



PERSONEL DEVAMSIZLIĞINDA ATAMA VE DENGEME PROBLEMİ İÇİN
KARAR DESTEK SİSTEMİ ÖNERİSİ: BİR UYGULAMA

Abdurrahim BULUT

Yüksek Lisans Tezi

Endüstri Mühendisliği Anabilim Dalı

Mayıs - 2019

PERSONEL DEVAMSIZLIĞINDA ATAMA VE DENGELEME PROBLEMİ İÇİN
KARAR DESTEK SİSTEMİ ÖNERİSİ: BİR UYGULAMA

Abdurrahim BULUT

Kütahya Dumlupınar Üniversitesi
Lisansüstü Öğretim ve Sınav Yönetmeliği Uyarınca
Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman: Doç. Dr. Şafak KIRIŞ

Mayıs - 2019

KABUL VE ONAY SAYFASI

Abdurrahim BULUT'un Yüksek lisans tezi olarak hazırladığı "Personel devamsızlığında atama ve dengeleme problemi için karar destek sistemi önerisi: Bir uygulama" başlıklı bu çalışma, jürimizce Kütahya Dumlupınar Üniversitesi Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

31 /05/2019

Prof. Dr. Önder UYSAL
Enstitü Müdürü, Fen Bilimleri Enstitüsü

Prof. Dr. Özden ÜSTÜN
Bölüm Başkanı, Endüstri Mühendisliği Bölümü

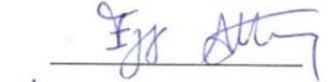
Doç. Dr. Şafak KIRIŞ
Danışman, Endüstri Mühendisliği Bölümü,
Kütahya Dumlupınar Üniversitesi

Sınav Komitesi Üyeleri

Prof.Dr. Ezgi AKTAR DEMİRTAŞ
Endüstri Mühendisliği Bölümü, Eskişehir Osmangazi Üniversitesi

Prof. Dr. Özden ÜSTÜN
Endüstri Mühendisliği Bölümü, Kütahya Dumlupınar Üniversitesi

Doç. Dr. Şafak KIRIŞ
Endüstri Mühendisliği Bölümü, Kütahya Dumlupınar Üniversitesi



ETİK İLKE VE KURALLARA UYGUNLUK BEYANI

Bu tezin hazırlanmasında Akademik kurallara riayet ettiğimizi, özgün bir çalışma olduğunu ve yapılan tez çalışmasının bilimsel etik ilke ve kurallara uygun olduğunu, çalışma kapsamında teze ait olmayan veriler için kaynak gösterildiğini ve kaynaklar dizininde belirtildiğini, Yüksek Öğretim Kurulu tarafından kullanılmak üzere önerilen ve Kütahya Dumlupınar Üniversitesi tarafından kullanılan İntihal Programı ile tarandığını ve benzerlik oranının %11 çıktığını beyan ederiz. Aykırı bir durum ortaya çıktığı takdirde tüm hukuki sonuçlara razı olduğumuzu taahhüt ederiz.



Doç. Dr. Şafak KIRIŞ



Abdurrahim BULUT

PERSONEL DEVAMSIZLIĞINDA ATAMA VE Dengeleme Problemi İÇİN KARAR DESTEK SİSTEMİ ÖNERİSİ: BİR UYGULAMA

Abdurrahim BULUT'

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2019

Tez Danışmanı: Doç. Dr. Şafak KIRIŞ

ÖZET

Günümüzün yoğun rekabet ortamı içerisinde işletmelerin varlıklarını sürdürebilmeleri için üretim kaynaklarını etkin bir şekilde kullanmaları gerekmektedir. Gün geçtikçe bu konunun önemi daha da artmaktadır. Bu nedenle işletmeler, çeşitli yaklaşımlar ile kaynaklarını en iyi şekilde yönetebilmek adına çalışmalar yapmaktadırlar.

İşletmeler için en önemli kaynaklardan birisi iş gücüdür. Emek yoğun çalışan işletmelerde insan faktöründen etkin olarak faydalanmak gerekmektedir. Bunun yanında işletmelerde işe gelmeme durumuyla da çok sık karşılaşılmaktadır. İşe gelmeyen personelin yerine çalışacak personelin atanması, özellikle küçük ve orta büyüklükteki işletmelerde, işletme yöneticilerinin kişisel görüşlerine göre yapılmakta olup, genellikle istenilen hedeflere ulaşılamamakta ve görünmeyen kayıplar oluşmaktadır.

Bu çalışmada bir otomotiv yan sanayi işletmesinde günlük devamsızlık yapan personelin yerine uygun personelin atanması ve gerekli işlerin dağıtılması için öncelikle bir matematiksel model önerilmiş, ardından işletme ihtiyaçlarına daha rahat cevap verebilmesi amacıyla bir Karar Destek Sistemi (KDS) tasarlanmıştır. Önerilen KDS ile işletmede personelin işe gelmemesi durumunda ilgili çalışma istasyonlarına, mevcut ekip içerisinde en yetkin personel atanmakta ve gerekli durumlarda da işler diğer istasyonlara paylaştırılmaktadır. Böylece sistematik bir süreç yönetimi ile oluşabilecek kayıpların en küçüklenmesi ve standart üretimin devam etmesi amaçlanmaktadır.

Anahtar Kelimeler: Atama problemi, Karar destek sistemi, Personel devamsızlığı.

A DECISION SUPPORT SYSTEM FOR THE ASSIGNMENT AND REBALANCING PROBLEM IN THE ABSENCE OF STAFF: A CASE STUDY

Abdurrahim BULUT

Industrial Engineering, M.S. Thesis, 2019

Thesis Supervisor: Assoc. Prof. Dr. Şafak KIRIŞ

SUMMARY

In today's intense competition environment, enterprises must use their production resources effectively in order to sustain their assets. The importance of this issue is increasing day by day. For this reason, enterprises are working to manage their resources in the best way with various methods.

One of the most important sources for businesses is labor. It is necessary to make effective use of human factor in labor intensive enterprises. In addition, absence in the business is very common. The appointment of the personnel who will work instead of the absentee personnel is done according to the personal opinions of the managers in the small and medium-sized enterprises, and generally the desired targets are not reached and the invisible losses occur.

In this study, a mathematical model was firstly proposed in order to assign the appropriate personnel instead of daily absentee staff in an automotive supplier industry and to distribute the work if necessary, afterwards, in order to answer the needs of the enterprise, the Decision Support System (DSS) has been designed. In the event that the personnel does not come to work in the operation with the proposed KDS, the most authorized personnel is assigned to the relevant work stations and the work is shared to the other stations where necessary. Therefore, with a systematic process management, sustaining of standard production and minimizing the losses which may occur are aimed.

Key Words: Assignment problem, Decision support system, Absence of staff.

TEŞEKKÜR

Tez araştırma konusunun belirlenmesinde, araştırılmasında, değerlendirilmesinde ve yazılmasında büyük emekleri olan, bilginin ve ilimin yolunda sağlam adımlar atmamda bana öncülük eden tez danışmanım Sayın Doç. Dr. Şafak KIRIŞ'a, hayatımın her noktasında ne olursa olsun hep yanımda olan annem, babam ve sevdiklerime hürmetlerimi ve şükranlarımı sunarım.

Ayrıca iş hayatım boyunca beni destekleyen ve engin tecrübelerini paylaşan değerli yöneticilerime desteklerinden dolayı sonsuz teşekkürlerimi arz ederim.



İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
ŞEKİLLER DİZİNİ.....	ix
ÇİZELGELER DİZİNİ	x
SİMGELER VE KISALTMALAR DİZİNİ.....	xi
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	3
3. PERSONEL DEVAMSIZLIĞI PROBLEMİ VE ÖNERİLEN MODEL	13
3.1. Matematiksel Model	13
3.2. Matematiksel Model için Uygulama.....	16
3.3. Joker Personel Sayısının Belirlenmesi ve Süreçlere Göre Yetkinlik Matrisi	16
3.4. Modelin Çözümü.....	18
4. KARAR DESTEK SİSTEMLERİ	20
5. PERSONEL DEVAMSIZLIĞINDA ATAMA VE DENGEME PROBLEMİ İÇİN KARAR DESTEK SİSTEMİ	23
5.1. Karar Destek Sisteminin Akış Şeması.....	24
5.2. Kullanıcı Arayüzleri ve Tanımları.....	26
5.3. KDS için Uygulama	44
5.SONUÇ VE ÖNERİLER	52
KAYNAKLAR DİZİNİ.....	54
EKLER	
EK 1. Matematiksel Modelin Lingo 11.0 Programındaki Yazılışı ve Çözüm Raporları	
EK 2. Personel yokluğunda Joker Personel Atama Karar Destek Sistemi Algoritmalarının C# Dilinde Kodları ve Açıklamaları	

ŞEKİLLER DİZİNİ

<u>Sekil</u>	<u>Sayfa</u>
2.1. Çözüm yöntemlerine göre çalışma yüzdeleri.....	12
2.2. Uygulama alanlarına göre çalışma yüzdeleri.....	12
3.1. Doksan günlük devamsızlık yapan personelin saçılım diyagramı.....	17
3.2. Doksan günlük devamsızlık yapan personelin histogram diyagramı.....	17
4.1. KDS'lerde karar verme süreci.....	22
5.1. Personel devamsızlığında atama ve dengeleme problemi için karar destek sistemi akış şeması.....	27
5.2. Giriş arayüzü.....	27
5.3. Ana menü arayüzü.....	28
5.4. Ana ekran arayüzünde kullanıcı ekleme silme butonu.....	30
5.5. Sisteme kullanıcı ekleme ve silme arayüzü.....	30
5.6. Personel listesi güncelleme arayüzü.....	32
5.7. Sisteme proje ekleme ve silme arayüzü.....	33
5.8. Süreç ekleme-silme ve süreçte var olan istasyonu silme arayüzü.....	34
5.9. Ekip oluşturma ve projelere ekip atama arayüzü.....	35
5.10. Ana ekran arayüzünde süreçlere ve istasyonlara personel atama veya personel çıkarma butonu.....	36
5.11. Süreçlere personel atama ve atanmış personeli sürecin ilgili istasyonuna atama arayüzü... ..	37
5.12. Ana ekran arayüzünde sisteme kayıtlı olan ekiplerin personel listesinin görüntüleme butonu	38
5.13. Sisteme kayıtlı ekiplere atanmış personel listesinin görüntülediği arayüz.....	39
5.14. Ana ekran arayüzünde ekip listesini güncelleme butonu.....	40
5.15. Atanmış joker personelin pasif hale getirilmesi için onay mesajı.....	40
5.16. Ana ekran arayüzünde işe gelmeyen personelin bilgilerini sisteme giriş butonu.....	41
5.17. İşe gelmeyen personelin bilgisinin sisteme giriş arayüzü.....	41
5.18. İşe gelmeyen personelin bilgisinin sisteme girişinin onay mesajı.....	42
5.19. İşe gelmeyen personelin yerine atama yapılırken sistemden alınan onay mesajı.....	42
5.20. İşe gelmeyen personelin yerine atama yapma arayüzü.....	43
5.21. İşe gelmeyen personelin yerine atanacak joker personel olmaması durumunda onay mesajı.....	44

ÇİZELGELER DİZİNİ

<u>Cizelge</u>	<u>Sayfa</u>
3.1. Kapı tesisatı projesinde çalışabilecek joker personelin süreç bazlı yetkinlik puanları	18
3.2. Kapı tesisatı üretim hattı istasyonlarının çevrim süreleri.....	18
3.3. Üç farklı senaryo sonuçları.	20
5.1. İşletmede bir günlük devamsızlık yapan personelin çalıştıkları istasyonlara atanan joker personelin süreç bazlı puanları.....	46
5.2. Joker personelin süreç bazlı puanları.	47
5.3. İşletmede devamsızlık yapan personelin bulunduğu ekip isimleri ve çalıştıkları istasyon numaraları.	48
5.4. Veri tabanında dengeleme öncesi Ekip10'un istasyon çevrim süreleri.	49
5.5. Ekip 10' un 9 ve 16 nolu istasyonlarına atanacak joker personel olmaması durumunda yeni çevrim süreleri.....	50

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltma</u>	<u>Açıklama</u>
KDS	Karar Destek Sistemi
YBS	Yönetim Bilgi Sistemi



1. GİRİŞ

Günümüzde hızla değişen müşteri talepleri, kaynakların kısıtlı olması, maliyetlerin gün geçtikçe artması, işletmelerin mevcut kaynaklarını etkin şekilde kullanmalarını ve maliyetlerini en küçüklemeye yönelik çalışmalarını arttırmayı gerektirmektedir.

Emek yoğun çalışan bir işletmede şüphesiz insan faktörünün rolü büyüktür. Bu nedenle işletme yöneticileri, işgücü kaynaklarını etkin kullanma konusundaki çalışmalarına yoğunlaşmalıdırlar. Bunun dışında emek yoğun işletmelerde sıklıkla karşılaşılan bir problem de devamsızlıktır. Bu gibi problemler için de işletmelerin doğru zamanda gerekli önlemleri almaları gerekmektedir.

Yüksek hacimlerde üretim gerçekleştirebilmek amacıyla tasarlanan montaj hatlarında iş istasyonlarına atanan personelin devamsızlık yapması durumunda, yerine uygun personel çalıştırma sorunu ortaya çıkmaktadır. Bu sorunun en küçük kayıp ile ortadan kaldırılması için işletmelerin uygun çözüm önerileri bulmaları gerekmektedir. Burada amaç, kurulacak montaj hattının verimli olarak çalışabilmesi için oluşturulan istasyon kapasitesinin en iyi şekilde kullanılmasıdır. İşletmelerde devamsızlık durumunda personel atama sürecinde karşılaşılan en önemli sorun, işe gelmeyen personelin yerine atanacak personelin ilgili istasyona atama probleminde sistematik bir karar sürecinin olmayışıdır. Devamsızlık yapan personelin yerine, o istasyonda çalışabilecek yetkinlikte bir personeli o istasyona atamak uygun olacaktır.

İşletmeler, bu yetkinlikteki personeli yetiştirerek ve kendi bünyesinde ayrı bir ekip oluşturarak bir çözüm sağlayabilmektedirler. Joker personel olarak adlandırılan bu yetkin personel atandığı istasyonda en az verim kaybı ile işlemleri yerine getirmekle yükümlü olmaktadır. Yeterli joker eleman olmadığı durumlarda ise en uygun istasyonlara işlemlerin dağıtımı ile problem çözülmeye çalışılmaktadır.

Yapılan literatür araştırmasında bu alanda kısıtlı sayıda çalışmaya rastlanmıştır olup, problemlerin işletmelere özel olması nedeniyle de bu çalışmanın literatüre katkısı olacağı düşünülmektedir.

Bu çalışmada günlük işe gelmeyen personelin yerine atama için öncelikle bir matematiksel model geliştirilmiştir. Bu model, yeterli joker personel varsa; joker personel ekibinden uygun personelin atanmasını, eğer yeterli joker personel yok ise; yeterli olan personelin atanıp, kalan işlerin iş yüklerinin ise uygun istasyonlara dağıtılmasını sağlamaktadır.

Önerilen bu model otomotiv sektöründe faaliyet gösteren bir işletmenin verileri kullanılarak gerçekleştirilmiştir. Model Lingo 11.0 programında çözülmüştür.

Bir matematiksel modelin çözümünün zaman alması ve günlük kullanıma çok uygun olmaması, işletme için daha kullanışlı bir çözüm gerektirmiştir. Bu nedenle, personel devamsızlığında atama ve dengeleme problemi için bir Karar Destek Sistemi (KDS) geliştirilmiştir. Bu sistem, işletme yöneticilerine kullanım kolaylığı sunan arayüzlerinden ve bilgi mesajlarından oluşmaktadır. Önerilen KDS, sistematik karar verebilen, kullanıcıya kolaylık sağlayan ve dinamik kullanma koşullarına uygun bir sistemdir. Böylece problem karşısında işletme, hızlı ve sorunsuz bir şekilde hedeflerine en uygun çözümü bulabilecektir.

Çalışmada yer alan bölümler ve bu bölümler içinde ele alınan konular şu şekildedir: İkinci bölümde atama ile ilgili literatür araştırması yapılmış ve literatürde atama ile ilgili çalışmaların yöntem ve uygulama alanları açısından sınıflandırılması ve bu sınıflandırmanın yüzdeleri hakkında bilgi verilmiştir. Üçüncü bölümde çözüm için matematiksel model oluşturulup, işletmede çalışan bir üretim ekibi için uygulaması yapılmıştır. Dördüncü bölümde KDS hakkında genel bir bilgi verilmiş ve karar verme süreci açıklanmıştır. Beşinci bölümde önerilen model için oluşturulan KDS üç başlık altında açıklanmış ve sistem üzerinde bir uygulama değerlendirilmiştir. Altıncı bölümde ise sonuç ve öneriler sunulmuştur.

2. LİTERATÜR ARAŞTIRMASI

İş gücü planlama literatürü, kaynak optimizasyonu (resource optimization), kaynak planlama (resource planing), iş gücü ihtiyaç tahmini (aggregate production planing), kaynak atama (workforce allocation) ve personel çizelgeleme ve görevlendirme (staff scheduling and rostering) anahtar kelimelerinden yararlanılarak araştırılmıştır.

Lee vd. (1995), personel planlama ve görevlendirme konusundaki makalelerinde, çeşitli kısıtlar altında ders programı çizelgeleme problemi üzerinde çalışmışlardır.

Niehauss (1995) çalışmasında, iş gücü kaynaklarını planlama konusunu incelemiştir. Çalışmada stratejik amaçlar ile birlikte matematiksel model kurulmuş ve sonuçlar değerlendirilmiştir.

Beaumont (1997) çalışmasında, personel çizelgeleme metodunu karma tamsayılı programlama ile geliştirmiştir. Modelde iş gücü olarak işletme elemanlarının dışında birde taşeron işçi olarak farklı bir iş gücü kaynağı kullanmıştır.

Mohanty vd. (1997) çalışmalarında, işletmenin planları ve yapısal ihtiyaçları baz alarak satış adetlerini dikkate alan doğrusal programlama yoluyla gerekli insan kaynağı ihtiyacı için bir öneri sunmuşlardır. Bu çalışmada ayrıca personelin geliştirilmesi ve terfi sisteminde bu sonuçlara istinaden değişiklikler yapılması önerilmiştir.

Huang vd. (1998) çalışmalarında, depolama yeri sınırlı olan çok aşamalı üretim sistemlerinin ve kapasite kısıtı olan iş istasyonlarının toplu planlama problemlerinin modellenmesinden bahsetmişlerdir. Yazarlar bu çalışmada bir bilgisayar çözümü sonrasında, çok aşamalı ve çeşitli üretim sistemlerinde planlamayı daha iyi hale getirebilecek optimum sonuç veren bir model ortaya koymuşlardır. Çalışmada dönem, çevrim süresi ve tesis içerisinde tutulan stok miktarı göz önünde bulundurulmuştur.

Ozdamar vd. (1998) çalışmalarında, hiyerarşik karar destek sistemi ile toplu üretim planlama konusunda bir öneri sunmuşlardır. Örnek uygulama için bir işletmenin yıllık satış verilerini kullanarak toplu üretim planı yapmışlardır.

Eimaraghy vd. (1999) çalışmalarında, işçi ve makine kısıtlarına sahip olan genetik algoritma tabanlı üretim sistemindeki çizelgeleme problemi için bir planlama yaklaşımı ortaya koymuşlardır. Bu genetik algoritma yeni bir kromozom örneğini kullanarak makine ve işçi atamalarını gerçekleştirmektedir.

Baykasoglu (2001) çalışmasında, çok amaçlı tabu arama algoritmasını kullanarak toplu üretim planlama konusunu ele almıştır. Yazar, toplu planlama modeline alt yüklenici (taşeron) ve kalıp değişimi için karar değişkenlerini ekleyerek, modelin çözümüne yönelik teknikler geliştirmeye çalışmıştır. Tabu arama algoritmasını kullanarak toplu üretim planlama modelinin daha hızlı hazırlanmasını amaçlamıştır.

Wang vd. (2001), pazarın talebini karşılayacak, ürün fiyatı, alt yüklenici (taşeron) maliyeti, işçi sayısı, üretim kapasitesi içeren toplu üretim planlama modelini çözebilmek için bulanık doğrusal programlama modeli geliştirmişlerdir. Talebin belirsiz olması nedeniyle yazarlar talebi belirlemek için bulanık mantık kullanmışlardır. Buna bağlı olarak diğer tüm değişkenler doğrusal programlama vasıtasıyla bulunmuştur. Modelde devamsızlık gibi özel durumlar için herhangi bir çözüm ortaya konmadığı için gerçek hayatta uygulama olanağı düşük olarak görülebilir.

Azmat vd. (2004), vardiya planlama ve çizelgeleme üzerine vaka çalışması yayınlamışlardır. Bu vaka çalışmalarında yıllık saat üzerinden tek vardiya planlaması yapılmıştır. Yıllık personelin çalışabilir saati çıkarılmış ve yıl içerisindeki resmi ve yıllık tatil günlerini kullanımı sezgisel metot ile planlanarak müşteri talebinin karşılanması için gerekli çalışma saati personele dağıtılmıştır.

Ernst vd. (2004) yayınladıkları çalışmalarında, personel planlama ve görevlendirme alanındaki modellere ve uygulama örneklerine yer verilmiştir.

Blöchliger (2004) yayınladığı makalesinde, bir sağlık kuruluşu için personel çizelgeleme problemi üzerine model geliştirmiştir.

Dellaert vd. (2004) çalışmalarında, çok aşamalı montaj hatları için stokastik talep altında çekme ve itme sistemleri için farklı senaryolar yazarak insan kaynağı teminini belirleyecek entegre bir model sunmuşlardır

Jaffry vd. (2005) çalışmalarında, personel planlamasında, tahmin riskleri için alternatif bir metot sunmuşlardır. Makalede kalitatif ve kantitatif yöntemlerle girdi ve çıktılar belirlenmiştir.

Gomes da Silva vd. (2006) çalışmalarında, toplu üretim planlama için karar destek sistemi kurmaya çalışmışlardır. Bu karar destek sisteminin temelinde karma tamsayılı doğrusal programlama kullanmışlardır.

Corominas vd. (2006) çalışmalarında, hizmet sektöründe çalışan personelin yeteneklerini göz önünde bulundurarak çoklu yeteneğe sahip personelin görevlere atanması konusunu ele almışlardır.

Thompson vd. (2006) makalelerinde, iş gücü planlamasında değişken personel verimliliği konusunu işlemişlerdir. Bu çalışmada personelin yeteneklerinin verimlilik üzerindeki etkisi incelenmiştir. Personelin yeteneğine uygun olmayan çalışma alanında çalıştırılmasının, verimlilik ve kalite açısından doğurabileceği sıkıntılardan söz edilmiştir.

Povh (2006), yayınladığı çalışmasında, lojistik alanında atama problemlerinin çözüm yöntemlerini detaylı bir şekilde sınıflandırmış ve açıklamıştır.

Nguyen vd. (2008), yılındaki çalışmalarında, önleyici bakım faaliyetlerinin planlanması ve fabrikadaki bakım faaliyetlerinin yürütülmesi, bakım kaynakları miktarı optimizasyonu için yeni bir formülasyon geliştirilmişlerdir.

De Castro vd. (2006), bakım kaynaklarının optimizasyonunu inceleyen bir çalışma yapmışlardır. Yazarlar bu çalışmalarında olarak genetik algoritmayı kullanmışlardır.

Yılman (2007) çalışmasında, toplu üretim planlama problemi konusunu ele almış ve hedef programlama yöntemi ile uygulamalı olarak çalışmıştır.

Ulusam Seçkiner vd. (2007) çalışmalarında, işçilerin haftalık çalışma çizelgeleri için tam sayılı matematiksel bir model önermişlerdir.

Wang vd. (2007) çalışmalarında, sipariş gecikmelerini en küçükmeyi amaçlayan bir model geliştirmişlerdir. Bulanık mantık ile siparişlere kaynak atamışlar ve genetik algoritma ile siparişlerin nasıl düzenlenmesi gerektiğini belirlemişlerdir.

Aktaş vd. (2007) dört adımdan oluşan sağlık hizmetlerinde kaynak atama verimliliği geliştirmeye yönelik bir karar destek sistemi önermişlerdir. Bu sistemin adımlarını aşağıdaki gibi sıralamışlardır. Birinci adımda sistem verimliliğini etkileyen temel faktörleri belirlemişlerdir. İkinci adımda, şartlı değişkenler ve kesin olmayan değişkenler için BBN (Bayesian Belief Networ) metodu uygulanmıştır. Üçüncü adımda, sisteme etki eden en kritik değişkenleri belirlemek amacıyla duyarlılık analizi yapılmıştır. Dördüncü adımda ise sistem verimliliğini geliştirmek için stratejiler önerilmiştir.

Eiselt vd. (2008) makalelerinde, personelin devamsızlık ve çalıştığı istasyonda memnuniyetsizliğini azaltmak amacıyla çalışanların pozisyonlarını belirleyip, iş yükünü

dağıtma konusunda çalışmışlardır. Çalışanların yetkinlik matrislerini kullanıp hedef programlama yöntemiyle çözüm aramışlardır.

Mohan (2008), çalışmasında tam zamanlı olmayan çalışmalarda çalışanların mesleki ihtiyaçlarını ve müşteri talebini karşılayacak şekilde bir planlama için matematiksel model tasarlanmıştır.

Fowler vd. (2008) makalelerinde, çalışan farklılıklarına göre iş gücü planlaması için sezgiseller başlığı altında, işe alma, işten çıkarma, iş gücü eğitim planı ve üretim kayıpları gibi parametre ve değişkenleri içeren tam sayılı programlama metodu önermişlerdir. Ayrıca bu metodun gerçek hayatta uygulanmasının çok zaman alacağı düşüncesiyle, genetik algoritma önerilmiştir.

Kabak vd. (2008), perakende sektörü için iki aşamalı vardiya planlama optimizasyon modelini konu alan bir makale yayınlamışlardır. Model iki aşamadan oluşmaktadır. Birinci aşamada saatlik kaynak ihtiyacı belirlenmiş, ikinci aşamada ise birinci aşamada sayısı belirlenen kaynakların vardiyalara kârını en büyükleyecek şekilde atanması amaçlanmıştır.

Lin vd. (2008) çalışmalarında, çok kriterli atama problemleri için çok amaçlı hibrit genetik algoritma yöntemini sunmuşlardır. Bir işletmenin on iki lokasyonunda çalışan yönetici ve mühendislerinin kârları ve maliyetleri baz alınarak optimum atama işlemleri gerçekleştirmişlerdir.

Çetinyokuş vd. (2008) çalışmalarında, işletmelerde performans değerlendirme sürecinde karşılaşılan en önemli sorun olan performans değerlendirme sürecinde kullanılacak faktörlerin belirlenmesi ve bu faktörlere bağlı olarak çalışan performanslarının objektif şekilde değerlendirilmesi ile ilgili bir karar destek sistemi sunmuşlardır.

Gülsün vd. (2009), doğrusal programlama temelli toplu üretim planlama strateji seçim metodolojisi alanında bir çalışma sunmuşlardır. Çalışmada altı strateji dikkate alınmış ve içlerinden en uygun planı seçmişlerdir. Çözüm metodu olarak doğrusal programlama kullanılmıştır.

Leung vd. (2008) çalışmalarında, toplu üretim planlaması konusunu kaynak kullanımı kısıtı ile hedef programlama modeli geliştirmişlerdir. Toplu üretim planlama probleminin içerdiği kısıtlar; üretim kapasitesi, işçi sayısı, fabrika yerleri, makine kullanımı, depolama alanı ve diğer kaynak kısıtlamalarıdır. Problemden toplam üç adet üretim tesis için aynı anda planlama yapması hedeflenmiştir. Bu çalışmada üç hedef belirlenmiştir. Bunlar: kusurlu parçalardan

kaynaklanan tamir maliyetinin enküçüklemesi, üç aylık kârın enbüyüklemesi ve bir üretim tesisindeki makine kullanım oranının enbüyüklemesidir.

Topaloğlu (2009), karma tamsayı programlama ile seviyelerine göre çalışanların tercihleri doğrultusunda istedikleri tarihlerdeki vardiyalara atamalarını sağlayan bir çalışma sunmuştur.

Valls vd. (2009) çalışmalarında, proje başlangıç zamanı, bitiş zamanı ve görevlerin aksaması durumunda ödenecek ceza gibi kısıtlar dikkate alınarak bir çalışma yayınlamışlardır. Yazarlar çalışmanın çözüm bölümünde genetik algoritma kullanmışlardır (Valls, vd. 2009).

Sabar vd. (2009) çalışmalarında, işçilerin montaj hatlarına günlük atama işlemlerinde, operasyonel maliyeti ve iş memnuniyetsizliğini enküçüklemek adına geliştirilen atama modelini sunmuşlardır.

Maenhout vd. (2010), genetik algoritma yöntemi ile havayolu endüstrisinde kişiselleştirilmiş mürettebat kadrosu için bir melez genetik algoritma araştırması yapmışlardır. Yazarlar çalışmalarında, mürettebat maliyetlerini en küçüklemeyi ve iş yükü ile ilgili tüm mürettebat üyelerine adil iş dağılımı sağlayan bir model geliştirmişlerdir.

Ho vd. (2010) çalışmalarında, tabu arama algoritması kullanılarak hava yolu işletmeleri için iş gücü planlaması yapmışlardır.

Carrasco (2010) çalışmasında, uzun dönemli personel planlaması konusunu işlemiştir. Önerilen model üç aşamadan oluşmaktadır. Birinci aşamada gerekli iş gücü hesaplanmakta, ikinci aşamada gerekli vardiya sayısı hesaplanmakta ve son aşama olan üçüncü aşamada ise sağlık kuruluşu çalışanları vardiya ve görevlere atamaktadır.

Avramidis vd. (2010) çalışmalarında, çağrı merkezi çalışanları için çoklu yetenekler göz önünde bulundurularak günlük görevlendirmeleri optimize edecek model geliştirmişlerdir.

Bayrak (2010) çalışmasında, hava hareket planı üzerine araştırmalar yapmıştır. Bu çalışmasında genetik algoritma önermiştir.

Hertz vd. (2010) çalışmalarında, çoklu vardiya çalışmalarında, esnek iş gücü planlaması yapabilmek için yıllık çalışma saatlerini baz alarak karma tam sayılı model önermişlerdir. Yıllık çalışan sayısı belirlemek için yıl içinde çalışma saati ve yıllık talep miktarı baz alınmıştır.

Mirzapour Al-E-Hashem vd. (2011) çalışmalarında, toplu üretim planlama modelinde, maliyetlerin en küçükleme denklemine çalışanların eğitim maliyetlerini de ilave ederek bir model tasarlamışlardır.

Sadjadi vd. (2011) çalışmalarında, işe alma, işten çıkarma, stok tutma ve talebi karşılayamama faktörlerini içeren bir stokastik model önermişlerdir. Çözüm metodu kıyaslama analizleri içeren bu çalışmada, üç farklı çözüm metodu birbiriyle kıyaslanmıştır. Bu çözüm metotları sırasıyla; Lingo 11.0'da doğrusal programlama modeli, C⁺⁺'ta genetik algoritma modeli ve arena programında simülasyon modelleridir. Çalışmalarının sonucunda Genetik algoritmanın, doğrusal programlamaya göre daha doğru sonuç verdiği simülasyon sonuçları yardımıyla da kıyaslanarak belirtilmiştir.

Lusa vd. (2011) çalışmalarında, maliyet unsurlarını baz alarak belirsizlik altında personel planlama alanında doğrusal programlama temelli uygulama senaryoları analizi yapılmıştır.

Hojati, vd, (2011) çalışmalarında, doğrusal programlama temelli sezgisel yöntemleri kullanarak yarı zamanlı çalışanlar için planlama ve çizelgeleme metodu önermişlerdir.

Brucker vd. (2011) çalışmalarında, personel planlama alanındaki modelleri ve bu modellerin karmaşık yapılarının araştırmışlardır.

Van Nieuwenhuyse vd. (2011) çalışmalarında, yöneticilerin çalışmalarda, satış ve pazarlama alanlarında karar alabilmeleri için değişken parametrelili ERP (Enterprise Resource Planning) sistemleri için ileri kaynak planlama için karar destek modeli önermişlerdir.

Ramezian vd. (2012) çalışmalarında, toplu üretim planlama alanında çift kademeli üretim sistemleri için model önermişlerdir. Bu çalışmada önerilen modelde çoklu dönem, çoklu ürün, çoklu makine sistemleriyle seri değişimleri de göz önünde bulundurarak iş gücü maliyeti ve envanter seviyesindeki dalgalanmaları enküçüklemeyi hedeflemektedirler. Kurulan modelin çözümünde tabu arama algoritması ve genetik algoritma denenmiştir. Çalışmanın sonucunda, iki farklı çözüm metodu karşılaştırılmış ve bu karşılaştırma sonucunda genetik algoritmanın daha kaliteli çözüm verdiği görülmüştür.

Lequy vd. (2012) çalışmalarında, iki aşamalı sezgisel bir model kullanarak çoklu aktivite ve vardiya atamaları için model önerilmiştir. Çalışmada karma tam sayılı programlama ile problem modeli kurulup ilk olarak görevlere atama yapılmış ve ikinci olarak da aktiviteye atama yapılmıştır.

Lin vd. (2012) çalışmalarında, bulanık mantık ve doğrusal programlama yaklaşımlarını kullanarak bir mağazanın müşteri servisinde çalışan işçilerinin doğru işlere atanması için bir model önermişlerdir. Çalışanlar yeteneklerine göre seviyelendirdikten sonra yetkinliklerine uygun işlere atamışlardır. Bu çalışmanın sonucunda atama modelinin çalışanların memnuniyeti ve verimliliği üzerinde olumlu etki yaratacağı gibi müşteri memnuniyetini de arttıracakını belirtmişlerdir.

Stuckless vd. (2012), Kanada'da on yıllık bir zaman dilimi içerisinde sağlık sektöründe radyoloji alanında öngörülen talep ile bu talebi karşılayabilecek arz hizmeti dengeleyebilmek amacıyla, iş gücü planlamasına olanak sağlayan bir model geliştirmeyi hedeflemişlerdir. Bu çalışmada geçmiş veriler göz önünde bulundurularak talep tahmini yapılmıştır. İki yönlü duyarlılık analizi ile arz ve talebe etki eden faktörler saptanmıştır.

Li vd. (2012) çalışmalarında atama modeli için genetik algoritma ile çözüm önermişlerdir.

Mesquita vd. (2013) çalışmalarında, daha önceden belirlenen tatil günlerini dikkate alarak, araç ve bu araçlarda çalışacak çalışanların haftalık atamalarını yapmak için doğrusal tamsayı temelli sezgisel model kullanmışlardır. Ulaşım sektöründe çalışanların ve müşterilerin memnuniyetinin artacağını belirtmişlerdir.

Maenhout vd. (2013), hemşirelerin atamalarının planlanması alanında bir çalışma yayınlamışlardır. Çalışmada personel memnuniyeti ve verim gibi kısıtların altında hemşirelerin uzun vadeli atama planlamalarını kapsayan bir model tasarlamışlardır.

Lapègue vd. (2013) çalışmalarında, ilaç sektöründe çalışanların görevlere atanması için kısıt programlaması kullanarak kısıtlara dayalı vardiya tasarımı ve personel görevlendirmesi alanında bir çalışma yayınlamışlardır.

Fernandes vd. (2013) çalışmalarında, farklı hizmet düzeylerinde vardiyalık işçilik planlaması için gerçek seçenekler üzerinden bir öneri geliştirmişlerdir.

Toledo vd. (2013), çok seviyeli sipariş büyüklüğü probleminin çözümü için yeni bir yaklaşım olan melez çoklu popülasyonlu genetik algoritma yaklaşımını sunmuşlardır. Modelde kaynak ve makinelere parametre olarak yer verilmiştir. Ayrıca önerilen yaklaşımın diğer çözüm yöntemleriyle karşılaştırılması yapılmış ve melez çoklu popülasyonlu genetik algoritma metodunun diğerlerine göre daha iyi sonuçlar verdiği sergilenmiştir.

Van den Bergh vd. (2013) çalışmalarında, önceki çalışmalar için bir literatür araştırması yapmışlardır. Yazarlar bu makalelerinde daha önce ilgili alanlarda yayınlanan çalışmaların sınıflandırılması ve alanlara göre analiz edilebilmesini kolaylaştırmışlardır. Detaylı inceleme için bu çalışma incelenebilir.

Smet vd. (2014) çalışmalarında, personel çizelgeleme problemini yeni bir melez yaklaşım ile çözmüşlerdir. Çoklu yetenekleri dikkate alarak iki aşamalı meta sezgisel yaklaşımdan söz etmişlerdir.

Harjunkoski, vd. (2014) çalışmalarında, üretim çizelgeleme modelleri ve çözüm metotlarını araştırmışlardır. Problem tipleri kendi aralarında sınıflandırılıp, çözüm metotlarından söz edilmiştir. Ayrıca gerçek hayatta var olan uygulamalar ile örnekler çözülmüştür. Yazarlar çalışmalarının son bölümünde özellikle otomotiv sektöründeki potansiyel problemler olan enerji, hammadde hakkındaki fikirlerine de yer vermişlerdir.

Wong vd. (2014) çalışmalarında, sağlık sektöründe çalışanların planlama problemi için iki aşamalı sezgisel bir yaklaşım sunmuşlardır. Bu çalışmanın birinci aşamasında tek vardiya atama sezgiseli ile kısıtların gerçekleştirilmesi sağlanmakta, ikinci aşamada ise hemşirelerin tercihleri göz önünde bulundurularak atama yapılmaktadır.

De Bruecker vd. (2015) çalışmalarında, yeteneklere göre iş gücü planlaması alanında literatür araştırmalarını yayınlamışlardır. Yazarlar bu çalışmalarında özellikle insanın yeteneği odaklı iş gücü planlaması üzerinde yoğunlaşarak, bu alanda çalışma yapan araştırmacılara yol göstermektedir.

Johnes (2014) çalışmasında, stokastik müşteri talebi ve vardiya sayısı ihtiyacı ve nakit akışını konu alan bir örnek ile hizmet düzeyleri ve bu düzeylerin geliştirilmeleri için öneriler sunmuştur.

Li vd. (2015) çalışmalarında, genetik algoritma ile ekipman, iş gücü ve servis planlama problemine bir çözüm önerisi sunmuşlardır.

Hidri vd. (2016), bir hastanenin üç bloğunda bulunan yoğun bakım servislerinde çalışan hemşireleri belirli gruplara ayırarak ihtiyaç duyulan hemşire sayısını belirleyen bir atama problemini çözmek için belirli kısıtlar kullanılarak bir model geliştirmişlerdir. Geliştirilen model tam sayılı programlama modelidir.

Varlı vd.(2017) çalışmalarında, günün her saatinde açık olan bir hastanenin yoğun bakım, acil ve ameliyathane servislerine atanan vardiyalarda ihtiyaç duyulan personel sayısını

karşılmak için bir model tasarlamışlardır. Bu modeli oluştururken hedef programlama yaklaşımını kullanmışlardır.

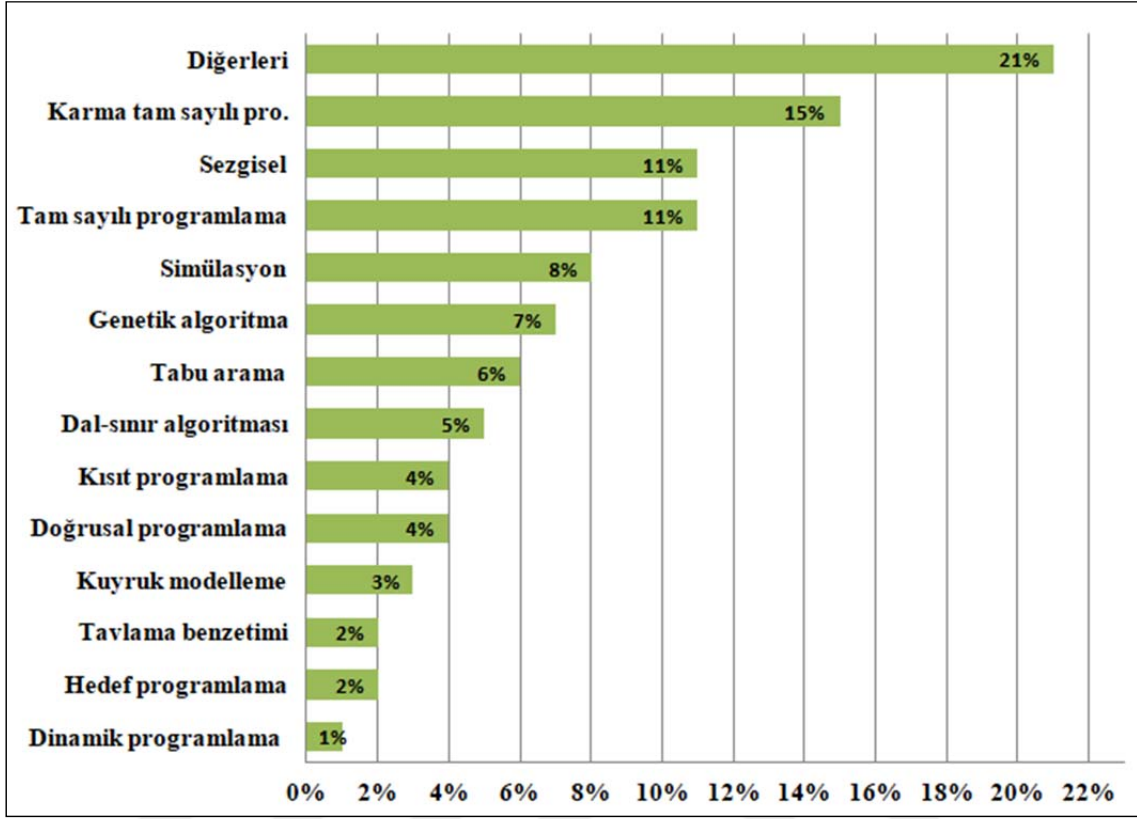
Polat vd (2018) çalışmalarında, herhangi bir istasyona bir işçinin atanması durumunda izin verilen en büyük iş yüklerinden sapmayı en küçükleyecek bir model geliştirmişlerdir.

Dolgui vd. (2018), belli başlı kısıtlar altında bir karma tam sayılı model önermişlerdir. Bu kısıtları; aynı anda kullanılan gerekli işçi sayısını en küçükleme, belirlenen montaj hattı çevrim süresinin aşılmaması ve her işlem için gerekli işçi sayısının belirlenmesidir. Bu kısıtları dikkate alarak istasyonlara atama yapılmasını amaçlayan bir model geliştirmişleridir.

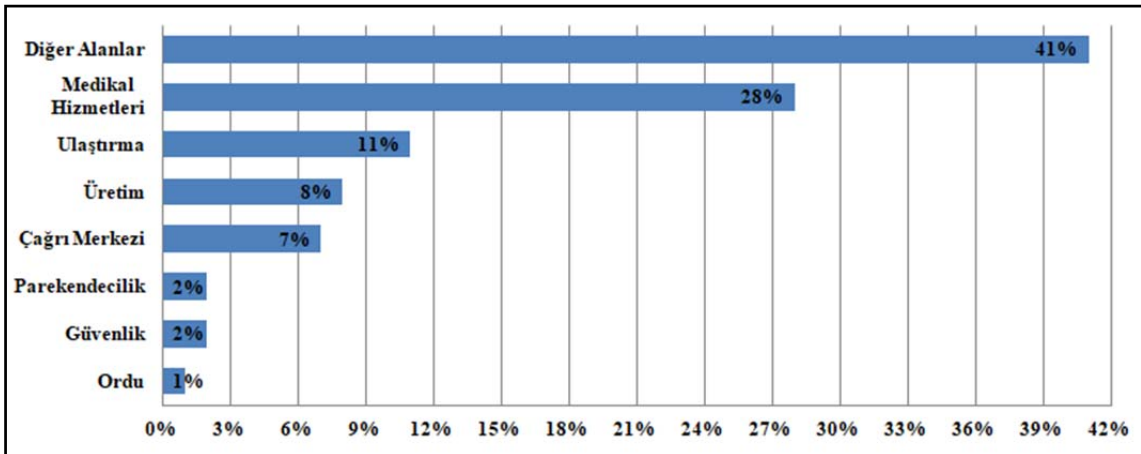
Son yıllarda hızla gelişen teknoloji ve sürekli değişen pazar koşulları yoğun bir rekabet ortamının oluşmasına neden olmuştur. Oluşan pazar şartlarında işletmeler ayakta kalabilmek, rekabeti sürdürebilmek ve kazancını arttırmak için üretim sistemlerinde maliyetleri düşürmek ve sınırlı olan kaynakları en iyi şekilde kullanmak gerekmektedir. Bu durum kaynak planlama alanında yapılan çalışmaların önemini ve sayısını artırmıştır.

Personel atama üzerine literatür araştırmaları analiz edildiğinde, çok sayıda çalışmada, tamsayılı programlama, doğrusal programlama, dinamik programlama ve hedef programlama gibi matematiksel programlama yaklaşımlarına rastlanmaktadır.

Literatür araştırmasında atama ile ilgili çalışmaların yöntem ve uygulama alanları açısından sınıflandırılması ve bu sınıflandırmanın yüzde bilgileri sırasıyla Şekil 2.1 ve Şekil 2.2'de verilmiştir.



Şekil 2.1. Çözüm yöntemlerine göre çalışma yüzdeleri.



Şekil 2.2. Uygulama alanlarına göre çalışma yüzdeleri.

3. PERSONEL DEVAMSIZLIĞI PROBLEMİ VE ÖNERİLEN MODEL

Bir otomotiv yan sanayi işletmesinde günlük işe gelmeyen personelin yerine uygun personelin atanması ve gerekli işe iş yüklerinin dağıtılması problemi için bir matematiksel model önerilmiştir. Geliştirilen bu matematiksel model, işletmede çalışan bir üretim hattının personel verileri üzerinde kullanılmıştır.

İşe gelmeyen personelin yerine, çalışabilecek yetkinlikte bir personelin atanması gerekmektedir. Bu amaçla işletme, yetkin personelden oluşan bir ekip oluşturmuştur. Ekip, joker personel olarak adlandırılan personelden oluşmaktadır. Joker personel atandığı istasyonda en az verim kaybı ile işlemleri yerine getirmekle yükümlüdür. Yeterli joker personel olmadığı durumlarda ise işe gelmeyen istasyona en yakın iki istasyona ilgili istasyonun iş yükü iş yüklerine bağlı olarak dağıtılmaktadır.

3.1. Matematiksel Model

Joker personelinin yetkinlikleri baz alınarak işe gelmeyen personelin yerine joker personelin atanması ve iş yüklerinin dağıtılması için matematiksel model Eşitlik (3.1) ile (3.17) arasında verilmiştir.

İndisler

$i = \{i | i = 1, 2, 3, \dots, 10\}$ istasyon indisi

$j = \{j | j = 1, 2, 3, \dots, 5\}$ joker personel indisi

$k = \{k | k = 1, 2, 3, \dots, 6\}$ Süreç indisi

Parametreler

$J =$ Joker personel sayısı, $J \in \mathbb{N}$

$t_i =$ i. istasyonunun çevrim süresi

$q_{jk} =$ j. jokerin k süreçteki puanı

$S =$ Eksik operatör sayısı, $S \in \mathbb{N}$

$A_{ji} =$ j. jokerin i. istasyona atanması sonucu i. istasyonun yeni çevrim süresi

$F_{i+1}=(i+1)$. istasyonun, işe gelmeyen i . istasyonun personelinin iş yükünün K_{i+1} katsayısı oranında dağıtılmasıyla oluşan yeni çevrim süresi

$F_{i-1}=(i-1)$. istasyonun, işe gelmeyen i . istasyonun personelinin iş yükünün K_{i-1} katsayısı oranında dağıtılmasıyla oluşan yeni çevrim süresi

Karar değişkenleri

$$x_{ji} = \begin{cases} 1, & j. \text{ Joker personelin } i. \text{ istasyon atanması,} \\ 0, & \text{diğer durumlar} \end{cases}$$

$$C_i = \begin{cases} 1, & i. \text{ istasyonun işe gelmemesi} \\ 0, & \text{diğer durumlar} \end{cases}$$

$$Z_i = \begin{cases} 1, & i. \text{ istasyonuna joker personel atanması} \\ 0, & \text{diğer durumlar} \end{cases}$$

Amaç fonksiyonu

$$\sum_{j=1}^5 \sum_{i=1}^5 \text{Min}Z = [C_i * |(t_i - A_{ji})|] + (1 - Z_i) * (C_i * F_{i,i+1} + C_i * F_{i,i-1}) \quad (3.1)$$

Kısıtlar

$$\sum_{j=1}^J C_i * X_{ji} \leq 1, \quad \forall i, \quad j = 1, 2, \dots, J, \quad (3.2)$$

$$\sum_{i=1}^m C_i * X_{ji} \leq 1, \quad \forall j, \quad j = 1, 2, \dots, m \quad (3.3)$$

$$\sum_{i=1}^S C_i * X_{ji} \leq \text{Min}(S, J), \quad \forall j, \quad i = 1, 2, \dots, s \quad (3.4)$$

$$1 - \sum_{j=1}^J X_{j1} \leq Z_1 * M, \quad \forall j \quad (3.5)$$

$$1 - \sum_{j=1}^J X_{j2} \leq Z_2 * M, \quad \forall j \quad (3.6)$$

$$1 - \sum_{j=1}^J X_{j3} \leq Z_3 * M, \quad \forall j \quad (3.7)$$

$$1 - \sum_{j=1}^J X_{j4} \leq Z_4 * M, \quad \forall j \quad (3.8)$$

$$1 - \sum_{j=1}^J X_{j5} \leq Z_5 * M, \quad \forall j \quad (3.9)$$

$$\sum_{j=1}^J X_{ji} \leq \sum_{i=1}^S (1 - Z_i) * C_i, \quad \forall j \quad (3.10)$$

$$\frac{t_{i-1}}{(t_{i-1}+t_{i+1})} = K_{i+1} \quad (3.11)$$

$$\frac{t_{i+1}}{(t_{i-1}+t_{i+1})} = K_{i-1} \quad (3.12)$$

$$Z_i * t_i * (K_{i+1}) + t_{i+1} = F_{i,i+1} \quad (3.13)$$

$$Z_i * t_i * (K_{i-1}) + t_{i-1} = F_{i,i-1} \quad (3.14)$$

$$x_{ji} * (t_i * q_{jk}) = A_{ji} \quad (3.15)$$

$$\sum_{j=1}^J X_{ji} \leq J \quad (3.16)$$

$$M \gg 0 \quad (3.17)$$

$X_{ji}, Z_i, C_i, 0-1$ tam sayı

Eşitlik (3.1), amaç fonksiyonunu göstermektedir. Amaç fonksiyonunda eğer yeterli joker personel var ise, işe gelmeyen istasyona en uygun joker personeli atanarak standart çevrim süresi ile joker personel atandığında oluşan yeni çevrim süresi arasındaki farkın en küçüklemesi hedeflenmektedir. Eğer yeterli joker personel yok ise, işe gelmeyen i. istasyonun iş yükü, kendisine en yakın iki istasyona, Eşitlik (3.11) ve (3.12)'de verilen K_{i+1} ve K_{i-1} dağıtım katsayılarına göre dağıtılmaktadır. Böylece iş yükü fazla olan istasyona az, iş yükü az olan istasyona da çok iş dağıtılmaktadır. Kısıt (3.2)- (3.3) sırası ile bir istasyona sadece bir joker personel atanabileceğini ve bir joker personelin sadece bir istasyona atanabileceğini göstermektedir. Kısıt (3.4) bir ekibe en fazla o ekibin eksik istasyonu kadar joker personel atanabileceğini göstermektedir. Ayrıca kısıt (3.4) yeterli joker personel olmaması durumunda, mevcut olan joker personelin atanmasını sağlamaktadır. (3.5)-(3.6)-(3.7)-(3.8) ve (3.9) numaralı kısıtlar, Z_i karar değişkeninin alabileceği değerlerin kısıtlarını göstermektedir. Kısıt (3.10), işe gelmeyen istasyonlara joker personel sayısı kadar personel atanabileceğini göstermektedir. Kısıt (3.11) ve (3.12) sırasıyla i. istasyonun işe gelmemesi durumunda, (i+1). ve (i-1). İstasyonlarının dağıtım katsayılarını göstermektedir. Kısıt (3.13) ve (3.14) sırası ile işe gelmeyen i. istasyonunun iş yükünün (i+1). ve (i-1). İstasyona dağıtılması durumunda bu istasyonların yeni çevrim sürelerini göstermektedir. Kısıt (3.15), işe gelmeyen i. istasyona j. joker personelin atanması durumunda i. istasyonun yeni çevrim süresini göstermektedir. Kısıt (3.16) atama olabilmesi için işe gelmeyen personelin sayısının joker personel sayısından küçük veya eşit olması gerektiğini göstermektedir. Kısıt (3.17), M sayısının çok büyük bir sayı olduğunu göstermektedir.

Matematiksel modelin günlük çalışmada karşılaşılabilecek farklı senaryolarına göre Lingo 11.0 programında yazılan modelleri ve bu modellerin çözümleri Ek 1’de verilmiştir.

3.2. Matematiksel Model için Uygulama

Uygulama için işletmede otomobillerin kapı tesisatının montajının yapıldığı üretim hattında günlük işe gelmeyen personelin verileri kullanılmıştır. İşe gelmeyen personelin yerine uygun joker personelin atanması ve iş yükünün dağıtılması için önerilen matematiksel model çözülmüştür. Ekibin süreçleri sırasıyla aşağıdaki gibidir:

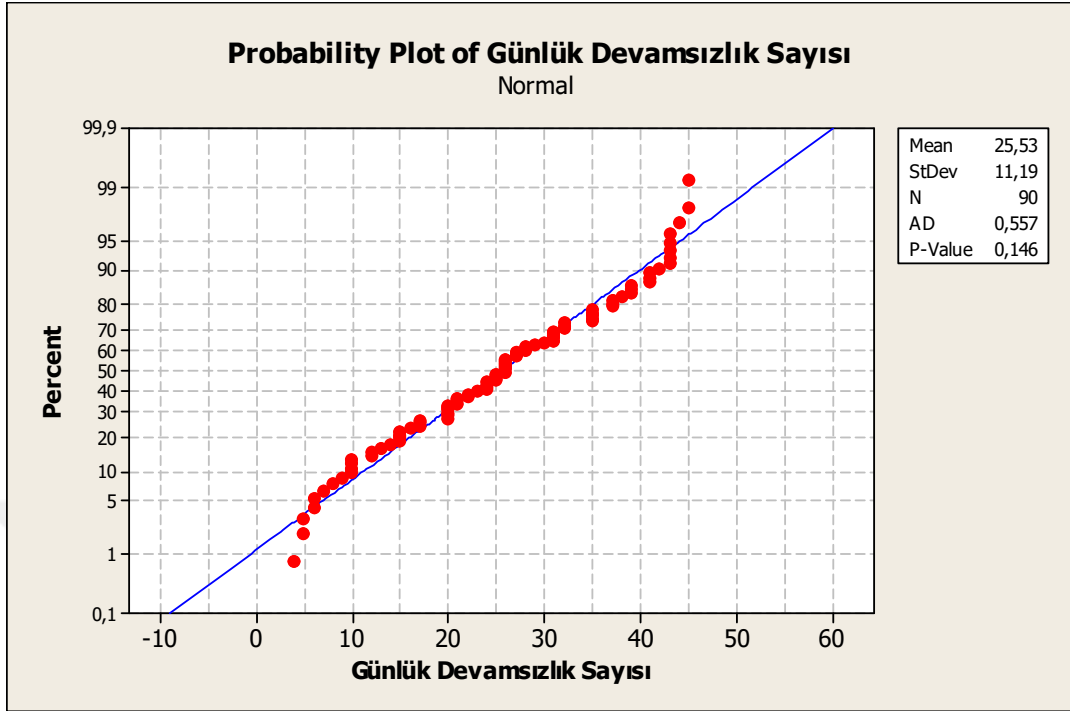
- Modül hazırlama işlemi,
- Montaj işlemi,
- Klip test süreci,
- Elektriksel test süreci,
- Göz kontrol ve paketleme sürecidir.

Üretim hattında her süreç için bir istasyon bulunmaktadır.

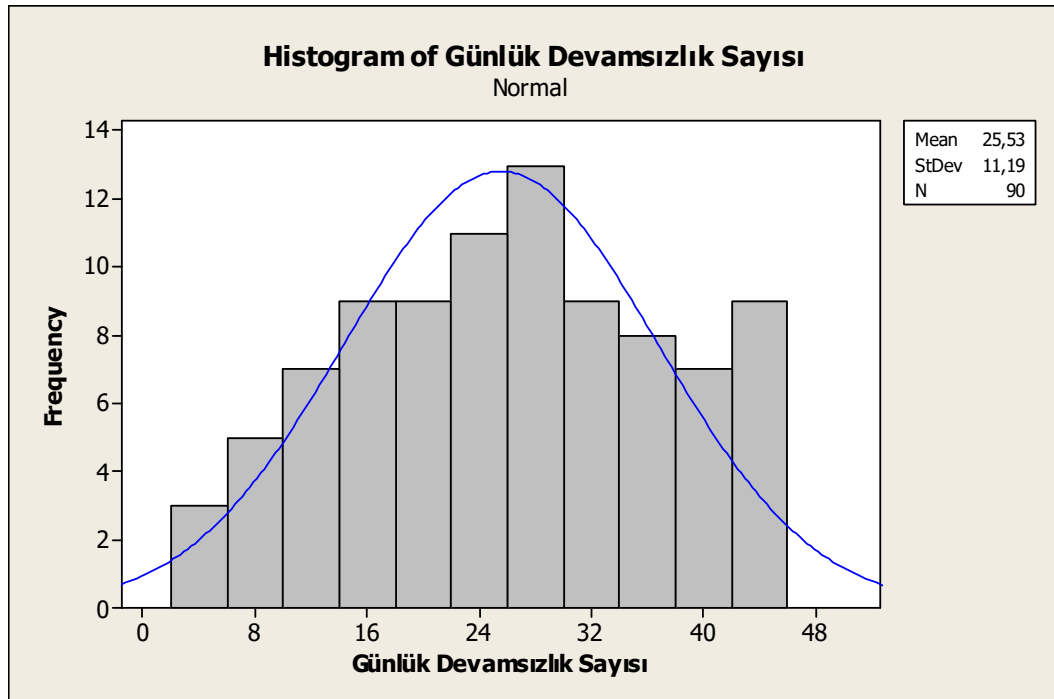
3.3. Joker Personel Sayısının Belirlenmesi ve Süreçlere Göre Yetkinlik Matrisi

İşletmenin 6 aylık devamsızlık verileri incelenerek yılın kalan 3 ayı için gerekli joker personel sayısı belirlenmektedir. İşletmenin doksan gün devamsızlık verileri MINITAB 16 programında analiz edildiğinde Şekil 3.1’de gösterilen saçılım grafiğinde görüldüğü gibi p değeri $\alpha=0,05$ ’ten büyük olduğu için veriler normal dağılım göstermektedir. İşletmenin 90 günlük devamsızlık verilerinin histogram grafiğinde de verilerin normal dağılım gösterdiği açıkça görülmektedir.

Analiz edilen veriler normal dağılım gösterdiği için bu verilerin ortalaması, ihtiyaç duyulan joker personel sayısının belirlenmesinde kullanılmıştır.



Şekil 3.1. Doksan günlük devamsızlık yapan personelin saçılım diyagramı.



Şekil 3.2. Doksan günlük devamsızlık yapan personelin histogram diyagramı.

Joker personel, yılın belirli periyotlarında düzenlenen sınavlara tabi tutularak seçilmektedir. İşletmenin üretimini yaptığı otomobilin kapı tesisatı projesinde çalışabilecek joker personelin süreç bazlı yetkinlik matrisi Çizelge 3.1’de gösterildiği gibi oluşturulmaktadır. Bu matris joker personelin süreç bazlı yetkinliklerini içermektedir.

Çizelge 3.1. Kapı tesisatı projesinde çalışabilecek joker personelin süreç bazlı yetkinlik puanları.

Joker personel no		Süreç no	1	2	3	4	5
		İstasyon no	1	1	1	1	1
Kapı tesisatı projesinde çalışabilecek joker personel ekibi	1	Ahmet Uslu	0,8	0,6	0,8	0,3	0,1
	2	Ayşe Kül	0,7	0,8	0,2	0,3	0,5
	3	Merve Arı	0,9	0,7	0,2	0,2	0,4
	4	Sevgi Çetin	0,8	0,5	0,3	0,1	0,3
	5	Hasibe Şen	0,5	0,8	0,3	0,1	0,1

Çizelge 3.2’de süreçlerde bulunan istasyonların çevrim süreleri gösterilmiştir. Bu süreler zaman analizleri çalışmaları sonucunda hesaplanan standart sürelerdir.

Çizelge 3.2. Kapı tesisatı üretim hattı istasyonlarının çevrim süreleri.

No	Süreç adı	Çevrim Süresi (sn)
1	Modül hazırlama istasyonu	162
2	Montaj istasyonu	162
3	Klip test süreci	138
4	Elektriksel test süreci	144
5	Göz kontrol ve paketlenme süreci	144

3.4. Modelin Çözümü

Oluşturulan matematiksel model işletmede üretilen otomobilin kapı tesisatının üretim hattında gerçekleşen üç farklı senaryoya göre çözülmüştür. Bu senaryoların modelleri ve çözümleri Ek 1’de verilmiş olup, sonuçların özeti Çizelge 3.3’de gösterilmektedir. Bu

senaryoların çözümleri için modelde sadece joker personel sayısı (J) ve işe gelmeyen personel sayısı (S) parametreleri değişmektedir.

Birinci senaryonun parametre değerleri aşağıda verilmiştir:

- ✓ Joker personel sayısı (J): 3,
- ✓ İşe gelmeyen personel sayısı (S): 2.

Bu senaryoya göre joker personel sayısı, işe gelmeyen personel sayısından daha fazla olduğu için sadece atama işlemi gerçekleşmektedir. Uygun olan joker personel işe gelmeyen personelin yerine atanmaktadır.

İkinci senaryonun parametre değerleri aşağıda verilmiştir:

- ✓ Joker personel sayısı (J): 1,
- ✓ İşe gelmeyen personel sayısı (S): 2.

Bu senaryoya göre joker personel sayısı işe gelmeyen personel sayısından daha az olduğu için, hem atama hem de dağıtma işlemi gerçekleşmektedir. Bu senaryoda mevcut durumda var olan joker personel uygun olan iş istasyonuna atandıktan sonra, joker personel atanamayan istasyonun iş yükü, kendisine en yakın iki istasyona dağıtılmaktadır. Bu dağıtım işlemi, yeni iş yükü alacak istasyonların çevrim sürelerine bağlı olarak dağıtılmaktadır. Böylece çevrim süresi az olan iş istasyonuna daha fazla iş yükü dağıtılmaktadır.

Üçüncü senaryonun parametre değerleri şu şekildedir,

- ✓ Joker personel sayısı (J) 2,
- ✓ İşe gelmeyen personel sayısı (S) 2,

Bu senaryoya göre joker personel sayısı ile işe gelmeyen personel sayısı birbirine eşittir. Dolayısıyla sadece atama işlemi gerçekleşmektedir. Uygun olan joker personel işe gelmeyen personelin yerine atanmaktadır.

Çizelge 3.3. Üç farklı senaryo sonuçları.

	İşe gelen joker personel no	İşe gelmeyen personelin istasyon no	İşe gelmeyen personelin istasyonunun çevrim süresi (sn)	Atanan joker personel no	Yeni çevrim süresi (sn)	İşe gelmeyen personelin istasyonuna en yakın birinci istasyonun çevrim süresi (sn)	İşe gelmeyen personelin istasyonuna en yakın ikinci istasyonun çevrim süresi (sn)
Senaryo 1	1,4,5	3	138	1	220	-	-
		5	144	4	216	-	-
		-	-	-	-	-	-
Senaryo 2	4	3	138	-	-	226	217
		5	144	4	216	-	-
Senaryo 3	1,4	3	138	1	220	-	-
		5	144	4	216	-	-

Matematiksel modelde işletmeye ait tek bir projede çalışan bir ekip ve bu projede çalışabilecek joker personel ele alınmıştır. Normal koşullarda işletme bünyesinde 18 farklı proje ve bu projelerde çalışan 36 personel ekibi mevcuttur. Matematiksel modelin bütün ekipleri kapsayacak şekilde oluşturulması problemin çözümünü zorlaştırmakta ve Lingo 11.0 programında çözümü yaklaşık her çözüm için 10 dk gibi bir süre gerekmektedir. Bu işlem her devamsızlık olması durumunda yapıldığı takdirde işletme için ciddi bir zaman kaybı demektir. Bu nedenle bir KDS'ye ihtiyaç duyulmaktadır.

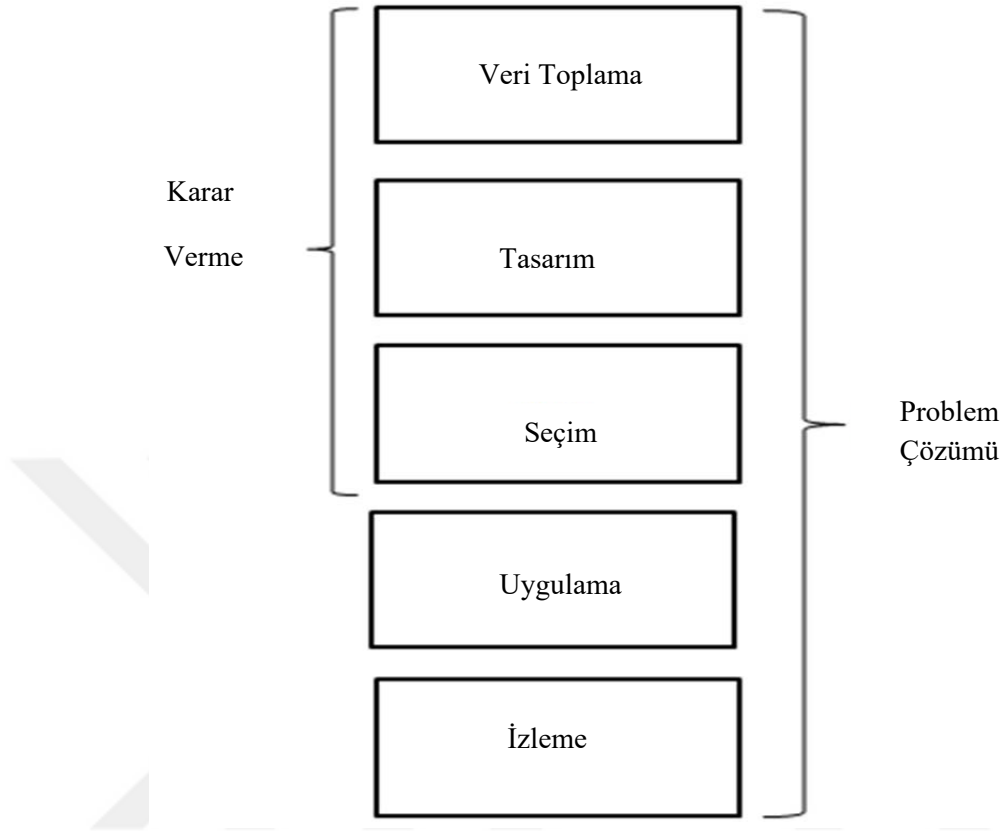
4. KARAR DESTEK SİSTEMLERİ

Karar Destek Sistemleri, kullanıcılara yardımcı olmak amacıyla veri, belge, bilgi ve iletişim teknolojilerini ve/veya modelleri kullanarak problemleri tanımlamayı ve çözmeyi, karar verme sürecini tamamlamayı ve karar vermeyi sağlayan interaktif bilgisayar sistemleridir.

Kullanıcı bazı durumlarda, kendi bilgi ve birikimine güvenerek karar verme durumunda kalabilir ya da Yönetim Bilgi Sisteminden (YBS) elde edilen bilgilerden başka ilave bilgiye bakmaya gerek duymaksızın karar verebilir. Özellikle stratejik anlamada önemli olan kararlarda kullanıcılar, sentez edilmesi zor ve karmaşık olan verileri tam olarak analiz edebilme ve daha sonra karar verebilme gibi insan yeteneğinin ötesinde olan zor durumlarla karşılaşır. Bu durumlar ile karşılaşan kullanıcıların KDS kullanmaları uygundur.

KDS'ler kullanıcılara tüm karar aşamaları, karar tipleri ve farklı yapıdaki problemlerle ilgilenebilme olanağı sağlamaktadır (Çetinyokuş, Gökçen, 2008).

Karar verme aşamasına kadar toplanan verilerin artışı, karar verme süreçlerinin karmaşık olması, insan beyninin sınırlı olması göz önünde bulundurulduğunda yazılım kullanımı zorunlu hale gelmektedir. KDS kapsamında karar verme süreci Şekil 4.1' de gösterilmiştir.



Şekil 4.1. KDS'lerde karar verme süreci.

Şekil 4.1'de görüldüğü gibi süreç karar verme, verilen kararın uygulanması ve izlenmesi şeklinde gerçekleşmektedir. KDS'lerde yazılım, temelde verilerin saklandığı veritabanı ve bu verileri işleyen algoritma ve formülleri çalıştıran bileşenlerden oluşmaktadır. Veri miktarının artması ve karmaşık hesaplar yapılması durumunda, donanımda önemli bir bileşen haline gelmektedir.

5. PERSONEL DEVAMSIZLIĞINDA ATAMA VE DENGELEME PROBLEMİ İÇİN KARAR DESTEK SİSTEMİ

Matematiksel modelin çözümünün zaman alması ve günlük ya da anlık kullanmaya uygun olmaması nedeniyle, işletmedeki değişikliklere kolay ve hızlı uyum sağlayabilmek için bir KDS tasarlanmıştır. Bu sistemi çalıştırabilmek için joker personel atama ve dağıtma amaçlı sezgisel algoritması adına bir sezgisel algoritma önerilmiştir. Bu algoritmanın adımları aşağıda verilmiştir:

Adım 1: Günlük işe gelmeyen personelin bilgisini sisteme gir.

Adım 2: Yeterli joker personel sayısını kontrol et, eğer yeterli joker personel varsa Adım 3 ile devam et, eğer yeterli joker personel yoksa Adım 5'e git.

Adım 3: Var olan joker personelin herhangi bir istasyona bağlı olup olmadığını kontrol et.

Adım 4: İstasyonlara bağlı olmayan joker personel arasından, devamsızlık bilgisi girilen personelin yerine en uygun olan joker personelin atamasını yap ve Adım 1'e geri dön.

Adım 5: Eğer Adım 4'te atanacak joker personel kalmadıysa, işe gelmeyen personelin çalıştığı istasyonun çevrim süresini kendisine en yakın iki istasyona Kısıt (3.11) ve (3.12)'deki katsayıları dikkate alarak dağıt.

Adım 6: İşe gelmeyen personel kaldıysa Adım 1'e dön. Değilse, sonlandır.

İşletme yöneticileri, geliştirilen sezgisel algoritmayla işe gelmeyen personelin yerine en uygun joker personelin atamasına veya atanacak joker personelin olmaması durumunda işe gelmeyen personelin iş yükünün dağıtılmasına karar vermektedir. Böylece işletme yöneticileri, işe gelmeyen personelin çalıştığı istasyona uygun joker personel atama veya işe gelmeyen personelin çalıştığı istasyonun iş yükünü dağıtma işlemini sistematik bir şekilde yapabilmektedir. Dolayısıyla oluşabilecek kaybı en küçüklemeyi hedeflemektedirler. Geliştirilen bu algoritma C# programlama dili kullanılarak kodlanmış ve kullanıcıya kolaylık sağlaması açısından aynı programlama dili kullanılarak kullanıcı arayüzleri oluşturulmuştur.

KDS için kullanılan yazılım dilleri

C# dili

C#, Microsoft'un geliřtirmiř olduėu yeni nesil programlama dilidir. Yine Microsoft tarafından .NET Teknolojisi iin geliřtirilmiř dillerden biridir. Microsoft tarafından geliřtirilmiř olsa da ECMA ve ISO standartları altına alınmıřtır.

Tasarım hedefleri

ECMA tarafından C# dilinin tasarım hedefleri, C# basit, modern, genel amalı, nesneye ynelik programlama dili olarak tasarlanmıřtır. ünkü yazılımın saėlamlıėı, gvenirliėi ve programcıların retkenliėi nemlidir. C# yazılım dili, gl tiplene kontrol (strong type checking), dizin sınırlar kontrol (array bounds checking), tanımlanmamıř deėiřkenlerin kullanım tespiti, (source code portability), ve otomatik artık veri toplama gibi zelliklerine sahiptir. Programcı portatifliėi zellikle C ve C++ dilleri ile tecrbesi olanlar iin ok nemlidir.

Microsoft SQL server

Microsoft SQL Server en ok kullanılan veritabanı sunucu yazılımıdır. Veritabanlarının oluřturulmasını ve ynetilmesini saėlayan kurumsal aplı Veritabanı Ynetim Sistemidir. Dnyada en ok kullanılan ynetim sistemi SQL Server'dır. SQL Server'ı kullanarak verilerinizi dilediėiniz Őekilde ynetebilir ve Stored prosedrleri kullanarak ok sayıda ve komplike sonular dndrebilirsiniz. Bylece istediėiniz verileri raporlayarak elde edebilmektedir.

SQL dili

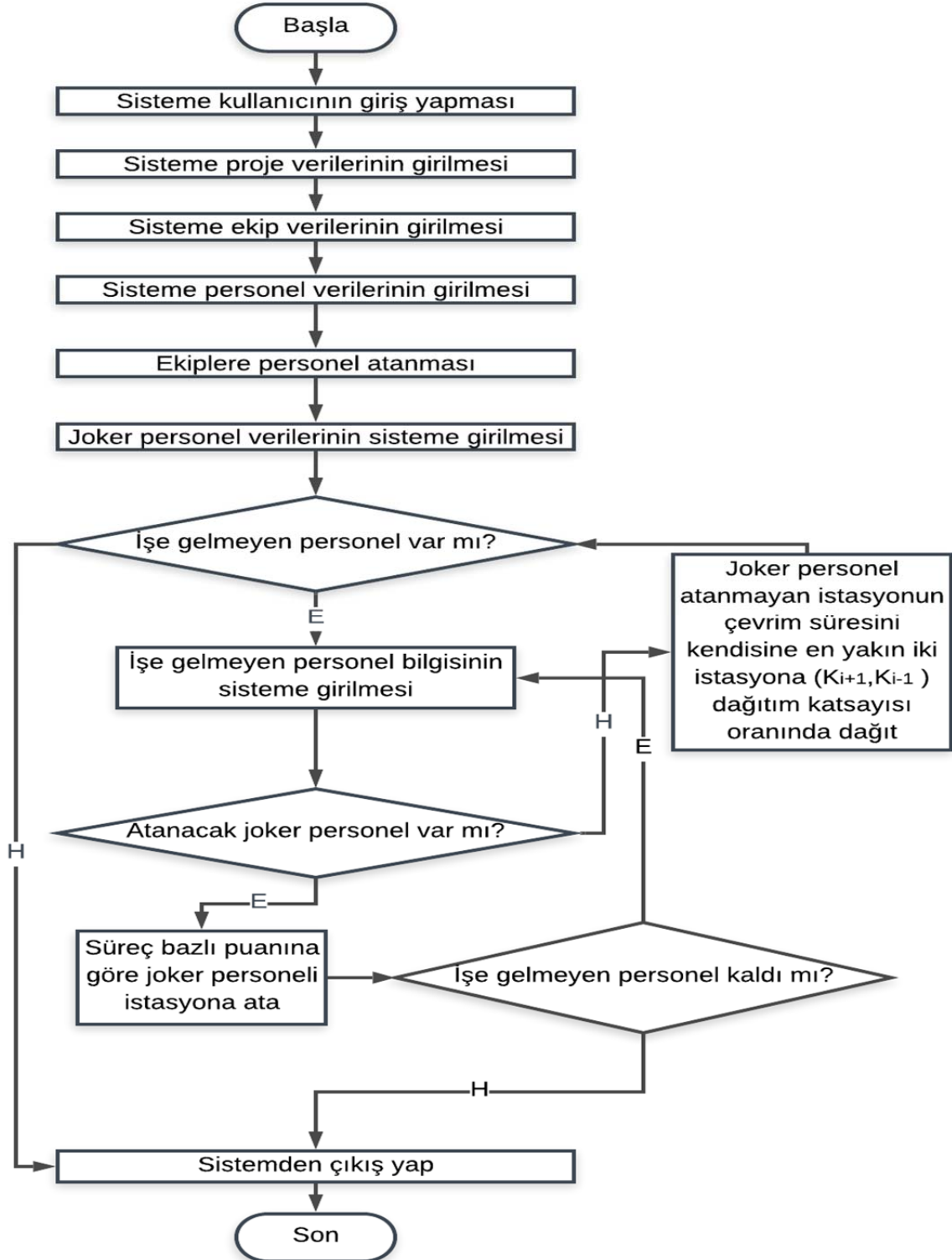
SQL (Structured Query Language), iliřkisel veritabanı ynetim sistemlerinden veri almak, veritabanında bulunan veriyi dzenlemek veya sisteme veri giriři yapmak iin kullanılan en popler sorgulama dilidir. SQL temelde, nesne-iliřkili (object-relational) veritabanı ynetim sistemlerini desteklemek iin tasarlanmıřtır. Fakat bu amacın tesinde, ANSI ve ISO standartları tarafından belirlenmiř, birok zelliėe sahip olduėu grlmektedir.

5.1. Karar Destek Sisteminin Akıř Őeması

Tasarlanan KDS'nin karar verme srecinin akıř Őeması Őekil 5.1'de verilmiřtir. Bu akıř Őeması KDS'nin karar verme mekanizmasını gstermektedir. Karar verme mekanizması tamamen kullanıcının kendi kararı dıřında ve sistematik bir Őekilde gerekleřmektedir. Karar verme ařamasında btn kararlarda insan faktrnn olmaması sistemin srekliėini ve tarafsız karar vermesini saėlamaktadır.

Şekil 5.1’de gösterilen personel yokluğunda atama KDS akış şemasında işe gelmeyen personelin istasyonuna, ilgili süreçte yetkin joker atanmaktadır. Eğer atanacak joker personel yoksa işe gelmeyen personelin iş yükü kendisine en yakın iki istasyona dağıtılmaktadır. İş yükü dağıtma işlemi dağıtım katsayıları (K_{i+1}, K_{i-1}) oranında yapılarak, dağıtma sonucu oluşacak çevrim süresinin en küçüklemesi amaçlanmaktadır.






Şekil 5.1. Personel devamsızlığında atama ve dengeleme problemi için karar destek sistemi akış şeması.

5.2. Kullanıcı Arayüzleri ve Tanımları

KDS’de kullanılan arayüzler sisteme veri girişi ve sistemden veri çıkışı için kullanılmaktadır.

Sisteme giriş arayüzü

Bu arayüz KDS’nin ilk arayüzüdür. Bu arayüzlerin açılması için yetkili personel, kullanıcı adını (ID) ve şifresini Şekil 5.2’de gösterilen arayüze girmelidir.

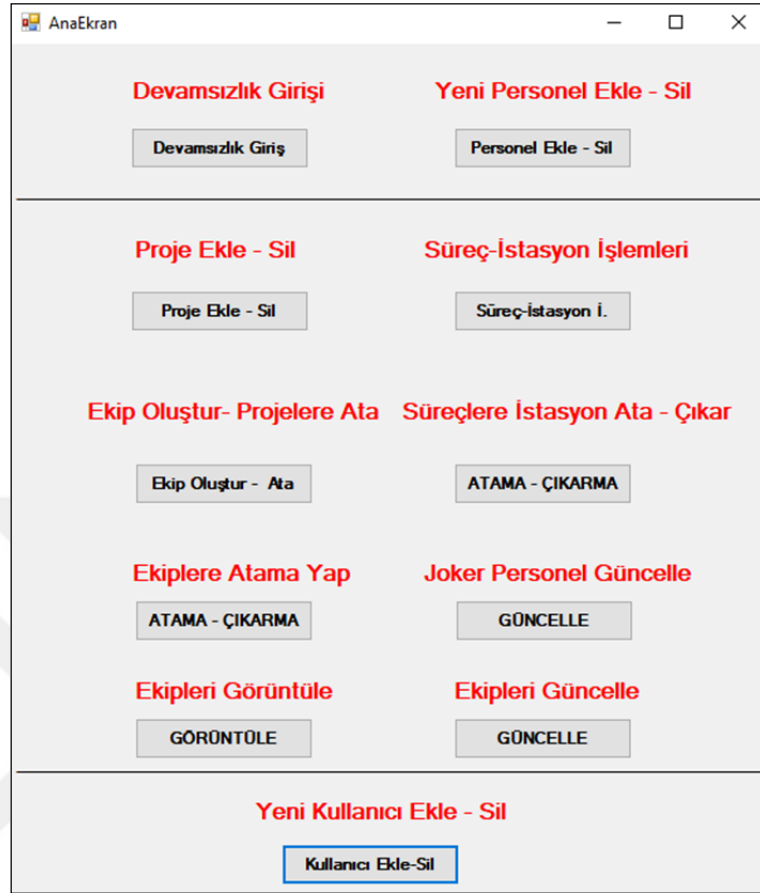


The image shows a Windows-style window titled "Giriş" (Login). Inside the window, there are two text input fields. The first is labeled "Kullanıcı Adı" (Username) and the second is labeled "Şifre" (Password). Below these fields are two buttons: "GİRİŞ" (Login) and "İPTAL" (Cancel). A red arrow points to the "GİRİŞ" button.

Şekil 5.2. Giriş arayüzü.

Ana ekran arayüzü

Kullanıcı, KDS’de Şekil 5.2’de gösterilen sisteme giriş arayüzüne kullanıcı adı ve şifresi ile giriş yaparak” Giriş” butonunu tıkladığında Şekil 5.3’de gösterilen ana ekran arayüzü açılmaktadır.



Şekil 5.3. Ana menü arayüzü.

Şekil 5.3’de gösterilen ara yüz kullanıcı açısından çok önemlidir. Kullanıcı sistemde bir değişiklik yapmak istiyorsa bu arayüzde ilgili değişikliğin butonunu tıklayarak ilgili değişikliğin yapılacağı arayüze yönlendirilmektedir. Kullanıcının bu arayüz ile yapabileceği işlemler aşağıdaki gibidir:

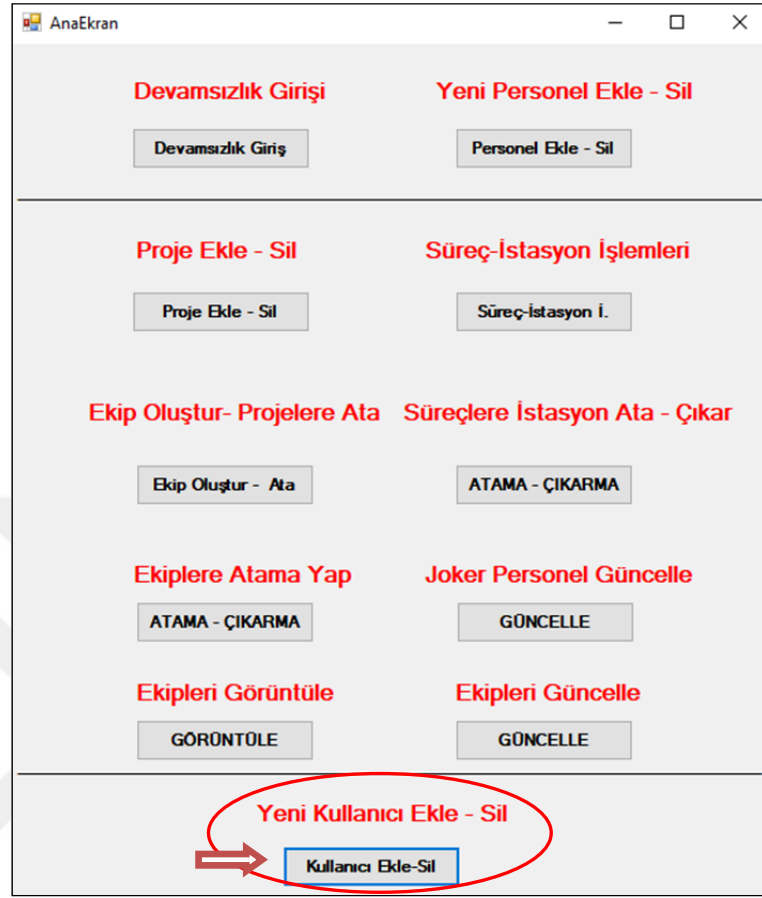
- “Devamsızlık girişi” bölümünden işe gelmeyen personelin verilerinin sisteme girişi,
- “Yeni personel ekle-sil” bölümünden sisteme yeni personel verilerinin girişi veya sistemden personel bilgisinin silinmesi,
- “Proje ekle-sil” bölümünden sisteme yeni projenin eklenmesi veya kayıtlı olan projenin sistemden silinmesi,
- Süreç-istasyon işlemleri” bölümünden sisteme kayıtlı olan süreçlere istasyon ataması veya süreçten istasyon çıkarılması,

- “Ekip oluştur-projelere ata” bölümünden sistemde yeni ekip oluşturulması ve bu ekipleri istemede kayıtlı olan projelere ataması,
- “Süreçlere istasyon ata-çıkar” bölümünden süreçlere istasyon ataması veya atanmış istasyonları çıkarılması,
- “Ekiplere atama yap” bölümünden ekiplere personel atanması,
- “Joker personel güncelle” bölümünden joker personel kadrosunun düzenlenmesi,
- “Ekipleri görüntüle” bölümünden ekiplerin kadrosunun görüntülenmesi,
- “Ekipleri günceller” bölümünden ekiplere atanan joker personelin çıkarılıp asıl kadrolarının güncellenmesi,
- “Yeni kullanıcı ekle-sil” bölümünden sisteme kullanıcı ekleme veya kayıtlı kullanıcı silme işleminin yapılmasıdır.

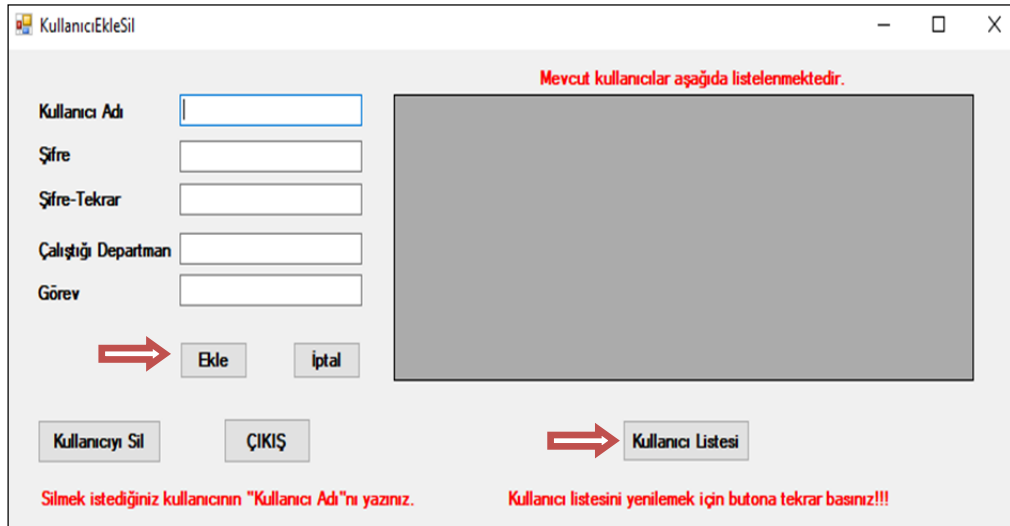
Kullanıcının bu arayüzdeki bölümler ile yönlendirileceği arayüzler ve açıklamaları aşağıdaki alt başlıklarda anlatılmaktadır.

Kullanıcı ekleme silme ve güncelleme arayüzü

Kullanıcı ekleme, silme arayüzünde sisteme kullanıcı ekleme, sistemin kullanıcı bilgilerini güncelleme ve kullanıcı silme işlemleri yapılabilmektedir. Bu işlemler sistem yöneticisi tarafından yapılabilmektedir. Kullanıcı Şekil 5.4’de gösterilen ana ekran arayüzündeki “Kullanıcı ekle-sil” butonu ile kullanıcı ekle sil arayüzüne yönlendirilmektedir.



Şekil 5.4. Ana ekran arayüzünde kullanıcı ekleme silme butonu.



Şekil 5.5. Sisteme kullanıcı ekleme ve silme arayüzü.

Sistem yöneticisi; diğer sistem kullanıcılarından farklı olarak sisteme kullanıcı tanımlama veya tanımlı olan kullanıcıları silebilme yetkisine sahip işletme yöneticisidir. Bu yönetici genellikle işletmenin üst düzey yöneticilerindedir.

Sistem yöneticisi, Şekil 5.5’de gösterilen arayüz ile sisteme yeni kullanıcılar ekleyebilmekte, var olan kullanıcıları silebilmekte veya sisteme kayıtlı kullanıcıların listesini görüntüleyebilmektedir.

Sistem yöneticisi, Şekil 5.5’de verilen arayüzde herhangi bir personelin bilgilerini ilgili alanlara girdikten sonra “Ekle” butonunu tıklayarak, sisteme kullanıcı tanımlayabilmektedir. Bunun yanında “ekle” butonu yerine “Kullanıcı sil” butonunu kullanarak sistemde kayıtlı olan kullanıcıyı silebilmektedir. Sistem yöneticisi “Kullanıcı listesi” butonunu tıklayarak kayıtlı kullanıcıların listesini görüntüleyebilmektedir.

KDS’de personel listesi güncelleme, yeni personel ekleme, joker personel bilgisini girme ve personel bilgisi arama arayüzü

Şekil 5.6’da verilen arayüz ile sisteme yeni personel bilgisinin girişi yapılabilmekte, sistemde var olan personel bilgisi silinebilmekte, joker personel bilgisi girişi yapılabilmekte, sistemde tanımlı olan personel listesi görüntülenebilmektedir.

Sisteme yeni personel ekleme işlemi

Kullanıcı tarafından Şekil 5.6’da yeni personel ekleme bölümünde ilgili alanlara eklenecek personelin verileri girilir. Daha sonra “Yeni personel ekle” butonuna tıklanarak, sisteme yeni personel ekleme işlemi tamamlanmaktadır.

Şekil 5.6. Personel listesi güncelleme arayüzü,

Sisteme yeni joker personel ekleme işlemi

Kullanıcı Şekil 5.6'da "Yeni personel bilgilerinizi giriniz" bölümünde ilgili alanlar doldurulduktan sonra "Joker personel onay" kutucuğunu işaretleyerek, personelin joker olacağına karar vermektedir. Daha sonra bu personelin süreç bazlı puanlarını sisteme girmek için "Joker personel ise doldurunuz" bölümündeki ilgili alanlar doldurulmakta ve "Yeni personel ekle" butonu tıklanarak, sisteme joker personel ekleme işlemi tamamlanmaktadır.

Kullanıcı bu işlemler dışında Şekil 5.6'da "Normal personel", "Joker personel" ve "Tüm personeli listele" butonlarını kullanarak sisteme kayıtlı personeli ayrı listeleylebilmektedir. Kullanıcı, Şekil 5.5'de "Silinecek personelin sicil numarasını giriniz" bölümündeki ilgili alanı doldurulduktan sonra, "Personeli sil" butonunu kullanarak sistemden personel silme işlemi yapabilmektedir. Kullanıcı aynı şekilde "Arama" bölümündeki ilgili alanları doldurup, "Ara" butonunu tıklayarak sistemde kayıtlı olan personelin bilgilerini görüntüleyebilmektedir.

Sistemde yeni proje oluşturma veya silme arayüzü

Kullanıcı, Şekil 5.7'de gösterilen arayüz ile sisteme yeni proje ekleyebilmekte, projenin üretimden kaldırılması durumunda ise proje bilgilerini sistemden silmekte veya sistemde kayıtlı projelerin bilgilerini listeleylebilmektedir.

ID	HatAdı	FomenAdı
1	IA FLOOR LHD	DILARA
2	IA FLOORT LHD1	ÖMER FA
*		

Şekil 5.7. Sisteme proje ekleme ve silme arayüzü.

Süreç ekleme-silme, sürece istasyon ekleme-silme arayüzü

Kullanıcı, Şekil 5.8’de verilen arayüz ile sisteme yeni sürecin eklemesi, sistemde kayıtlı olan sürecin silinmesi, sistemde kayıtlı olan süreçlerin listesinin görüntülenmesi, sisteme yeni istasyonların eklenmesi, var olan istasyonların silinmesi ve sistemde kayıtlı olan istasyonların süreç bazlı listelenmesi işlemlerini yapabilmektedir.

Kullanıcı “Süreç ekle” bölümünde yeni eklenecek sürecin adını yazarak “Ekle” butonu ile sisteme kayıt işlemi yapmaktadır. Kayıtlı olan sürecin silinme işlemi, “Süreç no” bölümünden ilgili sürece ait süreç numarası girilerek “Sil” butonu ile sistemden süreç silme işlemi yapılmaktadır. Sistemde kayıtlı olan süreçlerin listesinin görüntülenme işlemi “Mevcut süreç listesi” bölümünde “Listele” butonu ile kayıtlı olan süreçlerin listesi görüntülenmektedir.

İstasyon işlemleri bölümünden sistemde kayıtlı olan süreçlere bağlı olan istasyonların sayısı ve bu istasyonların süreleri belirlenmektedir. Ayrıca sürece bağlı olan bir istasyonun silme işlemi de “Silinecek istasyon no” bölümünden ilgili istasyona ait istasyon numarası girildikten sonra “Sil” butonu ile yapılabilmektedir.

Süreç-İstasyon İşlemleri

Süreç Adı

EKLE

Süreç No

SİL

İstasyon İşlemleri

İstasyon Sayısı

Süreç No

Standart Süre dk.(dakika)

EKLE

Standart Süre Güncelle

İstasyon No

Standart Süre

GÜNCELLE

Silinecek İstasyon No

SİL

Mevcut Süreçlerin Listesi

ID	BölümAdı
1	Modül Hazırlama ...
2	Bantlama Prosesi
3	Kküp Test Prosesi
4	Elektriksel Test P...
5	Paketleme Test ...

LİSTELE

Mevcut İstasyonlar Listesi

ID	StandartZaman	BolumID	VekilID
1	3	1	2
2	4,5	1	
3	4,5	1	

Süreç Seçiniz

Şekil 5.8. Süreç ekleme-silme ve süreçte var olan istasyonu silme arayüzü

Sisteme süreç eklenip ve bu eklenen süreçlere istasyonlar atandıktan sonra “Mevcut istasyonların listesi” bölümünden istenilen süreç ve o sürece atanmış istasyonların listesi görüntülenebilmektedir.

Ekip oluşturma ve bu ekipleri sistemde tanımlı olan projelere atama arayüzü

Kullanıcı Şekil 5.9’da verilen arayüz ile sistemde birbiri ile bağlantılı iki işlem yapmaktadır. Bunlardan birincisi yeni ekipler tanımlama veya tanımlı olan ekiplerin sistemden silinmesi işlemi, ikincisi ise daha önce tanımlanmış olan projelere ekipleri atama veya projelere atanmış olan ekiplerin atandıkları projeden çıkarılmasıdır.

“Ekip oluştur” bölümünde ilgili alanlar doldurularak “Ekle” butonu ile sisteme yeni ekipler eklenebilmektedir. Sistemde kayıtlı olan ekibin silinmesi, “Ekipleri sil” bölümünden ilgili ekibin numarası (ID) girilerek sil butonu ile olmaktadır. Sisteme kayıtlı olan ekiplerin listesinin görüntülenmesi işlemi, “Ekipleri görüntüle” butonu ile yapılmaktadır. Ekip listesinin

içerisinde sisteme kayıtlı olan ekiplerin numaraları, ekip isimleri ve ekip sorumluları görüntülenebilmektedir.

Kullanıcı sisteme eklediği ekipleri daha önce sisteme kaydedilmiş projelere atama veya herhangi bir projeye atanmış ekibi ilgili projeden çıkarma işlemini yapabilmektedir. Kullanıcı ekip atadığı projelerin listesini görüntülemek için projelerin listesi bölümünden “Proje listesini getir” butonu ile görüntüleyebilmektedir.

Şekil 5.9. Ekip oluşturma ve projelere ekip atama arayüzü.

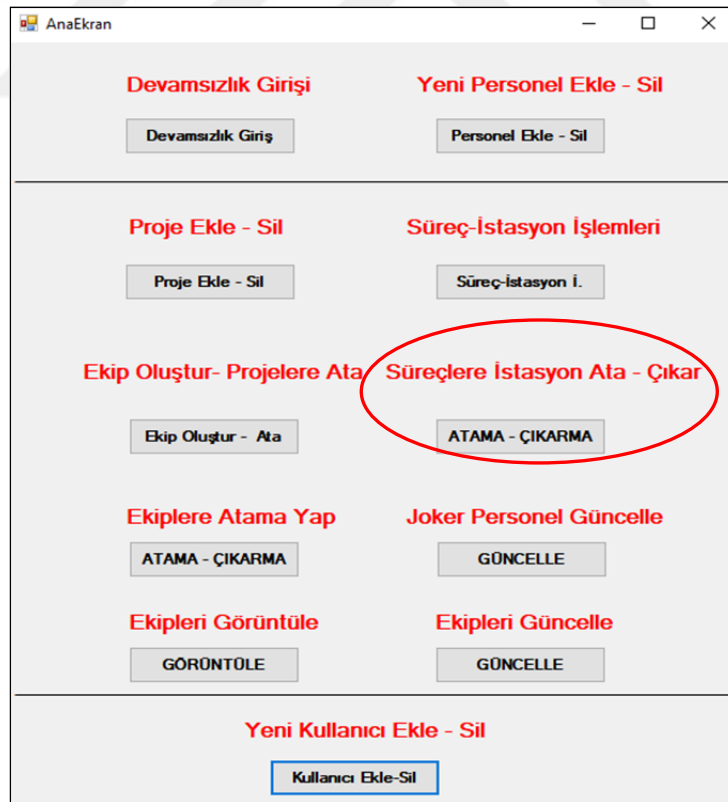
Süreçlere personel atama ve atanmış personelin sürecin ilgili istasyonuna atama arayüzü

Ana ekran. arayüzünde Şekil 5.10’de belirtilen “Süreç istasyon ata-çıkara” butonu ile Şekil 5.11’da gösterilen süreçlere personel atama ve atanmış personelin sürecin ilgili istasyonuna atama arayüzü açılmaktadır.

Şekil 5.11’da gösterilen arayüzün “Personel listesi” bölümünde “Normal personel” butonu ile personel listesi görüntülenmektedir. Kullanıcı, süreçlere personel atama yapmak için bu listeyi kullanmaktadır. Kullanıcı “Süreç listesi” bölümünde “Süreçleri listele” butonu ile sisteme kayıtlı süreçlerin listesini görüntüleyebilmektedir. Kullanıcı, aşağıdaki adımları takip ederek süreçlere personel atayabilmekte ve atadığı personeli ilgili sürecin istasyonlarına atayabilmektedir.

- ✓ Personel listesinden seçilen personelin numarasını “Personel no” alanına gir,
- ✓ Seçilen personelin atamak istenen sürecin numarasını “Süreç no” alanına gir,
- ✓ “Atama” butonunu tıkla
- ✓ “Süreçler” bölümünden ilgili süreci seç,
- ✓ İlgili sürece atanmış personel listesini ve sürece atanmış istasyonları görüntüle
- ✓ Sürece atanmış personel listesinden seçilen personelin numarasını “Personel no” alanına gir,
- ✓ Seçilen personeli atamak istediğin istasyonun numarasını “İstasyon no” alanına gir,
- ✓ “Atama” butonuna tıkla.

Aynı zamanda daha önce atamış olduğu personeli istasyondan silmek istendiğinde de ilgili alana personelin numarası girilerek bağlı olduğu istasyondan çıkarabilmektedir.



Şekil 5.10. Ana ekran arayüzünde süreçlere ve istasyonlara personel atama veya personel çıkarma butonu.

Süreç-İstasyon Atamaları

Personel Listesi

NORMAL PERSONEL

Seçilen Süreçteki Personel Listesi

ID	Adı	Soyadı	İstasyonID	Bölü
1	Ali	Efe	1	1
2	Mert	Can	2	1
3	Ahmet	Ufuk	3	1
*				

Personel ID

Süreç ID

➔ ATAMA

Personelin Süreç'ini Sil

Personel ID

SİL

Süreç Listesi

SÜREÇLERİN LİSTESİ

İstasyon Listesi

ID	StandartZaman	BölümAdı
1	3	Modül Hazırlama
2	4,5	Modül Hazırlama
3	4,5	Modül Hazırlama
*		

Süreçler

Modül Hazırlama

İstasyon Atamaları

Personel ID

İstasyon ID

➔ ATAMA

Personelin İstasyonunu Sil

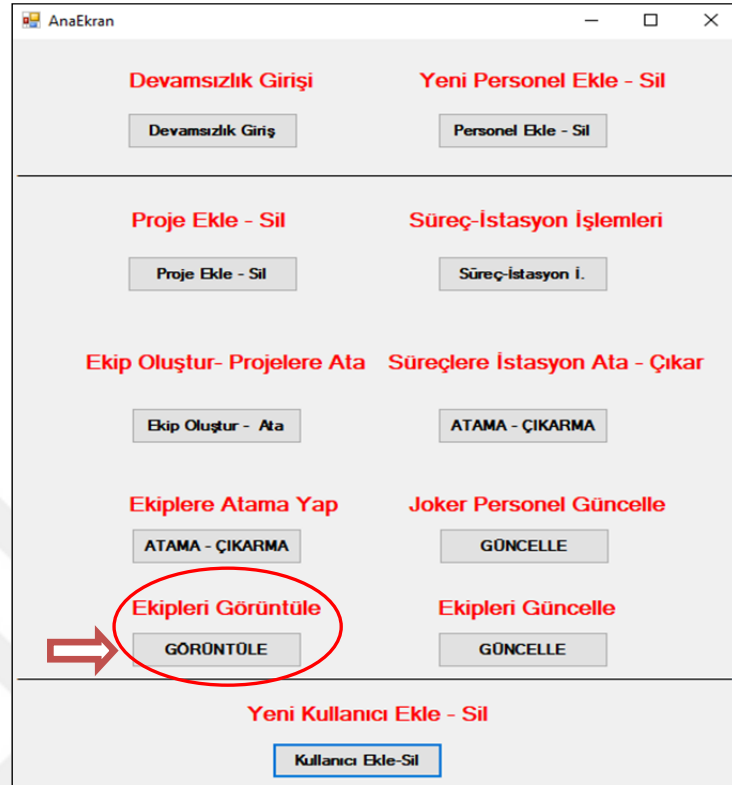
Personel ID

SİL

Şekil 5.11. Süreçlere personel atama ve atanan personeli sürecin ilgili istasyonuna atama arayüzü.

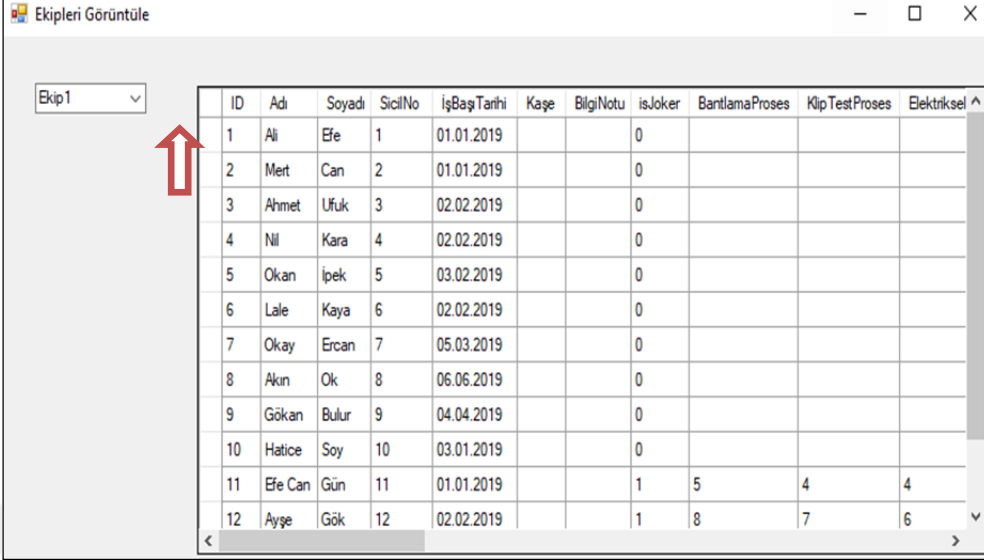
Sisteme kayıtlı ekiplere atanmış personel listesinin görüntülediği arayüz

Şekil 5.13’de verilen arayüzü, ana ekran arayüzünde Şekil 5.12’de belirtilen “Ekipleri görüntüle” butonu ile açılmaktadır. Kullanıcı bu arayüzün “Ekip adı” bölümünden ilgili ekibin ismini seçerek ilgili ekibin personel listesi görüntüleyebilmektedir. Görüntülenen bu ekip listesinde ekibe atanmış personel listesi ve personelin hangi istasyonlarda çalıştıkları görülebilmektedir.



Şekil 5.12. Ana ekran arayüzünde sisteme kayıtlı olan ekiplerin personel listesinin görüntüleme butonu.

Eğer ekip içerisinde işe gelmeyen personel varsa ve bu personelin yerine bir joker personel atanmış ise ilgili joker personelin bilgileri ve atandığı istasyon bilgisini de listede görülmektedir.

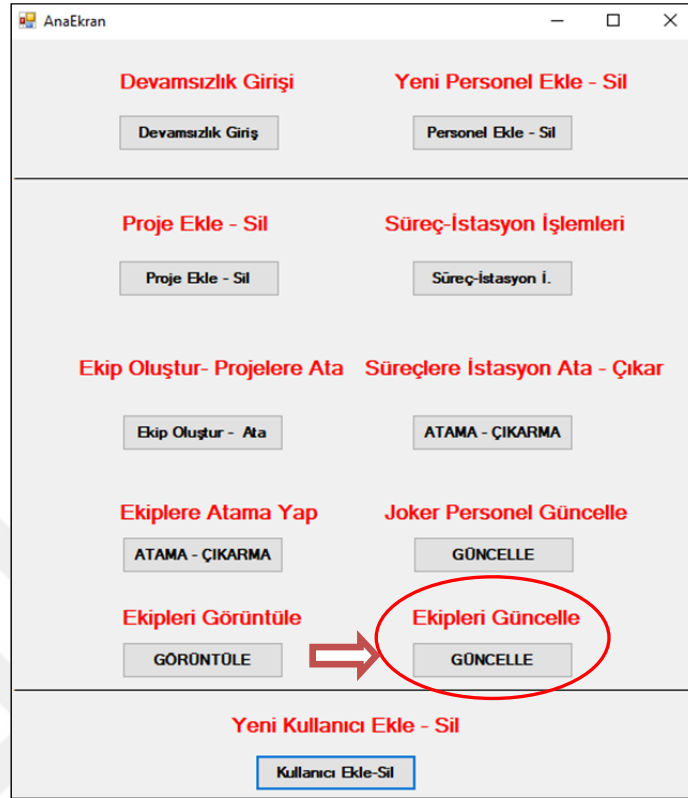


ID	Adı	Soyadı	SicilNo	İşBaşTarihi	Kaşe	BilgiNotu	isJoker	BartlamaProses	KlipTestProses	Elektriksel
1	Ali	Efe	1	01.01.2019			0			
2	Mert	Can	2	01.01.2019			0			
3	Ahmet	Ufuk	3	02.02.2019			0			
4	Nil	Kara	4	02.02.2019			0			
5	Okan	İpek	5	03.02.2019			0			
6	Lale	Kaya	6	02.02.2019			0			
7	Okay	Ercan	7	05.03.2019			0			
8	Akın	Ok	8	06.06.2019			0			
9	Gökân	Bulur	9	04.04.2019			0			
10	Hatice	Soy	10	03.01.2019			0			
11	Efe Can	Gün	11	01.01.2019			1	5	4	4
12	Ayşe	Gök	12	02.02.2019			1	8	7	6

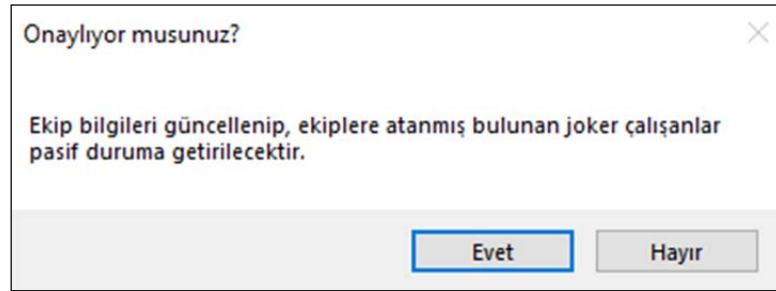
Şekil 5.13. Sisteme kayıtlı ekiplere atanmış personel listesinin görüntülediği arayüz.

Ekip listelerinin güncellenmesi

Ana ekran arayüzünde Şekil 5.14’de gösterilen “Ekipleri güncelle” butonu ile Şekil 5.15’deki sistem verdiği onay mesajı alınmaktadır. Bu mesaja “Evet” butonuna tıklayarak cevap verildiğinde daha önce işe gelmeyen personelin yerine atanan joker personel pasif hale gelmektedir. Böylece ekip listeleri ilk oluşturuldukları hale gelmektedirler.



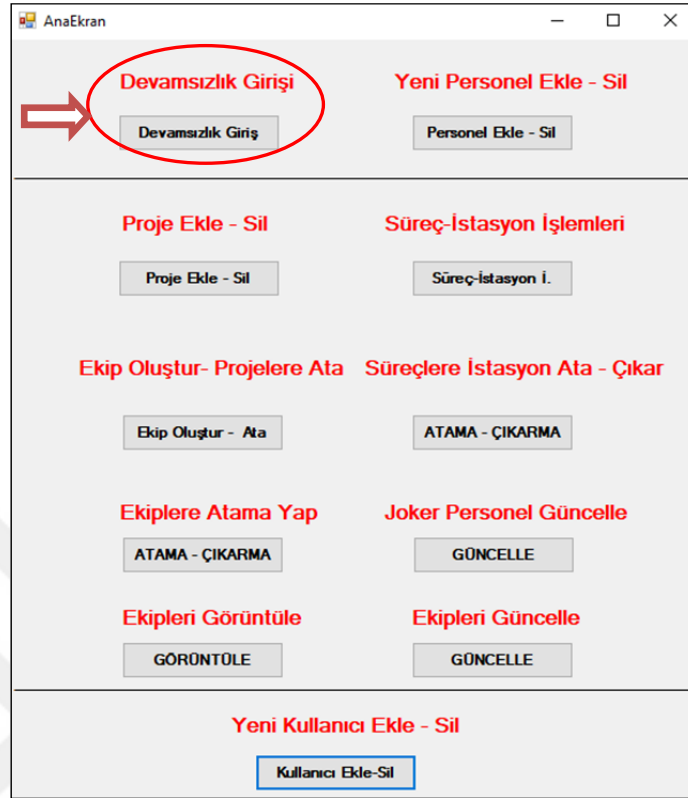
Şekil 5.14. Ana ekran arayüzünde ekip listesini güncelleme butonu.



Şekil 5.15. Atanmış joker personelin pasif hale getirilmesi için onay mesajı.

İşe gelmeyen personel bilgisinin sisteme giriş arayüzü

Bu arayüz Şekil 5.16'da gösterilen ana ekran arayüzünde "Devamsızlık girişi" butonu ile açılır. Kullanıcı bu arayüzde üç farklı işlem yapılabilmektedir. Bunlardan birincisi işe gelmeyen personelin bilgilerini sisteme girme işlemi, ikincisi devamsızlık listesinde arama yapabilme işlemi ve üçüncüsü de devamsızlık listesinden personel bilgisi silme işlemidir.



Şekil 5.16. Ana ekran arayüzünde işe gelmeyen personelin bilgilerini siteme giriş butonu.

The screenshot shows the 'DevamsızlıkGirisi' window. It contains a data entry form on the left and a table on the right.

Data Entry Form:

- Adı: Ali
- Soyadı: Efe
- Sicil No: 1
- Süresi: 1
- Sebebi: Raporlu
- Buttons: Onayla, İptal

Devamsızlık Listesi Table:

tarih	ID1	Adı	Soyadı	SicilNo	İşBaşTarhi	Kaşe	BilgiNotu	isJoker
29.04.2019	1	Ali	Efe	1	01.01.2019			0
5.05.2019	1	Ali	Efe	1	01.01.2019			0
30.04.2019	2	Mert	Can	2	01.01.2019			0
5.05.2019	2	Mert	Can	2	01.01.2019			0
5.05.2019	3	Ahmet	Ufuk	3	02.02.2019			0
30.04.2019	10	Hatice	Soy	10	03.01.2019			0
6.05.2019	10	Hatice	Soy	10	03.01.2019			0

Search Form (Arama):

- Adı: []
- Soyadı: []
- Sicil No: []
- Button: ARA

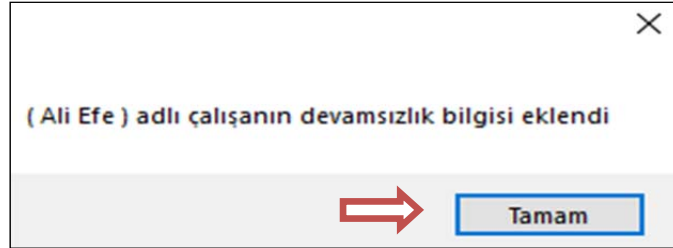
Devamsızlık Kaydı Sil:

- Devamsızlık ID: 3
- Button: SİL

Buttons: Devamsızlık Listesini Getir

Şekil 5.17. İşe gelmeyen personelin bilgisinin sisteme giriş arayüzü.

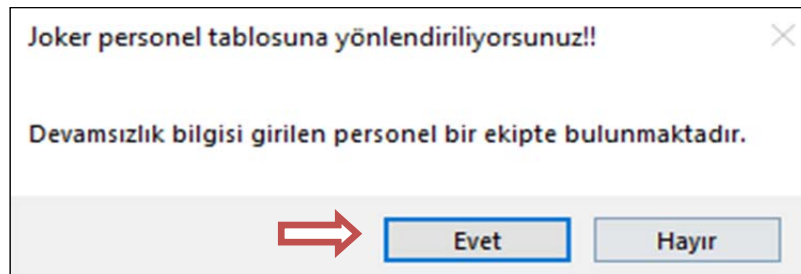
Kullanıcı Şekil 5.17’de gösterilen devamsızlık girişi arayüzüne ilgili personelin bilgilerini girip “Onay” butonuna tıkladıktan sonra Şekil 5.18’de gösterilen bilgi mesajı gelmektedir. Kullanıcı bu onay mesajına “Tamam” butonuna tıklayarak cevap verdiğinde sisteme personelin devamsızlık giriş işlemini tamamlamış olmaktadır.



Şekil 5.18. İşe gelmeyen personelin bilgisinin sisteme girişinin onay mesajı.

İşe gelmeyen personel yerine atama işlemi

Kullanıcı Şekil 5.17’de gösterilen işe gelmeyen personelin bilgisinin sisteme giriş arayüzünde işe gelmeyen personelin bilgileri, ilgili bölümlere girildikten sonra “Onay” butonu tıkladığında Şekil 5.18’de gösterilen onay mesajları sistemden alınmaktadır. Kullanıcı bu sistem mesajına “Tamam” butonuna tıklayarak cevap verdiğinde, eğer devamsızlık bilgisi girilen personel bir ekipte çalışan olarak kayıtlı ise sistem kullanıcıyı joker personel çizelgesine yönlendirmek üzere Şekil 5.19’da gösterilen onay mesajı gönderir. Kullanıcı bu mesaja “Evet” butonuna tıklayarak cevap verdiğinde Şekil 5.20’de gösterilen yerine atama yap arayüzü açılmaktadır.



Şekil 5.19. İşe gelmeyen personelin yerine atama yapılırken sistemden alınan onay mesajı.

YerineAtamaYap

DEVAMSIZLIK BİLGİSİ GİRİLEN PERSONEL

Adı	Soyadı	SicilNo	EkipID	IstasyonID	BölümAdı
▶ Alı	Efe	1	1	1	Modül Hazırlama ...
*					

MEVCUT JOKER PERSONEL LİSTESİ

ID	Adı	Soyadı	isJoker	BantlamaProses	Klip TestProses	ElektrikselTest
*						

Otomatik Atama

MEVCUT JOKER PERSONEL LİSTESİ

Personel Süreci Seçiniz

Modül HazırlamaProses

OTOMATİK ATA

Şekil 5.20. İşe gelmeyen personelin yerine atama yapma arayüzü.

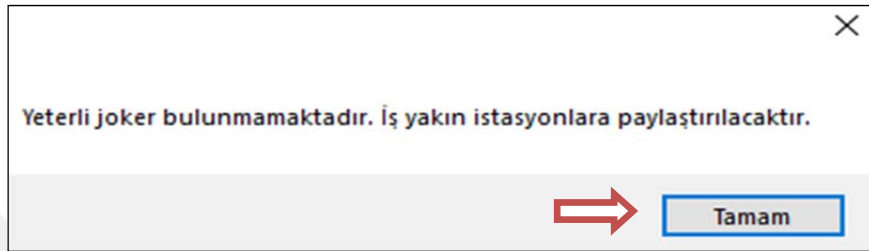
Kullanıcı bu arayüzde işe gelmeyen personelin yerine joker personel atama işlemini yapmaktadır.

Kullanıcının atama işlemi için aşağıdaki adımları takip etmesi gerekmektedir.

- ✓ Şekil 5.20’de gösterilen “Personelin sürecini seçiniz” alanından yerine atama yapılmasını istediği personelin sürecinin seçilmesi,
- ✓ “Otomatik ata” butonunun tıklanması.

Kullanıcı bu adımları sırasıyla yerine getirdiğinde joker personel arasından ilgili süreçte çalışacak joker personelin atanması süreç bazlı puanlara göre gerçekleştirilmektedir. Bu işlem işe gelmeyen her personel için yapılmaktadır. Kullanıcı bu işlemi günün her saatinde yapabilmektedir.

Sistemde işe gelmeyen personelin yerine mevcut joker personel otomatik olarak atandıktan sonra, eğer atanacak joker personel yoksa Şekil 5.21’de gösterilen sistemin onay mesajı alınmaktadır. Kullanıcı, bu mesaja “Tamam” butonuna tıklayarak cevap verdiğinde, işe gelmeyen personelin istasyonunun çevrim süresi kendisine en yakın iki istasyonun iş yüklerine bağlı olarak dağıtılmaktadır. Dolayısıyla iş yükü fazla olan istasyona az iş, iş yükü az olan istasyona çok iş dağıtılarak standart istasyon süresinden sapma en küçüklenmektedir.



Şekil 5.21. İşe gelmeyen personelin yerine atanacak joker personel olmaması durumunda onay mesajı.

5.3. KDS için Uygulama

Bu çalışmada önerilen KDS için, işletmenin bir çalışma gününde karşılaşılan iki farklı devamsızlık durumu verileri kullanılarak iki senaryo oluşturulmuştur.

Joker personel atama işlemi

KDS’de joker personel atama işlemleri aşağıdaki gibi yapılmaktadır:

- ✓ Şekil 5.22’de gösterilen devamsızlık bilgisi giriş arayüzündeki ilgili alanlara işe gelmeyen personelin bilgisinin girilmesi,
- ✓ “Onayla” butonunun tıklanması,
- ✓ Sistem tarafından gönderilen ve Şekil 5.19’da gösterilen onay mesajına “Evet” butonu tıklanarak cevap verilmesi,
- ✓ Onay mesajından sonra açılan ”Yerine atama yap” arayüzündeki “Otomatik atama yap” bölümünde ilgili sürecin seçilmesi,
- ✓ “Otomatik ata” butonunun tıklanması.

Bu adımlar işe gelmeyen her personel için tekrarlanır ve böylece işe gelmeyen personelin yerine en uygun joker personel atanabilmektedir.

Joker personel sayısının işe gelmeyen personel sayısından eşit veya fazla olması durumu

Bu senaryoda devamsızlık yapan personel sayısı ile joker personel sayısı eşit ve 30'dur.

Joker personel atama işleminin bütün adımları tamamlandıktan sonra, eğer atanacak joker personel yok ise sistem Şekil 5.21'de gösterilen bilgi mesajını vermektedir. Kullanıcı bu mesaja "Tamam" butonuna tıklayarak cevap verdiğinde, sistem otomatik olarak işe gelmeyen personelin çevrim süresini kendisine en yakın iki istasyona dağıtım katsayıları oranında dağıtmaktadır.



Çizelge 5.1. İşletmede devamsızlık yapan personelin çalıştıkları istasyonlara atanan joker personelin süreç bazlı puanları.

No (ID)	Ad-Soyad	Personelin joker olma durumu	Model hazırlama süreci puanı	Bantlama süreci puanı	Klip test süreci puanı	Elektriksel test süreci puanı	Göz kontrol süreci puanı	Paketleme süreci puanı	Malzeme taşıma süreci puanı	Süreç no	Ekip no	Devamsızlık yapan personelin istasyon no	Ekip adı
1	Ahmet Pire	1	9	6	4	7	8	2	1	1	1	1	Ekip1
2	Bahriye Demir	1	3	9	4	2	5	6	8	2	1	2	Ekip1
3	Fatma Zeybek	1	5	4	6	8	1	5	4	4	1	3	Ekip1
4	Adalet Çetin	1	8	2	3	4	8	4	6	1	2	4	Ekip2
5	Cennet Çelik	1	2	3	8	1	6	2	6	3	2	5	Ekip2
6	Gamze Man	1	7	9	5	7	8	8	9	2	2	6	Ekip2
7	Şeyma Özdal	1	9	1	7	2	5	6	4	1	2	7	Ekip2
8	Zehra Eker	1	4	2	8	7	4	6	4	3	2	8	Ekip2
9	Büşra Kaya	1	6	2	8	9	1	4	7	4	3	9	Ekip3
10	Gülsüm Basan	1	1	2	2	5	8	2	4	5	3	10	Ekip3
11	Melike Tuğlu	1	8	7	6	3	7	4	6	1	3	2	Ekip3
12	Gürol Aldemir	1	9	8	6	4	3	7	8	1	3	6	Ekip3
13	Gizem Kara	1	4	9	6	5	7	5	6	2	4	2	Ekip4
14	Sinem Tunç	1	1	2	8	4	7	5	6	3	4	4	Ekip4
15	Metin Sevinç	1	2	4	6	8	9	5	7	5	4	7	Ekip4
16	Halil Aksoy	1	8	5	7	9	2	3	4	4	5	21	Ekip5
17	Engin Çavlı	1	9	1	4	6	8	7	4	1	5	16	Ekip5
18	Vildan Kıymet	1	2	4	5	8	9	4	2	6	5	10	Ekip5
19	Ali Yeşilyurt	1	2	8	1	4	7	1	5	2	9	2	Ekip9
20	Hasan Coşar	1	2	4	5	7	5	4	9	7	5	3	Ekip5
21	Çiğdem Bilgin	1	1	2	8	3	3	3	9	7	6	8	Ekip6
22	Yusuf Düzal	1	1	4	4	9	2	4	8	4	6	10	Ekip6
23	İbrahim Üst	1	9	2	1	4	2	6	8	1	6	15	Ekip6
24	Fatma Atila	1	1	5	7	5	8	4	5	5	10	5	Ekip10
25	İsa Erkan	1	7	5	9	1	5	7	1	3	6	2	Ekip6
26	Ali Sal	1	2	4	7	9	5	8	4	4	7	7	Ekip7
27	Sibel Erbaş	1	1	9	8	5	6	4	7	2	7	4	Ekip7
28	Canip Gündüz	9	4	5	7	2	2	8	4	4	7	1	Ekip7
29	Furkan Men	1	8	9	6	5	7	5	6	2	7	9	Ekip7
30	Melih Akkaya	1	9	1	8	7	6	4	2	1	8	11	Ekip8

İşletmede farklı projelerde çalışan on ekibin işe gelmeyen 30 personelinin yerine Çizelge 5.2’de gösterilen joker personel kadrosundan yetkin olan joker personelin atanması ile Çizelge 5.1 oluşmaktadır. Çizelge 5.1’de gösterildiği gibi işe gelmeyen personelin yerine atanan joker personelin ilgili süreç puanları diğer personele göre daha yüksektir. Böylece işe gelmeyen personelin yerine yetkin joker personelin atandığı görülmektedir.

Çizelge 5.2. Joker personelin süreç bazlı puanları.

No (ID)	Ad-Soyad	Süreçler							
		Personelin joker olma durumu	Model hazırlama süreci puanı	Bantlama süreci puanı	Klip test süreci puanı	Elektriksel test süreci puanı	Göz kontrol süreci puanı	Paketleme süreci puanı	Malzeme taşıma süreci puanı
1	Yusuf Düzel	1	1	4	4	9	2	4	8
3	Çiğdem Bilgin	1	1	2	8	3	3	3	9
4	Gamze Kahraman	1	7	9	5	7	8	8	9
5	Ahmet Pire	1	9	6	4	7	8	2	1
6	Bahriye Demir	1	3	9	4	2	5	6	8
7	Fatma Zeybel	1	5	4	6	8	1	5	4
8	Adalet Çetin	1	8	2	3	4	8	4	6
9	Cennet Çelik	1	2	3	8	1	6	2	6
9	Melih Akkaya	1	9	1	8	7	6	4	2
10	Şeyma Özdal	1	9	1	7	2	5	6	4
11	Zehra Eker	1	4	2	8	7	4	6	4
12	Halil Alper Aksoy	1	8	5	7	9	2	3	4
13	Vildan Kıymetli	1	2	4	5	9	8	4	2
14	Hasan Coşar	1	2	4	5	9	5	4	1
15	Büşra Kaya	1	6	2	8	9	1	4	7
16	Ali Yeşilyurt	1	2	8	1	4	7	1	5
17	Engin Çavlı	1	9	1	4	6	8	7	4
18	Gülsüm Basan	1	1	2	2	5	8	2	4
19	Melike Tuğlu	1	8	7	6	3	7	4	6
20	İbrahim Üstünel	1	9	2	1	4	2	6	8
21	Gürol Aldemir	1	9	8	6	4	3	7	8
22	Fatma Atila	1	1	5	7	5	8	4	5
23	Furkan Gökmen	1	8	9	6	5	7	5	6
24	Muhammed Sam	1	2	4	7	9	5	8	4
25	Canip Gündüz	1	4	5	7	9	2	8	4
26	İsa Erkan	1	7	5	9	1	5	7	1
27	Gizem Karazeybek	1	4	9	6	5	7	5	6
28	Metin Sevinç	1	2	4	6	8	9	5	7
29	Sibel Erbaş	1	1	9	8	5	6	4	7
30	Sinem Tunç	1	1	2	8	4	7	5	6

Joker personel sayısının işe gelmeyen personel sayısından az olması durumu

Bu senaryoda devamsızlık yapan personel sayısı 32, joker personel sayısı ise 30'dur.

Eğer yeterli joker personel yok ise sistem Şekil 5.21'de gösterilen onay mesajını vermektedir. Kullanıcı bu mesajı onayladıktan sonra, sistem işe gelmeyen personelin çevrim süresini kendisine en yakın iki istasyona iş yüklerine bağlı olarak dağıtmaktadır. Böylece iş yükü fazla olan istasyona daha az iş dağıtılması sağlanmaktadır.

İşletmede aynı gün içinde Ekip10'un 9 ve 16 numaralı istasyonlarındaki personelinin rahatsızlanması durumunda devamsızlık yapan personel sayısı 32'ye yükselmiştir. Bu durumda joker personel sayısı 30 olduğu için devamsızlık yapan personel sayısından joker personel sayısı daha az olmaktadır. Sistem, joker personel atanamayan Ekip10'un 9 ve 16 nolu istasyonlarının çevrim sürelerini dağıtım katsayısı oranında 8,10 ve 15,17 nolu istasyonlara dağıtmaktadır. Bu dengeleme sonucunda oluşan Ekip10 istasyonlarının yeni çevrim süreleri Çizelge 5.5' de gösterilmektedir.

Çizelge 5.3. İşletmede devamsızlık yapan personelin bulunduğu ekip isimleri ve çalıştıkları istasyon numaraları

	Ekip adı									
	Ekip1	Ekip2	Ekip3	Ekip4	Ekip5	Ekip6	Ekip7	Ekip8	Ekip9	Ekip10
Devamsızlık yapan istasyon no	2	4	2	2	3	2	1	11	2	2
	3	5	6	4	10	8	4			9
	5	6	9	7	16	10	7			16
		7	10		21	15	9			
		8								

Süreçteki istasyonların işleri genellikle birbiri ile ilişkili olduğu için dağıtma işlemi yapılırken, iş yükü dağıtılacak istasyonun işleri kendisine en yakın iki istasyona atanmaktadır.

Çizelge 5.4. Veri tabanında dengeleme öncesi Ekip10'un istasyon çevrim süreleri.

İstasyon no	İş yükü atanan birinci istasyon	İş yükü atanan ikinci istasyon	Süreç no	Çevrim süresi (dk)
1	-	-	1	3
2	-	-	1	3
3	-	-	1	3
4	-	-	2	3,5
5	-	-	2	3,5
6	-	-	2	3,5
7	-	-	2	3,5
8	-	-	3	4
9	-	-	4	4
10	-	-	5	3
11	-	-	5	3
12	-	-	5	3
13	-	-	5	3
14	-	-	5	3
15	-	-	5	3
16	-	-	6	4
17	-	-	6	4
18	-	-	6	4
19	-	-	6	4
20	-	-	7	3
21	-	-	7	3
22	-	-	7	3

Çizelge 5.5. Ekip10'un 9 ve 16 nolu istasyonuna atanacak joker personel olmaması durumunda yeni çevrim süreleri.

İstasyon no	İş yükü atanan birinci istasyon	İş yükü atanan ikinci istasyon	Süreç no	Çevrim süresi (dk)
1	-	-	1	3
2	-	-	1	3
3	-	-	1	3
4	-	-	2	3,5
5	-	-	2	3,5
6	-	-	2	3,5
7	-	-	2	3,5
8	-	-	3	4,9
9	8	10	4	4
10	-	-	5	5,3
11	-	-	5	3
12	-	-	5	3
13	-	-	5	3
14	-	-	5	3
15	-	-	5	5,3
16	15	17	6	4
17	-	-	6	5,7
18	-	-	6	4
19	-	-	6	4
20	-	-	7	3
21	-	-	7	3
22	-	-	7	3

Ekip10'un istasyon çevrim süreleri Çizelge 5.4'de gösterilmektedir. Atamalar yapıldıktan sonra Ekip10'un istasyonlarının yeni çevrim süreleri Çizelge 5.5'te verilmiştir. 9 nolu istasyona joker personel atanmadığı için çevrim süresi kendisine en yakın iki istasyon olan 8 ve 10 nolu istasyonlara, 16 nolu istasyona joker personel atanmadığı için çevrim süresi kendisine en yakın iki istasyon olan 15 ve 17 nolu istasyonlara dağıtılmıştır. Böylece işe gelmeyen personelin çalıştığı istasyonun iş yükü iki istasyona dağıtım katsayıları (K_{i+1}, K_{i-1}) oranında dağıtılmaktadır.

Önerilen KDS ile işletmede personelin devamsızlık veya rahatsızlanması gibi durumlarda boş kalan istasyonların uygun şekilde çalışmaya devam edebilmeleri sağlanmış olmaktadır. Böylece ortaya çıkan problemlere sistematik olarak hızlı ve uygun çözümler bulunabilmekte ve işler aksamamaktadır.



6. SONUÇ VE ÖNERİLER

Otomotiv sektöründe artan rekabet ve azalan kâr oranları işletmeleri gelişime zorlamaktadır. İşletmeler için en önemli konulardan biri de insandır. Emek yoğun işletmelerde işgücünü etkin kullanabilmenin yanında, karşılaşılan insan faktörlü devamsızlık gibi problemlerde de en iyi çözümü bulmak ve hızlı bir şekilde bu çözümü uygulamak gerekmektedir. Rekabet ortamında ayakta kalabilmek için bu konuda da çalışmalar yapılmalıdır.

Bu çalışmada bir otomotiv yan sanayi işletmesinde işe gelmeyen personelin yerine personel atama ya da gerekli işe dengelenmesi yapabilmek için öncelikle bir matematiksel model geliştirilmiştir. Bu modelde, yeterli joker personel varsa; joker personel ekibinden uygun personel atanmakta, eğer yeterli joker personel yok ise; yeterli olan personel atanıp, kalan işlerin iş yükleri uygun istasyonlara dağıtılmaktadır.

İşletmede personelin devamsızlık durumunun günlük değişkenlik göstermesine karşılık, önerilen modelin her çalışma gününün başında çalıştırılması ve sonuçlarının analiz edilmesi gerekmektedir. İşletmede çalışan personel sayısının fazla olması nedeniyle, matematiksel modelin çözüm süresinin uzunluğu ve hata yapma olasılığı da yükselecektir. Bu nedenle personel devamsızlığında atama ve/veya dengeleme problemi için bir karar destek sistemi geliştirilmeye karar verilmiştir. Geliştirilen bu karar destek sistemi C# yazılım dilinde yazılmış ve kullanıcıya kullanım kolaylığı sağlayan arayüzler tasarlanmıştır.

İşletme yönetimi kararı ile işe gelmeyen personelin çalıştığı istasyonlarda çalışabilecek yetkinlikte personel içeren bir joker personel ekibi oluşturulmuştur. Bu ekip, süreç tabanlı yetkinlikleri göz önünde bulundurularak oluşturulmuştur. Joker personel dışındaki personelden birinin işe gelmemesi durumunda, bu ekipten ilgili çalışma istasyonunda çalışabilecek en yetkin personel seçilmekte ve ilgili istasyona atanmaktadır. Önerilen KDS her çalışma gününün başında kısa sürede çalıştırılıp, çözüm önerileri elde edilebilecektir. Joker personel sayısının yeterlilik durumuna göre çözüm yolu değişmektedir. İşe gelmeyen personelin çalıştığı istasyonlara yeterli joker personel olması durumunda uygun atamalar yapılabilmektedir. Yeterli joker personel bulunmaması durumunda ise, mevcut joker personelin atamasının ardından, işe gelmeyen personelin çalıştığı istasyonların iş yükleri en uygun şekilde en uygun istasyonlara dağıtılabilir. Böylece devamsızlık ve rahatsızlanma gibi durumların olması durumunda oluşabilecek kayıplar en küçüklenmeye çalışılmaktadır.

Çalışmada önerilen KDS ile işletmelerin önemli bir maliyet kalemi olan işgücü kaynaklarının verimli kullanılabilme olanağı sağlanmaktadır. KDS'nin kullanışlı, hızlı ve

dinamik bir yapıya sahip olması, sistematik bir çözüm önerisi getirmektedir. Bu durum karşılaşılan problemler sırasında oluşabilecek görünmeyen maliyetlerin de önüne geçmektedir.

İşletmenin diğer projeleri ve bu projelerde çalışan personel ekipleri sisteme tanımlanarak, KDS işletme geneli için genişletilmiş olmaktadır. İşletme yönetimi, KDS'yi işletmede kullanılan Kurumsal Kaynak Planlama (ERP) programı ile bütünleştirerek personelin anlık devamsızlık durumlarını sistemde takip edilebilmesini sağlayacaktır. Böylece sistemin kullanılabilirliğinin artacağı düşünülmektedir.

Önerilen KDS, farklı emek yoğun çalışan işletmelerde de kullanılabilir. Böylece personelin işe gelmemesi durumunda personel atama ve dengeleme sürecinde doğru ve sistematik kararların hızlı ve uygun bir şekilde alınması sağlanabilecektir.

KAYNAKLAR DİZİNİ

Aktaş, E., Ülengin, F. ve Önsel Şahin, Ş., (2007). "A Decision Support System To Improve The Efficiency Of Resource Allocation In Healthcare Management", *Socio - Economic Planning Sciences*, 41:130–146.

Avramidis, A. N., Chan, W., Gendreau, M., L'Ecuyer, P. ve Pisacane, O., (2010). "Optimizing Daily Agent Scheduling In A Multiskill Call Center", *European Journal of Operations Research*, 200(3):822–832.

Azmat, C. S. ve Widmer, M. A., (2004). "Case Study Of Single Shift Planning And Scheduling Under Annualized Hours: A Simple Threestep Approach", *European Journal of Operations Research*, 153:148–175.

Baykasoglu, A., (2001). "MOAPPS 1.0: Aggregate Production Planning Using The Multiple-objective Tabu Search", *International Journal of Production Research*, 39:3685–3702.

Bayrak, A. E., (2010). Optimization Algorithms For Resource Allocation Problem Of Air Tasking Order Preparation, Yüksek Lisans Tezi, ODTÜ Bilgisayar Mühendisliği Bölümü, Ankara.

Beaumont, N., (1997). "Scheduling Staff Using Mixed Integer Programming", *European Journal of Operations Research*, 98(97):473–484.

Blöchliger, I., (2004). "Modeling Staff Scheduling Problems. A tutorial", *European Journal of Operations Research*, 158:533–542.

Braunschweig, B., Joulia, X., Nguyen, D. ve Bagajewicz, M., (2008). "Optimization Of Preventive Maintenance Scheduling In Processing Plants", 18th European Symposium on

Computer Aided Process Engineering, (2006) 25:319–324.

Brucker, P., Qu, R. ve Burke, E., (2011). "Personnel Scheduling: Models And Complexity", *European Journal of Operations Research*, 210(3):467–473.

Carrasco, R. C., (2010). "Long-term Staff Scheduling With Regular Temporal Distribution", *Computer Methods and Programs in Biomedicine*, 100(2):191–199.

Corominas, A., Pastor, R. ve Rodríguez, E., (2006). "Rotational Allocation Of Tasks To Multifunctional Workers In A Service Industry", *International journal of production economics*, 103:3–9.

Çetinyokuş, T., Gökçen, H., (2008) "İşgören performansının değerlendirilmesi için bir karar destek sistemi ve uygulaması", *Gazi Üniv. Müh. Mim. Fak. Der. J. Fac. Eng. Arch. Gazi Univ. Cilt 23, No 1, 239-248, 2008 C.23, No.1, 239-248.*

De Bruecker, P., van den Bergh, J., Beliën, J. ve Demeulemeester, E., (2014). "Workforce Planning Incorporating Skills: State Of The Art", *European Journal of Operations Research*, 243: 1-16.

KAYNAKLAR DİZİNİ (devam)

De Castro, H. F. ve Cavalca, K. L., (2006). "Maintenance Resources Optimization Applied To A Manufacturing System", *Reliability Engineering and System Safety*, 91:413–420.

Dellaert, N. ve De Kok, T., (2004). "Integrating Resource And Production Decisions In A Simple Multistage Assembly System", *International journal of production economics*, 90: 281–294.

Dolgui A., Kovalev S. Kovalyovc , M.Y., Malyutina S. ve Soukhal A., (2018). *European Journal of Operational Research* 264 200–211.

Eimaraghy, H., (2000). "Scheduling of Manufacturing Systems Under Dual Resource Constraints Using Genetic Algorithms" *Journal Of Manufacturing Systems*, 19(3):186–201.

Eiselt, H. A. ve Marianov V., (2008). "Employee Positioning And Workload Allocation", *Computers and Operations Research*, 35:513–524.

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 Operations Research*, 153:3–27.

Fernandes, R., Gouveia, B. ve Pinho, C., (2013). "A Real Options Approach To Labour Shifts Planning Under Different Service Level Targets", *European Journal of Operations Research*, 231(1):182–189.

Fowler, J. W., Wirojanagud, P. ve Gel, E. S., (2008). "Heuristics For Workforce Planning With Worker Differences" *European Journal of Operations Research*, 190:724–740.

Gomes da Silva, C., Figueira, J., Lisboa, J. ve Barman, S., (2006). "An Interactive Decision Support System For An Aggregate Production Planning Model Based On Multiple Criteria Mixed Integer Linear Programming", *Omega*, 34(2):167–177.

Gulsun, B., Tuzkaya, G., Tuzkaya, U. R. ve Onut, S., (2009). "An Aggregate Production Planning Strategy Selection Methodology Based On Linear Physical Programming", *International Journal of Industrial Engineering*, 16(2):135–146.

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S. ve Grossmann, I. E., (2014). "Scope For Industrial Applications Of Production Scheduling Models And Solution Methods", *Computers and Chemical Engineering*, 62:161–193.

Hertz, A., Lahrichi, N. ve Widmer, M., (2010). "A Flexible MILP Model For Multiple-shift Workforce Planning Under Annualized Hours", *European Journal of Operations Research*, 200(3): 860–873.

Ho, S. C. ve Leung J. M. Y., (2010). "Solving A Manpower Scheduling Problem For Airline Catering Using Metaheuristics", *European Journal of Operations Research*, 202(3):903–921.

KAYNAKLAR DİZİNİ (devam)

Hojati, M. ve Patil, A.S.,(2011). "An Integer Linear Programming based Heuristic For Scheduling Heterogeneous, Part time Service Employees", *European Journal of Operations Research*, 209(1):37–50.

Huang, H. J. ve Xu, G., (1998). "Aggregate Scheduling And Network Solving Of Multistage And Multi-item Manufacturing Systems", *European Journal of Operations Research*, 105:52–65.

Jaffry, S., ve Capon, N., (2005). "Alternative Methods Of Forecasting Risks In Naval Manpower Planning", *International Journal of Forecasting*, 2: 73–85.

Johnes, J., (2014). "Operational Research In Education", *European Journal of Operations Research*, 000:1–14.

Kabak, Ö., Ülengin, F., Aktaş, E., Önsel, Ş. ve Topcu, Y. I., (2008). "Efficient Shift Scheduling In The Retail Sector Through Two-stage Optimization", *European Journal of Operations Research*, 184:76–90.

Lapègue, T., Bellenguez-Morineau, O. ve Prot, D.A., (2013). "Constraint-based Approach For The Shift Design Personnel Task Scheduling Problem With Equity", *Computers and Operations Research*, 40(10):2450–2465.

Lee, S. ve Wtl, C., (1995). "CLXPRT: A Rule-Based Scheduling System", *Expert Systems with Applications*, 9(2):153–164.

Lequy, Q., Desaulniers, G. ve Solomon, M. M., (2012). "A Two-stage Heuristic For Multi activity And Task Assignment To Work Shifts", *Computers & Industrial Engineering*, 63(4):831–841.

Leung, S. C. H. ve Chan, S. S. W., (2009). "A Goal Programming Model For Aggregate Production Planning With Resource Utilization Constraint", *Computers & Industrial Engineering*, 56(3):1053–1064.

Li, F., Xu L., Da., Jin, C. ve Wang, H., (2012). "Random Assignment Method Based On Genetic Algorithms And Its Application In Resource Allocation", *Expert Systems with Applications*, 39:12213–12219.

Li, G., Jiang, H. ve He, T., (2014). "A Genetic Algorithm-based Decomposition Approach To Solve An Integrated Equipment-workforce-service Planning Problem", *Omega*, 50:1–17.

Lin, C. ve Gen, M., (2008). "Multi-criteria Human Resource Allocation For Solving Multistage Combinatorial Optimization Problems Using Multiobjective Hybrid Genetic Algorithm", *Expert Systems with Applications*, 34:2480–2490.

KAYNAKLAR DİZİNİ (devam)

- Lin, H., Chen, Y., Chou, T. ve Liao, Y., (2012). "Crew Rostering With Multiple Goals: An Empirical Study", *Computers & Industrial Engineering*, 63(2):483–493.
- L. Hidri, M. Labidi, (2016). Optimal physicians schedule in an Intensive Care Unit. *IOP Conf. Series: Materials Science and Engineering* 131, 1-8.
- Lusa, A. ve Pastor, R., (2011). "Planning Working Time Accounts Under Demand Uncertainty", *Computers and Operations Research*, 38(2):517–524.
- Maenhout, B. Vanhoucke, M. (2010). A hybrid scatter search heuristic for personalized crew rostering in the airline industry, *European Journal of Operational Research* 206 (2010) 155–167.
- 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.
- Mesquita, M., Moz, M., Paias, A. ve Pato, M., (2013). "A Decomposition Approach For The Integrated Vehicle-crew-roster Problem With Days off Pattern", *European Journal of Operations Research*, 229(2):318–331.
- Mirzapour Al-E-Hashem, S. M. J., Malekly, H. ve Aryanezhad, M. B, (2011). "A Multi objective Robust Optimization Model For Multi-product Multi-site Aggregate Production Planning In A Supply Chain Under Uncertainty" *International Journal of Production Economics* 134(1): 28–42.
- Mohan, S., (2008). "Scheduling Parttime Personnel With Availability Restrictions And Preferences To Maximize Employee Satisfaction", *Mathematical and Computer Modelling*, 48(11-12):1806–1813.
- Mohanty, R. P. ve Deshmukhb, S. G., (1997). "Evolution Of A Decision Support System For Human Resource Planning In A Petroleum Company", *International journal of production economics*, 51:251-261.
- Niehaus, R. J., (1995). "Evolution Of The Strategy And Structure Of A Human Resource Planning DSS Application", *Decision Support Systems*, 14:187–204.
- Ozdamar, L., Bozyel, M. A. ve Birbil, S. I., (1998). "A Hierarchical Decision Support System For Production Planning (With case study)", *European Journal of Operations Research*, 104(97): 403–422.
- Patel, V. ve Division, D., (1999). "Genetic Algorithm Based Approach for Scheduling", *Annals of the CIRP*, 48:7–10.
- Polat, O., Mutlu, Ö., ve Özgörmüş, E. (2018). A Mathematical Model For Assembly Line Balancing Problem Type 2 Under Ergonomic Workload Constraint. *The Ergonomics Open Journal*, 11(1), 1-11

KAYNAKLAR DİZİNİ (devam)

Sabar, M., Montreuil, B. ve Frayret, J. M., (2009). "A Multi-agent based Approach For Personnel Scheduling In Assembly Centers", *Engineering Applications of Artificial Intelligence*, 22(7):1080–1088.

Sadjadi, S. J., Soltani, R., Izadkhah, M., Saberian, F. ve Darayi, M., (2011). "A New Nonlinear Stochastic Staff Scheduling Model", *Scientia Iranica*, 18(3):699–710.

Smet, P., Wauters, T., Mihaylov, M. ve Vanden Berghe, G., (2014). "The Shift Minimisation Personnel Task Scheduling Problem: A New Hybrid Approach And Computational Insights", *Omega*, 46:64–73.

Stuckless, T., Milosevic, M., De Metz, C., Parliament, M., Tompkins, B. ve Brundage, M., (2012). "Managing A National Radiation Oncologist Workforce: A Workforce Planning Model", *Radiotherapy and Oncology*, 103(1):123–129.

Thompson, G. M. ve Goodale, J. C., (2006). "Variable Employee Productivity In Workforce Scheduling", *European Journal of Operations Research*, 170:376–390.

Toledo, C. F. M., De Oliveira, R. R. R. ve Morelato França, P., (2013). "A Hybrid Multi population Genetic Algorithm Applied To Solve The Multi-level Capacitated Lot Sizing Problem With Backlogging", *Computers and Operations Research*, 40:910–919.

Topaloglu, S., (2009). "A Shift Scheduling Model For Employees With Different Seniority Levels And An Application In Healthcare", *European Journal of Operations Research*, 198(3):943–957.

Ulusam Seçkiner, S., Gökçen, H. ve Kurt, M., (2007). "An Integer Programming Model For Hierarchical Workforce Scheduling Problem", *European Journal of Operations Research*, 183:694–699.

Valls, V., Pérez Á. ve Quintanilla, S., (2009). "Skilled Workforce Scheduling In Service Centres", *European Journal of Operations Research*, 193:791–804.

Van Nieuwenhuysse, I., De Boeck, L., Lambrecht, M. ve Vandaele, N. J., (2011). "Advanced Resource Planning As A Decision Support Module For ERP", *Computers in Industry*, 62(1):1-8.

Van Den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E. ve De Boeck, L., (2013). "Personnel scheduling: A Literature Review", *European Journal of Operations Research*, 226(3) 367–385.

Varlı, E., Eren, T. (2017). Hemşire Çizelgeleme Problemi ve Bir Hastanede Uygulama. *Akademik Platform Mühendislik ve Fen Bilimleri Dergisi*, 5(1), ss. 34-40.

Wang, K. J. ve Lin, Y. S., (2007). "Resource Allocation By Genetic Algorithm With Fuzzy Inference", *Expert Systems with Applications*, 33:1025–1035.

KAYNAKLAR DİZİNİ (devam)

Wang, R. C. ve Fang, H H., (2001). "Aggregate Production Planning With Multiple Objectives In A Fuzzy Environment", *European Journal of Operations Research*, 133:521–536.

Wong, T. C., Xu, M. ve Chin, K. S., (2014). "A Two-stage Heuristic Approach For Nurse Scheduling Problem: A Case Study In An Emergency Department", *Computers and Operations Research*, 51:99–110.

Yılman, N., (2007). Bir toplu üretim planlama modeli ve uygulaması, Yüksek Lisans Tezi, İTÜ Fen Bilimleri Enstitüsü, İstanbul.



EKLER

EK 1. Matematiksel Modelin Lingo 11.0 Programındaki Yazılışı ve Çözüm Raporları

$$\min=(C1*-(t1-A11)*X11+C2*-(t2-A12)*X12+C3*-(t3-A13)*X13+C4*-(t4-A14)*X14+C5*-(t5-A15)*X15+C1*-(t1-A21)*X21+C2*-(t2-A22)*X22+C3*-(t3-A23)*X23+C4*-(t4-A24)*X24+C5*-(t5-A25)*X25+C1*-(t1-A31)*X31+C2*-(t2-A32)*X32+C3*-(t3-A33)*X33+C4*-(t4-A34)*X34+C5*-(t5-A35)*X35+C1*-(t1-A41)*X41+C2*-(t2-A42)*X42+C3*-(t3-A43)*X43+C4*-(t4-A44)*X44+C5*-(t5-A45)*X45)+(1-z1)*(C1*(F12)+C1*(F13))+(1-z2)*(C2*(F23)+C2*(F21))+(1-z3)*(C3*(F34)+C3*(F32))+(1-z4)*(C4*(F45)+C4*(F43))+(1-z5)*(C5*(F53)+C5*(F54));$$

@BIN(X11);@BIN(X12); @BIN(X13); @BIN(X14); @BIN(X15);
@BIN(X21);@BIN(X22); @BIN(X23); @BIN(X24); @BIN(X25);
@BIN(X31);@BIN(X32); @BIN(X33); @BIN(X34); @BIN(X35);
@BIN(X41);@BIN(X42); @BIN(X43); @BIN(X44); @BIN(X45);
@BIN(Z1); @BIN(Z2);@BIN(Z3);@BIN(Z4);@BIN(Z5);@BIN(C1);@BIN(C2);
@BIN(C3); @BIN(C4);@BIN(C5);

! istasyon süreleri;

t1=162; t2=162; t3=138; t4=144;t5=144;

!joker personelin istasyon bazlı puanları;

q11=1.8;q12=1.8;q13=1.6;q14=1.6;q15=1.6;
q21=1.7;q22=1.7;q23=1.8;q24=1.8;q25=1.8;
q31=1.9;q32=1.9;q33=1.7;q34=1.7;q35=1.7;
q41=1.8;q42=1.8;q43=1.5;q44=1.5;q45=1.5;

! dağıtma katsayıları;

t3/(t3+t2)=K12; !1. istasyon gelmediğinde 2. İstasyona dağıtma kat sayısı;
t2/(t2+t3)=K13; !1. istasyon gelmediğinde 3. İstasyona dağıtma kat sayısı;
t3/(t3+t1)=K21; !2. istasyon gelmediğinde 1. İstasyona dağıtma kat sayısı;
t1/(t3+t1)=K23; !2. istasyon gelmediğinde 3. İstasyona dağıtma kat sayısı;
t4/(t4+t2)=K32; !3. istasyon gelmediğinde 2. İstasyona dağıtma kat sayısı;
t2/(t4+t2)=K34; !3. istasyon gelmediğinde 4. İstasyona dağıtma kat sayısı;
t5/(t5+t3)=K43; !4. istasyon gelmediğinde 3. İstasyona dağıtma kat sayısı;
t3/(t5+t3)=K45; !4. istasyon gelmediğinde 5. İstasyona dağıtma kat sayısı;
t3/(t3+t4)=K54; !5. istasyon gelmediğinde 4. İstasyona dağıtma kat sayısı;
t4/(t3+t4)=K53; !5. istasyon gelmediğinde 3. İstasyona dağıtma kat sayısı;

1-(X11+X21+X31+X41)<Z1*M; !1.İstasyona Z değişkeni joker personel atandığında 1 değerini alır;

1-(X12+X22+X32+X42)<Z2*M; !2.İstasyona Z değişkeni joker personel atandığında 0, atanmadığında 1 değerini alır;

$1-(X_{13}+X_{23}+X_{33}+X_{43}) < Z_3 * M$; !3. İstasyona Z değişkeni joker personel atandığında 0, atanmadığında 1 değerini alır;
 $1-(X_{14}+X_{24}+X_{34}+X_{44}) < Z_4 * M$; !4. İstasyona Z değişkeni joker personel atandığında 0, atanmadığında 1 değerini alır;
 $1-(X_{15}+X_{25}+X_{35}+X_{45}) < Z_5 * M$; !5. İstasyona Z değişkeni joker personel atandığında 0, atanmadığında 1 değerini alır;

$Z_1 * C_1 * t_1 * K_{12} + t_2 = F_{12}$;
 $Z_1 * C_1 * t_1 * K_{13} + t_3 = F_{13}$;
 $Z_2 * C_2 * t_2 * K_{21} + t_1 = F_{21}$;
 $Z_2 * C_2 * t_2 * K_{23} + t_3 = F_{23}$;
 $Z_3 * C_3 * t_3 * K_{32} + t_2 = F_{32}$;
 $Z_3 * C_3 * t_3 * K_{34} + t_4 = F_{34}$;
 $Z_4 * C_4 * t_4 * K_{43} + t_3 = F_{43}$;
 $Z_4 * C_4 * t_4 * K_{45} + t_5 = F_{45}$;
 $Z_5 * C_5 * t_5 * K_{54} + t_4 = F_{54}$;
 $Z_5 * C_5 * t_5 * K_{53} + t_3 = F_{53}$;

! Yeterli joker personel olmaması durumunda işe gelmeyen personelin istasyonu;

$A_{11} = X_{11} * (q_{11} * t_1)$;
 $A_{21} = X_{21} * (q_{21} * t_1)$;
 $A_{31} = X_{31} * (q_{31} * t_1)$;
 $A_{41} = X_{41} * (q_{41} * t_1)$;
 $A_{12} = X_{12} * (q_{12} * t_2)$;
 $A_{22} = X_{22} * (q_{22} * t_2)$;
 $A_{32} = X_{32} * (q_{32} * t_2)$;
 $A_{42} = X_{42} * (q_{42} * t_2)$;
 $A_{13} = X_{13} * (q_{13} * t_3)$;
 $A_{23} = X_{23} * (q_{23} * t_3)$;
 $A_{33} = X_{33} * (q_{33} * t_3)$;
 $A_{43} = X_{43} * (q_{43} * t_3)$;
 $A_{14} = X_{14} * (q_{14} * t_4)$;
 $A_{24} = X_{24} * (q_{24} * t_4)$;
 $A_{34} = X_{34} * (q_{34} * t_4)$;
 $A_{44} = X_{44} * (q_{44} * t_4)$;
 $A_{15} = X_{15} * (q_{15} * t_5)$;
 $A_{25} = X_{25} * (q_{25} * t_5)$;
 $A_{35} = X_{35} * (q_{35} * t_5)$;
 $A_{45} = X_{45} * (q_{45} * t_5)$;

!joker personelin istasyonlara atanması durumunda oluşan yeni çevrim süreleri;

$C_1 * X_{11} + C_2 * X_{12} + C_3 * X_{13} + C_4 * X_{14} + C_5 * X_{15} \leq 1$; ! 1. Joker personel Ekibin en fazla bir istasyonuna atanabilir;
 $C_1 * X_{21} + C_2 * X_{22} + C_3 * X_{23} + C_4 * X_{24} + C_5 * X_{25} \leq 1$; ! 2. Joker personel Ekibin en fazla bir istasyonuna atanabilir;
 $C_1 * X_{31} + C_2 * X_{32} + C_3 * X_{33} + C_4 * X_{34} + C_5 * X_{35} \leq 1$; ! 3. Joker personel Ekibin en fazla bir istasyonuna atanabilir;
 $C_1 * X_{41} + C_2 * X_{42} + C_3 * X_{43} + C_4 * X_{44} + C_5 * X_{45} \leq 1$; ! 4. Joker personel Ekibin en fazla bir istasyonuna atanabilir;

$(C1*X11+C1*X21+C1*X31+C1*X41) \leq 1$; ! 1. İstasyonuna sadece bir joker personel atanabilir;
 $(C2*X12+C2*X22+C2*X32+C2*X42) \leq 1$; ! 2. İstasyonuna sadece bir joker personel atanabilir;
 $(C3*X13+C3*X23+C3*X33+C3*X43) \leq 1$; ! 3. İstasyonuna sadece bir joker personel atanabilir;
 $(C4*X14+C4*X24+C4*X34+C4*X44) \leq 1$; ! 4. İstasyonuna sadece bir joker personel atanabilir;
 $(C5*X15+C5*X25+C5*X35+C5*X45) \leq 1$; ! 5. İstasyonuna sadece bir joker personel atanabilir;

$(C1*X11+C1*X21+C1*X31+C1*X41+C2*X12+C2*X22+C2*X32+C2*X42+C3*X13+C3*X23+C3*X33+C3*X43+C4*X14+C4*X24+C4*X34+C4*X44+C5*X15+C5*X25+C5*X35+C5*X45) = @SMIN(J, S)$;

! Bu kısıt ile ekibin eksik istasyonu kadar joker personel atanabilir. Yeterli joker personelin olmaması durumunda ise var olan joker personelin en uygun istasyona atanması sağlanmıştır.

$X11+X12+X13+X14+X15+$
 $X21+X22+X23+X24+X25+$
 $X31+X32+X33+X34+X35+$
 $X41+X42+X43+X44+X45 \leq J$;

! Bu kısıt ile atanan joker personelin mevcut durumda ki joker personel sayısı kadar olması sağlanmıştır;

$S=2$;
 $M=1000000$;
 $J=3$;

$X11+X21+X31+X41 = (1-Z1)*C1$;
 $X12+X22+X32+X42 = (1-Z2)*C2$;
 $X13+X23+X33+X43 = (1-Z3)*C3$;
 $X14+X24+X34+X44 = (1-Z4)*C4$;
 $X15+X25+X35+X45 = (1-Z5)*C5$;

! Bu kısıtlar ile yeterli joker personel olmaması durumunda işe gelmeyen istasyona joker personel atanması sağlanmıştır;

$C1=0$; $C2=0$; $C3=1$; $C4=0$; $C5=1$; !işe gelmeyen istasyonlar;

End

Senaryo 1: Joker personel sayısı işe gelmeyen personel sayısından daha fazla olması durumunda modelin çözümü;

Local optimal solution found.

Objective value: 742.8000
Objective bound: 742.8000
Infeasibilities: 0.000000
Extended solver steps: 18
Total solver iterations: 2846

Variable	Value
X11	0.000000
X12	0.000000
X13	1.000000
X14	0.000000
X15	0.000000
X21	0.000000
X22	0.000000
X23	0.000000
X24	0.000000
X25	0.000000
X31	0.000000
X32	0.000000
X33	0.000000
X34	0.000000
X35	0.000000
X41	0.000000
X42	0.000000
X43	0.000000
X44	0.000000
X45	1.000000
Z1	1.000000
Z2	1.000000
Z3	0.000000
Z4	1.000000
Z5	0.000000
C1	0.000000
C2	0.000000
C3	1.000000
C4	0.000000
C5	1.000000
T1	162.0000
T2	162.0000
T3	138.0000
T4	144.0000
T5	144.0000
Q11	1.800000
Q12	1.800000
Q13	1.600000
Q14	1.600000
Q15	1.600000
Q21	1.700000
Q22	1.700000
Q23	1.800000
Q24	1.800000

Q25	1.800000
Q31	1.900000
Q32	1.900000
Q33	1.700000
Q34	1.700000
Q35	1.700000
Q41	1.800000
Q42	1.800000
Q43	1.500000
Q44	1.500000
Q45	1.500000
K12	0.460000
K13	0.540000
K21	0.460000
K23	0.540000
K32	0.4705882
K34	0.5294118
K43	0.5106383
K45	0.4893617
K54	0.4893617
K53	0.5106383
M	1000000.
F12	162.0000
F13	138.0000
F21	162.0000
F23	138.0000
F32	162.0000
F34	144.0000
F43	138.0000
F45	144.0000
F54	144.0000
F53	138.0000
A11	0.000000
A21	0.000000
A31	0.000000
A41	0.000000
A12	0.000000
A22	0.000000
A32	0.000000
A42	0.000000
A13	220.8000
A23	0.000000
A33	0.000000
A43	0.000000
A14	0.000000
A24	0.000000
A34	0.000000
A44	0.000000
A15	0.000000
A25	0.000000
A35	0.000000

A45	216.0000
J	3.000000
S	2.000000

Row	Slack or Surplus
1	0.000000
2	0.000000
3	0.000000
4	0.000000
5	0.000000
6	0.000000
7	0.000000
8	0.000000
9	0.000000
10	0.000000
11	0.000000
12	0.000000
13	0.000000
14	0.000000
15	0.000000
16	0.000000
17	0.000000
18	0.000000
19	0.000000
20	0.000000
21	0.000000
22	0.000000
23	0.000000
24	0.000000
25	0.000000
26	0.000000
27	0.000000
28	0.000000
29	0.000000
30	0.000000
31	0.000000
32	0.000000
33	0.000000
34	0.000000
35	0.000000
36	999999.0
37	999999.0
38	0.000000
39	999999.0
40	0.000000
41	0.000000
42	0.000000
43	0.000000
44	0.000000
45	0.000000
46	0.000000

47	0.000000
48	0.000000
49	0.000000
50	0.000000
51	0.000000
52	0.000000
53	0.000000
54	0.000000
55	0.000000
56	0.000000
57	0.000000
58	0.000000
59	0.000000
60	0.000000
61	0.000000
62	0.000000
63	0.000000
64	0.000000
65	0.000000
66	0.000000
67	0.000000
68	0.000000
69	0.000000
70	0.000000
71	0.000000
72	1.000000
73	1.000000
74	0.000000
75	1.000000
76	1.000000
77	0.000000
78	1.000000
79	0.000000
80	0.000000
81	1.000000
82	0.000000
83	0.000000
84	0.000000
85	0.000000
86	0.000000
87	0.000000
88	0.000000
89	0.000000
90	0.000000
91	0.000000
92	0.000000
93	0.000000
94	0.000000
95	742.8000

Senaryo 2: Joker personel sayısı işe gelmeyen personel sayısından daha az olması durumunda modelin çözümü;

Local optimal solution found.

Objective value: 354.0000
Objective bound: 354.0000
Infeasibilities: 0.000000
Extended solver steps: 9
Total solver iterations: 1762

Variable	Value	Reduced Cost
X11	0.000000	0.000000
X12	0.000000	0.000000
X13	0.000000	0.000000
X14	0.000000	0.000000
X15	0.000000	0.000000
X21	0.000000	0.000000
X22	0.000000	0.000000
X23	0.000000	0.000000
X24	0.000000	0.000000
X25	0.000000	-0.1979140E-07
X31	0.000000	0.000000
X32	0.000000	0.000000
X33	0.000000	0.000000
X34	0.000000	0.000000
X35	0.000000	-0.1971148E-07
X41	0.000000	0.000000
X42	0.000000	0.000000
X43	0.000000	0.000000
X44	0.000000	0.000000
X45	1.000000	431.9994
Z1	1.000000	0.000000
Z2	1.000000	0.000000
Z3	1.000000	-306.0000
Z4	1.000000	0.000000
Z5	0.000000	0.000000
C1	0.000000	0.000000
C2	0.000000	0.000000
C3	1.000000	0.000000
C4	0.000000	0.000000
C5	1.000000	0.000000
T1	162.0000	0.000000
T2	162.0000	0.000000
T3	138.0000	0.000000
T4	144.0000	0.000000
T5	144.0000	0.000000
Q11	1.800000	0.000000

Q12	1.800000	0.000000
Q13	1.600000	0.000000
Q14	1.600000	0.000000
Q15	1.600000	0.000000
Q21	1.700000	0.000000
Q22	1.700000	0.000000
Q23	1.800000	0.000000
Q24	1.800000	0.000000
Q25	1.800000	0.000000
Q31	1.900000	0.000000
Q32	1.900000	0.000000
Q33	1.700000	0.000000
Q34	1.700000	0.000000
Q35	1.700000	0.000000
Q41	1.800000	0.000000
Q42	1.800000	0.000000
Q43	1.500000	0.000000
Q44	1.500000	0.000000
Q45	1.500000	0.000000
K12	0.4600000	0.000000
K13	0.5400000	0.000000
K21	0.4600000	0.000000
K23	0.5400000	0.000000
K32	0.4705882	0.000000
K34	0.5294118	0.000000
K43	0.5106383	0.000000
K45	0.4893617	0.000000
K54	0.4893617	0.000000
K53	0.5106383	0.000000
M	1000000.	0.000000
F12	162.0000	0.000000
F13	138.0000	0.000000
F21	162.0000	0.000000
F23	138.0000	0.000000
F32	226.9412	0.000000
F34	217.0588	0.000000
F43	138.0000	0.000000
F45	144.0000	0.000000
F54	144.0000	0.000000
F53	138.0000	0.000000
A11	0.000000	0.000000
A21	0.000000	0.000000
A31	0.000000	0.000000
A41	0.000000	0.000000
A12	0.000000	0.000000
A22	0.000000	0.000000
A32	0.000000	0.000000
A42	0.000000	0.000000
A13	0.000000	0.000000
A23	0.000000	0.000000
A33	0.000000	0.000000

A43	0.000000	0.000000
A14	0.000000	0.000000
A24	0.000000	0.000000
A34	0.000000	0.000000
A44	0.000000	0.000000
A15	0.000000	0.000000
A25	0.000000	0.000000
A35	0.000000	0.000000
A45	216.0000	0.000000
J	1.000000	0.000000
S	2.000000	0.000000

Row	Slack or Surplus	Dual Price
1	0.000000	0.000000
2	0.000000	0.000000
3	0.000000	-1.000000
4	0.000000	-1.000000
5	0.000000	-0.4999985
6	0.000000	0.000000
7	0.000000	0.000000
8	0.000000	0.000000
9	0.000000	0.000000
10	0.000000	0.000000
11	0.000000	0.000000
12	0.000000	0.000000
13	0.000000	0.000000
14	0.000000	0.000000
15	0.000000	0.000000
16	0.000000	0.000000
17	0.000000	0.000000
18	0.000000	0.000000
19	0.000000	0.000000
20	0.000000	0.000000
21	0.000000	0.000000
22	0.000000	0.000000
23	0.000000	0.000000
24	0.000000	0.000000
25	0.000000	-143.9999
26	0.000000	0.000000
27	0.000000	0.000000
28	0.000000	0.000000
29	0.000000	0.000000
30	0.000000	0.000000
31	0.000000	0.000000
32	0.000000	0.000000
33	0.000000	0.000000
34	0.000000	0.000000
35	0.000000	0.000000
36	999999.0	0.000000
37	999999.0	0.000000
38	999999.0	0.000000

39	999999.0	0.000000
40	0.000000	0.000000
41	0.000000	0.000000
42	0.000000	0.000000
43	0.000000	0.000000
44	0.000000	0.000000
45	0.000000	0.000000
46	0.000000	0.000000
47	0.000000	0.000000
48	0.000000	0.000000
49	0.000000	1.000000
50	0.000000	1.000000
51	0.000000	0.000000
52	0.000000	0.000000
53	0.000000	0.000000
54	0.000000	0.000000
55	0.000000	0.000000
56	0.000000	0.000000
57	0.000000	0.000000
58	0.000000	0.000000
59	0.000000	0.000000
60	0.000000	0.000000
61	0.000000	0.000000
62	0.000000	0.000000
63	0.000000	0.000000
64	0.000000	0.000000
65	0.000000	0.000000
66	0.000000	0.000000
67	0.000000	-0.3533764E-06
68	0.000000	-0.3140742E-06
69	0.000000	-0.3325493E-06
70	0.000000	-0.9999990
71	1.000000	0.000000
72	1.000000	0.000000
73	1.000000	0.000000
74	0.000000	0.000000
75	1.000000	0.000000
76	1.000000	0.000000
77	1.000000	0.000000
78	1.000000	0.000000
79	0.000000	5.999837
80	0.000000	0.000000
81	0.000000	0.000000
82	0.000000	0.000000
83	0.000000	0.000000
84	0.000000	0.000000
85	0.000000	0.000000
86	0.000000	0.000000
87	0.000000	138.0000
88	0.000000	0.000000
89	0.000000	138.0000

90	0.000000	0.000000
91	0.000000	0.000000
92	0.000000	0.000000
93	0.000000	0.000000
94	0.000000	-221.9998
95	354.0000	-1.000000

Senaryo 3; Joker personel sayısı işe gelmeyen personel sayısına eşit olması durumunda 1 çözümü;

Local optimal solution found.

Objective value: 742.8000
Objective bound: 742.8000
Infeasibilities: 0.000000
Extended solver steps: 18
Total solver iterations: 2816

Variable	Value
X11	0.000000
X12	0.000000
X13	1.000000
X14	0.000000
X15	0.000000
X21	0.000000
X22	0.000000
X23	0.000000
X24	0.000000
X25	0.000000
X31	0.000000
X32	0.000000
X33	0.000000
X34	0.000000
X35	0.000000
X41	0.000000
X42	0.000000
X43	0.000000
X44	0.000000
X45	1.000000
Z1	1.000000
Z2	1.000000
Z3	0.000000
Z4	1.000000
Z5	0.000000
C1	0.000000
C2	0.000000
C3	1.000000
C4	0.000000

C5	1.000000
T1	162.0000
T2	162.0000
T3	138.0000
T4	144.0000
T5	144.0000
Q11	1.800000
Q12	1.800000
Q13	1.600000
Q14	1.600000
Q15	1.600000
Q21	1.700000
Q22	1.700000
Q23	1.800000
Q24	1.800000
Q25	1.800000
Q31	1.900000
Q32	1.900000
Q33	1.700000
Q34	1.700000
Q35	1.700000
Q41	1.800000
Q42	1.800000
Q43	1.500000
Q44	1.500000
Q45	1.500000
K12	0.4600000
K13	0.5400000
K21	0.4600000
K23	0.5400000
K32	0.4705882
K34	0.5294118
K43	0.5106383
K45	0.4893617
K54	0.4893617
K53	0.5106383
M	1000000.
F12	162.0000
F13	138.0000
F21	162.0000
F23	138.0000
F32	162.0000
F34	144.0000
F43	138.0000
F45	144.0000
F54	144.0000
F53	138.0000
A11	0.000000
A21	0.000000
A31	0.000000
A41	0.000000

A12	0.000000
A22	0.000000
A32	0.000000
A42	0.000000
A13	220.8000
A23	0.000000
A33	0.000000
A43	0.000000
A14	0.000000
A24	0.000000
A34	0.000000
A44	0.000000
A15	0.000000
A25	0.000000
A35	0.000000
A45	216.0000
J	2.000000
S	2.000000

Row	Slack or Surplus
1	0.000000
2	0.000000
3	0.000000
4	0.000000
5	0.000000
6	0.000000
7	0.000000
8	0.000000
9	0.000000
10	0.000000
11	0.000000
12	0.000000
13	0.000000
14	0.000000
15	0.000000
16	0.000000
17	0.000000
18	0.000000
19	0.000000
20	0.000000
21	0.000000
22	0.000000
23	0.000000
24	0.000000
25	0.000000
26	0.000000
27	0.000000
28	0.000000
29	0.000000
30	0.000000
31	0.000000

32	0.000000
33	0.000000
34	0.000000
35	0.000000
36	999999.0
37	999999.0
38	0.000000
39	999999.0
40	0.000000
41	0.000000
42	0.000000
43	0.000000
44	0.000000
45	0.000000
46	0.000000
47	0.000000
48	0.000000
49	0.000000
50	0.000000
51	0.000000
52	0.000000
53	0.000000
54	0.000000
55	0.000000
56	0.000000
57	0.000000
58	0.000000
59	0.000000
60	0.000000
61	0.000000
62	0.000000
63	0.000000
64	0.000000
65	0.000000
66	0.000000
67	0.000000
68	0.000000
69	0.000000
70	0.000000
71	0.000000
72	1.000000
73	1.000000
74	0.000000
75	1.000000
76	1.000000
77	0.000000
78	1.000000
79	0.000000
80	0.000000
81	0.000000
82	0.000000

83	0.000000
84	0.000000
85	0.000000
86	0.000000
87	0.000000
88	0.000000
89	0.000000
90	0.000000
91	0.000000
92	0.000000
93	0.000000
94	0.000000
95	742.8000



EK 2. Personel Devamsızlığında Atama ve Dengeleme Problemi İçin Karar Destek Sistemi Algoritmalarının C# Dilinde Kodları ve Açıklamaları

```
private void button3_Click(object sender, EventArgs e) // OTOMATİK ATAMA
BUTONU FONKSİYONLARI
{
    // otomatik ata butonu

    try
    {
        string queryBul = "select EkipID from Çalışan where SicilNo=@sicilno"; //Buradaki
try-catch bloğunda
        SqlCommand cmdBul = new SqlCommand(queryBul, con); //gelmeyen
kişinin ekip id si bulunuyor.
        con.Open(); // queryBul veritabanına sorguyu
yolluyor.
        cmdBul.Parameters.AddWithValue("@sicilno", sicilno);
        ekipidOto = (Int32)cmdBul.ExecuteScalar();
        cmdBul.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }

    try
    {
        string today = DateTime.Now.ToString("yyyy/MM/dd");
        string queryZero = "select isnull(min(c.ID),0) from Çalışan c left join Devamsızlık d
on c.ID=d.ÇalışanID where isJoker=1 and EkipID is null and (tarih>@today or tarih!=@today
or tarih is null)";
        string queryJoker = "select top 1 c.ID," + selectedIndex + " from Çalışan c left join
Devamsızlık d on c.ID=d.ÇalışanID where isJoker=1 and c.IstasyonID is null and EkipID is null
and (tarih>@today or tarih!=@today or tarih is null) ORDER BY cast(" + selectedIndex + " as
int) DESC";
        SqlCommand cmdJoker = new SqlCommand(queryJoker, con);
        SqlCommand cmdZero = new SqlCommand(queryZero, con);
        con.Open();
        //cmdJoker.Parameters.AddWithValue("@column",selectedIndex);
        cmdJoker.Parameters.AddWithValue("@today", today);
        cmdZero.Parameters.AddWithValue("@today",today);
        tempjoker = (Int32)cmdZero.ExecuteScalar();
        if (tempjoker == 0)
            selectedJoker = 0;
        else
    }
```

```

        selectedJoker = (Int32)cmdJoker.ExecuteScalar();    // Buradaki try-catch bloğu
joker personel sayısı ve o gün işte olup olmama durumları
        // kontrol ediliyor ve eğer sayı yeterli ise işe en
uygun joker personel seçiliyor.
        // queryJoker personel en uygun çalışanı seçiyor
queryZero uygun çalışan olup olmadığına bakıyor.
        cmdZero.ExecuteNonQuery();
        cmdJoker.ExecuteNonQuery();
        con.Close();
    } catch (Exception ex) { MessageBox.Show(ex.Message); }
    try
    {
        string queryFindIstasyon = "select i.ID from Istasyon i inner join Çalışan c on
i.ID=c.IstasyonID where c.SicilNo=@sicilno";
        SqlCommand cmdIstasyon = new SqlCommand(queryFindIstasyon, con);
        con.Open();
        cmdIstasyon.Parameters.AddWithValue("@sicilno", sicilno); // Buradaki try-catch
bloğunda gelmeyen personelin istasyonu belirleniyor.
        istasyonID = (Int32)cmdIstasyon.ExecuteScalar();    // queryFindIstasyon gerekli
sorguyu yapıyor.
        cmdIstasyon.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }

    if (selectedJoker == 0)    // Joker personel sayısı kontrol ediliyor, yeterli değilse
istasyonlara iş dağıtılıyor.
    {
        MessageBox.Show("Yeterli joker personel bulunmamaktadır. İş yakın istasyonlara
paylaştırılacaktır.");
        istasyonOto(istasyonID); // İstasyonlara otomatik dengeleme yapacak fonksiyon
çağırılıyor.
    }
    else    // Joker personel sayısı yeterli ise boş olan yere joker personel
atanıyor.
    {
        try
        {
            string queryOto = "update Çalışan set EkipID=@ekipid,IstasyonID=@istasyo
where ID=@id";
            SqlCommand cmdOto = new SqlCommand(queryOto, con);
            con.Open();

            cmdOto.Parameters.AddWithValue("@id", selectedJoker);
            cmdOto.Parameters.AddWithValue("@ekipid", ekipidOto);

```



```

        cmdOto.Parameters.AddWithValue("@istasyonid", istasyonID);
        cmdOto.ExecuteNonQuery(); // Buradaki try-catch
        // bloğunda da joker personel ataması gerçekleşiyor.
        con.Close(); // queryOto veritabanına değişiklikleri
        // yollayan sorguyu tutuyor
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
}
MessageBox.Show("Atama Tamamlandı.");
}
public void istasyonOto(int emptyStation) { // İSTASYONLARA ATAMA YAPACAK
// FONKSİYON
    int emptyBolumId = checkBolum(emptyStation);
    int maxIstasyon = findMaxID(emptyBolumId);
    if(emptyStation==1) //Atama yapılacak istasyon 1. istasyon ise
    { //kontrolFirst fonksiyonu çağırılıyor
        kontrolFirst(emptyStation);
    }
    else if (emptyStation == maxIstasyon) // Atama yapılacak istasyon son istasyon ise
    { // kontrolLast fonksiyonu çağırılıyor.
        kontrolLast(emptyStation);
    }
    else // Eğer ilk veya son istasyon değilse normal atama yapacak
    olan
    { // kontrol fonksiyonu çağırılıyor.
        kontrol(emptyStation);
    }
}
public int checkBolum(int id) { // checkBolum atama yapılacak istasyonun
// sürecinin olup olmadığı kontrol ediyor.
    int result = 0;
    try
    {
        string query = "select b.ID from Bölümler b inner join Istasyon i on b.ID=i.BolumID
        where i.ID=@id";
        SqlCommand cmdEmpty = new SqlCommand(query,con);
        cmdEmpty.Parameters.AddWithValue("@id",id);
        con.Open();
        result = (Int32)cmdEmpty.ExecuteScalar();
        cmdEmpty.ExecuteNonQuery();
        con.Close();

    }catch(Exception ex) { MessageBox.Show(ex.Message); }
    return result;
}
}

```

```

public int findMaxID(int id) { // FindMaxID boş istasyonun ait olduğu süre
toplam istasyon sayısını buluyor.
    int result = 0;
    try
    { //select isnull(max(i.ID),0) from Istasyon i inner join Bölümler b on i.BolumID=b.ID
where b.ID=@id
        string query = "select isnull(max(i.ID),0) from Istasyon i inner join Bölümler b on
i.BolumID=b.ID where b.ID=@id";
        SqlCommand cmdMax = new SqlCommand(query,con);
        cmdMax.Parameters.AddWithValue("@id",id);
        con.Open();
        result = (Int32)cmdMax.ExecuteScalar();
        cmdMax.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
    return result;
}

```

```

public void kontrol(int id) // NORMAL İSTASYON ATAMA FONKSİYONU
{
    int hedefArtı = id + 1; // Bir sonraki istasyonun id si
    int hedefEksi = id - 1; // bir önceki istasyonun id si
    int artı=0, eksi=0;
    try // try-catch bloğunda bir sağ istasyonun devamsızlık durumu
kontrol ediliyor.
    {
        string today = DateTime.Now.ToString("yyyy/MM/dd");
        string query = "select isnull(min(d.ID),0) from Çalışan c inner join Devamsızlık d
on c.ID=d.ÇalışanID where c.IstasyonID=@id and tarih=@today";
        SqlCommand cmdKontrolArtı = new SqlCommand(query,con);
        cmdKontrolArtı.Parameters.AddWithValue("@id",hedefArtı);
        cmdKontrolArtı.Parameters.AddWithValue("@today",today);
        con.Open();
        artı = (Int32)cmdKontrolArtı.ExecuteScalar();
        cmdKontrolArtı.ExecuteNonQuery();
        con.Close();
    }catch(Exception ex) { MessageBox.Show(ex.Message); } // mart değişiklik

    if (artı == 0) // Eğer uygunsa atama yapılıyor
        ataIstasyon(hedefArtı,id);
    else
        MessageBox.Show("Çalışanın sağındaki istasyon uygun değil!");

    try // try-catch bloğu istasyonun solundaki istasyonun devamsızlık
durumunu kontrol ediyor.

```

```

    {
        string today = DateTime.Now.ToString("yyyy/MM/dd");
        string query = "select isnull(min(d.ID),0) from Çalışan c inner join Devamsızlık d
on c.ID=d.ÇalışanID where c.IstasyonID=@id and tarih=@today";
        SqlCommand cmdKontrolEksi = new SqlCommand(query, con);
        cmdKontrolEksi.Parameters.AddWithValue("@id", hedefEksi);
        cmdKontrolEksi.Parameters.AddWithValue("@today",today);
        con.Open();
        eksi = (Int32)cmdKontrolEksi.ExecuteScalar();
        cmdKontrolEksi.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); } //mart değişiklik

    if (eksi == 0) // Eğer uygunsa atama yapıyor
        ataIstasyon(hedefEksi,id);
    else
        MessageBox.Show("Çalışanın solundaki istasyon uygun değil!");
        istasyonKayit(hedefArtı,hedefEksi); // değişiklik yapılan istasyon bilgisi
        veritabana kaydeden fonksiyon
        istasyonSurePaylasim(hedefArtı,hedefEksi,id); // istasyon süre paylaşım donksiyonu
    }
    public void kontrolFirst(int id) { // ilk istasyona atama yapılırken
        int hedefArtı = id + 1; // bir sağındaki istasyon id si
        int hedefArtıPlus = id + 2; // iki sağındaki istasyon id si
        int artı = 0, artıPlus = 0;
        try // try-catch bloğunda bir sağ istasyonun devamsızlık durumu kontrol ediliyor
        {
            string today = DateTime.Now.ToString("yyyy/MM/dd");
            string query = "select isnull(min(d.ID),0) from Çalışan c inner join Devamsızlık d on
c.ID=d.ÇalışanID where c.IstasyonID=@id and tarih=@today";
            SqlCommand cmdKontrolArtı = new SqlCommand(query, con);
            cmdKontrolArtı.Parameters.AddWithValue("@id", hedefArtı);
            cmdKontrolArtı.Parameters.AddWithValue("@today", today);
            con.Open();
            artı = (Int32)cmdKontrolArtı.ExecuteScalar();
            cmdKontrolArtı.ExecuteNonQuery();
            con.Close();
        }
        catch (Exception ex) { MessageBox.Show(ex.Message); } // değişiklik

        if (artı == 0) // Eğer uygunsa atama yap
            ataIstasyon(hedefArtı, id);
        else
            MessageBox.Show("Çalışanın sağındaki istasyon uygun değil!");
    }

```

```

try // try-catch bloğunda istasyonun iki sağındaki istasyonun devamsızlık durumu
kontrol ediliyor.
{
    string today = DateTime.Now.ToString("yyyy/MM/dd");
    string query = "select isnull(min(d.ID),0) from Çalışan c inner join Devamsızlık d on
c.ID=d.ÇalışanID where c.IstasyonID=@id and tarih=@today";
    SqlCommand cmdKontrolEksi = new SqlCommand(query, con);
    cmdKontrolEksi.Parameters.AddWithValue("@id", hedefArtıPlus);
    cmdKontrolEksi.Parameters.AddWithValue("@today", today);
    con.Open();
    artıPlus = (Int32)cmdKontrolEksi.ExecuteScalar();
    cmdKontrolEksi.ExecuteNonQuery();
    con.Close();
}
catch (Exception ex) { MessageBox.Show(ex.Message); } // değişiklik

if (artıPlus == 0) // Eğer uygunsuz atama yapılıyor.
    ataIstasyon(hedefArtıPlus, id);
else
    MessageBox.Show("Çalışanın 2 sağındaki istasyon uygun değil!");

istasyonKayıt(hedefArtı,hedefArtıPlus); // istasyon kayıt fonksiyonu.
istasyonSurePaylasim(hedefArtı,hedefArtıPlus,id); // istasyon süre paylaşım fonksiyonu.
}
public void kontrolLast(int id) // SON İSTASYONA ATAMA YAPILIRKEN
{
    int hedefEksiPlus = id - 2; // bir soldaki istasyon id
    int hedefEksi = id - 1; // iki soldaki istasyon id
    int eksiPlus = 0, eksi = 0;
    try // try-catch bloğu bir soldaki istasyonun devamsızlık durumunu kontrol
    ediyor.
    {
        string today = DateTime.Now.ToString("yyyy/MM/dd");
        string query = "select isnull(min(d.ID),0) from Çalışan c inner join Devamsızlık d on
c.ID=d.ÇalışanID where c.IstasyonID=@id and tarih=@today";
        SqlCommand cmdKontrolEksi = new SqlCommand(query, con);
        cmdKontrolEksi.Parameters.AddWithValue("@id", hedefEksi);
        cmdKontrolEksi.Parameters.AddWithValue("@today", today);
        con.Open();
        eksi = (Int32)cmdKontrolEksi.ExecuteScalar();
        cmdKontrolEksi.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); } // mart değişiklik

    if (eksi == 0) // Eğer uygunsuz atama yapıyor

```

```

        ataIstasyon(hedefEksi, id);
    else
        MessageBox.Show("Çalışanın solundaki istasyon uygun değil!");
    try // try-catch fonksiyonu istasyonun iki solundaki istasyonun devamsızlık
durumunu kontrol ediyor.
    {
        string today = DateTime.Now.ToString("yyyy/MM/dd");
        string query = "select isnull(min(d.ID),0) from Çalışan c inner join Devamsızlık d on
c.ID=d.ÇalışanID where c.IstasyonID=@id and tarih=@today";
        SqlCommand cmdKontrolArtı = new SqlCommand(query, con);
        cmdKontrolArtı.Parameters.AddWithValue("@id", hedefEksiPlus);
        cmdKontrolArtı.Parameters.AddWithValue("@today", today);
        con.Open();
        ekşiPlus = (Int32)cmdKontrolArtı.ExecuteScalar();
        cmdKontrolArtı.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); } // mart değişiklik

    if (ekşiPlus == 0) // eğer uygunsuz atama yapıyor.
        ataIstasyon(hedefEksiPlus, id);
    else
        MessageBox.Show("Çalışanın 2 solundaki istasyon uygun değil!");
        istasyonKayit(hedefEksi,hedefEksiPlus);
        istasyonSurePaylasim(hedefEksi,hedefEksiPlus,id); // süre paylaşım
    }
    public void ataIstasyon(int value,int istasyonid) { // İSTASYONA ATAMA
TAMAMLAMA FONKSİYONU
        int getCalisanID=0;
        int check=0;
        try // try-catch bloğunda istasyona vekil atanacak personelin idsi bulunuyor
        {
            string query = "select ID from Çalışan where IstasyonID=@istasyonid";
            SqlCommand cmdBul = new SqlCommand(query,con);
            cmdBul.Parameters.AddWithValue("@istasyonid",value);
            con.Open();
            getCalisanID = (Int32)cmdBul.ExecuteScalar();
            cmdBul.ExecuteNonQuery();
            con.Close();
        }catch(Exception ex) { MessageBox.Show(ex.Message); }
        try // try-catch bloğunda çalışanın solundan vekil olduğu istasyon var mı kontrol
ediliyor
        {
            string query = "select isnull(VekilID,0) from Istasyon where ID=@id";
            SqlCommand cmdCheck = new SqlCommand(query,con);
            cmdCheck.Parameters.AddWithValue("@id",istasyonid);

```

```

con.Open();
check = (Int32)cmdCheck.ExecuteScalar();
cmdCheck.ExecuteNonQuery();
con.Close();
} catch (Exception ex) { MessageBox.Show(ex.Message); }
if (check == 0) // eğer solundan vekil olduğu istasyon yok ise atama yapılıyor
{
    try
    {
        string queryata = "update Istasyon set Vekil1ID=@calisanid where ID=@id";
        SqlCommand cmdata = new SqlCommand(queryata, con);
        cmdata.Parameters.AddWithValue("@calisanid", getCalisanID);
        cmdata.Parameters.AddWithValue("@id", istasyonid);
        con.Open();
        cmdata.ExecuteNonQuery();
        con.Close();
        //MessageBox.Show("Atama tamamlandı");
    } catch (Exception ex) { MessageBox.Show(ex.Message); }
}
else // eğer solundan vekil ise sağ taraftan kontrol ediliyor
{ // vekil ataması tamamlanıyor.

    try
    {
        string queryata = "update Istasyon set Vekil2ID=@calisanid where ID=@id";
        SqlCommand cmdata = new SqlCommand(queryata, con);
        cmdata.Parameters.AddWithValue("@calisanid", getCalisanID);
        cmdata.Parameters.AddWithValue("@id", istasyonid);
        con.Open();
        cmdata.ExecuteNonQuery();
        con.Close();
        MessageBox.Show("Atama tamamlandı");
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
}
}
public void istasyonSurePaylasim(int value1, int value2, int istasyonid) //
İSTASYONLARIN SÜRE PAYLAŞIMI
{
    double Sure1 = 0, Sure2=0;
    double paylasimSure = 0;
    double yeniSure1 = 0, yeniSure2=0;

    try // try-catch bloğunda paylaşımı yapılan ve paylaşımı üstlenecek istasyonlarını
    orijinal süreleri bulunuyor.
    {

```

```

        string querySure = "select StandartZaman from Istasyon where ID=@value1"; //
1. istasyon süre
        string querySure2 = "select StandartZaman from Istasyon where ID=@value2"; //
2. istasyon süre
        string queryPaylasim = "select StandartZaman from Istasyon where ID=@istasyonid";
// paylaşılacak istasyon süresi
        SqlCommand cmdSure = new SqlCommand(querySure,con);
        SqlCommand cmdSure2 = new SqlCommand(querySure2, con);
        SqlCommand cmdPaylasim = new SqlCommand(queryPaylasim,con);
        cmdSure.Parameters.AddWithValue("@value1",value1);
        cmdSure2.Parameters.AddWithValue("@value2", value2);
        cmdPaylasim.Parameters.AddWithValue("@istasyonid",istasyonid);
        con.Open();

        SqlDataReader dReader = cmdSure.ExecuteReader();
        while (dReader.Read())
        {
            Sure1 = dReader.GetDouble(0);
        }
        dReader.Close();

        SqlDataReader dReader2 = cmdSure2.ExecuteReader();
        while (dReader2.Read())
        {
            Sure2 = dReader2.GetDouble(0);
        }
        dReader2.Close();

        SqlDataReader dReader3 = cmdPaylasim.ExecuteReader();
        while (dReader3.Read())
        {
            paylasimSure = dReader3.GetDouble(0);
        }
        dReader3.Close();
        cmdSure.ExecuteNonQuery();
        cmdSure2.ExecuteNonQuery();
        cmdPaylasim.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
    yeniSure1 = Sure1 + (paylasimSure * Sure2) / (Sure1 + Sure2); // paylaşılacak
istasyonun süresi diğer 2 istasyona istasyonların iş yüklerine bağlı olarak
    yeniSure2 = Sure2 + (paylasimSure * Sure1) / (Sure1 + Sure2); // paylaşılacak yeni
süreler hesaplanıyor.
    try // try-catch bloğunda istasyon süreleri yeni sürelerle güncelleniyor.
    {

```

```

        string querySureAta = "update Istasyon set StandartZaman=@yeniSure1 where
ID=@value1"; // 1. istasyon güncelleme
        string querySureAta2 = "update Istasyon set StandartZaman=@yeniSure2 where
ID=@value2"; // 2. istasyon güncelleme
        SqlCommand cmdSureAta = new SqlCommand(querySureAta,con);
        SqlCommand cmdSureAta2 = new SqlCommand(querySureAta2, con);
        cmdSureAta.Parameters.AddWithValue("@yeniSure1",yeniSure1);
        cmdSureAta.Parameters.AddWithValue("@value1",value1);
        cmdSureAta2.Parameters.AddWithValue("@yeniSure2", yeniSure2);
        cmdSureAta2.Parameters.AddWithValue("@value2", value2);
        con.Open();
        cmdSureAta.ExecuteNonQuery();
        cmdSureAta2.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
}
public void istasyonKayit(int value1,int value2)    // DEĞİŞEN İSTASYONLARIN
ORJİNAL SÜRELERİ BULUNUYOR
{
    double Sure1 = 0, Sure2 = 0;
    try    // try-catch bloğunda istasyonların kaydı yapılacak istasyonların orjinal süreleri
    bulunuyor.
    {
        string querySure = "select StandartZaman from Istasyon where ID=@value1"; //1.
        istasyon süre
        string querySure2 = "select StandartZaman from Istasyon where ID=@value2"; //2.
        istasyon süre
        SqlCommand cmdSure = new SqlCommand(querySure, con);
        SqlCommand cmdSure2 = new SqlCommand(querySure2, con);
        cmdSure.Parameters.AddWithValue("@value1", value1);
        cmdSure2.Parameters.AddWithValue("@value2", value2);
        con.Open();
        //Sure1 = (Int32)cmdSure.ExecuteScalar();
        //Sure2 = (Int32)cmdSure2.ExecuteScalar();
        SqlDataReader dReader = cmdSure.ExecuteReader();
        while (dReader.Read())
        {
            Sure1 = dReader.GetDouble(0);
        }
        dReader.Close();

        SqlDataReader dReader2 = cmdSure2.ExecuteReader();
        while (dReader2.Read())
        {
            Sure2 = dReader2.GetDouble(0);
        }
    }
}

```



```

    }
    dReader2.Close();
    cmdSure.ExecuteNonQuery();
    cmdSure2.ExecuteNonQuery();
    con.Close();
}
catch (Exception ex) { MessageBox.Show(ex.Message); }
istKayitTamamla(value1,Sure1); // Bulunan süreler kayıt fonksiyonuna yollanıyor
istKayitTamamla(value2,Sure2); //
}

```

public void istKayitTamamla(int istasyonID,double zaman) { // SÜRE KAYIT FONKSİYONU

```

try // try-catch bloğunda istasyonun veritabanına kaydı yapılıyor.
{
    string querymax = "select count(ID) from IstasyonKayit";
    SqlCommand cmdmax = new SqlCommand(querymax, con);
    con.Open();
    int id = (int)cmdmax.ExecuteScalar();
    cmdmax.ExecuteNonQuery();
    con.Close();
    int currentid = 0;
    if (id == 0)
    {
        currentid = 1;
    }
    else
    {
        string queryLast = "select max(ID) from IstasyonKayit";
        SqlCommand findLast = new SqlCommand(queryLast, con);
        con.Open();
        int id2 = (int)findLast.ExecuteScalar();
        findLast.ExecuteNonQuery();
        currentid = id2 + 1;
        con.Close();
    }
    /*string queryLast2 = "select max(ID) from Bölümler";
    SqlCommand findLast2 = new SqlCommand(queryLast2, con);
    con.Open();
    int id = (int)findLast2.ExecuteScalar();
    int currentid = 0;
    findLast2.ExecuteNonQuery();
    con.Close();
    currentid = id + 1;*/
}

```

```
        string query = "insert into IstasyonKayit(ID,Zaman,istasyonId)
values(@id,@zaman,@istasyonId)";
        SqlCommand cmd = new SqlCommand(query, con);
        con.Open();

        cmd.Parameters.AddWithValue("@id", currentid);
        cmd.Parameters.AddWithValue("@zaman", zaman);
        cmd.Parameters.AddWithValue("istasyonId",istasyonID);
        cmd.ExecuteNonQuery();
        con.Close();
    }
    catch (Exception ex) { MessageBox.Show(ex.Message); }
}
// end code
}
}
```

