



## Assembly line balancing problem with resource and sequence-dependent setup times (ALBPRS)

Özcan Kılınççı\*

Department of Industrial Engineering, Faculty Engineering, Dokuz Eylul University, 35390, Izmir, Türkiye

### Highlights:

- There is more than one resource at each workstation
- Sequence-dependent setups occur at each resource
- Mathematical model and genetic algorithm are proposed to solve

### Keywords:

- Assembly line balancing problem
- Assembly line balancing with resource and sequence dependent setup times
- Mathematical model
- Genetic algorithm

### Article Info:

Research Article  
Received: 24.06.2020  
Accepted: 27.02.2022

### DOI:

10.17341/gazimmfd.757276

### Correspondence:

Author: Özcan Kılınççı  
e-mail:  
ozcan.kilincci@deu.edu.tr  
phone: 90 232 3017 612

### Graphical/Tabular Abstract

In this study, the assembly line balancing problem with sequence dependent setup times is extended to more than one resource at each workstation. Task is performed any resource at each workstation. The problem is to assign the tasks to the workstations, to assign the tasks to the resource, and to sequence the tasks performed by the same resource at each workstation simultaneously. Task times and sequence dependent setup times change according to the resource. The station time includes the task times performed by the resource and setup times between the tasks performed by the resource at the workstation. A mathematical model and a genetic algorithm are developed to solve the problem. A flowchart of the proposed genetic algorithm is shown in Figure A.

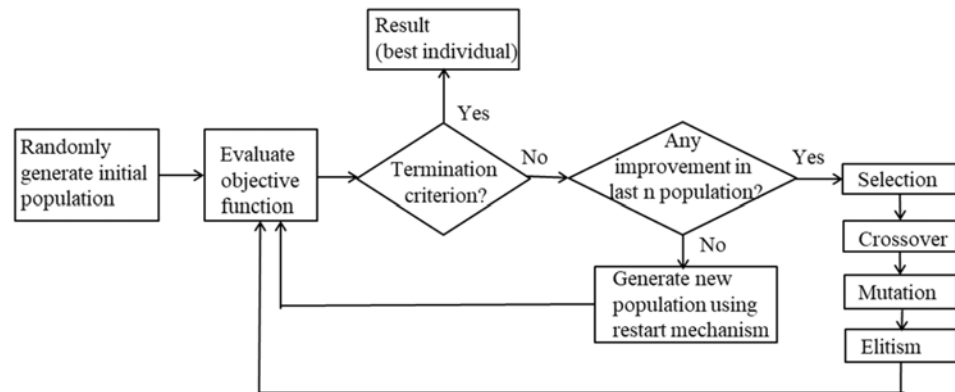


Figure A. Flowchart of the proposed genetic algorithm

### Purpose:

Using more than one resource at each workstation obtains more than one sequence at the workstation. The problem is more complex, but more than one resource at each workstation can decrease the number of setups. Thus, the required number of workstations on the line can be reduced.

### Theory and Methods:

To solve the problem, a binary linear mathematical model and a genetic algorithm are proposed. Lower bound is defined. The parameters of the proposed genetic algorithm are analyzed using a full factorial design. 160 test problems are generated.

### Results:

The mathematical model is written in LINGO 14, the genetic algorithm is coded in Matlab 2016a. Small sized problems are solved by the mathematical model. The proposed genetic algorithm is performed on the generated test problems. Average relative deviation from the lower bound is %17,99. Average cpu time is 239,18 sec. All results are analyzed according to the characteristics of the test problems. Better results are obtained in problems with more than 75 tasks and in problems with high setup times. .

### Conclusion:

Results show that the proposed methods are efficient and using more than one resource at each workstation decreases the total number of the workstations in many test problems.



## Kaynak ve sıra bağımlı hazırlık süreli montaj hattı dengeleme problemi (KMHDP)

Özcan Kılınçcı\*

Dokuz Eylül Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 35390, Buca, İzmir, Türkiye

### ÖNEÇIKANLAR

- Her iş istasyonunda birden fazla kaynak vardır
- Her bir kaynaktaki sıra bağımlı hazırlık süreleri oluşmaktadır
- Çözüm için matematiksel model ve genetik algoritma önerilmiştir

#### Makale Bilgileri

Araştırma Makalesi  
Geliş: 24.06.2020  
Kabul: 27.02.2022

#### DOI:

10.17341/gazimmfd.757276

#### Anahtar Kelimeler:

Montaj hattı dengeleme problemi,  
kaynak ve sıra bağımlı kurulum süreli montaj hattı dengeleme,  
matematiksel model,  
genetik algoritma.

#### ÖZ

Klasik Basit Montaj Hattı Dengeleme Problemi (BMHDP), son yıllarda birçok gerçek hayat uygulamaları ile farklılaştırılmaktadır. Bunlardan biri Sıra Bağımlı Hazırlık Süreli Montaj Hattı Dengeleme Problemidir (HMHDP). Bu çalışmada HMHDP, her iş istasyonunda birden fazla kaynak bulunması ile farklılaştırılmaktadır. İşlem, iş istasyonunda herhangi bir kaynak ile yapılır. Her iş istasyonunda birden fazla kaynak bulunması hazırlık sayısını azaltabilir, böylece hatta gerekli olan toplam iş istasyonu sayısı azalabilir. Problem, Kaynak ve Sıra Bağımlı Hazırlık Süreli Montaj Hattı Dengeleme Problemi (KMHDP) olarak tanımlanmıştır. KMHDP, eş zamanlı olarak, işlemin iş istasyonuna atanması, işlemin kaynağa atanması ve aynı kaynak kullanılarak yapılan işlemlerin sıralanmasıdır. Problemin çözümü için matematiksel model ve genetik algoritma geliştirilmiştir. KMHDP için veri seti oluşturulmuş, sunulan matematiksel model ve genetik algoritma veri seti kullanılarak test edilmiştir. Sonuçlar önerilen yöntemlerin etkinliğini ve her iş istasyonunda birden fazla kaynak kullanımının birçok test probleminde toplam iş istasyonu sayısında azalışlar sağladığını göstermektedir.

## Assembly line balancing problem with resource and sequence-dependent setup times (ALBPRS)

### HIGHLIGHTS

- There is more than one resource at each workstation
- Sequence-dependent setups occur at each resource
- Mathematical model and genetic algorithm are proposed to solve

#### Article Info

Research Article  
Received: 24.06.2020  
Accepted: 27.02.2022

#### DOI:

10.17341/gazimmfd.757276

#### Keywords:

Assembly line balancing problem,  
assembly line balancing with resource and sequence dependent setup times,  
mathematical model,  
genetic algorithm

#### ABSTRACT

The classical simple assembly line balancing problem (SALBP) has been extended with many real-life applications recently. One of these extensions is assembly line balancing with sequence-dependent setup times (ALBPS). In this study, ALBPS is extended with more than one resource at each workstation. Task is performed any resource at each workstation. More than one resource at each workstation can decrease the number of setups. Thus, the required number of the workstations on the line can reduce. The problem is called assembly line balancing with resource and sequence-dependent setup times (ALBPRS). ALBPRS is to assign the task to the workstations, to assign the tasks to the resource, and to sequence the tasks performed by the same resource at each workstation simultaneously. A mathematical model and a genetic algorithm are developed to solve the problem. The benchmark data set is generated for ALBPRS and the mathematical model and genetic algorithm are tested on the generated benchmark dataset. Results show that the proposed methods are efficient and using more than one resource at each workstation decreases the total number of the workstations in many test problems.

## 1. Giriş (Introduction)

Montaj hattı, üretim süreçleri birbirine benzeyen ürünlerin yüksek miktarda ve düşük maliyetle üretilmesini sağlayan akış tipi üretim sistemidir. Hat üzerindeki her iş istasyonunda, ürüne ait belirli işlemler tekrarlı olarak yapılarak sabit bir çevrim süresi içinde tamamlanır. Çevrim süresi sonunda istasyonda işlemleri tamamlanan ürün bir sonraki istasyona bir malzeme aktarma sistemi ile taşınır. Ürüne ait tüm işlemler hattaki son iş istasyonunda tamamlanır ve ürün hattan çıkar. Klasik Montaj Hattı Dengeleme Problemi (MHDP), işlem sürelerini ve işlemler arasındaki öncelik ilişkilerini dikkate alarak ürüne ait işlemlerin hat üzerindeki iş istasyonlarına atanması problemi. MHDP, literatürde araştırma yapılan temel problemlerden biridir. Araştırmacılar son yıllarda MHDP’de yapılan farklı çalışmaları, Becker ve Scholl [1], Scholl ve Becker [2], Boysen vd. [3, 4], Rashid vd [5], Battaia ve Dolgui [6], Sivasankaran ve Shahabudeen’e [7] ait kapsamlı literatür araştırmalarında bulabilirler.

Montaj hattı üzerinde sadece bir ürün çeşidi üretiliyorsa, ürüne ait işlem süreleri sabit ve biliniyorsa, problem Basit Montaj Hattı Dengeleme Problemi (BMHDP) olarak adlandırılır. BMHDP, verilen çevrim süresine uygun en küçük iş istasyonu sayısını bulmak amaçlanıyorsa Tip-1, verilen iş istasyonu sayısına uygun en küçük çevrim süresinin bulunması amaçlanıyorsa Tip-2 olarak sınıflandırılır. Klasik BMHDP’de hazırlık zamanları işlem zamanları ile karşılaştırıldığında çok küçük olduklarından dikkate alınmaz ya da işlem sürelerine eklenir. Andres vd. [8] klasik BMHDP’ye hazırlık sürelerini ekleyerek bu konuda ilk çalışmayı yapmış ve yeni bir problem tanımlamıştır, Sıra Bağımlı Hazırlık Süreli Montaj Hattı Dengeleme Problemi (HMHDP). Çalışmanın motivasyon kaynağı son yıllarda popüler olan HMHDP literatürüne yeni bir bakış açısı kazandırmaktır. Bu çalışmada, Andres vd. ’nin [8] tanımladığı HMHDP temel alınarak farklı bir problem tanımlanacaktır. Yeni problemde HMHDP’ye ilave olarak her iş istasyonunda işlemleri yerine getirmek için birden fazla kaynak kullanma olanağı vardır ve iş istasyonuna atanan işlem bu kaynaklardan herhangi biri ile yapılabilir. Bu durumda her iş istasyonunda işlemler için kullanılan kaynak sayısı kadar kaynağa bağımlı işlem sırası oluşacaktır. Toplam iş istasyon süresi hesaplanırken hem kullanılan kaynağa göre işlem süreleri, hem de kaynaklarda oluşan sıraya göre kaynak ve sıra bağımlı hazırlık süreleri dikkate alınmalıdır. Dolayısıyla problem, montaj hattındaki işlemlerin hangi iş istasyonunda, hangi kaynak kullanılarak ve o kaynağa hangi sırada yapılacağına karar vermek olarak ifade edilebilir. Bu yeni problem Kaynak ve Sıra Bağımlı Hazırlık Süreli Montaj Hattı Dengeleme Problemi (KHMHDP) olarak tanımlanmıştır. Bu çalışmada problemin çözümü için matematiksel model ve genetik algoritma sunulacaktır.

Makalenin bundan sonraki bölümleri şu şekilde planlanmıştır. İzleyen bölümde literatürdeki HMHDP çalışmaları verilecektir. Bölüm 3’te problem tanıtılacak, önerilen matematiksel model sunulacak ve örnek bir problemde HMHDP ve KHMHDP çözümleri yorumlanacaktır. Bölüm 4’te problem çözümü için önerilen genetik algoritma tabanlı çözüm yöntemi açıklanacaktır. Bölüm 5’te problem ile ilgili yapılan sayısal işlemler verilecek, son bölümde çalışma özetlenip gelecek çalışmalar hakkında öneriler sunulacaktır.

## 2. Literatür Taraması (Literature Review)

HMHDP ilk olarak Andres vd. ’nin [8] çalışmasında verilmiştir. Andres vd. [8] problemin çözümü için ikili doğrusal programlama modeli ve ağgözlü rasgele adaptif arama yöntemi (greedy randomized adaptive search procedure) tabanlı bir sezgisel geliştirmiştir. Literatürde bilinen BMHDP veri setine hazırlık süreleri ekleyerek yeni bir veri seti oluşturmuşlardır. Andres vd. ’nin [8] problem için

tanımladığı alt sınır, Pastor vd. [9] tarafından geliştirilmiştir. Martino ve Pastor [10] problemi genelleştirerek işlem ve iş istasyonu tabanlı sezgiseller önermiş ayrıca Andres vd. ’nin [8] veri setine eklemeler yapmıştır. Özcan ve Toklu [11] hazırlık süreli çift taraflı MHDP’ye odaklanmış, çözüm için karışık tam sayılı programlama modeli geliştirmiş ve COMSOAL yöntemini problem çözümüne uyarlamıştır. Seyed-Alagheband vd. [12], Andres vd. ’nin [8] Tip-1 için geliştirdiği matematiksel modelini Tip-2 problemine uygun hale getirmiştir. Ayrıca tavlama benzetimi tabanlı bir sezgisel önermişlerdir. Yolmeh ve Kianfar [13] da aynı problem için bir melez genetik algoritma geliştirmiştir.

Scholl vd. [14], Andres vd. ’nin [8] veri setlerinin aksine, tek hazırlık süresi kavramını iki farklı şekilde ileri ve geriye doğru hazırlık süreleri olarak tanımlamış, bu bilgilere göre yeni veri setleri oluşturmuştur. Problem çözümü için karışık ikili doğrusal matematiksel model geliştirmiş, Andres vd. ’nin [8] sunduğu sezgisel yöntemi uyarlayarak öncelik kuralı tabanlı yeni bir ağgözlü rasgele adaptif arama yöntemi önermişlerdir. Hamta vd. [15] probleme öğrenme etkisini katarak esnek işlem zamanları ve hazırlık süreleri oluşturmuş, problemi çözmek için üç amaçlı (çevrim süresinin en küçüklenmesi, düzgünlük endeksinin en küçüklenmesi ve ekipman maliyetinin en küçüklenmesi) bir matematiksel model geliştirmiştir. Ayrıca sürü optimizasyonu ve değişken komşuluk aramasını melezleyen bir metasezgisel algoritma önermişlerdir. Akpınar vd. [16] karışık MHDP’deki hazırlık sürelerine odaklanmış, problemin çözümü için karınca kolonisi ve genetik algoritmayı melezleyen bir çözüm yaklaşımı sunmuştur. Akpınar ve Baykasoglu [17, 18] hazırlık süreli karışık MHDP’nin çözümü için karışık tam sayılı doğrusal matematiksel model önermiş ve çok kolonili melez arı algoritması geliştirmiştir. Aghajani vd. [19] robotik çift taraflı MHDP’de hazırlık sürelerini ele almış, çözüm için karışık tam sayılı model ve tavlama benzetimi algoritması sunmuştur.

Esmailbeigi vd. [20] HMHDP için üç yeni matematiksel model geliştirmiştir. Modellerden ikisi iş istasyonu bazlı, diğeri çizelgeleme bazlıdır. Rabbani vd. [21] U tipi robotik montaj hatlarındaki hazırlık süreleri ile ilgilenmiş, çözüm için çevrim süresini, robot ve işlemlerin maliyetini, robot sayısını en küçükleyen bir matematiksel model sunmuş, ayrıca baskın olmayan sıralama genetik algoritması (NGSA) ve çok amaçlı parçacık sürü optimizasyonu algoritmaları önermiştir. Şahin ve Kellegöz [22] U tipi HMHDP için bir matematiksel model geliştirmiş ayrıca genetik algoritma ve tavlama benzetimi tabanlı iki metasezgisel sunmuştur. Akpınar vd. [23] problemin kesin çözümü için bir benders ayrıştırma algoritması önermiştir. Delice [24] çift taraflı HMHDP için öncelik kurallarını içeren bir genetik algoritma sunmuştur. Janardhan vd. [25] robotik montaj hatlarında robot hazırlık sürelerine odaklanmış, Andres vd. ’nin [8] matematiksel modelini kendi problemlerine uyarlamıştır. Ayrıca göçmen kuşlar (migrating birds) optimizasyonu algoritması önermişlerdir. Li vd.[26] hazırlık süreli çift taraflı robotik MHDP için matematiksel model geliştirmiş, yerel arama algoritması, sürü zekâsı (swarm intelligence) algoritması ve eş-evrimsel sürü zekâsı (co-evolutionary swarm intelligence) algoritması sunmuştur. Özcan [27] paralel montaj hatlarındaki hazırlık süreleri ile ilgilenmiş, çözüm için ikili doğrusal matematiksel model önermiş ve tavlama benzetimi algoritması geliştirmiştir. Yang ve Cheng [28] hazırlık süreli karışık modeli çift taraflı MHDP için karışık tam sayılı bir matematiksel model sunmuş, çözüm için değişken komşuluk algoritması (variable neighbourhood search) önermiştir.

## 3. Problemin Tanımlanması (Problem Definition)

Klasik MHDP’de işlemler arasındaki hazırlık süreleri işlem sürelerine eklenir ya da çok küçük olduklarından dikkate alınmaz. Andres vd.

[8] hat üzerindeki bir iş istasyonunda toplam süre hesaplanırken iş istasyonunda yapılan işlemlerin sürelerine ek olarak bir işlemde geçiş yaparken oluşan hazırlık sürelerini dikkate alır. HMHDP olarak tanımladıkları problemde, iş istasyonunda yapılan tüm işlemler her çevrim süresinde tekrarlandığı için istasyondaki son işlem ile bir sonraki çevrimdeki ilk işlem arasındaki hazırlık süresini de toplam iş istasyonu süresine eklenmesi gerektiğini belirtir. İşlemler arasındaki hazırlık sürelerinin toplam iş istasyonu süresine etkisinden dolayı klasik MHPD'deki öncelik ilişkileri, işlem süresi ve çevrim süresi bilgilerine göre işlemlerin iş istasyonlarına atanmasına ilave olarak HMHDP'de işlemlerin iş istasyonundaki iş sıraları da önem kazanır. Bu makalede sunulan problem Andres vd.'nin [8] HMHDP'nin bir uzantısıdır. HMHDP'ye getirilen ilave durum her iş istasyonunda tüm işlemleri yapabilecek yeterlilikte birden fazla farklı kaynak bulunmasıdır. Tüm iş istasyonları eşit sayıda farklı kaynak içerir. Her işlemin süresi kullanılan kaynağa göre farklıdır. İşlemler arasında hazırlık süreleri kaynaklara göre değişiklik gösterir. Hazırlık sürelerinin sıraya ve makineye bağımlı olması, bir işin hazırlık süresinin hem kendisinden önceki işe hem de atandığı makineye göre farklılık göstermesi anlamına gelmektedir [29]. Operatör, işlemi sadece bir kaynak kullanarak yapar ve aynı anda sadece tek bir işlem gerçekleştirilir. Operatör, iş istasyonundaki tüm işlemleri tek bir kaynak kullanarak yapabileceği gibi işlemleri farklı kaynaklar kullanarak da gerçekleştirebilir, herhangi bir kısıt yoktur. Bu şartlar altında işlemler iş istasyonlarına ve iş istasyonlarındaki farklı kaynaklara atandığında her bir kaynaktaki iş sıraları oluşacaktır. Dolayısıyla hem işlem süresi atan kaynakta göre farklılık gösterecek hem de işlemler arasındaki hazırlık süreleri kaynaklarda oluşan iş sıralarına göre belirlenecektir. Toplam iş istasyonu süresi, işlemlerin atandığı kaynaktaki sürelerine ve her bir kaynaktaki işlemler arası hazırlık sürelerine göre belirlenecektir. Sonuç olarak hangi işlemin, hangi istasyonda, hangi kaynaktaki, hangi sırada yapılacağı toplam iş istasyonu süresi açısından önemlidir. Makalede sunulan problem, kaynak ve sıra bağımlı hazırlık süreli montaj hattı dengeleme problemi (KHMHPD) olarak tanımlanmıştır. Amaç tüm bu bilgileri kullanarak, verilen çevrim süresi için en küçük iş istasyonu sayısını bulmaktır.

KHMHPD'nin varsayımları şunlardır:

- Öncelik ilişkileri, kaynağa bağlı işlem süreleri ve hazırlık süreleri sabit ve bilinmektedir.
- İşlem ve hazırlık süreleri iş istasyonlarından bağımsızdır, her iş istasyonunda aynıdır.
- İşlem süresi çevrim süresinden fazla olamaz.
- İş istasyonunda bulunan tüm kaynaklar tüm işlemler için kullanılabilir yeterliliktedir.
- Her iş istasyonu aynı donanıma sahiptir ve herhangi bir işlem herhangi bir iş istasyonuna atanabilir.
- Her işlem sadece bir iş istasyonunda bir kaynağa atanır. İş istasyonunda aynı anda sadece bir işlem yapılır.
- İş istasyonunda birden fazla farklı kaynak bulundurulması için herhangi bir yer ve finansman kısıtı yoktur.
- İş istasyonunda sadece aynı kaynak kullanılarak yapılan işlemler arasındaki hazırlık süreleri dikkate alınmıştır. Aynı iş istasyonunda farklı kaynaklar ile yapılan işlemler arasındaki geçişlerde hazırlık süreleri olabilir. Fakat aynı kaynakla yapılan işlemler arasındaki hazırlık sürelerine göre daha az olacağı düşünüldüğünden dikkate alınmamıştır.
- Hatta tek çeşit ürün üretilmektedir.
- Hattın hızı sabittir (paced line) ve ara stok yoktur.

### 3.1. Matematiksel Model (Mathematical model)

Makalede sunulan KHMHPD'nin çözümü için, Andres vd. 'nin [8] ikili doğrusal programlama modeli temel alınarak bir matematiksel

model geliştirilmiştir. Modelde kullanılan indis, parametre ve karar değişkenleri Tablo 1'de verilmiş, ardından matematiksel model sunulmuştur.

Amaç fonksiyonu

$$\text{Min} \sum_{j=m_{\min}+1}^{m_{\max}} j y_j \quad (1)$$

Kısıtlar

$$\sum_{j=E_i}^{L_i} \sum_{r=1}^{Rm_j} \sum_{s=1}^{Nm_j} x_{ijrs} = 1 \quad \forall i \quad (2)$$

$$\sum_{\forall i \in T_j} x_{ijrs} \leq 1 \quad \forall j; \forall r; s = 1, \dots, Nm_j \quad (3)$$

$$\sum_{\forall i \in T_j} x_{ijr,s+1} \leq \sum_{\forall i \in T_j} x_{ijrs} \quad \forall j; \forall r; s = 1, \dots, Nm_j - 1 \quad (4)$$

$$\sum_{j=E_i}^{L_i} \sum_{r=1}^{Rm_j} \sum_{s=1}^{Nm_j} (NTm_j(j-1) + s) x_{ijrs} \leq \sum_{j=E_k}^{L_k} \sum_{r=1}^{Rm_j} \sum_{s=1}^{Nm_j} (NTm_j(j-1) + s) x_{ikrs} \quad (5)$$

$$\forall (i, k) \in P \quad (5)$$

$$\sum_{\forall i \in T_j} \sum_{r=1}^{Rm_j} \sum_{s=1}^{Nm_j} t_{ir} x_{ijrs} + \sum_{\forall (i,k) \mid (i \neq k) \wedge (i,k \in T_j)} t_{ikr} z_{ikjr} \leq TC \quad j = 1, \dots, m_{\min} \quad (6)$$

$$\sum_{\forall i \in T_j} \sum_{r=1}^{Rm_j} \sum_{s=1}^{Nm_j} t_{ir} x_{ijrs} + \sum_{\forall (i,k) \mid (i \neq k) \wedge (i,k \in T_j)} t_{ikr} z_{ikjr} \leq TC y_j \quad (7)$$

$$j = m_{\min} + 1, \dots, m_{\max}$$

$$x_{ijrs} + x_{ikr,s+1} \leq 1 + z_{ikjr} \quad (8)$$

$$\forall j; \forall r; s = 1, \dots, Nm_j - 1; \forall (i, k) \mid (i \neq k) \wedge (i, k \in T_j) \wedge (k \notin PT_i)$$

$$x_{ijrs} - \sum_{\forall k \in T_j \mid (i \neq k) \wedge (k \notin PT_i)} x_{ikr,s+1} \leq w_{ijr} \quad (9)$$

$$\forall j; \forall r; s = 1, \dots, Nm_j - 1; \forall i \in T_j$$

$$w_{ijr} + x_{ikr1} \leq 1 + z_{ikjr} \quad (10)$$

$$\forall j; \forall r; \forall (i, k) \mid (i \neq k) \wedge (i, k \in T_j) \wedge (i \notin PT_k)$$

$$x_{ijrs} \in \{0, 1\} \quad \forall i; j = E_i, \dots, L_i; r = 1, \dots, Rm_j; s = 1, \dots, Nm_j \quad (11)$$

$$y_j \in \{0, 1\} \quad j = m_{\min}, \dots, m_{\max} \quad (12)$$

$$z_{ikjr} \in \{0, 1\} \quad \forall j; \forall (i, k) \mid (i \neq k) \wedge (i, k \in T_j); r = 1, \dots, Rm_j \quad (13)$$

$$w_{ijr} \in \{0, 1\} \quad \forall i; j = E_i, \dots, L_i; r = 1, \dots, Rm_j \quad (14)$$

Problem Tip-1 olduğu için matematiksel modelde verilen amaç fonksiyonu (Eş. 1) toplam iş istasyonu sayısını en küçüklemektedir. Eş. 2, her bir işlemin sadece bir iş istasyonunda bir kaynaktaki çizelgelemede bir pozisyona atanmasını sağlar. Aynı şekilde Eş. 3, her bir iş istasyonunda her bir kaynaktaki çizelgelemedeki her bir

**Tablo 1** Matematiksel modelde kullanılan indis, küme, parametre ve karar değişkenleri  
(Indices, sets, parameters, and decision variables used in the mathematical model)

İndisler	
$i, k$	İşlem
$j$	İş istasyonu
$r$	Kaynak
$s$	İş istasyonunda bir kaynakta oluşan çizelgelemedeki pozisyon
Kümeler	
$N$	İşlemler kümesi ( $i = 1, \dots, N$ )
$P_{ik}$	Birbirini takip eden ( $i, k$ ) işlemler çifti kümesi. ( $i$ işleminden hemen sonra gelen $k$ işlemi).
$PT_i$	$i$ işleminden önce yapılan tüm işlemler kümesi
$T_j$	$j$ . iş istasyonuna atanabilecek işlemler kümesi ( $E_i$ ve $L_i$ kullanılarak oluşturulur)
Parametreler	
$E_i, L_i$	$i$ işleminin en erken ve en geç atanabileceği iş istasyon numarası. ( $E_i = \left\lfloor \frac{\min\{t_{ir}\} + \sum_{k \in PT_i} \min\{t_{kr}\}}{TC} \right\rfloor$ , $L_i = m_{max} + 1 - \left\lfloor \frac{\min\{t_{ir}\} + \sum_{k \in ST_i} \min\{t_{kr}\}}{TC} \right\rfloor$ , $\min\{t_{ir}\}$ ( $i$ ( $k$ ) işleminin $r$ kaynaklarındaki en küçük işlem süresi, $ST_i$ $i$ işleminden sonra gelen tüm işlemler kümesi)
$m_{max}$	İş istasyonu sayısı için üst sınır (Önerilen Genetik Algoritma ile elde edilen toplam iş istasyonu sayısı $m_{max}$ olarak alınır. Eğer bulunan sonuç $m_{min}$ 'e eşitse $m_{max} = m_{min} + 1$ )
$m_{min}$	İş istasyonu sayısı için alt sınır
$Nm_j$	$j$ . iş istasyonuna atanabilecek maksimum işlem sayısı ( $T_j$ kümesindeki işlemler içinden $TC$ ve $\min\{t_{ir}\}$ dikkate alınarak atanabilecek maksimum işlem sayıları belirlenir)
$NTm$	Herhangi bir iş istasyonuna atanacak maksimum işlem sayısı. $NTm = \max\{Nm_j\}$ .
$Rm_j$	$j$ iş istasyonundaki maksimum kaynak sayısı.
$TC$	Çevrim süresi için üst sınır
$t_{ir}$	$i$ işleminin $r$ kaynağı kullanılarak yapılma süresi
$ts_{ikr}$	$r$ kaynağında $i$ işlemi ile $k$ işlemi arasındaki hazırlık süresi
Karar Değişkenleri	
$x_{ijrs}$	İkili karar değişkeni. Eğer $i$ işlemi $j$ . iş istasyonunda $r$ . kaynakta $s$ . sırada işlem görüyorsa 1 değeri alır. ( $i = 1, \dots, N; j = E_i, \dots, L_i; r = 1, \dots, Rm_j; s = 1, \dots, Nm_j$ ).
$y_j$	İkili karar değişkeni. Eğer $j$ iş istasyonuna işlem atanırsa 1 değeri alır. ( $j = m_{min} + 1, \dots, m_{max}$ )
$z_{ikjr}$	İkili karar değişkeni. Eğer $j$ . iş istasyonunda $r$ . kaynakta aynı çevrimde veya bir sonraki çevrimde $i$ işleminin hemen ardından $k$ işlemi yapılıyorsa 1 değeri alır. ( $j$ . iş istasyonunda $r$ . kaynakta $i$ ve $k$ işlemleri arasında hazırlık olduğunu gösterir). ( $\forall j; \forall (i, k) (i \neq k) \wedge (i, k \in T_j); r = 1, \dots, Rm_j$ )
$w_{ijr}$	İkili karar değişkeni. Eğer $i$ işlemi $j$ . iş istasyonunda $r$ . kaynakta yapılan son işlem ise 1 değerini alır. ( $\forall i; j = E_i, \dots, L_i; r = 1, \dots, Rm_j$ )

pozisyonda birden fazla işlem bulunmasını önler. Eş. 4, her bir iş istasyonunda her bir kaynaktaki çizelgeleme pozisyonlarını sıra ile oluşturur. İşlemler arasındaki öncelik ilişkileri koşulu Eş. 5 ile sağlanır. Eş. 6 ve Eş. 7, her bir iş istasyonu için iş istasyonu ve kaynaklara atanan işlemlerin kaynaklara bağlı sürelerini ve her bir kaynakta iş sırasına göre gerçekleşen kaynağa bağlı hazırlık sürelerini dikkate alarak oluşan toplam iş istasyonu süresinin çevrim süresini aşmamasını sağlar. Eş. 8 ile  $j$ . iş istasyonunda  $r$ . kaynakta  $s$ . pozisyonda  $i$  işlemi,  $(s+1)$ . pozisyonda  $k$  işlemi gerçekleşirse  $z_{ikjr}=1$  olur, bu da  $i$  ve  $k$  işlemleri arasında bir hazırlık yapıldığını belirtir.  $j$ . iş istasyonunda  $r$  kaynağındaki çizelgelemede en son pozisyona atanan  $i$  işlemi Eş. 9 ile  $w_{ijr}=1$  olarak belirlenir. Eş. 10,  $j$  iş istasyonunda  $r$  kaynağında çevrim bitişi ile bir sonraki çevrime başlanması sırasında işlemler arasında oluşan hazırlığı belirtir.  $j$  iş istasyonunda  $r$  kaynağındaki çizelgelemede  $i$  işlemi son pozisyondaydı ve yeni çevrim aynı kaynaktaki çizelgelemede ilk sıradaki  $k$  işlemi ile başlayacaksa  $z_{ikjr}=1$  olur. Eş. 11-Eş. 14, karar değişkenlerinin alacağı değerleri ifade eder.

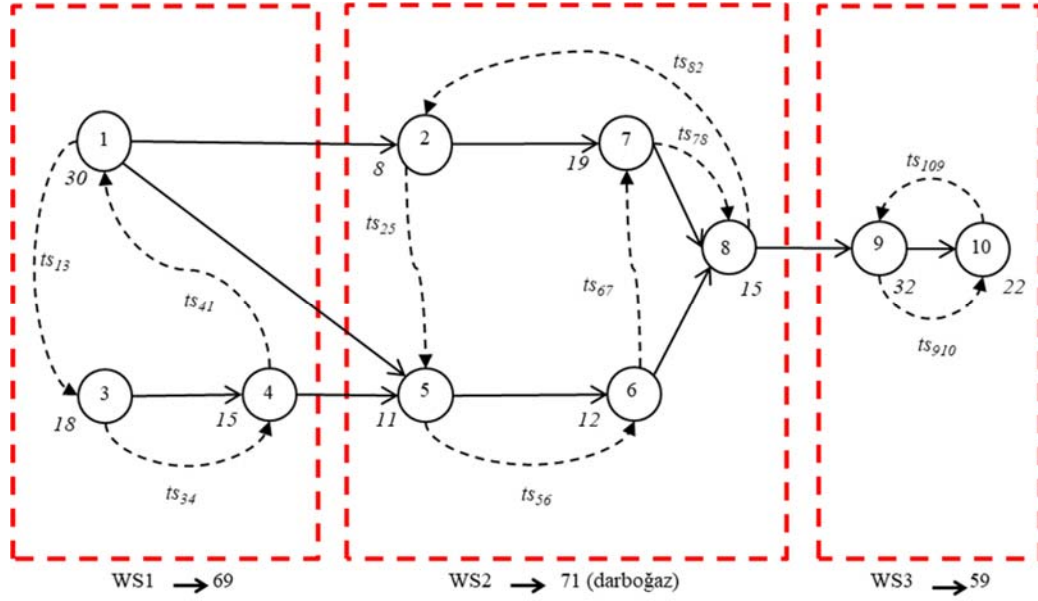
### 3.2. Örnek Problem (Example Problem)

Bu bölümde, makalede sunulan problemin klasik HMHPD'ye göre sağladığı avantajları ve önerilen matematiksel modelin sonuçlarını görmek için küçük boyutlu bir örnek verilmiştir. Seçilen örnek Andres vd. 'nin [8] HMHPD ile ilgili çalışmalarında verdiği 8 işlemlerli bir montaj hattıdır. Örneğe ait işlemler arası öncelik ilişkileri, işlem süreleri ve işlemler arasındaki hazırlık süreleri ( $ts_{ikr}$ ) Şekil 1 ve Tablo

2'de verilmiştir. HMHPD'nin çözümünde çevrim süresi, darboğaz oluşturan iş istasyonun süresi olan 71'dir [8]. Bu iş istasyonun süresi, ikinci iş istasyonuna atanmış işlemlerin süreleri ( $t_2+t_5+t_6+t_7+t_8$ ) ve bu işlemler arasında gerçekleşen hazırlık sürelerinden ( $ts_{25}+ts_{56}+ts_{67}+ts_{78}+ts_{82}$ ) oluşur.

Makalede sunulan KHMHPD'de, iş istasyonlarında birden fazla kaynak bulunur ve iş istasyonuna atanan işlem herhangi bir kaynakla yapılabilir. Örnek problemde her iş istasyonunda iki farklı kaynak olduğu varsayılmıştır. İlk kaynağa (A) ait işlem süreleri ve işlemler arasındaki hazırlık süreleri için Andres vd. 'de [8] verilen bilgiler kullanılmıştır (Şekil 1 ve Tablo 2). İkinci kaynak (B) için rasgele oluşturulan işlem süreleri ve hazırlık süreleri ( $ts_{ikB}$ ) Tablo 3'te verilmiştir. Andres vd. 'nin [8] çözümleri ile karşılaştırmak için problem, Bölüm 3.1.'de sunulan matematiksel model ile farklı çevrim süreleri için çözülmüş, üç iş istasyonunu sonucunu veren en küçük çevrim süresi belirlenmiştir.

Her iş istasyonunda iki kaynağın kullanılabilirdiği yeni durumda, önerilen matematiksel modelin çözümü Şekil 2'de verilmiştir. Çözümünden görüleceği gibi bazı işlemler A kaynağı, bazıları da B kaynağı kullanılarak yapılmaktadır. Hangi işlemin hangi kaynak kullanılarak yerine getirildiği, şekilde işlemlerin yanlarında belirtilmiştir. Ayrıca aynı iş istasyonuna atanıp aynı kaynakla yapılan işlemler arasındaki hazırlıklar da şekilde gösterilmiştir. Örneğin, ilk iş istasyonundaki bir ve üçüncü işlemler B kaynağı kullanılarak yapıldığından, bu işlemler arasında B kaynağında yapılması gereken



Şekil 1. Andres vd. 'deki [8] örnek problem ve çözümü. (Example and its solution in Andres et al. [8].)

hazırlıklar, şekilde  $ts_{13B}$  ve  $ts_{31B}$  olarak verilmiştir. Şekil 2'deki çözümde, örnek problemin tüm işlemleri yeni durumda da üç iş istasyonuna atanmış, ama çevrim süresi, darboğaz iş istasyonunun toplam süresi olan 58'e inmiştir. Bu iş istasyonunun süresi, A kaynağı ile yapılan işlemlerin süresi ( $ts_{8A}+ts_{10A}$ ), B kaynağı ile yapılan işlemin süresi ( $ts_{9B}$ ) ve A kaynağı ile yapılan işlemler arasındaki hazırlık sürelerinin ( $ts_{810A}+ts_{108A}$ ) toplamına eşittir. B kaynağı ile tek işlem yerine getirildiğinden, B kaynağında herhangi bir hazırlığa ihtiyaç yoktur. Sonuç olarak örnek problem için HMHDP'de toplam 10 hazırlık gerekirken, makalede sunulan KHMHDP'de kaynaklarda gerçekleşen toplam hazırlık sayısı 8'e düşmüştür. Hazırlık sayısının azalmış olması, hattaki toplam hazırlık süresinin azalması olasılığını arttıracaktır.

Tablo 2. Andres vd.'deki [8] örnek problemin hazırlık süreleri matrisi (Setup times matrix for the example in Andres et al. [8])

$ts_{ikA}$	1	2	3	4	5	6	7	8	9	10
1	0	3	1	3	1	1	0	1	1	1
2	2	0	2	1	0	1	1	1	3	3
3	3	2	0	2	2	1	2	2	1	1
4	3	1	3	0	2	3	1	1	2	1
5	1	1	4	1	0	0	3	2	1	2
6	1	3	1	0	2	0	1	4	1	3
7	1	1	4	1	1	1	0	4	1	0
8	1	1	1	2	1	1	1	0	2	1
9	1	1	2	1	2	2	3	1	0	3
10	3	1	1	2	3	1	3	1	2	0

#### 4. KHMHDP İçin Önerilen Genetik Algoritma (Proposed Genetic Algorithm for RSALBP)

BMHDP NP-zor problemdir dolayısıyla literatürdeki tüm BMHDP uzantıları NP-zordur [1, 3, 6]. Aynı şekilde HMHDP NP-zor olduğundan [8, 14] bu problemin genişletilmiş hali olan KHMHDP de NP-zordur. Problem boyutu büyüdükçe NP-zor problemlerde kesin sonuç bulmak zorlaşır. Bölüm 3.1.'de önerilen matematiksel model ile problem boyutu büyüdükçe optimum çözüm elde etmek güçleşir. Araştırmacılar NP-zor problemlerin çözümünde kurucu sezgiseller ya da metasezgisel tabanlı algoritmalarından faydalanır. Genetik Algoritma (GA), MHDP literatüründe problem çözümleri için en çok

kullanılan metasezgisellerden biridir [1, 3, 6]. Bu çalışmada KHMHDP çözümü için GA tabanlı bir algoritma önerilmiştir.

Tablo 3. İkinci kaynak için işlem süreleri ve hazırlık süreleri matrisi. (Task times and setup time matrix for second resource)

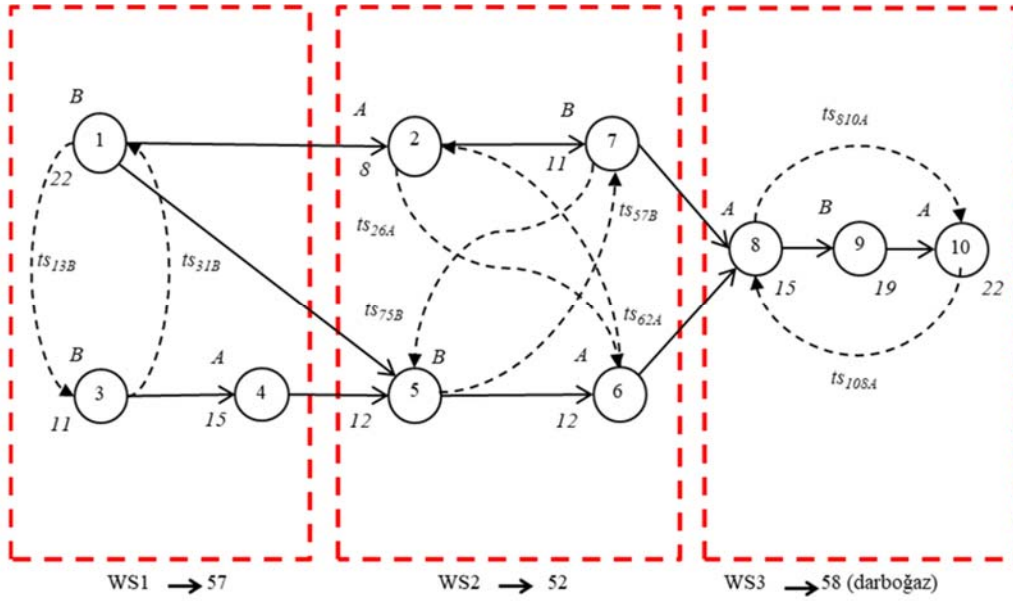
işlem	süre
1	22
2	12
3	11
4	21
5	12
6	18
7	11
8	15
9	19
10	33

#### Hazırlık süresi matrisi

$ts_{ikB}$	1	2	3	4	5	6	7	8	9	10
1	0	0	4	4	5	0	2	1	4	2
2	5	0	1	1	0	0	5	3	3	0
3	5	3	0	2	3	2	0	1	0	1
4	1	2	0	0	5	5	2	2	2	5
5	2	0	4	2	0	1	2	0	0	5
6	5	3	0	1	2	0	4	0	0	1
7	3	4	3	2	3	1	0	4	1	4
8	1	2	3	4	0	5	4	0	2	2
9	2	1	3	3	4	4	3	2	0	4
10	3	2	5	5	3	3	3	1	1	0

Önerilen GA başlangıç için rasgele bir popülasyon yaratır. Popülasyondaki her bireyde işlemlerin atama sırası ve işlemlerin hangi kaynakla yapıldığını gösteren bilgiler bulunur. Bu bilgiler kullanılarak, verilen çevrim süresine göre gerekli iş istasyonu sayısı amaç fonksiyonu olarak belirlenir. Rulet tekniği yöntemi kullanarak popülasyon içinden bireyler seçilir. Seçilen bireylere çaprazlama ve mutasyon işlemleri uygulanarak yeni popülasyonun bireyleri oluşturulur. Ayrıca elitizm uygulayarak mevcut popülasyondaki en iyi bireylerin bir sonraki popülasyona aktarılması sağlanır. Toplam tekrar





Şekil 2. Örnek problemin çözümü. (Solution of the example)

sayısı kadar yeni popülasyon oluşturma prosedürü yenilenir. Yeni popülasyonlar oluşturulurken, belli sayıda ardışık popülasyonda amaç fonksiyonu değerinde herhangi bir iyileşme olmazsa yeniden başlama mekanizması uygulanarak yeni popülasyon oluşturulur. Önerilen GA'nın algoritması Şekil 3'de verilmiştir. Algoritmanın işleyişi ve yapısı ile ilgili ayrıntılı bilgiler izleyen bölümlerde verilecektir. Her iş istasyonunda iki kaynak kullanılabileceği varsayılarak açıklamalar yapılacaktır.

#### 4.1. Gösterim (Representation)

GA'da bireyler kromozomlarda gösterilir. Önerilen GA'daki kromozom iki bölümden oluşur; işlemlerin atama sırası ve işlemler yapılırken kullanılan kaynak bilgisi. İşlem odaklı bir gösterim yapıldığından kromozomun her iki bölüm, problemdeki işlem sayısı kadar genden meydana gelir. İlk bölüm olan işlemlerin atama sırası, öncelik ilişkileri dikkate alınarak oluşturulur. Bu bölümdeki işlem sırası olurlu bir çözüm verecektir. İkinci bölüm, işlemlerin iş istasyonunda hangi kaynak kullanılarak yapıldığını gösterir. Şekil 2'de verilen problemin çözümü için örnek bir gösterim Şekil 4'de verilmiştir. Örneğin, atama sırasındaki ilk işlem olan üçüncü işlem, kromozomun ikinci bölümünde ilk gendeki bilgiye göre ikinci kaynak kullanarak yapılacaktır.

#### 4.2. Başlangıç Popülasyonu, Uygunluk Fonksiyonu ve Seçim (Initial Population, Fitness Function, and Selection)

Önerilen GA için, başlangıç popülasyonu rasgele olarak yaratılmıştır. Kromozomun ilk bölümü için rasgele oluşturulacak atama sırasının öncelik ilişkilerine uygun olurlu sıra içermesi gerekmektedir. Olurlu işlem atama sırası elde etmek için Leu vd. [30] yöntemi kullanılmıştır. Bu yöntemle göre ilk olarak önceliği olmayan işlemler belirlenir. Eğer önceliği olmayan işlem tekse o işlem olurlu atama sırasına yerleştirilir. Eğer önceliği olmayan birden fazla işlem varsa rasgele biri seçilerek olurlu atama sırasına eklenir. Her atamadan sonra öncelik ilişkileri güncellenir ve yeni duruma göre önceliği olmayan işlemler tekrar belirlenir. Yöntem bu şekilde tüm işlemler olurlu atama sırasına ekleninceye kadar devam eder. Böylece kromozomun ilk bölümü öncelik ilişkilerini sağlayan ve olurlu çözüm veren bir işlem atama sırasına sahip olur. Olurlu atama sırasının oluşturulması aşaması Algoritma 1 olarak aşağıda verilmiştir. İşlemlerin yerine

getirilmesinde kullanılan kaynak bilgilerinin yer aldığı ikinci bölümde herhangi bir kaynak kısıtı olmadığından, kaynaklar işlemlere rasgele atanır.

#### Algoritma 1

*Olurlu atama sırasının oluşturulması:*

**WHILE** atama sırasındaki işlem sayısı < problemdeki toplam işlem sayısı

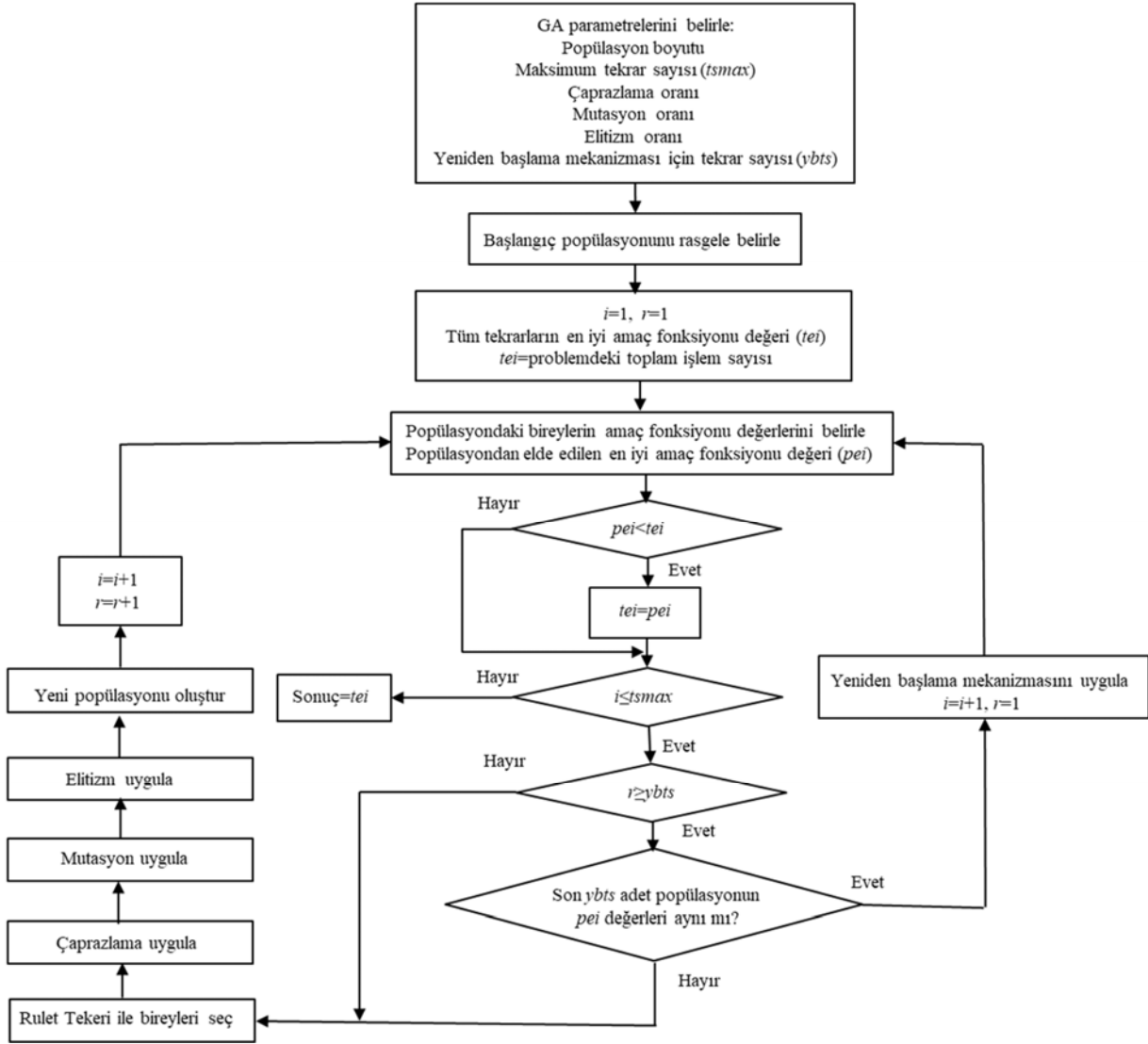
*Öncelik ilişkilerine göre atanabilecek işlemleri belirle*

*Atanabilecek işlemler arasından rasgele birini seç ve atama sırasına ekle*

*Öncelik ilişkilerini güncelle*

**END WHILE**

GA, en iyi uygunluk değerine sahip bireyi bir sonraki çözüme taşıyarak çözümünü geliştirir. Önerilen GA için uygunluk değeri, işlem atama sırası ve işlem için kullanılan kaynak bilgilerine göre bulunan toplam iş istasyonu sayısıdır. Algoritma her bir kromozom için toplam iş istasyonu sayısını şu şekilde belirler. Kromozomdaki işlem atama sırası ve hangi işlemin hangi kaynakla yapıldığı bilgisi kullanılarak işlemler tek tek iş istasyonuna atanır. İlk bölümdeki atama sırasından seçilen işlemin hangi kaynağa atanacağı bilgisi kromozomun ikinci bölümde bu işleme denk gelen genden sağlanır. İşlemin, atanacağı kaynağa ait işlem süresi ve bu kaynakta oluşacak hazırlık süresi, önceki toplam iş istasyonu süresine eklenir. Eğer toplam iş istasyonu süresi çevrim süresini aşmıyorsa ilgili işlem mevcut iş istasyonuna atanır. Eğer toplam iş istasyonu süresi çevrim süresini aşmıyorsa ilgili işlem açılan yeni iş istasyonuna yerleştirilir. Atama işleyişi kromozomun ilk bölümünde atama sırasındaki bir sonraki işlem için tekrarlanır ve atama sırasındaki tüm işlemlerin iş istasyonlarına atanması ile biter. Atama sırasındaki son işlemin yerleştirildiği iş istasyonu, verilen çevrim süresine ve kromozomdaki atama sırası ile işlem için kullanılan kaynak bilgilerine göre tüm işlemlerin yerleştirildiği toplam iş istasyonu sayısını verir. Atama sırası öncelik ilişkilerini sağlayacak şekilde oluşturulduğundan atama işleyişi sırasında sadece toplam iş istasyonu süresinin çevrim süresini aşmamasına dikkat edilir. Bireylerin popülasyondan seçilmesi rulet tekeri yöntemi [31] ile yapılır. Rulet tekeri yöntemi kullanılarak uygunluk değerleri iyi olan bireylerin bir sonraki nesle aktarılma olasılığı artırılmış olur.



Şekil 3. Önerilen GA algoritması (Algorithm for the proposed GA)

3	1	4	5	6	2	7	8	9	10	2	2	1	2	1	1	2	1	2	1
---	---	---	---	---	---	---	---	---	----	---	---	---	---	---	---	---	---	---	---

Şekil 4. Şekil 2'deki çözümün kromozom gösterimi (Representation of the chromosome for solution in figure 2)

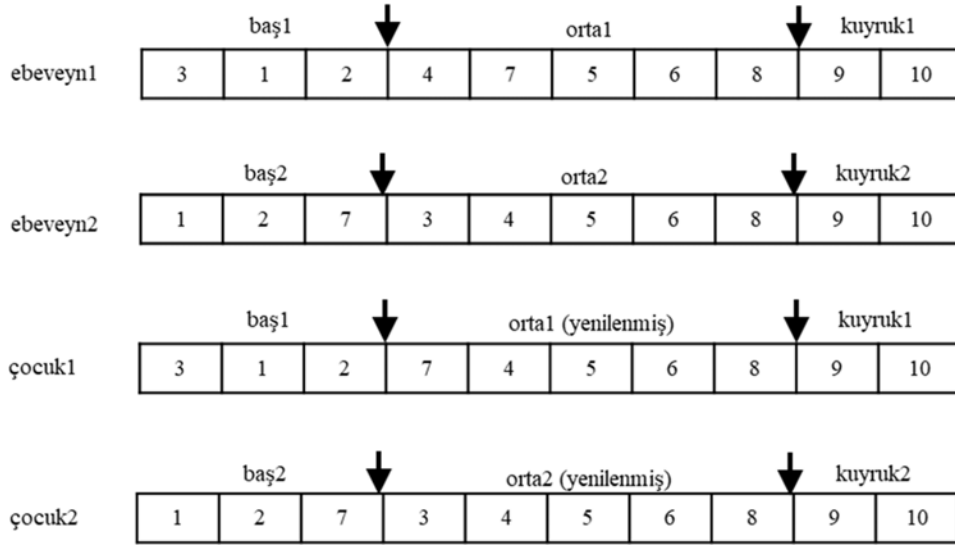
#### 4.3. Çaprazlama (Crossover)

Çaprazlama yöntemi olarak kromozomun her iki bölümüne iki noktalı çaprazlama yöntemi uygulanmıştır. Bu yöntemde kromozom üzerinde rasgele iki nokta belirlenir. Kromozom, baş, orta ve kuyruk kısmı olarak üç kısma ayrılır. Çaprazlama sonucu ilk çocuğun kromozomu, birinci ebeveynin baş ve kuyruk kısmı ile ikinci ebeveynin orta kısmından oluşur. Aynı şekilde ikinci çocuğun kromozomu da ikinci ebeveynin baş ve kuyruk kısmı ile birinci ebeveynin orta kısmından oluşmaktadır.

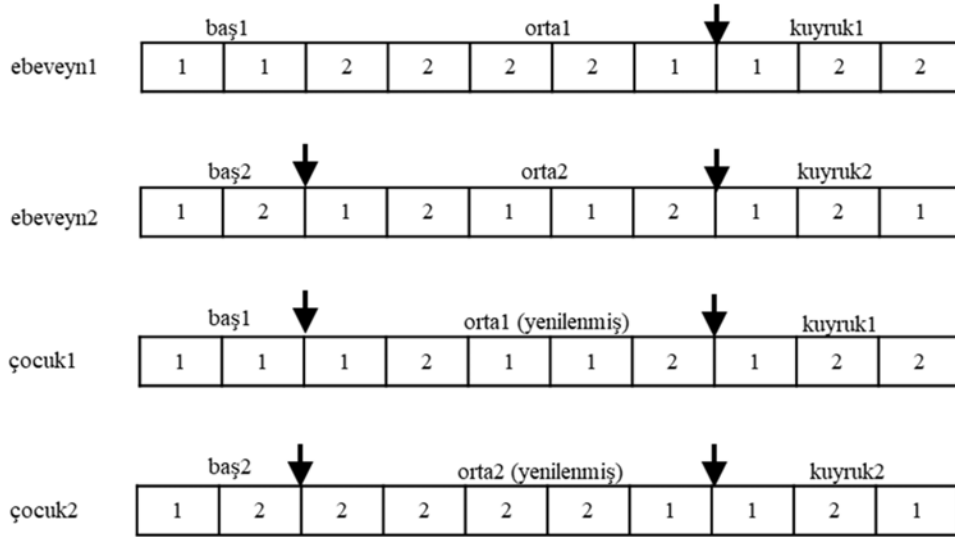
MHDP'de öncelik ilişkileri kısıtı olduğundan, yeni oluşturulan kromozomların olurlu bir çözüm verecek işlem atama sırası içermesi gerekir. Bunu sağlamak için kromozomun ilk kısmına Leu vd'nin [30] iki noktalı çaprazlama yöntemi uygulanmıştır. İlk çocuğun kromozomu ilk ebeveynin baş ve kuyruk kısımlarını içerir. Orta kısım ise ilk ebeveynin orta kısmındaki işlemlerin ikinci ebeveyndeki

pozisyonlarına göre sıralanarak oluşturulur. Benzer şekilde ikinci çocuk, ikinci ebeveynin baş ve kuyruk kısımlarını içerir, orta kısımdaki işlemler ilk ebeveyndeki pozisyonlara göre sıralanır. Şekil 5'te kromozomun ilk bölümü için bir örnek verilmiştir. Ebeveyn1'in baş1 (3-1-2) ve kuyruk1 (9-10) kısımları çocuk1'in baş1 ve kuyruk1 kısımlarına değiştirilmeden taşınır. Ebeveyn1'in orta1 kısmında kalan işlemler (4-7-5-6-8), ebeveyn2'de pozisyonlarına göre yeniden sıralanır (7-4-5-6-8) ve bu yeni sıra çocuk1'in orta kısmına orta1(yenilenmiş) olarak yerleştirilir. Çocuk2'deki kromozom sırasında orta2(yenilenmiş) kısmın ebeveyn2 orta2 kısmı ile aynı olmasının sebebi, orta2'de yer alan işlemlerin ebeveyn1'de aynı pozisyon sırasına sahip olmasıdır. Kromozomun ikinci bölümündeki kaynak bilgileri kısmında herhangi bir kaynak kullanma kısıtı olmadığı için geleneksel iki noktalı çaprazlama yöntemi uygulanmıştır. Kuyruk ve baş kısımları sabit kalıp orta kısımlar değiştirilmiştir. Kromozomun ikinci bölümü için çaprazlama örneği Şekil 6'ta verilmiştir.





Şekil 5. Kromozomun ilk bölümü için Leu vd.'nin [30] iki noktalı çaprazlama yöntemi (Leu et al.'s [30] two-point crossover method for the first part of the chromosome)



Şekil 6. Kromozomun ikinci bölümü için iki noktalı çaprazlama (Two-point crossover for the second part of the chromosome)

#### 4.4. Mutasyon (Mutation)

Önerilen GA'da kromozomun her bölümü için iki farklı mutasyon yöntemi uygulanmıştır. İlk bölüm için permütasyon (scramble) mutasyonu kullanılmıştır. Bu yöntemde, rasgele seçilen noktaya kadar kromozom yapısı korunur, seçilen noktadan sonra ise rasgele sıralama yapılır. Mutasyona uğrayan kromozomun olurlu işlem atama sırasında olması gerekir. Bu nedenle önerilen GA'da, Leu vd.'nin [30] permütasyon mutasyonu uygulanmıştır. Bu yöntemde rasgele seçilen noktaya kadar işlem atama sırası aynen korunur. Rasgele seçilen noktadan sonra, öncelik ilişkileri de dikkate alınarak kalan işlemler rasgele sıralanır. Böylece mutasyona uğrayan kromozomda olurlu işlem atama sırası korunmuş olur.

Kromozomun ikinci bölüm için ikili (binary) mutasyon uygulanmıştır. Bu yöntemde rasgele bir nokta seçilmekte, seçilen noktadaki işlem için kullanılan kaynak bilgisi değiştirilmektedir. Kaynaklar ile ilgili bir kısıt olmadığından mutasyon işlemi sonrası elde edilen yapı herhangi bir düzeltme gerektirmez.

#### 4.5. Elitizm (Elitism)

Çaprazlama ve mutasyon operatörleri uygulandıktan sonra oluşan yeni bireyler bir önceki popülasyonda elde edilen en iyi sonucu vermeyebilir. Bunu önlemek ve en iyi sonuca sahip bireylerin bir sonraki nesile aktarılmasını garantilemek için elitizm uygulanır. Elitizm, popülasyon içindeki en iyi uygunluk değerine sahip belli sayıda bireyin çaprazlama ve mutasyona uğramadan yeni nesile aktarılmasıdır. Önerilen GA'da elitizm uygulanarak popülasyondaki en iyi sonuçların bir sonraki nesilde kaybolma olasılığı önlenmiş olur.

GA'nın en büyük dezavantajlarından biri yerel optimuma takılıp prematüre sonuçlar üretmesidir. Bundan kaçınmak için yeniden başlama mekanizması oluşturularak popülasyon içindeki bireylerin farklılaştırılması amaçlanır. Böylece çözüm uzayında farklı noktalarda çözümler elde edilerek yerel optimumdan kurtulmaya çalışılır. Önerilen yeniden başlama mekanizmasında, arka arkaya belli sayıda tekrarda (nesilde) elde edilen en iyi sonuçlarda hiçbir iyileşme

gözlemlenmezse, o ana kadar oluşan popülasyon yenilenir. Yenileme işlemi yapılırken son popülasyondaki en iyi bireylerin ilk %40'ı hiçbir değişime uğramadan yeni popülasyona aktarılır. Bu en iyi bireyler mutasyona uğratılarak yeni popülasyonun ikinci %40'lık bölümü oluşturulur. Geriye kalan %20'lik kısım ise rasgele oluşturulur. GA, oluşturulan bu yeni popülasyonu kullanarak işleyişine devam eder.

## 5. Sayısal Çalışmalar (Computational Studies)

Yapılan sayısal çalışmalar dört bölümde toplanmıştır. İlk bölümde, makalede ele alınan problem için bir alt sınır tanımlanacaktır. İkinci bölümde, probleme ait herhangi bir veri seti bulunmadığından, önerilen matematiksel model ve GA'nın performansının değerlendirilmesi için bir veri seti oluşturulacaktır. Üç ve dördüncü bölümlerde ise önerilen matematiksel modelin ve GA'nın performansları test edilecektir. Bu bölümde tüm sayısal işlemler her iş istasyonunda iki farklı kaynak bulunduğu varsayımı ile yapılmıştır. Tüm sonuçlar 3,2 GHz hızlı Intel Core i5-3470 işlemcili 4GB RAM hafızalı bilgisayarda elde edilmiştir.

### 5.1. KMHDP için Alt Sınır (Lower Bound for RSALBP)

KMHDP Andres vd.'nin [8] HMHPD 'sini temel alarak geliştirildiğinden KMHDP için bir alt sınır tanımlarken HMHPD'nin alt sınırından yararlanacaktır. Scholl vd.'nin [14] HMHPD için tanımladığı alt sınırı Eş. 15 şeklindedir.

$$LB_{MP} = \min\{m \geq TLB | t_{sum} + ts_{sum}^{n+1-m} \leq TC(m)\} \quad (15)$$

Formülde,  $n$  problemdeki işlem sayısını,  $m$  iş istasyonu sayısını,  $t_{sum}$  işlem sürelerinin toplamını,  $ts_{sum}^{n+1-m}$  en küçük  $n+1-m$  adet hazırlık süresinin toplamını,  $TLB$  teorik iş istasyonu sayısını ( $t_{sum}/\text{çevrim süresi}$ ),  $TC(m)$  hattaki toplam kapasiteyi verir ( $TC(m) = m * \text{çevrim süresi}$ ). HMHPD'de hazırlık süreleri aynı iş istasyonunda gerçekleştirilen işlemler arasındaki hazırlık süreleridir. Alt sınır eşitliğinde, mümkün olan en az hazırlık sayısı dikkate alınır.  $n$  işlemden  $m-1$  adedi hattaki  $m-1$  iş istasyonuna teker teker atanırsa, geriye kalan  $n-m+1$  işlem hattaki tek iş istasyonuna atanır ve hattaki tüm işlemler arası hazırlıklar bu iş istasyonunda gerçekleşmiş olur. Bu iş istasyonunda işlemler arasındaki  $n-m$  ileriye doğru hazırlık ve bir adet de geriye doğru hazırlık (bir sonraki çevrime hazırlık) olmak üzere toplamda  $n+1-m$  hazırlık olacaktır.

Eş. 15'teki gösterime göre, işlem süreleri toplamı ve en küçük  $n+1-m$  adet hazırlık süresinin toplamı, hattaki toplam süre kapasitesini aşmamalıdır [14]. Problem Tip-1 olduğu için, verilen çevrim süresine göre en küçük iş istasyonu sayısı belirlenmelidir. Eş. 15'te, ilk olarak  $TLB$  ile bulunan  $m$  dikkate alınarak  $TC(m)$  hesaplanır. Eğer  $t_{sum} + ts_{sum}^{n+1-m} \leq TC(m)$  ise,  $m$  alt sınır olarak belirlenir. Eğer  $t_{sum} + ts_{sum}^{n+1-m} > TC(m)$  ise, bu eşitlik  $m=m+1$  için tekrarlanır.  $t_{sum} + ts_{sum}^{n+1-m} \leq m * \text{çevrim süresi}$  eşitliği sağlandığında,  $m$  alt sınır olarak belirlenir.

KMHDP için Eş. 15'ten yararlanılarak tanımlanan alt sınır Eş. 16 şeklindedir.

$$LB_K = \min\{m \geq TLB | tcom_{sum} + tscm_{sum}^{n+1-2m} \leq TC(m)\} \quad (16)$$

KMHDP'de, aynı iş istasyonunda aynı kaynak kullanılarak yapılan işlemler arasındaki hazırlıklara odaklanılmıştır. Toplam hazırlık süresi hatta olabilecek en az hazırlık sayısı ile ilişkilidir. Eğer  $m-1$  iş istasyonundaki her iki kaynağa birer işlem atanırsa toplamda  $2*m-2$  işlemin ataması yapılmış olur ve her kaynakta tek bir işlem olduğu için  $m-1$  iş istasyonunda hiçbir kaynakta hazırlık gerekmez. Atanmayan  $n-(2*m-2)$  işlem, son iş istasyonuna atanır. Bu iş istasyonunda bir işlem bir kaynakla, geriye kalan  $n-(2*m-2)-1$  işlem 566

diğer kaynakla yapılırsa, hattaki tüm  $n+1-2*m$  adet hazırlık, bu iş istasyonundaki tek kaynakta gerçekleşecektir.

KMHDP'de her iş istasyonunda iki kaynak da kullanılabilir. Bu nedenle işlemlerin iş süreleri ve hazırlık süreleri kullanılan kaynağa göre farklılık gösterecektir. Eş. 16'da, işlem süreleri için tek vektör ( $tcom$ ) ve hazırlık süresi için de tek matris ( $tscm$ ) oluşturulmuştur.  $tcom$  en küçük işlem sürelerinden ( $t_j = \min\{tA_j, tB_j\}$ ) meydana gelir. Alt sınır eşitliğindeki  $tcom_{sum}$  bu işlem sürelerinin toplamıdır.  $tscm$ , işlemler arası en küçük hazırlık sürelerinden ( $ts_{ij} = \min\{tsA_{ij}, tsB_{ij}\}$ ) oluşur.  $tscm_{sum}^{n+1-2m}$ ,  $tscm$  matrisinde işlemler arasındaki en küçük  $n-2*m+1$  adet hazırlık süresinin toplamıdır.

Eş. 15'e benzer şekilde Eş. 16'da ilk olarak  $TLB$  ile bulunan  $m$  dikkate alınarak  $TC(m)$  hesaplanır. Eğer  $tcom_{sum} + tscm_{sum}^{n+1-2m} \leq TC(m)$  ise,  $m$  alt sınır olarak belirlenir.  $TC(m)$  yeterli değilse, Eş. 16,  $m=m+1$  için tekrarlanır. Toplam kapasite aşılmadığında  $m$  alt sınır olarak belirlenir.

Alt sınır hesabı Bölüm 3.2.'de verilen örnek problem için uygulanırsa, ilk olarak  $tcom$  ve  $tscm$  oluşturulması gerekir. Tablo 4'te her iki kaynağın işlem süreleri ve hazırlık süreleri dikkate alınarak elde edilen  $tcom$  ve  $tscm$  verilmiştir. Bu bilgilere göre  $tcom_{sum} = 146$ 'dır. Çevrim süresi 58 için  $TLB$  üç iş istasyonudur.  $m=3$  için, işlemler arasında en az hazırlık sayısı beştir ( $n+1-2*m=10+1-2*3=5$ ). Beş adet en küçük hazırlık süresinin toplamı sıfırdır ( $tscm_{sum}^{10+1-2*3} = tscm_{sum}^5 = 0 + 0 + 0 + 0 + 0 = 0$ ). Eş. 16'ya göre, hattaki toplam kapasite ( $TC(m)$ ), işlem süreleri toplamını ve en küçük beş adet hazırlık süresinin toplamını karşılamaktadır ( $146 + 0 \leq 58 * 3$ ). Sonuç olarak örnek problem için alt sınır üç iş istasyonudur.

**Tablo 4.** Örnek problem için  $tcom$  ve  $tscm$  ( $tcom$  and  $tscm$  for the example problem)

$tcom$	
işlem	süre
1	22
2	8
3	11
4	15
5	11
6	12
7	11
8	15
9	19
10	22

$tscm$										
$ts_{ik}$	1	2	3	4	5	6	7	8	9	10
1	0	0	1	3	1	0	0	1	1	1
2	2	0	1	1	0	0	1	1	3	0
3	3	2	0	2	2	1	0	1	0	1
4	1	1	0	0	2	3	1	1	2	1
5	1	0	4	1	0	0	2	0	0	2
6	1	3	0	0	2	0	1	0	0	1
7	1	1	3	1	1	1	0	4	1	0
8	1	1	1	2	0	1	1	0	2	1
9	1	1	2	1	2	2	3	1	0	3
10	3	1	1	2	3	1	3	1	1	0

### 5.2. KMHDP için Test Problemleri (Test Problems for RSALBP)

Makalede ele alınan problem yeni olduğundan literatürde uygun bir veri seti yoktur. Her iki kaynak için hem işlem süreleri hem de işlemler arası hazırlık süreleri oluşturmak gerekir. Bu verilerin

tamamını rasgele oluşturmak yerine Martino ve Pastor'un [10] (MP) MHDP için oluşturdukları veri setinden yararlanılacaktır. Bir kaynak için gerekli bilgiler bu veri setinden alınacak, diğer kaynak için gerekli veriler MP verileri dikkate alınarak rasgele oluşturulacaktır.

Martino ve Pastor [10], montaj hattı literatüründe klasik MHDP için oluşturulan veri setinden [32] 16 problemin bilgilerini kullanmıştır. Hazırlık sürelerini belirlerken, problemdeki en küçük işlem süresini ( $t_{min}$ ) ve işlem sürelerinin ortalamasını ( $t_{ort}$ ) dikkate alarak uniform dağılımla  $[0 - 0,25*t_{min}]$ ,  $[0 - 0,75*t_{min}]$ ,  $[0 - 0,25*t_{ort}]$  ve  $[0 - 0,75*t_{ort}]$  aralıklarında dört farklı hazırlık süresi seviyesi tanımlanmıştır. Her bir hazırlık süresi seviyesinde rasgele 10 çevrim süresi ve işlemler arasında hazırlık süreleri yaratarak her bir problem için 40, toplamda 640 test problemi oluşturmuştur (MP veri seti).

KHMHDP için oluşturulacak veri seti için MP veri setindeki 16 problemden 8'i, literatürde problem karakteristiklerini açıklamak için kullanılan iki kritere, işlem sayısı (75'ten küçük olanlar ve 75'e eşit ve büyük olanlar) ve sıra kuvveti (order strength) (%50'den küçük olanlar ve %50'ye eşit ve büyük olanlar) değerlerine göre seçilmiştir. Sıra kuvveti, problemdeki öncelik ilişkilerini dikkate alarak hesaplanan ve problemin zorluk derecesini gösteren bir değerdir. Seçilen problemlere ait bilgiler Tablo 5'te verilmiştir.

Hazırlık sürelerinin problem çözümüne etkisini daha fazla görmek amacıyla işlem sürelerinin ortalamaları ile oluşturulan hazırlık süreleri seviyeleri kullanılacaktır. Tablo 5'te verilen problemlerin her biri için iki farklı hazırlık süresi seviyesinde, 10 farklı çevrim süresinde, toplamda 20 adet test problemi MP veri setinde mevcuttur. MP veri setinden gelen bu bilgiler kaynaklardan birine ait işlem süreleri ve işlemler arasındaki hazırlık süreleri olarak kabul edilmiştir. Diğer kaynak için işlem süreleri oluşturulurken iki kaynaktaki işlem sürelerinin birbirine tutarlı olması amaçlanmıştır. Bu nedenle ikinci kaynağın işlem süreleri, MP veri setinden alınan ilk kaynağın işlem süreleri kullanılarak uniform dağılımla  $[0,5*t_i - 1,5*t_i]$  aralığında rasgele türetilmiştir.  $t_i$ , ilk kaynak ile  $i$ . işlemin süresidir. Test problemlerine ait çevrim süresi bilgisi MP veri setinden sağlandığından, ikinci kaynak için oluşturulan işlem sürelerinin MP test problemindeki en küçük çevrim süresini aşmamasına dikkat edilmiştir. İkinci kaynağa ait hazırlık süreleri, ikinci kaynak için oluşturulan işlem süreleri dikkate alınarak uniform dağılımla  $[0 - 0,25*t_{ort}]$  ve  $[0 - 0,75*t_{ort}]$  aralıklarında rasgele türetilmiştir. Sonuç olarak 8 problem, iki farklı hazırlık süresi seviyesi ve her hazırlık süresi seviyesinde 10 farklı çevrim süresi olmak üzere toplamda 160 test problemine ait veriler oluşturulmuştur.

### 5.3. Matematiksel Modelin Performans Analizi (Performance Analysis of the Mathematical Model)

KHMHDP için önerilen matematiksel model LINGO 14 optimizasyon yazılımında modellenmiştir. İlk olarak KHMHDP için oluşturulan tüm test problemleri 7200 saniye süre limiti ile

çalıştırılmış, fakat herhangi bir sonuç elde edilememiştir. Daha sonra bir sonuç elde edebilmek amacıyla, veri setindeki en küçük boyutlu problem olan Mitchell 'in 21 işlemleri problemi için oluşturulan 20 test probleminin her biri 43200 saniye süre limiti ile çalıştırılmış, bu süre içinde olurlu çözüm elde edilirse çalışma süresi limiti 86400 saniyeye uzatılmıştır. 20 test probleminden sadece dördünde olurlu çözüm elde edilebilmiştir. Optimum çözüm elde edebilmek amacıyla, oluşturulan test problemlerinde bulunmayan fakat MHDP literatüründeki veri setinde [32] yer alan daha az sayıda işlemleri problemler seçilmiştir. 7 işlemleri Mertens, 8 işlemleri Bowman, 9 işlemleri Jaeschke, 11 işlemleri Jackson ve 11 işlemleri Mansoor problemleri için Bölüm 5.2.'de verilen işleyiş ile veriler oluşturulmuştur. Tüm problemler için her hazırlık süresi seviyesinde iki farklı çevrim süresi olmak üzere toplamda 20 yeni test problemi rasgele olarak yaratılmıştır. Önerilen matematiksel model küçük boyutlu tüm problemlerde optimum sonuç vermiştir.

Önerilen matematiksel model ile LINGO optimizasyon yazılımında elde edilen tüm sonuçlar Tablo 6'da özetlenmiştir. Probleme ait işlem sayısı (İşl. S.), hazırlık süresi seviyesi (Haz. S. sev.), çevrim süresi (Çev. S.) ve çözüm alt sınır değeri ( $LB_K$ ) bilgileri, problem için oluşturulan matematiksel modelin kısıt sayısı ve değişken sayısı (Değ. sayısı) bilgileri, matematiksel modelden elde edilen iş istasyonu sayısı (İş ist. S.) ve matematiksel modelin çözüm süresi (ÇS) Tablo 6'da ilgili sütunlarda gösterilmiştir. Ayrıca tüm problemler önerilen GA ile 10 kez çözülmüş, elde edilen sonuçların en iyi (Min.) ve ortalama değerleri (Ort.) ile ortalama çözüm süresi (Ort. ÇS) bilgisi tabloda verilmiştir. Önerilen GA'nın, matematiksel model ile elde edilen optimum ya da en iyi olurlu çözümlerle aynı sonuçları verdiği, problem boyutu büyüdükçe matematiksel modele göre çok daha kısa sürede sonuçlar elde ettiği görülmektedir.

### 5.4. Önerilen GA'nın Performans Analizi (Performance Analysis of the Proposed GA)

KHMHDP için önerilen GA Matlab 2016a programında kodlanmıştır. İlk olarak önerilen GA için parametre analizi yapılmıştır. Literatürden, farklı GA parametre seviyeleri belirlenerek tam faktöriyel deney tasarımı oluşturulmuştur. Problem olarak Tut2 probleminin yüksek hazırlık süresi seviyeli bir test problemi seçilmiş, farklı GA parametre seviyeleri için problem 10 kez çözülmüştür. Deney tasarımında kullanılan parametre seviyeleri ve deney tasarımı sonunda seçilen parametre seviyesi Tablo 7'de verilmiştir.

Seçilen parametre değerleri ile yapılan diğer denemeler sonucunda, elitizm oranı olarak %5, yeniden başlangıç mekanizması için arka arkaya benzer sonuç sayısı toplam tekrar sayısının üçte biri olarak belirlenmiştir.

160 test probleminin her biri önerilen GA ile 10 kez çözülmüş ve elde edilen en iyi sonuçlar kaydedilmiştir. Sonuçlar Tablo 8'de özetlenmiştir. Alt sınıra eşit çözüm sayısı (#sonuç=AS), yüzde olarak alt sınırdan sapma oranı (%sapma), saniye olarak ortalama çözüm

**Tablo 5.** Problemler bilgileri (Information related to problems)

Problem	İşlem sayısı	İşlem süreleri aralığı	Toplam işlem süresi	Çevrim süresi aralığı	Sıra kuvveti (%)
Arcus1	83	233-3691	75707	3923-7465	59,1
Heskiaoff	28	1-108	1024	146-329	22,5
Lutz2	89	1-10	485	12-21	77,6
Mitchell	21	1-13	125	15-39	71,7
Mukherje	94	8-171	4208	176-351	44,8
Rosenberg	25	1-13	125	14-20	71,7
Sawyer	30	1-25	324	26-74	44,8
WeeMag	75	2-27	1499	28-56	22,7

**Tablo 6.** Matematiksel model sonuçları (Mathematical model results)

Problem	Matematiksel Model							GA			
	İşl. S.	Haz. S. sev.	Çev. S.	$LB_K$	Kısıt sayısı	Değ. sayısı	İş ist. S.	ÇS (s)	Min.	Ort.	Ort. ÇS (s)
Mertens	7	Düşük	7	4	851	845	4	1	4	4	4,37
Mertens	7	Düşük	18	2	1487	633	2	1	2	2	3,96
Mertens	7	Yüksek	9	3	1126	732	3	2	3	3	4,16
Mertens	7	Yüksek	15	2	1261	591	2	20	2	2,1	4,16
Bowman	8	Düşük	20	4	776	1045	5	1	5	5	4,83
Bowman	8	Düşük	30	3	1081	900	3	4	3	3	4,42
Bowman	8	Yüksek	23	3	673	900	4	1	4	4	4,75
Bowman	8	Yüksek	28	3	1081	900	4	10	4	4	4,74
Jaesckhe	9	Düşük	9	4	874	1265	4	10	4	4	5,47
Jaesckhe	9	Düşük	18	2	1458	921	2	17	2	2	4,99
Jaesckhe	9	Yüksek	7	5	629	1518	5	1	5	5	5,72
Jaesckhe	9	Yüksek	16	2	1052	867	3	1	3	3	5,34
Jackson	11	Düşük	10	5	2721	2250	6	414	6	6	7,42
Jackson	11	Düşük	21	3	4047	1676	3	482	3	3	6,77
Jackson	11	Yüksek	13	4	3054	1985	5	14415	5	5	7,07
Jackson	11	Yüksek	22	3	4547	1764	3	1066	3	3	6,33
Mansoor	11	Düşük	48	4	3904	2205	4	4	4	4	6,42
Mansoor	11	Düşük	65	3	4873	1852	3	202	3	3	6,18
Mansoor	11	Yüksek	51	3	2889	1764	4	575	4	4	6,58
Mansoor	11	Yüksek	94	2	4878	1455	2	516	2	2	6,05
Mitchell	21	Düşük	26	4	12148	6725	5*	86400	5	5	17,21
Mitchell	21	Düşük	18	6	11208	8827	7*	86400	7	7	18,39
Mitchell	21	Düşük	35	3	15105	5716	4*	86400	4	4	16,36
Mitchell	21	Düşük	26	4	12148	6725	5*	86400	5	5	16,56

\*86400 saniye sonunda elde edilen en iyi iş istasyonu sayısı

**Tablo 7.** Parametre analizi (Parameter analysis)

Parametre	Seviyeler	Seçilen seviye
Çaprazlama oranı	0,7 – 0,8 – 0,9	0,9
Mutasyon oranı	0,01 – 0,1 – 0,2	0,1
Popülasyon büyüklüğü	100 – gen sayısı – 2*(gen sayısı)	2*(gen sayısı)
Tekrar sayısı	500 – 5*(gen sayısı)	5*(gen sayısı)

**Tablo 8.** Sonuçlar (Results)

	Tüm sonuçlar	İşlem sayısı		Sıra kuvveti		Hazırlık süresi seviyesi	
		$tn < 75$	$tn \geq 75$	$OS < 0,50$	$OS \geq 0,50$	Düşük	Yüksek
#sonuç=AS	13	13	0	5	8	12	1
%sapma	17,99	15,81	20,18	18,39	17,60	14,74	21,25
Ort. ÇS (s)	239,18	24,93	453,42	244,37	233,98	234,70	243,65

süresi (Ort. ÇS) bilgileri performans ölçütü olarak kullanılmıştır. Önerilen GA 13 test probleminde alt sınır ile aynı sonucu vermiştir. Ortalama sapma değeri %17,99 olmuştur. En büyük sapma %30,77'dir. Ortalama çözüm süresi 239,38 saniyedir. Çözüm süresi 15,67 – 653,31 saniye aralığındadır.

Ayrıca sonuçlar test problemlerin karakteristik bilgilerine göre, problemdeki işlem sayısına göre ( $tn < 75$ ,  $tn \geq 75$ ), problemin sıra kuvvetine ( $OS > 0,50$ ,  $OS \geq 0,50$ ) ve hazırlık süresi seviyesine (düşük, yüksek) göre sınıflandırılmıştır. Önerilen algoritmanın, işlem sayısı 75'ten küçük problemlerde ve hazırlık süresi seviyesi küçük problemlerde daha iyi sonuçlar verdiği gözlenmiştir. Sıra kuvveti sınıfları sonuçları fazla etkilememiştir. %sapma değerleri için sınıflar bazında  $\alpha = 0,05$  için 2-sample t-testi uygulanmış, bu çıkarımlar istatistiksel olarak da gözlenmiştir (Tablo 9). Tablo 8'de özellikle işlem sayısına göre çözüm sürelerinin arttığı görülmektedir. GA parametrelerinden hem popülasyon büyüklüğü hem de tekrar sayısı kromozomdaki gen sayısına göre seçildiğinden, kromozomdaki gen sayısı da problemdeki işlem sayısının iki katı olduğundan, problemdeki işlem sayısı arttıkça çözüm süresinin artması doğaldır.

**Tablo 9.** %sapma için 2-sample t-testi (2-sample t-tests for % deviation)

Sınıflandırma (alt sınıflandırma)	P-değeri
İşlem sayısı ( $tn < 75$ , $tn \geq 75$ )	0
Sıra kuvveti ( $OS < 0,50$ , $OS \geq 0,50$ )	0,156
Hazırlık süresi seviyesi (Düşük, Yüksek)	0

Bu çalışmada ele alınan KHMHPD'nin, Andres vd.'nin [8] HMHPD'ye göre, iş istasyonlarına atanan işlemler arasında hazırlık sayısını azaltması beklenmektedir. Bu azalma toplam iş istasyon süresine yansıtacak, böylece iş istasyonunda tek kaynak yerine iki kaynak kullanımı montaj hattında gerekli olan toplam iş istasyonu sayısını azaltacaktır. Bu durumun gerçekleştiğini gözlemlemek için, oluşturulan 160 test problemi aynı GA parametre bilgileri kullanılarak hem ilk hem de ikinci kaynağa göre tek tek çözülmüştür. İki çözüm arasındaki en iyi çözüm, makalede ele alınan problem için bulunan çözüm ile karşılaştırılmıştır. Böylece iki kaynak kullanılarak problemdeki iş istasyonu sayısındaki iyileşmeler gözlemlenmiştir. Elde edilen sonuçlar Tablo 10'da verilmiştir. Performans kriterleri

**Tablo 10.** İyileşme oranları (Improvement rates)

	Tüm sonuçlar	İşlem sayısı		Sıra kuvveti		Hazırlık süresi seviyesi	
		$t_n < 75$	$t_n \geq 75$	$OS < 0,50$	$OS \geq 0,50$	Düşük	Yüksek
#iyileşme	116	36	80	55	61	51	65
%ort. iyileşme	8,74	6,83	10,65	6,16	11,31	6,54	10,93
%min-max	0 – 25,42	0 – 23,08	1,70 – 25,42	0 – 22,22	0 – 25,42	0 – 20,41	0 – 25,42
ort. AİS	2,11	0,45	3,64	1,48	2,75	1,59	2,64
min-max AİS	0 – 15	0 – 1	1 – 15	0 – 9	0 – 15	0-10	0 – 15

olarak iyileşme görülen test problemi sayısı (#iyileşme), % olarak problemlerdeki iyileşmelerin ortalaması (%Ort. iyileşme) ve aralık değeri (%min.-max), problemlerde azalan iş istasyonları sayısının ortalaması (Ort. AİS) ve aralık değeri (min.-max AİS) belirlenmiştir.

160 test probleminin 116'sında hattaki iş istasyonu sayısında azalma görülmüştür. En büyük iyileşme 15 iş istasyonu azalışıdır. 160 test problemi için ortalama azalan iş istasyonu sayısı 2,11 olarak gözlenmiştir. % olarak bakıldığında ortalama iyileşme %8,74, en büyük iyileşme ise %25,42'dir. Test problemlerinin karakteristiklerine göre bakıldığında, tüm sınıf ve seviyelerinde iyileşmeler olduğu görülebilir. Sınıf bazlı en yüksek iyileşme oranları, işlem sayısı 75 ve üzeri problemlerde, sıra kuvveti %50 ve üzeri olan problemlerde ve hazırlık süresi seviyesi yüksek olan problemlerde gerçekleşmiştir. %ort. iyileşme değerleri için sınıflandırmalar bazında  $\alpha=0,05$  için 2-sample t-testi uygulanıp bu çıkarımlar istatistiksel olarak da gözlenmiştir (Tablo 11).

**Tablo 11.** % ort. iyileştirme için 2-sample t-testi (2-sample t-tests for % avg. improvement)

Sınıflandırma (alt sınıflandırma)	P-değeri
İşlem sayısı ( $t_n < 75$ , $t_n \geq 75$ )	0,001
Sıra kuvveti ( $OS < 0,50$ , $OS \geq 0,50$ )	0
Hazırlık süresi seviyesi (Düşük, Yüksek)	0

## 6. Sonuçlar (Conclusions)

Bu çalışmada, montaj hattındaki işlemlerin hangi iş istasyonuna, iş istasyonundaki hangi kaynağa ve o kaynaktan hangi iş sırasına atıldığına odaklanılmıştır. Eş zamanlı çözülen bu üç durum sonunda, iş istasyonlarında ilgili kaynaklarda oluşan sıra bağımlı hazırlık süreleri dikkate alınarak MHDP çözülmüştür. Ele alınan problem ile Andres vd.'nin [8] HMHDP arasındaki temel fark, bir iş istasyonunda birden fazla kaynak kullanılmasıdır. İki problem arasındaki benzerlikler ve farklılıklar sunulmuş, ele alınan problem için ikili doğrusal matematiksel model ve GA önerilmiştir. Problem, literatürde yeni olduğu için test problemleri oluşturulmuş, bu problemlerin kesin çözümleri bilinmediğinden KHMHDP için alt sınır tanımlanmıştır.

Geliştirilen matematiksel model ile küçük boyutlu problemlerde optimum çözümler elde edilmiş, bu problemlerde önerilen GA ve matematiksel model aynı sonuçları vermiştir. Önerilen GA oluşturulan 160 test problemine uygulanmış, sonuçlar farklı problem karakteristiklerine göre yorumlanmıştır. Tüm sonuçlar için alt sınırdan ortalama sapma %17,99 bulunmuş, özellikle işlem sayısı 75'ten az problemlerde ve düşük hazırlık süreli problemlerde daha iyi sonuçlar elde edilmiştir. Tüm sonuçlar ortalama 239,15 saniyede bulunmuştur. Ayrıca test problemlerinde KHMHDP'nin HMHDP'ye göre toplam iş istasyonu sayısında sağladığı iyileştirmeler araştırılmıştır. 160 test probleminin 116'sında daha az sayıda toplam iş istasyonu ile hat dengelenmiştir. Ortalama iyileştirme 2,11, en yüksek iyileştirme 15 iş istasyonu olmuştur. İyileştirmeler yüzde olarak ortalama %8,74, en büyük %25,42 olarak gözlemlenmiştir. İşlem sayısı 75 ve üzeri problemlerde, sıra kuvveti %50 ve üzeri problemlerde ve hazırlık süresi yüksek olan problemlerde, iyileştirme

oranları daha fazla gerçekleşmiştir. Sonuç olarak hem önerilen GA'nın iyi sonuçlar ürettiği hem de KHMHDP'nin HMHDP'ye göre yüksek iyileştirme oranları elde ettiği söylenebilir.

Bu makalede ele alınan problemin yeni olması ve elde edilen sonuçlar gelecekte yapılabilecek çalışmaları çeşitlendirmektedir. Probleme ait çeşitli varsayımlar esnetilerek yeni çalışma konuları belirlenebilir. Örneğin, iş istasyonunda kaynaklar arası geçişlerdeki hazırlık süreleri de dikkate alınabilir, bazı işlemlerin sadece belirli kaynaklarda yapılabilmesi kısıt eklenebilir, bazı iş istasyonlarında fiziki kısıtlardan dolayı belirli sayıda kaynak kullanımına izin verilebilir. Bu çalışmada problem, düz montaj hattında incelendiğinden, problem, U tipi montaj hatlarına, karışık montaj hatlarına uyarlanabilir. Benzer şekilde literatürde var olan MHDP çeşitleri de bu probleme uyarlanabilir. Bir başka çalışma konusu da problemin amaç fonksiyonunda yapılacak düzenlemeler olabilir. Sadece toplam iş istasyonu sayısı yerine montaj hattı ile ilgili diğer performans kriterleri, maliyet kavramı eklenerek problem çok amaçlı olarak düzenlenebilir.

## Kaynaklar (References)

1. Becker C. ve Scholl A., A Survey on Problems and Methods in Generalized Assembly Line Balancing, *Eur. J. Oper. Res.*, 168 (3), 694-715, 2006.
2. Scholl A. ve Becker C., State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, *Eur. J. Oper. Res.*, 168 (3), 666-693, 2006.
3. Boysen N., Fliedner M., Scholl A., A classification of assembly line balancing problems, *Eur. J. Oper. Res.*, 183 (2), 674-693, 2007.
4. Boysen N., Fliedner M., Scholl A., Assembly line balancing: Which model to use when?, *Int. J. Prod. Econ.*, 111 (2), 509-528, 2008.
5. Rashid M.F.F., Hutabarat W., Tiwari A., A review on assembly sequence planning and assembly line balancing optimisation using soft computing approaches, *Int. J. Adv. Manuf. Technol.*, 59 (1-4), 335-349, 2012.
6. Battaia O. ve Dolgui A., A taxonomy of line balancing problems and their solution approaches, *Int. J. Prod. Econ.*, 142 (2), 259-277, 2013.
7. Sivasankaran P. ve Shahabudeen P., Literature review of assembly line balancing problems, *Int. J. Adv. Manuf. Technol.*, 73 (9), 1665-1694, 2014.
8. Andres C., Miralles C., Pastor R., Balancing and scheduling tasks in assembly lines with sequence-dependent setup times, *Eur. J. Oper. Res.*, 187 (3), 1212-1223, 2008.
9. Pastor R., Andres C., Miralles C., Corrigendum to balancing and scheduling tasks in assembly lines with sequence-dependent setup, *Eur. J. Oper. Res.*, 201 (1), 336, 2010.
10. Martino L. ve Pastor R., Heuristic procedures for solving the general assembly line balancing problem with setups, *Int. J. Prod. Res.*, 48 (6), 1787-1804, 2010.
11. Özcan U. ve Toklu B., Balancing two-sided assembly lines with sequence-dependent setup times, *Int. J. Prod. Res.*, 48 (18), 5363-5383, 2010.
12. Seyed-Alagheband S., Ghomi S.F., Zandieh M., A simulated annealing algorithm for balancing the assembly line type II problem with sequence-dependent setup times between tasks, *Int. J. Prod. Res.*, 49 (3), 805-825, 2011.
13. Yolmeh A. ve Kianfar, F., An efficient hybrid genetic algorithm to solve assembly line balancing problem with sequence-dependent setup times, *Comput. Ind. Eng.*, 62 (4), 936-945, 2012.

14. Scholl A., Boysen N., Flidner M., The assembly line balancing and scheduling problem with sequence-dependent setup times: problem extension, model formulation and efficient heuristics., *OR Spectrum*, 35 (1), 291–320, 2013.
15. Hamta N., Ghomi S.F., Jolai F., Shirazi M.A., A Hybrid PSO Algorithm for a Multi-Objective Assembly Line Balancing Problem with Flexible Operation Times, Sequence-Dependent Setup Times and Learning Effect, *Int. J. Prod. Econ.*, 141 (1), 99–111, 2013.
16. Akpınar S., Bayhan G.M., Baykasoglu A., Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks, *Appl. Soft. Comput.*, 13 (1), 574–589, 2013.
17. Akpınar S. ve Baykasoglu A., Modeling and solving mixed-model assembly line balancing problem with setups. Part I: a mixed integer linear programming model, *J. Manuf. Sys.*, 33 (1), 177–187, 2014.
18. Akpınar S. ve Baykasoglu A., Modeling and solving mixed-model assembly line balancing problem with setups. Part II: a multiple colony hybrid bees algorithm, *J. Manuf. Sys.*, 33 (4), 445–461, 2014.
19. Aghajani M., Ghodsi R., Javadi B., Balancing of robotic mixed-model two-sided assembly line with robot setup times, *Int. J. Adv. Manuf. Technol.*, 74, 1005–1016, 2014.
20. Esmailbeigi R., Naderi B., Charkhgard P., New formulations for the setup assembly line balancing and scheduling problem, *OR Spectrum*, 38, 493-518, 2016.
21. Rabbani M., Mousavi Z., Farrokhi-Asl H., Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem, *Journal of Industrial and Production Engineering*, 33 (7), 472-484, 2016.
22. Şahin M. ve Kellegöz T., Increasing production rate in U-type assembly lines with sequence-dependent set-up times, *Eng. Optim.*, 49 (8), 1401-1419, 2017.
23. Akpınar S., Elmi A., Bektaş T., Combinatorial Benders cuts for assembly line balancing problems with setups, *Eur. J. Oper. Res.*, 259 (2), 527-537, 2017.
24. Delice Y., A genetic algorithm approach for balancing two-sided assembly lines with setups, *Assembly Automation*, 39 (5), 827-839, 2019.
25. Janardhanan M.N., Li, Z., Bocewicz, G., Banaszak, Z., Nielsen, P., Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times, *Appl. Math. Modell.*, 65, 256-270, 2019.
26. Li Z., Janardhanan M.N., Tang Q., Ponnambalam S.G., Model and metaheuristics for robotic two-sided assembly line balancing problems with setup times, *Swarm Evol. Comput.* 50, 2019.
27. Özcan U., Balancing and scheduling tasks in parallel assembly lines with sequence-dependent setup times, *Int. J. Prod. Econ.*, 213, 81-96, 2019.
28. Yang, W. ve Cheng, W., Modelling and solving mixed-model two-sided assembly line balancing problem with sequence-dependent setup time, *Int. J. Prod. Res.*, 58 (21), 6638-6659, 2020.
29. Saraç T., Tutumlu B., A mix integer programming model and solution approach to determine the optimum machine number in the unrelated parallel machine scheduling problem, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 37 (1), 329-345, 2022.
30. Leu Y.Y., Matheson L.A., Rees L.P., Assembly line balancing using genetic algorithms with heuristic generated initial populations and multiple criteria, *Decis Sci*, 15, 581–606, 1994.
31. Holland J.N., *Adaption in natural and artificial systems*, University of Michigan Press, A.B.D., 1975.
32. Scholl A. ve Voß, S., Simple assembly line balancing-heuristic approaches, *Journal of Heuristics*, 2, 217–244, 1996.