

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**ÖĞRENME VE BOZULMA ETKİSİ ALTINDA İŞ
REDDETMELİ ÇİZELGELEME PROBLEMLERİ**

**Hazırlayan
Berrin ATALAY**

**Danışman
Prof. Dr. M. Duran TOKSARI**

Doktora Tezi

**Aralık 2019
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**ÖĞRENME VE BOZULMA ETKİSİ ALTINDA İŞ
REDDETMELİ ÇİZELGELEME PROBLEMLERİ**

(Doktora Tezi)

**Hazırlayan
Berrin ATALAY**

**Danışman
Prof. Dr. M. Duran TOKSARI**

**Aralık 2019
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.



Berrin Atalay

“Öğrenme ve Bozulma Etkisi Altında İş Reddetmeli Çizelgeleme Problemleri” adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.



Hazırlayan

Berrin ATALAY



Danışman

Prof. Dr. M. Duran TOKSARI



Endüstri Mühendisliği ABD Başkanı

Prof. Dr. Lale ÖZBAKIR

Prof. Dr. M. Duran TOKSARI danışmanlığında **Berrin ATALAY** tarafından hazırlanan “**Öğrenme ve Bozulma Etkisi Altında İş Reddetmeli Çizelgeleme Problemleri**” adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Endüstri Mühendisliği** Anabilim Dalında **Doktora** tezi olarak kabul edilmiştir.

20.12.2019

JÜRİ:

Danışman : Prof. Dr. M. Duran TOKSARI

Üye : Prof. Dr. Emel KIZILKAYA AYDOĞAN

Üye : Doç. Dr. Celal ÖZTÜRK

Üye : Doç. Dr. Neslihan DEMİREL

Üye : Dr. Öğr. Üyesi Gazi Bilal YILDIZ

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 31/12/2019 tarih ve 2019/73-14...sayılı kararı ile onaylanmıştır.



Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

TEŐEKKÜR

Çalıőmalarım boyunca bana her türlü yardımda bulunan, katkılarıyla beni yönlendiren, danışman hocam sayın Prof. Dr. M. Duran TOKSARI'ya, Tez İzleme Komitesi üyeleri Prof. Dr. Emel KIZILKAYA AYDOĞAN ve Doç. Dr. Celal ÖZTÜRK'e ve manevi destekleriyle her zaman yanımda olan eşim Mehmet ATALAY ve oğlum Ahmet ATALAY başta olmak üzere tüm aileme teşekkürü bir borç bilirim.

Berrin ATALAY

Aralık 2019, KAYSERİ

ÖĞRENME VE BOZULMA ETKİSİ ALTINDA İŞ REDDETME Lİ ÇİZELGELEME PROBLEMLERİ

Berrin ATALAY

**Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Aralık 2019
Danışman: Prof. Dr. M. Duran TOKSARI**

ÖZET

Bu tez çalışmasında, literatürde ilk olarak öğrenme ve bozulma etkisi altında iş reddetmeli tek makina ve paralel makina çizelgeleme problemleri incelenmiştir. Öğrenme etkisi ve iş reddetme durumları ile öğrenme etkisi, bozulma etkisi ve iş reddetme durumları eş zamanlı olarak düşünülmüştür. Öğrenme etkisi işlerin işlem sürelerini azaltan etkiye sahipken, bozulma etkisi işlerin işlem sürelerini artıran bir etkiye sahiptir. Bunun yanında bazı işlerin reddedilmesi, yani işlenmemesi durumunda ise reddetme maliyeti ortaya çıkmaktadır. Bu çalışmada, bu durumlar altında tek makina ve paralel makina çizelgeleme problemleri düşünülmüştür. Amaç; işlenen işlerin işlem süreleri ve reddedilen işlerin reddetme maliyetleri toplamının en küçüklenmesidir. Mevcut problemlerin matematiksel modeli geliştirilmiş ve bir çözüm algoritması önerilmiştir. Matematiksel model sonuçları ile önerilen algoritmadan elde edilen sonuçlar karşılaştırılmıştır.

Anahtar Kelimeler: Çizelgeleme, öğrenme etkisi, bozulma etkisi, iş reddetme, tek makina, paralel makina

SCHEDULING PROBLEMS WITH JOB REJECTION UNDER THE EFFECT OF LEARNING AND DETERIORATION

Berrin ATALAY

Erciyes University, Graduate School of Natural and Applied Sciences

PhD Thesis, December 2019

Supervisor: Prof. M. Duran TOKSARI

ABSTRACT

In this thesis, single machine and parallel machine scheduling problems with job rejection under the effect of learning and deterioration are examined. Learning effect and job rejection situations and learning effect, deterioration effect, and job rejection situations were considered simultaneously. While the learning effect has the effect of decreasing the processing time of the jobs, the deterioration effect has the effect of increasing the processing time of the jobs. In addition, when some jobs are rejected, in other words, when they are not processed, rejection cost arises. In this study, single machine and parallel machine scheduling problems are considered under above given situations. The objective function is the minimization of the total processing time of the jobs processed and the rejection costs of rejected jobs. The mathematical model of the current problem is developed and the algorithm is proposed. The results of the mathematical model and the proposed algorithm are compared.

Keywords: Scheduling, learning effect, deterioration effect, job rejection, single machine, parallel machine

İÇİNDEKİLER

ÖĞRENME VE BOZULMA ETKİSİ ALTINDA İŞ REDDETMELİ ÇİZELGELEME PROBLEMLERİ

BİLİMSEL ETİĞE UYGUNLUK.....	i
YÖNERGEYE UYGUNLUK.....	ii
KABUL VE ONAY.....	iii
TEŞEKKÜR.....	iv
ÖZET.....	v
ABSTRACT.....	vi
İÇİNDEKİLER.....	vii
SİMGELER VE KISALTMALAR.....	ix
TABLolar LİSTESİ.....	x
ŞEKİLLER LİSTESİ.....	xi
GİRİŞ.....	1

1. BÖLÜM

LİTERATÜR ÇALIŞMASI.....	4
--------------------------	---

2. BÖLÜM

ÖĞRENME ETKİSİ ALTINDA İŞ REDDETMELİ ÇİZELGELEME PROBLEMLERİ

2.1. Öğrenme Etkisi Altında İş Reddetmeli Maksimum Tamamlanma Zamanı Probleminin En Küçüklenmesi	23
2.2. Öğrenme Etkisi Altında İş Reddetmeli Toplam Tamamlanma Zamanı Probleminin En Küçüklenmesi	24

3. BÖLÜM

BOZULMA VE ÖĞRENME ETKİSİ ALTINDA İŞ REDDETME Lİ ÇİZELGELEME PROBLEMLERİ

3.1. Bozulma ve Öğrenme Etkisi Altında İş Reddetmeli Maksimum Tamamlanma Zamanı Probleminin En Küçüklenmesi	28
3.2. Bozulma ve Öğrenme Etkisi Altında İş Reddetmeli Toplam Tamamlanma Zamanı Probleminin En Küçüklenmesi	29
3.3. Bozulma ve Öğrenme Etkisi Altında İş Reddetmeli Tamamlanma Zamanlarının Mutlak Farkları Toplamı Probleminin En Küçüklenmesi	31

4. BÖLÜM

ÇÖZÜM ALGORİTMALARI

4.1. Öğrenme ve Bozulma Etkisi Altında İş Reddetmeli Maksimum Tamamlanma Zamanı En Küçüklenmesi İçin Çözüm Algoritması	40
4.2. Öğrenme ve Bozulma Etkisi Altında İş Reddetmeli Toplam Tamamlanma Zamanı En Küçüklenmesi İçin Çözüm Algoritması	43
4.3. Öğrenme ve Bozulma Etkisi Altında İş Reddetmeli TZMFT En Küçüklenmesi İçin Çözüm Algoritması	45

5. BÖLÜM

BULGULAR.....	48
---------------	----

6. BÖLÜM

SONUÇ.....	53
KAYNAKÇA.....	52
EKLER.....	60
ÖZGEÇMİŞ.....	81

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simge ve kısaltmalar, açıklamaları ile birlikte aşağıda verilmiştir.

Simgeler	Açıklama
r	işlem sırası, $r = 1, \dots, n$
n	iş sayısı
m	makina sayısı
α	öğrenme indeksi
δ	ceza maliyeti
P_j	j işinin işlem süresi
$p_{j[r]}$	r pozisyonuna atanan j işinin gerçek işlem süresi
A	$(r - 1)$ pozisyonuna atanan işin tamamlanma zamanı
X_{ijr}	i makinasında j işinin r pozisyonuna atanmasını veya atanmamasını ifade eden karar değişkeni
w_r	r pozisyonunun ağırlığı
γ	bozulma katsayısı
T_r	r pozisyonunda işe başlayan işin işe başlama zamanı

Kısaltmalar	Açıklama
AEKİS	Ağırlıklı en kısa işlem süresi
ATÜ	Atölye tipi üretim
BE	Bozulma etkisi
BZMF	Bekleme zamanlarının mutlak farkı
EETT	En erken teslim tarihi
E/G	Erken/geç tamamlanma
EKBO	En küçük bozulma oranı
EKİS	En kısa işlem süresi
EUİS	En uzun işlem süresi
MİB	Merkezi işlem birimi
ÖE	Öğrenme etkisi
RM	Reddetme maliyeti
TM	Toplam maliyet
TPZTT	Tamamen polinomial zamanlı tahmin taslağı
TZMFT	Tamamlanma zamanlarının mutlak farkının toplamı

TABLULAR LİSTESİ

Tablo 1.1. Öğrenme etkisi, bozulma etkisi ve iş reddetme durumları üzerine yapılan çalışmaların literatür araştırması.....	21
Tablo 5.1. Öğrenme etkisi altındaki 10 iş, 20 iş ve 30 iş problemlerinin matematiksel ve algoritma sonuçları.....	48
Tablo 5.2. Öğrenme ve bozulma etkisi altındaki 10 iş, 20 iş ve 30 iş problemlerinin matematiksel ve algoritma sonuçları.....	50



ŞEKİLLER LİSTESİ

Şekil 4.1.*i* ve *j* işlerinin *S* ve *S'* çizelgelerinde gösterimi 35



GİRİŞ

Çizelgeleme sınırlı kaynakların, belirli bir amaç ya da amaçlar doğrultusunda ve belirli bir zaman aralığında atanmasını sağlayan bir karar verme sürecidir. Çizelgelemenin amaçları ise üretim ve hizmet olanaklarının en etkin biçimde kullanılmasını sağlamak, müşteri taleplerine hızlı bir şekilde cevap vermek, işleri tam zamanında teslim etmek, fazla mesai maliyetleri ve fason maliyetleri en küçükmek şeklinde özetlenebilir. Her bir iş öncelik seviyesi, teslim süresi, hazırlık süresi ve işe başlama zamanına sahip olabilir. Çizelgelemenin amaçları da bu farklı istekler doğrultusunda yönlendirilebilir.

Klasik çizelgeleme problemlerinde işlerin işlem sürelerinin sabit olduğu varsayılır ancak gerçek hayat problemlerinde öğrenme etkisi ve bozulma etkisi altında işlerin işlem süreleri değişebilmektedir. Ayrıca işlerin işlem sürelerinin fazla olması, üretim ve sabit giderlerin yüksek olması, ekipman ve işgücü yetersizliği gibi sebeplerden işletmeler ürünlerin işletme içerisinde üretilmemesini isteyebilir. Ürünler dışarıda üretildiğinde bu durum, işletmelere fason maliyeti olarak yansımaktadır. İşletmeler, ürünü üretmek isteyip istemedikleri kararını maliyetlere göre değerlendirerek vermektedirler. Direkt üretim maliyetlerinin yanı sıra, reddedilen işler için ceza maliyetleri oluşmaktadır. Bu ceza maliyetleri gelir kaybına sebep olmaktadır. Bu durumda çizelgelemeyi yapan kişi öncelikli olarak hangi işlerin işleneceği, hangi işlerin reddedileceği kararını vermek durumunda kalabilmektedir.

Öğrenme etkisi, işin sürekli yapıldığında öğrenilmesi sebebiyle işlem süresinin kısalması olarak açıklanabilir. Çizelgeleme probleminde öğrenme etkisi, tek makina veya paralel makinelerin olduğu üretim ortamında müşteriye ürün temin edilmesi esnasında ortaya çıkabilir.

Çizelgeleme probleminde öğrenme etkisi altındaki işin işlem zamanı, o işin çizelgedeki pozisyonuna bağlı olarak azalan fonksiyon şeklinde modellenir. Öğrenme etkisi farklı

şekillerde ifade edilmektedir. Literatürde sıklıkla kullanılan öğrenme etkileri; pozisyon tabanlı, zaman tabanlı ve işlem zamanlarının toplamına bağlı öğrenme etkileri şeklinde sınıflandırılabilir.

İşlenecek olan parça, sırada beklerken ya da makinanın/işçinin işinin olmasından dolayı işlerin işlem süreleri uzayabilir, bu durum işlem süresinde bozulma olarak bilinir. Elektronik parça üreten işletmelerde parça üretime girmek için beklerken, ortamdaki radyoaktif etkilerden ötürü ya da mikro düzeydeki toz parçalarının yüzeyine yapışmasından dolayı işlenemeyecek hale gelebilir, bu durum bozulma etkisine örnek gösterilebilir.

İş reddetme bazı işlerin içeride üretilmeyip dışarıda üretilmesi ya da üretilmemesi şeklinde ifade edilebilir. Ürünün içeride üretilmeme sebepleri; üretim maliyetlerinin, ürün hazırlık maliyetlerinin, işçilik maliyetlerinin yüksek oluşu veya fason üretim maliyetlerinin düşük olması olabilir. İşin reddedildiği durumda reddetme maliyeti ortaya çıkmaktadır. Tedarikçi en düşük maliyetli olan yöntemle göre işi reddedebilir ya da üretimini gerçekleştirebilir.

Bu tez çalışmasında, öğrenme ve bozulma etkileri altında eş zamanlı olarak iş reddetme durumlarını da ele alan tek ve paralel makinalı çizelgeleme ortamı incelenmiştir. Pozisyon tabanlı öğrenme ve iş reddetme ayrıca, pozisyon tabanlı öğrenme ve bozulmanın yanında iş reddetme durumlarını da içeren sekiz farklı model dikkate alınarak bu modellerdeki değişimlerin sonuçları analiz edilmiştir.

Çalışmanın birinci bölümünde literatürde mevcut olan öğrenme etkisi, bozulma etkisi ve iş reddetme üzerine yapılan çalışmalar incelenecektir.

Çalışmanın ikinci bölümünde öğrenme etkisi altında iş reddetmeli tek makina ve paralel makina çizelgeleme problemleri incelenmiştir. Problemlerle ilgili teoremlerin ispatları ve dört farklı amaç fonksiyonuna göre matematiksel modeller sunulmuştur.

Üçüncü bölümde öğrenme ve bozulma etkisi altında iş reddetmeli tek makina ve paralel makina çizelgeleme problemleri incelenmiştir. Problemlerle ilgili teoremlerin ispatları ve dört farklı amaç fonksiyonuna göre matematiksel modeller sunulmuştur.

Dördüncü bölümde önerilen çözüm algoritmaları sunulmuştur. Üç farklı algoritma, ikinci ve üçüncü bölümde bahsedilen modellerde kullanılacaktır.

Beşinci bölümde çözülen matematiksel model sonuçları ve algoritma sonuçları verilerek karşılaştırmalar yapılmıştır. Bu karşılaştırmalar; iki farklı bozulma oranı (0.1 ve 0.2) ve üç farklı öğrenme oranına (0.6, 0.7 ve 0.8) göre 10 iş, 20 iş ve 30 iş için yapılmıştır.



1. BÖLÜM

LİTERATÜR ÇALIŞMASI

Bu bölümde, tez çalışmasında incelenecek olan problemle ilgili daha önceden yapılmış çalışmalara ait literatür incelenecektir. Ele alınan problemlerde, öğrenme etkisi, bozulma etkisi ve iş reddetme durumlarının olduğu varsayılmıştır.

Mosheiov [1] “öğrenme etkisi” terminolojisini geliştirmiştir. Ayrıca Mosheiov [2] paralel eş makinalar için öğrenme etkisi altında akışın en küçüklenmesi problemlerini çalışmıştır. Biskup [4] öğrenme etkisini çizelgeleme üzerinde kullanan ilk kişidir. Aynı makina özellikleri durumunda, işlerin sayısının fonksiyonuna bağlı olarak işlem zamanının azalacağını varsaymıştır. Gerçek işlem zamanı için aşağıdaki modeli düşünmüştür.

$$p_{j[r]} = (p_j)r^\alpha \quad (1)$$

Burada $p_{j[r]}$, r pozisyonuna atanan j işinin gerçek işlem zamanı, p_j ise temel işlem zamanıdır. α ($\alpha < 0$) öğrenme indeksidir.

Mosheiov'un [3] çalışmasında; öğrenme eğrisi yaklaşımı, öğrenmeyi, üretim sürecinin tekrarının bir fonksiyonu olarak üstlendiğini varsaymaktadır ve herhangi bir belirli öğrenme modeline dayanmamaktadır. Tek makina çizelgeleme problemi için, en büyük tamamlanma zamanı ve toplam akış zamanının en küçüklenmesini; bağımsız paralel makinalar için teslim tarihi atama problemi ve toplam akış zamanı en küçüklenmesi probleminin polinomial zamanda çözülebildiğini göstermiştir.

Koulamas ve Kyparisis [5] çalışmasının ilk amacı Biskup [4]'un kullandığı öğrenme fonksiyonunu genellemektir. Akış tipi çizelgeleme problemleri için üretimde hazırlık

zamanlarının sıklığından dolayı pozisyon tabanlı öğrenme fonksiyonu daha uygundur ve çalışmalarının amacı, çok operasyonlu çizelgeleme problemleri için öğrenme etkisi altında en büyük tamamlanma zamanı ($F2|ÖE|C_{max}$) ve toplam tamamlanma zamanlarını ($F2|ÖE|\sum C_j$) en küçüklemeektir.

Yin ve ark. [6] çalışmasında, bir işin gerçek işlem zamanının, işlerin toplam temel işlem sürelerinin bir fonksiyonu olarak değil, aynı zamanda işin çizelgelendiği pozisyonuna bağlı olduğu bir fonksiyon olarak düşünen genel bir model geliştirmişlerdir. Önerilen modelin tek makina ve m -makina çizelgeleme problemleri için hala polinomial zamanda çözülebildiğini göstermişlerdir.

Kuo ve Yang [7, 8] çalışmalarında zaman bağımlı öğrenme etkisi altında tek makina çizelgeleme problemlerinde çalışmışlardır, bunlardan ilkinde grup çizelgeleme problemlerinde en büyük tamamlanma zamanı ve toplam tamamlanma zamanını en küçüklemişler, diğer çalışmalarında ise; toplam tamamlanma zamanını en küçüklemeyi amaçlamışlar ve En Kısa İşlem Süresi (EKİS) kuralının optimum olduğunu göstermişlerdir. Kuo ve Yang [9]'ın çalışmasında, zaman bağımlı öğrenme etkisi ve tek makina çizelgeleme problemi için en büyük gecikmeyi, tamamlanma zamanının k 'inci kuvvetlerinin toplamını ve ağırlıklandırılmış tamamlanma zamanı toplamını en küçükleyecek bir model düşünülmüştür. Tamamlanma zamanının k 'inci kuvvetlerinin toplamı işlerin EKİS sırasına göre sıralanmasıyla en küçüklendiği gösterilmiştir. Kuo [10] çalışmasında, grup çizelgeleme problemlerinde işlem süreleri için, öğrenme tabanlı işlem zamanları toplamı ve hazırlık zamanları için pozisyon tabanlı öğrenmeyi kullanmıştır. En büyük tamamlanma zamanı ve toplam tamamlanma zamanını en küçükleyecek, ikişer polinomial zamanlı algoritma önermiştir.

Low ve Lin [11] çalışmalarında, işlerin gerçek işlem zamanlarının işlerin sürelerinin oranı ve çizelgelenmiş işlerin pozisyonlarına bağlı olduğunu varsaymışlardır. Toplam tamamlanma zamanı ve toplam ağırlıklandırılmış tamamlanma zamanı en küçükleme problemleri için çözümün polinomial zamanda yapılabileceği gösterilmiş ve optimum çizelgenin işleri EKİS kuralına göre sıralayarak elde edilebileceği sunulmuştur.

Cheng ve ark. [12] çalışmalarında, öğrenme etkisinin işlerin işlem zamanı üzerinde logaritmik bir fonksiyonu şeklinde etkisi olduğunu gösteren yeni bir öğrenme modeli

düşünmüşlerdir. Önerilen öğrenme modeli ile en büyük tamamlanma zamanı ve toplam tamamlanma zamanı en küçükleme problemlerinin polinomiyal zamanda çözülebileceği gösterilmiştir. Eren ve Güner [13] çalışmalarında öğrenme etkisi altındaki işlerin toplam gecikmesini en küçükleyecek 0 – 1 tamsayılı programlama modeli geliştirmişlerdir. Ayrıca problemin çözümü için rassal arama, tabu arama ve tavlama benzetimi metotlarını kullanmışlardır.

Wu ve Lee ilk çalışmalarında [14] işlerin gerçek işlem zamanlarının sadece çizelgelendikleri pozisyona bağlı olmadığını ayrıca önceden işlenmiş olan işlerin işlem süreleri toplamına da bağlı olduğunu gösteren yeni bir model geliştirmişlerdir. İkinci çalışmalarında [15] öğrenmenin sadece işin pozisyonuna bağlı olmadığını ayrıca grup pozisyonuna da bağlı olduğunu gösteren yeni bir grup çizelgeleme öğrenme modeli geliştirmişlerdir. Önerdikleri modelin en büyük tamamlanma zamanı ve toplam tamamlanma zamanının en küçüklemesi amaçları için polinomiyal zamanda çözülebildiğini göstermişlerdir. Ji ve Cheng [16] çalışmalarında işe bağlı öğrenme etkisi ve çoklu hız değiştirme faaliyetleri ile bir programlama problemini düşünmüşlerdir. Bir işin gerçek işlem süresi, sadece bir sıradaki pozisyonuna değil, aynı zamanda bir hız değiştirici faaliyetin öncesinde mi yoksa sonrasında mı planlandığına da bağlı bir değişken olduğunu söylemişlerdir. Amaçları toplam tamamlanma zamanını en küçükleme. Cheng ve ark. [17] çalışmalarında, kontrol parametresi öğrenme fonksiyonu olan ve çizelgede, işin gerçek işlem zamanı ile çizelgelenen iş pozisyonunun fonksiyonu olan iki makinalı akış çizelgeleme problemi düşünmüşlerdir. Problemi çözmek için dal sınır algoritması ve üç tavlama algoritması geliştirmişlerdir, amaçları ise toplam tamamlama zamanını en küçükleme.

Liu [18] çalışmasında, elektronik üretim endüstrisinde, bir elektronik parçanın elektromanyetik alana maruz kalan parçalardaki kötü etkileri azaltmak için ekstra zamana ihtiyaç duymaktadır ve bu ekstra zamanı geçmiş zamanlı bağımsız dağıtım zamanı olarak modellemiştir. Çalışmada üç amaç fonksiyonu ele alınmıştır; işin toplam sapmalarının zamanını en küçükleme, makinaların toplam yükünü en küçükleme ve toplam tamamlanma zamanını en küçükleme. Problemin çözümü için polinomiyal zamanda çözülebilen bir algoritma geliştirilmiştir. Lee ve Wu [19] çalışmalarında yapay zeka ile çalışan robotların kendi kendine öğrenme süreci olduğunu, ayrıca operatörlerin de iş deneyimlerine göre daha iyi komutlar verdiğini gözlemlemişler ve bu iki ana

yaklaşımı birleştirerek yeni bir öğrenme modeli oluşturmuşlardır. Önerilen bu yaklaşımlar tek makina ve çok makina akış problemleri için uygulanmış, problemlerin polinomiyal zamanda çözülebileceği gösterilmiştir.

Eren [20] çalışmasında, hazırlık zamanlarının öğrenme etkisi ve çıkarma zamanları iki kriterli m adet eş paralel makina için düşünmüştür. Çalışmanın amacı ağırlıklı toplam tamamlanma zamanı toplamı ve toplam gecikmeyi en küçükleme olarak belirlenmiştir. Eren [21] çalışmasında, eşit olmayan dağıtım zamanları ve öğrenme etkisi altında tek makina çizelgeleme problemini düşünmüştür. Bu çalışmada amaç olarak ise ağırlıklı tamamlanma zamanı toplamını en küçükleme sunulmuştur. Wu ve Lee [22] çalışmalarında, hem makina hem de işçilerin öğrenmesini aynı anda düşünerek bir model önermişler, tek makina çizelgeleme ve akış tipi çizelgeleme problemleri için çözüm prosedürleri sunmuşlardır.

Yin ve Wang [23] çalışmalarında, kontrol edilebilir işlem süreleri ve öğrenme etkisi altında tek makinalı çizelgeleme problemlerini ele almışlardır. İki farklı amaç incelenmiştir. İlk olarak en büyük tamamlanma zamanı, toplam tamamlanma zamanı, tamamlanma zamanları arasında toplam mutlak fark ve toplam sıkıştırma süresinin en küçükleme maliyet fonksiyonu olarak düşünülmüştür. İkincide ise en büyük tamamlanma zamanı, toplam bekleme zamanı, bekleme zamanları arasında toplam mutlak fark ve toplam sıkıştırma süresinin en küçükleme maliyet fonksiyonu olarak düşünülmüştür. Problem atama problemi olarak modellenmiş ve iyi bilinen algoritmalarla çözülmüştür. Cheng ve Wang [24], öğrenme etkisi altında en büyük gecikmeyi en küçükleyecek tek makina çizelgeleme problemi üzerine çalışmışlardır. Bu problemin NP-zor olduğu gösterilmiş, iki özel durum tanımlanmış ve iki sezgisel ile polinomiyal zamanda çözülebileceği gösterilmiştir.

Biskup [25] öğrenme etkisi ile yapılan çalışmaların özetlerini sunmuş, farklı öğrenme etkilerine göre çalışmaları sınıflandırmıştır. Azzouz ve ark. [26] tarafından önerilen çalışmada öğrenme etkileri literatürü incelenmiş, yapılmış olan çalışmalar sınıflandırılmıştır.

Li ve ark. [27], kontrolsüz bir öğrenme etkisi altındaki bir işin gerçek işlem süresinin, işin pozisyonu ve öğrenme fonksiyonun kontrol parametresinin bir fonksiyonu olduğu iki makinalı akış çizelgeleme problemini ele almışlardır. Bu çalışmada toplam

tamamlanma zamanını en küçükleme amaçlanmıştır ve bunun için dal sınır ve üç tavlama benzetimi algoritması önerilmiştir. Toksarı ve Arık [28] çalışmalarında, bulanık işlem süreleri ile pozisyon tabanlı bulanık öğrenme etkisi altında tek makina çizelgeleme problemi incelemiştir. Bu çalışmada en büyük tamamlanma zamanı, toplam tamamlanma zamanı ve toplam ağırlıklandırılmış tamamlanma zamanını en küçükleme amaçlanmıştır. Bu üç amaç için bulanık karma tamsayı doğrusal olmayan programlama modeli önerilmiştir.

İşin reddedilerek çizelgelendiği, çizelgeleme problemlerinde tedarikçinin faydasına olmayacak ürünler işlenmeyebilir. Direkt üretim maliyetlerinin yanı sıra, reddedilen işler için ceza maliyetleri oluşmaktadır. Bu ceza maliyetleri gelir kaybına sebep olur. Bu durumda çizelgelemeyi yapan kişi hangi işlerin işleneceği hangi işlerin reddedileceği kararını öncelikli olarak vermek durumunda kalır. Makina çizelgelemede işlerin reddedilmesi durumu ilk olarak Bartal [29] tarafından düşünülmüştür. Zhang ve arkadaşlarının [30] çalışmasında, iş reddetme kısıtı altında tek makina çizelgeleme problemi incelenmiştir. İş ya reddedilir ve reddetme maliyeti ödenir ya da kabul edilir ve tek makinada işlenir. Ancak, reddedilen işlerin toplam reddetme maliyeti verilen üst sınırı aşamaz. Amaç, verilen f 'i en küçükleyecek çizelgenin bulunmasıdır. Burada f ; kabul edilen işlerin tamamlanma zamanının azalmayan bir fonksiyonudur. Bağımsız amaç fonksiyonları için problemlerin hesaplamasal karmaşası analiz edilmiştir ve yalancı polinomial zaman algoritması sunulmuştur. Ayrıca buna ek olarak, bırakma zamanları ile en büyük tamamlanma zamanı problemi için Tamamen Polinomial Zamanlı Tahmin Taslağı (TPZTT) sunulmuştur. Engels [31] genel çizelgeleme problemlerinde, reddedilen her bir iş için ceza maliyetine sebep olduğu durumu düşünmüştür. Reddedilen işlerin üzerine odaklanmıştır ve reddedilen işlerin ceza maliyetleri ile çizelgelenmiş işlerin ağırlıklı tamamlanma zamanlarını optimize etmeye çalışmıştır.

Li ve Yuan [32] tarafından sunulan çalışmada paralel eş makineler olduğunda iş reddedilmesinin genel yapısı verilmiştir. Çalışmanın amacı kabul edilen işlerin çizelgeleme maliyetine ek olarak reddedilen işlerin toplam ceza maliyetlerinin en küçülenmesidir. Üç farklı çizelgeleme maliyeti düşünülmüştür; en büyük tamamlanma zamanı, toplam ağırlıklandırılmış tamamlama zamanı ve toplam tamamlanma zamanıdır. Gerstl ve Mosheiov [33] tarafından sunulan çalışmada paralel makina

çizelgeleme problemi için pozisyon tabanlı işlem süreleri ve işlerin reddedilmesi eş zamanlı kullanılmıştır. İki amaç fonksiyonu sunulmuştur; ilk amaç fonksiyonu, akış zamanı toplamını ve reddedilen işlerin maliyetini en küçüklemek, ikinci amaç fonksiyonu ise toplam yüklemeyi ve reddedilen işlerin maliyetini en küçüklemektir.

Epstein ve ark. [34] tarafından önerilen çalışmada işlerin işlem sürelerini $p_j=1$ olarak alınmış, tek makina n adet iş düşünülmüştür. Çalışmanın amacı reddedilen işlerin ceza maliyetleri ile kabul edilen işlerin işlem sürelerinin toplamının en küçüklenmesi olarak belirlenmiştir. Çalışmada, işlerin tek tek geldiği ve sistemin geleceği ile ilgili hiçbir bilgiye sahip olunmadığı durum incelenmiştir. Koullamas ve Panwalkar [35]'in çalışmasında, tek makina çizelgeleme problemleri için iş reddetme ile erken tamamlanma/gecikme problemi için optimum bir algoritma sunulmuştur. Algoritmanın amacı, çizelgelenmiş işlerin en büyük erken bitirmesini ve reddedilen işlerin toplam ceza maliyetlerinin en küçüklenmesidir.

Cheng ve Sun [36]'in çalışmalarında, işin işlem zamanının doğrusal bir fonksiyonu olan işe başlama zamanı ve işlerin ceza maliyeti ödenerek reddedilmeleri sağlanan tek makina çizelgeleme problemi düşünülmüştür. Çalışmada amaç fonksiyonları olarak en büyük tamamlanma zamanı, toplam ağırlıklandırılmış tamamlama zamanı ve en büyük gecikme ve reddedilen işlerin toplam ceza maliyetlerinin en küçüklenmesi belirlenmiştir. Problemlerin NP-zor yapıda olduğu gösterilmiş ve dinamik programlama temeline dayanan tasarım algoritmalarıyla çözülmüştür. Shabtay ve ark. [37] tarafından yapılan çalışmada reddetme durumunda zaman tetikleyici işlerin olduğu çizelgeleme problemlerinin genel yapısı tanımlanmış ve çözümün kalitesi F_1 ve F_2 ölçülmüştür. Burada F_1 : İşlerin tamamlanma zamanıyla ilgili çizelgeleme kriteri, $F_2 = RM = \sum_{j \in \bar{A}} e_j$ toplam reddetme maliyetidir.

Öncelikleri olan işlerin çizelgelenmesi ve reddetme problemi Hoogeveen ve ark. [38] tarafından çalışılmıştır. Makinaları eş makinalar P , ilişkili makinalar Q ve ilişkisiz makinaları R ile göstererek, öncelikli çizelgeleme problemleri çalışılmıştır. Çalışmanın amacı olarak çizelgelenmiş işlerin en büyük tamamlanma zamanlarının ve reddedilen işlerin ceza maliyetlerinin en küçüklenmesi belirlenmiştir.

Shabtay ve ark. [39] tarafından yapılan çalışmada çözümün kalitesi 2 kriterle ölçülmüştür; bunlar kabul edilen işlerin tamamlanma zamanına bağlı olan $F1$ ve toplam reddetme maliyeti $F2$ 'dir. Çalışmada, pozisyon tabanlı cezalandırma maliyetlerini içeren çizelgeleme kriteri ile tek makina çizelgeleme problemleri için birleşik iki kriter analizi önerilmiştir. Çalışmada amaç fonksiyonu olarak en büyük tamamlanma zamanının, tamamlanma zamanı toplamının, tamamlama zamanları sapmaları toplamının, toplam E/G maliyetlerinin en küçüklenmesi ile ilgilenilmiştir. Dört farklı problem incelenmiştir. $F1 + F2$ 'nin en küçüklenmesi polinomiyal zamanda çözümlenirken, diğer üç varyasyonun çözümünün NP-zor yapıda olduğu sunulmuş ve bu problemler için yalancı polinomiyal süre algoritması geliştirilmiştir. Ayrıca uygun etkili bir çizelge elde etmek için TPZTT önerilmiştir.

Merten ve Muller'in [40] çalışmasında, tek makinalı çizelgeleme problemi için performans ölçütü olarak bekleme zamanı varyansı ve akış zamanı varyansı analiz edilmiştir. Aynı iş sıralamasıyla hem ortalama akış zamanı varyansı hem de bekleme zamanı varyansı en küçüklenmiştir. Akış zamanını en küçükleyecek sıralamanın EKİS olduğu ve önem ağırlıkları eşit olduğunda işlerin işlem süresi ve önem ağırlıkları oranı küçükten büyüğe sıralanması ile elde edilen çizelgenin yine aynı olduğu gösterilmiştir.

Toksarı ve Güner [41] tarafından önerilen çalışmada öğrenme etkisi, bozulma etkisi, sıra bağımlı hazırlık zamanları ve ortak teslim tarihi yaklaşımını eş zamanlı olarak düşünerek, paralel makina E/G çizelgeleme problemi için karma tamsayı doğrusal olmayan programlama formülasyonu sunmuşlardır. Geliştirilen model, sıra bağımlı hazırlık zamanları ve sıra bağımlı E/G cezalarına izin vermektedir. On bir iş, iki makina problemi için, optimum çözüm kolaylıkla elde edilmiştir. Toksarı ve Güner [42] çalışmalarında, paralel makina E/G çizelgeleme problemi için pozisyon tabanlı öğrenme ve doğrusal ve doğrusal olmayan bozulma etkisi altında, farklı ceza maliyetlerini düşünmüşlerdir. Öğrenme ve bozulma etkisi altında erken E/G çizelgeleme problemi için V-şekilli çizelgenin optimum çözümü verdiğini göstermişlerdir.

Toksarı ve ark. [43] çalışmalarında doğrusal olmayan bozulma ve zaman tabanlı öğrenme etkisini aynı anda düşünmüşlerdir. En büyük tamamlanma zamanı, tamamlanma zamanları toplamı ve tamamlanma zamanları kareleri toplamının en küçüklenmesi problemlerinin işlerin EKİS kuralına göre sıralandığında, en büyük

gecikmenin en küçüklenmesi probleminin ise işlerin En Erken Teslim Tarihi (EETT) kuralına göre sıralandığında problemlerin hala polinomiyal zamanda çözülebildiğini ispatlamışlardır. Toksarı ve Güner [44] bu çalışmalarında doğrusal olmayan bozulma etkisi ve doğrusal olmayan öğrenme etkisini aynı anda düşünmüşlerdir. Tek makina çizelgeleme problemi için en büyük tamamlanma zamanı, tamamlanma zamanı kareleri toplamı ve en büyük gecikmenin en küçüklenmesi için polinomiyal zamanda çözülebilen yaklaşımlar sunmuşlardır. m makina akış tipi problemi için en büyük tamamlanma zamanı ve toplam tamamlanma zamanını en küçükleyecek polinomiyal zamanlı çözümleri sunmuşlardır. Toksarı ve Güner [45, 46] zaman bağımlı öğrenme etkisi, doğrusal ve doğrusal olmayan bozulma etkisi, sıra bağımlı hazırlık zamanları ve bütün işler için ortak teslim tarihinin aynı anda düşünüldüğü paralel E/G çizelgeleme problemini analiz etmişlerdir. Problem için karma tamsayı programlama formülasyonu geliştirmişler ve optimum çizelgenin V-şekilli olduğunu göstermişlerdir.

Toksarı [47] tek makinalı çizelgeleme problemlerinde en büyük tamamlanma zamanını en küçüklemek için, eşit olmayan teslim tarihleri ile öğrenme etkisi ve bozulma etkisini aynı anda düşünmüştür. Dal sınır algoritması önerilmiş ve önerilen sezgisel algoritma optimuma yakın sonuç elde etmiştir.

Kanet [48] çalışmasında, tek makina n adet iş düşünülmüştür. Çalışmada amaç olarak akış zamanlarındaki değişkenliği en küçük yapacak şekilde çizelgeleme elde etmek sunulmuştur. Akış zamanı varyansını en küçükleyecek Tamamlanma Zamanları Mutlak Farkı Toplamı (TZMFT) önerilmiştir. Burada işlerin tamamlanma zamanları 2'li karşılaştırmalarla mutlak farkları alınarak hesaplanır. $TZMFT = \sum_{i=1}^n \sum_{j=1}^n |C_j - C_i|$. Çalışmada önerilen optimum TZMFT çizelgesi üretmek için basit bir metot aşağıdaki gibidir;

- U çizelgenmemiş işlerin kümesi,
- S son çizelge,
- B ve A boş kümeler,
- n çizelgelenecek işlerin sayısı iken

Procedure GEN

For i = 1 to n do

En büyük i işini U kümesinden sil ve i adını ver,

Eğer i tekse, B kümesinin son pozisyonuna koy, aksi taktirde A kümesinin ilk pozisyonuna koy

End

S = {B, A}

End GEN

En uzun süreli iş, en küçük ağırlıklı pozisyona atandığı için çizelge TZMFT için optimum sonuç verir. Ayrıca çalışmada TZMFT'yi en küçük yapabilecek sezgisel bir yöntem de verilmiştir.

Oron [49] çalışmasında, basit doğrusal bozulma durumunda tek makina çizelgeleme problemlerini ele almıştır. İşlerin yaklaşık olarak tamamlanma zamanları düşünülmüş ve teslim tarihine göre cezalandırılmamıştır. Tek makina durumunda bu problemin atama problemi gibi çözülebileceği gösterilmiştir. Çalışmada amaç fonksiyonu olarak TZMFT'nin en küçüklenmesi belirtilmiştir.

Mor ve Mosheiov [50] tarafından sunulan çalışmada, paralel eş makinalardaki pozisyon tabanlı işlem süreleri ve ayrı makinalardaki pozisyon tabanlı işlem süreleri ile TZMFT çalışılmıştır. Bu çalışmada Bekleme Zamanlarının Mutlak Farkı (BZMF) için de çalışılmış ve bu problemlerin polinomiyal zamanda çözümleri verilmiştir. Çalışmada m adet paralel makinada işlenecek n adet iş kümesi düşünülmüştür. Makinalara atanan iş sayıları n_i 'dir ($\sum_{i=1}^m n_i = n$). Bunların önceden bilindiği varsayılarak, model hem eş makinalar için hem de ayrı makinalar için çözülmüştür. Bu iki problem de atama problemi olarak çözülebilmıştır. Daha sonra eş makina ve ayrı makina durumlarında toplam tamamlanma zamanı ve TZMFT'nin ikisi birlikte en küçüklenmesi ve eş makina ve ayrı makina durumlarında toplam bekleme zamanı ve BZMF'nin ikisi birlikte en küçüklenmesi problemleri atama problemi olarak çözülmüştür.

Li ve ark. [51] çalışmalarında, bozulan işlerle tek makina çizelgeleme problemi araştırılmıştır. Çalışmada işin bozulmasını zamanın oransal bir fonksiyonu olarak tanımlanmıştır. Amaç olarak TZMFT'nin en küçüklenmesi belirlenmiş ve optimum

çizelgenin V-şekilli olduğu gösterilmiştir. Ayrıca, optimuma yakın çözümler elde etmek için etkili ve verimli olan 2 sezgisel algoritma sunulmuştur.

Wang ve Xia [52] tarafından sunulan çalışmada n adet iş için tek makina çizelgeleme problemini araştırmışlardır. Çalışmada amaç olarak toplam tamamlanma zamanını içeren maliyet fonksiyonunu ve tamamlanma zamanları arasındaki mutlak farkın toplamının en küçüklenmesi belirlenmiştir. Bu problem atama problemi olarak çözülmüş ve özel bir durum için optimum çözümü veren algoritma sunulmuştur.

Huang ve Wang [53] çalışmalarında, bozulma etkisi altındaki işlerin paralel eş makinalarda çizelgelenmesi problemi incelenmiştir. Modelde işlerin işlem zamanları, başlama zamanlarının bir fonksiyonu olarak tanımlanmıştır. İki ayrı amaç düşünülmüştür; bunlar TZMFT'nin en küçüklenmesi ve BZMF'nin en küçüklenmesidir. Bütün işler için bir bozulma oranı verilmiş ve işlem süresi bu bozulmaya bağlı olarak artmıştır. Önerilen algoritma ile problemin polinomial zamanda çözülebileceği gösterilmiştir.

İşlerin işlem süreleri, bakım faaliyetleri, makinaların etkinliğini kaybetmesinden yada başka sebeplerden dolayı uzayabilir, bu durum işlem süresinde bozulma olarak bilinir.

Mosheiov'un [54] çalışmasında çeşitli performans ölçütleri altında tek makina problemlerinde doğrusal bozulmaya uğrayan işlerin optimum sıralamasını bulmak amaçlanmıştır. Kullanılan performans ölçütleri; en büyük tamamlanma zamanı, toplam akış zamanı, ağırlıklandırılmış tamamlanma zamanı toplamı, toplam gecikme, en büyük gecikme ve geciken işlerin sayısıdır. Geciken işlerin sayısını işlem süreleri sabitken Moore algoritması en küçüklerken, bozulma etkisi altındayken Moore algoritmasına benzer bir algoritma önerilmiştir. Browne ve Yechiali [55] tarafından önerilen çalışmada farklı bozulma türlerinin etkileri analiz edilmiş ve beklenen en büyük tamamlanma zamanı ve en büyük tamamlanma zamanının varyansını en küçükleyecek optimum çizelgeleme politikaları sunulmuştur.

Lee [56] tarafından sunulan çalışmada öğrenme ve bozulma etkisi ilk defa beraber düşünülmüştür. n adet iş, tek makinalı çizelgeleme ortamında ele alınmıştır. Bütün işler $t_0 > 0$ anında uygundur. Makina aynı anda tek iş işleyebilir ve önceliğe izin verilmez.

Her bir J_i işinin bozulma oranı α_i 'dir ve işler En Küçük Bozulma Oranı (EKBO) kuralına göre sıralanmışlardır ($\alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_n$).

$p_{i,r}$: J_i işinin r pozisyonuna atıldığındaki işlem süresi iken

$$p_{i,r} = \alpha_i t r^a \quad \text{İlk Model}$$

$t \geq t_0$; J_i işinin başlama zamanı, $a \leq 0$: öğrenme etkisi iken

$$p_{i,r} = (p_0 + \alpha_i t) r^a \quad \text{İkinci Model}$$

Burada p_0 : temel işlem zamanıdır.

Çok iyi bilinen klasik tek makina çizelgeleme problemleri ilk modeldeki yaklaşımla değerlendirilmiştir.

- En büyük tamamlanma zamanı en küçüklenmesi için EKBO sıralamasıyla işlerin sıralanması gerektiği, işlerin ikili yer değiştirmeleriyle ispat edilmiştir.
- Toplam akış zamanı en küçüklenmesi, toplam gecikme en küçüklenmesi için EKBO sıralaması kullanılmıştır.
- En büyük gecikmenin en küçüklenmesi için EETT sıralamasının optimum olduğu gösterilmiştir.

Bu sonuçların ikinci modelde doğru olmadığı, yani sıralamaların artık optimumu vermediği sunulmuştur.

Wang [57] tarafından sunulan çalışmada bozulma etkisi ve öğrenme etkisi aynı anda ele alınmıştır. Problemin varsayımlarına göre, tek makina çizelgeleme ortamında n adet bağımsız ve önceliksiz iş vardır $J = \{J_1, J_2, \dots, J_n\}$. Makinada tek iş işlenmekte ve önceliğe izin verilmemektedir. Her bir j işinin temel işlem zamanı: p_j ve bozulma fonksiyonu $\alpha(t)$ ($\alpha(t) \geq 0$)dir. Ayrıca her işin ağırlığı w_j ve işlerin teslim tarihi d_j 'dir. Çalışmada, aşağıda maddelenmiş ispatlar yapılmıştır.

- $1|p_j(\alpha(t) + \beta \cdot r^a)|C_{max}$ problemi için optimum çizelge işlerin temel işlem zamanlarının EKİS kuralına göre sıralanmasıyla elde edilir.

- $1|p_j(\alpha(t) + \beta \cdot r^a)|\sum C_j$ problemi için optimum çizelge işlerin temel işlem zamanlarının EKİS kuralına göre sıralanmasıyla elde edilir.
- $1|p_j(\alpha(t) + \beta \cdot r^a)|\sum C_j^2$ problemi için optimum çizelge işlerin temel işlem zamanlarının SPT kuralına göre sıralanmasıyla elde edilir.
- $1|p_j(\alpha(t) + \beta \cdot r^a)|\sum w_j C_j$ problemi için $w_j \geq w_k$ ve $p_j \leq p_k$ olduğunda, optimum çizelge işlerin AEKİS kuralına göre p_j/w_j oranının azalmayan sırasına göre sıralandığında elde edilir.
- $1|p_j(\alpha(t) + \beta \cdot r^a), p_j = p|\sum w_j C_j$ problemi için optimum çizelge işlerin w_j 'lerinin azalmayan sırasına göre sıralandığında elde edilir.
- $1|p_j(\alpha(t) + \beta \cdot r^a)|L_{max}$ problemi için $p_i \leq p_j$ ve $d_i \leq d_j$ olduğunda, optimum çizelge işlerin d_j değerlerine göre azalmayan sırada sıralanmasıyla elde edilir.
- $1|p_j(\alpha(t) + \beta \cdot r^a), p_j = p|L_{max}$ problemi için optimum çizelge işlerin d_j değerlerine göre azalmayan sırada sıralanmasıyla elde edilir.

Mosheiov [58] tarafından sunulan çalışmada, doğrusal bozulma altındaki işlerin tek makinada çizelgelenmesi problemi incelenmiştir. Çalışmanın amacı olarak akış zamanını en küçükleme belirtilmiştir ve optimum çizelgenin V-şekilli olduğu gösterilmiştir. Wang ve Xia [59] tarafından önerilen çalışmada işlerin başlama zamanına göre azalan doğrusal fonksiyonlu işlem zamanını alan özel bir çizelgeleme problemi düşünülmüştür. Tek makina, n adet bağımsız önceliksiz iş ele alınmıştır. Tek makina ve akış tipi çizelgeleme problemleri için, en büyük tamamlanma zamanı, en büyük gecikme, en büyük maliyet fonksiyonu ve geciken işlerin sayıları gibi farklı amaçlar tartışılmıştır. Bütün problemler için çözümlerin klasik çizelgelemedeki gibi polinomiyal zamanda çözüldüğü gösterilmiştir.

Huang ve ark. [60] tarafından önerilen çalışmada zaman bağımlı bozulma ve üstel öğrenme etkisi altında, tek makina, n adet bağımsız ve önceliksiz iş düşünülmüştür. Makinada tek zamanda tek iş işlenebilmekte ve önceliğe izin verilmemektedir. En büyük tamamlanma zamanı, toplam tamamlanma zamanı ve tamamlanma zamanı kareleri toplamı problemlerinin polinomiyal zamanda çözülebileceği gösterilmiştir.

Toplam ağırlıklı tamamlanma zamanı ve en büyük gecikme problemlerinin belli özel koşullar eklendiğinde polinomiyal zamanda çözülebileceği gösterilmiştir.

Wu ve ark. [61] tarafından önerilen çalışmada işlerin başlama zamanlarının doğrusal fonksiyonu olan iki tip tek makina çizelgeleme problemi ele alınmıştır. Birincisinde işlem zamanları, ikincisinde hazırlık zamanları grup olarak incelenmiştir. En büyük tamamlanma zamanı ve toplam tamamlanma zamanının en küçüklenmesi için problemler çözülmüştür. Bu çalışma bozulma etkisi altındaki işlerin çizelgelenmesi problemlerinde grup teknolojisinin kullanıldığı ilk çalışmadır.

Yang ve ark. [62] tarafından sunulan çalışmada çoklu ortak teslim tarihi ataması ve genel pozisyon temelli bozulma etkisi aynı anda ele alınmıştır. İki farklı model üzerinde çalışılmıştır. Bunlardan ilki doğrusal kaynak tüketim fonksiyonunu içerirken ikincisi ise konveks kaynak tüketim fonksiyonunu içerir. Burada amaç olarak erken bitirmeyi, geç bitirmeyi, teslim tarihini ve kaynak tüketim maliyetlerini içeren toplam ceza fonksiyonunun en küçüklenmesi belirlenmiştir. Çalışmada 2 farklı algoritma sunulmuştur. İkisinin de polinomiyal zamanda çözülebilen etkin algoritmalar olduğu gösterilmiştir.

Wang ve Wang [63] tarafından sunulan çalışmada doğrusal olmayan bozulma etkisi altındaki işlerin tek makinada çizelgelenmesi problemi incelenmiştir. İşlerin işlem zamanlarının işe başlama zamanının doğrusal olmayan bir fonksiyonu ile ifade edilmesi, doğrusal olmayan bozulma etkisidir. Çalışmada en büyük tamamlanma zamanının en küçüklenmesi ve toplam tamamlanma zamanının en küçüklenmesi problemleri ele alınmıştır. $1|p_j^A(t) = p_j(a + bt)^c, 0 < c < 1| \sum C_j$ problemi için iki aşamadan oluşan sezgisel bir algoritma önerilmiştir. Burada $p_j: J_j$ işinin normal işlem zamanı, $a \geq 0, b \geq 0, c \geq 0$: bozulma indeksidir. Algoritma ortalama merkezi işlem birimi (MİB) zamanı, ortalama ve en büyük hata yüzdeleri kaydedilerek değerlendirilmiştir. $c \geq 1$ olduğunda toplam tamamlanma zamanının en küçüklenmesi probleminin optimum olarak çözüldüğü gösterilmiştir.

Kononov ve Gawiejnowicz [64] tarafından önerilen çalışmada işlerin işlem süreleri, işe başlama zamanlarının artan bir fonksiyonu şeklinde tanımlanmıştır. Çalışmada amaç olarak en büyük tamamlanma zamanını en küçükleme belirlenmiştir ve akış tipi

çizelgeleme modeli ve atölye tipi çizelgeleme modelleri incelenmiştir. Bozulma üç farklı formda düşünülmüştür;

- Basit Doğrusal $p_{ij} = b_{ij}S_{ij}$
- Doğrusal $p_{ij} = a_{ij} + b_{ij}S_{ij}$
- Oransal $p_{ij} = b_{ij}(a + bS_{ij})$ $a_{ij} \geq 0, b_{ij} \geq 0, a \geq 0, b \geq 0$

Burada a_{ij} : temel işlem zamanı, b_{ij} : bozulma oranıdır.

İki ve daha fazla makinalı akış tipi ve atölye tipi çizelgeleme problemleri için üç farklı bozulma oranı etkisi altında en büyük tamamlanma zamanının en küçüklenmesi ele alınmıştır. Çalışmada doğrusal bozulma etkisi altında iki makina akış tipi ve iki makina atölye tipi çizelgeleme problemlerinin NP-zor yapıda olduğu gösterilmiştir.

Wang ve Wang [65] tarafından önerilen çalışmada teslim tarihlerine eşit boşluk metodu ile karar verilen, bozulan işlerin çizelgelenmesi problemi incelenmiştir. Modelde işlerin işlem zamanı işe başlama zamanlarının basit doğrusal fonksiyonu olarak tanımlanmıştır. Çalışmanın amacı olarak hiçbir iş gecikmeden, toplam ağırlıklandırılmış erken bitirme cezasını en küçükleme ele alınmıştır. Polinomiyal zamanda çözülebilen iki özel durum incelenmiştir. İlk durumda, eşit bir şekilde ağırlıklandırılmış monoton ceza, diğerinde ise ağırlıklandırılmış doğrusal ceza kullanılmıştır. İki problemi de çözmek için algoritma önerilmiştir. İki problemin polinomiyal zamanda çözülebildiği teoremler ispatlanmıştır.

Alidaee ve Womer [66] tarafından sunulan çalışmada zaman bağımlı işlem sürelerine göre çizelgelenen tek makina çizelgeleme problemi literatürü incelenmiştir. İşler, işlem süreleri fonksiyonlarının tipine göre sıralanmış, amaç fonksiyonları, karmaşıklıkları ve önerdikleri algoritmalar açısından incelenmiştir.

Cai ve arkadaşları [67] tarafından önerilen çalışmada i işinin çizelgelenmesi probleminde genel zaman bozulması düşünülmüştür. t işe başlama zamanı ve α_i ceza oranı olmak üzere, i işinin işlem süresi $p_i + \alpha_i f(t_0, t)$ 'dir. $f(t_0, t)$ ise, t_0 'dan önce sıfıra eşit olan ve t_0 'dan sonra t 'yi artıran bir fonksiyondur. Guo ve Wang [68] tarafından sunulan çalışmada tek makina ve bozulan işlerin grup teknolojisi varsayımı

ile çizelgelenmesi incelenmiştir. Çalışmada amaç fonksiyonu en büyük tamamlanma zamanını en küçükleme. Burada grup teknolojisi varsayımı altında bozulan işlerin tek makinada çizelgelenmesinin polinomiyal zamanda çözülebileceği gösterilmiştir.

Xu ve arkadaşları [69] tarafından önerilen çalışmada n adet iş f gruba ayrılmıştır, doğrusal bozulma etkisi altında grup teknolojisi yaklaşımı kullanılarak tek makina çizelgeleme problemi incelenmiştir. Amaç fonksiyonu toplam ağırlıklandırılmış tamamlanma zamanını en küçükleme olarak belirlenmiştir. Çalışmada $1|S, GT, a_{ij}(a + bt)| \sum w_j C_j$ probleminin polinom zamanda çözüldüğü ispatlanmıştır. Wang ve ark. [70] tarafından önerilen çalışmada tek makina çizelgeleme problemlerinde grup teknolojisi ve işlerin bozulmaya uğraması durumu ele alınmıştır. Burada aynı gruptaki işler birlikte işlenirler, farklı gruplar işlenirken hazırlık zamanı oluşur. İşlerin işlem sürelerinin işe başlama zamanlarının azalan bir fonksiyonu olduğu düşünülmüştür. Amaç olarak en büyük tamamlanma zamanı ve toplam tamamlanma zamanını en küçükleme ele alınmıştır. Bu problemleri çözebilecek polinomiyal zamanlı çözüm algoritmaları önerilmiştir.

Hsieh ve Bricker [71] tarafından önerilen çalışmada işlerin işlem zamanı, işe başlama zamanının doğrusal artan bir fonksiyonu olarak incelenmiştir. Model 1 ve Model 2 olmak üzere 2 farklı bozulma etkisinde çizelgeleme modelinde çalışılmış, birden çok makina düşünülmüştür. Burada amaç en büyük tamamlanma zamanını en küçükleme olarak sunulmuştur. Model 1 için etkili bir sezgisel önerilmiştir ve sezgiselden üretilen optimum en büyük tamamlanma zamanı sınırlandırılmış ve en büyük tamamlanma zamanı oranı ispatlanmıştır. Model 2’de istatistiksel bir sezgiseli içeren, en büyük tamamlanma zamanını en küçükleme amaçlayan üç sezgisel önerilmiştir. Sayısal sonuçlar, modelin etkinliğini göstermek için sunulmuştur.

Kovalyov ve Kubiak [72] tarafından önerilen çalışmada n adet bozulan işin tek makinada çizelgenerek en büyük tamamlanma zamanını en küçükleme amaçlayan, TPZTT kullanılmıştır.

Liu [73] tarafından önerilen çalışmada Atöyle Tipi Üretim (ATÜ)’in özel bir çeşidini çözmeye odaklanılmıştır. Burada işlerin değerleri zamanla üstel olarak bozulmaktadır.

İşlerin toplam değerlerini en büyükmeyi amaçlayan ATÜ problemi çözülmüştür. Yeni bir genetik algoritma metodu önerilmiş, sonuçlar sayısal olarak verilmiştir.

Mosheiov [74] çalışmasında, iş bağımlı bozulma oranı kullanmıştır. Bu çalışmada tek makina ve doğrusal bozulma durumu ele alınmıştır. Amaç olarak ağırlıklandırılmış tamamlanma zamanını en küçükmek belirlenmiştir. Ağırlıklar temel işlem zamanının oransal bir değeri olarak alınmıştır. Optimal çizelgenin Λ -şekilli olduğu, temel işlem zamanlarının tek yerel en büyük sıralaması olduğu gösterilmiştir. Problemin polinomial zamanda çözülebileceği gösterilmiştir. Genel ağırlık durumları için sezgiseller test edilmiştir. Doğrusal bozulan işlerin, iş bağımlı bozulma oranı ve orantısal ağırlıkları için tek bir optimal çizelge bulunmuş ve bu çizelgenin ağırlıklandırılmış toplam tamamlanma zamanını en küçüklediği gösterilmiştir. Mosheiov [75]'un bu çalışmasında ise işlerin işlem zamanı başlama zamanlarının artan bir fonksiyonuna göre etkilenmektedir. Çalışmada amaç olarak çok makinalı en büyük tamamlama zamanını en küçükleme ele alınmıştır. Basit doğrusal bozulma ve paralel eş makineler düşünülmüştür. İki makina için problemin NP-zor yapıda olduğu ispatlanmıştır. Ji ve Cheng [76] tarafından önerilen çalışmada işlerin işlem zamanlarının işe başlama zamanlarının doğrusal bir fonksiyonu olduğu durumda paralel makina çizelgeleme problemi ele alınmıştır. Çalışmada amaç olarak toplam tamamlanma zamanını en küçükmek ele alınmıştır. m adet eş makina için, TPZTT kullanılmıştır.

Voutsinas ve Pappis [77] tarafından sunulan çalışmada bozulma etkisinin üstel olarak ifade edildiği, yeni tip tek makina çizelgeleme problemi çalışılmıştır. Problemin alt optimum çözümlerini bulmak için sezgisel bir algoritma önerilmiş ve algoritmayı test etmek için yeni bir bilgisayar programı sunulmuştur. Algoritma test edilmiş ve optimum çizelgeye yakın çizelge gösterilmiştir. Çalışmada amaç olarak işlerin toplam değerlerini en büyükmek belirlenmiştir. Orijinal problemin NP-zor yapıda olduğu varsayılırken, önerilen sezgisel algoritmanın problemi polinomial zamanda çözebileceği gösterilmiştir.

Wang ve Cheng [78] tarafından önerilen çalışmada işe başlama zamanlarının artan bir fonksiyonu olan işlem zamanları ile tek makina çizelgeleme problemi incelenmiştir. Burada işlerin öncelikleri dikkate alınarak grafik oluşturulmuştur. En büyük tamamlanma zamanını en küçükleyecek genel doğrusal problem için polinomial

algoritmalar sunulmuştur. Aynı zamanda toplam ağırlıklandırılmış tamamlanma zamanını en küçüklenmesi probleminin de polinomial olarak çözülebildiği gösterilmiştir.

Wang ve Chu [79] tarafından önerilen çalışmada, geçmiş sıra bağımlı dağıtım zamanları ve bozulma etkisi altındaki işler için paralel makina çizelgeleme problemi sunulmuştur. Toplam iş yükü, toplam tamamlanma zamanı, tamamlanma zamanındaki toplam farklılık amaç fonksiyonları için polinomial zamanlı bir algoritma sunulmuştur.

De ve ark. [80] tarafından sunulan çalışmada işlerin tamamlanma zamanlarının varyansını en küçüklemek için tek makina çizelgeleme problemi incelenmiştir. Bu problemin polinomial zamanda çözümü yapılamadığından dinamik programlama algoritması sunulmuştur.

Kubiak [81] çalışmasında, Merten ve Muller [40] tarafından yapılan çalışmada açık olarak kalan tamamlanma zamanı varyansı en küçüklenmesi probleminin karmaşıklığını incelemiştir ve tamamlanma zamanı varyansı en küçüklenmesi probleminin NP-zor yapıda olduğunu ispatlamıştır.

Manna ve Prasad [82] tarafından sunulan çalışmada işlerin tamamlanma zamanlarının varyanslarını en küçükleyecek tek makina çizelgeleme problemi ele alınmıştır. Bu çalışmada yazarlar optimum çizelgedeki en küçük işin pozisyonu için sınır geliştirmişlerdir. Bu durum, optimum çizelge V-şekilli sıralamada olduğu için önemlidir. İşlerin işlem süreleri homojen olduğunda, sayısal deneylerin sonuçlarındaki sınırlar çok etkili olmaktadır. Çalışmada bu sınırların kullanıldığı yeni bir sezgisel yöntem geliştirilmiştir.

Su ve Chou [83] tarafından önerilen çalışmada toplam tamamlanma zamanı ve tamamlanma zamanı varyansını aynı anda en küçükleyen iki kriterli çizelgeleme problemi düşünülmüştür. Optimum çizelgenin V-biçimli olduğu sunulmuştur. İş sayısı 18'e kadar çıktığında bile sonuçların optimum olduğu verilmiştir. Karar indeksi temeline dayanan yeni bir sezgisel metot önerilmiştir ve değişik boyutlardaki problemler için çok iyi sonuçların bulunması sağlanmıştır. Önerilen sezgisel algoritmanın çözüm kalitesi %97 civarındadır.

Su ve Wang [84] tarafından yapılan çalışmada tozları temizlemek için üretimin durdurulduğu tek makina çizelgeleme problemi ele alınmıştır. Çalışmanın amacı işlerin tamamlanma zamanlarının toplam mutlak farklarının en küçüklenmesidir. Problemden üretim çizelgeleme ve temizleme aktivitesi beraber düşünülmüştür ve bu problem NP-zor yapıdadır. Optimum çözümü bulmak için karma tamsayılı programlama modeli ve etkili bir sezgisel yöntem geliştirilmiştir.

Yukarıda verilmiş olan literatür çalışması Tablo 1.1’de sunulduğu gibi özetlenebilir.

Tablo 1.1. Öğrenme etkisi, bozulma etkisi ve iş reddetme durumları üzerine yapılan çizelgeleme çalışmaları

Kaynak	Amaç fonksiyonu	Öğrenme etkisi	Bozulma etkisi	İş reddetme
[1, 4, 5, 6, 7, 10, 12, 14, 15, 17, 19, 22, 23, 28]	C_{maks}	√		
[2, 4, 5, 6, 7, 8, 10, 11, 14, 15, 16, 18, 19, 23, 27, 28]	$\sum C$	√		
[3]	$C_{maks}, \sum C, \sum (\alpha d + \beta E + \gamma C)$	√		
[4]	$\sum wC, L_{maks}, \sum U$	√		
[6, 9, 12, 19]	$\sum wC, L_{maks}$	√		
[9]	$\sum C^k$	√		
[11, 14, 21, 28]	$\sum wC$	√		
[12]	$T_{maks}, \sum T$	√		
[13, 18]	$\sum L$	√		
[18, 23]	$\sum TADC$	√		
[20]	$\sum (\alpha d + \beta E)$	√		
[24]	L_{maks}	√		
[29, 30, 37]	C_{maks}			√
[31]	$\sum wC$			√
[32]	$C_{maks}, \sum C, \sum wC$		√	√
[33]	C_{maks}, TL		√	√
[34]	$\sum C$			√

Tablo 1.1. (Devam) Öğrenme etkisi, bozulma etkisi ve iş reddetme durumları üzerine yapılan çizelgeleme çalışmaları

[35]	$E_{maks}, \sum E$			√
[36]	$C_{maks}, \sum C, L_{maks}, T_{maks}$		√	√
[39]	C_{maks}			√
[43, 47, 56, 60]	C_{maks}	√	√	
[43]	$\sum C^2$	√	√	
[49, 51, 53, 69]	$\sum TADC$		√	
[54, 61, 63, 67, 70]	$C_{maks}, \sum C$		√	
Kaynak	Amaç fonksiyonu	Öğrenme etkisi	Bozulma etkisi	İş reddetme
[43, 56]	$\sum C$	√	√	
[57]	$\sum wC, L_{maks}$	√	√	
[59, 64, 68, 71, 72, 78]	C_{maks}		√	
[60]	$\sum C^\delta, \sum wC$	√	√	
[62]	$\sum (\alpha E + \beta T + \gamma D + Gu)$		√	
[65]	$\sum wE$		√	
[74]	$\sum wC$		√	
[75, 79]	TL		√	
[76, 79]	$\sum C$		√	
[79]	$\sum TADC$		√	

Tablo 1.1’de de görüldüğü üzere literatürde öğrenme etkisi, bozulma etkisi ve iş reddetme durumlarının eş zamanlı olarak ele alındığı başka bir çalışma bulunmamaktadır. Bu nedenle bu tez çalışmasında bu problem üzerine çözüm yaklaşımları araştırılacaktır.

2. BÖLÜM

ÖĞRENME ETKİSİ ALTINDA İŞ REDDETMELİ ÇİZELGELEME PROBLEMLERİ

Bu bölümde, tek makina ve paralel makina ortamında öğrenme etkisi altında, iş reddetmeli çizelgeleme problemleri incelenmiştir.

2.1. Öğrenme Etkisi Altında İş Reddetmeli Maksimum Tamamlanma Zamanı Probleminin En Küçüklenmesi

Çalışmamızda tek makina, n adet iş düşünülmüştür. p_j , j işinin işlem süresidir ($i, j, r = 1, \dots, n$). Eğer j işi makinaya atanmazsa, δ_j kadar ceza maliyeti alır. α öğrenme katsayısıdır. Literatürde genellikle 0,8 [1, 2, 5] olarak alınmaktadır. X_{ijr} karar değişkeni, j işi r pozisyonuna atanırsa 1, aksi durumda 0 değeri alır. $i = 1$ değerini aldığımda, kabul edilen işler kümesine ait bir iş olduğu, $i = 2$ değerini aldığımda, reddedilen işler kümesine ait bir iş olduğu anlamına gelmektedir ve δ_j kadar ceza maliyeti almaktadır. En büyük tamamlanma zamanı C_{max} olarak gösterilmiştir.

Bu çalışmada öğrenme etkisi altında çizelgeleme problemleri iş reddetme durumu altında ele alınması problemin Teorem 1'de olduğu gibi bir dağıtım kuralı ile çözülememesine neden olmaktadır. Bu nedenle öğrenme etkisi altında iş reddetmeli maksimum tamamlanma zamanının en küçüklenmesi problemi için aşağıdaki matematiksel model önerilmiştir.

$$\min \sum_{j=1}^n \sum_{r=1}^n r^2 \cdot (p_j \cdot r^\alpha \cdot X_{1jr}) + \delta_j \cdot X_{2jr} \quad (2.1.1)$$

$$\sum_{j=1}^n X_{ijr} \leq 1 \quad (i = 1,2) \quad (r = 1,2,\dots,n) \quad (2.1.2)$$

$$\sum_{i=1}^2 \sum_{r=1}^n X_{ijr} = 1 \quad (j = 1,2,\dots,n) \quad (2.1.3)$$

$$X_{ijr} \in (0,1) \quad (2.1.4)$$

Eşitlik (2.1.1)'de sunulan amaç fonksiyonu maksimum tamamlanma zamanı ve reddetme maliyetinden oluşan toplam maliyetin en küçüklenmesidir. Toplam çizelgeleme maliyeti reddedilen işlerin ceza maliyetlerine, işlenen işlerin işlem sürelerinin maliyete dönüştürülmesiyle elde edilen maliyetlerin toplamıdır. Eşitlik (2.1.2) her bir işin ya kabul edilmesi ya da reddedilmesini sağlar. Eşitlik (2.1.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (2.1.4) karar değişkeninin 0 veya 1 değerini almasını sağlar.

2.2. Öğrenme Etkisi Altında İş Reddetmeli Toplam Tamamlanma Zamanı Probleminin En Küçüklenmesi

Klasik tek makina çizelgeleme problemlerinde toplam tamamlanma zamanı en kısa işlem süresi politikasıyla çözülebilir. İşler öğrenme indeksine bağlı olsa da optimum sıralama EKİS ile elde edilir [33]. j işi r pozisyonuna atandığında onun pozisyonel ağırlığı $n - j + 1$ 'dir. p_j , j işinin tamamlanma zamanıdır.

Bu çalışmada öğrenme etkisi altında çizelgeleme problemleri iş reddetme durumu altında ele alınması problemin Teorem 2'de olduğu gibi bir dağıtım kuralı ile çözülememesine neden olmaktadır. Bu nedenle öğrenme etkisi altında iş reddetmeli toplam tamamlanma zamanının en küçüklenmesi problemi için aşağıdaki matematiksel model önerilmiştir.

$$\min \sum_{j=1}^n \sum_{r=1}^n r \cdot (n - j + 1) \cdot (p_j \cdot r^\alpha \cdot X_{1jr}) + \delta_j \cdot X_{2jr} \quad (2.2.1)$$

$$\sum_{j=1}^n X_{ijr} \leq 1 \quad (i = 1,2) \quad (r = 1,2, \dots, n) \quad (2.2.2)$$

$$\sum_{i=1}^2 \sum_{r=1}^n X_{ijr} = 1 \quad (j = 1,2, \dots, n) \quad (2.2.3)$$

$$X_{ijr} \in (0,1) \quad (2.2.4)$$

Eşitlik (2.2.1)'de sunulan amaç fonksiyonu toplam tamamlanma zamanı ve reddetme maliyetinden oluşan toplam maliyetin en küçüklenmesidir. Toplam çizelgeleme maliyeti reddedilen işlerin ceza maliyetlerine, işlenen işlerin işlem sürelerinin maliyete dönüştürülmesiyle elde edilen maliyetlerin toplamıdır. Eşitlik (2.2.2) her bir işin ya kabul edilmesi ya da reddedilmesini sağlar. Eşitlik (2.2.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (2.2.4) karar değişkeninin 0 veya 1 değerini almasını sağlar.

2.3. Öğrenme Etkisi Altında İş Reddetmeli Tamamlanma Zamanlarının Mutlak Farkları Toplamı Probleminin En Küçüklenmesi

Kanet [48], tamamlanma zamanlarının mutlak farkları toplamı için konum ağırlıklarının $w_r = (r - 1)(n - r + 1), r = 1, \dots, n$ olduğunu belirtmiştir. j işi r pozisyonuna atandığında $(j, r = 1, \dots, n)$ pozisyonel ağırlığı $n - j + 1$ dir. p_j j işinin işlem süresidir ve TZMFT problemi için matematiksel model aşağıda sunulmuştur.

$$\min \sum_{j=1}^n \sum_{r=1}^n (r - 1)(n - j + 1). (p_j \cdot r^\alpha \cdot X_{1jr}) + \delta_j \cdot X_{2jr} \quad (2.3.1)$$

$$\sum_{j=1}^n X_{ijr} \leq 1 \quad (i = 1,2) \quad (r = 1,2, \dots, n) \quad (2.3.2)$$

$$\sum_{i=1}^2 \sum_{r=1}^n X_{ijr} = 1 \quad (j = 1, 2, \dots, n) \quad (2.3.3)$$

$$X_{ijr} \in (0,1) \quad (2.3.4)$$

Eşitlik (2.3.1)'de sunulan amaç fonksiyonu TZMFT ve reddetme maliyetinden oluşan toplam maliyetin en küçüklenmesidir. Eşitlik (2.3.2) her bir işin ya kabul edilmesi ya da reddedilmesini sağlar. Eşitlik (2.3.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (2.3.4) karar değişkenininin 0 veya 1 değerini almasını sağlar.

2.4. Öğrenme Etkisi Altında Paralel Makinalı ve İş Reddetmeli Maksimum Tamamlanma Zamanı Probleminin En Küçüklenmesi

Paralel makina ortamında n adet iş düşünülmüştür. p_j j işinin işlem zamanı, C_{max} maksimum tamamlanma zamanıdır. $i = 1$ iken Makina 1'de $i = 2$ iken Makine 2'de çizelgelenen j işi $i = 3$ iken reddedilmektedir. δ_j j işinin ceza maliyetidir. α öğrenme etkisidir. X_{ijr} karar değişkeni, j işi r pozisyonuna atanırsa 1, aksi durumda 0 değerini alır.

Teorem 1'de $1|p_{j[r]} = r(p_j)r^\alpha|C_{max}$ problemi için optimum çözümün işlerin işlem sürelerine göre EUIS kuralına göre sıralanmasıyla elde edileceği sunulmuştu. $2|p_{j[r]} = r(p_j)r^\alpha|C_{max}$ problemi için de aynı kural ile optimum çizelge elde edilecektir.

Bu çalışmada öğrenme etkisi altında çizelgeleme problemleri iş reddetme durumu altında ele alınması problemin Teorem 1'de olduğu gibi bir dağıtım kuralı ile çözülememesine neden olmaktadır. Bu nedenle öğrenme etkisi altında paralel makinalı ve iş reddetmeli maksimum tamamlanma zamanının en küçüklenmesi problemi için aşağıdaki matematiksel model önerilmiştir.

$$\min \sum_{j=1}^n p_j \cdot r^\alpha \cdot r \cdot X_{1jr} + \sum_{j=1}^n p_j \cdot r^\alpha \cdot r \cdot X_{2jr} + \delta_j \cdot X_{3jr} \quad (2.4.1)$$

$$\sum_{j=1}^n X_{ijr} \leq 1 \quad (i = 1, 2, 3) \quad (r = 1, 2, \dots, n) \quad (2.4.2)$$

$$\sum_{i=1}^3 \sum_{r=1}^n X_{ijr} = 1 \quad (j = 1, 2, \dots, n) \quad (2.4.3)$$

$$X_{ijr} \in (0,1) \quad (2.4.4)$$

Eşitlik (2.4.1)'de sunulan amaç fonksiyonu maksimum tamamlanma zamanı ve reddetme maliyetinden oluşan toplam maliyetin en küçüklenmesidir. Eşitlik (2.4.2) her bir işin ya kabul edilmesi ya da reddedilmesini sağlar. Eşitlik (2.4.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (2.4.4) karar değişkeninin 0 veya 1 değerini almasını sağlar

Bu bölümde önerilen matematiksel modellerin Lingo paket programında yazılan kodları EK 1 - EK 4'de sunulmuştur.

3. BÖLÜM

BOZULMA VE ÖĞRENME ETKİSİ ALTINDA İŞ REDDETME Lİ ÇİZELGELEME PROBLEMLERİ

Bu bölümde, tek makina ve paralel makina ortamında bozulma ve öğrenme etkisi altında, iş reddetmeli çizelgeleme problemleri incelenmiştir.

3.1. Bozulma ve Öğrenme Etkisi Altında İş Reddetmeli Maksimum Tamamlanma Zamanı Probleminin En Küçüklenmesi

Çalışmamızda tek makina, n adet iş düşünülmüştür. p_{jr} , r pozisyonuna atanmış olan j işinin işlem süresidir ($i, j, r = 1, \dots, n$). Eğer j işi makinaya atanmazsa, δ_j kadar ceza maliyeti alır. α öğrenme katsayısıdır. Literatürde genellikle 0,8 [1, 2, 5] olarak alınmaktadır. γ bozulma katsayısıdır ($1 > \gamma > 0$). T_r r pozisyonunda işe başlayan işin işe başlama zamanıdır. X_{ijr} karar değişkeni, j işi r pozisyonuna atanırsa 1, aksi durumda 0 değeri alır. $i = 1$ değerini aldığıında, kabul edilen işler kümesine ait bir iş olduğu, $i = 2$ değerini aldığıında, reddedilen işler kümesine ait bir iş olduğu anlamına gelmektedir ve δ_j kadar ceza maliyeti almaktadır. En büyük tamamlanma zamanı C_{max} olarak gösterilmiştir.

Bu çalışmada öğrenme etkisi altında çizelgeleme problemleri iş reddetme durumu altında ele alınması problemin Teorem 3'de olduğu gibi bir dağıtım kuralı ile çözülememesine neden olmaktadır. Bu nedenle bozulma ve öğrenme etkisi altında iş reddetmeli maksimum tamamlanma zamanının en küçüklenmesi problemi için aşağıdaki matematiksel model önerilmiştir.

$$\min \sum_{j=1}^n \sum_{r=1}^n r^2 \cdot ((p_j + \gamma \cdot T_r) \cdot r^\alpha \cdot X_{1jr}) + \delta_j \cdot X_{2jr} \quad (3.1.1)$$

$$\sum_{j=1}^n X_{ijr} \leq 1 \quad (i = 1,2) \quad (r = 1,2, \dots, n) \quad (3.1.2)$$

$$\sum_{i=1}^2 \sum_{r=1}^n X_{ijr} = 1 \quad (j = 1,2, \dots, n) \quad (3.1.3)$$

$$T_1 = 0 \quad (3.1.4)$$

$$T_r = \sum_{j=1}^n \sum_{k=1}^{r-1} r^2 \cdot ((p_j + \gamma \cdot T_k) \cdot k^\alpha \cdot X_{1jk}) \quad (r = 2,3, \dots, n) \quad (3.1.5)$$

$$X_{ijr} \in (0,1) \quad (3.1.6)$$

Eşitlik (3.1.1)'de sunulan amaç fonksiyonu öğrenme ve bozulma etkisi altında maksimum tamamlanma zamanı ve reddetme maliyetinden oluşan toplam maliyetin en küçüklenmesidir. Toplam çizelgeleme maliyeti reddedilen işlerin ceza maliyetlerine, işlenen işlerin işlem sürelerinin maliyete dönüştürülmesiyle elde edilen maliyetlerin toplamıdır. Eşitlik (3.1.2) her bir işin ya kabul edilmesi ya da reddedilmesini sağlar. Eşitlik (3.1.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (3.1.4) birinci işin işe başlama zamanının 0 olduğunu ifade eder. Eşitlik (3.1.5) işlenen işlerin işe başlama zamanlarını işlerin pozisyonlarına göre güncellemektedir. Eşitlik (3.1.6) karar değişkenininin 0 veya 1 değerini almasını sağlar.

3.2. Bozulma ve Öğrenme Etkisi Altında İş Reddetmeli Toplam Tamamlanma Zamanı Probleminin En Küçüklenmesi

j işi r pozisyonuna atandığında onun pozisyonel ağırlığı $n - j + 1$ 'dir. p_j , j işinin tamamlanma zamanıdır ve optimum akış zamanı çözümü aşağıdaki atama probleminin çözümüyle elde edilir.

Çalışmamızda tek makina, n adet iş düşünülmüştür. p_{jr} , r pozisyonuna atanmış olan j işinin işlem süresidir ($i, j, r = 1, \dots, n$). Eğer j işi makinaya atanmazsa, δ_j kadar ceza

maliyeti alır. α öğrenme katsayısıdır. Literatürde genellikle 0,8 [1, 2, 5] olarak alınmaktadır. γ bozulma katsayısıdır ($1 > \gamma > 0$). T_r r pozisyonunda işe başlayan işin işe başlama zamanıdır. X_{ijr} karar değişkeni, j işi r pozisyonuna atanırsa 1, aksi durumda 0 değeri alır. $i = 1$ değerini aldığıında, kabul edilen işler kümesine ait bir iş olduğu, $i = 2$ değerini aldığıında, reddedilen işler kümesine ait bir iş olduğu anlamına gelmektedir ve δ_j kadar ceza maliyeti almaktadır.

Bu çalışmada öğrenme etkisi altında çizelgeleme problemleri iş reddetme durumu altında ele alınması problemin Teorem 4'te olduğu gibi bir dağıtım kuralı ile çözülememesine neden olmaktadır. Bu nedenle bozulma ve öğrenme etkisi altında iş reddetmeli toplam tamamlanma zamanının en küçüklenmesi problemi için aşağıdaki matematiksel model önerilmiştir.

$$\min \sum_{j=1}^n \sum_{r=1}^n \left(\frac{r}{2}\right) \cdot (n - j + 1) \cdot ((p_j + \gamma \cdot T_r) \cdot r^\alpha \cdot X_{1jr}) + \delta_j \cdot X_{2jr} \quad (3.2.1)$$

$$\sum_{j=1}^n X_{ijr} \leq 1 \quad (i = 1,2) \quad (r = 1,2, \dots, n) \quad (3.2.2)$$

$$\sum_{i=1}^2 \sum_{r=1}^n X_{ijr} = 1 \quad (j = 1,2, \dots, n) \quad (3.2.3)$$

$$T_1 = 0 \quad (3.2.4)$$

$$T_r = \sum_{j=1}^n \sum_{k=1}^{r-1} r^2 \cdot ((p_j + \gamma \cdot T_k) \cdot k^\alpha \cdot X_{1jk}) \quad (r = 2, 3, \dots, n) \quad (3.2.5)$$

$$X_{ijr} \in (0,1) \quad (3.2.6)$$

Eşitlik (3.2.1)'de sunulan amaç fonksiyonu öğrenme ve bozulma etkisi altında toplam tamamlanma zamanı ve reddetme maliyetinden oluşan toplam maliyetin en küçüklenmesidir. Toplam çizelgeleme maliyeti reddedilen işlerin ceza maliyetlerine, işlenen işlerin işlem sürelerinin maliyete dönüştürülmesiyle elde edilen maliyetlerin toplamıdır. Eşitlik (3.2.2) her bir işin ya kabul edilmesi ya da reddedilmesini sağlar. Eşitlik (3.2.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (3.2.4) birinci işin işe başlama zamanının 0 olduğunu ifade eder. Eşitlik (3.2.5) işlenen işlerin

işe başlama zamanlarını işlerin pozisyonlarına göre güncellemektedir. Eşitlik (3.2.6) karar değişkeninin 0 veya 1 değerini almasını sağlar.

3.3. Bozulma ve Öğrenme Etkisi Altında İş Reddetmeli Tamamlanma Zamanlarının Mutlak Farkları Toplamı Probleminin En Küçüklenmesi

Kanet [48], tamamlanma zamanlarının mutlak farkları toplamı için konum ağırlıklarının $w_r = (r - 1)(n - r + 1), r = 1, \dots, n$ olduğunu belirtmiştir. j işi r pozisyonuna atandığında ($j, r = 1, \dots, n$) pozisyonel ağırlığı $n - j + 1$ dir. p_j j işinin işlem süresidir ve TZMFT problemi için matematiksel model aşağıda sunulmuştur.

$$\min \sum_{j=1}^n \sum_{r=1}^n (r - 1)(n - j + 1) \cdot (p_j + \gamma \cdot T_r) \cdot r^\alpha \cdot X_{1jr} + \delta_j \cdot X_{2jr} \quad (3.3.1)$$

$$\sum_{j=1}^n X_{ijr} \leq 1 \quad (i = 1, 2) \quad (r = 1, 2, \dots, n) \quad (3.3.2)$$

$$\sum_{i=1}^2 \sum_{r=1}^n X_{ijr} = 1 \quad (j = 1, 2, \dots, n) \quad (3.3.3)$$

$$T_1 = 0 \quad (3.3.4)$$

$$T_r = \sum_{j=1}^n \sum_{k=1}^{r-1} ((p_j + \gamma \cdot T_k) \cdot k^\alpha \cdot X_{1jk}) \quad (r = 2, 3, \dots, n) \quad (3.3.5)$$

$$X_{ijr} \in (0, 1) \quad (3.3.6)$$

Amaç, TZMFT ve reddetme maliyetinin en küçüklenmesidir. Eşitlik (3.3.1) amaç fonksiyonumuzdur, Eşitlik (3.3.2) her bir işin ya kabul edilmesi ya da reddedilmesini sağlar. Eşitlik (3.3.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (3.3.4) birinci işin işe başlama zamanının 0 olduğunu ifade eder. Eşitlik (3.3.5) işlerin işe başlama zamanlarını pozisyon değerlerine göre güncellemektedir. Eşitlik (3.3.6) karar değişkenimizin 0 ya da 1 değerini alacağını ifade eder.

Eşitlik (3.3.1)'de sunulan amaç fonksiyonu TZMFT ve reddetme maliyetinden oluşan toplam maliyetin en küçüklenmesidir. Eşitlik (3.3.2) her bir işin ya kabul edilmesi ya da

reddedilmesini sağlar. Eşitlik (3.3.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (3.3.4) birinci işin işe başlama zamanının 0 olduğunu ifade eder. Eşitlik (3.3.5) işlenen işlerin işe başlama zamanlarını işlerin pozisyonlarına göre güncellemektedir. Eşitlik (3.3.6) karar değişkenininin 0 veya 1 değerini almasını sağlar.

3.4. Bozulma ve Öğrenme Etkisi Altında Paralel Makinalı ve İş Reddetmeli Maksimum Tamamlanma Zamanı Probleminin En Küçüklenmesi

Paralel makina ortamında n adet iş düşünülmüştür. p_j j işinin işlem zamanıdır. $i = 1$ iken Makina 1'de $i = 2$ iken Makine 2'de çizelgelenen j işi $i = 3$ iken reddedilmektedir. δ_j j işinin ceza maliyetidir. α öğrenme etkisidir. X_{ijr} karar değişkeni, j işi r pozisyonuna atanırsa 1, aksi durumda 0 değerini alır.

Teorem 3'te $I|p_{j[r]} = r(p_j + \gamma \cdot T_r)r^\alpha |C_{max}$ problemi için optimum çözümün işlerin işlem sürelerine göre EUIS kuralına göre sıralanmasıyla elde edileceği sunulmuştur. $2|p_{j[r]} = r(p_j + \gamma \cdot T_r)r^\alpha |C_{max}$ problemi için de aynı kural ile optimum çizelge elde edilecektir.

Bu çalışmada bozulma ve öğrenme etkisi altında çizelgeleme problemleri iş reddetme durumu altında ele alınması problemin Teorem 3'te olduğu gibi bir dağıtım kuralı ile çözülememesine neden olmaktadır. Bu nedenle bozulma ve öğrenme etkisi altında paralel makinalı ve iş reddetmeli maksimum tamamlanma zamanının en küçüklenmesi problemi için aşağıdaki matematiksel model önerilmiştir.

$$\min \sum_{j=1}^n r \cdot (p_j + \gamma \cdot T_r) \cdot r^\alpha \cdot X_{1jr} + \sum_{j=1}^n r \cdot (p_j + \gamma \cdot T_{1r}) \cdot r^\alpha \cdot X_{2jr} + \delta_j \cdot X_{3jr} \quad (3.4.1)$$

$$\sum_{j=1}^n X_{ijr} \leq 1 \quad (i = 1, 2, 3) \quad (r = 1, 2, \dots, n) \quad (3.4.2)$$

$$\sum_{i=1}^3 \sum_{r=1}^n X_{ijr} = 1 \quad (j = 1, 2, \dots, n) \quad (3.4.3)$$

$$T_1 = 0 \quad (3.4.4)$$

$$T_r = \sum_{j=1}^n \sum_k^{r-1} ((p_j + \gamma \cdot T_k) \cdot k^\alpha \cdot X_{1jk}) \quad (r = 2, 3, \dots, n) \quad (3.4.5)$$

$$T1_1 = 0 \quad (3.4.6)$$

$$T1_r = \sum_{j=1}^n \sum_k^{r-1} ((p_j + \gamma \cdot T1_k) \cdot k^\alpha \cdot X_{1jk}) \quad (r = 2, 3, \dots, n) \quad (3.4.7)$$

$$X_{ijr} \in (0,1) \quad (3.4.8)$$

Eşitlik (3.4.1)'de sunulan amaç fonksiyonu maksimum tamamlanma zamanı ve reddetme maliyetinden oluşan toplam maliyetin en küçüklenmesidir. Eşitlik (3.4.2) her bir işin ya kabul edilmesi ya da reddedilmesini sağlar. Eşitlik (3.4.3) tüm işlerin tek bir pozisyona atanmasını garanti eder. Eşitlik (3.4.4) birinci makinada kabul edilen birinci işin işe başlama zamanının 0 olduğunu ifade eder. Eşitlik (3.4.5) birinci makinada işlenen işlerin işe başlama zamanlarını işlerin pozisyonlarına göre güncellemektedir. Eşitlik (3.4.6) ikinci makinada kabul edilen birinci işin işe başlama zamanının 0 olduğunu ifade eder. Eşitlik (3.4.7) ikinci makinada işlenen işlerin işe başlama zamanlarını işlerin pozisyonlarına göre güncellemektedir. Eşitlik (3.4.8) karar değişkeninin 0 veya 1 değerini almasını sağlar.

Bu bölümde önerilen matematiksel modellerin Lingo paket programında yazılan kodları EK 5 - EK 8'de sunulmuştur.

4. BÖLÜM

ÇÖZÜM ALGORİTMALARI

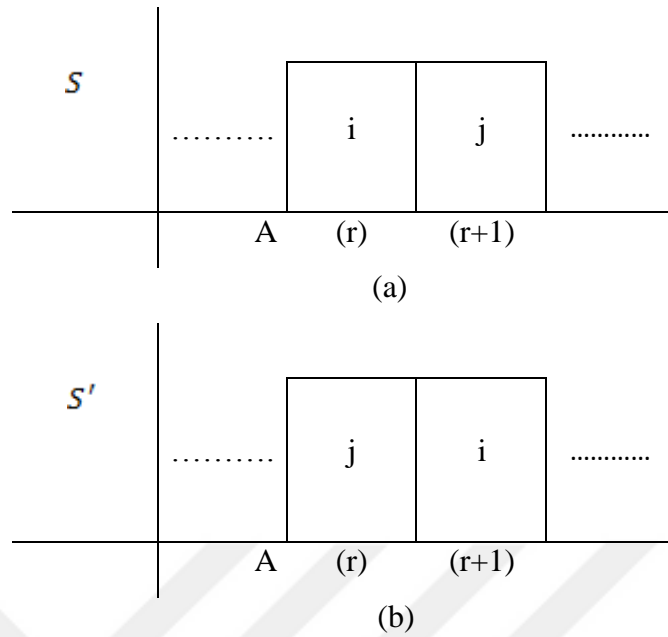
Öncelikle çözüm algoritmalarında kullanılacak olan teoremler sunulacaktır.

Tek makina çizelgeleme özelliklerinde, tamamlanma zamanı değeri sıradan bağımsızdır. Kabul edilen işler için öğrenme etkisi altında tamamlanma zamanının en küçüklenmesi çizelgeleme problemi en küçük işlem süresi kuralı ile optimum çizelge elde edilir, burada ardışık iş çiftinin yer değiştirmesi yaklaşımı ile teoremler ispatlanabilir.

Teorem 1: $|p_{j[r]} = r^2 p_j r^\alpha | C_{max}$ problemin optimum çözümü işlerin işlem sürelerine göre azalan sırada çizelgelenmesi ile elde edilir.

İspat:

Öğrenme etkisi altında, n adet iş tek makina problemi düşünülmüştür. $\alpha = \log(a)/\log(2) < 0$ 'dır burada α öğrenme etkisi, a ise öğrenme eğrisi değeridir $p_i \leq p_j$ varsayılmıştır. r işin çizelgelendiği pozisyon, $A(r-1)$ pozisyonunda çizelgelenen işin tamamlanma zamanıdır.



Şekil 4.1. i ve j işlerinin S ve S' çizelgelerinde gösterimi

Şekil 4.1. (a)'da gösterilen S çizelgesi için i ve j işlerinin tamamlanma zamanları

$$C_i(S) = A + r^2(p_i)r^\alpha \quad (4.1.1)$$

$$C_j(S) = A + r^2(p_j)r^\alpha + (r + 1)^2(p_j)(r + 1)^\alpha \quad (4.1.2)$$

S çizelgesinde ardışık iş çifti olan i ve j işlerinin yer değiştirmesi ile S' çizelgesi elde edilmiştir.

Şekil 4.1. (b)'de gösterilen S' çizelgesi için i ve j işlerinin tamamlanma zamanları

$$C_j(S') = A + r^2(p_j)r^\alpha \quad (4.1.3)$$

$$C_i(S') = A + r^2(p_i)r^\alpha + (r + 1)^2(p_i)(r + 1)^\alpha \quad (4.1.4)$$

Eşitlik (4.1.2) ve (4.1.4)'ün farkları alınarak aşağıdaki durum elde edilmiştir.

$$C_i(S') - C_j(S) = (p_i - p_j)(-r^2r^\alpha + (r + 1)^2(r + 1)^\alpha)$$

$C_i(S') - C_j(S) = (p_j - p_i)(r^{2+\alpha} - (r+1)^{2+\alpha})$ ve $\alpha < 0$ ve $p_i \leq p_j$ ise

$$C_i(S') - C_j(S) < 0.$$

Bu durum S' çizelgesinin S çizelgesini domine ettiğini gösterir ve $1|p_{j[r]} = r^2 p_j r^\alpha |C_{max}$ probleminin optimum çözümünün en uzun işlem süresi (EUİS) kuralı ile elde edildiğini ispatlar.

Biskup [4] tarafından sunulan $1|p_{j[r]} = p_j r^\alpha |C_{max}$ problemi çözümü için önerilen yöntem EKİS kuralı iken, çalışmamızdaki $1|p_{j[r]} = r^2 p_j r^\alpha |C_{max}$ problemi için önerdiğimiz yöntem EUİS kuralı olmuştur.

Teorem 2 sunulmadan önce bu teoremin ispatında kullanılacak olan yardımcı önerme belirtilmiştir.

Yardımcı Önerme 1: $2r^{\alpha+1} - (r+1)^{\alpha+1}$ ifadesinde $\alpha < 0$ ve $r > 0$ iken $2r^{\alpha+1} - (r+1)^{\alpha+1} \geq 0$ 'dir.

İspat:

$$f(r) = 2r^{\alpha+1} - (r+1)^{\alpha+1} \tag{4.1.5}$$

$$f'(r) = (\alpha+1)2r^\alpha - (\alpha+1)(r+1)^\alpha \tag{4.1.6}$$

$\alpha > 0$ iken

$$f(r) = (\alpha+1)(2r^\alpha - (r+1)^\alpha) > 0$$

Teorem 2: $1|p_{j[r]} = r p_j r^\alpha | \sum C$ problemin optimum çözümü işlerin EKİS kuralına göre sıralanmasıyla elde edilir.

İspat:

Öğrenme etkisi altında, n adet iş tek makina problemi düşünülmüştür. $\alpha = \log(a)/\log(2) < 0$ 'dir burada α öğrenme etkisi, a ise öğrenme eğrisi değeridir

$p_i \leq p_j$ varsayılmıştır. r işin çizelgelendiği pozisyon, $A(r-1)$ pozisyonunda çizelgelenen işin tamamlanma zamanıdır.

Şekil 4.1 (a)'da gösterilen S çizelgesi için i ve j işlerinin tamamlanma zamanları

$$C_i(S) = A + r(p_i)r^\alpha \quad (4.1.7)$$

$$C_j(S) = A + r(p_i)r^\alpha + (r+1)(p_j)(r+1)^\alpha \quad (4.1.8)$$

$$\sum C(S) = 2A + 2r(p_i)r^\alpha + (r+1)(p_j)(r+1)^\alpha \quad (4.1.9)$$

Şekil 4.1 (b)'de gösterilen S' çizelgesi için i ve j işlerinin tamamlanma zamanları

S çizelgesinde ardışık iş çifti olan i ve j işlerinin yer değiştirmesi ile S' çizelgesi elde edilmiştir.

$$C_j(S') = A + r(p_j)r^\alpha \quad (4.1.10)$$

$$C_i(S') = A + r(p_j)r^\alpha + (r+1)(p_i)(r+1)^\alpha \quad (4.1.11)$$

$$\sum C(S') = 2A + 2r(p_j)r^\alpha + (r+1)(p_i)(r+1)^\alpha \quad (4.1.12)$$

Eşitlik (4.1.9) ve (4.1.12)'in farkları alınarak aşağıdaki durum elde edilmiştir.

$$\sum C(S') - \sum C(S) = (p_j - p_i)(2rr^\alpha - (r+1)^{1+\alpha})$$

$$= (p_j - p_i)(2r^{\alpha+1} - (r+1)^{\alpha+1}) > 0$$

Yardımcı Önerme 1 ve Teorem 2'ye göre S çizelgesinin S' çizelgesini domine ettiği görülmektedir. Böylece $1|p_{j[r]} = rp_jr^\alpha | \sum C$ probleminin optimum çözümünün EKİS kuralı ile elde edildiğini ispatlar.

□

Biskup [4] tarafından yapılan çalışmada $I|p_{j[r]} = p_j r^\alpha | \Sigma C$ problemi çözümünün EKİS kuralı ile elde edildiği gösterilmiştir. Çalışmamızdaki $I|p_{j[r]} = r p_j r^\alpha | \Sigma C$ problemi için de EKİS kuralına göre çizelgelendiğinde optimum çizelgenin elde edileceği gösterilmiştir.

Teorem 3: $I|p_{j[r]} = r^2(p_j + \gamma \cdot T_r)r^\alpha | C_{max}$ problemin optimum çözümü işlerin EUIŞ kuralına göre çizelgelenmesiyle elde edilir.

İspat:

Öğrenme etkisi altında, n adet iş tek makina problemi düşünülmüştür. $\alpha = \log(a)/\log(2) < 0$ 'dır burada α öğrenme etkisi, a ise öğrenme eğrisi değeridir $p_i \leq p_j$ varsayılmıştır. r işin çizelgelendiği pozisyon, $A(r-1)$ pozisyonunda çizelgelenen işin tamamlanma zamanıdır.

Şekil 4.1 (a)'da gösterilen S çizelgesi için i ve j işlerinin tamamlanma zamanları

$$C_i(S) = A + r^2(P_i + \gamma \cdot T_r)r^\alpha \quad (4.1.13)$$

$$C_j(S) = A + r^2(P_i + \gamma \cdot T_r)r^\alpha + (r+1)^2(P_j + \gamma \cdot T_{r+1})(r+1)^\alpha \quad (4.1.14)$$

Şekil 4.1 (b)'de gösterilen S' çizelgesi için i ve j işlerinin tamamlanma zamanları

S çizelgesinde ardışık iş çifti olan i ve j işlerinin yer değiştirmesi ile S' çizelgesi elde edilmiştir.

$$C_j(S') = A + r^2(P_j + \gamma \cdot T_r)r^\alpha \quad (4.1.15)$$

$$C_i(S') = A + r^2(P_j + \gamma \cdot T_r)r^\alpha + (r+1)^2(P_i + \gamma \cdot T_{r+1})(r+1)^\alpha \quad (4.1.16)$$

Eşitlik (4.1.14) ve (4.1.16)'nın farkları alınarak aşağıdaki durum elde edilmiştir.

$$C_i(S') - C_j(S) = (P_j - P_i)(r^{2+\alpha} - (r+1)^{2+\alpha}(1-\gamma)) \text{ ve } \alpha < 0 \text{ ve } P_i \leq P_j \text{ ise}$$

$$C_i(S') - C_j(S) < 0.$$

Teorem 3'e göre S' çizelgesinin S çizelgesini domine ettiği görülmektedir. Böylece $1|p_{j[r]} = r^2(p_j + \gamma.T_r)r^\alpha | C_{max}$ probleminin optimum çözümünün EUİS kuralı ile elde edildiğini ispatlar.

Teorem 4: $1|p_{j[r]} = \left(\frac{r}{2}\right) \cdot (p_j + \gamma.T_r) \cdot r^\alpha | \sum C$ problemin optimum çözümü işlerin EKİS kuralına göre çizelgelenmesiyle elde edilir.

İspat:

Öğrenme etkisi altında, n adet iş tek makina problemi düşünülmüştür. $\alpha = \log(a)/\log(2) < 0$ 'dır burada α öğrenme etkisi, a ise öğrenme eğrisi değeridir $p_i \leq p_j$ varsayılmıştır. r işin çizelgelendiği pozisyon, $A(r-1)$ pozisyonunda çizelgelenen işin tamamlanma zamanıdır.

Şekil 4.1 (a)'da gösterilen S çizelgesi için i ve j işlerinin tamamlanma zamanları

$$C_i(S) = A + \left(\frac{r}{2}\right) (p_i + \gamma.T_r)r^\alpha \quad (4.1.17)$$

$$C_j(S) = A + \left(\frac{r}{2}\right) (p_i + \gamma.T_r)r^\alpha + \left(\frac{r+1}{2}\right) (p_i + \gamma(p_j + \gamma.T_r)) (r+1)^\alpha \quad (4.1.18)$$

$$\sum C(S) = 2A + r(p_i + \gamma.T_r)r^\alpha + \left(\frac{r+1}{2}\right) (p_i + \gamma(p_j + \gamma.T_r)) (r+1)^\alpha \quad (4.1.19)$$

Şekil 4.1 (b)'de gösterilen S' çizelgesi için i ve j işlerinin tamamlanma zamanları

S çizelgesinde ardışık iş çifti olan i ve j işlerinin yer değiştirmesi ile S' çizelgesi elde edilmiştir.

$$C_j(S') = A + \left(\frac{r}{2}\right) (p_j + \gamma.T_r)r^\alpha \quad (4.1.20)$$

$$C_i(S') = A + \left(\frac{r}{2}\right) (p_j + \gamma.T_r)r^\alpha + \left(\frac{r+1}{2}\right) (p_j + \gamma(p_i + \gamma.T_r)) (r+1)^\alpha \quad (4.1.21)$$

$$\sum C(S') = 2A + r(p_j + \gamma.T_r)r^\alpha + \left(\frac{r+1}{2}\right) (p_j + \gamma(p_i + \gamma.T_r)) (r+1)^\alpha \quad (4.1.22)$$

Eşitlik (4.1.19) ve (4.1.22)'nin farkları alınarak aşağıdaki durum elde edilmiştir.

$$\begin{aligned} \sum C(S') - \sum C(S) &= (p_j - p_i) \left(rr^\alpha - \left(\frac{r+1}{2}\right)(r+1)^\alpha \right) \\ &= (p_j - p_i) \left(2r^{\alpha+1} - \left(\frac{r+1}{2}\right)(r+1)^\alpha(1-\gamma) \right) > 0 \end{aligned}$$

Teorem 4'e göre S çizelgesinin S' çizelgesini domine ettiği görülmektedir. Böylece $1 \left| p_{j[r]} = \left(\frac{r}{2}\right) \cdot (p_j + \gamma \cdot T_r) \cdot r^\alpha \right| \sum C$ probleminin optimum çözümünün EKİS kuralı ile elde edildiğini ispatlar.

Oron [49] çalışmasında, basit bozulma etkisi altında tek makina çizelgeleme problemi incelemiştir ve tamamlanma zamanlarının mutlak farkları toplamını en küçüklemeyi amaçlamıştır. Problemin çözümü için sezgisel bir algoritma önerilmiştir.

Bu çalışmada Oron [49] tarafından sunulan sezgisel algoritma, öğrenme etkisi altında iş reddetmeli tamamlanma zamanlarının mutlak farkları toplamı çizelgeleme problemi için revize edilmiştir.

Bu bölümde ikinci ve üçüncü bölümde tanımlanan modellerin çözümü için algoritmalar önerilmiştir. Öğrenme etkisi altındaki tek makina ve paralel makina ortamında ve hem öğrenme hem bozulma etkisi altındaki tek makina ve paralel makina ortamında maksimum tamamlanma zamanı ve reddetme maliyeti en küçüklenmesi problemleri için, öğrenme ve bozulma etkisi altında iş reddetmeli maksimum tamamlanma zamanı en küçüklenmesi algoritması geliştirilmiştir.

4.1. Öğrenme ve Bozulma Etkisi Altında İş Reddetmeli Maksimum Tamamlanma Zamanı En Küçüklenmesi İçin Çözüm Algoritması

A : Kabul edilen işler, R : Reddedilen işler, n : İş sayısı, p_j j işin işlem süresi, δ_j j işinin ceza maliyeti

iken

TM : Toplam Maliyet = $\sum_{n=1}^A C_{max} + \sum_{n=1}^R \delta_j$ ise iş reddetmeli, öğrenme ve bozulma etkisi altında maksimum tamamlanma zamanının en küçüklenmesi için algoritmanın adımları aşağıda sunulmuştur.

Adım 1:

$i = 0, A = n, R = 0$. Tüm işleri kabul et. Kabul edilen işleri EUİS'e göre sırala.

$C(n, i)$ Kombinasyonu kadar çizelgenin TM'sini hesapla.

Adım 2:

$i = i + 1, A = n - 1, R = i$. Kabul edilen işleri 1 azalt, reddedilen işleri i kadar artır.

Kabul edilen işleri EUİS'e göre sırala.

$C(n, i)$ Kombinasyonu kadar çizelgenin TM'sini hesapla.

Adım 3:

Eğer $i < n$ ise Adım 2'yi tekrarla. $i = n$ ise dur ve Adım 4'e git.

Adım 4:

$\min \{TM_1, TM_2, \dots, TM_{2^n}\}$ ile elde edilen en küçük toplam maliyetli çizelgeyi belirle.

Sayısal Örnek 1:

İşlerin işlem süreleri $p_1 = 20, p_2 = 12, p_3 = 30, p_4 = 18$ ve ceza maliyetleri $\delta_1 = 30, \delta_2 = 20, \delta_3 = 25, \delta_4 = 25$ 'dir. Yukarıdaki algoritmaya göre algoritma adımları aşağıdaki gibidir.

Adım 1:

$A = 4, R = 0$,

Sıralama: 3 - 1 - 4 - 2, $TM = 80$.

Adım 2 (İterasyon 1):

$A = 3, R = 1$,

Sıralama: 3 - 1 - 2, $TM = 90$,

Sıralama: 1 - 4 - 2, $TM = 88$,

Sıralama: 3 - 1 - 4, $TM = 75$,

Sıralama: 3 - 4 - 2, $TM = 87$.

Adım 2 (İterasyon 2):

$$A = 2, R = 2,$$

$$\text{Sıralama: } 1 - 2, TM = 82,$$

$$\text{Sıralama: } 3 - 1, TM = 95,$$

$$\text{Sıralama: } 1 - 4, TM = 83,$$

$$\text{Sıralama: } 3 - 2, TM = 97,$$

$$\text{Sıralama: } 4 - 2, TM = 85,$$

$$\text{Sıralama: } 3 - 4, TM = 98.$$

Adım 2 (İterasyon 3):

$$A=1, R=3,$$

$$\text{Sıralama: } 1, TM = 90,$$

$$\text{Sıralama: } 2, TM = 92,$$

$$\text{Sıralama: } 3, TM = 105,$$

$$\text{Sıralama: } 4, TM = 93,$$

Adım 3:

$$A = 0, R = 4, TM = 100.$$

Adım 4:

$$\min \{80, 90, \dots, 100\} = 75, \text{ Sıralama: } 3 - 1 - 4.$$

Sadece öğrenme etkisi altında ve hem öğrenme hem bozulma etkisi altında tek makina ortamında toplam tamamlanma zamanı ve reddetme maliyeti en küçüklenmesi problemleri için, yukarıda önerilen çözüm algoritması kullanılmıştır.

4.2. Öğrenme ve Bozulma Etkisi Altında İş Reddetmeli Toplam Tamamlanma Zamanı En Küçüklenmesi İçin Çözüm Algoritması

A : Kabul edilen işler, R : Reddedilen işler, n : İş sayısı, p_j j işin işlem süresi, δ_j j işinin ceza maliyeti

iken

TM : *Toplam Maliyet* = $\sum_{n=1}^A C + \sum_{n=1}^R \delta_j$ ise iş reddetmeli, öğrenme ve bozulma etkisi altında toplam tamamlanma zamanının en küçüklenmesi için algoritmanın adımları aşağıda sunulmuştur.

Adım 1:

$i = 0, A = n, R = 0$. Tüm işleri kabul et. Kabul edilen işleri EKİS'e göre sırala.

$C(n, i)$ Kombinasyonu kadar çizelgenin TM 'sini hesapla.

Adım 2:

$i = i + 1, A = n - 1, R = i$. Kabul edilen işleri 1 azalt, reddedilen işleri i kadar artır. Kabul edilen işleri EKİS'e göre sırala.

$C(n, i)$ Kombinasyonu kadar çizelgenin TM 'sini hesapla.

Adım 3:

Eğer $i < n$ ise Adım 2'yi tekrarla. $i = n$ ise dur ve Adım 4'e git.

Adım 4:

$\min \{TM_1, TM_2, \dots, TM_{2^n}\}$ ile elde edilen en küçük toplam maliyetli çizelgeyi belirle.

Sayısal Örnek 2:

İşlerin işlem süreleri $p_1 = 8, p_2 = 10, p_3 = 6, p_4 = 12$ ve ceza maliyetleri $\delta_1 = 20, \delta_2 = 25, \delta_3 = 15, \delta_4 = 30$ 'dur. Yukarıdaki algoritmaya göre algoritma adımları aşağıdaki gibidir.

Adım 1:

$$A = 4, R = 0,$$

$$\text{Sıralama: } 3 - 1 - 2 - 4, TM = 36.$$

Adım 2 (İterasyon 1):

$$A = 3, R = 1,$$

$$\text{Sıralama: } 3 - 1 - 2, TM = 54,$$

$$\text{Sıralama: } 1 - 2 - 4, TM = 45,$$

$$\text{Sıralama: } 3 - 1 - 4, TM = 51,$$

$$\text{Sıralama: } 3 - 2 - 4, TM = 48.$$

Adım 2 (İterasyon 2):

$$A = 2, R = 2,$$

$$\text{Sıralama: } 1 - 2, TM = 63,$$

$$\text{Sıralama: } 3 - 1, TM = 69,$$

$$\text{Sıralama: } 1 - 4, TM = 60,$$

$$\text{Sıralama: } 3 - 2, TM = 66,$$

$$\text{Sıralama: } 2 - 4, TM = 57,$$

$$\text{Sıralama: } 3 - 4, TM = 63.$$

Adım 2 (İterasyon 3):

$$A=1, R=3,$$

$$\text{Sıralama: } 1, TM = 78,$$

$$\text{Sıralama: } 2, TM = 75,$$

$$\text{Sıralama: } 3, TM = 81,$$

$$\text{Sıralama: } 4, TM = 72,$$

Adım 3:

$$A = 0, R = 4, TM = 90.$$

Adım 4:

$\min \{36, 54, \dots, 90\} = 36$, Sıralama: 3 – 1 – 2 – 4.

Sadece öğrenme etkisi altında ve hem öğrenme hem bozulma etkisi altındaki tek makina ortamında toplam tamamlanma zamanları mutlak farkı ve reddetme maliyeti en küçüklenmesi problemleri için, yukarıda önerilen çözüm algoritması kullanılmıştır.

Oron [49] tarafından tarif edilen sezgisel algoritma, iş reddi ve öğrenmenin etkisi altındaki işler için geliştirilmiş ve kullanılmıştır.

4.3. Öğrenme ve Bozulma Etkisi Altında İş Reddetmeli TZMFT En Küçüklenmesi İçin Çözüm Algoritması

A : Kabul edilen işler, R : Reddedilen işler, n : İş sayısı, p_j j işin işlem süresi, δ_j j işinin ceza maliyeti, t : Kabul edilen iş sayısı

iken

TM : Toplam Maliyet = $\sum_{k=1}^A TZMFT + \sum_{n=1}^R \delta_j$ ise iş reddetmeli, öğrenme ve bozulma etkisi altında TZMFT en küçüklenmesi için algoritmanın adımları aşağıda sunulmuştur.

Eğer t çiftse Adım 1'e git. t tekse Adım 3'e git.

Adım 1:

$J_{t/2+2-i}$ İşini i pozisyonuna ata. $i = 1, 2, \dots, \frac{t}{2} + 1$

Adım 2:

J_i İşini i pozisyonuna ata. $i = \frac{t}{2} + 2, \frac{t}{2} + 3, \dots, t$.

Sonuç Formu: $\{J_{\frac{t}{2}+1}, J_{\frac{t}{2}}, \dots, J_3, J_2, J_1, J_{\frac{t}{2}+2}, J_{\frac{t}{2}+3}, \dots, J_t\}$

Adım 3:

$J_{\frac{t+5}{2}-i}$ İşini i pozisyonuna ata. $i = 1, 2, \dots, \frac{t+3}{2}$

Adım 4:

J_i işini i pozisyonuna ata. $i = \frac{t+5}{2}, \frac{t+7}{2}, \dots, t$. Adım 5'e git.

Sonuç Formu: $\{J_{\frac{t+3}{2}}, J_{\frac{t+1}{2}}, \dots, J_3, J_2, J_1, J_{\frac{t+5}{2}}, J_{\frac{t+7}{2}}, \dots, J_t\}$

Adım 5:

TM'leri hesapla.

Adım 6:

$\min \{TM_1, TM_2, \dots, TM_{2^n}\}$. Tüm TM'leri karşılaştır, en küçük olanı al. En küçük TM'nin elde edildiği çizelge optimumdur.

Sayısal Örnek 3:

İşlerin işlem süreleri $p_1 = 8, p_2 = 10, p_3 = 6, p_4 = 12$ ve ceza maliyetleri $\delta_1 = 20, \delta_2 = 25, \delta_3 = 15, \delta_4 = 30$ 'dur. Yukarıdaki algoritmaya göre algoritma adımları aşağıdaki gibidir.

$t = 4$ olduğu için Adım 1'den devam et,

Adım 1 (İterasyon 1):

$A = 4, R = 0, t = 4$ için, $J_{t/2+2-i} = J_3$ işi 1. pozisyona atanır.

Adım 2 (İterasyon 1):

Sıralama: 3 – 2 – 1 – 4, $TM = 36$.

Adım 3 (İterasyon 1):

$A = 3, R = 1, t = 3$ için, $J_{\frac{t+5}{2}-i} = J_3$ işi 1. pozisyona atanır.

Adım 4 (İterasyon 1):

Sıralama: 3 – 2 – 1 $TM = 54$.

Adım 1 (İterasyon 2):

$A = 2, R = 2, t = 2$ için, $J_{t/2+2-i} = J_2$ işi 1. pozisyona atanır.

Adım 2 (İterasyon 2):

Sıralama: $2 - 1, TM = 63$.

Adım 3 (İterasyon 2):

$A = 1, R = 3, t = 1$ için, $J_{\frac{t+5}{2}-i} = J_2$ işi 1. pozisyona atanır.

Adım 4 (İterasyon 2):

Sıralama: $2, TM = 75$.

Adım 1 (İterasyon 3):

$A = 0, R = 4, t = 0$ için, $TM = 90$.

$\min \{36, 54, \dots, 90\} = 36$, Sıralama: $3 - 2 - 1 - 4$.

5. BÖLÜM

BULGULAR

Öğrenme etkisi altında, öğrenme ve bozulma etkisi altındaki işlerin reddetme maliyeti en küçüklenmesi problemleri için öğrenme oranı 0.8, 0.7 ve 0.6 alınarak, bozulma oranı 0.1 ve 0.2 alınarak, 10 iş, 20 iş ve 30 iş için hem matematiksel modeller çözülmüş, hem de önerilen algoritmalarla çözümler elde edilmiştir. İşlerin işlem süreleri [1,25] aralığında, işlerin ceza maliyetleri ise [20,60] aralığında rassal olarak üretilmiştir.

Bölüm 2 ve Bölüm 3'te tanımlanan matematiksel modeller LINGO 11.0 paket programı ile kodlanmış ve çözümler 10 iş, 20 iş ve 30 işli test problemleri için üç farklı öğrenme oranı ve iki farklı bozulma oranına göre çözülmüştür.

Önerilen algoritmalar Visual C# 2017 programında kodlanmış ve matematiksel modellerle karşılaştırmaları yapılmıştır.

Tablo 5.1. Öğrenme etkisi altındaki 10 iş, 20 iş ve 30 işli test problemlerinin matematiksel model ve algoritma sonuçları

İş Sayısı	Amaç Fonksiyonu	Öğrenme Oranı	Matematiksel Model Sonucu	Algoritma Sonucu	Sonuçların Karşılaştırılması
10 iş	C_{max}	80%	299,20	299,20	100,00*
		70%	296,80	296,80	100,00*
		60%	294,40	294,40	100,00*
	Toplam C	80%	344,40	344,40	100,00*
		70%	339,60	339,60	100,00*
		60%	334,80	334,80	100,00*
	$TZMFT$	80%	275,7010	315,1	87,50
		70%	266,2727	305,4	87,19
		60%	254,8009	296,4	85,97
	Paralel Makina C_{max}	80%	229,0757	229,07	100,00*
		70%	218,6545	218,65	100,00*
		60%	201,4616	201,98	99,74

Tablo 5.1.(Devam) Öğrenme etkisi altındaki 10 iş, 20 iş ve 30 işli test problemlerinin matematiksel model ve algoritma sonuçları

İş Sayısı	Amaç Fonksiyonu	Öğrenme Oranı	Matematiksel Model Sonucu	Algoritma Sonucu	Sonuçların Karşılaştırılması
20 iş	C_{max}	80%	683,7557	683,7557	100,00*
		70%	672,1345	672,1345	100,00*
		60%	651,9363	651,9363	100,00*
	Toplam C	80%	765,9568	765,9568	100,00*
		70%	758,3409	758,3409	100,00*
		60%	749,6508	749,6508	100,00*
	TZMFT	80%	695,5641	696,44	99,87
		70%	682,7572	692,43	98,60
		60%	657,3797	687,54	95,61
	Paralel Makina C_{max}	80%	456,5677	458,06	99,67
		70%	386,7888	388,09	99,66
		60%	306,9943	307,92	99,70
30 iş	C_{max}	80%	1048,819	1048,819	100,00*
		70%	1026,9	1026,9	100,00*
		60%	994,7478	994,7478	100,00*
	Toplam C	80%	1162,02	1162,02	100,00*
		70%	1150,39	1150,39	100,00*
		60%	1139,37	1139,37	100,00*
	TZMFT	80%	1065,19	1084,26	98,24
		70%	1041,33	1054,89	98,71
		60%	1008,8	1019,34	98,97
	Paralel Makina C_{max}	80%	748,363	750,22	99,75
		70%	658,357	661,56	99,52
		60%	523,41	531,78	98,43

* ile işaretlenen değerler optimum değerlerdir.

Tablo 5.1’de öğrenme etkisi altında iş reddetmeli; 10 iş, 20 iş ve 30 iş problemleri için matematiksel model ve algoritma sonuçları verilmiştir. Üç farklı öğrenme oranına (0.8, 0.7 ve 0.6) göre hesaplamalar yapılmıştır. Sonuçların karşılaştırılması değeri matematiksel model sonucunun algoritma sonucuna bölünmesiyle elde edilmiştir. Yani $Sonuçların\ Karşılaştırılması = \frac{Matematiksel\ Model\ Sonucu}{Algoritma\ Sonucu}$ dur.

Tablo 5.1’de de görüldüğü üzere 10 iş için maksimum tamamlanma zamanı ve tamamlanma zamanları toplamı problemleri için %100, TZMFT problemi için ortalama

%86 ve paralel makina maksimum tamamlanma zamanı problemi için ortalama %100 değeri elde edilmiştir.

Tablo 5.1'de de görüldüğü üzere 20 iş için maksimum tamamlanma zamanı ve tamamlanma zamanları toplamı problemleri için %100, TZMFT problemi için ortalama %98 paralel makina maksimum tamamlanma zamanı problemi için ortalama %99 değeri elde edilmiştir.

Tablo 5.1'de de görüldüğü üzere 30 iş için maksimum tamamlanma zamanı ve tamamlanma zamanları toplamı problemleri için %100, TZMFT problemi için ortalama %98,6 paralel makina maksimum tamamlanma zamanı problemi için ortalama %99 değeri elde edilmiştir.

İş sayısı arttığında doğruluk değerinin de arttığı görülmektedir

Tablo 5.2. Öğrenme ve bozulma etkisi altındaki 10 iş, 20 iş ve 30 işli test problemlerinin matematiksel model ve algoritma sonuçları

İş Sayısı	Amaç Fonksiyonu	Öğrenme Oranı	Bozulma Oranı	Matematiksel Model Sonucu	Algoritma Sonucu	Sonuçların Karşılaştırılması
10 iş	C_{max}	80%	0,1	303,04	303,04	100,00*
		80%	0,2	306,88	306,88	100,00*
		70%	0,1	300,16	300,16	100,00*
		70%	0,2	303,52	303,52	100,00*
		60%	0,1	297,28	297,28	100,00*
		60%	0,2	300,16	300,16	100,00*
	Toplam C	80%	0,1	310,68	310,68	100,00*
		80%	0,2	311,16	311,16	100,00*
		70%	0,1	304,22	304,22	100,00*
		70%	0,2	306,457	306,457	100,00*
		60%	0,1	297,566	297,565	100,00*
		60%	0,2	299,358	299,358	100,00*
	TZMFT	80%	0,1	282,995	324,406	87,23
		80%	0,2	290,558	326	89,13
		70%	0,1	272,409	309,353	88,06
		70%	0,2	278,736	326	85,50
		60%	0,1	262,413	295,822	88,71
		60%	0,2	267,594	310,034	86,31
	Paralel Makina C_{max}	80%	0,1	247,489**	213,72	115,80
		80%	0,2	258,244**	218,75	118,05
		70%	0,1	235,213**	200,108	117,54
		70%	0,2	248,0227**	206,43	120,15
		60%	0,1	221,1609**	189,545	116,68
		60%	0,2	229,111**	193,532	118,38

* ile işaretlenen değerler optimum değerlerdir.

** ile işaretlenen değerler 5 saatlik çalışma sonunda elde edilmiş değerlerdir.

Tablo 5.2. (Devam) Öğrenme ve bozulma etkisi altındaki 10 iş, 20 iş ve 30 işli test problemlerinin matematiksel model ve algoritma sonuçları

İş Sayısı	Amaç Fonksiyonu	Öğrenme Oranı	Bozulma Oranı	Matematiksel Model Sonucu	Algoritma Sonucu	Sonuçların Karşılaştırılması
20 iş	C_{max}	80%	0,1	669,41	669,41	100,00*
		80%	0,2	705,25	686,989	102,66
		70%	0,1	660,0025	660	100,00*
		70%	0,2	672,9083	672,909	100,00*
		60%	0,1	642,02	642,016	100,00*
		60%	0,2	659,5101	660,973	99,78
	Toplam C	80%	0,1	755,083	741,815	101,79
		80%	0,2	783,72	742,986	105,48
		70%	0,1	720,87	738,35	97,63
		70%	0,2	721,397	739,254	97,58
		60%	0,1	666,43	722,92	92,19
		60%	0,2	667,556	726,067	91,94
	TZMFT	80%	0,1	649,9	684,872	94,46
		80%	0,2	662,648	706,073	93,85
		70%	0,1	633,294	662,227	95,63
		70%	0,2	644,799	691,13	93,30
		60%	0,1	617,18	644,34	95,78
		60%	0,2	629,8	666,926	94,43
	Paralel Makina C_{max}	80%	0,1	515,082**	505,923	101,81
		80%	0,2	547,573**	499,316	109,66
		70%	0,1	457,664**	465,08	98,41
		70%	0,2	506,676**	494,553	102,45
		60%	0,1	362,634**	374,616	96,80
		60%	0,2	410,821**	193,532	101,25
30 iş	C_{max}	80%	0,1	1083,16	1073,858	100,87
		80%	0,2	1105,33	1087,684	101,62
		70%	0,1	1059,79	1059,789	100,00*
		70%	0,2	1177,61	1072,05	109,85
		60%	0,1	1039,59	1031,77	100,76
		60%	0,2	1052,3	1045,187	100,68
	Toplam C	80%	0,1	1126,73	1109,673	101,54
		80%	0,2	1125,16	1086,94	103,52
		70%	0,1	1067,42	1055,78	101,10
		70%	0,2	1069,39	1060,86	100,8
		60%	0,1	971,308	1014,356	95,76
		60%	0,2	973,65	1022,706	95,20
	TZMFT	80%	0,1	1144,5	1164	98,32
		80%	0,2	1090,738	1164	93,71
		70%	0,1	1097,204	1157,96	94,75
		70%	0,2	1073,058	1164	92,19
		60%	0,1	1038,64	1147,96	90,48
		60%	0,2	1044,337	1159,22	90,09
	Paralel Makina C_{max}	80%	0,1	867,894**	873,527	99,36
		80%	0,2	905,968**	867,065	104,49
		70%	0,1	846,391**	842,933	100,41
		70%	0,2	866,881**	865,08	100,21
		60%	0,1	676,287**	662,485	102,08
		60%	0,2	759,909**	742,194	102,39

* ile işaretlenen değerler optimum değerlerdir.

** ile işaretlenen değerler 5 saatlik çalışma sonunda elde edilmiş değerlerdir.

Tablo 5.2’de öğrenme ve bozulma etkisi altında 10 iş, 20 iş ve 30 iş problemleri için matematiksel model ve algoritma sonuçları verilmiştir. Üç farklı öğrenme oranına (0.8, 0.7 ve 0.6) ve iki farklı bozulma oranına (0.1 ve 0.2) göre hesaplamalar yapılmıştır. Sonuçların karşılaştırılması değeri matematiksel model sonucunun algoritma sonucuna bölünmesiyle elde edilmiştir. Yani

$$\text{Sonuçların Karşılaştırılması} = \frac{\text{Matematiksel Model Sonucu}}{\text{Algoritma Sonucu}}, \text{dur.}$$

Tablo 5.2’de de görüldüğü üzere 10 iş için maksimum tamamlanma zamanı ve tamamlanma zamanları toplamı problemleri için %100, TZMFT problemi için ortalama %87 ve paralel makina maksimum tamamlanma zamanı problemi için ise algoritma sonuçlarımız daha iyi çıkmıştır.

Tablo 5.2’den görüleceği üzere 20 iş için maksimum tamamlanma zamanı için %100, tamamlanma zamanları toplamı problemi için ortalama %97, TZMFT problemi için ortalama %94 ve paralel makina maksimum tamamlanma zamanı problemi için ortalama %100 değerleri elde edilmiştir.

Tablo 5.2’de de görüldüğü üzere 30 iş için maksimum tamamlanma zamanı için %100, tamamlanma zamanları toplamı problemi için ortalama %99, TZMFT problemi için ortalama %93 ve paralel makina maksimum tamamlanma zamanı problemi için ortalama %100 değerleri elde edilmiştir.

Tablo 5.2’de de gösterildiği üzere paralel makina maksimum tamamlanma zamanı problemi için matematiksel model sonuçları 5 saatlik çalıştırma sonucunda elde edilmiştir.

Tablo 5.1 ve Tablo 5.2’den de görüleceği üzere önerdiğimiz algoritmalar ve matematiksel model sonuçları karşılaştırıldığında değerlerin ortalama yüzde 90’ın üzerinde olduğu görülmektedir. TZMFT için optimum çözüm bulan bir algoritma olmadığından elde ettiğimiz çözümlerin doğruluk değerleri daha büyük önem kazanmaktadır.

6. BÖLÜM

SONUÇ

Gerçek hayat problemlerinde işler öğrenme etkisi ve bozulma etkisi altına girebilirler. İşlerin işlem süreleri de bu etkiler sonucunda değişebilmektedir. Ayrıca işlerin işlem sürelerinin uzun olması, mesai maliyetlerinin fazla olması ya da fason maliyetlerinin daha ucuz olmasından dolayı işlerin işletmede üretilmemesi durumu da söz konusu olabilmektedir.

Öğrenme etkisi ve bozulma etkisinin önemi yıllar içerisinde daha da artmıştır. Bu çalışmada öğrenme etkisi altında iş reddetmeli çizelgeleme problemleri ile öğrenme etkisi ve bozulma etkisi altında iş reddetmeli çizelgeleme problemleri incelenmiştir. Öğrenme etkisi olarak sıra bağımlı öğrenme etkisi, bozulma etkisi olarak doğrusal bozulma etkisi ele alınmıştır.

Literatürde yapılan çalışmalarda, öğrenme etkisi ve iş reddetme ile öğrenme etkisi, bozulma etkisi ve iş reddetme durumları bir arada hiç kullanılmamıştır. Bu çalışma, üç öğrenme etkisi ve iki bozulma etkisinin birleşimlerinden elde edilen toplam sekiz farklı modeli ele alarak, bu etkilerin değişimlerinin sonuca etkisini analiz edebilmektedir. Öğrenme etkisi altında iş reddetmeli çizelgeleme problemleri ile öğrenme etkisi ve bozulma etkisi altında iş reddetmeli çizelgeleme problemleri paralel makina ortamı için de düşünülmüştür. Amaç fonksiyonu olarak toplam çizelgeleme maliyeti düşünülmüştür. Toplam çizelgeleme maliyeti reddedilen işlerin ceza maliyetlerine, işlenen işlerin işlem sürelerinin maliyete dönüştürülmesiyle elde edilen maliyetlerin toplam değeridir. Bütün bu yönleri ile yapılan çalışmanın sonuçlarının, literatüre büyük katkıları olduğu düşünülmektedir.

Bu çalışmada, iş reddetme kısıtı altında öğrenme ve bozulma etkilerinin kombinasyonlarından elde edilen modeller altında problemlerin optimum çözümlerinin

EUIS ve EKIS kuralı ile elde edilebileceği ispatlanmıştır. Çalışılan problemin matematiksel modeli LINGO paket programı ile modellenmiş, önerilen algoritmalar C# programı kullanılarak kodlanmıştır. LINGO sonuçları ve algoritma sonuçları karşılaştırılarak algoritmanın etkinliği gösterilmiştir.

Öğrenme etkisi altında iş reddetmeli çizelgeleme problemlerinde üç farklı öğrenme oranı (0.8,0.7,0.6) kullanılmıştır. Öğrenme etkisi altındaki 10 iş, 20 iş ve 30 iş problemlerinin matematiksel ve algoritma sonuçlarında iş sayısı arttığında algoritma sonucu ile matematiksel model sonucunun yaklaşık olarak aynı sonuç verdiği görülmektedir.

Bozulma ve öğrenme etkisi altında iş reddetmeli çizelgeleme problemlerinde üç öğrenme oranı (0.8, 0.7, 0.6) ve iki bozulma oranı (0.1, 0.2) kullanılmıştır. Bozulma ve öğrenme etkisi altındaki 10 iş, 20 iş ve 30 iş problemlerinin matematiksel ve algoritma sonuçlarında iş sayısı arttığında algoritma sonucu ile matematiksel model sonucunun yaklaşık olarak aynı sonuç verdiği görülmektedir.

TZMFT problemlerinin çözümü için literatürde optimum çözümü veren bir yöntem olmamasına rağmen, kullandığımız sezgisel algoritmadan elde edilen sonuçlarımızın en kötü değeri bile %85 doğrulukla elde edilmiştir.

KAYNAKÇA

1. Mosheiov, G., 2001. Scheduling problems with a learning effect. **European Journal of Operational Research**, **132**: 687– 693.
2. Mosheiov, G., 2001. Parallel machine scheduling with a learning effect. **Journal of the Operational Research Society**, **52**(10): 1165-1169.
3. Mosheiov, G., Sidney, J. B., 2003. Scheduling with general job-dependent learning curves. **European Journal of Operational Research**, **147**(3), 665-670.
4. Biskup, D., 1999. Single-machine scheduling with learning considerations. **European Journal of Operational Research**, **115**(1): 173-178.
5. Koulamas, C., Kyparisis, G. J., 2007. Single-machine and two-machine flowshop scheduling with general learning functions. **European Journal of Operational Research**, **178**(2): 402-407.
6. Yin, Y., Xu, D., Sun, K., Li, H., 2009. Some scheduling problems with general position-dependent and time-dependent learning effects. **Information Sciences**, **179**(14): 2416-2425.
7. Kuo, W. H., Yang, D. L., 2006. Single-machine group scheduling with a time-dependent learning effect. **Computers & Operations Research**, **33**(8): 2099-2112.
8. Kuo, W. H., Yang, D. L., 2006. Minimizing the total completion time in a single-machine scheduling problem with a time-dependent learning effect. **European Journal of Operational Research**, **174**(2): 1184-1190.
9. Kuo, W. H., Yang, D. L., 2007. Single-machine scheduling problems with the time-dependent learning effect. **Computers & Mathematics with Applications**, **53**(11): 1733-1739.
10. Kuo, W. H., 2012. Single-machine group scheduling with time-dependent learning effect and position-based setup time learning effect. **Annals of Operations Research**, **196**(1): 349-359.

11. Low, C., Lin, W. Y., 2011. Minimizing the total completion time in a single-machine scheduling problem with a learning effect. **Applied Mathematical Modelling**, **35**(4): 1946-1951.
12. Cheng, T. E., Lai, P. J., Wu, C. C., Lee, W. C., 2009. Single-machine scheduling with sum-of-logarithm-processing-times-based learning considerations. **Information Sciences**, **179**(18): 3127-3135.
13. Eren, T., Güner, E., 2007. Minimizing total tardiness in a scheduling problem with a learning effect. **Applied Mathematical Modelling**, **31**(7): 1351-1361.
14. Wu, C. C., Lee, W. C., 2008. Single-machine scheduling problems with a learning effect. **Applied Mathematical Modelling**, **32**(7): 1191-1197.
15. Lee, W. C., Wu, C. C., 2009. A note on single-machine group scheduling problems with position-based learning effect. **Applied Mathematical Modelling**, **33**(4): 2159-2163.
16. Ji, M., Cheng, T. E., 2010. Scheduling with job-dependent learning effects and multiple rate-modifying activities. **Information Processing Letters**, **110**(11), 460-463.
17. Cheng, T. C. E., Wu, C. C., Chen, J. C., Wu, W. H., Cheng, S. R., 2013. Two-machine flowshop scheduling with a truncated learning function to minimize the makespan. **International Journal of Production Economics**, **141**(1), 79-86.
18. Liu, M., 2013. Parallel-machine scheduling with past-sequence-dependent delivery times and learning effect. **Applied Mathematical Modelling**, **37**(23), 9630-9633.
19. Lee, W. C., Wu, C. C., 2009. Some single-machine and m-machine flowshop scheduling problems with learning considerations. **Information Sciences**, **179**(22), 3885-3892.
20. Eren, T., 2009. A bicriteria parallel machine scheduling with a learning effect of setup and removal times. **Applied Mathematical Modelling**, **33**(2), 1141-1150.

21. Eren, T., 2009. Minimizing the total weighted completion time on a single machine scheduling with release dates and a learning effect. **Applied Mathematics and Computation**, **208**(2), 355-358.
22. Wu, C. C., Lee, W. C., 2009. Single-machine and flowshop scheduling with a general learning effect model. **Computers & Industrial Engineering**, **56**(4), 1553-1558.
23. Yin, N., Wang, X. Y., 2011. Single-machine scheduling with controllable processing times and learning effect. **The International Journal of Advanced Manufacturing Technology**, **54**(5-8), 743-748.
24. Cheng, T. E., Wang, G., 2000. Single machine scheduling with learning effect considerations. **Annals of Operations Research**, **98**(1-4), 273-290.
25. Biskup, D., 2008. A state-of-the-art review on scheduling with learning effects. **European Journal of Operational Research**, **188**(2), 315-329.
26. Azzouz, A., Ennigrou, M., Ben Said, L., 2017. Scheduling problems under learning effects: classification and cartography. **International Journal of Production Research**, 1-20.
27. Li, D. C., Hsu, P. H., Wu, C. C., Cheng, T. E., 2011. Two-machine flowshop scheduling with truncated learning to minimize the total completion time. **Computers & Industrial Engineering**, **61**(3), 655-662.
28. Toksan, M. D., Arik, O. A., 2017. Single machine scheduling problems under position-dependent fuzzy learning effect with fuzzy processing times. **Journal of Manufacturing Systems**, **45**, 159-179.
29. Bartal, Y., Leonardi, S., Marchetti-Spaccamela, A., Sgall, J., Stougie, L., 2000. Multiprocessor scheduling with rejection. **SIAM Journal on Discrete Mathematics**, **13**(1): 64-78.
30. Zhang, L., Lu, L., Yuan, J., 2010. Single-machine scheduling under the job rejection constraint. **Theoretical Computer Science**, **411**(16): 1877-1882.
31. Engels, D. W., Karger, D. R., Kolliopoulos, S. G., Sengupta, S., Uma, R. N., Wein, J., 2003. Techniques for scheduling with rejection. **Journal of Algorithms**, **49**(1): 175-191.

32. Li, S., Yuan, J., 2010. Parallel-machine scheduling with deteriorating jobs and rejection. **Theoretical Computer Science**, **411**(40), 3642-3650.
33. Gerstl, E., Mosheiov, G., 2012. Scheduling on parallel identical machines with job-rejection and position-dependent processing times. **Information Processing Letters**, **112**(19), 743-747.
34. Epstein, L., Noga, J., Woeginger, G. J., 2002. On-line scheduling of unit time jobs with rejection: minimizing the total completion time. **Operations Research Letters**, **30**(6), 415-420.
35. Koulamas, C., Panwalkar, S. S., 2015. On the equivalence of single machine earliness/tardiness problems with job rejection. **Computers & Industrial Engineering**, **87**, 1-3, 2015.
36. Cheng, Y., Sun, S., 2009. Scheduling linear deteriorating jobs with rejection on a single machine. **European Journal of Operational Research**, **194**(1), 18-27.
37. Shabtay, D., Gaspar, N., Kaspi, M., 2013. A survey on offline scheduling with rejection. **Journal of Scheduling**, **16**(1), 3-28.
38. Hoogeveen, H., Skutella, M., Woeginger, G. J., 2003. Preemptive scheduling with rejection. **Mathematical Programming**, **94**(2-3), 361-374.
39. Shabtay, D., Gaspar, N., Yedidsion, L., 2012. A bicriteria approach to scheduling a single machine with job rejection and positional penalties. **Journal of Combinatorial Optimization**, **23**(4), 395-424.
40. Merten, A. G., Muller, M. E., 1972. Variance minimization in single machine sequencing problems. **Management Science**, **18**(9), 518-528.
41. Toksarı, M. D., Güner, E., 2008. Minimizing the earliness/tardiness costs on parallel machine with learning effects and deteriorating jobs: a mixed nonlinear integer programming approach. **The International Journal of Advanced Manufacturing Technology**, **38**(7-8), 801-808.
42. Toksarı, M. D., Güner, E., 2009. Parallel machine earliness/tardiness scheduling problem under the effects of position based learning and linear/nonlinear deterioration. **Computers & Operations Research**, **36**(8), 2394-2417.

43. Toksarı, M. D., Oron, D., Güner, E., 2009. Single machine scheduling problems under the effects of nonlinear deterioration and time-dependent learning. **Mathematical and Computer Modelling**, **50**(3-4), 401-406.
44. Toksarı, M. D., Güner, E., 2009. Scheduling problems with the nonlinear effects of learning and deterioration. **The International Journal of Advanced Manufacturing Technology**, **45**(7-8), 801-807.
45. Toksarı, M. D., Güner, E., 2010. Parallel machine scheduling problem to minimize the earliness/tardiness costs with learning effect and deteriorating jobs. **Journal of Intelligent Manufacturing**, **21**(6), 843-851.
46. Toksarı, M. D., Güner, E., 2010. The common due-date early/tardy scheduling problem on a parallel machine under the effects of time-dependent learning and linear and nonlinear deterioration. **Expert Systems with Applications**, **37**(1), 92-112.
47. Toksarı, M. D., 2011. A branch and bound algorithm for minimizing makespan on a single machine with unequal release times under learning effect and deteriorating jobs. **Computers & Operations Research**, **38**(9), 1361-1365.
48. Kanet, J. J., 1981. Minimizing variation of flow time in single machine systems. **Management Science**, **27**(12), 1453-1459.
49. Oron, D., 2008. Single machine scheduling with simple linear deterioration to minimize total absolute deviation of completion times. **Computers & Operations Research**, **35**(6), 2071-2078.
50. Mor, B., Mosheiov, G., 2011. Total absolute deviation of job completion times on uniform and unrelated machines. **Computers & Operations Research**, **38**(3), 660-665.
51. Li, Y., Li, G., Sun, L., Xu, Z., 2009. Single machine scheduling of deteriorating jobs to minimize total absolute differences in completion times. **International Journal of Production Economics**, **118**(2), 424-429.
52. Wang, J. B., Xia, Z. Q., 2007. Single machine scheduling problems with controllable processing times and total absolute differences penalties. **European Journal of Operational Research**, **177**(1), 638-645.

53. Huang, X., Wang, M. Z., 2011. Parallel identical machines scheduling with deteriorating jobs and total absolute differences penalties. **Applied Mathematical Modelling**, **35**(3), 1349-1353.
54. Mosheiov, G., 1994. Scheduling jobs under simple linear deterioration. **Computers & Operations Research**, **21**(6): 653-659.
55. Browne, S., Yechiali, U., 1990. Scheduling deteriorating jobs on a single processor. **Operations Research**, **38**(3): 495-498.
56. Lee, W. C., 2004. A note on deteriorating jobs and learning in single-machine scheduling problems. **International Journal of Business and Economics**, **3**(1): 83.
57. Wang, J. B., 2007. Single-machine scheduling problems with the effects of learning and deterioration. **Omega**, **35**(4): 397-402.
58. Mosheiov, G., 1991. V-shaped policies for scheduling deteriorating jobs. **Operations Research**, **39**(6): 979-991.
59. Wang, J. B., Xia, Z. Q., 2005. Scheduling jobs under decreasing linear deterioration. **Information Processing Letters**, **94**(2): 63-69.
60. Huang, X., 2010. Single machine scheduling with time-dependent deterioration and exponential learning effect. **Computers & Industrial Engineering**, **58**(1): 58-63.
61. Wu, C. C., Shiau, Y. R., Lee, W. C., 2008. Single-machine group scheduling problems with deterioration consideration. **Computers & Operations Research**, **35**(5): 1652-1659.
62. Yang, S. J., Lee, H. T., Guo, J. Y., 2013. Multiple common due dates assignment and scheduling problems with resource allocation and general position-dependent deterioration effect. **The International Journal of Advanced Manufacturing Technology**, **67**(1)-4: 181-188.
63. Wang, J. B., Wang, M. Z., 2012. Single-machine scheduling with nonlinear deterioration. **Optimization Letters**, **6**(1): 87-98.

64. Kononov, A., 2001. NP-hard cases in scheduling deteriorating jobs on dedicated machines. **Journal of the Operational Research Society**, **52**(6): 708-717.
65. Wang, D., Wang, J. B., 2010. Single-machine scheduling with simple linear deterioration to minimize earliness penalties. **The International Journal of Advanced Manufacturing Technology**, **46**(1): 285-290.
66. Alidaee, B., Womer, N. K., 1999. Scheduling with time dependent processing times: review and extensions. **Journal of the Operational Research Society**, **50**(7): 711-720.
67. Cai, J. Y., Cai, P., Zhu, Y., 1998. On a scheduling problem of time deteriorating jobs. **Journal of Complexity**, **14**(2): 190-209.
68. Guo, A. X., Wang, J. B., 2005. Single machine scheduling with deteriorating jobs under the group technology assumption. **Int J Pure Appl Math**, **18**(2): 225-231.
69. Xu, F., Guo, A., Wang, J., Shan, F., 2006. Single machine scheduling problem with linear deterioration under group technology. **International Journal of Pure and Applied Mathematics**, **28**(3), 401.
70. Wang, J. B., Guo, A. X., Shan, F., Jiang, B., Wang, L. Y., 2007. Single machine group scheduling under decreasing linear deterioration. **Journal of Applied Mathematics and Computing**, **24**(1-2), 283-293.
71. Hsieh, Y. C., Bricker, D. L., 1997. Scheduling linearly deteriorating jobs on multiple machines. **Computers & Industrial Engineering**, **32**(4), 727-734.
72. Kovalyov, M. Y., Kubiak, W., 1998. A fully polynomial approximation scheme for minimizing makespan of deteriorating jobs. **Journal of Heuristics**, **3**(4), 287-297.
73. Liu, C. H., 2011. Scheduling jobs with values exponentially deteriorating over time in a job shop environment. **In Proceedings of the International MultiConference of Engineers and Computer Scientists**, Vol. 2.
74. Mosheiov, G., 1996. Λ -shaped policies to schedule deteriorating jobs. **Journal of the Operational Research Society**, **47**(9), 1184-1191.

75. Mosheiov, G., 1998. Multi-machine scheduling with linear deterioration. **INFOR: Information Systems and Operational Research**, **36**(4), 205-214.
76. Ji, M., Cheng, T. E., 2008. Parallel-machine scheduling with simple linear deterioration to minimize total completion time. **European Journal of Operational Research**, **188**(2), 342-347.
77. Voutsinas, T. G., Pappis, C. P., 2002. Scheduling jobs with values exponentially deteriorating over time. **International Journal of Production Economics**, **79**(3), 163-169.
78. Wang, J. B., Ng, C. T., Cheng, T. E., 2008. Single-machine scheduling with deteriorating jobs under a series-parallel graph constraint. **Computers & Operations Research**, **35**(8), 2684-2693.
79. Liu, M., Wang, S., Chu, C., 2013. Scheduling deteriorating jobs with past-sequence-dependent delivery times. **International Journal of Production Economics**, **144**(2), 418-421.
80. De, P., Ghosh, J. B., Wells, C. E., 1992. On the minimization of completion time variance with a bicriteria extension. **Operations Research**, **40**(6), 1148-1155.
81. Kubiak, W., 1993. Completion time variance minimization on a single machine is difficult. **Operations Research Letters** **14**(1), 49-59.
82. Manna, D. K., Prasad, V. R., 1999. Bounds for the position of the smallest job in completion time variance minimization. **European Journal of Operational Research**, **114**(2), 411-419.
83. Su, L. H., Chou, F. D., 2001. Heuristic for scheduling in a bicriteria single machine problem. **Journal of the Chinese Institute of Industrial Engineers**, **18**(5), 39-46.
84. Su, L. H., Wang, H. M., 2017. Minimizing total absolute deviation of job completion times on a single machine with cleaning activities. **Computers & Industrial Engineering**, **103**, 242-249.

EKLER

EK 1. Öğrenme etkisi altında iş reddetmeli maksimum tamamlanma zamanı probleminin en küçüklenmesi problemi için önerilen matematiksel modelin Lingo paket program kodları (10 iş)

```

Model:
n=10;
alfa=0.8;

Sets:
hangiis/1..2/;
isler/1..10/;
pozisyonlar/1..10/;

link (hangiis,isler,pozisyonlar):
X;

link1 (isler):
P,
gama;
endsets

Data:
P=15 20 18 8 7 11 13 19 22 25;
gama=20 30 40 50 45 40 30 60 60 55;

enddata

Min=@sum(isler(j):
@SUM(pozisyonlar(r):((r^2)*(P(j))*r^(@log(alfa)/@log(2))*X(1,j,r)
))+gama(j)*X(2,j,r)));

@for(hangiis(i):
@for(pozisyonlar(r):
@sum(isler(j):X(i,j,r))<=1));
@for(isler(j):
@sum(hangiis(i):
@sum(pozisyonlar(r):X(i,j,r))=1);

@for(link:@bin(X));
end

```

EK 2. Öğrenme etkisi altında iş reddetmeli toplam tamamlanma zamanı probleminin en küçüklenmesi problemi için önerilen matematiksel modelin Lingo paket program kodları (10 iş)

```

Model:
n=10;
alfa=0.6;

Sets:
hangiis/1..2/;
isler/1..10/;
pozisyonlar/1..10/;

link (hangiis,isler,pozisyonlar):
X;

link1 (isler):
P,
gama;

endsets

Data:
P=17 12 20 24 17 6 16 6 25 24;
gama=26 58 30 30 37 52 41 22 32 50;

enddata

Min=@sum(isler(j):
@SUM(pozisyonlar(r):(r*(n-
j+1)*(P(j)*r^(@log(alfa)/@log(2))*X(1,j,r)))+gama(j)*X(2,j,r))));

@for(hangiis(i):
@for(pozisyonlar(r):
@sum(isler(j):X(i,j,r))<=1));
@for(isler(j):
@sum(hangiis(i):
@sum(pozisyonlar(r):X(i,j,r))=1));

@for(link:@bin(X));
end

```

EK 3. Öğrenme etkisi altında iş reddetmeli TZMFT probleminin en küçüklenmesi problemi için önerilen matematiksel modelin Lingo paket program kodları (10 iş)

```

Model:
n=10;
alfa=0.8;

Sets:
hangiis/1..2/;
isler/1..10/;
pozisyon/1..10/;

link (hangiis,isler,pozisyon):
X;

link1 (isler):
P,
gama;

endsets

Data:

P=17 12 20 24 17 6 16 6 25 24;
gama=26 58 30 30 37 52 41 22 32 50;

enddata

Min=@sum(isler(j):
@SUM(pozisyon(r):((r-1)*(n-
j+1)*(P(j)*r^(@log(alfa)/@log(2))*X(1,j,r))+gama(j)*X(2,j,r))));

@for(pozisyon(r):
@for(hangiis(i):
@sum(isler(j):X(i,j,r))<=1));

@for(isler(j):
@sum(hangiis(i):
@sum(pozisyon(r):X(i,j,r))=1);

@for(link:@bin(X));
end

```

EK 4. Öğrenme etkisi altında iş reddetmeli paralel makina maksimum tamamlanma zamanı probleminin en küçüklenmesi problemi için önerilen matematiksel modelin Lingo paket program kodları (10 iş)

```

Model:
n=10;
alfa=0.8;

Sets:
makine/1..3/;
isler/1..10/;
pozisyon/1..10/;

link (makine,isler,pozisyon):
X;
link1 (isler):
P,
gama;

endsets

Data:
P=17 12 20 24 17 6 16 6 25 24;
gama=26 58 30 30 37 52 41 22 32 50;

enddata

Min=@sum(isler (j):
@SUM(pozisyon(r):(r)*(P(j)*r^(@log(alfa)/@log(2))*X(1,j,r)))+
@SUM(pozisyon(r):(r)*(P(j)*r^(@log(alfa)/@log(2))*X(2,j,r)))+
@sum(pozisyon(r):(gama(j)*X(3,j,r))));

@for(makine(i):
@for(pozisyon(r):
@sum(isler(j):X(i,j,r))<=1));

@for(isler(j):
@sum(makine(i):
@sum(pozisyon(r):X(i,j,r)))=1);

@for(link:@bin(X));
end

```


EK 5. Bozulma ve öğrenme etkisi altında iş reddetmeli maksimum tamamlanma zamanı probleminin en küçüklenmesi problemi için önerilen matematiksel modelin Lingo paket program kodları (10 iş)

```

Model:
n=10;
alfa=0.8;
boz=0.1;
Sets:
hangiis/1..2/;
isler/1..10/;
pozisyonlar/1..10/;

link (hangiis,isler,pozisyonlar):
X;

link1 (isler):
P,
gama;

link2(pozisyonlar):
T;
endsets

Data:
P=17 12 20 24 17 6 16 6 25 24;
gama=26 58 30 30 37 52 41 22 32 50;

enddata

Min=@sum(isler(j):
@SUM(pozisyonlar(r):((r^2)*((P(j)+boz*T(r))*r^(@log(alfa)/@log(2)
))*X(1,j,r)))+gama(j)*X(2,j,r)));

@for(hangiis(i):
@for(pozisyonlar(r):
@sum(isler(j):X(i,j,r))<=1));
@for(isler(j):
@sum(hangiis(i):
@sum(pozisyonlar(r):X(i,j,r))=1);

T(1)=0;
@for(pozisyonlar(r)|r#GT#1:
@sum(isler(j):@sum(pozisyonlar(k)|k#LT#r:((P(j)+boz*T(k))*k^(@log(alfa)
)/@log(2))*X(1,j,k))))=T(r));

@for(link:@bin(X));
end

```

EK 6. Bozulma ve öğrenme etkisi altında iş reddetmeli toplam tamamlanma zamanı probleminin en küçüklenmesi problemi için önerilen matematiksel modelin Lingo paket program kodları (10 iş)

```

Model:
n=10;
alfa=0.8;
boz=0.1;
Sets:
hangiis/1..2/;
isler/1..10/;
pozisyonlar/1..10/;
link (hangiis,isler,pozisyonlar):
X;
link1 (isler):
P,
gama;

link2(pozisyonlar):
T;
endsets
Data:
    P=17 12 20 24 17 6 16 5 1 24;
    gama=26 58 30 30 37 52 41 22 32 50;

enddata

Min=@sum(isler(j):
    @SUM(pozisyonlar(r):((r/2)*(n-
r+1)*((P(j)+boz*T(r))*r^(@log(alfa)/@log(2)))*X(1,j,r))+gama(j)*X(2,j,
r))));

@for(hangiis(i):
    @for(pozisyonlar(r):
        @sum(isler(j):X(i,j,r))<=1));
@for(isler(j):
    @sum(hangiis(i):
        @sum(pozisyonlar(r):X(i,j,r))=1);
T(1)=0;

    @for(pozisyonlar(r)|r#GT#1:
@sum(isler(j):@sum(pozisyonlar(k)|k#LT#r:((P(j)+boz*T(k))*k^(@log(alfa)
)/@log(2))*X(1,j,k)))=T(r));

@for(link:@bin(X));
end

```

EK 7. Bozulma ve öğrenme etkisi altında iş reddetmeli TZMFT probleminin en küçüklenmesi problemi için önerilen matematiksel modelin Lingo paket program kodları (10 iş)

```

Model:
n=10;
alfa=0.8;
boz=0.1;

Sets:
hangiis/1..2/;
isler/1..10/;
pozisyonlar/1..10/;

link (hangiis,isler,pozisyonlar):
X;

link1 (isler):
P,
gama;

link2(pozisyonlar):
T;

endsets

Data:
P=17 12 20 24 17 6 16 6 25 24;
gama=26 58 30 30 37 52 41 22 32 50;

enddata

Min=@sum(isler(j):
    @SUM(pozisyonlar(r):((r-1)*(n-
j+1)*(P(j)+(boz*T(r)))*r^(@log(alfa)/@log(2))*X(1,j,r))+gama(j)*X(2,j,
r)));

@for(pozisyonlar(r):
    @for(hangiis(i):
        @sum(isler(j):X(i,j,r))<=1));

@for(isler(j):
    @sum(hangiis(i):
        @sum(pozisyonlar(r):X(i,j,r)))=1);
T(1)=0;

@for(pozisyonlar(r)|r#GT#1:
@sum(isler(j):@sum(pozisyonlar(k)|k#LT#r:((P(j)+boz*T(k))*k^(@log(alfa)
)/@log(2))*X(1,j,k)))=T(r));

@for(link:@bin(X));
end

```

EK 8. Bozulma ve öğrenme etkisi altında iş reddetmeli paralel makina maksimum tamamlanma zamanı probleminin en küçüklenmesi problemi için önerilen matematiksel modelin Lingo paket program kodları (10 iş)

```

Model:
n=10;
alfa=0.8;
boz=0.1;
Sets:
makine/1..3/;
isler/1..10/;
pozisyonlar/1..10/;
link (makine,isler,pozisyonlar):
X;
link1 (isler):
P,
gama;
link2 (pozisyonlar):
T,
T1;
endsets
Data:
    P=17 12 20 24 17 6 16 6 25 24;
    gama=26 58 30 30 37 52 41 22 32 50;
enddata
Min=@sum(isler (j):
    @SUM(pozisyonlar (r): (r) * (P(j)+boz*T(r)) *r^(@log(alfa)/@log(2)) *X
(1,j,r))+
    @SUM(pozisyonlar (r): (r) * (P(j)+boz*T1(r)) *r^(@log(alfa)/@log(2)) *
X(2,j,r))+
    @sum(pozisyonlar (r): (gama(j)*X(3,j,r))));

@for (makine (i):
    @for (pozisyonlar (r):
        @sum (isler (j): X(i,j,r)) <= 1));

@for (isler (j):
    @sum (makine (i):
        @sum (pozisyonlar (r): X(i,j,r)) = 1);
T(1)=0;

    @for (pozisyonlar (r) | r#GT#1:
@sum (isler (j): @sum (pozisyonlar (k) | k#LT#r: ((P(j)+boz*T(k)) *k^(@log(alfa)
)/@log(2)) *X(1,j,k)))) = T(r));

T1(1)=0;

    @for (pozisyonlar (r) | r#GT#1:
@sum (isler (j): @sum (pozisyonlar (k) | k#LT#r: ((P(j)+boz*T1(k)) *k^(@log(alf
a)/@log(2)) *X(2,j,k)))) = T1(r));
@for (link:@bin(X));
End

```

EK 9. Tüm problemler için C# kodları

```

namespace LPTApp
{
    public class Helper
    {
        private readonly object ThreadLock = new object();
        private double alpha;
        private Random rand;
        private FonkType ActiveFonkType;
        private double[] Alphas;
        public List<MyJob> Jobs { get; internal set; } = new List<MyJob>();
        private TC MinTC = new TC();
        private double decay;

        public void ReadJobsFromFile(string filePath)
        {
            Jobs.Clear();
            MinTC = new TC();
            using (StreamReader file = File.OpenText(filePath))
            {
                JsonSerializer serializer = new JsonSerializer
                {
                    MissingMemberHandling = MissingMemberHandling.Ignore
                };

                Jobs = (List<MyJob>)serializer.Deserialize(file, typeof(List<MyJob>));
            }

            for (int i = 0; i < Jobs.Count; i++)
            {
                Jobs[i].Name = $"job{i + 1}";
                Jobs[i].Position = i + 1;
            }
        }

        internal void SetAlpha(double training)
        {
            alpha = Math.Round(Math.Log(training / 100) / Math.Log(2), 4);
            Alphas = new double[Jobs.Count + 1];
            Alphas[0] = 0d;
            for (int i = 1; i < Alphas.Count(); i++)
            {
                Alphas[i] = Math.Round(Math.Pow(i, alpha), 4);
            }
        }

        internal void SetGama(double decay)
        {
            this.decay = decay / 100d;
        }

        public string RunFunctions(FonkType fonkType)
        {
            rand = new Random(Guid.NewGuid().GetHashCode());
            MinTC.AcceptedJobs = string.Empty;
            MinTC.DeniedJobs = string.Empty;
            MinTC.Value = double.MaxValue;

            ActiveFonkType = fonkType;
        }
    }
}

```

```

string result = "";
switch (ActiveFonkType)
{
    case FonkType.LPT:
        Jobs = Jobs.OrderByDescending(a => a.P).ToList();
        result = LptSpt();
        break;
    case FonkType.SPT:
        Jobs = Jobs.OrderBy(a => a.P).ToList();
        result = LptSpt();
        break;
    case FonkType.TADC:
        Jobs = Jobs.OrderBy(a => a.P).ToList();
        result = Tadc();
        break;
    case FonkType.ParallelLPT:
        Jobs = Jobs.OrderByDescending(a => a.P).ToList();
        result = LptSpt();
        break;
    case FonkType.LPT_DECAY:
        Jobs = Jobs.OrderByDescending(a => a.P).ToList();
        result = LptSpt();
        break;
    case FonkType.SPT_DECAY:
        Jobs = Jobs.OrderBy(a => a.P).ToList();
        result = LptSpt();
        break;
    case FonkType.TADC_Decay:
        Jobs = Jobs.OrderBy(a => a.P).ToList();
        result = Tadc();
        break;
    case FonkType.ParallelLPT_Decay:
        Jobs = Jobs.OrderBy(a => a.P).ToList();
        result = LptSpt();
        break;
}
return result;
}

private string Tadc()
{
    StringBuilder sb = new StringBuilder();
    sb.AppendLine($"Process Started for {Jobs.Count} job(s) at
{DateTime.Now.ToString("F")}");
    sb.AppendLine("".PadLeft(30, '='));
    for (int i = 0; i < Jobs.Count; i++)
    {
        double tc = 0;
        List<MyJob> acc = Jobs.Take(i).Reverse().ToList();
        double[] Ts = acc.TValues(decay, alpha);
        if (ActiveFonkType.Equals(FonkType.TADC_Decay))
        {
            acc.ForEach(a => tc += Calculate(acc.IndexOf(a) + 1, a.P + (Ts[i] *
decay), a.Position));
        }
        else
        {
            acc.ForEach(a => tc += Calculate(acc.IndexOf(a) + 1, a.P, a.Position));
        }
        List<MyJob> den = Jobs.Skip(i).ToList();
        tc += den.Sum(a => a.G);
    }
}

```

```

        SetTC(acc.GetJobNames(), den.GetJobNames(), tc);
    }
    sb.AppendLine($"Minimum TC:\t{MinTC.Value}");
    sb.AppendLine($"Accepted Jobs:\t{MinTC.AcceptedJobs}");
    sb.AppendLine($"Denied Jobs:\t{MinTC.DeniedJobs}");
    sb.AppendLine($"".PadLeft(30, '='));
    sb.AppendLine($"Process Completed at {DateTime.Now.ToString("F")}");
    return sb.ToString();
}

public string LptSpt()
{
    StringBuilder sb = new StringBuilder();
    sb.AppendLine($"Process Started for {Jobs.Count} job(s) at
{DateTime.Now.ToString("F")}");
    sb.AppendLine($"".PadLeft(30, '='));

    Parallel.For(0, Jobs.Count, ix =>
    {
        GetTCs(ix);
    });
    //for (int i = 0; i < Jobs.Count; i++)
    //{
    //    GetTCs(i);
    //}

    GetDeniedTCs();
    sb.AppendLine($"Minimum TC:\t{MinTC.Value}");
    sb.AppendLine($"Accepted Jobs:\t{MinTC.AcceptedJobs}");
    sb.AppendLine($"Denied Jobs:\t{MinTC.DeniedJobs}");
    sb.AppendLine($"".PadLeft(30, '='));
    sb.AppendLine($"Process Completed at {DateTime.Now.ToString("F")}");
    return sb.ToString();
}

private void GetTCs(int ix)
{
    Debug.WriteLine($" {ix} started at {DateTime.Now}");
    MyJob[] accepted = new MyJob[ix];
    double tc = Jobs.Sum(a => a.G);

    switch (ActiveFonkType)
    {
        case FonkType.LPT:
            CombinationLPT(accepted, 0, Jobs.Count - 1, 0, ix, tc);
            break;
        case FonkType.SPT:
            CombinationSPT(accepted, 0, Jobs.Count - 1, 0, ix, tc);
            break;
        case FonkType.ParallelLPT:
            ParallelCombination(accepted, 0, Jobs.Count - 1, 0, ix, tc);
            break;
        case FonkType.LPT_DECAY:
            CombinationLPT_decay(accepted, 0, Jobs.Count - 1, 0, ix, tc);
            break;
        case FonkType.SPT_DECAY:
            CombinationSPT_decay(accepted, 0, Jobs.Count - 1, 0, ix, tc);
            //Combination(accepted, 0, Jobs.Count - 1, 0, ix, tc);
            break;
        case FonkType.ParallelLPT_Decay:
            ParallelCombinationDecay(accepted, 0, Jobs.Count - 1, 0, ix, tc);
    }
}

```

```

        break;
    default:
        break;
    }
    Debug.WriteLine($"{ix} completed at {DateTime.Now}");
}

private void Combination(MyJob[] accepted, int start, int end, int index, int
r, double tc)
{
    if (index == r)
    {
        tc -= accepted.Sum(a => a.G);
        for (int i = 0; i < accepted.Length; i++)
        {
            if (ActiveFonkType.Equals(FonkType.LPT_DECAY) ||
ActiveFonkType.Equals(FonkType.SPT_DECAY))
            {
                double[] Ts = accepted.TValues(decay, alpha);
                tc += Calculate(i + 1, accepted[i].P + (Ts[i] * decay),
accepted[i].Position);
            }
            else
            {
                tc += Calculate(i + 1, accepted[i].P, accepted[i].Position);
            }
        }
        SetTC(accepted, tc);
        return;
    }

    for (int i = start; i <= end && end - i + 1 >= r - index; i++)
    {
        accepted[index] = Jobs[i];
        Combination(accepted, i + 1, end, index + 1, r, tc);
    }
}

private void ParallelCombination(MyJob[] accepted, int start, int end, int
index, int r, double tc)
{
    if (index == r)
    {
        string empty = string.Empty;
        tc -= accepted.Sum(a => a.G);
        int ix1 = 1;
        int ix2 = 1;
        string str1 = "1m: ";
        string str2 = "2m: ";
        for (int i = 0; i < accepted.Length; ++i)
        {
            if (rand.Next(150) < 75)
            {
                str1 = str1 + accepted[i].Name + ",";
                //tc += Calculate(index1, accepted[i].P, accepted[i].Position);
                tc += ix1 * accepted[i].P * Alphas[ix1]; ;
                ++ix1;
            }
            else
            {
                //tc += Calculate(index2, accepted[i].P, accepted[i].Position);
                tc += ix2 * accepted[i].P * Alphas[ix2];
            }
        }
    }
}

```



```

        str2 = str2 + accepted[i].Name + ",";
        ++ix2;
    }
}
SetTC(str1 + " --- " + str2, string.Empty, tc);
}
else
{
    for (int index1 = start; index1 <= end && end - index1 + 1 >= r - index;
++index1)
    {
        accepted[index] = Jobs[index1];
        ParallelCombination(accepted, index1 + 1, end, index + 1, r, tc);
    }
}
private void ParallelCombinationDecay(MyJob[] accepted, int start, int end,
int index, int r, double tc)
{
    if (index == r)
    {
        string empty = string.Empty;
        tc -= accepted.Sum(a => a.G);
        int ix1 = 1;
        int ix2 = 1;
        string str1 = "1m: ";
        string str2 = "2m: ";
        List<double> t1 = new List<double> { 0d };
        List<double> t2 = new List<double> { 0d };
        for (int i = 1; i < accepted.Length; ++i)
        {
            if (rand.Next(150) < 75)
            {
                t1.Add(((accepted[i - 1].P + Math.Round(decay * t1[ix1 - 1], 2)) *
Alphas[ix1]) + t1[ix1 - 1]);

                //tc += Calculate(ix1, accepted[i].P + decay * t1[ix1 - 1],
accepted[i].Position);
                tc += ix1 * (accepted[i - 1].P + decay * t1[ix1 - 1]) * Alphas[ix1];
                str1 = str1 + accepted[i - 1].Name + ",";
                ++ix1;
            }
            else
            {
                t2.Add(((accepted[i - 1].P + Math.Round(decay * t2[ix2 - 1], 2)) *
Alphas[ix2]) + t2[ix2 - 1]);

                //tc += Calculate(ix2, accepted[i].P + decay * t2[ix2 - 1],
accepted[i].Position);
                tc += (ix2 * (accepted[i - 1].P + decay * t2[ix2 - 1]) *
Alphas[ix2]);

                str2 = str2 + accepted[i - 1].Name + ",";
                ++ix2;
            }
        }
        SetTC(str1 + " --- " + str2, string.Empty, tc);
    }
}
else
{

```

```

        for (int index1 = start; index1 <= end && end - index1 + 1 >= r - index;
++index1)
        {
            accepted[index] = Jobs[index1];
            ParallelCombinationDecay(accepted, index1 + 1, end, index + 1, r, tc);
        }
    }
}

private void CombinationLPT(MyJob[] accepted, int start, int end, int index,
int r, double tc)
{
    if (index == r)
    {
        tc -= accepted.Sum(a => a.G);
        for (int i = 0; i < accepted.Length; i++)
        {
            tc += Math.Pow(i + 1, 2) * accepted[i].P * Alphas[i + 1];
            //acc += accepted[i].Name + ",";
        }

        SetTC(accepted,tc);
        return;
    }

    for (int i = start; i <= end && end - i + 1 >= r - index; i++)
    {
        accepted[index] = Jobs[i];
        CombinationLPT(accepted, i + 1, end, index + 1, r, tc);
    }
}

private void CombinationLPT_decay(MyJob[] accepted, int start, int end, int
index, int r, double tc)
{
    if (index == r)
    {
        tc -= accepted.Sum(a => a.G);
        double[] Ts = accepted.TValues(decay, alpha);
        //if (tc == 268)
        //{
        //    Debug.WriteLine(tc);
        //}
        for (int i = 0; i < accepted.Length; i++)
        {
            tc += (Math.Pow(i + 1, 2) * (accepted[i].P + (Ts[i] * decay)) *
Alphas[i + 1]);
        }

        SetTC(accepted, tc);
        return;
    }

    for (int i = start; i <= end && end - i + 1 >= r - index; i++)
    {
        accepted[index] = Jobs[i];
        CombinationLPT_decay(accepted, i + 1, end, index + 1, r, tc);
    }
}

private void CombinationSPT(MyJob[] accepted, int start, int end, int index,
int r, double tc)
{

```

```

        if (index == r)
        {
            tc -= accepted.Sum(a => a.G);
            for (int i = 0; i < accepted.Length; i++)
            {
                tc += (i + 1) * (Jobs.Count - accepted[i].Position + 1) * accepted[i].P
* Alphas[i + 1];
            }

            SetTC(accepted, tc);
            return;
        }

        for (int i = start; i <= end && end - i + 1 >= r - index; i++)
        {
            accepted[index] = Jobs[i];
            CombinationSPT(accepted, i + 1, end, index + 1, r, tc);
        }
    }
    private void CombinationSPT_decay(MyJob[] accepted, int start, int end, int
index, int r, double tc)
    {
        if (index == r)
        {
            tc -= accepted.Sum(a => a.G);
            double[] Ts = accepted.TValues(decay, alpha);
            for (int i = 0; i < accepted.Length; i++)
            {
                tc += (double)(i + 1) / 2 * (Jobs.Count - accepted[i].Position + 1) *
(accepted[i].P + (Ts[i] * decay)) * Alphas[i + 1];
            }

            SetTC(accepted, tc);
            return;
        }

        for (int i = start; i <= end && end - i + 1 >= r - index; i++)
        {
            accepted[index] = Jobs[i];
            CombinationSPT_decay(accepted, i + 1, end, index + 1, r, tc);
        }
    }

    private double Calculate(double index, double P, int position)
    {
        double res = 0d;
        switch (ActiveFonkType)
        {
            //case FonkType.LPT:
            //    res = Math.Pow(index, 2) * P * Math.Pow(index, alpha);
            //    break;
            //case FonkType.SPT:
            //    res = index * (Jobs.Count - position + 1) * P * Math.Pow(index,
alpha);
            //    break;
            case FonkType.TADC:
                res = Math.Pow(index, 2) * P * Math.Pow(index, alpha);
                break;
            //case FonkType.ParallelLPT:
            //    res = index * P * Math.Pow(index, alpha);
            //    break;
        }
    }

```

```

        //case FonkType.LPT_DECAY:
        // res = Math.Pow(index, 2) * P * Math.Pow(index, alpha);
        // break;
        //case FonkType.SPT_DECAY:
        // res = index / 2 * (Jobs.Count - position + 1) * P * Math.Pow(index,
alpha);
        // break;
        case FonkType.TADC_Decay:
            res = (index - 1) * (Jobs.Count - position + 1) * P * Math.Pow(index,
alpha);
            break;
        //case FonkType.ParallelLPT_Decay:
        // res = index * P * Math.Pow(index, alpha);
        // break;
    }
    return res;
}
private void SetTC(string acc, string den, double tc)
{
    lock (ThreadLock)
    {
        if (MinTC.Value > tc)
        {
            MinTC.AcceptedJobs = acc;
            MinTC.DeniedJobs = den;
            MinTC.Value = tc;
        }
    }
}
private void SetTC(MyJob[] jobs, double tc)
{
    lock (ThreadLock)
    {
        if (MinTC.Value > tc)
        {
            MinTC.AcceptedJobs = "";
            for (int i = 0; i < jobs.Count(); i++)
            {
                MinTC.AcceptedJobs += jobs[i].Name + ",";
            }
            MinTC.Value = tc;
        }
    }
}

private void GetDeniedTCs()
{
    string str = MinTC.AcceptedJobs;
    if (str.Contains("1m: "))
    {
        str = str.Replace("1m: ", "");
        str = str.Replace("2m: ", "");
        str = str.Replace(" --- ", "");
    }
    var list = str.Split(',');
    var second = Jobs.Where(a => !list.Contains(a.Name)).ToList();
    second.ForEach(a => MinTC.DeniedJobs += a.Name + ",");
}
}

public class MyJob

```

```
{  
    public string Name { get; set; }  
    public double P { get; set; }  
    public double G { get; set; }  
    public int Position { get; set; }  
}  
public class TC  
{  
    public double Value { get; set; }  
    public string AcceptedJobs { get; set; }  
    public string DeniedJobs { get; set; }  
}  
}
```



EK 10. Class tanımlamaları

```

namespace LPTApp
{
    public static class Extensions
    {
        //public static long TC(this List<MyJob> myJobs)
        //{
        //    return myJobs.Sum(a => a.P);
        //}
        public static string ToBinaryString(this BigInteger bigint)
        {
            var bytes = bigint.ToByteArray();
            var idx = bytes.Length - 1;

            // Create a StringBuilder having appropriate capacity.
            var base2 = new StringBuilder(bytes.Length * 8);

            // Convert first byte to binary.
            var binary = Convert.ToString(bytes[idx], 2);

            // Ensure leading zero exists if value is positive.
            //if (binary[0] != '0' && bigint.Sign == 1)
            //{
            //    base2.Append('0');
            //}

            // Append binary string to StringBuilder.
            if (binary[0] != '0')
            {
                base2.Append(binary);
            }

            // Convert remaining bytes adding leading zeros.
            for (idx--; idx >= 0; idx--)
            {
                base2.Append(Convert.ToString(bytes[idx], 2).PadLeft(8, '0'));
            }

            return base2.ToString();
        }
        public static string GetJobNames(this List<MyJob> myJobs)
        {
            string res = "";
            myJobs.ForEach(a => res += a.Name + ",");
            return res;
        }
        public static double[] TValues(this MyJob[] myJobs, double decayRate, double
alpha)
        {
            int c = myJobs.Count() + 1;
            var list = new double[c];
            list[0] = 0;
            for (int i = 1; i < c; i++)
            {
                double a = Math.Round(decayRate * list[i - 1], 2);
                list[i] = (myJobs[i - 1].P + a) * Math.Round(Math.Pow(i, alpha), 5);
                list[i] += list[i - 1];
            }
            return list;
        }
    }
}

```

```

    }
    public static double[] TValues(this List<MyJob> myJobs, double decayRate,
double alpha)
    {
        int c = myJobs.Count() + 1;
        var list = new double[c];
        list[0] = 0;
        for (int i = 1; i < c; i++)
        {
            double a = Math.Round(decayRate * list[i - 1], 2);
            list[i] = (myJobs[i - 1].P + a) * Math.Round(Math.Pow(i, alpha), 5);
            list[i] += list[i - 1];
        }
        return list;
    }
}
}
}

```

```

namespace LPTApp
{
    public static class PermutationsAndCombinations
    {
        public static long CombinationN(int n, int r)
        {
            // naive: return Factorial(n) / (Factorial(r) * Factorial(n - r));
            return PermutationN(n, r) / Factorial(r);
        }

        public static long PermutationN(int n, int r)
        {
            // naive: return Factorial(n) / Factorial(n - r);
            return FactorialDivision(n, n - r);
        }

        private static long FactorialDivision(int topFactorial, int divisorFactorial)
        {
            long result = 1;
            for (int i = topFactorial; i > divisorFactorial; i--)
                result *= i;
            return result;
        }

        private static long Factorial(int i)
        {
            if (i <= 1)
                return 1;
            return i * Factorial(i - 1);
        }
    }
}
}
}

```

```

namespace LPTApp
{
    public partial class Form1 : Form
    {
        public Helper Helper { get; set; }
        public Form1()
        {
            Helper = new Helper();
            InitializeComponent();
        }
    }
}

```

```

}

private void btn_fileSelect_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = openFileDialog1.ShowDialog();
    if (dialogResult == DialogResult.OK)
    {
        string file = openFileDialog1.FileName;
        try
        {
            Helper.ReadJobsFromFile(file);
            lbl_jobCount.Text = $"Job Count: {Helper.Jobs.Count}";
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

private void btn_lpt_Click(object sender, EventArgs e)
{
    Execute(FonkType.LPT);
}

private void btn_spt_Click(object sender, EventArgs e)
{
    Execute(FonkType.SPT);
}

private void btn_tadc_Click(object sender, EventArgs e)
{
    Execute(FonkType.TADC);
}

private void btn_parallelLpt_Click(object sender, EventArgs e)
{
    Execute(FonkType.ParallelLPT);
}

private void btn_lptDecay_Click(object sender, EventArgs e)
{
    Execute(FonkType.LPT_DECAY);
}

private void btn_sptDecay_Click(object sender, EventArgs e)
{
    Execute(FonkType.SPT_DECAY);
    //MyJob[] a = new MyJob[]
    //{
    //    new MyJob{P=12},
    //    new MyJob{P=6},
    //};
    //var m = a.TValues(0.1d, -0.3219d);
    //MessageBox.Show(m[0].ToString() + " " + m[1].ToString() + " " +
m[2].ToString());
}

private void btn_tadcDecay_Click(object sender, EventArgs e)
{
    Execute(FonkType.TADC_Decay);
}

```



```

    }

    private void Execute(FonkType fonkType)
    {
        var train = tb_training.Text == string.Empty ? 80 :
Convert.ToInt16(tb_training.Text);
        var decay = tb_decay.Text == string.Empty ? 10 :
Convert.ToInt16(tb_decay.Text);
        if (Helper.Jobs.Count < 1)
        {
            MessageBox.Show("Please Select File.", "File Error!");
        }
        else if (train < 0 || train > 100)
        {
            MessageBox.Show("The Training rate should be between 0 and 100.",
"Training Rate Error!");
        }
        else if (decay < 0 || decay > 100)
        {
            MessageBox.Show("The Decay rate should be between 0 and 100.", "Decay
Rate Error!");
        }
        else
        {
            Helper.SetAlpha(train);
            Helper.SetGama(decay);
            MessageBox.Show(Helper.RunFunctions(fonkType), $"{fonkType} - Result for
{lbl_jobCount.Text} job(s) with {tb_training.Text}% Training rate.");
            //MessageBox.Show(Helper.AlternativeRun(FonkType.LPT));
        }
    }

    private void btn_parallelLptDecay_Click(object sender, EventArgs e)
    {
        Execute(FonkType.ParallelLPT_Decay);
    }
}
}

```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Berrin ATALAY
Uyruğu: Türkiye (T.C)
Doğum Tarihi ve Yeri: 29.01.1987 - Kayseri
Medeni Durum: Evli
e-mail: berrinucarkus@erciyes.edu.tr
Yazışma Adresi: Erciyes Üniversitesi Endüstri Mühendisliği Bölümü Kat:3

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi, Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Ana Bilim Dalı	2014
Lisans	Erciyes Üniversitesi, Endüstri Müh. Bölümü	2009
Lise	Sümer Lisesi, Kayseri	2004

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2014-Halen	Erciyes Üniversitesi, Endüstri Mühendisliği Bölümü, Araştırma Görevlisi	5 Yıl
2010-2014	Yataş A. Ş.	3 Yıl

YABANCI DİL

İngilizce

PROJELER

TOKSARI M. D. (Yürütücü) , UÇARKUŞ B., Öğrenme etkisi altında iş bağımlı geçmiş sıra tabanlı teslimat süreli çizelgeleme problemleri, BAP Projesi, 2014-2015.

YAYINLAR

Toksarı, M. D., Uçarkuş, B., Solution Approaches for Some Scheduling Problems with Learning Effect and Job Dependent Delivery Times, International Journal of Social, Behavioral, Educational, Economic and Management Engineering, no.10, ss.3083-3086, 2015.

Canıyılmaz, E., Uçarkuş, B., Karaman, S., Optimization of Formulation Ingredients and Aging Time for Ice Cream Processing Using Combined Design Approach, Journal Of Food Processing And Preservation, cilt.40, ss.1325-1338, 2016.

Toksarı, M. D., Atalay, B., Pozisyon Tabanlı Öğrenme ve Doğrusal Olmayan Bozulma Etkisi Altında İş Reddetmeli Tek Makine Çizelgeleme Problemi, Konya Mühendislik Bilimleri Dergisi, cilt.7, ss.797-805, 2019.

Toksarı, M. D., Atalay, B., Some Scheduling Problems with Job Rejection under Learning and Deterioration Effects (hakem değerlendirmesinde).

Toksarı, M. D., Atalay, B., Job Rejection Scheduling Problems with Deterioration and Learning Effects (hakem değerlendirmesinde).

Toksarı, M. D., Atalay, B., Some Scheduling Problems with Job Rejection and a Learning Effect (hakem değerlendirmesinde).

KONFERANSLAR

Toksarı, M. D., Uçarkuş, B., Scheduling Problems Under Some Special Things (Job Rejection, Past Sequence Dependence Time And Learning And Deteriorating Effects), 23rd International Conference on Multiple Criteria Decision Making MCDM 2015, Hamburg, Almanya, 2 - 07 August 2015, ss.1.

Toksarı, M. D., Uçarkuş, B., Öğrenme Etkisi ve İşlerin Reddedilmesi Durumunda İşlerin Çizelgelenmesi, IMSEC 2016, Adana, Türkiye, 26 - 28 October 2016, ss.1-5.

Toksarı, M. D., Atalay, B., Öğrenme ve Bozulma Etkisi Altında İşlerin Reddedilerek Çizelgelenmesi Problemi, IMSEC 2017, Adana, Türkiye, 25 - 27 October 2017, ss.586-588.

Toksarı, M. D., Atalay, B., Öğrenme Etkisi Altında İş Reddetmeli Tek Makine Çizelgeleme Problemlerinde Toplam Tamamlanma Zamanı En Küçüklenmesi, IV. International Scientific and Vocational Studies Congress-Engineering, Ankara, Türkiye, 7 - 10 November 2019, ss.1-4.

Öğrenme Etkisi Altında İş Reddetmeli Tek Makine Çizelgeleme Problemlerinde Maksimum Tamamlanma Zamanı En Küçüklenmesi, IV. International Scientific and Vocational Studies Congress-Engineering, Ankara, Türkiye, 7 - 10 November 2019, ss.1-3.

