

**ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMASI İÇİN
MATEMATİKSEL MODELLER VE ÇÖZÜM YAKLAŞIMLARI**

Yüksek Lisans Tezi

Zeliha ERGÜL

Eskişehir, 2016

**ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMASI İÇİN
MATEMATİKSEL MODELLER VE ÇÖZÜM YAKLAŞIMLARI**

Zeliha ERGÜL

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Zehra KAMIŞLI ÖZTÜRK

Eskişehir

Anadolu Üniversitesi

Fen Bilimleri Enstitüsü

Haziran, 2016

JÜRİ VE ENSTİTÜ ONAYI

Zeliha ERGÜL'ün "Çok Oturumlu Sınavlarda Öğrenci Bina Ataması İçin Matematiksel Modeller Ve Çözüm Yaklaşımları" başlıklı tezi 30/06/2016 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin ilgili maddeleri uyarınca, Endüstri Mühendisliği Anabilim dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Unvanı-Adı Soyadı

İmza

Üye (Tez Danışmanı) : Yard. Doç. Dr. Zehra KAMIŞLI ÖZTÜRK

Üye : Doç. Dr. Onur KAYA

Üye : Yard. Doç. Dr. Tuğba SARAÇ

.....

Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMASI İÇİN MATEMATİKSEL MODELLER VE ÇÖZÜM YAKLAŞIMLARI

Zeliha ERGÜL

Endüstri Mühendisliği Anabilim Dalı

Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Haziran, 2016

Danışman: Yard. Doç. Dr. Zehra KAMIŞLI ÖZTÜRK

Bu tez kapsamında Anadolu Üniversitesi Açıköğretim Sistemi'ne kayıtlı öğrencilerin sınav binalarına atanması problemi ele alınmıştır. Ele alınan problemle benzer özelliklere sahip olan çalışmalar eğitimsel zaman çizelgeleme başlığı altında araştırılmıştır. Literatür araştırmasından sonra problemin çözümü için çok amaçlı karma tamsayılı doğrusal olmayan bir matematiksel model geliştirilmiş ve matematiksel modelin çözümleri incelenmiştir Öğrenci sınav yeri atama problemi Anadolu Üniversitesi Açıköğretim sistemine ait büyük boyutlu bir gerçek hayat problemi olduğu için matematiksel modelin problemin çözümünde yetersiz kaldığı görülmüştür. Buradan yola çıkarak problemin çözümü için sıralama temelli çok amaçlı bir sezgisel algoritma geliştirilmiştir. Geliştirilen sezgisel algoritma ile elde edilen sonuçlar, matematiksel modelleme ile elde edilen sonuçlar ve mevcut sistem çıktıları ile karşılaştırılmıştır.

Anahtar Sözcükler: Eğitimsel zaman çizelgeleme, Çok amaçlı optimizasyon, Atama problemi

ABSTRACT

Master of Science Thesis

MATHEMATICAL MODEL AND SOLUTION APPROACHES FOR STUDENT- EXAMINATION BUILDING ASSIGNMENT IN MULTISESSION EXAMS

Zeliha ERGÜL

Industrial Engineering Program

Anadolu University, Graduate School of Sciences, June,2016

Supervisor: Assist. Prof. Dr. Zehra KAMIŞLI ÖZTÜRK

In this study, we focused on assignment of students which enrolled at Anadolu University's distance education system to examination buildings. The studies similar to undertaken problem are investigated under the head of educational timetabling. After literature research multi objective mixed-integer nonlinear mathematical model is developed to solve the problem and the solution of the mathematical model is examined. Mathematical model has been found inadequate because students-examination building assignment which is belong to Anadolu University Open Education system is a large size real life problem. Starting from this point of view, an order-based multiobjective heuristic algorithm is developed to solve the problem. The obtain solutions by the proposed algorithm are compared with the solution obtained by the mathematical modelling and the out of existing system.

Key words: Educational timetabling, Multi-objective optimization, Assignment problem

ÖNSÖZ

Engin bilgisi ve pozitif enerjisi ile çalışmalarına ışık tutarak benden desteğini esirgemeyen, yüksek lisans eğitimine verdiğim aradan sonra akademik anlamda beni cesaretlendirip, geliştiren tez danışmanım Yard. Doç Dr. Zehra KAMIŞLI ÖZTÜRK'e en içten duygularıyla teşekkür ederim. Tez çalışmalarım boyunca bana destek olan çalışma arkadaşlarıma ve hayatım boyunca koşulsuz desteklerini hissettiğim annem Şaziye ERGÜL'e ve babam Abdullah ERGÜL'e teşekkürü bir borç bilirim. Bu çalışmanın öğrenci sınav yeri belirleme problemi ile ilgilenecek araştırmacılara faydalı olmasını dilerim.

Haziran 2016

Zeliha ERGÜL

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmamın Anadolu Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı”yla tarandığımı ve hiçbir şekilde “intihal içermediğimi” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

.....

(İmza)

.....

(Adı-Soyadı)

İÇİNDEKİLER

BAŞLIK SAYFASI.....	i
JÜRİ VE ENSTİTÜ ONAYI	ii
ÖZET.....	iii
ABSTRACT	iv
ÖNSÖZ.....	v
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ	vi
İÇİNDEKİLER.....	vii
TABLolar DİZİNİ	ix
ŞEKİLLER DİZİNİ	x
1. GİRİŞ.....	1
2. ATAMA PROBLEMLERİ.....	3
2.1 Doğrusal Atama Problemleri.....	3
2.2 Doğrusal Olmayan Atama Problemleri ..Hata! Yer işareti tanımlanmamış.	
2.3 Çok Boyutlu Atama Problemleri.....	4
3. EĞİTİMSEL ZAMAN ÇİZELGELEME PROBLEMLERİ.....	7
3.1 Ders Zaman Çizelgeleme Problemleri	8
3.2 Sınav Zaman Çizelgeleme Problemi	9
3.3 Çözüm Yaklaşımları ve Teknikleri	10
3.3.1 Matematiksel temelli yaklaşımlar.....	10
3.3.2 Sezgisel algoritmalar	10
3.3.3 Metasezgisel algoritmalar.....	10
3.3.4 Graf renklendirme.....	11
3.3.5 Sezgisel üstü (hyper heuristic) algoritmalar	11

4. ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMASI	
PROBLEMİ	12
4.1 Anadolu Üniversitesi Açıköğretim Sistemi.....	12
4.2 Anadolu Üniversitesi Açıköğretim Sistemi Sınavları	12
4.3 Açıköğretim Sistemleri Sınav Giriş Yeri Problemi	13
4.4 Literatür Araştırması.....	14
5. ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMA	
PROBLEMİ İÇİN ÖNERİLEN MATEMATİKSEL MODEL.....	17
5.1 Matematiksel Model İle Elde Edilen Sayısal Sonuçlar	21
6. ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMASI	
PROBLEMİ İÇİN GELİŞTİRİLEN ÇOK AMAÇLI SEZGİSEL	22
6.1 Çeşitlilik Onarma Alt Algoritması Adımları	23
6.1.1 Birinci oturum için başlangıç çözümün oluşturulması	23
6.1.2 Birinci oturum için oluşturulan başlangıç çözümün onarılması.	23
6.1.3 İkinci oturum için başlangıç çözümün oluşturulması	24
6.1.4 İkinci oturum için oluşturulan başlangıç çözümün onarılması ..	24
6.2 Kapasite Onarma Alt Algoritması	24
6.2.1 Birinci oturum için çözümün oluşturulması	24
6.2.2 İkinci oturum için başlangıç çözümün oluşturulması	24
6.2.3 İkinci oturum için oluşturulan başlangıç çözümün onarılması ..	25
6.3 Sezgisel Algoritma İle Elde Edilen Sayısal Sonuçlar	25
7. SONUÇ VE ÖNERİLER	30
KAYNAKÇA	31
EK-1 GAMS KODU	
EK-2 EXCEL VBA KODU	
ÖZGEÇMİŞ	

TABLULAR DİZİNİ

Tablo 2.1 Literatürdeki Atama Problemleri.....	6
Tablo 3.1 Eğitimsel zaman çizelgeleme problemleri	7
Tablo 5.1 Boyut Analizi	20
Tablo 5.2 Test Problemlerinin Boyutları	21
Tablo 5.3 GAMS Sonuçları	21
Tablo 6.1 Matematiksel Model ve Sezgisel Algoritma Karşılaştırılması.....	26
Tablo 6.2 Mesafe Değerleri Performans Karşılaştırılması	28
Tablo 6.3 Kitapçık Değerleri Performans Karşılaştırılması	29
Tablo 6.4 Bina Doluluk Oranı Değerler Performans Karşılaştırılması	29

ŞEKİLLER DİZİNİ

Şekil 2.1 Düzlemsel üç boyutlu atama problemi	4
Şekil 2.2 Eksensel üç boyutlu atama problemi	5
Şekil 6.1 Sezgisel Algoritmayla Test2'in Çözümü.....	26
Şekil 6.2 Sezgisel Algoritma ile Test3'ün Çözümü.....	27
Şekil 6.3 Sezgisel Algoritma ile Test4'ün Çözümü.....	27
Şekil 6.4 Sezgisel Algoritma ile Test1'in Çözümü	28

1. GİRİŞ

Anadolu Üniversitesi Türkiye'deki en büyük açıköğretim sistemine sahip eğitim kurumudur. Açıköğretim sistemi öğrencileri örgün eğitim almadıkları için kendilerini eğitim sistemin bir parçası olarak görmekte zorlanmaktadır. Bu zorluğu ortadan kaldırmak için öğrencilerin sistemle ilgili şikâyetleri dikkate alınmalı ve gerekli iyileştirmeler yapılmamıştır. Anadolu Üniversitesi Açıköğretim Sistemi'ne kayıtlı öğrencilerin sınav giriş yerleri açıklandıktan sonra sabah ve öğleden sonra oturumlarında sınava girdikleri binalarının bir birlerine uzaklıkları ve genel olarak sınav binalarının ikamet ettikleri adrese uzaklıkları ile şikâyetlerini Anadolu Üniversitesi Açıköğretim Fakültesi çağrı merkezi, sosyal medya hesapları ve forumlar üzerinden aktardıkları tespit edilmiştir. Öğrencilerin Açıköğretim sistemi ile ilgili memnuniyetlerini artırmak bu tezin motivasyon kaynağı olmuş ve tez konusu belirlenmiştir. Bu tez kapsamında çok oturumlu sınavlarda öğrenci sınav yeri (bina) atama problemi ele alınmıştır.

Gerçek hayatta en çok karşılaşılan ve Yöneylem Araştırmasında en çok ilgilenilen problemlerden biri atama problemleridir. Atama problemlerinin özel bir hali olan eğitimsel zaman çizelgeleme eğitim kurumlarındaki ders zaman çizelgeleme ve sınav zaman çizelgeleme ile ilgilenir. Bu tez kapsamında ele alınan problem eğitimsel zaman çizelgeleme başlığı altında incelenmiş ve benzer konularda yapılan çalışmaların sayıca yetersiz olduğu tespit edilmiştir. Çalışmanın özgünlük değerinin artması da bu tezin ikinci motivasyon kaynağı olmuştur.

Çalışmanın ikinci bölümünde atama problemleri ve atama problemlerinin matematiksel modelleri hakkında kısaca bilgi verilmiştir. Üçüncü bölümde ise atama problemlerinin özel bir hali olan eğitimsel zaman çizelgeleme problemlerinin alt konuları olan ders-zaman çizelgeleme ve sınav-zaman çizelgeleme ele alınmıştır.

Dördüncü bölümde ele alınan gerçek hayat problemine sahip kurumun işleyişi ve mevcut durum hakkında kısaca bilgi verildikten sonra literatürdeki ilgili çalışmalar listelenmiştir. Beşinci bölümde ise öğrenci sınav yeri atama problemi için kurulan matematiksel model ve bu modelin çözümüne ait sayısal sonuçlara yer verilmiştir.

Matematiksel modelin sonuçları incelendiğinde problemin çözümünde yetersiz kaldığı görülmüştür. Bu nedenle çok amaçlı sıralama temelli bir sezgisel algoritma geliştirilmiştir. Çalışmanın altıncı bölümünde öğrenci sınav yeri atama problemi için geliştirilen çok amaçlı sezgisel algoritma adımları sıralanıp algoritmaya ait sayısal sonuçlar ve matematiksel modelin çözümleri ve mevcut sistemin çıktıları ile karşılaştırılmış ve gerekli yorumlar yapılmıştır.

2. ATAMA PROBLEMLERİ

Atama problemleri genel olarak n tane öğenin (iş, öğrenci vb.) bir diğer n tane öğeye (iş, görev) atanması ile ilgilenir [1]. Atama problemi ifadesi ilk olarak Wotaw ve Orden 1952 yılındaki çalışmalarında [2] kullanılmıştır. 1952 yılından bugüne araştırmacılar atama problemi ile yakından ilgilenmiş ve literatüre birçok farklı konuda çalışmalar kazandırmıştır.

2.1 Doğrusal Atama Problemleri

Doğrusal atama probleminde belirli sayıda kaynağın belirli sayıda faaliyete en düşük maliyetle atanması amaçlanır. Örnek bir doğrusal atama modeli aşağıda verilmiştir.

Parametre

c_{ij} = i . kaynağın j . faaliyete atanmasının maliyet

Karar Değişkeni

x_{ij} = 1 i . kaynak j . faaliyete atanırsa; 0 diğer durumlar

Amaç Fonksiyonu

$$\text{Enk } Z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2.1)$$

Kısıtlar

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i, i = 1, \dots, n \quad (2.2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j, j = 1, \dots, n \quad (2.3)$$

$$x_{ij} = 0 \text{ ya da } 1 \quad \forall (i, j) \quad (2.4)$$

(2.2) 'nolu kısıt her bir kaynağın sadece bir faaliyete atanmasını sağlarken (2.3) 'nolu kısıt her bir faaliyete sadece bir kaynağın atanmasını sağlar.

2.2 Doğrusal Olmayan Atama Problemleri

Amaç fonksiyonu ve/veya kısıtları doğrusal yapıda olmayan olan atama problemleri bu sınıfta yer alır. En bilinen örneği karesel atama problemleridir. Karesel atama problemi n sayısı m sayısında büyük eşit olduğu sürece m tane tesisin n tane yere atanması problemidir [3]. Karesel atama probleminde amaç tesisler arasındaki etkileşimden kaynaklanan uzaklık ve akış maliyetlerini en küçükmektir. Karesel atama problemlerine ait örnek bir matematiksel model şu şekildedir:

Parametreler

f_{ij} = *i. tesis ile j. tesis arasındaki akış*

d_{kl} = *k.yer ve l.yer arasındaki uzaklık*

Karar Değişkeni

x_{ij} = *1 i. tesis j.yere atanırsa; 0 diğer durumlar*

Amaç Fonksiyonu

$$\text{Enk } Z = \sum_{i=1}^m \sum_{j=1}^n \sum_{k=1}^m \sum_{l=1}^N f_{ij} d_{kl} x_{ik} x_{jl} \quad (2.5)$$

Kısıtlar

$$\sum_{i=1}^M x_{ij} \leq 1 \quad \forall j, j = 1, \dots, n \quad (2.6)$$

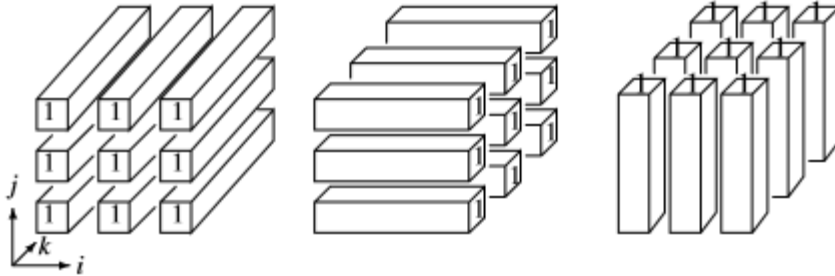
$$\sum_{j=1}^N x_{ij} = 1 \quad \forall i, i = 1, \dots, m \quad (2.7)$$

$$x_{ij} = 0 \text{ ya da } 1 \quad \forall (i, j) \quad (2.8)$$

2.3 Çok Boyutlu Atama Problemleri

Şu ana kadar incelenen modellerde öge kümeleri iki tanedir ancak öge küme sayısının üç ya da daha fazla olduğu atama problemleri de mevcuttur. Bu tür modeller de kendi içinde düzlemsel ve eksensel çok boyutlu atama problemleri olarak ikiye ayrılır.

***Düzlemsel çok boyutlu atama problemleri:** Şekil 2.1’de 3 boyutlu düzlemsel atama problemi görselleştirilmiştir. r sayısı s ’den büyük veya eşit ve s sayısının t ’den büyük veya eşit olduğu koşullarda; r tane ögenin t zaman aralığı boyunca s tane öge ile eşleşmesidir. Amaç eşleşmeden kaynaklanacak maliyeti en küçükmektir.



Şekil 2.1 Düzlemsel üç boyutlu atama problemi

Kaynak: Burkard, 2008, s.9

3 boyutlu düzlemsel atama problemlerine ait örnek bir matematiksel model şu şekildedir:

Parametre

$c_{ijk} = i.$ öğenin k zaman aralığı boyunca $j.$ öğe ile eşleşmesi maliyeti

Karar Değişkeni

$x_{ijk} = 1$ $i.$ öğe $k.$ zaman aralığı boyunca $j.$ öğe ile eşleşirse 1; 0 diğer durumlar

Amaç Fonksiyonu

$$Enk Z = \sum_{i=1}^r \sum_{j=1}^s \sum_{k=1}^t c_{ijk} x_{ijk} \quad (2.9)$$

Kısıtlar

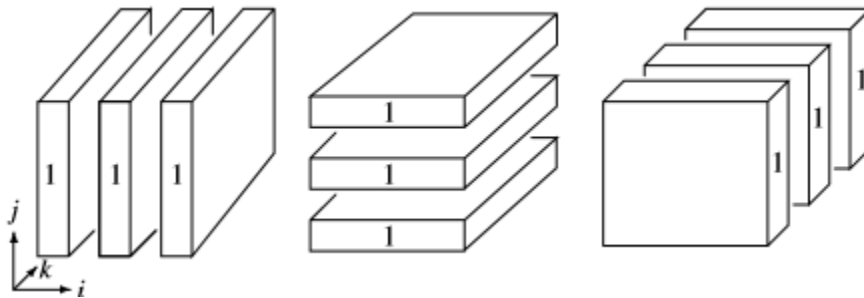
$$\sum_{i=1}^r x_{ijk} = 1 \quad \forall(j, k) \quad j = 1, \dots, s; k = 1, \dots, t \quad (2.10)$$

$$\sum_{j=1}^s x_{ijk} \leq 1 \quad \forall(i, k) \quad i = 1, \dots, r; k = 1, \dots, t \quad (2.11)$$

$$\sum_{k=1}^t x_{ijk} \leq 1 \quad \forall(i, j) \quad i = 1, \dots, r; j = 1, \dots, s \quad (2.12)$$

$$x_{ijk} = 0 \text{ ya da } 1 \quad \forall(i, j, k) \quad (2.13)$$

* **Eksensel çok boyutlu atama problemleri:** Şekil 2.2’ de 3 boyutlu eksensel atama problemi görselleştirilmiştir. r sayısı q ’ dan büyük veya eşit ve q sayısının p ’den büyük veya eşit olduğu koşullarda; p tane öğenin q ve r tane öğe ile eşleşmesidir. Amaç eşleşmeden kaynaklanacak maliyeti en küçüklemektir.



Şekil 2.2 Eksensel üç boyutlu atama problemi

Kaynak: Burkard, 2008, s.9

3 boyutlu eksensel atama problemlerine ait örnek bir matematiksel model şu şekildedir:

Parametre

$c_{ijk} = i.$ öğenin $j.$ ve $k.$ öge ile eşleşmesi maliyeti

Karar Değişkeni

$x_{ijk} = 1$ $i.$ öge $j.$ öge ve $k.$ öge ile eşleşirse 1; 0 diğer durumlar

Amaç Fonksiyonu

$$\text{Enk } Z = \sum_{i=1}^q \sum_{j=1}^p \sum_{k=1}^r c_{ijk} x_{ijk} \quad (2.14)$$

Kısıtlar

$$\sum_{i=1}^q \sum_{j=1}^p x_{ijk} \leq 1 \quad \forall(k) \quad i = 1, \dots, q; j = 1, \dots, p \quad (2.15)$$

$$\sum_{j=1}^p \sum_{k=1}^r x_{ijk} \leq 1 \quad \forall(i) \quad j = 1, \dots, p; k = 1, \dots, r \quad (2.16)$$

$$\sum_{i=1}^q \sum_{k=1}^r x_{ijk} = 1 \quad \forall(j) \quad i = 1, \dots, p; k = 1, \dots, r \quad (2.17)$$

$$x_{ijk} = 0 \text{ ya da } 1 \quad \forall(i, j, k) \quad (2.18)$$

Literatürde en çok karşılaşılan atama problemlerinin bahsedilen kategorilere göre sınıflandırılması Tablo 2.1’de verilmiştir.

Tablo 2.1 Literatürdeki Atama Problemleri

Kategori	Problem
Doğrusal Atama Problemleri	<ul style="list-style-type: none">• İşçi-makine atama• İş-makine Atama• Personel-iş Atama• Dengeli evlilik problemi
Doğrusal Olmayan Atama Problemleri	<ul style="list-style-type: none">• Karesel atama• Silah-hedef atama• Kapı Atama• Filo Atama• Çizelgeleme
Çok Boyutlu Atama Problemleri	<ul style="list-style-type: none">• Çizelgeleme• İş-işçi-makine atama• Kapı Atama• Filo Atama

3. EĞİTİMSEL ZAMAN ÇİZELGELEME PROBLEMLERİ

Bir tür atama problemi olan zaman çizelgeleme problemleri önceden belirlenmiş kısıtlar altında belirli sayıdaki faaliyetin sınırlı kaynaklara ve zaman dilimlerine ataması ile ilgilenir. Eğitim kurumlarındaki zaman çizelgelerinde ders, derslik gibi kaynaklar veya faaliyetler ile öğrenci, öğretici, gözetmen gibi kişiler yer almaktadır [4]. Eğitim kurumlarındaki çizelgeleme faaliyetleri eğitimsel zaman çizelgeleme başlığı altında incelenmektedir. Eğitimsel zaman çizelgeleme problemlerinde

- önceden belirlenmiş kısıtlar; klasik atama problemi kısıtları, derslik kapasiteleri, kurumun özel tercihleri, kurum çalışanlarının özel tercihleri, öğrencilerin özel tercihleri ve aynı kaynakları kullanan farklı faaliyetlerin aynı zaman dilimine atanmaması gibi koşullardır.

- faaliyetler; dersler, proje sunumları, sınavlar vb.
- kaynaklar; derslikler, laboratuvar, gözetmenler, öğretici vb.

olarak ele alınır. Öztürk (2010) çalışmasında [4] Tablo 3.1'deki gibi eğitimsel zaman çizelgeleme problemlerini ders zaman çizelgeleme ve sınav zaman çizelgeleme olarak iki başlık altında sınıflandırmıştır.

Tablo 3.1 Eğitimsel zaman çizelgeleme problemleri

Ders Zaman Çizelgeleme	Sınav Zaman Çizelgeleme
<ul style="list-style-type: none">• Ders Zaman Dilimi Ataması• Ders Derslik Ataması• Ders Öğretici Ataması• Ders Öğretici Derslik Zaman Dilimi Ataması	<ul style="list-style-type: none">• Sınav Zaman Dilimi Ataması• Sınav Derslik Ataması• Sınav Zaman Dilimi Derslik Ataması• Sınav Gözetmen Ataması

Ders ve sınav çizelgeleme problemleri arasında önemli farklılıklar vardır. Sınav çizelgeleme problemlerinde sınavlar birden fazla dersliğe atanabilirken, ders çizelgeleme problemlerinde dersler sadece bir sınıfa atanabilmektedir. Sınav çizelgeleme problemlerinde ardışık zaman dilimindeki sınav sayısı en küçüklenmeye çalışılırken, ders çizelgeleme problemlerinde öğrencinin bir günde alacağı ders sayısının birden fazla olması istenmektedir [5].

3.1 Ders Zaman Çizelgeleme Problemleri

Ders çizelgeleme problemi çok boyutlu bir atama problemi olup, öğrenci veya öğreticinin ders, derslik ve zaman dilimlerine atanması ile ilgilidir [6]. Ders zaman çizelgeleme esnek ve sıkı kısıtlar altında gerçekleştirilir. Eğitim kurumunun derecesine göre (ilköğretim, lise ve üniversite) esnek ve sıkı kısıtlar değişebilmektedir. Üniversite ders zaman çizelgeleme problemine ait genel sıkı ve esnek kısıtlar izleyen şekilde verilebilir [7]:

Sıkı Kısıtlar:

- Öğrenci veya öğretici bir zaman dilimde en fazla bir dersliğe atanabilir.
- Bir ders bir zaman dilimde en fazla bir dersliğe atanabilir.
- Öğrenci ve öğretici bir zaman dilimde bir derslikte en fazla bir derse atanabilir.
- Bir derslikte bir zaman dilimde en fazla bir öğretici ve bir dersi alan öğrenci grubu bulunabilir.
- Öğretici bir zaman dilimde en fazla bir dersi alan öğrenci grubuna ders verebilir.
- Bir zaman dilimde bir dersliğe atanan öğrenci sayısı derslik kapasitesini geçemez.

Esnek Kısıtlar:

- Dersler önceden belirlenen dersliklerde yapılabilir.
- Dersler önceden belirlenen öğretmenlere atanabilir.
- Öğrenci ve öğretmenler için günlük maksimum eğitim saati mümkün olduğunca aşılmamalı [8].

Ders zaman çizelgeleme çalışmalarının bazılarında derslik kapasitelerinin sınırsız olduğu varsayımı bulunmaktadır [9]. Eğer kapasiteler için böyle bir varsayım varsa kapasite ile ilgili olan sıkı kısıt çıkarılabilir.

Ders zaman çizelgeleme problemleri gerçek hayat problemlerinin özellikleri nedeniyle çalışmalarda ders-zaman dilimi, ders-derslik, ders-öğretici ve öğrenci-ders atama alt boyutları ile de ele alınmıştır.

3.2 Sınav Zaman Çizelgeleme Problemi

En yalın tanımı ile sınav çizelgeleme problemi; öğrencilerin aynı zaman diliminde birden fazla sınavı olmayacak şekilde sınavların zaman dilimlerine atanmasıdır [10]. Sınav çizelgeleme de ders çizelgeleme problemi gibi eğitim kurumunun derecesine özgü esnek ve sıkı kısıtlar altında gerçekleştirilir. Üniversite ders zaman çizelgeleme problemine ait genel sıkı ve esnek kısıtlar şu şekildedir:

Sıkı Kısıtlar:

- Bir öğrenci bir zaman diliminde en fazla bir sınava atanabilir. Bir sınav yalnızca bir zaman dilimine atanabilir.
- Bir sınav bir zaman diliminde en fazla bir dersliğe atanabilir.
- Bir gözetmen bir zaman diliminde en fazla bir sınava atabilir.
- Bir zaman diliminde sınavın atandığı ders/derslik kapasiteleri sınava girecek öğrenci sayısını geçemez.

Esnek Kısıtlar:

- Bir sınav mümkün daha önceden belirlenen derslikte yapılabilir.
- Bir sınav daha önceden belirlenen bir zaman dilimine atanabilir.
- Öğrencinin iki sınavı arasında en az belirli bir süre bulunmalıdır [7].
- Bir sınavın başka bir sınavdan önce ya da sonra atanması gerekebilir [11].

Sınav zaman çizelgeleme çalışmalarını bir kısmında bir sınavın bir zaman diliminde sadece bir dersliğe atanması sıkı kısıtı kullanılırken, bazı çalışmalarda sınavlar bir zaman diliminde birden fazla dersliğe yayılabilmektedir. Herhangi bir zaman diliminde bir derslikte sadece bir sınavın yapılabildiği problemlerde, sınavların dersliklere atanması polinom zamanda çözülebilirken, bir derslikte birden fazla sınavın yapılması durumunda problem NP-tam olmaktadır [12]. Sınav zaman çizelgeleme problemleri gerçek hayat problemlerinin özellikleri ve problem boyutları nedeniyle; sınav zaman dilimi ataması, sınav derslik ataması, sınav gözetmen ataması alt boyutları ile incelenmektedir. Bu tez kapsamında ele alına ardışık oturumlu öğrenci atama problemi sınav derslik atama başlığında değerlendirilebilir. Adından da anlaşılacağı üzere sınav derslik atama problemlerinde zaman boyutu yoktur.

Sınavlar zaman dilimleri düşünülmezsizin dersliklere atanır. Bu tip problemleri ele alan çalışmaların çoğunda zaman çizelgeleme anahtar kelimesi kullanılmıştır [13].

3.3 Çözüm Yaklaşımları ve Teknikleri

Eğitimsel zaman çizelgeleme problemlerinin çözümü için matematiksel programlama temelli yaklaşımlar, sezgisel algoritmalar, sezgisel üstü algoritmalar ve hazır paket programların kullanıldığı görülmektedir.

3.3.1 Matematiksel temelli yaklaşımlar

Matematiksel temelli yaklaşımlar küçük boyutlu eğitimsel zaman çizelgeleme problemleri için iyi sonuç verseler de büyük boyutlu problemleri etkin zamanlarda çözememektedir. Eğitimsel zaman çizelgeleme problemlerinde kullanılan matematiksel temelli yaklaşımları aşağıdaki gibi özetleyebiliriz:

- Doğrusal Modeller [14], [15]
- Doğrusal Olmayan Modeller
- Tamsayılı Modeller [16]
- 0-1 Tamsayılı Modeller [17]
- Karma Tamsayılı Modeller [18], [19]
- Hedef Programlama [20]
- Bulanık Hedef Programlama [21]

3.3.2 Sezgisel algoritmalar

Sezgisel algoritmalar en iyi sonucu garanti etmeyen, makul sürelerde iyiye yakın çözümler veren ve problemlere özgü geliştirilen algoritmalarlardır. Matematiksel temelli yaklaşımlarla çözülemeyecek boyutlardaki problemler sezgisel algoritmalarla çözülebilmektedir. Eğitimsel zaman çizelgeleme kapsamına ele alınan problemlerin bir bölümü ([13], [22], [9]) eğitim kurumuna özgü özelliklere sahip olduğu çalışmalarda problemlere özgü sezgisel algoritmalar geliştirilmiştir.

3.3.3 Metasezgisel algoritmalar

Metasezgisel algoritmalar problem yapısından bağımsız bir şekilde alt seviye sezgisellere rehberlik ederek kısa sürelerde kaliteli çözümler üretir. Metasezgisel yöntemler yerel en iyi çözümlere takılmaktan kaçabildikleri için büyük boyutlu eğitimsel zaman çizelgeleme problemlerinin çözümünde sezgisel algoritmalarından daha iyi sonuçlar

vermektedir. Eğitimsel çizelgeleme problemlerinde kullanılan metasezgisel yöntemler yerel arama ve topluluk temelli olarak ikiye ayrılabilir [23]:

Yerel Arama Temelli Metasezgiseller:

- Yasaklı Arama
- Tavlama Benzetimi
- Değişken Komşu Arama Algoritması (VNS)
- Yinemeli Komşu Arama Algoritması
- Büyük Komşu Arama Algoritması (LNS)

Topluluk Temelli Metasezgiseller:

- Genetik Algoritmalar
- Memetic Algoritmalar
- Karınca Kolonisi
- Yapay Bağışıklık Algoritmaları
- Arı Kolonisi Algoritmaları
- Parçacık Sürü Algoritması

3.3.4 Graf renklendirme

Graf renklendirme, graf üzerinde birbirine komşu olan düğümlere farklı renk atama işlemidir. Amaç, en az sayıda renk kullanılarak tüm düğümlere komşularından farklı birer renk vermektir. Eğitimsel zaman çizelgeleme problemlerinde her bir zaman dilimi bir renge, her ders ya da sınav bir köşeye denk gelmektedir [24].

3.3.5 Sezgisel üstü (hyper heuristic) algoritmalar

Sezgisel üstü algoritmaların arama uzayı sezgisel algoritmaların arama uzaylarının birleşiminden oluşmaktadır. Sezgisel üstü algoritmalar her yineleme değerinde en iyi sonucu veren sezgisel algoritmayı seçerek çözüme devam ederler [25].

4. ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMASI PROBLEMİ

4.1 Anadolu Üniversitesi Açıköğretim Sistemi

Anadolu Üniversitesi 20 Temmuz 1982’de çıkartılan 41 sayılı Kanun Hükmündeki Kararname ile Türkiye’de uzaktan öğretim hizmeti ile görevlendirildiğinden bu yana ülkemizde çağdaş anlamda uzaktan öğretim hizmeti vermektedir. Anadolu Üniversitesi Açıköğretim Sistemi, sahip olduğu teknik alt yapı, kalite düzeyi yüksek ve verimli çalışabilen üretken insan kaynağı ile kendini sürekli yenileyerek uyguladığı açıköğretim sistemi ile Türk Yükseköğretim Sisteminde ülkemizde yükseköğretim alanında birçok ilkleri ve yenilikleri uygulamaya koymaktadır. Bu yönü ile dünyanın mega üniversiteleri arasında yer almaktadır. Anadolu Üniversitesi Açıköğretim Sistemi 17 lisans, 34 önlisans programıyla hizmet vermektedir [26].

4.2 Anadolu Üniversitesi Açıköğretim Sistemi Sınavları

2016 yılı itibariyle Anadolu Üniversitesi Açıköğretim Sisteminin toplamda 17 farklı lisans ve 34 farklı önlisans programına kayıtlı yaklaşık 1 milyon 400 bin öğrenciye sahiptir. Güz döneminde güz dönemi ara sınavı ve dönem sonu sınavı, bahar döneminde bahar dönemi ara sınavı ve dönem sonu sınavı ve bahar dönemi bittikten sonra yapılan tek ders sınavı olmak üzere öğrencilere beş sınav yapılmaktadır. Tek ders sınavı dışındaki tüm sınavlar Cumartesi ve Pazar sabah bir ve öğleden sonra da bir oturum olmak üzere toplam 4 oturumda gerçekleştirilmektedir. Her dönem öncesinde Anadolu Üniversitesi Test Araştırma Birimi tarafından sınav oturum düzeni hazırlanır. Hazırlanan bu oturum düzeninde hangi kitapçıklar hangi derslerin olacağı ve hangi kitapçıklarının sınavının hangi oturumda yapılacağı kararlaştırılır. Mevcut durumda bir kitapçıkta en fazla 5 ders yer almaktadır. Her dersin sınav süresinin 30 dakika olduğu için bir oturum en fazla 150 dakika sürmektedir. Güz döneminde; Cumartesi ve Pazar sabah oturumunda 1. ve 5. yarıyılların, öğleden sonra oturumunda 3. ve 7. yarıyılların; bahar döneminde Cumartesi ve Pazar sabah oturumunda 2. ve 6. yarıyılların, öğleden sonra oturumunda 4. ve 8. yarıyılların sınavları yapılmaktadır.

Sınavlar 81 il 17 ilçe olmak üzere toplamda 98 merkezde yapılmaktadır. Tek ders sınavı hariç yapılan sınavlarda ortalama 9666 adet bina, 144232 adet sınav salonu ve

431445 görevli kullanılmaktadır. Sınav için kullanılabilir binalar ve salonlar her bir sınav merkezinin koordinatörlükleri tarafından belirlenir.

Öğrencilerin hangi oturumda hangi binada ve salonda sınava gireceği Anadolu Üniversitesi Bilgisayar Araştırma ve Uygulama Merkezi tarafından belirlenmektedir. Öğrencilerin sınav yerleri belirlenirken şu faktörler dikkate alınmaktadır:

- Öğrencinin bedensel engel durumu
- Öğrencinin sınav merkezi tercihi
- Görevli sayısını azaltmak için her bir oturumda bina kapasitelerinin maksimum oranda kullanması
- Aynı salonda sınava girecek öğrencilerin sınav sürelerinin mümkün olduğunca aynı olması ya da birbirine yakın olması
- Kitapçık paketleme operasyonunu hızlandırmak ve binalardaki yedek kitapçık sayılarını azaltmak için bir binadaki kitapçık çeşitliliğinin mümkün olduğunca en küçüklenmesi.

4.3 Açıköğretim Sistemleri Sınav Giriş Yeri Problemi

Mevcut sistemde, aynı gün içerisinde 2 oturuma giren öğrencilerin sabah ve öğleden sonraki oturumlarında atandıkları binaları birbirine mesafe olarak çok uzak olabilmektedir. Özellikle İstanbul, Ankara ve İzmir gibi büyük sınav merkezlerindeki trafik yoğunluğu da göz önüne alındığında öğrencinin öğleden sonraki oturumda sınava gireceği binaya ulaşması daha da zorlaşmaktadır. Öğrenciler sınav giriş yerleri açıklandıktan sonra bu duruma tepkilerini sosyal medya aracılığı ile göstermektedir. Öğrencilerin konu hakkında attıkları örnek tweetlere Şekil 4.1’ de yer verilmiştir. Ayrıca çağrı merkezine bu konu hakkında gelen telefon sayısı da oldukça fazladır.



Şekil 4.1 Öğrenci tweetleri

4.4 Literatür Araştırması

Öğrencilerin bir eğitim programı kapsamında girmeleri gereken sınavlarına hangi kampüste, hangi binada, hangi derslikte ve hangi sıra numarasında girmeleri gerektiğini ele alan problemler bu çalışma kapsamında öğrenci sınav yeri atama problemi olarak ele alınmaktadır. Literatür incelendiğinde öğrenci sınav yeri atama problemine çok benzeyen problemin sınav-derslik atama ve ders-derslik atama problemi olduğu görülmüştür. Ancak sınav-derslik atama problemlerinin ve ders-derslik matematiksel modeli öğrenci sınav yeri belirleme problemi ile örtüşmemektedir. Bu probleme en çok benzeyen çalışmalar şu şekilde listelenebilir.

Dammak ve arkadaşları 2006 yılındaki çalışmalarında [13] sınırlı kapasitelere sahip derslikleri ve bir dersliğe birden fazla sınavın atanabildiği problemleri ele

almışlardır. Derslik atama probleminin ulaştırma problemi olarak düşünölebileceğini belirtip matematiksel modeli ulaştırma modeli problemlerine benzeterek kurmuşlardır. Yine aynı çalışma kapsamında büyük boyutlu problemler için maksimum-boyut atama sezgiseli adını verdikleri bir sezgisel algoritma geliştirmişlerdir.

Kahar ve Gendall 2010 [22] yılındaki çalışmalarında Universiti Malaysia Pahang'daki sınav çizelgeleme probleminden yola çıkarak kurdukları matematiksel modelde öğrencilerin sınavlarını sınav dönemine yaymayı, sınavların bir zaman diliminde yapıldığı derslik sayısını (sınavların yayılmasını) en küçöklemeyi ve aynı anda farklı dersliklerde yapılan sınavların yapıldığı derslikler arasındaki mesafeleri en küçöklemeye çalışmışlardır. Sınavların sınav dönemine yayılması için Carter ve arkadaşlarının [27] geliştirdikleri yakınlık değerini ceza maliyetleri ile çarpma yoluna gitmişlerdir. Problemin çözümü için en büyük derece, en büyük ağırlıklandırılmış derece, doygunluk derecesi ve en büyük kayıt gibi graf sezgiseli yaklaşımlarından yola çıkarak kendileri bir sezgisel geliştirmişlerdir.

Sarin ve arkadaşları 2010 yılındaki çalışmalarında ders çizelgeleme problemi için üç alt problemden oluşan bir matematiksel model geliştirmişlerdir. Virginia Polytechnic Institute and State University'deki özel kısıtlar altında dersliklerden yararlanma oranını en büyükmeye ve öğreticilerin ofisleri ve ders verdikleri derslikler arasındaki mesafeyi en küçöklemeye çalışmışlardır. Alt problemlerden birinin çözümünde Bender's ayrıştırma algoritmasını kullanmışlardır.

Ayob ve Malik 2011 yılındaki çalışmalarında [9] Dammak ve arkadaşlarının yaptığı gibi sınırlı kapasitelere sahip derslikleri ve bir dersliğe birden fazla sınavın atanabildiği problemi ele almışlardır. Problem verisi olarak Universiti Kebangsaan Malaysia(UKM) ait verileri kullanmışlardır. Sınavların zaman dilimleri ve öğrencilerin hangi sınavlara ve zaman dilimlerine atandıkları veriler de mevcuttur. Bu çalışmada kurulan matematiksel modelde iki amaç vardır Birincisi ardışık zaman dilimlerinde sınava giren ve bu zaman dilimleri arasında bir derslikten başka bir dersliğe hareket eden toplam öğrenci sayısını en küçökleme; ikincisi ise bir zaman diliminde bir sınavın atandığı farklı derslik sayısını en küçökleme. Modelin amacı derslik ceza maliyeti olarak adlandırılmıştır. Bu tez kapsamında ele alınan probleme en çok benzeyen çalışma Ayob ve Malik'in çalışmasıdır. Aynı çalışma kapsamında bu tür problemlerin çözümü için en uygun derslik atama adında bir sezgisel algoritma geliştirilmiştir. Bu algoritmada ilk

olarak her bir zaman dilimi için derslikler kapasitelerine ve sınavlarda ilgili derse kayıtlı olan öğrenci sayılarına göre büyükten küçüğe azalacak şekilde sıralanır. Özel bir koşula sahip sınavlar varsa ilk olarak bu sınavlar dersliklere atanır ve yapılan sıralamalar güncellenir. Böyle bir durum yoksa her bir zaman dilimi için sıralamada ilk sırada olan sınavlar sıralamaya göre eşleştikleri dersliklere atanır. Her bir atamadan sonra atamaya uygun olan dersliklerin sıralaması güncellenir. Bu işlem tüm sınavlar dersliklere atanana kadar devam eder. Geliştirilen sezgisele ait sonuçlara ise Ayob ve Malik başka bir çalışmalarında yer vereceklerini ifade etmişlerdir.

Vermuyten ve arkadaşları 2015 yılındaki çalışmalarında [28] dersliklerin binaların farklı katlarına dağıldığı üniversitelerde ardışık dersler arasında binanın içinde oluşan öğrenci akışı nedeniyle derslerin başlama saatlerinin gecikmesinden yola çıkmışlardır. Ardışık dersler arasında bina içerisinde öğrencilerden kaynaklanan akışı en küçüklemek için iki aşamalı tamsayı bir model geliştirmişlerdir. Modelin birinci aşamasında öğreticileri zaman dilimlerine ve dersliklere atarken, öğreticilerden ve eğitimsel amaçlardan kaynaklanan tercihler en küçüklenmiştir. Modelin ikinci aşamasında ise öğreticiler dersliklere ve zaman dilimlerine tekrar atanarak öğrenci akışları en küçüklenmiştir. Test problemlerinden alınan sonuçlara göre iki aşamalı modelin tek aşamalı modele göre daha fazla uygun çözüm bulduğu belirtilmiştir.

5. ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMA PROBLEMİ İÇİN ÖNERİLEN MATEMATİKSEL MODEL

Bu çalışma kapsamında ele alınan problem daha önce modellenmemiş bir gerçek hayat problemidir. Problem modellenirken mevcut durumda göz önüne alınan; aynı salonda sınava girecek öğrencilerin sınav sürelerinin mümkün olduğunca aynı olması ya da birbirine yakın olması faktörleri göz ardı edilmiştir.

Çok oturumlu sınavlarda öğrenci bina ataması probleminin için önerilen çok amaçlı doğrusal olmayan karma tamsayılı matematiksel model izleyen şekilde verilmiştir.

İndis Kümeleri

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

Oturum indisleri kümesi: $J = \{1, \dots, 4\}$

Bina indisleri kümesi $K = \{1, \dots, p\}$

Bina indisleri kümesi $L = \{1, \dots, p\}$

Kitapçıklar indisleri kümesi $H = \{1, \dots, o\}$

Parametreler

d_{kl} = binalar arası uzaklık matrisi

q_k = bina kapasiteleri

KT_{hj} = kitapçık oturum matrisi

OK_{ih} = öğrenci kitapçık matrisi

Karar Değişkenleri

$x_{ijk} = 1$ i. öğrenci j. oturumda k. binaya atanırsa ; 0 dd

$y_{ij} = 1$ i. öğrenci j. oturuma atanırsa ; 0 dd

$b_{jk} = 1$ j. oturum k. binaya atanırsa ; 0 dd

$c_{jk} = j.$ oturumda k. binada kullanılmayan kapasite

$p_{kh} = k.$ binaya atanan h. kitapçık sayısı

$r_{kh} = 1$ h. kitapçık k. binaya atanırsa ; 0 dd

$$\sum_{j=1}^2 y_{ij} \geq 1, \forall i \quad (5.1)$$

$$y_{ij} = \sum_{h=1}^o OK_{ih} KT_{hj}, \forall(i, j) \quad (5.2)$$

$$\sum_{k=1}^p x_{ijk} \leq 1, \forall(i, j) \quad (5.3)$$

$$\sum_{j=1}^4 b_{jk} \leq 4, \forall k \quad (5.4)$$

$$\sum_{i=1}^m x_{ijk} \leq q_k, \forall(j, k) \quad (5.5)$$

$$y_{ij} \leq \sum_{k=1}^p x_{ijk} \leq y_{ij} * M, \forall(i, j) \quad (5.6)$$

$$b_{jk} \leq \sum_{i=1}^m x_{ijk} \leq b_{jk} * M, \forall(j, k) \quad (5.7)$$

$$c_{jk} = b_{jk} * q_k - \sum_{i=1}^m x_{ijk}, \forall(j, k) \quad (5.8)$$

$$p_{kh} = \sum_{j=1}^4 \sum_{i=1}^m x_{ijk} * OK_{ih} * KT_{hj}, \forall(k, h) \quad (5.9)$$

$$r_{kh} \leq p_{kh} \leq r_{kh} * M, \forall(k, h) \quad (5.10)$$

$$x_{ijk}, y_{ij}, b_{jk}, r_{kh} \in \{0,1\}, c_{jk} \geq 0 \quad (5.11)$$

$$F_1 = \left(\sum_{i=1}^m \sum_{k=1}^p \sum_{l=1, k \neq l}^p x_{i1k} * x_{i2l} * d_{kl} + \sum_{i=1}^m \sum_{k=1}^p \sum_{l=1, k \neq l}^p x_{i3k} * x_{i4l} * d_{kl} \right) / \max(d_{kl}) \quad (5.12)$$

$$F_2 = \sum_{k=1}^p \sum_{j=1}^4 (c_{jk} / q_k) \quad (5.13)$$

$$F_3 = \sum_{k=1}^p \sum_{h=1}^0 (r_{hk} / o) \quad (5.14)$$

Kısıtları altında

$$\text{Min } Z = F_1 + F_2 + F_3 \quad (5.15)$$

(5.1) nolu kısıt her öğrencinin en az bir oturuma atanmasını, (5.2) nolu kısıt her öğrencinin oturum düzeninde belirlenen kitapçık aldığı oturuma atanmasını, (5.3) nolu kısıtı bir öğrencinin bir oturumda en fazla bir binaya atanmasını sağlar. (5.4) nolu kısıt bir binanın bir sınavda en fazla iki oturuma atanmasını, (5.5) nolu kısıt bir oturumda bir binaya atanan öğrenci sayısı binanın kapasitesine eşit ya da binanın kapasitesinden küçük olmasını sağlarken (5.6), (5.7), (5.8), (5.9) ve (5.10) nolu kısıtlar karar değişkenleri arasındaki ilişkiyi sağlar.

Kurulan matematiksel model çok amaçlı doğrusal olmayan karma tam sayılı modeller sınıfına girmektedir. Model; üç tane amaç fonksiyonuna sahip olduğu için çok amaçlı, (5.12) nolu amaç fonksiyonun yapısından dolayı doğrusal olmayan ve karar değişkelerinin yapısından dolayı karma tamsayılı özelliklerine sahiptir.

Problemde aynı anda eniyilenmesi istenen amaçlar; öğrencilerin ardışık oturumlarda atandıkları binalar arası mesafenin, kullanılmayan bina kapasitesinin ve binalara atanan kitapçık çeşitliği sayısının enküçüklenmesidir. Bu amaçlar sırasıyla (5.12), (5.13) ve (5.14) nolu eşitliklerde ifade edilmiştir. Amaç fonksiyonlarının birimleri farklı olduğu için normalleştirme yoluna gidilmiştir. Normalleştirme yapılırken; her bir amaç fonksiyonu alabileceği en büyük değere bölünmüştür. Örneğin, ardışık oturumlara giren bir öğrencinin kat edeceği mesafe en fazla en uzak binalar arasındaki mesafe

olabilir. Bu yüzden (5.12) nolu ifade uzaklık matrisindeki maksimum uzaklık değerine bölünerek normalleştirilmiştir.

Öğrenci sınav giriş yeri belirleme problemi için kurulan matematiksel modelin karmaşıklığını belirtmek için, modelin boyut analizi Tablo 4.1 'de verilmiştir. Bu çalışma kapsamında mevcut sistemden yola çıkılarak hazırlanan test problemlerinin boyutları da Tablo 4.2' de verilmiştir.

Tablo 5.1 Boyut Analizi

Değişken	İndisler	Toplam değişken sayısı
x_{ijk}	i,j,k	$i*j*k$
y_{ij}	i,j	$i*j$
b_{jk}	j,k	$j*k$
c_{jk}	j,k	$j*k$
p_{kh}	k,h	$k*h$
r_{kh}	k,h	$k*h$
Kısıt no	İndis	Toplam Kısıt Sayısı
1	i	i
2	i,j	$i*j$
3	i,j	$i*j$
4	k	k
5	k	k
6	i,j	$i*j$
7	j,k	$j*k$
8	j,k	$j*k$
9	k,h	$k*h$
10	k,h	$k*h$
Toplam Değişken Sayısı		$ijk+ij+2jk+2kh$
Toplam Kısıt Sayısı		$i+2k+3ij+2k+2jk+2kh$

Tablo 5.2 Test Problemlerinin Boyutları

	Test 1	Test2	Test3	Test4
Öğrenci Sayısı	21408	257	257	50
Oturum Sayısı	2	2	2	2
Bina Sayısı	62	62	12	5
Kitapçık çeşiti	115	124	124	20
Toplam Değişken sayısı	2711916	48006	9706	820
Toplam Kısıt Sayısı	164612	17671	4871	590

5.1 Matematiksel Model İle Elde Edilen Sayısal Sonuçlar

Test problemleri The General Algebraic Modeling System (GAMS) programının 23.3.3 sürümü ile çalıştırılmıştır. Kurulan matematiksel modelin doğrusal olmayan karma tamsayılı yapısından dolayı MINLP çözümleri kullanılmıştır. SBB ve DICOPT çözümleri karma tamsayılı doğrusal olmayan modellerde hızlı sonuç vermektedir ancak SBB ve DICOPT çözümlerinin çözemediği bazı modeller BARON ve AlphaECP çözümleri tarafından çözülmektedir. Bu çalışmada tam sürümlerine ulaşılabilen DICOPT, BARON ve SBB çözümleri kullanılmıştır. Tablo 5.3 incelendiğinde GAMS programının test1 problemi için tüm çözümlerde, test 2 problemi için BARON ve SBB çözümlerinde limit aşımına bağlı olarak sonuç vermediği görülmüştür. Tablo 5.3'deki amaç fonksiyonu değerleri bulunan en iyi çözümleri temsil etmektedir.

Tablo 5.3 GAMS Sonuçları

Çözümler	Test1		Test2		Test3		Test4	
	Çözüm süresi (saniye)	Amaç fonksiyonu değeri	Çözüm süresi (saniye)	Amaç fonksiyonu değeri	Çözüm süresi (saniye)	Amaç fonksiyonu değeri	Çözüm süresi (saniye)	Amaç fonksiyonu değeri
BARON	-	-	-	-	1489	44,342	1005	1,35
SBB	-	-	-	-	1441	44,342	62	21,6
DICOPT	-	-	514	2,228	5	2,672	1	1,35

6. ÇOK OTURUMLU SINAVLARDA ÖĞRENCİ BİNA ATAMASI PROBLEMİ İÇİN GELİŞTİRİLEN ÇOK AMAÇLI SEZGİSEL

Öğrenci sınav yeri atama problemi NP-Zor sınıfına ait bir problemdir. NP-Zor sınıfına ait problemler çözüm süresi matematiksel modellerle çok uzun zaman gerektirmekte, çoğunlukla da ele alınan problem için bir en iyi değere ulaşılamamaktadır. Öğrenci sınav yeri atama problemine ait test 1'in verileri Açıköğretim Fakültesi'nden alınan veriler doğrultusunda Eskişehir sınav merkezinde yapılmış iki oturumdan oluşan bir günlük öğrenci ve bina verilerinden yola çıkarak hazırlanmıştır. Test 1 probleminin matematiksel modelinde 2.711.916 adet karar değişkeni ve 164.612 adet kısıt bulunduğu düşünüldüncce bu problemin çözümünde matematiksel modelin kullanılmayacağı aşikârdır. Bu nedenle problemin çözümü için sıralama temelli çok amaçlı sezgisel yöntem geliştirilmiştir. Geliştirilen sezgisel algoritmanın kısıntıları ve amaç fonksiyonu, matematiksel modelin atama kısıntıları ve amaç fonksiyonu ile aynıdır.

Geliştirilen çok amaçlı sezgisel algoritma parametre olarak bir binada katlanılabilecek en yüksek kitapçık çeşitliliği ve en düşük bina doluluk oranını almaktadır. Geliştirilen çok amaçlı sezgisel algoritma binalardaki doluluk oranını en büyükleyecek; kapasite onarma ve binalardaki kitapçık çeşitliliği sayısını en küçükleyecek; çeşitlilik onarma adındaki iki alt algoritmadan oluşmaktadır. Çeşitlilik onarma alt algoritmasının çözüm uzayı en yüksek kitapçık çeşitliliği parametresi ile üstten sınırlandırılmaktadır. Benzer şekilde kapasite onarma alt algoritmanın çözüm uzayı ise en düşük doluluk oranı parametresi ile alttan sınırlandırılmaktadır. Öğrencilerin ardışık oturumlar arasındaki kat ettiği mesafe ise her iki alt algoritmanın amaçları göz önüne alınarak en küçülenmektedir. Alınan parametreler doğrultusunda her iki alt algoritmadan gelen amaç fonksiyonu değerlerinden en küçük olan seçilir ve atama bu amaç fonksiyonunu veren alt algoritmaya göre yapılır. Amaç fonksiyonu ve klasik atama kısıntıları matematiksel model iyi aynıdır. Sezgisel algoritmadaki mesafe fonksiyonu matematiksel modeldeki mesafe ilgili olan amaç fonksiyonuna (f_1), kapasite fonksiyonu matematiksel modeldeki kapasitelerle ilgili olan amaç fonksiyonuna (f_2) ve son olarak çeşitlilik fonksiyonu matematiksel modeldeki kitapçık çeşitliliği ile ilgili olan amaç fonksiyonuna (f_3) karşılık gelmektedir. Geliştirilen algoritmada kullanılan parametreler ve amaç fonksiyonları şu şekildedir.

Parametreler

- Katlanılabilecek en yüksek kitapçık çeşitliliği
- Katlanılabilecek en düşük bina doluluk oranı
- Öğrenci-kitapçık matrisi
- Kitapçık- oturma matrisi
- Bina kapasiteleri
- Bina enlem ve boylam bilgileri

6.1 Çeşitlilik Onarma Alt Algoritması Adımları

Çeşitlilik onarma alt algoritmasına vba kodlarına ekler bölümünden erişilebilir. Algoritmanın adımları şu şekildedir:

6.1.1 Birinci oturum için başlangıç çözümün oluşturulması

Adım 1: Kitapçık çeşitleri öğrenci sayılarına, binalar kapasitelerine göre en büyükten en küçüğe sıralanır.

Adım 2: Adım 1’de yapılan sıralamaya göre öğrenciler binalara atanır.

6.1.2 Birinci oturum için oluşturulan başlangıç çözümün onarılması

Adım 3: Kitapçık çeşitliliği parametresinden daha büyük çeşitliliğe sahip olan binalardaki kitapçıklar öğrenci sayılarına göre en küçükten en büyüğe sıralanır.

Adım 4: Öğrenci atanan binalar boş kalan kapasitelerine göre küçükten büyüğe sıralanır.

Adım 5: Kitapçık çeşitliliği parametresinden daha büyük çeşitliliğe sahip olan binalardaki kitapçıklar yapılan sıralamalara göre aynı kitapçığa sahip binalara kapasite ve kitapçık çeşitliliği parametresi aşılmayacak şekilde taşınır.

Adım 6: : Kitapçık çeşitliliği parametresinden daha büyük çeşitliliğe sahip olan binalar yoksa algoritma sonlandırılır.

Adım 7: Adım3 ve Adım 4 tekrar edilir.

Adım 8: Kitapçık çeşitliliği parametresinden daha büyük çeşitliliğe sahip olan binalardaki kitapçıkları alan öğrenciler öğrenci atanmış olan diğer binalara kapasite ve kitapçık çeşitliliği parametresi aşılmayacak şekilde taşınır.

6.1.3 İkinci oturum için başlangıç çözümün oluşturulması

Adım 9: Birinci ve ikinci oturuma giren öğrenciler birinci oturumda girdikleri binalara atanırlar.

Adım 10: Sadece ikinci oturuma giren öğrenciler için bina kapasiteleri göz önüne alınarak Adım 1 ve Adım 2 gerçekleştirilir.

6.1.4 İkinci oturum için oluşturulan başlangıç çözümün onarılması

Adım 11: Adım 3 tekrar edilir.

Adım 12: Kitapçık çeşitliliği parametresinden daha büyük çeşitliliğe sahip olan binalardaki kitapçıkları alan öğrenciler aynı kitapçığa sahip öğrenci atanmış olan en yakındaki binalara taşınır.

Adım 13: Kitapçık çeşitliliği parametresinden daha büyük çeşitliliğe sahip olan binalar yoksa son adıma geçilir

Adım 14: Adım 3 ve Adım 4 tekrar edilir.

Adım 15: Kitapçık çeşitliliği parametresinden daha büyük çeşitliliğe sahip olan binalardaki kitapçıkları alan öğrenciler öğrenci atanmış olan en yakındaki binalara kapasite ve kitapçık çeşitliliği parametresi aşılmayacak şekilde taşınır.

6.2 Kapasite Onarma Alt Algoritması

Kapasite onarma alt algoritmasına ait vba kodlarına ekler bölümünden erişilebilir. Algoritmanın adımları şu şekildedir:

6.2.1 Birinci oturum için çözümün oluşturulması

Adım 1: Kitapçık çeşitleri öğrenci sayılarına, binalar kapasitelerine göre en büyükten en küçüğe sıralanır.

Adım2: Adım 1’de yapılan sıralamaya göre öğrenciler binalara atanır.

6.2.2 İkinci oturum için başlangıç çözümün oluşturulması

Adım 3: Birinci ve ikinci oturuma giren öğrenciler birinci oturumda girdikleri binalara atanırlar.

Adım 4: Sadece ikinci oturuma giren öğrenciler için bina kapasiteleri göz önüne alınarak Adım 1 ve Adım 2 gerçekleştirilir.

6.2.3 İkinci oturum için oluşturulan başlangıç çözümün onarılması

Adım 5: Öğrencisi olan binalar boş kalan kapasitelerine göre en büyükten en küçüğe sıralanır.

Adım 6: Doluluk oranı parametresinden daha küçük doluluk oranına sahip olan binalardaki kitapçıklar öğrenci sayılarına göre en küçükten en büyüğe sıralanır.

Adım 7: Doluluk oranı parametresinden daha küçük doluluk oranına sahip olan binalardaki kitapçıklar sırası ile en yakındaki aynı kitapçığa sahip binalara doluluk oranı parametresi ve kapasite göz önüne alınarak taşınır.

Adım 8: Doluluk oranı parametresinden daha küçük doluluk oranına sahip bina yoksa algoritma sonlandırılır.

Adım 9: Adım 5 ve Adım 6 tekrar edilir.

Adım 10: Doluluk oranı parametresinden daha küçük doluluk oranına sahip olan binalardaki kitapçıklar en yakın binadaki en yüksek doluluk oranına sahip binaya doluluk oranı parametresi ve kapasite göz önüne alınarak taşınır.

Adım 11: Doluluk oranı parametresinden daha küçük doluluk oranına sahip bina yoksa algoritma sonlandırılır.

Adım 12: Doluluk oranı parametresinden daha küçük doluluk oranına sahip olan binalardaki öğrenciler en yakın binadaki en yüksek doluluk oranına sahip binaya doluluk oranı parametresi ve kapasite göz önüne alınarak taşınır.

6.3 Sezgisel Algoritma İle Elde Edilen Sayısal Sonuçlar

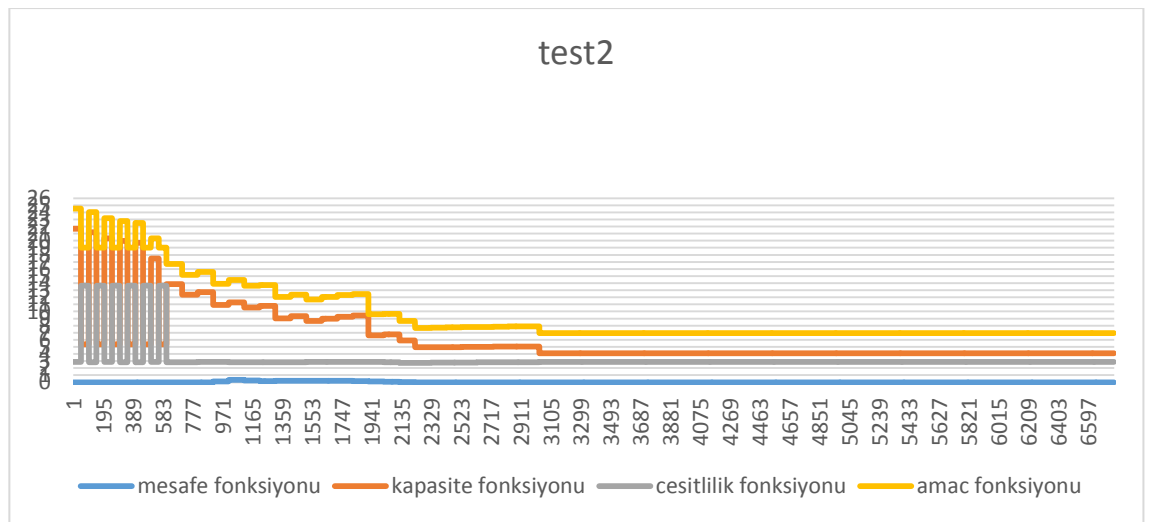
Geliştirilen çok amaçlı sezgisel algoritma Excel VBA ile kodlanmıştır. Matematiksel modelin çözümü ile karşılaştırma yapabilmek için test2, test3 ve test4 verileri kitapçık çeşitliliği parametresi değeri olarak test verisindeki en büyük kitapçık çeşitliliği ve bina doluluk oranı parametresi için sıfır kullanılarak sezgisel algoritmada çözüm aranmıştır. Bulunan sonuçlar ve GAMS programında bulunan en iyi çözüm değerleri ile karşılaştırılmıştır. Tablo 6.1' deki tablodaki z değeri sezgisel algoritmanın amaç fonksiyonu değerini temsil etmekte; f1, f2 ve f3 sırasıyla öğrencilerin ardışık oturumlarda atandıkları binalar arası mesafeyi, kullanılmayan bina kapasitesini ve

binalara atanan toplam kitaplık çeşitliliği sayısını veren normalleştirilmiş fonksiyonlarını temsil etmektedir.

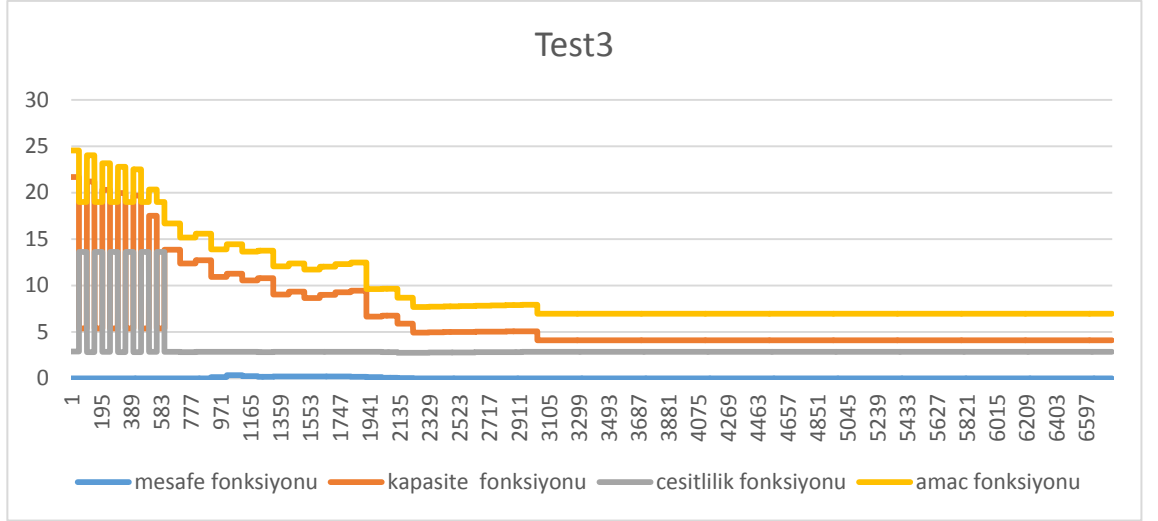
Tablo 6.1 Matematiksel Model ve Sezgisel Algoritma Karşılaştırılması

	Test2				Test3				Test4			
	f1	f2	f3	z	f1	f2	f3	z	f1	f2	f3	z
Matematiksel Model	0,058	0	2,169	2,227	0	1,12	1,548	2,672	0	0	1,35	1,35
Sezgisel	0	11,67	3,62	15,29	0	4,087	2,856	6,943	0	1	3,2	4,2

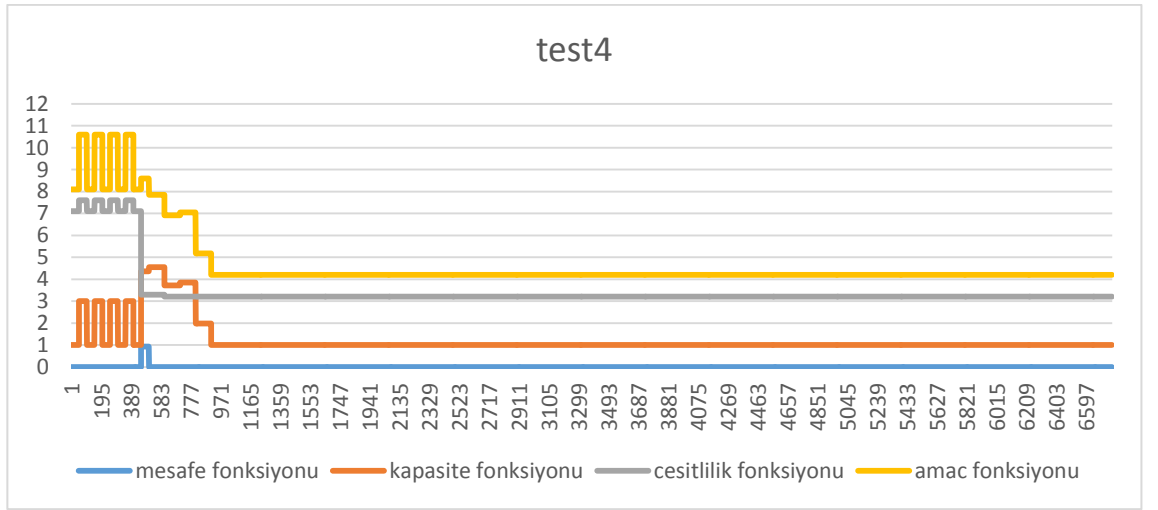
Test2, Test3 ve Test4 problemlerinin sezgisel algoritmayla adım adım çözümleri Şekil 6.1, Şekil 6.2 ve Şekil 6.3 'de görülmektedir. Problemin boyutları arttıkça sezgisel algoritmanın en iyi çözümüne ulaştığı yineleme sayısı da artmıştır. Geliştirilen çok amaçlı sezgisel algoritmanın ardışık oturumlar arasında öğrencilerin kat ettiği mesafeyi en küçükleme konusundaki performansının matematiksel modelin GAMS ile bulunan bilinen en iyi çözümleri ile Test3 ve Test4 problemlerinde aynı olduğu, Test2 probleminde ise daha iyi sonuç verdiği görülmektedir. Binalarda boş kalan kapasitenin ve kitaplık çeşitliliğinin en küçüklenmesi ele alındığında ise GAMS ile bulunan bilinen en iyi çözümlerin geliştirilen sezgiselden daha iyi sonuç verdiği görülmektedir.



Şekil 6.1 Sezgisel Algoritmayla Test2'in Çözümü

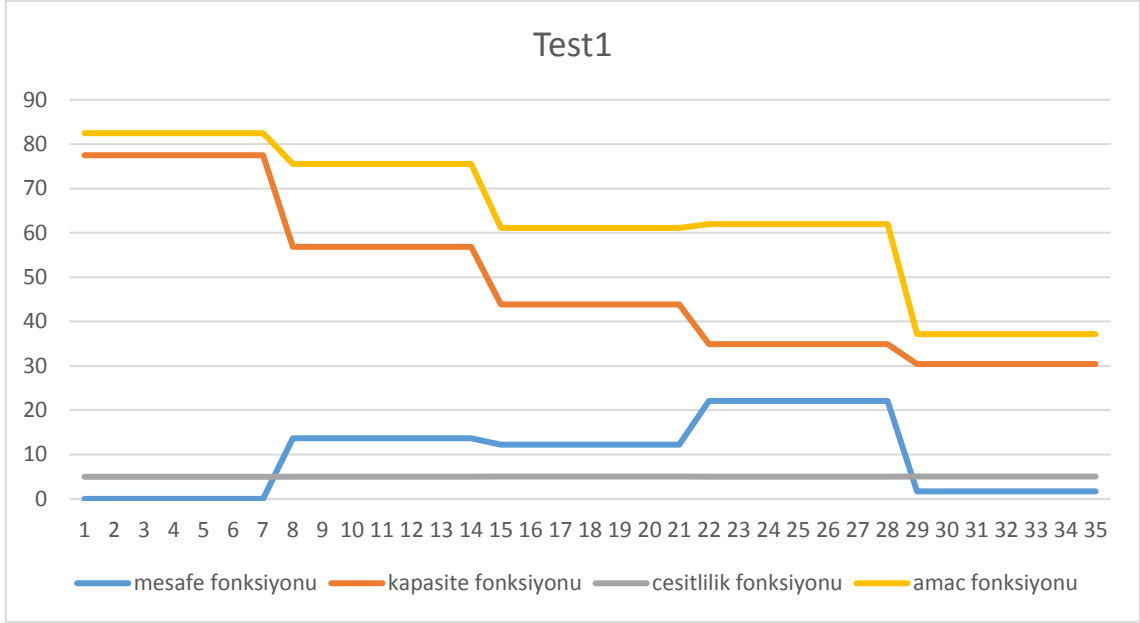


Şekil 6.2 Sezgisel Algoritma ile Test3'ün Çözümü



Şekil 6.3 Sezgisel Algoritma ile Test4'ün Çözümü

Açıköğretim sınavlarındaki orta ölçekli bir sınav merkezin ilk gün verilerini içeren Test1 problemi sezgisel algoritma ile çözdürülürken ilgili sınavdaki ortalama kitapçık çeşitliliği değeri olan 5 ve ortalama doluluk oranı olan %95 değerleri kullanılmıştır. Test1 probleminin sezgisel algoritmayla adım adım çözümleri Şekil 6.4 'de görülmektedir



Şekil 6.4 Sezgisel Algoritma ile Test1'in Çözümü

Geliştirilen sezgisel algoritma ölçekli bir sınav merkezine ait eski bir sınav verisi için çalıştırılmış ve sezgisel algoritmanın sonuçları performans göstergeleri üzerinden karşılaştırılmıştır. Algoritma Cumartesi ve Pazar günleri için ayrı ayrı çalıştırılmış, her günün ortalama kitapçık çeşitliliği ve ortalama doluluk oranı değerleri algoritmanın parametre değerleri olarak girilmiştir.

Tablo 6.2 'de mevcut sistem ile önerilen sezgiselin sonuçları ardışık oturumlar arasında öğrenci başına düşen ortalama kat edilen mesafe, öğrencilerin ardışık oturumlar arasında kat ettiği maksimum mesafe ve ardışık oturumlara giren öğrencilerin kat ettikleri mesafeler açısından karşılaştırılmıştır. Mevcut sistemde mesafe ile ilgili özel bir çalışma yapılmadığı için bu performans ölçütündeki iyileştirme sayılarıyla ifade edilemeye gerek kalmayacak kadar büyük olmuştur.

Tablo 6.2 Mesafe Değerleri Performans Karşılaştırılması

	1. gün			2. gün		
	Ortalama (km)	Maksimum (km)	Toplam (km)	Ortalama (km)	Maksimum (km)	Toplam (km)
Mevcut Sistem	3,317001089	9,546211103	23427,97869	3,382856374	9,546211103	25723,23987
Çok Amaçlı Sezgisel	0,002149805	1,851965202	15,18407594	0,001255774	0,519572308	9,71341114

Tablo 6.3 ‘de mevcut sistem ile önerilen sezgiselin sonuçları bina başına düşen ortalama kitapçık çeşitliliği ve tüm gün boyunca bir binada rastlanan maksimum kitapçık çeşitliliği açısından karşılaştırılmıştır. Önerilen sezgisel algoritma her iki performans ölçütü için daha iyi sonuç vermiştir.

Tablo 6.3 *Kitapçık Değerleri Performans Karşılaştırılması*

	1. gün		2. gün	
	Ortalama	Maksimum	Ortalama	Maksimum
Mevcut Sistem	5,029411	14	5,22222	17
Çok Amaçlı Sezgisel	3,448275862	5	3,50877193	5

Tablo 6.4 ‘de mevcut sistem ile önerilen sezgiselin sonuçları ortalama bina doluluk oranları ve tüm gün boyunca bir binada rastlanan minimum bina doluluk oranı açısından karşılaştırılmıştır. Önerilen sezgisel algoritma her iki performans ölçütü için mevcut duruma göre bir düşüş olmuştur.

Tablo 6.4 *Bina Doluluk Oranı Değerleri Performans Karşılaştırılması*

	1. gün		2. gün	
	Minumum (%)	Ortalama (%)	Minumum (%)	Ortalama (%)
Mevcut Sistem	60,03236246	94,11	40,22988506	95,67
Çok Amaçlı Sezgisel	7,731958763	56,11752758	7,731958763	56,11752758

Sonuçlar değerlendirilecek olursa öğrencilerin şikâyet ettikleri mesafe değerlerinde ve binalardaki kitapçık çeşitliliği sayılarında kayda değer bir iyileştirme sağlanmıştır. Üniversitenin kar amacı gütmeyen bir kuruluş olduğu düşünüldüğünde, bina doluluk oranlarındaki düşüğe katlanılabilir.

7. SONUÇ VE ÖNERİLER

Çalışmanın kapsamında Anadolu Üniversitesi'nin Açıköğretim Fakültesi'ne ait sınavlardan yola çıkılarak çok oturumlu sınavlarda öğrenci sınav yeri atama problemi ele alınmıştır. Her eğitim-öğretim döneminde iki kere yapılan bu sınavlara Açıköğretim programlarına kayıtlı yaklaşık 1 milyon 400 bin öğrenci girmektedir. Her bir sınav toplamda 98 sınav merkezinde sabah ve öğleden sonra birer oturum olacak şekilde cumartesi ve pazar günleri yapılmaktadır. Çok oturumlu sınavlarda öğrenci sınav yeri atama problemi bu çalışmada eğitimsel zaman çizelgeleme başlığı altında ele alınmıştır.

Problemin çözümü için öncelikle çok amaçlı karma tamsayılı doğrusal olmayan bir matematiksel model geliştirilmiştir. Gerçek sınav verilerinden yola çıkılarak hazırlanan 4 tane test problemi matematiksel model ile çözülmüştür. Orta ölçekli bir sınav merkezinin cumartesi günü verilerini içeren Test1 problemi boyutları nedeniyle matematiksel model ile çözülememiştir. Literatürdeki benzer problemler incelendiğinde matematiksel modelle çözülemeyen problemler için ilgili eğitim kuruma özgü sezgisel algoritmalar geliştirildiği tespit edilmiştir. Bu nedenle çok oturumlu sınavlarda öğrenci sınav yeri atama problemi için çok amaçlı bir sezgisel yöntem geliştirilmiştir. Geliştirilen sezgisel yöntem tek çözüm tabanlı olup, başlangıç çözüm üzerinde belirlenen amaç fonksiyonları doğrultusunda yapılan onarmalara ve bozulmalara dayanmaktadır.

Geliştirilen çok amaçlı sezgisel yöntem test problemlerine uygulanarak matematiksel modelin bilinen en iyi çözümleri ile karşılaştırılmıştır. Çalışmanın son aşamasında önerilen sezgisel algoritma gerçek verilere uygulanmış ve mevcut sistemle sonuçlar karşılaştırılmıştır. Elde edilen sayısal sonuçlara göre sezgisel algoritmanın üstünlüğü gösterilmiştir.

Gelecek çalışmalarda, geliştirilen sezgisel algoritma gerekli ara yüzler hazırlanarak bir karar destek sistemine dönüştürülebilir. Öğrenci sınav yeri belirleme problemine ardışık oturumlar arasındaki mesafeye ek olarak, öğrencilerin günün ilk oturumuna giderken çıkış noktaları ile ilk oturumdaki sınav yerleri arasındaki mesafe de eklenebilir. Büyük boyutlu öğrenci sınav yeri atama problemleri için uygun meta sezgisel ve sezgi üstü algoritmalar kullanılabilir.

KAYNAKÇA

- [1] Burkard, R., Dell'Amico, M. ve Martello S. (2008). Assignment Problems. *SIAM*. Philadelphia.
- [2] Votaw, D. ve Orden, A. (1952). The personnel assignment problem.
- [3] Pentico, D. W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176 (2), 774-793.
- [4] Kamişlı Öztürk, Z. (2010). Eğitimsel Zaman Çizelgeleme Problemleri İçin Çözüm Yaklaşımları ve Web Tabanlı Bir Karar Destek Sistemi Önerisi. Eskişehir.
- [5] Burke, E. K. ve Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140 (2), 266-280
- [6] Karter, M. W. ve Gilbert, L. (1998). Recent Developments in Practical Course. *Lecture Notes in Computer Science*,
- [7] Burke, E., Jackson, K., Kingston, J. ve Weare, R. (1997). Automated university timetabling: the state of the art. *The Computer Journal*, 40 (9), 565-571.
- [8] Babaei, H., Karimpour, J. ve Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86 (1), 43-59.
- [9] Ayob, M. ve Malik, A. (2011). A New Model for an Examination-Room Assignment Problem. *International Journal of Computer Science and Network Security*, 11, 187-190.
- [10] Boizumault, P., Delon, Y. ve Peridy, L. (1996). Constraint logic programming for examination timetabling. *The Journal of Logic Programming*, 26 (2), 217-233.

- [11] Burke, E ve Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140 (2), 226-280.
- [12] Schaerf, A. (1990). A survey of automated timetabling. *Artificial Intelligence Review*, 13 (2), 87-127.
- [13] Dammak, A., Elloumi, A. ve Kamoun, H. (2006). Classroom assignment for exam timetabling. *Advances in Engineering Software*, 37 (10), 659–666.
- [14] Akkoyunlu, E. (1973). A linear algorithm for computing the optimum university. *The Computer Journal*, 16 (4), 347-350.
- [15] Gosselin, K. ve Truchon, M. (1986). Allocation of classrooms by linear programming. *The Journal of the Operational Research Society*, 37 (6), 561-569.
- [16] Phillips, A.E., Hamish, W., Matthias, E. ve David, M. R. (2015). Integer programming methods for large-scale practical classroom assignment problems. *Computers & Operations Research*, 53, 42-53.
- [17] Akif, B. M (2008). A 0-1 Integer Programming Approach to a University. *Hacettepe Journal of Mathematics and Statistics*, 37 (1), 41-55.
- [18] Kanişlı Öztürk, Z. ve Sagır, M. (2010). Exam scheduling: Mathematical modeling and parameter estimation with the Analytic Network Process approach. *Mathematical and Computer Modelling*, 52 (5), 930-941.
- [19] Al-Yakoob, S., Sherali, H. ve Al-Jazzaf, M. (2010). A mixed-integer mathematical modeling approach to exam timetabling. *Computational Management Science*, 7 (1), 19-46,
- [20] Farida, H., Mira, R., Romliyah, A. ve Toni, B. (2015). Exam Invigilators Assignment Problem: A Goal Programming Approach. *Applied Mathematical Sciences*, 58 (9), 2871 - 2880.
- [21] Cavdur, F. ve Kose, M. (2016). A Fuzzy Logic and Binary-Goal Programming-Based Approach for Solving the Exam Timetabling Problem

- to Create a Balanced-Exam Schedule. *International Journal of Fuzzy Systems*, 18 (1), 119-129.
- [22] Kahara, M. ve Kendalla, G. (2010). The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution. *European Journal of Operational Research*, 207 (2), 557-565.
- [23] Qu, R., Burke, E., McCollum, B. ve Merlot, L. (2008). A survey of search methodologies and automated system. *Journal of Scheduling*, 12(1), 55-89.
- [24] Wearea, R., Burke, E. ve Ellimana, D. (1994). A University Timetabling System Based on Graph Colouring and Constraint Manipulation. *Journal of Research on Computing in Education*, 27 (1), 1-18.
- [25] Ahmed, L.N., Ozcan, E. ve Kheire, A. (2016). Solving high school timetabling problems worldwide using selection hyper-heuristics. *Expert Systems with Applications*, 42 (13), 5463-5471.
- [26] Anadolu Üniversitesi, 20/04/2016. [Çevrimiçi]. <https://www.anadolu.edu.tr/acikogretim/acikogretim-sistemi/acikogretim-sistemi-1>.
- [27] Carter, M. W., Laporte, G. ve Lee, S.Y. (1996). Examination Timetabling: Algorithmic Strategies and Applications. *The Journal of the Operational Research Society*, 47 (13), 373-383.
- [28] Vermuytena, H., Lemmensb, S., Marques, I. ve Beliëna, J. (2015). Developing compact course timetables with optimized student flows. *European Journal of Operational Research*, 251 (2), 651-661.

EK-1: GAMS KODU

*student-exam-building assignment

sets

i students /1*m/

j sessions /1*2/

k buildings/1*p/

h booklets/1*o/;

scalar tcomp,texec,telapsed;

*tcomp=TimeCOMP;*texec=TimeExec;*telapsed=TimeElapsed;*optionreslim=1000000;*optioniterlim=100000000;

parameters

q(k) capacity of building k

/bina kapasiteleri/;

alias(k,l);

table d(k,l) building distance matrix

Sondelim

\$include "filename.csv"

\$offdelim ;

table KT(h,j) booklet session matrix

Sondelim

\$include "filename.csv"

\$offdelim ;

table OK(i,h) student booklet matrix

Sondelim

\$include "filename.csv"

\$offdelim ;

binary variables

x(i,j,k) if student i in session j is assigned building k

y(i,j) if student i is in session j

b(j,k) if session j is at building k

r(k,h) if booklet h is at building k;

Scalar M ;

M=Card(i)*Card(j)*Card(k)

Variables f1,f2,f3;

positive variables

c(j,k) excess capacity of building k is in session j

p(k,h) number of booklets h is at building k ;

variable z objective function;

equations

F objective function

o1 obj 1,o2 obj 2,o3 obj3,c1(i) , c2(i,j) ,c3(i,j) ,c4(k) ,c5(j,k) ,c61(i,j) ,c62(i,j) ,c71(j,k) ,c72(j,k) ,c8(j,k) ,c9(k,h),c101(k,h) ,c102(k,h)

F.. z=e=f1+f2+f3;

o2.. SUM((j,k),c(j,k)/q(k))=e=f2;

o1.. SUM((i,k,l)\$ (ord(k) ne ord(l)),(x(i,'1',k)* x(i,'2',l)*d(k,l))/smax((k,l)\$ (ord(k) ne ord(l)),d(k,l))=e=f1 ;

*nonlinear o1

*(\$ ord(k) ne ord(l))

o3.. SUM ((k,h),r(k,h)/card(h))=e=f3;

c1(i).. SUM((j,k),x(i,j,k))=g=1;

c2(i,j).. SUM(h,OK(i,h)*KT(h,j))=e=y(i,j);

c3(i,j).. SUM(k,x(i,j,k))=l=1;

c4(k).. SUM(j,b(j,k))=l=2;

c5(j,k).. SUM(i,x(i,j,k))=l=q(k);

c61(i,j)..SUM(k,x(i,j,k))=g=y(i,j);

c62(i,j).. SUM(k,x(i,j,k))=l= y(i,j)*M;

c71(j,k).. sum(i,x(i,j,k))=g=b(j,k);

c72(j,k).. sum(i,x(i,j,k))=l=b(j,k)*M;

c8(j,k).. c(j,k)=e=b(j,k)*q(k)-SUM(i,x(i,j,k));

c9(k,h).. p(k,h)=e=SUM((i,j),x(i,j,k)*OK(i,h)*KT(h,j));

c101(k,h).. r(k,h)=l=p(k,h);

c102(k,h).. p(k,h)=l=r(k,h)*M;

MODEL studentassig/ALL/;

*studentassig.Nodlim=10000;

SOLVE studentassig using MINLP minimizing z;

EK-2: EXCEL VBA KODU

Option Base 1

Dim bsayi As Integer

Dim osayi As Integer

Dim otsayi As Integer

Dim ksayi As Integer

Dim aosayi As Integer

Dim dmatris As Integer

Dim maxksayi

Dim toplam As Double

Dim doluluk As Double

Dim otno As Integer

Type bina 'bina sinifi

 b_id As Integer

 bolge As Integer

 kapasite As Integer

 kalankap() As Integer

 kcesit() As Integer

 doluluko() As Double

 kcesitsayi() As Integer

 x As Double

 y As Double

 d_binaid() As Double

End Type

Type ogrenci 'ogrenci sinifi

 id As Integer

 kitapcik() As Integer

 bina() As Integer

End Type

Dim kitapcikilk() As Integer

Dim binalar() As bina

Dim ogrenciler() As ogrenci 'tum ogrenciler

Dim aogrenciler() As ogrenci 'her iki oturuma da giren ogrenciler

Dim bolgeler() As region

Dim binabilgi As Range

Dim ogrencibilgileri As Range

Dim kitapcikbilgi As Range

Dim x1 As Double
Dim x2 As Double
Dim y1 As Double
Dim y2 As Double
Dim binadoluluk As Double
Dim kitapcikcesit As Integer
Dim bolgesayi As Integer
Dim toplammesafe As Double
Dim maxmesafe As Double
Dim ortalamamesafe As Double
Dim ortalamadoluluk As Double
Dim ortalamakitapcik As Double
Dim dolulukw As Double
Dim mesafew As Double
Dim kitapcikw As Double
Dim option1value As Double
Dim option2value As Double
Dim objectivedoluluk1 As Double
Dim objectivekitapcik1 As Double
Dim objectivemesafe1 As Double
Dim objectivedoluluk2 As Double
Dim objectivekitapcik2 As Double
Dim objectivemesafe2 As Double
Dim optiontype As Byte

Dim maxcesit As Integer
Dim mincesit As Integer
Dim mindoluluk As Double
Dim maxdoluluk As Double
Dim totalkalangkap As Integer
Dim totalkap As Integer
Dim maxbinauzaklik As Double
Dim toplamcesit As Integer
Dim totalkapasiteo As Double
Dim binadolulukdown As Double


```

Dim kitapcikcesitup As Integer
Dim minval As Double
Type analiz
p_doluluk As Integer
p_cesit As Integer
objective As Double
objectivedoluluk As Double
objectivekitapcik As Double
objectivemesafe As Double
opt As Integer
End Type
Dim analizler() As analiz
Dim starttime As Double
Dim endtime As Double

Sub main()
starttime = Timer
binadolulukdown =
kitapcikcesitup =
ReDim analizler((100 - binadolulukdown + 1) * kitapcikcesitup)
minval = 10000000
ReDim analiz(kitapcikcesitup * 100)
say = 0
For i = 1 To kitapcikcesitup
    For j = 100 To binadolulukdown Step -1
        say = say + 1

        kitapcikcesit = i
        binadoluluk = j
        analizler(say).p_cesit = i
        analizler(say).p_doluluk = j
        optiontype = 0
        Sheets("data").Select
        Call parametreal
        optiontype = 1
        Call option1
        optionvalue1 = dolulukw * objectivedoluluk1 + objectivekitapcik1 * kitapcikw + objectivemesafe1 * mesafew
    
```

```

'MsgBox optionvalue1
optiontype = 2
Call option2
optionvalue2 = dolulukw * objectivedoluluk2 + objectivekitapcik2 * kitapcikw + objectivemesafe2 * mesafew
'MsgBox optionvalue2
If optionvalue1 > optionvalue2 Then

    analizler(say).objective = optionvalue2
    analizler(say).objectivedoluluk = objectivedoluluk2
    analizler(say).objectivekitapcik = objectivekitapcik2
    analizler(say).objectivemesafe = objectivemesafe2
    analizler(say).opt = 2
    'MsgBox "option2 selected"
    If minval > optionvalue2 Then
        minval = optionvalue2
        a = i
        b = j
    End If
Else
    analizler(say).objective = optionvalue1
    analizler(say).objectivedoluluk = objectivedoluluk1
    analizler(say).objectivekitapcik = objectivekitapcik1
    analizler(say).objectivemesafe = objectivemesafe1
    analizler(say).opt = 1
    If minval > optionvalue1 Then
        minval = optionvalue1
        a = i
        b = j
    End If
End If
Next
Next
endtime = Timer - starttime
kitapcikcesit = a
binadolulukdown = b
optiontype = 0
Sheets("data").Select

```

```

Call parametreal
optiontype = 1
Call option1_select
optionvalue1 = dolulukw * objectivedoluluk1 + objectivekitapcik1 * kitapcikw + objectivemesafe1 * mesafew
MsgBox optionvalue1
optiontype = 2
Call option2_select
optionvalue2 = dolulukw * objectivedoluluk2 + objectivekitapcik2 * kitapcikw + objectivemesafe2 * mesafew
MsgBox optionvalue2
If optionvalue1 > optionvalue2 Then
MsgBox "option2 selected"
Else
MsgBox "option1 selected"
End If

End Sub

Sub option1()
Sheets("data").Select
otno = 1
Call sort(kitapcikilk(), 2, 2)
Call binakapsort(otno, 2) 'kapasite büyükten küçüğe
Call ilkoturumatama
Call cesitlilik(otno)
Call dolulukorani(otno)
Call ilkoturumgeriatama
Call dolulukorani(otno)
objectivedoluluk1 = totalkapasiteo
Call cesitlilik(otno)
objectivekitapcik1 = toplamcesit / ksayi
'Call report(otno)
Call ardisikogrenci
Call binakapsort(otno, 2)
Call ikincioturumatamastep1
Call ikincioturumatamastep2
otno = 2
Call cesitlilik(otno)
Call binasort

```

Call ikincioturumgeriatama
Call cesitlilik(otno)
 $\text{objectivekitapcik1} = \text{objectivekitapcik1} + \text{toplamesit} / \text{ksayi}$
Call dolulukorani(otno)
 $\text{objectivedoluluk1} = \text{totalkapasiteo} + \text{objectivedoluluk1}$
Call mesafe
 $\text{objectivemesafe1} = \text{toplammesafe} / \text{maxbinauzaklik}$
End Sub
Sub option2()
Sheets("data").Select
Call parametreal
otno = 1
Call sort(kitapcikilk(), 2, 2)
Call binakapsort(otno, 2)
Call ilkoturumatama
otno = 1
Call dolulukorani(otno)
Call cesitlilik(otno)
 $\text{objectivekitapcik2} = \text{ortalamakitapcik}$
Call dolulukorani(otno)
 $\text{objectivedoluluk2} = \text{totalalangkap} / \text{totalkap}$
'Call report(otno)
Call ardisikogrenci
Call binakapsort(otno, 2)
Call ikincioturumatamastep1
Call ikincioturumatamastep2
otno = 2
Call cesitlilik(otno)
Call binasort
Call dolulukorani(otno)
Call dolulukgeriatama(otno)
Call cesitlilik(otno)
 $\text{objectivekitapcik2} = \text{objectivekitapcik2} + \text{toplamesit} / \text{ksayi}$
Call dolulukorani(otno)
 $\text{objectivedoluluk2} = \text{totalkapasiteo} + \text{objectivedoluluk2}$
Call mesafe
 $\text{objectivemesafe2} = \text{toplammesafe} / \text{maxbinauzaklik}$

```

'Call report(otno)
End Sub
Sub parametreal()
Set binabilgi = Range(Range("A1"), Range("A1").End(xlDown).End(xlToRight))
Set ogrencibilgileri = Range(Range("J1"), Range("J1").End(xlDown).End(xlToRight))
Set kitapcikbilgi = Range(Range("N1"), Range("N1").End(xlDown))
bsayi = binabilgi.Rows.Count - 1
osayi = ogrencibilgileri.Rows.Count - 1
otsayi = 2
ReDim binalar(bsayi)
ReDim ogrenciler(osayi)
ReDim kitapcikilk(ksayi, 2)
ReDim bolgeler(bolgesayi)
For i = 1 To bolgesayi
    ReDim bolgeler(i).regions(bolgesayi, 2)
    bolgeler(i).id = i
Next
For i = 1 To bolgesayi
    boluzaklik = 0
    For j = i To bolgesayi
        bolgeler(i).regions(j, 1) = j
        bolgeler(i).regions(j, 2) = boluzaklik
        bolgeler(j).regions(i, 2) = bolgeler(i).regions(j, 2)
        bolgeler(j).regions(i, 1) = i 'index
        boluzaklik = boluzaklik + 1
    Next
Next
Next
For i = 1 To bsayi
    ReDim binalar(i).kalankap(otsayi)
    ReDim binalar(i).kcesit(1, maxksayi)
    ReDim binalar(i).kcesit(2, maxksayi)
    ReDim binalar(i).kcesitsayi(otsayi)
    ReDim binalar(i).doluluk(otsayi)
    ReDim binalar(i).d_binaid(bsayi, 2) ' dizinin 1. elemanı diğer binanın idsine ikincisi uzaklığa
    binalar(i).b_id = binabilgi.Cells(i + 1, 1)
    binalar(i).oncelik = binabilgi.Cells(i + 1, 3)
    binalar(i).kapasite = binabilgi.Cells(i + 1, 4)

```

```

binalar(i).x = binabilgi.Cells(i + 1, 6)
binalar(i).y = binabilgi.Cells(i + 1, 7)
For j = 1 To otsayi
    binalar(i).kalankap(j) = binabilgi.Cells(i + 1, 4)
Next
binalar(i).bolge = binabilgi.Cells(i + 1, 5)
Next
For i = 1 To ksayi
    kitapcikilk(i, 1) = kitapcikbilgi.Cells(i + 1, 1)
Next
For i = 1 To osayi
    ogrenciler(i).id = ogrencibilgileri.Cells(i + 1, 1)
    ReDim ogrenciler(i).kitapcik(otsayi)
    ReDim ogrenciler(i).bina(otsayi)
    For j = 1 To otsayi
        ogrenciler(i).kitapcik(j) = ogrencibilgileri.Cells(i + 1, j + 1)
    Next
    For k = 1 To ksayi
        If ogrencibilgileri.Cells(i + 1, 2) = kitapcikilk(k, 1) Then
            kitapcikilk(k, 2) = kitapcikilk(k, 2) + 1
        End If
    Next
Next
maxbinauzaklik = 0
For i = 1 To bsayi
    For j = i + 1 To bsayi
        binalar(i).d_binaid(j, 1) = binalar(j).b_id
        binalar(i).d_binaid(j, 2) = oklid(binalar(i).x, binalar(j).x, binalar(i).y, binalar(j).y)
        binalar(j).d_binaid(i, 2) = binalar(i).d_binaid(j, 2)
        binalar(j).d_binaid(i, 1) = binalar(i).b_id
        If binalar(i).d_binaid(j, 2) > maxbinauzaklik Then maxbinauzaklik = binalar(i).d_binaid(j, 2)
    Next
Next
End Sub
Sub ilkoturumsiralama()
Dim temp As Integer

```

```

For i = 1 To ksayi - 1 'bubble sort
  For j = i + 1 To ksayi
    If kitapcikilk(j, 2) > kitapcikilk(i, 2) Then
      For k = 1 To 2
        temp = kitapcikilk(i, k)
        kitapcikilk(i, k) = kitapcikilk(j, k)
        kitapcikilk(j, k) = temp
      Next
    End If
  Next
Next
End Sub

'Sub binasirala()
'Dim temp As bina
'For i = 1 To bsayi - 1
'  For j = i + 1 To bsayi
'    If binalar(j).oncelik < binalar(i).oncelik Then
'      temp = binalar(i)
'      binalar(i) = binalar(j)
'      binalar(j) = temp
'    End If
'  Next
' Next
'End Sub

Sub ilkoturumatama()
For i = 1 To ksayi
  For j = 1 To osayi
    If ogrenciler(j).kitapcik(1) = kitapcikilk(i, 1) Then
      For k = 1 To bsayi
        If binalar(k).kalankap(1) > 0 Then
          ogrenciler(j).bina(1) = binalar(k).b_id
          binalar(k).kalankap(1) = binalar(k).kalankap(1) - 1
        Exit For
      End If
    Next
  End If
End If

```

```

    Next
Next
End Sub

Sub ardisikogrenci()
aosayi = 0
For i = 1 To osayi
If ogrenciler(i).kitapcik(1) <> 0 And ogrenciler(i).kitapcik(2) <> 0 Then
    aosayi = aosayi + 1
    If aosayi = 1 Then
        ReDim aogrenciler(1)
        For k = 1 To otsayi
            ReDim aogrenciler(1).bina(k)
            ReDim aogrenciler(1).kitapcik(k)
        Next
        aogrenciler(1).id = ogrenciler(i).id
        aogrenciler(1).bina(1) = ogrenciler(i).bina(1)
        For j = 1 To otsayi
            aogrenciler(1).kitapcik(j) = ogrenciler(i).kitapcik(j)
        Next
    ElseIf aosayi > 1 Then
        ReDim Preserve aogrenciler(aosayi)
        aogrenciler(aosayi).id = ogrenciler(i).id
        For k = 1 To otsayi
            ReDim Preserve aogrenciler(aosayi).bina(k)
            ReDim Preserve aogrenciler(aosayi).kitapcik(k)
        Next
        aogrenciler(aosayi).bina(1) = ogrenciler(i).bina(1)
        For j = 1 To otsayi
            aogrenciler(aosayi).kitapcik(j) = ogrenciler(i).kitapcik(j)
        Next
    End If
End If

Next

End Sub

```



```

Sub ikincioturumatamastep1()
For i = 1 To aosayi
    For j = 1 To bsayi
        If aogrenciler(i).bina(1) = binalar(j).b_id Then
            If binalar(j).kalankap(2) > 0 Then
                aogrenciler(i).bina(2) = aogrenciler(i).bina(1)
                binalar(j).kalankap(2) = binalar(j).kalankap(2) - 1
            Exit For
        Else
            a = binalar(j).bolge
            For k = 1 To bsayi
                If k <> j And binalar(k).bolge = a And binalar(k).kalankap(2) > 0 Then
                    aogrenciler(i).bina(2) = binalar(k).b_id
                    binalar(k).kalankap(2) = binalar(k).kalankap(2) - 1
                Else
                    MsgBox "yok"
                End If
            Next
        End If
    Next
End If
End If
Next
Next
End Sub

Sub ikincioturumatamastep2()
Set kitapcikbilgi = Range(Range("O1"), Range("O1").End(xlDown))
ksayi = kitapcikbilgi.Rows.Count - 1
ReDim kitapcikilk(ksayi, 2)

For i = 1 To ksayi
    kitapcikilk(i, 1) = kitapcikbilgi.Cells(i + 1, 1)
Next

For i = 1 To osayi
    If ogrenciler(i).kitapcik(1) = 0 And ogrenciler(i).kitapcik(2) <> 0 Then
        For k = 1 To ksayi
            If ogrencibilgileri.Cells(i + 1, 3) = kitapcikilk(k, 1) Then
                kitapcikilk(k, 2) = kitapcikilk(k, 2) + 1
            End If
        Next
    End If
Next
End Sub

```

```

Next
End If
Next
Call sort(kitapcikilk(), 2, 2) 'max to min kitapcik siralama

For i = 1 To ksayi
  For j = 1 To osayi
    If ogrenciler(j).kitapcik(1) = 0 And ogrenciler(j).kitapcik(2) <> 0 And ogrenciler(j).kitapcik(2) = kitapcikilk(i, 1)
    Then
      For k = 1 To bsayi
        If binalar(k).kalankap(2) > 0 Then
          ogrenciler(j).bina(2) = binalar(k).b_id
          binalar(k).kalankap(2) = binalar(k).kalankap(2) - 1
          Exit For
        End If
      Next
    End If
  Next
Next
Next

For i = 1 To osayi 'ogrenciler dizisine ardisikoturum ogrencileri de dahil edilir
  For j = 1 To aosayi
    If ogrenciler(i).id = aogrenciler(j).id Then
      ogrenciler(i).bina(2) = aogrenciler(j).bina(2)
    End If
  Next
Next

End Sub

Sub cesitlilik(oturum As Integer)
  ortalamakitapcik = 0
  toplamcesit = 0
  For i = 1 To bsayi

    ReDim binalar(i).kcesit(oturum, maxksayi) 'maximum kitapçık sayısı!!
    ReDim binalar(i).kcesitsayi(oturum)
  Next

```

```

Dim temp() As Integer
For i = 1 To bsayi
ReDim temp(1)
  For j = 1 To osayi
    Count = 0
    If ogrenciler(j).bina(oturum) <> 0 And ogrenciler(j).bina(oturum) = binalar(i).b_id Then
      For k = 1 To UBound(temp)
        If ogrenciler(j).kitapcik(oturum) <> temp(k) Then
          Count = k
        ElseIf ogrenciler(j).kitapcik(oturum) = temp(k) Then
          Count = 0
        End If
      Next
      If Count <> 0 Then
        ReDim Preserve temp(Count + 1)
        temp(Count + 1) = ogrenciler(j).kitapcik(oturum)
      End If
    End If
  Next
  For k = 1 To UBound(temp) - 1
    binalar(i).kcesit(oturum, k) = temp(k + 1)
  Next
  Count = 0
  sayac = 0
  For i = 1 To bsayi
    Count = 0
    For j = 1 To ksayi
      If binalar(i).kcesit(oturum, j) <> 0 Then Count = Count + 1
    Next
    binalar(i).kcesitsayi(oturum) = Count
    If Count <> 0 Then
      toplamcesit = toplamcesit + Count
      sayac = sayac + 1
    End If
  Next

```

Next

maxcesit = 0

mincesit = 10000

For i = 1 To bsayi

If binalar(i).kcesitsayi(otno) <> 0 Then

 If binalar(i).kcesitsayi(otno) > maxcesit Then maxcesit = binalar(i).kcesitsayi(otno)

 If binalar(i).kcesitsayi(otno) < mincesit Then mincesit = binalar(i).kcesitsayi(otno)

End If

Next

ortalamakitapcik = toplamcesit / sayac

End Sub

Sub mesafe()

toplammesafe = 0

maxmesafe = 0

For i = 1 To aosayi

 If aogrenciler(i).bina(1) <> aogrenciler(i).bina(2) Then

 For j = 1 To bsayi

 If aogrenciler(i).bina(1) = binalar(j).b_id Then

 x1 = binalar(j).x

 y1 = binalar(j).y

 End If

 If aogrenciler(i).bina(2) = binalar(j).b_id Then

 x2 = binalar(j).x

 y2 = binalar(j).y

 End If

 Next

 toplammesafe = toplammesafe + oklid(x1, x2, y1, y2)

 If oklid(x1, x2, y1, y2) > maxmesafe Then

 maxmesafe = oklid(x1, x2, y1, y2)

 End If

```

    End If
Next
ortalamamesafe = toplammesafe / aosayi
End Sub
Sub dolulukorani(otno As Integer)
ortalamadoluluk = 0
totalkap = 0
sayac = 0
totalkalangkap = 0
Dim conv As Double
conv = 0
mindoluluk = 100
maxdoluluk = 0
For i = 1 To bsayi
binalar(i).doluluko(otno) = 0
    If binalar(i).kalangkap(otno) <> binalar(i).kapasite Then
conv = (CLng(binalar(i).kapasite) - binalar(i).kalangkap(otno))
        binalar(i).doluluko(otno) = (conv / binalar(i).kapasite) * 100
    If binalar(i).doluluko(otno) < mindoluluk Then mindoluluk = binalar(i).doluluko(otno)
    If binalar(i).doluluko(otno) > maxdoluluk Then maxdoluluk = binalar(i).doluluko(otno)
        totalkalangkap = binalar(i).kalangkap(otno) + totalkalangkap
        totalkap = binalar(i).kapasite + totalkap
        totalkapasiteo = totalkapasiteo + binalar(i).kalangkap(otno) / binalar(i).kapasite
        ortalamadoluluk = ortalamadoluluk + binalar(i).doluluko(otno)
        sayac = sayac + 1
    End If
Next
ortalamadoluluk = ortalamadoluluk / sayac
End Sub

Sub alternativeilkoturum()
Sum = 0
Count = 1
For i = 1 To ksayi
    For j = 1 To osayi
        If ogrenciler(j).kitapcik(1) = kitapcikilk(i, 1) Then
            If 0 < binalar(Count).kalangkap Then

```

```

    ogrenciler(j).bina(1) = binalar(Count).b_id
    binalar(Count).kalankap = binalar(Count).kalankap - 1
ElseIf 0 >= binalar(Count).kalankap Then
    Count = Count + 1
    If binalar(Count).kalankap > 0 Then
        ogrenciler(j).bina(1) = binalar(Count).b_id
        binalar(Count).kalankap = binalar(Count).kalankap - 1
    End If
End If
End If
Next
Next
End Sub

```

```

Sub binakapsort(oturum As Integer, sorttype As Integer) '1=min to max 2=max to min

```

```

    Dim temp As bina

```

```

    If sorttype = 1 Then

```

```

        For i = 1 To bsayi - 1

```

```

            For j = i + 1 To bsayi

```

```

                If binalar(j).kalankap(oturum) < binalar(i).kalankap(oturum) Then

```

```

                    temp = binalar(i)

```

```

                    binalar(i) = binalar(j)

```

```

                    binalar(j) = temp

```

```

                End If

```

```

            Next

```

```

        Next

```

```

    End If

```

```

    If sorttype = 2 Then

```

```

        For i = 1 To bsayi - 1

```

```

            For j = i + 1 To bsayi

```

```

                If binalar(j).kalankap(oturum) > binalar(i).kalankap(oturum) Then

```

```

                    temp = binalar(j)

```

```

                    binalar(j) = binalar(i)

```

```

                    binalar(i) = temp

```

```

        End If
    Next
Next

End If
End Sub

Sub ilkoturumgeriatama()
Dim tempk() As Integer

Call binakapsort(otno, 1) 'minimum boş yer olandan max boş yer oalan kadar sirala
For i = 1 To bsayi
    If binalar(i).kcesitsayi(otno) > kitapcikcesit Then
        Change = binalar(i).kcesitsayi(otno) - kitapcikcesit 'tasinmasi gereken kitapcik sayisi
        ReDim tempk(binalar(i).kcesitsayi(otno), 2)
        Count = 0
        Min = 100000
        ischange = 0
        For k = 1 To ksayi
            If binalar(i).kcesit(otno, k) <> 0 Then
                Count = Count + 1
                tempk(Count, 1) = binalar(i).kcesit(otno, k)
                count2 = 0
                For m = 1 To osayi
                    If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = binalar(i).kcesit(otno,
k) Then
                        count2 = count2 + 1
                    End If
                Next
                tempk(Count, 2) = count2
            End If
        Next
        Call sort(tempk(), 2, 1)
        For j = 1 To UBound(tempk, 1)
Line1:
            For k = 1 To bsayi

```

```

    If ischange <> Change And binalar(k).kcesitsayi(otno) < kitapcikcesit And binalar(k).kalankap(otno) >=
tempk(j, 2) Then
        For Z = 1 To ksayi
            If binalar(k).kcesit(otno, Z) = tempk(j, 1) Then 'tasinacak kitapcik koduna sahip binada yer varsa oraya
ata
                For m = 1 To osayi
                    If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = tempk(j, 1) Then
                        ogrenciler(m).bina(otno) = binalar(k).b_id 'tasıma
                    End If
                Next
                binalar(k).kalankap(otno) = binalar(k).kalankap(otno) - tempk(j, 2) 'kapasite guncelle
                binalar(i).kalankap(otno) = binalar(i).kalankap(otno) + tempk(j, 2) 'kapasite guncelle
                ischange = ischange + 1
                binalar(i).kcesitsayi(otno) = binalar(i).kcesitsayi(otno) - 1
                j = j + 1
                GoTo Line1
            End If
        Next
    End If
Next
If ischange <> Change Then
    For k = 1 To bsayi
        If ischange <> Change And binalar(k).kcesitsayi(otno) < kitapcikcesit And binalar(k).kalankap(otno) >=
tempk(j, 2) Then 'kitapcik cesitliliği uygun olan bir binaya ata
            For m = 1 To osayi
                If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = tempk(j, 1) Then
                    ogrenciler(m).bina(otno) = binalar(k).b_id 'tasıma
                End If
            Next
            binalar(k).kalankap(otno) = binalar(k).kalankap(otno) - tempk(j, 2) 'kapasite guncelle
            binalar(i).kalankap(otno) = binalar(i).kalankap(otno) + tempk(j, 2) 'kapasite guncelle
            binalar(k).kcesitsayi(otno) = binalar(k).kcesitsayi(otno) + 1 'kitapcik cesitliliği sayisi guncelle
            binalar(i).kcesitsayi(otno) = binalar(i).kcesitsayi(otno) - 1
            For Z = 1 To ksayi
                If binalar(k).kcesit(otno, Z) = 0 Then
                    binalar(k).kcesit(otno, Z) = tempk(j, 1) 'binadaki kitacik kodlarını guncelle
                Exit For
            End If
        Next
    End If
Next

```



```

        For Z = 1 To ksayi
            If binalar(i).kcesit(otno, Z) = tempk(j, 1) Then
                binalar(i).kcesit(otno, Z) = 0 'binadaki kitapcik kodlarımı guncelle
                Exit For
            End If
        Next
        ischange = ischange + 1
        Exit For
    End If
Next
End If
Next
End If
Next

End Sub

Sub ikincioturumgeriatama()
    Dim tempk() As Integer
    break = False
    Call binakapsort(otno, 1) 'minimum boş yer olandan max boş yer oalan kadar sirala
    For i = 1 To bsayi
        If binalar(i).b_id = 1175 Then
            break = True
        End If
        If binalar(i).kcesitsayi(otno) > kitapcikcesit Then
            Change = binalar(i).kcesitsayi(otno) - kitapcikcesit 'tasinmasi gereken kitapcik sayisi
            ReDim tempk(binalar(i).kcesitsayi(otno), 2)
            Count = 0
            Min = 100000
            ischange = 0
            For k = 1 To ksayi
                If binalar(i).kcesit(otno, k) <> 0 Then
                    Count = Count + 1
                    tempk(Count, 1) = binalar(i).kcesit(otno, k)
                    count2 = 0
                    For m = 1 To osayi

```

```

k) Then      If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = binalar(i).kcesit(otno),
              count2 = count2 + 1
              End If
              Next
              tempk(Count, 2) = count2
            End If
            Next
            Call sort(tempk(), 2, 1)
            For j = 1 To UBound(tempk, 1)
Line2:
              For ord = 1 To bsayi
                For k = 1 To bsayi
                  If ischange <> Change And binalar(k).kcesitsayi(otno) < kitapcikcesit And binalar(k).kalankap(otno) >=
tempk(j, 2) And binalar(i).d_binaid(ord, 1) = binalar(k).b_id Then
                    For Z = 1 To ksayi
                      If binalar(k).kcesit(otno, Z) = tempk(j, 1) Then 'tasinacak kitapcik koduna sahip binada yer varsa
oraya ata
                        For m = 1 To osayi
                          If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = tempk(j, 1)
Then
                              ogrenciler(m).bina(otno) = binalar(k).b_id 'tasima
                              End If
                              Next
                              binalar(k).kalankap(otno) = binalar(k).kalankap(otno) - tempk(j, 2) 'kapasite guncelle
                              binalar(i).kalankap(otno) = binalar(i).kalankap(otno) + tempk(j, 2) 'kapasite guncelle
                              ischange = ischange + 1
                              binalar(i).kcesitsayi(otno) = binalar(i).kcesitsayi(otno) - 1
                              For a = 1 To ksayi
                                If binalar(i).kcesit(otno, a) = tempk(j, 1) Then
                                  binalar(i).kcesit(otno, a) = 0 'binadaki kitapcik kodlarını guncelle
                                  Exit For
                                End If
                              Next
                              j = j + 1
                              GoTo Line2
                            End If
                          Next
                        End If
                      Next
                    End If
                  Next
                End If
              End If
            Next
          End If

```

```

Next
If ischange <> Change Then
  For k = 1 To bsayi
    If ischange <> Change And binalar(k).kcesitsayi(otno) < kitapcikcesit And binalar(k).kalankap(otno)
    >= tempk(j, 2) And binalar(i).d_binaid(ord, 1) = binalar(k).b_id Then 'kitapcik cesitliliği uygun olan bir binaya ata
      For m = 1 To osayi
        If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = tempk(j, 1) Then
          ogrenciler(m).bina(otno) = binalar(k).b_id 'tasıma
        End If
      Next
      binalar(k).kalankap(otno) = binalar(k).kalankap(otno) - tempk(j, 2) 'kapasite guncelle
      binalar(i).kalankap(otno) = binalar(i).kalankap(otno) + tempk(j, 2) 'kapasite guncelle
      binalar(k).kcesitsayi(otno) = binalar(k).kcesitsayi(otno) + 1 'kitapcik cesitliliği sayisi guncelle
      binalar(i).kcesitsayi(otno) = binalar(i).kcesitsayi(otno) - 1
      For Z = 1 To ksayi
        If binalar(k).kcesit(otno, Z) = 0 Then
          binalar(k).kcesit(otno, Z) = tempk(j, 1) 'binadaki kitapcik kodlarını guncelle
        Exit For
      End If
    Next
    For Z = 1 To ksayi
      If binalar(i).kcesit(otno, Z) = tempk(j, 1) Then
        binalar(i).kcesit(otno, Z) = 0 'binadaki kitapcik kodlarını guncelle
      Exit For
    End If
  Next
  ischange = ischange + 1
  j = j + 1
  GoTo Line2
End If
Next
End If
Next
Next
End If
Next

```

```
For i = 1 To osayi 'ogrenciler dizisine ardisikoturum ogrencileri de dahil edilir
```

```
  For j = 1 To aosayi
```

```
    If aogrenciler(j).id = ogrenciler(i).id Then
```

```
      aogrenciler(j).bina(2) = ogrenciler(i).bina(2)
```

```
    End If
```

```
  Next
```

```
Next
```

```
End Sub
```

```
Sub binasort()
```

```
  For i = 1 To bsayi
```

```
    Call doublesort(binalar(i).d_binaid(), 2, 1)
```

```
  Next
```

```
End Sub
```

```
Sub doublesort(s_array() As Double, sortindex As Integer, sorttype As Integer) 'min to max sort
```

```
  temp = 0
```

```
  If sorttype = 1 Then 'min to max
```

```
  For i = 1 To UBound(s_array, 1) - 1
```

```
    For j = i + 1 To UBound(s_array, 1)
```

```
      If s_array(j, sortindex) < s_array(i, sortindex) Then
```

```
        For k = 1 To UBound(s_array, 2)
```

```
          temp = s_array(i, k)
```

```
          s_array(i, k) = s_array(j, k)
```

```
          s_array(j, k) = temp
```

```
        Next
```

```
      End If
```

```
    Next
```

```
  Next
```

```
End If
```

```
  If sorttype = 2 Then
```

```
  For i = 1 To UBound(s_array, 1) - 1
```

```

For j = i + 1 To UBound(s_array, 1)
  If s_array(j, sortindex) > s_array(i, sortindex) Then
    For k = 1 To UBound(s_array, 2)
      temp = s_array(i, k)
      s_array(i, k) = s_array(j, k)
      s_array(j, k) = temp
    Next
  End If
Next
Next
End If
End Sub

Sub dolulukgeriatama(otno As Integer)
  Dim tempk() As Integer
  Call binakapsort(otno, 2) 'max boş yer olandan min boş yer olan kadar sirala
  For i = 1 To bsayi

    conv = CLng(binalar(i).kalankap(otno)) * 100
    If binalar(i).kapasite > binalar(i).kalankap(otno) And conv / binalar(i).kapasite > 100 - binadoluluk Then
      Change = binalar(i).kapasite - binalar(i).kalankap(otno) 'tasinmasi gereken kitapcik sayisi
      ReDim tempk(binalar(i).kcesitsayi(otno), 2)
      Count = 0
      Min = 100000
      ischange = 0
      For k = 1 To ksayi
        If binalar(i).kcesit(otno, k) <> 0 Then
          Count = Count + 1
          tempk(Count, 1) = binalar(i).kcesit(otno, k)
          count2 = 0
          For m = 1 To osayi
            If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = binalar(i).kcesit(otno,
k) Then
              count2 = count2 + 1
            End If
          Next
          tempk(Count, 2) = count2
        End If
      Next
    End If
  Next
End Sub

```

```

        End If
    Next
    Call sort(tempk(), 2, 1)
    For j = 1 To UBound(tempk, 1)
Line2:
        For ord = 2 To bsayi
            For k = bsayi To 1 Step -1
                If ischange <> Change Then
                    If binalar(k).kapasite > binalar(k).kalankap(otno) And binalar(k).kalankap(otno) >= tempk(j, 2) And
binalar(i).d_binaid(ord, 1) = binalar(k).b_id Then
                        For Z = 1 To ksayi
                            If binalar(k).kcesit(otno, Z) = tempk(j, 1) Then 'tasinacak kitapcik koduna sahip binada yer varsa
oraya ata
                                For m = 1 To osayi
                                    If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = tempk(j, 1)
Then
                                        ogrenciler(m).bina(otno) = binalar(k).b_id 'tasima
                                        End If
                                    Next
                                binalar(k).kalankap(otno) = binalar(k).kalankap(otno) - tempk(j, 2) 'kapasite guncelle
                                binalar(i).kalankap(otno) = binalar(i).kalankap(otno) + tempk(j, 2) 'kapasite guncelle
                                binalar(i).kcesitsayi(otno) = binalar(i).kcesitsayi(otno) - 1
                                For a = 1 To ksayi
                                    If binalar(i).kcesit(otno, a) = tempk(j, 1) Then
                                        binalar(i).kcesit(otno, a) = 0 'binadaki kitapcik kodlarımı guncelle
                                        Exit For
                                    End If
                                Next
                                ischange = ischange + tempk(j, 2)
                                j = j + 1
                                GoTo Line2
                            End If
                        Next
                    End If
                End If
            Next
        If ischange <> Change Then
            For k = bsayi To 1 Step -1
                If ischange <> Change Then

```

```

        If binalar(k).kapasite > binalar(k).kalankap(otno) And binalar(k).kalankap(otno) >= tempk(j, 2) And
        binalar(i).d_binaid(ord, 1) = binalar(k).b_id Then
            For m = 1 To osayi
                If ogrenciler(m).bina(otno) = binalar(i).b_id And ogrenciler(m).kitapcik(otno) = tempk(j, 1) Then
                    ogrenciler(m).bina(otno) = binalar(k).b_id 'tasıma
                End If
            Next
            binalar(k).kalankap(otno) = binalar(k).kalankap(otno) - tempk(j, 2) 'kapasite guncelle
            binalar(i).kalankap(otno) = binalar(i).kalankap(otno) + tempk(j, 2) 'kapasite guncelle
            binalar(k).kcesitsayi(otno) = binalar(k).kcesitsayi(otno) + 1 'kitapcik cesitliliği sayisi guncelle
            binalar(i).kcesitsayi(otno) = binalar(i).kcesitsayi(otno) - 1
            For Z = 1 To ksayi
                If binalar(k).kcesit(otno, Z) = 0 Then
                    binalar(k).kcesit(otno, Z) = tempk(j, 1) 'binadaki kitapcik kodlarını guncelle
                Exit For
            End If
            Next
            For Z = 1 To ksayi
                If binalar(i).kcesit(otno, Z) = tempk(j, 1) Then
                    binalar(i).kcesit(otno, Z) = 0 'binadaki kitapcik kodlarını guncelle
                Exit For
            End If
            Next
            ischange = ischange + tempk(j, 2)
            j = j + 1
            GoTo Line2

        End If
    End If
Next
End If

Next
Next
End If

Next
Next
End If

Next
Call dolulukorani(otno)

```

```

Call binakapsort(otno, 2)
For i = 1 To bsayi
conv = CLng(binalar(i).kalankap(otno)) * 100
If binalar(i).kapasite > binalar(i).kalankap(otno) And conv / binalar(i).kapasite > 100 - binadoluluk Then
    For j = 1 To osayi
line3:
        If ogrenciler(j).bina(otno) = binalar(i).b_id And ogrenciler(j).kitapcik(1) <> 0 Then
            For ord = 2 To bsayi
                For k = bsayi To 1 Step -1
                    If binalar(k).kalankap(otno) > 0 And binalar(k).kalankap(otno) < binalar(k).kapasite And
binalar(i).d_binaid(ord, 1) = binalar(k).b_id Then
                        ogrenciler(j).bina(otno) = binalar(k).b_id
                        binalar(i).kalankap(otno) = binalar(i).kalankap(otno) + 1
                        binalar(k).kalankap(otno) = binalar(k).kalankap(otno) - 1
                        j = j + 1
                        GoTo line3
                    End If
                Next
            Next
        End If
    Next
End If
conv = CLng(binalar(i).kalankap(otno)) * 100
If binalar(i).kapasite > binalar(i).kalankap(otno) And conv / binalar(i).kapasite > 100 - binadoluluk Then
    For j = 1 To osayi
line4:
        If ogrenciler(j).bina(otno) = binalar(i).b_id Then
            For k = bsayi To 1 Step -1
                If binalar(k).kalankap(otno) > 0 And binalar(k).kalankap(otno) < binalar(k).kapasite Then
                    ogrenciler(j).bina(otno) = binalar(i).b_id
                    binalar(i).kalankap(otno) = binalar(i).kalankap(otno) + 1
                    binalar(k).kalankap(otno) = binalar(k).kalankap(otno) - 1
                    j = j + 1
                    GoTo line4
                End If
            Next
        End If
    Next
End If
Next

```



```

End If

Next

For i = 1 To osayi
  For j = 1 To aosayi
    If aogrenciler(j).id = ogrenciler(i).id Then
      aogrenciler(j).bina(2) = ogrenciler(i).bina(2)
    End If
  Next
Next

Next

End Sub

Sub kcesitsort(otno As Integer, sorttype As Integer)
  Dim temp As bina
  If sorttype = 1 Then
    For i = 1 To bsayi - 1
      For j = i + 1 To bsayi
        If binalar(j).kcesitsayi(otno) < binalar(i).kcesitsayi(otno) Then
          temp = binalar(i)
          binalar(i) = binalar(j)
          binalar(j) = temp
        End If
      Next
    Next
  End If
  If sorttype = 2 Then
    For i = 1 To bsayi - 1
      For j = i + 1 To bsayi
        If binalar(j).kcesitsayi(otno) > binalar(i).kcesitsayi(otno) Then
          temp = binalar(j)
          binalar(j) = binalar(i)
          binalar(i) = temp
        End If
      Next
    Next
  End If
End Sub

```

End Sub

End Sub