

# STOKASTİK MONTAJ HATLARININ KISIT PROGRAMLAMA VE KAPALI KUYRUK AĞLARI İLE DENGELENMESİ

**Hacı Mehmet ALAĞAŞ, Mustafa YÜZÜKIRMIZI, Ahmet Kürşad TÜRKER**

Kırıkkale Üniversitesi, Mühendislik Fakültesi, Endüstri Mühendisliği Bölümü, 71451 Yahşihan, Kırıkkale  
[hmalagas@kku.edu.tr](mailto:hmalagas@kku.edu.tr), [myuzukirmizi@kku.edu.tr](mailto:myuzukirmizi@kku.edu.tr), [kturker@kku.edu.tr](mailto:kturker@kku.edu.tr)

(Geliş/Received: 11.12.2012; Kabul/Accepted: 18.04.2013)

## ÖZET

Montaj hatları, akış tipi üretim sistemlerinin önemli parçası olarak otomotiv, dayanıklı tüketim mamulleri, konfeksiyon vd. sanayilerde çok yaygın olarak kullanılmaktadır. Bu çalışmada, Tip-2 olarak tanımlanan, istasyon sayısının sabit kabul edilip çevrim süresini minimize edildiği montaj hatlarında optimum görev atamasının bulunması amaçlanmıştır. İşlem sürelerinin belirli bir olasılık dağılımına uygun varsayıldığı stokastik montaj hatları ele alınmıştır. Kısıt programlama ve Kuyruk ağları kullanarak optimum çözümü veren yeni bir algoritma önerilmiştir. Bu algoritmada, Kısıt programlama metodu ile muhtemel görev atama kombinasyonlarını belirlenmiş ve Kuyruk Ağı modeli ile performansları değerlendirilerek en iyi sonuçlar elde edilmiştir. Önerilen metod literatürde bulunan deney setleri ile test edilmiş, yöntemin uygulanabilirliği kanıtlanmıştır.

**Anahtar kelimeler:** Tip-2 montaj hattı dengeleme, kısıt programlama, kuyruk ağları

## BALANCING STOCHASTIC ASSEMBLY LINES USING CONSTRAINT PROGRAMMING AND QUEUEING NETWORKS

### ABSTRACT

As an important component of flow type production systems, assembly lines are widely used from automotive, appliance to apparel industry. In this research, the aim is finding the optimum task assignment that minimizes the cycle time under the assumption that station quantities are constant which are described as Type-2 assembly lines. Stochastic assembly lines are considered in which task times are distributed according to a statistical distribution. A new algorithm which gives optimum solution using Constraint Programming and Queuing Network is proposed. In this algorithm, the possible combinations are determined by Constraint Programming, and then, the performance measures are evaluated by Queuing Network. The method is tested with several numerical experiments from literature and the applicability is confirmed.

**Keywords:** Type-2 assembly line balancing, constraint programming, queuing networks

### 1. GİRİŞ (INTRODUCTION)

Montaj hattı seri üretim sistemlerinde önemli bir yer tutmaktadır. Bir ürünün elde edilmesi için gerekli görevlerin, öncelik ilişkileri, atama kısıtları ve çevrim zamanı gibi kısıtlar altında gruplandırılarak oluşturulan istasyonların bir hat boyunca sıralanmasıyla meydana gelen sisteme Montaj hattı ve ürünün oluşumu sırasında yapılması gereken görevlerin belirlenen kısıt şartları altında bir performans ölçüsünü eniyileyecek şekilde istasyonlara atanmasına Montaj Hattı Dengeleme Problemi (MHDP) denir [1, 2].

Montaj hatlarında daha fazla ayrıştırılması mümkün olmayan en küçük iş ögesi görev olarak adlandırılır. Bir görevin iş istasyonunda tamamlanması için geçen süre görev zamanı olarak tanımlanır. Genel olarak, MHDP'de istasyonlarda oluşan boş zamanları minimize ederek hattın performansının iyileştirilmesi amaçlanmaktadır. MHDP'nin çözümünde dikkate alınan amaçlar ve buna bağlı olarak yapılan sınıflandırma şu şekildedir[3]:

- Sabit çevrim zamanı ile istasyon sayısının minimize edilmesi amaçlanan Tip 1,
- Sabit istasyon sayısı ile çevrim zamanının minimize edilmesi amaçlanan Tip 2,

- Belirli bir şekilde formülize edilmiş etkinlik oranının maksimize veya minimize edilmesi amaçlanan Tip E,
- Doğrudan optimum görev ataması ile hat dengesini kurmayı amaçlayan Tip F.

Montaj hatları yerleşim tiplerine, üretilen model çeşitliliğine, görev zamanlarına, gecikmeli ve gecikmesiz hatlar gibi özelliklerine göre çeşitli sınıflara ayrılmaktadır. Belirtilen hat sınıflarına ilişkin detaylar için Scholl[4]'e bakılabilir.

MHDP ilk olarak Bryton[5] tarafından tanımlanmıştır. Daha sonrasında, Salveson[6] montaj hattı dengeleme problemi için 0-1 tam sayılı modelini geliştirmiştir. Bu çalışmalar Tip-1 MHDP için temel teşkil etmektedir. Tip-2 MHDP için ise ilk çalışma Helgeson ve Bernie[7]'nin yaptığı çalışmadır. Pozisyon ağırlığı yöntemi tabanlı bir yaklaşımı önermişlerdir. Uğurdağ vd.[8] ve Carnahan vd.[9] çalışmalarında literatürde Tip-2 MHDP çözümüne yönelik yöntemler önermişlerdir.

Son yıllara bakıldığında ise Tip 2 MHDP ile ilgili şu çalışmalar bulunmaktadır. Liu vd.[10] stokastik görev zamanlı tip-2 montaj hattı için üç aşamalı bir yöntem önermişlerdir. Pastor ve Ferrer[11] görevlerin atanabileceği iş istasyonlarını, iş istasyonlarının üst sınırı veya çevrim sürelerinin üst sınırına göre belirledikleri yeni bir matematiksel model önermişlerdir. Wei ve Chao[12] hat etkinliğini artırmayı amaçlayan tip-E montaj hattı dengeleme problemi için bir matematiksel model önermişlerdir. Bu modelde tip-2 problemini çözen bir algoritma kullanmışlardır. Nourmohammadia ve Zandieh[13] genetik algoritma tabanlı bir algoritmayı kullanarak çevrim süresini ve düzgünlük endeksinin birlikte minimizasyonunu amaçlamışlardır.

MHDP'ler, işlem zamanları dikkate alındığında ise deterministik ve stokastik görev zamanlı MHDP olarak ikiye ayrılmaktadır. Görev sürelerinin kesin olarak bilindiği ve çözüm sürecinde değişmediği hatlar deterministik montaj hatları olarak literatüre girmiştir. Stokastik montaj hatları ise, görev sürelerinin değişkenliğinin de dikkate alındığı sürelerin belirli bir olasılık dağılımıyla ifade edilebildiği sistemlerdir. Bu hatlar ise stokastik montaj hattı olarak literatüre girmiştir. Deterministik montaj hatları literatürde daha çok ele alınmıştır ve NP-zor bir problem olduğu bilinmektedir.[4] Yine literatüre bakıldığında, stokastik görev zamanlı durumlar daha karmaşık olmasından dolayı deterministik görev zamanlı durumlara göre daha az çalışılmıştır.

Bilindiği kadarıyla, stokastik görev zamanını dikkate alan ilk çalışma Modie ve Young[14] tarafından yapılmıştır. Bu çalışmada, görev zamanlarının normal dağılıma uyduğu durum için istasyon sürelerinin

çevrim süresinin aşmaması olasılığının belirli bir değerden büyük ya da eşit olması şeklinde görevler istasyonlara atanmıştır. Literatürde tek modellenmiş düz stokastik görev zamanlı MH için Gökçen ve Baykoç[15], Ağpak ve Gökçen[16] ve Ayazi vd[17] yaptıkları çalışmalar bulunmaktadır. MHDP hakkında daha fazla çalışmaya ulaşmak için Becker ve Scholl[18]; Scholl ve Becker[3]; Boysen vd,[19]; Boysen vd,[20] yaptıkları tarama makaleleri incelenebilir.

Kısıt programlama ile yapılan çalışmalarda ise Topaloğlu vd.[21] çalışmaları bulunmaktadır. Çalışmalarında alternatif öncelik ilişkilerine sahip montaj hattı için kural tabanlı bir model önermişlerdir. Modeli tam sayılı programlamayı ve kısıt programlamayı kullanarak çözmüşlerdir. Sonuç olarak kısıt programlamanın çözüme ulaşmada daha etkin olduğunu göstermişlerdir.

Bu çalışmada stokastik, düz, tek modellenmiş Tip-2 MHDP için kapalı kuyruk ağı ve kısıt programlama modeli temelli yeni bir çözüm algoritması geliştirilmiştir. Görev zamanlarının üssel dağılıma uyduğu ve görevlerin tüm öncelik ilişkilerini sağlaması koşulu ile çıktı hızını en büyüleyecek şekilde istasyonlara atanması amaçlanmıştır. Yapılan literatür araştırmasından belirlenebildiği kadarıyla bu çalışma, MHDP'nin çözümünde kısıt programlama ve kuyruk ağları modellerini birlikte kullanan ilk çalışma olması dolayısı ile özgündür. Önerilen çözüm yönteminde kısıt programlama yaklaşımı ile görev atamaları belirlenirken, kuyruk ağları modelleri kullanılarak üretilen çözümlere ilişkin performans ölçütü değerleri hesaplanmaktadır.

Çalışma giriş bölümünden sonra şu bölümlerden oluşmaktadır: İkinci bölümde kısıt programlama metodu açıklanmıştır. 3. bölümde kapalı kuyruk ağı modelinden ve ortalama değer analizinden bahsedilmiştir. 4. bölümde önerilen yöntem açıklanmıştır ve algoritma akış şeması verilmiştir. Örnek bir problem için çözüm süreci detaylı olarak gösterilmiştir. Daha sonrasında, literatürden belirlenen test problemleri ile yapılan deneysel çalışmalar ve sonuçları verilmiştir. Son bölümde ise sonuçlar ve gelecekte yapılabilecek çalışma önerileri verilmiştir.

## 2. KISIT PROGRAMLAMA METODU (CONSTRAINT PROGRAMMING APPROACH)

Bu çalışmada önerilen algoritmada, öncelik ilişkilerine bağlı olarak muhtemel atama kombinasyonları Kısıt Programlama yaklaşımı ile belirlenmiştir. Kısıt programlama hesaplama ve mantığa dayalı tekniklerin kombinasyonuna dayanan bir programlama yaklaşımıdır. Kısıtlar, her birisi tanımlanan bir alandaki değerlerden birisini alabilen değişkenler arasındaki ilişkileri ifade etmek amacıyla kullanılan formlardır. Kısıt programlama, tanımlı alanlardan oluşan değişkenlerin kombinasyonu olarak

tanımlanabilir. Bu yaklaşımda, ana mantık problemin sahip olduğu kısıtların matematiksel ve mantıksal ifadelerle oluşturulması ve uygun çözüm alanlarının tanımlanmasıdır. Tanımlanan çözüm alanlarından değişkenlerin alabileceği uygun değerler bulunur ve problemin çözümü elde edilir. Verilen bir problemi kısıt programlama ile çözmek için kısıt sağlama problemi olarak formülize edilmelidir[22].

### 2.1. Kısıt Sağlama Problemi (Constraint Satisfaction Problem)

Kısıt sağlama problemleri bir amaca bağlı olmadan problemin bütün uygun çözümlerinin bulunmasını sağlayan yöntemdir. Kısıt programlama ile optimum çözümün bulunması amaçlanan problemlerde verilen bir amaç fonksiyonuna göre her bir çözüm değerlendirilir ve bu yolla verilen amaç fonksiyonu tipine uygun şekilde maksimum veya minimum amaç fonksiyonu değerine sahip çözüm veya çözümler bulunur. Bu problem tarzı kısıt optimizasyonu problemi olarak tanımlanır. Kısıt optimizasyon problemleri, kısıt sağlama problemlerine amaç fonksiyonu eklenerek geliştirilmiş bir problem çözme yöntemidir[22].

### 2.2. Arama Stratejileri (Search Strategies)

Kısıt programlama metodunda, genellikle üç tip arama stratejisi kullanılmaktadır[22]. Birinci arama stratejisi olan öncelikle derinlik arama stratejisi ile dal-sınır benzeri bir arama yapılmaktadır. En alt seviyedeki düğümlerden başlanarak ilgili dal için bütün çözümler bulunduktan sonra sıra ile diğer dallar araştırılmakta ve arama tamamlanmaktadır. Bu arama stratejisi ile bütün olası çözüm kombinasyonları elde edilmektedir. Diğer bir arama stratejisi olan çok noktalı arama algoritması, çözüm uzayının farklı bölgelerindeki çözümleri bulmayı sağlamaktadır. Algoritmada başlangıç çözüm havuzunda bulunan çözümlerin bir kombinasyonu ile oluşturulan yeni havuzlarda daha iyi çözümlere ulaşmak amaçlanmaktadır. Algoritma optimalliği veya daha iyi çözümün bulunmasını garanti etmemektedir. Bir başka arama stratejisi ise yeniden başlamalı arama stratejisidir. Bu arama stratejisinde çözüm aramada belirlenen hata değerine ulaşıldığında optimum çözüme ulaşmak amacıyla yeniden başlatılır. Yeniden başlatma ile arama uzayının ulaşılmayan bölgelerini de aramaya dâhil etmek amaçlanmaktadır[23].

## 3. KUYRUK AĞLARI VE ORTALAMA DEĞER ANALİZİ (ODA) (QUEUE NETWORKS AND MEAN VALUE ANALYSIS(MVA))

Bu çalışmada, Kısıt Programlama ile oluşturulan görev atama kombinasyonlarının performansları Kuyruk Ağları yöntemi ile hesaplanmış ve karşılaştırmalar yapılmıştır. Kuyruk ağları, sistem performansını ölçmede ve sistem karakteristiklerini belirlemede kullanılan matematiksel bir analiz

aracıdır. Kuyruk ağları, birden fazla kuyruk sisteminin birbirini takip etmesinden oluşan sistemler olarak tanımlanmaktadır. Kuyruk ağları ile sistemin performans değerleri analitik olarak belirlenebilmekte, en iyi çalışma parametrelerinin bulunması mümkün olmaktadır.[24]

Kuyruk ağları genel yapıları itibari ile sistemin yapısı da göz önüne alınarak sistemdeki müşteri sayısının değişimine, sistemin topolojik yapısına, sistemde bulunan ürün sayısı çeşitliğine ve sistemin bloke durumuna göre sınıflara ayrılmıştır. Bu sınıflar içerisinde sistemdeki müşteri sayısının değişim durumuna göre kuyruk ağları iki gruba ayrılmaktadır:[24]

- Açık kuyruk ağları; müşterilerin/ürünlerin bir veya daha fazla yolla dışarıdan gelecek hizmet aldıkları ve sonrasında sistemi terk ettikleri modellerdir. Açık ağlarda sistemdeki müşteri sayısı random (rastgele) gelişlere ve terk edilere bağlı olarak zamanla değişir.
- Kapalı kuyruk ağları; sistemdeki müşteri sayısının sabit olduğu ve sürekli bir şekilde sistemde dolaştıkları modellerdir. Bu ağlarda sisteme dışarıdan girişler veya sistemden dışarı çıkışlar yoktur.

Montaj hatları ürünleri taşıyan paletler ve hatta bulunan palet sayılarının sabitliğinden dolayı kapalı kuyruk ağlarına benzemektedir. Bu nedenle, montaj hattının performans değerleri kapalı kuyruk ağlarının performans ölçümünde kullanılan Ortalama Değer Analizi (ODA) yaklaşımı ile bulunmuştur.

ODA, kapalı kuyruk ağları için çıktı hızını, istasyon doluluk oranını ve ortalama kuyruk uzunluğunu hesaplayan bir tekniktir. Ayrıca, yöntem çarpım formu olarak ifade edilebilen ağların analizinde kullanılan tam sonuç veren bir algoritmadır. İlk defa Reiser ve Lavenberg[25] tarafından ortaya konulmuştur. Çalışmamızda performans ölçümleri için tek sınıflı blokesiz sistemler için önerilen ODA algoritması kullanılmıştır. Algoritmada kullanılan notasyonlar Tablo 1'de verilmiştir. Algoritmanın adımları aşağıdaki gibidir:[24]

*Adım 1.* Başlangıç değerlerini ata.  $i=1, \dots, M$  için  $\pi_i(0|0)=1$

*Adım 2.* İterasyon:  $n=1, \dots, N$  tüm popülasyon kombinasyonlarını itere et

*Adım 2.1.* Her  $i=1, \dots, M$  için işlem oranlarını hesapla

$$\mu_i(n) = \begin{cases} n\mu_i, & \text{\&eğ}er n \leq c_i \\ c_i\mu_i, & \text{\&eğ}er n > c_i \end{cases}$$

*Adım 2.2.* Her  $i=1, \dots, M$  için istasyon  $i$ 'deki ortalama istasyon zamanlarını hesapla

$$W_i(n) = \sum_{j=1}^n \frac{j}{\mu_i(j)} \pi_i(j-1|n-1)$$

Adım 2.3. Sistem çıktı hızını hesapla

$$\lambda(n) = \frac{n}{\sum_{i=1}^M W_i(n) V_i}$$

Adım 2.4.  $i = 1, \dots, M$  için koşullu olasılıkları hesapla

$$\pi_i(j|n) = \frac{\lambda(n)}{\mu_i(j)} \pi_i(j-1|n-1) V_i, \quad \text{her } j=1, \dots, n \text{ için}$$

$$\pi_i(0|n) = 1 - \sum_{j=1}^n \pi_i(j|n) \quad j=0 \text{ için}$$

**Tablo 1** ODA notasyonları (MVA Notations)

$M$	İstasyon sayısı
$N$	Müşteri/Ürün sayısı
$n_i$	$i^{\text{inci}}$ istasyondaki müşteri sayısı ( $\sum_i^M = N$ )
$\mu_i$	$i^{\text{inci}}$ istasyonun işlem oranı ( $i = 1, \dots, M$ )
$1/\mu_i$	$i^{\text{inci}}$ istasyondaki işlerin ortalama süresi ( $i = 1, \dots, M$ )
$V_i$	$i^{\text{inci}}$ istasyona geliş oranı ( $i = 1, \dots, M$ )
$\pi_i(j n)$	Sistemde $n$ müşteri varken $i^{\text{inci}}$ istasyonda $j$ iş olma koşullu olasılığı ( $i = 1, \dots, M, j = 0, \dots, n, n = 0, \dots, N$ )
$c_i$	$i^{\text{inci}}$ istasyondaki paralel işlemcilerin sayısı ( $c_i \geq 1, i = 1, \dots, M$ )
$\lambda$	Sistemin çıktı hızı
$W$	Çevrim zamanı, müşterilerin sisteme giriş anından çıktıkları ana kadar geçen veya bir çevrimi tamamladıkları süre.
$W_i$	$i^{\text{inci}}$ istasyondaki ortalama süre ( $i = 1, \dots, M$ )

#### 4. PROBLEMİN TANIMI (DESCRIPTION OF THE PROBLEM)

Ele aldığımız problem Tip 2 stokastik görev zamanlı, tek modelli, düz MHDP'dir. Giriş kısmında bahsedildiği üzere, Tip 2 MHDP problemlerinde amaç istasyon sayısı kısıdı altında çevrim süresini minimize etmektir. Ele aldığımız problem için yapılan kabul ve varsayımlar şu şekilde sıralanabilir:

- Her görev mutlaka bir istasyona atanmalıdır.
- Düz montaj hattı olduğu varsayılmıştır.
- Görev süreleri stokastiktir ve üssel dağılıma uymaktadır.
- Görevler bölünemezler.
- İstasyon sayısının sabit olduğu ve bilindiği kabul edilmiştir.
- Montaj hattında tek tip ürün üretildiği kabul edilmiştir.

- Görevlerin öncelik kısıtları bellidir ve görev-istasyon atamalarında bozulmamalıdır.
- Görev zamanları görev sıralarından bağımsızdır.
- Her istasyonda mutlaka en az bir görev bulunmalıdır.

Deterministik Tip 2 MHDP problemi için oluşturulan matematiksel model şu şekildedir.[4]

$$\text{Min } z = ct \quad (1)$$

Şu kısıtlara göre:

$$\sum_{k=1}^{|W|} x_{ik} = 1 \quad \forall t_i \in T \quad (2)$$

$$\sum_{k=1}^{|W|} k * x_{ik} \leq \sum_{k=1}^{|W|} k * x_{jk} \quad \forall \{t_i \rightarrow t_j | t_i, t_j \in T\} \quad (3)$$

$$ct \geq \sum_{i=1}^{|T|} \tau_i * x_{ik} \quad \forall w_k \in W \quad (4)$$

$$x_{ik} \in \{0, 1\} \quad \forall (t_i, w_j) \in (T \times W) \quad (5)$$

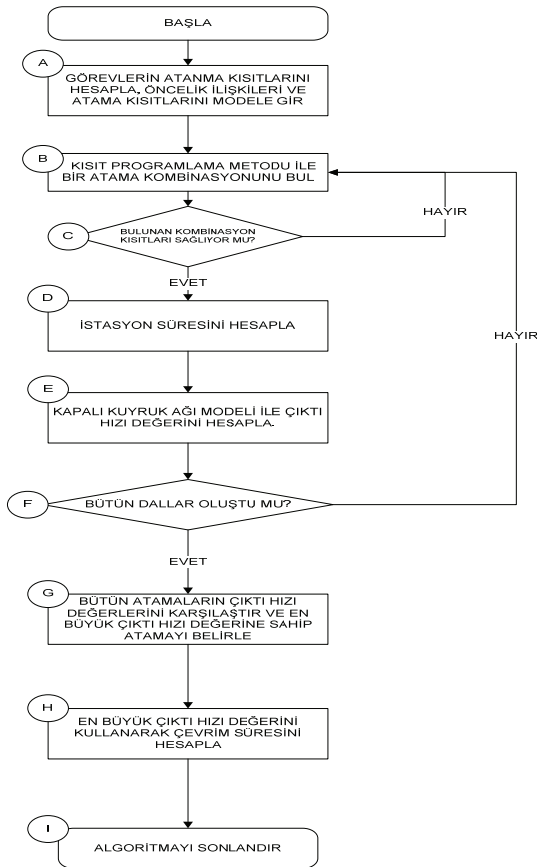
**Tablo 2.** Notasyonlar ve açıklamaları (Notations and descriptions)

$T$	Görev seti
$W$	İş istasyonları seti
$t_i$	T seti içindeki $i$ . Görev
$w_i$	W seti içindeki $i$ . İstasyon
$\tau_i$	$t_i$ 'nin proses zamanı
$\omega_i$	$W_j$ istasyonundaki iş yükü
$\bar{\omega}$	İstasyonlardaki iş yüklerinin ortalaması
$Ct$	Çevrim zamanı
$t_i \rightarrow t_j$	Öncelik kısıdı, $t_i$ görevi mutlaka $t_j$ görevin önce yapılmalıdır. =1 eğer $t_i$ görevi $w_k$ istasyonuna atanmış ise =0 diğer durumlarda
$x_{ik}$	

Burada, denklem 1, amacımız olan çevrim süresinin minimizasyonudur, Denklem 2, her görevin mutlaka bir istasyona atanmasını garanti etmektedir. 3 nolu denklem görevlerin istasyonlara atanmasında öncelik ilişkilerinin sağlanmasını garanti eder. Denklem 4 ise çevrim süresinin en büyük istasyon süresine eşit olmasını sağlamaktadır. Denklem 5 ise değişkenlerin ikili değerler almasını sağlamaktadır. Görev süreleri stokastik alındığında model stokastik optimizasyon problemine dönüşmektedir.

## 5. ÖNERİLEN ÇÖZÜM ALGORİTMASI (PROPOSED SOLUTION ALGORITHM)

Önerilen yöntemde görevlerin istasyonlara atanmasında kısıt programlama yöntemi kullanılarak bütün alternatif atamaların bulunabildiği yeni bir atama modeli oluşturulmuştur. Kısıt programlama yöntemi ile oluşturulan alternatif hat tasarımları kuyruk ağı modeli kullanılarak çıktı hızı değeri hesaplanmıştır. Bulunan çıktı hızı değerleri karşılaştırılarak en büyük çıktı hızına sahip atama kombinasyonu belirlenmiştir. Bu değer kullanılarak optimum çevrim süresi değeri elde edilmiştir. Önerilen yöntemin akış diyagramı şekil 2'de verilmiştir.



Şekil 2. Algoritma akış diyagramı (The algorithm flow chart)

Tip-2 Stokastik MHDH çözümü için önerilen kısıt programlama tabanlı algoritma şu şekildedir:

$ST_i$  değişkeni  $i$  görevinin atandığı  $j$  istasyon numarasını tutmaktadır.  $i$  görevinin atanabileceği ilk ve son istasyonları (algoritmamızda değişkenlerin alabileceği değerlere karşılık gelmektedir) bulmak için Klein ve Scholl[9]'un kullandıkları 6 ve 7 numaralı denklemlerde verilen formülasyon kullanılmıştır.  $E_i$  değeri görevin atanabileceği en düşük istasyon numarasını temsil etmektedir.  $L_i$  değeri ise görevin atanabileceği en yüksek istasyon numarasını temsil etmektedir. Burada şunu belirtmek

gerekir ki, bu denklemler sabit zamanlı görevlerin olduğu montaj hatları için geliştirilen alt ve üst sınır değerleridir. Bu çalışmada görev zamanları üssel dağılım olarak kabul edilmiştir. Üssel zamanlar toplandığı zaman Hipüssel dağılım veya bir diğer adıyla genelleştirilmiş Erlang dağılımı olmaktadır. Yalnız oluşan Hipüssel dağılımın ortalaması da bunları oluşturan üssel dağılımların toplamıdır. ( $\tau = \sum_{i=1}^k \tau_i$ ) Dolayısı ile denklemlerin geçerliliği devam etmektedir.  $P_i^*$ ,  $i$  görevinden önce gelen bütün görevlerin kümesidir.  $F_i^*$ ,  $i$  görevinden sonra gelen görevlerin kümesidir.

$$E_i(ct) \leq \left\lceil \frac{(\tau_i + \sum_{h \in P_j^*} \tau_h)}{ct} \right\rceil \quad (6)$$

$$L_i(ct) \leq m+1 - \left\lceil \frac{(\tau_i + \sum_{h \in F_j^*} \tau_h)}{ct} \right\rceil \quad (7)$$

Bu formülasyonda  $ct$  çevrim süresini belirtmektedir. Amacımız çevrim süresini minimize etmek olduğu için çevrim süresi olarak ulaşılmak istenen değer olan teorik çevrim süresi alınmış ve hesaplamalar buna göre yapılmıştır. Teorik çevrim süresinin hesaplanması denklem 8'de verilmiştir.

$$ct = \max \left( \max(\tau_i), \frac{\sum_{i=1}^{|T|} \tau_i}{|W|} \right) \quad (8)$$

Bu hesaplamalar sonucunda görevlerin atanabileceği alternatif istasyon numaraları bulunmuş ve modele girilmiştir.  $ST_i$  değişkeninin alabileceği değerler,  $E_i$  değerinden büyük veya eşit olmalı ve  $L_i$  değerinden küçük veya eşit olmalıdır ( $E_i \leq ST_i \leq L_i$ ). Öncelik ilişkilerinde ise  $i$  görevi  $k$  görevinden hemen önce gelmek üzere Denklem 9'daki gibi kısıtlar bütün önceliği olan görevler için girilmiştir.(A)

$$ST_i \leq ST_k \quad (9)$$

Kısıt programlama metodu ile bir atama kombinasyonu bulunur(B). Bu atama kombinasyonunun kısıtları sağlayıp sağlamadığı belirlenir. Eğer tüm kısıtları sağlıyor ise diğer adıma geçilir, uygun değilse yeni bir atama kombinasyonu oluşturulur(C). Tüm kısıtları sağlayan atama kombinasyonu için istasyon süreleri hesaplanır(D). Bulunan istasyon süreleri kullanılarak kapalı kuyruk ağı modeli ile çıktı hızı değerleri hesaplanır(E). İlgili problem için bütün dallara ulaşıldı ise algoritmada bir sonraki adım olan en büyük çıktı hızının bulunmasına geçilir, ulaşılmadı ise yeni atama kombinasyonu

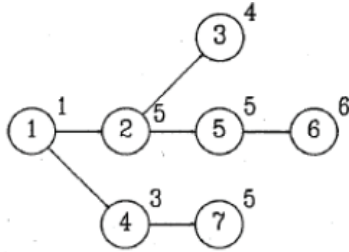
oluşturma adımına geçilir(F). Bütün dallara ulaşıldı ise, bulunan bütün atama kombinasyonlarının çıktı hızı değerleri karşılaştırılır ve en büyük çıktı hızına sahip atama optimum atama kombinasyonu olarak belirlenir.(G). En büyük çıktı hızı değeri kullanılarak çevrim süresi hesaplanır(H) ve algoritma sonlandırılır(I).

### 5.1. Açıklayıcı Örnek (Illustrative Example)

Bu bölümde, önerilen yöntem literatürde yer alan 7 görevli Merten montaj hattı deney seti üzerinden açıklanmıştır. Merten montaj hattı için 3 istasyon kurulmak istenmektedir. Öncelikle denklem 8 ile hesaplanan teorik çevrim süresi (ct) değerine göre görevlerin atanabileceği istasyonlar bulunmuştur. Denklem 6 ile  $E_j$  ve denklem 7 ile  $L_j$  değerleri her bir görev için hesaplanmış ve Tablo 3’de verilmiştir. Şekil 3’te görevler arasında öncelik ilişkileri verilmiştir. Her bir düğüm görevleri temsil etmektedir ve görev zamanları sağ üst köşede verilmiştir.

**Tablo 3.** Görevlerin atanabileceği en düşük ve en yüksek istasyon numaraları (The lowest and the highest station numbers can be assigned to the tasks)

Görevler	1	2	3	4	5	6	7
$E_j$	1	1	2	1	2	1	1
$L_j$	1	1	3	3	2	3	3



**Şekil 3.** Merten montaj hattı görevleri arasındaki öncelik ilişkileri (Precedence relationships between tasks in Merten assembly line)

Bu değerler ILOG CP Optimizier 6.3 ile kısıt programlama modeline girilmiştir. Değişken olarak

tanımlanan  $ST(i)$  değişkeni görevlerin atandığı istasyonları ifade etmektedir ve atanmış istasyonun değerini almaktadır. Örneğin, 3 nolu görev 2 nolu istasyona atanmış ise  $ST(3)=2$  değerini almaktadır.

Görevler arasındaki öncelik ilişkileri her görevin  $ST(i)$  değişkeninin aldığı değerin kendinden hemen sonra gelen görevin  $ST(i)$  değişkeninin aldığı değerden küçük veya eşit olması kısıdı eklenerek sağlanmıştır. Örneğin,  $ST(2) \leq ST(3)$  ifadesi ile 2 nolu görevin atandığı istasyon numarasının 3 nolu görevin atandığı istasyon numarasından küçük veya eşit olması sağlanmıştır.

Görevlerin atanabileceği istasyon numaraları ise  $ST(i)$  değişkeninin alacağı değerin ilgili görev için  $E_j$  değerinden büyük veya eşit ve  $L_j$  değerinden küçük veya eşit değerler olması ile modele dâhil edilmiştir. Örneğin, 4 nolu iş için  $ST(4) \geq 1$  ve  $ST(4) \leq 3$ . Bu şekildeki kısıtlar her bir görev için modele eklenmiştir.

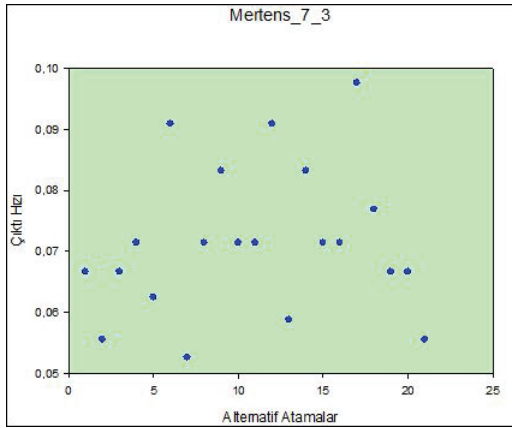
Bu kısıtlar ile modellenen Merten montaj hattının bütün uygun atama kombinasyonları ve bu atamaların ortalama değer analizi ile hesaplanan çıktı hızı, istasyon doluluğu gibi performans değerleri Şekil 2’de akış şeması verilen algoritma ile elde edilmiştir. Bulunan değerlerden çıktı değerleri karşılaştırılarak en büyük çıktı hızına sahip olan atama optimum çözüm olarak bulunmuştur. Bulunan bütün atamaların çıktı hızı ve çevrim süresi değerleri Tablo 4’te verilmiştir. Çıktı hızı değerlerine ilişkin grafik Şekil 4’te verilmiştir.

Tablo 4’te deney sonuçlarında görüldüğü üzere, en yüksek çıktı hızına sahip atama 0,09763 olarak bulunmuştur. Yapılan deneyde montaj hattının bütün alternatif atamaları elde edilmiş ve değerlendirilmiş olduğundan bulunan bu atama optimum atamadır.

Ayrıca, algoritma ile en yüksek çıktı hızı değerine sahip alternatif atama kombinasyonları da bulunabilmektedir. Bu sayede planlamacılara farklı hat tasarımlarını seçebilmeleri imkân sağlamaktadır.

**Tablo 4.** Merten montaj hattı çözüm sonuçları (The results of Merten assembly line solution)

Atama No	ST(1)	ST(2)	ST(3)	ST(4)	ST(5)	ST(6)	ST(7)	Çıktı Hızı	Çevrim süresi
1	1	1	2	1	2	2	3	0,06667	15,0000
2	1	1	2	2	2	2	3	0,05556	18,0000
3	1	1	2	3	2	2	3	0,06667	15,0000
4	1	1	3	1	2	2	1	0,07143	14,0000
5	1	1	3	1	2	2	2	0,06250	16,0000
6	1	1	3	1	2	2	3	0,09090	11,0008
7	1	1	3	2	2	2	2	0,05263	19,0000
8	1	1	3	2	2	2	3	0,07143	14,0000
9	1	1	3	3	2	2	3	0,08323	12,0144
10	1	1	2	1	2	3	1	0,07143	14,0000
11	1	1	2	1	2	3	2	0,07143	14,0000
12	1	1	2	1	2	3	3	0,09090	11,0008
13	1	1	2	2	2	3	2	0,05882	17,0000
14	1	1	2	2	2	3	3	0,08323	12,0144
15	1	1	2	3	2	3	3	0,07143	14,0000
16	1	1	3	1	2	3	1	0,07143	14,0000
<b>17</b>	<b>1</b>	<b>1</b>	<b>3</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>2</b>	<b>0,09763</b>	<b>10,2425</b>
18	1	1	3	2	2	3	2	0,07692	13,0000
19	1	1	3	1	2	3	3	0,06667	15,0000
20	1	1	3	2	2	3	3	0,06667	15,0000
21	1	1	3	3	2	3	3	0,05556	18,0000

**Şekil 4.** Merten 7 iş 3 istasyon montaj hattı çözüm sonuçları grafiği (The results of the Merten 7 work 3 station assembly line solution graph)

## 5.2. Deneysel Çalışmalar (Experimental Studies)

Önerilen yöntemi test etmek amacı ile deney setleri oluşturulmuştur. Literatürde stokastik MHDP için deney seti bulunmadığından Uğurdağ vd.[8] çalışmasında yer alan deterministik deney setlerinden 17 adet deney seti seçilmiştir. Süreler stokastik olarak alınıp deneyler yapılmıştır.

Deney setleri ILOG CP 6.3 ile modellenmiştir. Öncelikle derinlik arama ve çok noktalı arama stratejileri kullanılarak problemler çözülmüştür. Küçük boyutlu problemler için öncelikle derinlik arama stratejisi kullanılmıştır. Bu problemler için bulunan sonuçlar Tablo 6'da verilmiştir. Bu sonuçlarda, deney setleri için bütün kombinasyonlar değerlendirilmiş olduğundan bulunan çıktı hızı ve çevrim süresi değerleri optimumdur.

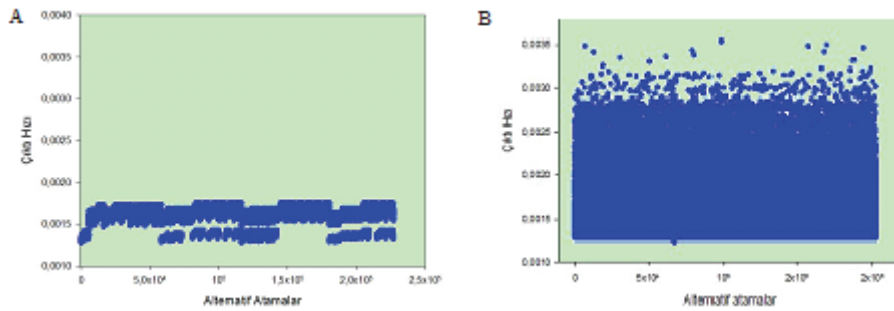
Görev sayısı ve istasyon sayısına bağlı olarak alternatif sayısı çoğalan büyük boyutlu problemlerde bütün atama kombinasyonlarının değerlendirilmesi mümkün olmamaktadır. Bu problemler, bilgisayar hafıza yetersizliklerinden dolayı öncelikle derinlik arama stratejisi ile çözülemediğinden, çok noktalı arama stratejisi kullanılarak çözülmüştür. Her bir problem 24 saat çalıştırılmış ve elde edilen çıktı hızı değerleri tablo 7'de verilmiştir. Problemlerde bütün atama kombinasyonları değerlendirilemediğinden, bulunan çıktı hızı değerleri optimuma yakın (suboptimal) değerlerdir. Bu durum, hesaplanan teorik çevrim süreleri ile çıktı hızı değeri kullanılarak elde edilen çevrim süreleri karşılaştırıldığında görülmüştür.

**Tablo 6.** Öncelikle derinlik arama stratejisi için sistem çıktı hızı değerleri (Values of the system output rate for depth first search strategy)

Deney Seti	İstasyon Sayısı	En büyük çıktı hızı değeri	Çevrim süresi	Alternatif Atama Kombinasyonu Sayısı
Merten's 7 İş Problemi	2	0,0700	14,280	28
	3	0,0976	10,243	21
	5	0,1427	7,006	304
Jaeschke's 9 İş Problemi	3	0,0714	14,002	4
	4	0,1085	9,220	4
	7	0,1428	7,005	60
Jackson's 11 İş Problemi	3	0,0619	16,151	250
	4	0,0764	13,093	656
	5	0,0968	10,105	3810
Mitchell's 21 İş Problemi	3	0,0275	36,400	960
	5	0,0441	22,680	16578
	8	0,0625	16,007	221000

**Tablo 7.** Çok noktalı arama stratejisi için sistem çıktı hızı değerleri (Values of the system output rate for multi point search strategy)

Deney Seti	İstasyon Sayısı	En büyük çıktı hızı değeri	Çevrim süresi	Alternatif Atama Kombinasyonu Sayısı
Heskiaoff's 28 İş Problemi	4	0,0035	285,714	250000
	5	0,0042	238,095	250000
Sawyer's 30 İş Problemi	5	0,0133	75,188	206000
	8	0,0182	54,945	206000
	13	0,026	38,462	206000

**Şekil 6.** Heskiaoff 28 iş 4 istasyonlu montaj hattı için öncelikle derinlik (A) ve çok noktalı (B) arama stratejileri çözüm sonuçlarının (The solution results of depth-first and multi-point search strategies for 28 task 4 station Heskiaoff experiment)



Yapılan deneysel çalışmalarda iki tür arama stratejisi kullanılmıştır. İlk olarak optimum sonucu bulmak amaçlandığından bütün atama kombinasyonlarının bulunabildiği öncelikle derinlik arama stratejisi kullanılmıştır. Ancak, Heskiaoff montaj hattı deney seti için yapılan çözümde görülmüştür ki, problem boyutu istasyon sayısı ve görev sayısına bağlı olarak büyüğünde optimum çözüme ulaşmak mümkün olmamaktadır. Bu sebeple, uygun çözüm uzayının değişik noktalarında arama yapabilen çok noktali arama stratejisi kullanılarak büyük boyutlu deney setleri yeniden çözülmüştür. Elde edilen çözümlere ilişkin grafikler şekil 6'da verilmiştir.

Şekil 6'daki grafiklerde görüldüğü üzere öncelikle derinlik arama stratejisinde arama bir bölgede sıkışmaktadır ve iyi çözümlere ulaşamamaktadır. Ancak, çok noktali arama stratejisinin değişik bölgelerde arama yapabilmesi optimum olmamasına karşın çok daha iyi çözümlerin elde edilmesini sağlamıştır.

## 6. SONUÇLAR VE TARTIŞMA (CONCLUSION AND DISCUSSION)

Bu çalışmada tek modellenli düz stokastik montaj hatları ele alınmıştır. Tip-2 olarak tarif edilen sabit istasyon sayısı kısıdı altında çevrim süresinin minimizasyonuna çalışılmıştır. Ele alınan MHDP için yeni bir metodoloji geliştirilmiştir. Bu metodolojide, montaj hattında istasyonlara görev atanmasında, kısıt programlama modeli geliştirilmiştir. Kısıt programlama ile belirlenen atama kombinasyonlarının performansı, stokastik montaj hattı dengeleme problemi olduğu için, kapalı kuyruk ağı modeli ile ölçülmüş ve en iyi çevrim süresi değerine sahip olan atama belirlenmiştir.

Geliştirilen yöntemi test etmek amacıyla literatürde bulunan örnek montaj hattı modelleri kullanılmıştır. Bu modellerden 12 deney seti öncelikle derinlik arama stratejisi ile çözülmüş ve optimum sonuçlar elde edilmiştir. 5 adet deney seti çoklu noktali arama stratejisi kullanılarak çözülmüş ve optimuma yakın suboptimal değerler elde edilmiştir.

Tip-2 MHDP'yi çalışan araştırmacılar, genel olarak tip-1 MHDP problemlerini çözen algoritmaları kullanılarak adım adım çözüme ulaşılmışlardır. Bu çalışmada ise, direkt Tip-2 MHDP'yi çözen yeni bir yöntem önerilmiştir. Ayrıca ilk defa olarak, görevlerin istasyonlara atanmasında, kısıt programlama modeli kullanılmıştır. Ek olarak, stokastik hatların dengelenmesi esnasında atamaların performansının değerlendirilmesinde kuyruk ağlarının kullanımı ilk defa uygulanmıştır.

Bu çalışmanın devamı olarak; MHDP için açık kuyruk ağı modelleri kullanılabilir, büyük boyutlu problemlerin çözümü için arama uzayını daraltacak

yeni kısıtlar eklenebilir veya sezgisel bir yöntem ile desteklenerek daha iyi bir çözümün elde edilebileceği araştırmalar yapılabilir.

## KAYNAKLAR (REFERENCES)

1. Çerçioğlu, H., **Stokastik Paralel Montaj Hattı Dengeleme Problemi için Yeni Modeller**, Doktora Tezi, Gazi Üniversitesi F.B.E., Ankara, (2009).
2. Gökçen, H., **Karışık Modelli Deterministik Montaj Hattı Dengeleme Problemleri İçin Yeni Modeller**, Doktora Tezi, Gazi Üniv., F.B.E., Ankara, (1994).
3. Scholl, A. ve Becker, C., State-of-the-art exact and heuristic solution procedures for simple assembly line balancing, **European J. of Oper. Research**, Cilt:168, 666-693, 2006.
4. Scholl, A., **Balancing and Sequencing of Assembly Lines**, second ed. Physica-Verlag Press, Heidelberg, Germany, 1999.
5. Bryton, B., **Balancing of a Continuous Production Line**, Yüksek Lisans tezi. Northwestern University, Northwestern, (1954).
6. Salvesson, M. E. The Assembly Line Balancing Problem, **J. of Industrial Engineering**, Cilt:6, No:3, 18-25, 1955.
7. Helgeson, W., & Birnie, D. Assembly Line Balancing Using The Ranked Positional Weight Technique, **J. of Ind. Eng.**, Cilt:12 No:6, 384-389, 1961.
8. Ugurdag, H., Rachamadugu, R., Papachristou, C. A., Designing paced assembly lines with fixed number of stations, **European J. of Operational Research**, Cilt:102, 488-501, 1997.
9. Carnahan, B. J., Norman, B. A., Redfern, M. S., Incorporating Physical Demand Criteria into Assembly Line Balancing. **IIE Transactions**, Cilt:33, No:10, 875-887, 2001.
10. Liu, S., Ong, H., Huang, H. A, Bi-directional Heuristic for Stochastic Assembly Line Balancing Problems. **Int. J. Adv. Manuf. Technol.**, Cilt:25: 71-77, 2005.
11. Pastor, R., & Ferrer, L., An improved mathematical program to solve the simple assembly line balancing problem, **Int. J. of Production Research**, Cilt:47 No:11: 2943-2959, 2009.
12. Wei, N., & Chao, M., A solution procedure for type E simple assembly line balancing problem, **Computers & Ind. Eng.**, Cilt:61, No:3, 824-830, 2011.
13. Nourmohammadia, A., Zandieh, M., Assembly line balancing by a new multi-objective differential evolution algorithm based on TOPSIS. **Int. J. of Production Research**, Cilt:49, No:10, 2833-2855, 2011.
14. Modie, C. L., Young, H. H., A heuristic method of assembly line balancing for assumptions of

- constant or variable work element times. **Journal of Industrial Engineering**, Cilt:16, 23-29, 1965.
15. Gökçen, H., ve Baykoç, Ö., A new line remedial policy for the paced lines with stochastic task times, **Int. J. of Prod. Economics**, Cilt:58, No:2, 191-197, 1999.
  16. Ağpak, K., ve Gökçen, H. A chance-constrained approach to stochastic line balancing. **European J. Of Oper. Res.**, Cilt:180, No:3, 1098-1115, 2007.
  17. Ayazi, S., Hajizadehb, A. E., Nooshabadic, M. R., Jalaiea, H., ve Moradi, Y., Multi-objective assembly line balancing using genetic algorithm, **Int. J. of Industrial Engineering Computations**, Cilt:2: No:4, 863-872, 2011
  18. Becker, C., ve Scholl, A., A survey on problems and methods in generalized assembly line balancing, **European J. of Oper. Res.**, Cilt:168, 694-715, 2006.
  19. Boysen, N., Fliedner, M., ve Scholl, A., A classification of assembly line balancing problems. **European J. of Oper. Res.**, Cilt:183, 674-693, 2007.
  20. Boysen, N., Fliedner, M. ve Scholl, A., Assembly line balancing: Which model to use when?, **International J. of Prod. Economics**, Cilt:111, 509-528, 2008.
  21. Topaloglu, S., Salum, L. ve Supciller A.A., Rule-based modeling and constraint programming based solution of the assembly line balancing problem, **Expert Systems with Applications**, Cilt:39, No:3, 3484-3493, (2012)
  22. Apt, K. R., **Principles of Constraint Programming**, Cambridge University Press, Amsterdam, The Netherlands, 2003.
  23. IBM. ILOG CP Optimizer User's Manual.,2012
  24. Bolch G., Greiner S., Meer H, Trivedi K. S., **Queuing Networks and Markov Chains Modeling and Performance Evaluation with Computer Science Applications**, John Wiley & Sons, New Jersey, 2006.
  25. Reiser, M., & Lavenberg, S., Mean Value Analysis For Closed Multi-Chain Queueing Networks, **Journal of A.C.M.**, Cilt:27, No:2, 313-322, 1980.