



Logaritmik İşlem Süreleri Toplamı Tabanlı Öğrenme ve İş Bozulması Etkileri Altında Demontaj Hattı Problemi

Disassembly Line Balancing Problem Under The Effects of Sum-of-Logarithm-Processing-Time-Based Learning and Job Deterioration

Halime Somtürk^{1*}, Mehmet Duran Toksarı²

¹ Tokat Gaziosmanpaşa Üniversitesi, Endüstri Mühendisliği Bölümü, halime.somturk@gop.edu.tr
ORCID: <https://orcid.org/0000-0001-7329-495X>

² Erciyes Üniversitesi, Endüstri Mühendisliği Bölümü, dtoksari@erciyes.edu.tr
ORCID: <https://orcid.org/0000-0001-9577-1956>

MAKALE BİLGİLERİ

Makale Geçmişi:

Geliş 25 Eylül 2023
Revizyon 11 Aralık 2023
Kabul 12 Aralık 2023
Online 31 Aralık 2023

Anahtar Kelimeler:

*Demontaj hattı dengeleme (DHD),
Logaritmik işlem süreleri toplamı
tabanlı öğrenme etkisi,
İş bozulması etkisi,
Hibrit PSO-GA yaklaşımı*

ÖZ

Artan çevresel farkındalık, ekonomik kaygılar ve yasal kurallar sayesinde, son zamanlarda ürün geri kazanımı veya yeniden üretimi yoğun ilgi görmektedir. Demontaj, ömrünü tamamlamış ürünlerin geri kazanımı aşamasındaki en önemli süreçlerden biridir. Bu yüzden, etkin ve dengeli kurulmuş demontaj hatları önem taşımaktadır. Bu çalışmada, demontaj hattı dengeleme (DHD) problemleri, eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında incelenmiştir. DHD literatüründe, logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması eş zamanlı olarak ilk kez çalışılmıştır. Herhangi bir iş, istasyondaki sırasına göre, kendisinden önceki işlerin işlem sürelerinin logaritmik toplamından etkilenmektedir. İş bozulması ise, işin işleme başlamasını geciktiren bozulmalardır. Öğrenme işlerin işlem sürelerinin azaltırken, bozulma artırmaktadır. Bu çalışmada amaç fonksiyonu, açılan istasyon sayısı minimizasyonudur. Öğrenme ve bozulma etkisi altındaki DHD probleminin çözümü için hibrit (Parçacık Sürüsü Optimizasyonu- Genetik Algoritma) PSO-GA algoritması geliştirilmiştir. Farklı öğrenme ve bozulma oranları için sonuçlar elde edilip karşılaştırmalar yapılmıştır. Öğrenme ve bozulma etkisinin, DHD problemlerinde dikkate alındığında amaç fonksiyonu değerinde iyileşmeler görülmüştür.

ARTICLE INFO

Article history:

Received 25 September 2023
Received in revised form 11 December 2023
Accepted 12 December 2023
Available online 31 December 2023

Keywords:

*Disassembly line balancing (DLB),
Sum-of-logarithm-processing-time-
based learning effect,
Job deterioration effect,
Hybrid PSO-GA approach*

ABSTRACT

Product recovery or remanufacturing has recently received a lot of attention as a result of rising environmental consciousness, economic worries, and laws and regulations. Disassembly is one of the most important processes in the recovery phase of end-of-life products. It's crucial to develop efficient and balanced disassembly lines. In this study, disassembly line balancing (DLB) problems are investigated under the effects of simultaneous sum-of-logarithm-processing-time-based learning and job deterioration. In the DLB literature, it is the first time to study simultaneous sum-of-logarithm-processing-time-based learning and job deterioration. Any job's processing time is influenced by the logarithmic sum of the processing times of its predecessors, in the station's order. Job deterioration delays the job start time. While learning reduces the processing time of jobs, deterioration increases. In this study, the objective function is the minimization of the number of opened stations. A hybrid PSO-GA algorithm has been developed to solve the DLB problem under the effects of learning and deterioration. Results for different learning and deterioration rates were obtained and comparisons were made. When the learning and deterioration effects were taken into account in DLB problems, improvements were observed in the objective function value.

Doi: 10.24012/dumf.1366117

* Sorumlu Yazar

Giriş

Doğal kaynakların azalması, depolama maliyetlerinin artması, toplumun zamanla daha bilinçlenmesi ve çevre ile ilgili yasa ve yönetmeliklerin etkisi ile üreticiler, tüketiciden çıkan tekrar kullanılabilir olan ürünlerin geri kazanımını sağlamaya yönelmiştir [1], [2]. Son yıllarda teknolojinin hızla gelişmesi ve yaşam döngüsünün kısalması ile birlikte ömrünü tamamlamış ürünlerin sayısı artmaktadır [3], [4]. Son kullanma tarihi geçmiş veya ömrü tükenmiş bu ürünlerin yeniden kullanılması veya yeniden üretilmesi ekonomik fayda sağlamakta ve aynı zamanda bu ürünlerden kaynaklanan çevre kirliliğini azaltmaktadır [3].

Demontaj, ürün geri kazanımında ürünlerin sistematik olarak alt bileşenlere ayrılmasını sağlayan en önemli ve zaman alıcı adımdır. Demontaj işlemi genellikle iş istasyonlarından oluşan bir hat üzerinde gerçekleştirilir. Demontaj hatları, ömrünü tamamlamış ürünlerin geri dönüştürülmesi veya yeniden üretilmesi için fırsatlar sunmaktadır [5]. Kaynak kullanımını optimize etmek için bir demontaj hattı dengeli olmalıdır. Demontaj hattı dengeleme, verimli ve etkili ürün geri dönüşümü sağlamanın en uygun yoludur. Demontaj hattı dengeleme problemleri, görevler arasındaki öncelik ilişkilerini sağlamak, çevrim süresini aşmamak gibi birkaç optimizasyon kısıtını karşılayacak şekilde belirli demontaj görevlerinin istasyonlara atanmasına izin verir. Bu nedenle, verimli ve dengeli bir şekilde tasarlanmış demontaj hattı, çevresel ve endüstriyel açıdan öneme sahiptir [6].

Demontaj işlemi, üründen ayrılacak parçalara göre kısmi demontaj ve komple demontaj olmak üzere iki şekilde olabilir. Kısmi demontajda ürün tamamen demonte olmaz. Tekrar kullanılabilir durumda olan veya ihtiyaç duyulan parçalar sökülür. Komple demontajda ise ürünün tüm parçaları ayrılır [7]. Farklı endüstri alanlarına göre düz, paralel ve U şeklinde demontaj hatları bulunmaktadır [3].

DHD problemini çözerken, görevler arasındaki öncelik ilişkileri ve çevrim süresi kısıtları dikkate alınmalıdır. Öncelik ilişkileri DHD problemini daha karmaşık hale getirir. Çözülmesi ve doğrulanması zor olan problemler, NP zor problem olarak tanımlanmaktadır. Dolayısıyla, DHD problemleri, kısıtlar dahilinde karmaşıklaştığı için optimum sonuca ulaşılması zorlaşmaktadır ve NP zor problemlerdir [8].

DHD problemi ilk olarak Güngör ve Gupta [9] tarafından sistematik yaklaşıma yönelik sezgisel bir yöntem kullanılarak çalışılmıştır. Güngör ve Gupta [10], tehlikeli ürünlerin öncelikli olarak sökülmesine izin veren minimum istasyon sayısını hedefleyen sezgisel bir yaklaşım önermiştir.

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 [11], farklı durumlarda DHD problemi için basit bir sezgisel yöntem geliştirmiştir. McGovern ve Gupta [12], çok amaçlı DHD problemi için bir ağırlıklı/2 optimal(2-opt) hibrit algoritma önermişlerdir. Ren ve ark. [13], bu yöntemi ağırlıklara dayalı çok kriterli kararlarla genişletmiştir. Mete ve ark. [14], istasyon sayısını en aza indirmek için ışın aramayı önermiştir. İkinci kategori metasezgisel yöntemlerdir. Genetik algoritmalar[8], [15], [16], karınca kolonisi optimizasyonu [17], [18], yapay arı kolonisi algoritması [19], [20], tabu arama algoritması [21],

parçacık sürü optimizasyonu [22], [23], yerçekimsel arama algoritması [24], yapay balık sürüsü algoritması [25], değişken komşuluk arama algoritması [24], ateş böceği algoritması [26] ve ayrı guguk kuşu arama algoritması [3]. Üçüncü yöntem olarak, DHD problemlerini optimal olarak çözmek için matematiksel programlama teknikleri kullanılmaktadır. Altekin ve ark. [27], kar odaklı kısmi DHD problemi için bir karma tamsayı programlama formülasyonu geliştirmiştir. Altekin [28], stokastik DHD'ye parçalı doğrusal programlama uygulanmasını önermiştir. Kaynak kısıtlı DHD sorunu için bir matematiksel model sunulmuştur [29]. Ilgın ve ark. [30], DHD hattı için doğrusal fiziksel programlama, Paksoy ve ark. [31] bulanık hedefler kullanan karma bir model önermiştir. Özceylan ve Paksoy [32], doğrusal olmayan karma tamsayı programlama modeli kullanarak çok amaçlı DHD sorununu çözmüştür. DHD problemleriyle ilgili detaylı literatür çalışması Özceylan ve ark. [33] ve Laili ve ark. [34] tarafından yapılmıştır.

Klasik DHD problemlerinde işlerin işlem sürelerinin sabit olduğu varsayılır. Ancak gerçek hayattaki uygulamalarda işler çeşitli etkiler altında olabilir ve işlerin işlem süreleri değişkenlik gösterebilir. Gerçek hayatta, çalışanlar yaptıkları işleri tekrarladıkları takdirde iş süreçlerini öğrenir ve daha verimli çalışabilirler. İşçilerin beceri ve deneyimlerinin artmasıyla birlikte, işlerin işlem süreleri de giderek azalacaktır [35]. Bu olgu literatürde öğrenme etkisi olarak bilinir.

Wright [36], üretim operasyonlarına dayalı öğrenme kavramını düşünen ilk kişidir. Biskup [37] ve Cheng ve Wang [38], öğrenme etkisini çizelgeleme alanında çalışan öncülerdir. Biskup, pozisyona dayalı bir öğrenme modeli önermiştir ve bunu tek makineyi çizelgeleme problemine uygulamıştır. Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi çizelgeleme problemlerinde nadiren kullanılmıştır [39]–[41].

Öğrenme etkisi ve iş kaybı literatürde derinlemesine incelenmiştir. Gerçek hayatta, öğrenme etkisi ve iş kaybı aynı anda meydana gelir. Örneğin, üretim ortamı giderek daha rekabetçi hale geldikçe, işletmeler müşterilere daha fazla ürün çeşitliliği sağlamak için daha kısa üretim sürelerine ve daha sık ürün değişikliklerine yönelmektedir. Bu tür iş ortamlarında çalışanların yaşadıkları öğrenmeler ve unutmalar çok önemlidir. Unutma olarak adlandırılan, düşük performansla neden olan ürün ve süreç değişiklikleri, çalışanları etkiler [42].

Bir iş, makinede işlenirken veya işlenmeyi beklerken bozulabilir. Bu nedenle, işe başlama saatine bağlı olarak bozulma etkisi altında işlem süresi artmaktadır. Gupta [43], bozulma etkisini çizelgeleme problemine ilk olarak eklemiştir. Haddehaneye girmeyi beklerken, külçe sıcaklığının düşmesi durumunda külçenin tekrar ısıtılması gerekecektir. Yangınla mücadelede gecikme olursa yangını söndürmek için gereken süre artacaktır [44], [45]. Kötüleştiren koşullar altında diğer faktörler şunlardır: Kötüleştiren hava veya artan karanlık, makineleri veya araçları onarmak, vb. Tüm bu durumlarda, gecikmeler genellikle işlem sürelerini uzatır [55].

Klasik DHD problemlerinde işlerin işlem süreleri süreç boyunca sabit kabul edilmektedir. Bu çalışmada, DHD problemlerine logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile iş bozulması etkisi eş zamanlı uygulanmıştır ve belirli bir döngü süresi için iş istasyonu

sayılarının en aza indirilmesi amaçlanmaktadır. Öğrenme ve bozulma etkisinin tüm görevler için eşit olduğu varsayılmaktadır.

Bu makalenin katkısını şu şekilde özetleyebiliriz:

•DHD problemi ilk defa, eş zamanlı olarak logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile iş bozulması etkisi düşünülerek çözdürülmüştür. Klasik DHD problemlerinde işlerin işlem süreleri, süreç boyunca sabittir. Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ile sıradaki işin işlem süresi, kendisinden önceki sıralarda yer alan işlerin işlem sürelerine bağlıdır. Aynı istasyonda, önceki sıralara atanan işlerin işlem sürelerinin logaritmik toplamının bir fonksiyonu olarak hesaplanmaktadır. Ayrıca, iş bozulması işler normalden daha uzun sürede tamamlanabilmektedir. Bu çalışmada her iki etkinin de düşünüldüğü DHD problemlerinin işlem sürelerindeki değişiklik sonucunda açılacak istasyon sayısındaki değişim incelenmiştir.

• Logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında DHD problemi için matematiksel model geliştirilmiştir. Problem NP-zor yapıda olduğu için Lingo.11 optimizasyon programı ile çok küçük boyutlu problem için çözüm aranmıştır. Daha büyük boyutlu problemlerin çözümünde ise hibrit PSO-GA algoritması kullanılmıştır. Önerilen hibrit metasezgisel yöntemle optimal sonuca veya optimal sonuca en yakın sonuca ulaşmak hedeflenmiştir.

Materyal ve Yöntem

Problem tanımı ve formülasyonu

DHD problemi, talebi karşılarken demontaj hattındaki kaynakları en verimli şekilde kullanmayı amaçlar. Demontaj hattı yerleşiminin iyileştirilmesi, kaynakların verimli kullanılması, demontaj görevlerinin öncelik ilişkileri dikkate alınarak iş istasyonlarına en uygun şekilde atanması ve gerekli minimum iş istasyonu sayısının bulunmasıdır (Güngör ve Gupta, 2002). Bu çalışmada, DHD probleminin modellenmesinde logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri dikkate alınmıştır.

Logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri modelinde, p_{ir} , r. sırada yer alan i işinin güncel işlem süresidir. Bu durumda;

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

burada $a < 0$ öğrenme oranı ($1/r$) ve p_i , i işinin gerçek işlem süresidir. Öğrenme etkisi indeks değeri, öğrenme oranının 2 tabanında logaritmik değerine eşittir. Denklem(1), logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında i işinin işlem süresini hesaplamaktadır.

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

$a > 0$, bozulma oranı (dr), $t_r > 0$, i görevinin başlangıç zamanıdır.

Denklem(2), r. sıradaki i işinin, logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altındaki işlem süresini hesaplamaktadır. Görevler, önce bozulmadan, daha sonra öğrenmeden etkilenir. Çünkü öğrenme etkisi işin pozisyonu ile ilgiliyken, bozulma etkisi işin başlama zamanı

ile ilişkilidir. Öğrenme, işin kendisinden önceki işlerin işlem sürelerinin logaritmik değerlerinin toplamı ile ilgilidir. Yani, iş ne kadar ileri sıraya atanırsa, öğrenme etkisi o kadar fazla olur. Bozulma ise işin başlangıcında vardır.

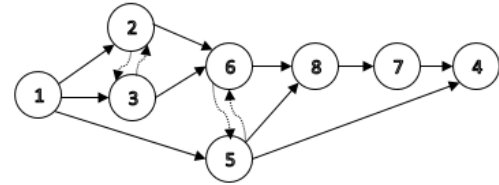
Güngör ve Gupta [10] tarafından oluşturulan DHD probleminin matematiksel modeli, logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında DHD problemi için geliştirilmiştir. Mevcut problem 8 görevden oluşmaktadır. Tablo 1, görevleri ve ilgili parametreleri göstermektedir.

Görevler arasındaki öncelik ilişkileri Şekil 1'de görüldüğü gibidir. Şekil 1'de de görüldüğü gibi Parça-1, Parça-2, 3 ve 5'ten önce demonte edilmelidir.

Tablo 1. Görev tanımları ve işlem süreleri

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

Parça-6, Parça-2 ve 3'ten sonra demonte edilmelidir. İşlerin istasyonlara atanmasında işler arasındaki öncelik ilişkileri önemlidir. Öncelik ilişkileri, DHD problemini karmaşıktırıp zorlaştırmaktadır.



Şekil 1. 8-parçalı verinin öncelik ilişkileri

- Model geliştirilirken dikkate alınması gereken varsayımlar;
1. Her iş sadece bir istasyonda sadece bir pozisyona atanabilir.
 2. İşlerin istasyona atanmasında öncelik ilişkileri dikkate alınmalıdır.
 3. Tek tip ürün işlenir.
 4. Bir demonte görevi birden fazla iş istasyonuna bölünemez.
 5. Parça çıkarma süreleri deterministik ve sabittir.
 6. Bir istasyona atanan tüm işlerin işlem sürelerinin toplamı çevrim süresini geçmemelidir.
 7. Sökme görevleri arasında yıkıcı işlemler yoktur.

Notasyonlar:

İndeksler:

$i, k \in N$ görev kümesi

$(i, k) \in SP$ etkileşimli görevler kümesi

$j \in M$ istasyon kümesi

$r \in N$ görevin istasyondaki konumu

Parametreler:

a : öğrenme indeksi,

α : bozulma indeksi,

M : büyük bir sayı,

ct =her istasyonun çevrim süresi,

Karar değişkenleri:

y_{ij} : i görevi j istasyonuna atanırsa, 1; Aksi takdirde 0,

z_j : j istasyonu açıksa, 1; Aksi takdirde 0,

x_{ijr} : i görevi j istasyonunda r pozisyonuna atanırsa, 1; Aksi takdirde 0,

b_{jr} : j istasyonunda r . sıraya atama yapıldıysa, 1; Aksi takdirde 0,

t_{jr} : j istasyonundaki r . sıradaki görevin işlem süresi,

c_{jr} : j istasyonundaki r . sıradaki görevin tamamlanma zamanı,

Aşağıdaki gibi verilen model formülasyonu:

$$f = \text{Min} \sum_{j=1}^n z_j \quad (3)$$

Kısıtlar:

$$\sum_{j=1}^n y_{ij} = 1 \quad \text{for } \forall i \in n \quad (4)$$

$$\sum_{j=1}^n y_{ij} \leq z_j \times ct \quad \text{for } \forall i \in n \quad (5)$$

$$\sum_{r=1}^n c_{jr} \times x_{ajr} \times b_{jr} - \sum_{R=1}^n c_{jR} \times x_{kjr} \times b_{jR} \leq 0 \quad \text{for } \{a, k\} \quad (6)$$

$$\sum_{i=1}^n x_{ijr} \leq 1 \quad \text{for } \forall j \in M, r \in N \quad (7)$$

$$\sum_{r=1}^n x_{ijr} = y_{ij} \quad \text{for } \forall j \in M, i \in N \quad (8)$$

$$\sum_{i=1}^n \left((g_i + \alpha \times c_{j(r-1)} \times b_{jr}) \times x_{ijr} \right) \times \left(1 + \sum_{l=1}^{r-1} (\ln g_l \times x_{ijl}) \right)^{\alpha} = t_{jr} \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (9)$$

$$\sum_{i=1}^n g_i \times x_{ijr} \times b_{jr} = t_{jr} \quad \text{for } \forall i \in N, r = 1 \quad (10)$$

$$c_{jr} = c_{j(r-1)} + t_{jr} \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (11)$$

$$t_{jr} \times b_{jr} \leq t_{j(r-1)} \times b_{j(r-1)} \times M \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (12)$$

$$\frac{c_{jr} \times b_{jr}}{M} \leq \frac{c_{j(r-1)} \times b_{j(r-1)}}{M} \quad \text{for } \forall j \in M, r \in N \text{ and } r \neq 1 \quad (13)$$

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

$$x_{ij(r-1)} \leq \sum_{i=1}^n x_{ijr} \quad (I \neq i), (I = 1, \dots, N) \quad (15)$$

$$x_{ijr}, y_{ij}, b_{jr}, z_j \in \{0,1\} \quad (16)$$

Denklem (3)' te verilen çevrim süresini aşmayacak şekilde açılacak istasyon sayısını en aza indirmek amaçlanmaktadır.

Kısıt (4) ve (8)' de, herhangi bir görevin yalnızca bir istasyona atanabilmesi, bu istasyonda da yalnızca bir sıraya atanabilmesi sağlanmaktadır. Kısıt (5), herhangi bir istasyona atanan işlerin işlem süreleri toplamının çevrim süresini aşmamasını sağlamaktadır. Kısıt (6), işler istasyona atanırken öncelik ilişkilerinin dikkate alınmasını sağlamaktadır. Kısıt(7), bir işin herhangi bir istasyonda yalnızca bir pozisyona atanabilmesini sağlamaktadır. Kısıt(9), herhangi bir görev ilk sıradan farklı bir sıraya atandığında işlem süresini, logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkisi altında hesaplamaktadır. Kısıt(10), herhangi bir işin bir istasyonda ilk sıraya atandığında ki işlem süresini hesaplamaktadır. Kısıt(11), herhangi bir j istasyonundaki iş tamamlanma süresini hesaplamaktadır. Kısıt(12), iş istasyonlarında farklı pozisyonlardaki işlerin işlem süreleri arasındaki ilişkiyi düzenlemektedir. Kısıt (13,14), herhangi bir istasyonda r pozisyonundaki işin işlem tamamlama süresi ile $r-1$ pozisyonundaki işin işlem tamamlama süresi arasındaki ilişkiyi sağlamaktadır. Kısıt(15), bir istasyonda bir işten önce yalnızca bir tane işin yapılabileceğini belirtir. Kısıt (16) ise ikili değişkenleri tanımlamaktadır.

Hibrit PSO-GA metasezgisel yaklaşım

Logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında DHD probleminin çözümünde hibrit PSO-GA metasezgisel yaklaşımı kullanılmıştır. Yöntemin uygulanmasında algoritmanın başlangıç çözümü aşamasında PSO algoritması belirli sayıda iterasyon çalıştırılmıştır. Daha sonra Genetik algoritmanın temel aşamaları olan çaprazlama ve mutasyon yöntemleri ile optimal çözüm aranmaya devam edilmiştir.

Parçacı Sürüsü Optimizasyonu (PSO)

Eberhart ve Kennedy [47], kuş-balık sürülerinin yiyecek ararken ki hareketlerinden hatta bazen insan davranışlarından esinlenerek, stokastik optimizasyon stratejisine dayalı bir yöntem olan PSO'yu geliştirmişlerdir. PSO'nun temel amacı, sürüyü oluşturan parçacıklar arasındaki sosyal bilgi paylaşımını iyileştirmektir. PSO, rastgele çözüm adaylarıyla popülasyon başlatma ve ardışık nesillerde küresel optimumu arama açısından GA'ya benzese de, çaprazlama ve mutasyona uğramaz. PSO'nun uygulanması diğer sezgisel algoritmalara göre çok kolaydır. PSO daha az hesaplama gerektirir ve düzenlenmesi gereken az sayıda parametresi vardır [48].

PSO, rastgele üretilmiş başlangıç çözümlerinden oluşan bir popülasyonla başlar. Optimal çözüm, çözüm uzayında iteratif olarak aranır. Çözümler, arama uzayında hareket eden parçacıklar olarak adlandırılmaktadır ve her bir parçacık bir çözüm temsil etmektedir. Her bir çözüm, parçacığın arama uzayındaki konumunu gösterir. Ayrıca, her parçacık kendi deneyimi ve tüm popülasyonun deneyiminden faydalanarak sürekli güncellenen bir hızla hareket eder. Her parçacık, bulunduğu iterasyona kadar elde ettiği pbest olarak adlandırılan en iyi sonucu ve tüm sürünün o ana kadar elde ettiği gbest olarak adlandırılan en iyi sonucu hatırlar. Dolayısıyla her bir parçacık, arama uzayındaki konumu, x_i^k , ve mevcut hızı, v_i^k , olmak üzere iki niteliğe sahiptir. x_i^k ve v_i^k , sırasıyla i parçacığının k . iterasyondaki konumu ve hızını temsil etmektedir. Parçacığın $(k+1)$. iterasyondaki konumu ve hızı güncellenirken aşağıdaki denklemler kullanılmaktadır.

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (17)$$

$$v_i^{k+1} = w \times v_i^k + c_1 \times \text{rand1}(pbest_i^k - x_i^k) + c_2 \times \text{rand2}(gbest^k - x_i^k) \quad (18)$$

$$w = w_{max} - \left(\frac{w_{max} - w_{min}}{\text{iterasyon sayısı}} \right) \times \text{mevcut iterasyon} \quad (19)$$

Burada; rand1 ve rand2, 0-1 aralığında rassal üretilmiş iki sayıdır, w atalet ağırlığıdır. w_{min} , 0.4 ile w_{max} , 0.9 arasında bir sayıdır. c_1 ve c_2 ivme katsayısı sabitleridir ve $c_1=c_2=2$ olarak alınmıştır [47], [49]. Ayrıca, $pbest_i^k$ i parçacığının en iyi konumudur, $gbest^k$ k. iterasyona kadarki küresel en iyi konumdur.

Genetik Algoritma (GA)

Genetik algoritma DLB problemlerinde en yaygın kullanılan evrimsel algoritmalarından biridir [4], [8]. Evrimsel algoritmalar, sadece rastgele çözümler bulmak için tasarlanmamıştır. Olası hata ve belirsizlik oranı içeren yöntemler kullanarak rastgele çözümlerden çok daha iyi çözümler üretmek temel amacdır. GA'da popülasyon, genlerin bir araya gelmesiyle kromozomlardan oluşur ve iki ebeveyn kromozomdan yeni bireylerin üretildiği bir çözüm yapısına sahiptir. Genetik algoritma, kromozomların sürekli çaprazlandığı, mutasyona uğradığı ve en iyi çözüme ulaşılan kadar birbirleriyle rekabet ettiği bir ortam sağlar [8]. GA ile çözüm aşamasında uygulanan adımlar: başlangıç popülasyonunu oluşturma, uygunluk fonksiyonu ve seçimi, turnuva seçim yöntemi, çaprazlama ve mutasyondur.

Başlangıç popülasyonu oluşturma

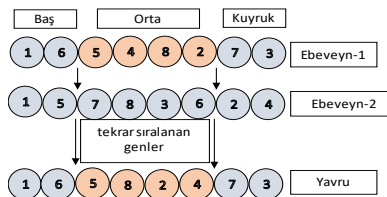
DHD problemini oluşturan işler GA'da ki genlerdir ve işlerin sıralı hali olan bir çözüm adayı ise kromozomdur. Problemi oluşturan işlerin sayısı kadar büyüklükte bir rassal sıra oluşturulur. GA başlangıç popülasyonu bu rassal sıralardan oluşmaktadır. Rassal sıraya göre işler istasyonlara öncelik ilişkileri ve çevrim süresi kısıtları göz önünde bulundurularak atanmaktadır.

Uygunluk fonksiyonu ve seçimi

Bir bireyin arama uzayındaki performansının ölçüsü uygunluk fonksiyonudur. Çalışmamızda uygunluk fonksiyonu amaç fonksiyonumuz olan minimum istasyon sayısıdır. Ebeveyn seçiminde turnuva seçim tekniği kullanılmıştır. Tüm aday kromozomların uygunluk değerleri hesaplanmış ve ebeveyn popülasyonu, daha iyi performans gösteren adaylardan oluşturulmuştur.

Çaprazlama Yöntemi

Bu çalışmada çaprazlama yöntemi olarak parça yeniden sıralama yöntemi [15] kullanılmıştır. Şekil 2, bu çaprazlama yönteminin ayrıntılarını göstermektedir.



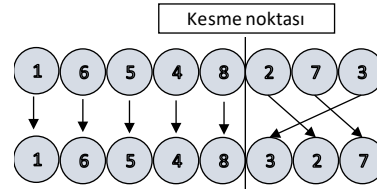
Şekil 2. Parçaları yeniden sıralama yöntemi

Her ebeveyn kromozom rastgele seçilen iki nokta ile 3 parçaya bölünür. Baş ve kuyruk kısımdaki genlerin sıralaması

aynı şekilde yavru kromozoma aktarılır, Ebeveyn-1' de orta parçadaki genler ise Ebeveyn-2'deki sıraları doğrultusunda yavru kromozomun orta kısmına tekrar sıralanarak aktarılır (Şekil 2). Diğer yavru kromozomda ise baş ve kuyruk kısmı Ebeveyn-2 'den aynen aktarılırken, orta kısım için genler Ebeveyn-1'deki sıralarına göre aktarılır.

Mutasyon Yöntemi

Bu çalışmada mutasyon işlemi olarak tek nokta sağ operatör yöntemi (one point right operatör method) [15] kullanılmıştır. Şekil 3, bu çaprazlama yönteminin ayrıntılarını göstermektedir.



Şekil 3. Tek nokta sağ operatör yöntemi

Mutasyon yönteminin uygulanacağı kromozom üzerinde mutasyon oranı dikkate alınarak bir nokta seçilir. Seçilen noktanın sağ tarafında kalan genler rassal olarak tekrar sıralanır (Şekil 3).

Hibrit PSO-GA algoritmasında problem için başlangıç popülasyonu oluşturulduktan sonra ilk uygunluk fonksiyonu değerleri elde edilir. Ardından oluşturan popülasyon için PSO algoritmasının adımları belirlenen iterasyon sayısı kadar uygulanmıştır. GA'nın aşamaları olan çaprazlama ve mutasyondan önce PSO ile çözüm adayları iyileştirilmiştir. Daha sonra, çaprazlama ve mutasyon işlemleri belirlenen çaprazlama ve mutasyon oranları dahilinde uygulanmıştır ve optimum veya optimuma yakın sonuçlar büyük boyutlu problemler için de elde edilmiştir. Hibrit PSO-GA algoritması C# programlama dili kullanılarak yazılmıştır.

Bulgular ve Tartışma

Bu bölümde literatürde kullanılan klasik DHD problemleri logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında incelenmiş olup ve hibrit PSO-GA algoritması ile çözümler bulunmuştur. Problemlerin çözüm yöntemlerinin uygulanmasında Intel Core (TM) i5-5200U CPU, 2.20 GHz, 8 GB RAM özellikli bilgisayar kullanılmıştır. Çalışmada, DHD problemlerinde eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkisini değerlendirebilmek için 0.90, 0.80 ve 0.70 olmak üzere 3 farklı öğrenme oranı, 0.10 ve 0.20 olmak üzere iki farklı iş bozulması oranı kullanılmıştır. Bahsedilen iki etki altında demontaj işlemlerinin işlem sürelerinin ve amaç fonksiyonunun nasıl etkilendiği incelenmiştir. Öğrenme etkisi oranı 0.80 iken hesaplamada kullanılan öğrenme etkisi değeri $\log_2^{0.80} = -0,322$ olarak hesaplanmaktadır.

8 parçalı veri seti için, görev parametreleri kullanılarak çevrim süresi 40 iken istasyon sayısı 4 olarak bulunmuştur [46]. Bu çalışmada, aynı problem için Lingo.11 optimizasyon programı kullanılarak amaç fonksiyonu değeri 4 olarak bulunmuştur. Görevlerin istasyonlara atanması şu şekildedir: Görev-1-2-3, istasyon-1'e atanır, Görev 5-6, istasyon 2'ye

atanır, Görev 7-4, istasyon-3'e atanır ve Görev-8, sırasıyla istasyon-4'e atanır.

Bu çalışmada öğrenme oranı değerleri 0.9, 0.8 ve 0.7, bozulma oranları ise 0.1 ve 0.2 kabul edilmiştir. Öğrenme oranı 0.9 ve bozulma oranı sırasıyla 0.1 ve 0.2 iken, logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında 8-parçalı DHD problemi için amaç fonksiyonu değerleri her iki durum için de 4 bulunmuştur. Öğrenme oranı 0.8 iken, bozulma oranı sırasıyla 0.1 ve 0.2 iken etkiler altında amaç fonksiyonu değerleri 3 bulunmuştur. Öğrenme oranı 0.7 iken ise bozulma oranı sırasıyla 0.1 ve 0.2 olduğunda, etkiler altında amaç fonksiyonu değerleri yine 3 olarak bulunmuştur.

25-parçalı telefon demontajı örneğinin logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında incelenmesi

DHD literatüründe sıklıkla kullanılan 25 parçadan oluşan hücreli telefon demontaj [50] örneği için sonuçlar elde edilmiştir. 25-parçalı DHD örneğinin görev tanımları, işlem süreleri ve işler arasındaki öncelik ilişkileri Tablo 2'de görülmektedir. Problem için çevrim süresi 18'dir.

Tablo 2. 25-parçalı DHD örneği için görev tanımları, işlem 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

Hibrit PSO-GA sezgisel yaklaşımı 500 iterasyon çalıştırılarak sonuçlar üretilmiştir. Öğrenme oranı ve

bozulma oranı değerleri sıfır iken, yani problemin literatürdeki baz hali için amaç fonksiyonu değeri olan açılan istasyon sayısı değeri 9 olarak bulunmuştur. Tablo 3'te, DHD literatüründe optimum sonuç olarak belirtilen sonucun, oluşturmuş olduğumuz algoritma ile de bulunduğu görülmektedir.

Elde edilen sonuca göre, istasyondaki işlerin tamamlanma süresi çevrim süresini aşmayacak ve öncelik ilişkilerini sağlayacak şekilde atamalar yapılmıştır. 23 ve 3 nolu işler İstasyon-1'e, 11 ve 9 nolu işler İstasyon-2'ye, 25-13-10-22-18-16 ve 12 nolu işler İstasyon-3'e, 8 nolu iş İstasyon -4'e, 7 ve 14 nolu işler İstasyon-5'e, 19 nolu iş İstasyon-6'ya, 6 ve 24 nolu işler İstasyon-7'ye, 2-1-14 ve 4 nolu işler İstasyon-8'e atanırken son olarakta 20-15-5 ve 21 nolu işler İstasyon-9'a atanmıştır (Tablo 4).

Tablo 3. 25-parçalı DHD probleminin literatürdeki çalışmalarda bulunan sonuçlarının karşılaştırılması

Makale	Yaklaşım	İstasyon sayısı
McGovern ve Gupta (2005)	H-K	10
McGovern ve Gupta (2006)	ACO	9
Ding vd. (2010)	MDACO	9
Kalayci vd. (2012a)	SA	9
Kalayci ve Gupta (2012b)	PSO	9
Kalayci vd.(2016)	VNSGA	9
Duta vd.(2016)	CGA	9
Edis(2021)	CP	9
Li vd.(2020)	BBRA	9
Bizim çalışmamız	Hibrit PSO-GA	9

Logaritmik işlem süreleri toplamı tabanlı öğrenmenin ve iş bozulmasının eş zamanlı etkisi altında mevcut 25-parçalı DHD probleminde, öğrenme ve bozulma oranları değiştikçe amaç fonksiyonu değeri de değişiklik göstermektedir (Tablo 4). Program, problem için sonuçları elde ederken 500 iterasyon çalıştırılmıştır. Tablo 4'te de görüldüğü üzere, eş zamanlı öğrenme ve bozulma etkisi altında amaç fonksiyonu değeri genellikle azalmıştır. Öğrenme oranı değeri 0.90 iken, bozulma oranı değeri 0.20 olduğunda açılan istasyon sayısı 10'a çıkmıştır. Öğrenme oranının 0.90 olması, 1 birim zamanda yapılabilen bir işin 0.90 birim zamanda yapılabilen olmasıdır. Yani öğrenme oranı değeri düşükçe işin daha kısa sürede yapılıyor olması demektir.

Bozulma etkisi değeri arttıkça işlerin işlem süreleri artış göstermektedir. Yani, öğrenme ile işin işlem süresi azalırken, bozulmanın etkisi ile işlem süresinde artış görülebilmektedir. Gerçek hayat problemlerinde, bir iş hem öğrenmeden hem de bozulmadan eş zamanlı etkilendiği için, bu çalışmada da işlemlerin her iki etkiden de etkilendiği varsayılmıştır.

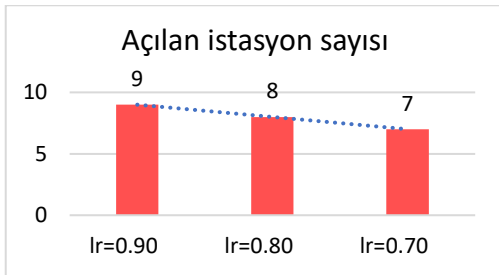
Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi ve iş bozulması etkisi dikkate alınmadığı durumda 9 istasyon açılması gerekirken, öğrenme oranı 0.90 ve bozulma oranı 0.20 olduğunda açılan istasyon sayısı 10 olarak bulunmuştur.

Tablo 4. 25-parçalı veri seti için logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkisi altında görevlerin işlem süreleri ve açılan istasyon sayıları

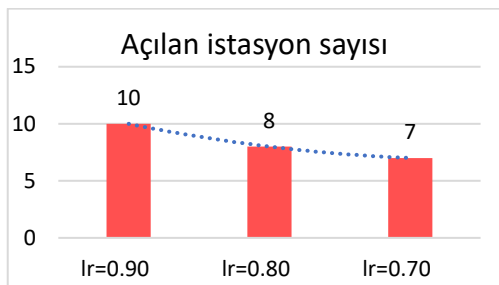
Senaryo	Atanan İşler	İşlem süreleri	İş Tamamlama süresi	Açılan istasyon sayısı
P25 with $lr=0.00$, $dr=0.00$, $ct=18$				
Station 1	23-3	15-3	18	
Station 2	11-9	2-15	17	
Station 3	25-13-10-22-18-16-12	2-2-2-5-3-2-2	18	
Station 4	8	15	15	
Station 5	7-14	15-2	17	9
Station 6	19	18	18	
Station 7	6-24	15-2	17	
Station 8	2-1-14-4	2-3-2-10	17	
Station 9	20-15-5-21	5-2-10-1	18	
P25 with $lr=0.90$, $dr=0.10$, $ct=18$				
Station 1	9	15	15	
Station 2	5-21-3-15	10-1,67-3,39-2,74	17,8	
Station 3	19	18	18	
Station 4	12-13-25-4-16	2-2,03-2,1-8,91-2,72	17,76	
Station 5	17-10-23	2-2,03-13,48	17,51	9
Station 6	11-7	2-14,03	16,03	
Station 7	24-8	2-14,03	16,03	
Station 8	22-2-20-18-14	5-2,16-4,75-3,29-2,67	17,87	
Station 9	1-6	3-13,67	16,67	
P25 with $lr=0.90$, $dr=0.20$, $ct=18$				
Station 1	3-21-12-13	3-1,43-2,52-2,82	9,77	
Station 2	17-6	2-14,21	16,21	
Station 3	15-1-25-4	2-3,14-2,58-9,43	17,15	
Station 4	18-23	3-13,94	16,94	
Station 5	2-11-20-14-16	2-2,22-5,09-3,11-3,48	15,9	10
Station 6	19	18	18	
Station 7	5-22	10-5,84	15,84	
Station 8	8	15	15	
Station 9	24-9	2-14,21	16,21	
Station 10	10-7	2-14,21	16,21	
P25 with $lr=0.8$, $dr=0.1$, $ct=18$				
Station 1	5-18-17-1	10-2,72-2,04-2,66	17,42	
Station 2	4-15-16	10-2,04-2,05	14,09	
Station 3	8-13	15-2,29	17,29	
Station 4	25-3-11-23	2-2,7-1,79-10,68	17,17	
Station 5	24-20-7	2-4,39-10,78	17,17	8
Station 6	14-19	2-15,36	17,36	
Station 7	10-2-21-12-9	2-1,86-1,06-1,89-11	17,81	
Station 8	22-6	5-11,38	16,38	
P25 with $lr=0.8$, $dr=0.2$, $ct=18$				
Station 1	17-11-9	2-2,03-11,9	15,93	
Station 2	2-10-14-23	2-10-14-23	17,37	
Station 3	22-6	5-11,75	16,75	
Station 4	21-8	1-15,2	16,2	
Station 5	25-1-7	2-2,87-11,54	16,41	8
Station 6	12-18-5-20	2-2,87-7,93-4,55	17,35	
Station 7	16-19	2-15,53	17,53	
Station 8	15-24-3-13-4	2-2,03-2,87-2,27-7,41	16,58	
P25 with $lr=0.7$, $dr=0.1$, $ct=18$				
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	7
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	
P25 with $lr=0.7$, $dr=0.2$, $ct=18$				
Station 1	24-22-6	2-4,12-9,05	13,75	
Station 2	21-18-12-8-13	1-3,2-1,91-9,53-2,22	18	
Station 3	23	15	15,45	
Station 4	1-5-14-4	3-7,24-1,96-5,58	17,81	7
Station 5	16-15-10-19	2-1,83-1,8-11,08	16,03	
Station 6	11-3-25-9	2-2,59-1,77-8,92	16,51	
Station 7	2-17-7-20	2-1,83-10,28-3,55	17,98	

Bu durumda, iş bozulması etkisi, öğrenme etkisine baskın gelmiştir. Bazı işlerin işlem sürelerinde artış görülmektedir. Örneğin, 5. istasyona 2. sırada atanan 11 numaralı işin işlem süresi normalde 2 iken, etkiler altında 2.22 olmuştur, aynı şekilde aynı istasyonda 4. sıraya atanan 14 numaralı işin işlem süresi normalde 2 iken etkiler altında 3.11 olmuştur. İşlerin işlem süreleri arttığından, çevrim süresini koruyabilmek için istasyonlara atanan iş sayıları azalmıştır ve açılan istasyon sayısında artış meydana gelmiştir.

Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi değeri 0.7 iken, iş bozulması etkisi 0.1 olduğunda 7 istasyon açılması gerekirken, iş bozulması oranı 0.2 olduğunda 8 istasyon açılması gerekmektedir. Öğrenme oranı sabit tutulduğunda, iş bozulması oranındaki artış ile amaç fonksiyonu değerinde de ~ 14.3 artış olduğu görülmektedir (Tablo 4). Benzer şekilde bozulma oranının 0.1 olduğu durumda, logaritmik işlem süreleri toplamı tabanlı öğrenme oranı 0.7 iken 7 istasyon açılırken, 0.8 olduğu durumda 8 istasyon açılması gerekmektedir. Dolayısıyla, iş bozulması oranı sabit olduğunda öğrenme oranındaki 0.1 birimlik artış ile amaç fonksiyonu değerinde ~ 14.3 oranında artış meydana gelmiştir.



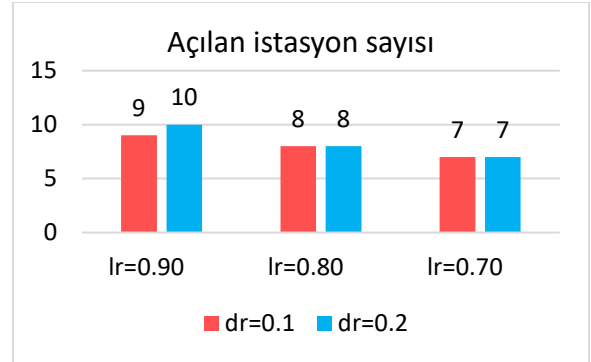
Şekil 4. İş bozulması oranı 0.1 iken farklı logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi değerleri ile hesaplanan amaç fonksiyonu değerleri



Şekil 5. İş bozulması oranı 0.2 iken farklı logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi değerleri ile hesaplanan amaç fonksiyonu değerleri

İş bozulması oranı sırasıyla, 0.1 ve 0.2 olduğunda farklı logaritmik işlem süreleri toplamı tabanlı öğrenme oranları altında 25-parçalı DHD problemi için açılan istasyon sayıları Şekil 4 ve Şekil 5'te görülmektedir. Öğrenme oranı azaldıkça, açılan istasyon sayısı değeri de azalmaktadır. Logaritmik işlem süreleri toplamı tabanlı öğrenme oranları sabit iken farklı iş bozulması oranları için elde edilen sonuçlar Şekil 6'da görülmektedir. İş bozulması oranı arttıkça açılan istasyon sayısı artmaktadır. Çünkü iş

bozulması işlerin işlem süreleri üzerinde artırıcı etkiye sahiptir.



Şekil 6. Aynı logaritmik işlem süreleri toplamı tabanlı öğrenme oranı ile farklı iş bozulması oranları için hesaplanan amaç fonksiyonu değerleri

Literatürdeki 15 farklı problem için sonuçları karşılaştırılması

Literatürde bulunan 15 farklı DHD problemi [54], eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkisi dikkate alınarak çözdürülmüştür. 0.9, 0.8, 0.7 olmak üzere 3 farklı öğrenme oranı ve 0.1 ve 0.2 olmak üzere 2 farklı iş bozulması oranı kullanılmıştır.

Tablo 5'te belirtilen veri setleri için öğrenme ve bozulma etkileri dikkate alınmadığı durumda bulunan istasyon sayısı değerleri, literatürde bulunan optimum amaç fonksiyonu değerleri ile aynıdır. Yani, hibrit PSO-GA algoritması ile parça sayısı 70'den küçük olan problemler için 500 iterasyon, 70'den büyük problemler için ise 1000 iterasyon çalıştırılarak optimum sonuçlar elde edilmiştir.

Tablo 6'da, optimal sonuç sütununda literatürdeki bulunan optimal sonuçlar yer alırken, lr=0-dr=0 sütununda bu çalışmada elde edilen sonuçlar yer almaktadır.

Öğrenme oranı 0.9 iken, bozulma oranı 0.2 olduğunda çalışılan veri setlerinin çoğunluğunda herhangi bir öğrenme veya bozulma etkisi dikkate alınmadığı durumla aynı veya daha kötü sonuçlar elde edilmiştir. Birkaç problem için daha iyi sonuç elde edilebilmiştir.

Bozulma oranının 0.2 olduğunda, öğrenme oranı 0.9 ise iş bozulmasının artırıcı etkisi, logaritmik işlem süreleri toplamı tabanlı öğrenme etkisinden baskın gelmiştir. Örneğin; 45-parçalı (Kilbrid) problemde, etkiler düşünülmediğinde açılan istasyon sayısı 10 iken, öğrenme oranı 0.9 ve bozulma oranı 0.2 olduğunda logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında açılan istasyon sayısı 11 bulunmuştur.

Öğrenme oranı 0.8 ve 0.7 olduğu tüm durumlarda ise amaç fonksiyonu değerinde iyileşme görülmektedir. 89-parçalı Lutz-3 veri setinde öğrenme ve bozulma etkisi dikkate alınmadığı durumda açılan istasyon sayısı 22 bulunmuştur. Öğrenme oranı 0.9 iken, bozulma oranları sırasıyla 0.1 ve 0.2 iken açılan istasyon sayıları sırasıyla 21 ve 23'tür. Burada, bozulma etkisi arttığında istasyon sayısı beklenen şekilde artış göstermiştir. Öğrenme oranı 0.8 iken ve bozulma oranları sırasıyla 0.1 ve 0.2 iken açılan istasyon sayıları sırasıyla 15 ve 18 olarak bulunmuştur. Öğrenme oranındaki 0.10 luk değişim, bozulma oranları 0.1 ve 0.2 iken amaç

Tablo 5. Literatürdeki farklı demontaj problemleri için logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkisi altında elde edilen optimum sonuçlar

Yazar	n	Hibrit PSO-GA Algoritması ile Elde Edilen Sonuçlar									
		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	

Tablo 6. Logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında amaç fonksiyonundaki değişim (%)

Yazar	Öğrenme ve Bozulma Etkileri Altında Amaç Fonksiyonundaki Değişim (%)					
	öo=0.9		öo=0.8		öo=0.7	
	bo=0.1	bo=0.2	bo=0.1	bo=0.2	bo=0.1	bo=0.2
Güngör and Gupta	0,00	0,00	-25,00	-25,00	-25,00	-25,00
Mansoor	0,00	0,00	-25,00	-25,00	-50,00	-50,00
Mitchell	0,00	0,00	-12,50	-12,50	-37,50	-25,00
Gupta et al.	0,00	11,11	-11,11	-11,11	-22,22	-22,22
Heskiaoff	-12,50	0,00	-37,50	-25,00	-62,50	-50,00
Buxey	-8,33	0,00	-25,00	-16,67	-41,67	-41,67
Lutz-1	-18,18	-9,09	-45,45	-36,36	-72,73	-63,64
Kilbrid	0,00	10,00	-30,00	-10,00	-50,00	-40,00
Hahn	-16,67	0,00	-50,00	-33,33	-50,00	-33,33
Tonge	-10,00	10,00	-40,00	-30,00	-60,00	-60,00
Wee-Mag	-3,03	0,00	-12,12	-9,09	-42,42	-36,36
Arcus-1	-43,75	-31,25	-62,50	-62,50	-68,75	-68,75
Lutz-3	-4,55	4,55	-31,82	-18,18	-54,55	-50,00
Mukherjee	-10,00	5,00	-40,00	-25,00	-65,00	-55,00
Arcus-2	-17,24	-10,34	-51,72	-48,28	-75,86	-75,86

Bu iyileşme oranları, DHD problemlerinin öğrenme ve bozulma etkisi düşünülerek çözülmesi gerektiğini göstermektedir.

fonksiyonu değerinde sırasıyla ~%28.5 ve ~%21.7 azalma sağlanmıştır (Tablo 6). Aynı problem için öğrenme oranı 0.7 olduğunda, bozulma oranları sırasıyla 0.1 ve 0.2 iken açılan istasyon sayıları sırasıyla 10 ve 11 olarak bulunmuştur.

DHD problemlerinin çözümünde öğrenme ve bozulma etkileri dikkate alınmalıdır. Çünkü bu iki olgu gerçek hayatta var olan durumlardır. Yapmış olunan çalışmada, her iki olgunun da problemin çözümünde önemli etkisi olduğu görülmektedir. Logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında amaç fonksiyonu değeri üzerinde %75'e varan iyileşme görülebilmektedir.

Farklı DHD problemlerinde, aynı öğrenme ve bozulma etkileri ile hesaplanan amaç fonksiyonu değerlerindeki iyileşme oranları farklılık gösterebilmektedir. Çünkü, her DHD problemi için istasyonların çevrim süresi kısıtları farklıdır. DHD problemlerinin çözümünde, istasyonların işleri tamamlama süreleri çevrim kısıtını aşmamalıdır. İşlerin işlem sürelerinin farklı olması ve istasyonlara işler atanırken öncelik ilişkilerinin dikkate alınması gerekliliğinden kaynaklı olarak, istasyonların iş tamamlama süreleri genellikle çevrim süresine eşit olmamaktadır. Böylece istasyonlarda arta kalan zamanlar meydana gelebilmektedir. Öğrenme ve bozulma etkileri altında DHD problemlerinin işlem süreleri genellikle kısaldığı için, işler mevcut istasyonlardaki boş zamanlara kısıtları sağlayacak şekilde dağılır. Dolayısıyla, istasyon sayılarında azalma görülebilmektedir. İstasyonların çevrim süresi arttıkça, istasyona yerleşen iş sayısında artmaktadır. Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisi altında, işin işlem süresi kendisinden önceki işlere bağlıdır. Dolayısıyla, bir istasyona ne kadar çok iş atanabiliyorsa, istasyonda daha ileri sıraya atanan işler öğrenmeden o kadar fazla etkilenmektedir. Örneğin; 53 görevden oluşan DHD problemi için herhangi bir etki düşünülmediği durumda 6 istasyon açılması uygun bulunmuştur. Bu durumda istasyon başına ortalama 8.8 iş düşmektedir. 89 parçalı DHD probleminde ise, etkiler düşünülmediğinde istasyon başına 5.2 görev düşmüştür. Öğrenme oranı 0.8 ve bozulma oranı 0.1 olduğunda 53 parçalı problem için açılan istasyon sayısı %50 iyileşme ile 3 bulunurken, 89 parçalı problem için %31.8 iyileşme ile 15 bulunmuştur (Tablo 6).

Sonuç

Bu çalışmada DHD problemi, eş zamanlı logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altında incelenmiştir. İşler tekrarlandıkça öğrenilir ve daha kısa sürede yapılır. Öğrenme etkisi de DHD problemlerinde işlerin işlem süreleri üzerinde azaltıcı etkiye sahiptir. Logaritmik işlem süreleri toplamı tabanlı öğrenme etkisinde, bir istasyonda herhangi bir sıraya atanan işin işlem süresi aynı istasyonda kendisinden önceki sıralara atanmış işlerin işlem sürelerinin logaritmik toplamının üstel öğrenme etkisi kuvvetiyle çarpılmasıyla hesaplanmaktadır. İş bozulması ise, bir işin başlangıç zamanının ötelenmesine sebep olan her türlü bozulmadır. İş bozulması, işin başlama zamanını artırdığı için, işin işlem süresini de artırmış olmaktadır. DHD problemlerinde işlerin işlem sürelerinin sabit olarak alınmaktadır. Oysaki, gerçek hayatta öğrenme ve bozulma, demontaj sürecinde olan durumlardır. Bu çalışmada, DHD literatüründe yaygın olarak kullanılmış olan problemler logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri dikkate alınarak, literatürdeki temel amaç

fonksiyonu olan istasyon sayısı minimizasyonu üzerine çalışılmıştır. Bu problem yapısının çözümü için hibrit PSO-GA algoritması geliştirilmiştir. Kullanılan öğrenme ve bozulma oranları doğrultusunda, açılması gereken istasyon sayısı değerlerinde önemli iyileşmeler kaydedilmiştir. Gelecek çalışmalarda, logaritmik işlem süreleri toplamı tabanlı öğrenme ve iş bozulması etkileri altındaki DHD problemleri için kar maksimizasyonu, artan zaman (idle time) minimizasyonu gibi farklı amaç fonksiyonları için çalışılabilir.

Etik kurul onayı ve çıkar çatışması beyanı

Hazırlanan makalede etik kurul izni alınmasına gerek yoktur.

Hazırlanan makalede herhangi bir kişi/kurum ile çıkar çatışması bulunmamaktadır.

Yazar Katkıları

SOMTÜRK

Modelin oluşturulması, C# kodunun oluşturulması ve yorumlanması

Çalışmanın yazılması görevlerini üstlenmiştir.

TOKSARI

Çalışma taslağının oluşturulması ve genel düzeninin planlamasını yapmıştır.

Kaynaklar

- [1] S. Hezer and Y. Kara, "A network-based shortest route model for parallel disassembly line balancing problem," *Int. J. Prod. Res.*, vol. 53, no. 6, pp. 1849–1865, Mar. 2015.
- [2] S. Agrawal and M. K. Tiwari, "A collaborative ant colony algorithm to stochastic mixed-model U-shaped disassembly line balancing and sequencing problem," *Int. J. Prod. Res.*, vol. 46, no. 6, pp. 1405–1429, Mar. 2008.
- [3] Z. Li and M. N. Janardhanan, "Modelling and solving profit-oriented U-shaped partial disassembly line balancing problem," *Expert Syst. Appl.*, vol. 183, no. October 2019, p. 115431, 2021.
- [4] K. Wang, X. Li, and L. Gao, "Modeling and optimization of multi-objective partial disassembly line balancing problem considering hazard and profit," *J. Clean. Prod.*, vol. 211, pp. 115–133, 2019.
- [5] E. B. Edis, "Constraint programming approaches to disassembly line balancing problem with sequencing decisions," *Comput. Oper. Res.*, vol. 126, p. 105111, 2021.
- [6] Z. Li, Z. A. Çil, S. Mete, and I. Kucukkoc, "A fast branch, bound and remember algorithm for disassembly line balancing problem," *Int. J. Prod. Res.*, vol. 58, no. 11, pp. 3220–3234, 2020.
- [7] E. Goksoy Kalaycilar, S. Batun, and M. Azizoğlu, "A stochastic programming approach for the

- disassembly line balancing with hazardous task failures,” *Int. J. Prod. Res.*, vol. 60, no. 10, pp. 3237–3262, 2022.
- [8] S. M. McGovern and S. M. Gupta, “A balancing method and genetic algorithm for disassembly line balancing,” *Eur. J. Oper. Res.*, vol. 179, no. 3, pp. 692–708, Jun. 2007.
- [9] A. Gungor and S. M. Gupta, “Disassembly line balancing,” in *Proceedings of the 1999 annual meeting of the northeast decision sciences institute*, 1999, pp. 193–195.
- [10] A. Güngör and S. M. Gupta, “A solution approach to the disassembly line balancing problem in the presence of task failures,” *Int. J. Prod. Res.*, vol. 39, no. 7, pp. 1427–1467, May 2001.
- [11] A. Güngör and S. M. Gupta, “Disassembly line in product recovery,” *Int. J. Prod. Res.*, vol. 40, no. 11, pp. 2569–2589, Jul. 2002.
- [12] S. M. McGovern and S. M. Gupta, “<Title>2-Opt Heuristic for the Disassembly Line Balancing Problem</Title>,” *Environ. Conscious Manuf. III*, vol. 5262, pp. 71–84, 2004.
- [13] Y. Ren *et al.*, “Disassembly line balancing problem using interdependent weights-based multi-criteria decision making and 2-Optimal algorithm,” *J. Clean. Prod.*, vol. 174, pp. 1475–1486, 2018.
- [14] S. Mete, Z. A. Çil, K. Ağpak, E. Özceylan, and A. Dolgui, “A solution approach based on beam search algorithm for disassembly line balancing problem,” *J. Manuf. Syst.*, vol. 41, pp. 188–200, 2016.
- [15] C. B. Kalayci, O. Polat, and S. M. Gupta, “A hybrid genetic algorithm for sequence-dependent disassembly line balancing problem,” *Ann. Oper. Res.*, vol. 242, no. 2, pp. 321–354, 2016.
- [16] A. Aydemir-Karadag and O. Turkbey, “Multi-objective optimization of stochastic disassembly line balancing with station paralleling,” *Comput. Ind. Eng.*, vol. 65, no. 3, pp. 413–425, 2013.
- [17] C. B. Kalayci and S. M. Gupta, “Ant colony optimization for sequence-dependent disassembly line balancing problem,” *J. Manuf. Technol. Manag.*, vol. 24, no. 3, pp. 413–427, 2013.
- [18] L. P. Ding, Y. X. Feng, J. R. Tan, and Y. C. Gao, “A new multi-objective ant colony algorithm for solving the disassembly line balancing problem,” *Int. J. Adv. Manuf. Technol.*, vol. 48, pp. 761–771, May 2010.
- [19] C. B. Kalayci and S. M. Gupta, “Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem,” *Expert Syst. Appl.*, vol. 40, no. 18, pp. 7231–7241, 2013.
- [20] J. Liu and S. Wang, “Balancing disassembly line in product recovery to promote the coordinated development of economy and environment,” *Sustain.*, vol. 9, no. 2, 2017.
- [21] C. B. Kalayci and S. M. Gupta, “A tabu search algorithm for balancing a sequence-dependent disassembly line,” *Prod. Plan. Control*, vol. 25, no. 2, pp. 149–160, Jan. 2014.
- [22] C. B. Kalayci and S. M. Gupta, “A particle swarm optimization algorithm with neighborhood-based mutation for sequence-dependent disassembly line balancing problem,” *Int. J. Adv. Manuf. Technol.*, vol. 69, pp. 197–209, 2013.
- [23] S. Xiao, Y. Wang, H. Yu, and S. Nie, “An entropy-based adaptive hybrid particle swarm optimization for disassembly line balancing problems,” *Entropy*, vol. 19, no. 11, p. 596, 2017.
- [24] Y. Ren, D. Yu, C. Zhang, G. Tian, L. Meng, and X. Zhou, “An improved gravitational search algorithm for profit-oriented partial disassembly line balancing problem,” *Int. J. Prod. Res.*, vol. 55, no. 24, pp. 7302–7316, Jun. 2017.
- [25] Z. Zhang, K. Wang, L. Zhu, and Y. Wang, “A Pareto improved artificial fish swarm algorithm for solving a multi-objective fuzzy disassembly line balancing problem,” *Expert Syst. Appl.*, vol. 86, pp. 1339–1351, 2017.
- [26] L. Zhu, Z. Zhang, and Y. Wang, “A Pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation,” *Int. J. Prod. Res.*, vol. 56, no. 24, pp. 7354–7374, 2018.
- [27] F. T. Altekin, L. Kandiller, and N. E. Ozdemirel, “Profit-oriented disassembly-line balancing,” *Int. J. Prod. Res.*, vol. 46, no. 10, pp. 2675–2693, May 2008.
- [28] F. T. Altekin, “A comparison of piecewise linear programming formulations for stochastic disassembly line balancing,” *Int. J. Prod. Res.*, vol. 55, no. 24, pp. 7412–7434, Jul. 2017.
- [29] S. Mete, Z. Abidin Çil, E. Özceylan, and K. Ağpak, “Resource constrained disassembly line balancing problem,” *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 921–925, 2016.
- [30] M. A. Ilgin, H. Akçay, and C. Araz, “Disassembly line balancing using linear physical programming,” *Int. J. Prod. Res.*, vol. 55, no. 20, pp. 6108–6119, Oct. 2017.
- [31] T. Paksoy, A. Güngör, E. Özceylan, and A. Hancılar, “Mixed model disassembly line balancing problem with fuzzy goals,” *Int. J. Prod. Res.*, vol. 51, no. 20, pp. 6082–6096, Oct. 2013.
- [32] E. Özceylan and T. Paksoy, “Reverse supply chain optimisation with disassembly line balancing,” *Int. J. Prod. Res.*, vol. 51, no. 20, pp. 5985–6001, 2013.
- [33] E. Özceylan, C. B. Kalayci, A. Güngör, and S. M. Gupta, “Disassembly line balancing problem: a review of the state of the art and future directions,” *International Journal of Production Research*, vol. 57, no. 15–16. Taylor and Francis Ltd., pp. 4805–4827, 2019.

- [34] Y. Laili, Y. Li, Y. Fang, D. T. Pham, and L. Zhang, "Model review and algorithm comparison on multi-objective disassembly line balancing," *J. Manuf. Syst.*, vol. 56, no. July, pp. 484–500, 2020.
- [35] G. Mosheiov, "Scheduling problems with a learning effect," *Eur. J. Oper. Res.*, vol. 132, no. 3, pp. 687–693, 2001.
- [36] T. P. Wright, "Factors Affecting the Cost of Engineering," *J. Aeronaut. Sci.*, vol. 3, no. 4, pp. 122–128, 1936.
- [37] D. Biskup, "Single-machine scheduling with learning considerations," *Eur. J. Oper. Res.*, vol. 115, pp. 173–178, 1999.
- [38] T. C. E. Cheng and G. Wang, "Single machine scheduling with learning effect considerations," *Ann. Oper. Res.*, vol. 98, pp. 273–290, 2000.
- [39] M. D. Toksari, E. K. Aydogan, B. Atalay, and S. Sari, "Some scheduling problems with sum of logarithm processing times based learning effect and exponential past sequence dependent delivery times," *J. Ind. Manag. Optim.*, vol. 18, no. 3, pp. 1795–1807, 2022.
- [40] T. C. E. Cheng, P. J. Lai, C. C. Wu, and W. C. Lee, "Single-machine scheduling with sum-of-logarithm-processing-times-based learning considerations," *Inf. Sci. (Ny)*, vol. 179, no. 18, pp. 3127–3135, 2009.
- [41] X. X. Liang, B. Zhang, J. B. Wang, N. Yin, and X. Huang, "Study on flow shop scheduling with sum-of-logarithm-processing-times-based learning effects," *J. Appl. Math. Comput.*, vol. 61, no. 1–2, pp. 373–388, 2019.
- [42] J. B. Wang and Z. Q. Xia, "Flow-shop scheduling with a learning effect," *J. Oper. Res. Soc.*, vol. 56, no. 11, pp. 1325–1330, 2005.
- [43] J. N. D. Gupta and S. K. Gupta, "Single facility scheduling with nonlinear processing times," *Comput. Ind. Eng.*, vol. 14, no. 4, pp. 387–393, 1988.
- [44] M. M. Mazdeh, F. Zaerpour, and F. F. Jahantigh, "A fuzzy modeling for single machine scheduling problem with deteriorating jobs," *Int. J. Ind. Eng. Comput.*, vol. 1, no. 2, pp. 147–156, 2010.
- [45] O. A. Arik and M. D. Toksari, "Minimizing makespan with fuzzy processing times under job deterioration and learning effect," *J. Ind. Eng.*, vol. 31, no. 1, pp. 1–17, 2020.
- [46] S. M. McGovern and S. M. Gupta, "Ant colony optimization for disassembly sequencing with multiple objectives," *Int. J. Adv. Manuf. Technol.*, vol. 30, no. 5–6, pp. 481–496, 2006.
- [47] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *In Proceedings of ICNN'95-international conference on neural networks*, 1995, pp. 1942–1948.
- [48] J. M. Nilakantan and S. G. Ponnambalam, "Robotic U-shaped assembly line balancing using particle swarm optimization," *Eng. Optim.*, vol. 48, no. 2, pp. 231–252, 2016.
- [49] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292–305, 2016.
- [50] S. M. Gupta, E. Erbis, and S. M. McGovern, "Disassembly sequencing problem: a case study of a cell phone," *Environ. Conscious Manuf. IV*, vol. 5583, pp. 43–52, 2004.
- [51] S. M. McGovern and S. M. Gupta, "Uninformed and probabilistic distributed agent combinatorial searches for the unary np-complete disassembly line balancing problem," *Environ. Conscious Manuf. V*, vol. 5997, pp. 81–92, 2005.
- [52] C. B. Kalayci, S. M. Gupta, and K. Nakashima, "A Simulated Annealing Algorithm for Balancing a Disassembly Line," *Des. Innov. Value Towar. a Sustain. Soc.*, no. 1, pp. 714–719, 2012.
- [53] L. Duta, I. Caciula, and P. C. Patric, "Column generation approach for disassembly line balancing," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 916–920, 2016.
- [54] A. Scholl, "Data of assembly line balancing problems," *Schriften zur quantitativen Betriebswirtschaftslehre*, vol. 93, no. 16, pp. 1–28, 1995.
- [55] G. Mosheiov, "A-shaped policies to schedule deteriorating jobs," *Journal of the Operational Research Society*, 47(9), pp.11841191,1996.