İnsan Kaynakları Yön.	End 10	30
	16:00	Pazarlama Yön. (I. Öğr.)	End 7	40
16:00	Pazarlama Yön. (II. Öğr.)	FL	61	
13.12.2005	09:00	Matematik I (I. Öğr.)	End 6 + End 7	83
	09:00	Matematik I (II. Öğr.)	FL	58
	09:00	Calculus I	End 10	33
	11:00	İngilizce I (I. Öğr. + II. Öğr.)	FL	23+22
	14:00	Benzetim (I. Öğr.)	End 7 + End 8	70
	14:00	Benzetim (II. Öğr.)	End 4 + End 5	64
	16:00	Diferansiyel Denk. (I. Öğr.)	End 6 + End 7	56
	16:00	Diferansiyel Denk. (II. Öğr.)	FL	60
16:00	Differential Equ.	End 10	26	
14.12.2005	09:00	Türk Dili I (I. Öğr.)	End 7 + End 8	67
	09:00	Türk Dili I (II. Öğr.)	FL	68
	11:00	Bilgisayar Programlama II (I. Öğr.)	FL + End 4	105
	11:00	Bilgisayar Programlama II (II. Öğr.)	End 6 + End 7 + End 10	99
	14:00	Teknik Resim (I. Öğr.)	İnşaat Müh.	67
	14:00	Teknik Resim (II. Öğr.)	İnşaat Müh.	85
	14:00	Mühendislik Ekonomisi (I. Öğr.)	End 6 + End 7	93

<sup>13</sup> Geliştirilen modelde bir dersin I., II. öğretim ile A ve B grupları ve varsa İngilizce olarak açılan grupları, sınavları aynı zaman diliminde yapıldığından, tek bir ders olarak ele alınmıştır. Dolayısıyla, toplamda çizelgelenen sınav sayısı 27 olmuştur.

**2005-2006 Güz dönemi 2. arasınav çizelgesi (devam)**

	14:00	Mühendislik Ekonomisi (II. Öğr.)	FL	72
	18:00	Üretim Planlama (I. Öğr.)	End 6 + End 7	61
	18:00	Üretim Planlama (II. Öğr.)	FL	63
15.12.2005	09:00	Kimya (I. Öğr.)	End 6 + End 10	75
	09:00	Kimya (II. Öğr.)	FL	65
	09:00	Chemistry	End 1	29
	09:00	KKİT	End 2	16
	09:00	ESKA	End 3	13
	09:00	NKV (I. Öğr. + II. Öğr.)	End 7	22+26
	11:00	Olasılık (I. Öğr.)	FL	61
	11:00	Olasılık (II. Öğr.)	End 6 + End 7	78
	11:00	Probability (I. Öğr.)	End 10	28
	14:00	Üretim Yöntemleri (I. Öğr.)	End 6 + End 7	85
	14:00	Üretim Yöntemleri (II. Öğr.)	FL	65
	16:00	Teknik İngilizce III (I. Öğr.)	FL	69
	16:00	Teknik İngilizce III (II. Öğr.)	End 7	47
	16.12.2005	09:00	TBTK (I. Öğr.)	End 7
09:00		TBTK (II. Öğr.)	FL	49
09:00		Teknik İngilizce V (I. Öğr.)	End 1 + End 2	56
09:00		Teknik İngilizce V (II. Öğr.)	End 4 + End 5	45
11:00		Tesis Planlaması (I. Öğr.)	FL	55
11:00		Tesis Planlaması (II. Öğr.)	End 6+End 7	83
14:00		İstatistik II ( I. + II. Öğr.A)	FL	37+20
14:00		İstatistik II ( I. + II. Öğr.B)	End 6+End 7	45+42
16:00		Mühendislik Mekaniği (I.Öğr. A)	End 1 + End 2	44
16:00		Mühendislik Mekaniği (II.Öğr. A)	End 6 +End 7	72
16:00		Mühendislik Mekaniği (I.Öğr. B)	FL	65
	11:00	Sosyal Seçmeli I		
	14:00	İş Etüdü(I.Öğr. A)	End 6 + End 10	55
	14:00	İş Etüdü(II.Öğr. A)	End 7	47
	14:00	İş Etüdü(I.Öğr. B)	FL	68