

Semih ÖZDAMAR

Yüksek Lisans Tezi

KÜ 2015

KIRIKKALE ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI  
YÜKSEK LİSANS TEZİ

SINAV ÇİZELGELEME, SINAV YERİ VE  
GÖZETMEN ATAMA PROBLEMLERİNİN ÇÖZÜMÜ  
İÇİN MODEL ÖNERİLERİ VE UYGULAMASI

Semih ÖZDAMAR

Temmuz 2015

**Endüstri Mühendisliđi Anabilim Dalında** Semih ÖZDAMAR tarafından hazırlanan SINAV ÇİZELGELEME, SINAV YERİ VE GÖZETMEN ATAMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN MODEL ÖNERİLERİ VE UYGULAMASI adlı Yüksek Lisans Tezinin Anabilim Dalı standartlarına uygun olduğunu onaylarım.

Prof. Dr. Burak BİRGÖREN  
Anabilim Dalı Başkanı

Bu tezi okuduđumu ve tezin Yüksek Lisans Tezi olarak bütün gereklilikleri yerine getirdiđini onaylarım.

Doç. Dr. Süleyman ERSÖZ  
Danışman

Jüri Üyeleri

Başkan : Prof. Dr. Burak BİRGÖREN \_\_\_\_\_  
Üye (Danışman) : Doç. Dr. Süleyman ERSÖZ \_\_\_\_\_  
Üye : Doç. Dr. Mehmet KABAK \_\_\_\_\_

14/07/2015

Bu tez ile Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onaylamıştır.

Prof. Dr. Mustafa YİĞİTOđLU  
Fen Bilimleri Enstitüsü Müdürü

## ÖZET

### SINAV ÇİZELGELEME, SINAV YERİ VE GÖZETMEN ATAMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN MODEL ÖNERİLERİ VE UYGULAMASI

ÖZDAMAR, Semih

Kırıkkale Üniversitesi

Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi

Danışman: Doç. Dr. Süleyman ERSÖZ

Temmuz 2015, **143 sayfa**

Hizmet sistemlerinden biri olan eğitim kurumlarında akademik personel, derslik ve öğrenciler kapasite yönetiminin önemli unsurlarını oluşturmaktadır. Kapasitenin etkin kullanılması, akademik personelin katma değerli işlerde daha fazla mesai harcaması, öğrencilerin ihtiyaçlarının doğru bir şekilde analizi ve yönetimin daha verimli strateji üretmesini sağlamaktadır. Eğitim kurumlarında; her yarıyılıda uygulanan ara sınav, final ve mazeret sınavları ile yaz döneminde uygulanan bütünleme, not yükseltme, tek ders ve engel sınavları için etkin planlanma yapılması hizmet kalitesi açısından önemli bir faaliyet olarak karşımıza çıkmaktadır. Bu çalışmada da, sınav çizelgeleme, sınav yeri ve gözetmen atama problemlerinin çözümü için dört ayrı model geliştirilmiştir. Sınav çizelgeleme modelinde, öğretmenlerin mazeretleri ile öğrencilerin isteklerinin dikkate alınarak bir öğrenciye aynı oturumda birden fazla sınav gelmeyecek şekilde sınav tarih ve oturumları belirlenmektedir. Sınav yeri atama modelinde, oturumları ilk modelde belirlenen sınavlar için her bir oturumda sınav salonu/dersliklerdeki kapasite aşılmadan, bir arada yapılabilecek ve yapılamayacak sınavlar dikkate alınarak hangi kısım/şube/öğrencinin hangi sınavda nerede sınava gireceği tespit edilmektedir. Gözetmen atama modelinde ise daha önce atama yapılan sınav salonu/dersliklerdeki sınava girecek öğrenci sayısına bağlı olarak belirlenen gözetmen ihtiyacına göre; personelin bu süreçteki mazeretleri, art arda gelen oturumlarda sınav görevi yazılmaması, oturumların ortalaması kadar görev yazılması, görev süresinin dengelenmesi ile personelin kıdem düzeyi dikkate alınarak gözetmenlik görevleri

yazılmaktadır. Yöntem olarak karma tam sayılı programlama ile hedef programlama kullanılmıştır. Modellerin kodlanması Ilog Cplex Optimization Studio 12.6 ile gerçekleştirilmiştir. Çalışma bir yükseköğrenim kurumunda uygulanmıştır. Uygulama sonuçları ayrıntılı olarak analiz edilmiştir. Ayrıca; kullanıcıların veri girişi, analizi ve uygulama sonuçlarını rahat bir şekilde tamamlayabilmesi için bir veri tabanı ile arayüz uygulaması da gerçekleştirilmiştir.

**Anahtar kelimeler:** Sınav Çizelgeleme, Sınav Yeri ve Gözetmen Atama Problemi, 0-1 Tam Sayılı Programlama, Karma Tam Sayılı Programlama, Hedef Programlama.

## **ABSTRACT**

### **MODEL PROPOSALS AND SOLUTIONS FOR EXAM SCHEDULING, EXAM LOCATION AND PROCTOR ASSIGNMENT PROBLEMS WITH APPLICATION**

ÖZDAMAR, Semih

Kırıkkale University

Graduate School Of Natural and Applied Sciences

Department of Industrial Engineering, M.Sc. Thesis

Supervisor: Assoc. Prof. Süleyman ERSÖZ

July 2015, **143 pages**

Academic staff, classrooms and students are important components of capacity management in educational institutions, one of the service systems. Efficient use of capacity ensures that academic staffs spend more working time on value added works, an accurate analysis of students' needs and generation of more productive strategies by the management. In educational institutions; the efficient planning of mid-term, final and excuse exams of every academic term and the make-up, increase grade, single course and excuse exams conducted in summer period pose to be an important activity for the quality of services. In this study, to solve the problems of exam scheduling, assignment of proctors and exam locations, four different models have been developed. In the scheduling model, exam dates and sessions are determined by considering student excuses and demands while ensuring that no one student has more than one exam in every session. In exam location assignment model for the exams whose sessions are determined in the first model, where each section/branch/student will take the exam is determined by considering the exams that could be held together with other exams and without the others and without exceeding the capacity of the exam locations/classrooms. In the proctor assignment model, the number of proctors needed is determined according to the number of students who will take the exam in the exam locations/classrooms which were assigned earlier. The supervision of the exams are written by taking the following factors into consideration: the excuses of

the staff during the exam period, not assigning same proctors in consecutive sessions, assigning proctorship by the average number of sessions and balancing the assignment period with the rank level of the staff. For the method, mixed-integer programming and goal programming have been used. Ilog Cplex Optimization Studio 12.6 has been used for encoding the models. The study has been implemented in an undergraduate school. The results of the implementation have been analyzed in detail. Furthermore; a database and an interface application have been produced in order to help the users with the entry and analysis of data, and complete the exam results easily.

**Key Words:** Exam Scheduling, Exam Location and Proctor Assignment Problems, 0-1 Integer Programming, Mixed-Integer Programming, Goal Programming.

## TEŐEKKÜR

Tezimin hazırlanması esnasında hiçbir yardımı esirgemeyen ve biz genç arařtırmacılara büyük destek olan, bilimsel deney imkanlarını sonuna kadar bizlerin hizmetine veren, tez yöneticisi hocam, Sayın Doç.Dr.Süleyman ERSÖZ'e, tez çalışmalarım esnasında, bilimsel konularda daima yardımını gördüğüm hocam Sayın Doç.Dr.Mustafa YÜZÜKIRMIZI'ya, Yrd.Doç.Dr.Adnan AKTEPE'ye, Dr.Cüneyt SEVİM'e ve Dr.Ercan TURGUT'a, bilgisini esirgemeyerek bana destek olan değerli arkadaşım Dr.Erkam GÜREŐEN'e, tezimin birçok aşamasında yardım ve destek gördüğüm Ahmet OKYAY, Kemal ÇİZMECİLER ile Muammer ÖZÇELİK ve diğer mesai arkadaşlarıma, son olarak bana birçok konuda olduğu gibi, tezimi hazırlamam esnasında da yardımlarını esirgemeyen aileme teşekkür ederim.

# İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
<b>ÖZET</b> .....	I
<b>ABSTRACT</b> .....	III
<b>TEŞEKKÜR</b> .....	V
<b>İÇİNDEKİLER DİZİNİ</b> .....	VI
<b>ÇİZELGELER DİZİNİ</b> .....	VIII
<b>ŞEKİLLER DİZİNİ</b> .....	IX
<b>KISALTMALAR DİZİNİ</b> .....	XI
<b>1. GİRİŞ</b> .....	1
<b>2. LİTERATÜR ARAŞTIRMASI</b> .....	4
<b>3. SINAV ÇİZELGELEME, SINAV YERİ VE GÖZETMEN ATAMA PROBLEMLERİ</b> .....	9
3.1. Sınav Çizelgeleme Problemi .....	10
3.1.1. Sınav Takvimi ve Sınav Gün İsteklerinin Oluşturulması .....	11
3.1.2. Sınav Oturumlarının Belirlenmesi .....	13
3.2. Sınav Yeri Atama Problemi .....	15
3.2.1. Kısım Bazlı Atama Problemi .....	16
3.2.2. Öğrenci Bazlı Atama Problemi .....	17
3.3. Gözetmen Atama Problemi.....	20
<b>4. SINAV ÇİZELGELEME, SINAV YERİ VE GÖZETMEN ATAMA PROBLEMLERİNDE KULLANILAN ÇÖZÜM YÖNTEMLERİ</b> .....	25
4.1. Tam Sayılı Programlama .....	26
4.1.1. Dal-Sınır Algoritması.....	27
4.1.2. Ekleme Algoritması .....	29
4.1.3. Kesme Düzlemi Algoritması.....	30
4.1.4. Tam Sayılı Programlamada Kullanılan Özel Durumlar.....	31
4.2. Ulaştırma ve Atama Problemleri .....	34
4.2.1. Ulaştırma Problemleri .....	35
4.2.2. Atama Problemleri (Macar Metodu).....	45
4.3. Hedef Programlama .....	48
4.4. Matematiksel Modellemede Tablo İndis Kullanımı .....	50



4.5. İlişkisel Veritabanı .....	52
4.5.1. İlişkisel Veritabanının Özellikleri .....	53
4.5.2. Anahtar Alanlar ve İlişkiler.....	53
4.5.3. İlişkisel Bütünlük Kuralı .....	54
4.5.4. Normalizasyon .....	54
<b>5. GELİŞTİRİLEN MODELLER.....</b>	<b>55</b>
5.1. Sınav Çizelgeleme için Geliştirilen Model .....	55
5.2. Sınav Yeri Atama için Geliştirilen Model .....	59
5.2.1. Kısım Bazlı Atama için Geliştirilen Model .....	59
5.2.2. Öğrenci Bazlı Atama için Geliştirilen Model .....	62
5.3. Gözetmen Atama için Geliştirilen Model .....	66
<b>6. UYGULAMA .....</b>	<b>73</b>
6.1. Veritabanı Şema Tasarımı.....	73
6.2. Tablo ve Sorguların Oluşturulması .....	74
6.3. Veritabanı Ara Yüzü .....	74
6.3.1. Giriş ve Ana Sayfa .....	75
6.3.2. Sınav Çizelgeleme Modülü .....	77
6.3.3. Sınav Yeri Atama Modülü .....	82
6.3.4. Gözetmen Atama Modülü .....	87
6.4. Uygulama Sonuçları.....	90
6.4.1. Sınav Çizelgeleme Sonuçları .....	90
6.4.2. Sınav Yeri Atama Sonuçları.....	93
6.4.3. Gözetmen Atama Sonuçları .....	96
<b>7. SONUÇ VE ÖNERİLER .....</b>	<b>99</b>
<b>KAYNAKLAR .....</b>	<b>103</b>
<b>EKLER.....</b>	<b>107</b>
EK-1 Ilog Sınav Çizelgeleme Modeli .....	107
EK-2 Ilog Kısım Bazlı Sınav Yeri Atama Modeli .....	112
EK-3 Ilog Öğrenci Bazlı Sınav Yeri Atama Modeli .....	116
EK-4 Ilog Gözetmen Atama Modeli .....	121
EK-5 Ms Access Veri Tabanı Tablo İlişkileri.....	129
<b>ÖZGEÇMİŞ.....</b>	<b>130</b>

## ÇİZELGELER DİZİNİ

<u>ÇİZELGE</u>	<u>Sayfa</u>
3.1. Sınav Takvimi .....	11
3.2. Öğrenci ve Öğretmen için Sınav Günü İstek Tablosu .....	12
3.3. Sınav Günü İstek Tablosu .....	12
3.4. Kümülatif Oturum Tablosu .....	14
3.5. Sınav Kümülatif Oturum Tablosu .....	15
3.6. Sınav Uygulama Stratejileri .....	15
3.7. Sınav Yerleri Tablosu .....	16
3.8. Sınav Kısım Tablosu .....	17
3.9. Sınav Yeri Teorik Sınav Şablonu.....	18
3.10. Sınav Yeri Mevcuduna Göre Teorik Sınav Mevcutları .....	19
3.11. Sınav Öğrenci Tablosu .....	20
3.12. Personel Gözetmenlik Mazeret Tablosu .....	21
3.13. Gözetmenlik Yazılabilecek Personel Listesi.....	21
3.14. Gözetmenlik Atama Tablosu .....	22
3.15. Gözetmenlik Talep Tablosu .....	23
3.16. Gözetmenlik Atama Sonuçları Tablosu .....	24
4.1. Ya-Ya da Kısıtları Dönüşümü.....	32
4.2. Taşıma Modeli Örneği .....	39
4.3. KBK Çözümü.....	41
4.4. EKM Çözümü .....	41
4.5. İki İndisli Bir Değişkenin Tablo Karşılığı .....	50
4.6. İndise Bağlı Tablo Yapısı .....	51
6.1. Kısım Bazlı Sınav Salonu Atama Sonuçları .....	93
6.2. Öğrenci Bazlı Sınav Salonu Atama Sonuçları-1 .....	94
6.3. Öğrenci Bazlı Sınav Salonu Atama Sonuçları-2.....	95
6.4. Art Arda Görev Yazılma Durumu.....	96
6.5. Gözetmenlik Atama Kontrol Tablosu - Talep.....	97
6.6. Gözetmenlik Atama Kontrol Tablosu - Arz.....	97
6.7. Mazerete Görev Yazılma Durumu Kontrol Tablosu.....	98

## ŞEKİLLER DİZİNİ

<u>ŞEKİL</u>	<u>Sayfa</u>
3.1. Sınav Planlama İş Akış Şeması.....	9
3.2. Sınav Çizelgeleme İş Akışı .....	10
4.1. Gevşetilmiş Çözüm .....	26
4.2. Tam Sayılı Çözüm .....	26
4.3. Tam Sayılı Programlama Grafik Çözümü.....	26
4.4. Ulaştırma Problemi Gösterimi .....	34
4.5. İşletmelerde Dağıtım Ağı .....	36
4.6. Ulaştırma Modeli Şebeke ve Tablo Gösterimi.....	36
6.1. Veri Kullanımı Optimizasyonu .....	73
6.2. Giriş Ekranı .....	75
6.3. Program Ana Sayfası .....	76
6.4. Optimizasyon Geçiş Ekranı .....	76
6.5. Sınav Çizelgeleme Ana Ekranı .....	77
6.6. Sınav Tipi Seçme Ekranı.....	78
6.7. Sınav Açma Butonu .....	78
6.8. Sınav Takvimi Belirleme Formu.....	79
6.9. Öğrenci ve Öğretmenlerin Sınav Gün İstekleri Kayıt Tablosu.....	79
6.10. İlog Sorgularını Açma Ekranı .....	80
6.11. İlog'da Kullanılan Öğrenci Sınav Listesi.....	80
6.12. İlog'da Kullanılan Sınav Günü İstek Listesi.....	81
6.13. Sınav Çizelgeleme Modeli İlog Ekranı .....	82
6.14. Kısımlı Bazlı Sınav Yeri Atama Modülü Ana Ekranı.....	82
6.15. Sınav Yerleri .....	83
6.16. Kısımlı Bazında Sınav Listesi .....	84
6.17. İlog'da Kullanılacak Sınav Yeri Listesi.....	84
6.18. İlog'da Kullanılacak Kısımlı Bazında Sınav Listesi .....	85
6.19. Kısımlı Bazlı Sınav Yeri Atama İlog Modeli Ekran Görüntüsü .....	85
6.20. Öğrenci Bazlı Sınav Yeri Atama Modülü Ana Ekranı.....	86
6.21. Sınav Yeri Listesi.....	86

6.22. Ilog'da Kullanılacak Sınav Listesi .....	87
6.23. Gözetmen Atama.....	87
6.24. Personel Bilgileri Düzenleme Ekranı .....	88
6.25. Gözetmen Olacak Personel Listesinin Belirlenmesi .....	88
6.26. Gözetmen Mazeret Kayıt Tablosu .....	89
6.27. Personel Bazında Mazeret Toplamları .....	89
6.28. Gözetmen İhtiyaç Ekranı .....	89
6.29. Gözetmen Atama Tablosu.....	90
6.30. Gözetmen Atama Ilog Modeli Ekran Görüntüsü .....	90
6.31. Ilog Atama Sonrası Ara Yüzü .....	91
6.32. Sınav İstekleri Kontrol .....	91
6.33. Sınav Çizelgeleme Sonuç Tablosu.....	92
6.34. Sınav Günleri Öğrenci Mevcutları .....	92
6.35. Ilog Sınav Çizelgeleme İstatistikleri .....	93
6.36. Ilog Kısım Bazlı Sınav Yeri Atama İstatistikleri .....	95
6.37. Ilog Öğrenci Bazlı Sınav Yeri Atama İstatistikleri .....	96
6.38. Ilog Gözetmen Atama İstatistikleri .....	98

## KISALTMALAR DİZİNİ

KBK	Kuzeybatı Köşe
EKM	En Düşük Maliyetler
VAM	Vogel Approximation Method
TD	Temel Değişkenler
TOD	Temel Olmayan Değişkenler
TDP	Tam Sayılı Doğrusal Programlama
KO	Kümülatif Oturum
TEE	Term-End Exams
DS	Dal-Sınır
KDP	Karar Destek Programı
DP	Doğrusal Programlama
ISO	International Organization for Standardization

## 1. GİRİŞ

Günümüzde hizmet sektöründe yer alan organizasyonlar oldukça geniş bir alanda faaliyet sürdürmektedirler. Geçmiş zamana kıyasla, her geçen gün daha da artan bir uygulama alanına sahip olan bu organizasyonlar, hedeflerine ulaşabilmek için hizmet kapasitesini daha etkin ve verimli kullanmak durumundadırlar.

Hizmet sistemlerinde kapasite planlamasına yönelik olarak arz ve talep yönetim stratejileri geliştirilmiştir. Arz yönetim stratejileri, işgücü seviyesini değiştirme, çalışanları başka işler için eğitime, yarı zamanlı çalışan kullanma, otomasyon, hizmet sürelerini uzatma, müşteri katılımını artırma, çalışma vardiyalarını çizelgeleme, ayarlanabilir kapasite oluşturma, ekipman kiralama ve tesislerin genişletilmesi/yenilenmesi olarak tanımlanmaktadır. Talep yönetim stratejileri ise; tamamlayıcı hizmetleri geliştirme, fiyat özendirme sağlama, reklam, satış ve promosyon, rezervasyon/randevu sistemlerini geliştirme, iletişim çabaları ve karşı devirli (devresel) talep dokusuna sahip hizmetler sunma şeklinde ifade edilmektedir.

Organizasyonların daha etkin çalışması bakımından otomasyon ve çizelgeleme stratejilerinin öncelikli olduğu düşünülmektedir. Otomasyon, organizasyonlar için geliştirilecek modellerin çözümü, sonuçlarının analizi ve duyurulması aşamalarında ihtiyaç duyulan verilere dinamik ve güncel bir şekilde ulaşımı sağlayan önemli bir ihtiyaçtır. Çizelgeleme ise, otomasyonla birlikte daha verimli kullanılan ve çalışma yapılan karar noktalarında mevcut kısıtlar dâhilinde sürekli olarak en iyi sonuçların üretilmesine imkân sağlayan oldukça önemli bir diğer ihtiyaçtır.

Hizmet sistemlerinden biri olan eğitim kurumlarının yaygın hizmet ağı ve kapasitesi dikkate alındığında, arz yönetim stratejilerinin hayata geçirilmesinin yönetime etkinlik kazandırma bakımından önemli bir yeri vardır. Özellikle bünyesinde çok sayıda personel, ders, derslik ve öğrenci gibi kaynaklar bulunan yükseköğretim kurumlarında, her bir yarıyıldaki bu kaynakların yeniden kullanımına yönelik olarak tekrarlanan ara sınav, final, bütünleme, not yükseltme, tek ders ve mazeret sınavları için planlama süreçleri bulunmaktadır.

Her bir sınav döneminde sınavların ne zaman, nerelerde ve hangi gözetmenlerle yapılacağı belirlenmesi gerekmektedir. Bu çerçevede, sınavların ne zaman yapılacağı sorusu sınav çizelgeleme, öğrencilerin sınav salonlarına yerleştirilmesi sınav yeri atama ve belirlenen sınav salonlarına personel görevlendirilmesi ise gözetmen atama problemi olarak tanımlanmaktadır. Bu süreçte, akademik personelin katma değerli işlerde daha fazla mesai harcamasına olanak sağlayacak, öğrencilerin ihtiyaçlarını daha çok dikkate alan ve yönetimin daha verimli çalışmasına imkân tanıyacak şekilde planlama yeteneğine sahip olunması, eğitim kurumlarında arz yönetiminin başarısı açısından geçerli bir ölçüt olacaktır.

Bu çalışmanın amacı, eğitim kurumlarında sınav planlama süreçlerinde yaşanan sorunlara personel mazeretleri, öğrencilerin istekleri ve idarenin ihtiyaçlarını dikkate alan hızlı ve etkin çözüm yöntemleri geliştirmektir. Bu amaçla, hizmet sistemlerinde otomasyon ve çizelgeleme stratejileri dikkate alınarak sınav çizelgeleme, sınav yeri ve gözetmen atama problemleri için toplam dört adet doğrusal programlama modeli geliştirilmiştir. Sınav yeri atama problemi kısım ve öğrenci bazlı olarak iki alt başlıkta modellenmiştir. Geliştirilen modellerde kullanılan veri hacminin oldukça büyük olması nedeniyle, kullanıcıların veri girişi, analizi ve uygulama sonuçlarını rahat bir şekilde tamamlayabilmesi için bir veri tabanı ile ara yüz uygulaması da gerçekleştirilmiştir.

Çalışmanın ikinci bölümünde, sınav çizelgeleme, sınav yeri ve gözetmen atama konuları ile ilgili literatür araştırmasına yer verilmiştir.

Üçüncü bölümde, sınav çizelgeleme, sınav yeri ve gözetmen atama problemleri bütün boyutları ile üç ayrı başlıkta incelenerek kavramsal alt yapının oluşturulmasına çalışılmıştır.

Dördüncü bölümde, sınav çizelgeleme, sınav yeri ve gözetmen atama problemleri ile ilgili doğrusal programlama çözüm yöntemleri incelenmiştir. Bu bağlamda, karma tam sayılı problemlerin çözümünde kullanılan dal-sınır algoritması, 0-1 tam sayılı programlamada geliştirilen ekleme algoritması ve tümü tam sayılı programlamada faydalanılan kesme düzlemi algoritması açıklanmıştır. İlave olarak tam sayılı

programlamada karşılaşılan özel çözüm teknikleri incelenmiştir. Bir diğer başlıkta ise, tam sayılı programlama algoritmaları ile çözülebilen ancak hususi yapısı gereği daha pratik çözüm teknikleri bulunan ulaştırma ve atama problemleri özetlenmiştir. Yine bu bölümde, tek bir amacı ve sınırları kesin kısıtları bulunan klasik modelleme yerine daha esnek yapıdaki çok amaçlı ve sınırları değiştirilebilen hedef programlama konusu incelenmiştir. Ayrıca, özellikle çok aşamalı matematiksel modellemede kullanılan tablo indis kavramı açıklanmıştır. Bu bölümde son olarak, büyük boyutlarda verilerin tekrarsız ve bir sistematik dâhilinde organize edilmesini sağlayan veri tabanı konusuna değinilmiştir.

Beşinci bölümde sınav çizelgeleme, sınav yeri ve gözetmen atama problemleri için geliştirilen matematiksel modeller açıklanmaktadır. Bu çerçevede, sınav çizelgeleme modelinde öğretmenlerin mazeretleri ile öğrencilerin isteklerini dikkate alan ve bir öğrenciye aynı oturumda birden fazla sınav gelmeyecek şekilde sınavların tarih ve oturumları belirlenmektedir. Sınav yeri atama modelinde, oturumları belirlenen sınavlar için sınav yerlerindeki kapasite aşılmadan, bir arada yapılabilecek ve yapılamayacak sınavlar dikkate alınarak hangi kısım/şube/öğrencinin hangi sınavda nerede olacağı tespit edilmektedir. Üçüncü modelde ise sınav yapılacak yerlerdeki öğrenci sayısına bağlı olarak belirlenen gözetmen ihtiyacına göre; personelin bu süreçteki mazeretleri, art arda gelen oturumlarda sınav görevi yazılmaması, görev süresinin dengelenmesi, kıdem düzeyinin dikkate alınması ve personele her bir sınav döneminde ortalama kadar görev yazılması hususları göz önünde bulundurularak gözetmenlik ataması yapılmaktadır.

Altıncı bölümde, bir yükseköğrenim kurumunda uygulanan bu çalışmayla ilgili veri tabanı ara yüzü ve uygulama sonuçları incelenmekte olup son bölümde bulgular hakkında yorumlar, çalışmanın literatüre katkısı ve geleceğe yönelik çalışmalarla ilgili önerilere yer verilmiştir.



## 2. LİTERATÜR ARAŞTIRMASI

Elabdalla ve Husein (1991) çalışmalarında, sınav çizelgelemek için; hiçbir öğrencinin aynı anda iki sınavının olmaması, hiçbir öğrencinin bir günde ikiden fazla sınav almaması, sınav gün sayısının az olması, bir gün içinde iki sınava sahip toplam öğrenci sayısının en az olması ve bir gün içinde iki ardışık sınava sahip öğrenci sayısı az olması şeklindeki hedefleri karşılayacak bir prosedür olması gerektiğini ifade etmiştir.

Balacrishnan (1991) makalesinde, sınav planlaması birincil amaç olarak hiçbir öğrencinin aynı dönemde birden fazla sınava girmemesi gerektiğini ancak; gerçek hayat şartlarında diğer karmaşık kısıtlar nedeniyle bu problemin karmaşık hale dönüştüğünü belirtmiştir.

Lotfi ve Cerveney (1991) makalelerinde, çok oturumlu final sınavlarının çizelgelenmesi için üç yıllık sürede geliştirilen bir uygulama üzerinde çalışmışlardır. İlk aşamada bir öğrencinin eş zamanlı sınavı en az olacak şekilde final sınavlarında bloklar oluşturmuşlardır. Bir günde iki ya da daha fazla sınavları olan öğrencilerin sayısını en aza indirerek sınav bloklarını sınav günlerine atamışlardır. Zaman içindeki sınav günleri ve sınav bloklarında ardışık sınavı olan öğrencilerin sayısını en aza indirmişler ve son olarak alan kullanımını maksimize edecek şekilde sınavları dersliklere atamışlardır. Yukarıda belirtilen dört aşamada yeni formül ve çözüm yöntemleri geliştirmişler ve büyük çaplı bir üniversitede uygulamışlardır.

Carter, Laporte ve Chinneck (1994) makalelerinde, sınav çizelgelemenin sınırlı bir zamanda yapılması gereken ve hemen hemen her lise, kolej ve üniversitede bir yaşanan sorun olduğunu belirtmiştir. 10 yılı aşkın bir süre içerisinde bu konu üzerinde çalışmak suretiyle bilgisayar üzerinde çalışan kullanıcı dostu, esnek ve güçlü bir sınav çizelgeleme sistemi geliştirmişlerdir. Geliştirdikleri programda bir öğrencinin sınav, oda veya zaman durumu ile diğer faktörlerle ilgili gereksinimlerine ilişkin çözüm üretmişlerdir. İki yıl boyunca, Toronto Üniversitesi Mühendislik fakültesi ve Carleton Üniversitesi ile işbirliği yapılarak yazılım paketi geliştirilmiş ve her iki enstitüde kullanmışlardır.

Yıldız (2004) yüksek lisans tezinde, Dumlupınar Üniversitesi Endüstri Mühendisliği Bölümü sınavları için görevli atamasını, 0-1 tam sayılı bulanık hedef programlama yaklaşımı ile model kurmuştur. Bulanık modeli Bellman ve Zadeh'in Max-Min yaklaşımı ve Tiwari, Dharmar ve Rao'nun toplamsal model yaklaşımı ile çözmüş ve işlem sayısı, çözüm süresi ve tatmin düzeyi bakımından Bellman ve Zadeh'in Max-Min yaklaşımının, atama probleminin çözümünde kullanılabilir en uygun metot olduğu sonucunu ifade etmiştir.

Burke, Bykov ve Petrovic (2001) bildirimlerinde, üniversite sınav çizelgeleme problemlerini çok ölçütlü karar problemi olarak incelemiştir. Bu tür problemin çözümüne yeni bir, çok ölçütlü yaklaşım sunmuştur. Kriterlerin sayısını çizelgeleme kısıtlarının (her enstitü için değişebilecek olan uygun derslik kapasiteleri, öğrenciler için sınavların yakınlık durumu vb.) sayısı olarak tanımlamışlardır. İki aşamalı yaklaşımda, ilk olarak her bir kriter için en iyi çizelgeleme sonuçlarına ulaşmışlardır. İkinci aşamada ise tüm kriterler için eş zamanlı olarak istenen sonuçlara ulaşmayı başarmışlardır. Bu yaklaşımda kriter uzayında tüm kriterleri tek seferde optimize eden ideal nokta olduğunu düşünmüşlerdir. Gerçekte genellikle bu ideal nokta tam olarak bulunamayacağından dolayı sezgisel arama yaparak bu ideal noktaya en yakın nokta için çözüm aramışlardır. Bununla birlikte geliştirdikleri yöntemle çeşitli üniversitelerden gelen gerçek verileri kullanarak test etmişlerdir.

Özcan ve Ersoy (2005) bildirimlerinde, çizelgeleme problemine memetik algoritma ile çözüm geliştirmişlerdir. Sınav çizelgeleme problemlerini çözmek için bir adaptif ihlal imkanı olan hiyerarşik tepe tırmanma yöntemini kullanmışlardır.

Dammak, Elloumi ve Kamoun (2006) makalelerinde, belli kapasiteleri olan dersliklere bir dizi belirli bir boyuta sahip bağımsız (sıfır-çelişkili) sınavları atama sorununu, mümkün olduğunca her derslikte bir sınavdan daha fazla sınav olmayacak şekilde 0-1 tam sayılı doğrusal programlama yöntemi ve aynı zamanda ulaştırma problemi olarak modellemişlerdir. Grafik renklendirme yönetimine dayalı çeşitli sezgisel tarama kullanan Sfax Ekonomi ve Yönetim Bilimleri Fakültesi'ndeki sınav atama sorununa uygulamışlardır.

Çoruhlu (2007) yüksek lisans tezinde, yaklaşık dört bin öğrencinin eğitim öğretimine devam ettiği bir yükseköğretim kurumundaki sınav çizelgeleme problemini incelemiş ve sınav sürecinde, farklı kapasitelere ve fiziksel koşullara sahip sınav salonlarına sınav görevlilerinin atanmasını sağlayan tam sayılı doğrusal programlama modeli geliştirmiştir. Modele veri girişinin kolaylaştırılması amacıyla basit ara yüzler hazırlamıştır. Modeli öncelikle kısıt tatmin problemi olarak ele almış ve sonrasında modeli hedef programlama problemi olarak ele almış ve kabul edilebilir süreler içinde uygun çözümlere ulaşmıştır.

Kalaycı'nın (2008) yüksek lisans tezinde, sınav çizelgelemenin öğrencilerin başarısını açığa çıkaracak şekilde düzenlenmesi gerektiği fikri üzerine odaklanılmıştır. Çalışmasında sınavların çizelgelenmesi için, zorunlu kısıtların sağlanması ön koşulu ile zor olan sınavların birbirinden daha uzak zaman dilimlerine atanmasıyla en uygun sınav dönemi yayılımını sağlamaya yönelik iki farklı genetik algoritma modeli geliştirmiştir.

Palamutçuoğlu (2008) yüksek lisans tezinde, Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakültesi için ders ve sınav programlarını çözen bir uygulama geliştirmiştir. Öncelikle, matematiksel model kullanarak melez bir genetik algoritma geliştirmiş ve 2007-2008 bahar yarıyılı için modeli uygulamıştır.

Acar (2010) yüksek lisans tezinde, öğretim üyelerinin ve öğrencilerin istekleri göz önünde bulundurarak sınav çizelgeleme problemini çözmeye çalışmıştır. Problemin çözümü için yeni bir matematiksel model oluşturmuştur. Ancak; bu matematiksel modelle büyük verilere sahip problemleri kısa zamanda çözemediği için sınıflandırma yapan iki yeni sezgisel yöntem geliştirmiştir. Araştırmasında eniyileme alanındaki en popüler yazılımlardan biri olan Xpress-MP adlı yazılım kullanmış ve geliştirdiği ikinci sezgisel yöntemi Fatih Üniversitesi verilerine uygulamıştır.

Wang, Bussieck, Guignard, Meeraus ve O'Brien (2010) makalelerinde, West Point Askeri Akademisi'ndeki dönem sonunda yüzlerce sınavların kısa bir süre içerisinde çizelgelenmesi yapıldığından diğer çizelgeleme problemlerine benzemediğini belirtmişlerdir. Buradaki zorluğun bazı sınavlar için birden fazla oturum oluşturulması

ve böylece her bir öğrencinin tüm sınavlara girmesinin sağlandığını bu çerçevede amaç fonksiyonunun bu tip çift sınavları en aza indirmek olduğunu belirtmişlerdir.

Bunula birlikte, 2001 yılından beri başarılı bir şekilde uyguladıkları ve GAMS Development şirketi işbirliği ile birlikte geliştirilen, çok sayıda modülü bulunan ve yeni teknikler de barındıran matematiksel optimizasyon sisteminin esnek bir yapıya sahip bulunduğunu ve bu sayede -hiyerarşik seviyesine göre- eğitmenlerden gelen istekler gibi beklenmedik kısıtların kolaylıkla eklenebildiğini ve öğrenci bilgi sistemi ile entegre olduğundan oldukça etkili bir uygulama geliştirdiklerini ifade etmişlerdir.

İlkuçar (2011) bildirisinde, özellikle meslek yüksekokullarında sınav salonlarında kimlerin, nerelerde, hangi oturumda, kaç görev tutacağı, gözetmenlik görevlerinin ücretle orantılı olması, sınav uygulamasıyla ilgili olarak öğretim elemanlarının isteklerinin göz önüne alarak sınav görevleri yerleşimi yapan bir yazılımla ilgili çalışma yapmıştır. Çalışması Mehmet Akif Ersoy Üniversitesi Meslek Yüksek Okulunda (MAKÜ-MYO) 2009 güz ve bahar 2010 güz dönemlerindeki vize ve final sınavlarının dağılımında kullanılmıştır.

Altıntaş (2011) yüksek lisans tezinde, çizelgeleme problemlerinin tanımlanmasını araştırmış ve bu yöntemlerle sınav çizelgeleme probleminin nasıl çözülebileceğini incelemiştir. Sınav programı çizelgeleme problemlerinin NP zor diye adlandırılan optimizasyon problemleri olduğunu ifade etmiş ve bu tip problemlerin her zaman tam çözümün bulunamadığını ve sadece yaklaşık sonuçların elde edilebildiğini ancak bunun yeterli olabildiğini, problemlerin çözümünde; matematiksel yöntemler, sayısal analiz ve sezgisel yöntemler kullanıldığını belirtmiş ve çalışmasında sezgisel yöntemlerden, Genetik Algoritma, Karınca Kolonisi, Tabu Arama, Benzetimli Tavlama (Simulated Annealing) algoritmalarını ele almıştır.

Gürel (2012) yüksek lisans tezinde, üniversitelerin sınav çizelgeleme problemine bir çözüm üretmek amacıyla, NP (Non-polinomal, polinomal olmayan) zor diye adlandırılan ve her zaman tam çözümü bulunamayan sınav çizelgeleme problemlerini incelemiştir. Algoritmaların ürettiği sonuçların elle kolayca değiştirilebilen görsel ara

yüz ile birlikte yaklaşık çözüm üreten üç farklı algoritma geliştirerek sonuçlarını karşılaştırmıştır.

Seyyide (2014) yüksek lisans tezinde, sınav çizelgeleme problemlerinin NP-complete karmaşıklık sınıfına ait kombinatoriyel optimizasyon problemleri olarak tanımlandığını ifade etmiş, konuyla ilgili olarak son 20 yılda bu zor problemin çözümü için yüksek seviyeli arama stratejisi kullanan pek çok meta-sezgisel yöntem önerildiğini ancak bu yöntemlerin çoğunun etkin state-of-art (en gelişkin) yaklaşımlar olması ile birlikte oldukça kompleks ve karmaşık yöntemler olduğunu belirtmiş ve uygulanabilirlik açısından basit ve yalın bir yöntem olan Öğretme-Öğrenme tabanlı optimizasyon (ÖÖTO) algoritmasını sınav çizelgeleme problemleri için araştırmıştır. Çalışma sonuçlarına göre ÖÖTO algoritmasının sınav çizelgeleme problemleri için basit ve yalın bir yöntem olma avantajı ile birlikte uygulanabilir yeni bir alternatif yöntem olduğunu söylemiştir.

Matcı (2014) yüksek lisans tezinde, Anadolu Üniversitesi bünyesinde yılda dört kez yapılan açık öğretim sınavlarına gözetmen atama problemini atama problemlerini örnek alarak bir matematiksel model ile çözmüştür. Aynı zamanda otomasyonun geliştirilmesi amacıyla bir algoritma oluşturmuş ve bu algoritmaya uygun bir otomasyonu C# dilinde yazmıştır.

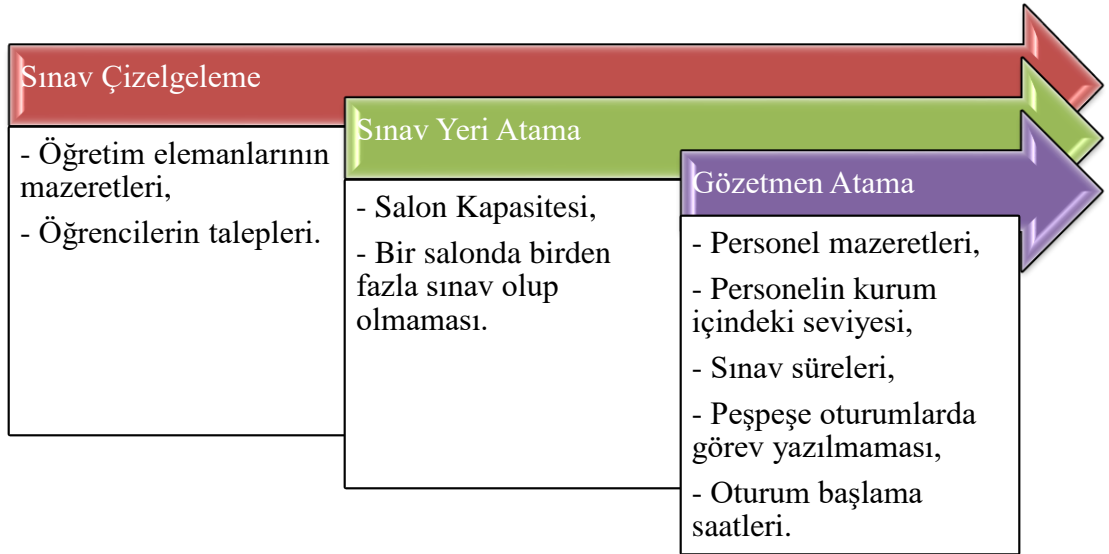
Yapılan çalışmalar incelendiğinde; Yıldız (2004) için görevli ataması için 0-1 tam sayılı bulanık hedef programlama yaklaşımı, Dammak, Elloumi ve Kamoun (2006) ile Çoruhlu (2007) sınav atama sorununu, mümkün olduğunca her derslikte bir sınavdan daha fazla sınav olmayacak şekilde 0-1 tam sayılı doğrusal programlama yönetimi ve aynı zamanda ulaştırma problemi olarak modellemişlerdir.

Kalaycı (2008) çalışmalarında sınav çizelgelemenin probleminin çözümü için genetik algoritma, Palamutçuoğlu (2008) matematiksel model kullanarak melez bir genetik algoritma geliştirmiş Altıntaş (2011) Genetik Algoritma, Karınca Kolonisi, Tabu Arama, Benzetimli Tavlama algoritmalarını ele almıştır.

### 3. SINAV ÇİZELGELEME, SINAV YERİ VE GÖZETMEN ATAMA PROBLEMLERİ

Eğitimde amaç, öğrencinin istenilen alışkanlıkları kazanmasıdır. Bu alışkanlıkların kazandırılması ve eğitim kalitesinin yükseltilmesi öğretim elemanı, motivasyon, çevre şartları, eğitimde zamanlama gibi birçok değişkene bağlıdır. Doğal olarak bu konularda çok çeşitli araştırmalar yapılmaktadır (Taban ve Özdemir, 2004).

Eğitim kurumları için zaman çizelgeleme problemleri literatürde, çoğunlukla üç temel grup altında incelenmektedir. Bunlar ders çizelgeleme, sınav çizelgeleme, sınav yeri ve gözetmen atama problemleridir. Bu çalışmada Şekil 3.1.de gösterilen sınav çizelgeleme ve sonrası ile ilgili planlama süreci incelenmiştir.



Şekil 3.1. Sınav Planlama İş Akış Şeması

Sınav çizelgeleme, sınav yeri ve gözetmen atama problemleri incelendiğinde yapılarının daha ziyade karma tam sayılı problem tipinde olduğu görülmüştür. Taha (2007) kitabında, tam sayılı programlamada hesaplamaları etkileyen en önemli

faktörün tam sayılı deęişken sayısı olduęunu, mevcut algoritmalar ile TDP problemlerinin istikrarlı bir biçimde çözülemediğinden TDP modelindeki tam sayılı deęişken sayısını mümkün olduęunca azaltmanın hesaplama açısından avantajlı olduęunu belirtmiştir.

Bu bağlamda, tam sayıya yakın deęişkenlerin mümkün olduęunca sürekli deęişken haline getirilmesi, tam sayı uygunluk aralığının mümkün olduęunca kısıtlanmasını ve modelde doğrusal olmayan durumlardan kaçınılmasının faydalı olacağını ifade etmiştir. Bu çalışmada geliştirilen modellerde bu hususlar dikkate alınmıştır.

### 3.1. Sınav Çizelgeleme Problemi

Sınav çizelgeleme problemi, öğretmenlerin mazeretleri ile öğrencilerin isteklerinin dikkate alınarak bir öğrenciye aynı oturumda birden fazla sınav gelmeyecek şekilde ve idare tarafından önceden belirlenen sınav takvimi içerisinde sınavların istenen gün ve oturumlara atanmasıdır. Şekil 3.2.de sınav çizelgeleme faaliyet sıralaması belirtilmiştir.



Şekil 3.2. Sınav Çizelgeleme İş Akışı

### 3.1.1. Sınav Takvimi ve Sınav Gün İsteklerinin Oluşturulması

Öncelikle idare tarafından eğitim öğretim yılı faaliyet takviminde belirlenen sınav tarihlerinin Çizelge 3.1.de olduğu gibi sırasıyla 1 'den g'ye kadar olan günlere atanması gerekmektedir.

**Çizelge 3.1.** Sınav Takvimi

Eğt.Öğt.Yılı	Dönem	Sınav Tipi	Tarih	Gün
2014-2015	1	Ara Sınav	26.1.2015	1
			27.1.2015	2
			28.1.2015	3
			.	.
			.	.
			6.2.2015	g

Belirtilen tarih aralığında yapılacak sınavların günlerinin öğretim elemanlarının mazeretleri ile öğrencilerin istekleri doğrultusunda belirlenebilmesi için Çizelge 3.2.de gösterildiği şekilde her bir sınav için sınav takvimini kapsayacak şekilde bu taleplerin öğretmen ve öğrencilerden alınması gerekmektedir.

Öğretmen ve öğrencilerin taleplerinin 0 ile 3 arasında bir ölçekle (0:Çok istiyorum, 1:İstiyorum, 2:Fark etmez, 3:Asla olmasın) bildirilmesi gerekmektedir. Konuyla ilgili olarak öğretmenlerin 3 olarak belirttikleri günlerde öğrenciler o gün için 0 değerinde istek bildirseler dahi sonuçta öğretmenlerin mazereti bağlayıcı olduğu için öğrenci istekleri dikkate alınmamaktadır.

Sonuç olarak; öğrenci ve öğretmenlerden alınan istekler bu açıdan değerlendirilerek tek bir değere dönüştürülmektedir. Bu sayede Çizelge 3.3.te gösterilen sınav gün istek tablosu oluşmaktadır.



**Çizelge 3.2.** Öğrenci ve Öğretmen için Sınav Günü İstek Tablosu

Sınav	Gün	İstekler	
		Öğretmen	Öğrenci
1	1	3	3
1	2	0	0
1	.	.	.
1	g	3	0
2	1	3	3
2	2	0	0
2	.	.	.
2	g	3	0
s	1	3	3
s	2	0	0
.	.	.	.
s	g	3	0

**Çizelge 3.3.** Sınav Günü İstek Tablosu

Sınav	Gün	Sonuç
1	1	9
1	2	0
1	.	.
1	g	9999
.	.	.
s	1	9
s	2	0
.	.	.
s	g	9999

Sınav çizelgelemede esas olan, bir öğrenci için aynı anda birden fazla sınav gelmeyecek şekilde planlama yapılmasıdır. Ancak, her ne kadar idare tarafından genel sınav takvimi belirlense de bazı durumlarda bir öğrencinin gireceği sınav sayısına bağlı olarak takvimde belirtilenden daha fazla gün gerekebilecektir.

Örneğin; sınav takvimi 10 gün olarak belirlenmiş olsun ve aynı süreçte bir öğrencinin de alttan ya da üstten aldığı bir ders sebebiyle 11 sınava girmesi söz konusu olursa, sınav çizelgelemesi yapılacak takvim otomatik olarak 11 güne uzayacaktır. Fazla olan ders sayısı kadar gün, çizelgeleme sırasında kullanılmak üzere modele ilave sanal günler olarak eklenmektedir. Sonradan eklenen sanal günler ise, mevcut sınav takvimine içerisinde bulunan günlerde ilave bir oturum olarak değerlendirilecektir.

### **3.1.2. Sınav Oturumlarının Belirlenmesi**

Açılacak ilave oturum sayısı ile ilgili, sınavların hangi tip salonlarda yapılacağına bağlı olarak ihtiyaç doğabilecektir. Örneğin; bir okulda uygulanan sınavlar dersliklerin haricinde amfi gibi büyük salonlarda uygulanması durumunda ve bu salonlar için kapasite kısıtı olması halinde, yine aynı günde birden fazla oturumda sınav uygulaması durumu ortaya çıkmaktadır. Bu çalışmada, derslik haricinde kullanılacak amfilerin aynı anda en fazla bir sınıftaki toplam öğrenci sayısı kadar kapasitesi olduğu varsayılmaktadır.

Sınav günleri belirlendikten sonra dört sınıfta da öğrencisi olan bir eğitim kurumu için öğrencilerin sınıfına bağlı olarak her gün için günde dört oturum olacak şekilde toplam sınav yapılması söz konusu olacaktır.

İdarenin yetkisinde olan bu hususla ilgili olarak, sınavlar amfilerde yapılmak suretiyle her gün birden fazla oturum açılması söz konusu olabileceği gibi sınavların normal dersliklerde yapılması neticesinde mevcut sınav takvimi içerisinde ve genellikle her gün bir oturum şeklinde de planlama yapılabilmektedir.

Sınav çizelgeleme sonrasında gerek sanal günlere atanan sınavlar, gerekse sınav yeri kapasitesine bağlı olarak oluşacak ilave oturum sayısının bu aşamada kesinleştirilmesi gerekmektedir.

Bu çerçevede; açılan oturumlara bağlı olarak birikimli (kümülatif) oturumlar Çizelge 3.4.teki gibi 1'den t'ye kadar numaralandırılmalıdır. Bu aşamadan sonra oturum olarak genellikle bu birikimli oturum verisi kullanılacaktır.

**Çizelge 3.4. Kümülatif Oturum Tablosu**

Gün	Oturum	Kümülatif Oturum (KO)
1	1	1
1	2	2
1	.	.
1	0	.
2	1	.
2	2	.
2	.	.
2	0	.
.	.	.
g	1	.
g	2	.
g	.	.
g	0	t

Sonuç olarak; sınav çizelgeleme modeli neticesinde elde edilecek sınav gün tablosunun Çizelge 3.5.teki gibi sınav birikimli oturum tablosu olarak oluşturulması gerekmektedir.

**Çizelge 3.5.** Sınav Kümülatif Oturum Tablosu

Sınav	KO
1	1
2	2
.	.
S	t

### 3.2. Sınav Yeri Atama Problemi

Sınav yeri atama, oturumları ilk modelde belirlenen sınavlar için her bir oturumda sınav salonu/dersliklerdeki kapasite aşılmadan, bir arada yapılabilecek ve yapılamayacak sınavlar dikkate alınarak hangi kısım/şube/öğrencinin hangi sınavda nerede sınava gireceği tespit edilmesidir. Çizelge 3.6.da kapasite durumuna göre kategorize edilen salonlara, salonda yapılacak sınav sayısına bağlı olarak genellikle dört farklı tarzda atama stratejisi uygulanabilecektir.

**Çizelge 3.6.** Sınav Uygulama Stratejileri

Kapasite	Sınav Yeri Türü	Salonda Yapılacak Sınav Sayısı	
		Bir Çeşit Sınav	Birden Fazla Sınav
~30	Normal Derslik	H	E
>>30	Büyük Salon	E	E

Bu bağlamda, birinci alternatif öğrencilerin sınava girebilecekleri büyük salonlarda yalnızca bir dersin sınavının uygulanması ya da birden çok dersin sınavının yapılmasıdır. İkinci seçenek ise normal dersliklerde yalnızca bir dersin sınavının uygulanması ya da birden çok dersin sınavının yapılmasıdır. Burada yapılacak tercih sınav salonunda kopya çekilmesine imkân vermeyecek alternatif yönünde olmalıdır.

Sınavların nasıl ve nerede uygulanacağı kararı verildikten sonra sınavların yapılacağı salonlar Çizelge 3.7.de belirtilen şekilde listelenmelidir. Sınav yerlerine yapılacak atamada öğrenci sayısının salonların kapasitelerini geçmeyecek şekilde uygulanması gerekmektedir.

**Çizelge 3.7.** Sınav Yerleri Tablosu

Sınav Yeri Nu	Sınav Yeri Adı	Sınav Kapasitesi	Kullanım Maliyeti	Grup Nu.
1	S_1	120	50	1
2	S_2	150	50	1
.	.	.	.	.
.	.	.	.	.
y	S_y	200	0	2

### 3.2.1. Kısım Bazlı Atama Problemi

Kısım tabiri, öğrencilerin her bir dersi alırken bir araya geldikleri grup olarak kullanılmaktadır. Kısım bazlı sınav yeri atama; bir önceki sınav çizelgeleme modelinde oturumları belirlenen sınavlardaki öğrencilerin kısım bütünlüğü bozulmaksızın ve kapasite aşılmadan salonlara yapılan atamadır. Problemin çözülebilmesi için her bir oturumdaki sınavlar ve bu sınavlara girecek olan kısımların Çizelge 3.8.de olduğu şekilde listelenmesi gerekmektedir.

Kısım bazlı sınav yeri atamanın uygulama bakımından avantaj ve dezavantajları bulunmaktadır. Avantajları, sınav uygulama kolaylığı, çizelgeleme sonuçlarının kolaylıkla öğrencilere duyurulabilmesi ve daha az satır için atama yapılmasıdır. Dezavantajları ise birden fazla kısmın aynı anda sınava girebileceği uygun büyüklükte

salonlara ihtiyaç olması, bir salonda çok sayıda gözetmen ihtiyacı bulunması ve bu büyüklükteki salonlar için gözetmen kontrolünün zorluğu hususlarıdır.

**Çizelge 3.8.** Sınav Kısım Tablosu

KO	Sınıf	Sınav	Sınav Süresi	Kısım	Öğrenci Mevcudu
1		1	60	5	27
1		2	60	4	28
1		.	.	.	.
1		S	60	13	29
2		1	60	14	27
2		2	90	15	25
2		.	.	.	.
2		S	90	17	25
t		1	60	18	25
t		2	60	19	25
t		.	.	.	.
t		S	60	19	25

### 3.2.2. Öğrenci Bazlı Atama Problemi

Öğrenci bazlı sınav yeri atama problemi; kısım bazlı atamada olduğu gibi oturumları daha önceki modelde belirlenen sınavlardaki öğrencilerin, sınav yerlerinin kapasitesi aşılmadan ve bir salonda aynı anda çok sayıda sınav olacak şekilde sınav salonlarına

atanmasıdır. Sınav yerleşiminde kopya çekmenin önüne geçilebilmesi bakımından bir tedbir olarak, aynı sınava giren öğrencilerin yan yana, arka arkaya ve çaprazda olmayacak şekilde atanması önemlidir. Bu durumun sağlanabilmesi için Çizelge 3.9.daki gibi her bir salonda öğrencilerin oturma düzeni önceden teorik olarak belirlenmelidir.

**Çizelge 3.9.** Sınav Yeri Teorik Sınav Şablonu

Sıra/Sütun	1	2	3	4	5	6
1	1	2	3	4	1	2
2	3	4	1	2	3	4
3	1	2	3	4	1	2
4	3	4	1	2	3	4
5	1	2	3	4	1	2

En az dört sınava göre tasarlanan bu örnekte her bir sınav yeri için uygulanacak teorik sınavların öğrenci kapasitesi Çizelge 3.10.da gösterildiği şekilde belirlenmelidir. Yapılacak atamalarda teorik sınav kapasitesinin aşılmaması hususu gözetilmelidir. Problemin model ile çözülebilmesi için her bir oturumdaki sınavlar ve bu sınavlara girecek olan öğrencilerin Çizelge 3.11.de olduğu şekilde listelenmesi gerekmektedir.

Öğrenci bazlı atamanın avantajları olarak, sınavların mevcut derslikler kullanılarak bu dersliklerde karışık sınav uygulamak suretiyle yapılabilmesi, mevcut derslikler haricinde ilave sınav yerlerine -büyük oranda- ihtiyaç olmaması ve gözetmen kontrolünün daha küçük bir gruba nezaret etmesi bakımından kolay olmasıdır. Dezavantajları ise sınav evraklarının dağıtım ve toplanması bakımından sınav uygulama zorluğu bulunmasıdır.

**Çizelge 3.10.** Sınav Yeri Mevcuduna Göre Teorik Sınav Mevcutları

Sınav Yeri Toplam Mevcudu	Teorik Ders Öğrenci Mevcudu				En Az Öğrenci Mevcudu	En Çok Öğrenci Mevcudu
	1	2	3	4		
14	4	4	3	3	3	4
15	4	4	4	3	3	4
17	5	4	4	4	4	5
18	5	5	4	4	4	5
20	5	5	5	5	5	5
21	6	5	5	5	5	6
22	6	6	5	5	5	6
23	6	6	6	5	5	6
24	6	6	6	6	6	6
25	7	6	6	6	6	7
26	7	7	6	6	6	7
27	7	7	7	6	6	7
28	7	7	7	7	7	7
29	8	7	7	7	7	8
30	8	8	7	7	7	8



**Çizelge 3.11.** Sınav Öğrenci Tablosu

KO	Sınıf	Sınav	Sınav Süresi	Sınav Grup Nu	Öğrenci Mevcudu
1		1	60	1	29
1		2	60	1	55
.		.	.	.	.
1		S	60	3	109
2		1	60		105
2		2	90		55
.		.	.		.
2		S	90		190
t		1	60		.
t		2	60		.
t		.	.		.
t		S	60		.

### 3.3. Gözetmen Atama Problemi

Gözetmen; sınavın kurallara uygun bir biçimde yapılmasını sağlayan kimse, gözcü, gözetmenlik ise gözetmenin yaptığı iş olarak tanımlanmaktadır.<sup>1</sup> Bu çerçevede gözetmen atama problemi, sınav gözetmenlik görevi yapacak personelin kısım ya da öğrenci bazlı sınav yeri atama modeli sonucunda sınav yerleri belirlenen sınav yerlerine görevlendirilmesidir. Gözetmenlik atama modeli için;

---

<sup>1</sup> TDK: Türk Dil Kurumu Genel Türkçe Sözlük.

1. Personelin sınav oturumlarındaki mazeretleri Çizelge 3.12.deki şekilde alınarak mazeret tablosu oluşturulur.

**Çizelge 3.12.** Personel Gözetmenlik Mazeret Tablosu

Personel	KO	Mazeret Tipi
1	1	
1	2	
1	.	
1	t	
.	.	
.	.	
m	1	
m	2	
m	.	
m	t	

2. Oluşturulan çizelge incelenerek, Çizelge 3.13. ile gösterildiği gibi personelin mazeretleri toplamı hesaplanır.

**Çizelge 3.13.** Gözetmenlik Yazılabilecek Personel Listesi

Personel	Toplam Mazeretli Gün Sayısı
1	
2	
.	
.	
m	

3. Hesaplanan mazeret sayısı toplam oturum sayısından küçük olan personel belirlenerek gözetmenlik yazılacak personel listesi kesinleştirilir.

4. Görev yazılacak personel, sınav oturumları çizelgesinde mazeret verisi eşleştirilmiş olarak Çizelge 3.14.de belirtilen şekilde tüm sınav oturumları için çoğaltılarak, modelde kullanıma hazır hale getirilir.

**Çizelge 3.14.** Gözetmenlik Atama Tablosu

KO	Gün	Oturum	Personel	Unvan	Bölüm	Mazeret Tip No
1	1	1	1			1
2	1	2	1			6
.	.	.	1			3
t	g	o	1			0
1	1	1	2			0
2	1	2	2			0
.	.	.	2			1
t	g	o	2			1
1	1	1	.			5
2	1	2	.			5
1	1	1	m			2
2	1	2	m			2
.	.	.	m			0
t	g	o	m			0

5. Sınav oturumlarında bulunan her bir sınav salonundaki sınava girecek öğrenci sayısı, salondaki maksimum sınav süresi ile salonda görev yapması planlanan gözetmen sayısı bilgilerini içeren Çizelge 3.15.teki verilerin oluşturulması gerekmektedir.

**Çizelge 3.15.** Gözetmenlik Talep Tablosu

KO	Gün	Oturum	Sınav Yeri	Maksimum Süre	Öğrenci Mevcudu	Gözetmen Sayısı
1	1	1	1			
2	1	2	1			
.	.	.	1			
t	g	o	1			
1	1	1	2			
2	1	2	2			
.	.	.	2			
t	g	o	2			
1	1	1	.			
2	1	2	.			
.	.	.	.			
t	g	o	.			
1	1	1	j			
2	1	2	j			
.	.	.	j			
t	g	o	j			

Çizelge 3.14.te bulunan personel mazeretleri ve Çizelge 3.15.teki gözetmen ihtiyacı dikkate alınarak; mümkün olduğunca personelin mazeretlerine görev yazılmamalıdır. Bir oturumun bitişi ile sonraki oturumun başlama saatleri arasında gözetmenin sınav salonuna gitmek için yeterli süre olmadığı varsayımı ile art arda gelen oturumlarda sınav görevi yazılmamalıdır. Gün içinde icra edilecek oturum sayısına bağlı olarak; her oturumda mümkün olduğunca her bir oturumdaki ortalama görev sayısı kadar görev yazılmalıdır (örneğin; 1.oturumda ortalama görev sayısı 2 ise herkese 2 adet ilk oturum görevi yazılması). Her bir oturumda her sınav salonundaki maksimum sınav süresi dikkate alınarak personele toplamda yapacağı görev süresi dengelenmelidir. Personelin kıdem düzeyi dikkate alınmak suretiyle görev dağılımı yapılarak gözetmen atama modeli çözümü sonrasında Çizelge 3.16.da gösterilen sonuçlar elde edilmektedir.

**Çizelge 3.16.** Gözetmenlik Atama Sonuçları Tablosu

Personel	KO	Sınav Yeri	Sonuç
1	1	1	1
1	2	2	1
1	.	.	1
1	t	j	0
2	1	1	0
2	2	2	0
2	.	.	1
2	t	j	1
.	.	1	1
.	.	.	1
.	.	j	1
m	1	1	0
m	2	2	1
m	.	.	0
m	t	j	0

#### 4. SINAV ÇİZELGELEME, SINAV YERİ VE GÖZETMEN ATAMA PROBLEMLERİNDE KULLANILAN ÇÖZÜM YÖNTEMLERİ

Bu bölümde; sınav çizelgeleme, sınav yeri ve gözetmen atama problemleri ile ilgili doğrusal programlama çözüm yöntemleri incelenmektedir. Bu bağlamda, karma tam sayılı problemlerin çözümünde kullanılan dal-sınır algoritması, 0-1 tam sayılı programlamada geliştirilen ekleme algoritması ve tümü tam sayılı programlamada faydalanılan kesme düzlemi algoritması açıklanmaktadır. İlave olarak tam sayılı programlamada karşılaşılan özel çözüm teknikleri incelenmiştir. Bir diğer başlıkta ise, tam sayılı programlama algoritmaları ile çözülebilen ancak hususi yapısı gereği daha pratik çözüm teknikleri bulunan ulaştırma ve atama problemleri özetlenmektedir. Yine bu bölümde, tek bir amacı ve sınırları kesin kısıtları bulunan klasik modelleme yerine daha esnek yapıdaki çok amaçlı ve sınırları değiştirilebilen hedef programlama konusu açıklanmaktadır. Ayrıca, özellikle çok aşamalı matematiksel modellemede kullanılan tablo indis kavramı incelenmiştir. Son olarak, büyük boyutlarda verilerin tekrarsız ve bir sistematik dâhilinde organize edilmesini sağlayan veri tabanı konusuna değinilmiştir.

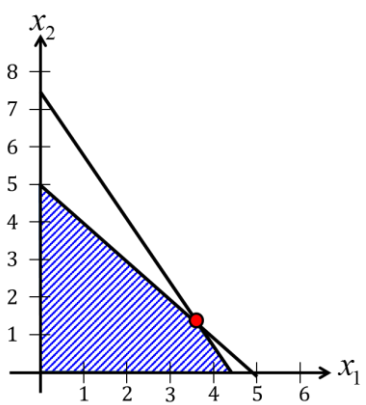
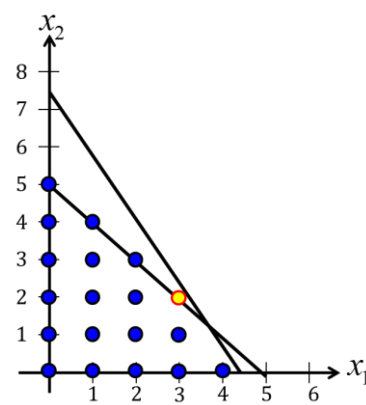
Atama ve çizelgeleme problemlerinin çözümünde, yöntem olarak genellikle ulaştırma, dağıtım ve atama problemleri ile hedef programlama ve tam sayılı programlama modelleri kullanılmaktadır. Konuyla ilgili literatür incelendiğinde, 0-1 tam sayılı bulanık hedef programlama, çok ölçütlü problem yaklaşımı, çeşitli sezgisel yaklaşımlar, memetik ve genetik algoritmalar, karınca kolonisi, tabu arama, benzetimli tavlama algoritmaları ve meta-sezgisel yöntemler de kullanılmıştır.

Bu çalışmada, veri tabanı programı olarak MS Office Access 2013 programı ile veri tabanları ile bütünleşik olarak çalışabilen ve sınırsız karar değişkenine karar verebilen IBM Ilog Cplex Optimzation Studio 12.6 profesyonel bir optimizasyon programı temel materyal olarak kullanılmıştır.

#### 4.1. Tam Sayılı Programlama

Tam sayılı doğrusal programlama (TDP), değişkenlerden bazılarının ya da tümünün tam sayılı (ya da kesikli) değerler aldığı bir doğrusal programlama problemidir (Taha, 2007). Tam sayılı programlama modelleri, tümü tam sayılı, karışık tam sayılı ve 0 - 1 tam sayılı modeller olarak kategorize edilerek, bu kategorilerde çözüm yöntemleri mevcuttur.

Genel olarak gevşetilmiş çözümler tam sayılı çözümlere göre daha iyi amaç fonksiyonu değeri vermektedir Şekil 4.3.(Taha, 2007, s. 379). Bu kapsamda, Şekil-4.1. ve Şekil-2’de gösterildiği gibi amaç fonksiyonu minimizasyon problemlerinde  $Z_{\text{Gevşetilmiş}} \leq Z_{\text{Tamsayılı}}$  ve maksimizasyon problemlerinde  $Z_{\text{Tamsayılı}} \leq Z_{\text{Gevşetilmiş}}$  olarak değer alır.

Maks. $Z = 5x_1 + 4x_2$	Maks. $Z = 5x_1 + 4x_2$
$x_1 + x_2 \leq 5$ $10x_1 + 6x_2 \leq 45$ $x_1, x_2 \geq 0$	$x_1 + x_2 \leq 5$ $10x_1 + 6x_2 \leq 45$ $x_1, x_2 \geq 0$ ve tam sayı
	
Şekil 4.1. Gevşetilmiş Çözüm	Şekil 4.2. Tam Sayılı Çözüm

Şekil 4.3. Tam Sayılı Programlama Grafik Çözümü

#### 4.1.1. Dal-Sınır Algoritması

Amaç fonksiyonu maksimizasyon olan tam sayılı programlama modelinin dal sınır (DS) tekniği ile çözüm algoritması aşağıdadır.

**Adım-1:** Verilen model, değişkenlerin tamsayı değer alma koşulu göz önüne alınmadan (gevşetilmiş tam sayılı programlama modeli) olarak çözülür. Bu durum sonucunda üç durum ile karşılaşılabilir.

1. Sınırsız çözüm bölgesi veya geçersiz çözüm bulunması. Bu durumda durulur. Çünkü gevşetilmiş tam sayılı programlama probleminin çözümü yok ise, tam sayılı programlama probleminin de çözümü yoktur.

2. Bulunan çözümde değişkenlerin tamsayı değer alması. Bu durumda da durulur. Çünkü gevşetilmiş tam sayılı programlama probleminin çözümünde tam sayılı bir çözüm elde edilmiş ise, bu çözüm tam sayılı programlama probleminin de çözümüdür. Yani ilk adımda optimal çözüm elde edilmiş olur.

3. Bulunan çözümde tam sayı değer alması istenen değişkenlerden en az bir tanesinin tam sayı olmaması. Bu durumda amaç fonksiyonunun alt ve üst sınırları belirlenerek, ikinci adıma geçilir.

**Adım-2:** Tam sayı değer almamış bir değişkene göre dallara ayrılarak alt modeller elde edilir. Bulunan optimal çözümde amaç fonksiyonunun değeri alt sınırdan küçükse bu dal işlem dışı bırakılır. Alt sınırdan büyük değer elde edilmesi durumunda;

1. Değişkenler tam sayı değer almış ise üçüncü adıma
2. Değişkenler tam sayı değer almamış ise dördüncü adıma geçilir.

**Adım-3:** Alt sınır güncellenir. Yeni alt sınır üst sınıra eşit ise beşinci adıma geçilir, değilse dördüncü adıma geçilir.



**Adım-4:** İşlem dışı olmamış alt problem var ise ikinci adıma dönülür, yoksa beşinci adıma geçilir.

**Adım-5:** Algoritma sona erer. Son alt sınıra gelen çözüm optimal çözümdür. (Binay, ve diğerleri, 2001)

Kesirli değer alan değişkenlerden hangisinin seçileceğine ilişkin önerilen bazı seçim kuralları aşağıda belirtilmiştir.

1. Çözümde elde edilen kesirli değerler içinde en büyüğü seçilir.
2. Tam sayı değer alması gereken değişkenlere önem derecesi verilir ve seçim, en önemli değişkeni ilk olarak seçmek ile devam eder. Değişkenlere önem vermek için kullanılacak kriterler ise;

1. Değişkenin modelde önemli bir kararı temsil etmesi,
2. Amaç fonksiyonu katsayısının diğer değişkenlere göre daha büyük olması,
3. Modeli çözümlenmiş kişinin deneyimlerine göre modelde kritik bir öneme sahip değişken olması.

3. En küçük indise sahip olan değişkenden başlamak üzere uygulanabilecek herhangi bir seçim kuralı. (Binay, ve diğerleri, 2001)

Dal-sınır algoritması kısaca özetlenecek olursa;

1. Tam sayılı programlama modeli gevşetilmiş olarak çöz.
2. Tam sayı olması istenen fakat kesirli çıkan değişkenlere bağlı olarak yeni kısıtlar ekle ve yeni modeli tekrar gevşetilmiş olarak çöz.
3. Sonuçları ve karar değişkenlerini değerlendirerek ya dallan ya da dalı buda.
4. Tüm dallar budanana kadar devam et.

#### 4.1.2. Ekleme Algoritması

0-1 tam sayılı problemlerin çözümünde kullanılan bir algoritmadır. Ekleme algoritmasındaki kurallar DS algoritmasındakilerle aynıdır. Ana fark, LP<sub>s</sub>'leri çözmeyip sezgisel yaklaşım kullanılmasıdır. (Taha, 2007). Ayrıca; problemin başlangıçta aşağıda belirtildiği gibi düzenlemesi gerekmektedir.

1. Amaç fonksiyonu minimizasyon olmalıdır.
2. Amaç fonksiyonu katsayılarının tümü POZİTİF olmalıdır.
3. Kısıtların tümü ( $\leq$ ) şeklinde ifade edilmelidir.
4. Kısıtlardaki eşitsizliklerin tümü dolgu değişken kullanılarak eşitlik şeklinde ifade edilmelidir.

Ekleme algoritmasında dallanma, herhangi bir karar değişkeninin 0 ya da 1 değeri alma durumuna göre yapılır. Bu algoritmadaki sezgisel yaklaşım gereği olarak herhangi bir dalda elde edilen uygun bir çözüm ile ilgili olarak dolgu değişkenleri üzerinden değerlendirme yapılır.

Dallanma değişkeninin seçimi dolgu değişkenlerinin uygunsuzluk ölçüsü esas alınarak gerçekleştirilir. Bu ölçü sıfır değerli bir  $x_j$  değişkeninin 1'e yükselteceği varsayımını temel alır. (Taha, 2007)

$$I_j = \sum_{\text{tüm } i\text{'ler}} \min\{0, s_i - a_{ij}\}$$

Tüm dolgu değişkenleri  $\geq 0$  olması durumunda bu çözümün optimum olduğu sonucu çıkar. (Taha, 2007)

Her alt problem incelendiğinde aşağıda belirtilen üç durumdan biri ile karşılaşılır;

1. Uygun bir çözüme ulaşamaz.
2. İyi bir üst sınır bulamaz.
3. Uygun bir tam sayılı çözüm oluşur.

#### 4.1.3. Kesme Düzlemi Algoritması

DS algoritmasında olduğu gibi, kesme düzlemi algoritması da sürekli bir doğrusal programlama probleminin optimum çözümü ile başlar. Ancak bu yöntemde dallanma ve sınırlamadan çok, kesme adı verilen özel kısıtlar artarda oluşturularak çözüm uzayının düzenlenmesine gidilir. (Taha, 2007)

Kesme düzlemi algoritması, optimum çözümün tam sayılı bir uç noktada meydana gelmesi için kesmeler ekleyerek çözüm uzayını düzenler. Eklenen kesmeler, orijinal uygun tam sayı noktalarının herhangi birini elimine etmez, ama bu kesmeler en az bir tane uygun ya da uygun olmayan tam sayı noktasından geçmelidir. (Taha, 2007)

#### Kesmelerin Cebirsel Oluşumu:

Kesme düzlemi algoritması doğrusal programlama problemini sürekli değişkenler ile çözerek işe başlar. Elde edilen optimum tabloda tam sayılı olmayan temel değişken satırlarından biri (**kaynak satır**) seçilir, bu kaynak satırın kesirli bileşenlerinden oluşturulur. Bundan dolayı **kesirli kesme** adını alır. (Taha, 2007)

Kesirli kesmenin oluşturulması için, tamsayı olmayan katsayıların her biri tam sayılı ve kesirli kısımlarına ayrılır. Burada dikkat edilmesi gereken husus, kesirli kısmın daima pozitif olmasıdır. (Taha, 2007)

Tam sayılı kısımlar sol tarafta, kesirli kısımları da sağ tarafta toplanır (Taha, 2007). Keyfi verilecek karara bağlı olarak seçilecek kesme denklemi optimum tabloya yeni bir kısıt olarak eklenir. (Taha, 2007) Sonrasında iterasyonlar devam ettirilerek sonuca gidilir.

Burada dikkat edilmesi gereken önemli bir husus, kesirli kesmenin dolgu ve artık değişkenleri de içeren bütün değişkenleri tam sayı kabul ettiğidir. Bu da kesmenin sadece tümüyle tam sayılı olan problemlerle ilgili olması demektir. (Taha, 2007)

Örneğin;

$$\frac{5}{2} = \left( 2 + \frac{1}{2} \right)$$
$$-\frac{7}{3} = \left( -3 + \frac{2}{3} \right)$$

#### 4.1.4. Tam Sayılı Programlamada Kullanılan Özel Durumlar

Bir karar modelinde problemin yapısı gereği seçenekler arası zorunlu ilişkiler söz konusu olabilir. Bu durumlarda, her bir tutum veya zorunlu ilişki karşılığı modele özel kısıtların yazılması gerekir. Aşağıda bu durumlar genel ifadeleri ile ve örneklerle açıklanmıştır. (Binay, ve diğerleri, 2001, s. 210)

##### 4.1.4.1. “Ya-ya da” Şeklindeki Kısıtlar

İki kısıt aşağıdaki şekilde verilmiş olsun.

$$(1) \quad f(x_1, x_2 \dots x_n) \leq 0, \quad (2) \quad g(x_1, x_2 \dots x_n) \leq 0$$

(1) ve (2) numaralı kısıtlardan en az birinin sağlanması “ya-ya da” şeklinde kısıtlar olarak adlandırılır. Modele (3) ve (4) kısıtları ilave edilmesi (1) ve (2) numaralı kısıtların en azından birinin sağlanmasını temin eder. (Binay, ve diğerleri, 2001)

$$(3) \quad f(x_1, x_2 \dots x_n) \leq My$$

$$(4) \quad g(x_1, x_2 \dots x_n) \leq M(1 - y)$$

$$(5) \quad y \in \{0,1\}$$

M modeldeki diğer kısıtlara göre elde edilen tüm değişkenlerin  $x_1, x_2, \dots, x_n$  değerlerini  $f(x_1, x_2, \dots, x_n) \leq My$  ve  $g(x_1, x_2, \dots, x_n) \leq My$  kısıtlarında sağlayacak kadar büyük bir sayıyı temsil eder. (Binay, ve diğerleri, 2001, s. 211), (Winston, 2004, s. 487)

#### 4.1.4.2. M Adet Kısıttan K Adedinin Etkili Olması

Eğer modelde aşağıdaki gibi m adet kısıt olsaydı ve bunlardan sadece k adedinin sağlanması istenseydi;

**Çizelge 4.1.** Ya-Ya da Kısıtları Dönüşümü

Model	Modeldeki Değişiklikler	
$f_1(x_1, x_2, \dots, x_n) \leq d_1$	$f_1(x_1, x_2, \dots, x_n) \leq d_1 + My_1$	$\sum_{i=1}^m y_i = m - k$
$f_2(x_1, x_2, \dots, x_n) \leq d_2$	$f_2(x_1, x_2, \dots, x_n) \leq d_2 + My_2$	
·	·	
·	·	
$f_m(x_1, x_2, \dots, x_n) \leq d_m$	$f_m(x_1, x_2, \dots, x_n) \leq d_m + My_m$	

Yukarıdaki yapı incelendiğinde aslında “ya-ya da” kısıtlarının genelleştirilmiş hali olduğu görülür. m=2 ve k=1 alınırsa “ya-ya da” kısıt tipi elde edilir. (Binay, ve diğerleri, 2001)

#### 4.1.4.3. Bir Kısıtın N Adet Farklı Değer Alması

Aşağıdaki gösterim, bir kısıtın sağ taraf değerinin alabileceği N farklı değeri ifade etmektedir.

$$f(x_1, x_2, \dots, x_n) = b_1, b_2, \dots, b_N$$

Sağ taraf değerlerinden sadece bir tanesinin kullanılması durumu, modele eklenecek şekilde aşağıdaki kısıtlar ile sağlanabilir. (Binay, ve diğerleri, 2001)

$$f(x_1, x_2, \dots, x_n) = \sum_{i=1}^N b_i y_i$$

$$\sum_{i=1}^N y_i = 1$$

$$y \in \{0,1\} \quad i = 1,2,\dots, N$$

#### 4.1.4.4. 0-1 Değişkenlerle İlgili Durumlar

0-1 değişkenlerin kullanımını buraya kadar anlatılan değişik örneklerde incelenmekle birlikte, aşağıda kullanımların özetinin verilmesi faydalı olacaktır.

1. m adet kısıttan k adedinin geçerli olması (her kısıtın sağ taraf değerine  $M y_i$  ilave edildikten sonra);

$$\sum_{i=1}^m y_i = m - k$$

2. n adet değişkenden **tam** k adedinin geçerli olması;

$$\sum_{i=1}^n x_i = k$$

3. n adet değişkenden **en fazla** k adedinin geçerli olması;

$$\sum_{i=1}^n x_i \leq k$$

4. n adet değişkenden **en fazla** k adedinden vazgeçilecek olması;

$$\sum_{i=1}^n x_i \geq n - k$$

5.  $j$  inci deęişkenin seęilmesinin  $i$  inci deęişkene baęlı olması;

$$x_i \geq x_j$$

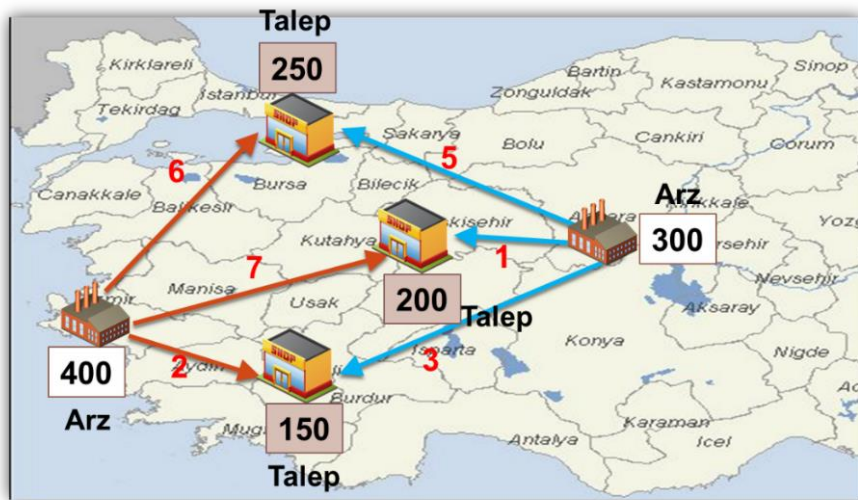
6.  $j$  inci ve  $i$  inci deęişkenlerin birlikte seęilmesi veya reddi;

$$x_i = x_j$$

#### 4.2. Ulaştırma ve Atama Problemleri

Ulaştırma ve atama problemleri genel gösterimleri doğrusal programlama modeli olmakla birlikte kısıtlarının özel yapısı nedeniyle, simpleks algoritmasına göre daha etkin çözüm yöntemleri geliştirilmiş olan model türleridir.

Ulaştırma problemleri Şekil 4.4.teki gibi belirli bir kaynaktan belirli bir merkeze taşınacak malzemelerin minimum maliyetle ve ne miktarda taşınma yapılacağı konusu üzerinde yoğunlaşırken, atama modellerinde belirli bir kaynaktan belirli hedeflere atama yapılıp yapılmayacağına karar verilmektedir. Ayrıca; atama problemleri ulaştırma modelleri gibi gösterilmekle birlikte karar deęişkenlerinin türü 0-1 olduğundan ulaştırma problemlerine göre farklı bir çözüm yöntemi bulunmaktadır. (Binay, ve dięerleri, 2001, s. 257)



Şekil 4.4. Ulaştırma Problemi Gösterimi

#### 4.2.1. Ulaştırma Problemleri

Doğrusal amaç ve kısıt fonksiyonlarına sahip DP'nin özel bir problem türü olan ulaştırma problemlerini simpleks yöntemle çözmek mümkündür. Bununla birlikte ulaştırma probleminin özel yapısından dolayı işlemlerde bir kısım azaltmalar yapılabilmektedir. Bu konuda ilk çalışma Rus matematikçisi L.V. Kantorovich tarafından yapılmıştır. Ulaştırma modellerindeki asıl gelişme ise Dantzig tarafından İlk defa 1947 yılında Hitchcock Petrol Endüstrisinde uygulanmıştır. (Binay, ve diğerleri, 2001, s. 257)

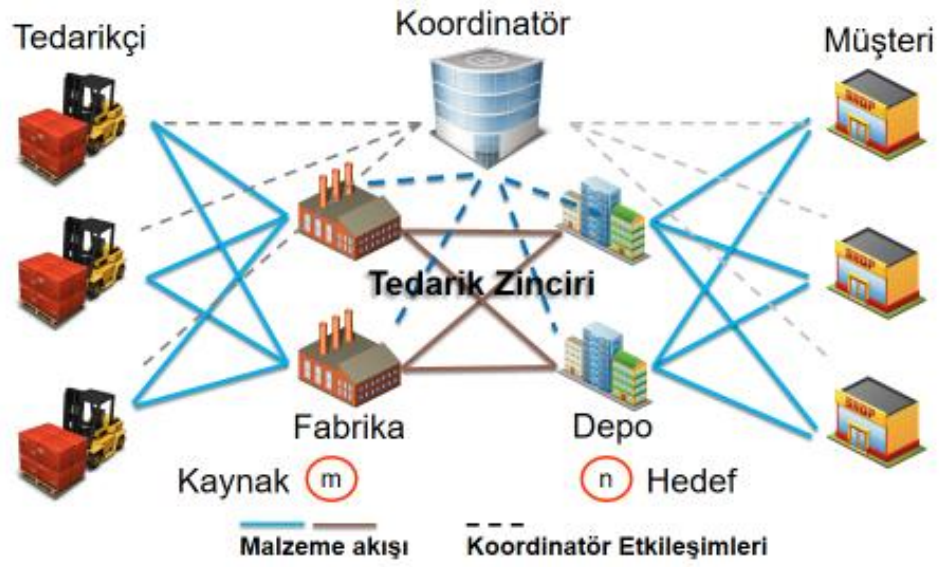
Ulaştırma modelinin amacı bir işletmenin belirli kapasitedeki kaynaklardan (üretim merkezleri, fabrika vb.), belirli talebi olan tüketim merkezlerine (depo vb.) göndereceği malların toplam ulaştırma maliyetini en küçük yapacak biçimde gönderilmesini sağlamaktır.

Ulaştırma modelleri ile genel olarak; üretim ve tüketim merkezleri arasında optimal mal dağıtım programlarının belirlenmesi, yapılacak işlerin makinalara dağıtımı, üretim planlaması, çeşitli şebeke ağ problemleri ve işletmelerin kuruluş yeri seçimi problemleri gibi konularda çalışılmaktadır. Örneğin; organizasyonların Şekil 4.5.te görülen tedarik zinciri yapısı içerisinde yer alan dağıtım ağı incelendiğinde, ulaştırma modelleri kullanılarak gerek tedarikçi-fabrika, fabrika-depo ve depo-müşteri gibi çeşitli kademeler arasında bulunan dağıtım sorunlarına, gerekse bu aşamaların kendi içerisinde karşılaşılan atama problemlerine çözümler geliştirilmektedir.

Bir ulaştırma modelinin genel olarak şematik yapısı ile tablo yapısı Şekil 4.6.daki gibidir. Ancak modellerin yapısı üretim merkezleri ile tüketim merkezleri arasındaki ilişkilere göre değişkenlik gösterebilmektedir.

Ulaştırma modeli şeklinde formüle edilen bir problem, simpleks yöntemi ile çözülebildiği gibi kendine has ulaştırma algoritması, atama ve aktarma modelleri gibi tekniklerle, daha az zamanda ve daha az hesaplamalarla da çözülebilir.





Şekil 4.5. İşletmelerde Dağıtım Ağı

Şebeke Gösterimi		Tablo Gösterimi				
<p><b>Kaynak</b>      <b>Hedef</b></p>		Kaynak\Hedef	1	...	n	Arz
1	1	$C_{11}$	$C_{1.}$	$C_{1n}$	$A_1$	
2	2	$C_{.1}$	$C_{..}$	$C_{.n}$	$A_{.}$	
...	...					
m	n	$C_{m1}$	$C_{m.}$	$C_{mn}$	$A_m$	
<b>Talep</b>	<b>B<sub>1</sub></b>	<b>B.</b>	<b>B<sub>n</sub></b>			

Şekil 4.6. Ulaştırma Modeli Şebeke ve Tablo Gösterimi

### Matematiksel Model

Bir ulaştırma modelinin genel olarak matematiksel gösterimi ise aşağıdaki gibidir.

### Tanımlar:

$s_i$  = i kaynağının arz miktarı (kapasitesi). ( $i = 1, \dots, m$ )

$d_j$  = j hedefinin talep miktarı. ( $j = 1, \dots, n$ )

$c_{ij}$  = i kaynağından j hedefine gönderilen malların birim ulaştırma maliyeti.

$x_{ij}$  = i kaynağından j hedefine gönderilen mal miktarı.

### Amaç Fonksiyonu

$$\text{Min. } Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

### Kısıtlar

#### 1. Arz Kısıtları

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

#### 2. Talep Kısıtları

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

#### 3. Yeter Koşul

$$\sum_{j=1}^m s_i = \sum_{j=1}^n d_j$$

#### 4. Negatif olmama koşulu

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

### Varsayımlar

- Modelde kullanılan tüm bilgiler ve probleme konu olan mal ve hizmetler, bütün üretim ve tüketim merkezleri için aynı birim ve türden olmalıdır.

- Her bir üretim merkezi ile her bir tüketim merkezi arasında bir birim malın taşınma ücreti belirli olmalıdır.
- Her bir arz ve tüketim merkezlerindeki toplam arz ve toplam talep tam olarak bilinmelidir.
- Üretim merkezlerinin toplam kapasitesinin, tüketim merkezlerinin toplam talep miktarına eşit olması, (eğer eşitlik yoksa ve elde bulundurma veya fırsat maliyeti bulunmuyorsa, eşitliği sağlamak için çözümde hayali üretim veya kukla tüketim merkezleri oluşturulur ve birim başına maliyetleri sıfır alınır).
- Gönderilen malların doğrudan doğruya üretim merkezlerinden, tüketim merkezlerine direkt gönderilmesi, yani üretim merkezleri veya tüketim merkezleri arasında nakil yapılmasının olanaksız olması,
- Taşıma maliyetinin birim başına sabit olması gerekir.

### **Ulaştırma Algoritması**

Ulaştırma algoritması, simpleks yönteminin adımlarını aynen takip eder.

Normal simpleks tablosunu kullanmak yerine, ulaştırma modelinin özel yapısının avantajını kullanarak algoritma daha uygun bir hale getirilir.

**Adım-1:** Modelin dengeli olup olmadığı kontrol edilir. Dengeli ise adım-2'ye geç, değilse dengele.

**Adım-2:** Başlangıç uygun çözümünü belirle adım-3'e geç.

**Adım-3:** Tüm temel dışı değişkenler içerisinde giren değişkeni belirlemek üzere simpleks yöntemin optimumluk koşulunu kullan. Optimumluk koşulu sağlanmışsa dur, yoksa adım-4'e git.

**Adım-4:** Mevcut tüm temel değişkenler içinden çıkan değişkeni belirlemek üzere simpleks yöntemin uygunluk koşulunu kullan ve yeni temel çözümü bul. Adım-3'geri dön.

#### 4.2.1.1. Başlangıç Çözümünün Bulunması

Dengeli ulaştırma problemlerinin başlangıç uygun temel çözümünün bulunmasında 3 farklı metot kullanılır;

1. Kuzeybatı Köşesi Metodu
2. En Düşük Maliyet Metodu
3. VOGEL (VAM) Metodu

Konunun daha iyi anlaşılması açısından bir örnekle tüm durumları inceleyelim.

**Çizelge 4.2.** Taşıma Modeli Örneği

		İşleme Merkezleri				Arz
		1	2	3	4	
Silo	1	<b>10</b> $x_{11}$	<b>2</b> $x_{12}$	<b>20</b> $x_{13}$	<b>11</b> $x_{14}$	<b>15</b>
	2	<b>12</b> $x_{21}$	<b>7</b> $x_{22}$	<b>9</b> $x_{23}$	<b>20</b> $x_{24}$	<b>25</b>
	3	<b>4</b> $x_{31}$	<b>14</b> $x_{32}$	<b>16</b> $x_{33}$	<b>18</b> $x_{34}$	<b>10</b>
Talep		<b>5</b>	<b>15</b>	<b>15</b>	<b>15</b>	

Örnek 1: Güneş yolu taşımacılık şirketi üç silodan dört işleme merkezine kamyonlarla hububat taşımaktadır. Arz ve talep miktarları (kamyon sayısı cinsinden) eşit olup, farklı rotalardaki kamyon başına ulaştırma maliyetleri ile birlikte ulaştırma modeli Çizelge 4.2.de verilmiştir. Birim taşıma maliyetleri ( $c_{ij}$ ) her kutunun kuzey doğusunda yer almaktadır.

Modelin amacı, silolarla işleme merkezleri arasındaki taşıma maliyetlerini minimum kılan taşıma programını belirlemektir. Bu,  $i$  silosundan  $j$  işleme merkezine ( $i=1,2,3$ ;  $j=1,2,3,4$ ) taşınan  $x_{ij}$  miktarını belirlemekle eşdeğerdir.

#### 4.2.1.1.1 Kuzeybatı Köşesi Metodu

$m$  kaynaklı  $n$  hedefli bir ulaştırma modeli  $m+n$  adet kısıt denklemine sahiptir. Ulaştırma modelleri daima dengelenmiş olduğu için bu denklemlerden biri fazla (hayali) olabilir. Dolayısıyla modelin  $m+n-1$  tane kısıt denklemi olur. Bu da başlangıç temel çözümünün  $m+n-1$  temel değişkenden oluşması demektir. Yöntem doğal olarak  $x_{11}$  değişkeninin yer aldığı hücre ile başlar. (Taha, 2007)

**Adım 1:** En kuzeybatıdaki hücreyi ( $i, j$  hücresi) bul. (Taha, 2007)

**Adım 2:**  $x_{ij}$ 'ye alabileceği en büyük değeri ata ( $x_{ij} = \min(s_i, d_j)$ ). Atanan miktarı arz ve talep miktarlarından çıkar. (Taha, 2007)

**Adım 3:** İleride tekrar atama yapılmasını önlemek için çıkarma sonucu sıfır arz ya da talebe ulaşan satır ya da sütunu iptal et. Hem satır hem de sütun aynı anda sıfıra gelmişse birini seç ve iptal edilmeyen satır (sütun)'daki sıfır arzı (talebi) dikkate alma (terk et). (Taha, 2007)

**Adım-4:** İptal edilmeyen sadece bir satır ya da sütun kaldığında dur. Aksi halde, bir önceki işlemde sütun iptal edilmişse sağ kutuya, satır iptal edilmişse bir aşağıdaki kutuya geç. Adım-1'e git. (Taha, 2007)

Çizelge 4.3. KBK Çözümü

		İşleme Merkezleri				Arz
		1	2	3	4	
Silo	1	10 5 → 10	2	20 $x^{13}$	11 $x^{14}$	15
	2	12 $x^{21}$	7 ↓ 5 →	9 15 →	20 5	25
	3	4 $x^{31}$	14 $x^{32}$	16 $x^{33}$	18 ↓ 10	10
Talep		5	15	15	15	

#### 4.2.1.1.2 En Düşük Maliyet Metodu

En düşük maliyetler yöntemi en ucuz rota üzerine yoğunlaştığı için daha iyi bir başlangıç çözümü bulmaktadır. (Binay, ve diğerleri, 2001)

Çizelge 4.4. EKM Çözümü

		İşleme Merkezleri				Arz
		1	2	3	4	
Silo	1	10 $x^{11}$	2 15	20 $x^{13}$	11 0	15
	2	12 $x^{21}$	7 $x^{22}$	9 15	20 10	25
	3	4 5	14 $x^{32}$	16 $x^{33}$	18 5	10
Talep		5	15	15	15	

**Adım 1:** Maliyeti en az olan hücreyi (i, j hücresi) bul. (Binay, ve diğerleri, 2001)

**Adım 2:**  $x_{ij}$ 'ye alabileceği en büyük değeri ata ( $x_{ij} = \min(s_i, d_j)$ ). KBK metodunda olduğu gibi, i satırını ya da j sütununu iptal et, iptal edilmeyen satır veya sütundaki arz veya talebi  $x_{ij}$  kadar azalt. (Binay, ve diğerleri, 2001)

**Adım 3:** İptal edilmeyen bir satır ya da sütun kalıncaya kadar devam et. (Binay, ve diğerleri, 2001)

#### **4.2.1.1.3VOGEL (VAM) Metodu**

En düşük maliyetler yönteminin geliştirilmiş hali olup, genelde en iyi başlangıç çözümünü vermektedir. (Taha, 2007, s. 181)

**Adım 1:** Pozitif arzlı(talepli) her satır ve satırdaki (sütundaki) en küçük birim maliyeti aynı satırın (sütunun) ikinci en küçük birim maliyetinden çıkararak bir ceza ölçüsü belirle. (Taha, 2007)

**Adım 2:** En büyük cezaya sahip satır ya da sütunu saptı. Eşitlik halinde rasgele seçim yapılabilir. Bu satır ya da sütundaki en düşük maliyetli kutuya mümkün olduğunca fazla miktarda atama yap. Kalan arz ve talepleri hesapla ve sıfırlanan satır ya da sütunu iptal et. Aynı anda sıfırlanan satır ya da sütunlar varsa, sadece birini iptal ederek kalan satıra (sütuna) sıfır miktarda arz (talep) ata. (Taha, 2007)

#### **4.2.1.2. Ulaştırma Simpleks Algoritması (MODI Metodu)**

Ulaştırma problemlerinin simpleks yöntemi ile çözümü için asıl modelin duali oluşturulur. Asıl modeldeki her bir kısıta karşılık gelen aşağıdaki gibi yeni değişkenler tanımlanır. (Bakır ve Altunkaynak, 2003, s. 130)

$u_i$  : Asıl modelin arz kısıtlarına karşılık gelen değişken

$v_j$  : Asıl modelin talep kısıtlarına karşılık gelen değişken

### Amaç Fonksiyonu

$$\text{Min } Z = \sum_{i=1}^m s_i u_i + \sum_{j=1}^n d_j v_j$$

### Kısıtlar

$$u_i + v_j \leq c_{ij} \quad i = 1, 2, \dots, m \text{ ve } j = 1, 2, \dots, n$$

$u_i, v_j$  sınırlandırılmamış

Dualite teoreminden dual ve asıl modellerin optimumunun karşılıklı olarak diğeri için de optimum çözüm olduğu bilinmektedir. Yukarıdaki dual modelin kısıtı için aylak değişkeni aşağıdaki gibi tanımlayalım. (Bakır ve Altunkaynak, 2003, s. 131)

$$w_{ij} = c_{ij} - u_i - v_j$$

Gerçekte  $w_{ij}$ ,  $x_{ij}$  değişkenli asıl modelle ilgili indirgenmiş maliyettir ve dolayısıyla  $x_{ij}$ 'dek birim değişimle birlikte amaç fonksiyonundaki değişmeyi ölçer. Asıl modeldeki **temel değişken** için karşılık gelen dual kısıt şu eşitliği sağlamalıdır. (Bakır ve Altunkaynak, 2003, s. 131)

$$u_i + v_j = c_{ij} \text{ ya da } w_{ij} = 0, \forall x_{ij} \text{ için}$$

Asıl modelin kısıtlarından birisi fazladan olduğundan dual değişkenlerden bir tanesini keyfi olarak sıfıra eşitleriz. Diğerleri ise bir üstteki eşitlikten hesaplanır. (Bakır ve Altunkaynak, 2003, s. 131)



Eğer dual problemin kısıtları dual çözüm tarafından sağlanıyorsa, hem asıl hem de dual çözüm optimaldir. Dual uygunluk için koşul ise, tüm **temelde olmayan değişkenler** için (yükleme yapılmayan hücreler)

$$u_i + v_j \leq c_{ij} \text{ ya da } w_{ij} \geq 0, \forall x_{ij} \text{ için}$$

biçiminde olmalıdır. Bu sonuç ulaştırma simpleks algoritmasının temelini oluşturmaktadır. (Bakır ve Altunkaynak, 2003, s. 132).

**Adım 1:** (*Başlangıç tablosunun oluşturulması*). Daha önce belirtilen yöntemlerden birisiyle başlangıç tablosu için bir uygun çözüm elde edilir.

**Adım 2:** Cari uygun çözüm için dual değişkenlerin hesaplanması ve optimumluk kontrolü.

- Temel değişkenlere karşılık gelen  $u_i$  veya  $v_j$  dual değişkenlerden birine keyfi olarak 0 değeri atanır.  $u_i + v_j = c_{ij}$  ilişkisinden diğer dual değişken değerleri hesaplanır.

- Temelde olmayan hücreler (değişkenler) için dual aylak değişken  $w_{ij}$ 'leri hesaplanır.

- Eğer  $w_{ij}$ 'lerin tüm değerleri negatif olmayan değerler ise ( $w_{ij} \geq 0$ ), optimal çözüme ulaşılmış demektir. Aksi durumda Adım 3'e geçilir.

**Adım 3:** (*Temel çözümün değiştirilmesi*).

- *Temele girecek değişkenin bulunması:* En küçük negatif değere sahip  $w_{ij}$ 'nin olduğu hücredir.

- *Temeli terk edecek hücrenin bulunması (Oran Testi):* Temele giren hücreden başlayan ve sadece diğer tüm temeldeki hücrelerden geçerek tekrar temele girecek

hücreye dönen bir döngü tanımlanır. Bu döngü tablo üzerinde 90<sup>0</sup>lik açılarla gösterilir. Giren hücre + ve diğer temel hücreler -+-. ardışıklığında işaretlenir. Sonra temeli terk edecek hücreyi bulmak için – işaretli hücreler arasından en küçük değere sahip olan belirlenir. (Eğer aynı değerli böyle birden fazla hücre varsa keyfi olarak birisi seçilir.) Seçilmeyen hücreler temelde kalacak pivot işlemi boyunca 0'a gidecektir. Terk eden hücre için temel değişkenin değerini  $\delta$  ile gösterelim.

- Temel çözümü değiştirme (pivot işlemi): + ile işaretlenmiş her hücre için temel değişkenin değeri  $\delta$  kadar artırılır, - ile işaretlenmiş değişken değerleri ise  $\delta$  kadar azaltılır. Atılacak hücre çıkarılır ve girecek hücre  $\delta$  değeri ile dâhil olur. Adım 2'ye dön. (Bakır ve Altunkaynak, 2003, s. 133)

Konuyla ilgili daha önce yapılan çalışmalarda ulaştırma probleminin kullanılma durumu incelendiğinde; Yıldız (2004), Dammak, Elloumi ve Kamoun (2006) ile Çoruhlu (2007)'nin çalışmalarında bu konudan yararlandıkları görülmüştür.

#### **4.2.2. Atama Problemleri (Macar Metodu)**

“İşe en uygun kişi” muhtemelen atama modeliyle halledilebilecek bir konudur. Bu durum işçilerin işlere atanması şeklinde bir ifadeyle daha iyi anlatılabilir. Modelin amacı, işçilerin işlere optimum (en düşük maliyetli) atanmasını belirlemektir. (Taha, 2007, s. 192)

Tipik bir atama probleminde işçilerin makinelere dağıtımı, kişilerin işlere tayini, satış personelinin satış bölgelerine dağıtımı vb. durumlarla karşılaşılabılır. Bu türden problemler için amaç, maliyetin en küçüklenmesi, karın en büyüklenmesi, toplam zamanın en küçüklenmesi vb. olabilir. (Bakır ve Altunkaynak, 2003, s. 137)

Aslında, atama modeli, kaynakları işçiler, hedefleri de işler olan özel bir ulaştırma modelidir. Ulaştırma problemleri ile arasındaki temel fark; her bir kaynaktaki arz miktarı ve her bir hedefteki talep miktarı sürekli olarak 1'e eşittir. Atama modeli, normal bir ulaştırma modeliymiş gibi çözülebilir. Bununla birlikte, arz ve talep miktarının 1'e eşit olması Macar Yöntemi diye adlandırılan basit bir çözüm

algoritmasının geliştirilmesine yol açmıştır. Yeni yöntem, ulaştırma modeliyle tamamen ilişkisiz görünse bile, ulaştırma modeli gibi simpleks yöntemden gelmektedir. (Taha, 2007, s. 193)

## Matematiksel Model

### Tanımlar:

$a_i$  = i kaynağının arz miktarı (kapasitesi). ( $i = 1, \dots, m$ )

$b_j$  = j hedefinin talep miktarı. ( $j = 1, \dots, n$ )

$c_{ij}$  = i kaynağından j hedefine gönderilen malların birim ulaştırma maliyeti.

$x_{ij}$  = i kaynağından j hedefine gönderilen mal miktarı.

### Amaç Fonksiyonu

$$\min Z = \sum_{i=1}^m \sum_{j=1}^n x_{ij} c_{ij}$$

### Kısıtlar

#### 1. Arz Kısıtları

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

#### 2. Talep Kısıtları

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

### 3.Yeter koşul

$$\sum_{j=1}^m s_i = \sum_{j=1}^n d_j$$

### 4.Negatif olmama koşulu

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

## Macar Metodu

**Adım 1:** nxn boyutlu orijinal maliyet matrisindeki her satırın minimumunu bul ve bunu satırın tüm elemanlarından (miktarlar) çıkar.

**Adım 2:** Adım 1’de elde edilen matrisin her sütununun minimumunu bulduktan sonra, bunu sütundaki tüm elemanlardan (miktarlar) çıkar.

**Adım 3:** Uygun atama adım 1 ve 2’de gerçekleştirilemiyorsa;

1. Son indirgenmiş matriste tüm sıfırları kapsayacak şekilde ve minimum sayıda yatay ve düşey çizgiler çiz.

2. Üzeri çizilmemiş elemanlar içinden en küçüğünü seç ve bu elemanı üzeri çizilmemiş her elemandan çıkar. Aynı elemanı, iki kapatma çizgisinin kesiştiği yerdeki elemanla topla.

3. Bu yeni tabloda sıfır değerini alan elemanlarla atama yapılamıyorsa adım 3’ü tekrarla. Aksi halde adım 4’e giderek optimum atamayı belirle.

**Adım 4:** Adım 2’deki matriste ortaya çıkan sıfır elemana ilgili optimum atamayı yap.

### 4.3. Hedef Programlama

Hedef programlama; verilen kısıtlayıcılar altında amaç ölçütünü doğrudan maksimum veya minimum kılmaktan ziyade hedeflerin kendi içindeki sapmalarını minimum kılmaya odaklanan çok amaçlı bir tekniktir. Örneğin hedef kısıtlayıcısı;

$\leq$  ise  $d_i^-$  istenen,  $d_i^+$  ise istenmeyen,

$\geq$  ise  $d_i^+$  istenen,  $d_i^-$  ise istenmeyen,

$=$  ise  $d_i^+$  ve  $d_i^-$  her ikisi de istenmeyen sapma değişkenleridir.

Klasik bir problemin hedef programlama olarak modellenmesi için; modele sıfırdan büyük eşit olan sapma değişkenleri eklenmelidir. Bu değişkenler hedef kısıtlayıcılarına bağlı olarak istenen veya istenmeyen değişken olarak adlandırılır.

Klasik minimizasyon ya da maksimizasyon problemleri; genellikle tek amaçlı ve kısıtlayıcılarının sınırları kesin olarak modellenmektedir. Ancak; bu durum gerçekte her zaman mümkün değildir. Bu sorunun çözümü ise hedef programlama ile iki aşamada mümkündür.

Öncelikle istenilen denkleme sıfırdan büyük ya da eşit olan iki sapma değişkeni eklenir. Bu değişkenlerden  $d^-$  hedeften negatif yöndeki sapmayı,  $d^+$  ise pozitif yöndeki sapmayı ifade eder. Bu sayede kısıt, söz konusu değişkenlerden birinin değer almasına göre hedeften negatif ya da pozitif yönde değişme imkânına kavuşur. İkinci aşamada, eşitsizliğin yönüne bağlı olarak istenmeyen yöndeki sapma değişkeni amaç fonksiyonuna eklemek suretiyle model çok amaçlı bir yapıya evrilir.

Hedef programlama modellerinin çözümünde iki algoritma kullanılmaktadır. Her iki yöntem de çok sayıda amaç fonksiyonunun tek bir amaç fonksiyonu gibi temsil edilmesine dayanır. Ağırlıklandırma yönteminde, tek bir amaç fonksiyonu, problemin hedeflerini temsil eden fonksiyonların ağırlıklandırılmış toplamı haline getirilir. Öncelik koruma yöntemi, önem derecelerine göre hedeflerin önceliklendirilmesi ile başlar. Model daha sonra, yüksek öncelikli hedefin optimum değerinin düşük öncelikli

hedef tarafından kötüleştirilmesine izin verilmeyecek şekilde, her seferinde bir hedefi optimum kılar. (Taha, 2007, s. 348)

Önerilen iki yöntem aynı çözümü üretmez, bu bakımdan birbirinden farklıdır. Bununla birlikte, yöntemlerden her biri belirli karar verme tercihlerini karşılamak için tasarlanmış olduğundan, ikisinden herhangi birinin daha üstün olduğu ileri sürülemez. (Taha, 2007, s. 349)

Ağırlıklandırma yönteminde kullanılan birleştirilmiş amaç fonksiyonu aşağıdaki şekilde tanımlanır.

$$\min .z = w_1G_1 + w_2G_2 + \dots + w_nG_n$$

Burada  $w_i, i = 1, 2, \dots, n$  her bir hedefin göreceli önemiyle ilgili karar vericinin tercihlerini yansıtan pozitif ağırlıklardır.

Önceliği koruma yönteminde ise, problemin  $n$  sayıda hedefi, karar vericinin değerlendirmesine göre önem sırasına sokulur ve bu aşağıdaki şekilde ifade edilir.

$$\min .G_1 = P_1 \quad (\text{en yüksek öncelik})$$

.

.

$$\min .G_n = P_n \quad (\text{en düşük öncelik})$$

$P_i$  değişkeni  $i$ . hedefi tanımlayan  $d_i^-$  veya  $d_i^+$  sapma değişkenlerinin bileşenidir.

Çözüm prosedürü, en yüksek öncelikli  $G_1$  hedefiyle başlayıp en düşük öncelikli  $G_n$  hedefiyle bitirilerek her seferde tek hedefli bir problemi çözer. Süreç öyle gerçekleştirilir ki, düşük öncelikli hedefle elde edilen çözüm, yüksek öncelikli hedefler için daha önce bulunmuş çözümleri kötüleştirmez.

#### 4.4. Matematiksel Modellemede Tablo İndis Kullanımı

İndis, özellikle matematikte kullanılan bir harf, benzer fakat yine de değişik biçimlerde iki veya daha çok kez kullanılmak istendiğinde harfin üstüne veya altına eklenen ayırıcı işaret<sup>2</sup> anlamına gelmektedir. Matematiksel modellemelerde indis kullanımı oldukça yaygındır. Gerçek hayatta karşılaşılan problemler genellikle büyük boyutlarda olduğundan çok sayıda indis kullanımı ihtiyacı ortaya çıkmaktadır.

Herhangi bir  $x$  değişkeni matematiksel bir modelde kullanılmak üzere  $i$  ve  $j$  gibi iki indisli olarak tanımlansın ( $x_{ij}$ ). Buradaki  $i$  indisi satırları,  $j$  indisi ise sütunları ifade etmektedir. Model 2 boyutta olduğu için bu tip değişkenlerin verileri genellikle çapraz tablo olarak Çizelge 4.5.te olduğu gibi kullanılmaktadır. Ancak; bu değişkene üçüncü bir indis olarak örneğin  $k$  indisini ekleyecek olursak ( $x_{ijk}$ ) bu değişken için her bir  $i$  ya da  $k$  indisi için birden fazla çapraz tablo oluşturulması gerekecektir.

Çizelge 4.5. İki İndisli Bir Değişkenin Tablo Karşılığı

$i / j$	1	2	3
1			
2			
3			

Görüldüğü gibi bu durum modellemeyi oldukça zorlaştırmaktadır. Elbette indis sayısının daha da artması halinde modelleme artık içinden çıkılmaz bir hale

---

<sup>2</sup> TDK: Türk Dil Kurumu Genel Türkçe Sözlük.

dönüşecektir. Ancak; gelişen bilgisayar yazılımları ile birlikte indis bir değişkene değil de bir tabloya bağlı olarak Çizelge 4.6.daki şekliyle tanımlanmak suretiyle bu sorun daha kolay aşılabilmektedir.

**Çizelge 4.6.** İndise Bağlı Tablo Yapısı

i	j	Değer
1	1	
1	2	
2	2	
.	.	

Konuyla ilgili olarak; gözetmenlik ataması için bir personelin hangi gün hangi oturumlarda mazereti olduğunu verisini içeren bir değişken tanımlanacak olsa; i:gün, j:oturum ve k:personel için indis ise, bu değişken klasik olarak  $x_{ijk}$  şeklinde üç indisli olarak düşünülecektir. Ancak; bunun yerine gün, oturum, personel, mazeret ve hatta birikimli oturum gibi -istendiği kadar bu konuyla ilgili alan eklenebilir- alanları temsil eden için aşağıdaki tablo tanımlanması durumunda o zaman söz konusu  $x_{ijk}$  değişkeni  $x_i$  olarak tanımlanacaktır. Örneğin;

{per\_mazeret} : Her bir personelin gözetmenlikle ilgili hangi gün hangi oturumlarda mazereti olduğunu gösteren bir tablo olsun.

G: Günler, O:Oturumlar, PM :per\_mazeret tablosunu temsil eden değişken olsun.



$i \in PM$  için:

**tuple** per\_mazeret {

**int** brkml\_oturum : PM tablosundaki i.brkml\_oturum, birikimli oturum değişkenini,

**int** gün : PM tablosundaki i.gün, gün değişkenini,

**int** oturum : PM tablosundaki i.oturum, oturum değişkenini,

**int** per\_id : PM tablosundaki i.per\_id, personel değişkenini,

**boolean** mazeret : PM tablosundaki i.mazeret, mazeret değişkenini temsil etsin.}

$$\sum_{i \in G} \sum_{j \in O} x_{ijk} = 1 \quad \forall k \in P \quad (1)$$

$$\sum_{i, per\_id=k} x_i = 1 \quad \forall k \in P \quad (2)$$

Yukarıda belirtilen iki formülü karşılaştıracak olursak; (1) numaralı denklemde 3 adet indis ve 2 adet toplam sembolü bulunmaktadır. (2) numaralı denklemde ise 1 adet değişken ve toplam sembolü kullanılmıştır. Görüldüğü gibi tablo indis ile matematiksel modeller çok daha kolay bir şekilde ve sade bir şekilde tanımlanmaktadır.

#### 4.5. İlişkisel Veritabanı

İlişkisel veritabanı tablo, ilişki ve bunlar üzerinde tanımlanan kurallardan oluşur. Tablo sınırlı sayıda tanımlı sütundan ve sınırsız sayıda tanımsız satırdan oluşan bir veri yapısıdır. Veri tabanı birden çok tablodan oluşabilir ve tablolar arasında sütunlar üzerinden ilişki kurulur. İlişkisel veri tabanı, veri doğruluğunu ve tutarlılığını sağlamak için gerekli kuralları tanımlayabilmek için de bir alt yapı sunar. (Nizam, 2011, s. 51)

#### **4.5.1. İlişkisel Veritabanının Özellikleri**

- Veri iki boyutlu satır ve sütunlardan oluşan mantıksal ilişkilerde saklanır.
- Kayıtların (yani satırların) sırası önemsizdir.
- Tüm kayıtlar tektir. Satır ve sütunun kesiştiği yerlerdeki değerler de tekil olmak zorundadır.
- Sütun isimlendirmesi önemlidir.
- Sütunlara kullanıldıkları alanlara göre mantıksal isimler verilir. Personel için adı, soyadı ve doğum yeri gibi. (Nizam, 2011, s. 54)

#### **4.5.2. Anahtar Alanlar ve İlişkiler**

Gerçek yaşamda veriler arasında kurulan ilişkilerin veri tabanına yansıtılması için anahtar alanlar ve ilişki yapıları kullanılmaktadır. (Nizam, 2011, s. 61)

##### **4.5.2.1. Birincil Anahtar**

Birincil anahtar (primary key) bir tablodaki kaydı kesin olarak belirleyen sütun veya sütun grubudur. Birincil anahtar bir tablonun kayıtlarını kesin olarak belirlediğinden değeri her zaman tanımlı olmalıdır, hiçbir zaman boş (NULL) olamaz ve tekrarı olamaz! Veri tabanında bir tablonun belirli bir kaydına ulaşmak birincil anahtar vasıtasıyla olur. Birincil anahtarın tekliği ve boş olmaması ekleme, değişiklik ve silme gibi tüm işlemlerde korunmalıdır. (Nizam, 2011, s. 61)

##### **4.5.2.2. Yabancı Anahtar**

Yabancı anahtar (foreign key) değerleri diğer bir tablonun birincil anahtarı ile bağlantılı tek ya da birleşik sütunlardır. İki tablo arasındaki ilişki, bir tablodaki birincil

anahtar ile diđer tabloda buna referans eden yabancı anahtar kullanılarak gerçekleştirilir. Yabancı anahtarın veri türü, veri uzunluğu ilişkili oldukları birincil anahtarla aynı olmak zorundadır. (Nizam, 2011, s. 61)

#### **4.5.3. İlişkisel Bütünlük Kuralı**

İlişkisel bütünlüğün sağlanabilmesi için her yabancı anahtar değeri ana tablo birincil anahtar değerinden birisi olmak zorundadır. Yabancı anahtarlar ana tablo birincil anahtar değeri haricinde boş değer alabilir. Eşleşmeyen hiçbir kayıt detay tabloda olamaz. Yabancı anahtar tanımlandığında ana tablo silinirken dahi veri tabanı tarafından tutarlılık kontrolü yapılır. Böylece birbirleri ile ilişkili varlıklar arasındaki bilgi tutarlılığı sağlanır. (Nizam, 2011, s. 64)

#### **4.5.4. Normalizasyon**

Normalizasyon, fazlalık veriyi ortadan kaldırmak için, tabloların sadece doğrudan ilişkili verileri içerecek daha küçük tablolara bölünmesi işlemidir. Bu işlem sonucunda her nitelik kendisine en uygun tabloya eklenmiş olur ve bir tabloda sadece direkt ilişkili nitelikler bulunur. Gereksiz tekrar eden veri olmayan, bilgi bütünlüğü ve tutarlılığı korunarak değiştirilebilen bir ilişkisel model elde edilir. Neticede karmaşık iş veri yapılarını daha basit ve kararlı veri yapılarına dönüştürmek mümkün olur. (Nizam, 2011, s. 137)

## 5. GELİŞTİRİLEN MODELLER

Bu bölümde; sınav çizelgeleme, sınav yeri ve gözetmen atama problemlerinin çözümü için geliştirilen modeller açıklanacaktır. Yapılan tüm modellerde değişkenlerin kullanımına yönelik bölüm 4.4'te belirtilen tablo indis özelliğinden faydalanılmıştır.

### 5.1. Sınav Çizelgeleme için Geliştirilen Model

Modelde hem 0-1, hem de sıfırdan büyük ve eşit değer alabilen tam sayılı karar değişkenleri kullanıldığından bu tür modeller **karma tam sayılı model** olarak tanımlanmaktadır.

#### Tanımlar

#### Parametreler

**M** : Sıfırdan oldukça büyük bir sayı

**T** : Sınav çizelgelenecek yapılacak oturum sayısı

**Tcp** : Oturum Kapasitesi

**t\_ogr\_sınav**: Her bir öğrencinin aldığı sınavlar ile bu sınavlara ait sınav grupları ve sınıf verilerini de içeren tablodur.

```
tuple t_ogr_sınav {  
    int sınav_grup_id :Birlikte yapılması istenen sınavlar için oluşturulan numaralar  
    int sınav_id      :Sınav numaraları.  
    int ogr_id       :Öğrenci Numaraları.  
    int sınıf        :Sınavı planlanacak sınıflar.  
    int sınav_suresi :Sınav Süreleri.};
```

**OS** : t\_ogr\_sınav tablosunu temsil eden değişkendir.

**O** : Sınavı planlanacak öğrencilerin numaralarını temsil eden değişkendir.

**C** : Sınavı planlanacak sınıfları temsil eder.

$i \in OS$  için:

$i.sınav\_grup\_id$  : OS değişkenindeki  $i$  numaralı sınav grubunu,  
 $i.sınav\_id$  : OS değişkenindeki  $i$  numaralı sınavı,  
 $i.ogr\_id$  : OS değişkenindeki  $i$  numaralı öğrenciyi,  
 $i.sinif$  : OS değişkenindeki  $i$  numaralı sınıfı,  
 $i.sınav\_suresi$  : OS değişkenindeki  $i$  numaralı sınavın süresini ifade etmektedir.

**t\_sınav** : Sınav bilgilerini içeren tablodur.

```
tuple t_sınav {  
  key int sınav_id;  
  int sınıf;  
  int sınav_suresi;};
```

**S** : t\_sınav tablosunu temsil eden değişkendir.

$s \in S$  için:

$s.sınav\_id$  : S değişkenindeki  $s$  numaralı sınavı,  
 $s.sinif$  : S değişkenindeki  $s$  numaralı sınıfı,  
 $s.sınav\_suresi$  : S değişkenindeki  $s$  numaralı sınavın süresini ifade etmektedir.

```
tuple t_istek {  
  int sınav_id;  
  string tarih;  
  int deger;  
  int gun_id; };
```

**IS** : t\_istek tablosunu temsil eden değişkendir.

$k \in IS$  için:

$k.sınav\_id$  : IS değişkenindeki  $k$  numaralı sınavı,

$k.tarih$  : IS değişkenindeki  $k$  numaralı tarihi,

$k.deger$  : IS değişkenindeki  $k$  numaralı istek değerini,

$k.gun\_id$  : IS değişkenindeki  $k$  numaralı günü ifade etmektedir.

### **Karar Değişkenleri**

$x_{ij}$  : 1, eğer  $k$ . Öğrenci  $s$ . Sınavı için  $j$ . Oturuma atanıyorsa.

$y_{sj}$  : 1, eğer  $s$ . Sınav  $j$ . Oturuma atanıyorsa.

$d^+_j$  :  $j$ . oturumdaki maksimum sınav süresinin minimum süreden pozitif yöndeki sapma miktarı.

$d^-_j$  :  $j$  oturumdaki maksimum sınav süresinin minimum süreden negatif yöndeki sapma miktarı.

### **Amaç Fonksiyonu**

*Sınavlar öğrenci isteklerini en yüksek düzeyde karşılamak suretiyle en az sayıda oturuma atansın.*

$$Z_{\min} = \sum_{s \in S} \sum_{j \in T} (y_{sj} * (j + istek_{sj})) + \sum_{j \in T} d^+_j$$

### **Kısıtlar**

#### **Zorunlu Kısıtlar**

1.Talep:  $k$ . Öğrenci  $s$ . Sınavı için mutlaka bir oturuma atansın.

$$\sum_{j \in T} x_{ij} = 1 \quad \forall i \in OS$$

2.Arz:  $j$ . Oturumdaki arz kadar atama yapılsın.

$$\sum_{i \in OS} x_{ij} \leq T_{cp} \quad \forall j \in T$$

3.k. Öğrenci  $j$ . Oturumda 1'den fazla sınava girmesin.

$$\sum_{\substack{i \in OS : \\ i.ogrid = k}} x_{ij} \leq 1 \quad \forall k \in O, \forall j \in T$$

4.s. Sınav  $j$ . Oturuma atanıyor mu?

$$\sum_{\substack{i \in OS : \\ i.snvid = \\ s.snvid}} x_{ij} \leq M * y_{sj} \quad \forall s \in S, \forall j \in T$$

5.Aynı sınav farklı oturumlarda olmasın.

$$\sum_{j \in B} y_{sj} \leq 1 \quad \forall s \in S$$

6.sg. Grubunda olan sınavlar grup olarak atansın.

$$\sum_{sg \in SG} \sum_{\substack{j \in T : \\ j \neq sg.istgumid}} y_{sj} = 0 \quad \forall sg \in SG : sg.snvgrpid > 0$$

**Esnek Kısıtlar**

7.c. Sınıfın  $j$ . Oturuma ataması olmasın.

$$\sum_{\substack{j \in T: \\ j=10}} y_{sj} = 0 \quad \forall s \in S : s.class = 3$$

8.Son günler mevcudu az olan sınavlar olsun.

$$\sum_{i \in OS} x_{ij} \leq 450 \quad \forall j \in T : j > 7$$

9.c. Sınıfın  $j$ . Oturumdaki maksimum süre mümkün olduğunca 60' olsun.

$$\max_{\substack{s \in S: \\ s.class=c}} (y_{sj}) + d^-_j - d^+_j = 60 \quad \forall j \in T : j \in \{1,3,5\}, \forall c \in C : c = 3$$

10.Belirtilen sınavların istenen oturumlarda olsun.

$$\sum_{\substack{j \in T: \\ j=5}} y_{sj} = 1 \quad \forall s \in S : s.snvid = 1$$

11.Negatif olmama koşulu

$$x_{ij} \in \{0,1\}$$

$$y_{sj} \in \{0,1\}$$

$$d^+_j, d^-_j \geq 0 \text{ ve tam sayı}$$

## 5.2. Sınav Yeri Atama için Geliştirilen Model

### 5.2.1. Kısım Bazlı Atama için Geliştirilen Model

Modelde 0-1 tam sayılı karar değişkenleri kullanıldığından bu tür modeller

**0-1 tam sayılı model** olarak tanımlanmaktadır.



## Tanımlar

### Parametreler

**t\_snav\_kisim** : Bir sınava girecek kısım/şube bazlı verileri içeren tablodur.

**KS** : t\_snav\_kisim tablosunu temsil eden değişkendir.

**tuple** snav\_kisim {

**int** oturum : Sınav Oturumları  
**int** sınıf : Sınavın ait olduğu sınıf  
**int** snav\_id : Sınavlar  
**int** snav\_suresi : Sınavların Süreleri  
**int** kisim\_id : Sınava Giren Kısımlar  
**int** mevcut : Sınava Giren Kısımların Mevcutları};

**t\_snav\_oturum** : Bir sınava girecek kısım/şube bazlı verileri içeren tablodur.

**S** : t\_snav\_oturum tablosunu temsil eden değişkendir.

**tuple** snav\_oturum {

**int** oturum;  
**int** snav\_id; }

**O** : Sınav Oturumları

**t\_snav\_yeri** : Sınavların yapılacağı salonlara ilişkin sin\_yer tuple'nda belirtilen verileri içeren tablodur.

**SY** : t\_snav\_yeri tablosunu temsil eden değişkendir.

**tuple** sin\_yer {

**int** sin\_yer\_id :  
**string** sin\_yer\_adi :

int kap :  
int agirlik :};

**h:** Toplam Sınav Yeri Sayısı

### Karar Değişkenleri

$x_{ij}$  : 1, eğer k. Kısım s. Sınavı için j. Salona atanıyorsa.

$y_{tj}$  : 1, eğer t. Oturumda j. Salona atama oluyorsa.

$z_{sj}$  : 1, eğer s. Sınav j. Salona atanıyorsa.

### Amaç Fonksiyonu

*Her bir oturumda açılacak en az sınav salonu ile her bir sınavın en az salona dağıtılması suretiyle ve sınavların öncelikle istenen salonlara atama yapılması amaçlanmaktadır.*

$$Z_{\min} = \sum_{t \in Otrm} \sum_{j \in SinYer} y_{tj} + \sum_{s \in Snv} \sum_{j \in SinYer} z_{sj} + \sum_{i \in SnvKsm} \sum_{j \in SinYer} x_{ij} * j.agirlik$$

### Kısıtlar

#### Zorunlu Kısıtlar

1. Talep: s. sınavın k. kısmı mutlaka bir salona atansın.

$$\sum_{j \in SinYer} x_{ij} = 1 \quad \forall i \in KS$$

2. Arz:  $t$ . oturumda  $j$ . sınav yerine atama yapılıyor mu? Atama yapılırsa arzı geçmesin.

$$\sum_{\substack{i \in KS: \\ i.otrm=t}} x_{ij} * i.mevcut \leq j.kap. * y_{ij} \quad \forall t \in KO, \forall j \in SY$$

### Esnek Kısıtlar

3.s. sınav  $j$ . sınav yerine atanyor mu?

$$\sum_{\substack{i \in SnvKsm: \\ i.snvid=s.snvid}} x_{ij} \leq h * z_{sj} \quad \forall s \in Snv, \forall j \in SinYer$$

4.s. sınav  $j$ . sınav yerine atama yapılırsa salonda olması istenen maksimum sayıyı geçmesin.

$$\sum_{\substack{s \in Snv: \\ s.oturum=t}} z_{sj} \leq m \quad \forall j \in SinYer, \forall t \in Otrm$$

5. Negatif olmama koşulu

$$x_{ij}, y_{ij}, z_{sj} \in \{0,1\}$$

### 5.2.2. Öğrenci Bazlı Atama için Geliştirilen Model

Modelde hem 0-1, hem de sıfırdan büyük ve eşit değer alabilen tam sayılı karar değişkenleri kullanıldığından bu tür modeller **karma tam sayılı model** olarak tanımlanmaktadır.

### Tanımlar

### Parametreler

**t\_snav\_yeri\_sablon** : Sınavların yapılacağı salonlara ilişkin sin\_yer tuple'nda belirtilen verileri içeren tablodur.

**SYS** : t\_snav\_yeri\_sablon tablosunu temsil eden değişkendir.

```
tuple sinav_yeri {  
  int sinav_yer_id      :  
  int teo_ders_id      :  
  int teo_ders_kap     :};
```

{sinav\_yeri} t\_snav\_yeri\_sablon=...;

t\_snav\_yeri: Sınavların yapılacağı salonları temsil eden dizidir.

**SY**: t\_snav\_yeri dizisini temsil eden değişkendir.

{int} t\_snav\_yeri = { a1 | <a1,a2,a3> in t\_snav\_yeri\_sablon };

t\_snavlar : Bir sınava girecek kısım/şube bazlı verileri içeren tablodur.

**S** : t\_snavlar tablosunu temsil eden değişkendir.

```
tuple sinavlar {  
  int kum_oturum;  
  int sinif;  
  int sinav_id;  
  int sure;  
  int sinav_grp_id;  
  int ogr_mevcut;  
}
```

{sinavlar} t\_snavlar = ...;

kum\_oturum: Sınavların yapılacağı oturumları temsil eden dizidir.

**KO**: kum\_oturum dizisini temsil eden deęişkendir.

{int} kum\_oturum = {a1 | <a1,a2,a3,a4,a5,a6 > in t\_snavlar };

sinav\_grp : Snavların yapılacağı gruplanmış snavların grup numaralarını temsil eden dizidir.

**SG**: sinav\_grp dizisini temsil eden deęişkendir.

{int} sinav\_grp = {a5 | <a1,a2,a3,a4,a5,a6 > in t\_snavlar : a5>0 };

### Karar Deęişkenleri

$x_{ij}$  : i. snavdan j. salona atanan öęrenci sayısı.

$t_{ij}$  : 1, eęer i. snav, j. salona atanıyorsa.

$y_{gh}$  : 1, eęer h. snav yeri, g. oturuma atanıyorsa.

### Amaç Fonksiyonu

*Her bir gün için en az salon ve amfi açılsın, amfi açılırsa buraya yapılacak atama en az olsun.*

$$Z_{\min} = \sum_{g \in KO} \sum_{\substack{h \in S: \\ h \neq 6}} y_{gh} + \sum_{g \in KO} \sum_{\substack{h \in S: \\ h = 6}} y_{gh} * 9999 + \sum_{i \in S} \sum_{\substack{j \in SYS: \\ j.\text{sin\_yer\_id}=6}} t_{ij} * x_{ij} * 1000$$

### Kısıtlar

*1.Snava girecek tüm öęrenciler snav yerlerine atansın.*

$$\sum_{j \in SYS} x_{ij} = i.\text{ogrMevcut} \quad \forall i \in S$$

2. Her bir oturumda aynı salonda aynı sınava girecek öğrencilerin toplamı, salonda atandığı teorik dersin öğrenci kapasitesini aşmasın.

$$\sum_{\substack{i \in S: \\ i.sin\_id=s \& \\ i.kum\_otrm=g}} x_{ij} \leq j.TeoDersKap * t_{sj} \quad \forall s \in S, \forall j \in SYS, \forall g \in KO$$

3. Bir sınav, salondaki sadece bir teorik ders grubuna atansın.

$$\sum_{\substack{j \in SYS: \\ j.sin\_yer\_id=h}} t_{sj} \leq 1 \quad \forall s \in S, \forall h \in SY$$

4. Aynı salondaki teorik ders gruplarında en fazla 2 farklı sınav olsun.

$$\sum_{s \in S} t_{sj} \leq 2 \quad \forall j \in SYS$$

5. Aynı sınav grubunda olan sınavlar, salondaki sadece bir teorik ders grubuna atansın.

$$\sum_{\substack{j \in SYS: \\ j.sin\_yer\_id=h}} \sum_{\substack{s \in S: \\ s.SnvGrp=grp}} t_{sj} \leq 1 \quad \forall grp \in SG, \forall h \in SY$$

6. Her bir oturumda hangi sınav yerleri açılıyor.

$$\sum_{\substack{i \in S: \\ i.kum\_otrm=g}} x_{ij} \leq j.TeoDersKap * y_{gh} \quad \forall h \in SY, \forall j \in SYS : j.sin\_yer\_id = h, \forall g \in KO$$

7. Negatif olmama koşulu

$$x_{ij} \geq 0 \text{ ve tamsayı } \forall i \in S, \forall j \in SYS$$

$$t_{sh} \in \{0,1\} \quad \forall s \in S, \forall h \in SYS,$$

$$y_{gh} \in \{0,1\} \quad \forall g \in KO, \forall h \in SY$$

### 5.3. Gözetmen Atama için Geliştirilen Model

Modelde hem 0-1, hem de sıfırdan büyük ve eşit değer alabilen tam sayılı karar değişkenleri kullanıldığından bu tür modeller **karma tam sayılı model** olarak tanımlanmaktadır.

#### Tanımlar

#### Parametreler

**t\_otrm\_snv\_yerleri** : Her oturumdaki sınav salonlarında yapılacak sınavlar ve bu sınavlara ilişkin sınav süresi ve öğrenci mevcudu verilerini içeren tablodur.

```
tuple otrm_snv_yer {  
    int gun_oturum      : Kümülatif oturumlar  
    int gun_id         : Sınav Günleri  
    int oturum         : Sınav Oturumları  
    int snv_yer_id     : Sınav Yerleri  
    int max_sure       : Sınav Yerlerindeki en büyük sınav süreleri  
    int ogr_mevcudu    : Sınav Yerlerindeki öğrenci sayıları  
    int top_gztnn      : Sınav Yerlerindeki Gözetmen Talebi}
```

**SYO** : t\_otrm\_snv\_yerleri tablosunu temsil eden değişkendir.

**SY** : Sınav yerleri listesi.

**t\_per\_oturum** : Her oturumdaki sınav salonlarında yapılacak sınavlar ve bu sınavlara ilişkin sınav süresi ve öğrenci mevcudu verilerini içeren tablodur.

```
tuple per_oturum {  
    int gun_oturum      : Kümülatif oturumlar  
    int gun_id         : Sınav Günleri  
    int oturum         : Sınav Oturumları
```

**int** per\_id : Personel Numaraları  
**int** rutbe\_id : Personel Ünvan Numaraları  
**int** bolum\_id : Personel Bölüm Numaraları  
**int** mazeret : Personel Mazeretleri}

**PO** : t\_per\_oturum tablosunu temsil eden değişkendir.

**t\_per\_list** : Unvan bilgisi ile birlikte personel listesini içeren tablodur.

**P** : t\_per\_list tablosunu temsil eden değişkendir.

**tuple** per\_tablo {  
**int** per\_id : Personel Numaraları  
**int** rutbe\_id : Personel Ünvan Numaraları}

**KO** : Kümülatif Sınav Oturumları

**G** : Sınav Günleri

**O** : Sınav Oturumları

**ODkGS** : Ortalama Doksan Dakika Görev Sayısı

**OGS** : Ortalama Görev Sayısı

**OBGS** : Oturum Ortalaması

### **Karar Değişkenleri**

$x_{ij}$  : 1, eğer o. Oturumda p. Personel j. Sınav Yerine atanıyorsa.

$d_{1p}^+$  : p. Personelin Kişi Başı Ortalama Görev Sayısından Pozitif Yöndeki Sapma Değişkeni

$d_{1p}^-$  : p. Personelin Kişi Başı Ortalama Görev Sayısından Negatif Yöndeki Sapma Değişkeni



$d_{2p}^+$  : p. Personelin Kişi Başı Ortalama Doksan Dakika Görev Sayısından Pozitif Yöndeki Sapma Değişkeni

$d_{2p}^-$  : p. Personelin Kişi Başı Ortalama Doksan Dakika Görev Sayısından Negatif Yöndeki Sapma Değişkeni

$d_{3p}^+$  : p. Personelin Mazeretine Yazılan Görev Sayısından Pozitif Yöndeki Sapma Değişkeni

$d_{3p}^-$  : p. Personelin Mazeretine Yazılan Görev Sayısından Negatif Yöndeki Sapma Değişkeni

$d_{po}^+$  : p. Personelin o. Oturumda oturum bazında kişi başına düşen görev ortalamasından Pozitif Yöndeki Sapma Değişkeni

$d_{po}^-$  : p. Personelin o. Oturumda oturum bazında kişi başına düşen görev ortalamasından Pozitif Yöndeki Sapma Değişkeni

### **Amaç Alt Fonksiyonları**

*Personele en az sayıda görev yazılması.*

$$Z_1 = \sum_{i \in TP} \sum_{j \in SY} x_{ij}$$

*Kişi başına düşen genel görev ortalamasından sapmalar en az olsun.*

$$Z_2 = \sum_{p \in P} (d_{1p}^- + d_{1p}^+)$$

*Kişi başına düşen doksan dakika görev ortalamasından sapmalar en az olsun.*

$$Z_3 = \sum_{p \in P} (d_{2-p}^- + d_{2-p}^+)$$

*Personelin mazeretine mümkün olduğunca görev yazılmasın.*

$$Z_4 = \sum_{p \in P} d_{3-p}^+$$

*Oturum bazında kişi başına düşen görev ortalamasından sapmalar en az olsun.*

$$Z_5 = \sum_{p \in P} \sum_{t \in O} d_{p0}^-$$

### **Amaç Fonksiyonu**

Personelin mazeretine -mümkün olduğunca- görev yazılmayacak şekilde ve kişi başına düşen genel görev ortalaması, doksan dakika görev ortalaması ile oturum bazındaki görev ortalamasından sapmalar en az olacak şekilde atama yapılsın.

Bu amaç yukarıdaki alt fonksiyonları olarak ayrı yazıp toplamak suretiyle aşağıdaki gibi yazılabileceği gibi;

$$Z_{\min} = \sum_{z \in \{1,2,3,4,5\}} Z_z$$

Bütün olarak da ifade edilebilecektir.

$$Z_{\min} = \sum_{i \in TP} \sum_{j \in SY} x_{ij} + \sum_{p \in P} \left( \sum_{t \in O} d_{p0}^- + d_{1-p}^- + d_{1-p}^+ + d_{2-p}^- + d_{2-p}^+ + d_{3-p}^+ \right)$$

## Kısıtlar

### Zorunlu Kısıtlar

1. Her oturum için sınav yerlerindeki talebi karşılansın

$$\sum_{i \in PO:} x_{ij} = st.topgztnn \quad \forall st \in SYO, \forall j \in SY : j = st.snvyerid$$

$i.gunotrm = st.gunotrm$

2. Sabah ya da öğleden sonra olacak şekilde bir günde en fazla iki görev gelmesi

$$\sum_{i \in PO:}^{2r} \sum_{j \in SnyYer} x_{ij} \leq 1 \quad \forall p \in P, \forall g \in G, \forall r \in \{1,2\}$$

$i.perid = p$  &  
 $i.gunid = g$  &  
 $i.oturum = 2r-1$

3. Personelin mazeretine mümkün olduğunca görev yazma

$$\sum_{i \in PO:} \sum_{j \in SY} x_{ij} = 0 \quad \forall p \in P$$

$i.perid = p.perid$  &  
 $i.mazeret > 0$

4. Personele kişi başına düşen ortalama görev sayısı kadar görev yazılsın

$$\sum_{i \in PO:} \sum_{j \in SY} x_{ij} + d_{1-p}^- - d_{1-p}^+ = OGS \quad \forall p \in P$$

$i.perid = p.perid$

5. Herkese üst sınır (ortalamadan bir/iki fazla görev)

$$\sum_{i \in PO:} \sum_{j \in SY} x_{ij} \leq OGS + 2 \quad \forall p \in P$$

$i.perid = p.perid$

**İstenen unvanlara görev üst sınırı**

6.1. - Unvanı 1 ve 2 olanlara en fazla 5 görev yaz

$$\sum_{\substack{i \in PO: \\ i.perid=p.perid}} \sum_{j \in SY} x_{ij} \leq 5 \quad \forall p \in P: p.unvanid = \{1,2\}$$

6.2. - Unvanı 3 ve 4 olanlara en fazla 6 görev yaz

$$\sum_{\substack{i \in PO: \\ i.perid=p.perid}} \sum_{j \in SY} x_{ij} \leq 6 \quad \forall p \in P: p.unvanid = \{3,4\}$$

7. Kişi başına düşen doksan dakika görev sayısının düzgün dağıtılması

$$\sum_{\substack{i \in PO: \\ i.perid=p.perid}} \sum_{j \in SY} x_{ij} \leq \text{ODkGS} \quad \forall p \in P$$

9. Gözetmenin istenen bir gündeki istenen salona atanması

$$x_{ij} = 1, \quad \forall i.perid = 1 \& i.GunOtrm = 11, \quad \forall j \in SnvYer: j = 1$$

**Salonlardaki istenen personel dağılımları**

10.1. - Yakın ünvanlar aynı salonda olmasın

$$\sum_{\substack{i \in PO: \\ i.gunotrm=st.gunotrm \& \\ i.rutbeid < 5}} x_{ij} \leq 1 \quad \forall st \in SYO, \forall j \in SY: j = st.snvyerid$$

10.2. - Her salona en az bir tane 5 ve 6 unvanı olan personel

$$\sum_{\substack{i \in PO: \\ i.gunotrm=st.gunotrm \& \\ i.rutbeid = \{5,6\}}} x_{ij} \geq 1 \quad \forall st \in SYO, \forall j \in SY: j = st.snvyerid$$

11.İlgili oturumdaki görevliler ilgisiz kişiler olmasın

$$\sum_{\substack{i \in PO : \\ i.gunotrm=st.gunotrm \ \& \\ i.bolumid \neq 11}} \sum_{j \in SY} x_{ij} \geq 1 \quad \forall st \in SYO : st.gunotrm = \{63,64\} \ \& \ st.SnyYerGrpid = 7,$$

$$\forall j \in SY : j = st.snyyerid$$

12.Negatif olmama koşulu

$$x_{ij} \in \{0,1\}$$

$$d_{1p}^+, d_{1p}^-, d_{2p}^+, d_{2p}^-, d_{3p}^+, d_{3p}^-, d_{po}^+, d_{po}^- \geq 0 \text{ \textit{vetam sayı}}$$

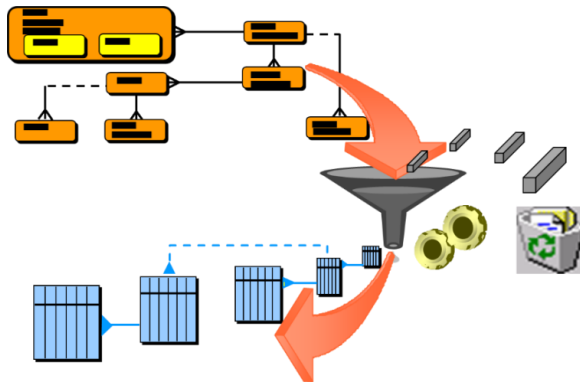
## 6. UYGULAMA

İnsanlık tarihinde bilim, sanayi ve teknolojinin gelişmesine bağlı olarak arz ve talep dengesi içerisinde üretim ve hizmet sektörleri ile günlük insan ihtiyaçları artmıştır. Artan bu ihtiyaçların sistematik olarak kaydedilmesi ve bu verilere hızlı bir şekilde ulaşılması ya da söz konusu verilerden hızlı ve doğru bilginin üretilmesi oldukça önemli bir gereksinim olarak karşımızda durmaktadır. Bu çerçevede bilim dünyası iş çevreleri ile birlikte sürekli olarak veri tabanı programları ve çeşitli uygulamalar geliştirmektedirler.

Sınav optimizasyonu olarak adlandırdığımız projede kullanılan Karar Destek Programı (KDP) ve optimizasyon programının kullanımı bu bölümde anlatılacaktır.

### 6.1. Veritabanı Şema Tasarımı

Veritabanı sistemleri Şekil 6.1.de gösterildiği gibi veri kullanımının optimize edildiği programlardır. Bu programlar sistemlerin gereksinim duyduğu verilerin saklandığı, birbiri ile ilişkili tablolar sayesinde bu verilere kolaylıkla erişim sağlanabildiği ve sistemde herhangi bir güncelleme gerektiğinde yalnızca ilgili tablolarda ve bir defaya mahsus olmak üzere değişiklik yapıldığında sonuçları tüm sisteme yansıtılabilen son derece gerekli ve kullanışlı programlardır.



Şekil 6.1. Veri Kullanımı Optimizasyonu

Veri tabanı programı kullanımında bölüm 4.5.4'te belirtilen normalizasyon kavramı veri tekrarının önüne geçilmesi bakımından oldukça önemlidir. Bu hususun gerçekleştirilmesi için program tasarımı öncesinde iyi bir ihtiyaç ve sistem analizi yapılarak öncelikle veri tabanı şema tasarımı oluşturulması gerekmektedir.

Veri tabanı şema tasarımı; sistemde hangi tablo ve alanların olacağı, birincil ve yabancı anahtar ile tablo ilişki ve izinlerinin belirlendiği yani gerçekte veri tabanı oluşturulurken yapılan tüm işlemlerin yapılabildiği planlama ve tasarım evresidir.

Veritabanı şema tasarımı tablolar oluşturulmadan önce birçok profesyonel yazılımda (Oracle, Mysql vb.) yapılabilmektedir. Ancak; maalesef bu husus MS Access programında desteklenmemektedir. Diğer taraftan, yazılım sırasında oluşturulan tablo ve ilişkilerle ilgili şema tüm programlarda sağlanmaktadır. Bu uygulama için geliştirilen veri tabanı şeması (Bkz. EK 5) sunulmuştur.

## **6.2. Tablo ve Sorguların Oluşturulması**

Veri tabanı şema tasarımı tamamlandıktan sonra doğrulama ve geçirme yapılarak yazılım safhasına geçilmektedir. Yazılım için öncelikle tabloların oluşturulması gerekmektedir. Aslında yukarıda belirtilen şema hazırlanabilen programlarda tablolar, tasarlanan şemaya göre otomatik olarak oluşturulabilmektedir. Yazılım esnasında planlama evresinde gözden kaçan hususlar ilave edilebilmektedir.

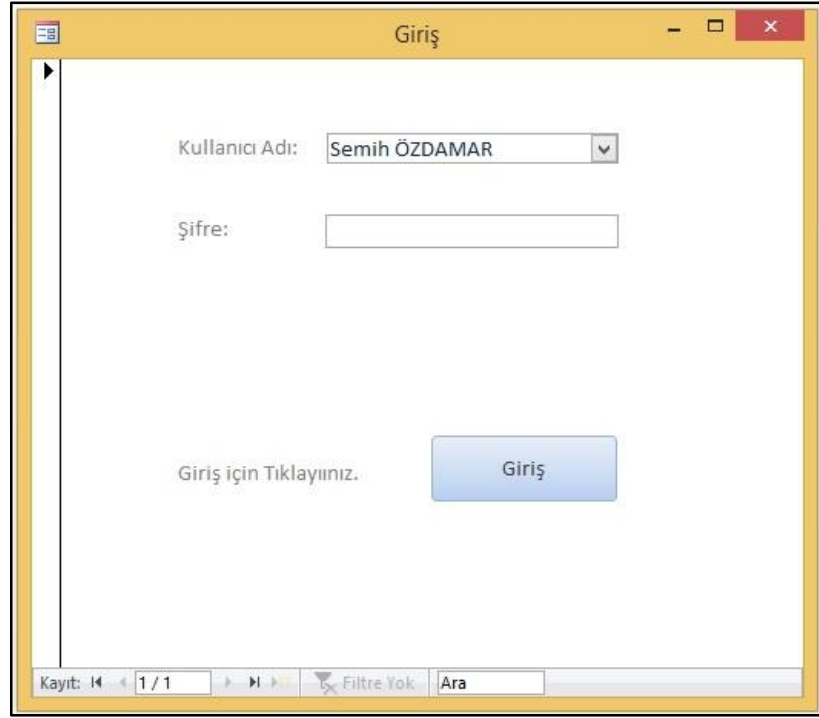
Tabloların oluşturulması sonrasında kullanıcı ve optimizasyon modelleri için gerekli tüm sorgular (view) da oluşturulmaktadır. Bu uygulama için oluşturulmuş çok sayıda sorgu bulunmaktadır.

## **6.3. Veritabanı Ara Yüzü**

Bu bölümde, veritabanı programı görselleri ile birlikte anlatılacaktır.

### 6.3.1. Giriş ve Ana Sayfa

Program çalıştırıldıktan sonra Şekil-8'deki giriş formu ile birlikte açılmaktadır. Bu form, yetkisiz kullanıcıların programa erişimine bir güvenlik önlemi ve kullanıcı istatistiğinin tutulması için tasarlanmıştır. Bu ekrandaki açılır kutuda tanımlı kullanıcılar kendi şifrelerini girmek suretiyle giriş butonunu kullanarak Şekil 6.2.deki ana sayfaya geçiş sağlamaktadır.



The image shows a Windows-style window titled "Giriş". Inside the window, there is a form with the following elements:

- A dropdown menu labeled "Kullanıcı Adı:" with the text "Semih ÖZDAMAR" and a downward arrow.
- A text input field labeled "Şifre:".
- A blue button labeled "Giriş".
- The text "Giriş için Tıklayınız." is displayed below the password field.

At the bottom of the window, there is a status bar with the following elements:

- "Kayıt: 1 / 1" with navigation arrows.
- "Filtre Yok" with a funnel icon.
- An "Ara" search box.

**Şekil 6.2.** Giriş Ekranı

Ana sayfa ekranında, sınav uygulaması için gerekli tablo ve formlara erişim sağlanması için çeşitli butonlar bulunmaktadır. Bu ekran üzerinden Bölümler, dersler, öğrenciler, personel bilgileri için tanımlama ve düzenlemeler yapılabilen formlar bulunmaktadır.





Şekil 6.3. Program Ana Sayfası

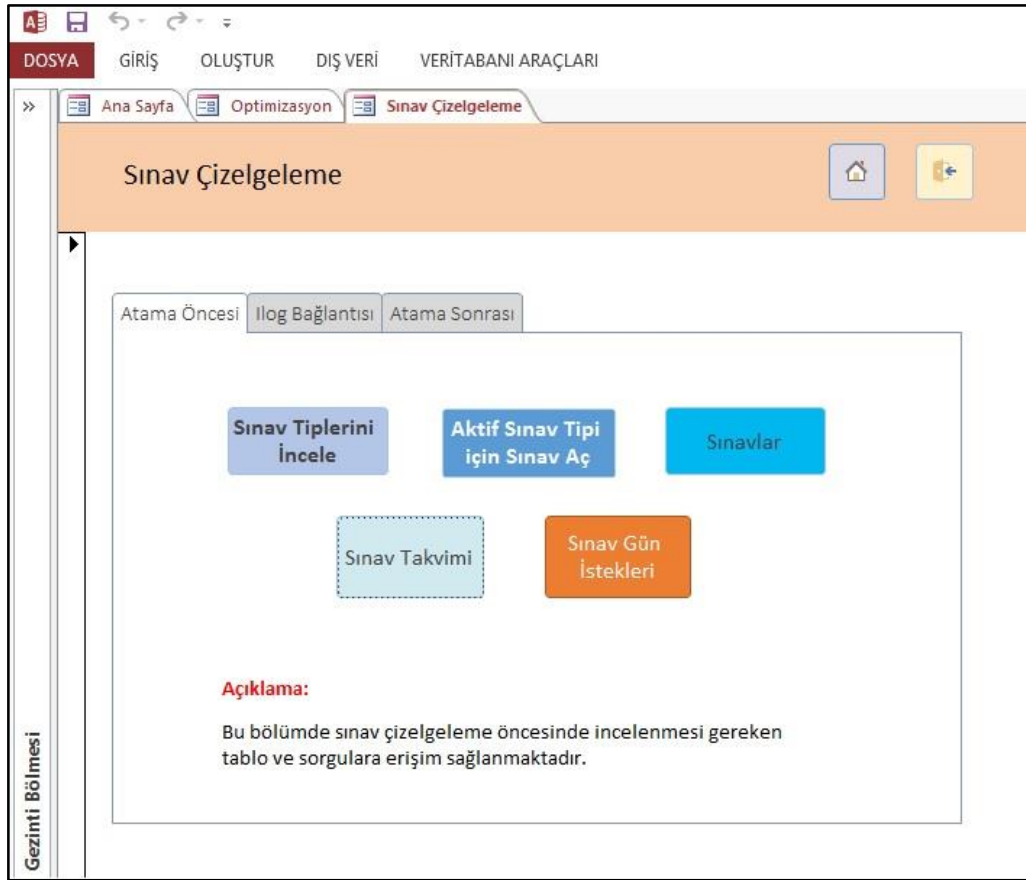
Ana sayfadan “optimizasyon” butonu ile Şekil 6.4.teki sınav sistemi optimizasyonu geçiş formu açılmaktadır.



Şekil 6.4. Optimizasyon Geçiş Ekranı

### 6.3.2. Sınav Çizelgeleme Modülü

Optimizasyon geçiş ekranında bulunan ‘‘Sınav Çizelgeleme’’ butonu kullanılarak sınav çizelgeleme ana ekranı açılmaktadır. Bu ekranda atama öncesi ve sonrasında ulaşılacak tablo ve formlar bulunmaktadır. Şimdi sırasıyla sınav çizelgeleme yapılabilmesi için gerekli işlemler anlatılacaktır.



Şekil 6.5. Sınav Çizelgeleme Ana Ekranı

Sınav çizelgelemesi öncesinde hangi sınav tipi ilgili çalışma yapılacağı Şekil 6.6.daki ekran üzerinden belirlenmelidir.

Belirlenen sınav tipi için eğitim verilen tüm derslere Şekil 6.7.de gösterilen ‘‘aktif sınav tipi için sınav aç’’ butonu ile sınavlar açılmalıdır.

eo_yil_id	eo_yili_uzun	donem_ic	donem	donem_snav_tip_id	snav_tip_id	engel_sira	snav_tipi	agirlik	aktif mi
14	2014-2015	40	1.Dönem	1	1		Ara	0,32	1
14	2014-2015	40	1.Dönem	2	2		Final	0,6	0
14	2014-2015	40	1.Dönem	3	3		Engel		0
14	2014-2015	40	1.Dönem	4	4		Bütünleme	1	0
14	2014-2015	40	1.Dönem	5	5		No Yükseltme	1	0
14	2014-2015	40	1.Dönem	6	6		Tek Ders	1	0

Şekil 6.6. Sınav Tipi Seçme Ekranı



Şekil 6.7. Sınav Açma Butonu

Sınavların yapılacağı tarihler sınav takvimine Şekil 6.8.deki ekran üzerinden girilmelidir.

yıl	donem	sta_id	tarih	sinav_tipi	gun_id	otrm_1	otrm_2	otrm_3	otrm_4
2014-2015	1.Dönem	1	26.1.2015	Ara	1	3	0	0	0
2014-2015	1.Dönem	2	27.1.2015	Ara	2	3	0	0	0
2014-2015	1.Dönem	3	28.1.2015	Ara	3	3	0	0	0
2014-2015	1.Dönem	4	29.1.2015	Ara	4	3	0	0	0
2014-2015	1.Dönem	5	30.1.2015	Ara	5	3	0	0	0
2014-2015	1.Dönem	6	2.2.2015	Ara	6	3	0	0	0
2014-2015	1.Dönem	7	3.2.2015	Ara	7	3	0	0	0
2014-2015	1.Dönem	8	4.2.2015	Ara	8	3	0	0	0
2014-2015	1.Dönem	9	5.2.2015	Ara	9	3	0	0	0
2014-2015	1.Dönem	10	6.2.2015	Ara	10	3	0	0	0

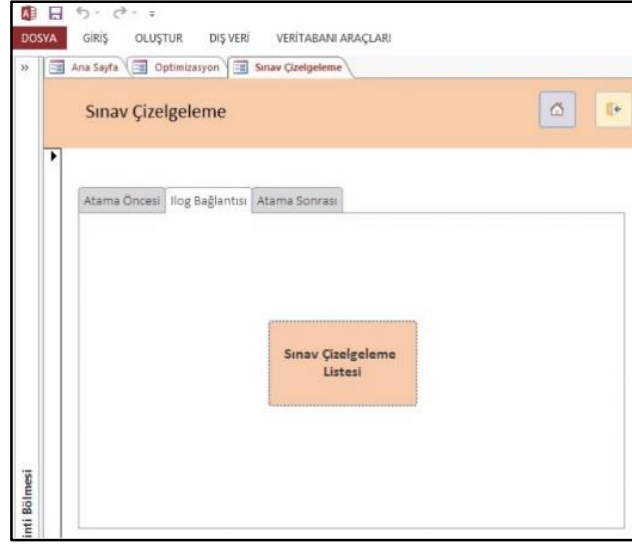
Şekil 6.8. Sınav Takvimi Belirleme Formu

Belirlenen sınav takvimi için öğretmen ve öğrencilerden önceden alınan talepler Şekil-6.9.daki tabloya aktarılır.

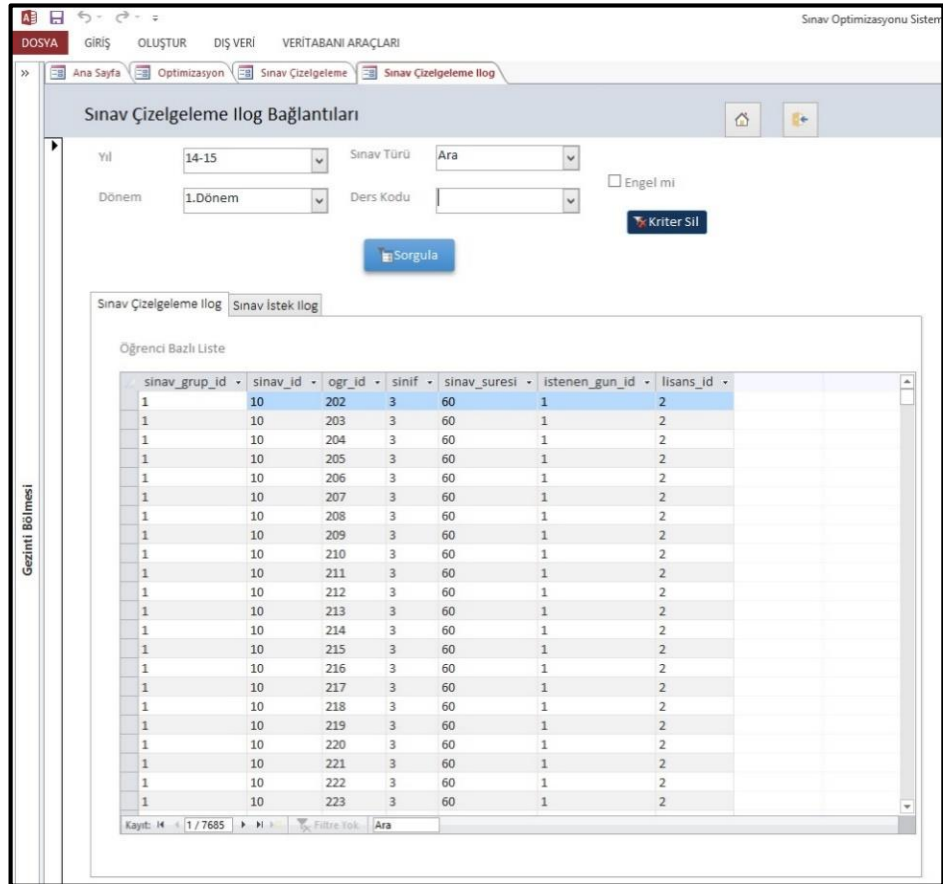
sinav_id	sta_id	ogretim_ele_istek	ogrenci_istek
1	1	3	3
1	2	1	3
1	3	3	3
1	4	3	2
1	5	0	2
1	6	2	0
1	7	2	3
1	8	3	3
1	9	3	1
1	10	3	3
2	1	2	3
2	2	2	3
2	3	3	1
2	4	3	1
2	5	1	0
2	6	0	1
2	7	1	3
2	8	1	3
2	9	3	2
2	10	3	2
3	1	3	3

Şekil 6.9. Öğrenci ve Öğretmenlerin Sınav Gün İstekleri Kayıt Tablosu

Şimdiye kadar belirtilen işlemlerin ardından ilog programında kullanılacak sorguların oluşturulması gerekmektedir. Bu konudaki hazırlanmış sorgulara Şekil 6.10.daki ekran üzerinden ulaşılabilecektir.



Şekil 6.10. Ilog Sorgularını Açma Ekranı



Şekil 6.11. Ilog'da Kullanılan Öğrenci Sınav Listesi

Şekil 6.11. “Sınav Çizelgeleme Ilog” sekmesinde sınav açılan derslerin öğrenci bazlı listesi görülmektedir. Bu tabloda herhangi bir sorun olup olmadığı kontrol edilir.

Aynı sayfadaki Şekil 6.12. “sınav istek ilog” sekmesi üzerinden öğrenci ve öğretmenlerden alınan isteklerin sonucu elde edilir. Bu sayede sınavların sınav takvimindeki her bir gün için bir istek değeri elde edilmiş olur.

Sınav Optimizasyonu Sistem

DOSYA GİRİŞ OLUŞTUR DIŞ VERİ VERİTABANI ARAÇLARI

Ana Sayfa Optimizasyon Sınav Çizelgeleme Sınav Çizelgeleme Ilog

Sınav Çizelgeleme Ilog Bağlantıları

Yıl: 14-15 Sınav Türü: Ara

Dönem: 1.Dönem Ders Kodu:

Engel mi

Kriter Sil

Sorgula

Sınav Çizelgeleme Ilog Sınav İstek Ilog

Sınav gün İstek listesi

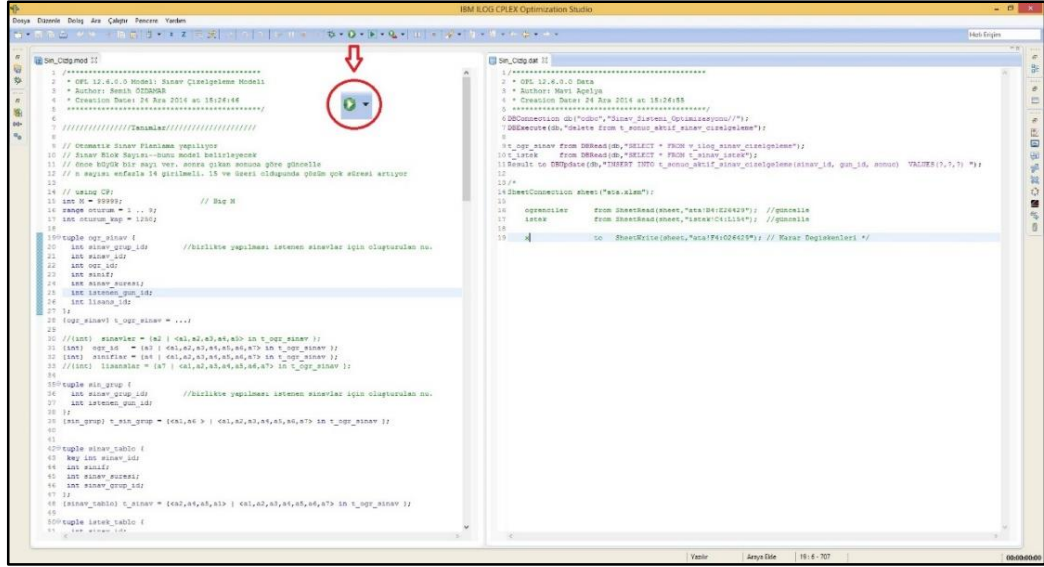
Sınav İstek Ilog

sinav_id	sta_id	deger
1	1	999
1	2	1
1	3	999
1	4	999
1	5	0
1	6	2
1	7	2
1	8	999
1	9	999
1	10	999
2	1	2
2	2	2
2	3	999
2	4	999
2	5	1

Kayıt: 1 / 900 Filtre Yok Ara

Şekil 6.12. Ilog’da Kullanılan Sınav Günü İstek Listesi

IBM Ilog Programı kullanılarak geliştirilen model (Bkz. EK 1) Şekil 6.13.deki ekranda gösterildiği şekilde çalıştırılarak sonuçlar elde edilmektedir.



Şekil 6.13. Sınav Çizelgeleme Modeli Ilog Ekranı

### 6.3.3. Sınav Yeri Atama Modülü

#### 6.3.3.1. Kısım Bazlı Atama Modülü



Şekil 6.14. Kısım Bazlı Sınav Yeri Atama Modülü Ana Ekranı

Optimizasyon geiř ekranında bulunan ‘‘Sınav Yeri Atama-Kısım Bazlı’’ butonu kullanılarak Őekil 6.14.de gsterilen Sınav Yeri Atama ana ekranı aılmaktadır. Bu ekranda atama ncesi ve sonrasında ulařılabilecek tablo ve formlar bulunmaktadır. Őimdi sırasıyla sınav yeri atama yapılabilmesi iin gerekli iřlemler anlatılacaktır.

Kısım Bazlı Atama ncesinde hangi sınav salonlarının kullanılacağı Őekil 6.15.de gsterilen ‘‘sınav yerleri’’ tablosu zerinden belirlenmelidir.

sinav_yer_id	sinav_yer_adi	sinav_yer_adi_kisa	sinav_kapasitesi	kisim_id	normal_kapasite	maliyeti	sinav_yer_grup_id	aciklama
1	S_1		110		0	50	1	110
2	S_2		100		0	50	1	100
3	S_3		180		0	0	1	180
4	S_4		190		0	0	2	190
5	S_5		200		0	0	2	200
6	S_6		60		0	300	2	40
7	S_7		40		0	400	2	40
8	S_8		40		0	500	2	30
9	S_9		80		0	600	3	30
10	S_10		60		0	700	3	30
11	S_11		80		0	800	3	20
12	S_12		80		0	900	3	20
13			23	1	23	0	4	
14			35	2	35	0	4	
15			30	3	30	0	4	
16			30	4	30	0	4	
17			24	5	24	0	4	
18			32	6	32	0	4	
19			29	7	29	0	4	

**Őekil 6.15.** Sınav Yerleri

Sınav yerleri belirlenmesinden sonra, ğrencilerin birlikte eđitim grdđ derslikteki ğrencileri temsil eden kısım bazlı sınav listesi Őekil 6.16.daki gibi hazırlanmalıdır.

Sınavların yapılacağı salonlar Őekil 6.17.deki gibi ilog programında kullanıma hazır hale getirilir.



Sınav Optimizasyonu Sistemi (Yüksek Lisans Tez)

DOSYA GİRİŞ OLUŞTUR DIŞ VERİ VERİTABANI ARAÇLARI

Ana Sayfa Optimizasyon Sınav Yeri Çizelgeleme (Ks.Bazlı) Sınav Girecek Kısımlar

Sınav Girecek Kısımlar

Yıl: 14-15 Sınav Türü:

Dönem: 1.Dönem  Engel mi

Tüm Liste

f sinava\_girecek\_kisimler:

donem_id	sinif	gun_id	sinav_id	donem_snr	sinav_tipi	sinav_sı	sinav_gru	secmeli_mi	ders	ders_kodu	ders_ad	kategori_id	mekan	ogr_mevcut
40	3	5	1	1	Ara	90			1	BM3121		1	1	35
40	3	5	1	1	Ara	90			1	BM3121		1	2	35
40	3	5	1	1	Ara	90			1	BM3121		1	3	35
40	3	6	2	1	Ara	90			2	BM3122		1	1	35
40	3	6	2	1	Ara	90			2	BM3122		1	2	35
40	3	6	2	1	Ara	90			2	BM3122		1	3	35
40	3	2	3	1	Ara	90			3	BM3123		1	1	35
40	3	2	3	1	Ara	90			3	BM3123		1	2	35
40	3	2	3	1	Ara	90			3	BM3123		1	3	35
40	3	4	4	1	Ara	60			4	BM3124		1	1	35
40	3	4	4	1	Ara	60			4	BM3124		1	2	35
40	3	4	4	1	Ara	60			4	BM3124		1	3	35
40	3	3	5	1	Ara	60			5	BM3125		1	1	35
40	3	3	5	1	Ara	60			5	BM3125		1	2	35

Gezinti Bölmesi

Şekil 6.16. Kısım Bazında Sınav Listesi

Sınav Optimizasyonu Sistemi (Yüksek Lisans Tez)

DOSYA GİRİŞ OLUŞTUR DIŞ VERİ VERİTABANI ARAÇLARI FORM ARAÇLARI VERİ SAYFASI

Ana Sayfa Optimizasyon Sınav Yeri Çizelgeleme (Ks.Bazlı) Sınav Yerleri

Sınav Yerleri

Yıl:  Sınav Türü:

Dönem:   Engel mi

Tüm Liste

f ilog\_sinav\_yerleri:

sinav_yer_id	sinav_yer_adi	sinav_kapasitesi	maliyeti
1	S_1	110	50
2	S_2	100	50
3	S_3	180	0
4	S_4	190	0
5	S_5	200	0
6	S_6	60	300
7	S_7	40	400
8	S_8	40	500
9	S_9	80	600
10	S_10	60	700
11	S_11	80	800
12	S_12	80	900
* Yeni		0	0

Gezinti Bölmesi

Şekil 6.17. Ilog'da Kullanılacak Sınav Yeri Listesi

Kısım bazında düzenlenen sınav listesi Şekil 6.18.de olduğu gibi listelenir.

gun_id	sinif	sinav_id	sinav_suresi	kisim_id	ogr_mevcut
1	3	8	60	2	29
1	3	10	60	4	28
1	3	10	60	5	27
1	3	25	90	1	28
1	3	25	90	2	28
1	3	25	90	3	28
1	3	25	90	4	25
1	3	28	60	12	29
1	3	28	60	13	29
1	3	28	60	14	27
1	3	39	90	15	25
1	3	39	90	16	25
1	3	39	90	17	25

Şekil 6.18. Ilog'da Kullanılacak Kısım Bazında Sınav Listesi

IBM Ilog Programı kullanılarak geliştirilen model (Bkz. EK 2) Şekil 6.19.daki ekranda gösterildiği şekilde çalıştırılarak sonuçlar elde edilmektedir.

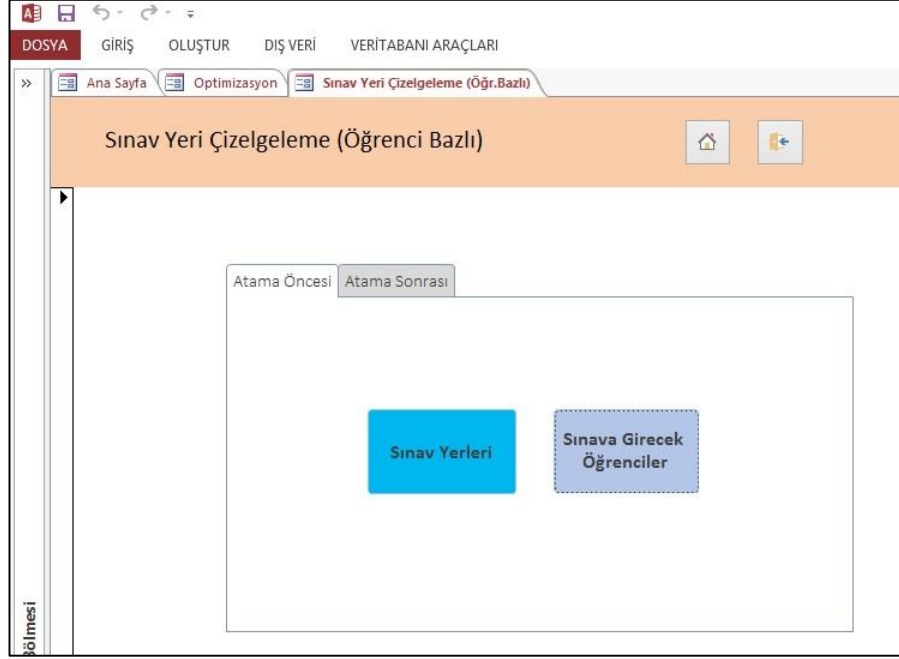
```

1 //*****
2 * OPL 12.6.0.0 Model
3 * Kısım Bazlı Sınav Yeri Çızelgeme (Sınav Salonlarına Öğrenci Atama) Modeli
4 * Author: Semih ÖZDAMAR
5 * Creation Date: 16 Tem 2014 at 23:39:05
6 *****
7 //*****TAMIMLAR*****
8
9 //using CP;
10 tuple sinav_kisim {
11 int oturum;
12 int sinif;
13 int sinav_id;
14 int sinav_suresi;
15 int kisim_id;
16 int mevcut;
17 };
18 {sinav_kisim} t_sinav_kisim = ...;
19
20 tuple t_sinav_oturum {
21 int oturum;
22 int sinav_id;
23 };
24
25 {int} oturumlar = {el | < el,a2,a3,a4,a5,a6 > in t_sinav_kisim};
26 {t_sinav_oturum}
27 sinavlar = {< el,a3 > | < el,a2,a3,a4,a5,a6 > in t_sinav_kisim};
28
29 tuple sinav_yeri {
30 int sinav_yer_id;
31 string sin_yer_adi;
32 int kap;
33 int spizlilik;
34 };
35 {sinav_yeri} t_sinav_yeri = ...;
36
37 //toplam sinav_yeri sayisi
38 int n = card(t_sinav_yeri);
39

```

Şekil 6.19. Kısım Bazlı Sınav Yeri Atama Ilog Modeli Ekran Görüntüsü

### 6.3.3.2. Öğrenci Bazlı Atama Modülü



Şekil 6.20. Öğrenci Bazlı Sınav Yeri Atama Modülü Ana Ekranı

İlog Sınav Yerleri Öğrenci Bazlı Model için

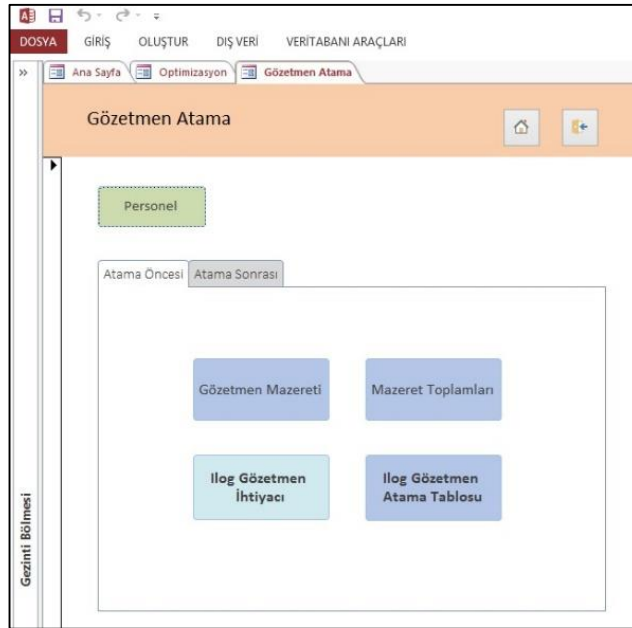
sınav_yer_id	sınav_yer_adi	sınav_kapasitesi	maliyeti
13		23	0
14		35	0
15		30	0
16		30	0
17		24	0
18		32	0
19		29	0
20		28	0
21		27	0
22		32	0
23		27	0
24		28	0
25		38	0
26		28	0

Şekil 6.21. Sınav Yeri Listesi

kum_oturum	sinif	sinav_id	sinav_suresi	sinav_grup_id	mevcut
11	3	8	60	1	29
11	3	10	60	1	55
11	3	25	90		109
11	3	28	60		85
11	3	39	90		75
11	3	53	90		20
11	3	55	90		180
11	3	69	90		90
11	3	79	90		10
11	3	85	90		190
21	3	3	90		105
21	3	11	90		55
21	3	21	60		195
21	3	33	60		85
21	3	42	60		75
21	3	49	90		75
21	3	60	90		180
21	3	71	90		30
21	3	76	90		50
21	3	83	60		190
31	3	5	60		105
31	3	14	90		55
31	3	22	60		195
31	3	30	60		85
31	3	41	90		75
31	3	48	60		75
31	3	58	90		180

Şekil 6.22. Ilog'da Kullanılacak Sınav Listesi

### 6.3.4. Gözetmen Atama Modülü



Şekil 6.23. Gözetmen Atama

Sınav Optimizasyonu Sistemi (Yüksek Lisans Tezi)

PERSONEL

Personel Bilgileri Düzenleme Ekranı

Bölüm:  Personel Num:

Unvan:   Pasif Personel

Tüm Liste |

Personel Listesi

per_id	adi	soyadı	tc_kimlik_no	dahili_tel	cep_tel	Sözetmen Pasif mi	eposta	brans_id	bolum_id	bolum_adi_kisa	fakulte_id	fakulte	ogr_kontenjani	bolum_baskani_mi
1	Süleyman	ERSÖZ	0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	1
2	Adnan	AKTEPE	0	0	0	1		0	1	Blg.Müh.	1	Müh.	105	0
3	Semih	ÖZDAMAR	0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
4	Kadir	ERTAN	0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
5			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
6			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
7			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
8			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
9			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
10			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
11			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
12			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
13			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
14			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0
15			0	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0

Şekil 6.24. Personel Bilgileri Düzenleme Ekranı

Sınav Optimizasyonu Sistemi (Yüksek Lisans Tezi)

FORM ARAÇLARI

PERSONEL

Personel Bilgileri Düzenleme Ekranı

Bölüm:  Personel Num:

Unvan:   Pasif Personel

Tüm Liste |

Gözetmen Olacak Personel Listesi

per_id	adsoyad	tc_kimlik_no	dahili_tel	cep_tel	eposta	brans_id	bolum_id	bolum_adi	fakulte_id	fakulte	ogr_kontenjani	bolum_bask	unvan_id	
3	Semih ÖZDAMAR	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	1	Prc
4	Kadir ERTAN	0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	1	Prc
5		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	1	Prc
6		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	1	Prc
7		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	1	Prc
8		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	1	Prc
9		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	1	Prc
10		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	2	Doç
11		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	2	Doç
12		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	2	Doç
13		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	2	Doç
14		0	0	0		0	1	Blg.Müh.	1	Müh.	105	0	3	Yar

Şekil 6.25. Gözetmen Olacak Personel Listesinin Belirlenmesi

gozetmen_mazeret_id	per_id	sta_id	oturum	sin_tur_id	aciklama
1	87	1	3	1	
2	88	1	3	1	
3	92	1	3	1	
4	93	1	3	1	
5	94	1	3	1	
6	96	1	3	1	
7	100	1	3	1	
8	102	1	3	1	
9	103	1	3	1	
10	105	1	3	1	
11	121	1	3	1	
12	131	1	3	1	
13	140	1	3	1	
14	151	1	3	1	
15	9	1	4	1	
16	12	1	4	1	
17	18	1	4	1	
18	22	1	4	1	

Şekil 6.26. Gözetmen Mazeret Kayıt Tablosu

per_id	gun_toplam
103	25
78	23
12	23
100	22
82	22
23	22
42	21
18	21
43	21
97	21
118	21

Şekil 6.27. Personel Bazında Mazeret Toplamları

gun_oturum	gun_id	oturum	sinav_yer_id	max_sure	90_dk_mi	mevcut	hesaplanan_gozetmen_sayisi	SA	ES	KG	G	Toplam
1	1	1	1	90	1	90	2,25	1	1	0	0	2
11	1	1	2	90	1	96	2,4	1	1	0	0	2
11	1	1	3	90	1	180	4,5	1	1	1	1	4
11	1	1	4	90	1	190	4,75	1	1	1	2	5
11	1	1	5	90	1	195	4,875	1	1	1	2	5
11	1	1	6	90	1	60	1,5	1	1	0	0	2
11	1	1	7	90	1	32	0,8	0	1	0	0	1
21	2	1	1	60	0	94	2,35	1	1	0	0	2
21	2	1	2	60	0	97	2,425	1	1	0	0	2
21	2	1	3	90	1	178	4,45	1	1	1	1	4

Şekil 6.28. Gözetmen İhtiyacı Ekranı

gun_oturum	sta_id	gun_id	oturum	per_id	unvan_id	bolum_id	mazereti
11	1	1	1	1	1	1	0
12	1	1	2	1	1	1	0
13	1	1	3	1	1	1	0
14	1	1	4	1	1	1	0
21	2	2	1	1	1	1	0
22	2	2	2	1	1	1	0
23	2	2	3	1	1	1	0
24	2	2	4	1	1	1	0
31	3	3	1	1	1	1	0
32	3	3	2	1	1	1	0
33	3	3	3	1	1	1	0
34	3	3	4	1	1	1	0
41	4	4	1	1	1	1	0
42	4	4	2	1	1	1	0
43	4	4	3	1	1	1	0
44	4	4	4	1	1	1	0

Şekil 6.29. Gözetmen Atama Tablosu

```

1 /*****
2 * OPL 12.6.0.0 Model
3 * Gözetmen Atama Modeli
4 * Author: Semih ÖZDAMAR
5 * Creation Date: 05 Şub 2015 at 13:07:47
6 *****/
7
8 //*****Tanımlar***/
9
10@tuple otrim_snv_yer {
11 int gun_oturum;
12 int gun_id;
13 int oturum;
14 int snv_yer_id;
15 int max_gunes;
16 int doksan_dk_mi;
17 int ogr_mevcutdu;
18 int hes_goz_ays;
19 int sa;
20 int es;
21 int kg;
22 int g;
23 int top_gztnm;
24 }
25
26 (otrm_snv_yer) t_otrm_snv_yerleri = ...;

```

```

1 /*****
2 * OPL 12.6.0.0
3 * Gözetmen Atama Modeli Data Bağlantısı
4 * Author: Semih ÖZDAMAR
5 * Creation Date: 05 Şub 2015 at 13:07:47
6 *****/
7
8 DBConnection db("odbc","Sinev_Sistemi_Optimizasyonu/");
9 DBExecute(db,"delete from t_gozetmen_aktif_sonuc");
10
11 t_per_oturum from DBRead(db,"SELECT * FROM v_ilog_gozetmen_atama where oturum=");
12 t_otrm_snv_yerleri from DBRead(db,"SELECT * FROM v_ilog_gozetmen_istiyaci");
13
14 Result to DBUpdate(db,"INSERT INTO t_gozetmen_aktif_sonuc(sta_id, oturum, per_id, snv_yer_id,

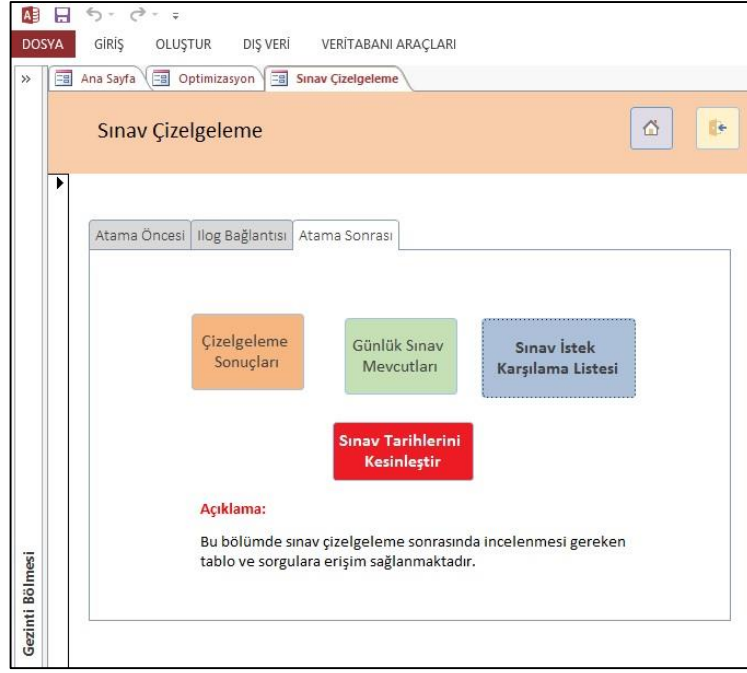
```

Şekil 6.30. Gözetmen Atama Ilog Modeli Ekran Görüntüsü

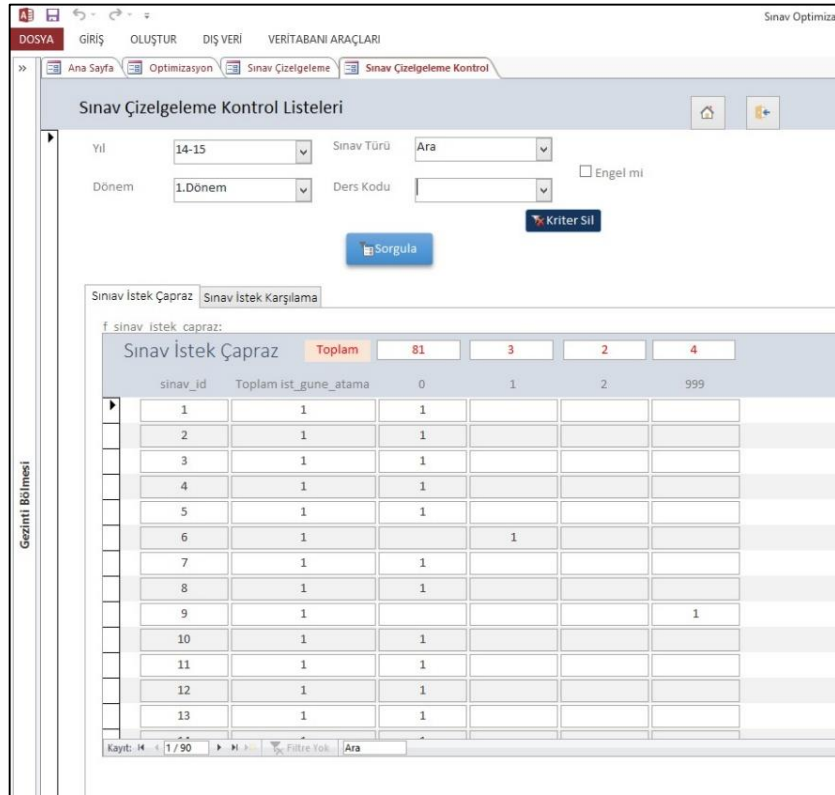
## 6.4. Uygulama Sonuçları

### 6.4.1. Sınav Çizelgeleme Sonuçları

Sınav çizelgeleme sonuç tablosu incelenir ve atama sonuçları ile istekler karşılaştırılarak sınavlar kesinleştirilir.



Şekil 6.31. İlog Atama Sonrası Ara Yüzü



Şekil 6.32. Sınav İstekleri Kontrol



Sınav Optimizasyonu Sistemi (Yüksek Lisans Tezi)

DOSYA GİRİŞ OLUŞTUR DIŞ VERİ VERİTABANI ARAÇLARI

Ana Sayfa Optimizasyon Sınav Çizelgeme Aktif Sınav Çizelgeme Sonuçları

Aktif Sınav Çizelgeme Sonuçları

donem_id	donem	sinif	sinav_id	sin_tip_id	sinav_turu	sinav_suresi	sinav_grup_id	taslak_gun_id	Kesin Gün	oturma	ders_kodu	ders_adi	bolum_ad	Mevcut
40	1.Dönem	3	1	1	Ara	90		5	5	1	BM3121		Blg.Müh.	105
40	1.Dönem	3	2	1	Ara	90		6	6	1	BM3122		Blg.Müh.	105
40	1.Dönem	3	3	1	Ara	90		2	2	1	BM3123		Blg.Müh.	105
40	1.Dönem	3	4	1	Ara	60		4	4	1	BM3124		Blg.Müh.	105
40	1.Dönem	3	5	1	Ara	60		3	3	1	BM3125		Blg.Müh.	105
40	1.Dönem	3	6	1	Ara	90		7	7	1	BM3126		Blg.Müh.	105
40	1.Dönem	3	7	1	Ara	90		8	8	1	BM3151		Blg.Müh.	41
40	1.Dönem	3	8	1	Ara	60	1	1	1	1	BM3152		Blg.Müh.	29
40	1.Dönem	3	9	1	Ara	60		9	7	1	BM3153		Blg.Müh.	35
40	1.Dönem	3	10	1	Ara	60	1	1	1	1	CV3121		Çev.Müh.	55
40	1.Dönem	3	11	1	Ara	90		2	2	1	CV3122		Çev.Müh.	55
40	1.Dönem	3	12	1	Ara	60		4	4	1	CV3123		Çev.Müh.	55
40	1.Dönem	3	13	1	Ara	90		5	5	1	CV3124		Çev.Müh.	55
40	1.Dönem	3	14	1	Ara	90		3	3	1	CV3125		Çev.Müh.	55
40	1.Dönem	3	15	1	Ara	60		7	7	1	CV3126		Çev.Müh.	55
40	1.Dönem	3	16	1	Ara	60		6	6	1	CV3151		Çev.Müh.	15
40	1.Dönem	3	17	1	Ara	60		8	8	1	CV3152		Çev.Müh.	20

Şekil 6.33. Sınav Çizelgeme Sonuç Tablosu

Sınav Optimizasyonu Sistemi (Yüksek Lisans Tezi)

DOSYA GİRİŞ OLUŞTUR DIŞ VERİ VERİTABANI ARAÇLARI

Ana Sayfa Optimizasyon Sınav Çizelgeme f\_sinav\_gunleri\_ogr\_mevcutlari

Sınav Günleri Öğrenci Mevcutları

donem_id	Dönem	donem_sin_tip_id	Sınav Türü	Taslak Gün id	Taslak Tarih	Toplam Mevcut	Max Süre
40	1.Dönem	1	Ara	1	26 Ocak 2015 Pazartesi	898	90
40	1.Dönem	1	Ara	2	27 Ocak 2015 Salı	1040	90
40	1.Dönem	1	Ara	3	28 Ocak 2015 Çarşamba	965	90
40	1.Dönem	1	Ara	4	29 Ocak 2015 Perşembe	1100	90
40	1.Dönem	1	Ara	5	30 Ocak 2015 Cuma	1040	90
40	1.Dönem	1	Ara	6	2 Şubat 2015 Pazartesi	975	90
40	1.Dönem	1	Ara	7	3 Şubat 2015 Salı	830	90
40	1.Dönem	1	Ara	8	4 Şubat 2015 Çarşamba	425	90
40	1.Dönem	1	Ara	9	5 Şubat 2015 Perşembe	412	90

Şekil 6.34. Sınav Günleri Öğrenci Mevcutları



**Çizelge 6.2. Öğrenci Bazlı Sınav Salonu Atama Sonuçları-1**

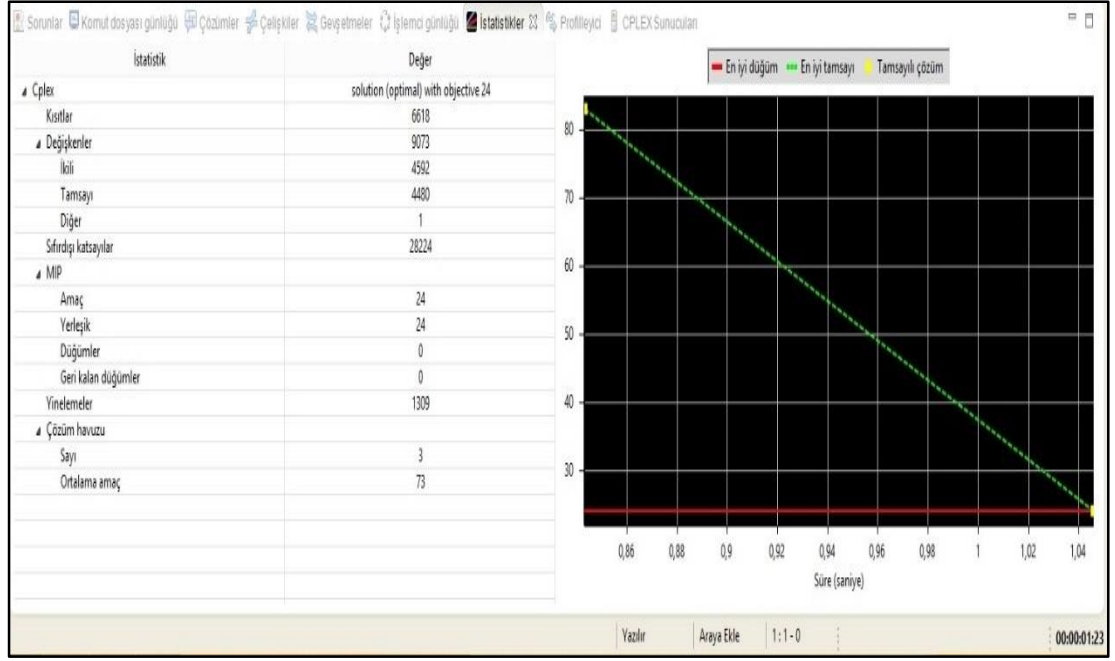
Sınav Yeri	Sınavlar										Toplam
	1	13	19	32	38	47	57	70	78	82	
14	9		9			9	8				35
18					2		1				3
25			10	9	1		9		9		38
38	9		9				8			8	34
42	8	9				9				8	34
50				9		8		8		8	33
54			9	9		8				8	34
61		8					8		8	8	32
62			8	8	8					8	32
65			9				10	10		9	38
67	1		8				9	8		9	35
70			10	10	10					10	40
71		9	8		9		8				34
72	8		8	8			8				32
73		3	8		8	6			8		33
76			8				8		8	8	32
78	9		9				8			9	35
79	9		9				9			9	36
84	8		8		9					8	33
88	9		9			9				9	36
90		9	9	6			9	3			36
92		9				9	9			9	36
94	9		9	9						9	36
96	9		5	8	3		8				33
98					8		9		9	9	35
100			9		8		8			9	34
101			7		9		8	1		8	33
104			9			9	9			9	36
106	8						9		8	9	34
113	9		8				9			8	34
120		8		9		8	8			1	34
<b>Toplam</b>	<b>105</b>	<b>55</b>	<b>195</b>	<b>85</b>	<b>75</b>	<b>75</b>	<b>180</b>	<b>30</b>	<b>50</b>	<b>190</b>	<b>1040</b>

Çizelge 6.3. Öğrenci Bazlı Sınav Salonu Atama Sonuçları-2

Sınav Yeri	Teorik Sınav	Sınavlar										
		1	13	19	32	38	47	57	70	78	82	Toplam
14	1						9					9
	2			9								9
	3	9										9
	4							8				8
<b>Toplam 14</b>		<b>9</b>		<b>9</b>			<b>9</b>	<b>8</b>				<b>35</b>
18	1							1				1
	4					2						2
<b>Toplam 18</b>						<b>2</b>		<b>1</b>				<b>3</b>
25	1					1				9		10
	2			10								10
	3				9							9
	4							9				9
<b>Toplam 25</b>			<b>10</b>	<b>9</b>	<b>1</b>		<b>9</b>		<b>9</b>		<b>38</b>	
38	1			9								9
	2	9										9
	3									8		8
	4							8				8
<b>Toplam 38</b>		<b>9</b>		<b>9</b>				<b>8</b>		<b>8</b>	<b>34</b>	



Şekil 6.36. İlog Kısımlı Bazlı Sınav Yeri Atama İstatistikleri



Şekil 6.37. İlog Öğrenci Bazlı Sınav Yeri Atama İstatistikleri

### 6.4.3. Gözetmen Atama Sonuçları

Art arda oturumlarda görev yazılıp yazılmadığının kontrol tablosu.

Çizelge 6.4. Art Arda Görev Yazılma Durumu

Personel	Gün	Oturum 1_2	Oturum 3_4	Gün Toplam
m	1	1	0	1
m	2	0	1	1
m	.	0	1	1
m	g	0	1	1

Talebin karşılanmasının kontrol tablosu

**Çizelge 6.5.** Gözetmenlik Atama Kontrol Tablosu – Talep

Oturum	Sınav Yeri	Atanacak Görevli Sayısı	Atanan Görevli Sayısı	Atanmayan Görevli Sayısı
t	1	2	2	0
t	2	3	3	0
t	.	7	7	0
t	j			

Personele yazılan görev sonuçlarının incelenmesi

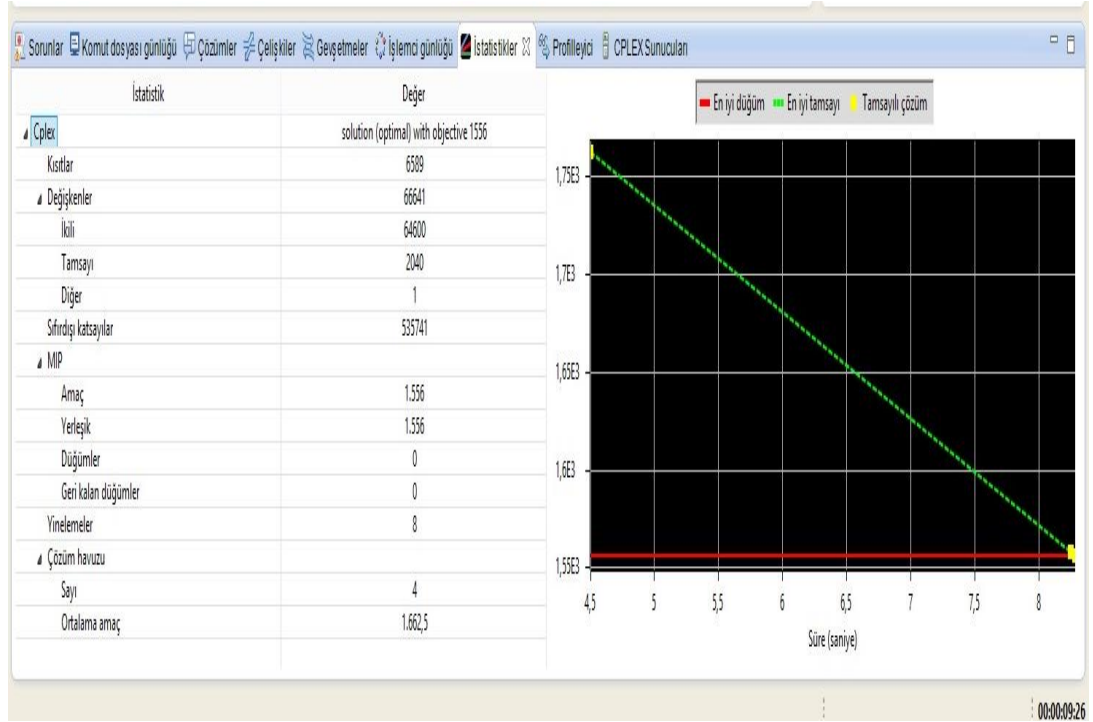
**Çizelge 6.6.** Gözetmenlik Atama Kontrol Tablosu – Arz

Unvan	Personel	Toplam Görev Sayısı	90' Görev Sayısı	Otrm 1 Görev Sayısı	Otrm 2 Görev Sayısı	Otrm 3 Görev Sayısı	Otrm 4 Görev Sayısı
	1	5	3	1	0	2	2
	2	5	2	2	0	2	1
	.		3	2	0	2	1
	.		2	1	1	2	1
	m	7	2	2	1	1	1

Mazerete görev yazılıp yazılmadığının kontrolü

**Çizelge 6.7.** Mazerete Görev Yazılma Durumu Kontrol Tablosu

Personel	KO	Sınav Yeri	Atanma Durumu	Mazereti
m	1	1	1	0
m	2	2	1	0
m	.	.	1	0
m	t	j	1	0



**Şekil 6.38.** Ilog Gözetmen Atama İstatistikleri

## 7. SONUÇ VE ÖNERİLER

Hizmet sektöründe yer alan organizasyonlar, varlıklarını sürdürebilmek için hizmet sistemlerini eniyileştirmek zorundadırlar. Hizmet sistemlerinde arz ve talep yönetim stratejileri, söz konusu organizasyonların kullandıkları sistemlerin eniyileştirilmesine katkı sağlayacak çözüm yöntemleri sunmaktadır. Özellikle arz yönetim stratejilerinden olan otomasyon ve çizelgeleme stratejilerinin organizasyonların faaliyetlerinin daha etkin yürütülmesini sağladığı tespit edilmiştir.

Otomasyon stratejisi kullanımı, gerçek zamanlı olarak en alt birimden en üst katmana kadar işletme içerisinde birbiri ile bütünleşik sistemlerin kurulmasını sağlamaktadır. İşletmelerde gerçekleşen faaliyetlerin ne oranda otomasyon içerisinde yapıldığı hususu, o işletmenin otomasyon stratejisini uygulamadaki gücünü gösterecektir. Elbette bu konuda güçlü olan işletme, rekabette büyük bir avantaj sağlamaktadır.

Çizelgeleme stratejisi uygulanması; geniş bir alanda faaliyet sürdüren hizmet sektöründeki işletmelerin kendi içerisinde yaşanan dinamik karar verme süreçlerinde sürekli olarak en iyi kararların alınabilmesi işletmeye oldukça fazla bir katma değer katacak ve yine rekabet gücüne olumlu yönde tesiri olmaktadır.

Hizmet sistemlerinden biri olan eğitim kurumlarında da durum farklı değildir. Eğitim kurumları oldukça geniş bir alanda faaliyet sürdürmektedirler. Büyüklü küçüklü çok sayıda eğitim kurumu olmakla birlikte bu kurumların ortalama büyüklüğünün hizmet sektöründe faaliyet gösteren birçok işletme ve kurumdan daha yaygın ve sayıca fazla olduğunu görmekteyiz.

Eğitim kurumlarında başta öğretmen, personel, ders, derslik ve öğrenciler olmak üzere çok sayıda kaynak bulunmaktadır. Verilen toplam ders sayısı, dersi alan öğrenci sayıları, dersi veren öğretmen miktarı, bu derslerin işlendiği yerler ve kurumdaki toplam öğrenci sayısı gibi veriler, ders programlarının hazırlanması, sınav, sınav yeri ve gözetmen planlarının kimlerle, nasıl, ne zaman ve nerelerde yapılacağı konularında karar verme zorluk seviyesini belirlemektedir.



Sınav uygulaması konusunda, okullarda kritik karar verme süreçlerindeki sınav çizelgeleme, öğrencilerin sınav salonlarına atanması ile söz konusu salonlara gözetmen ataması gibi dar boğazlarda etkili ve hızlı karar oluşturabilmek için bilgi sistemi ile bütünleşik bir optimizasyon programı vasıtasıyla, optimal çözümler elde edilmiştir. Bu problemlerin daha hızlı ve dinamik olarak çözümüne katkı sağlaması için tasarlanan veri tabanı ara yüzü ile personel, kendi isteklerini kendileri girebildiği gibi öğrenciler de isteklerini uzaktan erişimle tamamlayabilmektedir. Ayrıca, veri sınav tarihi, sınav yeri, görev tarih, saat ve yeri gibi konularda sorgulamalar yapılabilmektedir.

Bu güne kadar yapılan çalışmalarda sınav çizelgeleme, sınav yeri ve gözetmen atama problemlerinin birbirinden ayrı olarak çözümler geliştirildiği görülmüştür. Ancak, bu çalışmada sınav planlama sürecinde, sonraki modellerin çözümü önceki modellerin sonuçlarına bağlı olduğu düşünüldüğünden, sınav planlama süreci bir bütün olarak ele alınmak suretiyle modeller geliştirilmiştir.

Sınav çizelgeleme modeli bir eğitim öğretim yılında, sınav yeri atama modeli iki eğitim öğretim yılında ve gözetmen ataması modeli üç eğitim öğretim yılında bir yükseköğretim kurumunda uygulanmıştır.

Bu çalışma sonucunda; geliştirilen her bir modelden ortalama bir dakika içerisinde tüm kısıtları dikkate alan çözümler üretilmiştir. Sınav planlama süreci için geliştirilen modeller ile karar destek programı sınav uygulama kalitesinde iyileştirme sağlamıştır. Sınav yeri ataması bilgisayar ortamında yapıldığından öğrenciler eposta yolu ile bilgilendirilmiştir. Sınav salonlarına oturma planları çıktısı hazırlanmıştır. Sınav uygulama kalitesinin artmasına bağlı olarak kurumun güvenilirliği, personel ve öğrenci memnuniyeti, sınav güvenliği, sınav öncesinde sınav salonlarında yaşanacak karmaşanın önüne geçilmesi, iyi bir sınav uygulaması, objektif sonuçların ortaya çıkması, istatistiki olarak anlamlı analizlerin yapılabilmesi ve öğrenci ve öğretmenlerin emeklerinin daha iyi korunması gibi birçok fayda elde edilmiştir.

Sınav çizelgeleme modeli uygulanması sonucunda öğrenci ve öğretmenlerin istekleri maksimum düzeyde karşılandığı, sınav yeri atama modelinde özellikle öğrenci bazı

atama modelinde yan yana, arka arkaya ve çaprazda aynı sınava giren öğrenci olmayacak şekilde atama yapıldığı, gözetmen atama modelinde ise personelin mazeretleri ve modelde belirtilen tüm kriterler dikkate alınarak ve bir önceki modelde en az sınav yeri kullanılması nedeniyle personele en az sayıda görevlendirmenin yapıldığı sonuçlar elde edilmiştir.

Sınav planlama sürecinde verilecek kararlarla ilgili olarak, bilgisayar destekli olarak konusunda tecrübeli bir uzman tarafından çalışılması durumunda dahi zaman ve kaynak israfı olduğu tespit edilmiştir. Eğitim kurumlarının sayısının çokluğu dikkate alındığında kurumlarda yaşanacak böylesi bir israf neticesinde oldukça büyük bir kaynak kullanım sorunu ortaya çıkacaktır.

Sınav uygulama kalitesindeki artışa bağlı olarak kopyanın en aza indirilmesi, öğrencilerin seviyesinin objektif bir şekilde tespiti, öğrencilerin sınava odaklanması, sınavda herhangi bir problem olması durumunda bilgi sisteminde kayıtlı oturma planı sayesinde sorunlar daha kolay bir şekilde çözülmüştür.

Sınav çizelgeleme modeli neticesinde; manüel uygulamada uzun plan süresi, öğrenci ve öğretmenlerin isteklerini yerine getirememe, bir öğrenciye aynı anda birden fazla sınav yazılabilmesi, buna karşılık optimizasyon uygulamasında ortalama 60 saniye çözüm süresi, isteklerin karşılanması ve bir öğrenciye aynı anda birden fazla sınav yazılmaması gibi istenen sonuçların ortaya çıkması sağlanmaktadır.

Sınav yeri atama modeli neticesinde; manüel uygulamada en az iki iş günü (en az  $8*2=16$  saat) salon kapasitesini aşma ve salonda birden çok sınav olmasını minimize edememe, buna karşılık optimizasyon uygulamasında ortalama 60 saniye, salon kapasitesini aşmama, salonda birden çok sınav olmasının minimize edilmesi ve dersliklerin etkin kullanılması ve gözetmene daha çok hakimiyet imkanı tanıyacak istenen sonuçların ortaya çıkması sağlanmaktadır.

Gözetmenlerin atama modeli neticesinde; manüel uygulamada en az beş iş günü ( $8*5=40$  saat), mazerete görev yazma, sürelerin dengesiz dağıtılması, oturumların dikkate alınamaması, personelin akademik kıdemine göre görev yazılamaması, art

arda görev yazılması, buna karşılık optimizasyon uygulamasında ortalama 60 saniyede personelin tüm taleplerinin dikkate alınması gibi istenen sonuçların ortaya çıkmasını sağlamaktadır. Bununla birlikte; personelin akademik katma değeri olmayan işlerde minimum sayıda ve sürede görev alması sağlanmış, personel lehine tüm talepler karşılanmış ve personelin iş motivasyonu artmıştır.

Bu çalışmada geliştirilen modellerin esnek bir yapıya sahip olma avantajı da dikkate alınarak tüm eğitim kurumlarında özellikle de yükseköğretimde yaygınlaştırılmasının kaynak kullanımında etkinlik ve verimliliğe katkı sağlayacağı düşünülmektedir.

## KAYNAKLAR

- M. F. Acar, Sınav Çizelgelemesi için Matematiksel Model Yaklaşımı. Yüksek Lisans Tezi. Fatih Üniversitesi, İstanbul, 2010.
- C. Altıntaş, Sezgisel Algoritmalarla Sınav Çizelgeleme Problemi Çözümü: Yüksek Lisans Tezi. Süleyman Demirel Üniversitesi, Isparta, 2011.
- Bakır, M., Altunkaynak, B., Tamsayılı Programlama (Teori Modeller ve Algoritmalar). Nobel Yayın Dağıtım, Ankara, 2003.
- Balakrishnan, N. Examination Scheduling - A Computerized Application. Omega-International Journal Of Management Science(19), 37-41, 1991.
- Bazaraa, M. S., Linear Programming and Network Flows (2 b.). Wiley, New York, 1990.
- Binay, H., Aygüneş, H., Çetin, A., Oral, H., Güneri, A., Dalgıç, N., Yöneylem Araştırması Ders Kitabı. K.H.O. Matbaası,: Ankara, 2001.
- Burke, E., Bykov, Y., Petrovic, S. A Multicriteria Approach to Examination Timetabling. Practice And Theory Of Automated Timetabling III. Constance: Germany. s. 118-131, 2001.
- Carter, M., Laporte, G., Chinneck, J. A General Examination Scheduling System. Interfaces(24), 109-120. 1994.
- A. Çoruhlu, Sınav Personel Çizelgeleme Modeli. Yüksek Lisans Tezi. Gazi Üniversitesi, Ankara, 2007.

- Dammak, A., Elloumi, A., Kamoun, H. Classroom Assignment for Exam Timetabling. *Advances In Engineering Software*(37), 659-666. 2006.
- Elabdalla, A., Husein, R. An Optimal Examination Scheduler. *Computers & Education*(17), 267-269. 1991.
- Ü. B. Gürel, Sınav Çizelgeleme Probleminin Optimizasyonu. Yüksek Lisans Tezi:, Maltepe Üniversitesi, İstanbul, 2012.
- İlkuçar, M. Sınav Gözetmenlik Çizelgeleme Probleminin Optimizasyonu ve Bir Uygulama Yazılımı. Akademik Bilişim'11-XIII. Akademik Bilişim Konferansı Bildirileri (s. 413-420). Malatya: İnönü Üniversitesi. 2011.
- Ismayilova, N., Sağır, M., Gasimov, R. A multiobjective faculty–course–time slot assignment problem with preferences. *Elsevier*, 1017-1029. 2007.
- C. B. Kalaycı, Öğrenci Başarısına Odaklı Sınav Çizelgeleme Modeli ve Yazılım Uygulaması. Yüksek Lisans Tezi. Pamukkale Üniversitesi, Denizli, 2008.
- Karalar, R. Genel İşletme. Anadolu Üniversitesi, Eskişehir,. 2001.
- Lotfi, V., Cervený, R. A Final-Exam-Scheduling Package. *Journal Of The Operational Research Society*, 205-216. 1991.
- D. K. Matcı, Sınav Görevli Atama Problemine Bir Çözüm Önerisi: Anadolu Üniversitesi Uygulaması. Yüksek Lisans Tezi. Anadolu Üniversitesi, Eskişehir, 2014.
- Nizam, A., Veri Tabanı Tasarımı İlişkisel Veri Modeli ve Uygulamaları. Papatya, İstanbul, 2011.

Ozcan, E., Ersoy, E. Final Exam Scheduler - FES. 2005 IEEE Congress On Evolutionary Computation, Vols 1-3, Proceedings (s. 1356-1363). Edinburgh: Scotland. 2005.

Özdamar, S., Ersöz, S. Sınav Salonlarına Öğrenci ve Gözetmen Ataması Optimizasyonu. Yöneylem Araştırması Kongresi (YAEM). Bursa. 2014.

Öztürk, A. Yöneylem Araştırması (12 b.). Ekin Yayınevi. 2009.

B. Palamutçuoğlu, Üretim ve Hizmet Planlamasında Çizelgeleme Problemlerinin Yöneylem Teknikleriyle Çözümü: Ders ve Sınav Programlarının Optimizasyonu Üzerine Bir Uygulama. Yüksek Lisans Tezi: Celal Bayar Üniversitesi, Manisa, 2008.

Y. Seyyide, Öğretme ve Öğrenme Tabanlı Optimizasyon Algoritması ile Sınav Çizelgeleme Problemlerinin Çözümü. Yüksek Lisans Tezi. Dokuz Eylül Üniversitesi, İzmir, 2014.

Taban, F., Özdemir, E. Çok Parametrelili Genetik Algoritma Kullanarak Ders Programlarının Hazırlanması. II. Elektrik Elektronik Bilgisayar Mühendislikleri Eğitimi Sempozyumu. (s. 10-15). Samsun: Ondokuz Mayıs Üniversitesi, Mühendislik ve Mimarlık. 2004.

Taha, H. A. Yöneylem Araştırması (6 b.). (Ş. Baray, & Ş. Esnaf, Çev.). Literatür Yayınları, İstanbul, 2007.

Ulucan, A. Yöneylem Araştırması (2 b.). Siyasal Kitabevi, Ankara, 2007.

Wang, S., Bussieck, M., Guignard, M., Meeraus, A., & O'Brien, F. Term-end exam scheduling at United States Military Academy/West Point. Journal of Scheduling(13), 375-391. Ağustos 2010.

Winston, W. L. Operations Research Applications and Algorithms. Brooks/Cole, Cengage Learning, Belmont, 2004.

A. Yıldız, 0-1 Tam Sayılı Bulanık Hedef Programlama Yaklaşımı ile Sınav Görevi Atama Probleminin Çözümü. Yüksek Lisans Tezi. Dumlupınar Üniversitesi, Kütahya, 2004.

## EKLER

### EK-1 İlog Sınav Çizelgeleme Modeli

#### Model

```
//////////Tanımlar//////////
```

```
//Otomatik Sınav Planlama yapıyor
```

```
// Sınav Oturum Sayısı--bunu model belirleyecek
```

```
// önce büyük bir sayı ver. Sonra çıkan sonuca göre güncelle
```

```
// n sayısı en fazla 14 girilmeli. 15 ve üzeri olduğunda çözüm çok süresi artıyor
```

```
int M = 99999;          // Big M
```

```
range oturum = 1 .. 9;
```

```
int oturum_kap = 1250;
```

```
tuple ogr_snav {
```

```
    int sinav_grup_id;//birlikte yapılması istenen sınavlar için oluşturulan nu.
```

```
    int sinav_id;
```

```
    int ogr_id;
```

```
    int sinif;
```

```
    int sinav_suresi;
```

```
    int istenen_gun_id;
```

```
    int lisans_id;
```

```
};
```

```
{ogr_snav} t_ogr_snav = ...;
```

```
//{int} sinavler = {a2 | <a1,a2,a3,a4,a5> in t_ogr_snav };
```

```
{int} ogr_id = {a3 | < a1, a2, a3, a4, a5, a6, a7 > in t_ogr_snav};
```

```
{int} siniflar = {a4 | < a1, a2, a3, a4, a5, a6, a7 > in
```

```
t_ogr_snav};
```

```
//{int} lisanslar = {a7 | <a1,a2,a3,a4,a5,a6,a7> in t_ogr_snav };
```

```
tuple sin_grup {
```



```

    int sinav_grup_id;//birlikte yapılması istenen sınavlar için
oluşturulan nu.
    int istenen_gun_id;
};
{sin_grup} t_sin_grup = {< a1, a6 > | < a1, a2, a3, a4, a5, a6,
    a7 > in t_ogr_snav};

tuple sinav_tablo {
    key int sinav_id;
    int sinif;
    int sinav_suresi;
    int sinav_grup_id;
};
{sinav_tablo} t_snav = {< a2, a4, a5, a1 > | < a1, a2, a3, a4, a5,
a6,
    a7 > in t_ogr_snav};

tuple istek_tablo {
    int sinav_id;
    int gun_id;
    int deger;
};
{istek_tablo} t_istek = ...;

int sinif = card(siniflar);
int ogr_sys = card(ogr_id);

//////////////////Karar Degiskenleri//////////////////

dvar boolean x[t_ogr_snav][oturum];
//y:s.sinav j.oturuma atanıyor mu?
dvar boolean y[t_snav][oturum];
//h1:oturumlardaki en az süreden sapma değişkeni
dvar int+ h1ns;
dvar int+ h1ps;

//////////////////AMAC FONKSİYONU//////////////////

dexpr int amac2 = h1ps;

```

```

dexpr int amac = sum (s in t_snav, j in oturum,
                    is in t_istek:is.sinav_id==s.sinav_id && is.gun_id==j)
                    y[s][j] * j ;

dexpr int amac3 = sum (s in t_snav, j in oturum,
                    is in t_istek:is.sinav_id==s.sinav_id && is.gun_id==j)
                    y[s][j] * is.deger;

//////////AMAC//////////

minimize amac + amac2 + amac3;

//////////KISITLAR//////////

subject to {
//Talep: her öğrenciyi her bir sınavı için bir oturuma ata.1
forall ( i in t_ogr_snav )
    ct_gorev:
        sum ( j in oturum ) x[i][j] == 1;

//Arz: j. Oturumdaki arz kadar atama yapılsın.2
//aynı oturumda aynı öğrenciyi birden fazla sınav yazma
forall ( j in oturum )
    ct_oturum_kap_asma:
        sum ( i in t_ogr_snav ) x[i][j] <= oturum_kap;

//Bir öğrenciyi bir oturumda en fazla 1 sınav yazılsın.3
forall ( k in ogr_id, j in oturum )
    ct_snav_oturum_belirle2:
        sum ( i in t_ogr_snav : i.ogr_id == k ) x[i][j] <= 1;

//Sınavların oturumlara atanma durumu.4
forall ( j in oturum, s in t_snav )
    ct_snav_oturum_belirle:
        sum ( i in t_ogr_snav : i.sinav_id == s.sinav_id )
            x[i][j] <= M * y[s][j];

//Aynı sınav farklı oturumlara atanmasın.5
forall ( s in t_snav )
    ct_ayni_snav_farkli_oturumda_olmasin:

```

```

        sum ( j in oturum ) y[s][j] <= 1;

//İstenen bazı sınavların çakışmayan birden fazla sınavla birlikte
//istenen güne atanması.6
forall ( sg in t_sin_grup : sg.sinav_grup_id > 0 )
    ct_ayni_sinav_farkli_oturumda_olmasin2:
        sum ( s in t_sinav : s.sinav_grup_id == sg.sinav_grup_id, j in
oturum :
            j != sg.istenen_gun_id ) y[s][j] == 0;

//Belirtilen sınavlar aynı oturumlarda olsun.6.1
forall ( s in t_sinav : s.sinav_id == 1 )
    ct_birlikte_olsun:
        sum ( j in oturum : j == 5 ) y[s][j] == 1;

//s.sınıfın j.oturumda sınavı olmasın.7
forall ( s in t_sinav : s.sinif == 3 )
    ct_ayni_sinav_farkli_oturumda_olmasin3:
        sum ( j in oturum : j == 10 ) y[s][j] == 0;

//Son günler mevcudu az olan sınavlar olsun.8
forall ( j in oturum : j > 7 )
    ct_son_gunler_az_mevcut:
        sum ( i in t_ogr_sinav ) x[i][j] <= 450;

//4.oturumdaki sınavlar mümkün olduğunca en az süreli sınavlardan
oluşsun.9
forall ( c in siniflar : c == 3, j in oturum : j == 7 )
    ct_esit_sure:
        max ( s in t_sinav : s.sinif == c ) ( y[s][j] * s.sinav_suresi
)
        + h1ns - h1ps == 60;
};

//////////SON//////////

tuple result {
    int sinav_id;
    int gun_id;
    int sonuc;

```

```

}

{result} Result = { < i.sinav_id,
                    j,
                    x[i][j]> |
                    i in t_ogr_sinav, j in oturum:
                    x[i][j]>0 };

//////////*KOMUT DOSYASI*//////////

execute DISPLAY_RESULTS {
    writeln("Belirlenen Blok Sayisi= ", amac/4); //amaç değerine göre excelde
    yazdırılacak alan belirlenecek
    writeln("Sınıf Sayısı=", sinif);
    writeln("Öğrenci Sayısı=", ogr_say);
}

/*****
* OPL 12.6.0.0 Data
* Author: Mavi Açelya
* Creation Date: 24 Ara 2014 at 15:26:55
*****/

DBConnection db("odbc","Sinav_Sistemi_Optimizasyonu//");
DBExecute(db,"delete from t_sonuc_aktif_sinav_cizelgeleme");

t_ogr_sinav from DBRead(db,"SELECT * FROM
v_ilog_sinav_cizelgeleme");
t_istek from DBRead(db,"SELECT sinav_id, sta_id, deger FROM
v_sinav_istek_ilog");
Result to DBUpdate(db,"INSERT INTO
t_sonuc_aktif_sinav_cizelgeleme(sinav_id, gun_id, sonuc)
VALUES(?,?,?) ");

```

## EK-2 İlog Kısım Bazlı Sınav Yeri Atama Modeli

```
/*
*****
* OPL 12.6.0.0 Model:
* Kısım Bazlı Sınav Yeri Atama Modeli
* Author: Semih ÖZDAMAR
* Creation Date: 16 Tem 2014 at 23:39:05
*****
//////////////////////////////////TANIMLAR//////////////////////////////////
tuple sinav_kisim {
    int oturum;
    int sinif;
    int sinav_id;
    int sinav_suresi;
    int kisim_id;
    int mevcut;
};
{sinav_kisim} t_sinav_kisim = ...;

tuple t_sinav_oturum {
    int oturum;
    int sinav_id;
}

{int} oturumlar = {a1 | < a1,a2,a3,a4,a5,a6 > in t_sinav_kisim};
{t_sinav_oturum}
    sinavlar = {< a1,a3 > | < a1,a2,a3,a4,a5,a6 > in
t_sinav_kisim};

tuple sinav_yeri {
    int sin_yer_id;
    string sin_yer_adi;
    int kap;
    int agirlik;
};
{sinav_yeri} t_sinav_yeri=...;

//toplam sinav yeri sayısı
int h = card(t_sinav_yeri);
```

```

//////////////////////////////////Karar Degiskenleri//////////////////////////////////

//s.sınavda i.kısımın j.sınav yerine atanması
dvar boolean  x[t_sınav_kisim][t_sınav_yeri];
//t. oturumunda j. sınav yerine atama yapılması
dvar boolean  y[oturumlar][t_sınav_yeri];

//dvar int+ hlms[oturumlar][t_sınav_yeri];
//dvar int+ hlps[oturumlar][t_sınav_yeri];
//hedef1: j. sınav yerine mümkün olduğunca
//--istenen sınav sayısı kadar atama yapılması

dvar boolean  z[sınavlar][t_sınav_yeri];
//s.dersin j.sınav yerine atanması

//////////////////////////////////AMAC FONKSİYONU//////////////////////////////////

dexpr int amac1 = sum (t in oturumlar,  j in t_sınav_yeri )
y[t][j];
dexpr int amac2 = sum (i in t_sınav_kisim, j in t_sınav_yeri )
x[i][j]*j.agirlik;
dexpr int amac3 = sum (s in sınavlar,    j in t_sınav_yeri )
z[s][j];

//dexpr int amac  = sum ( u in oturumlar, j in t_sınav_yeri ) (
hlps[u][j] + y[u][j] ); //sapma
//dexpr int amac  = sum ( u in oturumlar, j in t_sınav_yeri ) (
hlps[u][j] ); //sapma

//////////////////////////////////AMAC//////////////////////////////////

minimize amac1 + amac2 + amac3;
//minimize amac3 ;

```

```

//////////////////////////////////KISITLAR//////////////////////////////////

subject to {
  //1.Talep
  forall ( i in t_sinav_kisim )
    ct_sinav_kisim_ata:
      sum ( j in t_sinav_yeri ) x[i][j] == 1;

  //2.Arz -- t.oturumda j. sınav yerine atama yapılıyor mu?
  //yapılacak atama arzı geçmesin.
  forall ( t in oturumlar, j in t_sinav_yeri )
    ct_kapasiteyi_asma:
      sum ( i in t_sinav_kisim : i.oturum == t )
        x[i][j] * i.mevcut <= j.kap * y[t][j];

  //3. -- s.sınavın j.sınav yerine atanması
  forall ( s in sınavlar, j in t_sinav_yeri )
    ct_sinav_ata:
      sum ( i in t_sinav_kisim : i.sınav_id == s.sınav_id )
        x[i][j] <= h * z[s][j];

  //4. -- t.oturumda, j.sınav yerine yapılacak sınav sayısı kısıtı
  forall ( t in oturumlar, j in t_sinav_yeri )
    ct_sinav_sayisi:
      sum ( s in sınavlar : s.oturum == t )
        z[s][j] <= 8;
  //    z[v][j] + h1ns[u][j] - h1ps[u][j] == 1;

  //5. -- t.oturumda, j.sınav yerine yapılacak sınav sayısı kısıtı
  // forall ( t in oturumlar, j in t_sinav_yeri )
  //    ct_sinav_sayisi2:
  //      sum ( s in sınavlar : s.oturum == t )
  //        z[s][j] <= 8;
  //    z[v][j] + h1ns[u][j] - h1ps[u][j] == 1;

  //6. -- özel istek ataması
  //sum ( i in t_sinav_kisim : i.sınav_id ==1,
  //    j in t_sinav_yeri: j.sin_yer_id==5 )
  //    x[i][j] ==1;
};

```

```

////////////////////////////////SON////////////////////////////////////

tuple result {
    int sinav_id;
    int kisim_id;
    int sin_yer_id;
    int sonuc;
}

{result} Result = { < i.sinav_id,
                    i.kisim_id,
                    j.sin_yer_id,
                    x[i][j]> |
                    i in t_sinav_kisim, j in t_sinav_yeri:
                    x[i][j]>0 };

/*****
* OPL 12.5 Data
* Author: Semih ÖZDAMAR
* Creation Date: 18 Mar 2014 at 11:30:23
*****/

DBConnection db("odbc","Sinav_Sistemi_Optimizasyonu//");
DBExecute (db,"delete from
t_sinav_yeri_cizelgeleme_kisim_bazli_aktif_sonuc");

t_sinav_kisim from DBRead(db,"SELECT * FROM
v_ilog_sinava_girecek_kisimler");
t_sinav_yeri   from DBRead(db,"SELECT * FROM v_ilog_sinav_yerleri");

Result to DBUpdate(db,"INSERT INTO
t_sinav_yeri_cizelgeleme_kisim_bazli_aktif_sonuc(sinav_id, kisim_id,
sin_yer_id, sonuc) VALUES(?,?,?,?) ");

```



### EK-3 İlog Öğrenci Bazlı Sınav Yeri Atama Modeli

```
/******  
* OPL 12.6.0.0 Model  
* Sınav Salonlarına Öğrenci Atama Modeli-Öğrenci Bazlı  
* Creation Date: 08 Haz 2015 at 23:32:53  
*****/  
  
/*////AÇIKLAMA/////////  
1.Sınav yerlerinde aynı dersteki öğrenciler yan yana,  
arka arkaya ve çapraz gelmeyecek şekilde  
bir salonda sınava girecek teorik ders sayısı ve  
herbir ders için öğrenci sayısı hesaplanır.  
  
2.Hesaplanan teorik derslerin kapasitesini  
geçmeyecek şekilde atama yapılır.  
*/  
  
//////////**Tanımlar**//////////  
  
tuple sinav_yeri {  
    int sinav_yer_id;  
    int teo_ders_id;  
    int teo_ders_kap;  
};  
{sinav_yeri} t_sinav_yeri_sablon=...;  
  
{int} t_sinav_yeri = { a1 | <a1,a2,a3> in t_sinav_yeri_sablon };  
  
tuple sinavlar {  
    int kum_oturum;  
    int sinif;  
    int sinav_id;  
    int sure;  
    int sinav_grp_id;  
    int ogr_mevcut;  
}  
{sinavlar} t_sinavlar = ...;  
  
{int} kum_oturum = {a1 | <a1,a2,a3,a4,a5,a6 > in t_sinavlar };
```

```

{int} sinav_grp = {a5 | <a1,a2,a3,a4,a5,a6 > in t_sinavlar : a5>0
};

////////////////////////////////**Karar Degiskenleri**////////////////////////////////

// x: sinav yeri teorik ders grubuna herhangi bir sınavdan atanan
öğrenci sayısı
dvar int+ x[t_sinavlar][t_sinav_yeri_sablon];
// t: sınavlar hangi salonlara dağılıyor
dvar boolean t[t_sinavlar][t_sinav_yeri_sablon];
// y: her bir günde hangi salonlar açılıyor
dvar boolean y[kum_oturum][t_sinav_yeri];

////////////////////////////////**AMAC FONKSIYONU**////////////////////////////////

//amac1: her bir gün için en az salon açılsın
dexpr float amac1 = sum ( k in kum_oturum,
                        h in t_sinav_yeri )
                        y[k][h] ;

//amac2: en az amfi açılsın
dexpr float amac2 = sum ( k in kum_oturum,
                        h in t_sinav_yeri: h==6 )
                        y[k][h]*9999 ;

//amac3: amfi açılırsa buraya yapılacak atama en az olsun

dexpr float amac3 = sum ( s in t_sinavlar,
                        j in t_sinav_yeri_sablon: j.sinav_yer_id==6 )
                        t[s][j]*x[s][j]*1000 ;

////////////////////////////////**AMAC**////////////////////////////////

minimize amac1 + amac2 + amac3;

```

```
//////////////////////////////////**KISITLAR**//////////////////////////////////
```

```
subject to {
```

```
//1- sınava girecek tüm öğrenciler sınav yerlerine atansın
```

```
forall ( i in t_sinavlar )  
    ct_1_ogr_sin_yer_ata:  
        sum ( j in t_sinav_yeri_sablon )  
            x[i][j] == i.ogr_mevcut;
```

```
//2- aynı salonda aynı sınava girecek öğrencilerin toplamı
```

```
//salonda atandığı teorik dersin öğrenci kapasitesini aşmasın
```

```
forall ( k in kum_oturum,  
        s in t_sinavlar: s.kum_oturum==k,  
        j in t_sinav_yeri_sablon )  
    ct_2_teo_derse_ataniyor_mu:  
        sum ( i in t_sinavlar: i.kum_oturum==k &&  
            i.sinav_id==s.sinav_id )  
            x[i][j] <= j.teo_ders_kap*t[s][j];
```

```
//3- bir sınav, salondaki sadece bir teorik ders grubuna atansın
```

```
forall ( s in t_sinavlar,  
        h in t_sinav_yeri )  
    ct_3_bir_sinav_sadece_bir_salona_atansin:  
        sum ( j in t_sinav_yeri_sablon: j.sinav_yer_id==h )  
            t[s][j] <= 1;
```

```
//4- aynı salondaki teorik ders gruplarında en fazla 2 farklı sınav  
olsun
```

```
forall ( j in t_sinav_yeri_sablon)  
    ct_4_teorik_ders_sinav_ust_sinir:  
        sum ( s in t_sinavlar)  
            t[s][j]<=2;
```

```
//5- aynı sınav grubunda olan sınavlar, salondaki sadece bir teorik  
ders grubuna atansın
```

```
forall ( grp in sinav_grp,  
        h in t_sinav_yeri )  
    ct_5_sinav_grp_atama:  
        sum ( s in t_sinavlar: s.sinav_grp_id==grp,
```

```

        j in t_sinav_yeri_sablon: j.sinav_yer_id==h )
t[s][j] <= 1;

//6- her bir oturumda hangi sınav yerleri açılıyor.
forall ( k in kum_oturum, h in t_sinav_yeri,
        j in t_sinav_yeri_sablon:j.sinav_yer_id==h )
ct_6_acilan_salonlar:
    sum ( i in t_sinavlar: i.kum_oturum==k)
        x[i][j]<=j.teo_ders_kap*y[k][h];

};

////////////////////////////////**SON**////////////////////////////////

tuple result {
    int kum_oturum;
    int sinav_id;
    int sinav_yer_id;
    int teo_ders_id;
    int sonuc;
}

{result} Result = { < i.kum_oturum,
                    i.sinav_id,
                    j.sinav_yer_id,
                    j.teo_ders_id,
                    x[i][j]> |
                    i in t_sinavlar, j in t_sinav_yeri_sablon:
                    x[i][j]>0 };

////////////////////////////////**NOTLAR**////////////////////////////////

/*****
* OPL 12.6.0.0 Data
* Sınav Salonlarına Öğrenci Atama Modeli-Öğrenci Bazlı
* Creation Date: 08 Haz 2015 at 23:33:08
*****/

DBConnection db("odbc","Sinav_Sistemi_Optimizasyonu//");

```

```
DBExecute (db,"delete from
t_snav_yeri_cizelgeleme_ogrenci_bazli_aktif_sonuc");

t_snavlar      from DBRead(db,"SELECT * FROM v_ilog_snav_mevcutlari
where kum_oturum=21");

t_snav_yeri_sablon from DBRead(db,"SELECT snav_yer_id, oturum_id,
maks_ogr_sys FROM t_snav_yeri_snav_atama_sablonu ");

Result          to          DBUpdate (db,"INSERT          INTO
t_snav_yeri_cizelgeleme_ogrenci_bazli_aktif_sonuc(kum_oturum,
snav_id, snav_yer_id, oturum_id, sonuc) VALUES(?,?,?,?,?) ");
```

## EK-4 Ilog Gözetmen Atama Modeli

```

/*****
* OPL 12.6.0.0 Model
* Gözetmen Atama Modeli
* Author: Semih ÖZDAMAR
* Creation Date: 05 Şub 2015 at 13:07:47
*****/

//////////**Tanımlar**//////////

tuple otrm_snv_yer {
    int gun_oturum;
    int gun_id;
    int oturum;
    int snv_yer_id;
    int max_sure;
    int doksan_dk_mi;
    int ogr_mevcudu;
    int hes_goz_sys;
    int sa;
    int es;
    int kg;
    int g;
    int top_gztnm;
}

{otrm_snv_yer} t_otrm_snv_yerleri = ...;

sorted {int} snv_yerleri = { a4 |
    < a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13 > in
t_otrm_snv_yerleri };

tuple per_oturum {
    int gun_oturum;
    int sta_id;
    int gun;
    int oturum;
    int per_id;
    int rutbe_id;
}
```

```

    int bolum_id;
    int mazeret;
}

{per_oturum} t_per_oturum = ...;

tuple per_tablo {
    int per_id;
    int rutbe_id;
}

{per_tablo} per_list = {<b5,b6> |
                        <b1,b2,b3,b4,b5,b6,b7,b8> in t_per_oturum };

{int} gunler    = {b3 | <b1,b2,b3,b4,b5,b6,b7,b8> in t_per_oturum };
{int} oturumlar = {b4 | <b1,b2,b3,b4,b5,b6,b7,b8> in t_per_oturum
};

int ort_doksandk_gorev_sayisi=1;
float ort_gorev_sayisi=1.17;

//////////**Karar Degiskenleri**//////////

dvar boolean  x [t_per_oturum][snv_yerleri];

dvar int+    per_grv_ns [per_list];
dvar int+    per_grv_ps [per_list];
dvar int+    dksn_dk_ns [per_list];
dvar int+    dksn_dk_ps [per_list];
dvar int+    otrm_ort_ns [per_list][oturumlar];
dvar int+    otrm_ort_ps [per_list][oturumlar];

//////////indexli_kısıtlar//////////
constraint ct_1_gztmn_40_ogr[t_otrm_snv_yerleri];
constraint ct_mazerete_grv_yazma[per_list];
constraint ct_otrm_ort_grv[per_list][oturumlar];

constraint ct_doksan_dk [per_list];
constraint ct_gorev_yaz_per [per_list];

```

```

constraint ct_doksan_dk_maks [per_list];
constraint ct_gorev_yaz_per_maks[per_list];

execute PARAMS {
  cplex.tilim = 120;
}

//////////**AMAC FONKSIYONLARI**//////////

//talep kadar görev yazılması
dexpr float a_gorev_sayisi = sum (i in t_per_oturum,
                                j in snv_yerleri)
                                x[i][j];

//genel ort.'dan sapmalar min. olsun
dexpr float b_genel_ort_sapma = sum (p in per_list)
                                (per_grv_ns[p] + per_grv_ps[p]);

//doksan_dk ort.'sından sapmalar min. olsun
dexpr float c_d_dk_ort_sapma = sum (p in per_list)
                                (dksn_dk_ps[p] + dksn_dk_ns[p]);

//oturum ort.'sından sapmalar min. olsun
dexpr float d_otrm_ort_sapma = sum (p in per_list,
                                    o in oturumlar)
                                    otrm_ort_ns[p][o];

//personel mazeretine min. görev yazılsın
//dexpr float amac5 = sum (p in per_list, t in gun_oturum)
//                    ( mazeret_ns[p][t] );
//////////**AMAC**//////////

minimize
  a_gorev_sayisi + b_genel_ort_sapma + c_d_dk_ort_sapma +
  d_otrm_ort_sapma ;

```



```

//////////////////**KISITLAR**//////////////////

subject to {

//1- her oturum için sınav yerlerindeki talebi karşıla
forall ( st in t_otrm_snv_yerleri,
         j in snv_yerleri : j==st.snv_yer_id )
ct_1_gztnm_40_ogr[st]:
    sum ( i in t_per_oturum : i.gun_oturum == st.gun_oturum )
        x[i][j] == st.top_gztnm ;

//2- ya sabah ya öğleden sonra olacak şekilde
//günde en fazla iki görev gelmesi
forall ( p in per_list, g in gunler, r in 1 .. 2)
ct_pespepe_gorev:
    sum ( o in 2*r-1 .. 2*r,
          i in t_per_oturum:
            i.per_id==p.per_id && i.gun==g && i.oturum==o,
          j in snv_yerleri )
        x[i][j] <= 1;

//3- personelin mazeretine görev yazma
forall ( p in per_list )
ct_mazerete_grv_yazma[p]:
    sum ( i in t_per_oturum: i.per_id==p.per_id && i.mazeret>0 ,
          j in snv_yerleri )
//x[i][j]*i.mazeret + mazeret_ns[p][t] - mazeret_ps[p][t] == 0;
        x[i][j] == 0;

//4- personele ortalama kadar görev yaz
forall ( p in per_list )
ct_gorev_yaz_per[p]:
    sum ( i in t_per_oturum: i.per_id==p.per_id,
          j in snv_yerleri )
        x[i][j] + per_grv_ns[p] - per_grv_ps[p] ==
trunc(ort_gorev_sayisi);

//5- herkese üst sınır - ort.dan bir fazla
forall ( p in per_list )
ct_gorev_yaz_per_maks[p]:

```

```

        sum ( i in t_per_oturum: i.per_id==p.per_id,
              j in snv_yerleri )
x[i][j] <= trunc(ort_gorev_sayisi)+2;

//6- istenen unvanlara üst sınır
//6.1- unvanı 1 ve 2 olanlara en fazla 5 görev yaz
forall ( p in per_list: p.rutbe_id in 1..2 )
    ct_max_gorev_1_2:
        sum ( i in t_per_oturum: i.per_id==p.per_id,
              j in snv_yerleri )
x[i][j] <= 5;

//6.2- unvanı 3 ve 4 olanlara en fazla 6 görev yaz
forall ( p in per_list: p.rutbe_id in 3..4 )
    ct_max_gorev_3_4:
        sum ( i in t_per_oturum: i.per_id==p.per_id,
              j in snv_yerleri )
x[i][j] <= 6;

//6.3- unvanı 2'den büyük olanlara en az 6 görev yaz
// forall ( p in per_list: p.rutbe_id>2 )
//     ct_min_gorev_b_2:
//         sum ( i in t_per_oturum: i.per_id==p.per_id,
//               j in snv_yerleri )
//     x[i][j] >= 1;

//7- doksan dk. görevlerin düzgün dağıtılması
forall ( p in per_list ) {
    ct_doksan_dk[p]:
        sum ( st in t_otrm_snv_yerleri: st.max_sure==90,
              i in t_per_oturum:
                i.per_id==p.per_id && i.gun_oturum==st.gun_oturum ,
              j in snv_yerleri :
                j==st.snv_yer_id )
x[i][j] + dksn_dk_ns[p] - dksn_dk_ps[p] ==
ort_doksandk_gorev_sayisi ;

    ct_doksan_dk_maks[p]:
        sum ( st in t_otrm_snv_yerleri: st.max_sure==90,
              i in t_per_oturum:

```

```

        i.per_id==p.per_id && i.gun_oturum==st.gun_oturum ,
        j in snv_yerleri :
        j==st.snv_yer_id )
        x[i][j] <= 2 ;
};

//8- oturum bazlı kısıtlar
/*
forall (p in per_list,
        o in oturumlar ) {
//8.1- herkese oturum ortalaması kadar görev yazılsın
ct_otrm_ort_grv[p][o]:
    sum (i in t_per_oturum:
        i.per_id==p.per_id && i.oturum==o ,
        j in snv_yerleri)
    x[i][j] + otrm_ort_ns[p][o] - otrm_ort_ps[p][o] == 2 ;
//8.2- herkese 4.otrmda en az 1 grv yazılsın
ct_4_otrmda_min_1_grv:
    sum (i in t_per_oturum:
        i.per_id==p.per_id && i.oturum==4 ,
        j in snv_yerleri)
    x[i][j] >= 1 ;
};
*/

//9- gözetmenin istenen bir gündeki istenen salona atanması
forall (i in t_per_oturum :
        i.per_id == 5 && i.gun_oturum==11,
        j in snv_yerleri:j==3 )
ct_tekil:
    x[i][j] == 1;

//10- salonlardaki istenen rütbelerin dağılımları
//10.1- yakın rütbeler aynı salonda olmasın
forall ( st in t_otrm_snv_yerleri,
        j in snv_yerleri : j==st.snv_yer_id )
ct_gztnm_1_2_3:
    sum ( i in t_per_oturum :
        i.gun_oturum == st.gun_oturum && i.rutbe_id in 1..4 )
    x[i][j] <= 1 ;

```

```

        //10.2- her salona en az bir tane 5 ve 6 unvanı olan
personel
forall ( st in t_otrm_snv_yerleri: st.top_gztn>1,
        j in snv_yerleri : j==st.snv_yer_id )
ct_gztn_5_6:
    sum ( i in t_per_oturum :
        i.gun_oturum == st.gun_oturum && i.rutbe_id in 5..6 )
        x[i][j] >= 1 ;

/*11- ilgili oturumadaki görevliler ilgisiz kişiler olmasın
forall ( st in t_otrm_snv_yerleri:
        st.gun_oturum in {63,64} && st.snv_yer_id in 15..20,
        j in snv_yerleri : j==st.snv_yer_id )
ct_ilgili_gorev:
    sum ( i in t_per_oturum :
        i.gun_oturum == st.gun_oturum && i.bolum_id!=11)
        x[i][j] == 0 ;

forall ( st in t_otrm_snv_yerleri:
        st.gun_oturum ==24 && st.snv_yer_id == 15,
        j in snv_yerleri : j==st.snv_yer_id )
ct_ilgili_2:
    sum ( i in t_per_oturum :
        i.gun_oturum == st.gun_oturum && i.per_id==11111)
        x[i][j] == 1 ;*/

};
////////////////////////////////**SON**////////////////////////////////

tuple result {
    int sta_id;
    int oturum;
    int per_id;
    int snv_yer_id;
    int sonuc;
}

```

```

{result} Result = { < i.sta_id,
                    i.oturum,
                    i.per_id,
                    j,
                    x[i][j]> |
                    i in t_per_oturum, j in snv_yerleri:
                    x[i][j]>0 };

/*****
* OPL 12.6.0.0
* Gözetmen Atama Modeli Data Bağlantısı
* Author: Semih ÖZDAMAR
* Creation Date: 05 Şub 2015 at 13:07:47
*****/

DBConnection db("odbc","Sinav_Sistemi_Optimizasyonu//");
DBExecute(db,"delete from t_gozetmen_aktif_sonuc");

t_per_oturum      from DBRead(db,"SELECT * FROM v_ilog_gozetmen_atama
where oturum=1");
t_otrm_snv_yerleri from DBRead(db,"SELECT * FROM
v_ilog_gozetmen_ihtiyaci");

Result to DBUpdate(db,"INSERT INTO t_gozetmen_aktif_sonuc(sta_id,
oturum, per_id, sin_yer_id, sonuc) VALUES(?,?,?,?,?) ");

```



## ÖZGEÇMİŞ

Adı Soyadı : Semih ÖZDAMAR

Doğum Tarihi : 27.10.1985

Yabancı Dil : İngilizce

Eğitim Durumu :

Lisans : İstanbul Ticaret Üniversitesi Mühendislik ve Tasarım Fakültesi  
Endüstri Mühendisliği Bölümü (2007)

Yüksek Lisans : Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Endüstri  
Mühendisliği Bölümü (2015)

Çalıştığı Kurum/Kurumlar ve Yıl/Yıllar:

Kara Kuvvetleri Lojistik Komutanlığı Bakım Komutanlığı Bakım, Üretim ve  
Koordinasyon Şubesi 2008-2012

Kara Harp Okulu Komutanlığı Endüstri ve Sistem Mühendisliği Bölümü (Öğretim  
Görevlisi) 2012

Kara Harp Okulu Komutanlığı Öğrenci İşleri Şube Müdürlüğü Sınav Plan ve  
Koordinasyon Subaylığı ve Öğretim Görevlisi 2012-...

Yayımları (SCI) :

Yayımları (Diğer):

Özdamar, S. ve Ersöz, S. (2014). Sınav Salonlarına Öğrenci ve Gözetmen Ataması  
Optimizasyonu. *Yöneylem Araştırması Kongresi (YAEM)*. Bursa.

Araştırma Alanları :

Yöneylem Araştırması, Sistem Simülasyonu, Optimizasyon, Uzman Sistemler, Bilgi  
Sistemleri Tasarımı..