

**T.C.  
DOKUZ EYLÜL ÜNİVERSİTESİ  
SOSYAL BİLİMLER ENSTİTÜSÜ  
EKONOMETRİ ANABİLİM DALI  
EKONOMETRİ PROGRAMI  
YÜKSEK LİSANS TEZİ**

**HEMŞİRE ÇİZELGELEME PROBLEMLERİNİN  
GENETİK ALGORİTMALARLA OPTİMİZASYONU VE  
BİR UYGULAMA**

**Ahsen KÜÇÜK**

**Danışman  
Prof. Dr. İpek DEVECİ KOCAKOÇ**

**İZMİR - 2016**

**TEZ ONAY SAYFASI**



## YEMİN METNİ

Yüksek Lisans Tezi olarak sunduğum “Hemşire Çizelgeleme Problemlerinin Genetik Algoritmalarla Optimizasyonu ve Bir Uygulama” adlı çalışmanın, tarafımdan, akademik kurallara ve etik değerlere uygun olarak yazıldığını ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

...../...../.....  
Ahsen KÜÇÜK

**ÖZET**  
**Yüksek Lisans Tezi**  
**Hemşire Çizelgeleme Problemlerinin Genetik Algoritmalarla**  
**Optimizasyonu ve Bir Uygulama**  
**Ahsen KÜÇÜK**

**Dokuz Eylül Üniversitesi**  
**Sosyal Bilimler Enstitüsü**  
**Ekonometri Anabilim Dalı**  
**Ekonometri Programı**

Bu tez çalışmasında hemşire çizelgeleme problemi ele alınmıştır. Çizelgeleme çalışmaları yaygın olarak işçi makine veya ders programı gibi alanlarda yapılmaktadır. Ancak sağlık sektöründeki önemi de oldukça büyüktür. Çalışma, Buca Seyfi Demirsoy Devlet Hastanesi'nden alınan gerçek verilerle yapılmıştır. Verilerin toplanması aşamasında, çizelgelerin elle oluşturulmasının birim sorumlusu olan hemşirelerin 6-8 saat kadar zamanını aldığı gözlenmiştir. Bu koşullar altında bu süreyi minimuma indirecek bir bilgisayar uygulaması şarttır. Çizelgeleme problemleri, çözümü zor olan problemler grubuna girmektedir. Bu nedenle yaklaşık en iyi çözüm değerlerine ulaşabilmek için genetik algoritmalar kullanılmıştır. Hemşirelerin en uygun çalışma saatlerini bulmak için oluşturulan modelin çözümünde MATLAB programının genetik algoritmalar aracından yararlanılmıştır. Son olarak elde edilen sonuçlarla hastanenin gerçek nöbet çizelgeleri karşılaştırılarak çalışma tamamlanmıştır.

**Anahtar Kelimeler: Optimizasyon, Genetik Algoritma, Hemşire Çizelgeleme.**

## **ABSTRACT**

**Master's Thesis**

### **Optimization of Nurse Scheduling Problem with Genetic Algorithms and an Application**

**Ahsen KÜÇÜK**

**Dokuz Eylül University**

**Graduate School of Social Sciences**

**Department of Econometrics**

**Econometrics Program**

The study focuses on scheduling problem of nurses. Scheduling studies have been carried out in the realm of machine scheduling or lesson scheduling. However it is very important in medical area. The study has been conducted with the help of the data gathered from Seyfi Demirsoy State Hospital. During the stage of data gathering, we acknowledged that it takes 6 to 8 hours to form the schedules by hand. A computer application is obligatory to minimize this time under this circumstances. Scheduling problems are among the ones whose solutions are rather difficult. Hence; Genetic Algorithms have been used in order to reach approximately the best solution values. So as to find the most available work-hours of nurses, Genetic algorithms toolbox of MATLAB is utilized. Finally, the study has been completed by comparing the authentic shift-schedules of the hospital with the attained results.

**Keywords: Optimization, Genetic Algorithms, Nurse Scheduling.**

**HEMŐİRE İZELGELEME PROBLEMLERİNİN  
GENETİK ALGORİTMALARLA OPTİMİZASYONU VE BİR UYGULAMA**

**İİNDEKİLER**

TEZ ONAY SAYFASI	ii
YEMİN METNİ	iii
ÖZET	iv
ABSTRACT	v
İİNDEKİLER	vi
KISALTMALAR	ix
TABLOLAR LİSTESİ	xi
ŐEKİLLER LİSTESİ	xii
EKLER LİSTESİ	xiv
GİRİŐ	1

**BİRİNCİ BÖLÜM  
İZELGELEME**

1.1. İZELGELEMENİN AMACI VE ÖNEMİ	6
1.2. İZELGELEME PROBLEMİNİN KARMAŐIKLIĐI	8

**İKİNCİ BÖLÜM  
HEMŐİRE İZELGELEME**

2.1. HEMŐİRE İZELGELEMENİN TANIMI	12
2.2. HEMŐİRE İZELGELEME YAKLAŐIMLARI	13
2.2.1. Tavlama Benzetimi	14
2.2.2. Tabu Arama	15

2.2.3. Evrimsel Algoritmalar	16
2.2.4 Diğer Yaklaşımlar	16
2.3. LİTERATÜR TARAMASI	17

## ÜÇÜNCÜ BÖLÜM

### GENETİK ALGORİTMALAR

3.1. GENETİK ALGORİTMALARIN KULLANILMA NEDENLERİ	33
3.2. GENETİK ALGORİTMALARIN DİĞER YÖNTEMLERDEN FARKI	34
3.3. TEMEL KAVRAMLAR	35
3.3.1. Gen	35
3.3.2. Kromozom	35
3.3.3. Populasyon	36
3.3.4. Kodlama	36
3.3.4.1. İkili (Binary) Kodlama	37
3.3.4.2. Sıralı (Permütasyon) Kodlama	37
3.3.4.3. Değer (Alpha-Numeric Encoding) Kodlaması	37
3.3.5. Seçim	38
3.3.5.1. Rulet Tekerli Seçim Yöntemi	38
3.3.5.2. Sıralı (Rank) Seçim Yöntemi	39
3.3.5.3. Turnuva Seçim Yöntemi	40
3.3.5.4. Sabit Durum (Kararlı Hal) Seçim Yöntemi	41
3.3.5.5. Seçkinlik (Elitizm) İşlemi	41
3.3.6. Genetik Algoritmaların Çözümünde Takip Edilecek İşlem Adımları	42
3.4. GENETİK OPERATÖRLER	44
3.4.1. Çaprazlama	44
3.4.1.1. İkili kodlama Düzeninde Çaprazlama Yöntemleri	45
3.4.1.2. Sıralı Kodlama Düzeninde Çaprazlama Yöntemleri	46
3.4.2. Mutasyon	48
3.4.2.1. Ters Mutasyon	49
3.4.2.2. Komşu İki Geni Değiştirme	49
3.4.2.3. Keyfi İki Gen Değiştirme	49

3.4.2.4. Keyfi Üç Gen Değişirme	50
3.3.4.5. Araya Yerleştirme	50
3.4.3. Durdurma Kriteri	50
3.5. ÇİZELGELEME PROBLEMLERİNDE GENETİK ALGORİTMA	51
3.6. GENETİK ALGORİTMALARIN ÇÖZÜMÜNDE KULLANILAN BİLGİSAYAR PROGRAMLARI	54
3.7. MATLAB PROGRAMINDA GENETİK ALGORİTMALAR	56
3.7.1. Problem Kurulumu ve Sonuçları	57
3.7.2. Seçenekler	59

## **DÖRDÜNCÜ BÖLÜM**

### **UYGULAMA ÇALIŞMASI**

4.1. HASTANEDEKİ MEVCUT ÇALIŞMA KOŞULLARI	75
4.1.1. Kısıtlar	75
4.1.2. Kısıtların Matematiksel Gösterimi	77
4.1.3. Hastanenin Mevcut Çizelgeleri	79
4.1.4. Karma Tamsayı Doğrusal Programlama (MILP)	82
4.2. UYGULAMA	84
4.2.1. Kodlama	84
4.2.2. Modellerin Çözümlemesi	87
4.2.3. Sonuç ve Yorumlar	94
SONUÇ VE ÖNERİLER	95
KAYNAKÇA	97



## KISALTMALAR

<b>BCO</b>	Bee Colony Optimization-Arı Kolonisi Algoritması
<b>BEA</b>	Bacterial Evolutionary Algorithm- Bakteriyel Evrimsel Algoritma
<b>CGA</b>	Cooperative Genetic Algorithms- Kooperatif Genetik Algoritmalar
<b>CP</b>	Constraint Programming- Kısıtlı Programlama
<b>CPA</b>	Critical Path Analysis- Kritik Yol Analizi
<b>EA</b>	Evolutionary Algorithms- Evrimsel Algoritmalar
<b>ERG</b>	Expert Rotation Generator- Uzman Rotasyon Jeneratör
<b>FGP</b>	Fuzzy Goal Programming- Bulanık Hedef Programlama
<b>FM</b>	Fuzzy Modeling- Bulanık Modelleme
<b>FMP</b>	Fuzzy Mathematical Programming- Bulanık Matematiksel Prog.
<b>GA</b>	Genetic Algorithm- Genetik Algoritmalar
<b>GH</b>	Greedy Heuristic- Açgözlü Sezgisel
<b>GOA</b>	Greedy Optimization Algorithm- Açgözlü Optimizasyon Algoritması
<b>GP</b>	Goal Programing- Hedef Programlama
<b>HA</b>	Heuristic Algorithm- Sezgisel Algoritma
<b>HEA</b>	Hybrid Evolutionary Algorithm- Hibrid Evrimsel Algoritma
<b>HSA</b>	Harmony Search Algorithm- Harmony Arama Algoritması
<b>IP</b>	Integer Programming- Tamsayı Programlama
<b>LGP</b>	Linear, Goal Programming- Doğrusal, Hedef Programlama
<b>LP</b>	Linear Programming- Doğrusal programlama
<b>MA</b>	Memetic Algorithms- Memetik Algoritmalar
<b>MAPA</b>	Multi Assignment Problem Based Algorithm- Çok Atamalı Probleme Dayalı Algoritma
<b>MP</b>	Mathematical Programming- Matematiksel Programlama
<b>NÇ</b>	Nöbet Çıkışı
<b>NP</b>	Non Polynomial- Polinomal Olmayan
<b>NetP</b>	Network Programming- Ağ Programlama
<b>NP-complete</b>	Polinomal Olmayan-tam
<b>NP-hard</b>	Polinomal Olmayan-zor
<b>OLAP</b>	Online Analytical Processing- Çok boyutlu veri modeli

<b>P</b>	Polynomial- Polinomal
<b>PC</b>	Personal Computer- Kişisel Bilgisayar
<b>S</b>	Senelik İzin
<b>SI</b>	Swarm Intelligence- Sürü Zekası
<b>TS</b>	Tabu Search- Tabu Arama



## TABLULAR LİSTESİ

<b>Tablo 1:</b> Literatür Tarama Tablosu	s.28
<b>Tablo 2:</b> Katkıların Yıllara Göre Dağılımı	s.53
<b>Tablo 3:</b> Çizelgelemede Genetik Algoritma	s.54
<b>Tablo 4:</b> Hastane Birimleri	s.74
<b>Tablo 5:</b> Kısıtların Matematiksel Gösterimi	s.77
<b>Tablo 6:</b> Koroner Yoğun Bakım Yıllık İzin	s.80
<b>Tablo 7:</b> Koroner Yoğun Bakım Gece Nöbeti Listesi	s.80
<b>Tablo 8:</b> Koroner Yoğun Bakım Matris Gösterimi	s.81
<b>Tablo 9:</b> Modellerin Karşılaştırması	s.89
<b>Tablo 10:</b> Karma Tamsayı Programlama Büyük Model Çözüm Matrisi	s.91

## ŞEKİLLER LİSTESİ

<b>Şekil 1:</b> Karmaşıklığın Sınıflandırılması	s. 9
<b>Şekil 2:</b> Araştırmacılar, Hemşire Çizelgeleyiciler ve Satıcılar Arasındaki Olası İşbirliği Yolları	s. 24
<b>Şekil 3:</b> Gen, Kromozom ve Populasyon Yapısı	s. 36
<b>Şekil 4:</b> İkili Düzendeki Kodlama Yapısı	s. 37
<b>Şekil 5:</b> Sıralı Kodlama Yapısı	s. 37
<b>Şekil 6:</b> Değer kodlama yapısı	s. 38
<b>Şekil 7:</b> Rulet Tekerleği Seçme Operatörü	s. 39
<b>Şekil 8:</b> Sıralı Seçim	s. 40
<b>Şekil 9:</b> Turnuva Seçim Yöntemi	s. 41
<b>Şekil 10:</b> Genetik Algoritmanın Akış Diyagramı	s. 43
<b>Şekil 11:</b> Tek Noktalı Çaprazlama	s. 45
<b>Şekil 12:</b> İki Noktalı Çaprazlama	s. 46
<b>Şekil 13:</b> Pozisyona Dayalı Çaprazlama	s. 46
<b>Şekil 14:</b> Sıraya Dayalı Çaprazlama	s. 47
<b>Şekil 15:</b> Kısmi Planlı Çaprazlama	s. 47
<b>Şekil 16:</b> Kısmi Planlı Çaprazlama	s. 48
<b>Şekil 17:</b> Ters Mutasyon	s. 49
<b>Şekil 18:</b> Komşu İki Geni Değiştirme	s. 49
<b>Şekil 19:</b> Keyfi İki Gen Değiştirme	s. 50
<b>Şekil 20:</b> Keyfi Üç Gen Değiştirme	s. 50
<b>Şekil 21:</b> Araya Yerleştirme	s. 50
<b>Şekil 22:</b> Problem Kurulumu ve Sonuçları	s. 58
<b>Şekil 23:</b> Populasyon	s. 60
<b>Şekil 24:</b> Uygunluk Ölçeklendirilmesi	s. 60
<b>Şekil 25:</b> Seçim	s. 61
<b>Şekil 26:</b> Yeniden Üreme	s. 61
<b>Şekil 27:</b> Mutasyon	s. 61
<b>Şekil 28:</b> Çaprazlama	s. 62
<b>Şekil 29:</b> Geçiş	s. 62

<b>Şekil 30:</b> Kısıtlama Parametreleri	s. 63
<b>Şekil 31:</b> Hibrit Fonksiyon	s. 63
<b>Şekil 32:</b> Durdurma Kriteri	s. 65
<b>Şekil 33:</b> Çizim Fonksiyonları	s. 66
<b>Şekil 34:</b> Çıktı Fonksiyonu	s. 66
<b>Şekil 35:</b> Komut Penceresinde Gösterim Ekranı	s. 66
<b>Şekil 36:</b> Kullanıcı Fonksiyonunu Değerlendirme	s. 67



## **EKLER LİSTESİ**

<b>Ek 1:</b> Koroner Yoğun Bir Aylık Çalışma Çizelgesi	ek s.1
<b>Ek 2:</b> ga_big.m	ek s. 3
<b>Ek 3:</b> ga_small.m	ek s. 4
<b>Ek 4:</b> Makelincon.m	ek s. 6
<b>Ek 5:</b> Objectvec.m	ek s.11



## GİRİŞ

Sağlık hizmetlerinin temel amacı; kişi, aile ve toplumların sağlıklarının korunması, geliştirilmesi, hasta olanların tedavi edilmesi ve tedavi edilenlerin geri kalan yaşamlarını sağlıklı olarak sürdürülebilmelerini sağlamaktır. İnsanların sağlık hizmetlerinden yeterince, yerinde, zamanında ve gereksiz masraflardan kaçınarak yararlanmaları önemlidir. Ancak bir kurumun 7 gün 24 saat açık olarak hizmet vermesi gerçekten muazzam bir çalışmadır. Personelin doğru atanabilmesi büyük bir planlama gerektirir. İşte tüm bu ihtiyaçlar bu tezde çizelgelemenin tercih edilmesine sebep olmuştur.

Bu bölümde Türkçe’de kullanımını çok da yaygın olmayan çizelgeleme (scheduling) kavramının ortaya çıkışı, süreç içerisinde ki gelişimi ve bugün ki halini alış ele alınmıştır. Çizelge ve çizelgeleme kavramlarının bilimsel tanımlarına geçmeden önce literatüre girişlerinden söz edilerek söz konusu yabancılık giderilmeye çalışılacaktır.

Kritik Yol Analizi (CPA) tarafından tanımlanan 'çizelgeleme' bilimi, 2007 yılında 50. yıldönümünü kutlamıştır. Algoritmalar 1956/57 yılında Kelly ve Walker tarafından geliştirilmeye başlanmıştır. İlk kritik yol planlaması Mart 1959 tarihinde yayımlanmıştır. Çizelgelemenin evrimi, bilgisayarların gelişimini yakından takip etmiştir. İlk sistemler anabilgisayarlarda devasa karmaşıklık içerirken 1970 ve 1980’li yıllarda kişisel bilgisayarların (PC) gelişimiyle çizelgeleme sonsuza dek değişmiştir (Hyatt ve Weaver, 2006).

Çizelge (schedule); çizgilerle bölümlere ayrılmış kâğıt anlamına gelir. Çizelge yardımıyla; kadro, kademe, basamak ve derecelerin yer aldığı bir liste elde edilebilir. Çizelgeleme de eldeki işlerin bir grup kaynağa atanmasıdır. Burada Türçe’ye hemen hemen aynı şekilde çevirilen “scheduling” ve “rostring” kavramları ile ilgili küçük bir not düşmek gerekmektedir; bu iki kavram scheduling’de tarihlerin de kullanılması sebebiyle birbirinden ayrılmaktadır. Aslında günlük hayatta bir şekilde kişisel not defteri kullanarak yaptığımız planlar da bir çizelge oluşturmaktır. Ancak elbette bir okul, hastane ya da fabrika söz konusu olduğunda değerler büyüyecek ve değişken sayısı artacaktır. Bu sebeple daha kısa sürede bu problemi çözmemizi sağlayacak yollar aranacaktır. Sezgisel algoritmalar genellikle en iyiye

yakın olan çözüm yoluna hızlı ve kolay bir şekilde ulaştıklarından bu tez çalışmasında sezgisel algoritmalarından olan Genetik Algoritma kullanılmıştır.

İlk kez 1985’de Davis tarafından çizelgeleme problemine uygulanan genetik algoritma, evrim teorisini esas alarak çalışır. En iyi çözüm veya çözümleri bulmaya çalışırken, karar değişkenlerinin genetik sayı sistemindeki kodları ile ilgilenir. En iyi çözüme ulaşabilmek için çaprazlama ve mutasyon gibi operatörlerden yararlanır. Bu operatörlerin işlemleri sonucunda oluşan yeni bireyler, ebeveynlerin bazı özelliklerini taşıyarak çözüme bir adım daha yaklaşan bireylerdir. Belli bir süre sonra populasyon içerisindeki bireylerin gitgide birbirlerine benzemesine engel olmak amacıyla yapılan bu işlemler çözüm uzayının daralmasına engel olmaktadır. Genetik algoritmanın tercih edilme sebepleri; tek çözüm değil birden fazla optimum çözüm elde etmesi, çok sayıda parametre ile çalışma imkanı olması, amaç fonksiyonunu geniş bir açıda araştırmasıdır. Genetik algoritma uygulanırken MATLAB programının genetik algoritma aracı kullanılmıştır.

Tezin amacı; hemşirelerin en uygun çalışma saatlerini bulmaktır. Bunun için önce 2015 Şubat ayına ait nöbetlerin çizelgesi çıkartılmıştır. Daha sonra zorunlu kısıtlar matematiksel şekilde ifade edilmiştir. Son olarak bu ifadeler; programa, vektörler ve matrisler şeklinde tanıtılmıştır.

Tezin ilk bölümünde çizelge ve çizelgeleme kavramlarının tanımı ve önemi üzerinde durulmaktadır. İkinci bölümde çözülmesi hedeflenen hemşire çizelgeleme probleminin detaylı literatür taraması ve çözümde benimsenen çeşitli yöntemlere ilişkin örnekler yer almaktadır. Üçüncü bölümde yöntem olarak belirlenen genetik algoritma, çizelgelemedeki rolü ve çözüm aracı olan MATLAB programı yer almıştır. Son bölümde ise gerçek verilerle yapılan uygulama çıktıları değerlendirilmiş ve hastaneden alınan nöbet çizelgeleri ile kıyaslanmıştır.



## BİRİNCİ BÖLÜM

### ÇİZELGELEME

Birinci bölümde çizelgeleme kavramına, amacına ve hizmet sektöründeki önemine yer verilmiştir. Çizelgeleme kavramını daha iyi anlamak için önce çizelgeden bahsedecektir. Çizelgeler bize çeşitli faaliyetlerin hangi zamanda yapılacağı bilgisinin yanında hangi sıra ile yapılacağı bilgisini veren listelerdir (Altındaş, 2011: 2).

Uygun çizelge, her işe ait işlemlerin belirlenen sıra dâhilinde gerçekleştirilmesini sağlamak durumundadır. (Özkazanç, 2001: 127).

Çizelgeleme ise çizelgenin oluşturulma süreci olarak tanımlanabilir. Çizelge doküman üzerindeki bilgi iken, o dokümanın oluşturulması için yapılan çalışmalar bütünü de çizelgeleme işlevi olarak adlandırılabilir (Altındaş, 2011: 2).

Kısaca yapılması gereken işlerin bir grup kaynağa tahsis edilmesidir. Bu süreçte işler belirli bir sıra ile kaynaklardan geçerek tamamlanmalıdır.

Daha kapsamlı olarak; zaman, mekân ve mevcut kaynaklar göz önünde bulundurularak düzenlenen bir problem olarak ele alınır. Çizelgeleme problemi pratikte, bir arama problemi olarak herhangi mümkün bir çizelgeyi bulma veya bir optimizasyon problemi olarak en iyi mümkün çizelgeyi arama şeklinde kabul edilir (Silva ve Burke, 2004: 1).

Çizelgeleme problemleri, kısıtlı en iyileme problemleri olarak adlandırılan geniş problem sınıfına girmektedir. Verilen kısıtlara bağlı olarak problemin belli bir amaç fonksiyonunu en iyileyen bileşiminin bulunması ile ilgilenmektedirler (Seçme, 2006: 1).

Özkazanç'a (1999) göre çizelgeleme problemleri, yerine getirilmesi gereken bir grup görev ve bu görevlerin gerçekleştirilmesinde kullanılacak uygun kaynakları içerirler. İki temel girdinin nitelikleri iyi belirlenmeli ve alacakları değerler mümkün olduğunca kesin ve doğru olarak hesaplanmalıdır. Ancak böyle doğru zaman planları ortaya çıkarılabilir. Kaynaklar belirlendiğinde çizelgeleme probleminin sınırları etkin bir şekilde çizilmiş olmaktadır (aktaran Seçme, 2006: 5).

Çizelgelemeyi, belirlenmiş bazı görevleri yerine getirmek için kaynak tahsis etmek olarak tanımlamak da mümkündür. Görevler, içinde bulunulan duruma bağlı olarak farklı şekillerde isimlendirilebilir (Parlak, 2012: 4).

Örneğin, bir okuldaki derslerin çizelgelenmesi ele alındığında iş, öğrencilerin farklı derslere atanmasıdır. Dersliğin öğrenci kapasitesi, derslik sayısı, ders verebilecek öğretmen sayısı, okuldaki toplam çalışma saati, ders araları için gereken süre v.b. pek çok kısıt söz konusudur. Oldukça basit gibi görünen bir okuldaki ders çizelgesini hazırlanma işlemi bile, oldukça karmaşık bir yapıya sahip karar verme problemine dönüşebilmektedir. (Aladağ, 2010: 5).

Çizelgeleme aynı zamanda sınırlı kaynakların belirli bir amaç veya amaçlar doğrultusunda, belirli kısıtlar altında ve belirli bir zaman aralığında işlere atanması ile ilgili karar verme sürecidir. Sürecin çıktısı ise atamalardır (Macerauskas ve Teresius, 2015: 22 ).

Başka bir tanıma göre çizelgeleme, belirli amaç fonksiyonlarını optimize etmek için kıt kaynakların görevlere tahsis edilmesi ile ilgilidir (Lee ve diğerleri, 1997:1).

Çizelgeleme verilen süreler boyunca görevlere kaynak tahsisi ile ilgilidir. Bir kuruluşun kaynakları ve görevleri farklı şekillerde olabilir (Pinedo, 2012: 1).

Kaynaklar;

- Bir atölyedeki makineler,
- Bir havaalanındaki pist,
- Bir inşaat sahasındaki ekipler,
- Bir bilgisayar ortamında birimlerin işlenmesi vb. olabilir.

Görevler;

- Bir üretim sürecindeki operasyonlar,
- Bir inşaat projesinin aşamaları,
- Bir havaalanındaki kalkış inişler ve bilgisayar programlarının uygulamaları olabilir.

Her görevin belirli bir öncelik düzeyi ile mümkün olan en erken başlama saati ve bir bitiş tarihi vardır.

Çizelgelemeyi, işlerin belirli sırada gerçekleşebilmesi için bir program yapılması ve bu programın çeşitli kısıtlar altında çeşitli ölçütleri en iyilemesi faaliyeti olarak tanımlamak da mümkündür (Conway, 2012: 1).

Çizelgeleme belirlenen görevleri yerine getirmek için kaynakların zaman içindeki kullanımını göstermektedir. Buna ek olarak hangi kaynakların ne zaman ve nasıl kullanılacağı da belirtilmektedir. Çizelgeledeki asıl amaç, daha az kaynakla ve daha az sürede istenilen kıstaslara uyacak biçimde problemin çözümünü geliştirmektir (Biroğul, 2005: 16).

Çizelgeleme problemleri çalışanların en düşük işgücü maliyetleri ile çalışma vardiyalarına atanmasının belirlenmesi, personel hizmet kalitesi ve çalışma yasalarıyla ilgili kısıtlamaların karşılanmasını konu alan problemler olarak tanımlanmıştır (Ernst ve diğerleri, 2004: 4).

Baker'a (1974) göre çizelgeleme probleminde dört ana evre bulunur. Bunlar; formülasyon, analiz, sentez ve değerlendirmedir (aktaran Sağlam, 2011: 7-8):

- İlk evrede, temel olarak bir problem tanımlanır ve karar verme sürecine yardımcı olacak kıstaslar belirlenir.
- Analiz evresi, bir problemin elemanlarının ve bunların ilişkilerinin incelendiği detaylı bir basamaktır. Karar değişkenlerinin belirlenmesini hedefler. Bununla birlikte karar değişkenlerinin ilişkileri ve uyulması gereken kısıtlar burada tanımlanır.
- Sentez, probleme alternatif sonuçlar üretme basamağıdır. Oynadığı rol, var olan uygulanabilir seçeneklerin özelliklerini belirlemektir.
- Son olarak değerlendirme evresi, yapılabilir alternatiflerin karşılaştırılması ve istenen faaliyetin seçilmesi sürecidir.

Çizelgeleme model ve metod çalışmaları, bu yeteneklerin artmasına yardımcı olmaktadır.

## 1.1. ÇİZELGELEMENİN AMACI VE ÖNEMİ

Aynı sektörde etkinlik gösteren işletmeler için, içinde oldukları rekabet ortamında bir adım öne geçebilmenin yolu; mevcut işlerin bir grup kaynağa atanması olarak tanımlanan çizelgelemedir. Çizelgeleme problemleri için pek çok yöntem geliştirilmiştir. Çizelgeleme artan rekabet ortamında işletmelere problemlerini çözmeye yardımcı olmaktadır (Parlak, 2012: 1).

Personel çizelgeleme oldukça önemlidir çünkü personel hastanenin en önemli girdi kaynaklarından biridir. Hastane amaçlarının gerçekleştirilmesi, personel yönetimine ve personelin davranışlarına gereken önemin verilmesine bağlıdır (Erigüç, 1999: 4).

Başka bir ifadeyle müşteri tatmininin her geçen gün daha da zorlaştığı hizmet sektörlerinde, işverenler sunulan hizmetin kalitesini arttırmak ve hizmetin devamlılığını sağlamak için yeni arayışlar içine girmektedirler. Dolayısıyla müşteriden önce hizmeti sunan kişilerin memnuniyetini sağlayarak rekabet gücünü artırma yollarına başvurumaktadırlar. Bu yollardan biri de çizelgelemedir (Öztürkoğlu ve Çalıřkan, 2014: 115).

Çizelgeleme birçok üretim ve hizmet sektöründe düzenli olarak kullanılan bir karar verme sürecidir ve bir ya da birden çok amaç için optimizasyon yapmayı hedefler. Amaçlar aynı zamanda birçok farklı biçim alabilir. Sonuçta kaynakların işlemlere atanması ve bunu yaparken işlemlerin bir an önce bitirilmesi gibi belirli amaçların en iyilenmesinin hedeflendiğini söylemek mümkündür (Pinedo, 2012: 1).

Çizelgeleme yaparken göz önünde bulundurulması gereken birçok önemli unsur vardır. Çizelgelemesi yapılacak işlerin ve kullanılacak kaynakların pek çok özelliğinin çizelgeleme esnasında dikkate alınması gerekir.

Çizelgeleme yaparken dikkate alınması gereken önemli unsurlar genel olarak şu şekildedir (Erdem, 2008: 4):

- Yapılması gerekli işler,
- İşlerin yapılması için gerekli kaynaklara ilişkin kapasite gereksinimleri,
- İşlerin belirlenmiş zamanları veya gerçekleştirilmesi gereken zaman aralıkları,
- İşlerin önem veya aciliyet dereceleri,

- İŖi yapmada kullanılacak kaynaklar,
- Kaynakların toplam kapasiteleri,
- İŖ yapma özellikleri.

Kurumların her türlü unsuru göz önünde bulundurması ve gerçeęe uygun bir çizelge oluşturabilmesi gerekir. Çizelgelemedeki asıl amaç, daha az kaynakla ve daha az sürede istenilen kıstaslara uyacak biçimde problemin çözümüne ulaşmaktır.

Güldalı'ya göre etkin bir çizelgeleme sayesinde belirli faaliyetlerin daha az kaynak kullanarak daha kısa zamanda yapılabilme olanağı ortaya çıkmaktadır (aktaran Gürel 2012: 4)

Çizelgeleme problemlerinin çözümündeki temel amaçlar, kısıtları gözönüne alarak, en iyi performans ve en az çakışma ile sonucumuzu optimize eden bir çizelgeleme yapabilmektir (Gürel, 2012: 20-21). Amaçlar aşağıdaki gibi toparlanıp özetlenebilir:

- Kaynakların en etkin biçimde kullanılması,
- Sürenin en etkin biçimde kullanılması,
- Kısıtların belirlenmesi ve gerektięi şekilde kullanılması.

Çizelgelemenin bu denli önemli olmasının sebeplerinden biri de aşağıdaki maddelerde belirtildięi gibi birçok problemin çözümünde kullanmasıdır.

- Sınav tarih ve zamanlarına ders atanması,
- Sınavların dersliklere atanması,
- Sınavlara gözetmen atanması,
- Ulaştırmada; personel, güzergâh ve saat planlaması,
- Projelere, kaynak ve zamanlama ataması,
- İŖ çizelgeleri.

Çizelgeleme problemlerinin geneli çok amaçlıdır. Bu problemlerde; varolan kaynakların optimizasyonu, insan kaynaklarının beklentilerinin karşılanması, çizelge uzunluğunun ve talep gecikmelerinin minimizasyonu gibi birçok amaç aynı anda göz önünde bulundurulur. Uzun yıllardır çok amaçlı çizelgeleme probleminin çözümü üzerinde durulmaktadır. Uygulamalarda benimsenen yöntem, genellikle birden çok amaç için karar mekanizması tarafından belirlenen ağırlıkları yalnızca bir değerde toplayıp, daha sonra bunlara uygun çözümün bulunmasıdır. Çünkü gerçek hayattaki çok amaçlı çizelgeleme problemlerinde, karar mekanizmasına birden fazla iyi

çözümün sunulması ve bunlar arasından seçim yapılabilme yeteneğine sahip olunması oldukça önemlidir (Özbakır, 2004: 5,6).

Başka bir görüşe göre, çizelgelemenin en az iki farklı amacı vardır. Biri, bütün görevlerin en kısa sürede bitirilmesi gerekliliğidir. Bir diğeri, görevlerin kendi göreceli aciliyetine göre tamamlanmasıdır (McNaughton, 1959: 1).

Gündelik hayatımızda çizelgeleme kavramına sık sık rastlarız. Çizelgeleme çalışmaları çoğunlukla bir üretim tesisindeki işlerin çizelgelenmesinde, okullarda ders planlarının yapılmasında, havaalanlarındaki uçakların iniş kalkışlarında ve otobüs seferlerinin belirlenmesinde karşımıza çıkar. Sistem incelendiğinde fazla mesai, geciken işler, düşük işgücü kullanımı gibi sayısal veriler çizelgeleme probleminin göstergesidir. İyi bir çizelgeleme yapılmasıyla bu gibi sorunlar çözülebilir (Parlak, 2012: 5).

Genellikle çizelgeleme problemine iki adımda yaklaşılr; sıralama ve çizelgeleme. İlk adım olarak; görevlerin hangi sıra ile gerçekleştirileceğinin belirlenmesi yani sıralama yapılır. İkinci aşamada, her görevin, başlangıç saati ve mümkün tamamlanma süresi planlanır (Baker ve Trietsch, 2009: 1-2). Bu ise kaynakların kullanımındaki verimliliği ve dolayısıyla maliyetleri etkileyecektir.

Sonuçta çizelgeleme problemleri planlama dönemi boyunca bir işyerindeki çalışanların iş çizelgelerinin belirlenmesiyle ilgili problemlerdir. Çizelgeleme problemlerinin pek çok uygulama alanı vardır. Bunların başında hastaneler ve sağlık merkezleri bulunmaktadır (Karatlı, 2010: 23).

## **1.2. ÇİZELGELEME PROBLEMİNİN KARMAŞIKLIĞI**

Çözmeyi hedeflediğimiz problemin çözüm yöntemini belirlerken problemin tipi ve zorluk derecesinin belirtilmesi oldukça önemli bir konudur. Bunu problemlerin çözüm için ne kadar zor olduğunu temel alan (sınıflandırma teorisi) karmaşıklık teorisi ile ifade edebiliriz. Aşağıda, Şekil 1'deki sıra takip edilerek problemin hangi kategoride olduğu gösterilmeye çalışılacaktır.

**Şekil 1:** Karmaşıklığın Sınıflandırılması



Gass'e göre optimizasyon, bir sistemde varolan kaynakların en verimli şekilde kullanılarak belirli amaçlara ulaşılmasını sağlayan bir teknoloji olarak tanımlanmaktadır (aktaran Türkay, 2015).

Optimizasyon problemleri karar değişkenlerinin sürekli ve süreksiz olmasına göre iki kısma ayrılır. N boyutlu uzayda; herhangi bir reel değeri alabilenler sürekli değişkenler, herhangi bir tamsayı değeri alabilenler ise tamsayıli yani kesikli değişkenlerdir. (Parlak, 2012: 13; Türkay, 2015)

Kesikli çözüm uzayına sahip problemler için en iyi çözümü arayan ya da bulan yöntem kombinatoriyel en iyileme denir (Parlak, 2012: 13).

Kombinatoriyel kelimesi nesnelere sıralanması ve seçilmesi ile ilgilenen, matematiğin bir dalı olan "combinatorics" kelimesinden türetilmiştir. Bu problemlerin uygun çözümler kümesi kesiklidir ya da kesikli bir sonuca indirgenebilirler. Amaç mümkün olan en iyi sonucu bulmaktır (Hezer, 2014).

Kombinatoriyel yani kesikli en iyileme problemleri P ya da NP problemler olarak adlandırılır. P harfi "polynomial", NP harfleri ise "non-deterministic

polynomial" ifadelerini temsil eder. Türkçe karşılıkları "polinom" ve "belirleyici olmayan polinom" dur (Wikipedi, 2015).

Eğer problemin çözümü için gereken adım sayısı problem büyüklüğünün bir kuvveti ile sınırlandırıyorsa bu tür problemler P problem sınıfında kabul edilirler. Ancak eğer problem non-deterministik bir çözüme izin veriyor ve çözümü doğrulamak için gereken adımların sayısı problem ölçüsünün bir kuvveti olarak sınırlandırılabiliriyorsa bu tür problemler NP-problem sınıfına girerler (Weisstein, 2015).

NP problem türü de kendi içinde ikiye ayrılır; NP problem kadar zor olan problemlerin bulunduğu sınıfa NP-Zor (NP-hard) denir. NP-Tam (NP-complete), hem NP olup hem NP-Zor olan problemlerin sınıfıdır (Wikipedi, 2014). Dolayısıyla bu sınıftaki problemler NP sınıfının en zor problemleridir.

Bir problemin NP-Complete sınıfında olduğu kabul edilmişse o problem için polinomsal (çokterimli) zamanda çözüm bulunmasının çok zor olduğu anlaşılabilir.

Ancak NP hard problemlerde bir problemi çözen algoritma, başka herhangi bir NP problemi çözen algoritmaya dönüştürülebilir (Weisstein, 2015).

NP problem grubunda, optimum çözüme ulaşmak yerine, en yakın çözümlere ulaşmak tercih edilir. Bu tip problemlerin kesin çözümleri çok zaman aldığından, yerel arama ve stokastik arama yöntemleri ile yaklaşık çözümler elde edilmeye çalışılır (Parlak, 2012: 14). NP problemlerinin çözümünde kullanılan metotlar şöyle sınıflandırılabilir (aktaran Parlak, 2012: 15) :

- Yerel arama metotları,
- Yapay sinir ağları,
- Tabu araştırmaları,
- Tavlama benzetimi,
- Karınca kolonileri,
- Yapay bağışıklık sistemleri,
- Genetik algoritmalar,
- Arı algoritması,
- Tepe tırmanma,
- Parçacık sürü optimizasyonu,



Pinedo ve Clifford'da göre çizelgeleme problemleri kombinatoriyel optimizasyon problemlerine örnektir. Çoğu çizelgeleme problemi polinomial zamanlı algoritmalarla çözülemediği için NP-hard problemler olarak tanımlanmaktadır.

Hemşire çizelgeleme problemleri de NP-hard problemler grubuna girmektedirler.



## İKİNCİ BÖLÜM

### HEMŞİRE ÇİZELGELEME

Küreselleşmenin etkisi ile sağlık sektörünün hizmet sektörü yapısı içindeki önemi gün geçtikçe artmaktadır. Sektördeki hastanelerin amacı maliyetlerini minimize etmek ve çalışma kalitesi ile verimliliğini arttırmaktır. Toplumun ihtiyacı olan sağlık hizmetlerinin; müşterinin istediği kalitede, istediği zamanda ve mümkün olan en düşük maliyetle sunulması için, yeterli sayıda hemşire çalışanı olması oldukça önemlidir. Hemşire sayısının az olması iş yükünü arttıracığından hizmet kalitesini düşürecektir. Fazla olması ise maliyetleri arttıracığından tercih edilmeyecektir (Atmaca, 2014: 351).

Hastane ile ilgili araştırmaların çoğunda hemşire çizelgelemesi üzerinde durulmuştur. Bunun sebebi ise; hemşirelere ödenen maaşların hastane bütçesinin büyük bir kısmını oluşturmasıdır. Ayrıca hemşirelerin verdikleri hizmet kalitesi ile çalışma koşulları ve haftalık çalışma saatleri arasında doğrusal bir bağ olduğu gözlenmiş, bu da araştırmaların bu konu üzerinde yoğunlaşmasına sebep olmuştur (Öztürkoğlu ve Çalışkan, 2014: 116)

Bu bölümde ilk olarak hemşire çizelgeleme kavramından bahsedilecek daha sonra 1965'ten günümüz 2015'e kadar nasıl değerlendirildiği açıklanacak ve son olarak hemşire çizelgeleme probleminin karmaşıklığı ele alınacaktır.

#### 2.1. HEMŞİRE ÇİZELGELEMENİN TANIMI

Personel çizelgeleme problemlerinin bir alt dalı olan hemşire çizelgeleme problemleri, aylık hemşire vardiyalarının belirli kısıtlar altında hemşirelere nasıl dağıtılacağı sorusuna cevap arar (Atmaca, 2014: 352).

Hemşire çizelgeleme problemleri için çözüm teknikleri önce doğrusal programlama ile başlamıştır. Problemlerin yapısı karmaşıklaştıkça sezgisel ve metasezgisel yöntemler uygulanmaya başlamıştır (Karaatlı, 2010: 23).

Bir hemşire çizelgeleme probleminde sık karşılaşılabilecek kısıtlar şöyle listelenmiştir (Cheang ve diğerleri, 2003: 449):

- Hemşire işyükü,
- Ardışık çalışma vardiyası,
- Hemşire beceri düzeyleri ve kategorileri,
- Hemşire tercihleri ve ihtiyaçları,
- Hemşire izin günleri,
- Vardiya tipi atamaları,
- Bayramlar, tatiller ve yıllık izinler,
- Hemşire grupları arasındaki kısıtlar.

## 2.2. HEMŞİRE ÇİZELGELEME YAKLAŞIMLARI

İkinci bölüm hemşire çizelgeleme problemlerine yönelik literatürdeki son dönem yaklaşımlarını açıklamaktadır. Bu metasezgisel yaklaşımlar, özellikle evrimsel yaklaşımlara önem vermektedir. Yöntemlerin özeti bir sonraki bölümdeki detaylı literatür taramasına bir giriş niteliğindedir.

1970'lere dönüp baktığımızda hemşire çizelgeleme problemlerini çözmek için kullanılan ilk metotlardan biri matematiksel programlamadır. Matematiksel programlama yaklaşımı bazı kısıtların ihlalinin değerlendirilmesi ile belirli bir amaç fonksiyonunu optimize etmeyi dener. Matematiksel programlamanın optimal çözümleri bulabilmesi mümkün olabilir ama sadece küçük boyutlu problemlerde. Hemşire çizelgeleme problemleri ise uygulamada çok daha karmaşık ve zordur (Le, 2011: 12).

Örneğin Tien ve Kamiyama'nın 1982'de yaptıkları açıklamaya göre hemşire çizelgeleme gezgin satıcı probleminden daha karmaşıktır (aktaran Burke, 2004: 452). Bu yüzden matematiksel programlama uygulama için uygun ve pratik olmayabilir.

1980'lere gelindiğinde hemşire çizelgeleme çözümü için hedef programlama yaklaşımları bazı araştırmacıların dikkatini çekmeye başlamıştır. Hedef programlamada her bir kısıtasa ulaşmak için istenen seviye tanımlanmaktadır. Kriterler arasındaki öncelikler araştırmadan önce belirlenir. Daha sonra hedef programlama kısıtlarının her biri kendi önceliklerini takip ederek en yakın çözümü bulmaya çalışır. En son araştırmalar, çok amaçlı çerçevede sezgisel yöntemleri araştırır (Le, 2011: 12).

1980'lerden sonra hemşire çizelgeleme problemleri için yapay zekâ teknikleri incelenmeye başlanmıştır ve bugün hala araştırmacıların büyük ilgisini çekmektedir. Bu teknikler kısıt programlama ve olay temelli düşünmeyi içerir. Kısıt programlamada, değişkenler sırayla bu değişkenler üzerinde uygulanan tüm kısıtlamaları karşılamak için sonlu alanlardaki değerlere atanır. Beddoe'ya göre olay temelli akıl yürütme önceki benzer problemlerin çözümüne ilişkin bilgilerden yararlanarak yeni problemleri çözmeyi dener. Olay temelli akıl yürütme, problem çözümlerine yardımcı olan yeni bir deneyime bakarak yeni bilgiler öğrenme yeteneğine sahiptir (aktaran Le, 2011: 13).

Son 20 yılda, hemşire çizelgeleme problemini çözmek için meta sezgisel yaklaşımlar büyük ilgi görmüştür. Metasezgisellerle mümkün sonuçları bulmak çok zor olsa bile kabul edilebilir bir çözüm oluşturmak için diğer yaklaşımlardan daha uygun olduklarına inanılmaktadır (Burke, 2004: 464). Bu metasezgisellere tavlama benzetimi, tabu arama, evrimsel algoritmalar vb örnek gösterilebilir.

### **2.2.1. Tavlama Benzetimi**

Tavlama benzetimi yaklaşımları, malzemeyi ısıtıktan sonra soğumasını sağlayarak katı malzeme kalitesini artıran fiziksel tavlama sürecini taklit etmektedir. Tavlama benzetimi algoritması yerel optimumdan kaçışı araştırmaya yardım eden ve küresel optimum için aramaya devam eden ilkeyi açıklar. Soğutma işlemi kontrol eden fonksiyon, arama sırasında bulunan düşük kaliteli çözümlerin kabul edilebilirlik olasılığını ifade etmektedir. Bu olasılık, arama başlangıcında kalitesiz çözümleri kabul eden ve dolayısıyla kolayca yerel optimumdan kaçmaya izin veren yüksek bir değere ayarlanır. Arama boyunca, olasılık giderek azalır dolayısıyla, yerel optimumdan kaçmak zorlaşır. Temel amaç mevcut yerel optimum komşuluğunda yeni bir yerel optimuma sıçramaktansa global optimuma ulaşmaya çalışmaktır (Le, 2011: 12).

Burke ve diğerleri (2012), çok amaçlı hemşire çizelgeleme sorunu için bir tavlama benzetimi yaklaşımı sunmuştur. İlk olarak, her hemşire için vardiya ile ilgili bütün sert kısıtları karşılayan yasal bir vardiya kalıbı oluşturmuşlardır. Daha sonra her hemşireyi kullanılabilir vardiya kalıplarından birine atayarak sert kısıt ihlallerinin

en aza indirildiği bir çözüm bulmak için uyarlanabilir bir sezgisel yaklaşım kullanılmışlardır. Son olarak, esnek kısıtları karşılamak için iki seçenekli tavlama benzetimi tabanlı arama yöntemini sunmuşlardır: bir tanesi ağırlıklı toplamı değerlendirme fonksiyonu, diğeri hâkimiyete dayalı fonksiyondur. Hesapladıkları sonuçlar önerilen tekniğin modern hastane ortamında uygulanabilir olduğunu göstermektedir.

### **2.2.2. Tabu Arama**

Tabu arama yaklaşımları son zamanlarda ziyaret edilmiş çözümleri tutmak için bir liste kullanır. Bu listedeki çözümler, yerel optimuma takılmaktan kaçınmak için tekrar ziyaret edilmesi tercih edilmeyen çözümlerdir (Le, 2011: 14).

Burke ve diğeri (2003), tabu aramayı, hemşire çizelgeleme problemlerini çözmeye bir hiper sezgisel sürücü kuvvet olarak sunmuşlardır. Genellikle bir hiper sezgisel her bir sezgiselin mevcut performansına dayalı problemi çözmek için sezgisel ve metasezgisellerin denenmesini tercih eder.

Burke ve diğeri (2006) da yaptıkları çalışmada ise, hemşire planlaması için etkin ve etkili bir hibrit tabu arama yaklaşımı sunulmuştur. Tüm sert kısıtları karşılayan bir ilk mümkün çözüm oluşturulur. Zorunlu kısıtları karşılayan uygulanabilir başlangıç çözümü oluşturulduktan sonra mümkün olduğunca çok esnek kısıtı yerine getiren hibrit tabu arama, çözümleri ararken asla zorunlu kısıtları ihlal etmemektedir. Hibrit tabu arama ilk olarak tam haftasonu kısıtlarını çözmeyi dener. Daha sonra başka bir hemşire çizelgesi ile bu çizelgeye ait bir parçayı değiştirerek en kötü hemşire çizelgesini geliştirir. Mümkün olan en iyi değişim seçilir ancak her hemşire çizelgesi için genel bir iyileşme olacağına dair garanti yoktur. Son olarak iki hemşire programı parçaları arasındaki olası tüm değişimleri arayan açgözlü karma algoritma ile büyük bir adım gerçekleştirir (aktaran Le, 2011: 14-15).

### **2.2.3. Evrimsel Algoritmalar**

Evrimsel algoritmalar biyolojik evrime benzetme için temel alınır. Çözüm popülasyonunun devamlılığını sağlarlar. Daha sonra popülasyondaki çözümler biraraya getirilir ve yeni çözümler oluşturmak için mutasyona uğratılır. Popülasyon mevcut çözümlerden en iyi çözümlerin ve aynı zamanda gelecek nesil için yeni oluşturulan çözümlerin seçilmesi ile geliştirilir. Hemşire çizelgeleme problemlerinin çözümünde evrimsel algoritmaların çözüm sürecinin mutasyonu gibi bazı yerel teknikler kullanılması yaygındır. Amaç yalnızca çözüm kalitesini artırmak değil aynı zamanda hemşire çizelgelemenin ana zorluklarından biri olan uygun çözümü korumaktır (Le, 2011: 15).

Aickelin ve Dowsland (2000), zamanlama ve çizelgeleme alanlarında ortaya çıkan sorunları çözmek için genetik algoritmalar kullanılmasına hayli büyük bir ilgi olduğunu düşünmektedir. Ancak klasik genetik algoritma paradigmasının amaçlar ve kısıtlar arasında ortaya çıkan çatışmayı ele almada donanımlı olmadığını belirtmişlerdir. Çalışmalarında büyük bir İngiliz hastanesinin çizelgenmesi için genetik algoritma geliştirilmesi ile ilgilenmektedirler. Gerçek verilerle 52 haftalık deneysel sonuçlar sunulmuşlardır.

### **2.2.4 Diğer Yaklaşımlar**

Burke ve arkadaşları (2001), hemşire planlaması sorunu için memetik algoritma ve tabu aramayı birleştiren melez bir algoritma önermiştir. Memetik algoritma yerel arama gibi bir dik iniş iyileştirme sezgiseli kullanır. Memetik algoritmaların tabu aramadan daha dayanıklı ancak daha yavaş olduğunu belirtmişlerdir. Aynı zamanda hibrit bir yaklaşımın tek başına tabu arama ya da memetik algoritmadan daha iyi olduğuna kanaat getirmişlerdir.

### 2.3. LİTERATÜR TARAMASI

Literatür taraması bölümünde hemşire çizelgeleme çalışmalarının başladığı 1960'lı yıllardan günümüz 2000'li yıllara kadar yapılan çalışmalar özetlenmiştir. Özellikle yakın dönemde yapılan çalışmaların sayısında artış gözlemlenmiştir.

Wolfe ve Young, 1965'de ortaya koydukları matematiksel programlama ile yapılan ilk hemşire çizelgelemesinin modelinde, hemşirelerin değişik görevlere atanma maliyetlerini en aza indirmeyi amaçlamışlardır. Yaygın olarak kullanılan tur modeli ise ilk olarak 1966'da Howell tarafından geliştirilmiştir. Modelde her bir hemşire n hafta boyunca dönüşümlü olarak aynı düzende çalışmasını tekrar etmektedir (aktaran Öztürkoğlu, 2014: 117).

Rothstein'in (1973) makelesi ardışık tatil günlerini maksimize etme fikrine sahiptir, Baker'ın (1974) temel modelinde ise iki önemli varsayım vardır: tam gün çalışanlar her hafta iki gün üst üste tatil hakkına sahiptir ve işgücü, tam zamanlı çalışanlardan oluşur.

Warner (1976), insangücü karar araştırmasını; personel seçimi, çizelgeleme ve yeniden atama şeklinde üç ana alana ayırır. Problemin çizelgeleme kısmı için altı farklı kıstas tanımlamıştır:

- Kapsam: gerekli hemşire sayısını minimum yapmaya çalışarak göreve kaç farklı atama yapılabileceğidir.
- Kalite: bir çizelgenin arzu edilebilme ölçüsü ve kimin çalışmak zorunda olduğunun hemşireler tarafından belirlenmesidir. Bu ölçüm; haftasonu tatilleri, çalışma esnekliği ve belirli bir çizelgeleme döneminin tatil günleri için hemşirenin isteklerinin nasıl olduğu ile ilgilenir.
- Kararlılık: bu ölçü hemşirelerin gelecek tatil günlerini ve çalışma günlerini ne ölçüde bildikleri ve çizelgelerin bir dizi sabit veri ile tutarlı olarak oluşturulduğunu hissetmeleridir.
- Esneklik: bir çizelgeleme sisteminin tam zamanlıdan yarı zamanlıya, döngülü çalışmadan yalnızca tek vardiyaya gibi değişiklikleri sağlama ve özel gereksinimleri karşılama yeteneğidir.
- Hakkaniyet: her bir hemşirenin çizelgeleme sistemindeki diğer hemşirelerle aynı miktarda emek sarfetmesidir.

- Maliyet: çizelgeleme kararı verilmesinde kaynak kullanımınıdır.

Tüm kıstasları ölçmek neredeyse çok zordur. Warner, bu altı kıstasla üç çizelgeleme yaklaşımını karşılaştırır:

- Geleneksel yaklaşımda çizelgeler elle oluşturulur.
- Döngüsel çizelgeleme, genellikle iyi çizelgeleri sağlar ancak personelin ihtiyaçlarına hitap etmez.
- Bilgisayar destekli geleneksel çizelgeleme, iyi çizelgeler için hızlı ve daha kapsamlı arama sağlar. Bu yaklaşım avantajları açısından göz önünde bulundurulduğunda, bu avantajlar tüm diğer kıstasların avantajlarından yüksektir.

Brownell ve Lowerre (1976) her çalışana haftada iki gün izin verilirken, yedi gün için çalışanların planlamasını gösteren önemli bir bibliyografya ortaya çıkarmıştır.

Miller, Pierskalla ve Rath (1976) tam sayılı program kullanarak yaptıkları çalışmada dört haftalık çizelgeleme boyunca hemşirelerin kişisel istekleri ile hastane idaresinin istekleri arasındaki farkı minimize etmeyi amaçlamışlardır. Belirli bir hastane birimi için altı aylık uygulamaya ilişkin sonuçları sunmuşlardır. Algoritma ve hastanede oluşturulan program arasında karşılaştırmalar yapılmıştır.

Burns (1978) 14 günlük çizelgeleme boyunca 10 günlük çalışmayı yani en azından iki hafta sonu boş olması ve hiçbir iş süresinin üst üste altı günü aşmaması için çabalamıştır.

Meglino (1979), çizelgelemede, hemşire memnuniyetini sağlamada başarılı olan çalışmalardan sonra hastane yönetiminin tatminine de sıra geldiğini fark etmiştir. Çalışmasında fazla mesai ve yarı zamanlı hemşire kullanımını sınırlamaya ve hemşire fazlalıklarını önlemeye çalışmıştır. Aynı zamanda insan davranışı ve organizasyon tasarımı alanlarında da araştırmaları bulunmaktadır. Örgütsel ve psikolojik stresin, çalışanların tutum ve performansı üzerine etkisi hakkında makaleler yazmıştır.

Sonuçta 1970'li yılların bitimine doğru hemşire çizelgelemede yapılan çalışmalar farklı bir hal almış ve hemşire isteklerinin önem derecesinin arttığı görülmüştür.



Trivedi (1981), makalesinde hemşirelik bölümü için karışık tamsayılı hedef programlama modeli ile bir gider bütçesi göstermektedir. Model maliyet kontrolü ve kaliteli hemşirelik bakımı sunmak için uygun hemşirelik saatleri sağlama gibi birkaç farklı hedef içermektedir. Mümkün olan tam zaman, yarı zaman ve fazla mesailer arasındaki dengeler hafta içlerinin yanı sıra hafta sonları için de dikkate alınmıştır. Bütçelemeye de tatiller, hastalık izinleri, resmi tatiller, kıdemlilik politikaları ve hemşirelik biriminin belirlemiş olduğu politikalar dikkate alınmıştır.

Tien ve Kamiyama (1982), personel çizelgeleme algoritmalarının yalnızca sağlık sektörü ile sınırlı olmayan bir listesini sunmuş ve ABD'de hastane çizelgeleme durumuna yoğunlaşmışlardır.

Özkarahan ve Bailey (1988), daha esnek bir model kurarak hem hastane hem de hemşire isteklerini yerine getirmeye çalışmışlardır. Yapılan çalışmada amaç programlanmış modellerle çeşitli hedef önceliklerine göre çözümler hazırlamaktır.

Sitompul ve Randhawa'nın 1989'da yaptığı çalışmaya göre amaç personel maliyetini azaltmaktır. Bu yüzden finansal maliyetlere odaklanmıştır. Ayrıca hemşire planlamasını dört aşamada tanımlarlar (aktaran Burke, 2004: 444).

- Kısıtların karşılandığı bir dizi mümkün çizelge belirlenmesi,
- Maliyet, kapsama alanı ya da diğer kıstaslar açısından en iyi çizelgelemenin seçilmesi,
- Değişikliklere uyum sağlaması için ayarlar yapılması,
- Belirli vardiya atamalarının yapılması.

Siferd ve Benton (1992), Amerika Birleşik Devletleri'ndeki hastane personel ataması ve çizelgelemeyi etkileyen faktörler için oldukça iyi bir derleme sunmuşlardır. Hastane yöneticileri arasında bir anketle problemin karmaşıklığını ortaya koymuşlardır. Çalışma ilk olarak maliyet azaltmanın gün geçtikçe önem kazandığı personel çizelgeleme tarihini anlatmaktadır. Çalışmanın ikinci kısmında ise hemşire çizelgelmede hangi kısıtların göz önünde bulundurulacağı ile kısa vadeli personel planlaması tartışılmıştır. Araştırmacılar, 31 farklı hastane ve toplam 348 klinikten veri toplamıştır. Araştırılan hastanelerin personelleri, farklı beceri kategorilerinde çalışmaktadır. Personel sıkıntısı; genellikle fazla mesaiye olanak tanıyarak ve diğer kliniklerinden gelen personeller ile çözülmüştür. Tam zamanlı çalışma, yarı zamanlı çalışmaya göre daha popüler gibi görünmektedir. Hemşirelerin

hem gündüz hem gece vardiyası yapması çok nadir olmaktadır. Bazı durumlarda gece çalışmaları özel bir grup personel tarafından gerçekleştirilmektedir. Uygulamada, çok sayıda hemşire bir dizi vardiyaya atanır. Çoğu vardiyanın başlangıç ve bitiş saatleri sabittir. Hastanelerin %50'si hafta içi gündüz vardiyası için üç başlama zamanı ile çalışmaktadır ve %30'unun 5 farklı başlangıç zamanı vardır. Çoğu hastane katı kurallarla çalışmakta gibi görünmektedir. Bu makale, hemşire atamalarının uygulamadaki zorluklarını tam olarak örnekleyen bir makaledir.

Hung (1995), 60'lardan 1994 yılına kadar hemşire planlaması ile ilgili toplanan 128 makalenin kısa bir özetini sunar. Çeşitli araştırma alanlarından toplanan bu makale genel bir kaynakça olmasına karşın literatür taraması için yararlı olabilir.

Kwak ve Lee (1997) bir sağlık merkezinde çalışan doktorlar, hemşireler ve teknik elemanların mevcut vardiyalara atanmasını sağlayacak bir model ortaya koymuşlardır. Çalışma hedef programlama olarak formüle edilmiştir. Sırasıyla; tüm personele ödenen ücretlerin minimizasyonu, belirli bir dönemde belirli bir departmanda belirli bir düzeyde doktorun bulunmasını sağlamak, doktorların uygun bir oranda tutulmasını sağlamak, hemşire-doktor oranı doğrultusunda uygun miktarda hemşire atamasını sağlayabilmek, teknik eleman-doktor oranı doğrultusunda uygun miktarda teknisyen atanmasını sağlamak olmak üzere beş amaçları vardır. Çalışmada ilk iki amaç tamamen başarılıken diğer üç amaç kısmen başarılmıştır. Model bir sağlık kuruluşu tarafından sağlanan veriler kullanılarak gösterilmiştir. Çözüm esnasında belirlenmiş hedeflere öncelik verilir. Önerilen model, diğer insan kaynakları planlama süreçleri için de kolayca uygulanabilir.

Millar ve Kiragu (1998), hemşirelerin 12 saatlik vardiyalarına döngüsel ve döngüsel olmayan programlama için bir matematiksel model sunmaktadır.

Huang (1999), çalışmasında bazı önemli ergonomi kriterlerinin ve sirkadiyen ritmi yani insan vücudunun biyolojik saatinin hemşire çizelgelemesi üzerine etkilerini tartışmıştır. Tayvan'da çizelgeleme genellikle manuel olarak başhemşire tarafından yapılmakta ve bu da onun çok zaman almaktadır. 20 hemşire için iki haftalık çalışma periyodunu manuel olarak çizelgelemek başhemşirenin 2-8 saatini almaktadır. Başhemşireye ödenen yüksek ücreti düşürmek ve hemşirelerin kendi nöbet görevlerinden memnuniyetlerini arttırmak için bilgisayar destekli bir çizelgeleme aracı tasarlanmıştır. Böylece hemşire tercihleri dikkate alınmış,

çalışanlar arasında adil bir dağılım sağlanmıştır. Vardiya atamaları 0-1 tamsayılı doğrusal hedef programlama ile modellenmiştir.

Jan ve diğerleri (2000), tarafından yapılan bu araştırmanın temel amacı, genetik algoritmalar ve evrimsel algoritmalar kullanarak hemşire çizelgeleme problemi ve çözümünü sırasında meydana gelen zorlukları incelemektir. İncelemenin sonunda deneyler boyunca edinilen bilgisayar benzetim sonuçları rapor edilmiştir.

Dowland ve Thompson (2000), İngiltere'deki büyük bir hastanede hemşire çizelgeleme problemini çözmek için sezgiselliği ve iki klasik tamsayılı programlama modelini bir araya getirmişlerdir.

Cai ve Li (2000), hemşire çizelgeleme sorunu için üç amaçlı bir genetik algoritma önermişlerdir. Bunlardan ilki personel maliyetinin minimize edilmesi, ikincisi personel fazlasınının maksimize edilmesi, sonuncusu ise personel fazlası varyasyonunun en aza indirilmesidir.

Hare (2001), çalışmasında Uzman Rotasyon Jeneratör isimli bir bilgisayar programı geliştirmiştir. Kullanıcı kısıtlamaları değiştirebilmekte ve memnun olduğu şekilde seçmek için Uzman Rotasyon Jeneratör ile etkileşimde bulunabilmektedir. Arama işleminin yapılma şekli de kontrol edilebilmektedir.

Güngör (2002), çalışmasında 7 gün 24 saat açık olan, günlük mesailerin 08.00-16.00 ve 16.00-08.00 saatleri arasında yapıldığı, tüm hemşirelerin kadrolu olduğu ve haftada 40 saat çalıştığı, bir hastanenin hemşire planlaması için tamsayılı doğrusal model önermektedir. Model iki aşamalıdır. İlk aşamada minimum hemşire sayısı belirlenmiş, ikinci aşamada ise hemşirelerin iki haftalık dönem için çalışma günleri, çalışma saatleri ve izin günlerinin belirlendiği bir çizelge oluşturulmuştur. Çalışmada hastane yönetimi tarafından yapılan çizelgeye göre daha az işgücü maliyeti gerektiren, hastane yönetimi ve hemşire isteklerini daha çok yerine getirerek hizmet kalitesini artıran optimum çalışma çizelgesi elde edilmiştir.

Miwa ve diğerleri (2002), 14 günlük bir dönem için 10 hemşirenin çalıştığı bir birimde tur planlama çalışması yapmışlardır. Çalışmalarında bakteriyel evrimleşme algoritması kullanmışlardır.

Cheang ve diğerleri (2003), hemşire atama problemine güncel bir model ve yaklaşım anketi sunmaktadır. Makalenin amacı yazarların da belirttiği gibi çözüm konusuna değinmek değil kısa bir özet bilgi vermektir. Makelede daha çok

matematiksel programlama yöntemleri, kısıtlı programlama yöntemleri ve sezgisellik üzerine kısa bir özet vurgulanmıştır.

Burke ve diğerleri (2004), için hemşire atama tüm dünyada hastane personelini etkileyen karmaşık bir çizelgeleme sorunudur. Çalışmalarında, zaman ve emekten verimli şekilde yararlanmak için insanlar arasında iş yükünü dengelemenin ve personel tercihlerini karşılamanın çok önemli olduğunu belirtmişlerdir. Oldukça kapsamlı olan bu çalışma ilk olarak hemşire çizelgeleme modellerinin açıklamasını yaparak detaylı bir literatür taraması sunar. Daha sonra hemşire atama yaklaşımları olan; matematiksel programlama, çok kistaslı hedef programlama ve yapay zekâ yöntemlerini özetler. Son olarak da sezgisel çizelgeleme içerisinde tavlama benzetimi, tabu arama ve genetik algoritmaların kullanımını açıklar.

Ernst ve diğerleri (2004), hemşire atamaya yoğunlaşmak yerine daha genel bir tanımlama ile personel çizelgelemeye odaklanmışlardır. Makaleyi üç ana bölüme bölmüşlerdir:

- Tanımlar, personel çizelgeleme problemlerinin sınıflandırılması,
- Uygulama alanlarında literatür taramasının sınıflandırması,
- Uygulanabilirlik yorumları ile çözüm yöntemleri.

Bilinçli olarak atama ve çizelgeleme arasında ayrım yapmamaktadırlar. Açıkladıkları özelliklerin çoğu hemşire atama için kullanıldığı gibi çizelgeleme için de kullanılmaktadır. Matematiksel programlama ve metasezgisel yaklaşımların, personel planlaması ile ilgili literatür taramalarında, bugüne kadar en çok incelenen teknikler olduğuna işaret etmektedirler. Bilahassa metasezgiselin, çok zor sorunlar ve kesin yaklaşımlarla optimal çözümlerin elde edilemediği gerçek dünya sorunları için umut verici olduğuna dikkat çekmişlerdir.

Azaiez ve Sharif (2005), çalışmalarında 0-1 hedef programlama yaklaşımı ile bilgisayarlı hemşire planlama modeli geliştirilmişlerdir. Program Riyadh Al-Kharj hastanesinde mevcut manuel çizelgelemenin geliştirilmesi için uyarlanmıştır. Geliştirilen model hem hastane hedeflerini hem de hemşirelerin önceliklerini hesaplamakta ve bunlara ek olarak literatürde gösterilen bazı önerileri gözden geçirmektedir. Hastane fazla mesai için ek maliyetlerden kaçınarak, uygun personel sayısı ile kesintisiz hizmet sağlamayı hedeflemektedir. Model Lingo kullanarak altı aylık bir deneysel aşamada uygulanmıştır.

Belien (2005), çalışmasında üç amaç gütmüştür. İlki; ameliyathane oda sayısına bağlı olarak hemşire talebinin belirlenmesidir. İkincisi çizelgeleme problemleri için kullanılan sütun oluşturma tekniğinin kullanılmasıdır. Üçüncüsü ise çok sayıda vasıtayla maliyet tasarrufudur.

Özcan (2005), problemin çözümü için memetik algoritma yöntemini kullanmıştır. Problemin çözümünün zor olduğunu ancak çalışanların istek ve ihtiyaçlarının da tatmin edilmesi gerektiğini belirtmiştir. Hemşire atama problemini gerçek verilerle çözmüştür.

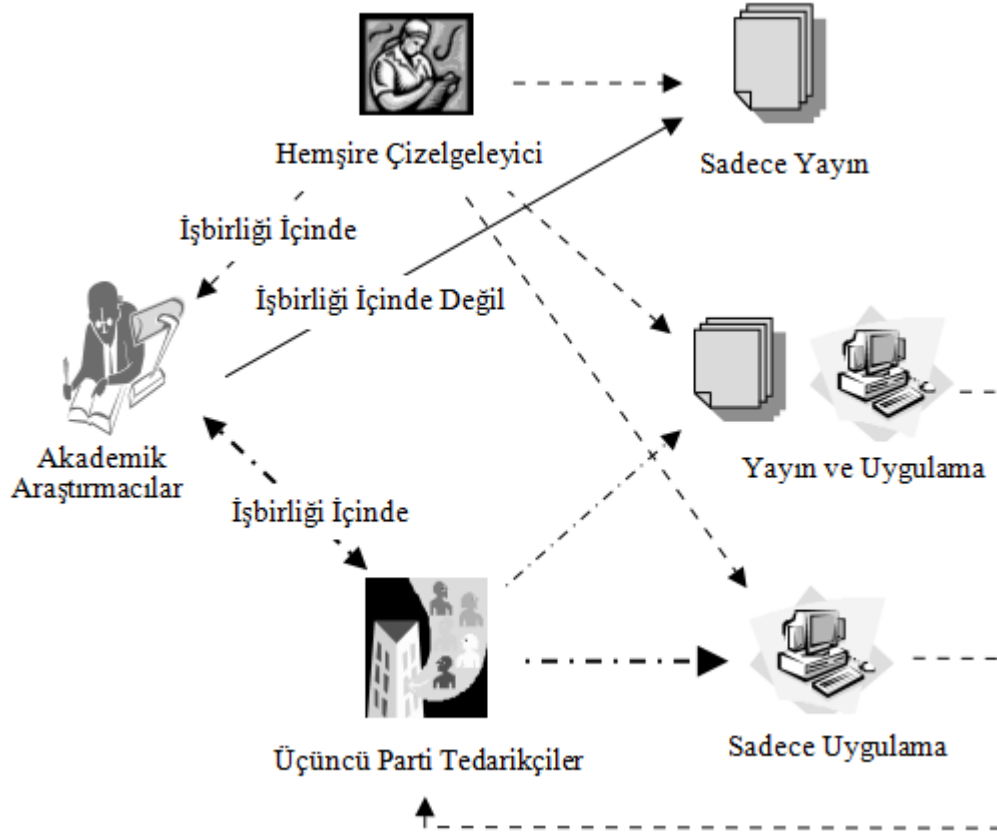
Ohki ve diğerleri (2006), tarafından yapılan çalışma, hemşire planlaması sorununu çözmek için genetik algoritma kullanmanın verimli olacağını önermektedir. Algoritmaları işbirliğine dayalı genetik algoritmayı temel almaktadır. Çaprazlama ve mutasyondan sonra yerel minimum alana yakalandığını düşündükleri çözümlerini geliştirip optimuma ulaştırmak için bir dağ tırmanışı operatörü kullanmışlardır. Sonuçlar pratik deneylerle gösterilmiştir.

Moz ve Pato (2007), daha önce atanan vardiyalarda çalışmak istemeyen hemşirelerin yeniden çizelgenemesi için bir genetik algoritma yaklaşımı kullanmışlardır. Eğer hemşire yokluğunu gidermek için bir rezerve hemşire havuzu yoksa geçerli çizelgenin yeniden oluşturulması gerekir. Bu yeni liste işgücü kuralları ve kurumsal kısıtlamalara uymak zorundadır. Yaklaşım, hemşire yeniden çizelgeleme sorunu ve sıralama problemlerine uygulamak için genetik algoritmaların çeşitli sürümleri dışında özgün kodlama ve operatörlere dayalı yapıcı sezgiseller incelemiştir. Genetik algoritmalarda açıklanan popülasyondaki her birey, görevler ve hemşirelerin permutasyonlarını sunan bir çift kromozomla ilişkilidir. Bu permutasyonlar kadroları üreten bir prosedüre girdi olarak kullanılmaktadır. Bireyin uygunluğu, yeniden oluşturulan çizelge ve mevcut çizelge arasındaki benzerlik temel alınarak tanımlanır. Oluşturdukları sezgisel Lizbon'daki bir hastaneden gerçek veriler alınarak test edilmiş ve kaliteli çözümler elde edilmiştir.

Kellogg ve Walczak (2007), çalışmalarında hem akademinin ürettiği hem de hastanelerde kullanılmış olan modelleri incelemiştir. Araştırma verisi olarak; makaleler, e-mail ve telefon anketleri, endüstri veri tabanı ve katalog yazılımı dâhil pek çok kaynaktan yararlanmışlardır. Aynı zamanda uygulama boşluğu nedenlerini incelemiş ve daha uygulanabilir akademik araştırma yollarını aramışlardır.

Çalışmanın ilk bölümünde, akademik araştırmaların pratikte nasıl kullanıldığını tartışmışlardır. Daha sonra ABD'de hemşire planlaması çözümlerinin uygulanması, hemşire planlamasında araştırma uygulama boşluğu ve gelecekteki araştırmalara yön verici bir keşif tartışılmıştır. Çalışma 1985'ten 2005 dönemine kadar yayınlanan akademik araştırmalarla sınırlandırılmıştır. Şekil 2'de birkaç işbirliği yöntemi tasarlamışlardır.

**Şekil 2:** Araştırmacılar, Hemşire Çizelgeleyiciler ve Satıcılar Arasındaki Olası İşbirliği Yolları



Kaynak: Kellogg ve Walczak, 2007: 359.

Landa-Silva ve Le (2008), çok amaçlı hemşire çizelgeleme problemi için kendi kendine uyarlanabilen basit bir evrimsel algoritma önermiştir. Karar vericinin anlamlı bir şekilde çizelgelerinin kalitesini ölçmesi için dört hedef kullanmışlardır. Birinci amaç personel memnuniyetini temsil eder ve bir hedef olarak ayarlanır. Diğer

üç amaç optimizasyona tabi olan iş düzenlemeleri ve işgücü talebini temsil eder. Algoritmaları yeni genetik materyalin üretimini teşvik etmek adına sert kısıtlamaları ve yenileme stratejisini ele almak için kendi kendini uyarlayan bir kod çözücü içerir.

Puente ve diğerleri (2009), tıbbi personel planlama sorunu için bir genetik algoritma sunmuşlardır. İspanya'daki bir hastanenin gerçek verilerinden yararlanmışlardır. Uygulanabilir çözümlerin ilk popülasyonu sert kısıtları karşılayan özel sezgisel çizelge oluşturucu ile üretilmektedir. Çözümler özel bir çaprazlama operatörü ile üretilir. Uygun olmayan çözümler daha sonra onarılır.

Brunner ve diğerleri (2009), bir Alman hastanesindeki vardiya çizelgeleme problemini araştırmışlardır. Amaç, iş sözleşmesi tarafından koyulan kısıtlamalar altında fazla mesai saatlerini düşürerek yapılan ödemelerin minimizasyonunu sağlamaktır.

Tein ve Ramli (2010), hemşire planlama modelleri için son gelişmeleri özetlemiş ve potansiyel bir yol önermişlerdir. Hemşire çizelgeleme probleminde çizelgeleme teknikleri olarak; optimizasyon teknikleri, yapıcı sezgiseller, araştırma teknikleri, bilgi temelli yaklaşımlar ve son olarak hibrit tekniklerden bahsetmişlerdir. 2000'den 2010 yılına kadar hemşire planlaması ve yeniden programlama sorununun tekniklerini içeren oldukça faydalı ve detaylı bir tablo oluşturmuşlardır. Önerdikleri potansiyel yol için ise hibrit bir evrimsel algoritma kullanmışlardır.

Bouarab ve diğerleri (2010), teorik modelden pratik uygulamaya gittikleri hemşire çizelgeleme çalışmaları için Montreal'deki Notre-Dame Hastanesi ve Sainte-Justine Hastanesi ile temasa geçmişlerdir. Bunlar iki büyük kamu ve üniversite hastaneleridir. Bu hastanelerin yapıları standarttır. Hastane 1 için düzenli bir ekip seçilmiş, hastane 2 için ise kayan noktalı bir ekip belirlenmiştir. 28 gün ve 30 işgücü için yapılmıştır. Bu makalenin amacı yöneylem araştırması ve yalın düşünce ilkelerinin verimli programlar geliştirilmesinde bir araç olarak nasıl kullanılabileceğini değerlendirmektir.

Brucker ve diğerleri (2010a), çalışmalarında hemşire atama problemlerini çözmek için adaptif yapıda bir yöntem araştırmışlardır. Problemlerde göz önünde bulundurulmuş kısıtlar iki kategoriye ayrılmıştır; sıra ile ilgili olanlar ve çizelge ile ilgili olanlar. Çözümü iki basamağın üzerine inşa etmeyi önermişlerdir: 1) Sadece sıra kısıtlarını dikkate alarak hemşireler için yüksek kaliteli sıralar oluşturmak, 2)

Hemşireler için iteratif çizelgeler oluşturulmak. Ayrıca, yeni hemşire atama değerlendirmesini gerçek dünya verilerine dayanan veri seti ile sunmuşlardır. Hemşire çizelgelemede teori ve pratik arasındaki boşluğu kapatmayı hedeflenmiştir.

Jenal ve diğerleri (2011), hedef programlama kullanarak döngüsel hemşire çizelgeleme üzerine bir çalışma yapmışlardır. Çalışmada 0-1 hedef programlama kullanmışlardır. Planlama süresi 4 hafta yani 3 farklı vardiya ile 28 gündür.

Causmaecker ve Berghe (2011), hemşire atama problemlerine bir kategorizasyon sunmuşlardır. Bunu yaparken çizelgelemeyi  $\alpha$ ,  $\beta$ ,  $\gamma$  notasyonu üzerine kurmuşlardır.

Ratnayaka ve diğerleri (2012), otomatik hemşire çizelgeleme sistemi için gelişmiş açgözlü optimizasyon algoritmasını kullanmışlardır. Gelişmiş açgözlü optimizasyon algoritmasını, hemşire kadrosunu optimize etmek için uyguladıktan sonra diğer optimizasyon algoritmaları (tavlama benzetimi ve genetik algoritma) ile karşılaştırmışlardır. Deneysel sonuçlar MYSQL, JAVA, OLAP (çok boyutlu veri modeli) ile elde edilmiştir. OLAP hemşire bilgilerini saklamak, almak ve yönetmek için bir veri eşya evi olarak, MYSQL ve Java önyüzleri ise veri ambarı olarak bilgilere erişmek için kullanılmıştır.

Soto ve diğerleri (2013), kısıt programlama ile hemşire ve ilk acil yardım atama için bir durum çalışması yapmışlardır. Bu çalışmanın amacı hemşire atama için bir kısıt programlama çözümünün tasarımını kapsayan gerçek bir vaka çalışması göstermektir. Çalışma, altı orta ölçekli Şili sağlık merkezine odaklı yapılmıştır. Her iki model de hızlı ve doğru hemşire ataması sağlayan Eclipse çözücü ile çözülmüştür. Burada ortaya konan çözüm devam eden bir çalışmadır ve büyük klinik merkezleri dikkate alınarak genişletilebilecektir.

Baskaran ve diğerleri (2013), etki dönüşüm yöntemi uygulanan hemşirelerin maliyet etkili çizelgelerine örnek bir çalışma yapmışlardır. Bu çalışma daha fazla sayıda personel ile tanınan esneklik arasındaki değiş tokuşu ve ve hemşire çizelgeleme sorunu bağlamında ekstra personel istihdamı ile örtülü maliyeti tartışır ve analiz eder. Baskaran ve diğerleri yüksek hesaplama gereksinimleri olmadan kaliteli programlar elde eden etki dönüşüm yöntemine dayalı, pratik bir yaklaşım sunmaktadır. Yumuşak kısıtların ve personellerin maliyetini dengelemeye



çalışmışlardır. Ancak önerdikleri yöntemin, rekabetçi ve şeffaf sonuçlar sağlamasına karşın küresel optimumu garanti etmediğine dikkat edilmelidir.

Öztürkoğlu ve Çalışkan (2014), hemşire çizelgeleme problemi için tam sayılı matematiksel bir model oluşturmuştur. Modelin varsayımlarına göre;

- Gece ve gündüz olmak üzere iki vardiya çalışılır,
- Yoğun bakım hemşirelerinin dinlenme imkânlarının kısıtlı olması nedeni ile çalışma süreleri gece ve gündüz vardiyaları için 10 saattir,
- Her hemşireye haftada bir gün tatil verilir,
- Hemşirelerin en fazla ardışık olarak üç gece vardiyasında çalışmasına izin verilir,
- Hemşire gece vardiyasından sonraki gün gündüz vardiyasına atanamaz.

Modelin başlıca amacı, hemşirelerin kendi tercihlerine göre haftalık çizelgelerinin oluşturulmasıdır. Sonuçta oluşturulan model, gerçek veriler kullanılarak bir üniversite hastanesinin genel cerrahi bölümünde denenmiştir ve %99,6 oranında hemşire tercihlerini yerine getirmiştir.

Tablo 1’de şimdiye hakkında bilgi verilen makalalar ve incelemeye dâhil tutulmamış ancak konu üzerine yapılmış çalışmaların özet görünümü sunulmuştur. Güncel olmasına gayret edilerek yalnızca 2000 yılından günümüze kadar olan çalışmalara yer verilmiştir. Bu tezde çözülmeye çalışılan problemin çözüm yöntemi olan GA ise bu sebeple ayrı bir sütunda incelenmiştir.

**Tablo 1:** Literatür Tarama Tablosu

Yıllar	Yazarlar	Optimizasyon	GA	Diğerleri
2000	Dowland ve Thompson			TS+IP
2000	Jan ve diğerleri		CGA	
2001	Hare			ERG
2002	Güngör	IP		
2002	Miwa ve diğerleri			BEA
2003	Cheang ve diğerleri			
2003	Inoue ve diğerleri			EA
2003	Kawanaka ve diğerleri		GA	
2004	Aickelin ve Dowland		GA	
2004	Burke ve diğerleri			
2004	Ernst ve ark.			
2005	Azaizez ve Sharif	0-1 LGP		
2005	Belien	IP		
2005	Özcan			MA
2006	Hayes ve diğerleri			
2006	Ohki ve diğerleri		CGA	
2007	Kellogg ve Walczak			
2009	Brunner ve diğerleri	IP		
2009	Çivril		GA	
2010	Bouarab ve diğerleri	MP		HA
2010	Brucker ve diğerleri (2010a)			NetP
2010	Brucker ve diğerleri(2010b)			GH
2010	Kitada ve diğerleri			HA
2010	Maenhout ve Vanhoucke			HA
2010	Tein ve Ramli			HEA
2010	Topaloğlu ve Selim	FMP+FGP		
2010	Tsai ve Lee			GA+MP
2011	Awadallah ve diğerleri			HSA
2011	Causmaecker ve Berghe			
2011	Clark ve Walker			
2011	Jenal ve diğerleri	0-1 GP		
2012	Ratnayaka ve diğerleri	GOA		
2012	Yilmaz	MP		
2013	Baskaran ve diğerleri			
2013	Maenhout ve Vanhoucke	IP		
2013	Todorovic ve Petrovic			BC+SI
2013	Soto ve diğerleri	CP		
2014	Constantino ve diğerleri	MAPA		

2014	Haspeslagh ve diğ erleri	MP		
2014	Öztürkoğ lu ve Çalıřkan	IP		
2015	Jafari ve diğ erleri	MP+FM		
2015	Legrain ve diğ erleri			
2015	Lin ve diğ erleri (2015a)		GA	
2015	Lin ve diğ erleri (2015b)	MA		
2015	Rivera ve Mesa	IP		

Tabloda Kullanılan Karakterler:

ERG: Expert Rotation Generator, TS: Tabu Search, HA: Heuristic Algorithm, EA: Evolutionary Algorithms, GA: Genetic Algorithm, CGA: Cooperative Genetic Algorithms, BEA: Bacterial Evolutionary Algorithm, BCO: Bee Colony Optimization, MA: Memetic Algorithms, HSA: Harmony Search Algorithm, GOA: Greedy Optimization Algorithm, GH: Greedy Heuristic, FM: Fuzzy Modeling, FMP: Fuzzy mathematical programming, FGP: Fuzzy goal programming, SI: Swarm Intelligence, CP: Constraint Programming, MAPA: Multi Assignment Problem Based Algorithm, HEA: Hybrid Evolutionary Algorithm, IP: Integer Programming, LP: Linear Programming, LGP: Linear, Goal Programming, GP: Goal Programing, MP: Mathematical Programming, NetP: Network Programming

Bu bölümde hemřire planlaması problemi için temel terminoloji, kavramlar ve modeller sunulmuřtur. Literatürdekilerin yanı sıra gerç ek dünya hemřire çizelgeleme senaryolarının da kısa bir açıklamasını sunulmuřtur. Bu bölümde ayrıca, literatürdeki hemřire planlaması için bazı yaklaşımlar özetlenmiřtir ancak evrimsel yaklaşımlara daha çok dikkat edilmiřtir.

Evrimsel yaklaşı m çok güçlüdür, ama ana zorluklardan biri çözümlerinin uygulanabilirliğini devam ettirmektir. Evrimsel yaklaşı m hemřire çizelgeleme problemlerine uygulanabilir çözümler üretmek ve hatta çözümlerin kalitesini artırmak için genellikle mutasyon veya onarım yöntemleri gibi yerel arama çeřitlerini kullanmaktadır. Literatürde hemřire çizelgeleme problemleri için pek çok model ve yaklaşı m olmasına rağ men, arařtırmada ve gerç ek hastane verilerinin zorlu gereksinimleri arasında hala büyük bir boşluk vardır (Burke ve diğ erleri, 2004: 469).

Bunun arkasındaki neden arařtırmadaki modellerin, genellikle gerek dnya hemřire izelgeleme problemlerinin basitleřtirilmiř hali olmasıdır. Mevcut eęilim gerek dnyanın gereksinimini ele almak ve arařtırma modelleri arasındaki bořluęu doldurmaya alıřmaktır. Her yaklařım genel olarak sadece birkaç hatta belli bir modeli özmek için tasarlanmıřtır. Hemřire izelgeleme problemlerini özmek için farklı yaklařımları melezleme eęilimi vardır (Burke ve dięerleri, 2004: 469).

Bir sonraki bölümde hemřire izelgeleme probleminin özüm yöntemi olarak belirlenen Genetik Algoritmalar incelenecek ve uygulama özümünde yararlanılan MATLAB Programı içindeki kullanımını açıklanacaktır.



## ÜÇÜNCÜ BÖLÜM

### GENETİK ALGORİTMALAR

İnsanoğlu tarihin derinliklerinden beri doğa ile iç içe yaşamaktadır. Bu birliktelik içerisinde doğa, insana birçok konuda ilham kaynağı olmuştur. Örneğin, uçağın yapılması, radarın icadı vb. gibi teknolojik gelişmeler insanoğlunun doğayı ve doğadaki canlıları incelemesi sonucu gerçekleşmiştir. Doğanın incelenmesi esnasında kimi sorunların çözümlenmesi için birçok yöntem geliştirilmiştir. Genetik algoritmalar da bu yöntemlerden biridir (Şen, 2004: 31).

Evrimsel hesaplamaların bir alt dalı olan genetik algoritmalar, yapay zekânın hızla gelişen alanlarından biridir (Onan, 2013: 119). Bu metasezigel yaklaşımların esin kaynağı Darwin'in evrim teorisidir. Evrimsel hesaplama kavramı, 1960'lı yıllarda I.Rechenberg'in "Evrimsel Stratejileri" adlı çalışmasında ortaya atılmıştır. Takip eden 1970'li yıllarda da Michigan Üniversitesi'nden John Holland tarafından genetik algoritmalar bulunmuş ve öğrencileri ile meslektaşları yardımıyla geliştirilmiştir. Bunu 1975 yılında Holland'ın "Doğal ve Yapay Sistemlerde Uyum" adlı kitabını yayınlaması izlemiştir. (Obitko, 1998a).

Genetik algoritmalar konusundaki esas gelişim ise, John Holland'ın doktora öğrencisi David E. Goldberg tarafından 1985 yılında hazırlanan "Gaz ve Boru Hatlarının Genetik Algoritma Kullanılarak Denetlenmesi" konulu doktora tezi ile sağlanmıştır. Ulusal Bilim Fonu tarafından verilen Genç Araştırmacı Ödülü'nü kazanan Goldberg, dört yıl sonra 1989 yılında yayımladığı "Makine Öğrenmesi, Arama ve Optimizasyon İçin Genetik Algoritma" adlı kitabı ile genetik algoritmaya yeni bir boyut kazandırmıştır. Bu çalışma günümüzde dahi genetik algoritma konusunda en kapsamlı referans olma özelliğindedir (Körez, 2005: 1).

Genetik algoritma yöntemi, evrim teorisi esaslarına göre çalışarak, verilen bir sorun için en iyi çözüm veya çözümleri bulmaya yarar. Bu arayışı, karar değişkeni uzayındaki birçok başlangıç noktasından başlayarak, paralel işlemler dizisi ile en iyi yöne doğru gelişerek yapar. Karar uzayındaki bu noktalarda uygunluk derecelerinden başka bilgilere gerek yoktur. Toplumdaki noktaların paralel çalışarak en iyiye doğru gelişmesi rastgelelik ilkeleri ile sağlanmaktadır. Genetik algoritmanın esası doğal

seçme ve genetik kurallarına dayanmaktadır. Bu kurallar ortama en fazla uyum sağlayan canlıların hayata devam etmesi ve uyum sağlayamayanların da elenmesi olarak algılanmalıdır (Şen, 2004: 32).

Genetik algoritmalar daha çok (Erdem, 2008: 23):

1. Matematiksel modeli kurulamayan,
2. Çözüm alanı oldukça geniş,
3. Problemi etkileyen faktörlerin çok fazla olduğu problemlerin çözümünde etkin olarak kullanılmaktadır.

Genetik algoritmalar, doğadaki canlıların geçirdiği süreci örnek alır ve iyi nesillerin kendi yaşamlarını muhafaza edip kötü nesillerin yok olması prensibine dayanır. Bu algoritma, anne ve baba bireyden doğan yeni çocuk bireylerin şartlara uyum sağlayıp yaşamlarını devam ettirmesine dayanır. Yeni bireyler, anne ve babasından gelen iyi genleri bünyelerinde muhafaza edebileceği gibi kötü genleri de almış olabilir. Bu durumda kötü genlere sahip çocuk bireyler varlıklarını sürdüremeyecektir. Genetik algoritmalar doğadaki en iyinin yaşamasını gerektirmekte ve bunu belirleyen uygunluk işlevi, yeni çözümler üretmek için çaprazlama, kopyalama ve değiştirme gibi operatörleri kullanmaktadır. Genetik algoritmanın önemli özelliklerinden birisi de bir grup üzerinde çözümü araması ve bu sayede çok sayıda çözümün içinden en iyiyi seçmesidir (Biroğlu, 2005: 28).

Probleme ait en iyi çözümün bulunabilmesi için;

- Bireylerin gösterimi doğru bir şekilde yapılmalı,
- Uygunluk fonksiyonu etkin bir şekilde oluşturulmalı,
- Doğru genetik işlemciler seçilmelidir.

Genetik algoritmalar, problemin en iyi çözümünü bulmak için garanti vermezler ancak problemlere makul bir süre içinde, kabul edilebilir, iyi çözümler bulurlar (Çivril, 2009: 23).

### 3.1. GENETİK ALGORİTMALARIN KULLANILMA NEDENLERİ

İlk olarak diğer yöntemlerin tercih edilmeme sebepleri incelenmelidir (Polat, 2006: 6-7). Denklem optimizasyonunda;

1. Türev-İntegral hesabına dayananlar,
2. Sayılamaya dayananlar,
3. Rastgele aramalar olmak üzere üç tip yöntemden bahsedilir.

Türev-integral hesaplamalarına dayanan hesaplama yöntemleri çok derinlemesine çalışılmıştır. Bu yöntemler fonksiyonun türevinin köklerinin, fonksiyonun en küçük ve en büyük değer veren noktaları olmasından yararlanır. Gerçek problemler için sıfır veren noktaları bulmak da ayrı bir problemdir. Diğer bir yöntem ise alınan bir noktadan sadece yukarı ilerleyerek en iyi sonucu bulmayı hedefler. Tepe tırmanma (hill climbing) denen bu yöntem, fonksiyon grafiğinin tepelerini tırmanır. Ancak çok sayıda dönme noktası içeren bir fonksiyonda çok sayıda tepe oluşur. Hangi tepenin en iyi çözüm olduğu bilinemez. Sayımlama yöntemleri ise oldukça alışlagelmiştir. Gerçek sayı aralıkları belli sayıda parçaya ayrılır ve parçalar sürekli denir. Ancak problemler, böyle çözmek için büyük olabilir. Yöntemin biraz daha geliştirilmiş şekli dinamik programlamayla oluşturulur. Parçalar arasından iyi görünenler seçilir. Bu parçalar da parçalara ayrılarak işlem tekrarlanır. Bu yöntem de tepe tırmanma yöntemi gibi yanlış tepeleri araştırabilir. Dinamik programlama tepelerin olmadığı aralıklarda, başarılı ve hızlıdır.

Optimizasyonun;

- Bir işin daha iyi yapılması,
- En doğru şekilde yapılması olmak üzere iki amacı vardır.

Günümüzde rastgele aramaların kullanımı artmaktadır. Bu tip aramalar, optimizasyonu daha iyi yapma amacını sağlamakta daha başarılılardır. Genetik algoritmalar gibi sezgisel yöntemler, klasik yöntemlerin çok uzun zamanda yapacakları işlemleri kısa bir zamanda çok net olmasa da yeterli bir doğrulukla yapabilmektedir.

### 3.2. GENETİK ALGORİTMALARIN DİĞER YÖNTEMLERDEN FARKI

Goldberg'e göre genetik algoritmayı diğer arama yöntemlerinden ayıran en belirgin özellikleri çözüm arama şeklinin farklı oluşudur. Aşağıda bu farklılıklar açıklanmaktadır (aktaran Gözen, 2007: 36).

1. Genetik algoritma, parametre setlerinin kodları ile ilgilenir, parametrelerin kendileri ile doğrudan ilgilenmez,
2. Genetik algoritmanın arama alanı, yığının veya popülasyonun tamamıdır; tek nokta veya noktalarda arama yapmaz,
3. Genetik algoritmalarda, amaç fonksiyonu kullanılır, sapma değerleri veya diğer hata faktörleri kullanılmaz,
4. Genetik algoritmaların uygulanmasında kullanılan operatörler, stokastik yöntemlere dayanır, deterministik yöntemler kullanılmaz.

Başka bir ifadeyle genetik algoritmalar doğal olayların gelişmesindeki genetik mekanik ilkelere göre çalışırlar (Şen, 2004: 32).

1) Genetik algoritmalar, çözümlemesinde karar değişkenlerini genetik sayı sistemine göre kodlayarak kullanır. Sayı sisteminde karar değişkenlerinin genleri topluca karar uzayında bir noktayı temsil eder.

2) Genetik algoritmalarda bir nokta yerine aynı anda noktalar topluluğundan hareket edilir. Topluluğun genetik algoritmalar evrimi ile gelişmesi sonucunda en iyi çözüme ulaşılır. Evrim sırasında sistem yerel en iyiye takılmaz.

3) Genetik algoritmalar evrimi sırasında, karar değişkenlerinin belirttiği noktalardaki hedef fonksiyonu değerleri kullanılır. Türev ve integral işlemlerine gerek olmadığından başlangıç ve sınır şartları ile bazı klasik kabullerin yapılmasına gerek yoktur.

4) Genetik algoritmalar evrim işlemleri belirlilik değil kurallarına dayanır. Seçim işlemleri ihtimal ilkeleri ışığı altında yapılır. En iyi için sayılan klasik yöntemlerin dışında daha basit, fazla matematik gerektirmeyen ve objektif olan GA (genetik algoritma) yaklaşımları kullanılabilir. Genetik algoritmalarındaki rastgele yaklaşımlar sonuç çözümün yerel en iyi çözümlere takılıp kalmamasını sağlar.



### **3.3. TEMEL KAVRAMLAR**

Genetik algoritmanın çalışmasında ve başarılı çözüm değerlerine ulaşılmasında temel kavramların iyi anlaşılmasının ve belirlenmesinin önemi büyüktür. Aşağıda bu kavramlardan bahsedilmiştir.

#### **3.3.1. Gen**

Kalıtsal molekülde bulunan ve organizmanın karakterlerinin tayininde rol oynayan kalıtsal birimlere denir. Yapay sistemlerde gen, kendi başına anlamlı birer genetik bilgi taşıyan en küçük yapı birimi olarak alınır (İşler, 2009: 22).

Genetik algoritmanın kullanıldığı programlama yapısında bu gen yapıları programcının tanımlamasına bağlıdır. Bir genin içerdiği bilgi sadece ikili tabandaki sayıları içerebileceği gibi onluk taban ve onaltılık tabandaki sayı değerlerini de içerebilir. Dolayısıyla yazılan programa göre gen içeriği çok önem kazanmaktadır (Biroğlu, 2005: 34).

#### **3.3.2. Kromozom**

Bir ya da birden fazla gen yapısının bir araya gelerek oluşturduğu problemin çözümüne ait tüm bilgiyi içeren diziye kromozom denir. Kromozomlar, alternatif aday çözümleri gösterirler (Engin, 2011: 17).

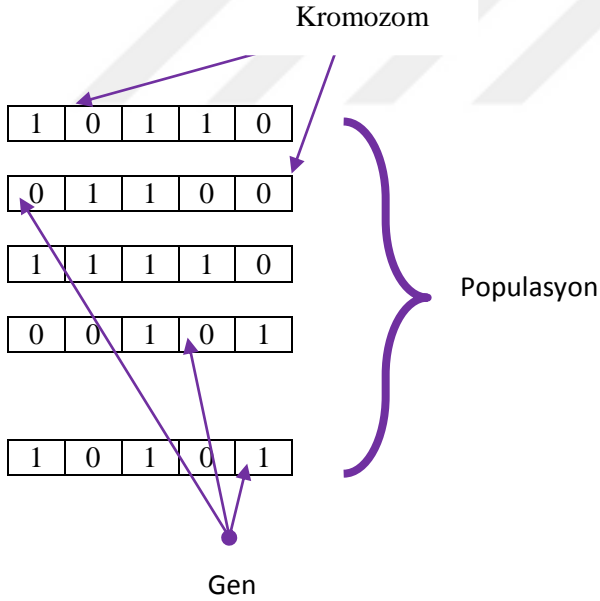
Kromozornların bir araya gelmesiyle popülasyon (yığın) oluşturulur. Yığındaki her bir bireye kromozom, kromozomdaki her bir bilgiye gen denir. Kromozomlar, üzerinde durulan problemin olası çözüm bilgilerini içermektedir (Biroğlu, 2005: 34). Kromozom, GA yaklaşımında üzerinde durulan en önemli birim olduğu için bilgisayar ortamında iyi ifade edilmesi gerekir (Elen, 2011: 16).

### 3.3.3. Populasyon

Populasyon, çözüm bilgilerini içeren kromozomların bir araya gelmesiyle oluşan olası çözüm yığına denir. Yığındaki kromozom sayısı sabit olup problemin özelliğine göre programlayıcı tarafından belirlenir (Biroğlu, 2005: 34).

Çözüm uzayı açısından populasyon büyüklüğünün küçük değerde olması çözüm uzayının küçük olmasını bu da aranan en iyi çözüm değerlerine ulaşılamamasına neden olmaktadır. Çözüm uzayının çok büyük değerde olması ise hem genetik algoritmanın etkinliğini azaltmakta hem de çözümün farklı noktalarda aranmasına neden olmaktadır (Biroğul ve Güvenç, 2007: 615). Şekil 3’de en küçük birim olan genin, kromozomları oluşturması; ardından biraraya gelen kromozomların ise populasyonu ortaya çıkarması özetlenmiştir.

**Şekil 3:** Gen, Kromozom ve Populasyon Yapısı



### 3.3.4. Kodlama

İlk adım; problem için arama uzayını en iyi temsil eden kodlama yapısının seçilmiş olmasıdır. Kodlama biçimi, genetik algoritmanın performansını oldukça

önemli oranda etkiler; fakat programa bağlı olduğundan bütün problemler için geçerli en uygun kodlama biçimini söylemek imkânsızdır (Elen, 2011: 14).

### 3.3.4.1. İkili (Binary) Kodlama

En yaygın olarak kullanılan kodlama türüdür. Burada kromozomlar 0 ve 1 şeklinde gen değerlerinde kodlanırlar. Bu dizideki her bit, çözümün belli karakteristiğini temsil eder veya tüm dizi bir sayıyı temsil eder. Şekil 4’de bir örnek ikili düzende kodlama verilmiştir (Bakır ve Altunkaynak, 2003: 462)

Şekil 4: İkili Düzende Kodlama Yapısı

Kromozom 1	1 0 1 0 1 1 0 1 1
Kromozom 2	0 0 1 0 1 0 1 0 1

### 3.3.4.2. Sıralı (Permütasyon) Kodlama

Sıralı (permütasyon) kodlama tekniği genellikle gezgin satıcı, çizelgeleme ve sıralama, şebeke tasarımları gibi sıra takibi olan problemlere uygulanır. Şekil 5’de örnek bir sıralı kodlama verilmiştir (Bakır ve Altunkaynak, 2003: 462).

Şekil 5: Sıralı Kodlama Yapısı

Kromozom A	1 3 5 7 2 8 4 6 9
Kromozom B	7 4 5 6 8 9 3 1 2

### 3.2.4.3. Değer (Alpha-Numeric Encoding) Kodlaması

Değer kodlama yönteminde ilgili parametre değerleri doğrudan alınır. Özel problemlerde alfa-sayısal ya da gerçel sayılar olarak kullanımı gerçekleştirilir. Şekil 6’de bir örnek değer kodlama verilmiştir (Bakır ve Altunkaynak, 2003: 462).

**Şekil 6:** Değer kodlama yapısı

Kromozom A	0.123 1.234 2.345 4.567 5.678
Kromozom B	A B C D E F G H I J K L M N

### 3.3.5. Seçim

Goldberg'e göre, yeni nesiller için ebeveyn kromozomların belirlenmesi sürecinde önceki popülasyondan gelen bazı kromozomların yeni popülasyona aktırılması gerekmektedir. Burada önemli olan bu kromozomların nasıl seçileceğidir. Darwin evrim teorisine göre; hayatta kalan en iyi kromozom ebeveyn olarak yeni oğul kromozomları oluşturur. En iyi kromozom seçilmesinde birçok yöntem bulunmaktadır. Her bir kromozomun uygunluk değeri hesaplandıktan sonra uygunluğu yüksek olan kromozomların seçilmesi için geliştirilmiş değişik seçim yöntemleri bulunmaktadır. Bu yöntemler şu şekilde sınıflandırılabilir (aktaran Aydemir, 2009: 47).

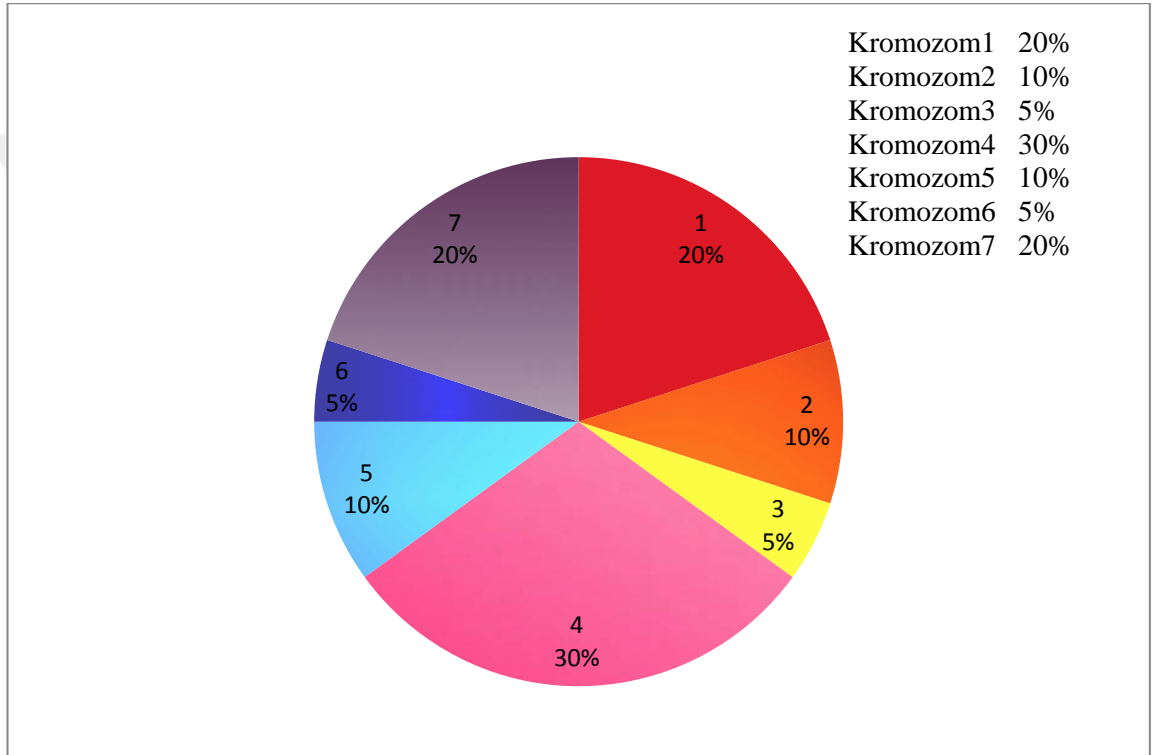
#### 3.3.5.1. Rulet Tekerli Seçim Yöntemi

Genetik algoritmalarda kullanılan en basit ve en yaygın seçim mekanizması rulet tekerleği (çemberi) seçimidir.

Rulet Tekerli seçim operatöründe, bütün kromozomlar uygunluk değerlerine göre bir rulet etrafında dizilirler. Rulet üzerinde uygunluk değerlerine göre sıralanan kromozomlar rasgele olarak seçilirler. Bu şekilde her birey seçilmek için kendi uygunluk değerine göre bu rulet tekerinden bir pay almaktadır. Daha büyük alana sahip bireyin seçilme şansı daha fazla olacaktır. Bu metot yardımıyla kromozomlar istatistiksel yöntemler kullanılarak uygunluk fonksiyonu değerlerinin toplam uygunluk fonksiyonuna oranları ölçüsünde seçilirler. Ancak bu seçim yöntemi, uyum değeri büyük olan bireylerin seçilme olasılığı yüksek olduğu için hep aynı kromozomların seçilmesine neden olmaktadır. Bu da popülasyon içindeki çeşitliliği etkileyerek sorun yaratır (Elen, 2011: 18).

Bu hata aynı zamanda algoritmanın zamansız yakınsamasına sebep olabilmektedir. Hatayı gidermek için çeşitli çalışmalar yapılmıştır. Dejong (1975) elitist, beklenen değer, elitist beklenen değer metodu gibi çeşitli seçim mekanizmalarını önermiştir (Biroğlu, 2005: 39). Şekil 7’de 5 kromozomun uygunluk değerleri ile dizilimi verilmiştir.

**Şekil 7:** Rulet Tekerleği Seçme Operatörü

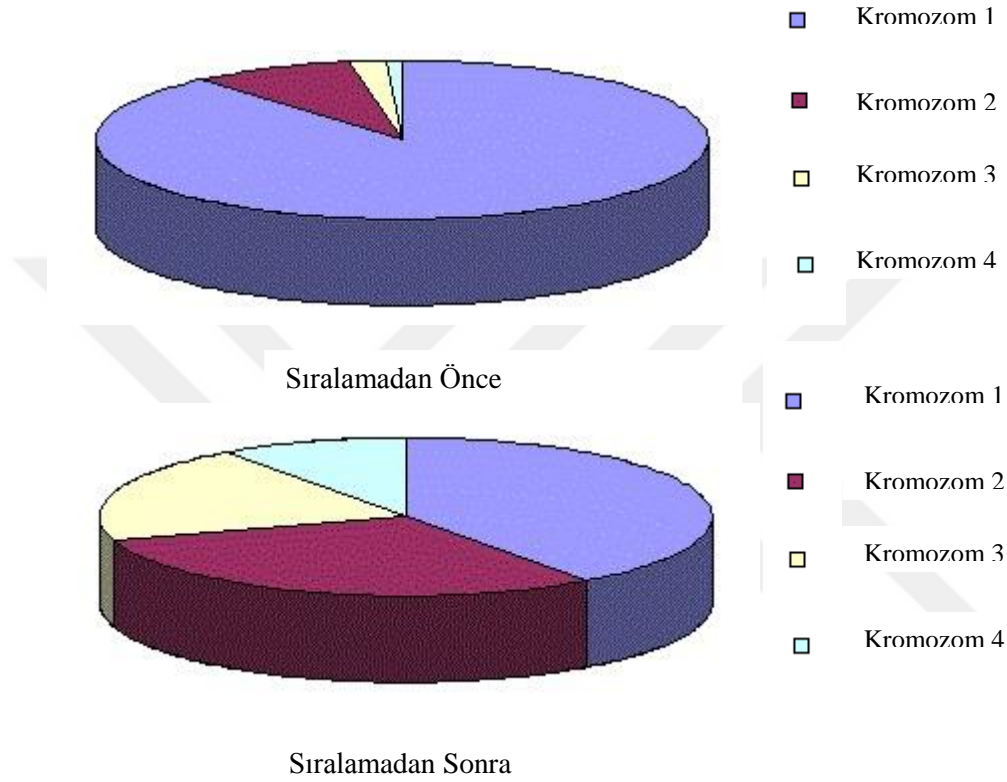


### 3.3.5.2. Sıralı (Rank) Seçim Yöntemi

Rulet seçimi eğer uyumluluk çok fazla değişiyorsa sorun çıkartabilir. Örneğin en iyi kromozomun uyumluluğu %90 ise diğer kromozomların seçilme şansı azalacaktır. Bunu önlemek için sıralı seçim kullanılabilir. Sıralı seçimde en kötü uyumlulukta olan kromozoma 1 değeri sonrakine 2 değeri verilir ve böylelikle seçilmede bunlara öncelik tanınmış olur. Bu şekilde onların da seçilme şansı artar. Fakat bu da çözümün daha geç yakınsamasına neden olabilir (Obitko, 1998b). Şekil

8’de sıralı seçim kromozomlara değer verilmesinden önceki ve sonraki durum gösterilmiştir.

**Şekil 8:** Sıralı Seçim



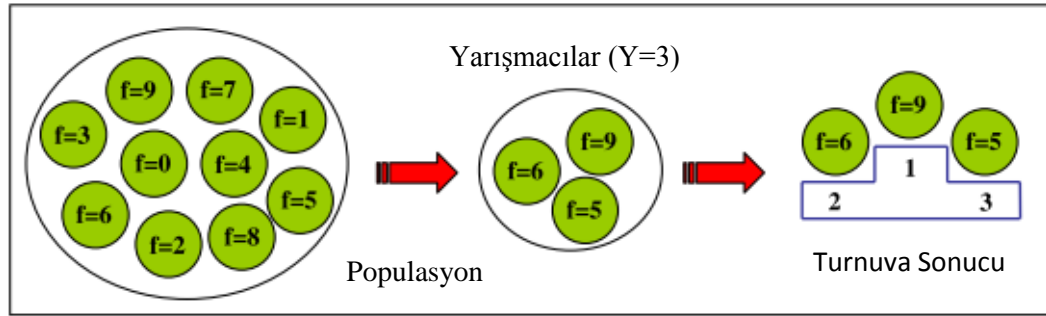
Kaynak: Obitko, 1998.

### 3.3.5.3. Turnuva Seçim Yöntemi

Bu seçim yönteminde, bireyler rasgele olarak gruplanır ve gruptaki bireyler aralarında seçim işlemi yapılmak üzere rekabete sokulur. Grup içinde en yüksek uygunluk değerine sahip olan birey, yeni nesli oluşturmak için ebeveyn bireylerden biri olarak seçilir. Bu işlem, toplam birey sayısına ulaşıncaya kadar devam eder. Bazı uygulamalarda grup büyüklüğü iki olarak seçilirken, bazılarında çok daha büyük gruplar oluşturulur. Şekil 9’da turnuva seçim yöntemi gösterilmiştir. Populasyondaki

bireylerin uygunluk değerleri  $f$  ile bireyler arasından rasgele seçilen grup büyüklüğü  $Y$  ile gösterilmiştir (Elen, 2011: 19).

**Şekil 9:** Turnuva Seçim Yöntemi



Kaynak: Elen, 2011: 20.

#### 3.3.5.4. Sabit Durum (Kararlı Hal) Seçim Yöntemi

Sabit durum metodunda, her nesilde yalnızca birkaç birey yer değiştirir. Çoğunlukla çok düşük uygunluk değerine sahip bireyler, çaprazlama ve mutasyon yöntemleriyle yeniden üretilerek yeni nesilde yer alırlar. Sabit durumlu GA'lar daha çok kural tabanlı sistemlerde kullanılır (Mitchell, 1996: 128).

#### 3.3.5.5. Seçkinlik (Elitizm) İşlemi

Mansfield'e göre üreme, çaprazlama ve mutasyon işlemleri sonrasında kuşakta bulunan en iyi uygunluk değerine sahip birey, sonraki kuşağa aktarılamayabilir. Bunu önlemek için bu işlemlerden sonra oluşan yeni kuşağa bir önceki kuşağın en iyi (elit) bireyi, yeni kuşaktaki herhangi bir birey ile değiştirilir. Buna elitizm adı verilir (aktaran Keskin, 2010: 41).

### 3.3.6. Genetik Algoritmaların Çözümünde Takip Edilecek İşlem Adımları

Croce ve diğerlerine göre bir problemin genetik algoritmalar ile çözümünde izlenecek işlem adımları aşağıdaki gibidir (aktaran İşler, 2009: 22-23).

Adım 1: Kullanıcının önceden tanımladığı kurallara göre genellikle rassal bir çözüm grubu seçilir veya kullanıcının kendisi ilk çözüm grubunu belirleyebilir. İlk çözüm grubuna başlangıç popülasyonu denir.

Adım 2: Her bir kromozom için bir uygunluk değeri hesaplanır; bulunan uygunluk değerleri dizilerin çözüm kalitesini gösterir. Popülasyonda yer alan en iyi uygunluk değerine sahip olan birey (kromozom), bir sonraki yeni nesile (popülasyon) doğrudan değiştirilmeden aktarılır.

Adım 3: İki grup dizi (kromozom), belirli bir seçim yöntemine (uygunluk değerlerine) göre rassal olarak seçilir.

Adım 4: Seçilen iki kromozom için rassal olarak genetik operatörler kullanılarak çaprazlama işlemi gerçekleştirilir. Sonuçta yeni popülasyonda yer alacak iki yeni birey (kromozom) oluşur. Çaprazlama, yeni popülasyonda yer alacak birey sayısına ulaşılan dek sürer.

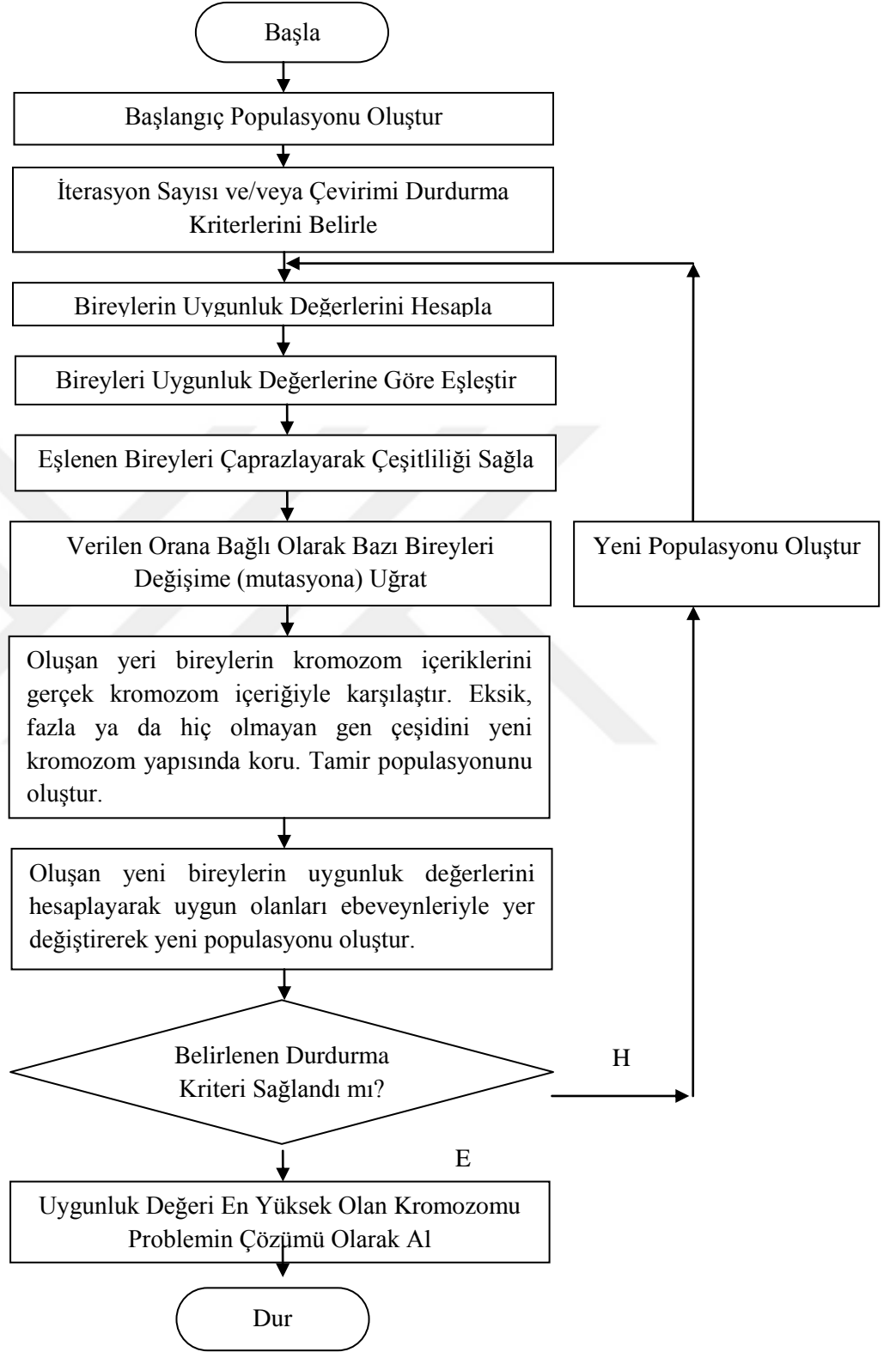
Adım 5: Yeni popülasyondaki bireyler, rassal olarak mutasyon işleminden geçerler.

Adım 6: Önceden belirlenen nesil sayısı boyunca yukarıdaki işlemler sürdürülür. Eğer en büyük nesil sayısına ulaşılmamışsa Adım 2'ye dönülür. En büyük nesil sayısına ulaşıncaya işlem bitirilir. Uygunluk değeri en yüksek olan kromozom (çözüm) seçilir.

Bu uygulama dizisi oldukça basit bir yöntemeye dayanır. En iyi uygunluk değerine sahip olan hayatta kalır. Çözümün üretkenlik başarısı, evrim boyunca belirlenmiş olan uygunluk değeriyle doğrudan bağlantılıdır. Bu sezgisel süreçte en az uygunluk değerine sahip olan çözüm, üremek için çok az bir şans bulabilecektir (Keskin, 2010: 37). Konunun özeti niteliğindeki genetik algoritmanın akış diyagramı Şekil 10'da verilmiştir.



Şekil 10: Genetik Algoritmanın Akış Diyagramı



Kaynak: Biroğlu, 2006: 31.

### 3.4. GENETİK OPERATÖRLER

Genetik algoritmalarda belirli noktalardan sonra nesil çeşitliliği olmamaktadır. Bunun için dizilere çaprazlama (crossing over) ve değişim (mutation) operatörleri belirli yüzdelik oranlarıyla uygulanarak nesil çeşitliliği sağlanır. Böylece çözümün belirli noktalara gelip tıkanması önlenmiş olur. Aşağıda bu genetik operatörlerin çeşitleri ve uygulamaları anlatılmıştır (Biroğlu, 2005: 41).

#### 3.4.1. Çaprazlama

Çaprazlama ebeveynlerden bazı genleri alarak yeni bireyler oluşturma işlemidir. Burada amaç, eldeki nesilden farklı nesiller elde etmektir. Çaprazlama yapılacak konum rastgele seçilir. Oluşan yeni birey ebeveynlerin bazı özelliklerini almış ve bir bakıma ikisinin kopyası olmuştur. Daha iyi performans almak amacıyla değişik çaprazlamalar kullanılabilir (Kurt ve Semetay, 2001: 3).

Kromozom 1 11011 | 00100110110

Kromozom 2 11011 | 11000011110

Birey 1 11011 | 11000011110

Birey 2 11011 | 00100110110

Çaprazlama operatörü, iki dizinin bir araya gelerek karşılıklı gen yapılarının değişimi ile yeni dizilerin oluşumunu sağlayan operatördür. Çaprazlanarak gen değişiminin yapılmasından önce dizilerin çaprazlamaya tutulma olasılığı belirlenmelidir. Literatürde bu oranın %50-%95 oranında uygulandığı görülmektedir (Biroğlu, 2005: 41). Çaprazlamada bir diğer önemli unsur ise ne tür bir çaprazlamanın yapılacağıdır.

### 3.4.3.1. İkili kodlama Düzeninde Çaprazlama Yöntemleri

İkili kodlama düzeni için çaprazlama yöntemleri tek nokta, iki nokta ve üç nokta çaprazlama yöntemleri olarak sınıflandırılmıştır.

- **Tek Nokta Çaprazlama Operatörü:** Bu işlemde kromozomlar rastgele bir yerinden kesilir ve sonra ilgili genler ile yer değiştirilir. Şekil 11’de rastgele değişiklik örneklenmiştir.

**Şekil 11:** Tek Noktalı Çaprazlama

	1	2	3	4	5	6	7	8	9	10	11	12
A <sub>1</sub> :	1	0	0	1	1	1	0	0	1	1	0	1
A <sub>2</sub> :	1	1	0	0	0	0	1	1	1	0	0	0
Çaprazlama Sonrası Oluşan Yeni Bireyler												
A <sub>1</sub> <sup>1</sup> :	1	0	0	1	1	1	0	0	1	0	0	0
A <sub>2</sub> <sup>1</sup> :	1	1	0	0	0	0	1	1	1	1	0	1

Kaynak: Biroğlu, 2005: 42.

- **İki Nokta Çaprazlama Operatörü:** Bazı durumlarda tek noktalı çaprazlama yöntemi yetersiz kalabilir ya da büyük parçalı blokların bozulması performansı düşürebilir. Bu sebeple iki noktalı çaprazlama yöntemi tercih edilebilir. İki noktalı çaprazlama operatöründe, rasgele iki nokta seçilir ve bu iki nokta arasında kalan bloklar kromozomlar arasında yer değiştirilir. Bu yöntem popülasyondaki kromozomların performansını arttırabilir. Şekil 12’de bu çaprazlama yöntemi için bir örnek verilmiştir (Elen, 2011: 20-21).

**Şekil 12:** İki Noktalı Çaprazlama

	1	2	3	4	5	6	7	8	9	10	11	12
A <sub>1</sub> :	1	0	0	1	1	1	0	0	1	1	0	1
A <sub>2</sub> :	1	1	0	0	0	0	1	1	1	0	0	0
Çaprazlama Sonrası Oluşan Yeni Bireyler												
A <sub>1</sub> <sup>1</sup> :	1	0	0	1	1	0	0	1	1	1	0	1
A <sub>2</sub> <sup>1</sup> :	1	1	0	0	1	1	0	0	1	0	0	0

Kaynak: Biroğlu, 2005: 42.

### 3.4.3.2. Sıralı Kodlama Düzeninde Çaprazlama Yöntemleri

Çizelgeleme problemlerinde sıkça kullanılan sıralı (permütasyon) kodlama düzeninde yer alan çeşitli çaprazlama yöntemleri mevcuttur. Bunlardan bazıları;

- **Pozisyona dayalı çaprazlama operatörü (PBX):** bu yöntemde çaprazlama kalıp olarak sabit kalacak olan gen hücrelerini belirler. Kalıpla işaretlenen noktalar dizide sabit kalırken diğer noktalarda iki birey yer değiştirilerek yeni bireylerin üremesi sağlanır. Bu durum Şekil 13’de gösterilmiştir (Aydemir, 2009: 44).

**Şekil 13:** Pozisyona Dayalı Çaprazlama

	1	2	3	4	5	6	7	8	9	10	11	12
A <sub>1</sub> :	3	4	7	1	1	0	4	8	9	2	3	3
A <sub>2</sub> :	0	0	1	4	7	2	8	9	2	1	0	0
Kalıp	1	1	1	0	0	0	1	1	0	0	1	0
Çaprazlama Sonrası Oluşan Yeni Bireyler												
A <sub>1</sub> <sup>1</sup> :	3	4	7	4	7	2	4	8	2	1	3	0
A <sub>2</sub> <sup>1</sup> :	0	0	1	1	1	0	8	9	9	2	0	3

Kaynak: Biroğlu, 2005: 43.

- **Sıraya dayalı çaprazlama operatörü (OBX):** kalıp üzerindeki 1 değerleri çaprazlamada kullanılacak değerleri gösterir. Bu tür çaprazlama, kromozomu oluşturan karakterlerin sayı ve sıralarının önem taşıdığı durumlarda kullanılır. Bu duruma örnek Şekil 14’de gösterilmiştir.

**Şekil 14:** Sıraya Dayalı Çaprazlama

	1	2	3	4	5	6	7	8	9	10	11	12
A <sub>1</sub> :	1	2	3	4	5	6	7	8	9	0	4	5
A <sub>2</sub> :	7	4	6	1	2	8	3	5	3	1	9	6
Kalıp	1	0	0	0	1	0	1	0	0	0	0	0
Çaprazlama Sonrası Oluşan Yeni Bireyler												
A <sub>1</sub> <sup>1</sup> :	1	7	2	4	5	6	3	8	9	0	4	5
A <sub>2</sub> <sup>1</sup> :	1	4	6	5	2	8	3	7	3	1	9	6

Kaynak: Biroğlu, 2005: 43.

- **Kısmi eşleşmeli çaprazlama operatörü (PMX):** iki bireyden rastgele bir aralık belirlenir. Bu aralıktaki değerler yer değiştirilir. Bu durum Şekil 15’de gösterilmiştir.

**Şekil 15:** Kısmi Planlı Çaprazlama

	1	2	3	4	5	6	7	8
A <sub>1</sub> :	2	8	6	4	5	7	1	3
A <sub>2</sub> :	8	7	2	1	3	4	5	6
Çaprazlama								
A <sub>1</sub> <sup>1</sup> :	2	8	2	1	3	7	1	3
A <sub>2</sub> <sup>1</sup> :	8	7	6	4	5	4	5	6

Kaynak: Biroğlu, 2005: 44.

Yer deęiřtirme sonunda dizide aynı olan deęerler deęiřtirilen deęerlerle tamamlanır. Őekil 16’da deęiřtirilen deęerler gzlenmiřtir.

**Őekil 16:** Kısmi Planlı aprazlama

	1	2	3	4	5	6	7	8
Oluřan Yeni Bireyler								
$A^1_1$ :	6	8	2	1	3	7	4	5
$A^1_2$ :	8	7	6	4	5	1	2	3

Kaynak: Biroęlu, 2005: 45.

### 3.4.2. Mutasyon

Kromozomların genleri veya genleri oluřturan kk birimleri zerinde deęiřiklik yapılmasını saęlayan iřlemcidir. Deęiřime uęratılacak kromozomun seiminde, kromozomun deęiřime uęrama ihtimalini gsteren ve bařlangıta sabit olarak tanımlanan bir deęiřim oranı sz konusudur. Genetik algoritmalarda deęiřime tabi tutulacak kromozomların belirlenmesinde bazılarının istisna tutulması veya zellikle deęiřime uęratılması gibi zel stratejiler tanımlanabilir (Erdem, 2008: 32).

Ama belli bir nesil sayısından sonra populasyon ierisindeki bireylerin gitgide birbirlerine benzemesine engel olmaktır. nk bu durum zm uzayının daralmasına neden olmaktadır. Bireylere ne kadar aprazlama operatr uygulansa da belli bir nesil sayısından sonra birey eřitlilięi saęlanmamaktadır. Bu durumda bireyi oluřturan genlerden rasgele bir tanesi seilir. Rasgele seilen genin deęeri deęiřtirilir. Bylelikle populasyon iindeki bireylerin eřitlilięinin devamı saęlanmış olunur (Ta, 2006: 18). Yapay sistemlerde mutasyon iřlemi esnasında kromozomdaki gen sayısı deęiřmez sabit kalır. Doęal populasyondaki mutasyon oranı olduka dřktr. Mutasyon frekansının byklę GA’nın performansını etkilemektedir. Mutasyon iřlemi bir tek kromozom zerinde yapılır (Engin ve Fıęlalı 2002).

Mutasyon oranı, tıpkı aprazlama oranında olduęu gibi mutasyonun olasılıęını ifade eden bir orandır. Yine rasgele yntemlerle kromozomun mutasyona

uğrayıp uğramayacağı belirlenir ve buna göre mutasyon gerçekleştirilir. Mutasyon oranı genellikle çok düşük (0,01) olduğundan mutasyon işlemi fertlerde az görülür (Baskak ve diğerleri, 2004: 7).

Mutasyonun sağladığı avantaj, problemin çözüm alanını araştırmada yön değişikliklerini sağlayarak araştırmannın kısır döngüye girmesini önlemektir (Erdem, 2008: 32). Mutasyon yöntemleri genel olarak beş farklı şekilde sınıflandırılmıştır.

#### 3.4.2.1. Ters Mutasyon

Bir bireyde rassal olarak iki pozisyon seçilir, bu iki pozisyondaki alt diziler ters çevrilir (Aydemir, 2009: 46-47). Şekil 17’de ters mutasyon örneklenmiştir.

Şekil 17: Ters Mutasyon

2	1	3	4	5	6	8	7
2	1	6	5	4	3	8	7

#### 3.4.2.2. Komşu İki Geni Değiştirme

Rassal olarak iki komşu iş yer değiştirebilir. Şekil 18’de bu durum örneklenmiştir.

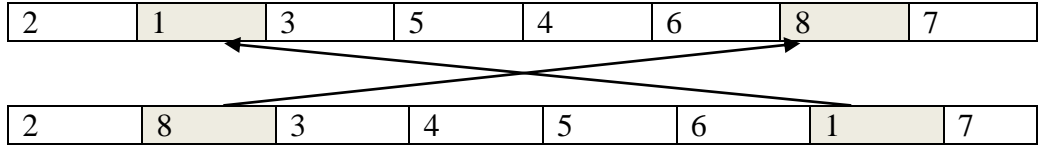
Şekil 18: Komşu İki Geni Değiştirme

2	1	3	5	4	6	8	7
2	1	3	4	5	6	8	7

#### 3.4.2.3. Keyfi İki Gen Değiştirme

Rassal olarak seçilen iki iş değiştirilebilir. Özel bir durum olarak, değiştirilebilen iki komşu işi bu mutasyon içerir. Şekil 19’da bu durum örneklenmiştir.

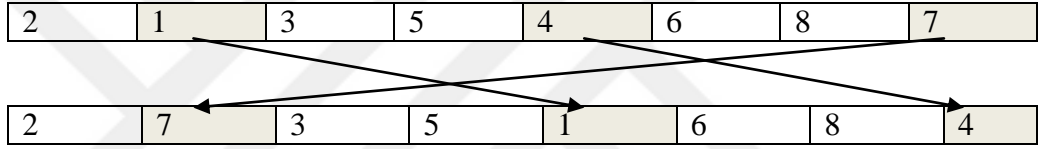
**Şekil 19:** Keyfi İki Gen Değişirme



#### 3.4.2.4. Keyfi Üç Gen Değişirme

Rassal seçilen üç iş keyfi olarak değiştirilir. Şekil 20’de bu durum örneklenmiştir.

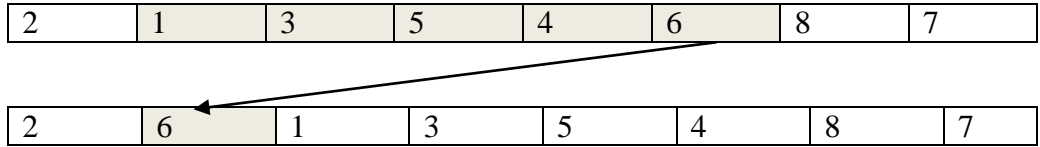
**Şekil 20:** Keyfi Üç Gen Değişirme



#### 3.3.4.5. Araya Yerleştirme

Rassal olarak seçilen bir kaydırma noktasında kromozomdaki bir iş kaydırılır ve diğer bir pozisyona yerleştirilir. Komşu iki iş değiştirme yönteminin özel bir durumudur. Keyfi üç iş değiştirmeye bir kesişime sahiptir. Şekil 21’de mutasyon bu durum örneklenmiştir.

**Şekil 21:** Araya Yerleştirme



#### 3.4.3. Durdurma Kriteri

Üreme, çaprazlama ve mutasyon işlemlerinden sonra yeni bir nesil oluşmaktadır. Tüm bu işlemler sonsuz döngü içerisinde yapılır. Eğer bir durdurma



kriteri belirlenmez ise bu süreç sonsuza kadar devam eder. Durdurma kriteri olarak şu yöntemler kullanılabilir (Elen, 2011: 28).

**a) Hesaplama zamanı kriteri:** bu yöntemde önceden bir hesaplama zamanı veya döngü sayısı belirlenmekte, bu zaman veya döngü sayısına ulaşıldığında durdurulmaktadır. Bu yöntemde belirlenen döngü sayısı gerektiğinden fazla ya da eksik olabilir.

**b) Optimizasyon hedefi kriteri:** önceden ulaşılmaması istenen amaç fonksiyonu değeri bilinmektedir. Uyum değeri bu değere ulaştığında algoritma durdurulmaktadır.

**c) Minimum iyileşme kriteri:** genetik algoritma problemlerinde bulunan en iyi çözümler önce hızlı daha sonra yavaş yavaş artış göstermektedir. Bulunan değerlerdeki iyileşme hızının giderek azalması ve sifıra yaklaşması, artık daha fazla iyileşme beklenmemesi gerektiğini gösterebilir. Çözümüne harcanacak zaman ile çözümden beklenecek kalite arasında bir denge kurularak durdurma gerçekleştirilir.

### 3.5. ÇİZELGELEME PROBLEMLERİNDE GENETİK ALGORİTMA

Önceki bölümlerde hemşire çizelgeleme problemine olan çözüm yaklaşımlarından detaylı bir şekilde bahsedilmiş ancak çözüm yöntemi olarak belirlenen genetik algoritmanın kullanım detayları verilmemişti. Bu bölümde genetik algoritmanın; nasıl, ne zaman, kimler tarafından çizelgeleme problemlerine uygulanmaya başlandığını açıklayan özet bir literatür taraması sunulacaktır.

Genetik algoritmaların çizelgeleme problemlerinin çözümünde kullanılması ilk kez 1985’de Davis tarafından yapılmıştır. Davis (1985) bu çalışmada, bir iş atölyesinden bahsetmekte ve kârı maksimize etmek için çizelgelemenin önemi üzerinde durmaktadır. Çalışmasında gerçekçi olmaktan ziyade basit ve öğretici bir çizelgeleme problemine genetik algoritmanın nasıl uygulanacağı anlatılmıştır. Bu çalışma daha sonraki çalışmalara temel olmuştur.

Davis’in çalışmasını, Liepins ve diğerlerinin (1987) “Açgözlü Genetikler” çalışması takip etmiş, açgözlü algoritmalar ile genetik algoritmalar kıyaslanmıştır. Genetik algoritma yapısı ilk kez iki makineli çizelgelemeye uygulanmıştır.

Bunlarla birlikte iş atölyesi çizelgeleme problemi için Biegel ve Davern'nin 1990'da, akış atölyesi problemi için Badami ve Parks'ın 1991'de, süreç planlama problemi için Vancza ve Markus'un 1991'de yayınlanmış çalışmaları bulunmaktadır (Yeniay, 2001: 43). Yeniay literatür taramasında çizelgemeyi 8 bölüme ayırarak 2000'li yıllara kadar yapılan çalışmaların taramasını vermiştir.

Dorndorf ve Pesch (1995), ilk defa öncelik kuralı tabanlı genetik algoritma yaklaşımı ile atölye tipi çizelgeleme problemi çözüm prosedürü geliştirmişlerdir (Aydemir, 2009: 14).

Ulusoy ve diğerlerinin (1997), çalışmalarındaki amaç eşzamanlı çalışan makinelerde, son işin tamamlanma zamanını en küçüklemedir. Problemin çözümü için genetik algoritma yöntemini uygulamışlar ve 180 test problemi üzerinde elde ettikleri sonuçları yayınlamışlardır. Sorunların %60'nın genetik algoritmada belirtilen alt sınıra göre optimale ulaştığı görülmüştür. Tüm sorunlar üzerinde alt sınır ortalama sapması 2.53% olarak bulunmuştur.

Çalışmaların sayısı giderek artmış, bilhassa 90'lı yıllarda yoğun ilgi görmüştür. Bu durum detaylı bir bibliyografyadan alınan Tablo 2'de özetlenmiştir. 80'li yıllardan günümüze kadar ulaşan çalışmaların sayısını gösteren bu bibliyografya 2014 yılıyla sınırlı olsa da 900 kaynağa atıfta bulunması açısından titizlikle hazırlanmış, oldukça faydalı bir metindir.

**Tablo 2:** Katkıların Yıllara Göre Dağılımı

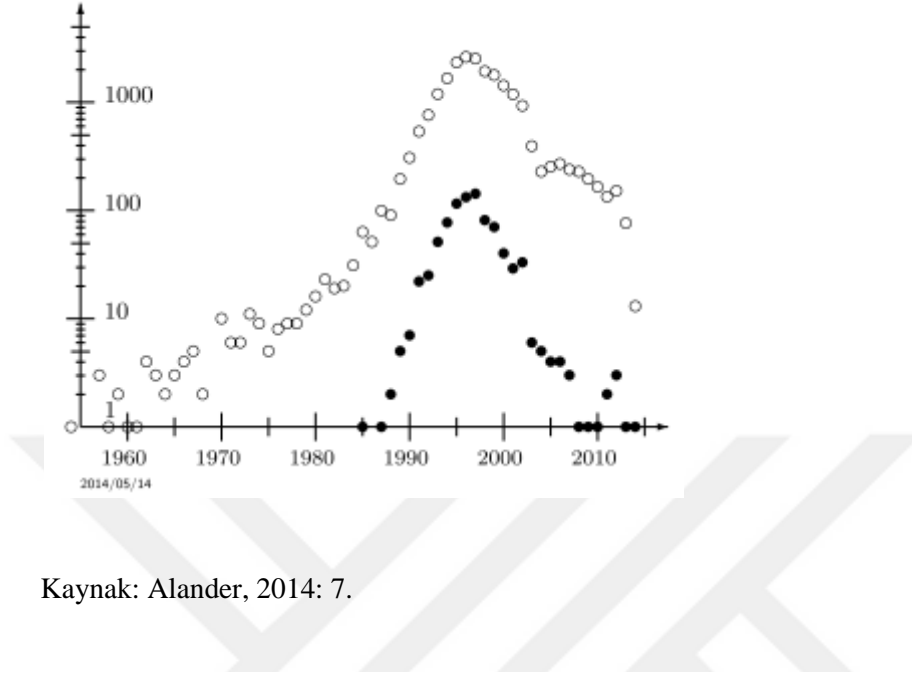
Yıl	Öğeler	Yıl	Öğeler
1985	1	1986	0
1987	1	1988	2
1989	5	1990	7
1991	22	1992	25
1993	51	1994	77
1995	115	1996	132
1997	142	1998	81
1999	70	2000	40
2001	29	2002	33
2003	6	2004	5
2005	4	2006	4
2007	3	2008	1
2009	1	2010	1
2011	2	2012	3
2013	1	2014	1
<b>Toplam</b>			<b>865</b>

Kaynak: Alander, 2014: 9.

Zhou ve diğerleri (2001) için geleneksel çözüm yöntemleri NP-Hard grubuna giren çizelgeleme problemleri için yeterli değildir. Genetik algoritmaların planlama ve zamanlama konusunda daha etkin olduğu kanıtlanmıştır. Ancak bu etkinlik tatmin edici olmadığından bu çalışmada hibrit sezgisel genetik algortimalar önerilmiştir. Geliştirdikleri melez GA yönteminin daha verimli ve etkili sonuçlar verdiğini tespit etmişlerdir. Buna ek olarak, komşuluk arama tekniği çözüm performansını artırmak için yardımcı bir yöntem olarak kabul edilmiştir.

Watanabe ve diğerleri (2005), atölye tipi çizelgeleme problemleri için modifiye edilmiş bir genetik algoritma önermişlerdir. Önerilen yöntemin etkinliğini göstermek için sayısal deneyler karşılaştırılmıştır. Mevcut genetik algoritmaların daha iyi performansa sahip olduğu görülmüştür.

**Tablo 3:** Çizelgelemede Genetik Algoritma



Kaynak: Alander, 2014: 7.

Xu ve diğerleri (2014) sezgisel tabanlı görev planlama üzerinde durmuşlardır. Heterojen bir bilgisayar sisteminde farklı öncelikler farklı tamamlanma süreleri doğurmaktadır. İyi bir çizelgeleme algoritması için kaynakların bağlı olduğu her bir alt görev için bir öncelik atamak gerekir.

Kucukkoc ve Zhang (2016) karma model paralel iki taraflı montaj hattı dengeleme sıralama sorununun çözümü için melez bazlı karınca kolonisi optimizasyonu genetik algoritma yaklaşımını geliştirmişlerdir. Mevcut karınca kolonisi optimizasyonu algoritması, genetik algoritma tabanlı yeni bir model sıralama mekanizması entegrasyonu ile geliştirilmiştir. Kromozomların evrimini göstermek için sayısal bir örnek verilmiştir. Yeni yöntemin daha iyi kalitede çözümler bulmaya yardımcı olduğu kanıtlanmıştır.

### **3.6. GENETİK ALGORİTMALARIN ÇÖZÜMÜNDE KULLANILAN BİLGİSAYAR PROGRAMLARI**

Genetik algoritmaların çözümü için tercih edilebilecek pek çok bilgisayar programı ve kullanıcı tarafından oluşturulan arayüzler mevcuttur. Bu bölümde

bunlara kısaca değinilecek ve neden bunların arasından MATLAB programının seçildiği üzerinde durulacaktır.

Cai ve Li (2000), genetik algoritmayı C dilinde programlamayı tercih etmişlerdir. Sorunu çok kriterli optimizasyon modeli olarak formüle etmişlerdir. Temel amaçları işgücü taleplerini karşılamak için personel atama esnasında, toplam maliyeti minimize etmektir.

Divina ve Marchiori (2002), Prolog programlama dilini kullanmışlardır. Prolog mantık programlama dillerindedir. Yapay zekâ uygulamalarında yaygın olarak kullanılır.

Taç (2006), eğitim kurumlarındaki ders, dersane, sınıf ve öğretmen bilgilerinin tanımlandığı C#.NET dilinde bir yazılım geliştirmiştir. Programda hazırlanan ara yüz pencereleri aracılığıyla genetik algoritmanın parametreleri, ders, dersane, sınıf, öğretmen bilgileri tanımlanabilmekte ve SQL Server 2000 veritabanında saklanabilmektedir.

Polat (2006) GOAL paket programıyla düzlem ve uzay kafes sistemlerin optimum boyutlandırılmasını yapan bir algoritma yazmıştır. Genetik algoritma ile kafes sistemlerinin çok amaçlı optimizasyon algoritmasını oluşturmuştur.

Paksoy (2007) kaynak kısıtlı proje çizelgeleme probleminin çözümüne yönelik olarak; Delphi 6.0 ve Access 2000 kullanmıştır. Geliştirilen bilgisayar destekli genetik algoritmada, kullanılan örneklerin özelliklerine, genetik algoritmaların performanslarının ölçümünde kullanılan yöntemlere yer vermiştir.

Gen ve diğerleri (2008) kitaplarının 9.bölümünde, gerçek bir problem olan Çin'deki Şangay konteyner terminali şirketlerinden birini incelemiştir. Simülasyon Pentium 4 işlemci (2.6 GHz saat) ile C # 'da yapılmıştır.

Erdem (2008) Delphi kodlama dilinde yazılmış MSSQL Server veri tabanı kullanan bir program kullanmıştır. Programın ana menüsünde üç modül vardır. İlki çizelge yapılacak işlerin, makinelerin girildiği veri girişi modülüdür. İkincisi genetik algoritmanın çalıştırıldığı hesaplatma modülü, üçüncüsü ise genetik algoritmanın oluşturduğu çizelgenin durumunun görüldüğü, durum izle modülüdür.

Çivril (2009) çalışmasında, genetik algoritma kullanılarak aylık hemşire nöbet çizelgesi yazılımını hazırlamıştır. Bu yazılım için C# programı ve veritabanı olarak Microsoft Access 2007 programı kullanmıştır.

Keskin'in (2010) kullandığı algoritma, Delphi programla dilinde kodlanmış, 7.0 versiyonu derleyicide derlenerek uygulama dosyası haline getirilmiştir. Kodlanan çok amaçlı melez genetik algoritma uygulaması, problem yapısına ve büyüklüklerine göre farklı parametreler için çalıştırılmış ve yapay sinir ağları algoritması ile karşılaştırılmıştır.

Lin ve diğerleri (2015) çalışmalarında tüm deneysel simülasyonları C ++ programlama dili ile geliştirmişlerdir. Uygulanan optimizasyonun çözücüsü IBM ILOG CPLEX Studio 12.4 'dür. Tüm deney sonuçları; Windows 7 64-bit platformu, 16 GB RAM bilgisayar ve 3.40 GHz hızın üzerinde çalışan bir Intel i7-3770 işlemci ile elde etmişlerdir.

Bu tezde ele alınan problemin çözümünde MATLAB programı tercih edilmiştir. MATLAB (matrix laboratory), çok paradigmatlı sayısal hesaplama yazılımı ve dördüncü nesil programlama dilidir. MathWorks tarafından geliştirilmektedir. MATLAB kullanıcıya, matris işleme, fonksiyon ve veri çizme, algoritma uygulama, kullanıcı arayüzü oluşturma, C, C++, Java, ve Fortran gibi diğer dillerde yazılmış programlarla arabağlama imkanı tanır (Vikipedi, 2016).

Özellikle mühendislik, bilim ve ekonomi gibi çeşitli alanlarda kullanıcılara hizmet veren MATLAB esnek bir program olması sebebiyle tercih edilmiştir. MATLAB'i çok gelişmiş özellikleri olan, programlanabilen bir bilimsel hesap makinesine benzetmek mümkündür. MATLAB'de yazılan programlar, MATLAB'in kendine özgü dili kullanılarak yazılır ve MATLAB içinden çalıştırılır. Adından da anlaşılacağı üzere matrisler (matrix) yani diğer bir deyişle diziler (array) ile çalışır. Bu kadar çok tercih edilmesindeki bir başka neden; içerisindeki hazır fonksiyon, teorem, vs. gibi araçlardır. MATLAB'ın aynı zamanda diğer dillerle de etkileşim halinde olması sayesinde, mevcut dillerdeki çeşitli kütüphanelere benzer fonksiyonlar eklenebilir.

### **3.7. MATLAB PROGRAMINDA GENETİK ALGORİTMALAR**

Bu bölümde MATLAB programının uygulamalarından biri olan genetik algoritmalar araç kutusuna daha yakında bakılacaktır. Ayrıca programda problem

kurulumunun nasıl yapılacağı ve sonuçlar incelenecek, ek olarak seçeneklere göz gezdirilecektir (RMATLAB, 2015b).

### 3.7.1. Problem Kurulumu ve Sonuçları

#### Problem

**Uygunluk fonksiyonu:** Amaç fonksiyonunun minimize edilmesini sağlar.

**Değişkenlerin sayısı:** uygunluk fonksiyonu için bağımsız değişken sayısıdır.

#### Kısıtlar

**Doğrusal eşitsizlikler formu**  $A*x \leq b$ , A matrisi ve b vektörü tarafından belirlenir.

**Doğrusal eşitlikler formu**  $A_{eq}*x = b_{eq}$ ,  $A_{eq}$  matrisi ve  $b_{eq}$  vektörü tarafından belirlenir.

**Sınırlar :** değişkenler üzerinde ki alt ve üst sınırlardır.

- Alt, vektör olarak alt sınırları belirtir.
- Üst, vektör olarak üst sınırları belirler.


**Doğrusal olmayan kısıtlama fonksiyonu :** doğrusal olmayan kısıtları tanımlar. Doğrusal olmayan eşitlikler  $ceq=0$  formu ve doğrusal olmayan eşitsizlikler  $c \leq 0$  formudur. Ayrıntılar için, Doğrusal Olmayan Kısıtlamalara bakılabilir.

**Tamsayı değişken endeksleri:** girdileri tamsayı değerli x bileşenlerini veren bir vektördür. Eğer boş değilse;

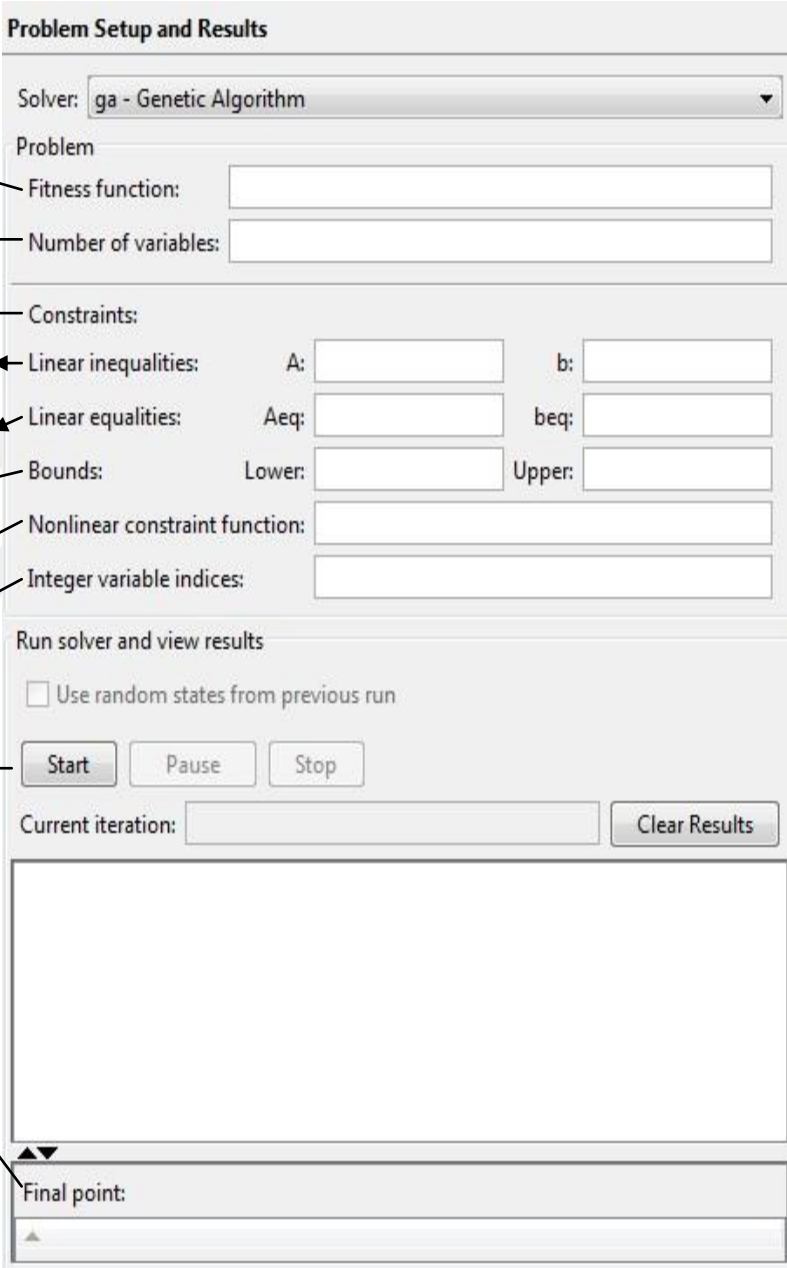
- $A_{eq}$ 'nin da  $b_{eq}$ 'nin da boş olması gerekir.
- Herbir doğrusal olmayan kısıt fonksiyonu,  $ceq$  için boşa ([]) dönmek zorundadır.
- Populasyon tipi çift vektör olmalıdır.

Ayrıntılar için Tamsayılı GA Çözücü Özelliklerine bakılabilir.

**Çözücüü çalıştırma ve sonuçları görüntüleme:** çözücüü çalıştırmak için **Başlat** düğmesine tıklattılır. Algoritma sona erdiğinde, durum ve sonuçlar bölümünde algoritmanın sonlandırılma nedeni görüntülenir. Nihai nokta güncellemeleri son

noktanın koordinatlarını göstermek içindir. **Final noktası** görülemiyorsa  düğmesine tıklanabilir.

Şekil 22: Problem Kurulumu ve Sonuçları



The image shows a software interface for setting up and running a Genetic Algorithm (GA) problem. The window is titled "Problem Setup and Results".

**Problem Setup:**

- Solver:** A dropdown menu set to "ga - Genetic Algorithm".
- Problem:**
  - Fitness function:** An empty text input field.
  - Number of variables:** An empty text input field.
- Constraints:**
  - Linear inequalities:** Two input fields labeled "A:" and "b:".
  - Linear equalities:** Two input fields labeled "Aeq:" and "beq:".
  - Bounds:** Two input fields labeled "Lower:" and "Upper:".
  - Nonlinear constraint function:** An empty text input field.
  - Integer variable indices:** An empty text input field.

**Run solver and view results:**

- Use random states from previous run
- Start** (button), **Pause** (button), **Stop** (button)
- Current iteration:** An empty text input field.
- Clear Results** (button)

**Results:**

- Final point:** A large empty text area for displaying the results.

**Annotations (Left side):**

- Uygunluk fonksiyonu (Fitness function)
- Değişkenlerin sayısı (Number of variables)
- Kısıtlar (Constraints)
- Doğrusal eşitsizlikler formu (Linear inequalities form)
- Doğrusal eşitlikler formu (Linear equalities form)
- Sınırlar (Bounds)
- Doğrusal olmayan kısıtlama fonksiyonu (Nonlinear constraint function)
- Tamsayı değişken endeksleri (Integer variable indices)
- Başlat (Start)
- Final noktası (Final point)



### 3.7.2. Seçenekler

Genetik Algoritma çözücü seçeneklerini belirler.

#### Populasyon

**Populasyon seçenekleri:** genetik algoritma için populasyon seçeneklerini belirtir.

**Populasyon tipi:** uygunluk fonksiyonu girdi türünü belirtir.

**Populasyon büyüklüğü:** her nesilde kaç kişi olduğunu belirtir. Eğer populasyon büyüklüğü 1'den büyük bir vektör olarak ayarlanırsa algoritma birden fazla altpopulasyon oluşturur. Vektörün her girdisi alt populasyonun boyutunu belirler.

**Oluşturma fonksiyonu:** ilk populasyonun populasyonunu belirleyen fonksiyondur.

**İlk populasyon:** genetik algoritma için bir başlangıç populasyonu belirlenmesini sağlar. Eğer ilk populasyon belirtilmezse algoritma oluşturma işlevini kullanarak oluşturur. Bireylerin populasyon büyüklüğü daha az belirtilebilir, bu yapılırsa, oluşturma fonksiyonu geri kalanını oluşturur.

**Başlangıç puanları:** ilk populasyon için puanları belirtmeyi sağlar. Eğer başlangıç puanları belirtilmediyse algoritma uygunluk fonksiyonunu kullanarak puanları hesaplar.

**Başlangıç aralığı:** başlangıç populasyonundaki vektörlerin girdileri için alt ve üst sınırları belirler. Başlangıç aralığı 2 satır ve başlangıç uzunluğu sütunları ile bir matris gibi gösterilebilir. İkinci satır üst sınırları içerirken ilk satır, ilk populasyon vektörlerinin girdileri için alt sınırı içerir.

Şekil 23: Populasyon

Population

Populasyon seçenekleri ← Population type: Double vector

Populasyon tipi ← Population size:  Use default: 50 for five or fewer variables, otherwise...  
 Specify:

Oluşturma fonksiyonu ← Creation function: Constraint dependent

İlk populasyon ← Initial population:  Use default: []  
 Specify:

Başlangıç puanları ← Initial scores:  Use default: []  
 Specify:

Başlangıç aralığı ← Initial range:  Use default: [-10;10]  
 Specify:

### Uygunluk ölçeklendirilmesi

Ölçekleme fonksiyonu, uygunluk fonksiyonu tarafından döndürülen işlenmemiş uygunluk puanlarını, seçim fonksiyonu için uygun aralıkta olan değerlere dönüştürür.

**Ölçekleme fonksiyonu:** ölçeklemeyi gerçekleştiren fonksiyonu belirtir.

Şekil 24: Uygunluk Ölçeklendirilmesi

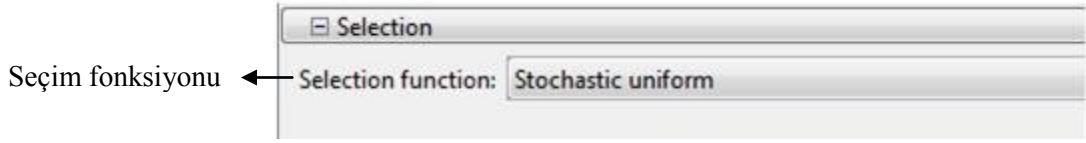
Fitness scaling

Ölçekleme fonksiyonu ← Scaling function: Rank

### Seçim

**Seçim fonksiyonu** bir sonraki nesil için kendi ölçekli değerlerine dayalı bir aile seçer.

**Şekil 25:** Seçim

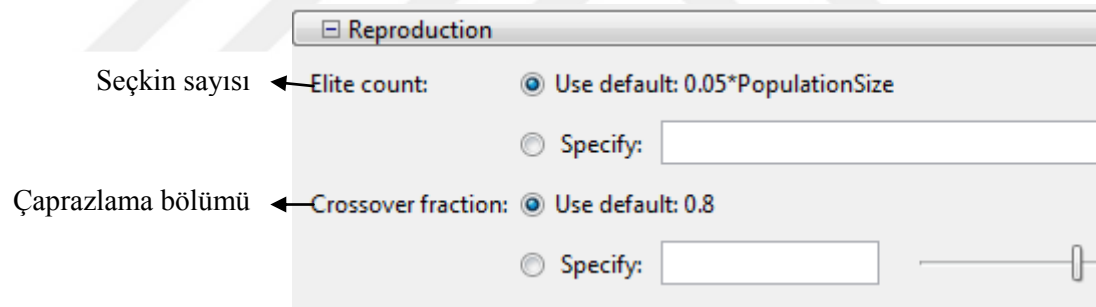


### Yeniden üreme

Üreme seçenekleri, genetik algoritmanın her yeni nesilde çocukları nasıl oluşturduğunu belirler.

**Seçkin sayısı:** gelecek kuşaklarda hayatta kalmasının garantisi olan bireylerin sayısını belirtir. **Çaprazlama bölümü:** çaprazlanarak üretilen gelecek neslin kesimini gösterir. Çaprazlama kesimi 0 ile 1 arasında bir bölüm olarak yapılabilir.

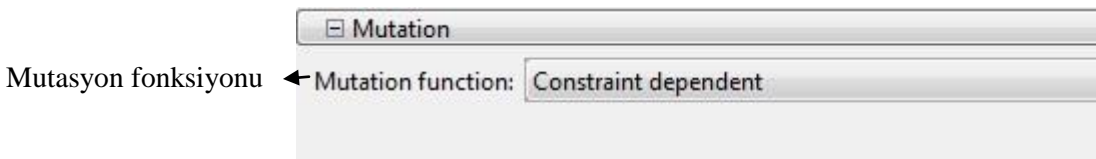
**Şekil 26:** Yeniden Üreme



### Mutasyon

**Mutasyon fonksiyonları** genetik çeşitlilik sağlamak ve daha geniş bir alan aramak için popülasyondaki bireylerde küçük rastgele değişiklikler yapar.

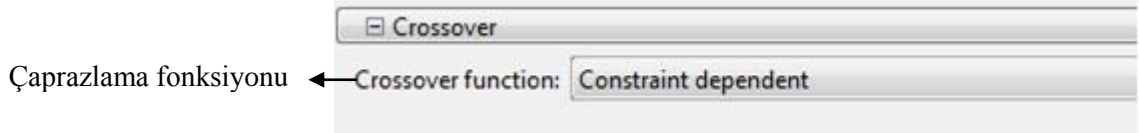
**Şekil 27:** Mutasyon



## Çaprazlama

**Çaprazlama fonksiyonu** yeni bir nesil üretmek için iki birey ya da aileyi bir araya getirir

**Şekil 28:** Çaprazlama



## Geçiş

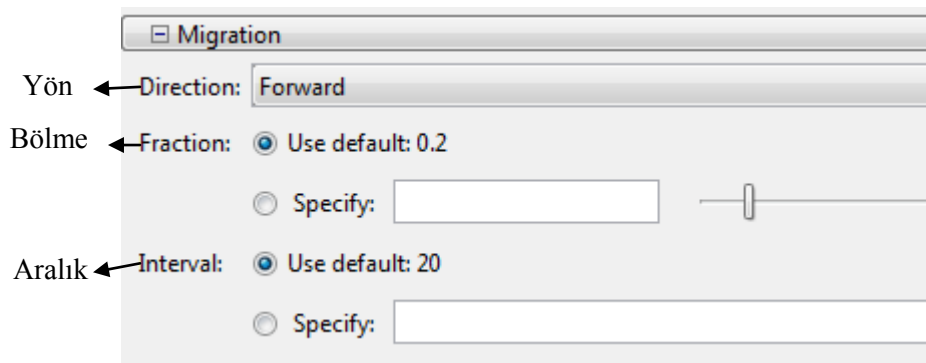
Geçiş bireylerin altpopulasyonlar arasındaki algoritmanın yarattığı hareketidir. Geçişin meydana gelmesi aşağıdaki üç parametre ile kontrol edilir.

**Yön:** göçün hangi yöne gerçekleşebileceğini belirler. Geçiş altpopulasyonları sonlara kaydırır. Bu son altpopulasyonun ilke, ilkin sona geçiş yapabileceğinin ifadesidir.

**Bölme:** alt populasyonlar arasında kaç kişinin hareket edeceğini denetler. Kesir hareket eden iki altpopulasyonun küçük kesimidir. Bir altpopulasyondan diğerine kopyalanan bireyler ana altpopulasyondan kaldırılmazlar.

**Aralık:** geçiş arasında kaç kuşağın olduğunu kontrol eder. Örneğin eğer 20 aralığı ayarlanırsa, alt populasyonlar arasındaki geçiş her 20 nesilde bir gerçekleşir.

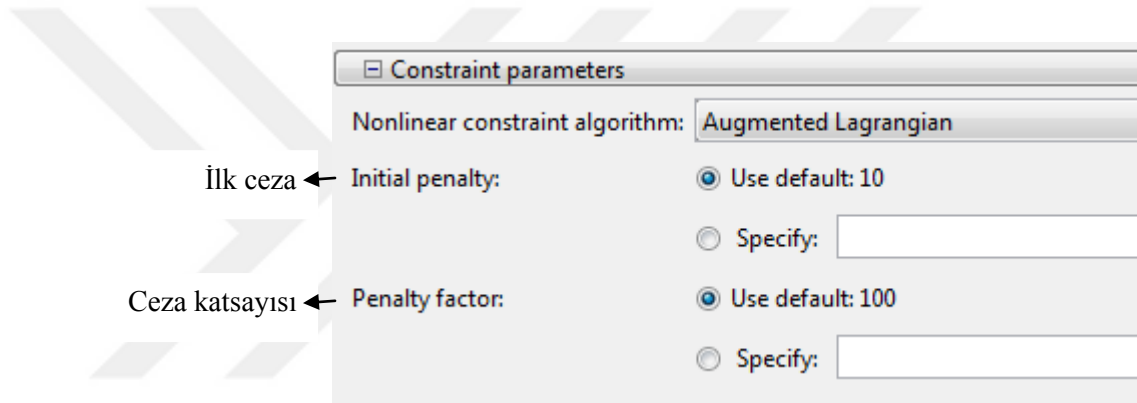
**Şekil 29:** Geçiş



## Kısıtlama Parametreleri

Kısıtlama parametreleri doğrusal olmayan kısıt çözücüyeye karşılık gelir. **İlk ceza** algoritma tarafından kullanılacak bir başlangıç değeri belirtir. İlk ceza 1'e eşit veya daha büyük olmalıdır. **Ceza katsayısı** problem gereken hassasiyete çözülmediyse ve kısıtlar karşılanmadıysa ceza parametresini artırır. Ceza değeri 1'den daha büyük olmalıdır. Algoritmalar hakkında daha fazla bilgi için Doğrusal Olmayan Kısıt Çözücü Algoritmalarına bakılabilir.

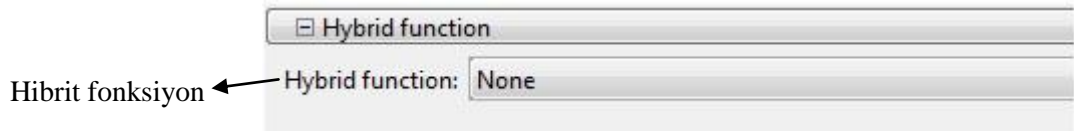
**Şekil 30:** Kısıtlama Parametreleri



## Hibrit Fonksiyon

Hibrit işlevi genetik algoritma sonlandıktan sonra çalışan başka minimize fonksiyonunun belirtilmesini sağlar.

**Şekil 31:** Hibrit Fonksiyon



## Durdurma Kriteri

Durdurma kriteri algoritmanın sonlandırılma nedenlerini belirler.

**Nesiller:** genetik algoritmanın gerçekleştireceği maksimum yineleme sayısını belirtir.

**Zaman sınırı:** genetik algoritmanın durmadan önce saniye cinsinden çalıştığı en uzun süreyi belirtir.

**Uygunluk değeri:** en iyi uygunluk değeri limitine eşitse algoritma durur.

**Geciktirme nesilleri:** uygunluk fonksiyon değerindeki ortalama değişim fonksiyonun toleransından azsa algoritma durur.

**Geciktirme zaman sınırı:** belirtilen saniye içinde bir zaman aralığı için en iyi uygunluk değerine dair hiçbir gelişme yoksa algoritma durur.

**Geciktirme testi:** N nesil için N geometrik ağırlık faktörünün geometrik ağırlıklı ortalama değişiminin (1/2) olup olmadığını belirler.

**Fonksiyon toleransı:** uygunluk fonksiyonunda ortalama değişim değeri fonksiyon toleransından azsa algoritma durur.

**Kısıt toleransı:** doğrusal olmayan kısıtlama ihlali için maksimum toleranstır.

Şekil 32: Durdurma Kriteri

Stopping criteria

Nesiller ← Generations:  Use default: 100\*numberOfVariables  
 Specify:

Zaman sınırı ← Time limit:  Use default: Inf  
 Specify:

Uygunluk değeri ← Fitness limit:  Use default: -Inf  
 Specify:

Geciktirme nesilleri ← Stall generations:  Use default: 50  
 Specify:

Geciktirme zaman sınırı ← Stall time limit:  Use default: Inf  
 Specify:

Geciktirme testi ← Stall test: average change

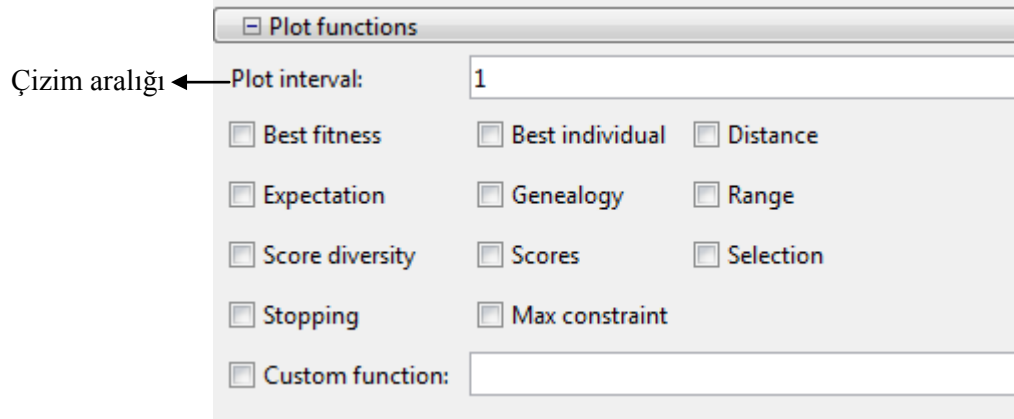
Fonksiyon toleransı ← Function tolerance:  Use default: 1e-6  
 Specify:

Kısıt toleransı ← Constraint tolerance:  Use default: 1e-3  
 Specify:

### Çizim Fonksiyonları

Çizim fonksiyonları genetik algoritmaların çeşitli yönlerini çizmeye olanak tanır. Her biri ekran penceresinden ayrı bir eksende çizilir. **Çizim aralığı** ardışık güncellemeler arasında nesillerin sayısını belirtir. Bir çalışma işlemini kesmek için pencerede Durdur düğmesi kullanılabilir.

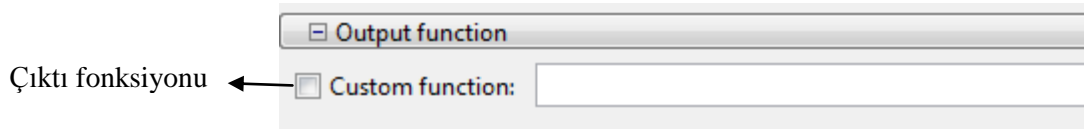
**Şekil 33:** Çizim Fonksiyonları



### Çıktı Fonksiyonu

**Çıktı fonksiyonları** genetik algoritmanın her adımda aradığı fonksiyonlardır. Fonksiyon kolları ile birden fazla çıkış fonksiyonu kullanılabilir.

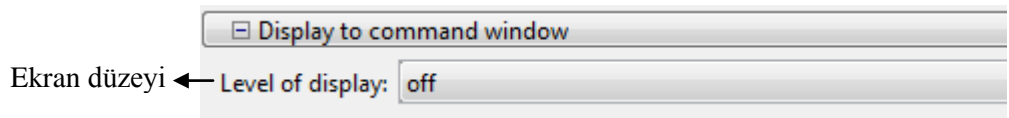
**Şekil 34:** Çıktı Fonksiyonu



### Komut Penceresinde Gösterim Ekranı

Ekranın düzeyi algoritma çalıştırıldığında MATLAB® komut penceresinde görüntülenen bilgi miktarını belirtir.

**Şekil 35:** Komut Penceresinde Gösterim Ekranı

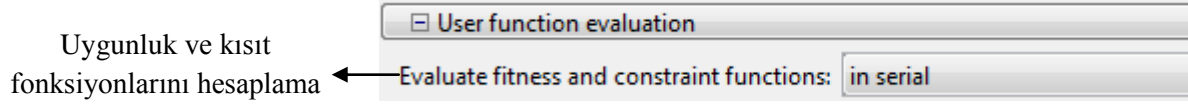


### Kullanıcı Fonksiyonunu Değerlendirme



Uygunluk ve kısıt fonksiyonlarının nasıl değerlendirildiğini gösterir.

**Şekil 36:** Kullanıcı Fonksiyonunu Değerlendirme



### Genetik Algoritma Kullanarak Fonksiyonun Minimumunu Bulma

MATLAB programında genetik algoritmalarla problemi çözmek için pek çok sözdizimi mevcuttur. Bunlardan birinin seçimi ihtiyaçlar doğrultusunda yapılmaktadır. Aşağıda bu dizimlere yer verilmiştir (MathWorks, 2016)

```
x= ga(fitnessfcn,nvars)
x= ga(fitnessfcn,nvars,A,b)
x= ga(fitnessfcn,nvars,A,b,Aeq,beq)
x= ga(fitnessfcn,nvars,A,b,Aeq,beq,LB,UB)
x= ga(fitnessfcn,nvars,A,b,Aeq,beq,LB,UB,nonlcon)
x= ga(fitnessfcn,nvars,A,b,Aeq,beq,LB,UB,nonlcon,options)
x= ga(fitnessfcn,nvars,A,b,[],[],LB,UB,nonlcon,IntCon)
x= ga(fitnessfcn,nvars,A,b,[],[],LB,UB,nonlcon,IntCon,options)
x= ga(problem)
[x,fval]= ga(fitnessfcn,nvars,...)
[x,fval,exitflag]= ga(fitnessfcn,nvars,...)
[x,fval,exitflag,output]= ga(fitnessfcn,nvars,...)
[x,fval,exitflag,output,population]= ga(fitnessfcn,nvars,...)
[x,fval,exitflag,output,population,scores]= ga(fitnessfcn,nvars,...)
```

Bu tezdeki problemin çözümünde tercih edilen ise şu şekildedir;

$x = \text{ga}(\text{fitnessfcn}, \text{nvars}, \text{A}, \text{b}, \text{Aeq}, \text{beq}, \text{LB}, \text{UB}, \text{nonlcon}, \text{options})$

Fonksiyon, genetik algoritma optimum set fonksiyonu kullanılarak oluşturulan yapı seçeneklerindeki değerler ile varsayılan optimizasyon parametrelerini değiştirerek, minimizasyon yapar. Daha iyi anlaşılması açısından bağımsız değişkenleri sırasıyla açıklamakta fayda vardır.

### Girdi Değişkenleri

**fitnessfcn:** uygunluk fonksiyonu ele alınacak olursa; fonksiyon değişken sayısını bir satır vektörünün uzunluğu olarak kabul etmeli ve skaler değere döndürmelidir. 'Vektörize' seçeneği açık olduğunda; fitnessfcn, **nvars** (değişken sayısı)\***pop** (mevcut popülasyon boyutu) matrisini kabul etmelidir. Bu durumda fitnessfcn, uygunluk fonksiyonu değerlerini içeren mevcut popülasyon boyutu gibi aynı uzunlukta olan bir vektöre dönmelidir. Uygunluk fonksiyonu, mevcut nüfus büyüklüğü için herhangi bir boyut tahminlememelidir çünkü genetik algoritma, vektörize bir hesaplamada bile nüfusun tek bir üyesini geçebilir.

**nvars:** problemdeki değişkenlerin sayısını temsil eden pozitif tamsayıdır.

**A:** doğrusal eşitsizlik kısıtları formunun matrisidir.

$$A*x \leq b$$

Eğer problem m tane doğrusal eşitsizlik kısıtına ve nvars değişkene sahipse,

- A matrisi m\*nvars boyutundadır.
- b, m uzunluğunda bir vektördür.

Genetik algoritma A\*x matris yapısını değerlendirir. x transpoz ise (A\*x') görünümünde olur.

**b:** doğrusal eşitsizlik kısıtları formunun vektörüdür.

$$A*x \leq b$$

Eğer problem m tane doğrusal eşitsizlik kısıtına ve nvars değişkene sahipse,

- A matrisi m\*nvars boyutundadır.
- b, m uzunluğunda bir vektördür.

**Aeq:** doğrusal eşitlik kısıtları formunun matrisidir.

$$Aeq*x = beq$$

Eğer problem m tane doğrusal eşitlik kısıtına ve nvars değişkene sahipse,

- Aeq matrisi m\*nvars boyutundadır.
- beq, m uzunluğunda bir vektördür.

Genetik algoritma Aeq\*x matris yapısını değerlendirir. x transpoz ise (Aeq\*x') görünümünde olur.

**beq:** doğrusal eşitlik kısıtları formunun vektörüdür.

$$Aeq*x = beq$$

Eğer problem m tane doğrusal eşitlik kısıtına ve nvars değişkene sahipse,

- Aeq matrisi m\*nvars boyutundadır.
- beq, m uzunluğunda bir vektördür.

**LB:** alt sınırlar vektörüdür. Genetik algoritma, iterasyonları alt sınırın üstünde kalmaya zorlar. Set  $LB(i) = -\text{Inf ifx}(i)$  sınırsız alt.

**UB:** üst sınırlar vektörüdür. Genetik algoritma, iterasyonları üst sınırın altında kalmaya zorlar. Set  $UB(i) = \text{Inf ifx}(i)$  sınırsız üst.

**nonlcon:** iki çıktı döndüren fonksiyonun kolu:

$$[c,ceq] = \text{nonlcon}(x)$$

genetik algoritma  $c \leq 0$  and  $ceq = 0$  ve  $ceq$  birden fazla kısıtlama olduğunda satır vektörleri elde etmeye için çalışır. Kullanılmayan çıkışlar, [ ] olarak ayarlanır. Bir dosyaya bir fonksiyon kolu olarak `nonlcon` yazılabilir, örneğin;

```
nonlcon = @constraintfile
```

`constraintfile.m` MATLAB® ağında bir dosyadır. Vektörize kısıtların nasıl kullanılacağına öğrenmek için `Vectorized Constraints`'e göz atılabilir.

**options:** optimizasyon seçeneklerini içeren yapıdır. Bu bölüm 35 alt başlıktan oluşmaktadır. Yalnızca özetlenecek detaylı gösterimine değinilmeyecektir.

1. `CreationFcn`: başlangıç popülasyonunun oluşumunu ele alır.
2. `CrossoverFcn`: çaprazlanan çocukların fonksiyonudur.
3. `CrossoverFraction`: çaprazlamadan sonra seçkin çocukların dâhil olmadığı yeni popülasyondur.
4. `Display`: ekran düzeyini ifade eder.
5. `DistanceMeasureFcn`: bireylerin mesafe ölçüsünü hesaplar.
6. `EliteCount`: şimdiki nesildeki kaç kişinin gelecek nesillerde hayatta kalacağını belirten pozitif tamsayıdır.
7. `FitnessLimit`: skalerdir. Uygunluk fonksiyonu `FitnessLimit` değerini kazanır ise, algoritma durur.
8. `FitnessScalingFcn`: uygunluk fonksiyonu değerlerini ölçekler.
9. `generations`: maksimum yinleme sayısını belirten pozitif tamsayıya ulaşmadan önce algoritma durur.
10. `HybridFcn`: melezleme fonksiyonudur.
11. `InitialPenalty`: ceza parametresinin başlangıç değeridir.
12. `InitialPopulation`: genetik algoritma tohumu kullanılan ilk popülasyondur.
13. `InitialScores`: uygunluk belirlemek için kullanılan ilk puanlardır.
14. `MigrationDirection`: göç yönü gösterir.
15. `MigrationFraction`: bireylerin bölümünü belirten 0 ile 1 arasında skalerdir.
16. `MigrationInterval`: alt popülasyonlar arasındaki bireylerin göç arasında gerçekleşecek nesillerinin sayısını belirten pozitif tamsayıdır.

17. MutationFcn: mutasyonlu çocukları üreten fonksiyonun koludur.
18. NonlinConAlgorithm: doğrusal olmayan kısıt algoritmasıdır.
19. OutputFcns: her adımda genetik algoritma çağırır fonksiyonlardır.
20. ParetoFraction: bireyleri, 0 ile 1 arasında skaler olan ilk Pareto cephesinde tutmak için kullanılır.
21. PenaltyFactor: ceza güncelleme parametresidir.
22. PlotFcns: çizim verileri algoritması tarafından hesaplanan işlevleri gösterir.
23. PlotInterval: çizim fonksiyonları için ardışık aramalar arasında nesillerin sayısını belirten pozitif tamsayıdır.
24. PopInitRange: ilk popülasyondaki bireylerin aralığını belirten matris veya vektördür.
25. PopulationSize: popülasyon boyutudur.
26. PopulationType: nüfusun veri türünü açıklar.
27. SelectionFcn: çaprazlama ve mutasyon geçirecek çocukların ailesini seçen işlev koludur.
28. StallGenLimit: algoritma, en iyi uygunluk fonksiyonu değerinde ortalama bağıl değişim olursa durur.
29. StallTest: durma testini açıklayan dizgedir.
30. StallTimeLimit: amaç fonksiyonunda herhangi bir gelişme varsa algoritma durur.
31. TimeLimit: algoritma zaman limiti olan saniye kadar çalıştıktan sonra durur.
32. TolCon: doğrusal olmayan kısıtlamaların olabilirliğini belirlemek için kullanılır.
33. TolFun: algoritma, en iyi uygunluk fonksiyonu değerinde ortalama bağıl değişim olursa durur.
34. UseParallel: paralelde, uygun ve doğrusal olmayan kısıt fonksiyonlarını hesaplar.
35. Vectorized: fonksiyonların vektörize olup olmadığını belirtir.

**IntCon:** 1'den nvars'a kadar değerler alan pozitif tamsayılar vektörüdür. IntCon'da her bir tamsayı değer olan bir x bileşenini temsil eder.

**problem:** ‘fitnessfcn, nvars, Aineq, Bineq, Aeq, Beq, lb, ub, nonlcon, rngstate, intcon, solver, options’ bölümlerinden oluşur.

### Çıktı Değişkenleri

**x:** genetik algoritmanın tekrarlamaları sırasında bulunan en iyi noktadır.

**fval:** x’de değerlendirilen uygunluk fonksiyonudur.

**exitflag:** bu tamsayı genetik algoritma iterasyonlarının durma sebebini açıklar.

**output:** her nesil ve algoritma performansı hakkında başka bilgilerin çıktısını içeren yapıdır.

**population:** matris olan satırlar son nüfus üyelerini içerir.

**scores:** nihai popülasyonun uygunluk değerlerinin sütun vektörüdür.

## **DÖRDÜNCÜ BÖLÜM**

### **UYGULAMA ÇALIŞMASI**

Dördüncü bölümde ilk üç bölümde aktarılan çizelgeleme, hemşire çizelgeleme ve genetik algoritmalar; gerçek bir örnek yardımıyla açıklanmıştır. Uygulamada kullanılan veriler Buca Seyfi Demirsoy Devlet Hastanesi, Koroner Yoğun Bakım servisinin Şubat 2015 çizelgeleridir. İlk olarak hastanenin tarihçesi ve bölümlerinden bahsedilmiş arkasından, Sağlık Bakanlığı ve Hastane Yönetimince belirlenmiş olan çalışma koşulları aktarılmıştır.

MATLAB'da genetik algoritmalar yardımıyla 14 hemşire için 28 günlük bir çizelgenin oluşturulması, oldukça fazla çözüm değerinin ortaya çıkmasına sebep olmaktadır. Buna ek olarak verilen kısıtların tamamının sağlanıp sağlanmadığını kontrol etmek oldukça güçtür. Güçlük sebebiyle kısıtların matematiksel olarak en iyi şekilde ifade edildiğinden ve kodlandığından emin olmak adına öncelikle küçük bir model için uygulama yapılmıştır. Yine benzer sebeplerle genetik algorithmadan önce bu tip problemlerin çözümünde yaygın olarak kullanılan karma tamsayılı programlamadan yardım alınmış daha sonra gerçek çizelge ve tüm bu değerlerin karşılaştırılması raporlanmıştır.

#### Buca Seyfi Demirsoy Devlet Hastanesi

İzmir'in Buca ilçesinin merkezinde yer alan Buca Seyfi Demirsoy Devlet Hastanesi, 140 dönüm arazi üzerinde hizmet vermektedir. 06.07.2002 tarihinde hizmete açılmış 19.02.2005 tarihinde Sağlık Bakanlığı bünyesine geçmiştir. Yataklı birim ve poliklinik binası başta olmak üzere toplam 8 adet bina, 24.000m2 kapalı alan içerisinde hizmet vermektedir. 406 yatak kapasiteli hastanede, tüm branşlarda poliklinik ve yataklı tedavi hizmeti verilmektedir. Tüm hasta odalarında telefon olup, 40 adet tek kişilik odada banyo, buzdolabı, televizyon, refakatçi koltuğu bulunmaktadır. Bünyesindeki acil, 10 ameliyathane salonu, yerel müdahale salonları, 10 yataklı 2.basamak yoğun bakım ünitesi, 8 yataklı 1. basamak yoğun bakım ünitesi, 9 yataklı koroner yoğun bakım ünitesi bulunmaktadır. Tablo 4'de hastanede mevcut tüm birimler yer almaktadır.

## Bölümler

**Tablo 4:** Hastane Birimleri

<p><u>Tıbbi Birimler</u></p> <ul style="list-style-type: none"><li>• Acil Servis</li><li>• Fizik Tedavi Servisi</li><li>• Kulak Burun Boğaz Hastalıkları Servisi</li><li>• Göz Hastalıkları Servisi</li><li>• Üroloji Servisi</li><li>• Dâhiliye Hastalıkları Servisi</li><li>• Göğüs Hastalıkları Servisi</li><li>• Kardiyoloji Servisi</li><li>• İntaniye Servisi</li><li>• Kalp Damar Cerrahisi Servisi</li><li>• Cildiye Servisi</li><li>• Plastik Cerrahi Servisi</li><li>• Genel Cerrahi Servisi</li><li>• Nöroloji Servisi</li><li>• Beyin Cerrahi Servisi</li><li>• Ortopedi Servisi</li><li>• Palyatif Bakım Servisi</li><li>• Fizik Tedavi Rehabilitasyon Servisi</li><li>• Diyet Hastalıkları</li></ul>	<p><u>Tanı Birimleri</u></p> <ul style="list-style-type: none"><li>• Biyokimya</li><li>• Mikrobiyoloji</li><li>• Patoloji</li><li>• Radyoloji</li><li>• Eeg</li><li>• Emg</li><li>• Ekg, Eforlu Ekg</li><li>• Endoskopi</li><li>• Odiyometri</li><li>• Holter</li><li>• Eko</li><li>• Tomografi</li></ul>
<p><u>Görüntüleme Birimleri</u></p> <ul style="list-style-type: none"><li>• Mr</li><li>• Tomografi</li><li>• Kemik Dansitometri (Yoğunluk Ölçümü)</li><li>• Doppler-Usg</li><li>• Mamografi</li></ul>	<p><u>Yoğun Bakımlar</u></p> <ul style="list-style-type: none"><li>• Koroner Yoğun Bakım</li><li>• 3.Basamak Yoğun Bakım</li><li>• Dâhiliye Yoğun Bakım</li></ul>

Kaynak: Sağlık Bakanlığı.



## 4.1. HASTANEDEKİ MEVCUT ÇALIŞMA KOŞULLARI

Buca Seyfi Demirsoy Devlet Hastanesi 7 gün 24 saat açık olan bir hastanedir. İlgilenmekte olduğumuz bölüm Koroner Yoğun Bakım Servisindeki hemşireler 08:00-16:00 (sabah vardiyası) ve 16:00-08:00 (gece nöbeti) şeklinde çalışmaktadır. Nöbet 16:00-24:00 şeklinde parçalanmamaktadır. Bunun sebebi 16 saatlik gece nöbetinden sonra hemşire taşıma hizmetinin yapılmaması ve hemşirelerin gece saat 12'de başka araçlarla evlerine gidip gelmekten endişe etmeleridir. Hemşireler haftada en az 40 dolayısıyla ayda 160 saat çalışmak durumundadır. Hastanede çalışmakta olan hemşirelerin resmi çalışma koşulları, Hemşirelik Yönetmeliği temel alınarak oluşturulmuştur. Buna ek olarak hastane yönetiminin hemşirelerin en uygun şekilde hizmet vermeleri ve hastaların gereken zamanda, gerektiği gibi tedavi görebilmeleri için belirledikleri bir takım kurallar vardır. Aşağıda bu kurallar gerekçeleri ile birlikte anlatılmaya çalışılacaktır.

### 4.1.1. Kısıtlar

Oluşturulmaya çalışılan zaman çizelgesinin hastanelerin tüm servisleri için eldeki kısıtları sağlayacak en uygun şekilde atanması hedeflenmektedir. Kısıtlar zorunlu (sabit katı) ve esnek kısıtlar olarak ikiye ayrılabilir. Zorunlu kısıtlar esnekliğin olmadığı, kısıtlardan birinin sağlanamaması durumunda problemin çözülemeyeceği kısıtlardır. Esnek kısıtlar; kanun ve yönetmeliğe bağlı olmayan elden geldiğince yerine getirilmeye çabalanan kısıtlardır.

#### Zorunlu (sabit katı) kısıtlar

1. Hemşireler gündüz ve akşam olmak üzere iki vardiya şeklinde çalışmaktadır.
2. Gündüz vardiyasında çalışan gece nöbet tutamaz. Gece nöbetçisi gündüz çalışamaz. Bir hemşire bir gün içinde yalnızca bir vardiyada çalışılabilir. Aksi halde hemşirenin çalışma koşulları ağırlaşacak ve performansı düşecektir.

3. Denetleme ve kontrol işini yapan sorumlu hemşire (supervisor) gece nöbeti tutmaz. Sorumlu hemşire zaten ciddi bir iş yükü altındadır, gece nöbeti bu nedenle ona verilmez ve denetleme görevi aksatılmaz.
4. Meslekte 30 yılını doldurmuş, 50 yaşın üstünde olan, sağlık sorunu olan ya da gebe olan hemşireler nöbet tutmaz. Doğum sonu 2 yıl nöbet muafiyeti olan 1 hemşire vardır. 50 yaşın üzerinde olan hemşireler yaşları gereği daha çabuk yorulacak ve gece nöbetinde gereken verimliliği gösteremeyeceklerdir. Doğum yapan hemşireler ise benzer sağlık zafiyetleri nedeniyle nöbet tutmadan muaf edilmişlerdir.
5. Akşam nöbetinde iki kişi çalışır.
6. Her hemşire nöbetçi olmadığı gün 8 saat çalışır. Dolayısıyla haftada 40 ayda 160 saat çalışma zorunluluğu vardır. Bu kısıt hemşirenin zorunlu olarak doldurmak durumunda olduğu mesai saatlerini ifade eder.
7. Akşam nöbeti tutanlara ertesi gün görev verilmez. Akşam 16 saatlik nöbetten çıkan hemşire yeteri kadar yorulmuş ve uykusuz kalmıştır. Sağlık sektörü bu sebeplerle yapılacak bir anlık dikkatsizliklerinin tolere edilemeyeceği riskli bir sektördür. Bu nedenle hemşire ertesi sabah çalışmaya devam etmez.
8. Hemşire aynı gün içinde yalnızca bir vardiyada çalışabilir ya da izinlidir. 2.kısıta benzer olan bu kısıt hemşire o gün içerisinde çalışmadıysa izinli olduğunu vurgular.
9. Akşam vardiyasında nöbet tutan iki kişiden birinin uzman olmasına çalışılır. Mesleğe yeni başlamış hemşirelerin olması ve gece vardiyasında desteğe ihtiyaç duyabilecek olmaları sebebiyle kıdem olarak daha yüksek ve deneyimli biri ile nöbet tutmalarına çalışılmaktadır.

#### Esnek Kısıtlar

1. Nöbetçi personel nöbeti teslim alacak personel gelmeden ayrılamaz.
2. Nöbetler sağlık bakım hizmetlerinin onayı olmadan değiştirilemez. Elle yapılan çizelgelerde kimi zaman hemşireler kendi aralarında anlaşıp günlerini değiştirmeyi talep etmektedir. Değiştirilmesinde sakınca olmasa da bu işlem ancak sağlık bakım hizmetlerinin onayı ile yapılabilmektedir.

#### 4.1.2. Kısıtların Matematiksel Gösterimi

İlk olarak problemin kısıtlarının, programa tanıtılabilmek için matematiksel ifadelere çevrilmesi gerekmektedir. Her bir kısıta karşılık gelen matematiksel gösterim aşağıdaki tabloda gösterilmiştir.

**Tablo 5:** Kısıtların Matematiksel Gösterimi

Kısıtın Açıklaması	Formülasyonu	Matrisi	Modeldeki Satır Sayısı
1) Hemşireler gündüz ve akşam olmak üzere iki vardiya şeklinde çalışmaktadır. Bu kısıt modellemede değişkenlerin belirlenmesi için kullanılmıştır.	$XM_{i,d} = 1$ , Eğer i. hemşire d. gündeki sabah vardiyasına atanmışsa $\forall (i=1,\dots,n; d=1,\dots,m)$ 0, diğer durumlarda $XE_{i,d} = 1$ , Eğer i. hemşire d. gündeki akşam vardiyasına atanmışsa $\forall (i=1,\dots,n; d=1,\dots,m)$ 0, diğer durumlarda $XD_{i,d} = 1$ , Eğer i. hemşire d. günde izinli ise $\forall (i=1,\dots,n; d=1,\dots,m)$ 0, diğer durumlarda	A	1738x784
2) Gündüz vardiyasında çalışan gece nöbet tutamaz. Gece nöbetçisi gündüz çalışamaz. Bir gün içinde yalnızca bir vardiyada çalışılabilir.	$XM_{i,d} + XE_{i,d} \leq 1$ $\forall (i=1,\dots,n; d=1,\dots,m)$	A2	392x784

3) Denetleme ve kontrol işini yapan sorumlu hemşire (supervisor) gece nöbeti tutmaz.	$XM_{13,d} = 1$ $\forall (i=1,\dots,n ; d=1,\dots,m)$	Aeq3	56x784
4) Meslekte 30 yılını doldurmuş, 50 yaşın üstünde olan, sağlık sorunu olan ya da gebe olan hemşireler nöbet tutmaz. Doğum sonu 2 yıl nöbet muhafiyeti olan 1 hemşire vardır.	$XM_{14,d} = 1$ $\forall (i=1,\dots,n ; d=1,\dots,m)$	Aeq4	56x784
5) Akşam nöbetinde iki kişi çalışır.	$\sum_{i=1}^{12} XE_{i,d} = 2$ $\forall (i=1,\dots,n ; d=1,\dots,m)$	Aeq5	56x784
6) Her hemşire nöbetçi olmadığı gün 8 saat çalışır. Dolayısıyla haftada 40 ayda 160 saat çalışma zorunluluğu vardır.	$8 \sum_{i=1}^{14} XM_{i,d} + 16 \sum_{i=1}^{14} XE_{i,d} \gg 160$	A6	14x784
7) Akşam nöbeti tutanlara ertesi gün görev verilmez.	$XM_{i,d} + XE_{i,d} \leq 1$ $\forall (i=1,\dots,n ; d=1,\dots,m)$	A7	378x784
8) Hemşire aynı gün içinde yalnızca bir vardiyada çalışabilir ya da izinlidir.	$XM_{i,d} + XE_{i,d} + XO_{i,d} = 1$ $\forall (i=1,\dots,n ; d=1,\dots,m)$	Aeq8	784x784

9) Akşam vardiyasında nöbet tutan iki kişiden birinin uzman olmasına çalışılır.	$\sum_{i=1}^6 XE_{i,d} \geq 1 \quad (d=1,\dots,m)$	A9	28x784
---	--	----	--------

### Amaç Fonksiyonu

Bu kısıtların sağlanması ile ulaşılmak istenen amaç çalışma sürelerinin minimizasyonu ve hemşireler için en iyi çalışma çizelgelerinin oluşturulmasıdır.

$$f_{min} = \sum (8 \sum XM_{i,d} + 16 \sum XE_{i,d})$$

### **4.1.3. Hastanenin Mevcut Çizelgeleri**

Buca Seyfi Demirsoy Devlet Hastanesi'nin hemşire çizelgeleri, aylık olarak her birimin sorumlu hemşiresi tarafından yapılmaktadır. Çizelgeler hiçbir program kullanmadan el yordamı ile yapılmaktadır. İşlem birim sorumlusu hemşirenin 6-8 saat kadar zamanını almaktadır. Gece nöbeti tutmayan her hemşire gündüz görev başındadır. Koroner yoğun bakım servisinde 14 hemşire bulunmaktadır. Aşağıdaki tablolarda Şubat ayında çalışan hemşirelerin toplam vardiya sayıları, her bir hemşirenin toplam gece nöbeti sayıları ve izin günleri incelenmiştir. Her gece vardiyasında en az 1 uzman hemşire olmasına ve hamile veya süt izninde olan hemşirelerin vardiya atamalarının sadece gündüz vardiyası şeklinde olmasına dikkat edilmiştir. Modelde çizelge uzunluğu 28 gündür.

Hemşirelerden 4 tanesi verilen tarihler aralığında yıllık izin haklarını kullanmıştır. Tablo 6’da yıllık izin kullanan hemşireler bulunmaktadır. Tablo 7’de gece göbet listesi verilmiştir.

**Tablo 6:** Koroner Yoğun Bakım Yıllık İzin

13	19.02.2015	19.02.2015	1gün	20.02.2015
2	24.02.2015	25.02.2015	2 gün	26.02.2015
11	27.02.2015	27.02.2015	1 gün	28.02.2015
9	02.02.2015	13.02.2015	12 gün	14.02.2015

**Tablo 7:** Koroner Yoğun Bakım Gece Nöbeti Listesi

Günler	Hemşireler	
1	1	11
2	2	12
3	3	8
4	4	11
5	1	8
6	5	12
7	2	7
8	6	10
9	3	12
10	6	10
11	7	11
12	5	6
13	2	10
14	7	1
15	8	10
16	9	6
17	4	10
18	9	6
19	2	5
20	3	6
21	10	12
22	4	9
23	8	10
24	7	6
25	11	5
26	9	1
27	3	10
28	4	6

Daha sonra çalışma günleri, programa tanıtımı kolaylaştırmak için matrisler ile ifade edilmiştir. Burada çalışılan günler 1 değerini, nöbet çıkışı olan ya da senelik izin olan günler 0 değerini almıştır. Bulunan birlerin toplamıyla, gündüz vardiyası için 8 ile gece nöbeti için 16 ile çarpılarak hastanenin mevcut amaç fonksiyonu değeri de elde edilmiştir.

- gündüz için;  $267 \cdot 8 = 2136$
- gece için;  $56 \cdot 8 = 896$
- amaç fonksiyonu değeri:  $2136 + 896 = 3032$

**Tablo 8:** Koroner Yoğun Bakım Matris Gösterimi

sabah	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
1	0	0	1	1	0	0	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	
2	1	0	0	1	1	1	0	0	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1	0	0	1	1	1	
3	1	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	0	0	
4	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	0	
5	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1	
6	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0
7	1	1	1	1	1	1	0	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	0	0	1	1	1
8	1	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	0	1	1	1	1
9	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	1	1	0	0	1
10	1	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0
11	0	0	1	0	0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1
12	1	0	0	1	1	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
gece	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
2	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1
5	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	0	1	0	0	0	1
7	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
8	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0
11	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
12	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Önceki bölümde MATLAB programında genetik algoritmaların nasıl kullanıldığı açıklanmıştı. Model çözümlerine geçmeden önce karma tamsayılı programlamanın MATLAB'daki kullanımını da kısaca özetlenmiştir.

#### 4.1.4. Karma Tamsayılı Doğrusal Programlama (MILP)

Karma tamsayı doğrusal programlama çözücü, belirtilen problemin minimumu aşağıdakiler aracılığı ile bulur:

$$\min_x f^T x \quad \left\{ \begin{array}{l} x \text{ (intcon) tamsayılı} \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub. \end{array} \right.$$

$f$ ,  $x$ ,  $intcon$ ,  $b$ ,  $beq$ ,  $lb$  ve  $ub$  vektörler ve  $A$  ve  $Aeq$  matrislerdir.

$f$ ,  $intcon$ ,  $lb$  ve  $ub$  vektör ya da dizi olarak da ifade edilebilir.

MATLAB programında karma tamsayılı doğrusal programlama ile problemi çözmek için pek çok sözdizimi mevcuttur. Bunlardan birinin seçimi ihtiyaçlar doğrultusunda yapılmaktadır. Aşağıda bu dizimlere yer verilmiştir.

```
x = intlinprog(f,intcon,A,b)
```

```
x = intlinprog(f,intcon,A,b,Aeq,beq)
```

```
x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub)
```

```
x = intlinprog(f,intcon,A,b,Aeq,beq,lb,ub,options)
```

```
x = intlinprog(problem) [x,fval,exitflag,output] = intlinprog ( _ )
```

Bu dizimlerle çözülen örnekler MathWorks sayfasında detaylı şekilde anlatılmıştır.



### Girdi Değişkenleri

- $f$ : katsayı vektörüdür. Notasyon,  $f$ 'in sütun vektörü olduğunu varsayar ancak bir satır vektörü kullanılması da serbesttir.  $f$  aynı zamanda bir dizi de olabilir.
- $intcon$ : tamsayılı kısıtlar vektörüdür. Pozitif tamsayılar vektörü olarak belirtilmiştir.  $intcon$ 'daki değerler, tamsayı değerli olan karar değişkeni  $x$  bileşenlerini göstermektedir.
- $A$ : Lineer eşitsizlik kısıtı matrisidir.  $A$  kısıtların doğrusal katsayılarını temsil eder.
- $b$ : Lineer eşitsizlik kısıtı vektörüdür.  $b$  kısıtların sabit vektörünü temsil eder.
- $Aeq$ : Doğrusal eşitlik kısıtı matrisidir.  $Aeq$  kısıtlamaları doğrusal katsayılarını temsil eder.
- $beq$ : Lineer eşitlik kısıtı vektörüdür.  $beq$  kısıtların sabit vektörünü temsil eder.
- $lb$ : alt sınırdır.  $lb$  bilge alt sınır elemanını temsil eder.
- $ub$ : üst sınırdır.  $ub$  bilge üst sınır elemanını temsil eder.

### Çıktı Değişkenleri

- $x$ : çözümdür.  $f$  için tüm sınırlara tabi, tamsayı kısıtlamaları ve lineer kısıtlamaları en aza indiren bir vektördür.
- Problemin çözümü mümkün olmadığında ya da sınırsız olduğunda;  $x$ , [ ] olur.
- $fval$ : amaç fonksiyonu değeridir. Problemin çözümü mümkün olmadığında ya da sınırsız olduğunda;  $fval$ , [ ] olur.

- exitflag: algoritmayı durdurma koşuludur. Aşağıda exitflag değerleri ve intlinprog'un durma sebebine karşılık gelen listeler vardır.

2	intlinprog erken durmuştur. Uygun tamsayı nokta bulunmuştur.
1	intlinprog x çözümüne yaklaşmıştır.
0	intlinprog erken durmuştur. Hiçbir uygun tamsayı noktası bulunamamıştır.
-1	intlinprog bir çıkış ya da çizim fonksiyonu tarafından durdurulmuştur
-2	Hiçbir uygun nokta bulunamamıştır.
-3	Kök LP problemin sınırsız olduğunu gösterir.

## 4.2. MODELLEME VE KODLAMA

Uygulama aşamasında, öncelikle tüm kısıtlar ve amaç fonksiyonu matris gösterimi haline getirilerek çözüm için girdi olacak şekilde hazırlanmıştır. Böylece Tablo 5'te adı ve boyutları verilen matrisler elde edilmiştir. Daha sonra bu matrisler kullanılarak optimum çözümlerin bulunabilmesi için karma tamsayılı programlama ile çözümleri yapılmıştır. Genetik algoritmalar yardımı ile optimum çözüme ulaşıp ulaşılamayacağını incelemek için GA çözücüsü ile de çözümler yapılmış ve sonuçları yorumlanmıştır.

### 4.2.1. Kodlama

Uygulamada kullanılan kodlar kısıtların karşılıkları olarak aşağıda verilmiştir.

Amaç Fonksiyonu: Çalışma sürelerinin en aza indirilmesi.

```
p=n*m;
costt=0;
work=0;
s=0;

for t=1:p
    costt(1,t)=8;
    costt(1,p+t)=16;
```

```
end
cost=sum(x.*costt);
```

Kısıt 2: Gündüz vardiyasında çalışan gece nöbet tutamaz. Gece nöbetçisi gündüz çalışamaz.

```
c=1;
for t=1:p
    A2(c,t)=1;
    A2(c,t+p)=1;
    b2(c,1)=1;
    c=c+1;
end
```

Kısıt 3: Denetleme ve kontrol işini yapan sorumlu hemşire (supervisor) gece nöbeti tutmaz.

```
d=1;
for t=2*p-m+1:2*p
    Aeq3(d,t)=1;
    beq3(d,1)=0;
    d=d+1;
end
Aeq3=[Aeq3;-Aeq3];
beq3=[beq3;-beq3];
```

Kısıt 4: Meslekte 30 yılını doldurmuş, 50 yaşın üstünde olan, sağlık sorunu olan ya da gebe olan hemşireler nöbet tutmaz.

```
d=1;
if h<1
    Aeq4=[];
    beq4=[];
else
    for t=2*p-m+1-m:2*p-m
        Aeq4(d,t)=1;
        beq4(d,1)=0;
        d=d+1;
    end
end
Aeq4=[Aeq4;-Aeq4];
beq4=[beq4;-beq4];
```

Kısıt 5: Akşam nöbetinde iki kişi çalışır.

```
c=1;
say=0;
for t=1:m
    say=0;
    for g=1:n
        Aeq5(c,p+m*say+t)=1;
        beq5(c,1)=2;
        say=say+1;
    end
    c=c+1;
end
Aeq5=[Aeq5;-Aeq5];
beq5=[beq5;-beq5];
```

Kısıt 6: Her hemşire nöbetçi olmadığı gün 8 saat çalışır. Dolayısıyla haftada 40 ayda 160 saat çalışma zorunluluğu vardır.

```
if n==5
    saat=40;
else saat=160;
end
c=1;
e=0;
for t=1:n
    for i=1:m
        A6(c,t+e)=-8;
        A6(c,p+t+e)=-16;
        e=e+1;
    end
    e=e-1;
    b6(c,1)=-saat;
    c=c+1;
end
```

Kısıt 7: Akşam nöbeti tutanlara ertesi gün görev verilmez.

```
c=1;
say=1;
for t=1:n
    say=1;
```

```

for g=1:m-1
    A7(c,(t-1)*m+say+1)=1;
    A7(c,(t-1)*m+p+say)=1;
    b7(c,1)=1;
    c=c+1;
    say=say+1;
end
end

```

Kısıt 8: Hemşire aynı gün içinde yalnızca bir vardiyada çalışabilir ya da izinlidir.

```

d=1;
for t=1:p
    Aeq8(d,t)=1;
    Aeq8(d,t+p)=1;
    beq8(d,1)=1-XO(t);
    d=d+1;
end
Aeq8=[Aeq8;-Aeq8];
beq8=[beq8;-beq8];

```

Kısıt 9: Akşam vardiyasında nöbet tutan iki kişiden birinin uzman olmasına çalışılır.

```

d=1;
for t=p+1:p+m
    for k=1:u
        A9(d,t+k*m-m)=-1;
    end
    b9(d,1)=-1;
    d=d+1;
end

```

#### 4.2.2. Modellerin Çözümlemesi

Oluşturduğumuz küçük model ve gerçek verilerden yararlandığımız büyük modelin, parametreleri ve çözüm değerleri Tablo 9'da yer almaktadır. ga\_big.m ve ga\_small.m adlı iki kod dosyası yardımıyla ana program hazırlanmıştır. Çözümler için ihtiyaç duyulan formatta amaç fonksiyonunu oluşturan kod dosyası objectvec.m dir. makelincon.m dosyası ise kısıtların matris haline getirilmesi için yazılmıştır. Bu

dört kod dosyası da eklerde verilmektedir. Karma tamsayı programlama ve GA için aşağıdaki kodlar kullanılmıştır.

```
[X,FVAL,EXITFLAG] = intlinprog(costt,intcon,A,b,[],[],lb,ub);
```

```
[x_ga,fval,exitflag]=  
ga(f,numberOfVariables,A,b,[],[],lb,ub,[],intcon,options)
```



**Tablo 9:** Modellerin Karşılaştırması

Modellerin Karşılaştırması	Küçük Model	Büyük Model
<u>Parametreler</u> n: hemşireler m:günler k:vardiya (2) u:uzman h: gece nöbetine kalmayacaklar	n: 5 m:7 k: 2 u: 3	n: 14 m:28 k: 2 u: 9
Karma Tamsayılı Programlamanın Çözümü	En Uygun Amaç Fonksiyonu Değeri: 392	En Uygun Amaç Fonksiyonu Değeri: 3456
Genetik Algoritmanın Çözümü	<u>10 Deneme için En Uygun Amaç Fonksiyonu Değerleri</u> min: 320 max: 400 1. 376 2. 400 3. 352 <b>4. 376</b> 5. 376 6. 344 7. 344 8. 320 9. 352 10. 384	<u>10 Deneme için En Uygun Amaç Fonksiyonu Değerleri</u> min: 3424 max: 3688 1. 3480 2. 3464 3. 3496 4. 3488 5. 3424 6. 3640 7. 3688 8. 3488 9. 3512 10. 3592

İlk olarak karma tamsayılı programlamadan başlanarak küçük ve büyük modeller için çözüm matrisleri sunulacaktır. 5 hemşire ve 1 haftalık veriler yardımıyla elde edilen çözüm aşağıdaki gibidir. Tüm bu değerler programın x\_int sayfasında çözüm vektörü olarak görüntülenmektedir. Amaç fonksiyonu değerleri ise fval yardımı ile elde edilir.

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	2	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1	3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	5	0	0	0	0	0	0	0

Mevcut çözüm bütün kısıtları sağlamaktadır ve tabloda görüldüğü gibi amaç fonksiyonu değeri 392'dir.



**Tablo 10:** Karma Tamsayılı Programlama Büyük Model Çözüm Matrisi

gündüz	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
2	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
3	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
4	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
5	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
6	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
7	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
13	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
gece	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Daha sonra genetik algoritma çözümleri yapılmıştır. Ancak genetik algoritma çözücüsü ile optimum çözüme ulaşamadığı için 10 deneme yapılmış ve bulunan değerler raporlanmıştır. Büyük model için tüm matrislerin aktarılması mümkün olmadığından yalnızca çözüm değerleri Tablo 9’da verilmiştir.

ga1: amaç fonksiyon değeri=376

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	2	1	1	0	1	1	1	1
3	1	1	0	0	0	1	0	3	0	0	1	1	0	0	1
4	1	1	0	1	0	0	0	4	0	0	1	0	1	1	1
5	1	1	1	1	1	1	1	5	0	0	0	0	0	0	0

ga2: amaç fonksiyon değeri=400

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	0	0	0	0	1	1	0	1	1	1	1	0	0	0	1
2	1	1	0	0	0	0	0	2	0	0	1	1	1	1	1
3	1	1	1	1	1	1	1	3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	5	0	0	0	0	0	0	0

ga3: amaç fonksiyon değeri=352

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	1	1	0	0	0	0	1	1	0	0	1	1	1	1	0
2	0	0	0	1	1	1	1	2	1	1	0	0	0	0	0
3	1	0	1	1	1	1	0	3	0	0	0	0	0	0	1
4	1	1	0	0	1	0	0	4	0	0	1	1	0	1	1
5	1	1	1	1	1	1	1	5	0	0	0	0	0	0	0

ga4: amaç fonksiyon değeri=376

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	2	1	1	1	0	0	1	1
3	1	1	1	0	0	0	1	3	0	0	0	1	1	0	0
4	0	0	0	0	0	0	0	4	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	5	0	0	0	0	0	0	0

ga5: amaç fonksiyon değeri=376

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	2	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	3	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	4	0	1	1	1	1	1	0
5	1	1	1	1	1	1	1	5	0	0	0	0	0	0	0

ga6: amaç fonksiyon değeri= 344

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
2	0	1	1	1	1	0	0	0	0	0	0	0	1	1	0
3	1	1	0	1	0	0	1	0	0	1	0	1	0	0	0
4	1	1	1	1	0	0	0	0	0	0	0	1	1	1	0
5	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

ga7: amaç fonksiyon değeri=344

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	1	1	1	0	0	0	0	0	0	0	1	1	0	1	0
2	1	1	0	1	1	0	0	0	0	0	0	0	1	0	0
3	0	0	0	1	1	0	0	1	1	1	0	0	1	1	0
4	1	1	0	0	0	1	0	0	0	1	1	0	0	1	0
5	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

ga8: amaç fonksiyon değeri=320

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	1	1	1	0	0	1	1	0	0	0	1	0	0	0	0
2	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
3	0	1	1	1	0	0	0	0	0	0	0	1	1	1	0
4	1	0	0	0	1	1	1	0	1	1	0	0	0	0	0
5	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

ga9: amaç fonksiyon değeri=352

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
4	1	1	0	1	1	1	1	0	0	1	0	0	0	0	0
5	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

ga10: amaç fonksiyon değeri= 384

gündüz	1	2	3	4	5	6	7	gece	1	2	3	4	5	6	7
1	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1
3	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

### 4.2.3. Bulgular

GA çözücüsü küçük modelde de büyük modelde de optimum çözüme ulaşamamıştır. Bazı denemelerde başlangıç çözümü için bile uygun bir başlangıç vektörü bulunamadığı uyarısı alınmıştır. Bunun nedeni, çizelgeleme probleminin boyutunun küçük olduğu durumda bile çok sayıda değişken ve kısıt olmasından dolayı çözüm uzayında global aramanın başarısız olmasıdır. Genetik algoritmalarda tüm kısıtların sağlanamamasından yana bir sıkıntıya düşülmüş, algoritma tekrarlı bir şekilde 5. ve 7. kısıtlarda engele takılmıştır. Küçük modelde yalnızca ga4 için tüm kısıtlar sağlanmıştır. Bu kısıtlar gevşetilebildiği takdirde karma tamsayılı programlamanın verdiği optimum çözümden daha küçük çözüm değerleri elde edilebilecektir.

## SONUÇ VE ÖNERİLER

Bu çalışmada hastanelerde 7 gün 24 saat sağlık hizmeti veren hemşirelerin çalışma çizelgelemesi konusu araştırılarak, Buca Seyfi Demirsoy Devlet Hastanesinde Koroner Yoğun Bakım biriminde çalışan hemşireler üzerinde bir uygulama yapılmıştır. Yöntem olarak genetik algoritma benimsenmiş, program olarak MATLAB kullanılmıştır.

Modelin kısıtlarının doğrusal olması sebebi ile genetik algoritmaya geçmeden önce karma tamsayılı programlama ile çözümü yapılmıştır. Karma tamsayılı programlama, tüm kısıtları sağlayan kesin bir sonuç vermiştir fakat genetik algoritma çalıştırıldıktan sonra elde edilen amaç fonksiyonu değerlerinin daha düşük olduğu ve en küçükmeye daha çok yaklaştığı görülmektedir. Program çaprazlama ve mutasyon değerlerini otomatik olarak belirleyip dışarıdan müdahaleye izin vermediği için popülasyondaki bireylerin bir süre sonra tamamen birbirine benzediği gözlemlenmiştir. Bir diğer dezavantaj MATLAB'da GA çözümü yapan kodların karma tamsayılı çözüm istendiğinde eşitlik kısıtlarını kabul etmemesidir. Bu sebeple her bir eşitlik kısıtı bir küçük, bir büyük eşit kısıtı olarak ifade edilip modele eklenmektedir. Bu da modeldeki kısıt sayısını neredeyse iki katına çıkarttığından, algoritmanın doğru başlangıç noktasını bularak çözüm değerlerine ulaşmasını güçleştirmektedir. Bu yüzden modelde çok fazla eşitlik kısıtı olduğu durumlarda genetik algoritma kullanılmaması önerilmektedir.

Hastanenin çözüm değeri oldukça düşüktür bunun sebebi çizelge elle yapıldığı için verdikleri kısıtlar noktasında katı davranmamış olmalarıdır.

Üzerinde deney yapılması daha kolay olan 10 küçük model çözümünün sonuçlarının hemen hemen hepsinin karma tamsayılı programlamada bulduğumuz çözümden daha düşük amaç fonksiyonu fonksiyonu değerleri verdiği görülmektedir. Fakat bunlardan bazıları bazı kısıtların esnetilmesi koşulunda bu sonuçları vermektedir. Eğer hastane maliyetini düşürmek adına bu kısıtları esnetmeye karar verirse onlara bu durumda yol gösterilebilmesi mümkündür.

Çizelgeleme problemlerinde, insan faktörü olmasından sebebiyle hesaplanamayacak çoklukta parametre görünse de yeterli sayıda matematiksel kısıt yardımıyla gerçek modele çok yakın bir sonuç geliştirilebilecektir.

Özetle, büyük boyutlu ve doğrusal olmayan kısıtların da olduğu çizelgeleme problemlerinde karma tamsayılı programlama da optimum sonucu bulmakta zorlanacaktır. GA kullanımı da standart çözücülerin kullanıldığı durumda etkin olmayacaktır. Farklı sezgisel algoritmaların da kullanılacağı hibrit yaklaşımlar daha etkin çözümler verebilir.

Günümüzde iş dünyasında en pahalı ve önemli kısıt aslına bakılırsa zamandır. Zaman geri getirilemez gibi görünse de bu tarz işlevsel çözümler kazanılabilir ve elle yapılan bir çalışma için harcanan emek başka alanlara kaydırılabilir.



## KAYNAKÇA

Aickelin, U. ve Dowsland, K. (2000). *Exploiting Problem Structure in a Genetic Algorithm Approach to a Nurse Rostering Problem*. 3(3): 139-153. <https://arxiv.org/ftp/arxiv/papers/0802/0802.2001.pdf>. (05.05.2016).

Aickelin, U. ve Dowsland, K.A. (2004). An Indirect Genetic Algorithm for a Nurse-Scheduling Problem. *Computers and Operations Research*. 31(5): 761-778.

Aladağ, A. (2010). *Tekrar İşlemeli Esnek Atölye Tipi Çizelgeleme Problemi İçin Yapay Bağışıklık Sistemi ile Bir Çözüm Yaklaşımı*. (Yayınlanmış Yüksek Lisans Tezi). Eskişehir: Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü.

Altındaş, M. (2011). *Üretim Çizelgeleme ve Bir Uygulama*. (Yayınlanmış Yüksek Lisans Tezi). Yozgat: Bozok Üniversitesi Sosyal Bilimler Enstitüsü.

An indexed bibliography of genetic algorithms: Years 1957-1993. *Art of CAD*, 1994.

Atmaca, E, Pehlivan, C, Aydoğdu, C.B. ve Yakıcı, M. (2014). Hemşire Çizelgeleme Problemi ve Uygulaması. *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Dergisi*. 28(4): 351-353.

Awadallah, M, Khader, A.T, Al-Betar, M.A. ve Bolaji, A.L.A. (2011). Nurse Scheduling Using Harmony Search. *2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications*. 58-63.

Aydemir, E. (2009). *Atölye Tipi Çizelgeleme Problemlerinin Öncelik Kuralı Tabanlı Genetik Algoritma Yaklaşımıyla Simülasyon Destekli Optimizasyonu*. (Yayınlanmış Yüksek Lisans Tezi). Isparta: Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü.

Azaiez, M.N. ve Al Sharif, S.S. (2005). A 0-1 Goal Programming Model for Nurse Scheduling. *Computers and Operations Research*. 32(3): 491-507.

Badami, V.S. ve Parks, C.M. (1991). A Classifier Based Approach to Flow Shop Scheduling. *Computers and Industrial Engineering*. 21(1): 329-333.

Baker, K. (1974). Scheduling a Full Time Workforce to Meet Cyclic Staffing Requirements. *Management Science*. 20 (12): 1561-1569.

Baker, K.R. ve Trietsch, D. (2009). *Principles of Sequencing and Scheduling*. New Jersey: Wiley.

Bakır, M.A, Altunkaynak, B. (2003). *Tamsayılı Programlama: Teori, Modeller ve Algoritmalar*. Ankara: Nobel Yayınevi.

Baskak, M. ve Erol, V. (2004). *Sipariş Tipi Atölyelerde İş Sıralama Problemi için Bir Genetik Algoritma Uygulaması*.

Baskaran, G, Bargiela, A. ve Qu, R. (2013). A Study Of Cost Effective Scheduling Of Nurses Based On The Domain Transformation Method. 309-314.

Belien, J. ve Demeulemeester, E. (2005). *Integrating Nurse and Surgery Scheduling*.

Biegel, J.E. ve Davern, J.J. (1990). Genetic algorithms and job shop scheduling. *Computers and Industrial Engineering*. 19(1): 81-91.

Birođlu, S, Güvenç, U. (2007). Genetik Algoritma ile Çözümü Gerçekleştirilen Atölye Çizelgeleme Probleminde Ürün Sayısının Etkisi. *Akademik Bilişim*.

Birođlu, S. (2005). *Genetik Algoritma Yaklaşımıyla Atölye Çizelgeleme*. (Yayınlanmamış Yüksek Lisans Tezi). Ankara: Gazi Üniversitesi Fen Bilimleri Enstitüsü.

Birođlu, S. (2005). *Genetik Algoritma Yaklaşımıyla Atölye Çizelgeleme*. (Yayınlanmış Yüksek Lisans Tezi). Ankara: Gazi Üniversitesi Fen Bilimleri Enstitüsü.



Bouarab, H, Champalle, S. ve Dagenais, M. (2010). Nurse Scheduling: From Theoretical Modeling to Practical Resolution.

Brownell, W.S. ve Lowerre, J.M. (1976). Scheduling of Workforces Required in Continuous Operations under Alternate Labour Policies. *Management Science*. 22(5): 597-605.

Brucker, P, Burke, E.K, Curtois, T, Qu, R. ve Berghe, G.V. (2010a). Adaptive Construction of Nurse Schedules: A Shift Sequence Based Approach. *Journal of Heuristics*. 16(4): 559-573.

Brucker, P, Burke, E.K, Curtois, T, Qu, R. ve Berghe, G.V. (2010b). A Shift Sequence Based Approach for Nurse Scheduling and a New Benchmark Dataset. *Journal of Heuristics*. 16(4): 559-573.

Brunner, J.O, Bard, J.F, ve Kolisch, R. (2009). Flexible Shift Scheduling of Physicians. *Health Care Management Science*. 12(3): 285-305.

Burke, E, Causmaecker, P, Berghe, G. ve Landeghem, H. (2004). The State of the Art of Nurse Rostering. *Journal of Scheduling*. 7(6): 441-499.

Burke, E, Cowling, P, De Causmaecker, P. ve Berghe, G.V. (2001). A Memetic Approach to the Nurse Rostering Problem. *Applied intelligence*. 15(3): 199-214.

Burke, E.K, Kendall, G. ve Soubeiga, E. (2003). A Tabu-Search Hyperheuristic for Timetabling and Rostering. *Journal of Heuristics*. 9(6), 451-470.

Burke, E.K, Li, J. ve Qu, R. (2012). A Pareto-Based Search Methodology for Multi-Objective Nurse Scheduling. *Annals of Operations Research*. 196(1): 91-109.

Burns, R.N. (1978). Manpower Scheduling with Variable Demands and Alternate Weekends off. 16(2): 101-112.

Cai, X. ve Li, K.N. (2000). A Genetic Algorithm for Scheduling Staff of Mixed Skills under Multi-Criteria. *European Journal of Operational Research*. 125(2): 359-369.

Causmaecker, P. ve Berghe, G.V. (2011). A Categorisation of Nurse Rostering Problems. *Journal of Scheduling*. 14(1): 3-16.

Cheang, B, Li, H, Lim, A. ve Rodrigues, B. (2003). Nurse Rostering Problems a Bibliographic Survey. *European Journal of Operational Research*. 151(3): 447-460.

Clark, A.R. ve Walker, H. (2011). Nurse Rescheduling with Shift Preferences and Minimal Disruption. *Journal of Applied Operational Research*. 3(3): 148-162.

Constantino, A.A, Landa-Silva, D, de Melo, E.L, de Mendonça, C.F.X, Rizzato, D.B. ve Romao, W. (2014). A Heuristic Algorithm Based on Multi-Assignment Procedures for Nurse Scheduling. *Annals of Operations Research*. 218(1): 165-183.

Conway, R.W., Maxwell, W.L. ve Miller, L.W. (2012). Theory of scheduling. New York: Courier Corporation.

Çivril, H. (2009). *Hemşire Çizelgeleme Problemlerinin Genetik Algoritma ile Çözümü*. (Yayınlanmış Yüksek Lisans Tezi). Isparta: Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü.

Davis, L. (1985). Job Shop Scheduling with Genetic Algorithm. Proceeding of the first International Conference on Genetic Algorithms (ss.136-140). Pittsburgh: Psychology Press.

Divina, F. ve Marchiori, E. (2002). Evolutionary Concept Learning. In GECCO. 343-350.

Dowland, K.A. ve Thompson, J.M. (2000). Solving a Nurse Scheduling Problem with Knapsacks. *Networks and Tabu Search. Journal of the Operational Research Society*. 51(7): 825-833.

Elen, A. (2011). *Çizelgeleme Probleminin Sezgisel Optimizasyon Yaklaşımıyla Çözümü*. (Yayınlanmış Yüksek Lisans Tezi). Karabük: Karabük Üniversitesi Fen Bilimleri Enstitüsü.

Engin, O. (2001). *Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Artırılmasında Parametre Optimizasyonu*. (Yayınlanmış Doktora Tezi). İstanbul: İstanbul Teknik Üniversitesi.

Engin, O. ve Fırlalı, A. (2002). Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu. *İTÜ Dergisi*. 1(1).

Erdem, A. (2008). *Bakım Kısıtı Altında Genetik Algoritmalarla Üretim Çizelgeleme*. (Yayınlanmış Yüksek Lisans Tezi). Sakarya: Sakarya Üniversitesi Fen Bilimleri Enstitüsü.

Erdem, A. (2008). *Bakım Kısıtı Altında Genetik Algoritmalarla Üretim Çizelgeleme*. (Yayınlanmış Yüksek Lisans Tezi). Sakarya: Sakarya Üniversitesi Fen Bilimleri Enstitüsü.

Erigüç, G. (1999). Ankara Meteropolitan Alanda Sağlık Bakanlığı Hastanelerinde Personel Devir Oranları ve Personelin İşinden Ayrılma Nedenleri. *Hacettepe Sağlık İdaresi Dergisi*. 4(1): 3-26.

Ernst, A.T, Jiang, H, Krishnamoorthy, M, ve Sier, D. (2004). Staff Scheduling and Rostering: A Review of Applications, Methods and Models. *European Journal of Operational Research*. 153(1): 3-27.

Gen, M., Cheng, R. ve Lin, L. (2008). Network models and optimization: Multiobjective genetic algorithm approach. *Springer Science and Business Media*.

Gözen, Ş. (2007). *Bulanık Esnek Akış Tipi Çok Prosesli Çizelgeme Problemlerinin Genetik Algoritma ve Tavlama Benzetimi ile Çözümü*. (Yayınlanmış Yüksek Lisans Tezi). Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

Güngör, İ. (2002). *Hemşire görevlendirme ve çizelgeme sorununa bir model önerisi*. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi. 7(2): 77-94.

Gürel, Ü.B. (2012). *Sınav Çizelgeme Probleminin Optimizasyonu*. (Yayınlanmış Yüksek Lisans Tezi). İstanbul: Maltepe Üniversitesi Fen Bilimleri Enstitüsü.

Hare, D.R. ve Kelowna, B.C. (2001). Staff Scheduling with Ilog Solver. Department of Mathematics and Statics. *Okagan University Collage*.

Haspesslagh, S, De Causmaecker, P, Schaerf, A. ve Stolevik, M. (2014). The First International Nurse Rostering Competition 2010. *Annals of Operations Research*. 218(1): 221-236.

Hayes, L. J, O'Brien-Pallas, L, Duffield, C, Shamian, J, Buchan, J, Hughes, F, Laschinger, H.K.S, North, N. ve Stone, P.W. (2006). Nurse Turnover: a Literature Review. *International Journal of Nursing Studies*. 43(2): 237-263.

Hezer, S. (2 Ocak 2014). Kombinatoriyel Optimizasyon Nedir?.  
<https://sedahezerr.wordpress.com/2014/01/02/kombinatoriyel-optimizasyon-nedir/>,  
(09.11.15).

Huang, F. (1999). A Primary Shift Rotation Nurse Scheduling Using Zero-one Linear Goal Programming. *Computers in nursing*. 17(3): 135-144.

Hung, R. (1995). Hospital Nurse Scheduling. *Journal of Nursing Administration*. 25(7-8): 21-23.

Hyatt, C. ve Weaver, P. (2006). *A Brief History of Scheduling*. Melbourne, Australia: Mosaic Project Services Pty Ltd.

Inoue, T, Furuhashi, T, Maeda, H. ve Takaba, M. (2003). A Proposal of Combined Method of Evolutionary Algorithm and Heuristics for Nurse Scheduling Support System. *Industrial Electronics*. 50(5): 833-838.

İşler, M. (2009). *Bulanık Esnek Akış Tipi Çizelgeleme Problemlerinin Paralel Doyumsuz Algoritma ile Çözümü: Bir Hazır Giyim İşletmesine Uygulanması*. (Yayınlanmış Yüksek Lisans Tezi). Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

Jafari, H, Bateni, S, Daneshvar, P, Bateni, S. ve Mahdioun, H. (2015). Fuzzy Mathematical Modeling Approach for the Nurse Scheduling Problem: A Case Study. *International Journal of Fuzzy Systems*. 1-13.

Jan, A, Yamamoto, M, ve Ohuchi, A. (2000). Evolutionary Algorithms for Nurse Scheduling Problem. *In Evolutionary Computation*. 196-203.

Jenal, R, Ismail, W.R, Yeun, L. C. ve Oughalime, A. (2011). A Cyclical Nurse Schedule Using Goal Programming. *Journal of Mathematical and Fundamental Sciences*. 43(3): 151-164.

Karatlı, M. ve Güngör, İ. (2010). Hemşire Çizelgeleme Sorununa Bir Çözüm Önerisi ve Bir Uygulama. *Alanya İşletme Fakültesi Dergisi*. 2(1): 22-52.

Kawanaka, H, Yoshikawa, T, Shinogi, T. ve Tsuruoka, S. (2003). Constraints and Search Efficiency in Nurse Scheduling Problem. In *Computational Intelligence in Robotics and Automation, 2003. Proceedings. 2003 IEEE International Symposium on IEEE*. 1: 312-317.

Kellogg, D.L. ve Walczak, S. (2007). Nurse Scheduling: from Academia to Implementation or Not?. *Interfaces*. 37(4): 355-369.

Keskin, K. (2010). *Beklemez Akış Tipi Çizelgeleme Problemlerinin Çok Amaçlı Genetik Algoritma ile Çözümü*. (Yayınlanmış Yüksek Lisan Tezi). Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü.

Kitada, M, Morizawa, K. ve Nagasawa, H. (2010). A Heuristic Method in Nurse Rerostering Following a Sudden Absence of Nurses. *In Proc. 11st Asia Pacific Industrial Engineering and Management Systems Conference*.

Körez, M.T. (2005). *Sıralı Akış Tipi Çizelgeleme Problemlerinde Genetik Algoritma Uygulaması*. (Yayınlanmış Yüksek Lisans Tezi). İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Kucukkoc, I. ve Zhang, D.Z. (2016). *Integrating Ant Colony and Genetic Algorithms in the Balancing and Scheduling of Complex Assembly Lines*. *The International Journal of Advanced Manufacturing Technology*. 82(1-4): 265-285.

Kurt, M. ve Semetay, C. (2001). Genetik Algoritma ve Uygulama Alanları. *Mühendis ve Makine*. 45(501): 19-24.

Kwak, N.K. ve Lee, C. (1997). A Linear Goal Programming Model for Human Resource Allocation in a Health-care Organization. *Journal of Medical Systems*. 21(3): 129-140.

Landa-Silva, D. ve Le, K.N. (2008). A Simple Evolutionary Algorithm with Self Adaptation for Multi Objective Nurse Scheduling. *Adaptive and Multilevel Metaheuristics*. (ss.133-155). *Verlag Berlin Heidelberg: Springer*.

Le, K.N. (2011). A study of Evolutionary Multiobjective Algorithms and Their Application to Knapsack and Nurse Scheduling Problems. (Published Doctoral Dissertation). Nottingham: University of Nottingham.

Lee, C.Y., Lei, L. ve Pinedo, M. (1997). Current trends in deterministic scheduling. *Annals of Operations Research*. 70: 1-41.

Legrain, A, Bouarab, H. ve Lahrichi, N. (2015). The Nurse Scheduling Problem in Real-life. *Journal of Medical Systems*. 39(1): 1-11.

Liepins, G.E., Hilliard, M.R., Palmer, M. ve Morrow, M. (1987). Greedy Genetics. In Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms: July 28-31, 1987 at the Massachusetts Institute of Technology. Cambridge, MA.

Lin, C.C, Kang, J. R, Chiang, D. J. ve Chen, C. L. (2015a). Nurse Scheduling with Joint Normalized Shift and Day-Off Preference Satisfaction Using a Genetic Algorithm with Immigrant Scheme. *International Journal of Distributed Sensor Networks*. 501: 595419.

Lin, C.C, Kang, J.R. ve Hsu, T.H. (2015b). A Memetic Algorithm with Recovery Scheme for Nurse Preference Scheduling. *Journal of Industrial and Production Engineering*. 32(2): 83-95.

Lin, C.C., Kang, J.R., Chiang, D. J, ve Chen, C. L. (2015). Nurse scheduling with joint normalized shift and day-off preference satisfaction using a genetic algorithm with immigrant scheme. *International Journal of Distributed Sensor Networks*.

Macerauskas, V. ve Teresius, V. (2015). Investigation Of Multi-Agent Control System. *Elektronika Ir Elektrotechnik*. 52(3): 21-26.

Maenhout, B. ve Vanhoucke, M. (2010). Branching Strategies in a Branch and Price Approach for a Multiple Objective Nurse Scheduling Problem. *Journal of Scheduling*. 13(1): 77-93.

Maenhout, B. ve Vanhoucke, M. (2013). An Integrated Nurse Staffing and Scheduling Analysis for Longer-term Nursing Staff Allocation Problems. *Omega*. 41(2): 485-499.

MathWorks, “ga”, 2016, <http://www.mathworks.com/help/gads/ga.html>, (06.07.16).

McNaughton, R. (1959). Scheduling with deadlines and loss functions. *Management Science*. 6(1): 1-12.

Meglino, B. M. (1979). A Methodology for Nurse Staffing. *California Management Review*. 21(3): 82-93.

Millar, H.H. ve Kiragu, M. (1998). Cyclic and Non-cyclic Scheduling of 12 h shift Nurses by Network Programming. *European Journal of Operational Research*. 104(3): 582-592.

Miller, H.E, Pierskalla, W.P. ve Rath, G.J. (1976). Nurse Scheduling Using Mathematical Programming. *Operations Research*. 24(5): 857-870.

Mitchell, M. (1998). *An introduction to genetic algorithms*. London: The MIT Press.

Miwa, M, Inoue, T, Matsuzaki, M, Furuhasi, T. ve Okuwa, S. (2002). Nurse Scheduling System Using Bacterial Evolutionary Algorithm Hardware. 3: 1801-1805.

Moz, M. ve Pato, M.V. (2007). A Genetic Algorithm Approach to a Nurse Rerostering Problem. *Computers and Operations Research*. 34(3), 667-691.

Obitko, Marek. "Introduction to Genetic Algorithms with Java Applets", 1998a, <http://www.obitko.com/tutorials/genetic-algorithms/introduction.php>, (19.04.16).

Obitko, Marek. "Selection", 1998b, <http://www.obitko.com/tutorials/genetic-algorithms/selection.php>, (19.04.16).

Ohki, M, Morimoto, A. ve Miyake, K. (2006). Nurse Scheduling by Using Cooperative GA with Efficient Mutation and Mountain Climbing Operators. *IEEE*. 164-169.



Onan, A. (2013). Metasezgisel Yöntemler ve Uygulama Alanları. *Çukurova Üniversitesi İİBF Dergisi*. 17(2): 113-128.

Özbakır, L. (2004). *Çok Objektifli Esnek Atölye Çizelgeleme Problemlerinin Sezgisel Yöntemlerle Modellenmesi, Analizi ve Çözümü*. (Yayınlanmış Doktora Tezi). Kayseri: Erciyes Üniversitesi Sosyal Bilimler Enstitüsü.

Özcan, E. (2005). In Computer and Information Sciences-ISCIS 2005 (ss.482-492), Boğaziçi University Department of Computer Engineering. 26-28 Ekim 2005.

Özkarahan, İ. ve Bailey, J.E. (1988). Goal Programming Model Subsystem of a Flexible Nurse Scheduling Support System. *IIE Transactions*. 20(3): 306-316.

Özkazanç, Ü.A. ve Anagün, S. (2001). Atelye Tipi Üretim Ortamında İşlerin Çizelgelenmesinde Toplam Tamamlanma Zamanının En Küçüklenmesine Dayalı Bir Sinirsel Ağ Yaklaşımı. *Anadolu Üniversitesi Bilim ve Teknoloji Dergisi*. 2(1): 125-136.

Öztürkoğlu, Y, Çalışkan, F. (2014). *Hemşire Çizelgeleminde Esnek Vardiya Planlaması ve Hastane Uygulması*. Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Dergisi. 16(1): 115-133.

Paksoy, S. (2007). *Genetik Algoritma ile Proje Çizelgeleme*. (Yayınlanmış Doktora Tezi). Adana: Çukurova Üniversitesi Sosyal Bilimler Enstitüsü.

Parlak, M. (2012). *Sıralı Akış Çizelgeleme Problemlerinin Arı Algoritmasıyla Çözümü*. (Yayınlanmış Yüksek Lisans Tezi). İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü.

Pinedo, M.L. (2012). *Scheduling: Theory, Algorithms, and Systems*. New York: Springer Science and Business Media.

Polat, A. (2006). *Kafes Sistemlerinin Genetik Algoritma ile Çok Amaçlı Optimizasyonu*. (Yayınlanmış Yüksek Lisans Tezi). Elazığ: Fırat Üniversitesi Fen Bilimleri Enstitüsü.

Puente, J, Gomez, A, Fernandez, I. ve Priore, P. (2009). *Medical Doctor Rostering Problem in a Hospital Emergency Department by Means of Genetic Algorithms*. Computers and Industrial Engineering. 56(4), 1232-1242.

Ratnayaka, R.K.T, Wang, Z.J, Anamalamudi, S. ve Cheng, S. (2012). *Enhanced Greedy Optimization Algorithm with Data Warehousing for Automated Nurse Scheduling System*. E-Health Telecommunication Systems and Networks. 1(04): 43.

Rivera, J.C. ve Mesa, S. (2015). *An Integer Programming Based Local Search Algorithm for the Nurse Scheduling Problem*.

Rothstein, M. (1973). *Hospital Manpower Shift Scheduling by Mathematical Programming*. *Health Service Research of Journal*. 8 (1): 60-66.

Sağlam, D. (2011). *Belirsizlik Altında Çizelgeleme Problemlerinin Optimizasyonu*. (Yayınlanmış Yüksek Lisans Tezi). Ankara: Hacettepe Üniversitesi Sosyal Bilimler Enstitüsü.

Seçme, G. (2006). *Akış Tipi Çizelgeleme Problemlerinin Yapay Sinir Ağları İle Modellenmesi*. (Yayınlanmış Yüksek Lisans Tezi). Kayseri: Erciyes Üniversitesi Sosyal Bilimler Enstitüsü.

Siferd, S.P. ve Benton, W.C. (1992). *Workforce Staffing and Scheduling: Hospital Nursing Specific Models*. European Journal of Operational Research. 60(3): 233-246.

Silva, J.L., ve Burke, E.K. (2004). *A Tutorial on Multiobjective Metaheuristics for Scheduling and Timetabling*. Multiple Objective Meta-Heuristics. 1-39.

Soto, R, Crawford, B, Monfroy, E, Palma, W. ve Paredes, F. (2013). *Nurse and Paramedic Rostering with Constraint Programming: A Case Study*. Romanian Journal of Information Science and Technology. 16(1): 52-64.

Şen, Z. (2004). *Genetik Algoritmalar ve En İyileme Yöntemleri*. İstanbul: Su Vakfı.

Taç, C.K. (2006). *Genetik Algoritma Kullanılarak Haftalık Ders Programı Zaman Çizelgeleme Yazılımının Geliştirilmesi*. İstanbul: Marmara Üniversitesi Fen Bilimleri Enstitüsü.

Tein, L.H. ve Ramli, R. (2010). *Recent Advancements of Nurse Scheduling Models and a Potential Path*. In Proc. 6th IMT-GT Conference on Mathematics, Statistics and its Applications. 395-409.

Tien, J. ve A. Kamiyama (1982). *On Manpower Scheduling Algorithms*. Society for Industrial and Applied Mathematics. 24(3): 275-287.

Todorovic, N. ve Petrovic, S. (2013). *Bee Colony Optimization Algorithm for Nurse Rostering*. Systems, Man, and Cybernetics: Systems, IEEE Transactions on. 43(2): 467-473.

Topaloglu, S. ve Selim, H. (2010). *Nurse Scheduling Using Fuzzy Modeling Approach*. Fuzzy Sets and Systems. 161(11): 1543-1563.

Trivedi, M.V. (1981). *A Mixed Integer Goal Programming Model For Nursing Service Budgeting*. Operations Research. 29(5): 1019-1034.

Tsai, C.C. ve Lee, C.J. (2010). *Optimization of Nurse Scheduling Problem with a Two Stage Mathematical Programming Model*. Asia Pacific Management Review. 15(4): 503-516.

Türkay, M. Optimizasyon Modelleri ve Çözüm Metodları.

Ulusoy, G., Sivrikaya-Şerifoğlu, F. ve Bilge, Ü. (1997). *A Genetic Algorithm Approach to the Simultaneous Scheduling of Machines and Automated Guided Vehicles*. *Computers and Operations Research*. 24(4): 335-351.

Vancza, J. ve Markus, A. (1991). *Genetic Algorithms in Process Planning*. *Computers in Industry*. 17(2-3): 181-194.

Vikipedi. (12 Haziran 2016). MATLAB. <https://tr.wikipedia.org/wiki/MATLAB>, (13.06.16).

Vikipedi. (12 Mart 2015). P ile NP Arasındaki İlişki. [https://tr.wikipedia.org/wiki/P\\_ile\\_NP\\_aras%C4%B1ndaki\\_ili%C5%9Fki](https://tr.wikipedia.org/wiki/P_ile_NP_aras%C4%B1ndaki_ili%C5%9Fki), (10.11.15).

Vikipedi. (29 Eylül 2014). NP (karmaşıklık). [https://tr.wikipedia.org/wiki/NP\\_\(karma%C5%9F%C4%B1kl%C4%B1k\)](https://tr.wikipedia.org/wiki/NP_(karma%C5%9F%C4%B1kl%C4%B1k)), (11.11.15).

Warner, D.M. (1976). Scheduling Nursing Personnel According to Nursing Preference: A Matematical Programming Approach. *Operation Research*. 24(5): 842-856.

Watanabe, M., Ida, K. ve Gen, M. (2005). A Genetic Algorithm with Modified Crossover Operator and Search Area Adaptation for the Job-Shop Scheduling Problem. *Computers and Industrial Engineering*. 48(4): 743-752.

Weisstein, E.W. Complexity Theory. MathWorld A Wolfram Web Resource. <http://mathworld.wolfram.com/ComplexityTheory.html>, (11.11.15).

Xu, Y., Li, K., Hu, J. ve Li, K. (2014). A Genetic Algorithm for Task Scheduling on Heterogeneous Computing Systems using Multiple Priority Queues. *Information Sciences* 270: 255-287.

Yilmaz, E. (2012). A Mathematical Programming Model for Scheduling of Nurses' Labor Shifts. *Journal of Medical Systems*. 36(2). 491-496.

Zhou, H., Feng, Y. ve Han, L. (2001). The Hybrid Heuristic Genetic Algorithm for Job Shop Scheduling. *Computers and Industrial Engineering*. 40(3): 191-200.





**EKLER**

**Ek 1: Koroner Yoğun Bir Aylık Çalışma Çizelgesi**

	01.02.2015	02.02.2015	03.02.2015	04.02.2015	05.02.2015	06.02.2015	07.02.2015
Hemşireler	Pazar	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi
1	16`08	NÇ	08`16	08`16	16`08	NÇ	08`16
2	08`16	16`08	NÇ	08`16	08`16	08`16	16`08
3	08`16	08`16	16`08	NÇ	08`16	08`16	08`16
4	08`16	08`16	08`16	16`08	NÇ	08`16	08`16
5	08`16	08`16	08`16	08`16	08`16	16`08	NÇ
6	08`16	08`16	08`16	08`16	08`16	08`16	08`16
7	08`16	08`16	08`16	08`16	08`16	08`16	16`08
8	08`16	08`16	16`08	NÇ	16`08	NÇ	08`16
9	08`16	S	S	S	S	S	S
10	08`16	08`16	08`16	08`16	08`16	08`16	08`16
11	16`08	NÇ	08`16	16`08	NÇ	08`16	08`16
12	08`16	16`08	NÇ	08`16	08`16	16`08	NÇ
13	08`16	08`16	08`16	08`16	08`16	08`16	08`16
14	08`16	08`16	08`16	08`16	08`16	08`16	08`16
	08.02.2015	09.02.2015	10.02.2015	11.02.2015	12.02.2015	13.02.2015	14.02.2015
Hemşireler	Pazar	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi
1	08`16	08`16	08`16	08`16	08`16	08`16	16`08
2	NÇ	08`16	08`16	08`16	08`16	16`08	NÇ
3	08`16	16`08	NÇ	08`16	08`16	08`16	08`16
4	08`16	08`16	08`16	08`16	08`16	08`16	08`16
5	08`16	08`16	08`16	08`16	16`08	NÇ	08`16
6	16`08	NÇ	16`08	NÇ	16`08	NÇ	08`16
7	NÇ	08`16	08`16	16`08	NÇ	08`16	16`08
8	08`16	08`16	08`16	08`16	08`16	08`16	08`16
9	S	S	S	S	S	S	08`16
10	16`08	NÇ	16`08	NÇ	08`16	16`08	NÇ
11	08`16	08`16	08`16	16`08	NÇ	08`16	08`16
12	08`16	16`08	NÇ	08`16	08`16	08`16	08`16
13	08`16	08`16	08`16	08`16	08`16	08`16	08`16
14	08`16	08`16	08`16	08`16	08`16	08`16	08`16

	15.02.2015	16.02.2015	17.02.2015	18.02.2015	19.02.2015	20.02.2015	21.02.2015
Hemşireler	Pazar	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi
1	NÇ	08``16	08``16	08``16	08``16	08``16	08``16
2	08``16	08``16	08``16	08``16	16``08	NÇ	08``16
3	08``16	08``16	08``16	08``16	08``16	16``08	NÇ
4	08``16	08``16	16``08	NÇ	08``16	08``16	08``16
5	08``16	08``16	08``16	08``16	16``08	NÇ	08``16
6	08``16	16``08	NÇ	16``08	NÇ	16``08	NÇ
7	NÇ	08``16	08``16	08``16	08``16	08``16	08``16
8	16``08	NÇ	08``16	08``16	08``16	08``16	08``16
9	08``16	16``08	NÇ	16``08	NÇ	08``16	08``16
10	16``08	NÇ	16``08	NÇ	08``16	08``16	16``08
11	08``16	08``16	08``16	08``16	08``16	08``16	08``16
12	08``16	08``16	08``16	08``16	08``16	08``16	16``08
13	08``16	08``16	08``16	08``16	S	08``16	08``16
14	08``16	08``16	08``16	08``16	08``16	08``16	08``16
	22.02.2015	23.02.2015	24.02.2015	25.02.2015	26.02.2015	27.02.2015	28.02.2015
Hemşireler	Pazar	Pazartesi	Salı	Çarşamba	Perşembe	Cuma	Cumartesi
1	08``16	08``16	08``16	08``16	16``08	08``16	08``16
2	08``16	08``16	S	S	08``16	08``16	08``16
3	08``16	08``16	08``16	08``16	08``16	16``08	NÇ
4	16``08	NÇ	08``16	08``16	08``16	08``16	16``09
5	08``16	08``16	08``16	16``08	NÇ	08``16	08``16
6	08``16	08``16	16``08	NÇ	08``16	08``16	16``09
7	08``16	08``16	16``08	NÇ	08``16	08``16	08``16
8	08``16	16``08	NÇ	08``16	08``16	08``16	08``16
9	16``08	NÇ	08``16	08``16	16``08	NÇ	08``16
10	NÇ	16``08	NÇ	08``16	08``16	16``08	NÇ
11	08``16	08``16	08``16	16``08	NÇ	S	08``16
12	NÇ	08``16	08``16	08``16	08``16	08``16	08``16
13	08``16	08``16	08``16	08``16	08``16	08``16	08``16
14	08``16	08``16	08``16	08``16	08``16	08``16	08``16



**Ek 2:** ga\_big.m

```
n=14; %hemşire sayısı
```

```
m=28; %gün sayısı
```

```
k=2; %vardiya sayısı
```

```
u=6; %uzman sayısı (ilk u adet hemşire uzmandır)
```

```
h=2; %gece nöbete kalmayacaklar
```

```
load('data');
```

```
OT=ones(n,1)*5; %overtime
```

```
lims=1:n*m*k;
```

```
lb=zeros(1,n*m*k);
```

```
ub=ones(1,n*m*k);
```

```
numberOfVariables = n*m*k;
```

```
f = @(x)objectfn(x,OT,n,m,k);
```

```
%normal çözümler
```

```
[A,b, Aeq,beq]=makelincon(n,m,u,h,XO);
```

```
[x,fval] = ga(f,numberOfVariables,A,b,Aeq,beq,lb,ub)
```

```
[x, fval] = fmincon(f,lb,A,b,Aeq,beq,lb,ub)
```

```
%integer çözüm
```

```
intcon=1:784;
```

```
[A,b, Aeq,beq]=makelincon2(n,m,u,h,XO);
```

```
[x,fval,exitflag]= ga(f,numberOfVariables,A,b,[],[],lb,ub,[],intcon)
```

**Ek 3:**ga\_small.m

```
n=5; %hemşire sayısı
m=7; %gün sayısı
k=2; %vardiya sayısı
u=3; %uzman sayısı (ilk u adet hemşire uzmandır)
h=1; %gece nöbete kalmayacaklar
load('data');

OT=OT(1:n,:); %overtime
XO=XO(1:n,1:m);
lims=1:n*m*k;
lb=zeros(1,n*m*k);
ub=ones(1,n*m*k);
numberOfVariables = n*m*k;
intcon=1:numberOfVariables;

f = @(x)objectfn(x,OT,n,m,k);

%normal çözümler
[A,b, Aeq,beq]=makelincon(n,m,u,h,XO);
[x,fval] = ga(f,numberOfVariables,A,b,Aeq,beq,lb,ub)
[x, fval] = fmincon(f,lb,A,b,Aeq,beq,lb,ub)

%integer çözüm

[A,b, Aeq,beq]=makelincon2(n,m,u,h,XO);
[x, fval] = fmincon(f,lb,A,b,[],[],lb,ub)

options.CreationFcn=[];
```

```
[Aeq8,A2,A7,A6,Aeq9,Aeq5,Aeq3,Aeq4,beq8,b2,b7,b6,beq9,beq5,beq3,beq4]=mak  
elincon3(n,m,u,h,XO);
```

```
A=[Aeq8;A7;A6;Aeq9;Aeq5;Aeq3;Aeq4];
```

```
b=[beq8;b7;b6;beq9;beq5;beq3;beq4];
```

```
A=[Aeq8;A7];
```

```
b=[beq8;b7];
```

```
[x,fval,exitflag]= ga(f,numberOfVariables,A,b,[],[],lb,ub,[],intcon,options)
```



**Ek 4:**Makelincon.m

function

[Aeq8,A2,A7,A6,Aeq9,Aeq5,Aeq3,Aeq4,beq8,b2,b7,b6,beq9,beq5,beq3,beq4]=mak  
elincon3(n,m,u,h,XO)

% i: hemşireler (n),j:günler (m), k:vardiya (2)

% sıralarını değiştirdim

p=n\*m;

A2=zeros(1,2\*p);

b2=zeros(1,1);

A7=zeros(1,2\*p);

b7=zeros(1,1);

A6=zeros(1,2\*p);

b6=zeros(1,1);

Aeq8=zeros(1,2\*p);

beq8=zeros(1,1);

Aeq9=zeros(1,2\*p);

beq9=zeros(1,1);

Aeq5=zeros(1,2\*p);

beq5=zeros(1,1);

Aeq3=zeros(1,2\*p);

beq3=zeros(1,1);

Aeq4=zeros(1,2\*p);

beq4=zeros(1,1);

c=1;

d=1; %counters

e=0;

% aynı gün ya gece ya gündüz vardiyasında çalışılmalı ya da izinli

% olunmalıdır. (8)

% % XO: yapısı A ile aynı olan  $1 \times n \times m$  boyutlu izin matrisi

d=1;

for t=1:p

Aeq8(d,t)=1;

Aeq8(d,t+p)=1;

beq8(d,1)=1-XO(t);

d=d+1;

end

for t=1:p

Aeq8(d,t)=-1;

Aeq8(d,t+p)=-1;

beq8(d,1)=-1+XO(t);

d=d+1;

end

%aynı gün hem gündüz hem gece nöbet tutulmaz (2)

c=1;

for t=1:p

A2(c,t)=1;

A2(c,t+p)=1;

b2(c,1)=1;

c=c+1;

end

%gece nöbet tutan ertesi gün gündüz çalışmaz (7)

c=1;

for t=1:p-m

A7(c,t+1)=1;

A7(c,t+p)=1;

b7(c,1)=1;

c=c+1;

```

end

% ayda en az 160 saat çalışmalı (6) (>=)
c=1;
e=0;
for t=1:n
    for i=1:m
        A6(c,t+e)=-8;
        A6(c,p+t+e)=-16;
        e=e+1;
    end
    e=e-1;
    b6(c,1)=-40;
    c=c+1;
end
end

```

%eşitlik kısıtları- eşitsizliğe dönüştürülüyor

%akşam vardiyada çalışan iki kişiden birinin uzman olması gerekir (9)

```

d=1;
for t=p+1:p+m
    for k=1:u
        Aeq9(d,t+k*m-m)=1;
    end
    beq9(d,1)=1;
    d=d+1;
end
for t=p+1:p+m
    for k=1:u

```

```
Aeq9(d,t+k*m-m)=-1;  
end  
beq9(d,1)=-1;  
d=d+1;  
end
```

% akşam nöbetinde iki kişi çalışır (kısıtlı olanlar hariç) (5)

```
d=1;  
for t=p+1:(p+(n-h)*m)  
    Aeq5(d,t)=1;  
end  
beq5(d,1)=1;  
d=d+1;
```

```
for t=p+1:(p+(n-h)*m)  
    Aeq5(d,t)=-1;  
end  
beq5(d,1)=-1;  
d=d+1;
```

% sorumlu hemşire gece nöbeti tutmaz (3)

```
d=1;  
for t=2*p-m+1:2*p  
    Aeq3(d,t)=1;  
    beq3(d,1)=0;  
    d=d+1;  
end
```

```
for t=2*p-m+1:2*p  
    Aeq3(d,t)=-1;  
    beq3(d,1)=0;  
    d=d+1;
```

```
end
% hariç hemşireler gece nöbeti tutmaz (4)
d=1;
for t=2*p-h*m+1:2*p-m
    Aeq4(d,t)=1;
    beq4(d,1)=0;
    d=d+1;
end
for t=2*p-h*m+1:2*p-m
    Aeq4(d,t)=-1;
    beq4(d,1)=0;
    d=d+1;
end
```



**Ek 5:** Objectvec.m

```
function [cost,costt]=objectvec(x,n,m,k)
```

```
%intlinprog vektör istiyor, costt kullanılacak
```

```
p=n*m;
```

```
costt=0;
```

```
work=0;
```

```
s=0;
```

```
for t=1:p
```

```
    costt(1,t)=8;
```

```
    costt(1,p+t)=16;
```

```
end
```

```
cost=sum(x.*costt);
```