

**Sınav Görevli Atama Problemine Bir Çözüm**  
**Önerisi: Anadolu Üniversitesi Uygulaması**

Dilek Küçük MATCI

Yüksek Lisans Tezi

Endüstri Mühendisliği Ana Bilim Dalı

Ocak 2014

## JÜRİ VE ENSTİTÜ ONAYI

Dilek Küçük MATCI'nın "Sınav Görevli Atama Problemine Bir Çözüm Önerisi: Anadolu Üniversitesi Uygulaması" başlıklı Endüstri Mühendisliği Anabilim Dalındaki, Yüksek Lisans Tezi 20.12.2013 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Adı Soyadı

İmza

Üye (Tez Danışmanı) : Yrd. Doç. Dr. Ilgın ACAR

Üye : Prof. Dr. Refail KASIMBEYLİ

Üye : Doç. Dr. Kamil ÇEKEROL

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

Enstitü Müdürü



## ÖZET

**Yüksek Lisans Tezi**

### **SINAV GÖREVLİ ATAMA PROBLEMİNE BİR ÇÖZÜM ÖNERİSİ: ANADOLU ÜNİVERSİTESİ UYGULAMASI**

**Dilek Küçük MATCI**

**Anadolu Üniversitesi**

**Fen Bilimleri Enstitüsü**

**Endüstri Mühendisliği Anabilim Dalı**

**Danışman: Yrd. Doç. Dr. Iğın Acar**

**2014, 71 Sayfa**

Anadolu Üniversitesi bünyesinde yapılan Açıköğretim sınavları yılda dört kez büyük bir organizasyon dâhilinde gerçekleştirilmektedir. Bu organizasyonun bir parçası da, sınavda görevlendirilmek üzere görevlileri uygun merkezlere atama işlemidir. Gerek atanacak görevli ve merkez sayılarının fazla olması, gerekse atama işlemlerinin el ile yapılmasının işlem süresini uzatmasından dolayı, bu prosedürü otomatik olarak gerçekleştirecek bir program ihtiyacı doğmuştur. Problemin çözümü amacıyla literatür taraması yapılmış ve genel olarak, m sayıda personelin n sayıda işe en küçük maliyetle atanmasını amaçlayan atama problemleri örnek alınarak bir matematiksel model kurulmuştur. Bu model ile en iyi çözüm elde edildikten sonra, otomasyonun geliştirilmesi amacıyla bir algoritma oluşturulmuş ve bu algoritmaya uygun bir otomasyon C# dilinde yazılmıştır.

**Anahtar Kelimeler:** Atama Problemi, Personel Atama Problemi

## ABSTRACT

Master of Science Thesis

### STAFF ASSIGNMENT PROBLEM AND THE SOLUTION APPROACH

Dilek KÜÇÜK MATCI

Anadolu University

Graduate School of Sciences

Industrial Engineering Program

Supervisor: Asst. Prof. Dr. İlgin ACAR

2014, 71 pages

The exams of the open education faculty are conducted four times a year with a great organization. As a part of this organization, academic staffs are being assigned to the tasks which are the part of the exams. Due to the assignment process is too hard to do by hand and it takes a lot of time, the assignment software was needed to automate the procedure. In order to solve the problem, literature survey was made. The assignment problems that aims to find the minimum cost of assignment of  $n$  jobs to  $m$  workers are taken as example to establish a mathematical model. After the optimal solution obtained with the model, an algorithm which implements the model is developed and an automation that suitable to the algorithm is written in C#.

**Keywords:** Assignment problem, Staff Assignment Problem, Personnel Assignment

## TEŞEKKÜR

Tez çalışmam boyunca bilgi ve tecrübelerini aktaran, beni yönlendiren ve destekleyen değerli hocam Yardımcı Doçent Doktor Ilgın ACAR'a en içten teşekkür ve saygılarımı sunarım.

Proje çalışması sırasında bilgi ve birikimleriyle çalışmama katkıda bulunan başta B.A.U.M. Müdür Yardımcısı Kamil ÇEKEROL olmak üzere, personel projesi sorumlusu Nilgün FIRAT ve B.A.U.M. genel yazılımlar sorumlusu Reha Oğuz ALTUĞ'a yardımlarından ve desteklerinden dolayı teşekkürlerimi sunarım.

Ayrıca çalışmalarım süresince gösterdikleri manevi destekten dolayı aileme sevgi ve teşekkürlerimi sunarım.

Dilek Küçük MATCI

Ocak 2014

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> .....	<b>i</b>
<b>ABSTRACT</b> .....	<b>ii</b>
<b>TEŞEKKÜR</b> .....	<b>iii</b>
<b>İÇİNDEKİLER</b> .....	<b>iv</b>
<b>ŞEKİLLER DİZİNİ</b> .....	<b>vi</b>
<b>TABLolar DİZİNİ</b> .....	<b>vii</b>
<b>1. GİRİŞ</b>	<b>1</b>
<b>2. ATAMA PROBLEMLERİ</b>	<b>9</b>
2.1. Görevli Başına En Fazla Bir Görev Atanan Modeller .....	10
2.2. Görevli başına birden çok görev atanan modeller.....	16
2.3. Çok Boyutlu Atama Problemleri.....	17
<b>3. PROBLEMİN TANIMI</b>	<b>19</b>
3.1. Önerilen Matematiksel Model.....	19
<b>4. UYGULAMA</b>	<b>22</b>
4.1. Genişletilmiş Kısıtlar ile Yapılan Atama Çalışmaları.....	34
<b>5. SONUÇ VE ÖNERİLER</b>	<b>37</b>
<b>KAYNAKLAR</b> .....	<b>40</b>
<b>Ek 1 . Personel Bilgileri</b> .....	<b>45</b>
<b>Ek 2 . Merkezlere Atanması Gereken Görevli Sayıları</b> .....	<b>46</b>
<b>Ek 3 . Modelin GAMS Dilinde Kodlanması</b> .....	<b>47</b>
<b>Ek 4 . GAMS ile Elde Edilen Çözüm</b> .....	<b>49</b>

<b>Ek 5 . Birinci Senaryo için Kullanılan Veriler ve Atama Sonucu .....</b>	<b>50</b>
<b>Ek 6 . İkinci Senaryo için Kullanılan Veriler ve Atama Sonucu .....</b>	<b>51</b>
<b>Ek 7 . Üçüncü Senaryo için Kullanılan Veriler ve Atama Sonucu .....</b>	<b>52</b>
<b>Ek 8 . Önerilen Algoritma .....</b>	<b>53</b>
<b>Ek 9 . Problemin Çözümü İçin C#'da Hazırlanan Programın Kodları.....</b>	<b>54</b>
<b>Ek 10 . Yeniden Düzenlenmiş Olan Önerilen Algoritma .....</b>	<b>62</b>

## ŞEKİLLER DİZİNİ

1.1 Profesörler genelinde görev alma sayıları.....	2
1.2 Doçentler genelinde görev alma sayıları.....	2
1.3 Yardımcı Doçentler genelinde görev alma sayıları.....	3
1.4 Araştırma Görevlileri genelinde görev alma sayıları.....	3
1.5 Öğretim Görevlileri genelinde görev alma sayıları.....	4
1.6 Uzmanlar genelinde görev alma sayıları.....	4
1.7 Okutmanlar genelinde görev alma sayıları .....	5
1.8 Genel Atama Süreci .....	6
4.1 Önerilen Algoritma .....	25
4.2 Yeniden Düzenlenmiş Olan Atama Algoritması .....	31



## TABLULAR DİZİNİ

1.1 Bölge ve Merkezler Tablosu .....	8
4.1 Atama Değerlendirme Verileri.....	22
4.2 Senaryo 1 için Atama Değerlendirme Verileri.....	23
4.3 Senaryo 2 için Atama Değerlendirme Verileri.....	24
4.4 Senaryo 3 için Atama Değerlendirme Verileri.....	24
4.5 Senaryo 1 için Atama Değerlendirme Verileri.....	28
4.6 Senaryo 2 için Atama Değerlendirme Verileri.....	29
4.7 Senaryo 3 için Atama Değerlendirme Verileri.....	29
4.8 Personelin 3 ayrı bölgeden birer il seçtiği durum .....	35
4.9 Personelin 3 ayrı bölgeden ikişer il seçtiği durum.....	36
4.10 Personelin 3 ayrı bölgeden toplam 6 il seçtiği durum.....	36
5.1 Farklı Tercih Senaryoları ile elde edilen sonuçlar .....	38

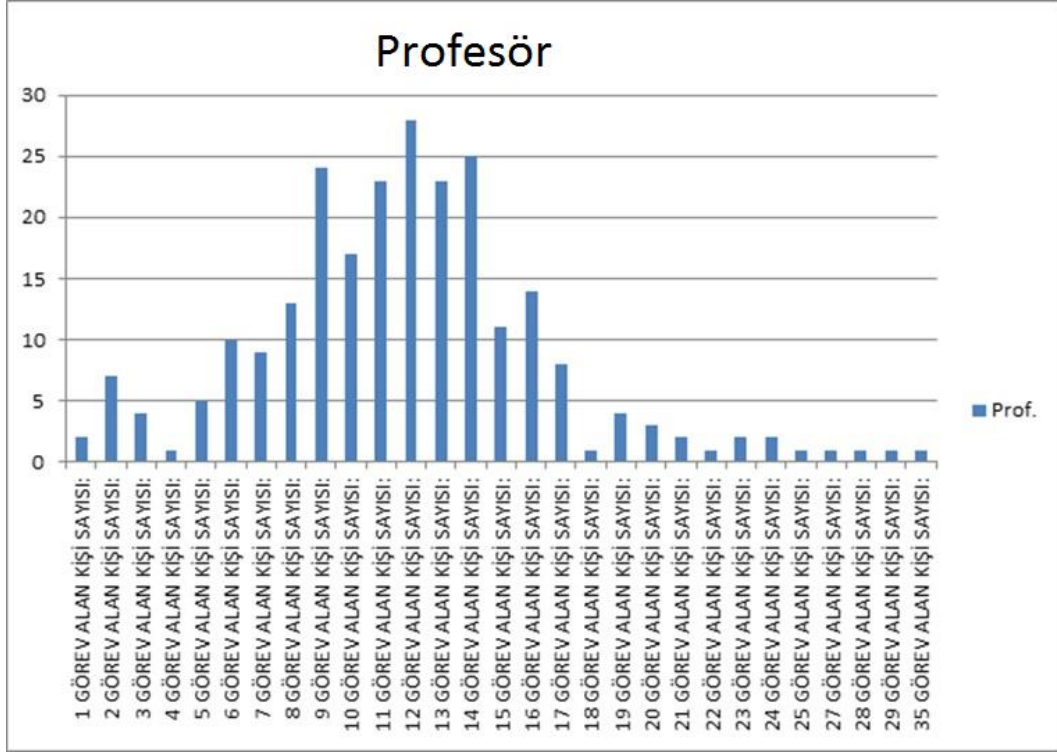
## 1. GİRİŞ

Anadolu Üniversitesi Açıköğretim Fakültesi 1982-83 öğretim yılından itibaren Türkiye'deki yüksek eğitim sisteminde yer almaktadır. Yurt dışında ve yurt içinde yaklaşık 12 lisans, 46 Önlisans programıyla eğitim hizmetini sürdürmektedir. Açıköğretim sistemi kapsamında, öğrencilerin başarısını ölçmek amacıyla yurt dışındaki merkezlerin yanı sıra yurt içinde 83 merkezde sınav organizasyonu yapılmaktadır.

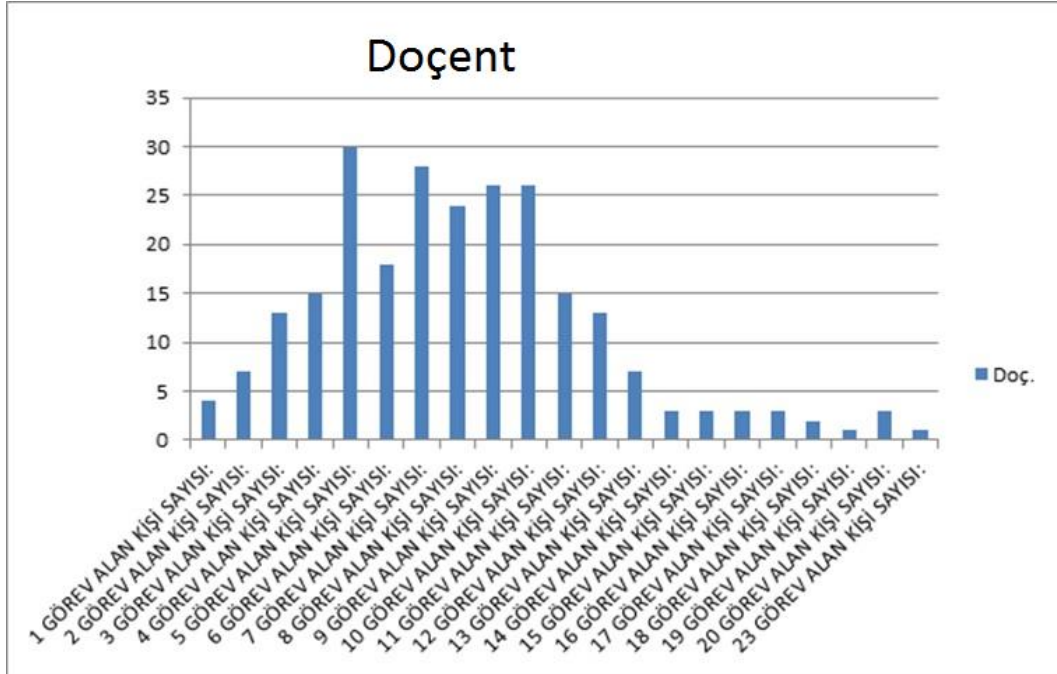
Organizasyon kapsamında, görevlilerin merkezlere atanması gerekmektedir. Bu atama işlemleri bir takım kısıtlar göz önüne alınmaya çalışılarak hâlihazırda el ile yapılmaktadır. Ancak işlemlerin el ile yapılması işlem sürelerinin uzamasına yol açmaktadır. Bununla birlikte, mevcut kısıtların uygulanmasında aksaklıklar yaşanabilmektedir. Ayrıca, bazı durumlarda kişilerin hep aynı merkezlerde görev almasının ya da hep aynı kişilerin görevlendirilmesinin önüne geçilememektedir. Adil bir görev dağılımı yapılabilmesi ve herkesin eşit şartlarda görev alabilmesi gerekliliğini karşılamak, çok kısıtlı süreçlerde işleri tamamlaması gereken organizasyon ekibinin karşı karşıya kaldığı zorluklardandır.

Bu çalışma, Açıköğretim sınavlarının yapıldığı yurt içinde bulunan 83 merkezde görev alması gereken akademik personelin geçmiş atama verileri ve modele ait kısıtlar göz önüne alınarak uygun merkeze ve uygun göreve atanmasını sağlamak amacıyla yapılmıştır. Bu doğrultuda atama süreci gözden geçirilmiş, tercihe dayalı bir atama sistemi kurgulanmış ve işlemleri gerçekleştiren bir otomasyon C # dilinde yazılmıştır. Çalışma kapsamında sınav görevi ataması yapılacak olan pozisyon Üniversite Temsilciliği görevidir.

Çalışmalara öncelikle atama süreci ve geçmiş atama sonuçları gözden geçirilerek başlanılmıştır. Geçmiş yıllarda gerçekleştirilmiş olan atama işlemleri, görevli unvanları göz önüne alınarak incelenmiştir. Profesör, doçent, yardımcı doçent, araştırma görevlisi, öğretim görevlisi, uzman ve okutmanlara ait görev alma sayıları sırasıyla Şekil 1.1 -1.7'de gösterilmiştir.

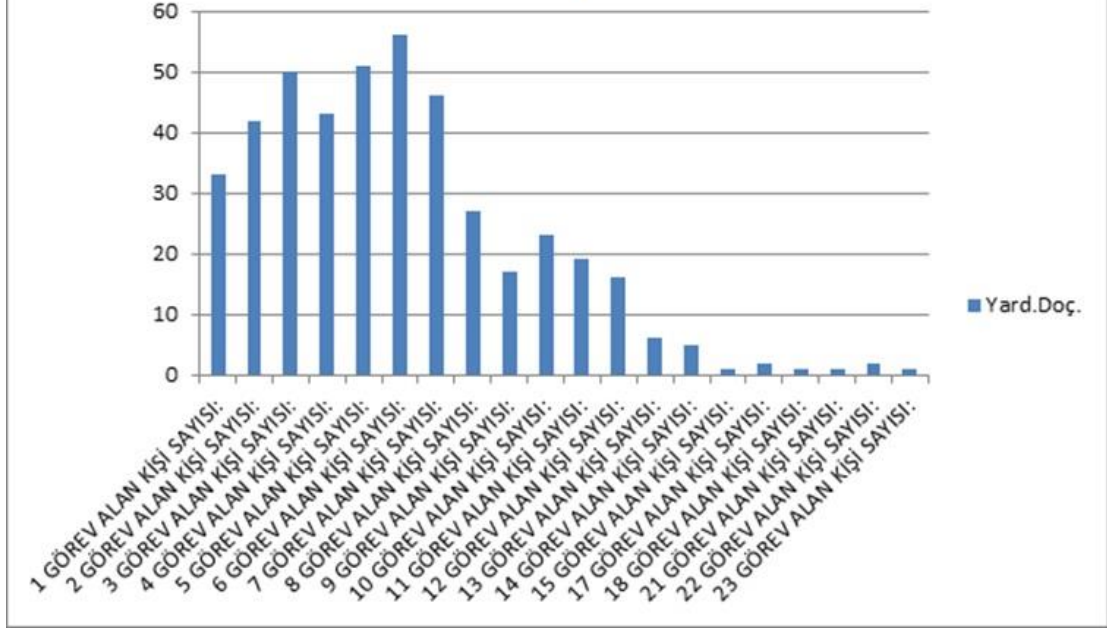


Şekil 1.1 Profesörler genelinde görev alma sayıları



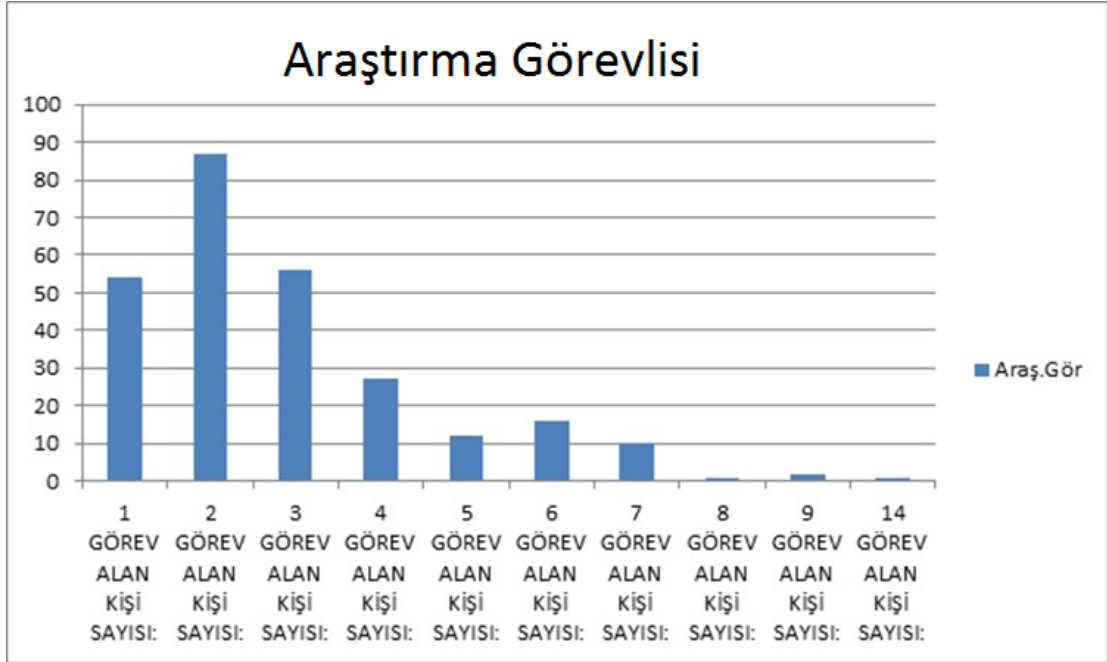
Şekil 1.2 Doçentler genelinde görev alma sayıları

## Yardımcı Doçent

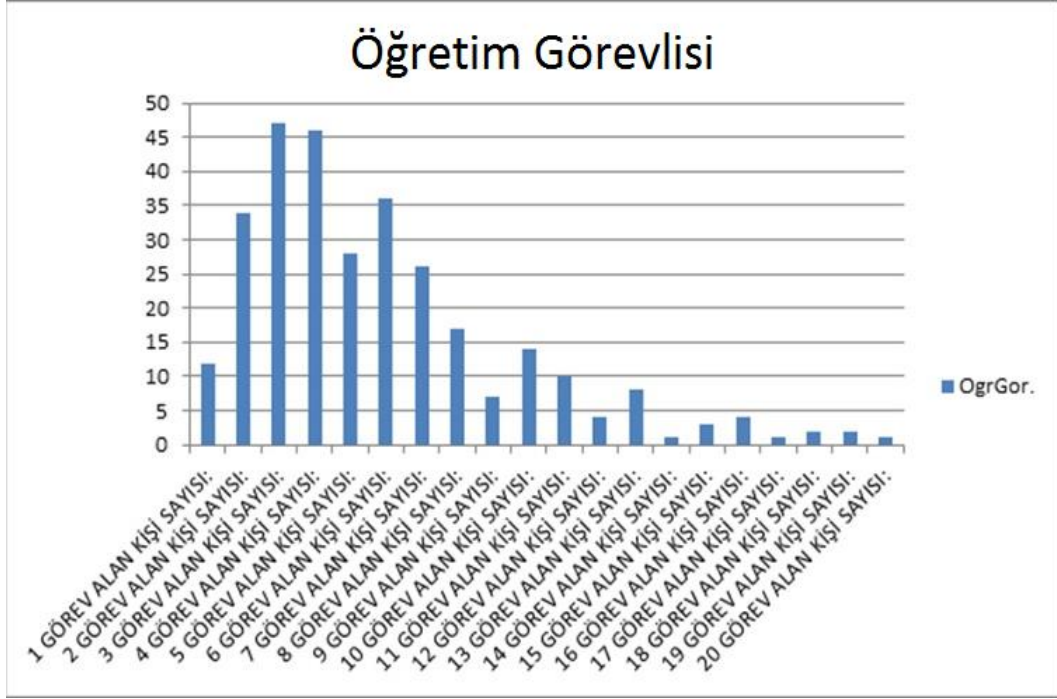


Şekil 1.3 Yardımcı Doçentler genelinde görev alma sayıları

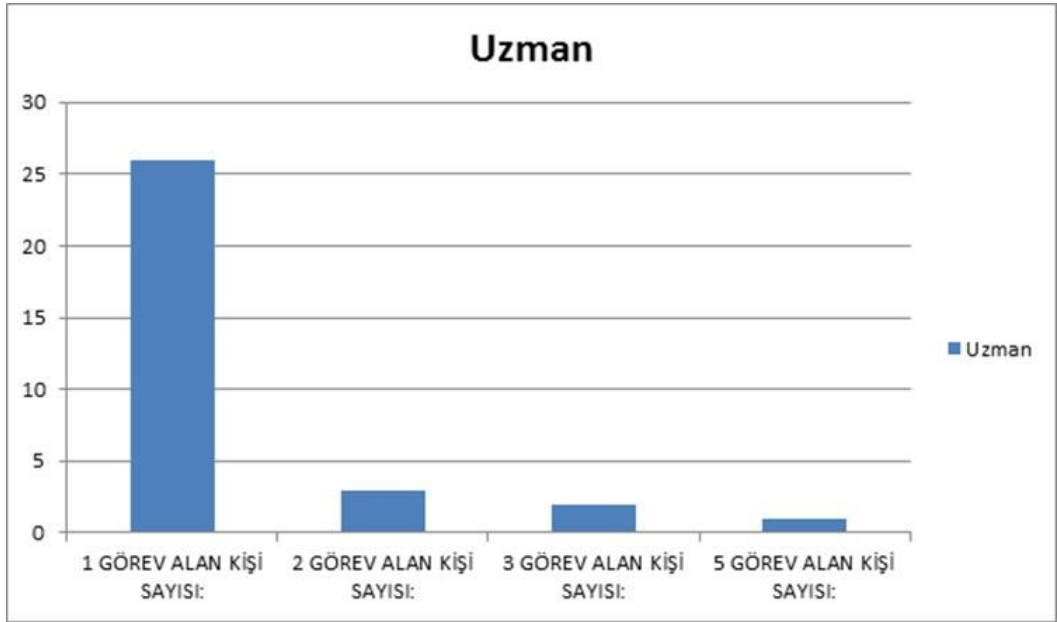
## Araştırma Görevlisi



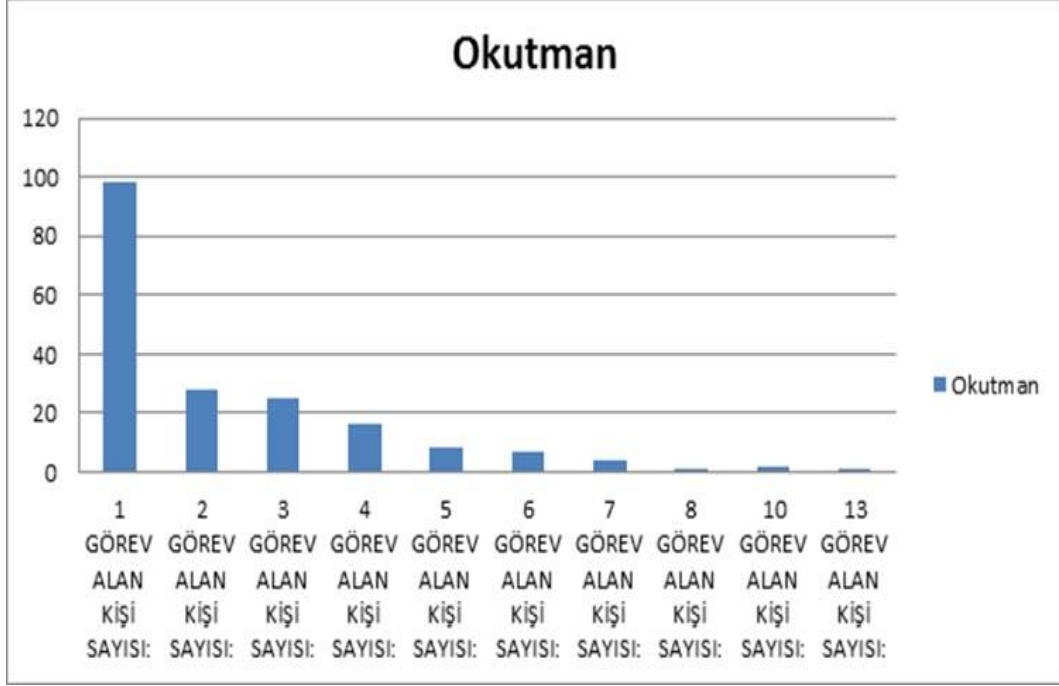
Şekil 1.4 Araştırma Görevlileri genelinde görev alma sayıları



Şekil 1.5 Öğretim Görevlileri genelinde görev alma sayıları



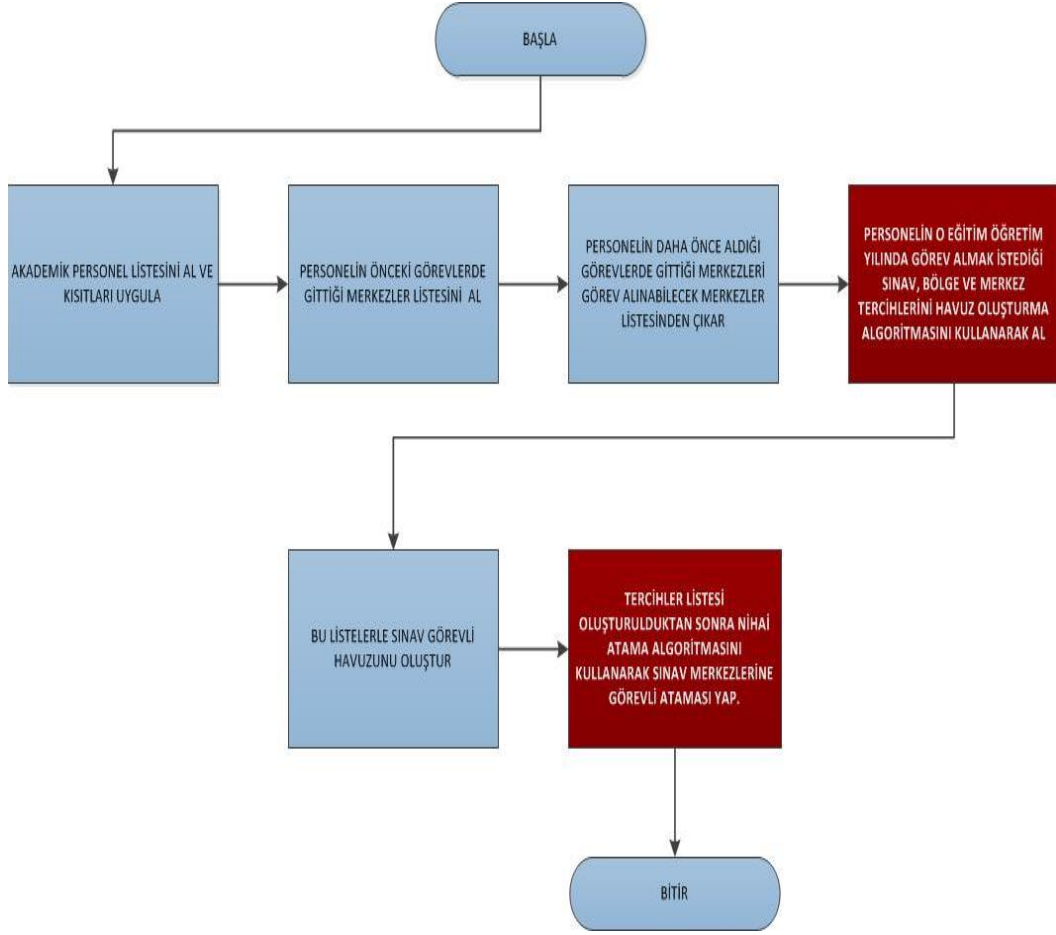
Şekil 1.6 Uzmanlar genelinde görev alma sayıları



Şekil 1.7 Okutmanlar genelinde görev alma sayıları

Görev alma sayıları üzerinde yapılan bu istatistiki çalışmada, bazı unvanlara sahip görevlilerin atanma sayılarının diğer görevlilere göre daha fazla olduğu gözlemlenmiştir.

Bu doğrultuda, görevlilerin memnuniyeti de göz önüne alınarak görev alma sayıları konusunda dengeli ve tercihe dayalı bir atama sistemi oluşturulması gerekliliği ortaya çıkmıştır. Bu amaçla genel atama süreci üzerinde çalışılarak yeni bir süreç tasarlanmıştır. Planlanan genel atama süreci Şekil 1.8 de gösterilmiştir.



Şekil 1.8 Genel Atama Süreci

Oluşturulan akışa göre atama işleminde kullanılmak amacıyla görevli havuzu oluşturulacaktır. Bu işlem görevlilerin yıl içindeki sınavlardan hangilerinde ve hangi merkezlerde görev almak istedikleri göz önüne alınarak yapılacaktır. Bu amaçla, sınav görevi almak isteyen akademik personel, her yılın Eylül ayında ilgili eğitim-öğretim dönemindeki sınavlardan hangisinde görevli olmak ve hangi merkezde görev almak istediklerini bir web arayüzü yardımıyla seçecektir. Bu web arayüzünde görevlilere seçim yapabilecekleri merkezler havuzu sunulacaktır.

Her görevlinin kurumda bulunduğu süre boyunca mümkün olduğunca tüm bölgelerde ve farklı merkezlerde görev alabilmesi, adil bir atama işlemi gerçekleştirmek için önemlidir. Bu nedenle kişinin daha önce görev yaptığı merkezlerde tekrar görevlendirilmemesi gerekmektedir. Bu doğrultuda,

görevlilerin tercihlerini yapabilecekleri merkezler, merkez seçim havuzu olarak kendilerine sunulurken, daha önceki görevlendirmelerde buldukları merkezler havuzdan çıkarılarak bu illeri tercih etmeleri dolayısı ile de bu illere atanmaları engellenmektedir.

Havuz oluşturulduktan sonra önerilen atama algoritması kullanılarak sınav merkezine görevli ataması yapılır. Atama algoritmasının uygulanması aşamasında yapılan tercihlerin yanında, önceki görevlendirmelerden toplanan puanlar ile görevlilerin unvan bilgileri kullanılır.

Görevlerden alınan puanlar şehirlerin puanlarından gelmektedir. Bu puanları belirlemek için ataması yapılacak merkezler, şehir özellikleri, genel olarak tercih edilme yoğunlukları, ulaşım, görevlendirme günleri gibi faktörler dikkate alınarak altı bölgeye ayrılmıştır. Şehrin özellikleri açısından en çok tercih edilen merkezler birinci bölgede yer alırken diğer merkezler aynı mantıkla diğer bölgelerde yer almıştır.

Bölgelerde bulunan merkezler ilgili bölgenin puanını aldığından, puanlandırma yapılırken en çok tercih edilen illerin bulunduğu bölgelere daha çok, tercih edilme oranı düşük olan illerin bulunduğu bölgelere daha az puan verilmesine dikkat edilmiştir. Böylelikle tercih edilme oranı daha az olan bölgedeki merkezlere atanan görevlilerin geçmiş görev puanı toplamları daha az olacağından bir sonraki atamada daha çok tercih edilen illere atanma olasılığı arttırılmıştır. Buna göre:

- Birinci bölgede yer alan merkezler altı puan
- İkinci bölgedeki merkezler beş puan
- Üçüncü bölgedeki merkezler dört puan
- Dördüncü bölgedeki merkezler üç puan
- Beşinci bölgedeki merkezler iki puan
- Altıncı bölgedeki merkezler bir puan almıştır.

Sınavlarda görevlendirilmesi uygun olan öğretim elemanlarının eldeki kısıtlar dâhilinde, atamasının yapılacağı merkez ve bölgeler Tablo 1.1’de bulunmaktadır:



**Tablo 1.1 Bölge ve Merkezler Tablosu**

BÖLGELER					
BölgeNo:1 Puanı:6	BölgeNo:2 Puanı:5	BölgeNo:3 Puanı:4	BölgeNo:4 Puanı:3	BölgeNo:5 Puanı:2	BölgeNo:6 Puanı:1
ANTALYA	MUĞLA	BURDUR	ISPARTA	UŞAK	ARDAHAN
İZMİR	AYDIN	KÜTAHYA	MANİSA	DÜZCE	MUŞ
ÇANAKKALE	AFYONKARAHİSAR	KARABÜK	DENİZLİ	KIRŞEHİR	SİİRT
SINOP	BALIKESİR	AMASYA	BOLU	NİĞDE	BİTLİS
ANKARA	BARTIN	GÜMÜŞHANE	ZONGULDAK	ÇORUM	ŞIRNAK
KONYA	NEVŞEHİR	KIRKLARELİ	KASTAMONU	TOKAT	HAKKARİ
BURSA	EDİRNE	KAHRAMANMARAŞ	KEŞAN	SİVAS	TUNCELİ
ESKİŞEHİR	TEKİRDAĞ	YALOVA	SAKARYA	KARS	BİNGÖL
ADANA	KOCAELİ	GEBZE	BİLECİK	AĞRI	
MERSİN	RİZE	ERZİNCAN	OSMANIYE	KIRIKKALE	
HATAY	VAN	ERZURUM	KİLİS	ÇANKIRI	
GAZİANTEP	ADIYAMAN	ELAZIĞ		AKSARAY	
ŞANLIURFA	MALATYA			KARAMAN	
MARDİN	SAMSUN			YOZGAT	
İSTANBUL	ARTVİN			BAYBURT	
TRABZON	KAYSERİ			IĞDIR	
DIYARBAKIR	ORDU			BATMAN	
	GİRESUN				

Problemin çözümüne yönelik literatür taraması gerçekleştirilmiş ve problemin bir atama problemi olarak ele alınmasına karar verilmiştir. Genel olarak atama problemlerinde, minimum maliyetle kaynakların hedeflere atanması gerçekleştirilmeye çalışılmıştır. Buradan hareketle ele alınan problemde görevlilerin merkezlere atanması sürecinde görev dağılımında adil olabilmek için en düşük puanlı görevlilere öncelik tanınmış, dolayısıyla merkeze atanan görevlilerin geçmiş görev puanları bir maliyet olarak kabul edilerek en küçüklenmeye çalışılmıştır.

Çalışmanın bundan sonraki bölümlerinde; literatür taraması, problemin tanımı ve oluşturulan matematiksel model, önerilen algoritma, algoritmaya uygun geliştirilen program, farklı tercih senaryoları kullanılarak yapılan uygulamalar ve sonuç bölümü yer alacaktır.

## 2. ATAMA PROBLEMLERİ

Belli sayıda kaynağın, değişik hedeflere dağıtılmasını amaçlayan atama problemleri, sonlu sayıda elemanı bulunan bir kümeden en iyi değerini elde edilmesi ile ilgilenen kombinatorik optimizasyon problemlerinden biridir.

Öner ve Ülengin [1] yaptıkları çalışmalarında, atama problemlerini genel olarak yapılması gereken (m) adet görev ve bu görevleri yapmaları için de (m) ayrı kişi bulunan, herhangi bir (i) kişinin (j) görevine atanması durumunda ( $c_{ij}$ ) maliyetinin doğduğu problemler olduğunu belirtmişlerdir. Her göreve mutlaka bir kişinin verilmesi ve bir kişinin sadece tek bir göreve atanması koşuluyla en küçük toplam maliyetle bir görevlendirme planı yapılması istendiği, diğer bir deyişle, en küçük toplam maliyeti doğuracak birebir kişi-görev eşleşmesinin bulunması gerektiğini belirtmişlerdir.

Özkan [2] çalışmasında, atama problemlerinde farklı kaynakların değişik görevlere etkin bir şekilde dağıtılmasının hedeflendiğini belirtmiştir. Ayrıca geleneksel atama problemlerinde kaynakların ve görevlerin eşit sayıda olması gerekirken, gerçek hayatta bu tür bir eşitlikle nadiren karşılaşıldığını eklemiştir. İstihdam edilen personel sayısının, yapılması gereken iş sayısından daha fazla olduğu problemler için aşağıda bulunan modeli önermiştir:

$$\text{Min. } Z = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \quad (2.1)$$

Kısıtlar:

$$\sum_{j \in J} x_{ij} = 1 \quad ; \quad i \in I_3 \quad (2.2)$$

$$\sum_{j \in J} x_{ij} \leq 1 \quad ; \quad i \in I_4 \quad (2.3)$$

$$\sum_{i \in I} x_{ij} = 1 \quad ; \quad j \in J \quad (2.4)$$

$$x_{ij} = \{0 \text{ veya } 1\} \quad i \in I, j \in J \quad (2.5)$$

Burada  $I_3$  işten çıkarılması düşünülmeyen personeli,  $I_4$  işten çıkarılabilecek personeli, I kümesi  $I_3$  ve  $I_4$  kümelerinin bileşimi olan personeli ve J kümesi de yapılması gereken işleri gösterir. Denklem (2.2), işten çıkarılmayacak personeli,

denklem (2.3) işten çıkarılabilecek personeli ve denklem (2.4) bir işin tek bir personel tarafından yapılabileceğini gösterir.

Bu modelde görevli sayısının yapılacak işlerden fazla olma durumu ele aldığımız problemle benzerlik göstermektedir. Ancak, denklem (2.4) ifade edilen bir işin tek bir personel tarafından yapılması gerekliliği sınav görevli atama probleminin yapısına uymamaktadır. Bunun nedeni sınav görevli atama probleminde bir merkeze belirlenmiş sayıda görevli atanması gerekliliğidir.

Atama problemleri, Burkard [3], Çela [4], Pentico [5], Özcan [6], Loiola ve arkadaşları [7] gibi araştırmacılar tarafından incelenmiş ve çalışmalar yapılmıştır. Bunlardan biri Burkard [3]'ün atama problemlerini sınıflandırdığı çalışmasıdır. Burkard, çalışmasında atama problemlerinin genel olarak  $n$  adet öğenin (iş, öğrenci vb.)  $n$  adet başka öğeye (makinalar, görevler vb.) nasıl atanacağı sorunu ele aldığını belirtmiş ve atama problemlerini doğrusal atama problemleri, çok boyutlu atama problemleri ve kareli atama problemleri olmak üzere üç ana gruba ayırmıştır.

Atama problemleri ile ilgili olarak yapılan bir diğer sınıflandırma ise Pentico [5] tarafından gerçekleştirilmiştir. Bu çalışmada atama problemleri ile ilgili son 50 yılda yapılan çalışmalar incelenmiş ve bu çalışma sonucunda atama problemleri üç ana sınıf altında toplanmıştır:

1. Görevli başına en fazla bir görev atanmış modeller,
2. Görevli başına birden çok görev atanmış modeller,
3. Çok boyutlu atama problemleri

### **2.1. Görevli Başına En Fazla Bir Görev Atanan Modeller**

Atama problemleri, Pentico [5]'nin sınıflandırması üzerinden incelendiğinde, ilk olarak görevli başına en fazla bir görev atanmış modelleri inceleyebiliriz. Klasik atama problemleri, K-Kardinalite atama problemleri, darboğaz atama problemi, dengelenmiş atama problemleri, minimum sapma atama problemi, çok kriterli atama problemi, yan kısıtları ile atama problemi, karesel atama problemleri bu grupta yer almaktadır.

Doğan [8] çalışmasında klasik atama problemlerini genel olarak, n sayıdaki işçinin m sayıdaki işe atanması şeklinde tanımlamış ve problemin modelini şu şekilde ifade etmiştir:

$c_{ij}$ : i işçisinin j işine atandığında oluşan maliyeti ifade eder.

$x_{ij}$  : i işçisi j işine atanırsa 1 atanmazsa 0 değerini alır.

$$\text{Min. } Z = \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \quad (2.6)$$

Kısıtlar:

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i = 1, 2, \dots, \dots, m \quad (2.7)$$

$$\sum_{i=1}^m x_{ij} = 1 \quad \forall j = 1, 2, \dots, \dots, n \quad (2.8)$$

$$x_{ij} = 0 \text{ veya } 1 \quad (2.9)$$

Klasik atama problemlerinin çözümü için önerilen yöntemlerin en çok bilineni Khun [9] tarafından geliştirilen Macar Algoritmasıdır. Bu yöntem ile maliyet matrisi her adımda sistematik bir şekilde yeni bir indirgenmiş matrisle dönüştürülerek atama problemine çözüm getirilmektedir. Algoritmanın alt yordamında matrisde sıfır içeren tüm hücreler en az sayıda çizgi ile kapatılmakta ve çizgilerin durumuna göre matris üzerinde işlem yapılmaktadır [10].

K-Kardinalite atama problemleri, atanacak m adet kaynak ve n adet hedefin bulunduğu ancak sadece k ( $k < m$ ,  $k < n$ ) adet hedef ve kaynağın eşleştirilmesi gereken durumlardır ve ilk olarak Amico ve Martello [11] tarafından sunulmuştur. Amico ve Martello [11] problemi kendi geliştirdikleri bir algoritma ile problemi çözmüşlerdir.

Volganent [12] çalışmasında, K-Kardinalite atama problemini standart doğrusal atama problemine dönüştüren transformasyon yöntemini kullanmıştır. Çeşitli k değerleri ve  $n=500$ 'ün üstünde olacak şekilde rassal test durumu için transformasyon yöntemini kullanmış ve standart kişisel bilgisayarlarla, çok kısa sürelerde bu yöntemi kullanarak K-Kardinalite atama problemlerinin çözümünü gerçekleştirmiştir.

İşgücü niteliğine göre atama problemleri, atanacak m kaynak ve n hedef olduğu halde bütün kaynakların tüm hedefleri yapabilecek nitelikte olmadığı durumlardır.

Caron [13] çalışmasında bu tür problemleri amaç fonksiyonundaki katsayıları nitelik ve öncelik ağırlıklarına göre değiştirmiş ve sonucu klasik atama problemi olarak çözmüştür.

Liu ve arkadaşları [14], İngilizce eğitimi veren bir kurumda, öğrencilerin uygun sınıflara atanması problemini incelemiştir. Daha önce sınıflar oluşturulurken kullanılan hazırlık sınav sonuçları kişilerin seviyelerinin belirlenmesi için çok uygun olmadığından bu yöntemin kötü sonuçlarının yıllar içinde görüldüğünü belirten Liu ve arkadaşları bu problemin çözümü amacıyla kümeleme veri madenciliği algoritmasını kullanmışlardır.

Volganent [15], bazı personellerin özelliklerinden dolayı sadece belli işlere atanabileceği işgücü niteliğine göre atama problemi üzerinde bir çalışma yapmıştır. Bu çalışmada problemi aşağıdaki gibi modellemiştir:

$$\text{Max } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.10)$$

Kısıtlar:

$$\sum_{j=1}^n a_{ij} x_{ij} \leq 1 \quad i = 1, 2, \dots, m \quad (2.11)$$

$$\sum_{i=1}^m a_{ij} x_{ij} \leq 1 \quad j = 1, 2, \dots, n \quad (2.12)$$

$$x_{ij} = 0 \text{ veya } 1 \quad i = 1, \dots, m; j = 1, \dots, n \quad (2.13)$$

Modelde  $c_{ij}$  j işinin i kişisine atanmasının getirisini ifade etmekte,  $x_{ij}$  eğer i kişisi j işine atanırsa 1, atanmazsa 0 değerini almaktadır.  $a_{ij}$  eğer i kişisi j işine uygunsa 1 uygun değilse 0 değerini almaktadır. Volganent [15], problemin çözümü için bir algoritma geliştirmiş ve bu algoritma ile problemin karmaşıklığını standart doğrusal atama problemleri seviyesine indirgeyerek çözmeyi sağlamıştır.

Yan kısıtları ile atama problemleri, her iş bir kişi tarafından yapılacak ve her kişi bir işi yapacak kısıtlarının yanında bütçe limiti, işçi nitelikleri ve zaman

kısıtları gibi atamayı etkileyen ilave kısıtlar bulunduğu durumlarda ortaya çıkar. Mazzola ve Neebe [16], Foulds ve Wilson [17] gibi araştırmacılar problemin çözümü için dal sınır algoritmasını, Wall [18], Majumdar ve Bhunia [19] genetik algoritmayı, Che ve arkadaşları [20] geliştirdikleri algoritmayı kullanarak yan kısıtlar ile atama problemini çözmüşlerdir. Toroslu [21], çalışmasında hiyerarşik kısıtlarla atama problemlerini incelemiş ve bu problemin çözümü için sezgisel bir yöntem önermiştir.

Ele aldığımız problemde bulunan atamaları gerçekleştirirken görevli tercihlerini göz önünde bulundurma gerekliliğinden dolayı problem yan kısıtları ile atama problemlerine benzerlik göstermektedir. Bu nedenle Majumdar ve Bhunia [19]'nın çalışması incelenmiştir. Bu çalışmada Majumdar ve Bhunia [19], yan kısıtları ile atama probleminin 0-1 tamsayı atama problemi olarak düşünülebileceğini belirtmiştir. Ayrıca çalışmasında aşağıdaki modeli önermiştir:

$$\text{Minimize } Z = \sum_{i=1}^n \sum_{j=1}^n C_{ij} x_{ij} \quad (2.14)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \quad (2.15)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (2.16)$$

$$\sum_{i=1}^n \sum_{j=1}^n r_{ij} x_{ij} \leq b \quad (2.17)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n \quad (2.18)$$

Burada  $C_{ij}$  i görevi j tesisine(makine, işçi) atandığında meydana gelecek sabit maliyet/zamanı ifade etmektedir.  $r_{ij}$  ise bu atamada gerçekleşecek olacak kaynak kullanımını, b ise kaynağın kapasitesini göstermektedir. Modelin kısıtlarında önerilen modelde her kaynağın bir hedefe ve her hedefin bir kaynağa atanması gerekliliğinin mevcut olduğu görülmektedir. Ancak ele aldığımız problemde her görevli en fazla bir merkeze atanabilir. Bunun nedeni, elimizde bulunan 1671 görevlinin ancak 379 adetinin merkezlere atanabileceğidir. Bunun yanında yine ele aldığımız problemde her hedefe sadece bir kaynak atanmakta, belirlenmiş olan görevli sayısı kadar görevli atanmaktadır. Bu nedenle Majumdar ve Bhunia [19]'nın modeli problemimize uygun değildir.

Elizondo [22] çalışmasında n adet öğrencinin m adet okula ataması problemini incelemiştir. Bu problem için önerdiği model aşağıda bulunmaktadır:

$$\text{Minimize } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.19)$$

*subject to :*

$$\sum_{i=1}^m x_{ij} \geq 1 \quad \forall j \quad (2.20)$$

$$\sum_{j=1}^n x_{ij} \leq p_i \quad \forall i \quad (2.21)$$

$$0 \leq x_{ij} \leq 1 \quad \forall i, j \quad (2.22)$$

$$x_{ij} = \begin{cases} 1, & \text{Eğer } j. \text{ öğrenci, } i. \text{ okula atandıysa} \\ 0, & \text{Eğer } j. \text{ öğrenci, } i. \text{ okula atanmadıysa} \end{cases}$$

Burada  $c_{ij}$  uzaklığı,  $p_i$  i. okulun kapasitesini ifade etmektedir. Amaç fonksiyonunda (2.19), atama sonucunda hesaplanan toplam uzaklık en küçüklenmeye çalışılmaktadır. Bu durum ele aldığımız problemin amacı olan toplam geçmiş görev puanının en küçüklenmesiyle benzerlik göstermektedir. Bunun yanında denklem (2.21)'de bulunan bir okula atanan öğrenci sayısı o okulun kapasitesinden küçük olmalıdır kısıtı, ele aldığımız problemde bulunan bir merkeze atanan görevli sayısı o merkezde görevlendirilmesi gereken kişi sayısına eşit olmalıdır kısıtına benzerlik göstermektedir. Ancak ele aldığımız problemde belirlenen sayının altında atama gerçekleştirilmesi söz konusu değildir.

Elloumi ve arkadaşları [23], sınavları uygun sınıflara atayabilmek için bir çalışma yapmış ve bu problem için aşağıdaki doğrusal programlama modelini önermiştir:

$$\text{Min } f(x) = \sum_{i=1}^m \sum_{j=1}^n b_j x_{ij} \quad (2.23)$$

*subject to :*

$$\sum_{j=1}^n b_j x_{ij} \geq a_i \quad \forall i \in \{1, \dots, m\} \quad (2.24)$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad \forall j \in \{1, \dots, n\} \quad (2.25)$$

$$x_{ij} = \begin{cases} 1, & \text{Eğer } j. \text{ sınıf, } i. \text{ sınava atandıysa} \\ 0, & \text{Eğer } j. \text{ sınıf, } i. \text{ sınıvala atanmadıysa} \end{cases}$$

Burada  $a_i$   $i$  sınavındaki kişi sayısını,  $b_j$  sınıfların kapasitesini göstermektedir. Problemin amaç fonksiyonunda sınıflar sınavlara atandığında boşta kalan kişilerin sayısının en küçüklenmesidir. Denklem (2.24), her sınav için toplam sıra sayısının kişi sayısından fazla olması gerektiğini ifade ederken, denklem (2.25) her sınıfın en fazla bir sınava atanabileceğini ifade etmektedir. Bu kısıt ele aldığımız problemde bulunan her görevli en fazla bir merkeze atanmalıdır kısıtıyla benzerlik göstermektedir. Ancak denklem (2.24)'de bulunan toplam sıra sayısının kişi sayısından fazla olması gerekliliği ele aldığımız problem ile uyumsuzdur.

Karimzadehgan ve Zhai [24], çalışmalarında çok yönlü uzmanlık gerektiren komite inceleme ekibi atama problemini tam sayılı programlama modeli olarak modellemiş ve kendi geliştirdikleri algoritma ile çözmüşlerdir. Modellerinde bulunan her makale gerekli sayıda uzmana atanmalıdır kısıtı ele aldığımız problemde bulunan her merkeze gerektiği sayı kadar personel atanmalıdır kısıtı ile benzerlik gösterse de, tüm makaleleri gözden geçirmek için yeterli sayıda uzman olmalıdır şeklindeki kapasite kısıtı problemimizin kısıtlarından farklılık göstermektedir.

Hosny [25], her eğitim öğretim döneminde asistanların laboratuvarlara atanması problemini bir fakülte atama problemi olarak ele almıştır. Ele aldığı problemde asistanların tercihleri bir yan kısıt olarak kabul edilmiştir. Hosny [25] bu problemin çözümü için yeni bir sezgisel algoritma geliştirmiştir.

Hübscher [26], öğrenci çalışma grubu ataması yaparken, grubun verimliliğini arttırmak amacıyla öğrencilerin kendi tercihlerini dikkate aldığını, ancak hem tercihler hem de genel ya da konuya özel kısıtlar göz önüne alınarak atamanın yapılmasının çok karışık ve zaman alan bir işlem olduğunu belirtmiştir. Bu nedenle grupları belirlemek amacıyla bir çalışma yapmış ve bu çalışmada tabu arama sezgiseli kullanılarak pek çok tercihi karşılayan sonuçlar elde edilmiştir.



Koopmans ve Beckman [27] tarafından ilk defa ortaya atılan karesel atama problemlerinde (KAP), eşit sayıda bulunan bölge ve tesisi toplam maliyeti en küçükleyecek şekilde bölgelere atamasını gerçekleştirmektedir. KAP'nin çözümü için çalışmalarında Bölte ve Thonemann [28], Nissen ve Paul [29], Ünsal M. [30] tavlama benzetimi yöntemini, Xia ve Yuan [31], Frieze ve Yadegar [32], Adams ve Johnson [33], doğrusallaştırma yöntemini, Stützel ve Dorigo [34] gibi araştırmacılar karınca kolonisi yöntemini kullanmış, Drezner [35] genetik algoritmayı, Jünger ve Kaibel [36] ise polyhedral çalışmalar gerçekleştirilmiştir.

## 2.2. Görevli başına birden çok görev atanmış modeller

Pentico [5]'nin çalışmasında belirttiği bir diğer kategori, görevli başına birden çok görev atanmış atama problemleridir ve genelleştirilmiş atama problemi (GAP), çok kaynaklı GAP, darboğaz GAP, karesel GAP'ni bu kategoriye dâhil etmiştir.

Yerleştirme problemleri, araç rotalama, grup teknolojisi, çizelgeleme gibi birçok uygulama alanı bulunan GAP'de kapasite kısıtları altında ve minimum atama maliyeti elde edilmeye çalışılarak kaynaklar hedeflere atanır. Her kaynak sadece bir hedefe atanır ancak hedefin kapasite kısıtları doğrultusunda her hedefe birden fazla kaynak atanabilir [37].

GAP'nin matematiksel modeli şu şekilde gösterilebilir:

$$\sum_{i=1}^n a_{ij}x_{ij} \leq b_j \quad \forall j, \quad 1 \leq j \leq m \quad (2.26)$$

$$\sum_{j=1}^m x_{ij} = 1 \quad \forall i, \quad 1 \leq i \leq n \quad (2.27)$$

$$x_{ij} \in \{0,1\} \quad \forall i, \quad 1 \leq i \leq n \quad 1 \leq j \leq m \quad (2.28)$$

k.a,

$$\text{enk} \sum_{i=1}^n \sum_{j=1}^m c_{ij}x_{ij} \quad (2.29)$$

Modelde yer alan parametrelerden  $c_{ij}$  i işini j görevlisine atama maliyetini,  $a_{ij}$  i işinin j görevlisine atanması sonucu oluşan kaynak kullanım miktarını,  $b_j$ , j görevlisinin kapasitesini göstermekte olup, I işler kümesini ( $i=1, \dots, n$ ), J

görevliler kümesini ( $j=1, \dots, m$ ) temsil etmektedir.  $x_{ij}$  ise modelin tek değişkeni olup  $i$  işi  $j$  görevlisine atanmışsa 1, aksi takdirde 0'dır. Amaç fonksiyonu (2.29), toplam atama maliyetini minimize ederken, ilk kısıt (2.26) kümesi görevlilerin kapasiteleriyle ilişkili olup, ikinci kısıt kümesi (2.27) her bir işin tek bir görevliye atanmasını garantilemektedir [37].

GAP'nin çözümü için kullanılan tekniklerin bazıları; Osman [38]'nin çalışmasında kullandığı tavlama benzetimi ve tabu arama algoritmaları, Tapkan, Özbakır ve Baykasoğlu [37]'nin çalışmalarında kullandığı arı algoritması, Diaz ve Fernandez [39]'in çalışmasında kullandığı tabu arama algoritması, Liu ve arkadaşlarının [40]'in kullandığı genetik algoritma olarak sayılabilir.

Lee ve Ma [41] birden fazla ekipmanı tek bir yere, yerdeki kaynakların izin verdiği ölçüde yerleştirme problemini genelleştirilmiş karesel atama problemi olarak ele almıştır. Bu problemin çözümü için doğrusallaştırma yaklaşımı ve dal sınır yöntemi ile uygulamalar yapmışlardır.

Hojati [42], çalışmasında vardiyaların işçilere atanması problemini incelemiştir. Burada vardiyaların kıdeme göre atanması gerekirken her işçinin en az bir vardiya almasını garantilemek gerekmektedir. Bu problemin çözümü için sezgisel bir yöntem önermiştir. Geliştirdiği yöntemi kullanarak New Brunswick Telefon Şirketinde bir uygulama gerçekleştirmiş ve en iyiye yakın sonuçlar elde etmiştir.

### 2.3. Çok Boyutlu Atama Problemleri

Düzlemsel üç boyutlu atama problemi, eksensel üç boyutlu atama problemi, üç boyutlu darboğaz atama problemi, çoklu zaman dilimlerinde atama problemleri Pentico [5]'nin belirttiği bir diğer kategori olan çok boyutlu atama problemleri sınıfındadır.

Çok boyutlu atama problemleri, Pierskalla [43] tarafından tanıtılmıştır. Çok boyutlu atama problemleri, Gilbert ve Hofstra [44]'nin  $p$  adet iş,  $q$  adet işçi ve  $r$  adet makinanın atanması örneğinde olduğu gibi birden fazla setin atanması olarak tanımlanmış ve aşağıdaki matematiksel modeli vermişlerdir:

$$EnK. \sum_{i=1}^r \sum_{j=1}^s \sum_{k=1}^t c_{ijk} x_{ijk} \quad (2.30)$$

s.t.

$$\sum_{i=1}^r x_{ijk} = 1 \quad , \forall j, k \quad (2.31)$$

$$\sum_{j=1}^s x_{ijk} \leq 1 \quad , \forall i, k \quad (2.32)$$

$$\sum_{k=1}^t x_{ijk} \leq 1 \quad , \forall i, j \quad (2.33)$$

$$x_{ijk} = 0 \text{ veya } 1 \quad , \forall i, j, k \quad (2.34)$$

$x_{ijk} = 1$  eğer satıcı  $i$ , müşteri  $j$ 'yi,  $k$  dönemi boyunca karşılarsa, karşılamazsa 0  $c_{ijk}$  bu atamanın maliyetidir. İlk kısıt seti (2.31) her müşterinin tam olarak 1 satıcı tarafından karşılandığını, ikinci kısıt seti (2.32) her satıcının her zaman döneminde en fazla bir müşteriyi karşılayacağını, üçüncü kısıt seti (2.33) müşterilerin birden fazla karşılaşamayacağını gösterir.

Çok boyutlu atama problemlerinin çözümü için, Pasilliao ve arkadaşları [45], Balas ve Saltzman [46], Larsen [47] gibi araştırmacılar tarafından dal-sınır algoritması kullanılmıştır. Bunun yanında, Poore ve Robertson [48]'in önerdiği Langaryan Gevşetmesine dayanan algoritmaları, Nguyen ve arkadaşlarının [49] çalışmalarında kullandığı çapraz entropi yöntemi gibi yöntemler kullanılmıştır. Frieze ve Yadegar [50] hangi öğrencinin, hangi okulda, hangi öğretici ile çalışacağı problemini, üç boyutlu atama problemi olarak ele alıp modellemişlerdir. Ayrıca bu problemin çözümü için Lagrangian gevşetme yöntemine dayalı bir algoritma geliştirmişlerdir.

Literatür kapsamında atama problemleri ve çözüm yöntemleri incelenmiş; ele alınan problem, literatürdeki atama problemleri ile karşılaştırılmıştır. Fakat problemin yapısı gereği ve probleme özgü kısıtların varlığından dolayı literatürdeki hiçbir atama problemi ile birebir örtüşmediği görülmüştür. Bundan dolayı akademik personel arasında adil bir atamayı gerçekleştirmek amacıyla, 83 merkezde yapılan Açıköğretim Fakültesi sınavlarında görev alma sayılarını dengelemeyi hedefleyen bir matematiksel model önerilmiştir.

### 3. PROBLEMİN TANIMI

Sınav görevli atama problemi göz önüne alındığında görev dağılımlarında adil bir yaklaşımı sağlamak için düşük puanlı personele öncelik verilmiştir. Bunun nedeni, az görev alan ya da düşük puanlı bölgelerde görev alan personelin diğerlerine göre daha az puan toplamasıdır. Bunu matematiksel modele yansıtmak için, amaç fonksiyonunda sınav merkezlerine atanan görevlilerin geçmiş görev puanları toplamının en küçüklenmesi kullanılacaktır. Geçmiş görev puanları görevlilerin daha önce gittiği merkezlerin puanlarının toplanmasıyla elde edilir. Probleme kabul edilen varsayımlar aşağıda yer almaktadır:

- Bir personel daha önce görev aldığı merkezde görev alamaz.
- Bir önceki sınavda görev alanlar mevcut sınavda görev alamaz.

Probleme etki eden kısıtlar şunlardır:

- Her bir görevli en fazla bir merkeze atanabilir.
- Her bir merkeze ancak o merkezde görev alması gereken personel sayısı kadar görevli atanabilir.
- Her bir merkeze en az bir Profesör atanmalıdır.
- Her bir merkeze en az bir Araş.Gör. atanmalıdır.

#### 3.1. Önerilen Matematiksel Model

Ele alınan sınav görevli atama problemi matematiksel olarak formüle edilmiştir. Matematiksel model ile mevcut kısıtlar dâhilinde amaç fonksiyonunu en küçükleyen görevli grubunu bulmak hedeflenmiştir. Amaç fonksiyonu, toplam geçmiş görev puanlarının en küçüklenmesi olarak belirlenmiştir.

#### Notasyon

Modele ait görevliler, sınav merkezleri ve unvan gruplarını içeren indis, parametre ve karar değişkeni tanımlanmıştır.

İndisler:

- $g=1 \dots G$ , görevliler: Problemde G adet görevli bulunmaktadır.
- $m=1 \dots M$ , merkezler: Problemde atanabilecek maksimum M adet merkez bulunmaktadır.

Parametreler:

- $P_g$  : Görevli g nin geçmiş görev puanı
- $Pr_g = \begin{cases} 1, & \text{Eğer görevli g profesör ise} \\ 0, & \text{Eğer görevli g profesör değil ise} \end{cases}$
- $A_g = \begin{cases} 1, & \text{Eğer görevli g araştırma görevlisi ise} \\ 0, & \text{Eğer görevli g araştırma görevlisi değil ise} \end{cases}$
- $S_m = m$ . merkeze atanması gereken görevli sayısı
- $T_{gm} = g$ . görevlinin m. merkezi tercih etme durumu.

Karar değişkeni:

$$Y_{gm} = \begin{cases} 1, & \text{Eğer g. görevli, m. merkeze atandıysa} \\ 0, & \text{Eğer g. görevli m. merkeze atanmadıysa} \end{cases}$$

### Model

$$\text{Min}Z = \sum_{m=1}^M \sum_{g=1}^G (P_g * Y_{gm} * T_{gm}) \quad (3.1)$$

$$\sum_{m=1}^M Y_{gm} \leq 1 \quad (g=1, \dots, G) \quad (3.2)$$

$$\sum_{g=1}^G Y_{gm} = S_m \quad (m=1, \dots, M) \quad (3.3)$$

$$\sum_{g=1}^G Pr_g Y_{gm} \geq 1 \quad (m=1, \dots, M) \quad (3.4)$$

$$\sum_{g=1}^G A_g Y_{gm} \geq 1 \quad (m=1, \dots, M) \quad (3.5)$$

$$Y_{gm} \in \{0,1\} \quad (3.6)$$

Amaç fonksiyonu (3.1), m merkezini tercih eden görevlilerin içinden atanan tüm g görevlilerinin geçmiş puanları toplamını en küçüklemek şeklinde ifade edilebilir. Burada kullanılan  $T_{gm}$  parametresi, modelin personel tercihlerinin dışında bir atama yapmasını engellemek amacıyla 1000 gibi büyük bir değer olarak kullanılmıştır.

Denklem (3.2)'de her bir görevlinin en fazla bir merkeze atanabilmesi kısıtı yer almaktadır. Denklem (3.3)'de her bir merkeze ancak o merkezde görev alması gereken personel sayısı kadar görevli atanabilir kısıtı yer almaktadır. Denklem (3.4)'te her bir merkeze en az bir Profesör atanmalıdır kısıtı yer almaktadır. Denklem (3.5)'te her bir merkeze en az bir Araş.Gör. atanmalıdır kısıtı yer almaktadır.

Çalışmanın devamında model, sınav organizasyon biriminden alınan veriler kullanılarak çalıştırılmıştır. Problemin sonuçları personel tercihleri ile doğrudan bağlantılı olduğundan farklı tercih senaryoları geliştirilip model bu yeni durumlar için çalıştırılmış ve sonuçlar incelenmiştir.

#### 4. UYGULAMA

Problemde 1671 adet personel ve 83 adet merkez bulunmaktadır. Her personelin üç adet tercih yaptığı durumun benzetimi yapılarak her görevli için rassal üç tercih oluşturulmuştur. Bu tercihler oluşturulurken personelin daha önce gittiği merkezler tercih havuzundan çıkarılmıştır. Ayrıca her personelin personel unvan bilgileri ve daha önce atanmış olduğu görev bilgileri sınav organizasyon biriminden alınmıştır. Görevlilerin daha önce almış görev bilgileri kullanılarak geçmiş görev puanları hesaplanmıştır. Bu bilgilerin 50 kişiden oluşan örneği Ek 1’de bulunan Personel Bilgileri tablosunda gösterilmiştir. Bunun yanında merkezlere atanması gereken personel sayıları sınav biriminden alınmış ve bu bilgiler Ek 2’de verilmektedir.

Problemin çözümü için oluşturulan matematiksel model, GAMS ile kodlanmıştır. Modelin kaynak kodları Ek 3’te bulunmaktadır. Hazırlanan model, sınav biriminden alınmış veriler ve her görevli için rassal olarak üçer merkez seçerek oluşturulmuş olan tercihlerle çalıştırılarak optimal sonuç elde edilmiştir. Atama sonucunun 50 kişilik örneği Ek 4’te gösterilmektedir. Buna göre atama sonucunu gösteren veriler Tablo 4.1’de bulunmaktadır.

**Tablo 4.1 Atama Değerlendirme Verileri**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	1839	379	376	99%

Yapılan atamanın sonucunda toplam geçmiş görev puanı 1839 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 376’sı tercihinde atanmıştır. Yapılan atamanın amacı toplam puanı en küçükleyecek şekilde

personelin tercihlerine atanması olduğundan bu atama %99 oranında başarılı bir atama olarak kabul edilebilir.

Çıktıların kısıtlara uygunluğu test edildikten sonra, farklı tercih senaryoları türetilerek model çalıştırılmıştır. Buna göre ilk senaryo personelin üç ayrı bölgeden birer merkez seçtiği durumdur. Bu senaryo için türetilmiş olan tercihleri de içeren atama sonucunun 50 kişilik örneği Ek 5 te verilmiştir. Model hazırlanan yeni veriler kullanılarak tekrar GAMS ile çalıştırılmıştır. Buna göre yapılan atama sonucunun değerlendirmesi Tablo 4.2’de bulunmaktadır.

**Tablo 4.2 Senaryo 1 için Atama Değerlendirme Verileri**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	1715	379	358	95%

Yapılan atamanın sonucunda hesaplanan toplam geçmiş görev puanı 1715 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 358’i tercihine atanmıştır. Bu sayı tercihlere atanma oranının %95 olduğunu göstermiştir.

Senaryolardan ikincisi personelin üç ayrı bölgeden ikişer merkez olmak üzere toplam altı merkez seçtiği durumdur. Bu senaryo için türetilmiş olan tercihleri de içeren atama sonucunun 50 kişilik örneği Ek 6’da verilmiştir. Model hazırlanan yeni veriler kullanılarak tekrar GAMS’te çalıştırılmıştır. Buna göre yapılan atama sonucunun değerlendirmesi Tablo 4.3’te bulunmaktadır.



**Tablo 4.3 Senaryo 2 için Atama Değerlendirme Verileri**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	1257	379	379	100%

Yapılan atamanın sonucunda toplam geçmiş görev puanı 1257 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 379'u tercihine atanmıştır. Bu atama sonucunda personelin %100'ü tercihlerine atanmıştır.

Senaryolardan sonuncusu personelin üç ayrı bölgeden toplamda rassal olarak altı merkez seçtiği durumdur. Bu senaryo için türetilmiş olan tercihleri de içeren atama sonucunun 50 kişilik örneği Ek 7'de verilmiştir. Model hazırlanan yeni veriler kullanılarak tekrar GAMS'te çalıştırılmıştır. Buna göre yapılan atama sonucunun değerlendirilmesi Tablo 4.4'te bulunmaktadır.

**Tablo 4.4 Senaryo 3 için Atama Değerlendirme Verileri**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	1244	379	379	100%

Yapılan atamanın sonucunda toplam geçmiş görev puanı 1244 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 379'u tercihine atanmıştır.

Model farklı senaryolar için test edildikten sonra, son kullanıcıların modelin kullanımında yaşayabileceği zorluklar nedeniyle, kullanıcı dostu bir arayüz içeren,



Geliştirilen programın çıktıları sınav merkezleri listesinin sıralamasına göre değişkenlik gösterdiği için, atama 1000 farklı sıralama oluşturularak yapılmakta ve bu atama sonuçları arasından en iyi olan seçilmektedir. Bu çalışmada kullanılan merkez sıralama sistemi üzerinde yeni çalışmalar yapılarak en iyi sonuca yaklaşılabılır. Ancak, bu işlemlerin toplam atama süresini etkileyeceği unutulmamalıdır.

Şekil 4.1 de gösterilmiş olan algoritmanın adımları şöyledir:

• Adım1: Sınav tercih aşamasında oluşturulan havuzda bulunan görevlilerin tercih ettiği üç merkezi içeren tercihler listesini al.

• Adım2: Havuzda bulunan tüm görevlilerin unvan bilgilerini al.

• Adım3: Havuzda bulunan tüm görevlilerin geçmiş görev puanlarını hesapla.

• Adım4: Tüm merkezleri puanlarına göre büyükten küçüğe sırala

• Adım5: Sıradaki merkezi al.

• Adım6: Atanmadıysa Adım 10'a git

• Adım7: Merkeze en az bir profesör atanıp atanmadığını kontrol et.

• Adım8: Eğer atanmadıysa merkezi seçen personel listesinde unvanı profesör olan görevli olup olmadığını kontrol et.

• Adım9: Profesör var ise puanı en düşük olanı merkeze ata. Ve Adım 10'a git.

• Adım10: Profesör yok ise Adım 10'a git.

• Adım11: Listede başka merkez olup olmadığını kontrol et.

• Adım12: Başka merkez var ise Adım 5'e git.

• Adım13: Başka merkez yok ise merkez listesinin başına dön.

• Adım14: Sıradaki merkezi al

• Adım15: Merkeze en az bir profesör atanıp atanmadığını kontrol et.

• Adım16: Eğer atanmadıysa merkezin bulunduğu bölgede bulunan diğer merkezleri seçen ancak ilk turda atanmayan ve unvanı profesör olan görevli olup olmadığını kontrol et.

• Adım17: Profesör var ise puanı en düşük olanı merkeze ata

• Adım18: Profesör yok ise listede başka merkez olup olmadığını kontrol et.

• Adım19: Başka merkez var ise Adım 14'e git

- Adım20: Başka merkez yok ise merkez listesinin başına git.
- Adım21: Sıradaki merkeze geç
- Adım22: Merkeze en az bir profesör atanıp atanmadığını kontrol et.
- Adım23: Atanmadıysa Adım26'ya git.
- Adım24: Eğer atanmadıysa tüm Türkiye genelinde ilk iki turda atanmamış profesör olup olmadığını kontrol et.
- Adım25: Profesör var ise puanı en düşük olan görevliyi merkeze ata.
- Adım26: Profesör yok ise merkezin listedeki son merkez olup olmadığını kontrol et.
- Adım27: Son merkez değil ise Adım22'e dön.
- Adım28: Son merkez ise merkezi profesör bulunamayanlar listesine ekle ve listenin başına dön.
- Adım29: Adım5 ve Adım28 arasındaki tüm adımları unvanı araştırma görevlisi olanlar için uygula.
- Adım30: Listenin başına dön ve Adım5 ve Adım28 arasındaki tüm adımları unvan ayırımı yapmadan merkeze atanması gereken personel sayısını tamamlayana kadar uygula.
- Adım31: Atama sonucunu kaydet.
- Adım32: Toplam atama sayısının 1000 olup olmadığını kontrol et.
- Adım33: 1000 değil ise merkez sıralaması için rassal sayı üreterek yeni bir sıralama oluştur ve Adım 5'ya dön
- Adım34: 1000 ise en iyi atama sonucunu seç.
- Adım34: Atama sonucunu veritabanına yaz.

Geliştirilen algoritmada genel yaklaşım, bir merkezi tercih eden personeller listesi içinde bulunan geçmiş görev puanı en düşük olan personele öncelik tanınması yönündedir. Böylece tüm personelin toplam göreve atanma sayıları ve tercih edilen sınav merkezlerine gitme sayılarının dengelenmesi amaçlanmaktadır.

Problemin kısıtlarında bulunan her merkeze en az bir profesör ve bir araştırma görevlisi atanmalıdır kuralını gerçekleştirmek amacıyla, algoritma profesör bulma, araştırma görevlisi bulma ve diğer görevlileri bulma olmak üzere üç ana bölüme ayrılır. Her ana bölüm de kendi içinde üç iterasyon içerir. Bu

iterasyonların ilkinde merkezi tercih eden görevliler içinde merkeze en uygun olan aranır ve bu işlem her merkez için tekrarlanır. İkinci iterasyonda eğer merkeze atanması gereken görevli atanmadıysa merkezin bulunduğu bölgedeki diğer merkezlerden birini seçmiş ancak ilk iterasyonda atanmamış olan görevlilerden en uygunu merkeze atanır. Bu işlemler de bütün merkezler için tekrarlanır. Son iterasyonda ise merkeze atanması gereken görevliler atanmadı ise tüm merkezler içinde tercih yapmış ancak atanmamış olan görevliler aranır ve bu görevlilerden en uygun olan merkeze atanır.

Şekil 4.1’de verilen algoritmayı çalıştıran bilgisayar programı birim içinde kullanımı yaygın, geliştirilmesi ve yönetilmesi açısından daha pratik olan C# dilinde yazılmıştır. Yazılmış olan programa ait kaynak kodları Ek 9’da bulunmaktadır.

Daha önce, GAMS modeli için kullanılan farklı tercih senaryoları, geliştirilen program ile de çalıştırılmıştır. Buna göre ilk senaryo personelin üç ayrı bölgeden birer merkez seçtiği durumdur. Bu senaryo hazırlanan programla çalıştırılmıştır. Buna göre yapılan atama sonucunun değerlendirilmesi Tablo 4.5’te bulunmaktadır.

**Tablo 4.5 Senaryo 1 için Atama Değerlendirme Verileri**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	1786	379	356	93,40%

Yapılan atamanın sonucunda hesaplanan toplam geçmiş görev puanı 1786 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 356’sı, toplam atanan personelin %93,4’ü tercihine atanmıştır.

Senaryolardan ikincisi personelin üç ayrı bölgeden ikişer merkez olmak üzere toplam altı merkez seçtiği durumdur. Bu senaryo için hazırlanan veriler kullanılarak tekrar program çalıştırılmıştır. Buna göre yapılan atama sonucunun değerlendirilmesi Tablo 4.6’ da bulunmaktadır.

**Tablo 4.6 Senaryo 2 için Atama Değerlendirme Verileri**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	1363	379	379	100%

Yapılan atamanın sonucunda hesaplanan toplam geçmiş görev puanı 1363 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 379’u tercihine atanmıştır.

Senaryolardan sonuncusu personelin üç ayrı bölgeden toplamda rassal olarak altı merkez seçtiği durumdur. Bu senaryo için türetilmiş veriler kullanılarak tekrar program çalıştırılmıştır. Buna göre yapılan atama sonucunun değerlendirilmesi Tablo 4.7’de bulunmaktadır.

**Tablo 4.7 Senaryo 3 için Atama Değerlendirme Verileri**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	1325	379	379	100%

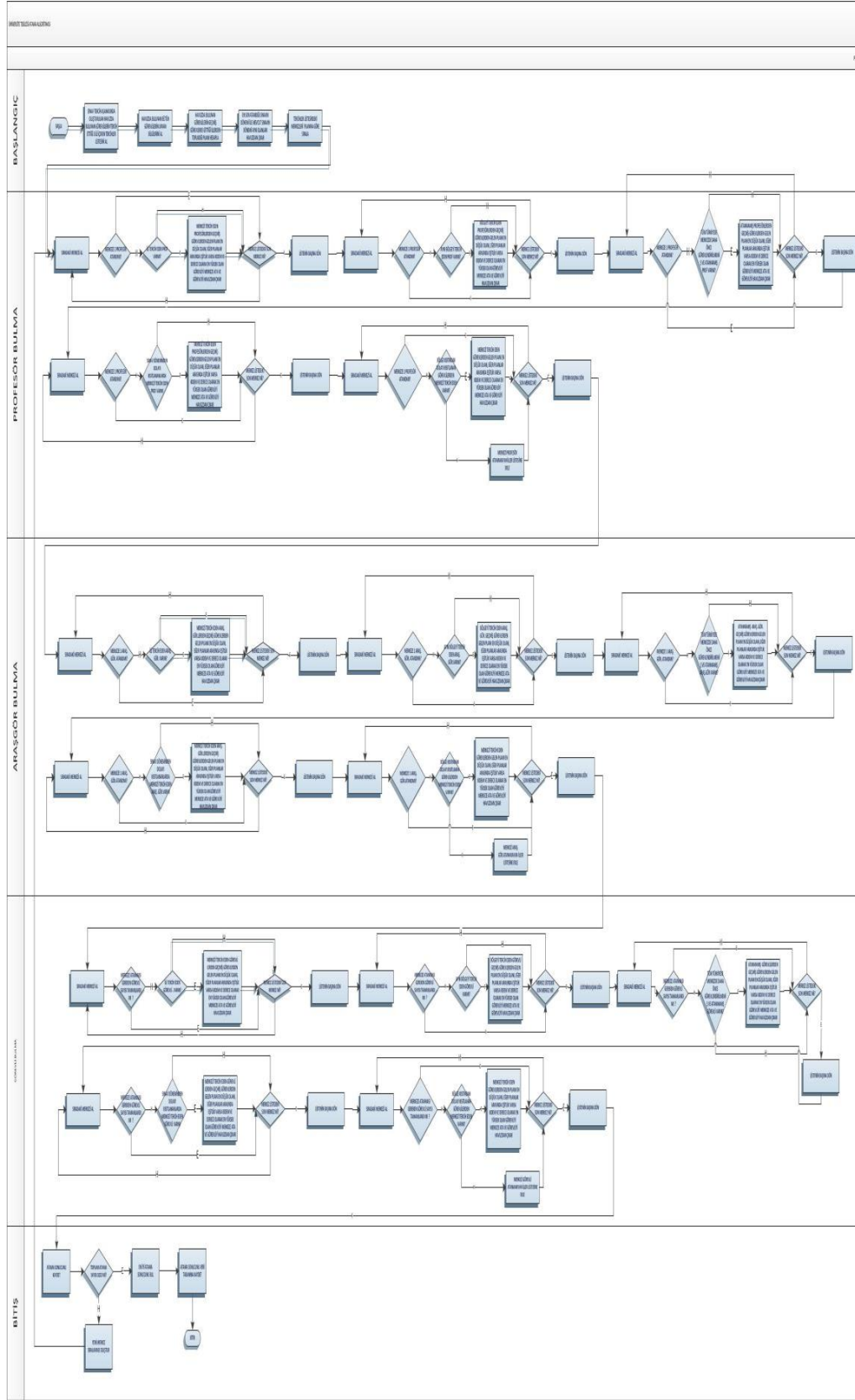
Yapılan atamanın sonucunda hesaplanan toplam geçmiş görev puanı 1325 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 379’u

tercihine atanmıştır. Sonuç olarak, farklı senaryolarla çalıştırılan program ile GAMS ile elde edilen sonuçlara yakın sonuçlar elde edilmiştir.

Problem bu haliyle çözüldükten sonra, sınav organizasyonu tarafından personel havuzunda bulunan her görevlinin tüm merkezlere eşit şekilde dağıtılabilmesi için, daha önce görev aldığı merkezleri tercih listesinden çıkarmanın yanında, belirlenmiş olan tüm bölgelerde eşit şekilde dağıtılmasının gerekliliği bildirilmiştir. Bunu sağlamak için personelin belirlenmiş olan altı bölgenin hepsinde birer kez görevlendirilmeden, daha önce görev aldığı bölgeye gönderilmemesi gerekmektedir. Son altı görevinde, belirlenmiş olan bölgelerin hepsinde görevlendirildi ise tura en baştan başlayabilecektir. Ayrıca personelin her defasında farklı sınav dönemlerinde görev almasının sağlanması gerekliliği bildirilmiştir. Bu amaçla da, personelin en son görev aldığı sınavın dönemiyle mevcut sınavın döneminin aynı olup olmadığının kontrol edilmesi gerekmektedir.

Talep edilen değişiklikler Şekil 4.1’de bulunan algoritmaya yansıtılmış ve Şekil 4.2’de bulunan algoritma elde edilmiştir. Ayrıca algoritmanın gerçek boyutlarıyla basılı hali Ek 10’da verilmiştir.

Şekil 4.2 Yeniden Düzenlenmiş Olan Atama Algoritması





Aşağıda yeniden düzenlenmiş algoritmanın adımları bulunmaktadır:

- Adım1: Sınav tercih aşamasında oluşturulan havuzda bulunan görevlilerin tercih ettiği üç ili içeren tercihler listesini al
- Adım2: Havuzda bulunan tüm görevlilerin unvan bilgilerini al.
- Adım3: Havuzda bulunan tüm görevlilerin geçmiş görev bilgilerini al son sınav dönemlerini belirle ve geçmiş görev puanlarını hesapla.
- Adım4: Görevli havuzunda bulunan görevlilerden en son atandığı sınavın dönemi ile mevcut sınavın döneminin aynı olanlar çıkarılır.
- Adım5: Tüm merkezleri puanlarına göre büyükten küçüğe sırala
- Adım 6: Sıradaki merkezi al.
- Adım7: Merkeze en az bir profesör atanıp atanmadığını kontrol et.
- Adım8: Eğer atanmadıysa merkezi seçen personel listesinde unvanı profesör olan görevli olup olmadığını kontrol et.
- Adım9: Profesör var ise puanı en düşük olanı merkeze ata. Adım 11'e git.
- Adım10: Profesör yok ise Adım 11'a git.
- Adım 11: Listede başka merkez olup olmadığını kontrol et.
- Adım 12: Başka merkez var ise Adım 6'ya git.
- Adım 13: Başka merkez yok ise listenin başına git
- Adım 14: Sıradaki merkezi al
- Adım 15: Merkeze en az bir profesör atanıp atanmadığını kontrol et.
- Adım16: Eğer atanmadıysa merkezin bulunduğu bölgede bulunan diğer merkezleri seçen ancak ilk turda atanmayan ve unvanı profesör olan görevli olup olmadığını kontrol et.
- Adım17: Profesör var ise puanı en düşük olanı merkeze ata
- Adım18: Profesör yok ise listede başka merkez olup olmadığını kontrol et.
- Adım19: Başka merkez var ise Adım 14'e git
- Adım20: Başka merkez yok ise listenin başına git.
- Adım21: Sıradaki merkeze geç
- Adım22: Merkeze en az bir profesör atanıp atanmadığını kontrol et.
- Adım23: Eğer atanmadıysa tüm Türkiye genelinde ilk iki turda atanmamış profesör olup olmadığını kontrol et.

- Adım24: Profesör var ise puanı en düşük olan görevliyi merkeze ata.
- Adım25: Profesör yok ise merkezin listedeki son merkez olup olmadığını kontrol et.
- Adım26: Son merkez değil ise Adım21'e dön.
- Adım27: Son merkez ise listenin başına dön.
- Adım28: Sıradaki merkeze geç
- Adım29: Merkeze en az bir profesör atanıp atanmadığını kontrol et.
- Adım30: Eğer atanmadıysa en son görev aldığı sınavın dönemi ile ataması yapılan sınavın dönemi aynı olduğundan dolayı kısıtlanmış personel arasından merkezi tercih eden profesör olup olmadığını kontrol et.
- Adım31: Profesör var ise puanı en düşük olan görevliyi merkeze ata.
- Adım32: Profesör yok ise merkezin listedeki son merkez olup olmadığını kontrol et.
- Adım33: Son merkez değil ise Adım28'e dön.
- Adım34: Son merkez ise listenin başına dön.
- Adım35: Sıradaki merkeze geç
- Adım36: Merkeze en az bir profesör atanıp atanmadığını kontrol et.
- Adım37: İlgili merkezin bulunduğu bölgeye daha önce gitmiş olduğundan dolayı kısıtlanmış olan görevliler arasında merkezi tercih eden profesör olup olmadığını kontrol et.
- Adım38: Profesör var ise puanı en düşük olan görevliyi merkeze ata.
- Adım39: Profesör yok ise merkezin listedeki son merkez olup olmadığını kontrol et.
- Adım40: Son merkez değil ise Adım35'e dön.
- Adım41: Son merkez ise merkezi profesör atanamayan merkezler listesine ekle
- Adım42: Adım5 ve Adım41 arasındaki tüm adımları unvanı araştırma görevlisi olanlar için uygula.
- Adım43: Listenin başına dön ve Adım5 ve Adım41 arasındaki tüm adımları unvan ayırımı yapmadan merkeze atanması gereken personel sayısını tamamlayana kadar uygula.
- Adım 44: Atama sonucunu kaydet.

- Adım 45: Toplam atama sayısının 1000 olup olmadığını kontrol et.
- Adım 46: 1000 değil ise merkez sıralaması için rassal sayı üreterek yeni bir sıralama oluştur ve Adım 6'ya dön
- Adım 46: 1000 ise en iyi atama sonucunu seç.
- Adım 48: Atama sonucunu veritabanına yaz.

Algoritmanın yeniden düzenlenmiş halinde, öncelikle görevlilerden en son atandığı sınavın dönemi ile mevcut sınavın döneminin aynı olanlar havuzdan çıkarılır. Daha sonra personelin görev aldığı sınavlarda bulunduğu bölgeler listelenir ve tüm adımlarda personelin ilgili merkezin bulunduğu bölgeye gidip gitmediği kontrol edilir. Eğer daha önce belirlenmiş olan algorithmada bulunan dört iterasyonda merkez için uygun görevli bulunamamışsa işlemlere iki iterasyon daha eklenir. Bu son iki iterasyonun ilkinde sınav dönemi kısıtından dolayı engellenen personelle ilgili kısıt kaldırılarak uygun personel aranır, ardından son iterasyonda personel atanmayan merkez kaldıysa görevlilerin ilgili merkezin bulunduğu bölgede daha önce görev alıp almadığı kontrol edilmeden sadece o merkezde daha önce görev alıp almadığı kontrol edilerek uygun görevli aranır.

#### **4.1. Genişletilmiş Kısıtlar ile Yapılan Atama Çalışmaları**

Probleme sonradan eklenen personelin en son görev aldığı sınavın dönemi ile mevcut sınavın dönemi aynı olamaz kısıtı ve bir personel önceden tanımlanmış olan altı bölgenin hepsinde görev almadan daha önce görev aldığı bölgelerde görevlendirilemez kısıtı, görevlilerin tercih ettiği merkezlere atanma oranını büyük ölçüde etkilemektedir. Görevli tercihlerinin ilgili eğitim öğretim yılının başında ve o dönemde yapılacak tüm sınavlar için alınmıyor olması nedeniyle, tercih alma aşamasında görevlinin bölgede daha önce görev alıp almadığı kontrol edilmemektedir. Bunun nedeni görevlinin yıl içinde hangi sınava, bölgeye ve merkeze atanacağını bilinmemesi, dolayısıyla bölge turunun tamamlanıp tamamlanmayacağını bilinmemesidir. Bölge kontrolünün atama sırasında yapılması, görevlinin tercih ettiği merkeze daha önce görevlendirildiği bölgede olması nedeniyle atanmamasına yol açmaktadır.

Bu nedenle personelin yaptığı tercihler üzerinde daha önce hazırlanmış olan senaryolar ile program tekrar çalıştırılmış ve yeni kısıtların atanmanın sonuçlarına olan etkisi incelenmiştir.

Senaryolardan ilki personelin üç ayrı bölgeden birer merkez olmak üzere toplam üç merkez seçtiği durumdur. Bu senaryo için hazırlanan veriler kullanılarak program çalıştırıldıktan sonra Tablo 4-8’de özetlenen sonuçlar elde edilmiştir.

**Tablo 4.8 Personelin 3 ayrı bölgeden birer il seçtiği durum**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	3996	379	326	86%

Yapılan atanmanın sonucunda hesaplanan toplam geçmiş görev puanı 3996 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 326’sı tercihine atanmıştır. Bu sayıda görevlilerin %86 oranında tercihlerine atandığını göstermektedir.

Senaryolardan ikincisi personelin üç ayrı bölgeden ikişer merkez olmak üzere toplam altı merkez seçtiği durumdur. Bu senaryo için hazırlanan veriler kullanılarak tekrar program çalıştırılmıştır. Buna göre yapılan atama sonucunun değerlendirmesi Tablo 4.9’da bulunmaktadır.

**Tablo 4.9 Personelin 3 ayrı bölgeden ikişer il seçtiği durum**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	3853	379	355	94 %

Yapılan atamanın sonucunda hesaplanan toplam geçmiş görev puanı 3853 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 355'i, yaklaşık olarak %94'ü tercihine atanmıştır.

Senaryolardan sonuncusu personelin üç ayrı bölgeden toplamda rassal olarak altı merkez seçtiği durumdur. Bu senaryo için türetilmiş veriler kullanılarak tekrar program çalıştırılmıştır. Buna göre yapılan atama sonucunun değerlendirmesi Tablo 4.10'da bulunmaktadır.

**Tablo 4.10 Personelin 3 ayrı bölgeden toplam 6 il seçtiği durum**

Toplam Geçmiş Görev Puanı	Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
26306	3815	379	375	98,9%

Yapılan atamanın sonucunda hesaplanan toplam geçmiş görev puanı 3815 olarak hesaplanmıştır. Buna göre ataması yapılmış olan 379 personelin 375'i, %98,9'u tercihine atanmıştır.

## 5. SONUÇ VE ÖNERİLER

Bu çalışmada, gerçek hayat problemi olan sınav görevli atama problemi incelenmiştir. Sınav görevli atama işlemleri bir bütün olarak ele alınmış ve tüm süreç gözden geçirilmiştir. Sürecin akışı ortaya çıkarıldıktan sonra literatürde bulunan kombinatorik optimizasyon problemleri genel olarak gözden geçirilerek problemin bir atama problemi olarak ele alınabileceği kararına varılmıştır. Atama problemleri araştırılarak, mevcut modeller incelendikten sonra problemin matematiksel modeli oluşturulmuştur.

Oluşturulan matematiksel model, sınav biriminden alınan verilerle çalıştırılmış ve kısıtları sağlayan çözüm elde edilmiştir. Önerilen matematiksel modelin, son kullanıcılar tarafından aktif olarak kullanılmasını sağlamak amacıyla, veri tabanlarıyla entegre olarak çalışan, kullanıcı dostu bir program haline getirilmesi için bir algoritma tasarlanmıştır. Tasarlanan algoritma C# dilinde kodlanmıştır. Bu program personel havuzu içerisinde bulunan görevlileri, tercihler, kısıtlar ve geçmiş görev puanlarını dikkate alarak uygun merkezlere atamaktadır. Atamanın kabul kriteri olarak merkezlere atanan görevlilerin toplam geçmiş puanını en küçükleyecek şekilde tercihlerine atanması olarak kabul edilmiştir. Program ilk haliyle oluşturulan tercih senaryolarının ilkinde 379 kişiden 356'sını tercihlerine atamıştır. Bu yaklaşık olarak % 93.4'lük bir başarı oranı demektir.

Daha sonra personelin hep aynı dönemde görev almasını engellemek ve tüm merkezlere ve bölgelere eşit şekilde dağıtılmasını sağlamak amacıyla programa ek kısıtlar getirilmiştir. Bu kısıtlar, personelin daha önce görevlendirilmediği merkezlere atanması kısıtına ek olarak daha önce görevlendirilmediği bölgelerde görevlendirilmesi kısıtı ve personelin en son görev aldığı sınavın dönemi ile atanması yapılan sınavın döneminin aynı olmaması kısıtıdır.

Algoritma yeni kısıtlar göz önünde bulundurularak yeniden düzenlenmiş ve program bu algoritmayı gerçekleyecek şekilde güncellenmiştir. Programda istenen değişiklik yapıldıktan sonra oluşturulan tercih senaryoları programın yeni hali ile tekrar çalıştırılmış ve Tablo 5.1'de bulunan sonuçlar elde edilmiştir.

**Tablo 5.1 Farklı Tercih Senaryoları ile elde edilen sonuçlar**

		Atanan Personelin Toplam Geçmiş Görev Puanları	Toplam Atanan Görevli Sayısı	Tercihine Atanan Personel Sayısı	Tercihlere Atanma Oranı
Senaryo 1	Önceki	1786	379	356	93,40%
	Sonraki	3996	379	326	86%
Senaryo 2	Önceki	1363	379	379	100%
	Sonraki	3853	379	355	94 %
Senaryo 3	Önceki	1325	379	379	100%
	Sonraki	3815	379	375	98,9%

Buna göre yapılan güncellemeyle görevlilerin tercihlerine atanma oranları önceki sayılara göre düşmüştür. Bunun nedeni, tercihlerin alınması aşamasında, sadece seçilen merkezlerin daha önce gitmediği merkezler olması kısıtının uygulanmasıdır. Bu aşamada seçilen merkezin daha önce görev alınmamış bölgede olup olmadığı kontrol edilmediğinden kişilerin tercihleri daha önce görevlendirildikleri bölgelerde olabilmektedir. Bölge kontrolü atama esnasında

gerçekleştirildiğinden, kişiler tercih ettikleri merkeze atanmayabilmektedir. Dolayısıyla personelin tercihlerine atanma oranı düşmüştür.

Bu durumun önüne geçmek amacıyla, görevli tercihlerinin alınması aşamasında, görevlinin hangi merkezi seçerse atanma oranının yükseleceğini görebilmesi için, toplam geçmiş görev puanı, bu puanın görevli havuzundaki diğer görevlilerin puanları içerisindeki durumu gibi bilgiler sunulacaktır. Böylece görevlinin daha bilinçli bir tercih yapması sağlanacaktır.

Program son haliyle yaklaşık üç dakika gibi bir sürede çalışmaktadır. Yapılan bu çalışma ile daha önce bir hafta kadar süren atama işlemleri makul sürelerle indirilmiştir. Kurulan matematiksel modele uygun olarak geliştirilen algoritma ve bu algoritmayı gerçekleyen program ile veri tabanından alınan verilerin kontrol edilip, kısıtların uygulanması, görevlilerin uygun merkezlere atanması ve atama sonuçlarının veri tabanına işlenmesi süreçleri otomatik olarak gerçekleştirilmektedir.

Geliştirilen programın çıktıları sınav merkezleri listesinin sıralamasına göre değişkenlik göstermektedir. Atama, farklı sıralama yöntemleriyle oluşturulan 1000 farklı sıralama seti oluşturularak yapılmakta ve bu atama sonuçları arasından en iyi olan seçilmektedir. Daha sonra yapılacak çalışmalarda en iyi çözüme ulaşmak amacıyla farklı sıralama teknikleri üzerinde çalışmalar yapılarak en iyi sonuca yaklaşılabılır.

Bunun yanında atamanın başarısı doğrudan görevli tercihleri ile bağlantılıdır. Geliştirilen sistemin gerçek hayatta kullanılmaya başlanmasıyla oluşacak olan tercih havuzlarının yapısı incelenerek algoritma bu yönde geliştirilebilir.



## KAYNAKLAR

- [1] Öner, A., Ülengin, F., Atama problemi için yeni bir çözüm yaklaşımı, *itüdergisi/d mühendislik*, 2(1) , 73-79, 2003.
- [2] Özkan, M.M., Atama Problemlerinde Kardinalite Sorunu, *Uludağ Üniversitesi,İ.İ.B.F.,Ekonometri Bölümü,Dr.*, 6(22) , 323-335, Haziran 2004.
- [3] Burkard, R.E., Selected topics on assignment problems, *Discrete Applied Mathematics*, 123(1) , 257-302, 2002.
- [4] Çela, E., "Assignment Problems," in *Handbook Of Applied Optimization*. Oxford, New York, USA: Oxford University Press, 2002, ch. 2, pp. 661-678.
- [5] Pentico, D.W., Assignment problems: A golden anniversary survey, *European Journal of Operational Research*, 176(2) , 774-793, 2007.
- [6] Özcan, T., A Survey of the Generalized Assignment Problem and Its Applications, *INFOR: Information Systems and Operational Research*, 45(3) , 123-141, Ağustos 2007.
- [7] Loiola, E., Abreu ,M., Netto, P., Hahn, P. ve Quer, A survey for the quadratic assignment problem., *European Journal of Operational Research*, 176(2) , 657-690, 2007.
- [8] Doğan, İ., *Yöneylem Araştırması Teknikleri ve İşletme Uygulamaları*. İstanbul: Bilim Teknik Yayınevi, 1995.
- [9] Kuhn, H. W., The Hungarian Method for the Assignment Problem, *Naval Research Logistics Quarterly*, 2(1-2) , 83-97, 1955.
- [10] Berberler, M.E., Uğurlu, O. ve Kızılateş, G., Macar Algoritmasının Sıfırları Kapatma Alt Yordamı Üzerine, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 18(2) , 85-94, 2012.

- [11] Dell'Amico, M., Martello, S., The k-cardinality assignment problem, *Discrete Applied Mathematics*, 76(1) , 103-121, 1997.
- [12] Volgenant, A., Solving the k-cardinality assignment problem by transformation, *European Journal of Operational Research*, 157(2) , 322-331, 2004.
- [13] Caron, G., Hansen, P. ve Jaumard, B., The assignment problem with seniority and job priority constraints, *Operations Research*, 47(3) , 449-453, Mayıs 1999.
- [14] Liu, G. , Wu, S. ve Zhang, Y., New Prediction Algorithm for English Class Personnel Assignment Problem, *Advances in Computer Science, Intelligent System and Environment*, 104, 95-98, 2011.
- [15] Volganent, A., A note on the assignment problem with seniority and job priority constraints, *European Journal Of Operational Research*, 154(1) , 330-335, 2004.
- [16] Mazzola, J. B., Neebe, A.W., Resource-Constrained Assignment Scheduling, *Operations Research*, 34(4) , 560-572, Ağustos 1986.
- [17] Foulds, L.R., Wilson, M., On an assignment problem with side constraints, *Computers & Industrial Engineering*, 37(4) , 847-858, 1999.
- [18] Wall, M.B., A Genetic Algorithm for Resource-Constrained Scheduling, 1991, PhD Thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, USA.
- [19] Majumdar, J., Bhunia, A.K., Penalty approaches for Assignment Problem with single side constraint via Genetic Algorithms, *Journal of Mathematical Modelling and Application*, 2(2) , 60-86, 2010.
- [20] Che, Y., Gale, I., Kim, J., Efficient assignment mechanisms for liquidity-constrained agents, *International Journal of Industrial Organization*, 31,

659-665, 2013.

- [21] Toroslu, İ., Personnel assignment problem with hierarchical ordering constraints, *Computers & Industrial Engineering*, 45(3) , 493-510, Ekim 2003.
- [22] Elizondo, R., Solving Very Large Scale School/Student Assignment Problem, 1993, A Thesis Submitted in Partial Fulfillment Of the Requirements for the Degree in Rice University.
- [23] Elloumi, A., Kamoun, H., Jarboui, B. ve Dammak, A., The classroom assignment problem: Complexity, size reduction and heuristics, *Applied Soft Computing*, 14, 677-686, 2014.
- [24] Karimzadehgan, M., Zhai, C., Integer Linear Programming For Constrained Multi-Aspect Committee Review, *Information Processing And Management*, 48(4) , 725-740, 2011.
- [25] Hosny, M., A Heuristic Algorithm for Solving the Faculty Assignment Problem, *Journal Of Communication and Computer*, 10, 287-294, 2013.
- [26] Hübscher, R., Assigning Students to Groups Using General and Context-Specific Criteria, *IEEE Transactions On Learning Technologies*, 3(3) , 178-189, Temmuz-Eylül 2010.
- [27] Koopmans, T.C., Beckman, M., Assignment Problems and the Location of Economic Activities, *Econometrica*, 25(1) , 53-76, Ocak 1957.
- [28] Bölte, A. , Thonemann, U., Optimizing simulated annealing schedules with genetic programming, *European Journal of Operational Research*, 92(2) , 402-416, 1996.
- [29] Nissen, V., Paul, H., A modification of threshold accepting and its application to the quadratic assignment problem, *ORSpektrum*, 17(2-3) , 205-210, 1995.

- [30] Ünsal, M.G., Karesel Atama Problemi için Deterministik Tavlama Benzetim Yöntemlerinin Karşılaştırılması, *Uludağ Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 18(2) , 37-46, 2013.
- [31] XIA, Y., YUAN, Y., A new linearization method for quadratic assignment problems, *Optimization Methods and Software*, 21(5) , 805-818, Ekim 2006.
- [32] Frieze, A., Yadegar, J., On The Quadratic Assignment Problem, *Discrete Applied Mathematics*, 5(1) , 89-98, 1983.
- [33] Adams, W., Johnson, T., Improved linear programming-based lower bounds for the quadratic assignment problem, 16, 43-77, 1994.
- [34] Stützle, T., Dorigo, M., ACO Algorithms for the Quadratic Assignment Problem, *New Ideas in Optimization*, 33-50, 1999.
- [35] Drezner, Z., A New Algorithm For The Quadratic Assignment Problem, *INFORMS Journal on Computing*, 15(3) , 320-330, 2003.
- [36] Jünger, M., Kaibel, V., The QAP-polytope and the star transformation, *Discrete Applied Mathematics*, 111(3) , 283-306, 2001.
- [37] Tapkan, P., Özbakır, L. ve Baykasoğlu, A., Arı Algoritması ve Genelleştirilmiş Atama Problemi: Farklı Komşuluk Yapılarının Karşılaştırılması, *Endüstri Mühendisliği Dergisi*, 21(2) , 2-13, 2008.
- [38] Osman, I. H., Heuristics for the generalised assignment problem: simulated annealing and tabu search approaches, *OR Spektrum*, 17(4) , 211-225, 1995.
- [39] Fernandez, E., Diaz, J., A Tabu Search Heuristic for the generalized assignment problem, *European Journal of Operational Research*, 132(1) , 22-28, 2001.
- [40] Liu, L., Mu, H., Song, Y., Luo, H., Li, X. ve Wu, , The equilibrium generalized assignment problem and genetic algorithm, *Applied Mathematics*

*and Computation*, 218(11) , 6526-6535, 2012.

- [41] Lee, C. , Ma, Z., The Generalized Quadratic Assignment Problem, *European Journal of Operational Research*, 206(1) , 54-63, Ocak 2010.
- [42] Hojati, M., Near-optimal solution to an employee assignment problem with seniority, *Annals of Operations Research*, 181(1) , 539-557, Aralık 2010.
- [43] Pierskalla, W., Letter to the Editor—The Multidimensional Assignment Problem, *Operations Research*, 16(2) , 422-431, 1968.
- [44] Gilbert, K., Hofstra, R., Multidimensional Assignment Problems, *Decision Sciences*, 19(2) , 306, 1988.
- [45] Pasiliao, E., Pardalos, P. ve Pitsoulis, L., Branch and bound algorithms for the multidimensional assignment problem, *Optimization Methods and Software*, 20(1) , 127-143, Eylül 2008.
- [46] Balas, E., Saltzman, M., An Algorithm for the Three-Index Assignment Problem, *Operations Research*, 39(1) , 150-161, Ocak-Şubat 1991.
- [47] Larsen, M., Branch and bound solution of the multidimensional assignment problem formulation of data association, *Optimization Methods and Software*, 27(6) , 1101-1126, Ocak 2012.
- [48] Poore, A.B., Robertson, A.J., A New Lagrangian Relaxation Based Algorithm for a Class of Multidimensional Assignment Problems, *Computational Optimization and Applications*, 8(2) , 129-150, 1997.
- [49] Nguyen, D., Thi, H. ve Dinh, T. (2012, Ekim) Solving the Multidimensional Assignment Problem by a Cross-Entropy method.
- [50] Frieze, M., Yadegar, J., An Algorithm for Solving 3-Dimensional Assignment Problems with Application to Scheduling a Teaching Practice, *The Journal of the Operational Research Society*, 32(11) , 989-995, 1981.

## Ek 1 . Personel Bilgileri

Personel	Tercih 1	Tercih 2	Tercih 3	Geçmiş Görev Puanları	Unvanlar
1	10	11	42	6	4
2	41	61	78	38	2
3	25	34	38	26	1
4	15	51	79	5	5
5	9	46	47	16	4
6	40	43	59	2	5
7	7	10	62	3	5
8	27	40	53	8	3
9	23	74	80	6	4
10	5	67	74	29	3
11	45	46	49	34	5
12	10	25	34	33	1
13	22	39	67	15	5
14	55	56	66	29	3
15	3	7	15	24	4
16	45	74	77	1	3
17	3	32	62	26	3
18	45	53	77	21	1
19	16	19	80	1	2
20	24	28	64	16	5
21	25	42	52	14	1
22	47	59	62	3	4
23	21	44	46	15	5
24	1	29	40	37	3
25	30	45	68	30	5
26	42	53	78	20	5
27	9	18	79	20	4
28	1	25	37	1	5
29	11	48	51	21	1
30	3	35	76	14	5
31	2	4	63	1	3
32	27	51	83	10	4
33	5	38	63	31	5
34	17	46	58	24	5
35	19	43	79	3	4
36	20	30	81	16	5
37	16	53	60	23	2
38	37	74	77	10	1
39	24	54	81	16	4
40	70	81	83	25	3
41	11	57	60	23	5
42	15	31	50	31	5
43	60	65	83	20	3
44	10	77	81	14	5
45	17	23	74	1	4
46	3	29	54	10	3
47	20	21	69	19	2
48	15	33	41	15	4
49	17	30	50	19	1
50	29	31	67	9	5

## Ek 2 . Merkezlere Atanması Gereken Görevli Sayıları

Merkez No	Atanması Gereken Personel Sayısı
1	9
2	3
3	3
4	4
5	3
6	23
7	12
8	3
9	4
10	5
11	3
12	3
13	6
14	3
15	3
16	11
17	3
18	3
19	3
20	4
21	8
22	3
23	4
24	4
25	4
26	8
27	5
28	3

Merkez No	Atanması Gereken Personel Sayısı
29	3
30	4
31	3
32	3
33	4
34	35
35	17
36	4
37	2
38	4
39	3
40	3
41	8
42	3
43	3
44	3
45	3
46	3
47	6
48	3
49	5
50	3
51	3
52	3
53	3
54	3
55	3
56	4

Merkez No	Atanması Gereken Personel Sayısı
57	3
58	3
59	3
60	2
61	7
62	3
63	4
64	3
65	4
66	3
67	3
68	3
69	3
70	2
71	2
72	5
73	5
74	3
75	3
76	4
77	3
78	3
79	2
80	3
81	2
82	2
83	2

### Ek 3 . Modelin GAMS Dilinde Kodlanması

sets

g PERSONNEL /1\*1671/

m CITY /1\*83/;

\*////////////////////////////////////

variables

Z MINIMUM SCORE

Y(g,m) 1 IF THE PERSONNEL g ASSIGNED TO CITY m ;

binary variable

Y(g,m);

\*////////////////////////////////////

\$onecho > taskin.txt

par=t rng=Sheet1!A1 Cdim=1 Rdim=1

par=p rng=Sheet1!CI1 Rdim=1

par=s rng=Sheet1!CM1 Rdim=1

par=Pr rng=Sheet1!CU1 Rdim=1

par=Ra rng=Sheet1!CY1 Rdim=1

\$offecho

\$call gdxrw.exe data\_tercih.xls @taskin.txt

\$gdxin data\_tercih.gdx

Parameters

p(g) SCORE OF THE PERSONNEL



s(m) THE NUMBER OF THE PERSONNEL THAT MUST BE ASSIGNED TO THE CITY

Pr(g) IF THE I THE PERSON'S TITLE IS PROF

Ra(g) IF THE I THE PERSON'S TITLE IS RESEARCH ASS.

t(g,m) PREFERENCES OF THE PERSONNEL

\$load t p s Ra Pr

\$gdxin

\*////////////////////////////////////

equations

OBJ,CONST1,CONST2,CONST3,CONST4;

OBJ.. Z=E=sum(m,sum(g,P(g)\*Y(g,m)\*t(g,m)));

CONST1(g).. SUM((m), Y(g,m))=L=1;

CONST2(m).. s(m) =E= SUM((g), Y(g,m));

CONST3(m).. sum(g,y(g,m)\*Pr(g))=g=1;

CONST4(m).. sum(g,y(g,m)\*Ra(g))=g=1;

MODEL ASSIGNMENT /ALL/ ;

SOLVE ASSIGNMENT using mip minimizing Z ;

display Y.l,z.l;

execute\_unload 'data\_tercih.gdx', Y;

execute 'gdxrw.exe data\_tercih.gdx var=Y.l rng=Sheet2!a1' ;

#### Ek 4 . GAMS ile Elde Edilen Çözüm

Personel	Tercih 1	Tercih 2	Tercih 3	Geçmiş Görev Puanları	Unvanlar	Atandığı Merkez	Tercihine Atanma
28	1	25	37	1	5	1	DOĞRU
550	1	9	77	2	4	1	DOĞRU
582	1	3	83	1	1	1	DOĞRU
717	1	71	78	3	5	1	DOĞRU
765	1	9	62	1	4	1	DOĞRU
970	1	53	77	5	3	1	DOĞRU
1444	1	8	24	2	5	1	DOĞRU
1503	1	8	43	1	4	1	DOĞRU
1616	1	48	69	1	5	1	DOĞRU
31	2	4	63	1	3	2	DOĞRU
349	2	14	39	1	4	2	DOĞRU
1583	2	32	72	6	1	2	DOĞRU
1255	3	31	83	1	1	3	DOĞRU
1565	3	37	83	1	3	3	DOĞRU
1615	3	43	60	6	4	3	DOĞRU
302	4	21	62	1	4	4	DOĞRU
332	1	4	28	11	1	4	DOĞRU
640	4	23	53	1	5	4	DOĞRU
1159	4	37	48	1	5	4	DOĞRU
241	5	54	65	5	1	5	DOĞRU
361	5	10	63	1	4	5	DOĞRU
1548	5	57	59	1	3	5	DOĞRU
140	2	6	43	1	4	6	DOĞRU
279	6	20	36	6	5	6	DOĞRU
365	6	15	54	11	4	6	DOĞRU
534	6	30	52	13	4	6	DOĞRU
597	6	55	78	15	5	6	DOĞRU
605	6	43	44	13	4	6	DOĞRU
690	6	68	83	15	5	6	DOĞRU
694	6	54	71	1	4	6	DOĞRU
753	6	12	28	12	5	6	DOĞRU
759	6	22	56	4	3	6	DOĞRU
893	6	50	51	1	3	6	DOĞRU
943	6	15	75	8	5	6	DOĞRU
1112	6	14	70	15	5	6	DOĞRU
1142	6	10	79	4	3	6	DOĞRU
1248	3	6	51	11	1	6	DOĞRU
1328	6	15	52	15	3	6	DOĞRU
1342	6	53	54	4	1	6	DOĞRU
1403	6	25	45	1	5	6	DOĞRU
1430	6	15	83	9	3	6	DOĞRU
1448	6	38	51	4	4	6	DOĞRU
1518	6	10	23	1	2	6	DOĞRU
1524	6	48	70	1	3	6	DOĞRU
1543	2	6	54	16	3	6	DOĞRU
7	7	10	62	3	5	7	DOĞRU
282	7	45	64	2	4	7	DOĞRU
310	7	23	59	1	4	7	DOĞRU
718	7	14	18	6	3	7	DOĞRU

## Ek 5 . Birinci Senaryo için Kullanılan Veriler ve Atama Sonucu

Personel	Tercih 1	Tercih 2	Tercih 3	Geçmiş Görev Puanları	Unvanlar	Atandığı Merkez
349	67	1	44	1	4	1
641	48	1	46	6	4	1
793	48	4	1	1	2	1
1077	10	58	1	6	4	1
1296	1	53	24	4	3	1
1365	64	67	1	1	3	1
1548	58	20	1	1	3	1
1553	76	1	3	11	1	1
1633	15	67	1	1	3	1
38	2	20	46	10	1	2
282	2	33	76	2	4	2
560	70	83	2	1	5	2
153	3	36	80	12	1	3
1011	17	3	13	1	4	3
1466	3	79	64	1	2	3
267	53	4	75	7	1	4
908	4	15	81	1	5	4
959	54	4	39	3	5	4
1184	7	50	4	1	4	4
802	38	5	70	2	3	5
1448	5	64	37	4	4	5
1649	5	21	50	7	1	5
8	44	6	24	8	3	6
206	10	6	20	1	3	6
265	6	71	23	11	5	6
310	6	9	51	1	4	6
339	81	6	67	6	5	6
421	45	6	25	1	2	6
427	46	52	6	1	4	6
536	4	6	78	3	5	6
660	81	55	6	1	3	6
769	3	6	19	12	3	6
823	70	6	43	1	2	6
955	4	11	6	4	5	6
980	77	6	49	1	3	6
1151	19	11	6	6	2	6
1277	6	40	59	8	1	6
1295	77	6	14	1	4	6
1371	6	56	32	12	5	6
1395	81	6	67	12	5	6
1430	80	76	6	9	3	6
1431	6	39	22	1	3	6
1483	81	74	6	5	4	6
1567	19	11	6	1	5	6
1654	39	6	81	10	1	6
45	7	40	43	1	4	7
431	41	23	7	1	5	7
446	55	58	7	4	3	7
550	13	15	7	2	4	7
555	7	41	4	1	3	7

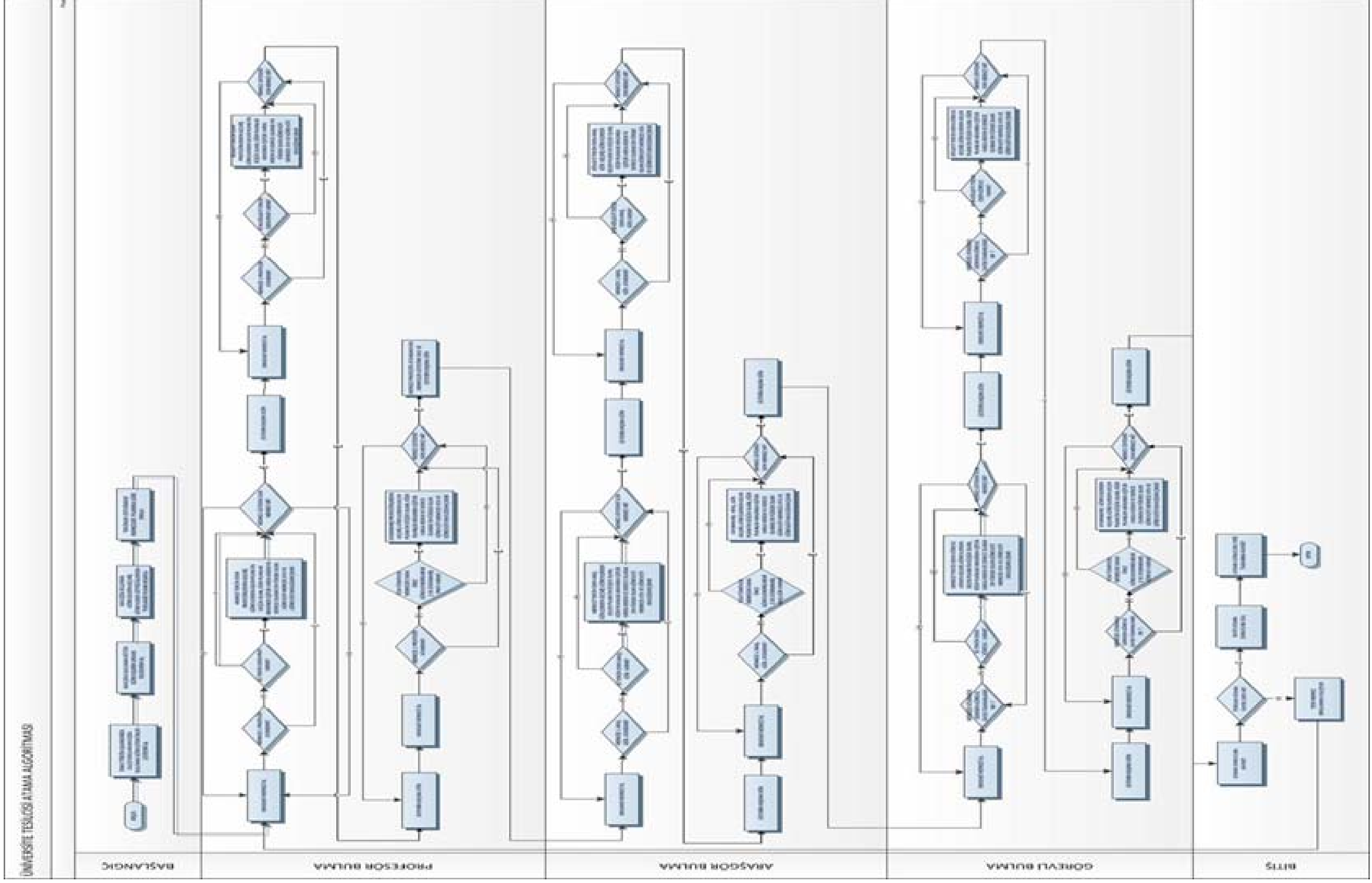
## Ek 6 . İkinci Senaryo için Kullanılan Veriler ve Atama Sonucu

Personel	Tercih 1	Tercih 2	Tercih 3	Tercih 4	Tercih 5	Tercih 6	Geçmiş Görev Puanları	Unvanlar	Atandığı Merkez
6	10	15	46	53	54	59	2	5	10
7	6	18	38	42	44	69	3	5	6
16	2	36	51	65	80	81	1	3	65
19	2	8	21	39	46	61	1	2	21
22	11	43	51	54	69	78	3	4	51
28	2	9	16	49	63	73	1	5	73
31	32	34	41	52	61	79	1	3	34
35	11	20	28	54	66	77	3	4	77
38	6	28	34	60	72	80	10	1	6
45	5	29	36	42	58	61	1	4	58
55	6	22	23	59	67	72	1	3	6
65	42	44	50	51	74	80	6	1	44
69	30	35	44	62	64	80	4	3	35
71	17	25	28	73	77	78	2	5	73
74	25	27	29	46	54	55	1	5	27
84	5	29	36	42	58	61	4	4	36
88	0	11	13	44	53	78	1	3	13
92	38	44	72	75	77	80	1	4	72
96	12	34	42	43	63	70	6	2	34
101	4	13	51	61	72	79	11	1	4
108	30	35	44	62	64	80	1	2	35
118	24	27	38	53	74	80	1	1	38
120	4	15	44	43	45	82	1	3	4
125	2	12	21	30	31	63	1	4	21
130	11	13	14	30	40	70	3	4	70
135	20	28	35	40	63	78	1	5	35
139	2	4	11	34	60	61	1	4	34
140	16	34	38	43	47	74	1	4	34
153	0	13	44	53	61	78	12	1	53
155	3	8	28	53	60	63	12	1	63
160	7	12	15	44	46	74	1	4	44
171	1	31	36	56	76	79	12	1	76
202	21	27	33	35	78	82	4	4	35
203	6	14	39	40	55	61	3	3	6
206	1	5	6	32	43	54	1	3	6
207	7	8	9	30	34	75	10	3	34
209	1	29	39	42	52	55	1	5	1
214	17	25	35	52	55	83	6	5	35
218	7	22	62	65	66	77	12	1	66
222	15	19	29	54	61	76	1	4	76
223	0	10	11	13	44	53	11	1	13
232	17	36	38	43	44	55	2	3	17
237	4	16	45	48	53	67	2	3	16
240	30	39	42	47	66	79	1	2	47
241	31	36	55	56	76	79	5	1	31
242	1	6	13	40	42	76	1	1	1
257	2	13	29	61	67	80	1	3	61
267	6	11	24	36	71	74	7	1	74
280	0	10	11	13	44	53	1	3	53
282	3	4	8	28	53	63	2	4	8

## Ek 7 . Üçüncü Senaryo için Kullanılan Veriler ve Atama Sonucu

Personel	Tercih 1	Tercih 2	Tercih 3	Tercih 4	Tercih 5	Tercih 6	Geçmiş Görev Puanları	Unvanlar	Atandığı Merkez
1	32	45	50	54	56	67	6	4	56
4	33	81	31	82	26	37	5	5	26
6	8	9	52	29	41	45	2	5	8
7	39	35	31	65	63	48	3	5	35
16	49	75	30	74	17	10	1	3	30
19	25	46	23	81	83	82	1	2	25
22	2	5	38	29	11	37	3	4	37
28	46	78	15	77	72	80	1	5	72
31	39	34	24	61	21	48	1	3	34
35	7	39	15	68	66	19	3	4	7
38	5	79	75	55	77	53	10	1	55
45	9	7	64	69	60	19	1	4	60
55	29	18	31	19	76	60	1	3	19
65	69	32	43	14	51	67	6	1	14
69	8	7	33	49	41	13	4	3	13
71	46	81	83	82	79	78	2	5	46
74	9	7	35	19	36	72	1	5	35
83	39	34	71	64	70	40	6	3	34
84	46	28	43	21	15	24	4	4	28
88	8	7	31	62	30	13	1	3	7
92	69	24	43	67	70	40	1	4	70
101	8	32	51	65	80	28	11	1	65
108	8	7	73	65	13	75	1	2	7
118	49	50	25	51	32	52	1	1	52
120	25	42	63	58	83	27	1	3	27
125	75	74	76	12	63	26	1	4	12
130	5	29	83	39	82	81	3	4	29
135	9	29	31	65	28	41	1	5	65
139	34	10	36	61	16	27	1	4	34
140	69	32	51	18	36	24	1	4	18
153	8	22	30	32	14	55	12	1	30
155	32	82	20	78	72	80	12	1	20
160	34	61	66	48	26	57	1	4	61
171	9	59	76	65	63	44	12	1	63
202	47	42	76	78	63	70	4	4	42
203	25	21	73	56	30	57	3	3	21
206	3	6	11	48	20	54	1	3	6
209	69	46	36	81	83	67	1	5	36
218	35	28	44	61	27	24	12	1	27
222	39	32	37	25	54	27	1	4	27
223	45	56	73	80	30	72	11	1	80
232	9	28	66	58	44	27	2	3	27
237	29	78	43	80	23	76	2	3	23
240	1	4	11	54	66	37	1	2	1
241	39	40	78	45	41	42	5	1	42
242	9	22	16	13	21	33	1	1	21
246	8	9	19	6	35	7	6	5	35
257	35	13	50	33	16	56	1	3	35
267	45	13	23	12	20	17	7	1	45
280	32	75	51	81	66	82	1	3	75

Ek 8 . Önerilen Algoritma



## Ek 9 . Problemin Çözümü İçin C#'da Hazırlanan Programın Kodları

```
using System;

using System.Collections;

using System.Collections.Generic;

using System.Globalization;

using System.IO;

using System.Linq;

using System.Reflection;

using System.Text;

using log4net;

namespace SınavGorevliAtama.atama_siniflari
{
    internal class PersonelSaglayici
    {
        public static ILog Log =
        LogManager.GetLogger(MethodBase.GetCurrentMethod().DeclaringType);

        public static Hashtable SınavMerkezleri;

        public static Hashtable PersonelHavuzu;

        public static List<Personel>
        AyniBolgedekiDigerMerkezleriSectigiIcinAtananlar = new List<Personel>();

        public static Dictionary<Personel, string>
        DonemdenDolayiKisitlanarlardanAtananlar = new Dictionary<Personel,
        string>();

        public static List<Personel> TumTurkiyedenAtananlar = new
        List<Personel>();

        public static List<Personel>
        GitmedigiBolgeyeGitmeKisitindanAyrılanlardanAtananlar = new
        List<Personel>();

        public static int ToplamGorevSayisi = 3;

        public List<Tercih> TercihlerListesi;

        private PersonelSorter _personelSorter = new PersonelSorter();

        public PersonelSaglayici(string sınavKodu)
```

```

    { PersonelHavuzu = PersonelHavuzuOlustur();
      SınavMerkezleri = SınavMerkezleriniOlustur(sınavKodu);    }
public Hashtable PersonelHavuzuOlustur()
{ var personelHavuzu = new Hashtable();
  # region Havuzu olusturma
  List<Personel> personelListesi = Listeleyici.AkademikPersonel;
  foreach (Personel personel in personelListesi)
  {      personelHavuzu.Add(personel.TCNo, personel);    }
  #endregion
  return personelHavuzu;    }
public Hashtable SınavMerkezleriniOlustur(string sınavKodu)
{ var sınavMerkezleri = new Hashtable();
  Hashtable sınavMerkeziListesi = Listeleyici.AtamaMerkezler;
  TercihlerListesi = new List<Tercih>();
  foreach (Tercih item in Listeleyici.Tercihler)
  { if (Listeleyici.SınavKodu.ToString(CultureInfo.InvariantCulture) ==
item.SınavKodu)
    { TercihlerListesi.Add(item); } }
  foreach (string kodd in sınavMerkeziListesi.Keys)
  { var sınavMerkezi = new SınavMerkeziHazirlik(kodd);
    sınavMerkezleri.Add(kodd, sınavMerkezi); }
  foreach (Tercih tercih in TercihlerListesi)
  { if (!PersonelHavuzu.ContainsKey(tercih.PersonelTC))
    {continue; }
    var personel = (Personel)PersonelHavuzu[tercih.PersonelTC];
(SınavMerkeziHazirlik)sınavMerkezleri[tercih.MerkezKod].TercihEdenPersonel.
Add(personel); }
  foreach (SınavMerkeziHazirlik sınavMerkezi in sınavMerkezleri.Values)
  {_personelSorter = new PersonelSorter();

```



```

sınavMerkezi.TercihEdenPersonel.Sort(_personelSorter);

        sınavMerkezi.TercihEdenPersSayisi =
sınavMerkezi.TercihEdenPersonel.Count;

        if (Listeleyici.AtamaMerkezler.ContainsKey(sınavMerkezi.Kod))
        {
            SınavMerkeziAtama sınavMerkez =
(SınavMerkeziAtama)Listeleyici.AtamaMerkezler[sınavMerkezi.Kod];

            sınavMerkez.TercihEdenPersSayisi =
sınavMerkezi.TercihEdenPersonel.Count;

            return sınavMerkezleri; }

public Personel SiradakiProfesoruVer(string merkezKod, AtamaTuru tur,
bool yedek)
{
    Personel adayPersonel = null;

    const short sınavGorevSayisi = 0;

    var sınavMerkezi = (SınavMerkeziHazirlik)SınavMerkezleri[merkezKod];

    var sınavMerkeziAtanan =
(SınavMerkeziAtama)Atayici.SınavMerkezleri[merkezKod];

    List<Personel> sortedList =
sınavMerkezi.TercihEdenPersonel.ToList().OrderBy(h =>
h.GecmisGorevPuani).ThenBy(j => j.TCNo).ToList();

    if (tur == AtamaTuru.BirinciTur)
    {
        foreach (Personel p in sortedList)
        {
            if (yedek == false)
            {
                if (p.GorevVerilmemeNedeni == GorevVerilmemeNedeni.Yok &&
p.Unvan == Unvan.Profesor &&p.Gorevlendirildi == false)
                {
                    p.Gorevlendirildi = true;

                    PersoneliTercihListelerindenCikar(p);

                    adayPersonel = p;

                    break; } }

            else{ if (p.GorevVerilmemeNedeni == GorevVerilmemeNedeni.Yok
&& p.Unvan == Unvan.Profesor && p.Gorevlendirildi == false)
            {
                p.Gorevlendirildi = true;

                PersoneliTercihListelerindenCikar(p);
            }
        }
    }
}

```

```

        adayPersonel = p; break; } } }

        if (tur == AtamaTuru.IkinciTur &&
sınavMerkeziAtanan.AtananUniversiteTemsilcileri.Count(a => a.Unvan ==
Unvan.Profesor) == 0/*adayPersonel == null*/)

        { adayPersonel = AyniBolgedekiDigerSınavMerkezlerdekiAdayPersoneliGetir
(merkezKod, Unvan.Profesor, yedek, sınavGorevSayisi);

if (adayPersonel != null)

        { adayPersonel.Gorevlendirildi = true;

                PersoneliTercihListelerindenCikar(adayPersonel);

        } }

        if (tur == AtamaTuru.UcuncuTur &&
sınavMerkeziAtanan.AtananUniversiteTemsilcileri.Count(a => a.Unvan ==
Unvan.Profesor) == 0/*adayPersonel == null*/)

        { adayPersonel = TumTurkiyedeSınavMerkezlerdekiAdayPersoneliGetir
(merkezKod, Unvan.Profesor, yedek, sınavGorevSayisi);

                if (adayPersonel != null)

                        { adayPersonel.Gorevlendirildi = true;

                                PersoneliTercihListelerindenCikar(adayPersonel); } }

        return adayPersonel;    }

public Personel SiradakiArasOgrGorVer(string merkezKod, AtamaTuru tur,
bool yedek)

{   Personel adayPersonel = null;

        var sınavMerkezi = (SınavMerkeziHazirlik)SınavMerkezleri[merkezKod];

        var sınavMerkeziAtanan =
(SınavMerkeziAtama)Atayici.SınavMerkezleri[merkezKod];

List<Personel> sortedList = sınavMerkezi.TercihEdenPersonel.ToList().OrderBy
(h => h.GecmisGorevPuani).ThenBy(j => j.TCNo).ToList();

        if (tur == AtamaTuru.BirinciTur)

                { foreach (Personel p in sortedList)

                        { if (yedek == false) {

                                if (p.GorevVerilmemeNedeni == GorevVerilmemeNedeni.Yok
&& p.Unvan == Unvan.ArasGor && p.Gorevlendirildi == false

```

```

        { p.Gorevlendirildi = true;
          PersoneliTercihListelerindenCikar(p);
          adayPersonel = p;
          break;          }          }

      else { if (p.GorevVerilmemeNedeni == GorevVerilmemeNedeni.Yok
&& p.Unvan == Unvan.ArasGor && p.Gorevlendirildi == false
        { p.Gorevlendirildi = true;
          PersoneliTercihListelerindenCikar(p);
          adayPersonel = p;
          break;          }          }          }

      if (tur == AtamaTuru.IkinciTur &&
sınavMerkeziAtanan.AtananUniversiteTemsilcileri.Count(a => a.Unvan ==
Unvan.ArasGor) == 0)

        { adayPersonel =
AyniBolgedekiDigerSınavMerkezlerdekiAdayPersoneliGetir(merkezKod,
Unvan.ArasGor, yedek, 0);

          if (adayPersonel != null) { adayPersonel.Gorevlendirildi = true;

            PersoneliTercihListelerindenCikar(adayPersonel);          }          }

          if (tur == AtamaTuru.UcuncuTur &&
sınavMerkeziAtanan.AtananUniversiteTemsilcileri.Count(a => a.Unvan ==
Unvan.ArasGor) == 0)

            { adayPersonel = TumTurkiyedeSınavMerkezlerdekiAdayPersoneliGetir
(merkezKod, Unvan.ArasGor, yedek, 0);

              if (adayPersonel != null) { adayPersonel.Gorevlendirildi = true;

                PersoneliTercihListelerindenCikar(adayPersonel); } } return adayPersonel;          }

      public Personel SiradakiGorevliyiVer(string merkezKod, AtamaTuru tur,
bool yedek)

        { Personel adayPersonel = null;

          var sınavMerkeziAtanan = (SınavMerkeziAtama)
Atayici.SınavMerkezleri[merkezKod];

          var sınavMerkezi = (SınavMerkeziHazırlık)SınavMerkezleri[merkezKod];

```

```

List<Personel> sortedList = SınavMerkezi.TercihEdenPersonel.ToList().OrderBy
(h => h.GecmisGorevPuani).ThenBy(j => j.TCNo).ToList();

if (tur == AtamaTuru.BirinciTur)

{if (yedek == false) {foreach (Personel p in sortedList) {if (!p.Gorevlendirildi
p.GorevVerilmemeNedeni == GorevVerilmemeNedeni.Yok ) {adayPersonel = p;
p.Gorevlendirildi = true;

PersoneliTercihListelerindenCikar(p);

break;          }          }          }

else

{foreach (Personel p in sınavMerkezi.TercihEdenPersonel.OrderBy(h =>
h.GecmisGorevPuani).ThenBy(j => j.TCNo))

{ if (!p.Gorevlendirildi && p.GorevVerilmemeNedeni ==
GorevVerilmemeNedeni.Yok)

        {adayPersonel = p; p.Gorevlendirildi = true;

        PersoneliTercihListelerindenCikar(p); break; }}}

return adayPersonel;    }

private List<Personel> KriterleriUygula(List<Personel> result2, string
merkezKod, Unvan unvanKod1, bool yedek, SınavMerkeziHazirlik sınavMerkezi)

{ if (unvanKod1 != Unvan.Hepsi)

    { List<Personel> result3 = (from Personel p in result2

        where p.Unvan == unvanKod1 orderby p.GecmisGorevPuani

        select p).ToList();result2 = result3; }

return result2;    }

private Personel
AyniBolgedekiDigerSınavMerkezlerdekiAdayPersoneliGetir(string merkezKod,
Unvan unvanKod1, bool yedek, Int16 gorevSayisi)

{ Personel result = null;

var sınavMerkezi = (SınavMerkeziHazirlik)SınavMerkezleri[merkezKod];

IEnumerable<string> ayniBolgedekiMerkezKodlari =
AyniBolgedekiMerkezKodlariniGetir (sınavMerkezi.BolgeNo);

List<Personel> result2 = (from Personel p in PersonelHavuzu.Values join Tercih
tercih in TercihlerListesi on p.TCNo equals tercih.PersonelTC join string

```

ayniBolgedekiMerkezKod in ayniBolgedekiMerkezKodlari on tercih.MerkezKod equals ayniBolgedekiMerkezKod

where p.Gorevlendirildi == false && p.GorevVerilmemeNedeni == GorevVerilmemeNedeni.Yok orderby p.GecmisGorevPuani, p.TCNo

select p).ToList();

result = KriterleriUygula(result2, merkezKod, unvanKod1, yedek, sinavMerkezi).FirstOrDefault();

if (result != null)

{result.Gorevlendirildi = true;

result.GorevVerilmemeNedeni = GorevVerilmemeNedeni.Yok;

PersoneliTercihListelerindenCikar(result);

AyniBolgedekiDigerMerkezleriSectigiIcinAtananlar.Add(result); }

return result; }

private Personel TumTurkiyedeSinavMerkezlerdekiAdayPersoneliGetir(string merkezKod, Unvan unvanKod1, bool yedek, Int16 gorevSayisi)

{Personel result = null;

var sinavMerkezi = (SinavMerkeziHazirlik)SinavMerkezleri[merkezKod];

List<Personel> result2 = (from Personel p in PersonelHavuzu.Values

where p.Gorevlendirildi == false && p.GorevVerilmemeNedeni == GorevVerilmemeNedeni.Yok orderby p.GecmisGorevPuani, p.TCNo

select p).ToList();

result = KriterleriUygula(result2, merkezKod, unvanKod1, yedek, sinavMerkezi).FirstOrDefault();

if (result != null) {

result.Gorevlendirildi = true;

result.GorevVerilmemeNedeni = GorevVerilmemeNedeni.Yok;

//bu personeli tum sinav merkezlerinden cikar

PersoneliTercihListelerindenCikar(result);

TumTurkiyedenAtananlar.Add(result); } return result; }

private bool TercihlerinIcindemi(string tcNo, string merkezNo)

{ List<Tercih> tercihler = Listeleyici.Tercihler;

```

        if (tercihler.Any(a => a.MerkezKod == merkezNo && a.PersonelTC ==
tcNo))
        { return true; }
        return false;    }

public static IEnumerable<string>
AyniBolgedekiMerkezKodlariniGetir(string p)
{ Hashtable merkezler = Listeleyici.HazirlikMerkezler;
  var digerMerkezler = new List<string>();
  foreach (object item in merkezler.Values)
  {string[] words = item.ToString().Split(',');
    string bolge = words[2];
    if (bolge == p)
    { digerMerkezler.Add(words[0]); } }
  return digerMerkezler; }

private void PersoneliHavuzdanCikar(Personel p)
{ ((Personel)PersonelHavuzu[p.TCNo]).Gorevlendirildi = true; } }}

```

