

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

**ÖĞRENME VE BOZULMA ETKİLERİ ALTINDA
DEMONTAJ HATTI Dengeleme Problemi**

**Hazırlayan
Halime SOMTÜRK**

**Danışman
Prof. Dr. Mehmet Duran TOKSARI**

Doktora Tezi

**Ocak 2024
KAYSERİ**

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

**ÖĞRENME VE BOZULMA ETKİLERİ ALTINDA
DEMONTAJ HATTI DENGELEME PROBLEMİ
(Doktora Tezi)**

**Hazırlayan
Halime SOMTÜRK**

**Danışman
Prof. Dr. Mehmet Duran TOKSARI**

**Ocak 2024
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.

Halime SÖMTÜRK

İmza

“Öğrenme ve Bozulma Etkileri Altında Demontaj Hattı Dengeleme Problemi” adlı Doktora tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ ne uygun olarak hazırlanmıştır.

Hazırlayan

Halime SOMTÜRK

İmza

Danışman

Prof. Dr. Mehmet Duran TOKSARI

İmza

Endüstri Mühendisliği ABD Başkanı

İmza

TEŞEKKÜR

Çalışmalarım boyunca farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan, bana güvenen, yardımlarını esirgemeyen değerli hocam Sayın Prof. Dr. M. Duran TOKSARI'ya teşekkürü borç bilirim.

Bu süreçte desteğini esirgemeyen, sabır ve anlayışla yanımda olan değerli eşim Çağrı SOMTÜRK'e ve küçük bedenleriyle beni anlamaya çalışan hayatıma anlam katan oğlum Emir ve kızım Meva'ya sonsuz teşekkür ederim. Dualarıyla her zaman manevi güç katan, hayatımın her aşamasında bana güvenerek her konuda destekleyen çok kıymetli, canım aileme minnettarım.

Geç tanışmış olduğum ama sonrasında dostluğunu ve manevi desteğini hiçbir zaman esirgemeyen kıymetli arkadaşım Arş. Gör. Dr. Zeynep ÜNSAL ASLAN'a teşekkür ederim.

Halime SOMTÜRK

Ocak 2024, KAYSERİ

ÖĞRENME VE BOZULMA ETKİLERİ ALTINDA DEMONTAJ HATTI DENGELEME PROBLEMİ

Halime SOMTÜRK

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Doktora Tezi, Ocak 2024
Danışman: Prof. Dr. Mehmet Duran TOKSARI

ÖZET

Son yıllarda, ürün geri kazanımı veya yeniden üretimi, çevre bilincinin artmasıyla, ekonomik nedenlerle ve hükümetlerin almış oldukları sıkı çevre politikaları ile büyük ilgi görmektedir. Ömrü tükenmiş bir ürünün geri kazanılması aşamasında en önemli işlem, ürünün parçalarına ayrılması olarak tanımlanan demontajdır. Bu çalışmada, demontaj hattı dengeleme probleminde pozisyon tabanlı ve logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile bozulma etkisi ele alınmıştır. İşler, çalışanlar tarafından tekrarlandıkça öğrenilmekte, öğrenilen iş daha kısa sürede yapılabilmektedir. Pozisyon tabanlı öğrenme, işlerin işlem sürelerinin istasyonda atandıkları sıraya bağlı olarak güncellendiği öğrenme türüdür. Logaritmik işlem süreleri toplamı tabanlı öğrenmede ise bir istasyonda herhangi bir sıraya atanmış olan işin işlem süresi, kendisinden önceki sıralara atanmış işlerin işlem sürelerinin logaritmasının toplamına bağlı olarak azalmaktadır. Bir işin başlangıç zamanının gecikmesine veya ertelenmesine sebep olan her türlü olay bozulma olarak adlandırılmaktadır. Bu çalışmada öğrenme etkileri ve bozulma etkileri altında demontaj hattı dengeleme problemleri ele alınmıştır. Çalışmada, amaç fonksiyonu, istasyon sayısının minimizasyonudur. Küçük boyutlu problemlerin çözümü için Lingo 11 paket programı kullanılmıştır. Orta ve büyük boyutlu problemler için Genetik Algoritma (GA), Hibrit Genetik Algoritma (HGA), Parçacık Sürüsü Optimizasyonu (PSO) ve PSO-GA hibrit yaklaşımları kullanılmıştır. Bu yaklaşımlar farklı öğrenme oranları ve bozulma oranlarının kombinasyonu için test problemleri çözülmüş ve elde edilen sonuçlar karşılaştırılmıştır.

Anahtar Kelimeler: Demontaj hattı dengeleme, öğrenme etkisi, iş bozulma etkisi, genetik algoritma, yerel arama metodu, parçacık sürüsü algoritması

DISASSEMBLY LINE BALANCING PROBLEM UNDER THE EFFECTS OF LEARNING AND DETERIORATION

Halime SOMTÜRK

Erciyes University, Graduate School of Natural and Applied Sciences

PhD Thesis, January 2024

Supervisor: Prof. Dr. Mehmet Duran TOKSARI

ABSTRACT

In recent years, product recovery or remanufacturing has attracted great attention with increasing environmental awareness, economic reasons and strict environmental policies taken by governments. The most important process in the recovery phase of an end-of-life product is disassembly, which is defined as separating the product into its parts. In this study, position-based and logarithmic sum-of-processing-time-based learning effects and deterioration effects were discussed in the disassembly line balancing problem. Jobs are learned by workers as they are repeated. The work learned can be done in a shorter time. Position-based learning is a type of learning in which the processing times of jobs are updated depending on the position in which they are assigned at the station. In logarithmic sum-of-processing-time-based learning, the processing time of a job assigned to any position at a station decreases depending on the sum of the logarithm of the processing times of the jobs assigned to the previous positions. Any event that causes the start time of a job to be delayed or postponed is called job deterioration. In this study, disassembly line balancing problems under the effects of learning and job deterioration were discussed. In this study, the objective function is the minimization of the number of stations. Lingo 11 program was used to solve small-sized problems. Genetic Algorithm (GA), Hybrid Genetic Algorithm (HGA), Particle Swarm Optimization (PSO) and PSO-GA hybrid approaches have been used for medium and large size problems. Test problems for these approaches were solved for the combination of different learning and deterioration rates and the results were compared.

Keywords: Disassembly line balancing, learning effects, deterioration effect, genetic algorithm, local search method, particle swarm optimization.

İÇİNDEKİLER

ÖĞRENME VE BOZULMA ETKİLERİ ALTINDA DEMONTAJ HATTI DENGELEME PROBLEMİ

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

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Demontaj Hattı.....	4
1.1.1. Demontaj hatlarında karşılaşılan zorluklar	7
1.1.2. Demontaj hattı dengeleme literatürü	11
1.2. Çizelgeleme	14
1.2.1. Öğrenme etkisi.....	15
1.2.2 Bozulma etkisi.....	16

2. BÖLÜM

YÖNTEM VE MATERYAL

2.1. Problem Tanımı.....	17
2.2. Notasyon.....	19

2.3. Öğrenme ve bozulma etkileri altında DHD problemleri için matematiksel model	20
2.4. Öğrenme ve bozulma etkileri altında DHD problemlerinin çözümü için önerilen metasezgisel yaklaşımlar	22
2.4.1. Genetik Algoritma.....	22
2.4.2 Hibrit Genetik Algoritma (HGA)	26
2.4.3 Parçacık Sürüsü Optimizasyonu (PSO)	29
2.4.4. Parçacık Sürüsü Optimizasyonu –Genetik Algoritma (PSO-GA) Hibrit Yaklaşımı.....	31

3. BÖLÜM

BULGULAR

3.1. Pozisyon tabanlı ve logaritmik işlem süreleri toplamı tabanlı öğrenme etkileri altında DHD problemi	32
3.2. Bozulması etkisi altında DHD problemi	50
3.3. Eşzamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında DHD problemi	59
3.4. Eşzamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında DHD problemi	70

4. BÖLÜM

SONUÇ ve ÖNERİLER

KAYNAKÇA	87
EKLER.....	99
ÖZGEÇMİŞ.....	127

KISALTMALAR

bo:	Bozulma oranı
DÇP:	Demontaj Çizelgele Planı
DHD:	Demontaj Hattı Dengeleme
GA:	Genetik Algoritma
HGA:	Hibrit Genetik Algoritma
İTZ :	İleri Tedarik Zinciri
KDTZ:	Kapalı Döngü Tedarik Zinciri
LİSTTÖE:	Logaritmik İşlem Süreleri Toplamı Tabanlı Öğrenme Etkisi
öö:	Öğrenme oranı
PSO:	Parçacık Sürüsü Optimizasyonu
PTÖE:	Pozisyon Tabanlı Öğrenme Etkisi
TTZ:	Tersine Tedarik Zinciri

TABLOLAR LİSTESİ

Tablo 1.	Montaj ve demontaj hatlarının karşılaştırılması.....	10
Tablo 2.	DHD probleminin literatür özeti	13
Tablo 3.	8-parçalı DHD probleminin görev tanımları ve parametreler	18
Tablo 4.	Rassal başlangıç kromozomu iş sırası örnekleri	24
Tablo 5.	Yerel arama yönteminin yalancı kodu (Gonçalves and Almedia, 2002)	27
Table 6.	Pozisyon ve logaritmik işlem süresi toplamı tabanlı öğrenme etkileri altındaki 8 parçalı DHD problemi için Lingo 11 ile elde edilen sonuçlar ...	33
Tablo 7.	8 parçalı DHD probleminde istasyonlardaki optimum iş sıraları.....	34
Tablo 8.	PTÖE ve LİSTTÖE altında görevlerin işlem süreleri.....	35
Tablo 9.	25 görevli cep telefonu probleminin görev tanımları, süreleri ve öncelik ilişkileri.....	36
Tablo 10.	25 parçalı DHD problemi örneği için literatürdeki çalışmaların sonuçlarının karşılaştırılması.....	37
Tablo 11.	LİSTTÖE altında 25 parçalı DHD problemi için atama detayları	38
Tablo 12.	PTÖE altında 25Parçalı DHD problemi için atama detayları	40
Tablo 13.	LİSTTÖE altında 83 Parçalı DHD problemi için atama detayları	43
Tablo 14.	PTÖE altında 83 Parçalı DHD problemi için atama detayları	45
Tablo 15.	Farklı veri kümeleri için LİSTTÖE ve PTÖE ile elde edilen sonuçlar	47
Tablo 16.	15 farklı DHD problemi için öğrenme etkileri altında açılan istasyon sayısı değerleri.....	48
Tablo 17.	GA, HGA, PSO ve PSO-GA yaklaşımlarının çözüm sonuçları (LİSTTÖE etkisi altında).....	49
Tablo 18.	Bozulma etkisi altında 8 parçalı DHD probleminin optimum sonuçları.....	51
Tablo 19.	Bozulma Oranı 0.1 iken işlerin işlem sürelerinin hesaplanması	51
Tablo 21.	Bozulma etkisi altında 83 parçalı DHD probleminin atama detayları	55
Tablo 22.	Farklı DHD problemleri için bozulma etkisi altında açılan istasyon sayıları	56
Tablo 23.	GA, HGA, PSO ve PSO-GA yaklaşımlarının çözüm sonuçları (Bozulma etkisi altında).....	58
Tablo 24.	8 parçalı DHD problemi için optimal sıra ve iş istasyonu sayısı	60

Tablo 25. Öğrenme oranı 0.70 ve bozulma oranı 0.10 iken eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında işlem süreleri.....	60
Tablo 26. Pozisyon tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için optimal sıra ve iş istasyonu sayısı.....	62
Tablo 27. Pozisyon tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi için iş istasyonu sayısı	65
Tablo 28. Farklı DHD problemleri için pozisyon tabanlı öğrenme ve bozulma etkileri altında açılan istasyon sayıları	67
Tablo 29. GA, HGA, PSO ve PSO-GA yaklaşımlarının çözüm sonuçları (PTÖE ve bozulma etkileri altında).....	71
Tablo 30. Logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 8 parçalı DHD probleminin optimum sonuçları.....	73
Tablo 31. Öğrenme oranı 0.70 ve bozulma oranı 0.20 iken işlerin işlem süreleri	73
Tablo 32. Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için iş istasyonu sayısı.....	74
Tablo 33. Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi için iş istasyonu sayısı.....	77
Tablo 34. Farklı DHD problemleri için LİSTTÖ ve bozulma etkileri altında açılan istasyon sayıları	79
Tablo 35. GA, HGA, PSO ve PSO-GA yaklaşımlarının çözüm sonuçları (LİSTTÖE ve bozulma).....	82

ŞEKİLLER LİSTESİ

Şekil 1.	Alternatif demontaj kurulumları: (a) tek iş istasyonu, (b) demontaj hücresi, (c) demontaj hattı (Güngör & Gupta, 2001).	6
Şekil 2.	8-parçalı DHD probleminin öncelik ilişkiler diyagramı	19
Şekil 3.	Öncelikli korunumlu çaprazlama yöntemi (Precedence preservative crossover method)	25
Şekil 4.	Öncelikli düzgün operatör yöntemi (Precedence uniform operator method).....	26
Şekil 5.	Parçaları yeniden sıralama çaprazlama yöntemi (Fragment reordering crossover method)	28
Şekil 6.	Sağda bir nokta operatör mutasyon yöntemi (One point right operator mutation method)	28
Şekil 7.	PSO algoritması akış diyagramı	30
Şekil 8.	LİSTTÖE ile PTÖE altında 25 parçalı DHD probleminde açılan istasyon sayıları	41
Şekil 9.	LİSTTÖE ile PTÖE altında 83 parçalı DHD probleminde açılan istasyon sayıları	46
Şekil 10.	İş bozulması etkisi altında 25 parçalı DHD probleminde açılan istasyon sayısı.....	53
Şekil 11.	Bozulma etkisi altında 83 parçalı DHD probleminde açılan istasyon sayıları	56
Şekil 12.	15 farklı DHD probleminin bozulma etkisi altında açılan istasyon sayıları	57
Şekil 13.	Eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için açılan istasyon sayılarının değişimi.....	63
Şekil 14.	Eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi için açılan istasyon sayılarının değişimi.....	66
Şekil 15.	Bozulma oranları sabit iken öğrenme oranlarındaki değişime göre amaç fonksiyonu değerleri.....	68
Şekil 16.	Öğrenme oranları sabit iken bozulma oranlarındaki değişime göre amaç fonksiyonu değerleri.....	69

Şekil 17.	Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için açılan istasyon sayılarının değişimi	76
Şekil 18.	Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi için açılan istasyon sayılarının değişimi	78
Şekil 20.	Öğrenme oranları sabit iken bozulma oranlarındaki değişime göre amaç fonksiyonu değerleri.....	81



GİRİŞ

Bir tedarik zinciri, tedarikçiler, üreticiler, dağıtıcılar ve satıcılar gibi birçok elemanın, hammaddeleri elde etmek, bu hammaddeleri son ürünlere çevirmek ve bu son ürünleri de nihai kullanıcılara ulaştırmak için birlikte çalıştığı entegre bir süreçtir (Beamon & Fernandes, 2004). Tedarik zinciri, hammaddenin elde edilmesi ile başlayıp, hammaddenin yarı mamule daha sonrada nihai ürüne dönüştürölüp son müşteriye ulaştırılması işlemlerini gerçekleştiren tesis ve dağıtım şirketlerinin ağıdır. Tedarik zinciri, ürünlerin hammaddeden müşteriye teslim edilene kadar geçirdikleri tüm faaliyetlerdir.

Artan çevresel kaygılar ve Dünya'nın doğal kaynaklarının tükenmesi, sanayileşmiş ülkelerin hükümetlerini önleyici tedbirler almaya zorlamıştır. Bu hükümetler, üreticileri çevre dostu ürünler tasarlamak ve ürünlerin çevre üzerindeki etkisini azaltmak için çevre dostu uygulamalar yapmaya teşvik etmektedir (Tuncel ve ark., 2014). Günümüzde, teknolojiideki hızlı gelişmeler nedeniyle pazara hızla yeni ürünler girmektedir. Sonuç olarak, teknolojiideki bu sürekli değişim sayesinde ürünler daha çabuk eski hale gelir ve her zamankinden daha hızlı atılır. Bu eski hale gelen ürünleri ekonomik açıdan değerlendirmek için yeni çözümler aranmaya başlanmıştır (Avikal ve ark., 2014). Yaşam süreci sonunda parça ve malzemeler, ürün geri kazanım işlemleri uygulanmadan önce çeşitli işlemlerden geçmektedir. Bu işlemler temizleme, demontaj, parçalama, sınıflandırma ve kontrol olarak belirtilmiştir (Duflou ve ark., 2008).

Tedarik zincirinde tedarikçi, üretici, dağıtıcı gibi birçok aktör yer alsa da zincirdeki son aktör genellikle son kullanıcı olan müşteridir. Bu şekilde olan, yani akış yönü hammadde tedarikçisinden son kullanıcı müşteriye doğru olan yapılara ileri tedarik zinciri (İTZ) denir.

Tedarik zinciri yönetimi uzun yıllarca ileriye doğru olan faaliyetlerle ilgilenmiştir. Ancak artan çevre sorunları, müşterilerin geri dönüşüm hakkında bilinçlenmesi ve yasal

düzenlemeler, şirketler üzerine çevresel sorumluluğun yanı sıra sosyal sorumluluk da yüklemiştir. Bunun sonucunda, tersine lojistik, ürün geri kazanımı ve tersine tedarik zinciri (TTZ) yönetimi gibi kavramlar ortaya çıkmıştır. Böylece şirketler, tersine lojistik üzerine yoğunlaşarak diğer şirketlere göre maliyet ve rekabet avantajı elde etmeyi düşünmüşlerdir (H. Wang & Hsu, 2010). Tedarik zinciri yapılarına yeniden kullanım, tamir, yenileme veya demontaj gibi geri dönüşüm faaliyetlerinin eklenmiş yeni halleri TTZ ve kapalı döngü tedarik zinciri (KDTZ) olmak üzere iki farklı şekilde ortaya çıkmaktadır (Yang ve ark., 2009). TTZ’nde müşterilerden toplanan kullanılmış ürünlerin ters yönde ilerleyişi söz konusudur. KDTZ’de ise ürün üreticiden tüketiciye doğru ilerledikten sonra tüketiciden de üreticiye doğru ilerleyerek döngüyü tamamlamaktadır. Yani KDTZ, hem ileri hem de tersine akışın olduğu bütünleşik bir ağ yapısına sahiptir. TTZ ve KDTZ’nin hammadde kaynağı İTZ’deki gibi tedarikçiler değil, son kullanıcılar ve müşterilerdir (Mondragon ve ark., 2011).

TTZ ve KDTZ ağlarının önem kazanmasındaki ana sebepler şunlardır (Mondragon ve ark., 2011):

- Ürün geri dönüşünün yüksek miktarda olması,
- Geri dönen ürünlerin ikincil pazarda satış imkanının olması,
- Geri dönüşüme olan önemin ve çevre duyarlılığının artması,
- Yaşam döngüsü bitmiş ürünlerin yasal yollarla geri dönüşümlerinin benimsenmesi,
- Müşterilerin, geri dönüşüm bilincinin artması.

Ürün geri dönüşünün artmasına neden olan diğer faktörler aşağıda verilmiştir (Nakıboğlu, 2007):

Üretici dönüşleri:

- Ürünün hammadde fazlası olması,
- Ürünün kalite kontrolde başarısız olması,
- Ürünün üretim fazlası olması,
- Üründeki hata, güvenlik veya sağlık problemleri gibi sebepler dahilinde üreticinin ürünü geri çağırması.

Dağıtıcı dönüşleri:

- Yanlış ve hasarlı teslimat yapılması,
- Kullanım süresi geçmiş ürünler,
- Stok fazlası ürünler,
- Mevsimsel dalgalanmalar.

Müşteri dönüşleri:

- Garanti kapsamındaki dönüşler,
- Ürünlerin tamir edilerek yeniden kullanılması,
- Değer kazanımı, hurda değeri vb.

Kullanım sonu dönüşleri:

- Ürünün ömrü sonu dönüşleri,
- Zararlı malzemelerle ilgili yasal düzenlemeler,
- Ürünün hasarlı olması.

Fonksiyonel Dönüşler:

- Paketleme malzemeleri, palet, paket, kutu gibi yeniden kullanılabilir ürünlerin olması,

Bu sebepler sonucunda ürün geri dönüşüm sistemlerini mevcut tedarik zinciri yapılarına entegre eden işletmeler, enerji ve doğal kaynak kullanımında tasarruf, temiz hava ve su tasarrufu, atık alanlarından tasarruf ve enerji tasarrufu gibi faydaları sağlayabilirler.

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR ÇALIŞMASI

1.1. Demontaj Hattı

Çevre yasaları ve yönetmelikleri, doğal kaynakların tükenmesi, depolama maliyetlerinin artması, artan tüketici bilinci ve sosyal sorumluluklar gibi farklı faktörler, üreticileri kullanılmış veya hatalı olarak geri dönen ürünlerin tekrar kazanılmasını sağlamaya zorlamıştır (Agrawal & Tiwari, 2008; Hezer & Kara, 2015). Son yıllarda teknolojinin hızla gelişmesi ve yaşam döngüsünün kısalması ile birlikte ömrünü tamamlamış ürünlerin sayısı artmaktadır (Li & Janardhanan, 2021; K. Wang, Li, & Gao, 2019). Son kullanma tarihi geçmiş bu ürünlerin yeniden kullanılması veya yeniden üretilmesi ekonomik fayda sağlamakta ve bu ürünlerin neden olduğu çevre kirliliğini azaltmaktadır (Li & Janardhanan, 2021). Ürün geri kazanımı, eski veya tarihi geçmiş ürünleri demonte ederek, tekrar üreterek veya geri dönüştürerek atık depolama alanlarına gönderilen atık miktarını en aza indirmeyi amaçlamaktadır (Aşkıner Güngör & Gupta, 2002; Kalayci & Gupta, 2013a). Geri dönüşüm, üründen sökme, fiziksel ve kimyasal ayırma işlemlerini gerçekleştirerek ürünün malzeme içeriğini kurtarır. Ancak yeniden üretim, ürünün kimliğini korur ve ürünü istenen kalite seviyesine getirmek için gerekli sökme, ayırma, yenileme ve montaj işlemlerini gerçekleştirir. Demontaj, istenen parçaların seçici olarak ayrılmasına izin vererek malzeme ve ürün geri kazanımındaki rolünü kanıtlamıştır (Güngör & Gupta, 2002; Kalaycılar ve ark., 2016; S. M. McGovern & Gupta, 2007).

Demontaj, ürün geri kazanımında ürünlerin sistematik olarak alt bileşenlere ayrılmasını sağlayan en önemli ve zaman alıcı adımdır. Demontaj sistemlerinde "ürün", bir geri dönüşüm veya yeniden üretim merkezine gelen "kullanılmış" veya "iade edilmiş" bir öğeyi ifade eder (Hezer & Kara, 2015). Demontaj, hala üretilmekte olan ürünler için ortak olan değerli parçaların ve alt montaj parçalarının geri kazanımını, bazı parçalar için gelebilecek ani taleplerin karşılanabilmesini, tehlikeli parçaların çıkarılmasını, ömrünü

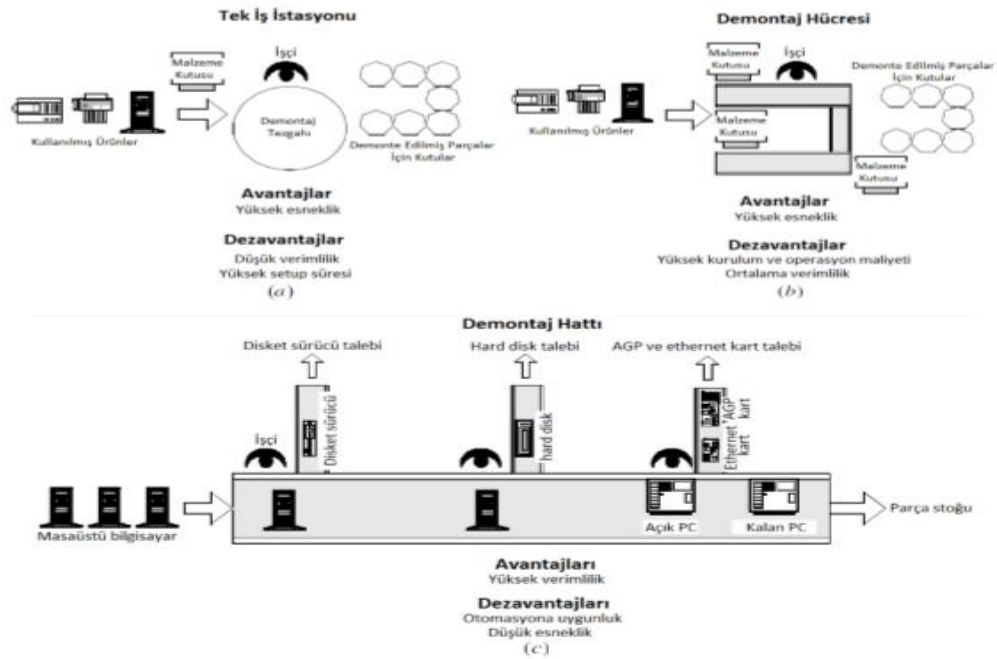
tamamlamış üründen ileride kullanılmak üzere envantere gönderilebilen parçaların çıkarılmasını, depolama alanına gönderilecek atık miktarının azaltılmasını ve çevre dostu üretim standartlarına ulaşmayı amaçlamaktadır (Aşkiner Güngör & Gupta, 2002). Böylece verimli bir demontaj hattının tasarımı büyük ölçüde endüstriyel ve çevresel öneme sahiptir (Kalayci ve ark., 2015).

Demontajdan sonra, yeniden kullanılabilir parçalar temizlenir, yenilenir, test edilir ve yeniden imalatla kullanılmak üzere envantere yönlendirilir. Geri gelen ürünlerin yapısında ve kalitesinde belirsizlikler olabilmektedir. Alınan ürünlerin durumu bilinmemektedir ve genellikle bileşenlerinin güvenilirliği ve sağlamlığı şüphelidir. Ürünün bazı parçaları tehlikeli olabilmektedir. Bu özellikler, demontaj iş istasyonlarının kullanımını etkileyebilecek özel işlemler gerektirebilmektedir (S. M. McGovern & Gupta, 2007).

Başka bir belirsizlik sorunu, ürünlerdeki parçaların miktarıyla ilgilidir. Ürün kullanımından kaynaklı olarak, ürünlerdeki parçaların sayısı orijinal yapılandırmasıyla eşleşmeyebilir. Gerçek parça sayısı, ürün alındığında beklenenden daha fazla veya daha az olabilir. Ürünün talep edilen parçaları olmayabilir veya miktarı beklenenden az olabilmektedir. Bu durumla başa çıkmak için, sökülecek parça sayısı için gerçek gereksinimi dinamik olarak ayarlamak gerekebilmektedir (Aşkiner Güngör & Gupta, 1999).

Demontajdaki bir diğer önemli husus, verimli bir demontaj çizelgeleme planının (DÇP) oluşturulmasıdır. DÇP, demonte edilecek bir ürünle başlayan ve ilgili tüm parçaların bağlantısının kesildiği bir durumda sonlanan bir demontaj görevi dizisidir. Etkili bir DÇP, sökme işleminin maliyetini en aza indirebilmektedir (Aşkiner Güngör & Gupta, 2001).

Demontaj sistemlerinin işçi, zaman ve maliyeti optimize edecek şekilde tasarlanması ve geliştirilmesi gerekmektedir. Geri dönüşüm merkezlerinde, geri dönen ürünlerin demontaj işlemleri tek iş istasyonunda, bir demontaj hücresinde veya bir demontaj hattında gerçekleştirilebilir (Güngör & Gupta, 2001).



Şekil 1. Alternatif demontaj kurulumları: (a) tek iş istasyonu, (b) demontaj hücresi, (c) demontaj hattı (Güngör & Gupta, 2001).

Tek bir iş istasyonu veya demontaj hücresi, parçaların miktarlarına ve kalitesine göre sınıflandırılması için en esnek ortamı sağlamasına rağmen, demontaj hattı en yüksek verimlilik oranını sağlar. Demontaj hattı büyük boyutlu ürünlerin veya büyük miktarlardaki küçük boyutlu ürünlerin demontajı için en uygun yöntemdir. Demontaj hattı gelecekteki otomatik demontaj sistemleri için de en uygun ve verimli seçimdir (Güngör & Gupta, 2002).

Demontaj hatları, hatta sökülen farklı ürünlerin sayısı ve düzenleri bakımından birkaç gruba ayrılabilir. Demonte edilen farklı ürünlere göre, sökme hatları "tek ürün" ve "çoklu / karışık ürün" sökme hatları olarak sınıflandırılabilir. Tek ürünlü sökme hattı, yalnızca bir ürün türünü sökmek içindir. Oysa çok / karışık ürün sökme hattı farklı ancak benzer ürün türleri veya belirli bir ürünün farklı türleri içindir (Hezer & Kara, 2015).

Montaj hattının bir ürünü birleştirmenin en verimli yolu olduğu düşünüldüğünde, demontaj hattının da bir ürünü sökmenin en etkili yolu olduğu görülmektedir. Demontaj, montaj ile benzerlikler gösterse de montaj işleminin tersi değildir. Demontaj hatlarındaki zorluk, kendine özgü özelliklere sahip olması gerçeğinde yatmaktadır (Kalaycı & Gupta, 2013b; McGovern & Gupta, 2007).

Demontaj işlemi montajdan daha karmaşıktır. Aslında, bir demontaj sisteminde montaj hattının aksine, bir ürün, kalitesi, miktarı ve güvenilirliği belirsizlik içeren çeşitli bileşenlere ve alt montajlara ayrılır. Ömrü dolmuş ürünlerin yapısı ve kalitesi son derece belirsizdir ve bu tür ürünlerdeki bileşenlerin sayısı bile tahmin edilemez. Ayrıca, ömrü tükenmiş bir ürün, bir demontaj hattının iş istasyonunda özel işlem gerektiren bazı tehlikeli maddeler içerebilir. Bu yüzden, demontaj hatlarının tasarımı ve dengelenmesi, zor bir optimizasyon problemidir (Bentaha ve ark., 2015).

Demontaj işlemi genellikle iş istasyonlarından oluşan bir hat üzerinde gerçekleştirilir. Demontaj hatları, ömrünü tamamlamış ürünlerin geri dönüştürülmesi veya yeniden üretilmesi için fırsatlar sunmaktadır (Edis, 2021). Demontaj hatları, yerleşimlerine göre düz, U tipi, paralel ve iki taraflı demontaj hatları olarak sınıflandırılabilir. Ancak demontaj hattı dengeleme (DHD) problemi literatüründe çalışmaların çoğu genellikle düz sökme hattına odaklanmaktadır (Hezer & Kara, 2015). Hat dengeleme, demontaj için harcanan zaman ve para gibi kaynakların en aza indirilmesi ve demontaj sürecinin otomasyon seviyesini ve geri kazanılan parça veya malzemelerin kalitesini en üst düzeye çıkarmak için çok önemlidir (McGovern & Gupta, 2007). Kaynak kullanımını optimize etmek için bir demontaj hattı dengelenmelidir. Demontaj hattı dengeleme, verimli ve etkili ürün geri dönüşümü elde etmenin en uygun yoludur. Demontaj hattı dengeleme problemi, görevler arasındaki öncelik kısıtlamalarını karşılamak ve döngü süresini aşmamak gibi birkaç optimizasyon kriteri ile belirli demontaj görevlerinin istasyonlara atanmasına izin verir. Bu nedenle, verimli bir şekilde tasarlanmış ve dengeli bir demontaj hattı, çevresel ve endüstriyel açıdan önemli bir öneme sahiptir (Li ve ark., 2020).

1.1.1. Demontaj hatlarında karşılaşılan zorluklar

Demontaj hatları yüksek verimlilik gösteren hatlar olsa da çeşitli zorluklarla karşılaşılabilir. Bunlar (Gupta & Gungor, 2001; Özceylan ve ark., 2014):

Ürünle ilgili zorluklar: ürünlerin değişen karakteristik yapıları, demontaj hattı üzerindeki operasyonları karmaşıktır. Bu durumlarda montaj hattı dengeleme çok kompleksleşebilir. Demontaj hattı bir grup ürün için dengeli iken, arkasından gelecek bir diğer grup ürün için denge durumu bozulabilir.

Demontaj hattının yapısı ile ilgili zorluklar: Çeşitli hat tipleri vardır: seri, paralel, hücresel, U tipi veya çift taraflı olabilir. Bu demontaj hatları, demontaj sürecinde ürün çeşitliliği ve düzensizliklerle baş edebilmek için önerilmiştir. Bir diğer durum ise, hat hızıdır. Demontaj hattı, sökülen ürünlere olan talebin değişken etkisini minimize etmek için dinamik olmalıdır.

Parça ile ilgili zorluklar: Demontaj hattına gelen ürünlerin kalite seviyesi ciddi belirsizlik içerebilir. Ürünler fiziksel veya fonksiyonel hasarlı olabilir. Demontaj sonrası ortaya çıkan ürünler de beklentiyi karşılamayabilir, arızasız olabileceği gibi arızalı da olabilir.

Operasyonel zorluklar: Demontaj görev süreleri kesin, belirsiz ve dinamik olabilmektedir. Bu görev süreleri farklılığı ürün durumu ve demontaj iş hattının durumuna bağlı olarak ortaya çıkabilir. Ürün kusurları olduğunda görevler farklı işleyebilir. Bu gibi durumlarda alınan önlemler arasında; iş istasyonunu erken terk etmek, bir sonraki iş istasyonuna / istasyonlarına atlamak, hattan çıkmak, önceki iş istasyonunu tekrar ziyaret etmek olabilir.

Taleple ilgili zorluklar: Üç talep tipi olabilir: tek bir parçaya talep, birkaç parçaya talep veya tüm parçalara talep. Talep edilen ürünlerde görülebilecek fiziksel veya fonksiyonel hata durumu demontaj hattını daha karmaşıktırır.

Atama ile ilgili zorluklar: Demontaj görevlerinin iş istasyonlarına atanmasında benzer görevlerin gruplandırılması, yürüme mesafesinin en küçük olduğu istasyonun seçilmesi, bazı ekipmanların sadece belirli iş istasyonlarında bulunması gibi durumları içeren sınırlamalar vardır.

Diğer zorluklar: İş istasyonlarının güvenilirliğiyle ilgili ekstra belirsizlikler vardır. Demontaj sırasında bazı parçaların kırılması, kirliliği veya zararlı olması demontaj hatlarıyla ilgili diğer zorluklardır.

Bazı durumlarda, düz bir sökme hattının kapasitesi, geri kazanılan parçaların yüksek talebi nedeniyle yetersizdir. Bu nedenle, talebi karşılamak için kapasitenin artırılması istenebilir. Bunu yapmanın yollarından biri, birkaç sökme hattının paralel şekilde yerleştirilerek paralel sökme hattı oluşturulmasıdır. Bu durumda iki veya daha fazla düz

sökme hattı aynı anda paralel ve dengelidir. Bu dengeye ulaşma problemine paralel sökme hattı dengeleme problemi denir (Hezer & Kara, 2015).

Demontaj hatları, hattın verimliliğini artırmak ve demontaj maliyetlerini azaltmak için etkin bir şekilde dengelenmelidir. Bu probleme DHD problemi denir. DHD probleminin amacı genellikle gereken minimum sökme iş istasyonu sayısını bulmaktır (Hezer & Kara, 2015).

DHD probleminin amacı, demontaj hattının kaynaklarını mümkün olduğunca verimli kullanmak ve talebi karşılamaktır. Kaynakların verimli kullanılması, gereken minimum sayıda demontaj iş istasyonunun bulunmasını, demontaj görevlerinin istasyonlara en uygun şekilde atanmasını ve demontaj hattının yerleşim ve malzeme taşıma özelliklerinin geliştirilmesini gerektirir. Bu hedeflere ulaşılırken, görevler arasındaki öncelik ilişkileri yerine getirilmelidir (Güngör & Gupta, 1999).

Demontaj hattı dengeleme problemi, iş istasyonu sayısını minimize etmek, iş istasyonlarındaki boş zamanı dengelemek ve tehlikeli parçaların çıkarılmasına öncelik vermek gibi hedefleri karşılayacak şekilde demonte edilecek ürünlerin iş istasyonlarına atanmasını sağlar (Özceylan ve ark., 2019).

Demontaj işlemi, demonte edilen parçalara göre kısmi demontaj ve komple demontaj olmak üzere iki şekilde olabilir. Kısmi sökmede ürün tamamen parçalarına ayrılmaz, ihtiyaç duyulan veya tekrar kullanılabilir parçalar sökülür. Tam sökme işleminde ise ürünün tüm parçaları demonte edilir (Goksoy Kalaycılar ve ark., 2022).

Demontaj hattı dengeleme, demontaj görevleri arasında çeşitli performans ölçütleri ve öncelik kısıtlamaları dikkate alınarak demontaj istasyonlarına demontaj görevlerinin atanmasını içerir (Güngör & Gupta, 2001; Ilgin ve ark., 2017).

Montaj hatlarının dengelenmesi için geliştirilen metodolojiler, demontaj sistemlerindeki karmaşıklıklar nedeniyle, demontaj hatlarının dengelenmesinde kullanılamaz. Demontaj hatlarında fonksiyonel ve fiziksel öncelik kısıtlamaları bulunurken, montaj hatları sadece fiziksel öncelik kısıtlamalarını içerir. İkincisi, demontaj hatlarında, hattın farklı istasyonlarında sökülen belirli bileşenler ve alt montajlar için talep oluşurken, montaj hatlarındaki son ürüne talep gelir. Üçüncüsü, şirketler tüketicilerden toplanan demonte

edilecek ürünlerin miktarı ve kalitesi üzerinde sınırlı kontrole sahip olduğundan, demontaj hatlarında sökülen ürünlerin miktarı ve kalitesi ile ilgili yüksek bir belirsizlik seviyesi vardır (Bentaha ve ark., 2015; Ilgin ve ark., 2017).

Montaj ve demontaj hatlarının teknik ve operasyonel açıdan karşılaştırılması Tablo 1'de verilmiştir (Güngör & Gupta, 2001).

Tablo 1. Montaj ve demontaj hatlarının karşılaştırılması

Özellikler	Montaj Hattı	Demontaj Hattı
Talep kaynakları	Tek	Çok
Talep edilen ürün	Nihai ürün	Bileşenler
Öncelik ilişkileri	Var ve sıkı	Var ve esnek
Öncelik ilişkilerine bağlı karmaşıklık	Yüksek	Orta
Parça kalitesine bağlı belirsizlik	Düşük	Yüksek
Parça sayısına bağlı belirsizlik	Düşük	Yüksek
İstasyonlara ilişkin belirsizlik	Düşük-orta	Yüksek
İstasyon güvenilirliği	Yüksek	Düşük
Çoklu ürün	Mevcut	Mevcut
Akış süreci	Yakınsak	İraksak
Hat esnekliği	Düşük-orta	Yüksek
Performans ölçütlerine bağlı karmaşıklık	Orta	Yüksek
Bilinen performans ölçütleri	Çok sayıda	Mevcut değil
Yok olan parça durumu	Mevcut değil	Mevcut
Patlayan parça durumu	Mevcut değil	Mevcut
Gürbüz hat gerekliliği	Orta	Yüksek
İstasyonlar arası stoğa bağlı karmaşıklık	Orta	Yüksek
Hat optimizasyonu için bilinen teknikler	Çok sayıda	Çok az

Bir demontaj hattının montajı genellikle uzun vadeli bir karara dayanır ve büyük bir sermaye yatırımı gerektirir. Bu nedenle, böyle bir sistemin mümkün olduğunca verimli çalışması için tasarlanması ve dengelenmesi önemlidir (Güngör & Gupta, 1999).

Literatürde demontaj işlemleri ve demontaj hatları ile ilgili, demontaj planlaması, demontaj sıralaması (çizelgelemesi) ve demontaj hattı dengelemesi gibi hatlarda ortaya çıkan bir dizi alt problem vardır (Özceylan ve ark., 2019).

DHD problemini çözerken, görevler arasındaki öncelik ilişkileri ve çevrim süresi kısıtı dikkate alınmalıdır. Öncelik ilişkileri DHD problemini daha karmaşık hale getirir, bu nedenle DHD problemleri NP zor problemlerdir. McGovern ve Gupta (2007), DHD problemlerinin NP-zor olduğunu kanıtlamıştır.

1.1.2. Demontaj hattı dengeleme literatürü

Son yıllarda teknolojinin hızla gelişmesi ve yaşam döngüsünün kısalması ile birlikte ömrünü tamamlamış ürünlerin sayısı da artmaktadır (Li & Janardhanan, 2021; K. Wang ve ark., 2019). Son kullanma tarihi geçmiş bu ürünlerin yeniden kullanılması veya yeniden üretilmesi ekonomik fayda sağlar ve bu ürünlerden kaynaklanan çevre kirliliğini azaltır (Li & Janardhanan, 2021). Demontaj, ürün geri kazanımında ürünlerin sistematik olarak alt bileşenlere ayrılmasını sağlayan en önemli ve zaman alıcı adımdır. Sökme işlemi genellikle iş istasyonlarından oluşan bir hat üzerinde gerçekleştirilir.

DHD problemleri, demontaj görevlerini sıralı bir istasyon dizisine atarken görevler arasındaki öncelik ilişkilerini sağlayıp, istasyon sayısının minimize edilmesi, çevrim süresinin minimize edilmesi, verimliliğin maksimize edilmesi, karın maksimize edilmesi veya maliyetin minimize edilmesi gibi ölçütleri optimize etmeyi amaçlar (Ding ve ark., 2010).

Demontaj hattı dengeleme problemlerini literatüre Gupta ve Güngör (1999) kazandırmıştır. Demontaj görevleri arasındaki öncelik ilişkilerinin ihlal edilmemesi koşuluyla, demontaj işlemlerinin demontaj iş istasyonlarına optimum şekilde atanması olarak tanımlanabilecek yeni bir sorun olan DHD problemini tartışmışlardır. Amaç, talebi karşılamak ve demontaj hattını mümkün olduğunca verimli kullanmaktır.

Güngör ve Gupta (2001), demonte edilecek parçaların bazılarının arızalı olmasının DHD probleminin çözümünü nasıl etkilediği üzerine çalışmışlardır. Arızalı parçalar, demontaj hattını olumsuz yönde etkileyebilmekte ve demontaj işlemlerinin yapılmamasına neden olabilmektedir. Örneğin, bir görev bir hata nedeniyle gerçekleştirilemezse, görevler arasındaki öncelik ilişkileri nedeniyle kalan görevlerin bazıları veya tümü devre dışı bırakılabilir. Bu, demontaj hattındaki iş parçalarının akışında erken ayrılma, atlama, kaybolma veya parçaları tekrar işleme gibi çeşitli komplikasyonlara neden olabilir. Bu komplikasyonlar ayrıntılı olarak açıklanmıştır ve tehlikeli ürünlerin öncelikli olarak sökülmesine izin veren minimum istasyon sayısını hedefleyen buluşsal bir yaklaşım önerilmiştir.

Parçaların yapısal belirsizliği ve bu komplikasyonlara ek olarak demontaj iş sürelerindeki değişkenliğin, demontaj hatlarına gelen ürünlerin durumundan, demontaj iş

istasyonlarından ve hat çalışanlarından kaynaklanabileceği belirtilmiştir. Demontaj hatlarının tasarlanması ve dengelenmesi onların en iyi şekilde çalışabilmesi için önemlidir (S.M. Gupta & Güngör, 2001). Güngör & Gupta (2002; 2001), demontaj hattı dengeleme problemi ve zorlukları üzerine çalışmışlardır.

DHD problemlerini çözmek için kullanılan yöntemler üç kategoride sınıflandırılabilir. Bunlar sezgisel yöntemler, metasezgisel yöntemler ve matematiksel programlama teknikleridir. İlk kategori sezgisel tabanlı çözüm yaklaşımlarıdır. Güngör ve Gupta (2002), farklı durumlarda DHD problemi için basit bir sezgisel yöntem geliştirmişlerdir. McGovern ve Gupta (2004), çok amaçlı DHD problemi için bir açgözlü/2 optimal(2-opt) hibrit algoritma önermişlerdir. Ren ve ark., (2018), bu yöntemi ağırlıklara dayalı çok kriterli kararlarla genişletmiştir.

İkinci kategori metasezgisel yöntemlerdir. Bunlar; genetik algoritma (Aydemir-Karadag & Turkbey, 2013; Kalayci ve ark., 2016; McGovern & Gupta, 2007), karınca kolonisi optimizasyonu (Agrawal & Tiwari, 2008; Ding ve ark., 2010; Kalayci & Gupta, 2013b; Seamus M. McGovern & Gupta, 2006), yapay arı kolonisi (Kalayci & Gupta, 2013c; Jia Liu & Wang, 2017), tabu arama (Kalayci & Gupta, 2014), parçacık sürü optimizasyonu (Kalayci & Gupta, 2013a; Xiao vd., 2017), yerçekimsel arama algoritması (Ren ve ark., 2017), yapay balık sürüsü algoritması (Zhang ve ark., 2017), ayırık arı algoritması (Liu ve ark., 2018), ateş böceği algoritması (Zhu ve ark., 2018), ayırık guguk kuşu arama algoritması (Li & Janardhanan, 2021) ve balina optimizasyon algoritması (Y. Zhang ve ark., 2022). Üçüncü yöntem olarak, DHD problemlerini optimal olarak çözmek için matematiksel programlama teknikleri kullanılmaktadır. Altekin ve ark. (2008), kar odaklı kısmi DHD problemi için bir karma tamsayılı programlama formülasyonu geliştirdiler. Altekin (2017), stokastik DHD problemi için parçalı doğrusal programlama yaklaşımı önerdi. Koç vd. (2009), AND/OR grafiğini kullanarak DHD problemi için dinamik programlama ve tamsayı programlama ile iki kesin formülasyon geliştirmişlerdir. Paksoy ve ark. (2013), DHD problemi için bulanık hedefler kullanan karma bir model önermiştir. Ilgın ve ark. (2017), DHD hattı için doğrusal fiziksel programlama önermiştir. Özceylan ve Paksoy (2013), çok amaçlı DHD problemi için doğrusal olmayan karma tamsayılı programlama modeli geliştirmiştir.

Kullanılan hedefleri içeren ilgili DHD literatürünün bir özeti Tablo 2'de verilmiştir. DHD problemleriyle ilgili detaylı literatür bilgisi Özceylan ve ark.(2019) ve Laili ve ark. (2020) tarafından sunulmuştur. Sökme işlemi DHD problemlerinde, kısmi veya tamamen olacak şekilde iki türlü olabilmektedir. Kısmi sökme işleminde, parçalar seçilerek sökülür. Ürünün tüm bileşenleri ayrılmaz (Kalaycılar ve ark., 2016; Liang ve ark., 2023; Yin ve ark., 2021; Yin ve ark., 2023).

Tablo 2. DHD probleminin literatür özeti

Makale (yılı göre sıralanmıştır)	Hat tipi	Parametre yapısı	Demontaj türü	Ürün Tipi	Amaç	Çözüm yaklaşımı
Güngör and Gupta(2001)	Düz	Deterministik	Tamamen	Tek tip	İS	Sezgisel
Güngör and Gupta(2002)	Düz	Deterministik	Tamamen	Tek tip	TÜEÇ, FTÜEK, YDEİ	Spesifik sezgisel
McGovern and Gupta(2006)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK, YDEİ	Karınca kolonisi optimizasyonu
McGovern and Gupta(2007b)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK, YDEİ	Genetik algoritma
Agrawal and Tiwari (2008)	U-tipi	Stokastik	Tamamen	Karma	HVMak, KKV	Karınca kolonisi optimizasyonu
Altıkin et al. (2008)	Düz	Deterministik	Kısmi	Tek tip	Kmak	Karınca tamsayılı lineer programlama
Koç et al.(2009)	Düz	Deterministik	Tamamen	Tek tip	İS	Karma tamsayılı lineer programlama
Ding et al. (2010)	Düz	Deterministik	Tamamen	Tek tip	İS, FTÜEK	Karınca kolonisi optimizasyonu
Kalaycı et al. (2013a)	Düz	Deterministik	Tamamen	Tek tip	TÜEÇ, FTÜEK	Parçacık sürüşü optimizasyonu
Kalaycı et al. (2013c)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK	Yapay arı kolonisi algoritması
Kalaycı et al. (2013b)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK	Karınca kolonisi optimizasyonu
Kalaycı and Gupta(2014)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK	Tabu araması
Avikal et al. (2014)	Düz	Bulanık	Tamamen	Tek tip	İS	Bulanık AHP ve PROMETHEE metod tabanlı yaklaşım
Kalaycı et al. (2015a)	Düz	Bulanık	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK	Hibrit ayrık karınca kolonisi algoritması
Hezer and Kara (2015)	Paralel	Deterministik	Tamamen	Tek tip	İS	Ağ tabanlı en kısa rota modeli
Kalaycı et al. (2016)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK	Hibrit genetik algoritma
İlgin et al. (2017)	Düz	Deterministik	Tamamen	Karma	İS, BSMn	Lineer fiziksel programlama
Zhang et al. (2017)	Düz	Bulanık	Tamamen	Tek tip	İS, Mcost, YDEİ	Yapay balık sürüşü
Ren et al. (2017)	Düz	Deterministik	Kısmi	Tek tip	KarMak	Genetik benzetilmiş tavlama
Liu et al. (2018)	Düz	Deterministik	Tamamen	Tek tip	İS, FTÜEK	Ayrık arı algoritması
Zhu et al. (2018)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK	Pareto ateşböceği algoritması
Ren et al. (2018)	Düz	Bulanık	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK, KarMak, KMak, Mcost	2-opt algoritması
Edis et al. (2019)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK, YDEİ	Karınca tamsayılı lineer programlama
Wang et al. (2019)	Düz	Deterministik	Kısmi	Tek tip	İS, KarMak	Genetik algoritma, Tavlama benzetimi
Liu et al. (2020)	Düz	Deterministik	Tamamen	Tek tip	ÇSMn, İS	Ayrık arı algoritması
Li et al. (2020)	Düz	Deterministik	Tamamen	Tek tip	İS	Dallanma ve sınırlandırma Yaklaşımı
Li and Janardhanan (2021)	U-tipi	Deterministik	Kısmi	Tek tip	KarMak	Karınca tamsayılı lineer programlama
Edis (2021)	Düz	Deterministik	Tamamen	Tek tip	İS, TÜEÇ, FTÜEK, YDEİ	Kısıt programlama yaklaşımı
Wang et al. (2021)	Paralel	Deterministik	Kısmi	Tek tip	İS, KarMak	Genetik benzetilmiş tavlama
Kalaycılar et al. (2022)	Düz	Deterministik	Kısmi	Tek tip	KMak, TÜEÇ	Stokastik programlama yaklaşımı
Zhang et al. (2022)	2-tarafı	Deterministik	Tamamen	Tek tip	TÜEÇ, FTÜEK, İYH, Mcost	Geliştirilmiş balina optimizasyon algoritması
Tian et al. (2023)	Düz	Bulanık	Tamamen	tek tip	Bsmn, ETM, HD	Kuzey çakar kuşu optimizasyonu
Proposed Study	Düz	Deterministik	Tamamen	Tek tip	İS	Hibrit genetik algoritma, Parçacık sürüşü optimizasyonu

İS: İş istasyonu sayısı, **BSMn:** Boşta kalan süre minimizasyonu, **TÜEÇ:** Tehlikeli ürünleri erken çıkarma, **FTÜEK:** Fazla talepli ürünleri erken çıkarma, **YDEİ:** Yön değişikliği sayısını en aza indirme, **Mmin:** Maliyet minimizasyonu, **İYH:** İş yükünü hafifletme, **HVMak:** Hat verimliliği maksimizasyonu, **EPEÇ:** Erişilebilir parçaları erken çıkarma, **APEİ:** Arızalı parçaların etkisini en aza indirme, **KKV:** Kaynak kullanımı verimliliği, **KMak:** Kazanç maksimizasyonu, **KarMak:** Kar maksimizasyonu, **ÇSMn:** Çevrim süresi minimizasyonu, **ETM:** Enerji tüketimi minimizasyonu, **HD:** Hat düzgünlüğü.

Tamamen sökme işleminde ise ürünün tüm parçaları sökülür (Liu & Wang, 2017; Ren ve ark., 2020; Zhang ve ark., 2017). Zhu ve ark. (2018), yirmi beş parçadan oluşan bir buzdolabının sökölme aşamalarını incelemiştir. Yani, buzdolabının tüm bileşenlerini sökerek tamamen demontaj işlemi gerçekleştirmişlerdir. Diğer yandan, Yin ve ark. (2021), kırk yedi bileşenden oluşan bir dizüstü bilgisayar demontajı incelemiştir. Kırk yedi parçalı bu ürünün yalnızca kırk iki parçası başarıyla sökülebilmektedir (Liang ve ark., 2023).

Farklı endüstri alanlarına göre demontaj hatları düz, 2-yönlü, U şeklinde, paralel veya çoklu paralel hatlar şeklinde yerleşim göstermektedir. DHD literatüründe en sık çalışılan yerleşim türü düz yerleşimdir (Liang ve ark., 2022; Ren ve ark., 2020; Zhou & Bian,

2022). En çok çalışılan diğer yerleşim türü ise 2-yönlü yerleşimdir (Kucukkoc ve ark., 2020; K. Wang ve ark., 2019; Y. Zhang ve ark., 2022). Bir diğer yerleşim türü olan U şeklindeki yerleşim, DHD literatüründe bazı araştırmacılar tarafından çalışılmıştır (Guo ve ark., 2022; Li & Janardhanan, 2021; Li ve ark., 2023; K. Wang ve ark., 2020). DHD literatüründe son zamanlarda paralel yerleşim (Qin ve ark., 2023; Zhu ve ark., 2020) ve çoklu paralel yerleşim (Liang ve ark., 2023) türleri çalışılmıştır.

1.2. Çizelgeleme

Bir ürünün üretilmesi veya bir hizmetin sunumu için, işgörenin ne zaman ve nerede görevli olduğunun, gerekli faaliyetlerin zamanlamasının, üretime başlama ve üretimi tamamlama zamanlarının, çalışma zamanlarının belirli bölümlerinde hangi iş merkezinde hangi işin yapılacağını, iş merkezlerinin yeni işler için ne zaman hazır olacağını belirlenmesi işi çizelgelemedir (Topoyan, 2019).

Çizelgeleme sayesinde müşteri taleplerine karşılık verilebilmek, tedarik sürelerini azaltmak, esnekliği artırmak ve kapasiteyi etkin kullanmak mümkün olmaktadır.

İki çeşit çizelgeleme yapılabilmektedir: 1. İleriye doğru çizelgeleme, 2. Geriye doğru çizelgeleme (Topoyan, 2019).

İleriye doğru çizelgeleme, iş gereklilikleri belli olduğunda en kısa zamanda işe başlamak üzere çizelge hazırlanır. Bu yöntemle oluşturulan çizelge son teslim tarihini karşılamayabilir. Çoğunlukla müşteri siparişi üzerine başlayan, en kısa zamanda teslim edilmesi gereken işlerde kullanılır.

Geriye doğru çizelgelemede, işin son teslim tarihinden başlayarak geriye doğru işi oluşturan her faaliyetin zamanlamasının belirlenmesidir. En son faaliyetten başlayarak çizelge hazırlanır ve faaliyetlerin başlaması gereken zamanlar elde edilir. Çizelgelenen iş için gerekli kaynaklar elde olmayabilir.

Üretim sistemleri içinde çizelgeleme problemleri önemli bir yere sahiptir. Klasik çizelgeleme problemlerinde işlerin işlem zamanının sabit olduğu kabul edilmektedir. Oysa ki uygulamalara bakıldığında işlerin işlem zamanı işin tekrarı olmasına göre değişim gösterebilmektedir. Gerçek hayat problemlerinde işler öğrenme ve bozulma etkileri altında kalabilmekte ve işlerin işlem süreleri değişiklik gösterebilmektedir. İşlem

zamanlarında zaman ve pozisyona bağılı olarak artan bir fonksiyon göstermesi bozulma, azalan bir fonksiyon göstermesi halinde ise öğrenme olarak tanımlanmaktadır. Bu durum literatürde öğrenme ve bozulma etkisi olarak geçmektedir (Eren, 2014).

1.2.1. Öğrenme etkisi

Çizelgeleme problemleri üretim sistemlerinde önemli bir yere sahiptir. Klasik çizelgeleme problemlerinde, işlerin işlem sürelerinin sabit olduğu varsayılır. Ancak uygulamalara bakıldığında işlerin tekrarlanma durumuna göre işlerin işlem süreleri değişkenlik gösterebilmektedir. Gerçek hayatta, çalışanlar görevlerini tekrarladıkça işi daha kısa sürede tamamlayabilirler. Bu nedenle, bir işin gerçek işlem süresi, daha sonraki bir pozisyonda işlenirse daha kısa olacaktır. Bu, öğrenme etkisi olarak bilinir (Wang ve ark., 2008).

Wright (1936), üretim operasyonlarına dayalı öğrenme kavramını ilk olarak ele almıştır. Biskup (1999) ve Cheng & Wang (2000), öğrenme etkisini çizelgeleme problemlerinde ele almışlardır. Biskup (1999), pozisyon tabanlı bir öğrenme modeli önermiştir ve bunu tek bir makine çizelgeleme problemine uygulamıştır. Pozisyon tabanlı öğrenme etkisi çizelgeleme problemlerinde (Mosheiov, 2001; Pei ve ark., 2021; Rudek, 2011; Xu ve ark., 2016), Toksarı ve ark. (2010), montaj hattı dengeleme probleminde çalışmışlardır. Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi çizelgeleme problemlerinde nadiren kullanılmıştır (Cheng ve ark., 2009; Liang ve ark., 2019; Toksarı ve ark., 2022).

Pozisyona bağlı öğrenme etkisi

Çalışanlar yaptıkları işi tekrarladıkça öğrenirler ve daha kısa sürede işi tamamlarlar. Pozisyon tabanlı öğrenme etkisinde, işler pozisyonlarına bağılı olarak öğrenmeden etkilenirler. Bir iş istasyonunda bir iş ne kadar ileri pozisyona atanırsa, öğrenme oranı o kadar artmaktadır. Yani, pozisyon tabanlı öğrenme etkisi altında, bir işin işlem süresi istasyonda bulunduğu sıraya bağlıdır.

Logaritmik işlem süresi toplamına bağlı öğrenme etkisi

Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile bir işin işlem süresi kendisinden önceki pozisyonlara atanmış işlerin işlem sürelerinin logaritmik toplamına dayalı bir fonksiyon ile hesaplanmaktadır. İşin işlem süresi, başlangıçta belirtilen işlem

süresinden daha kısa gerçekleşmektedir. Çizelgedeki her işin işlem süresi, atandığı istasyonda kendisinden önce gerçekleşen işlere bağlıdır.

1.2.2 Bozulma etkisi

Bir işin işlem süresini artıracak şekilde etki eden durumlar bozulma olarak adlandırılmaktadır. Örneğin; üretim ortamı giderek daha rekabetçi hale geldikçe, müşterilere daha fazla ürün çeşitliliği sağlamak için tesisler daha kısa üretim sürelerine ve sık ürün değişikliklerine yönelmektedirler. Dolayısıyla, işçiler tamamen yetkin hale gelmeden görevler arasında daha fazla zaman harcamak, sürekli yeni görevle karşılaşmak zorundadır. Çalışanların bu ortamda yaşadıkları öğrenmeler ve unutmalar önemlidir. Bu çalışanlar genellikle, unutma olarak adlandırılan, düşük performansa neden olan ürün ve süreç değişikliği ile kesintiye uğramaktadır (Wang & Xia, 2005).

Bir iş kuyruğunda işlenmeyi beklerken veya makinede işlenirken işler bozulabilir. Dolayısıyla işin bozulma etkisi altındaki işlem süresi, işe başlama zamanına bağlı olarak artmaktadır. Bozulma etkisi ilk olarak Gupta & Gupta (1988) tarafından çizelgeleme problemine entegre edilmiştir. Haddehaneye girmeyi beklerken külçe sıcaklığının düşmesi, külçenin haddelemeden önce yeniden ısıtılmasını gerektirecektir. Yangınla mücadelede gecikme olursa yangını söndürmek için gereken süre artacaktır (Arık & Toksarı, 2020; Mazdeh ve ark., 2010). Kötüleşen havalarda veya artan karanlıkta bir şeyler aramak, kötü mekanik koşullar altında makineleri veya araçları onarmakta işlerin işlem sürelerini artıran bozulmalardır (Mosheiov, 1994).

Klasik DHD problemlerinde, işlerin işlem sürelerinin süreç boyunca değişmeyip sabit kaldığı varsayılmaktadır. Oysa ki, gerçek hayatta işler tekrarlandıkça öğrenilebilmekte ve çeşitli sebeplerden dolayı başlangıç zamanları gecikebilmektedir. Yani, işler gerçek hayatta öğrenme ve bozulma etkisi altında gerçekleşmektedir. Bu çalışmada, DHD problemleri öğrenme ve bozulma etkileri altında ele alınmış, işlerin işlem süreleri bu etkiler altında hesaplanmıştır.

2. BÖLÜM

YÖNTEM VE MATERYAL

2.1. Problem Tanımı

DLB probleminde amaç, talebi karşılarken demontaj hattındaki kaynakları en verimli şekilde kullanmaktır. Kaynakların verimli kullanılması, gerekli minimum demontaj iş istasyonu sayısının bulunması, demontaj görevlerinin öncelik ilişkileri dikkate alınarak iş istasyonlarına en uygun şekilde atanması ve demontaj hattı yerleşiminin iyileştirilmesidir (Güngör & Gupta, 2002). DHD problemleri, öğrenme etkileri ve bozulma etkisinin farklı kombinasyonları ile çözülmüştür.

Klasik pozisyon tabanlı öğrenme etkisi modelinde (Biskup, 1999), P_{ir} , i görevinin gerçek işlem süresidir ve çizelgede i işi r 'nci sıradadır. Bu durumda,

$$p_{i(r)} = p_i \times r^a \quad (1)$$

$$p_{i(r)} = p_i \left(1 + \sum_{l=1}^{r-1} \ln p(l)\right)^a \quad (2)$$

burada $a < 0$ öğrenme oranı ve p_i , i işinin temel işlem süresidir. Öğrenme etkisi indeks değeri, öğrenme etkisinin 2 tabanında logaritmik değerine eşittir. Denklem (1), pozisyona bağlı öğrenme etkisi altında i işinin gerçek işlem süresini hesaplamaktadır. Denklem (2), logaritmik işlem zamanı toplamına bağlı öğrenme etkisi altında i işinin gerçek işlem süresini hesaplamaktadır. p_l , i işinin atanmış olduğu r 'nci sıradan önceki sıralarda bulunan işlerin temel işlem süreleridir.

$$P_{ir} = p_i + \alpha t_r \quad (3)$$

Denklem (3)'de $\alpha > 0$, işe başlama zamanındaki gecikmeden kaynaklanan bozulma etkisinin indeksidir. Denklem (3), bozulma etkisi ile işlerin işlem sürelerini hesaplamaktadır. Denklem (4), görevlerin önce bozulma etkisinden sonra öğrenme

etkisinden etkilendiğini göstermektedir. Çünkü, bozulma etkisi işin başlama zamanı ile öğrenme etkisi ise işin pozisyonuna bağlıdır. Öğrenme t_r anında başlarken, görevin bozulması t_i zamanında devam etmektedir (Toksarı ve ark., 2010).

$$P_{i(r)} = (p_i + \alpha \times t_r) \times r^a \quad (4)$$

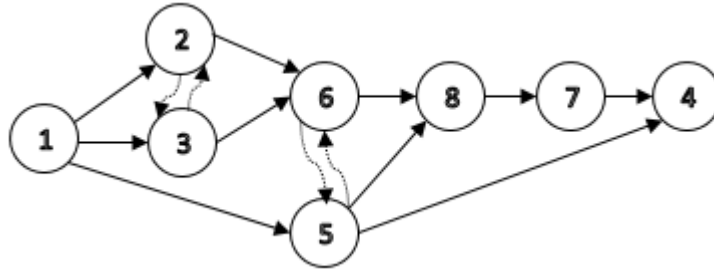
$$P_{i(r)} = (p_i + \alpha t_r) \times \left(1 + \sum_{l=1}^{r-1} \ln p(l) \right)^a \quad (5)$$

Burada, $t_i \geq 0$, i görevi için başlangıç zamanıdır. Denklem (4), pozisyon tabanlı öğrenme etkisi ile doğrusal bozulma etkisinin eş zamanlı uygulandığında, r 'nci sıraya atanan i işinin gerçek işlem süresini hesaplamaktadır. Denklem (5) ise logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile doğrusal bozulma etkisi eş zamanlı uygulandığında, r 'nci sıraya atanan i işinin gerçek işlem süresini hesaplamaktadır.

Gupta ve Güngör (2001) tarafından önerilen temel matematiksel modele dayanarak, DHD problemi için pozisyon tabanlı öğrenme, logaritmik işlem süreleri tabanlı öğrenme ve iş bozulması etkilerinin farklı kombinasyonları altında DHD problemi için matematiksel model geliştirilmiştir. Gupta ve Güngör (2001), 8 parçadan oluşan bir dizüstü bilgisayarın demontajı problemi üzerinde çalışmışlardır. Görevler ve parametreler Tablo 3'de sunulmuştur. DHD problemleri için çok önemli olan öncelik ilişkileri bu problem için Şekil 1'de verilmiştir.

Tablo 3. 8-parçalı DHD probleminin görev tanımları ve parametreler

Görev no	Görev Tanımı	İşlem süresi (sn)
1	PC'nin üst kapağının çıkarılması (TC)	14
2	Sabit sürücünün çıkarılması	10
3	Arka düzlemin çıkarılması (BP)	12
4	PCI kartların çıkarılması (PCI)	18
5	PCI kartların çıkarılması (PCI)	23
6	İki RAM modülünün çıkarılması (RAM)	16
7	Güç ünitesinin (PU) çıkarılması	20
8	Anakartın çıkarılması (MB)	36



Şekil 2. 8-parçalı DHD probleminin öncelik ilişkiler diyagramı

Yukarıdaki verilen veri seti için çevrim süresi 40'tır. Herhangi bir istasyonunun iş tamamlama süresi çevrim süresini aşmamalıdır. Gupta ve Güngör (2001), amaç fonksiyonu değerini yani minimum açılacak istasyon sayısını 4 olarak bulmuştur. DHD probleminin bazı özellikleri ve varsayımları aşağıda sunulmuştur:

- Her ürün tamamen demonte edilmelidir.
- Her sökme işleminden sadece bir parça demonte edilebilir.
- Tek ürün tipi demonte edilir.
- Sökme işlemi öncelik ilişkilerine göre yapılmalıdır.
- Herhangi bir iş istasyonunun iş tamamlama süresi, önceden belirlenmiş çevrim süresini aşmamalıdır.
- Parça sökme süreleri sabit ve deterministiktir.
- Her görev, tek bir iş istasyonunda yürütülmelidir.

2.2. Notasyon

İndisler

$i, k \in N$	işler ($i, k=1, 2, \dots, n$)
$(i, k) \in SP$	öncelik ilişkileri olan işler kümesi
$j \in M$	iş istasyonu ($j=1, 2, \dots, m$)
$r \in N$	istasyonda işin pozisyonu ($r=1, 2, \dots, n$)

Parametreler

α	öğrenme indeksi
α	bozulma indeksi
C	çevrim süresi

Karar Değişkenleri

y_{ij}	i işi j istasyonunda sökülüyorsa, 1 ; aksi halde 0
z_j	j istasyonu açılıyorsa, 1; aksi halde 0
x_{ijr}	i işi j istasyonunda r 'nci sıraya atanıyorsa, 1; aksi halde 0
b_{jr}	j istasyonunda r 'nci sıra doluysa, 1; aksi halde 0
t_{jr}	j istasyonunda r 'nci sıradaki işin işlem süresi
c_{jr}	j istasyonunda r 'nci sıradaki işin toplam işlem süresi
NWS	iş istasyonu sayısı

2.3. Öğrenme ve iş bozulmaları etkisi altında DHD problemleri için matematiksel model

Amaç fonksiyonu:

$$f = \text{Min} \sum_{j=1}^n z_j \quad (6)$$

Kısıtlar:

$$\sum_{j=1}^n y_{ij} = 1 \quad \text{for } \forall i \in n \quad (7)$$

$$\sum_{j=1}^n y_{ij} \leq z_j \times ct \quad \text{for } \forall i \in n \quad (8)$$

$$\sum_{r=1}^n c_{jr} \times x_{ajr} \times b_{jr} - \sum_{R=1}^n c_{jR} \times x_{kjR} \times b_{jR} \leq 0 \quad \text{for } \{a, k\} \quad (9)$$

$$\sum_{i=1}^n x_{ijr} \leq 1 \quad \text{for } \forall j \in M, r \in N \quad (10)$$

$$\sum_{r=1}^n x_{ijr} = y_{ij} \quad \text{for } \forall j \in M, i \in N \quad (11)$$

$$\sum_{r=1}^n (g_i \times x_{ijr} \times r^a \times b_{jr}) = t_{jr} \quad \text{for } \forall j \in M, r \in N, i \in n \quad (12a)$$

$$\sum_{r=1}^n (g_i \times x_{ijr}) \times \left(1 + \sum_{l=1}^{r-1} \ln(g_l \times x_{[j]l}) \right)^a = t_{jr} \quad \text{for } \forall j \in M, r \in N, i \in n (i \neq 1) \quad (12b)$$

$$\sum_{i=1}^n ((g_i + \alpha \times c_{j(r-1)} \times b_{jr}) \times x_{ijr}) = t_{jr} \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (12c)$$

$$\sum_{i=1}^n g_i \times x_{ijr} \times r^a + c_{j(r-1)} \times r^a \times \alpha \times b_{jr} = t_{jr} \quad \text{for } \forall j \in M, r \in N, i \in n (r \neq 1) \quad (12d)$$

$$\sum_{i=1}^n \left((g_i + \alpha \times c_{j(r-1)} \times b_{jr}) \times x_{ijr} \right) \times (1 + \sum_{l=1}^{r-1} (\ln g_l \times x_{ilr}))^a = t_{jr} \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (12e)$$

$$\sum_{i=1}^n g_i \times x_{ijr} \times b_{jr} = t_{jr} \quad \text{for } \forall i \in N, r = 1 \quad (13)$$

$$c_{jr} = c_{j(r-1)} + t_{jr} \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (14)$$

$$t_{jr} \times b_{jr} \leq t_{j(r-1)} \times b_{j(r-1)} \times M \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (15)$$

$$c_{jr} \times b_{jr} \leq c_{j(r-1)} \times b_{j(r-1)} \times M \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (16)$$

$$c_{jr} = t_{jr} \quad \text{for } \forall j \in M \text{ and } r = 1 \quad (17)$$

$$x_{Ij(r-1)} \leq \sum_{i=1}^n x_{ijr} \quad (I \neq i), (I = 1, \dots, N) \quad (18)$$

$$x_{ijr}, y_{ij}, b_{jr}, z_j \in \{0,1\} \quad (19)$$

Denklem (6)'da verilen amaç fonksiyonunda, belirli bir çevrim süresi kısıtı ile açılan iş istasyonu sayısının minimize edilmesi amaçlanmaktadır. Kısıt (7) ve Kısıt (11)'de, bir görevin yalnızca bir istasyona ve istasyonda yalnızca bir pozisyona atanabilmesini sağlamaktadır. Kısıt (8), bir istasyona atanan işlerin görev sürelerinin toplamının çevrim süresini aşmamasını sağlamaktadır. Kısıt (9), işlerin öncelik ilişkilerine göre istasyona atanmasını sağlamaktadır. Kısıt (10), herhangi bir işin herhangi bir istasyonda tek bir pozisyona atanabileceğini belirtmektedir. Bir model çalıştırılırken Kısıt (12a)-(12e) kısıtlarından sadece bir tanesi modele eklenir. Kısıt (12a), herhangi bir görev ilk sıradan farklı bir sıraya atandığında işlem süresini pozisyon tabanlı öğrenme etkisi altında hesaplamaktadır. Kısıt (12b), herhangi bir görev ilk sıradan farklı bir sıraya atandığında işlem süresini logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında hesaplamaktadır. Kısıt (12c), herhangi bir görev ilk sıradan farklı bir sıraya atandığında işlem süresini bozulma etkisi altında hesaplamaktadır. Kısıt (12d), herhangi bir görev ilk sıradan farklı bir sıraya atandığında işlem süresini pozisyon tabanlı öğrenme ve bozulmanın eş zamanlı etkileri altında hesaplamaktadır. Kısıt (12e), herhangi bir görev ilk sıradan farklı bir sıraya atandığında işlem süresini, logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulmanın eş zamanlı etkileri altında hesaplamaktadır. Kısıt (13), herhangi bir görev istasyondaki ilk pozisyona atandığında işlem süresini hesaplamaktadır. Kısıt (14), j istasyonunun görev tamamlama süresini hesaplamaktadır. Kısıt (15), j istasyonundaki ilk konum dışındaki herhangi bir r pozisyonundaki görevin işlem süresi

ile kendisinden önceki pozisyonundaki görevin işlem süresi arasındaki ilişkiyi sağlamaktadır. Kısıt (16) ve Kısıt (17), j istasyonundaki r 'nci sırada yer alan görevin tamamlanma süresi ile $(r-1)$ 'nci sırada yer alan görevin tamamlanma süresi arasındaki ilişkiyi sağlamaktadır. Kısıt (18), herhangi bir iş istasyonunda, bir görevden önce yalnızca bir görevin gelebileceğini tanımlar. Son olarak, Kısıt (19) ikili değişkenleri tanımlamaktadır.

Yukarıda verilen matematiksel modelde, öğrenme ve bozulma etkileri altında DHD probleminin, öğrenme ve bozulma etkilerinin farklı kombinasyonları ile işlerin işlem sürelerini hesaplayan fonksiyonlar Kısıt (12a-12e) olarak belirtilmiştir. Hangi etki altında hesaplama yapılıyorsa, işin istasyondaki işlem süresini hesaplarken o etkinin denklemi kullanılmıştır.

Uygulama aşamasında küçük boyutlu problemler için Lingo 11 paket programı kullanılarak matematiksel model ile, büyük boyutlu problemler için genetik algoritma (GA), hibrit genetik algoritma (HGA), parçacık sürüsü optimizasyonu (PSO) ve PSO-GA hibrit yaklaşımlarıyla çözümler elde edilmiştir.

2.4. Öğrenme ve bozulma etkileri altında DHD problemlerinin çözümü için önerilen metasezgisel yaklaşımlar

Sıra bağımlı demontaj hattı problemi, NP zor bir problemidir (Kalayci ve ark., 2016). Dolayısıyla problemin boyutu arttıkça karmaşıklığı da önemli ölçüde artmakta ve probleme en iyi çözümü bulmak zorlaşmaktadır. Bu çalışmada, literatürde bulunan 15 veri seti kullanılmıştır. Daha büyük veri setleri için lineer optimizasyon programı ile sonuç elde edilemeyeceğinden, öğrenme ve bozulma etkisi altında DHD problemlerine optimum veya optimuma yakın çözümler elde etmek için farklı metasezgisel yaklaşımlar önerilmiştir. Bu yaklaşımlar; GA, HGA, PSO ve PSO-GA hibrit yaklaşımlarıdır.

2.4.1. Genetik Algoritma

Genetik algoritma, doğal evrimsel süreçlerden yararlanır. Sanal ortamlarda, canlı organizmaların belirli özellikleri taklit edilerek modeller oluşturulmuş ve bu modeller kullanılarak çözüm yolları denenmeye çalışılmıştır. Bu modellerden biri olan genetik algoritma, canlıların genetik yapıları ve çevreye uyumlarına bakılarak oluşturulmuştur (İşçi & Korukoğlu, 2003).

Genetik algorithmada uygulanan temel adımlar şöyledir (Levitin ve ark., 2006):

- Problemin çözümünü temsil eden çözüm adayı kromozomları rastgele üretilir,
- Her bir çözüm adayının uygunluğu değerlendirilir,
- Rastgele iki çözüm adayı seçilir ve bazı ebeveyn kromozom yapılarını değiştirmeden çocuk kromozomları üretmek için çaprazlama yöntemi uygulanır,
- Çocuk kromozomun yapısında küçük değişikliklere neden olan mutasyon yöntemi uygulanır. Bu yöntem çözüm uzayında sıçramaya izin verir ve yerel bir optimuma yaklaşmayı önler.
- Çocuk kromozomların amaç fonksiyonu değerleri hesaplanır,
- Çözüm değeri daha iyi olan çocuk popülasyona katılır ve daha kötü sonuca sahip olan çocuklar elenir.

Başlangıç popülasyonunu oluşturma

Başlangıç kromozomu, Tablo 4'deki örnekte görüldüğü gibi problem büyüklüğü dikkate alınarak rastgele oluşturulmuştur. DHD problemini çözmedeki ilk hedef, uygulanabilir bir dengeleyici çözüm stratejisi oluşturmaktır. Çevrim süresini aşmadan iş istasyonlarının birbirini ardına doldurulmasını sağlayan istasyon odaklı bir yöntem kullanılmaktadır (Ding ve ark., 2010). İlk iş istasyonundan başlanarak atama yapılır. Rastgele sıralanan işler, ilk istasyondan başlanarak sırayla atanır. Bu atamalar yapılırken, öncelik ilişkileri dikkate alınmaktadır. Belirli bir sıraya atanan işin, kendisinden önce aynı istasyona atanmış işlerin öncülü olmaması gerekmektedir. Sıradaki, atanacak olan iş, bu istasyona kendisinden önce atanan herhangi bir işin öncülüyse, bu durumda, iş bu istasyona atanamaz ve yeni bir istasyon açılır. Atama işlemi sırasında öncül ilişkisine ek olarak istasyona atanan işlerin işlem sürelerinin toplamı hesaplanmalı ve atanan işlerin işlem sürelerinin toplamı, çevrim süresini aştığı anda iş bir sonraki istasyona atanmalıdır. Yeni açılan istasyona işler sırasıyla öncelik ilişkileri ve çevrim süresi kısıtı göz önünde bulundurularak atanmaya devam eder. Bu süreç tüm işler istasyonlara atanana kadar tekrarlanır.

8 parçadan oluşan DHD problemi düşünüldüğünde rastgele oluşturulmuş birkaç iş sırası Tablo 4'e görülmektedir.

Tablo 4. Rassal başlangıç kromozomu iş sırası örnekleri

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

Uygunluk fonksiyonu ve seçim yöntemi

Genetik algoritma en iyi çözümü veren kromozomu yani çözüm adayını bulmaya çalışır. Uygunluk fonksiyonu, bir bireyin arama uzayındaki performansının bir ölçüsüdür. Bizim çalışmamızda uygunluk fonksiyonu amaç fonksiyonudur. Denklem (5)'de de görüldüğü gibi uygunluk fonksiyonu açılan istasyon sayısı değeridir.

İki çeşit seçim yöntemi bulunmaktadır. Bunlar rulet çarkı seçim yöntemi ve turnuva seçim yöntemidir.

Rulet çarkı seçim yöntemi

Bu yöntemde, tüm başlangıç popülasyonu için uygunluk değerleri hesaplanır ve bu değerler 0 ile 1 arasında değerler alacak şekilde normalize edilir. Bu şekilde her bir çözüm adayının seçilme olasılıkları elde edilmiş olacaktır. Her bir çözüm adayı seçilme olasılığı oranında rulet çarkında yer almaktadır. Dolayısıyla, rulet çarkı üzerinde olasılık değeri yüksek olan adayın seçilme olasılığı da daha yüksektir.

Turnuva seçim yöntemi

Başlangıç popülasyonunda daha iyi uygunluk değerine sahip olan kromozomlar seçilirler. Dolayısıyla yeni popülasyon, başlangıç popülasyonunun kötü sonuç veren bireylerinden arındırılmış olmaktadır. Turnuva seçimi yöntemi hem daha kolay, hem de optimum çözüme daha hızlı ulaşması açısından önemli bir yöntemdir (Paksoy & Uzun, 2008).

Bu çalışmada seçim yöntemi olarak turnuva seçim yöntemi kullanılmıştır. Başlangıç popülasyonunu oluşturan tüm çözüm adayları için uygunluk değeri hesaplanmış ve daha iyi sonuca sahip olanlar bir sonraki topluma aktarılırken, kötü sonuç verenler elenmiştir.

Çaprazlama

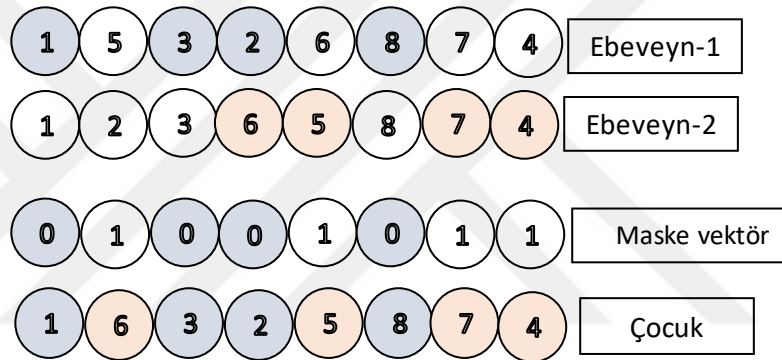
Seçilen ebeveynlerden çocuk kromozomlar olarak bilinen yeni çözüm adaylarını üretmek amacıyla yapılmaktadır. Seçilen iki ebeveynde, çaprazlama noktası dikkate alınarak ebeveyn kromozomlarının çapraz bir şekilde yer değiştirmesidir. Böylece ebeveyn

kromozomundaki genler kalıtsallaştırılmakta ve kromozomlar arasındaki genetik bilgi değişerek toplum iyileşmektedir.

Bu çalışmada öğrenme ve bozulma etkisi altında DHD problemi çalışılırken genetik algoritmanın çaprazlama aşamasında öncelikli korunumlu çaprazlama yöntemi (precedence preservative crossover method) kullanılmıştır (Kalayci ve ark., 2016).

Öncelikli korunumlu çaprazlama yöntemi

Bu çaprazlama yöntemi öğrenme ve iş bozulma etkisi altında DHD probleminin çözümü için kullanılmıştır.



Şekil 3. Öncelikli korunumlu çaprazlama yöntemi (Precedence preservative crossover method)

Bu çaprazlama yöntemini uygulamak için gerekli öğeler Şekil 3’de görülmektedir. Öncelikli koruyucu çaprazlama yönteminde, Ebeveyn-1 ve Ebeveyn-2’nin yanı sıra maske vektör vardır. Bu maske vektör, rastgele oluşturulmuş $[0,1]$ değerlerinden oluşmaktadır. Maske vektöre göre çocuk kromozom oluşturulurken, maske vektörde 0 olan genlerdeki işler Ebeveyn-1’den, 1 olan genlerdeki görevler Ebeveyn-2’den alınarak çocuk kromozomda ilgili genlere yerleştirilir. Böylece yavru kromozomdaki iş sırası Şekil 3’de görüldüğü gibi elde edilmiş olur.

Mutasyon

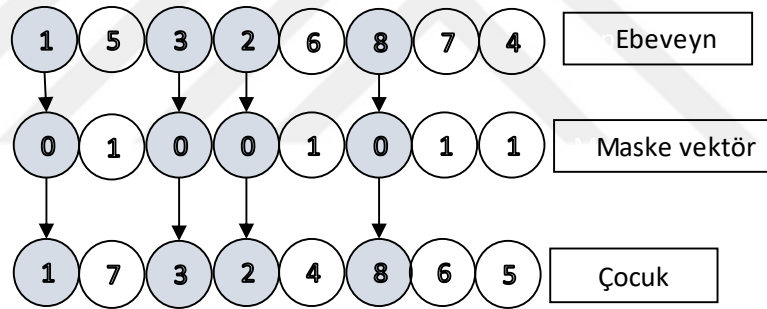
Mutasyon ise çaprazlamadan sonra üretilen her bir çocuğa uygulanan genetik operatördür. Mutasyonun amacı arama uzayında her noktanın denenmesine olanak sağlamaktır. Çocukların ebeveynlerden farklı olmasını ve popülasyonda çeşitliliğin artmasını sağlar.

Bu çalışmada öğrenme ve bozulma etkisi altında DHD problemi çalışılırken genetik algoritmanın mutasyon aşamasında öncelikli düzgün operatör yöntemi (Precedence uniform operator method) kullanılmıştır (Kalayci ve ark., 2016).

Öncelikli düzgün operatör yöntemi (Precedence uniform operator method)

Bu mutasyon yöntemi öğrenme ve bozulma etkisi altında DHD probleminin çözümü için kullanılmıştır.

Bu çaprazlama yöntemini uygulamak için gerekli öğeler Şekil 4'te görülmektedir. Bir ebeveyn vektörün yanı sıra rassal olarak $[0,1]$ 'den oluşan bir maske vektör oluşturulur. Maske vektörde 0 olan genlere karşılık gelen işlerin konumları korunurken, değeri 1 olan genlere karşılık gelen işler için yeni bir alt dizi oluşturulur. Yani bu işler rassal olarak tekrar sıralanır ve bu yeni sıraya göre çocuk kromozomdaki boş vektörlere sırasıyla yerleştirilir (bkz Şekil 5).



Şekil 4. Öncelikli düzgün operatör yöntemi (Precedence uniform operator method)

2.4.2. Hibrit Genetik Algoritma (HGA)

HGA yönteminde, genetik algoritmanın başlangıç çözümü aşamasında bir yerel arama yöntemi kullanılmıştır. Genetik algoritmanın çaprazlama ve mutasyon aşamalarında kullanılan ebeveyn popülasyonun daha seçkin bireylerden oluşturulması amaçlanmıştır. Bu çalışmada kullanılan yerel arama yöntemi ilk kez, Gonçalves ve Almeida (2002) tarafından önerilmiştir. Tablo 5'de bu yöntemi uygularken atılması gereken adımlar gösterilmektedir. Her çözüm adayı için ardışık iş istasyonlarından rastgele seçilen işler yer değiştirir. Yerel arama yönteminin amacı son istasyondaki toplam işlem süresini azaltmaktır. Böylece son istasyondaki iş sayısı sıfıra indirilerek amaç fonksiyonu değeri iyileştirilecektir. Bu yerel arama yönteminin yalancı kodu şu şekildedir:

Tablo 5. Yerel arama yönteminin yalancı kodu (Gonçalves and Almedia, 2002)

```

FOR workstation k=1,2,...n-1
{
    REPEAT
    {
        Let LEFT (k+1)=Set of operations in work-station k+1
        can be moved to work-station k.

        Move, in decreasing order of their processing times,
        as many as possible operations in LEFT (k+1) to work-station k.

        Update LEFT (k+1)

        Let RIGHT (k)=Set of operations in work-station k that
        can be moved to work-station k+1.

        Exchange, in decreasing order of  $t_j - t_j > 0$ , as many as possible pairs of
        exchangeable operations (i, j) with  $i \in \text{RIGHT}(k)$  and  $j \in \text{LEFT}(k+1)$ .
    }
    UNTIL no move and exchange of operations is possible
}
END

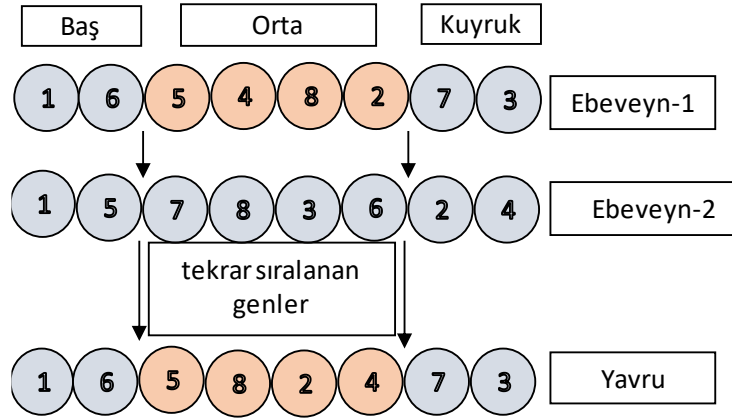
```

HGA yönteminde, çaprazlama ve mutasyon aşamasında, klasik GA yönteminde kullanılan yöntemlerden farklı yöntemler tercih edilmiştir. HGA uygulamasında, uygunluk fonksiyonu seçim yöntemi olarak turnuva seçim yöntemi kullanılmıştır. Çaprazlama aşamasında ise yeniden sıralama çaprazlama yöntemi kullanılmıştır.

Parçaları yeniden sıralama çaprazlama yöntemi

Bu çaprazlama yöntemi, öğrenme ve bozulma etkileri altında DHD problemlerinin HGA ile çözümünde kullanılmıştır.

Her ebeveyni baş, orta ve kuyruk olarak 3 parçaya bölmek için rastgele iki nokta seçilir. Ebeveyn-1 ve Ebeveyn-2, bir çocuk oluşturmak için yeniden birleştirilir. Çocuk, Ebeveyn-1'den baş ve kuyruğu alırken, kalan genler Ebeveyn-2'deki sıra ile çocuğa aktarılır (Şekil 5'te görüldüğü gibi). Ayrıca diğer çocuk, Ebeveyn-2'den baş ve kuyruğu alır ve kalan genler, Ebeveyn-1'deki sırasına göre bu çocuğa aktarılır. DHD problemlerinde öncelik ilişkileri önemlidir, yeni kromozomlar için uygunluk değerleri bulunurken dikkatli olunmalıdır.

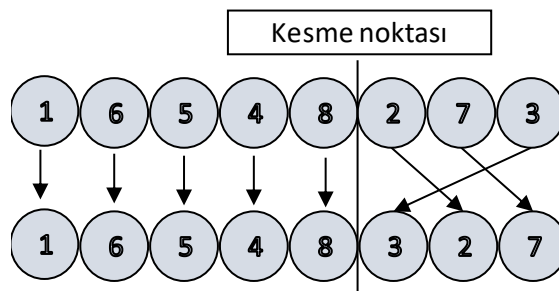


Şekil 5. Parçaları yeniden sıralama çaprazlama yöntemi (Fragment reordering crossover method)

HGA'nın mutasyon aşamasında ise sağda bir nokta operatör mutasyon yöntemi (One point right operator mutation method) kullanılmıştır.

Sağda bir nokta operatör mutasyon yöntemi (One point right operator mutation method)

Bu mutasyon yöntemi, öğrenme ve bozulma etkileri altında DHD problemlerinin HGA ile çözümünde kullanılmıştır.



Şekil 6. Sağda bir nokta operatör mutasyon yöntemi (One point right operator mutation method)

Bu yöntemde mutasyon oranı dikkate alınarak rastgele bir kesme noktası seçilir. Seçilen noktanın sağındaki genler rastgele yeniden sıralanır. Kesme noktasının solundaki genler ise aynı konumlarını korurlar (bkz. Şekil 6).

2.4.3 Parçacık Sürüsü Optimizasyonu (PSO)

PSO, Kennedy ve Eberhart (1995) tarafından geliştirilen ve kuş sürüsü, balık sürüsü gibi sürü teorisine dayanan bir evrimsel algoritmadır. Kuş ve balık sürülerinin, yerini bilmedikleri yiyecekleri ararken izledikleri yollardan esinlenilmiştir. PSO'nun temel amacı, popülasyon bireyler arasındaki sosyal bilgi paylaşımını iyileştirmektir. Popülasyonu değerlendirmek için uygunluk değerlerine sahip olan popülasyon tabanlı bir optimizasyon aracıdır. Optimal çözüme ulaşabilmek için popülasyon her iterasyonda güncellenir. Diğer metasezgisel yaklaşımlar gibi optimum noktaya ulaşmayı garanti etmez, ancak optimum veya optimuma en yakın değeri bulmaya çalışmaktadır. PSO çözümü daha hızlı bulduğu ve daha az parametre gerektirdiği için diğer algoritmalara göre üstünlüğü vardır (Cakar & Koker, 2015).

PSO, problem için farklı çözüm alternatifi üretmekte olan parçacık adı verilen çözüm adaylarından oluşmaktadır. Parçacıkların oluşturduğu yapıya ise sürü denilmektedir. Başlangıç popülasyonunu temsil eden sürüdeki tüm parçacıklar rastgele üretilmektedir ve problem için uygun çözümü aramaktadırlar. Her parçacığın konum (x) ve hız (v) vektörü olmak üzere iki vektörel bileşeni vardır. Konum vektörü parçacığın konum bilgisini tutarken, hız vektörü parçacığın yer değiştirme miktarını ve yönünü kaydetmektedir. Her iterasyonda konum ve hız vektörleri güncellenmektedir. Bir parçacığın yeni hız değeri, önceki iterasyondaki hızı, sürünün genel deneyimi ve rastgelelik kullanılarak hesaplanmaktadır. Parçacığın, çözüm uzayında hangi yönde ve hangi hızda hareket edeceği kendi en iyi konumuna ve tüm komşu parçacıkların en iyi konumlarına bağlıdır. Parçacık optimuma ulaşırken, bir sonraki iterasyondaki konumunu, kendi en iyi konumuna ve sürünün en iyi uygunluk değerine sahip parçacığın pozisyonuna göre hesaplar. Çözüm uzayında rastgele bir pozisyonla başlayan parçacık, her iterasyonda kendi en iyi pozisyonuna ve sürünün en iyi pozisyonuna göre adım adım hareket ederek optimum sonuca ulaşmaya çalışır.

$$v_i^{k+1} = w \times v_i^k + c_1 \times rand1(pbest_i^k - x_i^k) + c_2 \times rand2(gbest^k - x_i^k) \quad (20)$$

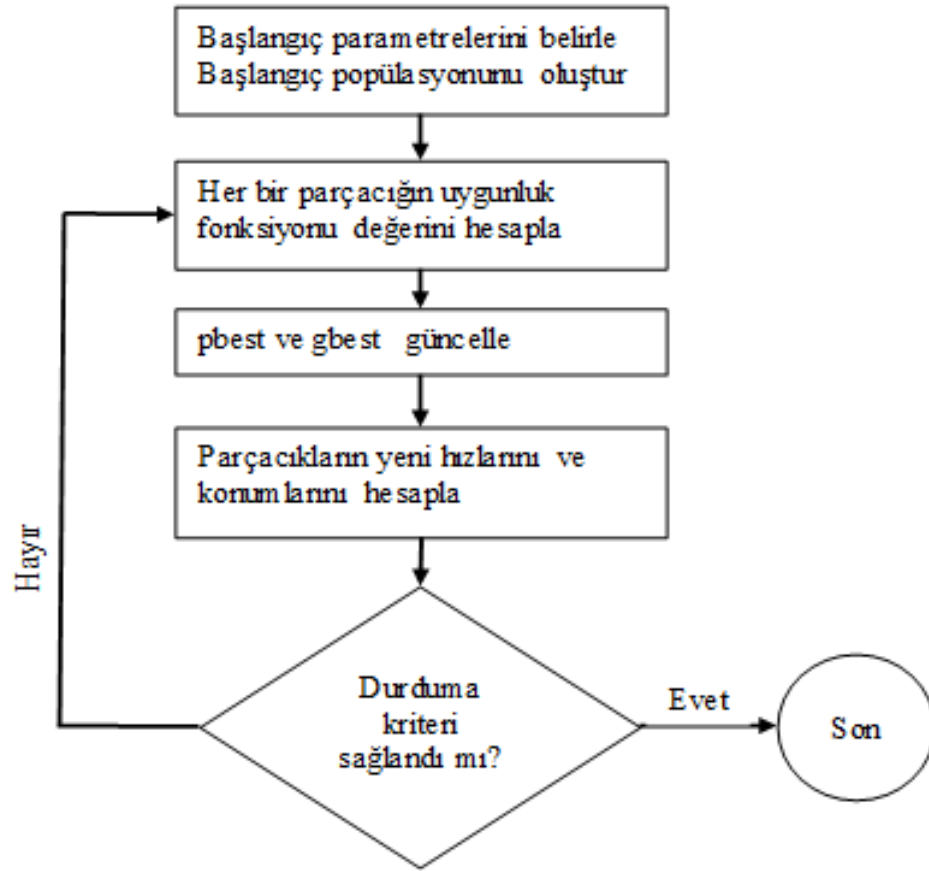
$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (21)$$

Burada, x_i^k ve v_i^k , i parçacığının k 'nci iterasyondaki konumunu ve hızını göstermektedir. c_1 kendi kendine öğrenme faktörü iken, c_2 sosyal öğrenme faktörüdür. c_1 ve c_2 değerleri sabittir ve 2 kabul edilmiştir (Garg, 2016; Kennedy & Eberhart, 1995). w atalet ağırlık

değeridir. $pbest_i^k$, i parçacığının k 'nci iterasyona kadar elde ettiği en iyi değer iken, $gbest^k$ ise k 'nci iterasyona kadar tüm sürüde bulunan en iyi değerdir.

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{\text{iterasyon sayısı}} \right) \times \text{mevcut iterasyon} \quad (22)$$

$0.4 < w < 0.9$ aralığında bir değerdir ($w_{min} = 0.4$ & $w_{max} = 0.9$).



Şekil 7. PSO algoritması akış diyagramı

Şekil 7’de PSO algoritmasının adımları sunulmuştur. Parçacıkların başlangıç konumları rastgele oluşturulur. Başlangıçta parçacıklar herhangi bir hıza sahip değillerdir. Dolayısıyla $v_0 = 0$ alınır. Başlangıç popülasyonu için uygunluk değerleri hesaplanır. Böylece ilk ($pbest_i^k$) ve ($gbest^k$) bulunmuş olur. Daha sonra, popülasyonu oluşturan parçacıkların hızları ve konumları hesaplanır. Parçacıklar konum vektörü (x_i^k) değerlerine göre küçükten büyüğe doğru sıralanır (Cakar & Koker, 2015). Ardından sürüdeki bütün parçacıkların uygunluk değerleri hesaplanır. Uygunluk değeri, bu çalışma için, aynı zamanda amaç fonksiyonu olan açılan istasyon sayısı değeridir. Her iterasyonda elde edilen değerler, tüm parçacıkların önceki iterasyonun en iyi ($pbest_i^k$) değerleri ile

karşılaştırılır, yeni parçacığın değeri daha iyi ise yer değiştirilir. Daha sonra, evrensel en iyi değerle ($gbest^k$) karşılaştırılır, yeni parçacığın değeri daha iyi ise evrensel en iyi olarak atanır. Parçacığın hız ve konum değerleri güncellenir. Durdurma kriteri sağlanana kadar bu aşamalar tekrar edilir.

2.4.4. Parçacık Sürüsü Optimizasyonu – Genetik Algoritma (PSO-GA) Hibrit Yaklaşımı

Öğrenme ve bozulma etkileri altında DHD problemlerinin çözümünde PSO-GA hibrit yaklaşımı da kullanılmıştır. Bölüm 2.4.1’de sunulan klasik GA yaklaşımı ile Bölüm 2.4.3’de sunulan PSO algoritması birlikte kullanılmıştır. GA yaklaşımının uygulanması aşamasında, başlangıç popülasyonu oluşturulduktan sonra çözüm adayı kromozomlara, PSO yaklaşımı ile çözümler elde edilmiştir. Rassal bir şekilde oluşturulan başlangıç popülasyonundan, PSO algoritmasının uygulanması ile daha seçkin ebeveynlerin oluşması sağlanmıştır. Böylece, GA’nın çaprazlama ve mutasyon aşamaları daha seçkin ebeveynler üzerinde yapılmıştır.

3. BÖLÜM

BULGULAR

Bu çalışmada, pozisyon tabanlı ve logaritmik işlem süreleri toplamı öğrenme etkileri ile bozulma etkisi altında DHD problemleri ele alınmıştır. DHD literatüründe, öğrenme ve bozulma etkilerinin çalışıldığı ilk çalışmadır. Küçük boyutlu, literatürün en temel örneği olan 8 görevden oluşan demontaj örneği, oluşturulan matematiksel modellerin doğruluğunu kanıtlamak için Lingo 11 paket programı ile çözdürülmüştür. Orta ve büyük boyutlu problemler ise GA, HGA, PSO ve PSO-GA hibrit yaklaşımı ile çözülmüş, elde edilen amaç fonksiyonu değerleri karşılaştırılmıştır. Metasezgisel yaklaşımların algoritmaları ise C# yazılım dili kullanılarak kodlanmıştır. Böylece, öğrenme ve bozulma etkileri ile DHD problemlerinde açılan istasyon sayılarındaki değişim irdelenmiş ve farklı yaklaşımların belirli iterasyonlar için gösterdikleri performanslar karşılaştırılmıştır. Tüm çalışmalar, Intel Core (TM) i5-5200U CPU, 2.20 GHz, 8 GB RAM özellikli bilgisayar kullanılarak gerçekleştirilmiştir.

3.1. Pozisyon tabanlı ve logaritmik işlem süreleri toplamı tabanlı öğrenme etkileri altında DHD problemi

Bu bölümde, DHD problemleri, pozisyon tabanlı ve logaritmik işlem süreleri toplamı tabanlı öğrenme etkileri altında incelenmiştir. Her bir öğrenmenin problem üzerindeki etkisi ayrı ayrı çalışılmıştır ve işlerin işlem sürelerindeki değişiklikler incelenmiş ve karşılaştırılmıştır. Sadece öğrenme etkileri çalışıldığı için sık aralıklı olacak şekilde 5 farklı öğrenme oranı kullanılmıştır. Bunlar: 0.90, 0.85, 0.80, 0.75 ve 0.70'dir. Bahsedilen öğrenme etkisi değerlerinden bazıları (0.80, 0.70) çizelgelemede öğrenme literatüründe sıklıkla kullanılmaktadır (Biskup, 1999; Toksarı ve ark., 2010).

8 parçalı DHD problemi için hesaplama sonuçları

8 parçalı veri seti, DHD literatüründe en sık kullanılan veri kümesidir (Tablo 3). Bu veri seti için, DHD literatüründe hiçbir etki göz önünde bulunmadığı durumda açılacak istasyon sayısı 4 olarak bulunmuştur. Görevlerin iş istasyonlarına atanması şu şekildedir: Görev-1-2-3, İş İstasyonu-1'e, Görev-5-6, İş İstasyonu-2'ye, Görev-7-4 İş İstasyonu-3'e, Görev-8 İş İstasyonu-4'e atanır. İstasyonların işlem tamamlama süreleri ise sırasıyla; 36, 39, 38 ve 36 olmuştur.

Table 6. Pozisyon ve logaritmik işlem süresi toplamı tabanlı öğrenme etkileri altındaki 8 parçalı DHD problemi için Lingo 11 ile elde edilen sonuçlar

	Öğrenme Oranı					
	0.00	0.90	0.85	0.80	0.75	0.70
LISTTÖE	4	4	3	3	3	2
PTÖE	4	4	4	4	3	3

***LISTTÖE**: Logaritmik işlem süresi toplamı tabanlı öğrenme etkisi, **PTÖE**: Pozisyon tabanlı öğrenme etkisi

Tablo 6' da pozisyon ve logaritmik işlem süreleri toplamı tabanlı öğrenme etkileri altında hesaplanan amaç fonksiyonu değerleri görülmektedir. Öğrenme oranı değeri düştükçe açılan istasyon sayısı değeri azalmaktadır. Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında, öğrenme oranları sırasıyla 0.90-0.85-0.80-0.75-0.70 iken açılan istasyon sayıları sırasıyla 4-3-3-3-2 olarak bulunmuştur. Öğrenme oranı 0.80 olduğunda, istasyon sayısında %25 oranında azalma görülmüştür. 8 görevli DHD problemi pozisyon tabanlı öğrenme etkisi altında çözüldüğünde ise, öğrenme oranları sırasıyla 0.90-0.85-0.80-0.75-0.70 iken, açılan istasyon sayıları sırasıyla 4-4-4-3-3 olarak bulunmuştur. Bu DHD probleminde öğrenme etkisi logaritmik işlem süreleri toplamına göre belirlendiğinde, amaç fonksiyonunda daha çok iyileşme görülmektedir. Öğrenme oranı 0.80 iken logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile 3 istasyon açılması yeterli iken, pozisyon tabanlı öğrenme etkisi ile hala 4 istasyon açılması gerekli bulunmaktadır (Tablo 6).

8 görevden oluşan DHD probleminin iki farklı öğrenme etkisi ile hangi işin hangi istasyonda hangi sıraya atandığı Tablo 7'de görülmektedir. Her iki öğrenme etkisi altında da öğrenme oranı azaldıkça açılan istasyon sayısı değeri iyileşmektedir. Bu veri seti için logaritmik işlem süresi toplamı tabanlı öğrenme etkisi daha etkin olmuştur. Pozisyon tabanlı öğrenmeye göre işin yapılması için daha az istasyon gerekmektedir. Öğrenme oranı 0.90 olduğunda işin yapılma süresindeki azalma diğer oranlara göre daha düşük

miktarda olduğu için açılan istasyon sayısı her iki öğrenme etkisi altında da değişiklik göstermemiştir. Yani öğrenme etkileri dikkate alınmadığı durumda 4 istasyon açılması gerekirken, öğrenme oranı 0.90 olduğu durumda da 4 istasyon açılması gerektiği bulunmuştur (Tablo 7). Öğrenme oranı 0.80 olduğunda, pozisyon ve logaritmik işlem süresi toplamı tabanlı öğrenme etkileri altında bulunan iş istasyonları, sırasıyla, 3 ve 4 tür.

Tablo 7. 8 parçalı DHD probleminde istasyonlardaki optimum iş sıraları

Öğrenme oranı	Öğrenme etkisi	Optimal sıra	İş istasyonu sayısı
0	LISTTÖE	8, 5-6, 3-4-2, 7-1	4
	PTÖE	7-2, 6-5, 1-3-4, 8	4
0.90	LISTTÖE	5-7, 2-3-6, 1-4, 8	4
	PTÖE	5-6, 1-7-3, 8, 4-2	4
0.85	LISTTÖE	3-8, 2-5-4, 1-6-7	3
	PTÖE	2-7, 4-1-6, 3-5, 8	4
0.80	LISTTÖE	7-2-5, 3-8, 1-6-4	3
	PTÖE	2-3-5, 1-6-4, 8, 7	4
0.75	LISTTÖE	7-1-5, 2-8-3, 6-4	3
	PTÖE	4-2-6, 1-5-7, 3-8	3
0.70	LISTTÖE	1-8-4, 3-2-5-7-6	2
	PTÖE	1-6-4, 3-5-7, 2-8	3

8 parçalı bu DHD problemi için öğrenme göz ardı edildiğinde 4 istasyon açılması gerekirken, öğrenme oranı 0.70 olduğunda logaritmik işlem süresi tabanlı öğrenme etkisi altında 2 istasyon açılması yeterlidir.

Yukarıda Tablo 6 ve Tablo 7’de sonuçlarına yer verdiğimiz öğrenme etkileri altında DHD probleminde, işlem sürelerinin hesaplamalarının nasıl yapıldığını gösteren detaylı matematiksel işlemler, Tablo 8’de yer almaktadır. Bir istasyondaki her bir iş için işlem süreleri hesaplanmıştır. Bu hesaplamalar yapılırken, logaritmik işlem süreleri tabanlı öğrenme etkisi altında herhangi bir istasyona atanan bir işin işlem süresi, kendisinden önce aynı istasyona atanan işlerin işlem sürelerinin toplamının e tabanındaki logaritmik değeri kullanılarak bulunmaktadır. Yani, bir işin işlem süresi kendisinden önceki sıralara atanmış işlerin işlem sürelerine bağlıdır. Pozisyon tabanlı öğrenme de işin işlem süresi, istasyonda atandığı sıraya bağlıdır. Bir iş ne kadar geç işlem görürse, daha fazla azalır. Çünkü, işlem sırası kendisine gelene kadar öğrenme sürekli olarak devam etmektedir.

8 parçalı DHD probleminde, öğrenme oranı 0.70 iken, logaritmik işlem süresi toplamı tabanlı öğrenme etkisi altında işlerin istasyonlara ataması gerçekleştiğinde, 7 numaralı iş İkinci istasyonda 4 nolu sıraya atanmıştır. İşin temel işlem süresi 20 birim zaman iken, logaritmik işlem süresi toplamı tabanlı öğrenme etkisi altında 7.1 birim zamanda tamamlanmaktadır. Benzer şekilde pozisyon tabanlı öğrenme etkisi altında işler istasyonlara atandığında 8 nolu iş 3 nolu İstasyonda 2. sıraya atanmıştır ve işlem süresi 36 dan 25.2' ye düşmüştür. Tablo 8'de de görüldüğü gibi öğrenme etkisi dikkate alındığında işlerin işlem süreleri önemli oranda azalmaktadır.

Tablo 8. PTÖE ve LISTTÖE altında görevlerin işlem süreleri

Öğrenme etkisi	İşlem İndeksi	İşlem süresi
LISTTÖE	p ₁₁₁	14
	p ₈₁₂	$36*(1+\ln(14))^{-0.514}=18.53$
	p ₄₁₃	$18*(1+\ln(14)+\ln(18.53))^{-0.514}=6.84$
	p ₃₂₁	12
LISTTÖE	p ₂₂₂	$10*(1+\ln(12))^{-0.514}=5.3$
	p ₅₂₃	$23*(1+\ln(12)+\ln(5.3))^{-0.514}=9.9$
	p ₇₂₄	$20*(1+\ln(12)+\ln(5.3)+\ln(9.9))^{-0.514}=7.1$
	p ₆₂₅	$16*(1+\ln(12)+\ln(5.3)+\ln(9.9)+\ln(7.1))^{-0.514}=5.1$
PTÖE	p ₁₁₁	14
	p ₆₁₂	$16*2^{-0.514}=11.2$
	p ₄₁₃	$18*3^{-0.514}=10.23$
	p ₃₂₁	12
PTÖE	p ₅₂₂	$23*2^{-0.514}=16.1$
	p ₇₂₃	$20*3^{-0.514}=11.37$
PTÖE	p ₂₃₁	10
	p ₈₃₂	$36*2^{-0.514}=25.2$

Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile öğrenme oranı 0.70 iken açılan istasyon sayısı 2 bulunmuştur. İkinci istasyona 3-2-5-7-6 işleri sırasıyla atanırken, bu işlerin işlem süreleri mevcutta, sırasıyla, 12-10-23-20-16 iken logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında sırasıyla 12-5.3-9.9-7.1 ve 5.1 olarak elde edilmiştir. İstasyonun işleri tamamlama süresi mevcut işlem süreleri ile 81 birim zaman iken, öğrenme etkisi altında 39.4 olarak elde edilmiştir.

25 parçalı DHD problemi için matematiksel model sonuçları

25 parçalı DHD problemini oluşturan işlerin görev tanımları, işlem süreleri ve işlerin öncelik ilişkileri Tablo 9’da yer almaktadır. DHD probleminde istasyonların iş tamamlama süreleri çevrim süresini aşmamalıdır. 25 parçalı DHD problemi için çevrim süresi 18’dir. Ayrıca işler istasyona atanırken, Tablo 9’da bulunan işlerin öncelik ilişkileri dikkate alınmalıdır.

Tablo 9. 25 görevli cep telefonu probleminin görev tanımları, süreleri ve öncelik ilişkileri

Görev	Parça ismi	İşlem süresi	Öncelik ilişkisi
1	Antenna	3	
2	Battery	2	
3	Antenna guide	3	1, 2
4	Bolt(Type 1) A	10	
5	Bolt(Type 1) B	10	
6	Bolt(Type 2) 1	15	2
7	Bolt(Type 2) 2	15	2
8	Bolt(Type 2) 3	15	2
9	Bolt(Type 2) 4	15	3
10	Clip	2	4, 5
11	Rubber seal	2	10
12	Speaker	2	11
13	White cable	2	6, 7, 8, 9
14	Red/blue cable	2	6, 7, 8, 9
15	Orange cable	2	6, 7, 8, 9
16	Metal top	2	6, 7, 8, 9
17	Front cover	2	13, 14
18	Back cover	3	15
19	Curciut board	18	13, 14, 16, 18
20	Plastic screen	5	17
21	Keyboard	1	17
22	LCD	5	21
23	Sub-keyboard	15	16, 21
24	Internal IC Board	2	19, 23
25	Microphone	2	21

Tablo 10. 25 parçalı DHD problemi örneği için literatürdeki çalışmaların sonuçlarının karşılaştırılması

Makale	Yaklaşım	İstasyon sayısı
McGovern & Gupta (2005)	H-K	10
McGovern & Gupta (2006)	ACO	9
Ding et al. (2010)	MDACO	9
Kalayci et al. (2012)	SA	9
Kalayci et al. (2016)	VNSGA	9
Duta et al. (2016)	CGA	9
Edis (2021)	CP	9
Li et al. (2020))	BBRA	9
Our approaches	GA, HGA, PSO, PSO-GA	9

Pozisyon ve logaritmik işlem süreleri toplamı tabanlı öğrenme etkileri altında 25 parçalı DHD problemi GA, HGA, PSO ve PSO-GA metasezgisel yöntemler kullanılarak çözülmüştür. Literatürde öğrenme etkisinin dikkate alınmadığı durumda, yani problemin baz hali için elde edilen sonuçlar Tablo 10'da görülmektedir. Benzer şekilde, 25 parçalı DHD probleminin baz hali için açılması gereken istasyon sayısı bizim çalışmamızda da 9 olarak bulunmuştur. İşlerin istasyonlara atanma detayları, işlem süreleri ve istasyonların iş tamamlama süreleri Tablo 11'de sunulmuştur. İşlerin istasyondaki işlem sıraları, toplam iş tamamlama süreleri için paylaşılan optimum veya optimuma en yakın sonuçlar HGA metasezgisel yöntemiyle hesaplanmıştır.

25 parçalı DHD problemi logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında çözüldüğünde, öğrenme oranları sırasıyla 0.00-0.90-0.85-0.80-0.75-0.70 iken, açılan istasyon sayıları sırasıyla, 9-9-8-7-7-6 olarak elde edilmiştir. Öğrenme oranı azaldıkça işlerin işlem sürelerinde daha fazla azalma gerçekleşmiştir ve herhangi bir istasyona atanabilen iş sayısı da artmıştır. Öğrenme oranı 0.90 iken, işlerin işlem sürelerinde azalma görülse de, açılan istasyon sayısını etkilememiştir. Öğrenme oranı 0.8 kabul edildiğinde açılan istasyon sayısı 9'dan 7'ye düşmüştür ve 7 nolu istasyona 18-25-10 ve 9 nolu işler sırasıyla atanmıştır. 18 nolu iş, istasyona ilk sırada atandığı için işlem süresinde bir değişiklik yoktur. Çünkü, istasyonda ilk pozisyonda henüz öğrenme başlamamıştır. İkinci sıraya atanan 25 nolu işin işlem süresi gerçekte 2 birim zaman iken öğrenmenin etkisi ile 1.57 birim zamana, üçüncü sıraya atanan 10 nolu işin işlem süresi 2 birim zamandan 1.47 birim zamana ve son olarak dördüncü sıraya atanan 9 nolu işin işlem süresi ise 18

Tablo 11. LİSTTÖE altında 25 parçalı DHD problemi için atama detayları

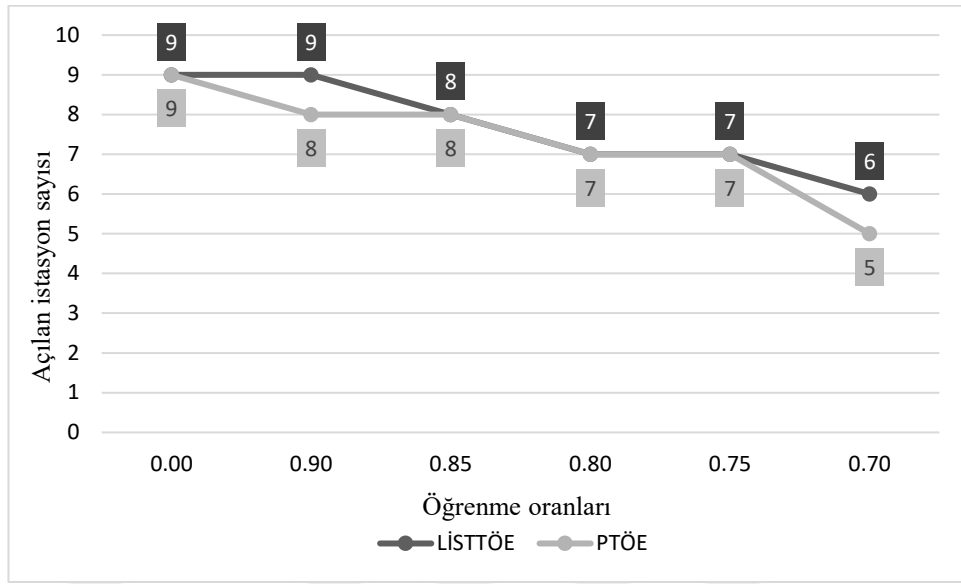
P25 , $\phi_s=18$	Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi			
$\phi_o=0.00$	Atanan İşler	İşlem süreleri	Toplam süre	İstasyon sayısı
İstasyon 1	6	15	15	9
İstasyon 2	19	18	18	
İstasyon 3	5-25-24-3	10-2-2-3	17	
İstasyon 4	10-23	2-15	17	
İstasyon 5	1-4-20	3-10-5	18	
İstasyon 6	7-13	15-2	17	
İstasyon 7	15-22-11-16-17-18-2	2-5-2-2-2-3-2	18	
İstasyon 8	8-14-21	15-2-1	18	
İstasyon 9	12-9	2-15	17	
$\phi_o=0.90$	Atanan İşler	İşlem süreleri	Toplam süre	
İstasyon 1	22-6	5-12,9	17,9	9
İstasyon 2	9-15	15-1,64	16,64	
İstasyon 3	24-1-2-13-12-18-10-17-16-21	2-2,77-1,71-1,63-2,4-1,56-1,54-1,52-0,75	17,51	
İstasyon 4	11-8	2-13,8	15,8	
İstasyon 5	5	10	10	
İstasyon 6	19	18	18	
İstasyon 7	14-23	2-13,85	15,85	
İstasyon 8	20-7	5-12,96	17,96	
İstasyon 9	4-3-25	10-2,5-1,6	14,1	
$\phi_o=0.85$	Atanan İşler	İşlem süreleri	Toplam süre	
İstasyon 1	15-3-4-10-24-12	2-2,65-7,94-1,39-1,37-1,35	16,7	8
İstasyon 2	22-18-2-5	5-2,39-1,49-7,27	16,15	
İstasyon 3	1-25-8	3-1,68-11,97	16,65	
İstasyon 4	21-9-13	1-15-1,47	17,47	
İstasyon 5	6	15	15	
İstasyon 6	20-7	5-11,98	16,98	
İstasyon 7	17-19	2-15,99	17,99	
İstasyon 8	16-14-11-23	2-1,77-1,65-11,82	17,24	
$\phi_o=0.80$	Atanan İşler	İşlem süreleri	Toplam süre	
İstasyon 1	14-13-15-23-1	2-1,69-1,55-10,95-1,78	17,97	7
İstasyon 2	16-5-4-21	2-8,44-6,49-0,57	17,5	
İstasyon 3	12-7-20	2-12,66-3,14	17,8	
İstasyon 4	17-22-24-8	2-4,22-1,38-10,06	17,66	
İstasyon 5	2-6-11-3	2-12,66-1,26-1,85	17,77	
İstasyon 6	19	18	18	
İstasyon 7	18-25-10-9	3-1,57-1,47-10,59	16,63	
$\phi_o=0.75$	Atanan İşler	İşlem süreleri	Toplam süre	
İstasyon 1	7	15	15	7
İstasyon 2	8-25-17	15-1,16-1,14	17,3	
İstasyon 3	12-3-20-10-23	2-2,41-3,37-1,15-8,5	17,43	
İstasyon 4	18-1-22-2-9	3-2-3,22-1,12-8,29	17,63	
İstasyon 5	5-4-14	10-6,1-1,02	17,12	
İstasyon 6	11-13-16-19	2-1,6-1,45-12,22	17,27	
İstasyon 7	6-24-15-21	15-1,16-1,14-0,56	17,86	
$\phi_o=0.70$	Atanan İşler	İşlem süreleri	Toplam süre	
İstasyon 1	22-23-13-14-12	5-9,16-0,89-0,9-0,91	16,86	6
İstasyon 2	3-25-4-9	3-1,36-6,36-7,12	17,84	
İstasyon 3	10-11-24-18-8-16	2-1,52-1,36-1,9-8,43-0,86	16,07	
İstasyon 4	2-5-7	2-7,63-7,63	17,26	
İstasyon 5	1-20-6	3-3,41-8,08	14,49	
İstasyon 6	17-21-15-19	2-0,76-1,67-12,82	17,25	

birim zamandan 10.59 birim zamana düşmüştür. Dolayısıyla, öğrenme oranı 0.80 iken 7 numaralı istasyonun iş tamamlama süresi $25(3+2+2+18)$ 'ten $16.63(3+1.57+1.47+10.59)$ 'e düşmüştür. Öğrenme oranı 0.70 olduğunda açılan istasyon sayısı 6 olarak hesaplanmıştır ve amaç fonksiyonu değerinde %33 oranında iyileşme görülmüştür. 3 nolu istasyona 6 tane iş atanmıştır, atanan işler sırasıyla; 10-11-24-18-8-16'dır. Bu işlerin gerçek işlem süreleri sırasıyla; 2-2-2-3-15-2 birim zamandır. Gerçek işlem süreleriyle, istasyonun iş tamamlama süresi hesaplanırsa, 26 birim zaman bulunmaktadır. Ancak, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile işlerin gerçekleşen işlem süreleri sırasıyla, 2-1.52-1.36-1.9-8.43-0.86 olmaktadır. İstasyonun iş tamamlama süresi ise 16.07 birim zaman elde edilmiştir. İstasyonun iş tamamlama süresinde %38.19 azalma görülmektedir ve böylece çevrim süresini aşmadan işler yapılabilir.

Tablo 12'de, 25 parçalı DHD probleminin pozisyon tabanlı öğrenme etkisi altında sonuçları sunulmuştur. İşlerin pozisyonlarına bağlı olarak öğrenmenin etkisiyle hesaplanan yeni işlem süreleri, istasyonların iş tamamlama süreleri ve açılan istasyon sayıları Tablo 12'de görülmektedir. Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altındaki hesaplama sonuçlarına, benzer sonuçlar elde edilmiştir. Pozisyon tabanlı öğrenme etkisi altında, öğrenme oranları sırasıyla; 0.00-0.90-0.85-0.80-0.75-0.70 iken, açılan istasyon sayıları sırasıyla; 9-8-8-7-7-5 olarak elde edilmiştir. Öğrenme oranı 0.80 ve 0.70 iken logaritmik işlem süreleri toplamı tabanlı öğrenmeye göre daha iyi sonuçlar elde edilmiştir. Öğrenme oranı 0.80 iken 1 nolu istasyona 7 adet iş atanabilmiştir. Bu işler sırasıyla; 14-15-13-12-21-16-19 ve gerçek işlem süreleri sırasıyla, 2-2-2-2-1-2-18'dir. 1 nolu istasyona atanan işlerin pozisyon tabanlı öğrenme etkisi altında elde edilen işlem süreleri ise sırasıyla, 2-1.6-1.4-1.3-0.6-1.1-9.6'dır. İşlerin temel işlem süreleri toplamı 29 birim zaman iken, pozisyon tabanlı öğrenme etkisi altında 17.6 birim zamandır. Böylece 1 nolu istasyonda 7 adet iş, çevrim süresini aşmadan işlem görebilmektedir. Yedinci sıraya atanan 19 nolu işin gerçek işlem süresi 18 birim zaman iken pozisyon tabanlı öğrenme ile 9.6 birim zaman olarak gerçekleşmektedir. Öğrenme ile işin işlem süresinde yaklaşık olarak %47 azalma meydana gelmektedir. Herhangi bir etki gözlemlenmediğinde hesaplanan istasyon sayısı 9 iken, öğrenme oranı 0.70 olduğunda, pozisyon tabanlı öğrenme etkisi altında açılan istasyon sayısı 5 olarak hesaplanmıştır. Dolayısıyla, açılan istasyon sayısında yaklaşık olarak %55 iyileşme görülmüştür.

Tablo 12. PTÖE altında 25Parçalı DHD problemi için atama detayları

P25 , çs=18	Pozisyon tabanlı öğrenme etkisi		Toplam süre	İstasyon sayısı
öo=0.00	Atanan İşler	İşlem süreleri		
İstasyon 1	6	15	15	9
İstasyon 2	19	18	18	
İstasyon 3	5-25-24-3	10-2-2-3	17	
İstasyon 4	10-23	2-15	17	
İstasyon 5	1-4-20	3-10-5	18	
İstasyon 6	7-13	15-2	17	
İstasyon 7	15-22-11-16-17-18-2	2-5-2-2-2-3-2	18	
İstasyon 8	8-14-21	15-2-1	18	
İstasyon 9	12-9	2-15	17	
öo=0.90	Atanan İşler	İşlem süreleri	Toplam süre	İstasyon sayısı
İstasyon 1	25-8-16	2-13.5-1.7	17.2	8
İstasyon 2	19	18	18	
İstasyon 3	2-1-9	2-2.7-12.7	17.4	
İstasyon 4	24-3-21-6	2-2.7-0.8-12.1	17.6	
İstasyon 5	10-14-23	2-1.8-12.7	16.5	
İstasyon 6	20-5-11-13	5-9-1.7-1.6	17.3	
İstasyon 7	22-18-17-4	5-2.7-1.7-8.1	17.5	
İstasyon 8	12-7-15	2-13.5-1.7	17.2	
öo=0.85	Atanan İşler	İşlem süreleri	Toplam süre	İstasyon sayısı
İstasyon 1	13-17-15-5-12-20	2-1.7-1.5-7.2-1.4-3.3	17.1	8
İstasyon 2	1-2-23	3-1.7-11.6	16.3	
İstasyon 3	4-3-14-16-24	10-2.5-1.6-1.4-1.4	16.9	
İstasyon 4	22-6	5-12.7	17.7	
İstasyon 5	9-25	15-1.7	16.7	
İstasyon 6	10-21-19	2-0.8-13.9	16.7	
İstasyon 7	11-8-18	2-12.7-2.3	17	
İstasyon 8	7	15	15	
öo=0.80	Atanan İşler	İşlem süreleri	Toplam süre	İstasyon sayısı
İstasyon 1	14-15-13-12-21-16-19	2-1.6-1.4-1.3-0.6-1.1-9.6	17.6	7
İstasyon 2	24-8-25-3	2-12-1.4-1.9	17.3	
İstasyon 3	22-23	5-11.99	16.99	
İstasyon 4	5-4	10-7.99	17.99	
İstasyon 5	20-9	5-11.99	16.99	
İstasyon 6	6-10	15-1.6	16.6	
İstasyon 7	1-2-11-17-18-7	3-1.6-1.4-1.3-1.8-8.4	17.5	
öo=0.75	Atanan İşler	İşlem süreleri	Toplam süre	İstasyon sayısı
İstasyon 1	10-23-22-25	2-11.2-3.2-1.1	17.4	7
İstasyon 2	11-24-9-16-20-2	2-1.5-9.5-1.1-2.6-1	17.7	
İstasyon 3	3-6	3-11.2	14.2	
İstasyon 4	1-18-8-12-21	3-2.2-9.5-1.1-0.6	16.4	
İstasyon 5	17-19	2-13.5	15.5	
İstasyon 6	14-4-15-5-13	2-7.5-1.3-5.6-1	17.4	
İstasyon 7	7	15	15	
öo=0.70	Atanan İşler	İşlem süreleri	Toplam süre	İstasyon sayısı
İstasyon 1	15-13-11-16-14-19-4-25	2-1.4-1.1-0.9-0.8-7.2-3.7-0.7	17.8	5
İstasyon 2	1-24-5-17-7	3-1.4-5.7-0.9-6.6	17.6	
İstasyon 3	20-18-10-3-22-8	5-2.1-1.1-1.5-2.2-6	17.9	
İstasyon 4	2-12-21-6-9	2-1.4-0.6-7.3-6.6	17.9	
İstasyon 5	23	15	15	



LİSTTÖE: Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi, PTÖE: Pozisyon tabanlı öğrenme etkisi

Şekil 8. LİSTTÖE ile PTÖE altında 25 parçalı DHD probleminde açılan istasyon sayıları

Logaritmik işlem süreleri toplamı tabanlı ve pozisyon tabanlı öğrenme etkisi altında 25 parçalı DHD probleminin farklı öğrenme oranları ile çözümünde hesaplanan sonuçlar Şekil 8’de görülmektedir. Öğrenme oranı 0.90 iken logaritmik işlem süreleri toplamı tabanlı öğrenme altında 9 istasyon açılması gerekirken, pozisyon tabanlı öğrenme etkisi altında 8 istasyon açılması yeterli bulunmuştur. Benzer şekilde, öğrenme oranları sırasıyla 0.80 ve 0.70 olduğunda, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında sırasıyla 7 ve 6 istasyon gerekli bulunurken, pozisyon tabanlı öğrenme etkisi altında 7 ve 5 istasyon yeterli bulunmuştur. Öğrenme oranları 0.85 ve 0.75 olduğunda ise her iki öğrenme türünde de açılacak istasyon sayısı değeri 8 ve 7 olacak şekilde eşit hesaplanmıştır. Şekil 8’de de görüldüğü gibi, pozisyon tabanlı öğrenme etkisi altında 25 parçalı DHD probleminde öğrenme oranının azalmasıyla açılan istasyon sayısı değerinde daha fazla iyileşme görülmüştür. Öğrenme etkisi arttıkça işlerin işlem süreleri azalmakta ve istasyonlara atanan iş sayısı artmaktadır. Böylece açılan istasyon sayısında azalma görülmektedir.

25 parçalı DHD problemi, açık sonuçlarını paylaşmış olduğumuz HGA metasezgisel yöntemi dışında GA, PSO ve PSO-GA hibrit yaklaşımı ile de çözdürülmüştür. Tüm yöntemlerle, problemin optimum veya optimuma yakın sonuçlar elde edilmiştir. 4 farklı metasezgisel yaklaşımın belirli iterasyon sayısı ile optimuma ulaşma oranları

hesaplanmıştır. Bu hesaplamalar, 1-21 parçadan oluşan DHD problemleri için 100 kez 1000 yeni çözüm adayı kullanılarak, 25-53 parçadan oluşan DHD problemleri için 100 kez 2000 çözüm adayı için ve 53 ten fazla olan DHD problemleri için ise 100 kez 5000 çözüm adayı kullanılarak bulunmuştur. Hibrit yöntemlerde, GA yöntemi içerisinde 2 aşamada çözüm aranmakta, hem de bu yönteme entegre edilen yaklaşımlarla çözüm aranmaktadır. Ancak, PSO da tek arama aşaması ve GA da 2 arama aşaması bulunmaktadır. Dolayısıyla, iterasyon sayısını eşit tuttuğumuzda doğru bir yaklaşım olmayacağı için karşılaştırma, kullanılan toplam çözüm adayları üzerinden yapılmıştır. 25 parçadan oluşan DHD probleminde 4 farklı yöntem ile hesaplanan optimuma ulaşma oranları Ek 1 de sunulmuştur.

83 parçalı DHD problemi için logaritmik işlem süreleri toplamı ve pozisyon tabanlı öğrenme etkileri altında hesaplama sonuçları

83 parçalı (Arcus I) problem örneği için işlerin işlem süreleri ve öncelik ilişkileri Ek-1’de sunulmuştur (Scholl, 1995). Çevrim süresi 7571 olarak kullanılmıştır. Çok görevli bu DHD problemi, pozisyon ve logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında incelenmiştir. GA, HGA, PSO ve PSO-GA yaklaşımları kullanılarak çözümler elde edilmiştir.

Tablo 13’de logaritmik işlem süresi tabanlı öğren etkisi altında 83 parçalı problem için elde edilen sonuçlar yer almaktadır. Tablo 14’de ise pozisyon tabanlı öğrenme etkisi altındaki sonuçlar sunulmuştur. Tablo 13 ve Tablo 14’de sunulan sonuçlar HGA yaklaşımı ile elde edilmiştir. Her iki tabloda da hangi işin hangi istasyona hangi sırada atandığı, öğrenme etkileri altında hesaplanan, işlerin gerçek işlem süreleri ve istasyonların iş tamamlama süreleri sunulmuştur. Öğrenme etkileri dikkate alınmadığı, problemin en baz hali için literatürde bulunan amaç fonksiyonu değeri bu çalışmada da elde edilmiştir. Öğrenme oranı değeri değiştikçe açılan istasyon sayısında da değişiklik görülmektedir. Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında 83 parçalı DHD problemi için, öğrenme oranları sırasıyla 0.90-0.85-0.80-0.75-0.70 olduğunda açılan istasyon sayısı değerleri sırasıyla 8-7-6-5-4 bulunmuştur. Öğrenme oranındaki her yüzde yarım birimlik düşüşte istasyon sayısında bir düşüş görülmektedir. Öğrenme oranı, çalışmada kullanılan en yüksek değeri 0.90 olduğunda bile amaç fonksiyonu değeri 11’den 8’e düşmüştür. Yani, istasyon sayısında yaklaşık olarak %27 oranında azalma gerçekleşmektedir.

Tablo 13. LİSTTÖE altında 83 Parçalı DHD problemi için atama detayları

P83 , çs=7571		Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi		
öo=0	Atanan işler	İşlem süreleri	Toplam süre	İstasyon sayısı
İstasyon 1	67-17-45-5-79-32-39-22	578-880-594-1700-286-842-1310-847	7037	11
İstasyon 2	53-1-60-72-41-78-47	578-1673-578-1298-494-1164-822	6607	
İstasyon 3	8-4-3-58-18-35-62	1040-973-1838-578-667-1143-578	6817	
İstasyon 4	38-77-48-70-14-7	1251-701-578-887-1133-2231	6781	
İstasyon 5	16-34-54-25-65-75-69-66-40-51	483-1234-578-780-578-764-467-578-83-578	6123	
İstasyon 6	28-82-56-37-10-20-19-27	1883-1300-578-792-1250-233-600-748	7384	
İstasyon 7	61-26-6-42-33	578-912-2881-1288-829	6488	
İstasyon 8	74-13-64-83	2543-500-578-3691	7312	
İstasyon 9	68-43-30-21-71-80-11-63-76-12	578-792-1004-408-396-2100-700-578-357-464	7377	
İstasyon 10	23-59-31-57-44-29-9-46-49-15-55	767-578-713-578-578-714-793-578-584-577-578	7038	
İstasyon 11	36-81-50-73-52-24-2	1266-450-578-1100-578-850-985	5807	
öo=0.90				
İstasyon 1	12-20-51-69-38-39-65-61-34-81-21-14-22-56-30-29-13	464-172,8-394,7-300,28-771,85-778,73-333,47-326,2-683,2-244,4-218,5-598,9-441,5-297,6-511,5-359,8-249,6	7147,03	7
İstasyon 2	16-80-1-26-45-43-75-40-35-47-66-42-57	483-1556,3-1113,8-571,9-358,2-463,8-436,2-46,4-630,99-445,5-308,4-678,1-300,1	7392,69	
İstasyon 3	83-2-8-55-11-6	3691-702,8-683,8-360,6-421,4-1683,9	7543,5	
İstasyon 4	31-67-64-15-7-4-33-24-19-41-18-52-50-36-71-76	713-424,9-388,6-367,1-1363,7-572,6-474,9-476,2-329,7-267,1-355,5-303,9-300,2-650,3-201-179,6	7368,3	
İstasyon 5	5-37-59-46-53-49-10-73-68-72-82	1700-572,7-383,8-364,6-350,9-344-718,1-616,9-317,6-701,9-691,4	6761,9	
İstasyon 6	54-63-79-70-27-23-3-62-17-25-58-60-78-77	578-426,7-192,7-568,4-458,5-454,8-1060,5-324,9-485,5-422,8-308,5-304,4-605,5-360	6551,2	
İstasyon 7	74-32-48-44-28-9	2543-604,6-382-363,4-1140,4-463,7	5497,1	
öo=0.85				
İstasyon 1	-11-72-43-57-46-42-70-41-79-52-6-82-8-29-	842-433,3-701,4-390,7-268,5-256,6-550,7-365,7-197,7-111,8-221,6-1082,3-476,8-374,2-252,5-168,3	6694,1	5
İstasyon 2	3-77-60-34-74-19-65-26-3-18-15-10-67-71-6-	793-434,7-312,6-615-1185,5-264,3-245,5-375,4-734,5-258,9-219,1-465,5-210,9-142,1-204,5	6461,5	
İstasyon 3	39-61-50-7-5-78-45-23-33-4-55-69-25-1-68	1310-353,1-311,1-1108,1-786,4-511,1-250,5-313,8-329,8-377,4-219-173,6-285-600,7-203,8	7133,4	
İstasyon 4	28-51-30-81-49-36-27-80-76-66-12-22-20-21-14-17-59-40-73	1883-349,6-537,2-221,2-271,8-563-318,7-865,7-142,3-225,4-177-317-85,5-147,7-404-308,6-199,7-28,3-372,2	7417,9	
İstasyon 5	14-56-13-83-38-47-31-35-75-37-48-44-62-54-	578-616,8-459,5-286,6-233,8-1649,6-531,7-337,2-284,2-444,5-289,9-294,5-210,9-207,5-204,3-201,4-198,7-196,2	7225,3	
öo=0.80				
İstasyon 1	33-52-5-51-69-7-15-76-28-30	829-299,3-736,7-220,2-164,7-742,1-180,2-106,9-545,1-279,1	4103,3	4
İstasyon 2	71-54-65-35-17-74-83-23-27-20-68-32-63-79-42-10-2-58-62-1-12-72-6	396-309,1-254,9-448,6-314,7-849,5-1155,4-226,9-213,2-64,2-155,6-220,7-147,6-71,4-315,9-299,7-231,3-133,2-131-373,5-101,8-281-614,3	7309,5	
İstasyon 3	66-43-29-47-8-44-9-73-56-19-37-34-16-64-48-60-70-38-39-13	578-416,5-309,6-317,8-369,4-191,7-249,9-331,2-166,9-167,6-214,7-325-123,7-144,8-141,8-139-209,4-289,9-297,9-111,7	5096,5	
İstasyon 4	50-4-61-49-77-75-36-59-21-78-11-81-53-25-55-45-41-3-57-31-14-22-82-80-18-26-24-40-46-67	578-511,7-249,4-225,8-250,1-255,4-400,7-174-118,3-327,1-190,1-118,8-149-196,5-142,5-143,4-117-428,1-131,9-160,2-250,6-184,4-278,9-443,9-138,9-187,6-172,7-16,7-115,4-114,2	6771,3	
öo=0.75				
İstasyon 1	51-50-33-62-36-22-13-58-53-1-43-55-29-35-41-47-83-3-8-67	578-252,4-286,9-172-340,3-207,8-114,5-125,6-119,7-332,1-150,1-105,6-126,5-196,6-82,4-133,9-586,5-283,3-156,3-85,1	4435,6	4
İstasyon 2	39-5-56-31-24-72-46-60-54-76-71-80-63-45-66-23-59-14-40-26-65-7-6-4-20	1310-710,7-189,2-205,7-222,3-313,3-129,9-123,4-117,8-69,9-74,9-385,3-101,9-101,8-96,5-124,9-91,9-176,2-12,6-137-85,1-322,5-407,4-134,6-31,7	5676,5	
İstasyon 3	78-69-34-64-15-11-73-81-52-61-21-57-68-12-25	1164-196,4-421,1-168,9-152,9-171,7-252,5-97,1-119,3-114,4-77,7-106,7-103,3-80,5-131,9	3358,4	
İstasyon 4	32-18-74-37-79-19-44-30-27-77-42-38-10-28-70-2-75-16-82-9-49-48-17	842-285,4-866,4-227,7-74,4-146,5-132,4-217,7-153,9-138-243,9-227,8-219,6-320,2-146,1-158,1-119,6-73,9-195,3-116,5-84,2-81,9-122,7	5194,2	
öo=0.70				
İstasyon 1	68-39-36-48-46-55-16-73-27-80-61-15-14-67-23-11-7-45-79-3-66-65-28-12-29-17-13	578-469,5-332,1-126,2-112,4-102,6-79,4-169,8-108,2-288-74,8-71,8-135,8-66,6-85,7-75,8-234,8-60,4-28,4-179,1-54,8-53,6-171,4-41,2-62,3-75,5-42,1	3880,3	4
İstasyon 2	38-71-37-30-77-1-58-34-4-83-60-20-10-63	1251-134,8-211,6-224,7-137,4-297,8-93,7-187,8-138,8-498,6-73,4-28,5-148,6-66,1	3492,8	
İstasyon 3	78-44-59-72-62-52-31-43-21-70-8-76-57-50-54-74-2-82-9-24-41-56-49-32-40-75-19-18-81-64-6	1164-197,7-152,6-290,7-112,7-102,8-117,4-121,8-59-122,5-136,6-44,8-70,1-67,7-65,5-279,8-104,3-133,7-79,2-82,8-46,9-53,9-53,4-75,5-7,3-66,6-51,3-56,1-37,2-47,1-231,5	4232,5	
İstasyon 4	5-42-69-35-22-26-25-47-33-53-51	1700-430,3-118,1-249,8-162,6-159,1-125,5-123,7-117,7-77,9-74,6	3339,3	

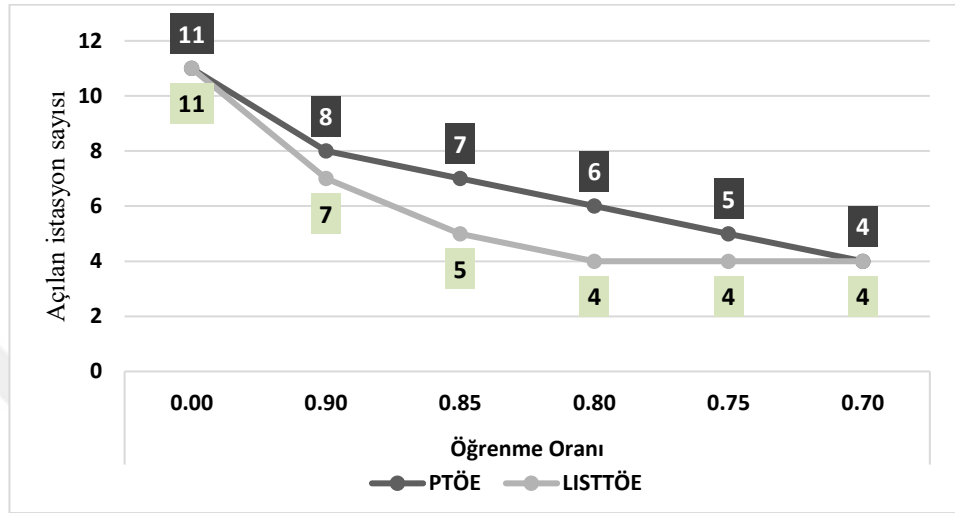
Öğrenme oranı 0.80 iken, açılacak istasyon sayısı 4 bulunmuştur. Öğrenme etkisi dikkate alınmadığı duruma göre istasyon sayısında yaklaşık olarak %63.6 oranında azalma görülmektedir. Öğrenme oranı dikkate alınmadığında bir istasyona en fazla 11 iş atanabilirken, öğrenme oranı 0.80 olduğunda bir istasyona atanan maksimum iş sayısı 30 olmuştur. Dolayısıyla, öğrenme etkisi nedeniyle işlerin işlem süreleri azaldığı için her hangi bir istasyona atanan maksimum iş sayısı artış göstermektedir. Yine, öğrenme etkisi düşünülmediği durumda istasyonların ortalama iş yükü 7.54 iş iken, öğrenme oranı 0.80 olduğunda 20.75 iştir.

Pozisyon tabanlı öğrenme etkisi altında da 83 parçalı DHD probleminde, öğrenme oranları sırasıyla 0.90-0.85-0.80-0.75-0.70 iken, açılan istasyon sayısı değerleri sırasıyla 8-7-6-5-4 bulunmuştur. Pozisyon tabanlı öğrenme etkisinin 83 parçalı DHD problemi için amaç fonksiyonu üzerindeki etkisi, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile aynıdır. Her iki öğrenme etkisi altında da 83 parçalı DHD probleminde açılan istasyon sayısı değerleri eşit bulunmuştur. Ancak, öğrenme oranı 0.80 iken, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında, herhangi bir istasyona atanan iş sayısı maksimum 16 iken, pozisyon tabanlı öğrenme etkisi altında 4 nolu istasyona 17 adet iş atanmıştır. Öğrenme oranı 0.70 iken, öncelik ilişkileri de dikkate alınarak hesaplanan istasyon sayısı 4 olarak bulunmuştur. Pozisyon tabanlı öğrenmede 3 nolu istasyona 20 adet iş atanmıştır. Bu işlerin gerçek işlem süreleri sırasıyla; 578-233-767-1040-850-713-1673-3691-578-985-792-578-578-467-1310-701-2100-1164 iken, pozisyon tabanlı öğrenme etkisi altında 578-163-436-510-372-284-615-1267-187-347-169-274-212-149-144-112-305-159-462-250 olarak hesaplanmıştır. Bu işlerin temel işlem süreleri toplamı 20509 iken, pozisyon tabanlı öğrenme etkisi ile hesaplanan işlem süreleri toplamı 6995'dir ve çevrim süresi olan 7571'i aşmamaktadır. 19. sıraya atanan 80 nolu işin temel işlem süresi 2100 birim zaman iken, öğrenme etkisi altında 462 birim zamana düşmektedir (Tablo 14). İstasyona atanan işlerin işlem süreleri temel işlem sürelerinden azdır. Beşinci sıraya atanan işin işlem süresindeki azalma oranı ikinci sıraya atanmasından fazla iken, altıncı sıraya atanmasından ise daha düşüktür.

Tablo 14. PTÖE altında 83 Parçalı DHD problemi için atama detayları

P83 , çs=7571		Pozisyon tabanlı öğrenme etkisi		
öo=0	Atanan işler	İşlem süreleri	Toplamsüre	İstasyon sayısı
İstasyon 1	67-17-45-5-79-32-39-22	578-880-594-1700-286-842-1310-847	7037	11
İstasyon 2	53-1-60-72-41-78-47	578-1673-578-1298-494-1164-822	6607	
İstasyon 3	8-4-3-58-18-35-62	1040-973-1838-578-667-1143-578	6817	
İstasyon 4	38-77-48-70-14-7	1251-701-578-887-1133-2231	6781	
İstasyon 5	16-34-54-25-65-75-69-66-40-51	483-1234-578-780-578-764-467-578-83-578	6123	
İstasyon 6	28-82-56-37-10-20-19-27	1883-1300-578-792-1250-233-600-748	7384	
İstasyon 7	61-26-6-42-33	578-912-2881-1288-829	6488	
İstasyon 8	74-13-64-83	2543-500-578-3691	7312	
İstasyon 9	68-43-30-21-71-80-11-63-76-12	578-792-1004-408-396-2100-700-578-357-464	7377	
İstasyon 10	23-59-31-57-44-29-9-46-49-15-55	767-578-713-578-578-714-793-578-584-577-578	7038	
İstasyon 11	36-81-50-73-52-24-2	1266-450-578-1100-578-850-985	5807	
öo=0.90	Atanan işler	İşlem süreleri	Toplamsüre	İstasyon sayısı
İstasyon 1	81-70-62-36-5-65-54-38-46-60-15-51	450-798-489-1025-1331-440-430-912-414-407-401-396	7493	8
İstasyon 2	30-49-58-3-76-41-23-73-55-80	1004-526-489-1489-279-376-571-802-414-1480	7430	
İstasyon 3	71-82-67-79-37-40-39-1-21-9-19-42	396-1170-489-232-620-63-975-1220-291-558-417-883	7314	
İstasyon 4	24-43-52-26-83-44-27-75	850-713-489-739-2890-440-556-557	7234	
İstasyon 5	53-56-14-61-34-6-13-28	578-520-959-468-966-2194-372-1373	7430	
İstasyon 6	32-68-35-22-12-63-59-33-18-69-78-4	842-520-967-686-363-440-430-604-478-330-808-667	7135	
İstasyon 7	47-11-8-64-66-45-2-74-10-20-	822-630-880-468-453-452-733-1854-895-164	7351	
İstasyon 8	17-31-16-48-7-77-50-72-25-29-57	880-642-409-468-1747-534-430-946-558-503-401	7518	
öo=0.85	Atanan işler	İşlem süreleri	Toplamsüre	İstasyon sayısı
İstasyon 1	2-72-36-71-33-24-35-42-11-82-65	985-1103-979-286-568-558-724-791-418-756-329	7497	7
İstasyon 2	43-61-38-52-78-64-50-69-18-26-7-25-79	792-491-967-418-798-380-366-287-399-532-1272-436-157	7295	
İstasyon 3	44-40-30-8-56-83-15-73-17-58-76	578-71-776-751-396-2425-366-676-526-337-203	7105	
İstasyon 4	81-34-77-54-13-80-3-27-20-51-39-41	450-1049-542-418-343-1379-1165-459-139-337-747-276	7304	
İstasyon 5	59-70-28-10-32-47-1-14-55-21-16	578-754-1455-903-577-540-1060-696-345-238-275	7421	
İstasyon 6	5-57-53-4-48-68-45-60-75-63-6	1700-491-447-703-396-380-376-355-456-337-1642	7283	
İstasyon 7	23-19-46-62-37-74-9-31-66-12-49-67-22-29	767-510-447-417-543-1671-502-438-345-270-333-323-464-385	7415	
öo=0.80	Atanan işler	İşlem süreleri	Toplamsüre	İstasyon sayısı
İstasyon 1	47-67-61-82-23-72-12-49-83-7-71	822-462-406-832-457-729-248-299-1819-1063-183	7320	6
İstasyon 2	11-19-62-30-46-55-58-25-74-69-52-228-36	700-480-406-642-344-325-309-399-1253-223-267-443-824-541	7156	
İstasyon 3	14-35-66-73-51-6-18-54-59-37-15-70-77-20	1133-914-406-704-344-1618-356-296-285-377-267-398-307-100	7505	
İstasyon 4	50-81-53-44-42-4-32-38-3-40-68-41-8-21-10-29-56	578-360-406-370-767-546-450-640-906-40-268-222-455-174-523-292-232	7229	
İstasyon 5	39-13-24-76-16-5-43-1-64-33-26-27-9-79-57-60	1310-400-597-228-288-955-423-856-285-395-421-336-347-122-242-237	7442	
İstasyon 6	31-45-65-80-75-17-22-63-48-34-78	713-475-406-1344-455-494-453-295-285-588-538	6046	
öo=0.75	Atanan işler	İşlem süreleri	Toplamsüre	İstasyon sayısı
İstasyon 1	22-31-26-71-67-40-19-29-53-16-34-56-82-61-28-47-75-51-38-25	847-535-578-223-296-39-267-301-232-186-456-206-448-193-612-260-236-174-369-225	6683	5
İstasyon 2	60-4-72-43-65-23-83-1-6-9-14	578-730-823-445-296-365-1646-706-1157-305-419	7470	
İstasyon 3	10-55-17-32-63-45-49-21-74-52-44-15-18-77-30-81-48-69-70-33-78	1250-433-558-474-296-282-260-172-1022-222-214-206-230-234-326-142-178-141-261-239-329	7469	
İstasyon 4	46-2-5-79-68-41-62-13-12-76-80-20-58-73-54-64-7-27-50-3	578-739-1077-161-296-235-258-211-186-137-776-83-199-368-188-183-688-225-170-530	7288	
İstasyon 5	8-36-39-66-42-35-24-11-37-57-59	1040-949-830-325-660-543-379-295-318-223-214	5776	
öo=0.70	Atanan işler	İşlem süreleri	Toplamsüre	İstasyon sayısı
İstasyon 1	55-26-53-32-7-72-65-30-36-58-40-62-51-15-5-10-19-63-42-12-44-35-27-16-56-71-17	578-639-329-412-975-517-213-345-409-177-24-161-155-149-423-301-140-131-283-99-121-233-149-94-110-74-162	7403	4
İstasyon 2	3-43-41-79-28-47-57-66-6-49-59-45-68-34-64-82-25-13-73-81-70	1838-555-281-140-823-327-213-198-931-179-169-167-154-318-144-312-182-113-242-96-185	7567	
İstasyon 3	60-20-23-8-24-31-1-83-67-14-46-2-37-48-61-69-39-77-80-78	578-163-436-510-372-284-615-1267-187-347-169-274-212-149-144-112-305-159-462-250	6995	
İstasyon 4	18-74-21-22-54-33-29-50-9-75-52-38-11-4-76	667-1781-232-415-253-330-263-198-255-234-169-349-187-251-89	5673	

Her iki öğrenme etkisi altında 83 parçalı DHD problemi için açılan istasyon sayısı değerleri Şekil 9’da sunulmuştur. Tüm öğrenme oranları için logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında daha iyi sonuçlar elde edilirken, öğrenme oranı 0.70 iken ise eşit sonuçlar elde edilmiştir.



Şekil 9. LİSTTÖE ile PTÖE altında 83 parçalı DHD probleminde açılan istasyon sayıları

Literatürdeki 15 farklı DHD problemi için logaritmik işlem süreleri toplamı ve pozisyon tabanlı öğrenme etkileri altında hesaplama sonuçları

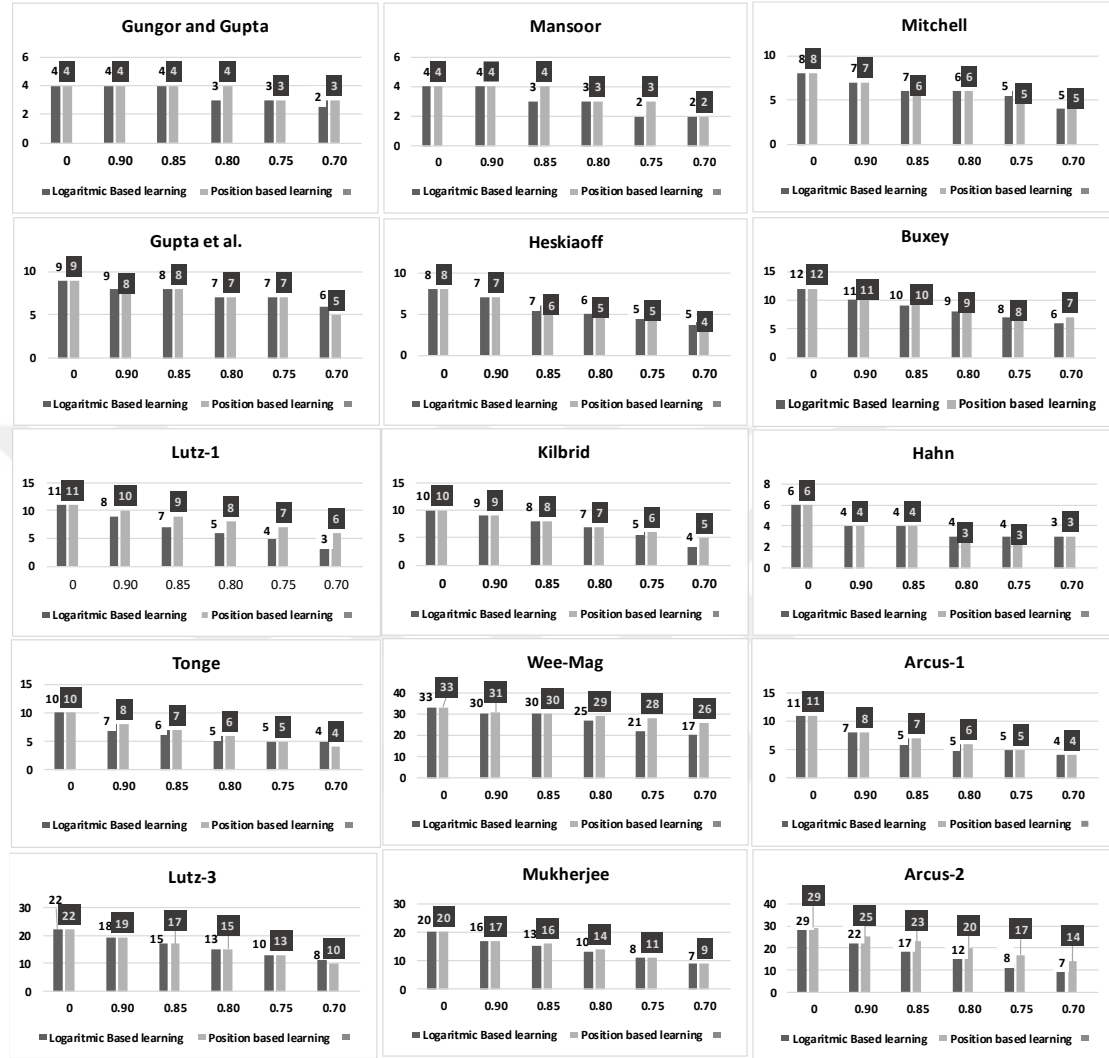
Literatürde bilinen 15 DHD problemi (Scholl, 1995) öğrenme etkisi altında ele alınmıştır. Altı farklı öğrenme oranı, 0, 0.90, 0.85, 0.80, 0.75 ve 0.70, kullanılmıştır. Öğrenme, işlerin gerçek işlem sürelerini azaltıcı etkiye sahiptir. Çünkü işler öğrenildikçe daha kısa sürede yapılmaktadır. Bir işin öğrenmeden etkilenme oranı kendinden önceki sıralara atanmış işlerin etkilenme oranından daha fazladır. Benzer şekilde, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altındaki DHD probleminde ise, iş yine ne kadar ileri pozisyona atanırsa, işlem süresi o kadar etkilenir ve kısalmır. Bu öğrenme türünde ise, bir işin öğrenmeden etkilenme oranı kendinden önceki işlerin işlem süreleri toplamının logaritmik değeri üzerinden hesaplanır. Öğrenme etkisi değeri negatif bir sayı olduğu için, iş ne kadar ileri pozisyona atanırsa ve önceki işlerin işlem sürelerinin logaritmik değerleri toplamı ne kadar fazla olursa işin gerçekleşecek işlem süresi daha fazla azalır.

Tablo 15. Farklı veri kümeleri için LİSTTÖE ve PTÖE ile elde edilen sonuçlar

Yazar/ Set	n	çevrim süresi	*Optimal sonuç	Öğrenme oranı	LİSTTÖE	PTÖE	Yazar/ Set	n	çevrim süresi	*Optimal sonuç	Öğrenme oranı	LİSTTÖE	PTÖE
Gungor and Gupta	8	40	4	0	4	4	Hahn	53	2806	6	0	6	6
				0.90	4	4					0.90	4	4
				0.85	3	4					0.85	4	4
				0.80	3	4					0.80	4	3
				0.75	3	3					0.75	4	3
Mansoor	11	48	4	0.70	2	3	Tonge	70	364	10	0.70	3	3
				0	4	4					0	10	10
				0.90	4	4					0.90	7	8
				0.85	3	4					0.85	6	7
				0.80	3	3					0.80	5	6
Mitchell	21	15	8	0.75	2	3	Wee-Mag	75	49	33	0.75	5	5
				0.70	2	2					0.70	4	4
				0	8	8					0	33	33
				0.90	7	7					0.90	30	31
				0.85	7	6					0.85	30	30
Gupta et al.	25	18	9	0.80	6	6	Arcus-1	83	7571	16	0.80	25	29
				0.75	7	7					0.75	21	28
				0.70	6	5					0.70	17	26
				0	9	9					0	11	11
				0.90	9	8					0.90	7	8
Heskiaoff	28	138	8	0.85	8	8	Lutz-3	89	79	22	0.85	5	7
				0.80	7	7					0.80	5	6
				0.75	7	7					0.75	5	5
				0.70	6	5					0.70	4	4
				0	8	8					0	22	22
Buxey	29	30	12	0.90	6	7	Mukherjee	94	222	20	0.90	18	19
				0.85	5	6					0.85	15	17
				0.80	5	5					0.80	13	15
				0.75	4	5					0.75	10	13
				0.70	3	4					0.70	8	10
Lutz-1	32	1414	11	0	12	12	Arcus-2	111	5785	29	0	20	20
				0.90	11	11					0.90	16	17
				0.85	10	10					0.85	13	16
				0.80	9	9					0.80	10	14
				0.75	8	8					0.75	8	11
Kilbrid	45	56	10	0.70	6	7					0.70	7	9
				0	11	11					0	29	29
				0.90	8	10					0.90	22	25
				0.85	7	9					0.85	17	23
				0.80	5	8					0.80	12	20

Tablo 15’ de 15 farklı DHD problemi için elde edilen amaç fonksiyonu değerleri görülmektedir. Öğrenme oranı azaldıkça her iki öğrenme etkisi ile de amaç fonksiyonu değeri azalmaktadır. Örneğin; 45 parçalı veri seti için çevrim süresi 56 olarak verilmiştir. Öğrenme düşünülmendiği durumda amaç fonksiyonu değeri 10 olarak bulunmuştur. Ancak, öğrenme etkisi değeri sırasıyla, 0.90-0.85-0.80-0.75-0.70 iken pozisyon tabanlı öğrenme etkisi altında amaç fonksiyonu değeri sırasıyla; 10-9-8-7-6-5, logaritmik işlem süreleri

Tablo 16. 15 farklı DHD problemi için öğrenme etkileri altında açılan istasyon sayısı değerleri



toplamı tabanlı öğrenme etkisi altında ise sırasıyla 10-9-8-7-5-4 olarak elde edilmiştir. 45 parçalı (Kilbrid) DHD problemi için logaritmik işlem süreleri toplamı ve pozisyon tabanlı öğrenme etkisi altında öğrenme oranları 0.9-0.85-0.80 iken aynı amaç fonksiyonu değerleri bulunmuştur. Öğrenme oranı 0.70 iken ise pozisyon tabanlı öğrenme ile istasyon sayısı 5 olarak bulunurken, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile istasyon sayısı 4 olarak bulunmuştur. 21 ve 25 parçalı problemlerde pozisyon tabanlı öğrenme etkisi ile daha az istasyon sayısı elde edilirken, diğer problemlerde ise logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile daha az istasyon sayısı elde edilmiştir. 111 parçalı DHD probleminde öğrenme oranları sırasıyla 0.90-0.85-0.80-0.75-0.70 iken, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında açılan istasyon sayısı sırasıyla, 22-17-12-8-7 olarak elde edilmişken, pozisyon tabanlı öğrenme etkisi altında

Tablo 17. GA, HGA, PSO ve PSO-GA yaklaşımlarının çözüm sonuçları (LİSTTÖE etkisi altında)

Set	Cevrim Süresi	Öğrenme Oranı	opt*	f(GA)	Min-Max	f(HGA)	Min-Max	f(PSO)	Min-Max	f(PSO-GA)	Min-Max
8-part	40	0	4	4	4-5	4	4	4	4-5	4	4
		0.9	-	4	4-5	4	4	4	4-5	4	4
		0.85	-	3	3-4	3	3	3	3-4	3	3
		0.80	-	3	3-4	3	3	3	3-4	3	3
		0.75	-	3	3-4	3	3	3	3-4	3	3
11-part	48	0.7	-	2	3	3	3	3	3	3	3
		0	4	4	4-5	4	4	4	4-5	4	4
		0.9	-	4	4	4	4	4	4	4	4
		0.85	-	3	3-4	3	3	3	3-4	3	3
		0.80	-	3	3-4	3	3	3	3-4	3	3
21-part	15	0.75	-	2	2-3	2	2-3	2	2-3	2	2
		0.7	-	2	2-3	2	2	2	2-3	2	2
		0	8	8	8	8	8	8	8-9	8	8
		0.9	-	7	7-8	7	7	7	7-8	7	7
		0.85	-	7	7-8	7	7	7	7-8	7	7
25-part	18	0.80	-	6	6-7	6	6	6	6-7	6	6
		0.75	-	5	5-6	5	5-6	5	5-7	5	5-6
		0.7	-	5	5-6	5	5	5	5-7	5	5
		0	9	9	9-10	9	9	9	9-10	9	9
		0.9	-	9	9-10	8	8-9	9	9-10	8	8-9
28-part	138	0.85	-	8	8-9	8	8	8	8-9	8	8
		0.80	-	7	7-8	7	7-8	7	7-9	7	7-8
		0.75	-	7	7-8	7	7	7	7-8	7	7
		0.7	-	6	6-8	6	6	6	6-8	6	6-7
		0	8	8	8	8	8	8	8-9	8	8
29-part	30	0.9	-	6	6-7	6	6	6	6-7	6	6
		0.85	-	5	5-7	5	5-6	5	5-7	5	5-6
		0.80	-	5	5-6	5	5-6	5	5-6	5	5-6
		0.75	-	4	4-5	4	4	4	4-5	4	4
		0.7	-	3	3-5	3	3-5	3	3-5	3	3-5
32-part	1414	0	12	12	12	12	12	12	12	12	12
		0.9	-	11	11-12	11	11-12	11	11-12	11	11-12
		0.85	-	10	10-11	10	10-11	10	10-11	10	10-11
		0.80	-	9	9-10	9	9-10	9	9-10	9	9-10
		0.75	-	8	8-9	8	8-9	8	8-9	8	8-9
45-part	56	0.7	-	6	6-8	6	6-8	6	6-8	6	6-8
		0	11	11	11-12	11	11	11	11-12	11	11
		0.9	-	9	9-11	8	8-10	9	9-11	8	8-10
		0.85	-	7	7-8	7	7-8	7	7-8	7	7
		0.80	-	5	5-6	5	5-6	5	5-6	5	5-6
53-part	2806	0.75	-	4	4-6	4	4-6	4	4-6	4	4-6
		0.7	-	3	3-5	3	3-5	3	3-5	3	3-5
		0	10	10	10	10	10	10	10	10	10
		0.9	-	9	9-11	9	9-10	9	9-11	9	9-10
		0.85	-	8	8-9	8	8-9	8	8-9	8	8-9
70-part	364	0.80	-	7	7-8	7	7-8	7	7-8	7	7-8
		0.75	-	5	5-8	5	5-8	5	5-8	5	5-8
		0.7	-	5	5-6	5	5-6	5	5-6	5	5-6
		0	6	6	6	6	6	6	6-7	6	6
		0.9	-	4	4-5	4	4-6	4	4-6	4	4-5
75-part	49	0.85	-	4	4-5	4	4-5	4	4-6	4	4-4
		0.80	-	4	4-5	4	4-5	4	4-5	4	4-5
		0.75	-	4	4-5	4	4	4	4-5	4	4
		0.7	-	3	3-5	3	3-4	3	3-5	3	3-4
		0	10	10	10-11	10	10-11	10	10-11	10	10-11
83-part	7571	0.9	-	7	7-9	7	7	7	7-9	7	7
		0.85	-	6	6-9	6	6-8	6	6-9	6	6-8
		0.80	-	6	6-8	6	6-8	6	6-8	6	6-8
		0.75	-	6	6-8	6	6-7	6	6-8	6	6-7
		0.7	-	4	4-5	4	4-5	4	4-5	4	4-5
89-part	79	0	22	22	22-23	22	22-23	23	23-24	22	22-23
		0.9	-	17	17-18	17	17-18	17	17-18	17	17-18
		0.85	-	15	15-16	15	15-16	15	15-16	15	15-16
		0.80	-	13	13-15	13	13-15	13	13-15	13	13-15
		0.75	-	10	10-14	10	10-13	10	10-14	10	10-15
94-part	222	0.7	-	8	8-11	8	8-10	8	8-12	8	8-10
		0	20	21	21-22	21	21-22	21	21-22	20	20-22
		0.9	-	16	16-18	16	16-18	16	16-19	16	16-18
		0.85	-	13	13-15	13	13-15	14	14-16	13	13-15
		0.80	-	10	10-14	10	10-13	11	11-14	10	10-13
111-part	5785	0.75	-	8	8-11	8	8-10	8	8-11	8	8-10
		0.7	-	7	7-10	7	7-9	8	8-10	7	7-9
		0	29	29	29-31	29	29-31	29	29-31	29	29-31
		0.9	-	22	22-25	22	22-24	22	22-26	22	22-24
		0.85	-	17	17-20	17	17-19	17	17-20	17	17-20
		0.80	-	12	12-16	12	12-15	13	13-16	12	12-15
		0.75	-	8	8-12	8	8-11	8	8-12	8	8-12
		0.7	-	7	7-9	7	7-8	7	7-9	7	7-8

sırasıyla; 25-23-20-17-14 olarak elde edilmiştir. Tüm öğrenme oranları için logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile daha az istasyon sayıları elde edilmiştir. Açılan istasyon sayılarındaki değişim oranı logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında sırasıyla %24,13- %41,3-%58,6-%72,4-%75,8 iken, pozisyon tabanlı öğrenme etkisi altında sırasıyla; %13,8-%20,7-%31-%41,4-%51,7'dir.

15 farklı DHD problemi için iki öğrenme türü etkisi altında hesaplanan amaç fonksiyonu değerleri Tablo 16'da sunulmuştur.

Logaritmik işlem süreleri toplamı ve pozisyon tabanlı öğrenme etkileri altında 15 farklı DHD problemi için GA, HGA, PSO VE PSO-GA yaklaşımları ile elde edilen sonuçlar ve bu sonuçların karşılaştırılması

Tüm DHD problemleri GA, HGA, PSO ve PSO-GA yaklaşımlarıyla çözdürülmüş ve elde edilen sonuçlar karşılaştırılmıştır.

Tablo 17'de tüm DHD problemleri için dört farklı yaklaşımın elde ettiği sonuçlar sunulmuştur. Bu karşılaştırmalar yapılırken, yöntemlerin kullanmış olduğu çözüm adayları sayıları eşitlenmeye çalışılmıştır.

3.2. Bozulma etkisi altında DHD problemi

Bozulma, bir işin işleme başlama zamanını geciktiren her türlü olaydır. İşler çeşitli sebeplerden işleme başlama zamanları gecikebilmektedir. Bu bölümde bozulma oranı 0.10, 0.15 ve 0.20 olarak kullanılmıştır. Bozulma etkisi altında DHD problemi için hesaplama yapılırken Denklem 12c kullanılmıştır.

8 parçalı DHD problemi için hesaplama sonuçları

Literatürde en sık kullanılan küçük boyutlu DHD probleminin çözümü Lingo 11 paket programı ile elde edilerek modelin doğruluğu kanıtlanmıştır. 8 parçalı DHD probleminin görev tanımları, işlem süreleri ve işler arasındaki öncelik ilişkileri Tablo 3'de sunulmuştur. Bozulma oranı 0.10, 0.15 ve 0.20 iken açılan istasyon sayıları Tablo 18'de görülmektedir. Tüm bozulma oranları için açılan istasyon sayısı eşittir ve 5'dir.

Bozulma oranı 0.10 iken işlerin istasyonlara atanma durumu, işlem süreleri ve istasyonların iş tamamlama süreleri Tablo 19'da görülmektedir. İstasyonlarda işlem

tamamlanma zamanları sırasıyla; 39.58, 35.8, 20, 36 ve 23 olarak hesaplanmıştır, tüm istasyonlarda çevrim süresi kısıtı sağlanmıştır.

Tablo 18. Bozulma etkisi altında 8 parçalı DHD probleminin optimum sonuçları

Bozulma oranı	İstasyon sayısı
0.00	4
0.10	5
0.15	5
0.20	5

Tablo 19. Bozulma Oranı 0.1 iken işlerin işlem sürelerinin hesaplanması

İşlerin atandığı istasyon ve sırası	İşin işle süresinin hesaplanması	İşin tamamlanma zamanı
(1,1,1)	14	14
(2,1,2)	$10+0.1*14=11.4$	$14+11.4=25.4$
(3,1,3)	$12+0.1*25.4=14.54$	39.58
(4,2,1)	18	18
(6,2,2)	$16+0.1*18=17.8$	35.8
(7,3,1)	20	20
(8,4,1)	36	36
(5,5,1)	23	23

25 parçalı bozulma etkisi altında DHD problemi için hesaplama sonuçları

DHD literatürünün en yaygın örneği olan 25 parçalı hücreli telefon demontajı probleminin görev tanımları, işlem süreleri ve işler arasındaki öncelik ilişkileri Tablo 9’da sunulmuştur.

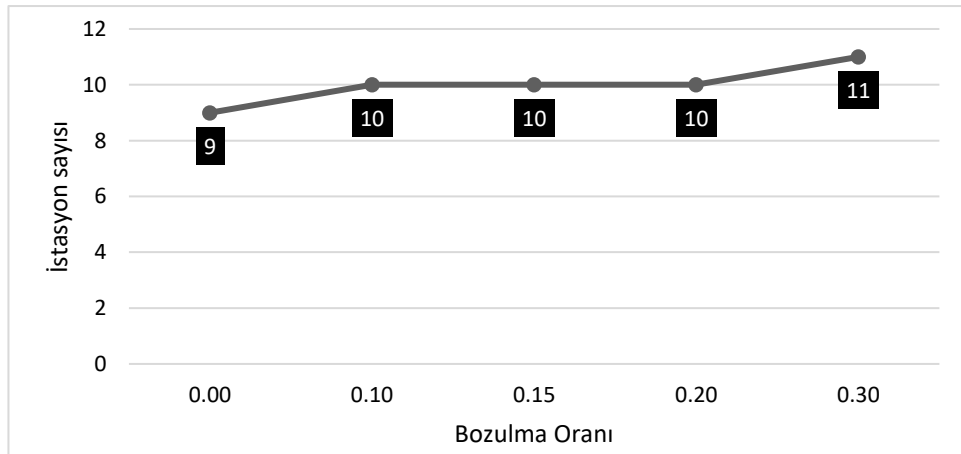
Bozulma oranları 0.1, 0.15 ve 0.2 olarak kullanıldığında 25 görevden oluşan DHD problemi için açılan istasyon sayısı değeri 10 olarak bulunmuştur. Bozulma etkisi dikkate alınmadığında 9 istasyon yeterli olurken, bozulma etkisi altında açılan istasyon sayısında artış görülmektedir. Bozulma oranı 0.1 iken 2 nolu istasyona 12-24-11-20-3 nolu işler sırasıyla atanmıştır. Bu işlerin gerçek işlem süreleri sırasıyla 2-2-2-5-3 iken bozulma etkisi hesaba katıldığında işlem süreleri sırasıyla 2-2.2-2.4-5.7-4.2 olacak şekilde değişmiştir. 24 nolu işin işlem süresi 2 birim zaman iken bozulma etkisi altında 2.2 birim zamana çıkmıştır. İş daha ileri bir pozisyona atandığında ise kendisinden önceki tüm gecikmelerden etkilenmektedir. 3 nolu işin temel işlem süresi 3 birim zaman iken istasyonda 5. sıraya atandığında yeni işlem süresi 4.2 birim zaman olarak gerçekleşmiştir.

Tablo 20. Bozulma etkisi altında 25 parçalı DHD problemi için farklı bozulma oranları ile açılan istasyon sayısı

Bozulma Oranı	İstasyona İşlerin Atanma Sırası	İşlem Süreleri	İş Tamamlama Süresi	Açılan İstasyon Sayısı
Bozulma oranı=0	20-13-11-3-18-14	5-2-2-3-3-2	17	9
	6-10	15-2	17	
	17-9	2-15	17	
	4-12-16-1-21	10-2-2-3-1	18	
	24-23	2-15	17	
	19	18	18	
	5-22-15	10-5-2	17	
	25-7	2-15	17	
	8-2	15-2	18	
Bozulma oranı: 0.1	7-21	15-2.5	17.5	10
	12-24-11-20-3	2-2.2-2.4-5.7-4.2	16.5	
	6	15	15	
	19	18	18	
	18-13-4	3-2.3-10.5	15.8	
	23	15	15	
	22-17-2-10-1	5-2.5-2.8-3-4.3	17.6	
	8	15	15	
	14-25-15-5	2-2.2-2.4-10.7	17.3	
	16-9	2-15.2	17.2	
Bozulma oranı=0.15	10-16-18-25-20	2-2.3-3.6-3.2-6.7	17.8	10
	19	18	18	
	3-2-4	3-2.5-10.9	16.3	
	25-7	2-15.3	17.3	
	9	15	15	
	12-23	2-15.3	17.3	
	21-6	1-15.15	16.1	
	14-13-1-22	2-2.3-3.6-6.2	14.1	
	17-8	2-15.3	17.3	
	24-11-5	2-2.3-10.7	15	
Bozulma oranı=0.2	1-20-16-18	3-5.6-3.72-5.46	17.78	10
	14-4-3	2-10.4-5.48	17.88	
	8	15	15	
	21-5-17	1-10.2-4.24	15.44	
	12-23	2-15.4	17.4	
	19	18	18	
	24-7	2-15.4	17.4	
	13-15-11-2-22	2-2.4-2.88-3.46-7.15	17.88	
	10-9	2-15.4	17.4	
	25-6	2-15.4	17.4	
Bozulma oranı: 0.3	19	18	18	11
	14-4-21	2-10.6-4.8	17.4	
	7	15	15	
	9	15	15	
	17-23	2-15.6	17.6	
	5-14	10-5	15	
	16-8	2-15.6	17.6	
	10-2-24-22	2-2.6-3.4-7.4	15.4	
	3-25-1-12	3-2.9-4.8-5.2	15.9	
	15-6	2-15.6	17.6	
	13-18-20	2-3.6-6.7	12.3	

3 nolu işin işlem süresinde %40 oranında bir artış gözlemlenmektedir. Bozulma oranı 0.2 olduğu durumda da 8 nolu istasyona sırasıyla 13-15-11-2-22 nolu işler atanmıştır. Bu işlerin işlem süreleri sırasıyla 2-2-22-5 iken, bozulma oranı 0.20 iken işlem süreleri sırasıyla 2-2.4-2.88-3.46-7.15 olarak gerçekleşmektedir. Temel işlem süresi 2 birim zaman olan 11 nolu iş istasyonda 3. sıraya atandığında işlem süresi %44 oranında artarak 2.88 olarak gerçekleşmektedir. İşlem süresi 5 birim zaman olan 22 nolu iş aynı istasyonda 5. sıraya atandığında ise gerçekleşen işlem süresi 2.15 birim zaman fazla olmuştur. Bozulma oranları 0.1, 0.15 ve 0.2 olduğunda bu DHD problemi için amaç fonksiyonu değeri değişmemiş, ihtiyaç duyulan istasyon sayısı aynı kalmıştır. Ancak işlerin işlem süreleri bozulma oranı arttıkça artış göstermektedir. Bu yüzden bozulma oranını 0.1 birim daha artırarak 0.30 iken de sonuç elde edilmiştir.

Bozulma oranını 0.3 olarak ele aldığımızda ise açılan istasyon sayısı 11 olarak bulunmuştur. Farklı bozulma oranları için hesaplanan amaç fonksiyonu değerleri Şekil 10'da sunulmuştur.



Şekil 10. İş bozulması etkisi altında 25 parçalı DHD probleminde açılan istasyon sayısı

Bozulma etkisi altında 83 parçalı DHD problemi için hesaplama sonuçları

83 parçalı DHD problemi bozulma etkisi altında, 0.10, 0.15 ve 0.20 bozulma oranları ile çözülmüştür. 83 parçalı DHD probleminin işlem süreleri ve işler arasındaki öncelik ilişkileri Ek 1'de sunulmuştur. Bu problemin çevrim süresi 7571'dir. GA, HGA, PSO ve PSO-GA yaklaşımları ile optimum sonuçlar hesaplanmıştır. Bu bölümde PSO yaklaşımı

ile bulunan optimum iş atama detayları Tablo 21’de sunulmuştur. İşlerin istasyonlara atanma sıraları, yeni işlem süreleri ve açılan istasyon sayısı hesaplanmıştır.

83 parçalı DHD problemi için işler arasındaki öncelik ilişkileri ve çevrim süresi kısıtı sağlandığı ve bozulma etkisi dikkate alınmadığı durumda açılan istasyon sayısı 11’dir. Bu çalışmada da 11 değeri tüm çözüm yaklaşımları ile bulunmuştur. Bozulma oranları sırasıyla 0.10-0.15-0.20 olduğunda, açılan istasyon sayıları da sırasıyla; 14-15-16 olmaktadır. Bozulma etkisi ile ihtiyaç duyulan istasyon sayısında artış meydana gelmiştir. Bozulma etkisi ile işlerin işlem sürelerinde artış olmakta dolayısıyla istasyonlara atanabilen iş sayısı azalabilmektedir.

Bozulma oranı hesaba katılmadığında bir istasyona en fazla 11 iş yerleşirken, bozulma oranı 0.10 iken maksimum 8 iş atanabilmektedir. Bozulma oranı 0.10 iken, 2 nolu istasyona 7 iş atanmıştır. Beşinci sıraya atanan 24 nolu işin gerçek işlem süresi 850 birim zaman iken, bozulma etkisi ile işin gerçekleşen işlem süresi 1200,5 birim zaman olmuştur. İşin işlem süresi % 41 oranında artış göstermiştir. Aynı istasyonda, altıncı sıraya atanan 16 nolu işin temel işlem süresi 483 birim zaman iken, bozulma etkisi altında 954 birim zaman olmuştur. Böylece, işin işlem süresinde %97 artış görülmüştür. İşin istasyondaki

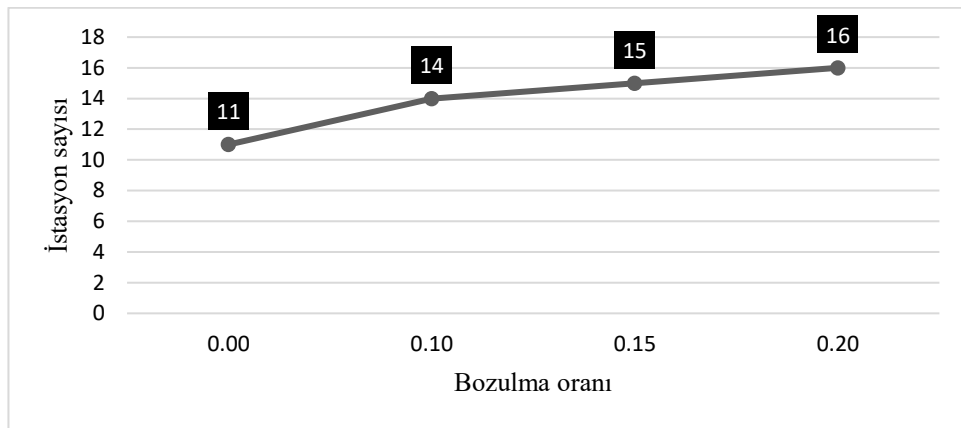
pozisyonuna göre bozulmadan etkilenme oranı da değişmektedir. İş beşinci sıraya atandığında bozulmadan etkilenme oranı %41 iken, altıncı sıraya atanması durumunda bu oran %97.5 olmuştur.

Bozulma oranı 0.15 iken herhangi bir istasyona atanan maksimum iş sayısı 7 iken, bozulma oranı 0.20 olduğunda bu sayı 6 olmuştur. Bozulma oranı dikkate alınmadığında bir istasyona 11 iş yerleşebilmişken, bozulma oranı 0.20 iken bu sayı 6’ya düşmüştür ve yaklaşık olarak %45 oranında azalma meydana gelmiştir. Bozulma etkisi altında 83 parçalı DHD problemi için açılan istasyon sayıları Şekil 11’de görülmektedir.

Şekil 11’de de görüldüğü gibi bozulma oranı arttıkça açılan istasyon sayısı değerinde artış olmuştur. Bozulma oranı 0.10 iken açılan istasyon sayısı 14 olarak elde edilmişken, bozulma oranı 0.20 olduğunda 16 olarak elde edilmiştir.

Tablo 21. Bozulma etkisi altında 83 parçalı DHD probleminin atama detayları

Bozulma Oranı	İstasyona İşlerin Atanma Sırası	İşlem Süreleri	İş Tamamlama Süresi	Açılan İstasyon Sayısı
Bozulma oranı=0				
İstasyon 1	49-70-48-34-81-43-40-15-32-73	584-887-578-1234-450-792-83-577-842-1100	7127	11
İstasyon 2	67-74-75-13-22	578-2543-764-500-847	5232	
İstasyon 3	6-20-68-66-39-38-21	2881-233-578-578-1310-1251-408	7239	
İstasyon 4	51-25-23-78-72-30-3	578-780-767-1164-1298-1004-1838	7429	
İstasyon 5	7-45-65-56-2-42-55	2231-594-578-578-985-1288-578	6832	
İstasyon 6	36-17-41-5-80-37	1266-880-494-1700-2100-792	7232	
İstasyon 7	18-26-52-11-77-69-50-9-14-64	667-912-578-700-701-467-578-793-1133-578	7107	
İstasyon 8	12-19-47-4-83	464-600-822-973-3691	6550	
İstasyon 9	1-71-61-33-63-76-8-16-58-79-53	1673-396-578-829-578-357-1040-483-578-286-578	7376	
İstasyon 10	35-31-29-59-10-54-28	1143-713-714-578-1250-578-1883	6859	
İstasyon 11	60-62-24-46-82-57-44-27	578-578-850-578-1300-578-578-748	5788	
Bozulma oranı: 0.1				
İstasyon 1	8-29-64-7-22	1040-818-763,8-2493,2-1358,5	6473,5	14
İstasyon 2	43-18-4-58-24-16-55	792-746,2-1126,8-844,5-1200,5-954-1144,5	6808,5	
İstasyon 3	5-72-83	1700-1468-4007,8	7175,8	
İstasyon 4	51-47-14-53-44-80	578-879,8-1278,6-851,6-936,8-2552,5	7077,3	
İstasyon 5	36-13-78-9-76-35	1266-626,6-1353,3-1117,6-793,3-1658,7	6815,5	
İstasyon 6	71-12-3-17-57-11-67	396-503,6-1928-1162,7-977-1196,7-1194,4	7358,4	
İstasyon 7	41-32-56-74-48	494-891,4-716,5-2753,2-1063,5	5918,6	
İstasyon 8	50-68-54-26-40-70-63	578-635,8-699,4-1103,3-384,6-1227,1-1040,8	5669	
İstasyon 9	75-82-52-42-33-27	764-1376,4-792-1581,2-1280,4-1327,4	7121,4	
İstasyon 10	77-38-65-30-73-39	701-1321,1-780,2-1284,2-1508,6-1869,5	7464,6	
İstasyon 11	60-2-66-6-46	578-1042,8-740-3117,1-1125,8	6603,7	
İstasyon 12	25-69-20-21-81-10-31-59	780-545-365,5-577,1-676,7-1544,4-1161,9-1143,1	6793,7	
İstasyon 13	28-23-61-34-1	1883-955,3-861,8-1604-2203,4	7507,5	
İstasyon 14	79-37-62-15-49-19-45	286-820,6-688,7-756,5-839,2-939,1-1027	5357,1	
Bozulma oranı=0.15				
İstasyon 1	4-9-55-76-1-25	973-938,9-864,8-773,5-2205,5-1643,4	7399,1	15
İstasyon 2	74-8-49-47	2543-1421,5-1178,7-1593,5	6736,7	
İstasyon 3	38-81-46-36-21-24	1251-637,6-861,3-1678,5-1072,3-1675,1	7175,8	
İstasyon 4	62-15-28-10-77	578-663,7-2069,3-1746,6-1459,6	6517,2	
İstasyon 5	2-82-22-50-73	985-1447,7-1211,7-1124,7-1815,4	6584,5	
İstasyon 6	69-33-45-61-30-78	467-899-798,9-902,7-1464,2-1843,8	6375,6	
İstasyon 7	53-23-79-17-35-31	578-853,7-500,7-1169,9-1608,3-1419,6	6130,2	
İstasyon 8	51-14-16-64-7	578-1219,7-752,6-960,5-2757,6	6268,4	
İstasyon 9	60-19-44-68-52-43-18	578-686,7-767,7-882,8-1015,3-1381,6-1463,8	6775,9	
İstasyon 10	32-39-56-29-54-67	842-1436,3-919,7-1193,7-1236,7-1422,3	7050,7	
İstasyon 11	71-11-6-72-13	396-759,4-3054,3-1929,4-1420,9	7560	
İstasyon 12	27-34-58-83	748-1346,2-892,1-4138,9	7125,2	
İstasyon 13	26-5-3-40-66	912-1836,8-2250,3-832,9-1452,8	7284,8	
İstasyon 14	59-20-42-75-63-80	578-319,7-1422,6-1112,1-1092,9-2778,8	7304,1	
İstasyon 15	57-70-48-65-41-12-37	578-973,7-810,7-932,4-988,2-1106,4-1600,4	6989,8	
Bozulma oranı=0.2				
İstasyon 1	38-79-9-27-19-18	1251-536,2-1150,4-1335,5-1454,6-1812,6	7540,3	16
İstasyon 2	4-34-17-67-30	973-1428,6-1360,3-1330,4-2022,5	7114,8	
İstasyon 3	77-41-47-76-10-63	701-634,2-1089-841,8-1903,2-1611,9	6781,1	
İstasyon 4	74-22-59-78	2543-1355,6-1357,7-2215,3	7471,6	
İstasyon 5	32-54-29-66-82-65	842-746,4-1031,7-1102-2044,4-1731,3	7497,8	
İstasyon 6	52-31-83-55	578-828,6-3972,3-1653,8	7032,7	
İstasyon 7	40-12-2-69-26-80	83-480,6-1097,7-799,3-1404,1-2872,9	6737,6	
İstasyon 8	71-25-7-43-64	396-859,2-2482-1539,4-1633,3	6909,9	
İstasyon 9	6-68-57-53	2881-1154,2-1385-1662	7082,2	
İstasyon 10	72-49-16-51-15-75	1298-843,6-911,3-1188,6-1425,3-1897,4	7564,2	
İstasyon 11	60-21-44-58-1-37	578-523,6-798,3-958-2244,6-1812,5	6915	
İstasyon 12	20-81-8-11-50-70	233-496,6-1185,9-1083,1-1177,7-1722,3	5898,6	
İstasyon 13	56-39-35-13-33	578-1425,6-1543,7-1209,5-1780,4	6537,2	
İstasyon 14	24-62-42-23-61	850-748-1607,6-1408,1-1500,7	6114,4	
İstasyon 15	73-46-3-28	1100-798-2217,6-2706,1	6821,7	
İstasyon 16	14-48-45-36-5	1133-804,6-981,5-1849,8-2653,8	7422,7	



Şekil 11. Bozulma etkisi altında 83 parçalı DHD probleminde açılan istasyon sayıları

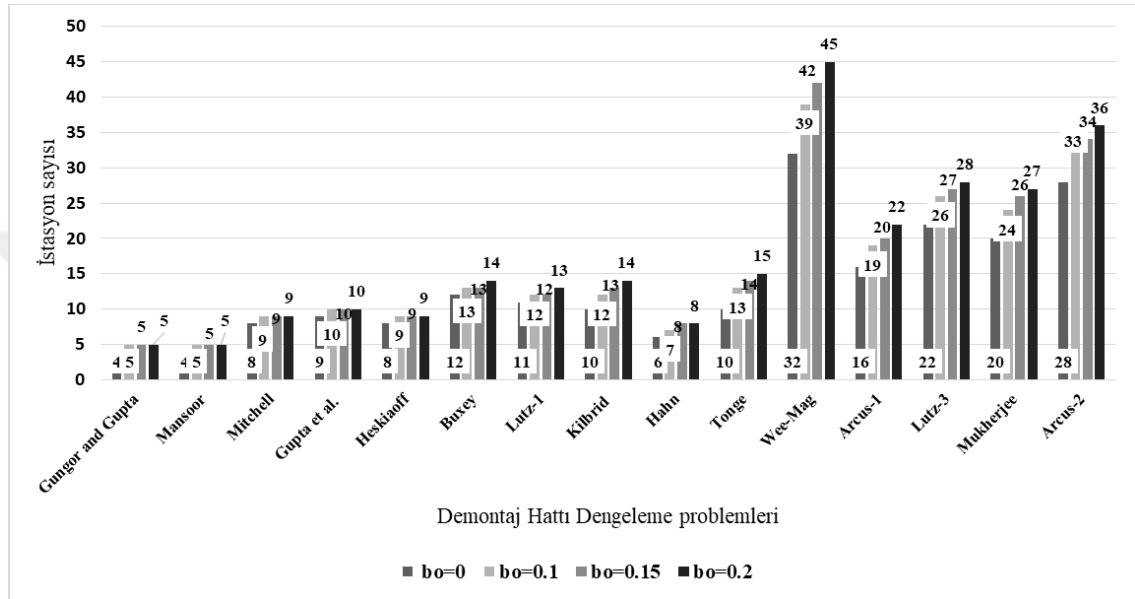
Literatürdeki 15 farklı DHD problemi için bozulma etkisi altında hesaplama sonuçları

Literatürde bilinen 15 DHD problemi (Scholl, 1995) bozulma etkisi altında ele alınmıştır. Üç farklı bozulma oranı, 0.10, 0.15, 0.20 kullanılmıştır. Bozulma, bir işin işe başlama zamanının gecikmesidir. Bozulma etkisi ile işlerin işlem sürelerinde artış görülmektedir. Gerçek hayatta da işlerin başlama zamanlarında, işçiden, hava durumundan, ortamdan veya tezgahattan kaynaklı sebeplerden gecikmeler meydana gelebilmektedir. Literatürdeki DHD problemleri için bozulma etkisi dikkate alınarak hesaplanan amaç fonksiyonu değerleri Tablo 22’de sunulmuştur.

Tablo 22. Farklı DHD problemleri için bozulma etkisi altında açılan istasyon sayıları

Yazar	n	C	*Optimum	PSO			
			Sonuç	bo=0	bo=0.1	bo=0.15	bo=0.2
Gungor and Gupta	8	40	4	4	5	5	5
Mansoor	11	48	4	4	5	5	5
Mitchell	21	15	8	8	9	9	9
Gupta et al.	25	18	9	9	10	10	10
Heskiaoff	28	138	8	8	9	9	9
Buxey	29	30	12	12	13	13	14
Lutz-1	32	1414	11	11	12	12	13
Kilbrid	45	56	10	10	12	13	14
Hahn	53	2806	6	6	7	8	8
Tonge	70	364	10	10	13	14	15
Wee-Mag	75	49	32	32	39	42	45
Arcus-1	83	5048	16	16	19	20	22
Lutz-3	89	79	22	22	26	27	28
Mukherjee	94	222	20	20	24	26	27
Arcus-2	111	5785	27	28	33	34	36

Tablo 22’de görüldüğü gibi bozulma oranı 0.10,0.15 ve 0.20 iken, 8, 11, 21, 25 ve 28 parçadan oluşan veri setleri için sırasıyla; 5-5-9-10-9 olacak şekilde açılan istasyon sayısı eşit bulunmuştur. Bozulma etkisinin dikkate alınmadığı duruma göre amaç fonksiyonu değerinde artış gözlemlenmiştir. Ancak, diğer tüm problemlerde bozulma oranı arttıkça açılan istasyon sayısı da artmıştır.



bo=bozulma oranı

Şekil 12. 15 farklı DHD probleminin bozulma etkisi altında açılan istasyon sayıları

70 parçalı DHD problemi için bozulma etkisi dikkate alınmadığında 10 istasyon açılması gerekirken, bozulma oranları sırasıyla; 0.10-.15-0.20 olduğunda açılan istasyon sayıları sırasıyla; 13-14-15 olarak elde edilmiştir. 94 parçadan oluşan DHD problemi için bozulma etkisi dikkate alınmadığında 20 istasyon açılması yeterli bulunurken, bozulma oranları sırasıyla; 0.10-.15-0.20 olduğunda açılan istasyon sayıları sırasıyla; 24-26-27 olarak elde edilmiştir. Bozulma etkisi ile işlerin işlem sürelerinde artış görüldüğü için, istasyonlara çevrim süresini aşmayacak şekilde atanabilecek iş sayısında azalma görülmektedir. Dolayısıyla, bu durum yeni istasyonların açılmasını gerekli kılmaktadır. Bu durum, problemin amaç fonksiyonu değerinde artışa sebep olmaktadır.

Bozulma etkisi altında 15 farklı DHD problemi için GA, HGA, PSO VE PSO-GA yaklaşımları ile elde edilen sonuçlar ve bu sonuçların karşılaştırılması

Tüm DHD problemleri bozulma etkisi altında GA, HGA, PSO ve PSO-GA yaklaşımlarıyla çözdürülmüş ve elde edilen sonuçlar karşılaştırılmıştır. Bu aşamada, her

Tablo 23. GA, HGA, PSO ve PSO-GA yaklaşımlarının çözüm sonuçları (Bozulma etkisi altında)

Çevrim sür	Bozulma oran	opt*	f(GA)	Min–Max	f (HGA)	Min–Max	f (PSO)	Min–Max	f (PSO-GA)	Min–Max
40	0	4	4	4-5	4	4	4	4-5	4	4
	0.1	-	5	5	5	5	5	5	5	5
	0.15	-	5	5	5	5	5	5	5	5
	0.2	-	5	5	5	5	5	5	5	5
48	0	4	4	4-5	4	4	4	4-5	4	4
	0.1	-	5	5	5	5	5	5	5	5
	0.15	-	5	5	5	5	5	5-6	5	5
	0.2	-	5	5	5	5	5	5-6	5	5
15	0	8	8	8	8	8	8	8	8	8
	0.1	-	8	8-9	8	8-9	9	9	8	8-9
	0.15	-	9	9-10	9	9	9	9-10	9	9
	0.2	-	9	9-10	9	9	9	9-10	9	9
18	0	9	9	9-10	9	9-10	9	9-11	9	9-10
	0.1	-	10	10-11	10	10-11	11	11-12	10	10-11
	0.15	-	11	11-12	10	10-11	11	11-12	10	10-11
	0.2	-	11	11-12	10	10-11	11	11-12	10	10-11
138	0	8	8	8	8	8	8	8-9	8	8
	0.1	-	9	9-10	9	9	9	9-10	9	9
	0.15	-	9	9-10	9	9-10	9	9-11	9	9-10
	0.2	-	9	9-10	9	9-10	10	10-11	9	9-10
30	0	12	12	12-13	12	12	12	12-13	12	12
	0.1	-	13	13-14	13	13-14	13	13-15	13	13-14
	0.15	-	14	14-15	13	13-14	14	14-15	13	13-14
	0.2	-	14	14-15	13	13-15	14	14-16	14	14-15
1414	0	11	11	11	11	11	11	11-12	11	11
	0.1	-	12	12-13	12	12	12	12-13	12	12
	0.15	-	12	12-13	12	12-13	12	12-13	12	12-13
	0.2	-	13	13-14	13	13	13	13-14	13	13
56	0	10	11	11	11	11	11	11-12	11	11
	0.1	-	12	12-13	12	12-13	12	12-13	12	12-13
	0.15	-	13	13-14	13	13-14	13	13-14	13	13
	0.2	-	13	13-14	13	13-14	13	13-15	13	13-14
2806	0	6	6	6	6	6	6	6-7	6	6
	0.1	-	7	7-8	7	7-8	7	7-8	7	7-8
	0.15	-	8	8-9	8	8-9	8	8-9	8	8
	0.2	-	8	8-9	8	8-9	8	8-10	8	8-9
364	0	10	10	10-11	10	10-11	10	10-11	10	10-11
	0.1	-	13	13-14	13	13-14	13	13-14	13	12
	0.15	-	14	14-15	14	14-15	14	14-15	14	14
	0.2	-	15	15-16	15	15	15	15-16	15	15
49	0	33	33	33-34	33	33-34	33	33-35	33	33-34
	0.1	-	39	39-41	39	39-41	39	39-42	39	39-41
	0.15	-	42	42-45	42	42-43	42	42-45	42	42-43
	0.2	-	45	45-48	45	45-47	46	46-48	45	45-47
7571	0	11	11	11	11	11	11	11	11	11
	0.1	-	14	14	14	14	14	14	14	14
	0.15	-	15	15-16	15	15-16	15	15-16	15	15
	0.2	-	16	16-17	16	16-17	16	16-17	16	16-17
79	0	22	22	22-25	22	22-23	22	22-25	22	22-23
	0.1	-	26	26-27	26	26-27	26	26-27	26	26-27
	0.15	-	27	27-29	27	27-28	27	27-29	27	27-28
	0.2	-	28	28-30	28	28-29	28	28-30	28	28-29
222	0	20	21	21-22	21	21-22	21	21-22	21	21-22
	0.1	-	24	24-25	24	24-25	24	24-25	24	24-25
	0.15	-	26	26-27	25	25-27	26	26-27	25	25-27
	0.2	-	27	27-28	27	27-28	27	27-28	27	27-28
5785	0	29	29	29-31	29	29-31	29	29-30	29	29-31
	0.1	-	33	33-35	33	33-35	33	33-35	33	33-35
	0.15	-	34	34-36	34	34-36	35	35-37	34	34-36
	0.2	-	36	36-38	36	36-38	36	36-38	36	36-38

yöntem yukarıda belirtmiş olduğumuz sayıda iterasyon çalıştırılarak optimum sonuca ulaşma oranları hesaplanmıştır. Böylece, çözüm yöntemlerinin performansları karşılaştırılmıştır.

Tablo 23’de bozulma etkisi altında tüm DHD problemleri için dört farklı yaklaşımın elde ettiği sonuçlar sunulmuştur. Bu karşılaştırmalar yapılırken, yöntemlerin kullanmış olduğu çözüm adayı sayıları eşitlenmeye çalışılmıştır.

3.3. Eşzamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında DHD problemi

Eşzamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında DHD problemleri için açılan istasyon sayıları belirlenmiştir. İşler istasyonlara atanırken, işler arasındaki öncelik ilişkileri ve istasyonların çevrim süresi kısıtları dikkate alınmıştır. Öğrenme etkisi, işlerin işlem sürelerini azaltıcı etki ederken amaç fonksiyonu üzerinde azaltıcı etki göstermektedir. Ancak, bozulma etkisi öğrenmenin aksine işlerin işlem süreleri üzerinde artırıcı etki yaptığı için amaç fonksiyonu üzerinde de artırıcı bir etkiye sahiptir. Bu bölümde, zıt etkilere sahip olan iki etkinin eş zamanlı uygulanması çalışılmıştır.

Bu çalışmada literatürde yer alan 15 test problemi (Scholl,1995) için sonuçlar elde edilmiş, pozisyon tabanlı öğrenme ve bozulma etkisi tartışılmıştır. Öğrenme etkisi için kullanılan parametre değerleri 0.90, 0.80 ve 0.70 iken bozulma etkisi parametreleri ise 0.10 ve 0.20’dir.

Eşzamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 8 parçalı DHD problemi

8 parçalı veri seti, DHD literatüründe ilk ve en sık kullanılan veri setidir (Bkz. Tablo 3). Bu veri seti için herhangi bir etkinin dikkate alınmadığı durum ve pozisyon tabanlı öğrenme ve bozulma etkilerinin eş zamanlı uygulandığı durum için sonuçlar elde edilmiştir. Öğrenme ve bozulma etkisi altında işin işlem süresi pozisyonuna bağlı olarak değişiklik göstermektedir.

Pozisyon tabanlı öğrenme ve bozulma etkilerinin dikkate alınmadığı durumda, yani literatürdeki çalışmalarda bulunan istasyon sayısı değeri 4’dür. Bu çalışmada da aynı şartlarda amaç fonksiyonu değeri 4 olarak bulunduğu Tablo 24’de gösterilmiştir. Görevlerin istasyonlara atanması ise şu şekildedir: Görev 1-2-3, İstasyon-1’e; Görev 5-6, İstasyon-2’ye; Görev 7-4, İstasyon-3’e ve Görev 8, İstasyon-4’e atanmaktadır.

Tablo 24. 8 parçalı DHD problemi için optimal sıra ve iş istasyonu sayısı

öğrenme oranı(öo) bozulma oranı(bo)	optimal sıra	iş istasyonu sayısı
öo=0.00, bo=0	1-2-3, 5-6, 7-4, 8	4
öo=0.90, bo=0.10	6-7, 1-5, 2-3-4, 8	4
öo=0.90, bo=0.20	6-5, 8, 7-4, 1-2-3	4
öo=0.80, bo=0.10	7-4, 5-3, 1-2-6, 8	4
öo=0.80, bo=0.20	8, 2-3-4, 7-6, 1-5	4
öo=0.70, bo=0.10	3-8, 2-5-7, 6-1-4	3
öo=0.70, bo=0.20	7-1, 8, 2-6-4, 5-3	4

Üç farklı pozisyon tabanlı öğrenme etkisi oranı ve iki farklı bozulma etkisi oranının farklı kombinasyonlarının eş zamanlı etkisi altında elde edilen sonuçlar Tablo 24’de sunulmuştur. 8 parçalı DHD problemi için amaç fonksiyonu değeri, öğrenme oranı 0.90 ve 0.80 iken, bozulma etkisi 0.10 ve 0.20 iken, bu iki etkinin eş zamanlı uygulanması ile yine 4 olarak bulunmuştur. Öğrenme oranı 0.70 ve bozulma etkisi 0.10 iken açılacak istasyon sayısında azalma görülmüş ve istasyon sayısı 3 olarak elde edilmiştir. Bu durumda, işlerin istasyonlara atanma sıraları şu şekildedir: 3-8 nolu işler İstasyon-1’e, 2-5-7 nolu işler istasyon-2’ye ve 6-1-4 nolu işler, İstasyon-3’e atanmaktadır. Tablo-24’de de görüldüğü gibi öğrenme etkisi 0.70 ve bozulma etkisi 0.10 olduğu durumda amaç fonksiyonu için daha düşük bir değer elde edilmiştir.

Tablo 25. Öğrenme oranı 0.70 ve bozulma oranı 0.10 iken eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında işlem süreleri

İşlerin atandığı istasyon ve sırası	İşin işlem süresinin hesaplanması	İşin tamamlanma zamanı
(3,1,1)	12	14
(8,1,2)	$(36+0.1*12)*2^{-0,514}=26.05$	$12+26.05=38.05$
(2,2,1)	10	10
(5,2,2)	$(23+0.1*10)*2^{-0,514}=16.8$	$10+16.8=26.8$
(7,2,3)	$(20+0.1*26.8)*3^{-0,514}=12.89$	$26.8+12.89=39.69$
(6,3,1)	16	16
(1,3,2)	$(14+0.1*16)*2^{-0,514}=10.92$	$16+10.92=26.92$
(4,3,3)	$(18+0.1*30.92)*3^{-0,514}=12.89$	$26.92+12.89=38.68$

Öğrenme oranı 0.70 ve bozulma oranı 0.10 iken bulunan sonuçta, istasyona atanan işler, işlerin işlem sürelerinin hesaplanması ile ilgili detaylar Tablo 25’de sunulmuştur.

İstasyonlarda işler, çevrim süresini aşmadan ve öncelik ilişkilerini sağlayacak şekilde atanmıştır. Buna göre 1 nolu istasyona 3-8 nolu işler, 2 nolu istasyona 2-5-7 nolu işler ve 3 nolu istasyona 6-1-4 nolu işler sırasıyla atanmıştır. İstasyonlarda işlerin tamamlanma süreleri ise sırasıyla; 38.05-39.69-38.68 olarak elde edilmiştir.

Eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi

25 parçadan oluşan cep telefonu demontajı problemi, demontaj hattı literatüründe en sık kullanılan veri setlerinden biridir. Demontaj görev tanımları, görev süreleri ve öncülleri Tablo-9'da sunulmuştur. Çevrim süresi 18'dir ve görevler arasındaki öncelik ilişkileri dikkate alınarak açılacak minimum istasyon sayısı hesaplanmıştır. Pozisyon tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için GA sezgisel yöntemiyle elde edilen sonuçlar, literatürde bildirilen sonuçlarla karşılaştırılmıştır. Kullandığımız GA yaklaşımı Tablo 10'da sunulduğu gibi optimum sonucu elde etmektedir.

Eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD probleminin farklı öğrenme ve bozulma oranları ile elde edilen sonuçları Tablo 26'da sunulmuştur. İşlerin öncelik ilişkilerini sağlayacak ve çevrim sürelerini aşmayacak şekilde istasyonlara atanma sıraları, istasyonların iş tamamlama süreleri ve problemin amaç fonksiyonu değeri elde edilmiştir. Öğrenme oranları 0.90, 0.80, 0.70 ve bozulma oranları 0.10 ve 0.20 olarak ele alınmıştır.

Öğrenme oranı 0.90 iken, bozulma oranı 0.10 iken 9 istasyon açılırken, bozulma oranı 0.20 iken 10 istasyon açılması gerekli bulunmuştur. Öğrenme etkisi, işlerin işlem süresinin azalttığı için açılan istasyon sayısını azaltıcı etki gösterirken, bozulma etkisi artırıcı etki göstermektedir. Öğrenme oranı sabit tutulduğunda, bozulma etkisi artınca, açılan istasyon sayısında artış görülmüştür. Benzer şekilde öğrenme oranı 0.70 iken, bozulma oranı 0.10 olduğunda amaç fonksiyonu değeri 6 bulunmuştur. Bozulma oranı 0.20 olduğunda ise amaç fonksiyonu değeri 7 bulunmuştur. Böylece, amaç fonksiyonu değerinde yaklaşık olarak %16 artış olmuştur. Bozulma etkisi 0.10 iken, temel işlem süresi 2 birim zaman olan 16 nolu iş 6 nolu istasyonda beşinci sıraya atandığında işin işlem süresi 1.2 birim zaman olarak hesaplanmışken, aynı iş bozulma oranı 0.20 iken ise 16 nolu iş 5 nolu istasyonda beşinci sıraya atanmıştır ve işlem süresi 1.9 olarak

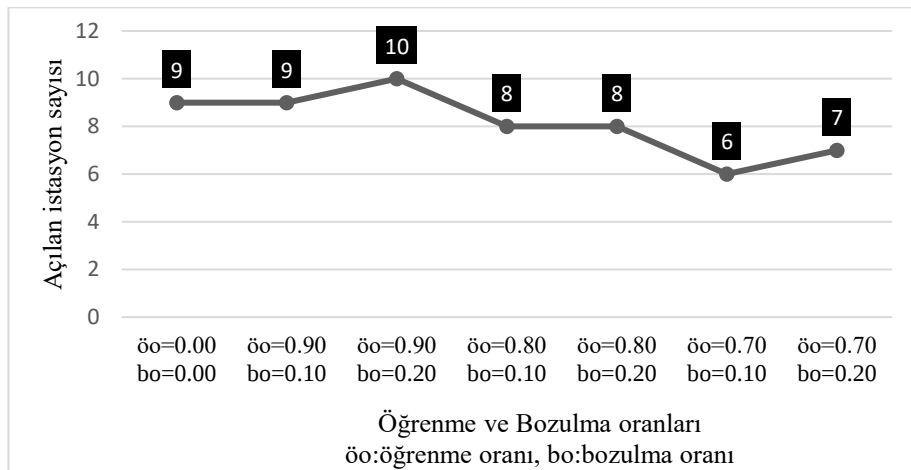
Tablo 26. Pozisyon tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için optimal sıra ve iş istasyonu sayısı

Senaryo	Atanan İşler	İşlem süreleri	İş Tamamlama süresi	Açılan istasyon sayısı
P25 with $lr=0.00$, $dr=0.00$, $ct=18$				
Station 1	21-23-14	1-15-2	18	9
Station 2	7-24	15-2	17	
Station 3	13-5-25-10-15	2-10-2-2-2	18	
Station 4	4-22-16	10-5-2	17	
Station 5	19	18	18	
Station 6	1-6	3-15	18	
Station 7	11-8	2-15	17	
Station 8	12-2-17-3-20	2-2-2-3-5	14	
Station 9	9-18	15-3	18	
P25 with $lr=0.90$, $dr=0.10$, $ct=18$				
Station 1	1-23	3-13,8	16,8	9
Station 2	12-3-8	2-2,9-13,1	18	
Station 3	21-14-24-10-20-11-16	1-1,9-1,9-2-4,5-2,4-2,5	16,2	
Station 4	19	18	18	
Station 5	22-5-2	5-9,4-2,9	17,3	
Station 6	18-7	3-13,8	16,8	
Station 7	13-4-25-15	2-9,2-2,6-2,7	16,5	
Station 8	9	15	15	
Station 9	17-6	2-13,7	15,7	
P25 with $lr=0.90$, $dr=0.20$, $ct=18$				
Station 1	23	15	15	10
Station 2	8	15	15	
Station 3	11-9	2-13,9	16	
Station 4	14-21-16-12-25-24	2-1,3-2,2-2,5-2,8-3,2	14	
Station 5	20-15-5	5-2,7-9,8	17,5	
Station 6	18-6	3-14	17	
Station 7	3-10-22-13	3-2,3-5,1-3,3	13,7	
Station 8	1-4-2	3-9,5-3,8	16,3	
Station 9	17-7	2-13,9	15,9	
Station 10	19	18	18	
P25 with $lr=0.8$, $dr=0.1$, $ct=18$				
Station 1	16-13-20-2-25-5	2-1.76-3.77-1.76-1.75-6.24	17.28	8
Station 2	18-8-11	3-12.24-2.47	17.71	
Station 3	24-7-21	2-12.16-1.7	15.86	
Station 4	9-15	15-2.8	17.8	
Station 5	19	18	18	
Station 6	14-23	2-12.16	14.16	
Station 7	22-17-4-1	5-2.7.5-2.86	17.36	
Station 8	12-3-10-6	2-2.56-1.72-10	16.28	
P25 with $lr=0.8$, $dr=0.2$, $ct=18$				
Station 1	14-24-13-15-16-5	2-1.92-1.96-2-2.2-6.7	16.78	8
Station 2	2-8	2-12.3	14.3	
Station 3	19	18	18	
Station 4	11-22-6	2-4.42-11.42	17.74	
Station 5	17-12-23	2-2-11	15	
Station 6	1-9	3-12.5	15.5	
Station 7	10-18-21-7	2-2.7-1.4-10.4	16.5	
Station 8	25-20-4-3	2-4.32-7.9-3.74	17.96	
P25 with $lr=0.7$, $dr=0.1$, $ct=18$				
Station 1	18-20-11-23-24	3-3.7-1.5-7.8-1.6	17.6	6
Station 2	21-9-5	1-10.6-6.3	17.9	
Station 3	17-2-4-7	2-1.5-5.9-7.9	17.3	
Station 4	1-3-8-10-13	3-2.3-8.8-1.7-1.6	17.4	
Station 5	6-25	15-2.5	17.5	
Station 6	14-15-22-12-16-19	2-1.5-3.1-1.3-1.2-7.5	16.6	
P25 with $lr=0.7$, $dr=0.2$, $ct=18$				
Station 1	12-17-5-18-4	2-1.7-6.1-2.4-5.5	17.7	7
Station 2	14-2-15-13-19-11	2-1.7-1.5-1.5-8.5-2	17.2	
Station 3	22-25-7	5-2.1-9.3	16.4	
Station 4	1-20-8	3-3.9-9.3	16.2	
Station 5	21-10-24-9-16	1-1.5-1.4-7.8-1.9	13.6	
Station 6	23	15	15	
Station 7	3-6	3-10.9	13.9	

hesaplanmıştır. Yani bir iş aynı sıraya atandığı halde bozulma etkisinin 0.10 birim artması ile işlem süresinde yaklaşık olarak %53 artış gerçekleşmiştir.

Bozulma oranı 0.10 iken, öğrenme oranları sırasıyla 0.90-0.80-0.70 olduğunda açılan istasyon sayıları sırasıyla 9-8-6 olarak hesaplanmıştır. Bozulma etkisi sabit iken, öğrenme oranı düştükçe açılan istasyon sayısında da düşüş görülmektedir. Benzer şekilde bozulma oranı 0.20 iken, öğrenme oranları sırasıyla 0.90-0.80-0.70 olduğunda açılan istasyon sayıları sırasıyla 10-8-7 olarak hesaplanmıştır. Bozulma oranı 0.10 olarak sabit tutulduğunda, temel işlem süresi 2 birim zaman olan 11 nolu iş, öğrenme oranı 0.80 iken 2 nolu istasyonda üçüncü sıraya atandığında işlem süresi 2.47 birim zaman olarak hesaplanmıştır. Aynı iş, öğrenme oranı 0.70 olduğunda 1 nolu istasyonda üçüncü sıraya atanmıştır ve işlem süresi 1.5 birim zaman olarak hesaplanmıştır. Öğrenme oranı 0.80 iken 11 nolu işin işlem süresinde bozulmanın da etkisiyle artış görülürken, öğrenme oranı 0.70 olduğunda azalma görülmüştür. Öğrenme oranı 0.80 iken işin işlem süresinde %23 artış görülmüştür. Öğrenme oranı 0.70 iken ise işin işlem süresinde %25 azalma görülmüştür.

Tablo 26’da da görüldüğü gibi, öğrenme oranı sabit tutulduğunda bozulma oranı arttıkça işlerin işlem sürelerinde artış görülmekte ve sonuç olarak amaç fonksiyonu değerinde de artış olmaktadır. Bozulma oranı sabit tutulduğunda ise, öğrenme oranı azaldıkça işlerin işlem süreleri de azalmakta ve dolayısıyla, amaç fonksiyonu değerinde de azalma meydana gelmektedir. Tüm istasyonların işlerin işlemlerini tamamlama süreleri çevrim süresini aşmamaktadır. Böylece, çevrim süresi kısıtı da sağlanmıştır.



Şekil 13. Eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için açılan istasyon sayılarının değişimi

Farklı senaryolar ile amaç fonksiyonundaki değişim Şekil 13’de sunulmuştur.

Eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi

Bu bölümde, 83 parçalı DHD problemi eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında, 0.90, 0.80, 0.70 öğrenme oranları ve 0.10, 0.20 bozulma oranları ile çözülmüştür. 83 parçalı DHD probleminin işlem süreleri ve işler arasındaki öncelik ilişkileri Ek 1’de sunulmuştur. Bu problemin çevrim süresi 7571’dir. GA, HGA, PSO ve PSO-GA yaklaşımları kullanılarak 83 parçalı DHD problemi için pozisyon tabanlı öğrenme ve bozulma etkileri ile optimum sonuçlar elde edilmiştir. Bu bölümde HGA yaklaşımı ile bulunan optimum iş atama detayları Tablo 27’de sunulmuştur. İşlerin istasyonlara atanma sıraları, yeni işlem süreleri ve açılan istasyon sayısı belirtilmiştir.

83 parçalı DHD problemi için işler arasındaki öncelik ilişkileri ve çevrim süresi kısıtı sağlanacak şekilde hesaplama yapılırken, pozisyon tabanlı öğrenme ve bozulma etkileri dikkate alınmadığı durumda açılan istasyon sayısı 11’dir. Öğrenme oranı 0.90 iken, bozulma oranı 0.10 için amaç fonksiyonu değeri 11 olarak elde edilmişken, bozulma oranı 0.20 olduğunda ise 13 istasyon açılması gerekli bulunmuştur. Öğrenme etkisi işlerin işlem sürelerini azaltırken, bozulma etkisi artırdığı için açılan istasyon sayısı bozulma etkisi arttıkça artmıştır. Öğrenme ve bozulma etkileri dikkate alınmadığında, bir istasyona maksimum 9 adet iş atanabilmişken, öğrenme oranı 0.90 ve bozulma oranı 0.20 olduğunda herhangi bir istasyona atanan maksimum iş sayısı 8 adet olmuştur. İşlerin öğrenilmesiyle işlem sürelerinde bir azalma olmuştur, ancak bozulma oranı 0.20, öğrenme oranı 0.90 iken bozulma etkisi öğrenme etkisine baskın gelmiştir ve işlerin işlem sürelerinde öğrenme etkisine rağmen artış görülmüştür. Dolayısıyla istasyonlara atanabilen iş sayılarında düşüşler meydana gelebilmiş ve yeni istasyonların açılması gerekmiştir.

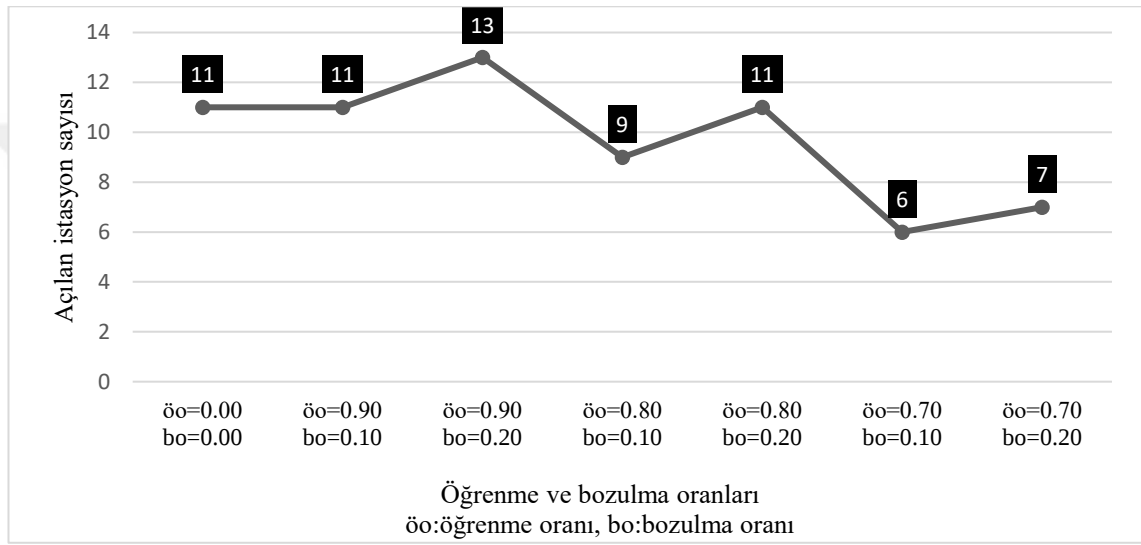
Bozulma oranı sabit tutulup öğrenme etkisindeki değişimin, açılan istasyon sayısında ne tür bir değişikliğe sebep olduğu Tablo 27’de görülebilmektedir. Bozulma oranı 0.10 iken öğrenme oranları sırasıyla 0.90-0.80-0.70 olduğunda, açılan istasyon sayıları sırasıyla; 11-9-6 olarak elde edilmiştir.

Tablo 27. Pozisyon tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi için iş istasyonu sayısı

Senaryo	Atanan İşler	İşlem süreleri	İş Tamamlama süresi	Açılan istasyon sayısı
P83 with Ir=0, dr=0, ct=7571				
Station 1	28-37-46-48-53-13-60-65-72	1883-792-578-578-578-500-578-578-1298	7363	11
Station 2	5-15-35-27-61-70-1	1700-577-1143-748-578-887-1673	7306	
Station 3	62-57-83-80-50	578-578-3691-2100-578	7525	
Station 4	81-9-39-49-18-38-7	450-793-1310-584-667-1251-2231	7286	
Station 5	31-63-25-43-73-14-42	713-578-780-792-1100-1133-1288	6384	
Station 6	30-40-66-41-23-16-3-24-17	1004-83-578-494-767-483-1838-850-880	6977	
Station 7	34-79-54-8-67-6	1234-286-578-1040-578-2881	6597	
Station 8	10-36-33-20-51-11-4	1250-1266-829-233-578-700-973	5829	
Station 9	74-64-19-47-44-12-76-55-45	2543-578-600-822-578-464-357-578-594	7114	
Station 10	32-78-77-22-29-69-82-52-58	842-1164-701-847-714-467-1300-578-578	7191	
Station 11	26-56-68-75-21-59-2-71	912-578-578-764-408-578-985-396	5199	
P83 with Ir=0.90, dr=0.10, ct=7571				
Station 1	69-59-47-23-20-43-67-83	467-562,2-782,7-768-384,4-828,9-712,2-3019,2	7524,6	11
Station 2	50-57-31-10-80-18	578-572,2-700,7-1162,4-1880,2-880,7	5774,2	
Station 3	58-16-5-68-41-46-63-49-14	578-486,7-1528,6-678,2-642,9-738,3-776,2-821,5-1528,9	7509,3	
Station 4	55-73-76-8-28-6	578-1042-439,2-1009,2-1714,6-2558,4	7341,4	
Station 5	56-32-11-19-60-7-27-29	578-809,8-709,8-655,9-668,2-1959,7-956,8-982,6	7320,8	
Station 6	2-30-15-44-66-36-17-54	985-992,2-655,6-681,4-712,1-1270,8-1048,8-884	7229,9	
Station 7	25-24-37-48-34-72-53-65	780-835,2-806,9-664,4-1207,9-1315,6-847,4-892,1	7349,5	
Station 8	62-81-78-21-51-9-77-4-13	578-457-1072,6-501,2-656,8-852,6-827,9-1069,9-788,8	6804,8	
Station 9	61-52-1-82-64-79-74	578-572,2-1513-1268,7-760,4-575,2-2283,7	7551,2	
Station 10	12-42-40-26-22-71-45-75-39	464-1200,9-211,1-890,7-879,8-579,3-756,3-920,2-1360,7	7263	
Station 11	70-33-35-38-3	887-825,9-1112,2-1242,1-1757,6	5824,8	
P83 with Ir=0.90, dr=0.20, ct=7571				
Station 1	25-52-54-27-29-7	780-660,6-732,9-958-1049,4-2335,9	6516,8	13
Station 2	53-16-33-5-35-39	578-538,7-890,5-1702,2-1475,8-1787,5	6972,7	
Station 3	41-63-37-47-12-32-70-79	494-609,1-856,9-983,3-824,2-1215,1-1401,3-1139,3	7523,2	
Station 4	69-48-22-17-77-51-76-31	467-604,3-898-1031,8-1018,8-1052,5-1020,3-1408,1	7500,8	
Station 5	15-64-62-60-75-80-26	577-624,1-692,4-774,9-1016,1-2160,5-1548,2	7393,2	
Station 6	66-19-40-45-58-4-18-36	578-644-277-724-800,1-1201,6-1124,9-1703	7052,6	
Station 7	20-46-1-43-61-44-59	233-562,1-1550,3-1021,5-979,8-1102,3-1240,7	6689,7	
Station 8	34-68-82-13-65-38	1234-742,3-1434,5-957,6-1136,6-1791,3	7296,3	
Station 9	3-10-9-49-81	1838-1455,8-1228,5-1205,7-1249,3	6977,3	
Station 10	78-57-24-6-14	1164-729,7-1039,8-2808,8-1786,4	7528,7	
Station 11	23-21-8-42-83	767-505,3-1095,4-1426,8-3484,2	7278,7	
Station 12	50-67-73-28-30-72	578-624,2-1134,3-1903,7-1450,1-1855,3	7545,6	
Station 13	71-11-56-74-55-2	396-701,3-674,8-2346,9-1097,6-1544,7	6761,3	
P83 with Ir=0.80, dr=0.10, ct=7571				
Station 1	19-11-51-15-22-76-82-34-62-38-71	600-608-491-478-634-358-864-838-525-853-472	6721	9
Station 2	72-80-14-75-45-58-33-66	1298-1784-1012-751-642-633-770-648	7538	
Station 3	26-5-12-17-83-77	912-1433-490-745-2411-730	6721	
Station 4	40-64-28-6-7-10-54	83-469-1361-1966-1560-1007-653	7099	
Station 5	57-41-4-27-43-68-79-9-36-52-55	578-441-755-592-613-492-338-601-841-526-534	6311	
Station 6	21-49-29-1-59-35-46-53-25	408-500-565-1165-501-818-520-525-632	5634	
Station 7	47-78-23-67-60-30-74-39-65	822-997-666-529-524-762-1589-972-623	7484	
Station 8	31-2-32-3-73-69-18-63-37-61	713-845-700-1321-868-512-621-582-694-602	7458	
Station 9	48-13-42-50-16-20-70-8-44-24-81-56	578-446-976-498-436-296-647-731-512-649-474-540	6783	
Optimal Station Number				
P83 with Ir=0.7, dr=0.20, ct=7571				
Station 1	2-7-57-14-52-37	985-1942-817-1204-934-1105	6987	11
Station 2	56-23-33-70-63-1-80-46	578-706-762-830-687-1340-1646-966	7515	
Station 3	77-79-31-17-59-69-19-36-51-75	701-341-647-779-638-611-718-1102-831-971	7339	
Station 4	68-61-73-28-3-62-50	578-555-931-1469-1515-892-944	6884	
Station 5	65-43-9-55-42-4-16-24-21	578-726-740-631-1086-969-764-997-841	7332	
Station 6	53-82-67-72-32-29-71-8	578-1132-646-1132-917-896-778-1155	7234	
Station 7	49-48-41-15-10-74-12-27	584-556-507-580-1010-1792-785-978	6792	
Station 8	18-39-5-47-25-6	667-1154-1449-945-967-2200	7382	
Station 9	83-34-76-58	3691-1577-990-1171	7429	
Station 10	20-45-44-11-60-81-54-22-78-66-30	233-512-510-609-566-526-625-800-1006-789-1035	7211	
Station 11	35-26-13-40-64-38	1143-912-640-398-713-1130	4936	
P83 with Ir=0.7, dr=0.10, ct=7571				
Station 1	13-7052-44-37-32-49-68-36-46-1-22-35-2-20-19-40-62-8	500-656-394-359-430-428-317-304-519-297-610-370-444-399-208-294-172-283-383	7367	6
Station 2	12-18-16-23-75-78-66-38-77-33-63-51-58-59-41-56-14-26	464-499-329-439-410-549-311-533-341-373-292-288-284-281-257-275-402-350	6677	
Station 3	55-47-53-34-50-73-4-64-60-31-45-21-29-67-65-15-6	578-616-397-683-352-543-474-324-315-349-308-251-330-291-288-286-820	7205	
Station 4	5-71-28-48-27-39-74-30-10	1700-396-1190-444-490-690-1116-552-617	7195	
Station 5	54-76-7-25-79-9-80-69-57-83-17-42-43	578-290-1318-490-242-432-896-306-334-1280-436-543-403	7548	
Station 6	3-82-81-11-24-72-61	1838-1039-419-505-538-690-397	5426	
P83 with Ir=0.7, dr=0.20, ct=7571				
Station 1	60-63-42-28-10-12-26-56-47-4-51	578-486-853-1111-811-491-654-541-623-674-566	7388	7
Station 2	81-77-5-53-50-73-11-75-80-33-82	450-554-1081-488-478-681-532-555-990-609-753	7171	
Station 3	49-20-67-24-14-61-79-31-41-83-74-44-48	584-245-423-540-652-425-316-464-395-1378-1057-522-529	7530	
Station 4	68-30-19-13-69-35-76-57-22-8-59-21-2-66	578-784-496-427-404-669-378-455-545-609-480-439-599-502	7365	
Station 5	52-39-43-62-70-9-32-64-78-17-37-46	578-998-630-500-624-581-597-508-701-620-600-548	7485	
Station 6	18-27-71-15-25-54-3-6-36-72-23	667-617-371-445-525-439-901-1262-747-763-617	7354	
Station 7		594-488-398-1239-490-922-516-348-727-728-584	7034	

Öğrenme oranının 0.90 olması ile amaç fonksiyonunda değişim olmazken, öğrenme oranı 0.80 olduğunda yaklaşık %27, 0.70 olduğunda ise yaklaşık %45 oranında azalma gerçekleşmektedir ve öğrenme etkisi bozulma etkisine göre daha baskın etki göstermiştir. Dolayısıyla, öğrenme oranı azaldıkça öğrenme etkisi bozulma etkisine daha baskın

geldiği için öğrenme etkisi azaldıkça işlerin işlem süreleri de azalmaktadır ve herhangi bir istasyona atanan iş sayısı artış gösterebilmektedir. Pozisyon tabanlı öğrenme ve bozulma etkileri düşünülmediği durumda bir istasyona maksimum 9 adet iş atanabilmiştir. Bozulma oranı 0.10 iken, öğrenme oranı 0.80 olduğunda herhangi bir istasyona maksimum 12 iş atanmışken, öğrenme oranı 0.70 olduğunda 18 iş bir istasyona atanmıştır. İşlerin işlem sürelerindeki azalma sayesinde, istasyonlara atanan iş sayısı arttığı için bazı istasyonlar boşa çıkmaktadır.



Şekil 14. Eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi için açılan istasyon sayılarının değişimi

Pozisyon tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD probleminin, öğrenme ve bozulma oranlarının farklı kombinasyonları için hesaplanan amaç fonksiyonu değerleri Şekil 14’de sunulmuştur.

Literatürdeki 15 farklı DHD problemi için pozisyon tabanlı öğrenme ve bozulma etkileri altında elde edilen sonuçlar

Literatürde bilinen 15 DHD problemi (Scholl,1995) eş zamanlı olarak pozisyon tabanlı öğrenme etkisi ve bozulma etkisi altında ele alınmıştır. 0.90-0.80-0.70 olmak üzere üç farklı öğrenme oranı ile 0.10 ve 0.20 olmak üzere iki farklı bozulma oranı kullanılmıştır. Öğrenme ve bozulma oranlarının altı farklı kombinasyonu için DHD problemleri eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri dikkate alınarak çözülmüştür. Bozulma, işin işleme başlama zamanının çeşitli sebeplerden gecikmesi veya ertelenmesinden kaynaklanmaktadır. Bozulma gerçek hayatta her alanda var olan bir

olgudur. Öğrenme etkisi ise işin tekrarlanarak öğrenilmesi ve daha kısa sürede yapılmasıdır. Pozisyon tabanlı öğrenme etkisi, öğrenmenin işin istasyonda atandığı pozisyonunun bir fonksiyonu olarak hesaplanmaktadır. Literatürdeki DHD problemleri için pozisyon tabanlı öğrenme ve bozulma etkileri eş zamanlı dikkate alınarak elde edilen amaç fonksiyonu değerleri Tablo 28’de sunulmuştur.

Tablo 28. Farklı DHD problemleri için pozisyon tabanlı öğrenme ve bozulma etkileri altında açılan istasyon sayıları

Problemler	n	çevrim süresi	*Optimal sonuç	öo=0.00 bo=0.00	öo=0.90 bo=0.10	öo=0.90 bo=0.20	öo=0.80 bo=0.10	öo=0.80 bo=0.20	öo=0.70 bo=0.10	öo=0.70 bo=0.20
Gungor and Gupta	8	40	4	4	4	4	4	4	3	4
Mansoor	11	48	4	4	4	4	4	4	3	3
Mitchell	21	15	8	8	8	8	6	7	5	6
Gupta et al.	25	18	9	9	9	10	8	8	6	7
Heskiaoff	28	138	8	8	8	8	6	7	4	5
Buxey	29	30	12	12	12	12	10	11	8	9
Lutz-1	32	1414	11	11	11	12	9	10	7	8
Kilbrid	45	56	10	10	11	12	9	10	6	7
Hahn	53	2806	6	6	6	7	4	5	3	4
Tonge	70	364	10	10	11	12	8	10	5	7
Wee-Mag	75	49	32	32	33	36	31	31	28	29
Arcus-1	83	7571	11	11	11	13	9	11	6	7
Lutz-3	89	79	22	22	22	25	18	21	14	16
Mukherjee	94	222	20	20	20	23	16	19	12	14
Arcus-2	111	7571	27	28	28	31	24	26	17	19

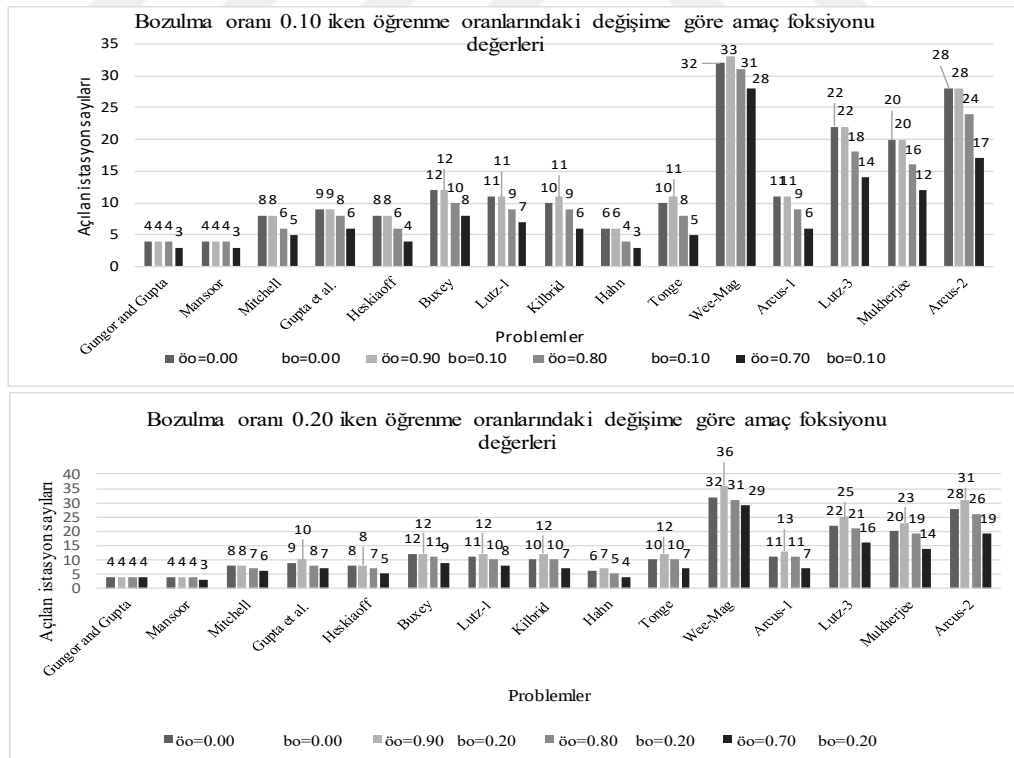
öo:öğrenme oranı, bo:bozulma oranı

Tablo 28’de de görüldüğü gibi öğrenme oranı sabit tutulduğunda, bozulma oranı arttıkça amaç fonksiyonu değeri çoğu DHD probleminde artmaktadır. 32 parçalı DHD probleminde öğrenme oranı 0.90 iken, bozulma oranı 0.10 olduğunda açılan istasyon sayısı 11’dir. Ancak bozulma oranı 0.20 olduğunda açılan istasyon sayısı 12 olmaktadır. Öğrenme oranı 0.80 iken, bozulma oranı sırasıyla; 0.10 ve 0.20 iken açılan istasyon sayıları sırasıyla; 9-10 olarak elde edilmiştir. Öğrenme oranı 0.70 iken, bozulma oranı sırasıyla; 0.10 ve 0.20 iken açılan istasyon sayıları sırasıyla; 7 ve 8 olarak elde edilmiştir. Bozulma oranı arttıkça işlerin işlem süreleri arttığı için, bu etki amaç fonksiyonu değerinde de artışa sebep olmaktadır.

Bir başka örnek olarak; 75 parçalı DHD probleminde bozulma oranları sırasıyla 0.10 ve 0.20 iken, öğrenme oranı 0.90 olduğunda açılan istasyon sayıları 33 ve 36, öğrenme oranı 0.80 olduğunda açılan istasyon sayıları eşit olup 31 ve öğrenme oranı 0.70 olduğunda açılan istasyon sayıları 28 ve 29 olarak elde edilmiştir. Öğrenme oranları sabit iken, tüm

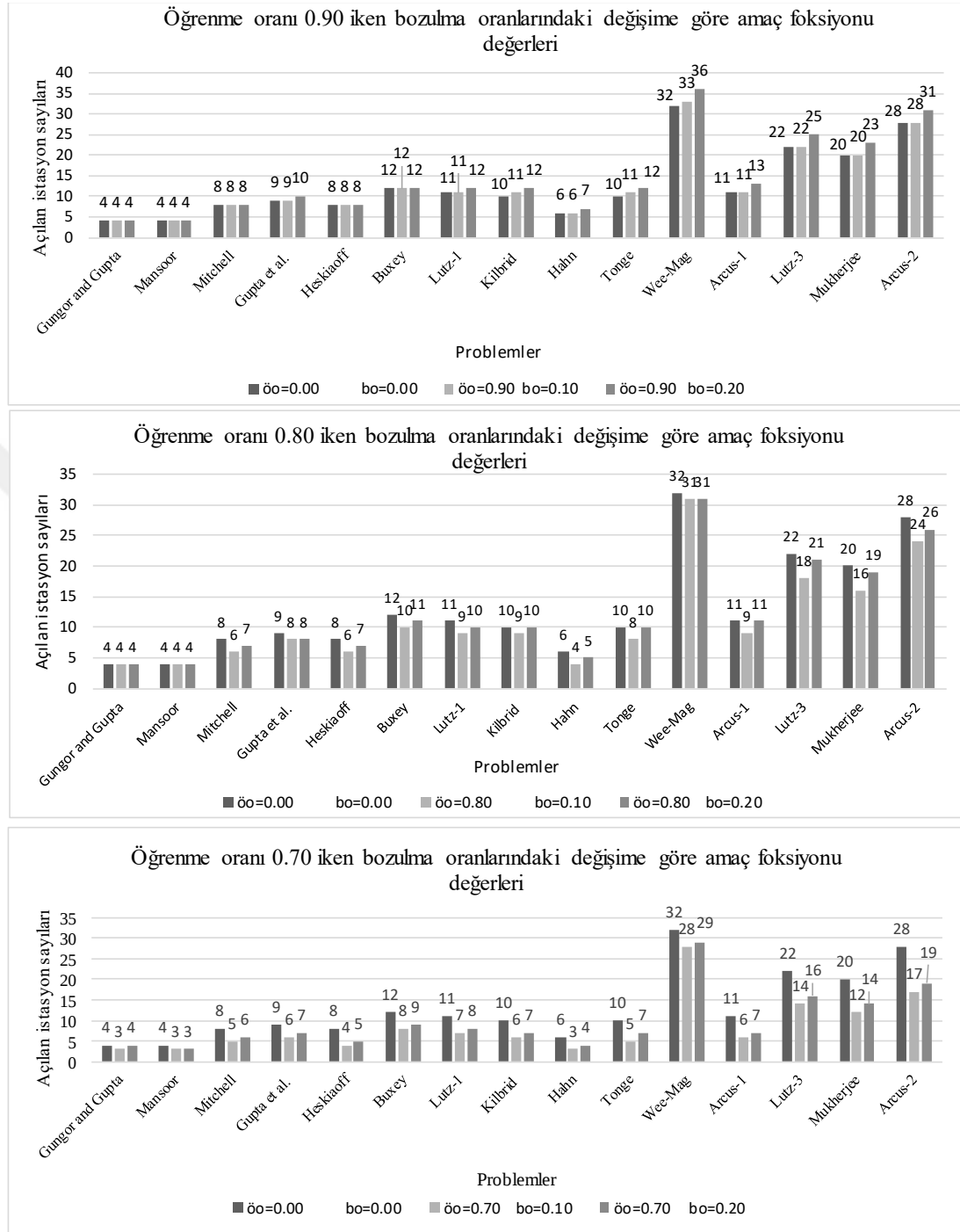
DHD problemlerinde bozulma oranı arttıkça amaç fonksiyonu değeri sabit kalmış veya çoğunlukla artmıştır.

Bozulma oranını sabit kabul edip, öğrenme oranları değerlendirildiği zaman ise öğrenme oranı azaldıkça amaç fonksiyonu değeri de azalmaktadır. Öğrenme oranının azalması, işin daha fazla öğrenildiğini göstermektedir. Bozulma oranı sabit olup 0.10 olduğu varsayılırsa, 111 parçalı DHD problemi için öğrenme oranı sırasıyla 0.90-0.80-0.70 olduğunda amaç fonksiyonu değerleri sırasıyla; 28-24-17 olarak hesaplanmıştır. 111 parçalı DHD probleminde bozulma etkisi olduğu için öğrenme oranı 0.90 iken açılan istasyon sayısı değeri değişmemiştir. Bu durumda bozulma oranı 0.10 ve öğrenme oranı 0.90 iken, pozisyon tabanlı öğrenme etkisi ile bozulma etkisinin DHD problemi üzerindeki etkileri başa baştır. Ne bozulmanın artırıcı etkisi ile amaç fonksiyonunda artış görülmüş, ne de öğrenme etkisinin azaltıcı etkisi ile azalma görülmüştür. Ancak, öğrenme oranı 0.80 olduğunda, amaç fonksiyonu değerinde %14 azalma olurken, öğrenme oranı 0.70 olduğunda, %39 oranında azalma görülmüştür.



öo:öğrenme oranı, bo:bozulma oranı

Şekil 15. Bozulma oranları sabit iken öğrenme oranlarındaki değişime göre amaç fonksiyonu değerleri



öo:öğrenme oranı, bo:bozulma oranı

Şekil 16. Öğrenme oranları sabit iken bozulma oranlarındaki değişime göre amaç fonksiyonu değerleri

Aynı DHD problemi için, bozulma oranı 0.20 olduğunda, öğrenme oranı 0.90 iken açılan istasyon sayısı 31'e çıkmıştır. Bu durumda, amaç fonksiyonu değeri DHD probleminin herhangi bir etki dikkate alınmadığı durumdaki amaç fonksiyonu değerine göre %14

artmıştır. Bu artış, öğrenmenin 0.90 oranında var olduğu düşünüldüğünde gerçekleşmiştir. Bozulma oranı 0.20 iken, öğrenme oranları sırasıyla 0.80 ve 0.70 olduğunda açılan istasyon sayıları sırasıyla; 26 ve 19 olarak bulunmuştur. Bu durumlarda, bozulma etkisine rağmen öğrenme etkisi baskındır ve herhangi bir etkinin düşünülmediği duruma göre amaç fonksiyonu değerinde azalmalar görülmektedir. Diğer tüm DHD problemlerinde de benzer durumlar söz konusudur. Tablo 28’de görüldüğü gibi pozisyon tabanlı öğrenme ve bozulma etkilerinin başa baş olduğu durumlar varken, bozulma etkisinin pozisyon tabanlı öğrenme etkisine baskın olduğu durumlar da vardır. Ancak öğrenme oranı azaldıkça, pozisyon tabanlı öğrenme etkisi bozulma etkisinden daha baskındır ve amaç fonksiyonu değerlerinde azalmalar görülmektedir.

Şekil 15’de bozulma oranları sabit tutulduğunda, öğrenme oranının farklı değerleri ile hesaplanan amaç fonksiyonu değerleri sunulmuştur. Öğrenme oranı düştükçe, açılan istasyon sayısında değişiklik görülebilmektedir. Şekil 16’da ise, öğrenme oranları sabit tutulduğunda, bozulma oranının farklı değerleri ile hesaplanan amaç fonksiyonu değerleri verilmiştir. Öğrenme oranı sabit iken, bozulma oranı değeri arttıkça amaç fonksiyonu değerinin de arttığı görülebilmektedir.

Eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri altında 15 farklı DHD problemi için GA, HGA, PSO VE PSO-GA yaklaşımları ile elde edilen sonuçları ve bu sonuçları karşılaştırılması

Tüm DHD problemleri pozisyon tabanlı öğrenme ve bozulma etkileri altında GA, HGA, PSO ve PSO-GA yaklaşımlarıyla çözdürülmüş ve elde edilen sonuçlar karşılaştırılmıştır.

Tablo 29’da öğrenme etkisi ve bozulma etkisinin eş zamanlı olarak etki ettiği DHD problemleri için 4 yaklaşımın elde ettiği sonuçlar sunulmuştur.

3.4. Eşzamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında DHD problemi

Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında DHD problemleri için açılan istasyon sayıları hesaplanmıştır. İşler istasyonlara atanırken, işler arasındaki öncelik ilişkileri ve istasyonların çevrim süresi kısıtları dikkate alınmıştır. Öğrenme, işlerin tekrarlandıkça öğrenilmeleri ve daha kısa sürede

Tablo 29. GA, HGA, PSO ve PSO-GA yaklaşımlarının çözüm sonuçları (PTÖE ve bozulma etkileri altında)

Set	Çevrim süre	Öğrenme oranı	Bozulma oranı	f(GA)	Min–Max	f(HGA)	Min–Max	f(PSO)	Min–Max	f(PSO-GA)	Min–Max
8-part	40	0.9	0.1	4	4-5	4	4-5	4	4-5	4	4-5
		0.9	0.2	4	4-5	4	4-5	4	4-5	4	4-5
		0.8	0.1	4	4	4	4	4	4	4	4
		0.8	0.2	4	4-5	4	4-5	4	4-5	4	4-5
		0.7	0.1	3	3-4	3	3-4	3	3-4	3	3-4
		0.7	0.2	4	4	4	4	4	4	4	4
11-part	48	0.9	0.1	4	4-5	4	4-5	4	4-5	4	4-5
		0.9	0.2	4	4-5	4	4-5	4	4-5	4	4-5
		0.8	0.1	4	4-5	4	4	4	4-5	3	3-4
		0.8	0.2	4	4	4	4	4	4	4	4
		0.7	0.1	3	3	3	3	3	3	3	3
		0.7	0.2	3	3-4	3	3-4	3	3-4	3	3-4
21-part	15	0.9	0.1	8	8-9	8	8	8	8-9	8	8
		0.9	0.2	8	8-9	8	8-9	8	8-9	8	8-9
		0.8	0.1	7	7-8	7	7	7	7-8	6	6-7
		0.8	0.2	7	7-8	7	7-8	7	7-8	7	7-8
		0.7	0.1	5	5-6	5	5-6	5	5-6	5	5-6
		0.7	0.2	6	6-7	6	6-7	6	6-7	6	6-7
25-part	18	0.9	0.1	9	9-10	9	9-10	9	9-11	9	9-11
		0.9	0.2	10	10-11	10	10-11	10	10-11	10	10-11
		0.8	0.1	8	8-9	8	8-9	8	8-9	8	8-9
		0.8	0.2	8	8-10	8	8-10	9	9-10	8	8-10
		0.7	0.1	7	7-8	7	7-8	7	7-8	7	7-8
		0.7	0.2	7	7-8	7	7-8	7	7-8	7	7-8
28-part	138	0.9	0.1	8	8-9	8	8-9	8	8-9	8	8-9
		0.9	0.2	8	8-10	8	8-9	8	8-10	8	8-9
		0.8	0.1	6	6-9	6	6-8	7	7-8	6	6-8
		0.8	0.2	7	7-8	7	7-8	7	7-8	7	7-8
		0.7	0.1	5	5-6	5	5-6	5	5-6	5	5-6
		0.7	0.2	5	5-7	5	5-7	5	5-7	5	5-7
29-part	30	0.9	0.1	12	12-13	12	12-13	12	12-13	12	12-13
		0.9	0.2	13	13-14	13	13-14	13	13-14	13	13-14
		0.8	0.1	11	11-12	10	10-11	11	11-12	10	10-11
		0.8	0.2	11	11-12	11	11-12	11	11-12	11	11-12
		0.7	0.1	9	9-10	8	8-10	9	9-10	8	8-10
		0.7	0.2	9	9-11	9	9-10	9	9-11	9	9-10
32-part	1414	0.9	0.1	11	11-12	11	11-12	11	11-12	11	11
		0.9	0.2	12	12-13	12	12-13	12	12-13	12	12
		0.8	0.1	9	9-10	9	9-10	10	10	9	9-10
		0.8	0.2	10	10-11	10	10-11	11	11	10	10-11
		0.7	0.1	8	8-9	8	8	8	8-9	7	7-8
		0.7	0.2	9	9-10	8	8-9	9	9-10	8	8-9
45-part	56	0.9	0.1	11	11-12	11	11	11	11-12	11	11
		0.9	0.2	12	12-13	12	12-13	12	12-13	12	12-13
		0.8	0.1	9	9-10	9	9-10	9	9-10	9	9
		0.8	0.2	10	10-11	10	10-11	10	10-11	10	10-11
		0.7	0.1	6	6-8	6	6-7	7	7-8	6	6-7
		0.7	0.2	8	8-9	7	7-8	8	8-9	7	7-8
53-part	2806	0.9	0.1	6	6-7	6	6-7	6	6-7	6	6-7
		0.9	0.2	7	7-8	7	7-8	7	7-8	7	7-8
		0.8	0.1	5	5-6	4	4-5	5	5-6	4	4-5
		0.8	0.2	5	5-7	5	5-6	5	5-8	5	5-6
		0.7	0.1	4	4-5	3	3-5	4	4-5	3	3-5
		0.7	0.2	4	4-6	4	4-5	4	4-6	4	4-5
70-part	364	0.9	0.1	11	11	11	11	11	11	11	11
		0.9	0.2	13	13	12	12-13	13	13-14	12	12-13
		0.8	0.1	8	8-9	8	8-9	8	8-9	8	8-9
		0.8	0.2	10	10-11	10	10-11	10	10-11	9	9-11
		0.7	0.1	6	6-7	5	5-7	6	6-7	5	5-7
		0.7	0.2	7	7-8	7	7-8	7	7-8	7	7-8
75-part	49	0.9	0.1	33	33-34	33	33-34	33	33-34	33	33-34
		0.9	0.2	37	37-39	37	37-39	37	37-39	37	37-39
		0.8	0.1	31	31-32	31	31	31	31-32	31	31
		0.8	0.2	32	32-33	32	32-33	32	32-33	32	32-33
		0.7	0.1	28	28-29	28	28-29	28	28-29	28	28-29
		0.7	0.2	29	29-30	29	29-30	29	29-30	29	29-30
83-part	7571	0.9	0.1	11	11-12	11	11-12	11	11-12	11	11-12
		0.9	0.2	14	14	14	14	14	14	14	14
		0.8	0.1	9	9	9	9	9	9	8	8-9
		0.8	0.2	11	11	10	10-11	11	11-12	10	10-11
		0.7	0.1	6	6-7	6	6-7	6	6-7	6	6-7
		0.7	0.2	8	8-9	7	7-8	8	8-9	7	7-8
89-part	79	0.9	0.1	23	23-24	22	22-23	23	23-24	22	22-23
		0.9	0.2	25	25-27	25	25-26	25	25-27	25	25-26
		0.8	0.1	18	18-19	18	18-19	19	19-20	18	18-19
		0.8	0.2	21	21-22	21	21-22	21	21-22	21	21-22
		0.7	0.1	14	14-15	13	13-15	14	14-15	14	14-15
		0.7	0.2	16	16-17	16	16-17	16	16-17	16	16-17
94-part	222	0.9	0.1	21	21-22	21	21-22	21	21-22	21	21-22
		0.9	0.2	23	23-25	23	23-25	23	23-25	23	23-25
		0.8	0.1	17	17-18	16	16-18	17	17-18	16	16-18
		0.8	0.2	20	20-21	19	19-20	20	20-21	19	19-20
		0.7	0.1	12	12-13	12	12-13	12	12-13	12	12-13
		0.7	0.2	15	15-16	14	14-16	15	15-16	14	14-16
111-part	5785	0.9	0.1	29	29-31	29	29-30	29	29-31	29	29-30
		0.9	0.2	32	32-34	31	31-33	32	32-34	31	31-33
		0.8	0.1	24	24-27	23	23-25	24	24-27	23	23-25
		0.8	0.2	27	27-30	26	26-27	27	27-30	26	26-27
		0.7	0.1	17	17-20	16	16-18	17	17-20	16	16-18
		0.7	0.2	20	20-22	19	19-21	20	20-22	19	19-21

yapılmalarıdır. Dolayısıyla öğrenme etkisi ile işler, gerçekte bilinen işlem sürelerinden daha kısa sürede yapılabilmektedir. Logaritmik işlem süreleri toplamı tabanlı öğrenme de, işlerin öğrenmeden kaynaklı işlem süreleri atandıkları istasyonda kendilerinden önceki pozisyonlara atanmış işlerin işlem sürelerinin logaritmik toplamının bir fonksiyonu ile hesaplanmaktadır. İş ne kadar ileri pozisyona atanırsa, kendisinden önceki pozisyonlara atanmış işlerdeki öğrenmelerden o kadar etkilenir ve daha kısa sürede yapılır. Bozulma ise, işin işleme başlama zamanını erteleyen her türlü olgudur. İşlerin işlem süreleri öğrenme ile azalırken, bozulma ile artmaktadır. Bu bölümde, öğrenme etkisi logaritmik işlem süreleri toplamına göre hesaplanmaktadır. Dolayısıyla, DHD problemlerinde eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri ile amaç fonksiyonundaki ve işlerin işlem sürelerindeki değişiklikler incelenmiştir.

Bu bölümde, öğrenme oranı değerleri 0.90, 0.80 ve 0.70 olarak alınırken, bozulma oranları 0.10 ve 0.20 olarak kullanılmıştır. Öğrenme ve bozulma oranlarının farklı kombinasyonları için açılan istasyon sayısı minimizasyonunu amaçlayan DHD problemleri için eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında elde edilmiştir.

Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 8 parçalı DHD problemi

8 parçalı DHD problemi DHD literatürünün ilk örneğidir. Problemin bileşenlerinin görev tanımları, işlem süreleri ve öncelik ilişkileri Tablo 3’de sunulmuştur. Bu DHD problemi, eş zamanlı logaritmik işlem süreleri toplamı ve bozulma etkileri altında çözümü Lingo 11 paket programı ile elde edilmiştir ve böylece modelin doğruluğu kanıtlanmıştır. Problemin çevrim süresi 40’tır.

Logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 8 parçalı DHD problemi için elde edilen sonuçlar Tablo 30’da yer almaktadır. Bozulma oranı 0.10 iken öğrenme oranları hem 0.80 hem de 0.70 için minimum istasyon sayısı 3 olarak elde edilmiştir. Öğrenme oranı 0.70 iken her iki bozulma oranı için de açılan istasyon sayısı 3 olarak bulunmuştur.

Tablo 30. Logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 8 parçalı DHD probleminin optimum sonuçları

öğrenme oranı(öo) bozulma oranı(bo)	optimal sıra	iş istasyonu sayısı
öo=0 , bo=0	1-2-3, 5-6, 7-4, 8	4
öo=0.90 , bo=0.10	2-4-6, 8, 1-7, 3-5	4
öo=0.90 , bo=0.20	1-5, 8, 2-4-3, 7-6	4
öo=0.80 , bo=0.10	3-8, 1-6-4, 2-5-7	3
öo=0.80 , bo=0.20	2-5-6, 1-8, 3-7-4	3
öo=0.70 , bo=0.10	1-3-8, 7-5-6, 4-2	3
öo=0.70 , bo=0.20	2-8-4, 5-3, 7-6-1	3

öo:öğrenme oranı, bo:bozulma oranı

Öğrenme oranı 0.70 ve bozulma oranı 0.20 iken bulunan optimum sonuçta, istasyona atanan işler, işlerin işlem sürelerinin hesaplanması ile ilgili detaylar Tablo 31’de yer almaktadır. İstasyonlarda işler, çevrim süresini aşmadan ve öncelik ilişkilerini sağlayacak şekilde atanmıştır. Buna göre 1 nolu İstasyona 2-8-4 nolu işler, 2 nolu İstasyona 5-3 nolu işler ve 3 nolu İstasyona 7-6-1 nolu işler sırasıyla atanmıştır. İstasyonlarda işlerin tamamlanma süreleri ise sırasıyla; 39,9-31-37,57 olarak elde edilmiştir.

Tablo 31. Öğrenme oranı 0.70 ve bozulma oranı 0.20 iken işlerin işlem süreleri

İşlerin atandığı istasyon ve sırası	İşin işle süresinin hesaplanması	İşin tamamlanma zamanı
(2,1,1)	10	10
(8,1,2)	$(36+0.20*10)*(1+\ln(10))^{-0,514}=20,56$	$10+20,56=30,56$
(4,1,3)	$(18+0.20*30,56)*(1+\ln(10)+\ln(20,56))^{-0,514}=9.34$	$30,56+9.34=39,9$
(5,2,1)	23	23
(3,2,2)	$(12+0.20*23)*(1+\ln(23))^{-0,514}=8$	$23+8=31$
(7,3,1)	20	20
(6,3,2)	$(16+0.20*20)*(1+\ln(20))^{-0,514}=9.81$	$20+9.81=29.81$
(1,3,3)	$(14+0.20*29.81)*(1+\ln(20)+\ln(9.81))^{-0,514}=7.76$	$29.81+7.76=37.57$

Eşzamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi

25 parçalı hücrese telefon demontajı probleminin görev tanımları, işlerin işlem süreleri ve istasyonların iş tamamlama süreleri Tablo 9’da sunulmuştur. Öğrenme oranları 0.90, 0.80 ve 0.70, bozulma oranları 0.10 ve 0.20 olarak kullanılmıştır. Öğrenme ve bozulma oranlarının farklı kombinasyonları için sonuçlar elde edilmiştir. İşlerin istasyonlara

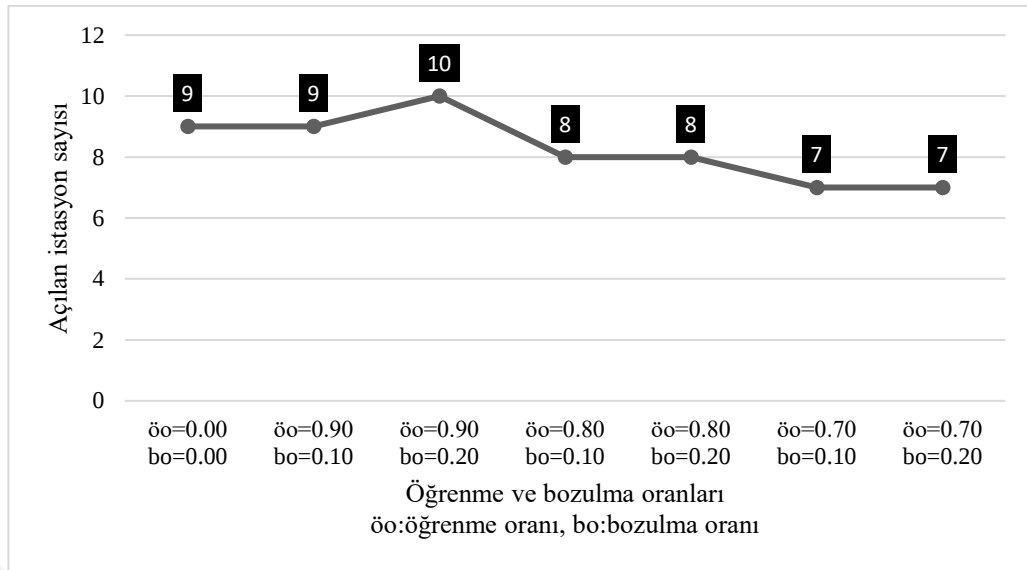
Tablo 32. Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için iş istasyonu sayısı

P25 , çs=18	Logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri			
öo=0.90 bo=0.1	Atanan işler	İşlem süreleri	Toplam süre	İstasyon sayısı
İstasyon 1	9	15	15	9
İstasyon 2	5-21-3-15	10-1,67-3,39-2,74	17,8	
İstasyon 3	19	18	18	
İstasyon 4	12-13-25-4-16	2-2,03-2,1-8,91-2,72	17,76	
İstasyon 5	17-10-23	2-2,03-13,48	17,51	
İstasyon 6	11-7	2-14,03	16,03	
İstasyon 7	24-8	2-14,03	16,03	
İstasyon 8	22-2-20-18-14	5-2,16-4,75-3,29-2,67	17,87	
İstasyon 9	1-6	3-13,67	16,67	
öo=0.90 bo=0.2				10
İstasyon 1	3-21-12-13	3-1,43-2,52-2,82	9,77	
İstasyon 2	17-6	2-14,21	16,21	
İstasyon 3	15-1-25-4	2-3,14-2,58-9,43	17,15	
İstasyon 4	18-23	3-13,94	16,94	
İstasyon 5	2-11-20-14-16	2-2,22-5,09-3,11-3,48	15,9	
İstasyon 6	19	18	18	
İstasyon 7	5-22	10-5,84	15,84	
İstasyon 8	8	15	15	
İstasyon 9	24-9	2-14,21	16,21	
İstasyon 10	10-7	2-14,21	16,21	
öo=0.80 bo=0.10				8
İstasyon 1	5-18-17-1	10-2,72-2,04-2,66	17,42	
İstasyon 2	4-15-16	10-2,04-2,05	14,09	
İstasyon 3	8-13	15-2,29	17,29	
İstasyon 4	25-3-11-23	2-2,7-1,79-10,68	17,17	
İstasyon 5	24-20-7	2-4,39-10,78	17,17	
İstasyon 6	14-19	2-15,36	17,36	
İstasyon 7	10-2-21-12-9	2-1,86-1,06-1,89-11	17,81	
İstasyon 8	22-6	5-11,38	16,38	
öo=0.80 bo=0.20				8
İstasyon 1	17-11-9	2-2,03-11,9	15,93	
İstasyon 2	2-10-14-23	2-10-14-23	17,37	
İstasyon 3	22-6	5-11,75	16,75	
İstasyon 4	21-8	1-15,2	16,2	
İstasyon 5	25-1-7	2-2,87-11,54	16,41	
İstasyon 6	12-18-5-20	2-2,87-7,93-4,55	17,35	
İstasyon 7	16-19	2-15,53	17,53	
İstasyon 8	15-24-3-13-4	2-2,03-2,87-2,27-7,41	16,58	
öo=0.70 bo=0.10				7
İstasyon 1	6	15	15	
İstasyon 2	22-25-12-9-10	5-1,53-1,5-8,38-1,5	17,91	
İstasyon 3	20-16-19	5-1,53-10,54	17,07	
İstasyon 4	24-5-4-14	2-7,78-5,57-1,47	16,82	
İstasyon 5	15-1-13-17-2-23	2-2,44-1,5-1,48-1,47-8,04	16,93	
İstasyon 6	18-11-7-21	3-1,57-9,55-1,07	15,19	
İstasyon 7	8-3	15-2,29	17,29	
öo=0.70 bo=0.20				7
İstasyon 1	24-22-6	2-4,12-9,05	13,75	
İstasyon 2	21-18-12-8-13	1-3,2-1,91-9,53-2,22	18	
İstasyon 3	23	15	15,45	
İstasyon 4	1-5-14-4	3-7,24-1,96-5,58	17,81	
İstasyon 5	16-15-10-19	2-1,83-1,8-11,08	16,03	
İstasyon 6	11-3-25-9	2-2,59-1,77-8,92	16,51	
İstasyon 7	2-17-7-20	2-1,83-10,28-3,55	17,98	

atanma süreleri, eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında hesaplanan yeni işlem süreleri ve istasyonların iş tamamlama süreleri Tablo 32’de sunulmuştur. Tablo 32’de paylaşılan sonuçlar PSO-GA ile elde edilen sonuçlardır.

Öğrenme ve bozulma oranı dikkate alınmadığı durumda açılan istasyon sayısı 9’dur. Bu sonuç, PSO-GA ile de elde edilmiştir. Öğrenme oranı 0.90 ve bozulma oranı 0.10 iken amaç fonksiyonu değerinde bir değişiklik meydana gelmemiştir. Yine 9 istasyon açılması gerekli bulunmuştur. Öğrenme oranı 0.90 ve bozulma oranı 0.20 iken ise, eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında açılan istasyon sayısı 10 olarak elde edilmiştir. Yani, öğrenme etkisi, problemin görevleri üzerinde azaltıcı etki yaparken, bozulma etkisi artırıcı etki yapmaktadır. Burada, bozulma etkisinin artırıcı etkisinin, logaritmik işlem süreleri toplamı tabanlı öğrenmenin azaltıcı etkisinden baskın olduğu görülmektedir. 3 nolu istasyona atanan işler sırasıyla; 15-1-25-4 nolu işlerdir. Bu işlerin temel işlem süreleri sırasıyla; 2-3-2-10 iken eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri ile 2-3.14-2.58-9.45 olarak elde edilmiştir. Temel işlem süreleri ile 3 nolu istasyonun iş tamamlama süresi 17 iken, 17.15 olarak değişmiştir. Öğrenme etkisine rağmen bozulma etkisi ile istasyon iş tamamlama süresinde yaklaşık olarak %24.2 artış görülmektedir.

Öğrenme oranları 0.80 ve 0.70 olduğunda ise amaç fonksiyonu değerlerinde azalma Şekil 17’de olduğu gibi görülmektedir. Öğrenme oranı 0.80 iken bozulma oranının her iki değeri için de açılan istasyon sayıları eşit ve 8 bulunmuştur. Öğrenme oranı 0.70 ve bozulma oranı 0.10 iken açılan istasyon sayısı 7’dir. Öğrenme ve bozulma oranı değerleri minimum iken amaç fonksiyonu değeri en düşüktür. İstasyonlara atanan ortalama iş sayısı 3.57’dir. Her hangi bir etki düşünülmediği durumda ise 2.8’dir. İş istasyonlarının ortalama yoğunluğu 1.25 katına çıkmıştır. Öğrenme oranı 0.70 ve bozulma oranı 0.20 iken açılan istasyon sayısı 7 bulunmuştur. Bozulma oranındaki yüzde 10 artış ile amaç fonksiyonu değerinde değişim olmamıştır. Bozulmanın işlem sürelerini artırması ile istasyonlara atanan iş sayılarında azalma olabilmekte ve yeni istasyonlar açılması gerekebilmektedir. Şekil 17’de öğrenme ve bozulma oranlarındaki değişim sonucu amaç fonksiyonu değerlerindeki değişim görülmektedir. Öğrenme oranı azaldıkça, açılan istasyon sayısında azalma görülmüştür.



Şekil 17. Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 25 parçalı DHD problemi için açılan istasyon sayılarının değişimi

Eşzamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi

Bu bölümde, 83 parçalı DHD problemi eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında, 0.90, 0.80, 0.70 öğrenme oranları ve 0.10, 0.20 bozulma oranları ile çözülmüştür. 83 parçalı DHD probleminin işlem süreleri ve işler arasındaki öncelik ilişkileri Ek 1’de sunulmuştur. Bu problemin çevrim süresi 7571’dir. GA, HGA, PSO ve PSO-GA yaklaşımları kullanılarak 83 parçalı DHD problemi için logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında sonuçlar elde edilmiştir. Bu bölümde PSO-GA yaklaşımı ile bulunan iş atama detayları Tablo 33’de görülmektedir. İşlerin istasyonlara atanma sıraları, yeni işlem süreleri ve açılan istasyon sayısı belirlenmiştir.

Öğrenme ve bozulma etkileri dikkate alınmadığında optimum açılan istasyon sayısı 11 değeri PSO-GA ile de bulunmuştur. Öğrenme oranı 0.90 iken bozulma oranı 0.10 olduğunda açılan istasyon sayısı 10’a düşerken, bozulma oranı 0.20 olduğunda 11’e çıkmıştır. Öğrenme etkisi dikkate alındığı halde bozulma oranı 0.20 iken, bozulma etkisi logaritmik işlem süreleri toplamı tabanlı öğrenme etkisine baskın gelmiştir. Bozulma oranının 0.10 olduğu duruma göre amaç fonksiyonunda %10 artış olurken, herhangi bir etkinin dikkate alınmadığı durumla eşit sonuç elde edilmiştir. Öğrenme oranları 0.80 ve 0.70 iken her iki bozulma oranı içinde amaç fonksiyonu değerlerinde azalma

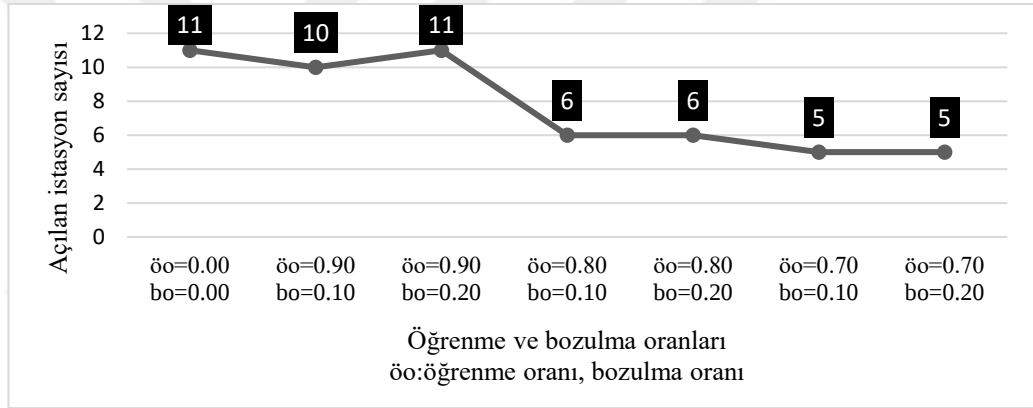
Tablo 33. Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi için iş istasyonu sayısı

P83 , çs=7571	Atanan İşler	İşlem süreleri	Tamamlama sür	Açılan istasyon sayısı
öo=0.00 bo=0.00				
İstasyon 1	28-37-46-48-53-13-60-65-72	1883-792-578-578-578-500-578-578-1298	7363	11
İstasyon 2	5-15-35-27-61-70-1	1700-577-1143-748-578-887-1673	7306	
İstasyon 3	62-57-83-80-50	578-578-3691-2100-578	7525	
İstasyon 4	81-9-39-49-18-38-7	450-793-1310-584-667-1251-2231	7286	
İstasyon 5	31-63-25-43-73-14-42	713-578-780-792-1100-1133-1288	6384	
İstasyon 6	30-40-66-41-23-16-3-24-17	1004-83-578-494-767-483-1838-850-880	6977	
İstasyon 7	34-79-54-8-67-6	1234-286-578-1040-578-2881	6597	
İstasyon 8	10-36-33-20-51-11-4	1250-1266-829-233-578-700-973	5829	
İstasyon 9	74-64-19-47-44-12-76-55-45	2543-578-600-822-578-464-357-578-594	7114	
İstasyon 10	32-78-77-22-29-69-82-52-58	842-1164-701-847-714-467-1300-578-578	7191	
İstasyon 11	26-56-68-75-21-59-2-71	912-578-578-764-408-578-985-396	5199	
öo=0.90 bo=0.10				
İstasyon 1	51-30-17-45-14-62-60-8-13	578-773,69-700,5-541,51-872,23-581,81-601,46-875,22-616,62	7107,40	10
İstasyon 2	32-67-44-21-9-78-73-43-39	842-476,28-480,64-387,26-625,35-857,02-848,45-708,44-1012,86	6723,56	
İstasyon 3	77-31-12-72-35-24-76-26-27-48	701-553,63-399,36-916,78-847,43-704,45-452-772,6-711,9-648,2	7102,49	
İstasyon 4	29-37-53-58-36-34-65-41-55-15	714-628,5-498,4-509,1-927,3-935,9-600,2-575,6-641,8-665,3	7473,31	
İstasyon 5	66-80-71-47-20-25-6-75	578-1484,2-409,2-676,9-349-673,7-1871-773,4	7284,94	
İstasyon 6	4-81-61-42-64-79-5-68-49	973-431,1-517-958-566,3-417,6-1221,4-649,1-676	7430,89	
İstasyon 7	40-33-19-59-18-3-56-22	83-685,2-541,6-538,6-605,3-1296,1-630,2-799,8	6911,92	
İstasyon 8	28-74-70-52-23-63	1883-1852,1-825-657,4-785,6-702,9	7347,21	
İstasyon 9	46-16-10-57-1	578-540,1-1027,3-642,6-1298,3	6854,15	
öo=0.90 bo=0.20				
İstasyon 1	73-22-54-11-48-49-33-60	1100-777,8-634-756,3-743,2-809,8-1019,5-969,5	6810,1	11
İstasyon 2	26-63-80-78-68-3	912-556,3-1600,3-1115-850-1655,1	6688,7	
İstasyon 3	79-72-76-16-61-2-46-71	286-1015,9-415,2-525,7-626,1-919,1-764,9-730,6	5283,5	
İstasyon 4	62-8-25-81-30-6-52	578-853,2-713,1-554,6-935,4-2112,4-980,5	6727,2	
İstasyon 5	47-70-20-21-7-36-1	822-770,7-367,8-506,4-1655,6-1223,6-1559,3	6905,4	
İstasyon 6	29-69-44-27-28-83	714-448,3-544,6-691-1435,5-2610,6	6444	
İstasyon 7	9-31-67-56-51-74-19-82	793-639,4-577,9-619,3-669,2-1878,67-930,2-1400,8	7508,47	
İstasyon 8	12-17-15-35-4-55-23-34	464-721,5-546,7-944,5-914,6-759,5-933,3-1275,3	6559,4	
İstasyon 9	43-38-75-10-58-40-45-39	792-1033,9-751,1-1109,5-791,5-570,2-911,1-1388,7	7348	
İstasyon 10	37-42-24-59-18-65-32-13	792-1061,1-811,7-697,7-807,7-824-1045,7-947,5	6987,4	
İstasyon 11	50-41-57-14-77-53-5-64-66	578-450,01-527,7-917,7-726,3-715,4-1418,8-916,2-997,5	7247,61	
öo=0.80 bo=0.10				
İstasyon 1	53-60-62-83-21-41-11-74-59	578-334,3-291,8-1480,6-236,5-258,5-317,8-862,1-288,3	4647,9	6
İstasyon 2	61-71-76-80-38-69-6-31-48-8-64-75-42-29-67-45-37-14	578-238,6-192,8-866,3-511,2-233,3-986,5-318,7-276,8-402,2-276,6-324,2-455,8-314,3-282,1-286,7-333,3-411,7	7289,1	
İstasyon 3	10-28-20-78-56-58-51-66-81-15	1250-1022,6-192,2-534,8-305,2-296,6-290,7-286,6-247,3-280,8	4706,8	
İstasyon 4	26-25-19-33-34-30-72-9-1-50-44-17-5-79-43-23-68	912-449,3-315,3-381,8-506,7-413,6-495,2-336,9-583,2-277,9-276,9-354,1-539,3-212,9-334,6-329,6-287,3	7026,6	
İstasyon 5	32-63-3-49-7-39-2-57-18-16-47-73-70-82-27-46-40	842-342,7-844,8-298,4-862,8-529,6-416,9-290,9-313,4-261,1-349,9-420,4-366,3-467,1-335,6-296,8-184,6	7423,3	
İstasyon 6	55-77-35-4-13-65-22-24-36-52-12-54	578-399-538,6-431,1-244,2-263,2-342,2-337,5-452,7-258,4-226,9-256,2	4328	
öo=0.80 bo=0.20				
İstasyon 1	21-14-28-47-81-50-42-71-33-60-26-4-30-68	408-648,7-906,2-460,7-325,7-368,7-591,9-335,2-463,8-403,8-500,4-526,6-545,9-455,1	6940,7	6
İstasyon 2	27-62-65-54-25-59-61-40-35-72-31-20-52-36-6-23	748-378,4-347-335,8-403,8-337-340-197,1-505,6-556,9-411,3-296,1-390,2-568,5-968,3-495,9	7279,9	
İstasyon 3	43-48-9-55-3-57-7-10-2-18-37-39	792-382,1-443,3-345,2-785,6-368,6-883,7-602,1-536,2-459,4-501,6-644,3	6744,1	
İstasyon 4	56-8-32-5-41-17-46-12-82-29-75-53-44-58-51	578-607,7-463,8-774,8-340,5-467,6-377,8-349,5-592,3-442,6-464,8-427,2-437,1-447,5-458,4	7229,6	
İstasyon 5	19-79-45-70-76-63-74-83-49-66-34-38-22-16	600-213,2-333,3-435,7-239,9-314,8-937,9-1285,3-413,5-419,5-601,9-618,4-530,8-454,6	7398,8	
İstasyon 6	24-15-78-1-11-73-13-67-64-77-69-80	850-386,4-607,5-776,8-424,7-555,4-374,6-402,8-409,7-450,9-398,6-824,9	6462,3	
öo=0.70 bo=0.10				
İstasyon 1	4-53-49-23-5-58-41-43-60-72-70-14-32-11	973-233,7-186,1-201,9-364,1-135,3-113,7-153,2-115,8-208,7-147,9-173,4-134-114,6	3255,4	5
İstasyon 2	61-29-39-69-26-17-1-6-33-67-57-38-31-65	578-276,6-373,6-130,2-205,6-183,3-300,2-463,2-150,7-111,7-108,1-186,7-249,4-116,5-99,4	3533,2	
İstasyon 3	37-54-13-63-51-75-80-77-35-42-15-83-45-68-24-59-21-2-81-7-44-34-22	792-230,5-160,4-156,7-141,3-163,9-373,6-136,9-193,9-205,1-106,4-484,9-105,8-101,5-128,9-97,6-78,4-134,7-80,4-248,1-91,2-149,2-112,9	4474,3	
İstasyon 4	76-28-16-64-82-48-27-25-36-73-47-8-79-40-18-52-9	357-712,1-155,1-156,4-283,5-131,4-151,2-148,1-210-179,2-136,9-159,4-66,2-41,8-105,7-93,9-114,6	3202,5	
İstasyon 5	50-19-71-10-74-78-20-66-56-62-12-30-55-46	578-235,8-128,6-307,1-527,6-234,9-69,7-119-113,9-109,6-91,3-155,6-100,3-97,9	2869,3	
öo=0.70 bo=0.20				
İstasyon 1	15-36-13-83-6-72-63-61-53-79-29-78-82-35-4-21-5	577-495,2-187-872,9-625,1-311,4-184,4-177,8-172,6-130,4-180,9-230,8-241,9-220,7-199,6-139,6-266,3	5213,6	5
İstasyon 2	39-48-25-40-28-31-75-7-65-20-38-51-23-64-45-1-27-10-80-59	1310-285,2-284,8-99,7-444,1-207,8-205,9-415,7-170-120,4-245,7-159,7-178,8-155,2-154,9-264,8-168,8-216-294,2-151,7	5533,4	
İstasyon 3	60-76-50-55-33-68-69-14	578-169,4-198,7-174,7-210,4-151,4-125,9-223,8	1832,3	
İstasyon 4	77-52-66-12-9-22-74-34-41-8-46-62-81-49-2-30-11-70-17-11-44-54-32	701-254-205-155,7-208,6-204,8-470,7-250,5-137,8-207-143,5-140,8-123,7-137,1-178,8-178,5-145,8-162,9-160,8-114,5-130,525338237373-129,8-152,2	4694	
İstasyon 5	3-58-24-16-26-57-42-67-19-37-47-73-56-18-43	1838-314,4-326,6-209,6-274,9-200,3-303,1-186,1-183,4-203,6-202,6-231,4-167,7-175-185,9	5002,6	

gözlemlenmiştir. Öğrenme oranı 0.80 iken, bozulma oranı 0.10 ve 0.20 olduğunda 6 istasyon açılması gerekli bulunmuştur. Öğrenme oranı 0.90 iken, bozulma oranı arttıkça açılan istasyon sayısı da artmıştır. 83 parçalı DHD problemi için logaritmik işlem süreleri tabanlı öğrenme ve bozulma etkileri altında en iyi sonuç öğrenme oranı 0.70 iken bulunmuştur. Bozulma oranının her iki değeri için de amaç fonksiyonu değeri 5 bulunmuştur. Bozulma oranı ne kadar az olursa işlerin işlem süresindeki artış ta o kadar

az olmaktadır. En düşük öğrenme oranı ve en düşük bozulma oranının kombinasyonu olan 0.70-0.10 öğrenme-bozulma oranı değerleri ile en iyi sonuç elde edilmiştir. Bu oranlarda açılan istasyon sayısı 11'den 5'e düşmüştür ve %54 azalma görülmüştür. Tablo 33'de görüldüğü gibi tüm istasyonlarda çevrim süresi kısıtı sağlanmıştır.

Öğrenme oranı 0.90 ve bozulma oranı 0.20 iken istasyonların ortalama iş yoğunluğu 7.45 iken öğrenme oranı 0.70 ve bozulma 0.10 iken 16.6 olarak hesaplanmıştır. Bu durum, hem bozulma oranının azalması hem de işin daha fazla öğrenilmesinden kaynaklanmaktadır. Bozulma oranının azalması ile işlerin işlem sürelerindeki artış baskılanmıştır. Öğrenmenin artması ile de işlerin işlem sürelerindeki azalma artmıştır. Dolayısıyla, istasyonlara daha fazla iş atanmış ve birçok istasyon boşa çıkmıştır.



Şekil 18. Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında 83 parçalı DHD problemi için açılan istasyon sayılarının değişimi

Logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında DHD probleminin, farklı öğrenme bozulma oranları kombinasyonları ile hesaplanan amaç fonksiyonu değerleri Şekil 18'de görüldüğü gibidir. Öğrenme oranı ve bozulma azaldıkça, açılan istasyon sayısı değerinde iyileşme görülmektedir.

Literatürdeki 15 farklı DHD problemi için logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında elde edilen sonuçlar

Logaritmik işlem süreleri toplamı tabanlı öğrenmenin ve bozulma etkisinin eş zamanlı etkileri altında literatürdeki 15 DHD problemi için minimum iş istasyonu sayısı elde edilmiştir. 0.90-0.80-0.70 olmak üzere üç farklı öğrenme oranı ile 0.10 ve 0.20 olmak üzere iki farklı bozulma oranı kullanılmıştır. Öğrenme ve bozulma oranlarının altı farklı

kombinasyonu için DHD problemleri eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri dikkate alınarak çözülmüştür. Literatürdeki DHD problemleri için pozisyon tabanlı öğrenme ve bozulması etkileri eş zamanlı dikkate alınarak hesaplanan amaç fonksiyonu değerleri Tablo 34’de sunulmuştur. Tablo 34’de de görüldüğü gibi öğrenme oranı sabit tutulduğunda, bozulma oranı arttıkça amaç fonksiyonu değeri çoğu DHD probleminde artmaktadır. Özellikle, öğrenme oranı 0.90 ve bozulma oranı 0.20 iken herhangi bir etki düşünülmediği duruma göre çoğu amaç fonksiyonu değerinde artış gözlenmiştir. 45 parçalı DHD probleminde açılan istasyon sayısı 10’dan 11’e, 89 parçalı DHD probleminde 22’ten 23’e, 94 parçalı DHD probleminde 20’den 21’e çıkmıştır.

Tablo 34. Farklı DHD problemleri için LİSTTÖ ve bozulma etkileri altında açılan istasyon sayıları

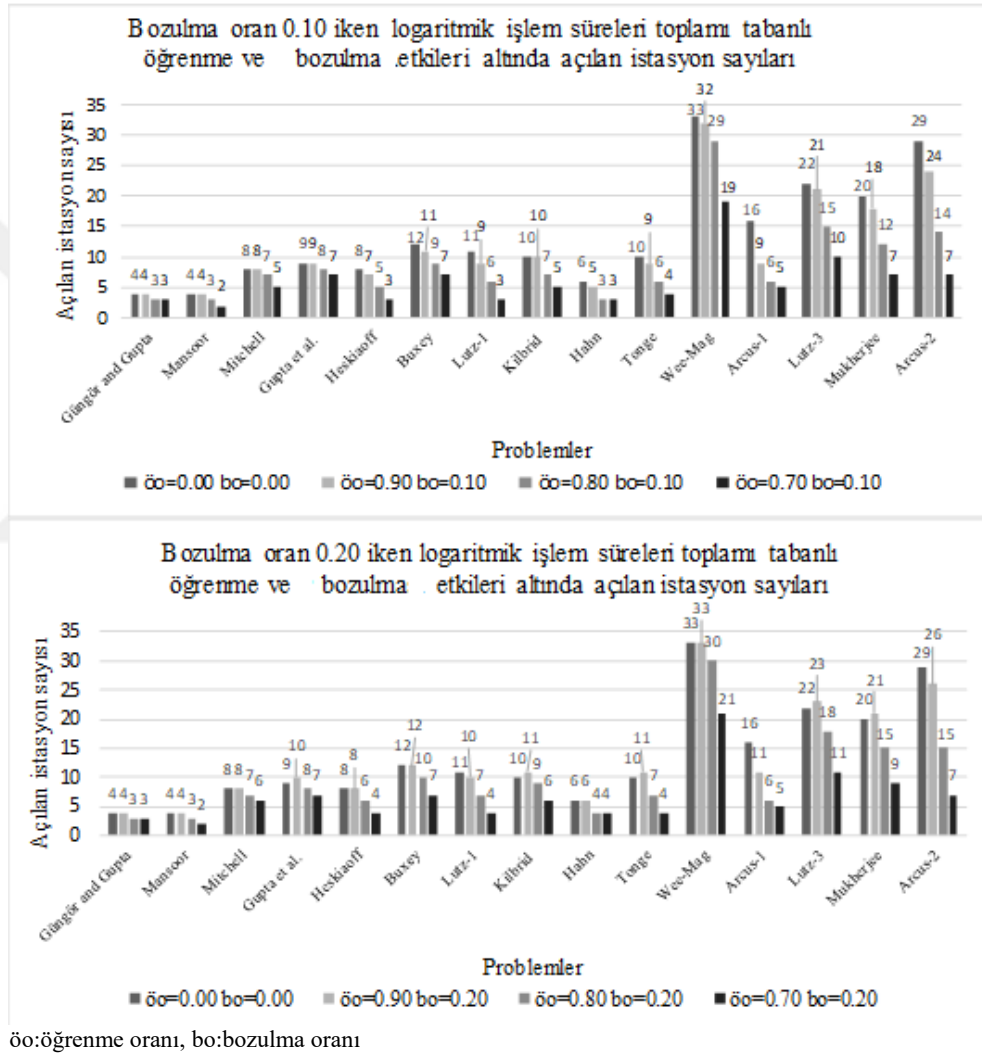
Yazar	n	Çevrim Süresi	*Optimal Sonuç	öo=0	öo=0.9		öo=0.8		öo=0.7	
				bo=0	bo=0.1	bo=0.2	bo=0.1	bo=0.2	bo=0.1	bo=0.2
Güngör and Gupta	8	40	4	4	4	4	3	3	3	3
Mansoor	11	48	4	4	4	4	3	3	2	2
Mitchell	21	15	8	8	8	8	7	7	5	6
Gupta et al.	25	18	9	9	9	10	8	8	7	7
Heskiaoff	28	138	8	8	7	8	5	6	3	4
Buxey	29	30	12	12	11	12	9	10	7	7
Lutz-1	32	1414	11	11	9	10	6	7	3	4
Kilbrid	45	56	10	10	10	11	7	9	5	6
Hahn	53	2806	6	6	5	6	3	4	3	4
Tonge	70	364	10	10	9	11	6	7	4	4
Wee-Mag	75	49	33	33	32	33	29	30	19	21
Arcus-1	83	7571	16	16	9	11	6	6	5	5
Lutz-3	89	79	22	22	21	23	15	18	10	11
Mukherjee	94	222	20	20	18	21	12	15	7	9
Arcus-2	111	5785	29	29	24	26	14	15	7	7

öo:öğrenme oranı, bo:bozulma oranı

Bozulma oranının yüksek olması ve öğrenme oranının düşük olması durumunda öğrenmenin az olması sebebiyle, işlerin işlem sürelerinde artış olmuş ve yeni istasyonların açılması gerekmiştir. Dolayısıyla açılan istasyon sayılarında artış meydana görülmüştür.

70 parçalı DHD problemi için etkiler gözetilmediğinde 10 istasyon açılması gerekmektedir. Öğrenme oranı 0.80 iken bozulma oranları sırasıyla; 0.10 ve 0.20 iken açılan istasyon sayıları sırasıyla; 6 ve 7 olarak elde edilmiştir. Bozulma oranı artınca açılan istasyon sayısı da artmıştır. Öğrenme oranı 0.70 iken bozulma oranları sırasıyla; 0.10 ve 0.20 iken açılan istasyon sayıları eşit olup; 4 bulunmuştur. Öğrenme oranı

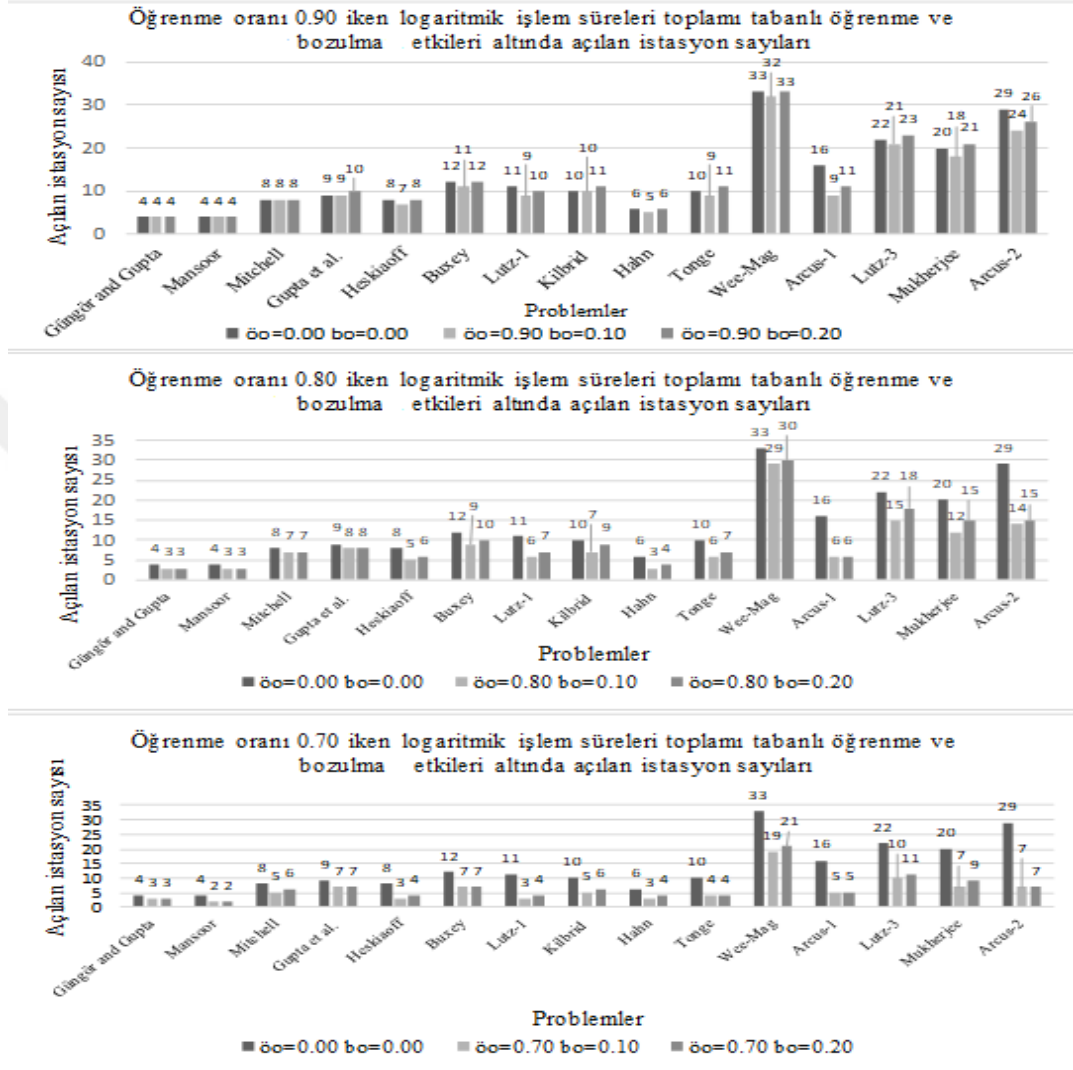
azalınca, bozulma oranlarına daha baskın gelerek açılan istasyon sayısının azalmasını sağlamıştır. Bozulma oranları sabit tutulursa, bozulma oranı 0.10 iken ve öğrenme oranları sırasıyla 0.90-0.80-0.70 olduğunda açılan istasyon sayısı sırasıyla 9-6-4 olarak elde edilmiştir. Bozulma oranı 0.20 iken öğrenme oranları sırasıyla 0.90-0.80-0.70 iken açılan istasyon sayısı sırasıyla 11-7-4 olarak bulunmuştur. Öğrenme oranı azaldıkça amaç fonksiyonu değeri de azalmaktadır.



Şekil 19. Bozulma oranları sabit iken öğrenme oranlarındaki değişime göre amaç fonksiyonu değerleri

Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında DHD problemlerinin, bozulma oranları sabit iken öğrenme oranlarının değişimi ile amaç fonksiyonlarındaki değişim Şekil 19'da görüldüğü gibidir. Öğrenme oranı azaldıkça açılan istasyon sayısı azalmaktadır. Öğrenme oranları sabit iken ise bozulma oranlarının değişimi ile DHD problemlerinin amaç fonksiyonlarındaki değişim Şekil 20'de

görülmektedir. Öğrenme oranı sabit iken, bozulma oranı arttıkça açılan istasyon sayısı genellikle artmaktadır.



öo:öğrenme oranı, bo:bozulma oranı

Şekil 20. Öğrenme oranları sabit iken bozulma oranlarındaki değişime göre amaç fonksiyonu değerleri

Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında farklı DHD problemi için GA, HGA, PSO VE PSO-GA yaklaşımları ile elde edilen sonuçlar ve bu sonuçların karşılaştırılması

Tüm DHD problemleri eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında GA, HGA, PSO ve PSO-GA yaklaşımlarıyla çözdürülmüş ve elde edilen sonuçlar Tablo 35’de sunulmuştur.

Tablo 35. GA, HGA, PSO ve PSO-GA yaklaşımlarının çözüm sonuçları (LİSTTÖE ve bozulma)

Set	Cycle time	opt*	ğrenme oranı	bozulma oranı	f(GA)	Min-Max	f(HGA)	Min-Max	f(PSO)	Min-Max	f(PSO-GA)	Min-Max
6-pant	40	4	0.9	0.1	4	4	4	4	4	4-5	4	4
			0.9	0.2	4	4-5	4	4-5	4	4-5	4	4-5
			0.8	0.1	3	3-4	3	3-4	3	3-4	3	3-4
			0.8	0.2	3	3-4	3	3-4	3	3-4	3	3-4
			0.7	0.1	3	3-4	3	3-4	3	3-5	3	3-4
11-pant	48	4	0.9	0.1	4	4-5	4	4	4	4-5	4	4
			0.9	0.2	4	4-5	4	4-5	4	4-5	4	4-5
			0.8	0.1	3	3-4	3	3	3	3-4	3	3
			0.8	0.2	3	3-4	3	3	3	3-4	3	3
			0.7	0.1	2	2-3	2	2-3	2	2-3	2	2-3
21-pant	15	8	0.9	0.1	8	8	7	7-8	8	8-9	7	7-8
			0.9	0.2	8	8-9	8	8-9	8	8-9	8	8-9
			0.8	0.1	7	7-8	7	7-8	7	7-8	7	7-8
			0.8	0.2	7	7-8	7	7-8	7	7-8	7	7-8
			0.7	0.1	5	5-6	5	5-6	5	5-6	5	5-6
25-pant	18	9	0.9	0.1	9	9-10	9	9-10	9	9-10	9	9-10
			0.9	0.2	10	10-11	10	10-11	10	10-11	10	10-11
			0.8	0.1	8	8-9	8	8-9	8	8-9	8	8-9
			0.8	0.2	8	8-9	8	8-9	8	8-9	8	8-9
			0.7	0.1	7	7-8	7	7-8	7	7-8	7	7-8
28-pant	138	8	0.9	0.1	7	7-8	7	7-8	7	7-8	7	7-8
			0.9	0.2	8	8-9	8	8-9	8	8-9	8	8-9
			0.8	0.1	5	5-6	5	5-6	5	5-6	5	5-6
			0.8	0.2	6	6-7	6	6-7	6	6-7	5	5-7
			0.7	0.1	3	3-4	3	3-4	3	3-4	3	3-4
29-pant	30	12	0.9	0.1	11	11-12	11	11-12	11	11-12	11	11-12
			0.9	0.2	12	12-13	12	12-13	12	12-13	12	12-13
			0.8	0.1	10	10-11	9	9-10	10	10-11	9	9-10
			0.8	0.2	10	10-11	10	10-11	10	10-11	10	10-11
			0.7	0.1	7	7-8	7	7-8	7	7-8	7	7-8
32-pant	1414	11	0.9	0.1	9	9-10	9	9-10	9	9-10	9	9-10
			0.9	0.2	10	10-11	10	10-11	10	10-11	10	10-11
			0.8	0.1	6	6-7	6	6-7	6	6-7	6	6
			0.8	0.2	7	7-8	7	7-8	7	7-8	7	7-8
			0.7	0.1	3	3-4	3	3-4	3	3-4	3	3-4
45-pant	56	10	0.9	0.1	10	10-11	10	10-11	10	10-11	10	10-11
			0.9	0.2	11	11-12	11	11-12	11	11-12	11	11-12
			0.8	0.1	7	7-8	7	7-8	7	7-8	7	7-8
			0.8	0.2	9	9-10	9	9-10	9	9-10	9	9-10
			0.7	0.1	5	5-6	5	5-6	6	6-7	5	5-6
53-pant	2806	6	0.9	0.1	5	5-6	5	5-6	5	5-6	5	5-6
			0.9	0.2	6	6-7	6	6-7	6	6-7	6	6-7
			0.8	0.1	3	3-5	3	3-5	4	4-5	3	3-5
			0.8	0.2	4	4-5	4	4-5	4	4-6	4	4-5
			0.7	0.1	4	4-5	3	3-5	4	4-5	3	3-5
70-pant	364	10	0.9	0.1	9	9-10	9	9-10	9	9-10	9	9-10
			0.9	0.2	11	11-12	11	11-12	11	11-12	11	11-12
			0.8	0.1	6	6-7	6	6-7	6	6-7	6	6
			0.8	0.2	7	7-8	7	7-8	7	7-8	7	7-8
			0.7	0.1	4	4-5	4	4-5	4	4-5	4	4-5
75-pant	49	33	0.9	0.1	32	32-33	31	31-32	32	32-33	31	31-32
			0.9	0.2	33	33-34	33	33-34	33	33-34	33	33-34
			0.8	0.1	29	29-31	29	29-31	29	29-32	29	29-30
			0.8	0.2	30	30	30	30	30	30	30	30
			0.7	0.1	19	19-20	18	18-20	19	19-20	18	18-20
83-pant	7571	11	0.9	0.1	9	9-10	9	9-10	9	9-10	9	9
			0.9	0.2	11	11	11	11	11	11	11	11
			0.8	0.1	6	6-7	6	6-7	6	6-7	6	6-7
			0.8	0.2	6	6-7	6	6-7	6	6-7	6	6-7
			0.7	0.1	5	5-6	5	5-6	5	5-6	5	5-6
89-pant	79	22	0.9	0.1	21	21-22	21	21-22	21	21-22	20	20-22
			0.9	0.2	23	23-24	23	23-24	23	23-24	23	23-24
			0.8	0.1	15	15-16	15	15-16	15	15-16	15	15-16
			0.8	0.2	18	19-20	18	19-20	18	19-21	18	19-21
			0.7	0.1	10	10-13	10	10-12	11	11-13	10	10-12
94-pant	222	20	0.9	0.1	18	18-19	18	18-19	18	18-19	18	18-19
			0.9	0.2	21	21-22	21	21-22	21	21-22	21	21-22
			0.8	0.1	12	12-13	12	12-13	12	12-13	12	12-13
			0.8	0.2	16	16-17	15	15-17	16	16-18	15	15-17
			0.7	0.1	7	7-8	7	7-8	7	7-8	7	7-8
111-pant	5785	29	0.9	0.1	24	24-26	24	24-26	24	24-26	24	24-26
			0.9	0.2	26	26-29	26	26-29	26	26-29	26	26-29
			0.8	0.1	14	14-16	14	14-16	14	14-17	14	14-16
			0.8	0.2	15	15-18	15	15-18	16	16-18	15	15-18
			0.7	0.1	7	7-8	7	7-8	7	7-8	7	7-8
			0.7	0.2	7	7-9	7	7-9	7	7-9	7	7-9

4. BÖLÜM

SONUÇ VE GELECEK ÇALIŞMALAR

Ürünler, kullanım ömürleri dolduğunda, bozulduğunda veya teknolojik gelişmelerle yeni sürümleri çıktığında atık haline gelmektedir. Son yıllarda, atık malzemelerin çok artması, halkların bilinçlenmesi ve hükümetlerin çeşitli çevre yasaları çıkarmasıyla birlikte geri dönüşüme ilgi artmaktadır. Atık malzemeler, geri dönüştürülebilmekte veya yeniden üretilebilmektedir. Geri dönüşüm sisteminin en önemli aşaması ürünlerin bileşenlerine ayrılmasıdır. Yani, demontaj işlemleridir. Demontaj işlemleri, tamamen veya kısmi olabilmektedir. Demontaj işlemleri, demontaj hatları üzerinde yapılmaktadır. Bu hatlar, düz, 2-yönlü, U-şeklinde, paralel veya çoklu paralel olabilmektedir. Bu çalışmada ürünleri tamamen demonte edilmektedir ve demontaj hattı düzdür. Demontaj işlemlerinin etkili ve etkin bir şekilde yapılabilmesi için demontaj hatları dengeli olmalıdır. Bu çalışmada 15 farklı DHD problemi üzerinde çalışılmıştır.

Açılan istasyon sayısı minimizasyonunu amaç edinen DHD problemleri, öğrenme ve bozulma etkileri altında ele alınmıştır. Literatürdeki DHD problemlerinde, işlerin işlem süreleri mevcuttur ve sabittir. Oysaki, gerçek hayatta işler zamanla tekrarlandıkça öğrenilmektedir ve işlerin işleme başlama zamanları çeşitli sebeplerden dolayı gecikebilmektedir. İşlerin, istasyonlarda işleme başlamalarının gecikmesi bozulma olarak bilinmektedir. Öğrenme etkisi bu çalışmada iki şekilde hesaplanmıştır. Birincisi pozisyon tabanlı öğrenme türüdür. İşin işlem süresi istasyonda atanmış olduğu pozisyona bağlıdır ve iş istasyonda ne kadar ileri pozisyona atanırsa, işlem süresi o düzeyde etkilenir ve kısalır. İkinci olarak, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi türü kullanılmıştır. Bu öğrenme türünde ise, işin işlem süresi, aynı istasyonda kendisinden önceki sıralara atanmış işlerin logaritmik işlem süreleri toplamının bir fonksiyonu ile hesaplanmaktadır. Herhangi bir iş, istasyonda ne kadar ileri pozisyona atanırsa,

kendisinden önceki tüm işler yapılırken öğrenme devam ettiği için, o kadar etkilenir ve işin işlem süresinde azalma görülür. Bozulma etkisi ise, işlerin işlem sürelerini artırıcı etkiye sahiptir. Bu çalışmada, DHD problemleri, pozisyon tabanlı öğrenme, logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında ayrı ayrı ele alınmıştır. Ayrıca, öğrenme ve bozulmanın gerçek hayatta eş zamanlı etki ettiği gerçeği de vardır. İşler zamanla öğrenilirken, aynı zamanda işlem sürelerini artırıcı etki edecek bozulmalara da maruz kalmaktadır. Dolayısıyla, DHD problemleri, eş zamanlı pozisyon tabanlı öğrenme ve bozulma etkileri ile eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında da çalışılmıştır.

DHD problemleri çözülürken, istasyonların çevrim süresi ve işler arasındaki öncelik ilişkileri dikkate alınmalıdır. İstasyonlara atanan işlerin tamamlanma zamanları çevrim süresini aşmamalı ve işler istasyonlara atanırken öncelik ilişkilerine uyulmalıdır. Öğrenme ve bozulma etkileri altında çalışılan DHD problemleri için elde edilen sonuçlar aşağıdaki gibi özetlenmiştir:

1. Pozisyon tabanlı öğrenme ve logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında DHD problemlerinde, işlem sürelerinde ve açılan istasyon sayılarında ciddi azalmalar görülmüştür. Öğrenme oranları 0.90, 0.85, 0.80, 0.75 ve 0.70 olarak ele alınmıştır. Öğrenme oranı azaldıkça işlerin işlem süreleri azalmaktadır. İşlerin işlem süreleri azaldıkça istasyonlara atanan iş sayılarında artış meydana gelmekte ve bazı istasyonlar boşa çıkmaktadır. Böylece, açılan istasyon sayısında azalma olmaktadır. Her iki öğrenme türünde de, açılan istasyon sayıları, tüm DHD problemlerinde, öğrenme etkileri altında azalmıştır. Öğrenme oranı düştükçe, açılan istasyon sayısı da düşmüştür.
2. Bozulması etkisi altında DHD problemlerinde, bozulma oranları 0.10, 0.15 ve 0.20 olarak ele alınmıştır. Bozulma oranı arttıkça işlerin işlem süreleri artmaktadır. Dolayısıyla, istasyonların çevrim süresini aşmayacak şekilde alabileceği iş sayısı azalmaktadır. Arta kalan işler için yeni istasyonlar açılması gerekir ve sonuç olarak açılan istasyon sayısı artar. Bozulması etkisi amaç fonksiyonu değerini olumsuz etkileyen bir olgudur.

3. Pozisyon tabanlı öğrenme ve bozulma etkileri altında DHD problemleri için, öğrenme oranları 0.90, 0.80 ve 0.70 ve bozulma oranları 0.10 ve 0.20 olarak ele alınmıştır. Öğrenme etkisi işlerin işlem sürelerini azaltırken bozulma etkisi artırıcı etki etmektedir. DHD problemlerinde, öğrenme oranı 0.90 ve bozulma oranı 0.10 iken açılan istasyon sayılarında değişim olmamıştır. Öğrenme oranı sabit kalıp bozulma oranı 0.20 olduğunda ise açılan istasyon sayılarında artış görülmüştür. Dolayısıyla, bozulma oranı 0.20 ve öğrenme oranı 0.90 iken, bozulmanın öğrenmeye baskın olduğu söylenebilmektedir. Ancak, öğrenme oranı arttıkça, açılan istasyon sayıları azalmıştır. Bozulmanın etkisi olsa da, öğrenme oranı 0.80 veya 0.70 iken, amaç fonksiyonu değerlerinde azalmalar görülmüştür.
4. Eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve bozulma etkileri altında DHD problemlerinde de öğrenme oranları 0.90, 0.80 ve 0.70 ve bozulma oranları 0.10 ve 0.20'dir. Öğrenme oranı 0.90 ve bozulma oranı 0.10 iken, açılan istasyon sayılarının yarısında değişiklik görülmezken diğer yarısında artış meydana gelmiştir. Aynı durumda, pozisyon tabanlı öğrenme etkisi ile hesaplandığında amaç fonksiyonunda azalış olmazken, logaritmik işlem süreleri toplamına göre etki hesaplandığında problemlerin %50'sinde azalma görülmüştür. Öğrenme oranı arttıkça, bozulma oranının değerindeki artışa rağmen açılan istasyon sayılarında azalma olmuştur.

Öğrenme ve bozulma gerçek hayatta her alanda var olan olgulardır. DHD problemlerinin işlem sürelerinin, işin süreci boyunca sabit kabul edilmesi, bu iki olgunun yok sayılması demektir. Demontaj hatları planlanırken daha doğru ve gerçekçi olarak gerçek hayat problemlerinin yansıtılması için öğrenme ve bozulmanın dikkate alınması gerekmektedir. Kontrolsüz gelişebilen bozulmaların varoluş ihtimali ve işin öğrenilmesi gerçeği kabul edilmelidir. Çünkü, küçük bir öğrenmenin, amaç fonksiyonu üzerinde ciddi azaltıcı etkileri olabilecekken, düşük orandaki bozulmanın da amaç fonksiyonunu artırdığı bu çalışmada görülmüştür.

Tez kapsamında incelenen konular dahilinde gelecekte yapılacak çalışmalar için öneriler şu şekildedir:

1. DHD problemlerinde, öğrenme ve bozulma etkilerinin istasyonlardaki kar maksimizasyonu, maliyet minimizasyonu ve benzeri farklı amaç fonksiyonu üzerindeki etkileri incelenebilir,
2. Bu çalışmada düz demontaj hattı ele alınmıştır. 2-yönlü, U şeklinde, paralel veya çoklu-paralel demontaj hatları üzerinde öğrenme ve bozulma etkileri incelenebilir.



KAYNAKÇA

1. Agrawal, S., & Tiwari, M. K. 2008. A collaborative ant colony algorithm to stochastic mixed-model U-shaped disassembly line balancing and sequencing problem. **International Journal of Production Research**, 46(6), 1405–1429. <https://doi.org/10.1080/00207540600943985>
2. Altekin, F. T. 2017. A comparison of piecewise linear programming formulations for stochastic disassembly line balancing. **International Journal of Production Research**, 55(24), 7412–7434. <https://doi.org/10.1080/00207543.2017.1351639>
3. Altekin, F. T., Kandiller, L., & Ozdemirel, N. E. 2008. Profit-oriented disassembly-line balancing. **International Journal of Production Research**, 46(10), 2675–2693. <https://doi.org/10.1080/00207540601137207>
4. Arik, O. A., & Toksarı, M. D. 2020. Minimizing Makespan With Fuzzy Processing Times Under Job Deterioration and Learning Effect. **Journal of Industrial Engineering**, 31(1), 1–17. Retrieved from <https://orcid.org/0000-0002-7088-2104>
5. Avikal, S., Mishra, P. K., & Jain, R. 2014. A Fuzzy AHP and PROMETHEE method-based heuristic for disassembly line balancing problems. **International Journal of Production Research**, 52(5), 1306–1317. <https://doi.org/10.1080/00207543.2013.831999>
6. Aydemir-Karadag, A., & Turkbey, O. 2013. Multi-objective optimization of stochastic disassembly line balancing with station paralleling. **Computers and Industrial Engineering**, 65(3), 413–425. <https://doi.org/10.1016/j.cie.2013.03.014>
7. Beamon, B. M., & Fernandes, C. 2004. Supply-chain network configuration for product recovery. **Production Planning and Control**, 15(3), 270–281. <https://doi.org/10.1080/09537280410001697701>
8. Bentaha, M. L., Battaiä, O., & Dolgui, A. 2015. An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times. **International Journal of Production Research**, 53(6), 1807–1818. <https://doi.org/10.1080/00207543.2014.961212>

9. Bentaha, M. L., Battaïa, O., Dolgui, A., & Hu, S. J. 2015. Second order conic approximation for disassembly line design with joint probabilistic constraints. **European Journal of Operational Research**, 247(3), 957–967. <https://doi.org/10.1016/j.ejor.2015.06.019>
10. Biskup, D. 1999. Single-machine scheduling with learning considerations. **European Journal of Operational Research**, 115(1), 173–178. [https://doi.org/10.1016/S0377-2217\(98\)00246-X](https://doi.org/10.1016/S0377-2217(98)00246-X)
11. Cakar, T., & Koker, R. 2015. Solving Single Machine Total Weighted Tardiness Problem with Unequal Release Date Using Neurohybrid Particle Swarm Optimization Approach. **Computational Intelligence and Neuroscience**, 2015, 1–13. <https://doi.org/10.1155/2015/838925>
12. Cheng, T. C. Edwin, & Wang, G. 2000. Single Machine Scheduling with Learning Effect Considerations. **Annals of Operations Research**, 98, 273–290. <https://doi.org/https://doi.org/10.1023/A:1019216726076>
13. Cheng, T. C.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. <https://doi.org/10.1016/j.ins.-2009.05.002>
14. Ding, L. P., Feng, Y. X., Tan, J. R., & Gao, Y. C. 2010. A new multi-objective ant colony algorithm for solving the disassembly line balancing problem. **International Journal of Advanced Manufacturing Technology**, 48(5–8), 761–771. <https://doi.org/10.1007/s00170-009-2303-5>
15. Duflou, J. R., Seliger, G., Kara, S., Umeda, Y., Ometto, A., & Willems, B. 2008. Efficiency and feasibility of product disassembly: A case-based study. **CIRP Annals - Manufacturing Technology**, 57(2), 583–600. <https://doi.org/10.1016/j.cirp.2008.09.009>
16. Duta, L., Caciula, I., & Patric, P. C. 2016. Column generation approach for disassembly line balancing. **IFAC-PapersOnLine**, 49(12), 916–920. <https://doi.org/10.1016/j.ifacol.2016.07.892>
17. Edis, E. B., Ilgin, M. A., & Edis, R. S. 2019. Disassembly line balancing with sequencing decisions: A mixed integer linear programming model and

- extensions. **Journal of Cleaner Production**, 238, 117826. <https://doi.org/10.1016/j.jclepro.2019.117826>
18. Edis, E. B. 2021. Constraint programming approaches to disassembly line balancing problem with sequencing decisions. **Computers and Operations Research**, 126, 105111. <https://doi.org/10.1016/j.cor.2020.105111>
 19. Eren, T. 2014. Learning and Deteriorating Effects on the Single Machine Scheduling Problems. **Uluslararası Muhendislik Arastirma ve Gelistirme Dergisi**, 6(1), 15–20. <https://doi.org/10.29137/umagd.346071>
 20. Garg, H. 2016. A hybrid PSO-GA algorithm for constrained optimization problems. **Applied Mathematics and Computation**, 274, 292–305. <https://doi.org/10.1016/j.amc.2015.11.001>
 21. Goksoy Kalaycilar, E., Batun, S., & Azizoğlu, M. 2022. A stochastic programming approach for the disassembly line balancing with hazardous task failures. **International Journal of Production Research**, 60(10), 3237–3262. <https://doi.org/10.1080/00207543.2021.1916119>
 22. Gonçalves, J. F., & Almeida, J. R. De. 2002. A Hybrid Genetic Algorithm for Assembly Line Balancing. **Journal of Heuristics**, 8, 629–642. <https://doi.org/10.1023/A:1020377910258>
 23. Güngör, Askiner, & Gupta, S. M. 2001. A solution approach to the disassembly line balancing problem in the presence of task failures. **International Journal of Production Research**, 39(7), 1427–1467. <https://doi.org/10.1080/00207540110052157>
 24. Güngör, Aşkiner, & Gupta, S. 1999. **Disassembly Line Balancing**. <https://doi.org/10.1201/9781420018790.ch6>
 25. Güngör, Aşkiner, & Gupta, S. M. 2002. Disassembly line in product recovery. **International Journal of Production Research**, 40(11), 2569–2589. <https://doi.org/10.1080/00207540210135622>
 26. Guo, X., Wei, T., Wang, J., Liu, S., Qin, S., & Qi, L. 2022. Multiobjective U-Shaped Disassembly Line Balancing Problem Considering Human Fatigue Index and an Efficient Solution. *IEEE Transactions on Computational Social Systems*.

<https://doi.org/10.1109/TCSS.2022.3217101>

27. Gupta, J. N. D., & Gupta, S. K. 1988. Single facility scheduling with nonlinear processing times. **Computers and Industrial Engineering**, **14**(4), 387–393. [https://doi.org/10.1016/0360-8352\(88\)90041-1](https://doi.org/10.1016/0360-8352(88)90041-1)
28. Gupta, S.M., & Gungor, A. 2001. Product recovery using a disassembly line: challenges and solution. *Proceedings of the 2001 IEEE International Symposium on Electronics and the Environment. 2001 IEEE ISEE (Cat. No.01CH37190)*, (1), 36–40. <https://doi.org/10.1109/ISEE.2001.924499>
29. Hezer, S., & Kara, Y. 2015. A network-based shortest route model for parallel disassembly line balancing problem. **International Journal of Production Research**, **53**(6), 1849–1865. <https://doi.org/10.1080/00207543.2014.965348>
30. Ilgin, M. A., Akçay, H., & Araz, C. 2017. Disassembly line balancing using linear physical programming. **International Journal of Production Research**, **55**(20), 6108–6119. <https://doi.org/10.1080/00207543.2017.1324225>
31. İşçi, Ö., & Korukoğlu, S. 2003. Genetik Algoritma Yaklaşımı ve Yöneylem Araştırmasında Bir Uygulama. **Yönetim ve Ekonomi: Celal Bayar Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, **10**(2), 191–208.
32. Kalayci, C. B., & Gupta, S. M. 2013a. A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem. **International Journal of Advanced Manufacturing Technology**, **69**(1–4), 197–209. <https://doi.org/10.1007/s00170-013-4990-1>
33. Kalayci, C. B., & Gupta, S. M. 2013b. Ant colony optimization for sequence-dependent disassembly line balancing problem. **Journal of Manufacturing Technology Management**, **24**(3), 413–427. <https://doi.org/10.1108/1741038-1311318909>
34. Kalayci, C. B., & Gupta, S. M. 2013c. Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. **Expert Systems with Applications**, **40**(18), 7231–7241. <https://doi.org/10.1016/j.eswa.2013.06.067>
35. Kalayci, C. B., & Gupta, S. M. 2014. A Tabu Search Algorithm For Balancing A Sequence-Dependent Disassembly Line. **Production Planning and Control**,

- 25(2), 149–160. <https://doi.org/10.1080/09537287.2013.782949>
36. Kalayci, C. B., Gupta, S. M., & Nakashima, K. 2012. A Simulated Annealing Algorithm for Balancing a Disassembly Line. **Design for Innovative Value Towards a Sustainable Society**, 714–719. https://doi.org/10.1007/978-94-007-3010-6_143
 37. Kalayci, C. B., Hancilar, A., Gungor, A., & Gupta, S. M. 2015. Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm. **Journal of Manufacturing Systems**, 37, 672–682. <https://doi.org/10.1016/j.jmsy.2014.11.015>
 38. Kalayci, C. B., Polat, O., & Gupta, S. M. 2016. A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem. **Annals of Operations Research**, 242(2), 321–354. <https://doi.org/10.1007/s10479-014-1641-3>
 39. Kalayci, E. G., Azizoglu, M., & Yeralan, S. 2016. A disassembly line balancing problem with fixed number of workstations. **European Journal of Operational Research**, 249(2), 592–604. <https://doi.org/10.1016/j.ejor.2015.09.004>
 40. Kennedy, J., & Eberhart, R. 1995. Particle Swarm Optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks.IEEE*, 1942–1948. https://doi.org/10.1007/978-3-031-17922-8_4
 41. Koc, A., Sabuncuoglu, I., & Erel, E. 2009. Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an AND/OR graph. *IIE Transactions (Institute of Industrial Engineers)*, 41(10), 866–881. <https://doi.org/10.1080/07408170802510390>
 42. Kucukkoc, I., Li, Z., & Li, Y. 2020. Type-E disassembly line balancing problem with multi-manned workstations. **Optimization and Engineering**, 21(2), 611–630. <https://doi.org/10.1007/s11081-019-09465-y>
 43. Laili, Y., Li, Y., Fang, Y., Pham, D. T., & Zhang, L. 2020. Model review and algorithm comparison on multi-objective disassembly line balancing. **Journal of Manufacturing Systems**, 56(December 2019), 484–500. <https://doi.org/10.1016/j.jmsy.2020.07.015>
 44. Levitin, G., Rubinovitz, J., & Shnits, B. 2006. A genetic algorithm for robotic

- assembly line balancing. **European Journal of Operational Research**, **168**(3), 811–825. <https://doi.org/10.1016/j.ejor.2004.07.030>
45. Li, Z., Çil, Z. A., Mete, S., & Kucukkoc, I. 2020. A fast branch, bound and remember algorithm for disassembly line balancing problem. **International Journal of Production Research**, **58**(11), 3220–3234. <https://doi.org/10.1080/0020-7543.2019.1630774>
 46. Li, Z., & Janardhanan, M. N. 2021. Modelling and solving profit-oriented U-shaped partial disassembly line balancing problem. **Expert Systems with Applications**, **183**(October 2019), 115431. <https://doi.org/10.1016/j.eswa.2021.115431>
 47. Li, Z., Kucukkoc, I., Tang, Q., & Zhang, Z. 2023. Models and two-phase bee algorithms for multi-objective U-shaped disassembly line balancing problem. In **Optimization and Engineering** (Vol. 24). <https://doi.org/10.1007/s11081-021-09696-y>
 48. Liang, W., Zhang, Z., Yin, T., Zhang, Y., & Wu, T. 2023. Modelling and optimisation of energy consumption and profit-oriented multi-parallel partial disassembly line balancing problem. **International Journal of Production Economics**, **262**(May), 108928. <https://doi.org/10.1016/j.ijpe.2023.108928>
 49. Liang, W., Zhang, Z., Yin, T., Zhang, Y., & Wu, T. 2023. Modelling and optimisation of energy consumption and profit-oriented multi-parallel partial disassembly line balancing problem. **International Journal of Production Economics**, **262**(108928).
 50. Liang, W., Zhang, Z., Zhang, Y., Xu, P., & Yin, T. 2022. Improved social spider algorithm for partial disassembly line balancing problem considering the energy consumption involved in tool switching. **International Journal of Production Research**, **61**(7), 2250–2266. <https://doi.org/10.1080/00207543.2022.2069059>
 51. Liang, X. X., Zhang, B., Wang, J. B., Yin, N., & Huang, X. 2019. Study on flow shop scheduling with sum-of-logarithm-processing-times-based learning effects. **Journal of Applied Mathematics and Computing**, **61**(1–2), 373–388. <https://doi.org/10.1007/s12190-019-01255-0>
 52. Liu, Jia, & Wang, S. 2017. Balancing disassembly line in product recovery to promote the coordinated development of economy and environment.

Sustainability (Switzerland), 9(2). <https://doi.org/10.3390/su9020309>

53. Liu, Jiayi, Zhou, Z., Pham, D. T., Xu, W., Yan, J., Liu, A., ... Liu, Q. 2018. An improved multi-objective discrete bees algorithm for robotic disassembly line balancing problem in remanufacturing. **International Journal of Advanced Manufacturing Technology**, 97(9–12), 3937–3962. <https://doi.org/10.1007/s00170-018-2183-7>
54. Liu, J., Zhou, Z., Pham, D. T., Xu, W., Ji, C., & Liu, Q. 2020. Collaborative optimization of robotic disassembly sequence planning and robotic disassembly line balancing problem using improved discrete Bees algorithm in remanufacturing☆. **Robotics and computer-integrated manufacturing**, 61, 101829. <https://doi.org/10.1016/j.rcim.2019.101829>
55. Mazdeh, M. M., Zaerpour, F., & Jahantigh, F. F. 2010. A fuzzy modeling for single machine scheduling problem with deteriorating jobs. **International Journal of Industrial Engineering Computations**, 1(2), 147–156. <https://doi.org/10.5267/j.ijiec.2010.02.004>
56. McGovern, S. M., & Gupta, S. M. 2007. Combinatorial optimization analysis of the unary NP-complete disassembly line balancing problem. **International Journal of Production Research**, 45(18–19), 4485–4511. <https://doi.org/10.1080/00207540701476281>
57. McGovern, Seamus M., & Gupta, S. M. 2004. <Title>2-Opt Heuristic for the Disassembly Line Balancing Problem</Title>. **Environmentally Conscious Manufacturing III**, 5262, 71–84. <https://doi.org/10.1117/12.516155>
58. McGovern, Seamus M., & Gupta, S. M. 2005. Uninformed and Probabilistic Distributed Agent Combinatorial Searches for the Unary NP-Complete Disassembly Line Balancing Problem. In Surendra M. Gupta (Ed.), **Environmentally Conscious Manufacturing V** (Vol. 5997, pp. 81–92). <https://doi.org/10.1117/12.629121>
59. McGovern, Seamus M., & Gupta, S. M. 2006. Ant Colony Optimization For Disassembly Sequencing With Multiple Objectives. **International Journal of Advanced Manufacturing Technology**, 30(5–6), 481–496. <https://doi.org/10.1007/s00170-005-0037-6>

60. McGovern, Seamus M., & Gupta, S. M. 2007. A balancing method and genetic algorithm for disassembly line balancing. **European Journal of Operational Research**, **179**(3), 692–708. <https://doi.org/10.1016/j.ejor.2005.03.055>
61. Mondragon, A. E. C., Lalwani, C., & Mondragon, C. E. C. 2011. Measures for auditing performance and integration in closed-loop supply chains. **Supply Chain Management**, **16**(1), 43–56. <https://doi.org/10.1108/135985411111103494>
62. Mosheiov, G. 1994. Scheduling jobs under simple linear deterioration. **Computers & Operations Research**, **21**(6), 653–659. [https://doi.org/10.1016/0305-0548\(94\)90080-9](https://doi.org/10.1016/0305-0548(94)90080-9)
63. Mosheiov, G. 2001. Scheduling problems with a learning effect. **European Journal of Operational Research**, **132**(3), 687–693. [https://doi.org/10.1016/S0377-2217\(00\)00175-2](https://doi.org/10.1016/S0377-2217(00)00175-2)
64. Nakıboğlu, G. 2007. Tersine Lojistik: Önemi ve Dünyadaki Uygulamaları. **Gazi Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, **2**(2), 181–196. Retrieved from <http://dergipark.gov.tr/gaziuiibfd/issue/28332/301099>
65. Özceylan, E., Kalayci, C. B., Güngör, A., & Gupta, S. M. 2019. Disassembly line balancing problem: a review of the state of the art and future directions. **International Journal of Production Research**, **57**(15–16), 4805–4827. <https://doi.org/10.1080/00207543.2018.1428775>
66. Özceylan, E., & Paksoy, T. 2013. Reverse supply chain optimisation with disassembly line balancing. **International Journal of Production Research**, **51**(20), 5985–6001. <https://doi.org/10.1080/00207543.2013.784405>
67. Özceylan, E., Paksoy, T., & Bektaş, T. 2014. Modeling and optimizing the integrated problem of closed-loop supply chain network design and disassembly line balancing. **Transportation Research Part E: Logistics and Transportation Review**, **61**, 142–164. <https://doi.org/10.1016/j.tre.2013.11.001>
68. Paksoy, S., & Uzun, A. 2008. Genetik Algoritma ile Kaynak Kısıtlı Proje Çizelgeleme. **Ç.Ü. Sosyal Bilimler Enstitüsü Dergisi**, **17**(2), 345–362.
69. Paksoy, T., Güngör, A., Özceylan, E., & Hancılar, A. 2013. Mixed model disassembly line balancing problem with fuzzy goals. **International Journal of**

- Production Research**, **51(20)**, 6082–6096. <https://doi.org/10.1080/00207543.2013.795251>
70. Pei, J., Song, Q., Liao, B., Liu, X., & Pardalos, P. M. 2021. Parallel-machine serial-batching scheduling with release times under the effects of position-dependent learning and time-dependent deterioration. **Annals of Operations Research**, **298(1–2)**, 407–444. <https://doi.org/10.1007/s10479-020-03555-2>
 71. Qin, S., Li, J., Wang, J., Guo, X., Liu, S., & Qi, L. 2023. A Salp Swarm Algorithm for Parallel Disassembly Line Balancing Considering Workers With Government Benefits. *IEEE Transactions on Computational Social Systems*, *PP*, 1–10. <https://doi.org/10.1109/TCSS.2023.3238965>
 72. Ren, Y., Meng, L., Zhang, C., Zhao, F., Saif, U., Huang, A., ... Sutherland, J. W. 2020. An Efficient Metaheuristics For A Sequence-Dependent Disassembly Planning. **Journal of Cleaner Production**, **245**, 118644. <https://doi.org/10.1016/j.jclepro.2019.118644>
 73. Ren, Y., Yu, D., Zhang, C., Tian, G., Meng, L., & Zhou, X. 2017. An improved gravitational search algorithm for profit-oriented partial disassembly line balancing problem. **International Journal of Production Research**, **55(24)**, 7302–7316. <https://doi.org/10.1080/00207543.2017.1341066>
 74. Ren, Y., Zhang, C., Zhao, F., Tian, G., Lin, W., Meng, L., & Li, H. 2018. Disassembly line balancing problem using interdependent weights-based multi-criteria decision making and 2-Optimal algorithm. **Journal of Cleaner Production**, **174**, 1475–1486. <https://doi.org/10.1016/j.jclepro.2017.10.308>
 75. Rudek, R. 2011. Computational complexity and solution algorithms for flowshop scheduling problems with the learning effect. **Computers and Industrial Engineering**, **61(1)**, 20–31. <https://doi.org/10.1016/j.cie.2011.02.005>
 76. Scholl, A. 1995. Data of assembly line balancing problems. **Schriften Zur Quantitativen Betriebswirtschaftslehre**, Vol. 93, pp. 1–28.
 77. Tian, G., Zhang, X., Fathollahi-Fard, A. M., Jiang, Z., Zhang, C., Yuan, G., & Pham, D. T. 2023. Hybrid evolutionary algorithm for stochastic multiobjective disassembly line balancing problem in remanufacturing. *Environmental Science and Pollution Research*, 1-16.

78. Toksarı, M. Duran, İşleyen, S. K., Güner, E., & Baykoç, Ö. F. 2010. Assembly line balancing problem with deterioration tasks and learning effect. **Expert Systems with Applications**, 37(2), 1223–1228. <https://doi.org/10.1016/j.eswa.2009.06.005>
79. Toksarı, Mehmet Duran, Aydogan, E. K., Atalay, B., & Sarı, S. 2022. Some Scheduling Problems With Sum of Logarithm Processing Times Based Learning Effect and Exponential Past Sequence Dependent Delivery Times. **Journal of Industrial and Management Optimization**, 18(3), 1795–1807. <https://doi.org/10.3934/jimo.2021044>
80. Topoyan, M. (2019). *Üretim Çizelgeleme*.
81. Tuncel, E., Zeid, A., & Kamarthi, S. 2014. Solving large scale disassembly line balancing problem with uncertainty using reinforcement learning. **Journal of Intelligent Manufacturing**, 25(4), 647–659. <https://doi.org/10.1007/s10845-012-0711-0>
82. Wang, H., & Hsu, H. 2010. Computers & Operations Research A closed-loop logistic model with a spanning-tree based genetic algorithm. **Computers and Operation Research**, 37(2), 376–389. <https://doi.org/10.1016/j.cor.2009.06.001>
83. Wang, J. B., Ng, C. T., Cheng, T. C. E., & Liu, L. L. 2008. Single-machine scheduling with a time-dependent learning effect. **International Journal of Production Economics**, 111(2), 802–811. <https://doi.org/10.1016/j.ijpe.2007.03.013>
84. Wang, J. B., & Xia, Z. Q. 2005. Flow-shop scheduling with a learning effect. **Journal of the Operational Research Society**, 56(11), 1325–1330. <https://doi.org/10.1057/palgrave.jors.2601856>
85. Wang, K., Gao, L., & Li, X. 2020. A multi-objective algorithm for U-shaped disassembly line balancing with partial destructive mode. **Neural Computing and Applications**, 32(16), 12715–12736. <https://doi.org/10.1007/s00521-020-04721-0>
86. Wang, K., Li, X., & Gao, L. 2019. Modeling and optimization of multi-objective partial disassembly line balancing problem considering hazard and profit. **Journal of Cleaner Production**, 211, 115–133. <https://doi.org/10.1016/j.jclepro.2019.03.013>

2018.11.114

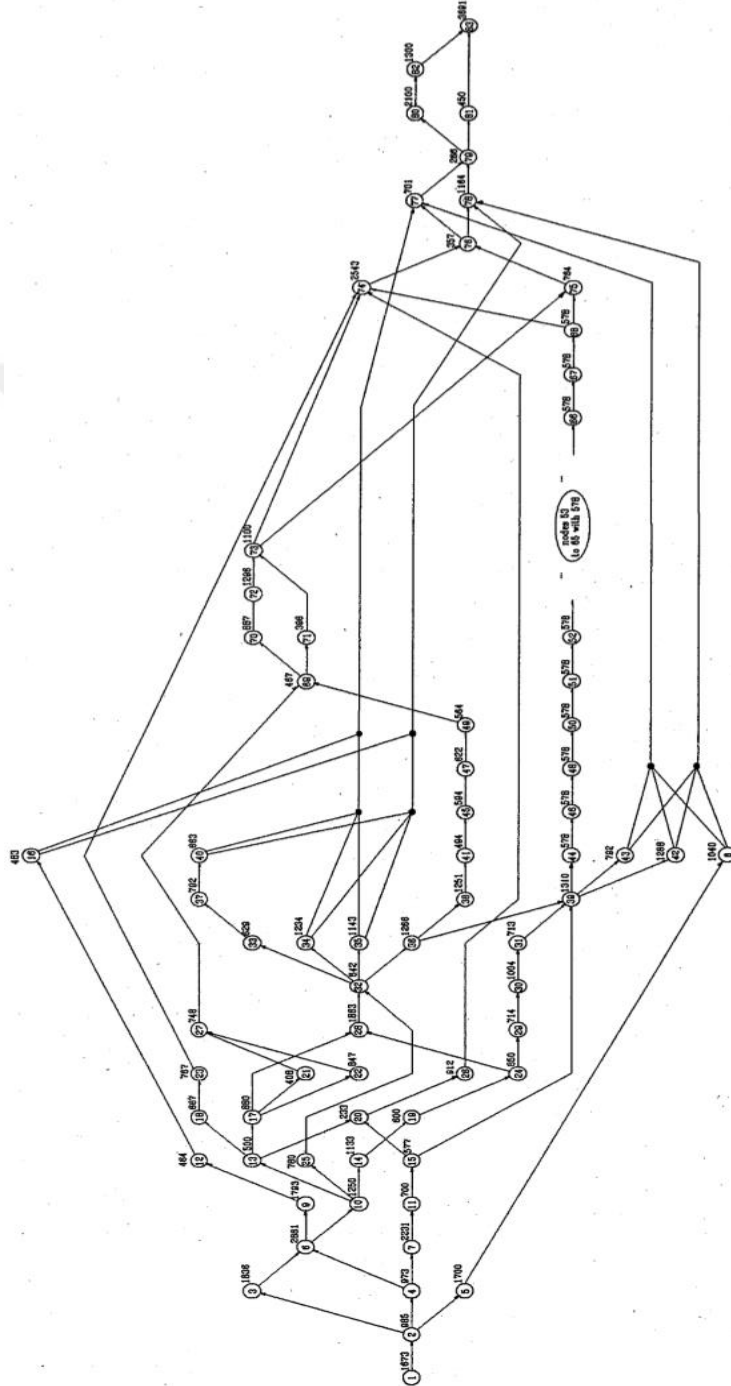
87. WRIGHT, T. P. 1936. Factors Affecting the Cost of Airplanes. **Journal of the Aeronautical Sciences**, 3(4), 122–128. <https://doi.org/10.2514/8.155>
88. Xiao, S., Wang, Y., Yu, H., & Nie, S. 2017. An entropy-based adaptive hybrid particle swarm optimization for disassembly line balancing problems. **Entropy**, 19(11). <https://doi.org/10.3390/e19110596>
89. Xu, J., Wu, C. C., Yin, Y., Zhao, C., Chiou, Y. T., & Lin, W. C. 2016. An order scheduling problem with position-based learning effect. **Computers and Operations Research**, 74, 175–186. <https://doi.org/10.1016/j.cor.2016.04.021>
90. Yang, G. fen, Wang, Z. ping, & Li, X. qiang. 2009. The optimization of the closed-loop supply chain network. **Transportation Research Part E: Logistics and Transportation Review**, 45(1), 16–28. <https://doi.org/10.1016/j.tre.2008.02.007>
91. Yin, T., Zhang, Z., & Jiang, J. 2021. A Pareto-discrete hummingbird algorithm for partial sequence-dependent disassembly line balancing problem considering tool requirements. **Journal of Manufacturing Systems**, 60(August 2020), 406–428. <https://doi.org/10.1016/j.jmsy.2021.07.005>
92. Yin, T., Zhang, Z., Wu, T., Zeng, Y., Zhang, Y., & Liu, J. 2023. Multimanned partial disassembly line balancing optimization considering end-of-life states of products and skill differences of workers. **Journal of Manufacturing Systems**, 66(July 2022), 107–126. <https://doi.org/10.1016/j.jmsy.2022.12.002>
93. Zhang, Y., Zhang, Z., Guan, C., & Xu, P. 2022. Improved whale optimisation algorithm for two-sided disassembly line balancing problems considering part characteristic indexes. **International Journal of Production Research**, 60(8), 2553–2571. <https://doi.org/10.1080/00207543.2021.1897178>
94. Zhang, Z., Wang, K., Zhu, L., & Wang, Y. 2017. A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem. **Expert Systems with Applications**, 86, 1339–1351. <https://doi.org/10.1016/j.eswa.2017.05.053>
95. Zhou, B., & Bian, J. 2022. Multi-mechanism-based modified bi-objective Harris Hawks optimization for sustainable robotic disassembly line balancing problems.

Engineering Applications of Artificial Intelligence, 116(September), 105479.
<https://doi.org/10.1016/j.engappai.2022.105479>

96. Zhu, L., Zhang, Z., & Guan, C. 2020. Multi-objective partial parallel disassembly line balancing problem using hybrid group neighbourhood search algorithm. **Journal of Manufacturing Systems**, 56(May), 252–269. <https://doi.org/10.1016/j.jmsy.2020.06.013>
97. Zhu, L., Zhang, Z., & Wang, Y. 2018. A Pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation. **International Journal of Production Research**, 56(24), 7354–7374. <https://doi.org/10.1080/00207543.2018.1471238>

EKLER

EK 1. 83-parçalı problemin işler arasındaki öncelik grafiği (Arcus1) (Scholl,1995)



EK 2. Pozisyon tabanlı öğrenme etkisi altında 8 parçalı DHD probleminin Lingo.11 kodu
(öğrenme oranı=0.90)

```

model:
sets:
islem/1..8/;
islem2/1..8/;
istasyon/1..6/;
sira/1..8/;
sira2/1..8/;
oncul (islem,islem)/1,2 1,3 1,5 2,6 3,6 6,8 2,8 3,8 5,8 8,7 7,4/;
link1(islem,istasyon):
y;
link2(islem,istasyon,sira):
x;
link3(istasyon,sira):
t;
link4(islem):
g;
link5(istasyon,sira):
c;
link6(istasyon):
z;
link7(istasyon,islem):
b;
endsets
data:
M=1000;
ct=40;
e=-0.152;
g=14 10 12 18 23 16 20 36;
enddata
min=@sum(istasyon(j):z(j));
@for(islem(i):@sum(istasyon(j):y(i,j))=1);
@for(istasyon(j):@sum(islem(i):y(i,j)*g(i))<=z(j)*ct);
@for(oncul(a,k):@for(istasyon(j):@sum(sira(r)|r#NE#R:r*(x(a,j,r))*b(j,r))<=@sum(sira2(R):R*(x(k,j,R))*b(j,R))));
@for(sira(r):@for(istasyon(j):@sum(islem(i):x(i,j,r))<=1));
@for(islem(i):@for(istasyon(j):@sum(sira(r):x(i,j,r))=y(i,j)));
@for(istasyon(j):@for(sira(r):@sum(islem(i):x(i,j,r))=b(j,r)));
@for(istasyon(j):@for(sira(r)|r#NE#1:@sum(islem(i):g(i)*x(i,j,r)*r^e)=t(j,r)));
@for(istasyon(j):@for(sira(r)|r#EQ#1:@sum(islem(i):g(i)*x(i,j,r))*r^e*b(j,r)=t(j,r)));
@for(istasyon(j):@for(sira(r)|r#NE#1:c(j,r)=(c(j,r-1)+t(j,r))));
@for(istasyon(j):@for(sira(r)|r#NE#1:t(j,r)*b(j,r)<=t(j,r-1)*b(j,r-1)*m));
@for(istasyon(j):@for(sira(r)|r#NE#1:c(j,r)*b(j,r)<=c(j,r-1)*b(j,r-1)*m));
@for(istasyon(j):@for(sira(r)|r#EQ#1:c(j,r)=t(j,r)));
@for(istasyon(j):@sum(sira(r):(t(j,r)))<=ct);
@for(istasyon(j):@for(sira(r):@for(islem2(I)|I#NE#i:x(I,j,r-1)<=@sum(islem(i):x(i,j,r))));
@for(islem(i):@for(istasyon(j):@for(sira(r):@bin(x(i,j,r))));
@for(islem(i):@for(istasyon(j):@bin(y(i,j))));
@for(istasyon(j):@for(sira(r):@bin(b(j,r))));
@for(istasyon(j):@bin(z(j)));

```

EK 3. Pozisyon tabanlı öğrenme ve bozulma etkileri altında 8 parçalı DHD probleminin Lingo.11 kodu (öğrenme oranı= 0.90 ve bozulma oranı=0.10)

```

model:
sets:
islem/1..8/;
islem2/1..8/;
istasyon/1..6/;
sira/1..8/;
sira2/1..8/;
oncul (islem,islem)/1,2 1,3 1,5 2,6 3,6 6,8 2,8 3,8 5,8 8,7 7,4/;
link1(islem,istasyon):
y;
link2(islem,istasyon,sira):
x;
link3(istasyon,sira):
t;
link4(islem):
g;
link5(istasyon,sira):
c;
link6(istasyon):
z;
link7(istasyon,islem):
b;
endsets
data:
M=1000;
ct=40;
e=-0.152;
alfa=0.1;
g=14 10 12 18 23 16 20 36;
enddata
min=@sum(istasyon(j):z(j));
@for(islem(i):@sum(istasyon(j):y(i,j))=1);
@for(istasyon(j):@sum(islem(i):y(i,j))*g(i)<=z(j)*ct);
@for(oncul(a,k):@for(istasyon(j):@sum(sira(r)|r#NE#R:r*(x(a,j,r))*b(j,r))<=@sum(sira2(R):R*(x(k,j,R))*b(j,R))));
@for(sira(r):@for(istasyon(j):@sum(islem(i):x(i,j,r))<=1));
@for(islem(i):@for(istasyon(j):@sum(sira(r):x(i,j,r))=y(i,j)));
@for(istasyon(j):@for(sira(r):@sum(islem(i):x(i,j,r))=b(j,r)));
@for(istasyon(j):@for(sira(r)|r#NE#1:@sum(islem(i):g(i)*x(i,j,r)*r^e)+
c(j,r-1)*r^e*alfa*b(j,r)=t(j,r)));
@for(istasyon(j):@for(sira(r)|r#EQ#1:@sum(islem(i):g(i)*x(i,j,r))*r^e*
b(j,r)=t(j,r)));
@for(istasyon(j):@for(sira(r)|r#NE#1:c(j,r)=(c(j,r-1)+t(j,r))));
@for(istasyon(j):@for(sira(r)|r#NE#1:t(j,r)*b(j,r)<=t(j,r-1)*b(j,r-
1)*m));
@for(istasyon(j):@for(sira(r)|r#NE#1:c(j,r)*b(j,r)<=c(j,r-1)*b(j,r-
1)*m));
@for(istasyon(j):@for(sira(r)|r#EQ#1:c(j,r)=t(j,r)));
@for(istasyon(j):@sum(sira(r):(t(j,r)))<=ct);
@for(istasyon(j):@for(sira(r):@for(islem2(I)|I#NE#i:x(I,j,r-
1)<=@sum(islem(i):x(i,j,r))));
@for(islem(i):@for(istasyon(j):@for(sira(r):@bin(x(i,j,r)))));
@for(islem(i):@for(istasyon(j):@bin(y(i,j))));
@for(istasyon(j):@for(sira(r):@bin(b(j,r))));
@for(istasyon(j):@bin(z(j)));

```

EK 4. İş bozulması etkisi altında 8 parçalı DHD probleminin Lingo.11 kodu (bozulma oranı=0.20)

```

model:
sets:
islem/1..8/;
islem2/1..8/;
istasyon/1..6/;
sira/1..8/;
sira2/1..8/;
oncul (islem,islem)/1,2 1,3 1,5 2,6 3,6 6,8 2,8 3,8 5,8 8,7 7,4/;
link1(islem,istasyon):
y;
link2(islem,istasyon,sira):
x;
link3(istasyon,sira):
t;
link4(islem):
g;
link5(istasyon,sira):
c;
link6(istasyon):
z;
link7(istasyon,islem):
b;
endsets
data:
M=1000;
ct=40;
alfa=0.2; !bozulma etkisi;
g=14 10 12 18 23 16 20 36;
enddata
min=@sum(istasyon(j):z(j));
@for(islem(i):@sum(istasyon(j):y(i,j))=1);
@for(istasyon(j):@sum(islem(i):y(i,j)*g(i))<=z(j)*ct);
@for(oncul(a,k):@for(istasyon(j):@sum(sira(r)|r#NE#R:r*(x(a,j,r))*b(j,r))<=@sum(sira2(R):R*(x(k,j,R))*b(j,R))));
@for(sira(r):@for(istasyon(j):@sum(islem(i):x(i,j,r))<=1));
@for(islem(i):@for(istasyon(j):@sum(sira(r):x(i,j,r))=y(i,j)));
@for(istasyon(j):@for(sira(r):@sum(islem(i):x(i,j,r))=b(j,r)));
@for(istasyon(j):@for(sira(r)|r#NE#1:@sum(islem(i):g(i)*x(i,j,r))+c(j,r-1)*alfa*b(j,r)=t(j,r)));
@for(istasyon(j):@for(sira(r)|r#EQ#1:@sum(islem(i):g(i)*x(i,j,r))*b(j,r)=t(j,r)));
@for(istasyon(j):@for(sira(r)|r#NE#1:c(j,r)=(c(j,r-1)+t(j,r))*b(j,r)));
@for(istasyon(j):@for(sira(r)|r#NE#1:t(j,r)*b(j,r)<=t(j,r-1)*b(j,r-1)*m));
@for(istasyon(j):@for(sira(r)|r#NE#1:c(j,r)*b(j,r)<=c(j,r-1)*b(j,r-1)*m));
@for(istasyon(j):@for(sira(r)|r#EQ#1:c(j,r)=t(j,r)));
@for(istasyon(j):@sum(sira(r):(t(j,r)))<=ct);
@for(istasyon(j):@for(sira(r):@for(islem2(I)|I#NE#i:x(I,j,r-1)<=@sum(islem(i):x(i,j,r))));
@for(islem(i):@for(istasyon(j):@for(sira(r):@bin(x(i,j,r)))));
@for(islem(i):@for(istasyon(j):@bin(y(i,j))));
@for(istasyon(j):@for(sira(r):@bin(b(j,r))));
@for(istasyon(j):@bin(z(j)));

```

EK 5. İş bozulması etkisi altında 8 parçalı DHD probleminin öğrenme ve bozulma etkileri altında HGA ile çözümünün C# kodu

```
using System;
using System.Collections.Generic;
namespace initialpopcreating
{
    class Program
    {
        static void Main(string[] args)
        {
            int sonucsayisi = 1;
            //double[] lrlist = new double[] { 0, -0.152, -0.23446, -0.322, -0.415, -0.514 };
            double[] lrlist = new double[] { -0.152, -0.322, -0.514 };
            double[] drlist = new double[] { 0.1, 0.2 };
            //double[] drlist = new double[] {0};
            for (int lrsay = 0; lrsay < lrlist.Length; lrsay++)
            {
                double lr = lrlist[lrsay];
                for (int drsay = 0; drsay < drlist.Length; drsay++)
                {
                    double dr = drlist[drsay];
                    int M = 1000000;
                    int globalsol = M;
                    double mutrate = 0.2;
                    double corate = 0.8;
                    ////8 veri için////////
                    int[,] precedence = new int[,]{ { 1, 2 }, { 1, 3 }, { 7, 4 }, { 1, 5 }, { 2, 6 }, {
3, 6 }, { 8, 7 }, { 5, 4 }, { 5, 8 }, { 6, 8 } };
                    int ct = 40;
                    int[] processtime = new int[] { 14, 10, 12, 18, 23, 16, 20, 36 };
                    int n = processtime.Length;//n= görev sayısı
                    int[] iskumesi = new int[n];
                    for (int i = 0; i < n; i++)
```

```

{
    iskumesi[i] = i + 1;
}
int[,] globalsira = new int[n, 3];
int ebeveynpopsize = 2000;
int[] globalissirasi = new int[n];
double[,] globaltjr = new double[n, n];
double[,] globalcjr = new double[n, n];
int[,] ebeveynpop = new int[ebeveynpopsize, n];
int[] sonuckumesi = new int[ebeveynpopsize];
int[,] istasyongorevsayisi = new int[ebeveynpopsize, n];
// int[,] ebeveynpop = new int[ebeveynpopsize, n, 3];
for (int ebs = 0; ebs < ebeveynpopsize; ebs++)
{
    int[,] localsira = new int[n, 3];
    int[] localissirasi = new int[n];
    double[,] localtjr = new double[n, n];
    double[,] localcjr = new double[n, n];
    int localsol = M;
    int popsize = 100;
    for (int itnum = 0; itnum < popsize; itnum++)
    {
        double toplamsonuc = 0;
        double sonuc = 0;
        double sonuc1 = 0;
        List<int> elemanlar = new List<int>();
        for (int k = 1; k < n + 1; k++)
        {
            elemanlar.Add(k);
        }
        int[] rassalsira = new int[n];
        int[,] sira = new int[n, 3];
        int[] ctlist = new int[n];

```

```

int[,] sr = new int[n, n];
int aktifist = 0;
double[] ctaktif = new double[n];
double[] ctaktifyapay = new double[n];
int aktifsirano = 0;
double[,] tjr = new double[n, n];
double[,] cjr = new double[n, n];
for (int i = 0; i < processtime.Length; i++)
{
    Random rnd = new Random();
    int sayi = rnd.Next(1, (elemanlar.Count) + 1);
    rassalsira[i] = elemanlar[sayi - 1];
    elemanlar.RemoveAt(sayi - 1);
}

for (int i = 0; i < rassalsira.Length; i++)
{
    int islemdekiis = rassalsira[i];
    if (aktifsirano == 0)
    {
        tjr[aktifist, aktifsirano] = processtime[islemdekiis - 1];
        cjr[aktifist, aktifsirano] = tjr[aktifist, aktifsirano];
    }
    if (aktifsirano > 0)
    {
        double suretoplamlam = 0;
        for (int sirasay = 0; sirasay < aktifsirano; sirasay++)
        {
            suretoplamlam = Math.Log(tjr[aktifist, sirasay] )+ suretoplamlam;
        }
        //tjr[aktifist, aktifsirano] = (processtime[islemdekiis - 1] * Math.Pow(aktifsirano + 1,
        lr)); //pozisyon tabanlı öğrenme etkisi altında

```

```

//tjr[aktifist, aktifsirano] = processtime[islemdekiis - 1] * Math.Pow((1 + suretoplami), lr);
//logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında
//tjr[aktifist, aktifsirano] = (processtime[islemdekiis - 1] + cjr[aktifist, aktifsirano - 1] *
dr); //bozulma etkisi altında
tjr[aktifist, aktifsirano] = (processtime[islemdekiis - 1] * Math.Pow(aktifsirano + 1, lr))
+ (cjr[aktifist, aktifsirano - 1] * Math.Pow(aktifsirano + 1, lr) * dr); //pozisyon tabanlı
öğrenme ve bozulma etkisi altında
//tjr[aktifist, aktifsirano] = (processtime[islemdekiis - 1] + cjr[aktifist, aktifsirano - 1] *
dr) * Math.Pow((1 + suretoplami), lr); //logaritmik işlem süreleri toplamı tabanlı öğrenme
e bozulma etkisi altında
cjr[aktifist, aktifsirano] = cjr[aktifist, aktifsirano - 1] + tjr[aktifist, aktifsirano];
    }
    ctaktifyapay[aktifist] = cjr[aktifist, aktifsirano];
    if (ctaktifyapay[aktifist] <= ct)
    {
        int bulundu = 0;
        if (aktifsirano > 0)
        {
            int a1 = islemdekiis;
            for (int j = 0; j < aktifsirano; j++)
            {
                int a2 = sr[aktifist, j];
                for (int z = 0; z < precedence.GetLength(0); z++)
                {
                    if (precedence[z, 1] == a2 && precedence[z, 0] == a1)
                    {
                        bulundu += 1;
                    }
                }
            }
        }
        if (bulundu == 0)
        {

```

```

    sira[i, 0] = rassalsira[i];
    sira[i, 1] = aktifist;
    sira[i, 2] = aktifsirano;
    sr[aktifist, aktifsirano] = islemdekiis;
    ctaktif[aktifist] = ctaktifyapay[aktifist];
    aktifsirano += 1;
}
if (bulundu > 0)
{
    tjr[aktifist, aktifsirano] = 0;
    cjr[aktifist, aktifsirano] = 0;
    aktifsirano = 0;
    aktifist += 1;
    sira[i, 0] = rassalsira[i];
    sira[i, 1] = aktifist;
    sira[i, 2] = aktifsirano;
    sr[aktifist, aktifsirano] = islemdekiis;
    ctaktif[aktifist] = processtime[islemdekiis - 1];
    ctaktifyapay[aktifist] = processtime[islemdekiis - 1];
    tjr[aktifist, aktifsirano] = processtime[islemdekiis - 1];
    cjr[aktifist, aktifsirano] = tjr[aktifist, aktifsirano];
    aktifsirano += 1;
}
}
if (ctaktifyapay[aktifist] > ct)
{
    tjr[aktifist, aktifsirano] = 0;
    cjr[aktifist, aktifsirano] = 0;
    aktifsirano = 0;
    aktifist += 1;
    sira[i, 0] = rassalsira[i];
    sira[i, 1] = aktifist;
    sira[i, 2] = aktifsirano;

```

```

        sr[aktifist, aktifsirano] = islemdekiis;
        ctaktif[aktifist] = processtime[islemdekiis - 1];
        ctaktifyapay[aktifist] = processtime[islemdekiis - 1];
        tjr[aktifist, aktifsirano] = processtime[islemdekiis - 1];
        cjr[aktifist, aktifsirano] = tjr[aktifist, aktifsirano];
        aktifsirano += 1;
    }
}
if (aktifist + 1 < localsol)
{
    localsol = aktifist + 1;
    localsira = sirasira;
    localissirasi = rassalsira;
    localcjr = cjr;
    localtjr = tjr;
}
}
sonuckumesi[ebs] = localsol;
int[, ] lsmipop = new int[sonuckumesi[ebs], n];
for (int i = 0; i < n; i++)
{
    ebeveynpop[ebs, i] = localissirasi[i];
    for (int j = 0; j < sonuckumesi[ebs]; j++)
    {
        lsmipop[j, i] = localissirasi[i];
    }
}
if (localsol < globalsol)
{
    globalsol = localsol;
    globalsira = localsira;
    globalissirasi = localissirasi;
    globalcjr = localcjr;
}

```

```

        globaltjr = localtjr;
    }
    /////local search method application
    int istasyonsay = 0;
    int sayackac = 0;
    for (int i = 0; i < n; i++)
    {
        if (istasyonsay == localsira[i, 1])
        {
            sayackac += 1;
            istasyongorevsayisi[ebs, istasyonsay] = sayackac;
        }
        else
        {
            istasyonsay += 1;
            sayackac = 0;
            istasyongorevsayisi[ebs, istasyonsay] = sayackac += 1; }
    }
    int[] kacinciislist = new int[sonuckumesi[ebs]];
    for (int i = 1; i < sonuckumesi[ebs]; i++)
    {
        kacinciislist[i] = kacinciislist[i - 1] + istasyongorevsayisi[ebs, i - 1];
    }
    for (int i = 0; i < sonuckumesi[ebs] - 1; i++)
    {
        Random rnd = new Random();
        int c1 = rnd.Next(istasyongorevsayisi[ebs, i]);
        int c2 = rnd.Next(istasyongorevsayisi[ebs, i + 1]);
        lsmipop[i, kacinciislist[i] + c1] = ebeveynpop[ebs, kacinciislist[i]
+ c2];
        lsmipop[i, kacinciislist[i + 1] + c2] = ebeveynpop[ebs, kacinciislist[i]
+ c1];
    }
    //lsm hesaplma işlemi

```

```

int[] localissirasilm = new int[n];
int[,] localsiralsm = new int[n, 3];
double[,] localtjrlsm = new double[n, n];
double[,] localcjrslm = new double[n, n];
int localsollsm = M;
for (int b = 0; b < lsmpop.GetLength(0); b++)
{
    int[,] siralsm = new int[n, 3];
    int[] ctlistlsm = new int[n];
    int[,] srlsm = new int[n, n];
    int aktifistlsm = 0;
    double[] ctaktiflsm = new double[n];
    double[] ctaktifyapaylsm = new double[n];
    int aktifsiranolsm = 0;
    int[] rassalsiralsm = new int[n];
    double[,] tjrlsm = new double[n, n];
    double[,] cjrslm = new double[n, n];
    for (int b1 = 0; b1 < n; b1++)
    {
        rassalsiralsm[b1] = lsmpop[b, b1];
    }
    for (int i = 0; i < n; i++)
    {
        int islemdekiis = rassalsiralsm[i];
        if (aktifsiranolsm == 0)
        {
            tjrlsm[aktifistlsm, aktifsiranolsm] = processtime[islemdekiis -
1];

            cjrslm[aktifistlsm, aktifsiranolsm] = tjrlsm[aktifistlsm,
aktifsiranolsm];

        }
        if (aktifsiranolsm > 0)
        {

```

```

double suretoplamlam = 0;
for (int sirasay = 0; sirasay < aktifsiranolsm; sirasay++)
{
    suretoplamlam = tjrlsm[aktifistlsm, sirasay] + suretoplamlam;
}

//tjrlsm[aktifistlsm, aktifsiranolsm] = (processtime[islemdekiis - 1] *
Math.Pow(aktifsiranolsm + 1, lr)); //pozisyon tabanlı öğrenme etkisi altında
//tjrlsm[aktifistlsm, aktifsiranolsm] = processtime[islemdekiis - 1] * Math.Pow((1 +
Math.Log(suretoplamlam)), lr); //logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi
altında
//tjrlsm[aktifistlsm, aktifsiranolsm] = (processtime[islemdekiis - 1] + cjrlsm[aktifistlsm,
aktifsiranolsm - 1] * dr); //bozulma etkisi altında
tjrlsm[aktifistlsm, aktifsiranolsm] = (processtime[islemdekiis - 1] *
Math.Pow(aktifsiranolsm + 1, lr)) + (cjrlsm[aktifistlsm, aktifsiranolsm - 1] *
Math.Pow(aktifsiranolsm + 1, lr) * dr); //pozisyon tabanlı öğrenme ve bozulma etkisi
altında
//tjrlsm[aktifistlsm, aktifsiranolsm] = (processtime[islemdekiis - 1] + cjrlsm[aktifistlsm,
aktifsiranolsm - 1] * dr) * Math.Pow((1 + Math.Log(suretoplamlam)), lr); //logaritmik işlem
süreleri toplamı tabanlı öğrenme e bozulma etkisi altında
cjrlsm[aktifistlsm, aktifsiranolsm] = cjrlsm[aktifistlsm, aktifsiranolsm - 1] +
tjrlsm[aktifistlsm, aktifsiranolsm];
}
ctaktifyapaylsm[aktifistlsm] = cjrlsm[aktifistlsm, aktifsiranolsm];
if (ctaktifyapaylsm[aktifistlsm] <= ct)
{
    int bulundu = 0;
    if (aktifsiranolsm > 0)
    {
        int a1 = islemdekiis;
        for (int j = 0; j < aktifsiranolsm; j++)
        {
            int a2 = srlsm[aktifistlsm, j];
            for (int z = 0; z < precedence.GetLength(0); z++)

```

```

{

if (precedence[z, 1] == a2 && precedence[z, 0] == a1)
{
    bulundu += 1;
}
}

}

if (bulundu == 0)
{
    siralsm[i, 0] = rassalsiralsm[i];
    siralsm[i, 1] = aktifistlsm;
    siralsm[i, 2] = aktifsiranolsm;
    srlsm[aktifistlsm, aktifsiranolsm] = islemdekiis;
    ctaktiflsm[aktifistlsm] = ctaktifyapaylsm[aktifistlsm];
    aktifsiranolsm += 1;
}

if (bulundu > 0)
{
    tjrlsm[aktifistlsm, aktifsiranolsm] = 0;
    cjrlsm[aktifistlsm, aktifsiranolsm] = 0;
    aktifsiranolsm = 0;
    aktifistlsm += 1;
    siralsm[i, 0] = rassalsiralsm[i];
    siralsm[i, 1] = aktifistlsm;
    siralsm[i, 2] = aktifsiranolsm;
    srlsm[aktifistlsm, aktifsiranolsm] = islemdekiis;
    ctaktiflsm[aktifistlsm] = processtime[islemdekiis - 1];
    ctaktifyapaylsm[aktifistlsm] = processtime[islemdekiis - 1];
    tjrlsm[aktifistlsm, aktifsiranolsm] = processtime[islemdekiis - 1];
    cjrlsm[aktifistlsm, aktifsiranolsm] = tjrlsm[aktifistlsm,
aktifsiranolsm];

```

```

        aktifsiranolsm += 1;
    }
}
if (ctaktifyapaylsm[aktifistlsm] > ct)
{
    tjrlsm[aktifistlsm, aktifsiranolsm] = 0;
    cjrlsm[aktifistlsm, aktifsiranolsm] = 0;
    aktifsiranolsm = 0;
    aktifistlsm += 1;
    siralsm[i, 0] = rassalsiralsm[i];
    siralsm[i, 1] = aktifistlsm;
    siralsm[i, 2] = aktifsiranolsm;
    srlsm[aktifistlsm, aktifsiranolsm] = islemdekiis;
    ctaktiflsm[aktifistlsm] = processtime[islemdekiis - 1];
    ctaktifyapaylsm[aktifistlsm] = processtime[islemdekiis - 1];
    tjrlsm[aktifistlsm, aktifsiranolsm] = processtime[islemdekiis - 1];
    cjrlsm[aktifistlsm, aktifsiranolsm] = tjrlsm[aktifistlsm, aktifsiranolsm];
    aktifsiranolsm += 1;
}
}
if (aktifistlsm + 1 < localsollsm)
{
    localsollsm = aktifistlsm + 1;
    localsiralsm = siralsm;
    localissirasilsm = rassalsiralsm;
    localcjrlsm = cjrlsm;
    localtjrlsm = tjrlsm;
}
}
if (localsollsm < globalsol)
{
    globalsol = localsollsm;
    globalsira = localsiralsm;

```

```

        globalissirasi = localissirasilsm;
        globalcjr = localcjrslm;
        globaltjr = localtjrslm;
    }
}

////////CROSSOVER PART////////
///fragment reordering method
int[,] cobinary = new int[ebeveynpopsize / 2, n];
double a = n * corate; //max iş sayısının yarısı kadar istasyon açık olsun
double z1 = Math.Round(a); //sayıyı alt tamsayıya yuvarla
int coissayisi = Convert.ToInt32(z1); //sayıyı integera dönüştürme
int cutsayisi = n - coissayisi - 1;
int[,] cutpoint = new int[cutsayisi, 2];
for (int i = 0; i < cutsayisi; i++)
{
    cutpoint[i, 0] = i + 1;
    cutpoint[i, 1] = cutpoint[i, 0] + coissayisi;
}
int[,] copopeslestirme = new int[ebeveynpopsize / 2, 2];
List<int> ebeveynpopindex = new List<int>();
for (int i = 0; i < ebeveynpopsize; i++)
{
    ebeveynpopindex.Add(i);
}
while (ebeveynpopindex.Count > 0)
{
    for (int i = 0; i < ebeveynpopsize / 2; i++)
    {

        Random rnd = new Random();
        int sayi1 = rnd.Next(0, ebeveynpopindex.Count);
        copopeslestirme[i, 0] = ebeveynpopindex[sayi1]; ;
        ebeveynpopindex.RemoveAt(sayi1);
    }
}

```

```

        int sayi2 = rnd.Next(0, ebeveynpopindex.Count);
        copopeslestirme[i, 1] = ebeveynpopindex[sayi2];
        ebeveynpopindex.RemoveAt(sayi2);
    }
}

int[] selectedcutpoint = new int[ebeveynpopsize / 2];
for (int i = 0; i < ebeveynpopsize / 2; i++)
{
    Random rnd = new Random();
    int sayi = rnd.Next(0, cutpoint.GetLength(0));
    selectedcutpoint[i] = sayi;
}

//crossover işlemi
int d1; //degisenilk görev
int d2; //degisen ikinci görev
int[,] copop = new int[ebeveynpopsize, n];
for (int i = 0; i < ebeveynpopsize / 2; i++)
{
    //int[] cop2 = new int[n];
    d1 = copopeslestirme[i, 0];
    d2 = copopeslestirme[i, 1];
    List<int> cop1 = new List<int>();
    List<int> cop2 = new List<int>();
    for (int p = 0; p < n; p++)
    {
        cop1.Add(ebeveynpop[d1, p]);
        cop2.Add(ebeveynpop[d2, p]);
    }

    for (int j1 = 0; j1 < cutpoint[selectedcutpoint[i], 0]; j1++)
    {
        copop[d1, j1] = ebeveynpop[d1, j1];
    }
}

```

```

        copop[d2, j1] = ebeveynpop[d2, j1];
        cop2.Remove(ebeveynpop[d1, j1]);
        cop1.Remove(ebeveynpop[d2, j1]);
    }
    for (int j2 = cutpoint[selectedcutpoint[i], 1]; j2 < n; j2++)
    {
        copop[d1, j2] = ebeveynpop[d1, j2];
        copop[d2, j2] = ebeveynpop[d2, j2];
        cop2.Remove(ebeveynpop[d1, j2]);
        cop1.Remove(ebeveynpop[d2, j2]);
    }

    for (int k = cutpoint[selectedcutpoint[i], 0]; k <
cutpoint[selectedcutpoint[i], 1]; k++)
    {
        copop[d1, k] = cop2[k - cutpoint[selectedcutpoint[i], 0]];
        copop[d2, k] = cop1[k - cutpoint[selectedcutpoint[i], 0]];
    }
}

int[] localissirasico = new int[n];
int[,] localsiraco = new int[n, 3];
double[,] localtjrco = new double[n, n];
double[,] localcjrco = new double[n, n];
int localsolco = M;

for (int b = 0; b < copop.GetLength(0); b++)
{
    int[,] siraco = new int[n, 3];
    int[] ctlistco = new int[n];
    int[,] srco = new int[n, n];
    int aktifistco = 0;

```

```

double[] ctaktifco = new double[n];
double[] ctaktifyapayco = new double[n];
int aktifsiranoco = 0;
int[] rassalsiraco = new int[n];
double[,] tjrco = new double[n, n];
double[,] cjrco = new double[n, n];
for (int b1 = 0; b1 < n; b1++)
{
    rassalsiraco[b1] = copop[b, b1];
}
for (int i = 0; i < n; i++)
{
    int islemdekiis = rassalsiraco[i];
    if (aktifsiranoco == 0)
    {
        tjrco[aktifistco, aktifsiranoco] = processtime[islemdekiis - 1];
        cjrco[aktifistco, aktifsiranoco] = tjrco[aktifistco, aktifsiranoco];
    }
    if (aktifsiranoco > 0)
    {
        double suretoplamlam = 0;
        for (int sirasay = 0; sirasay < aktifsiranoco; sirasay++)
        {
            //suretoplamlam = Math.Log(processtime[srco[aktifistco, sirasay] -
1]) + suretoplamlam;
            suretoplamlam = tjrco[aktifistco, sirasay] + suretoplamlam;
        }
        // tjrco[aktifistco, aktifsiranoco] = (processtime[islemdekiis - 1] *
Math.Pow(aktifsiranoco + 1, lr)); //pozisyon tabanlı öğrenme etkisi altında
//tjrco[aktifistco, aktifsiranoco] = processtime[islemdekiis - 1] * Math.Pow((1 +
Math.Log(suretoplamlam)), lr); //logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi
altında

```

```

//tjrco[aktifistco, aktifsiranoco] = (processtime[islemdekiis - 1] + cjrco[aktifistco,
aktifsiranoco - 1] * dr); //bozulma etkisi altında
tjrco[aktifistco, aktifsiranoco] = (processtime[islemdekiis - 1] * Math.Pow(aktifsiranoco
+ 1, lr)) + (cjrco[aktifistco, aktifsiranoco - 1] * Math.Pow(aktifsiranoco + 1, lr) * dr);
//pozisyon tabanlı öğrenme ve bozulma etkisi altında
//tjrco[aktifistco, aktifsiranoco] = (processtime[islemdekiis - 1] + cjrco[aktifistco,
aktifsiranoco - 1] * dr) * Math.Pow((1 + Math.Log(suretoplaml)), lr); //logaritmik işlem
süreleri toplamı tabanlı öğrenme e bozulma etkisi altında
cjrco[aktifistco, aktifsiranoco] = cjrco[aktifistco, aktifsiranoco - 1] + tjrco[aktifistco,
aktifsiranoco];

    }

//ctaktifyapay[aktifistco] = ctaktifyapay[aktifistco] + processtime[islemdekiis-1];
ctaktifyapayco[aktifistco] = cjrco[aktifistco, aktifsiranoco];
if (ctaktifyapayco[aktifistco] <= ct)
{
    int bulundu = 0;
    if (aktifsiranoco > 0)
    {
        int a1 = islemdekiis;
        for (int j = 0; j < aktifsiranoco; j++)
        {
            int a2 = srco[aktifistco, j];
            for (int z = 0; z < precedence.GetLength(0); z++)
            {
                if (precedence[z, 1] == a2 && precedence[z, 0] == a1) // 8
v 25 verili için
            {
                bulundu += 1;
            }
        }
    }
}
}

```

```

if (bulundu == 0)
{
    siraco[i, 0] = rassalsiraco[i];
    siraco[i, 1] = aktifistco;
    siraco[i, 2] = aktifsiranoco;
    srco[aktifistco, aktifsiranoco] = islemdekiis;
    ctaktifco[aktifistco] = ctaktifyapayco[aktifistco];
    aktifsiranoco += 1;
}
if (bulundu > 0)
{
    tjrco[aktifistco, aktifsiranoco] = 0;
    cjrco[aktifistco, aktifsiranoco] = 0;
    aktifsiranoco = 0;
    aktifistco += 1;
    siraco[i, 0] = rassalsiraco[i];
    siraco[i, 1] = aktifistco;
    siraco[i, 2] = aktifsiranoco;
    srco[aktifistco, aktifsiranoco] = islemdekiis;
    ctaktifco[aktifistco] = processtime[islemdekiis - 1];
    ctaktifyapayco[aktifistco] = processtime[islemdekiis - 1];
    tjrco[aktifistco, aktifsiranoco] = processtime[islemdekiis - 1];
    cjrco[aktifistco, aktifsiranoco] = tjrco[aktifistco, aktifsiranoco];
    aktifsiranoco += 1;
}
}
if (ctaktifyapayco[aktifistco] > ct)
{
    tjrco[aktifistco, aktifsiranoco] = 0;
    cjrco[aktifistco, aktifsiranoco] = 0;
    aktifsiranoco = 0;
    aktifistco += 1;
    siraco[i, 0] = rassalsiraco[i];

```

```

siraco[i, 1] = aktifistco;
siraco[i, 2] = aktifsiranoco;
srco[aktifistco, aktifsiranoco] = islemdekiis;
ctaktifco[aktifistco] = processtime[islemdekiis - 1];
ctaktifyapayco[aktifistco] = processtime[islemdekiis - 1];
tjrco[aktifistco, aktifsiranoco] = processtime[islemdekiis - 1];
cjrco[aktifistco, aktifsiranoco] = tjrco[aktifistco, aktifsiranoco];
aktifsiranoco += 1;
}

}

//crossover hesaplama bitiř
if (aktifistco + 1 < localsolco)
{
    localsolco = aktifistco + 1;
    localsiraco = siraco;
    localissirasico = rassalsiraco;
    localcjrco = cjrco;
    localtjrco = tjrco;
}
}
if (localsolco < globalsol)
{
    globalsol = localsolco;
    globalsira = localsiraco;
    globalissirasi = localissirasico;
    globalcjr = localcjrco;
    globaltjr = localtjrco;
}

////MUTATION PART/////
/// OPRO (one point right operator)
int[, ] mutbinary = new int[ebeveynpopsize, n];

```

```

int[,] mutpop = new int[ebeveynpopsize, n];
int[] cutpoints = new int[ebeveynpop.GetLength(0)];
double m = n * mutrate; //max iş sayısının yarısı kadar istasyon açık olsun
double m1 = Math.Round(m); //sayıyı alt tamsayıya yuvarla
int mutissayisi = Convert.ToInt32(m1); //sayıyı integera dönüştürme
int[,] mutparts = new int[ebeveynpopsize, mutissayisi];
int cutpointkarar = n - mutissayisi + 1;
int[] selectedcutpointmut = new int[ebeveynpopsize];
for (int i = 0; i < ebeveynpopsize; i++)
{
    Random rnd = new Random();
    int sayi = rnd.Next(0, cutpointkarar);
    selectedcutpointmut[i] = sayi;
}
int[] mp = new int[mutissayisi];
for (int i = 0; i < ebeveynpopsize; i++)
{
    List<int> mplist = new List<int>(mutissayisi);
    for (int k = 0; k < mutissayisi; k++)
    {
        mp[k] = ebeveynpop[i, selectedcutpointmut[i] + k];
        mplist.Add(mp[k]);
    }
    int[] mutrassal = new int[mutissayisi];
    for (int i1 = 0; i1 < mp.Length; i1++)
    {
        Random rnd = new Random();
        int sayi = rnd.Next(0, mplist.Count);
        mutrassal[i1] = mplist[sayi];
        mplist.RemoveAt(sayi);
    }
    for (int j = 0; j < n; j++)
    {

```

```

        mutpop[i, j] = ebeveynpop[i, j];
    }
    for (int i2 = 0; i2 < mutrassal.Length; i2++)
    {
        mutpop[i, selectedcutpointmut[i] + i2] = mutrassal[i2];
    }
}

int[] localissirasimut = new int[n];
int[,] localsiramut = new int[n, 3];
double[,] localtjrmut = new double[n, n];
double[,] localcjrmut = new double[n, n];
int localsolmut = M;
for (int b = 0; b < mutpop.GetLength(0); b++)
{
    int[,] siramut = new int[n, 3];
    int[] ctlistmut = new int[n];
    int[,] srmut = new int[n, n];
    int aktifistmut = 0;
    double[] ctaktifmut = new double[n];
    double[] ctaktifyapaymut = new double[n];
    int aktifsiranomut = 0;
    int[] rassalsiramut = new int[n];
    double[,] tjrmut = new double[n, n];
    double[,] cjrmut = new double[n, n];
    for (int b1 = 0; b1 < n; b1++)
    {
        rassalsiramut[b1] = mutpop[b, b1];
    }
    for (int i = 0; i < n; i++)
    {
        int islemdekiis = rassalsiramut[i];
        if (aktifsiranomut == 0)
        {
            tjrmut[aktifistmut, aktifsiranomut] = processtime[islemdekiis - 1];

```

```

        cjrmut[aktifistmut,    aktifsiranomut]    =    tjrmut[aktifistmut,
aktifirsiranomut];

    }
    if (aktifsiranomut > 0)
    {
        double suretoplamlam = 0;
        for (int sirasay = 0; sirasay < aktifsiranomut; sirasay++)
        {
            suretoplamlam = tjrmut[aktifistmut, sirasay] + suretoplamlam;
        }

        //tjrmut[aktifistmut,    aktifsiranomut]    =    (processtime[islemdekiis    -    1]    *
Math.Pow(aktifsiranomut + 1, lr));    //pozisyon tabanlı öğrenme etkisi altında
        //tjrmut[aktifistmut, aktifsiranomut] = processtime[islemdekiis - 1] * Math.Pow((1 +
Math.Log(suretoplamlam)), lr);    //logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi
altında
        //tjrmut[aktifistmut,    aktifsiranomut]    =    (processtime[islemdekiis    -    1]    +
cjrmut[aktifistmut, aktifsiranomut - 1] * dr); //bozulma etkisi altında
        tjrmut[aktifistmut,    aktifsiranomut]    =    (processtime[islemdekiis    -    1]    *
Math.Pow(aktifsiranomut + 1, lr)) + (cjrmut[aktifistmut, aktifsiranomut - 1] *
Math.Pow(aktifsiranomut + 1, lr) * dr); //pozisyon tabanlı öğrenme ve bozulma etkisi
altında
        //tjrmut[aktifistmut,    aktifsiranomut]    =    (processtime[islemdekiis    -    1]    +
cjrmut[aktifistmut, aktifsiranomut - 1] * dr) * Math.Pow((1 + Math.Log(suretoplamlam)), lr);
//logaritmik işlem süreleri toplamı tabanlı öğrenme e bozulma etkisi altında
        cjrmut[aktifistmut, aktifsiranomut] = cjrmut[aktifistmut, aktifsiranomut - 1] +
tjrmut[aktifistmut, aktifsiranomut];

    }
    ctaktifyapaymut[aktifistmut] = cjrmut[aktifistmut, aktifsiranomut];
    if (ctaktifyapaymut[aktifistmut] <= ct)
    {
        int bulundu = 0;
        if (aktifsiranomut > 0)
        {

```

```

int a1 = islemdekiis;
for (int j = 0; j < aktifsiranomut; j++)
{
    int a2 = srmut[aktifistmut, j];
    for (int z = 0; z < precedence.GetLength(0); z++)
    {
        if (precedence[z, 1] == a2 && precedence[z, 0] == a1) // 8
            v 25 verili için
            {
                bulundu += 1;
            }
    }
}
if (bulundu == 0)
{
    siramut[i, 0] = rassalsiramut[i];
    siramut[i, 1] = aktifistmut;
    siramut[i, 2] = aktifsiranomut;
    srmut[aktifistmut, aktifsiranomut] = islemdekiis;
    ctaktifmut[aktifistmut] = ctaktifyapaymut[aktifistmut];
    aktifsiranomut += 1;
}
if (bulundu > 0)
{
    tjrmut[aktifistmut, aktifsiranomut] = 0;
    cjrmut[aktifistmut, aktifsiranomut] = 0;
    aktifsiranomut = 0;
    aktifistmut += 1;
    siramut[i, 0] = rassalsiramut[i];
    siramut[i, 1] = aktifistmut;
    siramut[i, 2] = aktifsiranomut;
    srmut[aktifistmut, aktifsiranomut] = islemdekiis;
}

```

```

ctaktifmut[aktifistmut] = processtime[islemdekiis - 1];
ctaktifyapaymut[aktifistmut] = processtime[islemdekiis - 1];
tjrmut[aktifistmut, aktifsiranomut] = processtime[islemdekiis -
1];

cjrmut[aktifistmut, aktifsiranomut] = tjrmut[aktifistmut,
aktifsiranomut];

aktifsiranomut += 1;
}
}
if (ctaktifyapaymut[aktifistmut] > ct)
{
tjrmut[aktifistmut, aktifsiranomut] = 0;
cjrmut[aktifistmut, aktifsiranomut] = 0;
aktifsiranomut = 0;
aktifistmut += 1;
siramut[i, 0] = rassalsiramut[i];
siramut[i, 1] = aktifistmut;
siramut[i, 2] = aktifsiranomut;
srmut[aktifistmut, aktifsiranomut] = islemdekiis;
ctaktifmut[aktifistmut] = processtime[islemdekiis - 1];
ctaktifyapaymut[aktifistmut] = processtime[islemdekiis - 1];
tjrmut[aktifistmut, aktifsiranomut] = processtime[islemdekiis - 1];
cjrmut[aktifistmut, aktifsiranomut] = tjrmut[aktifistmut,
aktifsiranomut];

aktifsiranomut += 1;
}
}
if (aktifistmut + 1 < localsolmut)
{
localsolmut = aktifistmut + 1;
localsiramut = siramut;
localissirasimut = rassalsiramut;
localcjrmut = cjrmut;

```

```
        localtjrmut = tjrmut;
    }
}
if (localsolmut < globalsol)
{
    globalsol = localsolmut;
    globalsira = localsiramut;
    globalissirasi = localissirasimut;
    globalcjr = localcjrmut;
    globaltjr = localtjrmut;
}
}
}
}
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı: Halime SÖMÜRK

Uyruğu: Türkiye (T.C)

Doğum Tarihi ve Yeri:

Medeni Durum:

e-mail:

Yazışma Adresi:

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Yüksek Lisans	Erciyes Üniversitesi-Endüstri Mühendisliği	2018
Lisans	Gaziantep Üniversitesi- Endüstri Mühendisliği	2013

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2014-Halen	Gaziosmanpaşa Üniversitesi	9 Yıl
2013-2014	Adoçim Çimento A.Ş.	1 Yıl

YABANCI DİL

İngilizce