

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

**KARIŞIK MODELLİ MONTAJ HATTI Dengeleme Problemleri ve  
Genetik Algorİtmalar İle Bir Uygulama**

**YÜKSEK LİSANS TEZİ**

**İBRAHİM KÜÇÜKKOÇ**

**Balıkesir, Temmuz-2011**

T.C.  
BALIKESİR ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

KARIŞIK MODELLİ MONTAJ HATTI Dengeleme Problemleri ve  
Genetik Algorİtmalar İle Bir Uygulama

YÜKSEK LİSANS TEZİ

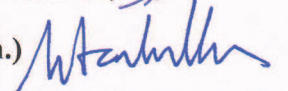
İBRAHİM KÜÇÜKKOÇ

Tez Danışmanı: Prof. Dr. Ramazan YAMAN

Sınav Tarihi: 12.07.2011

Jüri Üyeleri: Prof. Dr. Ramazan YAMAN (BAÜ-MMF-Endüstri Müh.) 

Yrd. Doç. Dr. A. Deniz KARAOĞLAN (BAÜ-MMF-Endüstri Müh.) 

Yrd. Doç. Dr. Ayhan İSTANBULLU (BAÜ-Bilgisayar Müh.) 

Enstitü Yönetim Kurulunun ..... tarih ..... sayılı oturumunun .....  
nolu kararı ile ..... Mezun olmuştur.

Balıkesir, Temmuz-2011

**Bu tez çalışması BALIKESİR ÜNİVERSİTESİ, BİLİMSEL ARAŞTIRMA PROJELERİ BİRİMİ tarafından 2011-40 nolu proje ile desteklenmiştir.**

## ÖZET

### KARIŞIK MODELLİ MONTAJ HATTI Dengeleme Problemleri ve Genetik Algorİtmalar İle BİR UYGULAMA

**İbrahim KÜÇÜKKOÇ**  
Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü,  
Endüstri Mühendisliği Anabilim Dalı

(Yüksek Lisans Tezi / Tez Danışmanı: Prof. Dr. Ramazan YAMAN)

Balıkesir, 2011

Sanayi devrimi ile birlikte Henry Ford öncülüğünde uygulamaya koyulan yürüyen bant sistemi, montaj hatlarına ilham kaynağı olmuştur. Geçmişten günümüze üretim planlama sürecinde anahtar rol oynayan montaj hatları ise yapısal açıdan gelişimini sürdürmekle birlikte, halen çeşitli çözüm algorİtmalarına da uygulama alanı oluşturmaktadır.

Bu çalışmada karışık modelli düz montaj hattı dengeleme problemleri ele alınmıştır. Gerçek hayat koşullarını daha iyi yansıtmayı amacıyla model kurgulanırken paralel iş istasyonu oluşumu durumu ve pozitif-negatif bölgeleme kısıtları göz önüne alınmıştır. Öncelikle konu ile ilgili yapılan çalışmalar incelenmiş ve literatür çalışmaları ile bazı sezgisel yöntemler özet halinde sunulmuştur. Daha sonra COMSOAL yöntemi ve genetik algoritmanın birleştirilmesiyle geliştirilen hibrid genetik algoritma ile çözüm yolları aranmıştır. Geliştirilen algoritma modellenerek Matlab programında kodlanmıştır. Algoritmanın işleyişi bir örnek üzerinde anlatılarak, geçerliliğini göstermek ve performansını ölçmek amacıyla 12 test problemi üzerinde denemeler yapılarak elde edilen sonuçlar yalnız genetik algoritma sonuçlarıyla kıyaslanmıştır. Sonuçta, geliştirilen hibrid genetik algoritmanın, 12 test problemi üzerinde yalnız genetik algorİtmaya göre daha iyi sonuçlar verdiği görülmüştür.

**ANAHTAR KELİMELER:** Karışık Modelli Montaj Hattı Dengeleme, Genetik Algorİtmalar, Üretim Planlama, Sezgisel ve Meta-Sezgiseller, Matlab

## **ABSTRACT**

### **MIXED MODEL ASSEMBLY LINE BALANCING AND AN APPLICATION WITH GENETIC ALGORITHMS**

**İbrahim KÜÇÜKKOÇ**  
**Balikesir University, Institute of Science,**  
**Department of Industrial Engineering**

**(M.Sc. Thesis / Supervisor : Prof. Dr. Ramazan YAMAN)**

**Balikesir-Turkey, 2011**

With the industrial revolution, conveyor belt system had been put into practice led by Henry Ford, and it was the source of inspiration to assembly lines. From past to present, assembly lines which have a key role in the production planning process, develop themselves structurally, and still continue to generate various solution algorithms.

In this study mixed-model straight assembly line balancing problems are addressed. While establishing the model, state of the formation of a parallel workstation and the positive-negative zoning constraints are taken into account in order to reflect real-life conditions better. First of all, relevant studies and literature were examined and some heuristic methods are presented in a summary. Then, solutions have been sought with the hybrid genetic algorithm that is obtained from the combination of COMSOAL method and the genetic algorithms. The developed algorithm is modeled and then coded in Matlab program. Explained the functioning of the algorithm with an example, trials were made on 12 test problems in order to show the validity and measure the performance, and the results were compared with the results of a simple genetic algorithm. Ultimately, it has been observed that better results were obtained from 12 test problems with the developed hybrid genetic algorithm than the simple genetic algorithm.

**KEYWORDS:** Mixed Model Assembly Line Balancing, Genetic Algorithms, Production Planning, Heuristics and Meta-Heuristics, Matlab

## İÇİNDEKİLER

	<u>Sayfa</u>
<b>ÖZET</b> .....	<b>ii</b>
<b>ABSTRACT</b> .....	<b>iii</b>
<b>İÇİNDEKİLER</b> .....	<b>iv</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>vi</b>
<b>TABLO LİSTESİ</b> .....	<b>vii</b>
<b>ÖNSÖZ</b> .....	<b>viii</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. MONTAJ HATTI Dengeleme Problemlerine Genel Bakış</b> .....	<b>3</b>
2.1 Montaj Hattı Dengeleme Problemlerinin Temel Özellikleri .....	3
2.2 Montaj Hattı Dengeleme Problemlerinin Sınıflandırılması .....	5
2.3 Basit Montaj Hattı Dengeleme (bMHD) Problemlerinin Genel Formülasyonu .....	7
2.4 Karışık Modelli Montaj Hattı Dengeleme Problemleri .....	10
2.5 Karışık Modelli Montaj Hattı Dengeleme (kMHD) Problemlerinin Genel Formülasyonu .....	14
<b>3. MONTAJ HATTI DEGELEME PROBLEMLERİ İÇİN ÇÖZÜM YAKLAŞIMLARI VE GENETİK ALGORİTMALAR</b> .....	<b>17</b>
3.1 Çözüm Yaklaşımlarının Sınıflandırılması .....	17
3.2 En Büyük Aday Kuralı (Largest Candidate Rule - LCR) .....	19
3.3 Sıralı Pozisyonel Ağırlık Yöntemi (Ranked Positional Weighted Method - RPW) .....	23
3.4 COMSOAL (Computer Method for Sequencing Operations for Assembly Lines) .....	24
3.5 Kilbridge ve Wester Yöntemi .....	31
3.6 Genetik Algoritmalar .....	36
3.6.1 Kodlama .....	41
3.6.2 GA Operatörleri .....	42
3.6.2.1 Çaprazlama .....	42
3.6.2.2 Mutasyon .....	42
3.6.3 Seçim Mekanizmaları .....	45
3.6.4 Elitizm .....	47
3.6.5 Yeni Neslin Oluşturulması .....	47

<b>4. KARIŞIK MODELLİ DÜZ HATLARIN DENGELENMESİ.....</b>	<b>49</b>
4.1 Problemin Varsayımları ve Kısıtları .....	49
4.2 Amaç Fonksiyonu .....	53
4.3 Matematiksel Model.....	54
<b>5. UYGULAMA.....</b>	<b>59</b>
5.1 Önerilen Hibrid Genetik Algoritma .....	59
5.2 Örnek Uygulama .....	65
5.3 Kabul Edilen Test Problemleri ve Sonuçlarının Kıyaslanması .....	70
<b>6. SONUÇ VE TARTIŞMA.....</b>	<b>72</b>
<b>7. KAYNAKÇA .....</b>	<b>74</b>
<b>8. EKLER.....</b>	<b>77</b>
<b>EK-A. Karışık Modelli Montaj Hatları İçin Test Problemleri ve Süreleri.....</b>	<b>77</b>

## ŞEKİL LİSTESİ

<u>Şekil Numarası</u>	<u>Şekil Adı</u>	<u>Sayfa Numarası</u>
Şekil 2. 1	İki farklı ürüne ait öncelik ilişkileri diyagramının birleştirilmesi.....	4
Şekil 2. 2	Montaj hattı dengeleme problemlerinin sınıflandırılması .....	5
Şekil 2. 3	Model tiplerine göre montaj hatları sınıflandırması .....	6
Şekil 2. 4	Yerleşime göre montaj hattı türleri, düz hat, U-tipi hat ve C-tipi hat.....	7
Şekil 3. 1	Montaj hattı dengelemede kullanılan çözüm yaklaşımlarının sınıflandırılması ...	18
Şekil 3. 2	Çözüm yaklaşımlarının ortak aşamaları .....	18
Şekil 3. 3	Örneğe ait öncelik ilişkileri diyagramı ve görev zamanları.....	21
Şekil 3. 4	Görevlerin öncelik ilişkileri ve süreleri .....	25
Şekil 3. 5	Öncelik ilişkileri diyagramı .....	31
Şekil 3. 6	İstasyonlara atanmış görevler .....	36
Şekil 3. 7	Geleneksel ve genetik yaklaşımların karşılaştırılması.....	38
Şekil 3. 8	Genetik algoritma akış diyagramı.....	39
Şekil 3. 9	Çözüm uzayı ve arama uzayı.....	39
Şekil 3. 10	Gen, kromozom ve popülasyon yapısı.....	40
Şekil 3. 11	GA'da kullanılan çeşitli kodlama biçimleri.....	42
Şekil 3. 12	Tek noktalı çaprazlama örneği .....	43
Şekil 3. 13	Çok noktalı çaprazlama örneği .....	43
Şekil 3. 14	Pozisyona dayalı çaprazlama örneği.....	44
Şekil 3. 15	Sıraya dayalı çaprazlama örneği.....	44
Şekil 3. 16	Komşu iki genin, rastgele iki genin ve rastgele üç genin değiştirilmesi.....	45
Şekil 3. 17	Rulet Çemberi.....	46
Şekil 5. 1	Geliştirilen algoritmaya ait akış diyagramı.....	62
Şekil 5. 2	Geliştirilen algoritmanın simülasyonu.....	64
Şekil 5. 3	Kodlamada kullanılan kromozom yapısı .....	65
Şekil 5. 4	Uygulama örneğine ait öncelik ilişkileri diyagramı (Gunther).....	68



## TABLO LİSTESİ

<b><u>Tablo Numarası</u></b>	<b><u>Tablo Adı</u></b>	<b><u>Sayfa Numarası</u></b>
<b>Tablo 2. 1</b>	<b>kMHD-I literatürüne genel bakış .....</b>	<b>13</b>
<b>Tablo 3. 1</b>	<b>Örneğe ait görev zamanları ve öncelik ilişkileri .....</b>	<b>21</b>
<b>Tablo 3. 2</b>	<b>LCR kuralına göre istasyonlara yapılan atama .....</b>	<b>22</b>
<b>Tablo 3. 3</b>	<b>RPW kuralına göre görevlerin pozisyonel ağırlıkları .....</b>	<b>23</b>
<b>Tablo 3. 4</b>	<b>RPW kuralına göre istasyonlara yapılan atama .....</b>	<b>24</b>
<b>Tablo 3. 5</b>	<b>Liste A ve Liste B .....</b>	<b>25</b>
<b>Tablo 3. 6</b>	<b>COMSOAL ile yapılan çözüm sonucu gerçekleşen atamalar .....</b>	<b>30</b>
<b>Tablo 3. 7</b>	<b>Çözüme yönelik bilgilerin çizelge şeklinde ifade edilmesi.....</b>	<b>32</b>
<b>Tablo 5. 1</b>	<b>Gerçekleştirilmesi gereken görevlerin modellere göre süreleri .....</b>	<b>66</b>
<b>Tablo 5. 2</b>	<b>Elde edilen en iyi çözüm .....</b>	<b>69</b>
<b>Tablo 5. 3</b>	<b>Test problemlerinin sonuçlarının kıyaslanması .....</b>	<b>71</b>
<b>Tablo A. 1</b>	<b>Vilarinho&amp;Simaria (2002) ve Heskiaoff test problemlerine ait süreler .....</b>	<b>77</b>
<b>Tablo A. 2</b>	<b>Sawyer ve Lutz1 test problemlerine ait süreler .....</b>	<b>78</b>
<b>Tablo A. 3</b>	<b>Kilbridge&amp;Wester ve Tonge test problemlerine ait süreler .....</b>	<b>79</b>
<b>Tablo A. 4</b>	<b>Kilbridge&amp;Wester ve Tonge test problemlerine ait süreler (devam) .....</b>	<b>80</b>

## ÖNSÖZ

Bilimsel anlamda, akademik kariyere attığım adımın ilk meyvelerinden birisi olmaya aday olarak nitelendirilebilecek ve öyle olmasını temenni ettiğim bu çalışmada, öncelikle beni her zaman büyük bir sabır ve gayretle dinleyen, yönlendiren, motive eden, ufkumu genişleten, problem üretmek yerine problem çözme felsefesini ilke edinmiş çok değerli hocam Prof. Dr. Ramazan YAMAN'a sonsuz teşekkürlerimi sunarım.

Başta Yrd. Doç. Dr. Aslan Deniz KARAOĞLAN olmak üzere bölümdeki tüm hocalarıma verdikleri destek ve çeşitli konularda gösterdikleri anlayış için minnettarım.

Bu zorlu ve bir o kadar da sabır gerektiren süreçte, her zaman her türlü konuda desteğiyle yanımda olduğunu hissettiğim çok değerli eşime; gerek fiziksel gerekse ruhsal anlamda beni büyüten, yetiştiren ve her zaman desteklerini hissettiğim anneme, babama ve kardeşlerime ne kadar teşekkür etsem azdır.

Yüksek lisans eğitimim boyunca burs vererek beni ve çalışmalarımı bir ileri boyuta taşıyan TÜBİTAK'a, Bilimsel Araştırma Projeleri kapsamında sağladığı altyapı ve araştırma olanaklarıyla çalışmalarımı kolaylaştıran ve kalitesini artıran Balıkesir Üniversitesi ve BAÜ Bilimsel Araştırma Projeleri Birimine teşekkürlerimi sunarım.

İyi ki varsınız...

Balıkesir, Temmuz 2011

İbrahim KÜÇÜKKOÇ

## 1. GİRİŞ

Geçmişten günümüze süregelen montaj işlemi, pek çok gelişim ve değişime uğramıştır. En önemli kilometre taşlarından birisi ise 1913'te Henry Ford'un yürüyen bant sistemini ilk defa otomobil fabrikasında uygulamasıdır [1]. İşçilerin yarı mamul şeklindeki otomobilin yanına gitmesi yerine, otomobilin yürüyen bir bant sistemi ile işçilerin (iş istasyonlarının) önünden belirli bir hızda geçmesi planlanmıştır. Henry Ford tarafından yapılan ve otomobil üretiminde bir devrim niteliğindeki bu yenilik, işgücü israfını azaltarak birim zamanda daha fazla sayıda otomobil üretimine olanak sağlamıştır.

Sanayi devrimiyle birlikte kitle üretim montaj sistemleri sayesinde montaj hatlarında yüksek verimlilik sağlanmıştır. Fakat günümüzde ürün yaşam döngülerinin kısa olması ve müşteri isteklerine göre özelleştirilmiş ürünlerin daha çok talep görmesi nedeniyle, üretimin küçük partiler halinde ve mümkün olan en kısa sürede yapılması gerekliliği ortaya çıkmıştır [2].

Montaj hattı, belirli bir ürünün üretimine yönelik gerçekleştirilmesi gereken işlerin gerçekleştiği iş istasyonlarından oluşan bir üretim sürecidir. Montaj hatları, yürüyen bir bant sistemi veya çeşitli malzeme taşıma sistemleri ile birbirlerine bağlanırlar [3]. Her iş istasyonunda bir grup iş, detayları ileriki kısımlarda açıklanacak olan çevrim süresi (iki ürün çıkışı arasında geçen zaman), öncelik ilişki kısıtları (işlerin teknolojik olarak sırasıyla yapılmasını gerektiren durumlar), önceden belirlenmiş işlem zamanları ve bölgeleme kısıtları (bazı görevlerin aynı veya farklı istasyonda yapılması zorunluluğu) dikkate alınarak gerçekleştirilir. Bölgeleme kısıtları pozitif veya negatif bölgeleme kısıtları olabilir.

Tip-1 montaj hattı dengeleme problemleri, belirli bir çevrim zamanı için işleri, iş istasyonlarına atama yoluyla minimum iş istasyonu sayısını bulmayı amaçlamaktadır.

Tip-2 montaj hattı dengeleme problemlerinde ise amaç işleri, belirli sayıdaki iş istasyonlarına atarken çevrim zamanını minimize etmektir.

Her iki problem tipinde de ortak amaçlardan birisi atıl süreyi en aza indirmektir. Bunun yanında, iş istasyonları arasındaki düzgün iş yükü dağılımı, pozitif veya negatif bölgeleme kısıtlarının sağlanması son yıllardaki çalışmalarda araştırmacılar tarafından dikkate alınmaktadır.

Bu çalışmada dikkate alınan problemler Tip-1 montaj hattı dengeleme problemleridir. Karışık modelli montaj hattı dengeleme problemleri konusunda öncelikle ilgili literatürdeki çalışmalardan bahsedilmiş, daha sonra bu çalışmalar sınıflandırılarak ünitelere ayrılmış ve Karışık modelli montaj hattı dengeleme problemlerinin matematiksel formülasyonu, varsayımları, eksikleri, uygulanmasında karşılaşılan problemler ve örnekleri verilmiştir. Ayrıca ilgili problemler incelenirken paralel iş istasyonları, pozitif veya negatif bölgeleme gibi gerçek uygulamaları yansıtan kısıtlar dikkate alınmıştır.

## 2. MONTAJ HATTI Dengeleme Problemlerine Genel Bakış

### 2.1 Montaj Hattı Dengeleme Problemlerinin Temel Özellikleri

Montaj hattı, işlerin verilen bir çevrim zamanı içerisinde, belirli sırayla iş istasyonlarında gerçekleştirildiği, her istasyonda ürüne değişik parçaların monte edildiği bir üretim sürecidir. Literatürde sık kullanılan bazı kavramlar ise şunlardır:

***İş elemanı (Görev):*** İşin parçalarına ayrılamayan en küçük birimdir.

***İş istasyonu:*** Kendisine ayrılan zaman (çevrim süresi) içerisinde, gerçekleştirmekle yükümlü olduğu işlerin gerçekleştirildiği birimdir.

***Çevrim zamanı:*** Görevleri gerçekleştirebilmesi için iş istasyonlarına ayrılan zamandır. Diğer bir deyişle, iki ürün çıkışı arasında geçen süredir.

***Hat dengeleme:*** Öncelik ilişkileri ve iş istasyonları arasındaki düzgün iş yükü dağılımı gözetilerek, işlerin iş istasyonlarına atanması prosedürüdür [4].

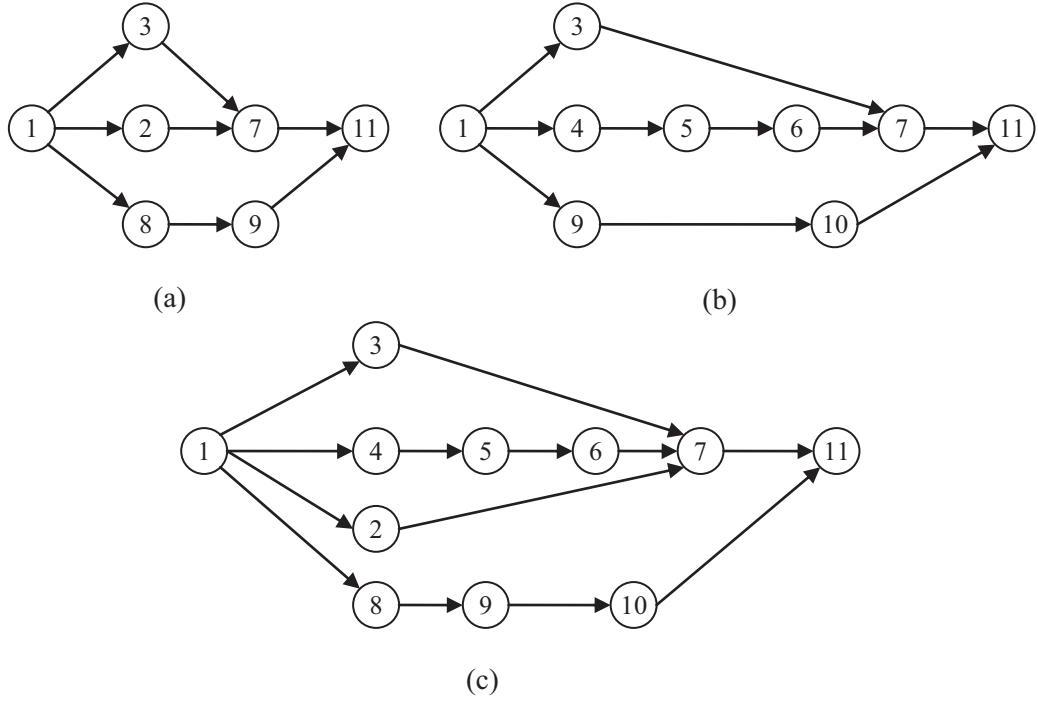
***Denge kaybı (gecikmesi):*** İş istasyonlarının montaj hattına dengesiz dağılımından kaynaklanan atıl süredir.

***En küçük iş elemanı:*** İşin olası en küçük parçasıdır.

***Akış süresi:*** Tüm istasyonlardaki işlemleri tamamlamak için ayrılan süredir.

**Toplam iş kapsamı:** Ürün montaj işlemlerinin sürelerinin toplamıdır. Diğer bir ifade ile, belirli bir ürünün montaj hattına girişi ile çıkışı arasında geçen süredir.

**Öncelik ilişkileri:** Ürünün üretiminde gerçekleştirilmesi gereken görevlerin teknolojik nedenlerle birbirlerinden önce veya sonra tamamlanması gerekliliğidir. Öncelik ilişkileri diyagram veya matris yoluyla ifade edilebilirler. Aşağıdaki şekillerde iki modele ait öncelik ilişkileri diyagramları ve bunların birleştirilmesiyle oluşan diyagram görülmektedir.



**Şekil 2. 1** İki farklı ürüne ait öncelik ilişkileri diyagramının (a ve b) birleştirilmesi (c)

**Bölgeleme Kısıtı:** Aynı veya farklı istasyonlarda yapılması gereken görevleri ifade eder. İki tür bölgeleme kısıtı vardır:

**Pozitif Bölgeleme Kısıtı:** Bazı işlerin aynı istasyonda yapılması gerekliliğidir.

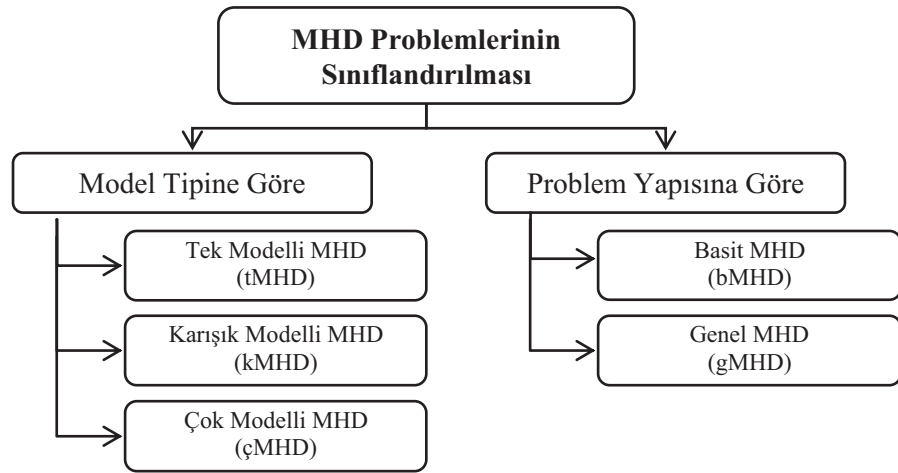
**Negatif Bölgeleme Kısıtı:** Bazı işlerin farklı istasyonlarda yapılması gerekliliğidir.

**Paralel İstasyonlar:** Bazı iş elemanlarının süresinin çevrim süresinden fazla olması veya birden fazla işçiye gerek duyması halinde iş istasyonları birleştirilir ve paralel iş istasyonları oluşturulur.

## 2.2 Montaj Hattı Dengeleme Problemlerinin Sınıflandırılması

Konuyla ilgili literatürde montaj hatları pek çok yönden sınıflandırılmaktadır. Sınıflandırmada en çok kullanılan kriterler ise montaj edilen model tipi, hattın konfigürasyonu ve problemin yapısıdır.

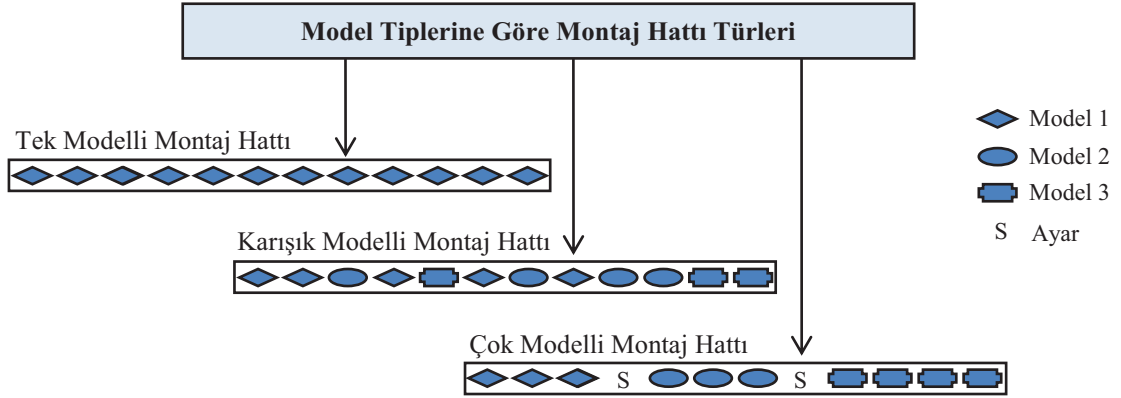
Montaj hattı dengeleme problemleri model yapılarına göre iki gruba ayrılabilir (Şekil 2.2). Tek modellenli montaj hattı dengeleme, çok modellenli montaj hattı dengeleme ve karışık modellenli montaj hattı dengeleme problemleri ilk grubu oluşturmaktadır [1].



Şekil 2. 2 Montaj hattı dengeleme problemlerinin sınıflandırılması

- tMHD: tek bir ürün üretilir,
- kMHD: benzer ürünler veya aynı ürünün değişik modelleri aynı hat üzerinde eş zamanlı ve sürekli üretilir (partiler halinde değil),

- çMHD: birden fazla ürün partiler halinde üretilir ve ürünler arası geçişlerde hazırlık zamanları söz konusudur.



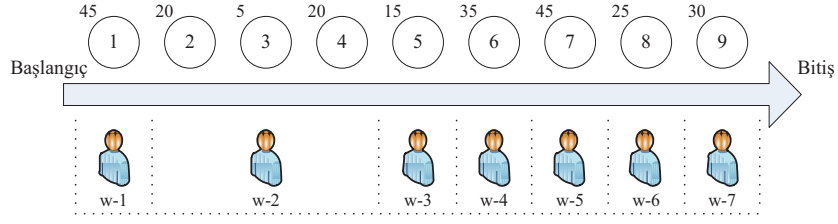
**Şekil 2. 3** Model tiplerine göre montaj hatları sınıflandırması

Diğer grupta ise basit montaj hattı dengeleme (bMHD) ve genel montaj hattı dengeleme (gMHD) problemleri yer alır [1]:

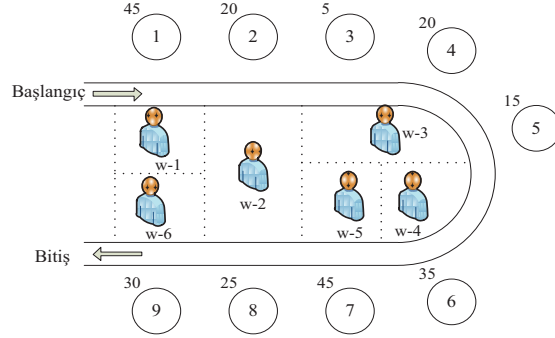
- bMHD: tek modelli montaj hattı dengelemenin özel bir versiyonudur. Sadece bir ürünün üretildiği ve çevrim zamanının sabit, işlem sürelerinin deterministik, montaj hattının düz ve tek yönlü olduğu; bölgeleme ve atama kısıtlarının olmadığı montaj hattı dengeleme problemleridir.
- gMHD: daha gerçekçi şartları yansıtan montaj hattı dengeleme problemleridir. Karışık modeller, paralel iş istasyonları, U-tipi ve iki yönlü hatlar, stokastik işlem zamanları söz konusudur.

Montaj hattı dengeleme problemleri, hattın konfigürasyonuna göre de sınıflandırılmaktadır. Literatürde en fazla üzerine çalışma yapılanlar ise düz (I-tipi) hat ve U-tipi hatlardır. Düz hatlarla ilgili detaylı bilgi ilerleyen bölümlerde verilecektir.

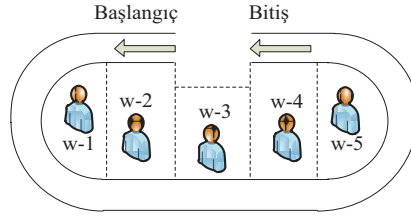




(a)



(b)



(c)

**Şekil 2. 4** Yerleşime göre montaj hattı türleri, (a) düz hat, (b) U-tipi hat ve (c) C-tipi hat

### 2.3 Basit Montaj Hattı Dengeleme (bMHD) Problemlerinin Genel Formülasyonu

Montaj hattı dengelemenin en basit versiyonu Basit Montaj Hattı Dengelemedir. Basit montaj hattı dengeleme (bMHD) de bir, Tek Modelli Montaj Hattı Dengeleme (tMHD) problemidir. Bu hat üzerinde, ürünün sadece tek bir modeli üretilir.

Montaj hattı dengeleme problemleri içerisinde en çok bilineni ve üzerinde en çok çalışma yapılanı şüphesiz bMHD problemleridir. Tabii ki bu problemlerin çözümünde kurulan modeller her zaman gerçek hayat koşullarını yansıtamamaktadır. Fakat diğer karmaşık dengeleme problemlerinde, bMHD modelinden türetilmiş yapılar kullanılmakta veya çözümün bir aşamasında bMHD modellerine başvurulmaktadır.

Modelin varsayımları şunlardır [1]:

- Hat üzerinde homojen ürünlerin kitle üretimi yapılmaktadır.
- Hat seridir, çevrim zamanı sabittir ve paralel hatlar söz konusu değildir.
- Görev süreleri deterministik yapıdadır.
- Tüm istasyonlardaki makine ve işçilerin aynı iş yapma kapasitesine sahip olduğu varsayılmaktadır.
- Bir görev sadece bir istasyona atanabilir ve alt parçalara bölünemez.
- Öncelik ilişkilerinden başka atama kısıtı yoktur.
- Tüm iş istasyonları, tüm görevleri aynı maliyetlerle yapabilmektedir.
- Bir görevin süresi atandığı istasyondan bağımsızdır ve sıralamadan etkilenmemektedir.

Tek modellenli montaj hattı dengeleme problemlerine ait matematiksel model aşağıdaki gibi verilebilir [1].

$i$ : istasyon indeksi ( $i = 1, \dots, m$ )

$j, k$ : görev indeksleri ( $j, k = 1, \dots, m$ )

$m$ : istasyon sayısı

$M$ : maksimum istasyon sayısı

$n$ : görev sayısı

$C$ : çevrim zamanı

$t_j$ :  $j$  görevinin süresi

$S_i$ :  $i$  istasyonuna atanan görevler kümesi

$T_s$ : bir görev sıralaması

$Suc(j)$ :  $j$  görevinin direkt ardıllarının kümesi

$Pre(j)$ :  $j$  görevinin direkt öncüllerinin kümesi

$x_{ij} : \begin{cases} 1, & \text{eğer } j \text{ görevi } i \text{ istasyonuna atanmışsa} \\ 0, & \text{aksi durumda} \end{cases}$

Amaç Fonksiyonu:

$$\min m = \sum_{i=1}^M \max_{j \in S_i} x_{ij} \quad (1 \leq j \leq n) \quad (2.1)$$

Kısıtlar:

$$\sum_{i=1}^M x_{ij} = 1, \quad \forall j \quad (2.2)$$

$$\sum_{i=1}^M i x_{ik} \leq \sum_{i=1}^M i x_{ij}, \quad \forall j, \forall k \in Pre(j) \quad (2.3)$$

$$t(S_j) = \sum_{j \in S_i} t_j = \sum_{j=1}^n t_j x_{ij} \leq C, \quad \forall i \quad (2.4)$$

$$x_{ij} = 0 \text{ veya } 1, \quad \forall i, j \quad (2.5)$$

Bu modelde, amaç fonksiyonu (2.1), istasyon sayısını minimize etmek üzere kurulmuştur. (2.2) kısıtı her görevin sadece bir iş istasyonuna atanabilmesine olanak vermektedir. (2.3) kısıtı öncelik ilişki kısıtlarını temsil etmektedir.  $j$  görevinin öncelikli görevleri  $j$  ile aynı istasyona veya daha önceki bir istasyona atanmalıdır. (2.4) kısıtı, istasyonların mevcut yüklerinin çevrim süresine eşit veya çevrim

süresinden daha az olmasını sağlamaktadır. Son olarak (2.5) kısıtı karar değişkenlerinin 0 veya 1 değerlerini alabileceğini göstermektedir.

Eğer bu modelde  $u_i$  ,  $i$  istasyonunun kullanımını;  $\bar{u}$  ise tüm istasyonların ortalama kullanımını temsil ederse alternatif amaç fonksiyonları (2.8) ve (2.9) eşitliklerindeki gibi olur.

$$u_i = \frac{1}{\max\{t(S_i)\}} t(S_i) \quad 1 \leq i \leq m \quad (2.6)$$

$$\bar{u} = \frac{1}{m} \sum_{i=1}^m u_i \quad (2.7)$$

$$\max E = \frac{1}{mc_T} \sum_{j \in S_i} t_j x_{ij} \quad (2.8)$$

$$\min V = \sqrt{\frac{1}{m} \sum_{i=1}^m (u_i - \bar{u})^2} \quad (2.9)$$

Burada (2.8) eşitliği hat etkinliğini maksimize etmeyi amaçlarken, (2.9) eşitliği iş yükü dağılımı dengesizliğini minimize etmeyi amaçlamaktadır.

#### 2.4 Karışık Modelli Montaj Hattı Dengeleme Problemleri

Gerçeği yansıtmaktan uzak olmasına rağmen literatürde üzerine en çok araştırma yapılmış konu basit montaj hattı dengeleme problemleridir [5].

Fakat günümüz şartlarında müşteri istekleri yönünde üretilen ürünlere olan talep artmaktadır ve bu durum üreticileri daha esnek üretim sistemleri kurmaya zorlamaktadır. Bu yüzden tek modelli montaj hatları yerini daha esnek karışık modelli montaj hatlarına bırakmaktadır. Karışık modelli montaj hatları bu esnekliği, bir ürünün benzer modellerinin aynı hat üzerinde eş zamanlı olarak üretilmesiyle sağlamaktadır [6].

Montaj hattı dengeleme problemlerinin NP-Hard (optimum sonucu bulmanın oldukça zor ve zaman alıcı olduğu) problemler grubunda yer alması nedeniyle genellikle çözümlerde optimumu garanti etmeyen fakat optimuma yakın çözümler üreten sezgisel ve meta sezgisel algoritmalar kullanılmaktadır [7]. Sezgisel ve meta sezgisel algoritmaların kullanımıyla kesin çözümü garanti etmemesine karşın hızlı ve geliştirilmesi kolay oldukları için kısa bir zamanda iyi sonuçlar elde edilebilmektedir [8].

Montaj hattı dengeleme fikri ilk olarak Bryton (1954) tarafından önerilmiş olmasına rağmen klasik montaj hattı dengeleme problemleri ilk defa 1955 yılında Salveson tarafından tanımlanmıştır. Salveson (1955), ilk defa montaj hattı dengeleme problemleri için matematiksel formülasyon geliştirmiş ve bir çözüm metodu önermiştir [9].

Salveson’u takiben yapılan en büyük aday kuralı, sıralı pozisyonel ağırlık metodu (Helgeson ve Birnie, 1961), Hoffman metodu, Kilbridge ve Wester metodu (Kilbridge ve Wester, 1961), COMSOAL (Arcus, 1965), Moodie ve Young metodu (Moodie ve Young, 1965) gibi çalışmalar literatürde en sık rastlanan sezgisellerdir.

İlk karışık modellenmiş montaj hattı dengeleme çalışması Thomopoulos (1967) tarafından yapılmıştır. Bu çalışmada karışık modellenmiş montaj hattı, tek modellenmiş montaj hattı gibi düşünülmüştür. Sonrasında ise Thomopoulos (1970) karışık modellenmiş montaj hatlarında iş istasyonlarının iş yükü dağılımının düzgünleştirilmesi için bir çalışma yapmıştır [7].

Ghosh ve Gagnon (1989), Fokkert ve Kok (1997), ve Scholl (1999) tarafından karışık modellenmiş montaj hatlarıyla ilgili literatür değerlendirmesi çalışmaları yapılmıştır.

Gökçen ve Erel (1998) 0-1 tamsayı formülasyonu geliştirmiş ve bir karışık modellenmiş montaj hattı dengeleme problemi için Erel ve Gökçen (1999) tarafından optimum çözüm bulunmuştur [6].

Kim, Kim, ve Kim (2000) karışık modelli montaj hatlarında sıralama problemi için evrimsel tabanlı bir algoritma önermişlerdir [26].

Karabatı ve Sayın (2003) maksimum alt çevrim zamanlarının minimizasyonu amaçlı bir matematiksel model ve sezgisel yöntem önermişlerdir.

İstasyon sayısının minimizasyonu amacıyla Bukchin and Rubinovitz (2006) geri dönüşlü dal-sınır algoritması, Haq, Jayaprakash, ve Rengarajan (2006) hibrid genetik algoritma geliştirmişlerdir.

Xu ve Xiao (2009) stokastik zamanlı karışık modelli montaj hatlarının dengelenmesi amacıyla kararlı bir optimizasyon yaklaşımı geliştirmişlerdir [7].

Son zamanlarda ise Akpınar ve Bayhan (2011)'in geliştirdiği bir hibrid genetik algoritma ve Yağmahan (2011) tarafından önerilen çok amaçlı karınca koloni optimizasyonu yaklaşımı, gerçek hayat problemlerini daha iyi yansıtan paralel istasyonlar ve bölgeleme kısıtlarını da dikkate alarak karışık modelli montaj hattı dengeleme yöntemleriyle ilgili literatürdeki yerlerini almışlardır [25].

tMHD-I problemleri konusunda pek çok çalışma yapılmıştır ve bunlar üç gruba ayrılabilir: matematiksel programlama, sezgisel ve meta-sezgiseller, ve hibrid yaklaşımlar. [10] ve [11]'den yararlanılarak son 20 yılda konu ile ilgili yapılmış çalışmalar, hat konfigürasyonu ve kullanılan metodoloji bilgileriyle birlikte Tablo 2.1'de görülmektedir.

Literatürde üç tip kMHD problemi tanımlanmıştır [12]:

- kMHD-I: verilen bir çevrim zamanı için istasyon sayısının minimizasyonu,
- kMHD-II: verilen istasyon sayısı için çevrim zamanının minimizasyonu,
- kMHD-E: çevrim zamanı ve istasyon sayısının birlikte minimizasyonu.

**Tablo 2. 1** kmHD-I literatürüne genel bakış

Çalışma	Hat Yerleşimi					Paralel İstasyon	Metodoloji
	Düz	U-Tipi	Paralel Hatlar	İki Yönlü	Paralel İstasyon		
Bard (1989)	X					E	Dinamik programlama
Bartholdi (1993)				X			Sezgisel prosedür
Inman ve Leon (1994)	X					E	Simülasyon modeli
McMullen (1997)	X		X			E	Sezgisel prosedür
Askin ve Zhou (1997)	X					E	Doğrusal olm. tamsayılı prog., sezgisel
Gökçen ve Erel (1997)	X						0-1 hedef prog.
McMullen ve Frazier (1997)	X					E	Sezgisel
Gökçen ve Erel (1998)	X						0-1 tamsayılı prog.
Suer (1998)	X		X			E	Doğrusal programlama modeli
Urban (1998)		X					Doğrusal programlama modeli
Sarker ve Pan (1998)	X						Doğrusal programlama modeli
Scholl ve Klein (1999)		X					Dal-Sınır algoritması
Kim ve ark. (1999)				X			Matematiksel model
Erel ve Gökçen (1999)	X						Ağ Programlama
Kim ve ark. (2000)				X			Genetik algoritma
Miltenburg (2001)		X					Literatür incelemesi ve bir uygulama
Lee ve ark. (2001)				X			Sezgisel prosedür
Rekiek ve ark. (2001)	X						Genetik algoritma
Sarker ve Pan	X						Doğrusal programlama modeli
Vilarinho ve Simaria (2002)	X					E	Matematiksel model, tavlama benz.
Bukchin ve ark. (2002)	X					E	Sezgisel prosedür
Karabatı ve Sayın (2003)	X						Sezgisel prosedür
McMullen ve Tarasewich (2003)	X					E	Karınca koloni optimizasyonu,
Gökçen ve Ağpak (2004)		X					Doğrusal programlama modeli
Urban ve Chiang (2004)		X					Doğrusal programlama modeli
Kim ve ark. (2004)		X					Genetik algoritma
Simaria ve Vilarinho (2004)	X					E	Genetik algoritma
Hop (2006)	X						Bulanık doğrusal model, Sezgisel prosedür
Beker ve Scholl (2004)	X	X	X	X	X		Literatür incelemesi
Bukchin ve Rabinowitch (2006)	X						Dal-Sınır algoritması
Bock (2006)	X						Dağıtılmış arama prosedürü
Haq ve ark. (2006)	X						Hibrid genetik algoritma
Kara ve ark. (2007)		X					Tavlama benzetimi
Bock (2008)	X						Tabu arama
Simaria ve Vilarinho (2009)				X			Karınca koloni optimizasyonu
Ozcan ve Toklu (2009)				X			Matematiksel model, tavlama benzetimi
Ozcan (2010)				X			Stokastik karışık modelli ve tavlama benz.
Akpınar ve Bayhan (2011)	X						Hibrid genetik algoritma
Yagmahan (2011)	X						Çok amaçlı karınca koloni optimizasyonu

\*E: Evet

Bu çalışma kapsamında kMHD-I problem grubu ile ilgilenilecektir ve kısaca kMHD şeklinde ifade edilecektir.

Tek modelli montaj hattı dengeleme problemleri NP-Hard tipi karmaşık problemlerdendir. Karışık modelli montaj hattı dengeleme problemleri de, tek modelli montaj hattı dengeleme problemlerinin daha karmaşık bir çeşidi olduğu için NP-Hard grubu problemler arasında yer almaktadır. Çünkü tMHD problemlerinde, kMHD problemlerinde ele alınan modellerden yalnızca birisi ile ilgilenilmektedir [13]. Yukarıda bahsedilen kMHD problemlerinin her üç versiyonu da NP-Hard problemlerdir.

## 2.5 Karışık Modelli Montaj Hattı Dengeleme (kMHD) Problemlerinin Genel Formülasyonu

kMHD problemine ait 0-1 tamsayılı programlama modeli aşağıda verilmiştir [14]:

- $N$ , birleştirilmiş öncelik diyagramındaki görevlerin sayısıdır.
- $M$ , hat üzerinde montaj edilen modellerin sayısıdır.
- $D_m$ ,  $m$  modelinin  $P$  planlama dönemindeki talebidir.
- $q_m$ ,  $m$  modelinin oransal ihtiyacıdır ve  $q_m = D_m / \sum_{p=1}^M D_p$  eşitliğinden hesaplanır.
- $S$ , iş istasyonu sayısıdır.
- $C$ , hattın çevrim süresidir ve  $C = P / \sum_{m=1}^M D_m$  eşitliğinden hesaplanır.
- $t_{im}$ ,  $i$  görevinin  $m$  modeli için işlem süresidir.
- $Suc_i$ ,  $i$  görevinden önce tamamlanamayacak görevlerin kümesidir ( $i$  görevinin ardılları).



- $x_{ik} = \begin{cases} 1, & \text{eğer } i \text{ işi } k \text{ istasyonuna atanmışsa} \\ 0, & \text{aksi durumda} \end{cases}$

Amaç Fonksiyonu:

$$\text{Min} \sum_{k=1}^S \left( C - \sum_{m=1}^M q_m \sum_{i=1}^N t_{im} x_{ik} \right) \quad (2.10)$$

Kısıtlar:

$$\sum_{k=1}^S x_{ik} = 1 \quad i = 1, \dots, N \quad (2.11)$$

$$\sum_{k=1}^S kx_{ik} - \sum_{k=1}^S kx_{jk} \leq 0 \quad i \in N, j \in \text{Suc}_i \quad (2.12)$$

$$\sum_{i=1}^N t_{im} x_{ik} \leq C \quad k = 1, \dots, S; m = 1, \dots, M \quad (2.13)$$

$$x_{ik} \in \{0,1\} \quad i = 1, \dots, N; k = 1, \dots, S \quad (2.14)$$

Amaç fonksiyonu (2.10) her modelin üretim oranına göre ağırlıklı atıl zamanları minimize eder. Bu amaç, hem istasyon sayısının minimizasyonunu amaçlayan kMHD-I problemleri hem de çevrim zamanının minimizasyonunu amaçlayan kMHD-II problemleri için ortaktır.

(2.11) numaralı kısıt her görevin yalnızca bir istasyona atanmasını sağlar. Tüm modellere ait ortak görevler aynı istasyonda gerçekleştirilir.

(2.12) numaralı kısıt öncelik ilişkileri kısıtlarını sağlamaktadır. Bir görevin ardılarının, kendisinden önceki istasyonlara atanması önlenmektedir.

(2.13) numaralı kısıt kapasite kısıtıdır ve bir istasyonun iş yükünün çevrim süresini aşmasını engellemektedir.

(2.14) numaralı kısıt karar değişkenlerinin 0-1 değer almasını sağlamaktadır.

Temel amaç iş istasyonu sayısı veya çevrim süresinin minimizasyonu olmasına karşın bunun yanında istasyonlar arası iş yükü dengesizliğinin minimizasyonu da yapılan son çalışmalarda dikkate alınan amaçlardan birisidir.

### **3. MONTAJ HATTI DEGELEME PROBLEMLERİ İÇİN ÇÖZÜM YAKLAŞIMLARI VE GENETİK ALGORİTMALAR**

#### **3.1 Çözüm Yaklaşımlarının Sınıflandırılması**

Daha önce de belirtildiği gibi montaj hattı, parçaların öncelik ilişkilerine dayanarak belirli bir biçimde monte edildiği ve bir istasyondan diğer istasyona ilerleyerek hattın sonunda nihai ürüne dönüştüğü bir üretim sürecidir. Dengeli bir montaj hattı geliştirmek için ürünün elde edilmesinde gerekli toplam işi tüm istasyonlara dağıtmak gerekmektedir. Böylece istasyonlar kendilerine verilen işi eşzamanlı bir biçimde yerine getirebileceklerdir. Eğer hat mükemmel dengelenmişse her istasyon kendisine atanan işleri gerçekleştirmek için eşit zamana ihtiyaç duyacaktır. Fakat bu durum gerçekte nadiren gerçekleşmektedir ve işleri tamamlamak için en uzun zamana gerek duyan istasyonun süresi, çevrim zamanını belirlemektedir [15].

Literatürde en çok kullanılan montaj hattı dengeleme yöntemlerinden bazıları Şekil 3.1'deki gibi sınıflandırılabilir.

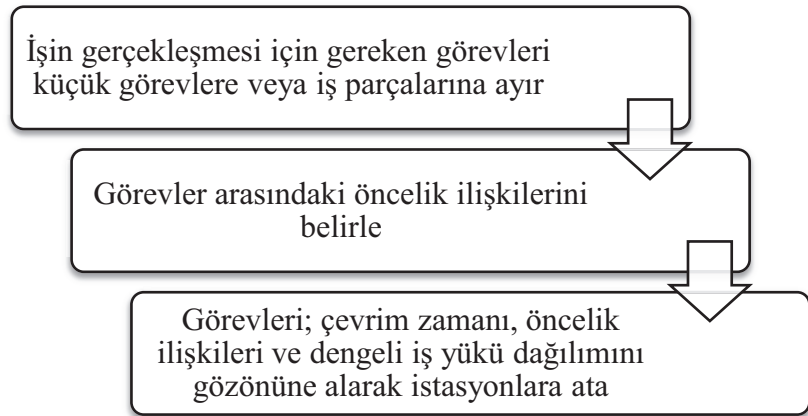
Klasik optimizasyon teknikleri, küçük boyuttaki problemler için optimum sonucu üretmekle birlikte, büyük boyutlu problemlerde optimum sonuca ulaşmakta zorluk çekmektedir. Çünkü problemi çözme süresi modelde kullanılan görev sayısı ile orantılı olarak üstsel şekilde artmaktadır.



**Şekil 3. 1** Montaj hattı dengelemede kullanılan bazı çözüm yaklaşımlarının sınıflandırılması

Sezgisel ve meta-sezgisel yöntemler ise rastgele araştırma yöntemleridir ve optimum sonucu garanti edemezler fakat optimum sonuca yakın değerler elde ederler. Bu yönüyle sezgisel ve meta-sezgisel yöntemlerin büyük boyutlu problemlerde kullanımı yaygındır.

Dengeleme yaklaşımlarında kullanılan ortak prosedür Şekil 3.2’de verildiği gibidir.



**Şekil 3. 2** Çözüm yaklaşımlarının ortak aşamaları

Önceki bölümde basit montaj hattı dengeleme problemlerine yönelik 0-1 tamsayı programlama modeli verilmişti. İzleyen kısımda montaj hattı dengeleme problemlerine yönelik sezgisel ve meta-sezgisel çözüm yöntemlerinden olan En Büyük Aday Kuralı, Sıralı Pozisyonel Ağırlık Yöntemi, COMSOAL (Computer Method for Sequencing Operations for Assembly Lines), Kilbridge&Wester Yöntemi ve Genetik Algoritmalar, çözüm yaklaşımları ve temel fonksiyonları açısından kısaca açıklanacak ve görsel öğelerle desteklenecektir.

### **3.2 En Büyük Aday Kuralı (Largest Candidate Rule - LCR)**

En Büyük Aday Kuralı (Largest Candidate Rule - LCR) ve Sıralı Pozisyonel Ağırlık Yöntemi (Ranked Positional Weighted Method - RPW), montaj hattı dengelemede kullanılan en basit yöntemlerdendir. Bu yöntemler, Sule (2008)'in belirttiği gibi izleyen kısımda kısa bir şekilde özetlenerek örneklendirilecektir.

#### **Adım 1:**

- Görevler, görev zamanlarına göre azalan şekilde sıralanır.
- Her görev için ilgili öncelikli görevler belirtilir.

#### **Adım 2:**

- İlk istasyonla başlanır ve kalan istasyonlar için Adım 3'te belirtildiği gibi işlemler uygulanır.

#### **Adım 3:**

- Görev listesinin en üstündeki görev ile başlanır ve atanmamış ilk uygun görev ilgili istasyona atanır.

- Bir görev atandığı zaman, o görev, önceliği olan diğer tüm görevlerin öncelikli görevler listesinden çıkartılır.
- Bir görevin uygun olabilmesi için, ya öncelikli görevlerinin olmaması ya da tüm öncelikli görevlerin tamamlanmış olması gerekir.
- Bir görevin bir istasyona atanabilmesi için, görevin süresi ile o istasyona atanmış görevlerin toplam süresinin, çevrim süresini aşmaması gerekir. Eğer aşarsa atama işlemi yapılmadan diğer göreve geçilir.
- Eğer atanabilecek uygun görev yoksa Adım 5'e gidilir.

**Adım 4:**

- İstasyona ataması yapılan görev, listeden silinir. Eğer listede başka atanacak görev kalmadıysa Adım 6'ya gidilir, aksi halde Adım 3'e gidilir.

**Adım 5:**

- İstasyon numarasının bir artırılması ile yeni bir istasyon oluşturulur ve Adım 3'e gidilir.

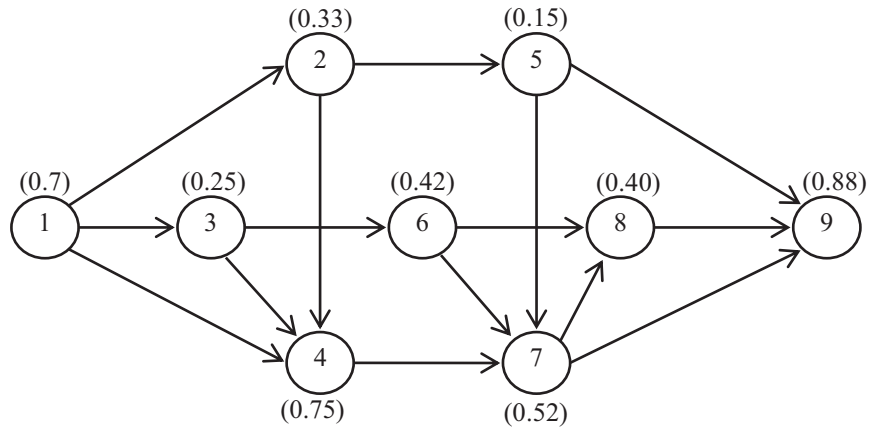
**Adım 6:**

- Tüm görevler atanmıştır. Mevcut istasyon numarası, gerekli toplam istasyon sayısını belirtmektedir.
- Hangi görevlerin hangi istasyonlarda yapılması gerektiği belirlenmiştir.
- En büyük toplam zamana sahip istasyonun zamanı, çevrim zamanı olarak kabul edilir.

Bir örnek, problemin çözülmesi ve yaklaşımın anlatımı için açıklayıcı olacaktır.

Öncelik ilişkileri ve zamanları izleyen şekilde verilen dokuz görevden oluşan bir ürün için 500 dakikada 526 adet talep söz konusudur. Bu durumda teorik çevrim zamanı (3.1) eşitliği ile hesaplanabilir.

$$C = \frac{500}{526} = 0.95 \text{ dakika/adet} \quad (3.1)$$



Şekil 3. 3 Örneğe ait öncelik ilişkileri diyagramı ve görev zamanları

Tablo 3. 1 Örneğe ait görev zamanları ve öncelik ilişkileri

Görev	Görev Zamanı	Öncelikli Görev(ler)
1	0.70	-
2	0.33	1
3	0.25	1
4	0.75	1, 2, 3
5	0.15	2
6	0.42	3
7	0.52	4, 5, 6
8	0.40	6, 7
9	0.88	5, 7, 8

Öncelikle, görevler zamanlarına göre azalan şekilde sıralanır.

Görev	Görev Zamanı	Öncelikli Görev(ler)
9	0.88	5, 7, 8
4	0.75	1, 2, 3
1	0.70	-
7	0.52	4, 5, 6
6	0.42	3
8	0.40	6, 7
2	0.33	1
3	0.25	1
5	0.15	2

En Büyük Aday Kuralı prosedürleri devam ettirilir ve görevler istasyonlara atanır. Sonuçta beş istasyona gerek duyulmaktadır. Çevrim süresini belirleyen istasyon ise tam kapasite ile çalışan 1 numaralı istasyondur.

Oluşturulan istasyonlar ve bu istasyonlara atanan görevler Tablo 3.2’de verilmektedir.

**Tablo 3. 2** LCR kuralına göre istasyonlara yapılan atama

İstasyon	Atanan Görev	Görev Zamanı	Kümülatif Zaman
1	1	0.70	0.70
	3	0.25	0.95
	6	0.42	0.42
2	2	0.33	0.75
	5	0.15	0.90
3	4	0.75	0.75
4	7	0.52	0.52
	8	0.40	0.92
5	9	0.88	0.88

Hattın verimliliği ( $e$ ); (3.2), (3.3) ve (3.4) eşitlikleri yardımıyla % 92.63 olarak hesaplanır.

$$e = (1 - p) \times 100 \quad (3.2)$$

$$p = \frac{(Toplam\ istasyon\ sayısı \times Çevrim\ Süresi) - (Gerekli\ toplam\ zaman)}{(Toplam\ istasyon\ sayısı \times Çevrim\ Süresi)} \quad (3.3)$$

$$e = \left[ 1 - \frac{5 \times 0.95 - 4.4}{5 \times 0.95} \right] \times 100 = \% 92.63 \quad (3.4)$$



### 3.3 Sıralı Pozisyonel Ağırlık Yöntemi (Ranked Positional Weighted Method - RPW)

Bir önceki yöntemde görevler, zamanlarına göre azalan şekilde sıralanmıştı. Bu metotta ise tüm işlerin tamamlanmasındaki önem derecelerine (ağırlıklarına) göre sıralanmaktadır. Bir görevin önem derecesi, o görevin süresi ile öncelik ilişkileri diyagramında kendisini izleyen görevlerin zamanlarının toplamı ile ölçülür.

Aynı örneği kullanacak olursak, örneğin 6 numaralı görevin önem derecesi; 6 numaralı görevin süresi ile kendisini izleyen 7, 8 ve 9 numaralı görevlerin sürelerinin toplamına ( $0.52 + 0.40 + 0.88 + 0.42 = 2.22$ ) eşittir.

Diğer görevlerin de ağırlıkları hesaplandığında Tablo 3.3 oluşturulur.

**Tablo 3. 3** RPW kuralına göre görevlerin pozisyonel ağırlıkları

Görev	PW	Görev Zamanı
1	4.40	0.70
2	3.45	0.33
3	3.22	0.25
4	2.55	0.75
6	2.22	0.42
5	1.95	0.15
7	1.80	0.52
8	1.28	0.40
9	0.88	0.88

Görevler ağırlıklarına göre azalan şekilde sıralandıktan sonra, En Büyük Aday Kuralının 2-6 adımları aynen uygulanır.

**Tablo 3. 4** RPW kuralına göre istasyonlara yapılan atama

İstasyon	Atanan Görev	Görev Zamanı	Kümülatif Zaman
1	1	0.70	0.70
	3	0.25	0.95
2	2	0.33	0.33
	6	0.42	0.75
	5	0.15	0.90
3	4	0.75	0.75
4	7	0.52	0.52
5	8	0.40	0.92
	9	0.88	0.88

Sonuçta yine çevrim süresi 0.95, istasyon sayısı 5, verimlilik ise % 92.63 çıkmıştır. Her iki yöntemde de sonuçların aynı çıkması tamamen tesadüftür.

$$e = \left[ 1 - \frac{5 \times 0.95 - 4.4}{5 \times 0.95} \right] \times 100 = \% 92.63 \quad (3.5)$$

### 3.4 COMSOAL (Computer Method for Sequencing Operations for Assembly Lines)

Arcus (1965) tarafından geliştirilen algoritma, olası dengeleme simülasyonları için bilgisayar kullanılarak iteratif yapıyla devam ettirilir. COMSOAL, hızlı hesaplama için listeler kullanır. Algoritmayı açıklayan prosedür şu şekildedir [16]:

Adım 1: Görevlerin öncelik ilişkisi olduğu görev sayılarını içeren Liste A'yı oluştur.

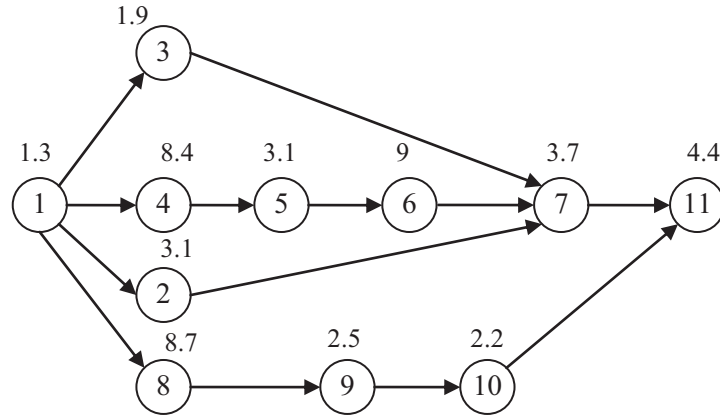
Adım 2: Öncelik ilişki sayısı "0" olan görevleri Liste B'ye al.

Adım 3: Liste B'den rastgele bir görev seç ve sıralamaya ekle. Daha sonra bu görevi Liste A'dan sil.

Adım 4: Liste A'daki görevlerin öncelik ilişki sayılarını yeniden hesapla.

Adım 5: Liste A'da atanmayan görev varsa Adım 2'ye git, aksi halde Adım'6 ya git

Adım 6: Sonucu al ve bitir.



Şekil 3. 4 Görevlerin öncelik ilişkileri ve süreleri

Tablo 3. 5 Liste A (solda) ve Liste B (sağda)

Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
1	0	1	0
2	1		
3	1		
4	1		
5	2		
6	3		
7	6		
8	1		
9	2		
10	3		
11	10		

Bu süreç, bütün görevler sıralamaya dahil edilene dek devam ettirilir. Daha sonra bu sıralamadaki işler sırayla iş istasyonlarına ayrılır. Gerekli iş istasyonu sayısı hesaplanır ve bir önceki çözümle karşılaştırılır. Eğer iyileşme varsa yeni sıralama saklanır ve eski sıralama yok edilir.

Yukarıda öncelik ilişkileri ve süreleri verilen görevleri COMSOAL yöntemini kullanarak  $C = 9$  için dengeleyelim.

*İterasyon 1:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
1	0	1*	0
2	1		
3	1		
4	1		
5	2		
6	3		
7	6		
8	1		
9	2		
10	3		
11	10		

\*: Seçim

Liste B'den görüleceği gibi seçilebilecek tek görev 1 numaralı görevdir. Böylece 1 numaralı görev seçilir ve sıralamaya dahil edilir:

Sıralama: 1 olur.

*İterasyon 2:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
2	0	2	0
3	0	3*	0
4	0	4	0
5	1	8	0
6	2		
7	5		
8	0		
9	1		
10	2		
11	9		

\*: Seçim

Seçilebilecek görevler 2, 3, 4 ve 8 numaralı görevlerdir. Bunlar arasından rastsal olarak birisi seçilir ve sıralamaya dahil edilir. Burada 3 numaralı görev seçilmiştir.

Sıralama: 1-3 olur.

*İterasyon 3:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
2	0	2*	0
4	0	4	0
5	1	8	0
6	2		
7	4		
8	0		
9	1		
10	2		
11	8		

\*: Seçim

Seçilebilecek görevler 2, 4 ve 8 numaralı görevlerdir. Bunlar arasından rastsal olarak birisi seçilir ve sıralamaya dahil edilir. Burada 2 numaralı görev seçilmiştir.

Sıralama: 1-3-2 olur.

*İterasyon 4:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
4	0	4	0
5	1	8*	0
6	2		
7	3		
8	0		
9	1		
10	2		
11	7		

\*: Seçim

4 ve 8 numaralı görevler arasından 8 numaralı görev seçilmiştir.

Sıralama: 1-3-2-8 olur.

*İterasyon 5:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
4	0	4*	0
5	1	9	0
6	2		
7	3		
9	0		
10	1		
11	6		

\*: Seçim

Seçilen görev 4 numaralı görevdir.

Sıralama: 1-3-2-8-4 olur.

*İterasyon 6:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
5	0	5*	0
6	1	9	0
7	2		
9	0		
10	1		
11	5		

\*: Seçim

5 numaralı görev seçilmiştir.

Sıralama: 1-3-2-8-4-5 olur.

*İterasyon 7:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
6	0	6	0
7	1	9*	0
9	0		
10	1		
11	4		

\*: Seçim

9 numaralı görev seçilmiştir.

Sıralama: 1-3-2-8-4-5-9 olur.

*İterasyon 8:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
6	0	6	0
7	1	10*	0
10	0		
11	3		

\*: Seçim

10 numaralı görevin seçilmesi sonucu;

Sıralama: 1-3-2-8-4-5-9-10 olur.

*İterasyon 9:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
6	0	6*	0
7	1		
11	2		

\*: Seçim

Seçilebilecek tek alternatif 6 numaralı görevdir.

Sıralama: 1-3-2-8-4-5-9-10-6 olur.

*İterasyon 10:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
7	0	7*	0
11	1		

\*: Seçim

Seçilebilecek tek alternatif 7 numaralı görevdir.

Sıralama: 1-3-2-8-4-5-9-10-6-7 olur.

*İterasyon 11:*

Liste A		Liste B	
Görev	Öncelik İlişkisi	Görev	Öncelik İlişki Sayısı
11	0	11*	0

\*: Seçim

Son olarak 11 numaralı görev seçilerek algoritma tamamlanır.

Nihai sıralama: 1-3-2-8-4-5-9-10-6-7-11 olur.

Daha sonra bu sıralamaya göre işler istasyonlara atanır.

Yapılan atamalar sonucu istasyonlara atanan görevler, istasyonların iş yükü ve atıl zamanları Tablo 3.6’da görülmektedir. En fazla atıl zamanı olan istasyon 1 numaralı istasyondur. 5 numaralı istasyon ise kapasitesinin tamamını kullanmaktadır ve atıl zamanı 0’dır. Toplam atıl zaman 5.3 zaman birimidir.

**Tablo 3. 6** COMSOAL ile yapılan çözüm sonucu gerçekleşen atamalar

İstasyon	Görev	Görev Süresi	İst. Süresi	Atıl Zaman
1	1	1.3	6.3	2.7
	3	1.9		
	2	3.1		
2	8	8.7	8.7	0.3
3	4	8.4	8.4	0.6
4	5	3.1	8.2	0.8
	9	2.5		
	10	2.2		
5	6	9	9	0
6	7	3.7	8.1	0.9
	11	4.4		
Toplam		48.3	48.3	5.3

Yapılan atamalar sonucu oluşan denge kaybı (3.6) eşitliği yardımıyla hesaplanabilir.

$$Denge kaybı = \frac{S(C) - \sum t}{S(C)} \times 100 = \frac{6(9) - 48.3}{6(9)} \times 100 = \% 10 \quad (3.6)$$

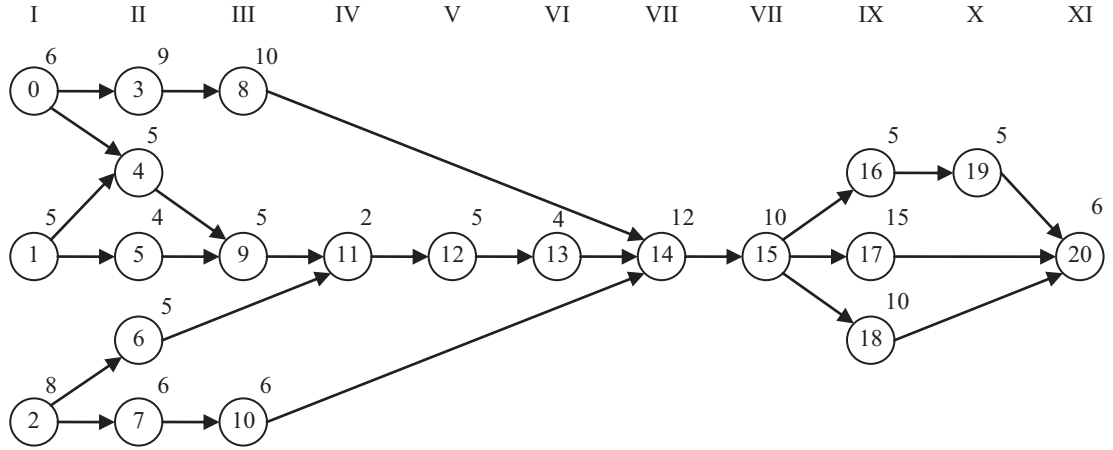


Arcus daha sonra COMSOAL algoritmasını daha da geliştirmiştir. Liste B'den, istasyondaki kullanılabilir süreyi aşmayan görevleri yeni bir listeye (Liste C) alarak, görevlerin rassal seçim işlemlerini bu listeden yapmıştır.

### 3.5 Kilbridge ve Wester Yöntemi

Bu basit sezgisel yöntem bir örnek [16] üzerinden açıklanacaktır.

Öncelik ilişkileri ve süreleri belli olan 21 göreve ait montaj süreci Şekil 3.5'te verilmektedir. Bu öncelik ilişkileri diyagramında görevleri belirten daireler mümkün olduğunca sol tarafa yerleştirilmeye çalışılmıştır. Diyagramın üzerindeki ifadeler kolon numaralarını göstermektedir. I kolonunda bulunan görevler hemen başlayabilir. II kolonunda bulunan görevler ise ancak I kolonundaki bir veya daha fazla görevin gerçekleştirilmesinden sonra başlayabilir.



Şekil 3. 5 Öncelik ilişkileri diyagramı

Öncelik ilişkileri diyagramındaki görevler çizelge şeklinde ifade edilerek işlemlere devam edilebilir. Tablo 3.7'de bulunan (c) kolonu, görevlerin yana doğru transfer edilebilirliğini göstermektedir. Örneğin 6 numaralı görev II kolonunda bulunduğu halde öncelik ilişkilerini ihlal etmeden III numaralı kolona transfer edilebilir. 8 numaralı ve 10 numaralı görevler kolon IV veya VI arasında herhangi

bir kolonda gerçekleştirilebilir. Bu durumda 3 ve 7 numaralı görevler de III ve V arasında transfer edilebilir.

**Tablo 3. 7** Çözümüne yönelik bilgilerin çizelge şeklinde ifade edilmesi

Kolon no (a)	Görev no (b)	Görevin transfer edilebilirliği (c)	Görev süresi (d)	Kolonun süresi (e)	Toplam süre (f)
I	0		6		
	1		5		
	2		8	19	19
II	3	III-V (8 ile)	9		
	4		5		
	5		4		
	6	III	5		
	7	III-V (10 ile)	6	29	48
III	8	IV-VI	10		
	9		5		
	10	IV-VI	6	21	69
IV	11		2	2	71
V	12		5	5	76
VI	13		4	4	80
VII	14		12	12	92
VIII	15		10	10	102
IX	16		5		
	17	X	15		
	18	X	10	30	132
X	19		5	5	137
XI	20		6	6	143

Çevrim zamanının 36 br zaman olduğunu varsayarsak yapacağımız dengeleme işlemi şu şekilde ilerler:

1. (f) kolonunda çevrim zamanı olan 36'ya eşit bir süre var mı? Hayır.
2. (f) kolonundaki 36'dan küçük olan en büyük süreyi seç, I kolonundaki 19.
3. 36'dan 19'u çıkar=17.
4. (II) kolonundaki görevlerden tek veya toplam şeklinde 17'ye eşit olan var mı? Hayır. En yakın süre 4, 6, ve 7 numaralı görevlerin süreleri toplamı olan 16. Böylece ilk istasyonun süresi=35.

5. (f) kolonundaki 36'dan büyük değerlerden en küçük olanını seç, I ve II kolonları için 48.
6. Süreyi 36'ya indirmek için I veya II kolonlarından herhangi bir görev transfer edilebilir mi? Hayır. Fakat 3 numaralı görev (8 ile birlikte) ve 6 numaralı görev transfer edilirse istasyon süresi 34 olur.
7. (f) kolonundaki diğer en büyük süreyi seç, I, II ve III kolonları için 69.
8. I, II ve III kolonlarından herhangi bir görev çıkartılarak 36'ya ulaşılabilir mi? Hayır, en yakın değer olarak 3, 7, 8 ve 10 numaralı görevlerin çıkartılmasıyla 38'e ulaşılabilir, çevrim süresinden büyük.
9. (f) kolonundaki büyük değerler dikkate alındığında daha iyi bir sonuca ulaşıldı mı? Hayır.
10. Bir önceki en iyi çözümü al, 4. adımdaki 35 br istasyon zamanına sahip çözüm.
11. Tabloyu yeniden çiz ve atamaları yap. (f) kolonundaki toplam zamanları yeniden hesapla.

Kolon no (a)	Görev no (b)	Görevin transfer edilebilirliği (c)	Görev süresi (d)	Kolonun süresi (e)	Toplam süre (f)
I	0		6		
	1		5		
	2		8		
II	4		5	19	(35) İst 1.
	6		5		
	7		6		
III	3	III-V (8 ile)	9	24	24
	5		4		
	9	IV-VI	5		
	10		6		
IV	8	V-VI	10	12	36
	11		2		
V	12		5	5	41
VI	13		4	4	45
VII	14		12	12	57
VIII	15		10	10	67
IX	16		5	30	97
	17	X	15		
	18	X	10		
X	19		5	5	102
XI	20		6	6	108

12. (f) kolonunda 36'ya eşit süre var mı? Evet. Kolon III ve IV.

13. Bu kolonlardaki görevleri istasyona ata ve tabloyu yeniden oluştur.

Kolon no (a)	Görev no (b)	Görevin transfer edilebilirliği (c)	Görev süresi (d)	Kolonun süresi (e)	Toplam süre (f)
	0		6		
	1		5		
	2		8		
	4		5	19	(35)
	6		5		İst 1.
	7		6		
III	3		9		
	5		4		
	9		5		
	8		10		(36)
	10		6		İst. 2
IV	11		2	36	
V	12		5	5	5
VI	13		4	4	9
VII	14		12	12	21
VIII	15		10	10	31
	16		5		
IX	17	X	15		
	18	X	10	30	61
X	19		5	5	66
XI	20		6	6	72

14. (f) kolonunda 36'ya eşit süre var mı? Hayır.

15. (f) kolonundaki 36'dan küçük olan en büyük süreyi seç, kolon V, VI, VII ve VIII'in toplamları 31.

16. 31'i 36'dan çıkar=5.

17. Diğer kolondaki (IX) bir veya daha fazla görevlerin toplamı 5'e eşit mi? Evet.  
16 numaralı görev.

18. İlgili kolonları ve görevi istasyona ata ve tabloyu yeniden oluştur.

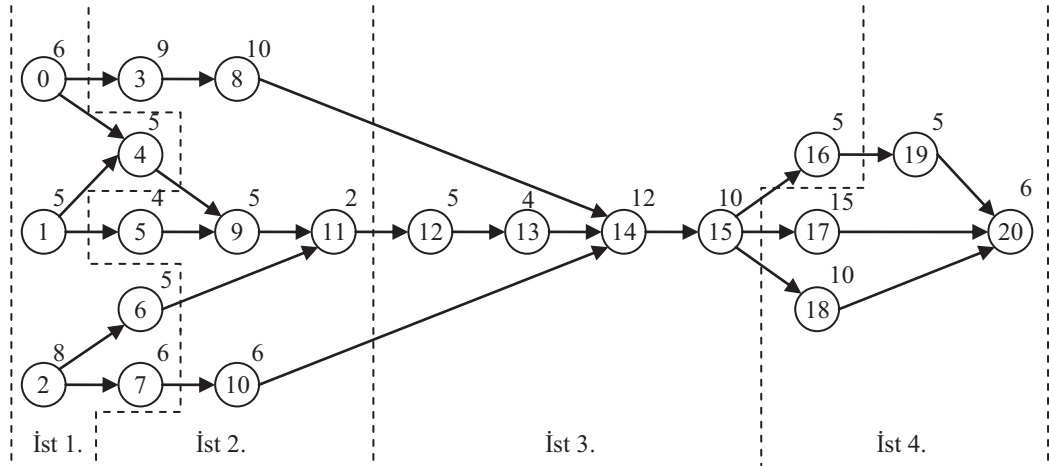
Kolon no (a)	Görev no (b)	Görevin transfer edilebilirliği (c)	Görev süresi (d)	Kolonun süresi (e)	Toplam süre (f)
	0		6		
	1		5		
	2		8		
	4		5	19	(35)
	6		5		İst 1.
	7		6		
	3		9		
	5		4		
	9		5		
	8		10		(36)
	10		6		İst. 2
	11		2	36	
V	12		5		
VI	13		4		
VII	14		12		(36)
VIII	15		10		İst. 3
IX	16		5	36	
IX	17	X	15		
	18	X	10	25	25
X	19		5	5	30
XI	20		6	6	36

19. (f) kolonunda 36'ya eşit süre var mı? Evet. IX, X ve XI kolonları.

20. Bu kolonlardaki görevleri istasyona ata.

Böylece tüm görevler istasyonlara atanmıştır. Yapılan atamalar sonucu denge gecikmesi (3.7) eşitliği yardımıyla hesaplanabilir.

$$\frac{S(C) - \sum t}{S(C)} \times 100 = \frac{4(36) - 143}{4(36)} \times 100 = \% 0.7 \quad (3.7)$$



Şekil 3. 6: İstasyonlara atanmış görevler

### 3.6 Genetik Algoritmalar

Klasik optimizasyon teknikleri, küçük boyutlu problemlerin çözümünde kısa sürede optimum sonucu verebilmektedir. Fakat bu klasik optimizasyon tekniklerinin büyük boyutlu ve NP-Hard sınıftaki problemlerin (gezin satıcı problemleri, paketleme problemleri vb.) çözümünde zorlandığı bilinmektedir. Bu zorluk problemin karmaşıklığının yanında boyutunun da büyük olmasından kaynaklanmaktadır. NP-Hard problemlerin çözümünde sezgisel ve meta-sezgisel teknikler her zaman optimum sonuçlar vermese de, kısa zamanda optimuma yakın sonuçlar vermektedir.

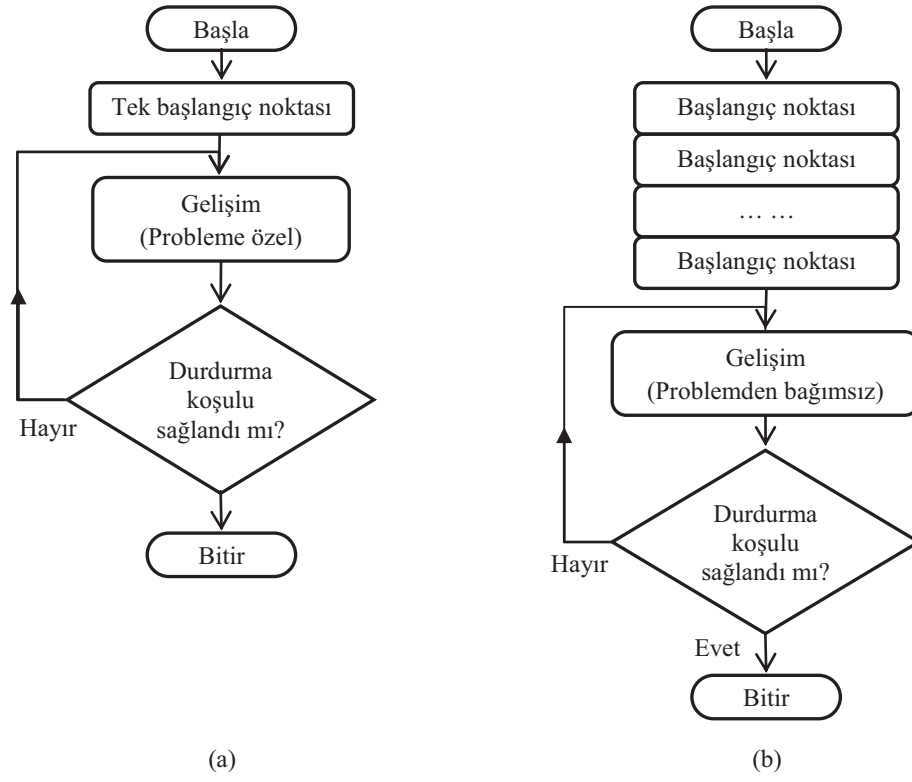
Daha önce de belirtildiği gibi hat dengeleme problemleri NP-Hard grubu problemler sınıfında yer almaktadır. Bu anlamda bu çalışmada ele alınan karışık modelli montaj hattı dengeleme probleminin çözümünde, modern sezgisel tekniklerden birisi olan genetik algoritmalarından yararlanılacaktır.

Genetik algoritmalar (GA), genetik ve doğal seleksiyonu temel alan evrimsel algoritmaların başını çeken, uyarlanabilir sezgisel arama algoritmalarındandır. GA, Charles Darwin tarafından öne sürülen doğal seleksiyon sürecini benimsemektedir. Diğer sezgisel yaklaşımlar gibi GA da rastlantısal aramaya dayanmaktadır [17].

Evrimsel programlama 1960'lı yıllarda I. Rechenberg'in "Evrimsel Stratejileri" adlı çalışmasıyla gündeme gelmiştir. Genetik algoritmaların bugünkü biçimi ise ilk olarak Michigan Üniversitesi'nden psikolog ve aynı zamanda bilgisayar bilimleri uzmanı John Holland tarafından ortaya atılmıştır [19].

Temeli 1960'lara dayanmakla birlikte, o günlerden bu yana başta mühendislik olmak üzere pek çok alanda uygulama şansı bulmuştur [17].

Geleneksel çözüm yaklaşımı ile genetik algoritmaya ait çözüm yaklaşımı karşılaştırmalı olarak Şekil 3.7'de verilmiştir. Şekilden de görüleceği gibi, GA diğer geleneksel arama yöntemlerinden farklı olarak tek bir başlangıç noktasından başlayarak değil, birçok başlangıç noktasından oluşan popülasyon içerisinde çözümü arar.



Şekil 3. 7 Geleneksel (a) ve genetik (b) yaklaşımların karşılaştırılması [1]

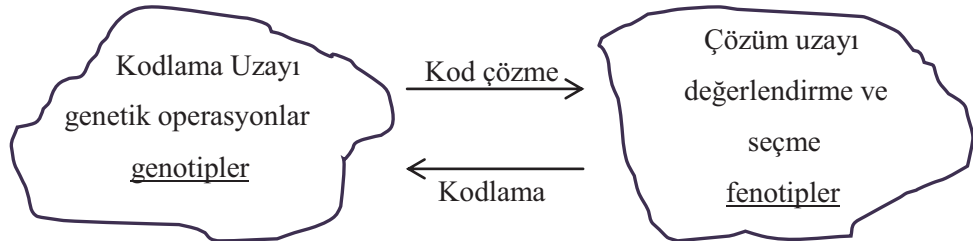
GA'ya ait akış diyagramı ise Şekil 3.8'de verilmektedir.





Şekil 3. 8 Genetik algoritma akış diyagramı [14]

GA direkt olarak çözüm uzayı üzerinde işlem yapmaz. Çözümler kromozomlar şeklinde kodlanır ve bunlar üzerinde işlem yapılır. Her bir kromozom, arama uzayındaki bir çözümü temsil eder. Daha sonra bu kromozomlar kullanıcı tarafından tanımlanan bilgilere göre çözümlenirler [14].



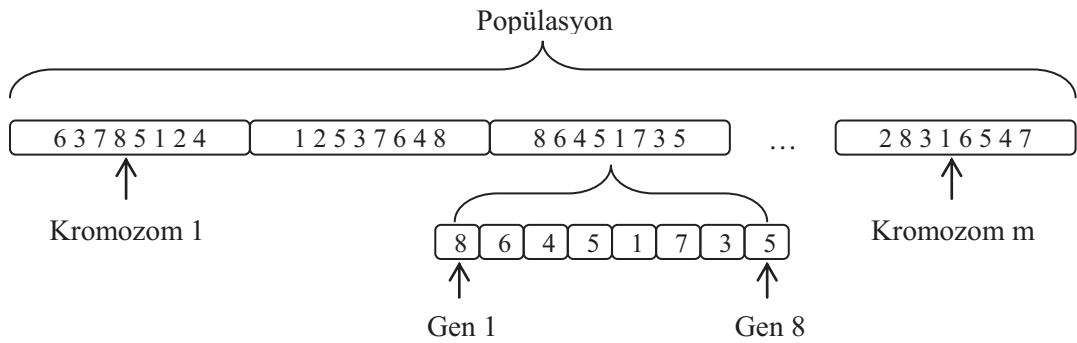
Şekil 3. 9 Çözüm uzayı ve arama uzayı [29]

Genetik algoritmalarda kullanılan bazı temel kavramlar aşağıda verilmektedir [18]:

*Gen*; kendi başına bir genetik bilgiyi taşıyan yapıdır ve bu ufak yapılar bir araya gelerek kromozomu oluşturur.

*Kromozom*; genlerin birleşmesiyle oluşan ve bir çözümü temsil eden dizidir. İlgili problemin olası çözümlerini içerirler. Kromozomlar bir araya gelerek popülasyonu oluşturur. Kromozomların hangi kısmının ne tür bilgiyi saklayacağı kullanıcının tanımına bağlıdır.

*Popülasyon*; kromozomların bir araya gelmesiyle oluşan olası çözüm yığınlarıdır. Bir popülasyondaki kromozom sayısı kullanıcı tarafından belirlenir ve sabittir. Genelde 30 ile 100 arasında belirlenmektedir. Popülasyon büyüklüğünün doğru saptanması çözümün hızı ve niteliği açısından önem arz etmektedir. Popülasyon büyüklüğünün fazla olması durumunda optimum çözüme yaklaşma ihtimali artabilir fakat bu durumda çözüme ulaşma süresi artacaktır. Kısaca popülasyon büyüklüğü problemin niteliğine göre kullanıcı tarafından iyi belirlenmelidir.



Şekil 3. 10 Gen, kromozom ve popülasyon yapısı [18]

*Başlangıç popülasyonu;* GA'nın ilk adımıdır. Başlangıç popülasyonunun oluşturulmasında uygunsuz çözümlerin popülasyona girmesini engellemek amacıyla probleme özel sezgisellere başvurulabilir.

*Uygunluk değeri;* kromozomların çözümde gösterdikleri başarının bir göstergesidir. Belirlenen kriterlere göre kromozomların sayısal olarak değerlendirilmesi söz konusudur.

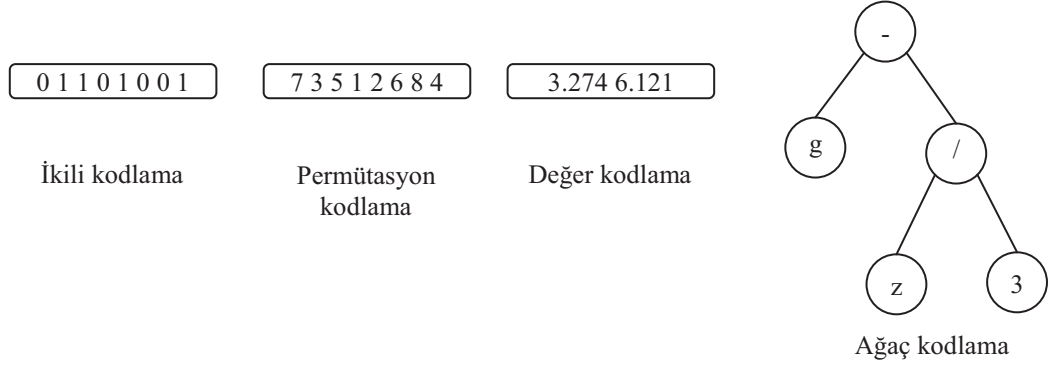
### **3.6.1 Kodlama**

Kromozomların kodlanma şekli probleme göre değişiklik gösterir ve yine kullanıcıya bağlıdır. Arama uzayını en iyi temsil edecek kodlama biçiminin seçilmesi, başarılı çözümlere ulaşım kolaylığı açısından önemlidir.

Sıkça kullanılan bazı kodlama biçimleri aşağıda verilmektedir [19]:

- **İkili Kodlama:** En eski kodlama biçimlerinden birisidir ve hala yaygın olarak kullanılmaktadır. Kromozomlar, 0 ve 1 değerli genlerden oluşur. Kromozomun tümü bir sayıyı ifade etmektedir.
- **Permütasyon Kodlama:** Her gen bir sayıya karşılık gelmektedir ve genelde sıralama problemlerinde kullanışlıdır. Çizelgeleme, gezgin satıcı problemleri gibi.
- **Değer Kodlama:** Kompleks sayıların yer aldığı problemlerde kullanılır. Burada her kromozom bazı değerler dizisidir. Bazı özel problemler (örneğin bir yapay sinir ağının ağırlık katsayılarının bulunması) için oldukça idealdir.
- **Ağaç Kodlama:** Genelde genetik programlamada programlar ve ifadeler oluşturmak için kullanılırlar. Kromozomlar, nesnelere ve nesnelere arası

işlemleri içeren bir ağaç yapısından oluşurlar. LISP, Prolog gibi programlama dillerinde sık kullanılırlar.



Şekil 3.11 GA'da kullanılan çeşitli kodlama biçimleri

### 3.6.2 GA Operatörleri

GA'nın operatörleri olarak kabul edilen *çaprazlama* ve *mutasyon*, çözümün performansını doğrudan etkilemektedir. Bu yüzden çaprazlama ve mutasyon stratejilerinin ve oranlarının doğru belirlenmesi çözümün niteliği açısından önemlidir.

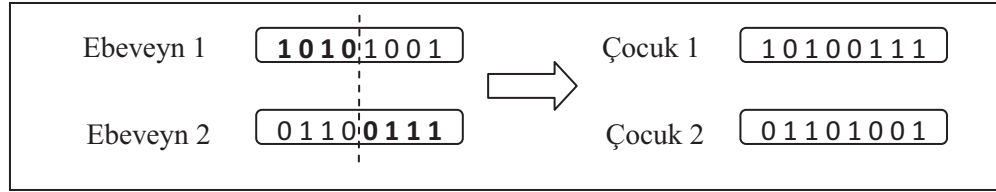
#### 3.6.2.1 Çaprazlama

Çaprazlama yaparken, önceden belirlenmiş bir seçim stratejisine göre seçilen ebeveyn kromozomlar, karşılıklı olarak belirli kısımlarını takas ederler. Bunun sonucunda ise yeni kromozomlar elde edilir. Seçilen kromozomların uygunluk değerlerinin iyi olması, yeni elde edilecek kromozomların da uygunluk değerlerinin iyi olma ihtimalini artıracaktır [19].

Çaprazlama sonrası oluşan kromozomların, ebeveyn kromozomlardan daha iyi olacağına teminatı yoktur [20]. Fakat daha iyi çözümlerin elde edilebilmesi için çaprazlama gerekli bir genetik operatördür.

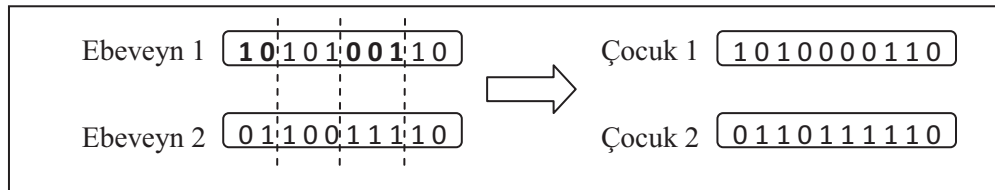
Genelde tavsiye edilen, problemin niteliğine göre değişmekle birlikte, çeşitliliği sağlamak amacıyla çaprazlama oranının 0.6-0.9 arasında seçilmesidir. Sıkça kullanılan çaprazlama yöntemlerinden bazıları aşağıda verilmiştir [18]:

- Tek noktalı çaprazlama: 1 ile L-1 arasında bir nokta seçilir ve ebeveyn kromozomlardaki bu çaprazlama noktasından sonraki bölümler karşılıklı yer değiştirilerek iki yeni kromozom oluşturulur (L, kromozomun uzunluğudur).



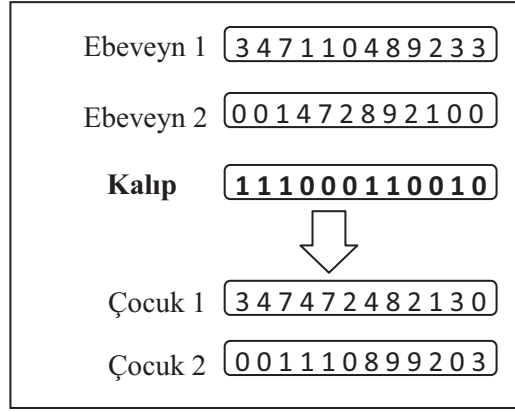
Şekil 3. 12 Tek noktalı çaprazlama örneği

- Çok noktalı çaprazlama: 1 ile L-1 arasında rastgele çoklu bölge seçilir. Belirlenen ebeveyn kromozomların bu çaprazlama noktaları arasında kalan kısımları karşılıklı yer değiştirilerek yeni bireyler elde edilir.



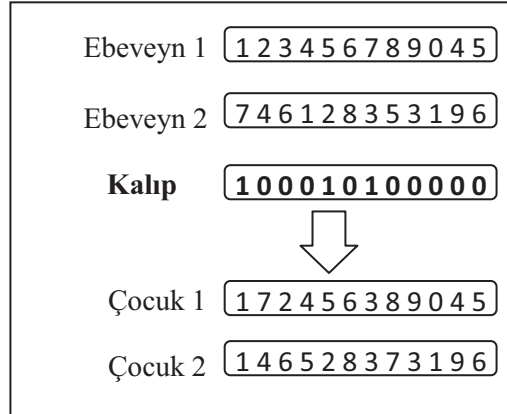
Şekil 3. 13 Çok noktalı çaprazlama örneği

- Pozisyona dayalı çaprazlama: Belirlenen bir kalıba göre sabit kalacak genler aynen kalır, değişecek olan genler diğer ebeveynden alınır. Aynı değerlerin tekrar etme riski vardır.



Şekil 3. 14 Pozisyona dayalı çaprazlama örneği

- Sıraya dayalı çaprazlama: Belirlenen bir kalıba göre sabit kalacak genler aynen kalır, değişecek genler ise diğer ebeveyndeki sıralamaya bağlı kalarak atanır.



Şekil 3. 15 Sıraya dayalı çaprazlama örneği

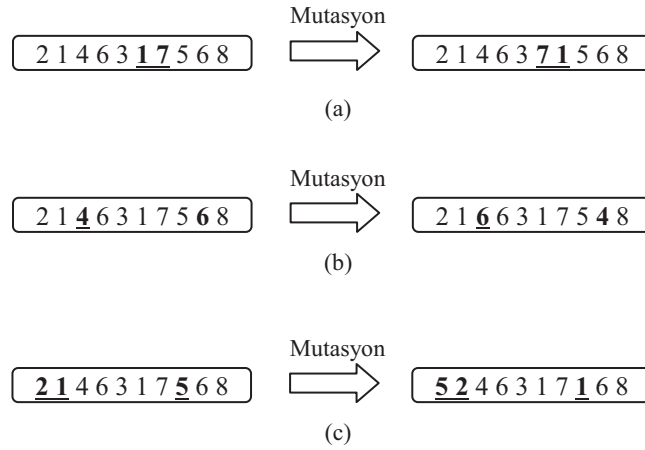
### 3.6.2.2 Mutasyon

Mutasyonda bir kromozomdaki belirli bir gen değiştirilir. Değiştirme oranının yüksek belirlenmesi, mevcut kromozomların bozulmasına; düşük belirlenmesi ise toplumdaki bazı özelliklerin kaybedilmesine yol açacaktır. Kabul görmüş yaklaşım,

eldeki iyi çözümlerin bozulmaması adına, mutasyon oranının % 0.1 ile % 15 aralığında seçilmesi yönündedir.

Mutasyon operatörünün uygulanmasına yönelik bazı yöntemler izleyen kısımda verilmektedir [18]:

- Komşu iki genin değiştirilmesi: Rastgele seçilen komşu iki gen yer değiştirir.
- Rastgele iki genin değiştirilmesi: Rastgele seçilen iki gen birbirleri ile değiştirilir.
- Rastgele üç genin değiştirilmesi: Rastgele seçilen üç gen yine rastgele değiştirilir.



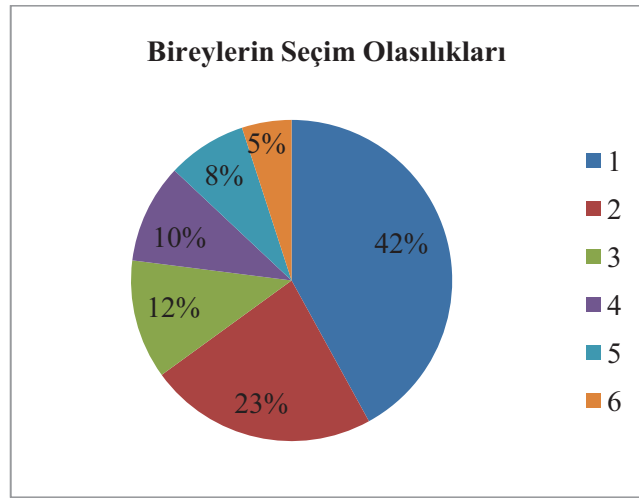
Şekil 3. 16 Komşu iki genin (a), rastgele iki genin (b) ve rastgele üç genin (c) değiştirilmesi

### 3.6.3 Seçim Mekanizmaları

Her iterasyon sonucunda popülasyon büyüklüğünün sabit tutulması amacıyla bir nesildeki kromozomların bir kısmı gelecek nesle aktarılırken, bir kısmı da elenmektedir. Hangi kromozomların gelecek nesle aktarılacağı belirlenirken bir takım seçme yöntemleri uygulanır [18].

Bu yöntemlerden en yaygın olanları şu şekildedir [19]:

- **Rulet Çemberi:** Bu seçimde, uygunluk değeri yüksek olan bireylerin seçilme olasılığı yüksektir, fakat garanti edilemez. Tüm bireylerin uygunluk değerleri bir tabloya yazılır ve toplanır. Sonra tüm bireylerin uygunluk değerleri ayrı ayrı toplama bölünerek seçilme olasılıkları belirlenir. Seçim yapılacağında bu olasılıklar birbirine eklenerek rastgele bir sayıya kadar ilerlenir. Bu sayıya ulaşıldığında ya da geçildiğinde son eklenen sayının ait olduğu birey seçilir.



Şekil 3. 17 Rulet Çemberi

- **Sıralı Seçim:** Uygunluklar arasında büyük farklılıklar olması durumunda, rulet seçiminin kullanılması uygun olmayabilir. Bu durumda sıralama seçimi kullanılabilir. Sıralama seçiminde bütün bireyler uygunluk değerlerine göre sıralanır ve bu sıralamaya göre her bireyin uygunluğu değerlendirilir. Uygunluk değeri en kötü birey 1 uygunlukta, ikinci en kötü 2 uygunlukta vb., en uygun olanı ise N uygunlukta olacaktır. Buradaki N değeri popülasyonun büyüklüğünü ifade eden değerdir. Bu şekilde bütün kromozomlara seçilme şansı doğacaktır. Fakat bu durumda da uygunluk değeri çok iyi olan kromozomlarla diğerleri arasında fazla fark olmayacağı için çözüme yaklaşma yavaş olabilir.



- Sabit Durum Seçimi: Her nesilde bir sonraki nesle aktarılmak üzere büyük uygunlukta ve iyi olan birkaç kromozom seçilir. Daha sonra en kötü uygunlukta olan kromozomlar popülasyondan atılır ve yeni nesil onların yerine getirilir.

#### **3.6.4 Elitizm**

Elit kromozomların bir sonraki nesle aktarımının sağlanması, iyi kromozomların kaybedilmesini önlemekte ve genetik operatörler yardımıyla elit kromozomlardan daha iyi çocuklar elde edilmesinin yolunu açmaktadır [20].

Bazı durumlarda, örneğin çizelgeleme problemlerinde, çaprazlama ve mutasyon operatörlerinin kullanımı sonrası kromozomlarda oluşan gen yapılarındaki bozukluğun düzeltilmesi amacıyla tamir operatörü kullanılır. Tamir operatörü, genetik operatörlerin kullanımı sonucu yeni kromozomda birden fazla oluşan veya kaybolan genlerin bulunup düzeltilmesinden sorumludur [18].

#### **3.6.5 Yeni Neslin Oluşturulması**

Uygunluk değeri hesaplanan bireylerden hangilerinin gelecek nesle aktarılıp hangilerinin yok edileceğine seçim mekanizmalarının da kullanılmasıyla karar verilir. Bu şekilde, çözüme ulaşmada yardımcı olacağı düşünülen iyi bireyler gelecek nesle aktarılır ve yok olmasının önüne geçilir.

Sonuç olarak, GA hat dengeleme problemlerinde 1990'lı yıllardan beri etkin olarak uygulanmakta ve özellikle büyük boyutlu problemlerin çözümünde etkin sonuçlar elde edilebilmektedir. Diğer sezgisel teknikler gibi GA da hibrid bir model olarak kurgulanıp doğru problemlere uygulandığında başarı şansı daha da artmaktadır. Bu çalışma kapsamında ele alınacak GA, COMSOAL yöntemi ile

desteklenerek hibrid bir GA elde edilecek ve karışık modelli montaj hattı dengeleme problemlerinin çözümünde kullanılacaktır.

#### 4. KARIŞIK MODELLİ DÜZ HATLARIN DENGELENMESİ

Daha önce de belirtildiği gibi son zamanlarda tüketici eğilimleri isteğe göre üretilmiş ürünlerden yana olmakta ve bu da firmaları esnek ve siparişe göre üretim konusunda zorlamaktadır. Bu konuda artan müşteri talepleri firmaları aynı hat üzerinde birden fazla modeli üretmeye itmekle birlikte firmaların ancak bu yolla potansiyel pazarını koruyabileceklerini düşünmelerini sağlamıştır.

Bu bölümde karışık modelli montaj hatları, yerleşim şekillerinin getirdiği özel durumlar da göz önüne alınarak incelenecektir. Bu bağlamda karışık modelli düz hatlar (KMHD) varsayımları, amaçları, kısıtları ve gerçek hayatta karşılaşılan zorluklar büyük oranda Simaria (2006)'dan yararlanılarak değerlendirilecektir.

##### 4.1 Problemin Varsayımları ve Kısıtları

Bu problem grubunda montaj hattı, aynı hat üzerinde bir ürüne ait birbirine benzer modellerin ( $m = 1, \dots, M$ ) herhangi bir sırada veya karışık bir şekilde daha önceden belirlenmiş bir planlama periyodunda ( $P$ ) üretilebileceği şekilde düzenlenmektedir.  $m$  modeli için tahmin edilen talep miktarı  $D_m$  iken hattın çevrim zamanı (4.1) eşitliği kullanılarak hesaplanabilir [14].

$$C = P / \sum_{m=1}^M D_m \quad (4.1)$$

$m$  modelinin üretimdeki genel oranı ise (4.2) eşitliği ile hesaplanabilir.

$$q_m = D_m / \sum_{p=1}^M D_p \quad (4.2)$$

Her model kendi öncelik ilişkileri kısıtlarına sahiptir. Fakat farklı modellere ait bu öncelik ilişkileri kısıtları birleştirilerek, tüm modellere ait,  $N$  adet göreve ( $i = 1, \dots, N$ ) sahip genel bir öncelik ilişkileri diyagramı oluşturulabilir. Burada  $m$  modeline ait  $i$  görevinin süresi  $t_{im}$  olacaktır. Eğer  $m$  modeli için  $i$  görevinin gerçekleştirilmesine gerek duyulmuyorsa  $t_{im} = 0$  olacaktır.

Modern montaj hatlarında işçilerin kalifiye olması ve kendisine verilen herhangi bir işi tamamlayabilmesi istenmektedir. Ancak bu şekilde, bir görevin farklı istasyonlarda gerçekleştirilmesi mümkün olacaktır. Fakat bu durum gerçek hayatta tam bir başarıyla uygulanamamaktadır. Bu yüzden bir görevin tüm modeller için aynı istasyonda gerçekleştirilmesi kısıtı doğmaktadır. Bu kısıt (4.3) eşitliği dikkate alınarak (4.4) eşitliği ile matematiksel olarak ifade edilebilir.

$$x_{ik} = \begin{cases} 1, & \text{eğer } i \text{ işi } k \text{ istasyonuna atanmışsa} \\ 0, & \text{aksi durumda} \end{cases} \quad (i = 1, \dots, N; \quad k = 1, \dots, LL) \quad (4.3)$$

$$\sum_{k=1}^{LL} x_{ik} = 1 \quad (i = 1, \dots, N) \quad (4.4)$$

Bu sayede birçok modele ait ortak herhangi bir görev sadece bir istasyona atanabilecektir. Burada  $LL$ , hattın uzunluğunu temsil etmektedir.

Öncelik ilişkileri kısıtlarının sağlanması için ise (4.5) denklemi ile verilen kısıta ihtiyaç duyulacaktır. Her bir  $i$  görevinin ardıllarının kümesi ( $Suc_i$ ) öncelik ilişkileri diyagramından çıkartılabilir.  $Suc_i$  kümesinin hiçbir elemanı  $i$  görevinden önceki bir istasyona atanmamalıdır.

$$\sum_{k=1}^{LL} kx_{ik} - \sum_{k=1}^{LL} kx_k \leq 0 \quad (i = 1, \dots, N; \quad j \in Suc_i) \quad (4.5)$$

Buxey (1974)'in de belirttiği gibi, istasyona atanmış işlerin toplam yükünün istasyonun maksimum kapasitesini aşmaması gerekir. Montaj hattı dengelemede kullanılan birçok teknikte en uzun işlemin süresi çevrim zamanını belirlemektedir. Yani üretim hızını belirleyen kısıt en uzun işlem süresidir. Bu kısıt paralel istasyonlar kullanılarak gevşetilebilir ve böylece hem daha esnek hem de daha kısa çevrim zamanına ihtiyaç duyan hatlar tasarlanabilir.

Paralel istasyon kullanmanın avantajları olduğu gibi dezavantajları da vardır. Montaj hatlarının avantajlarından birisi olan nitelikli işgücüne ihtiyaç duymama özelliği, paralel istasyonlar kullanılırsa ortadan kalkmaktadır. Bu anlamda, paralel şekilde oluşturulacak istasyonların sınırının belirlenmesi gerekmektedir. Schofield (1979) ve Sarker ve Shantikumar (1983) paralel istasyon sayısını kontrol altına alabilmek için maksimum istasyon sayısı limiti belirlerken; Buxey (1974) her istasyona atanabilecek işler sayısına bir limit getirmiştir.

Simaria ve Vilarinho (2001) ise *Minimum Replication Time-MRT* olarak bir değer belirlemişlerdir. Ancak bir istasyonun paralelleştirilebilmesi için o istasyona atanan herhangi bir modele ait bir görevin süresinin MRT değerini geçmesi gerekmektedir. Birleştirilecek istasyonların sayısını hesaplamak için en büyük işlem zamanına ait görev seçilerek (4.6) eşitliğinden yararlanılmaktadır. Burada  $R_k$ ,  $k$  istasyonunda çalışan işçi sayısı olarak düşünülebilir.

$$R_k = \left\lceil \frac{\max(t_{im} x_{ik})}{MRT} \right\rceil \quad (m = 1, \dots, M; \quad i = 1, \dots, N; \quad k = 1, \dots, LL) \quad (4.6)$$

Paralel istasyonların kullanımıyla, ilgili istasyonda aynı işi gerçekleştirecek işçi sayılarının da artması gerekmektedir. Montaj hattında çalışan işçi sayılarının hesaplanması için (4.7) eşitliği kullanılabilir.

$$S = \sum_{k=1}^{LL} R_k \quad (4.7)$$

Bir istasyonun kapasitesi, kendisine atanan işlerin süresine bağlıdır.  $m$  görevine ait  $h$  görevi  $k$  istasyonuna atanacaksa, bu istasyonun iş yükünü temsil eden  $W_k$  değeri (4.8) eşitliği ile hesaplanabilir.

$$W_{km} = \sum_{i=1}^N t_{im} x_{ik} + t_{hm} \quad (k = 1, \dots, LL; \quad m = 1, \dots, M) \quad (4.8)$$

İstasyonun kapasitesi ise kendisine atanan ve  $MRT$  değerinden büyük işleri gerçekleştirmek için yeterli olmalıdır. Bu anlamda iki durum karşımıza çıkmaktadır. Birinci durum,  $k$  istasyonunun en az bir model için  $MRT$  değerinden büyük bir işlem süreli göreve sahip olması veya en az bir model için  $h$  görevinin süresinin  $MRT$  değerinden büyük olmasıdır. Bu durumda (4.9) denkleminde verilen kısıt devreye girecektir.

$$W_{km} \leq R_k C \quad (k = 1, \dots, LL; \quad m = 1, \dots, M) \quad (4.9)$$

Eğer  $k$  istasyonundaki tüm görevler ve  $h$  görevi  $MRT$  değerinden küçük sürelerle sahipse kısıt (4.10) denklemi ile ifade edilebilir.

$$W_{km} \leq C \quad (k = 1, \dots, LL; \quad m = 1, \dots, M) \quad (4.10)$$

Bir istasyonun atıl zamanı, kapasitesi ile yükü arasındaki farktan oluşmaktadır.  $k$  istasyonunun  $m$  modeline ait atıl zamanı ( $s_{km}$ ), (4.11) eşitliği yardımıyla hesaplanabilir.

$$s_{km} = R_k C - \sum_{i=1}^N t_{im} x_{ik} \quad (k = 1, \dots, LL; \quad m = 1, \dots, M) \quad (4.11)$$

Pozitif ve negatif bölgeleme kısıtları da uygulamada sıkça karşılaşılan problemler olmasına karşın üzerinde fazla çalışma yapılmamış konulardır.  $ZP$ , aynı istasyonda yapılması gereken görevlerin kümesini ifade etmek üzere, pozitif bölgeleme kısıtı (4.12) eşitliğindeki gibi formüleleştirilebilir.

$$\sum_{k=1}^{LL} k x_{ik} - \sum_{k=1}^{LL} k x_{jk} = 0 \quad ((i, j) \in ZP) \quad (4.12)$$

$ZN$ , farklı istasyonlarda gerçekleştirilmesi gereken görevler kümesini belirtmek üzere, negatif bölgeleme kısıtı (4.13) denklemindeki gibi formülleştirilebilir.

$$\sum_{k=1}^{LL} k x_{ik} - \sum_{k=1}^{LL} k x_{jk} \neq 0 \quad ((i, j) \in ZN) \quad (4.13)$$

## 4.2 Amaç Fonksiyonu

Tip-1 montaj hattı dengelemenin temel amacı montaj hattında kullanılan toplam istasyon sayısını minimize etmektir. Bu aynı zamanda atıl sürenin de minimizasyonu anlamına gelmektedir. Çünkü görevlerin süreleri toplamı sabit kaldığı sürece, istasyon sayısının azalması toplam atıl süreyi de azaltacaktır. Bu amaçla geliştirilen amaç fonksiyonlarından bazıları izleyen kısımda verilmiştir [14].

$$\text{Min } WIT = \sum_{k=1}^{LL} \sum_{m=1}^M q_m \left( R_k C - \sum_{i=1}^N t_{im} x_{ik} \right) \quad (4.14)$$

Burada  $WIT$  ağırlıklı atıl zamanı temsil etmektedir. Fakat bu değer problemden probleme değişeceği için daha sonra geliştirilen hat etkinliği ( $WE$ ) kavramı (4.15) eşitliğinde ifade edilmiştir.

$$\text{Max } WE = \sum_{m=1}^M q_m \left( \frac{\sum_{i=1}^N t_{im}}{S C} \right) \quad (4.15)$$

$WE$ , 0 ile 1 arasında bir değer almaktadır.  $WE$ 'nin 1 olması hattın tam kapasite ile hiçbir atıl zaman olmadan kullanılması anlamına gelmektedir. Fakat bu durum pratikte söz konusu değildir.

Hem tip-1 hem de tip-2 problemlerde atıl zamanın minimizasyonu yanında ortak amaçlardan birisi, istasyonlar arası ve istasyon içi iş yükünün düzgün dağıtılmasıdır. Bu amaçla Vilarinho ve Simaria (2006) tarafından geliştirilen formülasyonlar (4.16) ve (4.17) eşitliklerinde verilmektedir.

$$\text{Min } B_b = \frac{LL}{LL-1} \sum_{k=1}^{LL} \left( \frac{\sum_{m=1}^M q_m S_{km}}{WIT} - \frac{1}{LL} \right)^2 \quad (4.16)$$

$$\text{Min } B_w = \frac{M}{LL(M-1)} \sum_{k=1}^{LL} \sum_{m=1}^M \left( \frac{q_m S_{km}}{S_k} - \frac{1}{M} \right)^2 \quad (4.17)$$

Burada  $S_k$  ifadesi  $k$  istasyonunun ağırlıklı atıl zamanını,  $B_b$  ifadesi istasyonlar arası iş yükü dağılım dengesini,  $B_w$  ifadesi ise istasyonlar içi iş yükü dağılım dengesini temsil etmektedir.

### 4.3 Matematiksel Model

Vilarinho ve Simaria (2002) tarafından paralel istasyonlu ve bölgeleme kısıtlı kMHD-I problemi için yapılan çalışmada kullanılan 0-1 tamsayı programlama modeli aşağıda verilmiştir:

- planlama zamanı sabittir ve  $P$  olarak alınmıştır,
- benzer  $M$  adet model eş zamanlı montaj edilmektedir,
- planlama periyodu boyunca  $m$  modeli için tahmin edilen talep  $D_m$  kadardır ve gerek duyulan çevrim süresi  $C = P / \sum_{m=1}^M D_m$ 'dir,



- $m$  modelinin oransal ihtiyacı  $q_m = D_m / \sum_{p=1}^M D_p$  'dir,
- her model kendi öncelik ilişkileri diyagramına sahiptir fakat tüm modeller için ortak görevler alt kümesi vardır. Bu yüzden, tüm modellerin öncelik ilişkileri diyagramları birleştirilir ve  $N$  adet görev vardır,
- $N$  adet görev ( $i = 1, \dots, N$ ),  $S$  adet iş istasyonunda ( $k = 1, \dots, S$ ) gerçekleştirilir,
- $m$  modeline ait  $i$  görevinin gerçekleştirilme zamanı  $t_{im}$  'dir ve görevlerin süresi modellere bağlı olarak değişebilmektedir ( $t_{im} = 0$  olması,  $m$  modelinin üretilmesinde  $i$  işine gerek duyulmadığını ifade etmektedir),
- her iş yalnızca bir iş istasyonuna atanabilmektedir ve birçok modele ait ortak işler aynı iş istasyonunda gerçekleştirilmelidir,
- $i$  görevinden önce gerçekleştirilemeyecek işler kümesi  $F_i$  ( $i$  görevinin ardılları) birleştirilmiş öncelik ilişkileri diyagramından elde edilen öncelik kısıtlarıyla verilmektedir,
- $MAXP$  değeri en fazla kaç iş istasyonunun birleştirilebileceğini (paralel istasyonlar) gösteren değerdir ve önceden tanımlanmaktadır. Bir modele ait herhangi bir görevin çevrim süresini aşması durumunda paralel istasyon oluşumuna izin verilmektedir,
- bölgeleme kısıtları vardır ve  $ZP$  pozitif bölgeleme kısıtlarını, yani aynı istasyonda gerçekleştirilmesi gereken işleri,  $ZN$  ise negatif bölgeleme kısıtlarını, yani farklı istasyonlarda gerçekleştirilmesi gereken işleri belirten kümelerdir.
- $x_{ik} = \begin{cases} 1, & \text{eğer } i \text{ işi } k \text{ istasyonuna atanmışsa} \\ 0, & \text{aksi durumda} \end{cases}$

- $r_k = \begin{cases} 1, & \text{eğer } k \text{ istasyonu birleştirilmişse} \\ 0, & \text{aksi durumda} \end{cases}$
- $S_{km} = m$  modeli için  $k$  istasyonunun atıl zamanı

Amaç fonksiyonu:

$$\min Z = \sum_{k=1}^S k x_{Nk} + \frac{S'}{S'-1} \sum_{m=1}^M q_m \sum_{k=1}^{S'} \left( \frac{S_{km}}{\sum_{l=1}^{S'} S_{lm}} \frac{1}{S'} \right)^2 + \frac{M}{S'(M-1)} \sum_{k=1}^{S'} \sum_{m=1}^M \left( \frac{q_m S_{km}}{S_k} - \frac{1}{M} \right)^2 \quad (4.18)$$

Kısıtlar:

$$\sum_{k=1}^S x_{ik} = 1 \quad i = 1, \dots, N \quad (4.19)$$

$$\sum_{k=1}^S x_{ak} - \sum_{k=1}^S x_{bk} \leq 0 \quad a \in N, b \in F_a \quad (4.20)$$

$$\sum_{k=1}^S x_{ak} - \sum_{k=1}^S x_{bk} = 0 \quad (a, b) \in ZP \quad (4.21)$$

$$x_{ak} + x_{bk} \leq 1 \quad (a, b) \in ZN, k = 1, \dots, S \quad (4.22)$$

$$\sum_{i=1}^N t_{im} x_{ik} + s_{km} = C[1 + r_k(MAXP - 1)] \quad k = 1, \dots, S, m = 1, \dots, M \quad (4.23a)$$

$$r_k \leq \sum_{i: \exists t_{im} > \alpha C; m=1, \dots, M} x_{ik} \quad k = 1, \dots, S, 0 < \alpha < \%100 \quad (4.23b)$$

$$M r_k \geq \sum_{i: \exists t_{im} > \alpha C; m=1, \dots, M} x_{ik} \quad k = 1, \dots, S, 0 < \alpha < \%100 \quad (4.23c)$$

$$S_{km} \geq 0 \quad k = 1, \dots, S, m = 1, \dots, M \quad (4.24a)$$

$$x_{ik} \in [0,1] \quad k = 1, \dots, S, i = 1, \dots, N \quad (4.24b)$$

$$r_k \in [0,1] \quad k = 1, \dots, S \quad (4.24c)$$

Verilen bir çevrim zamanı için istasyon sayısının minimizasyonunu amaçlayan paralel istasyon ve bölgeleme kısıtlı karışık modelli montaj hattı dengeleme problemi matematiksel model olarak yukarıdaki gibi ifade edilebilir.

Modelde kullanılan formülasyonların açıklamaları ise izleyen kısımda verilmektedir [6].

Amaç fonksiyonundaki (4.18), ilk terim son görevin atandığı iş istasyonu indeksini minimize etmektedir, böylece toplam iş istasyonu sayısı minimum olacaktır. İkinci terim, iş istasyonları arasındaki iş yükü dağılımı dengesizliğini minimize etmektedir. Üçüncü terim ise istasyon içindeki iş yükü dağılımını minimize etmektedir. Amaç fonksiyonunun ikinci ve üçüncü kısmı  $[0,1]$  aralığında değer almaktadır. Gerçek hayat uygulamalarında  $S' > 2$  olduğu için fonksiyonun ilk kısmı baskındır. Bu yüzden model, fonksiyonun ikinci kısmı aktif olmadan birinci kısmı minimize etmektedir.

(4.19) numaralı kısıt her görevin yalnızca bir istasyona atanmasını sağlamaktadır.

(4.20) numaralı kısıt  $i$  görevinin ardıllarının  $i$  görevinden önceki bir istasyona atanmasını engellemektedir.

(4.21) numaralı kısıt pozitif bölgeleme kısıtlarını ifade etmektedir.

(4.22) numaralı kısıt negatif bölgeleme kısıtlarını ifade etmektedir.

(4.23) numaralı kısıt grubu istasyonların kapasitelerinin üzerinde yüklenmesini ve iş istasyonlarının belirlenen sınırın üzerinde birleştirilmesini önlemektedir. İş istasyonlarının birleştirilmesi durumu yalnızca, herhangi bir görevin çevrim süresini aşması durumunda ortaya çıkmaktadır.

(4.24) numaralı kısıt grubu ise karar değişkenlerinin alabileceği değerleri ifade etmektedir.

Bugünlerde küçük ve orta büyüklükteki işletme (KOBİ)'lerin büyük çoğunluğunda karşılaşılan durum, ya ürünlerin üretimine yönelik belirgin ve dengelenmiş bir montaj hattının bulunmaması ya da her bir ürün için ayrı bir montaj

hattı bulunması şeklindedir. Burada tabii ki işletmenin kaynakları, her bir ürün için ayrı bir montaj hattı kurulması konusunda belirleyici kısıtlardan birisi olacaktır. Farklı modellerin aynı hat üzerinde montajı ise olması gereken fakat nadir rastlanan bir durumdur. Oysa, dengelenmiş bir hat üzerinde farklı modellerin üretimi hem termin zamanını kısaltacak, hem işletme kaynaklarının israfını önleyecek hem de müşteri istekleri doğrultusunda esnekliğin sağlanmasına yardımcı olacaktır.

## 5. UYGULAMA

Bu bölüme kadar, montaj hatlarının ne olduğundan, kaç tür montaj hattı olduğundan, problemin karmaşıklığından, montaj hattı dengelemenin öneminden, montaj hattı dengelemede kullanılan bazı yaklaşımlardan, karışık modelli hatların neden gerektiğinden, konuyla ilgili yapılmış çalışmalardan bahsedilmiştir. Bu bölümde ise yeni bir hibrid algoritma verilmektedir.

Genetik algoritmaların, permütasyon kodlamalı (gezgin satıcı problemi, iş çizelgeleme vb.) problem türlerinde gösterdiği başarı açıktır. Fakat başlangıç popülasyonunun oluşturulması ve sonrasında, uygunsuz çözümler oluşmaktadır. Bu uygunsuz çözümler iterasyonların sayısını veya popülasyon büyüklüğünü artırmayı gerekli kılmakta ve dolayısıyla çözüme yaklaşımı zorlaştırmaktadır. Bu bağlamda, bu çalışma kapsamında geliştirilen hibrid algoritma, başlangıç popülasyonunun oluşturulması sırasında sadece uygun çözümlerin oluşmasına izin vermekte ve bu sayede iterasyon sayısını düşürmektedir.

### 5.1 Önerilen Hibrid Genetik Algoritma

Bu çalışma kapsamında ele alınan karışık modelli düz montaj hatları, gerçek hayat koşullarını daha çok yansıtması amacıyla paralel istasyon oluşumu ve bölgeleme kısıtları da göz önüne alınarak modellenmiştir. Paralel istasyonlar yalnızca ağırlıklı görev sürelerinden herhangi birisinin istasyon kapasitesini aştığı durumlarda oluşmaktadır.

Bölgeleme kısıtları da pozitif ve negatif bölgeleme kısıtları olarak iki şekilde değerlendirilmektedir. Pozitif bölgeleme kısıtı aynı istasyonda yapılması gereken

görevleri ifade ederken negatif bölgeleme kısıtı ise farklı istasyonlarda yapılması gereken görevleri ifade etmektedir.

Geliştirilen yöntem, Arcus (1965) tarafından geliştirilen COMSOAL metodu ile popülasyon tabanlı algoritmalarından olan genetik algoritmanın birleştirilmesinden oluşturulmuştur ve hibrid GA olarak adlandırılmıştır. COMSOAL (Arcus, 1965) ile güçlendirilen GA, sayısal bir örnek üzerinde gösterilecektir. Daha sonra modelin ve algoritmanın başarısını ölçmek amacıyla 12 test problemi üzerinde deneysel çalışma yapılarak elde edilen sonuçlar ile önceki çalışmalarda elde edilen sonuçlar kıyaslanacaktır.

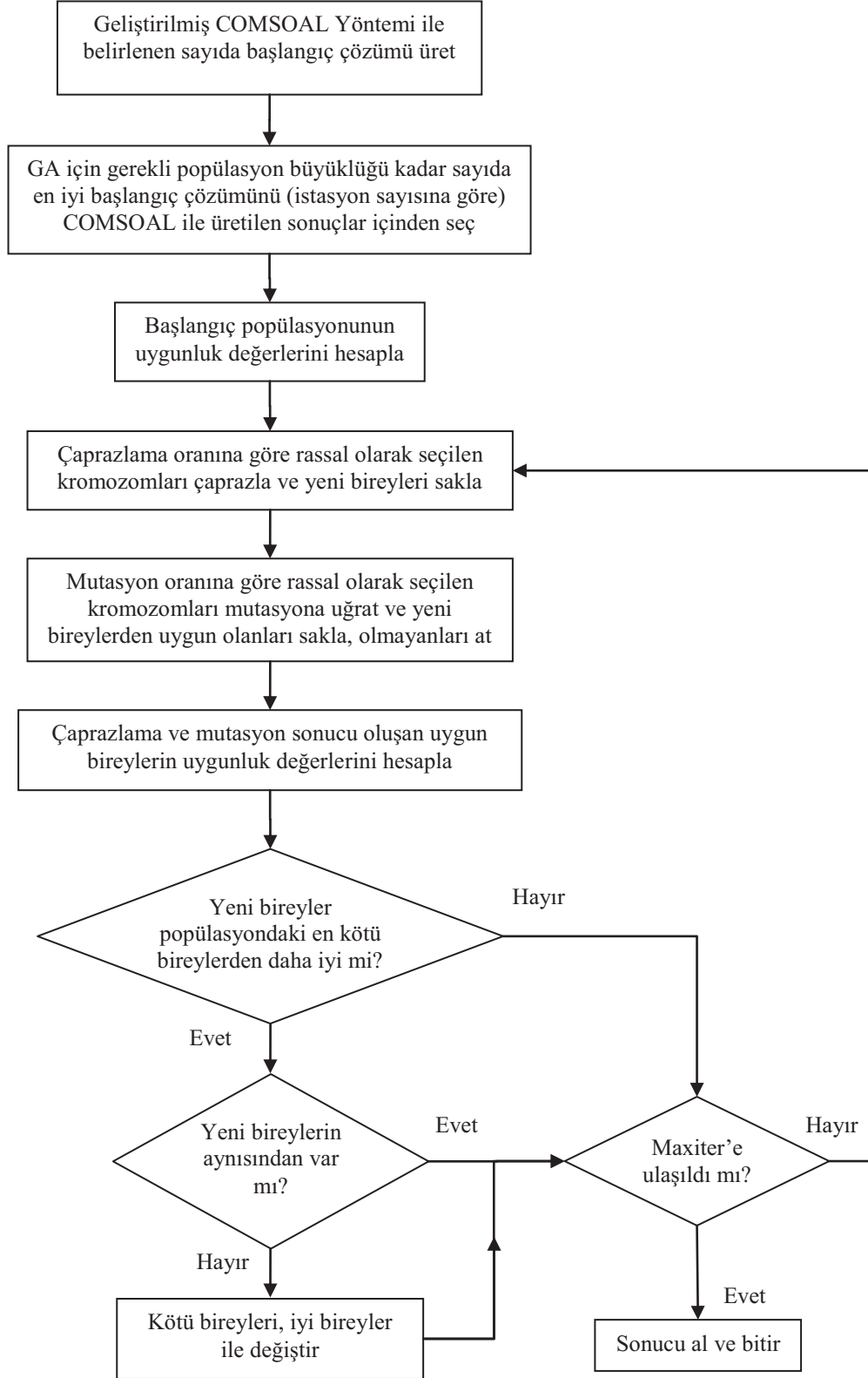
Daha önce de belirtildiği gibi sıralama problemlerinin GA ile çözümünde uygunsuz çözümlerin oluşması söz konusudur. Uygunsuz çözümler başlangıç popülasyonunun oluşumunda, çaprazlama ve mutasyon sırasında, görevlerin öncelik ilişki kısıtlarını ihlal edecek şekilde sıralanmasından kaynaklanmaktadır.

Burada algoritmanın performansını en çok etkileyen faktör, başlangıç popülasyonunun oluşumu sırasında ortaya çıkan uygunsuz çözümlerdir. Çünkü bu aşamada uygunsuz çözüm oluşma ihtimali, çaprazlama ve mutasyon sırasında uygunsuz çözüm oluşma ihtimalinden çok daha yüksektir. Bu nedenle, bu çalışmada başlangıç popülasyonu COMSOAL metodu kullanılarak elde edilen sonuçlar arasından en iyileri seçilerek oluşturulmaktadır. Böylece hem uygunsuz çözümlerin oluşumunun önüne geçilmiş olacak, hem de optimum çözüme daha kısa zamanda ve daha az sapmayla ulaşma ihtimali artacaktır.

Pek çok çalışmada kullanılan tamir operatörü bu çalışmada kullanılmamaktadır. Bunun yerine çaprazlama ve mutasyon sırasında uygunsuz çözümlerin oluşmasını engelleyecek önlemlerle, fazladan bir operatör kullanımına gerek duyulmadan, zamandan ve gereksiz işlemlerden tasarruf edilmektedir.

Algoritmada kullanılan COMSOAL yöntemi geleneksel COMSOAL'dan farklı olarak iteratif bir yapıyla kodlanmıştır. Belirlenen iterasyon sayısına ulaşıncaya dek, çözüm üretmeye devam etmektedir. İstenen sayıya ulaşıldığında ise istenen

popülasyon büyüklüğü kadar birey, uygunluk değerlerine göre, üretilen sonuçlar arasından en iyileri alınarak seçilmektedir (Şekil 5.1).



Şekil 5. 1 Geliştirilen algoritmaya ait akış diyagramı

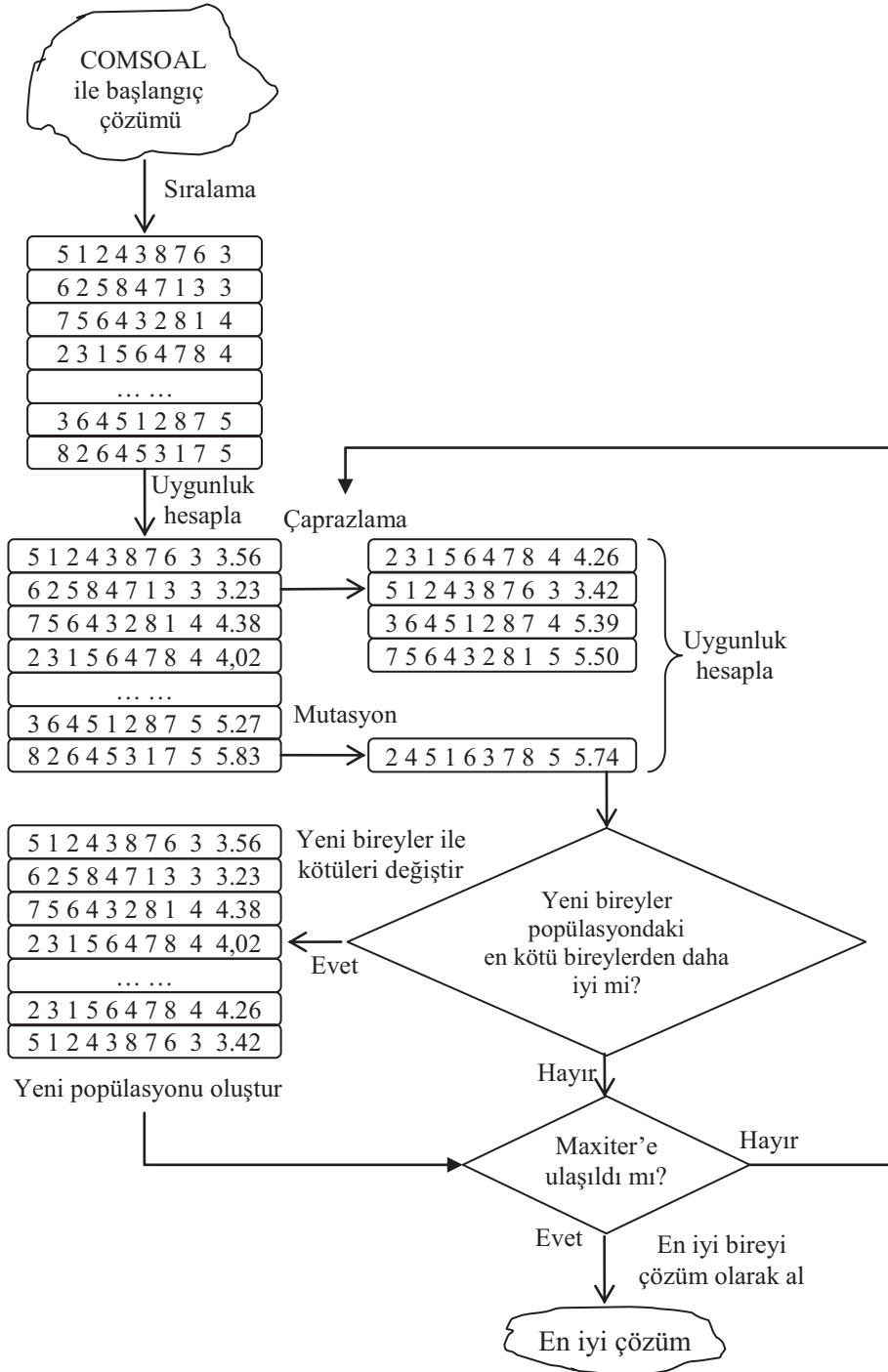


Daha sonra bu bireyler GA'ya dahil edilmekte ve uygunlukları hesaplanmaktadır. Çaprazlama oranına göre rastgele seçilen bireyler, iki noktalı çaprazlama yöntemine göre çaprazlanmaktadır. Çaprazlama sonrası yeni oluşan bireylerin uygunluk değerleri ebeveynlerin uygunluk değerleri ile kıyaslanır. Eğer yeni bireyin uygunluk değeri daha iyi ise ebeveynin yerine yeni birey atanır, değilse herhangi bir işlem yapılmaz.

Çaprazlamadan sonra mutasyon oranına göre rastgele seçilen bireyler mutasyona uğratılır. Eğer daha iyi bir çözüm elde edilmiş ise ebeveyn ile yeni birey değiştirilir, değilse herhangi bir işlem yapılmaz.

Sonraki nesilde kullanılacak bireyler, popülasyonda kalan bireyler olacaktır. Popülasyon büyüklüğü sabit tutulmaktadır ve durdurma kriteri sağlanana dek döngü devam ettirilir. Algoritmada kullanılan durdurma kriteri iterasyon sayısının belirli bir değeri aşp aşmadığıdır. Durdurma kriteri sağlanınca mevcut çözümler arasından uygunluk değeri en iyi olan çözüm seçilir ve problemin çözümü olarak kabul edilir.

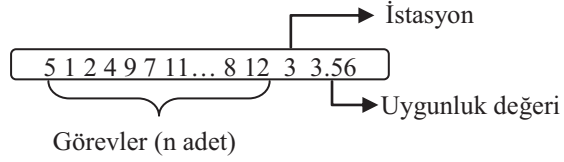
Algoritmanın daha açık ve görsel anlatımı için oluşturulan simülasyon Şekil 5.2'de verilmiştir.



Şekil 5. 2 Geliştirilen algoritmanın simülasyonu

Algoritmada kullanılan kodlama biçimi permütasyon kodlamadır. COMSOAL ile elde edilen çözümler popülasyon tabanlı kodlama şekli ile kodlandıktan sonra GA için hazır olmaktadır.

İşlemlerin kolaylığı için algoritmada kullanılan kromozomlarda, sıralı görevlerin hemen sonunda bu kromozomun oluşturduğu çözümde gerek duyduğu istasyon sayısı ve uygunluk değerleri yer alır. Bu bilgiler her kromozomda farklıdır ve kendine özgüdür.



Şekil 5.3 Kodlamada kullanılan kromozom yapısı

## 5.2 Örnek Uygulama

Geliştirilen algoritma, öncelik ilişkileri (Şekil 5.3) Gunther'den (Scholl 1993) alınan, görev süreleri (Tablo 5.1) rassal olarak oluşturulmuş iki modellen bir örnek üzerinde uygulanmıştır. Üretim miktarları dikkate alınarak  $q_1 = 0,40$  ve  $q_2 = 0,60$  olarak hesaplanmıştır. Bu durumda görevlerin ağırlıklı süreleri ( $t_i$ ) de  $q_1$  ve  $q_2$  değerleri dikkate alınarak hesaplanmış ve Tablo 5.1'de verilmiştir. Görevlerin ağırlıklı sürelerinin hesaplanmasında (5.1) eşitliği kullanılmaktadır.

$$t_{ort} = \sum_{m=1}^M q_m \times t_m \quad (5.1)$$

Burada  $m$ , modeli ifade etmekte,  $q_m$  modelin üretimdeki oranı,  $t_m$  ise ilgili görev için  $m$  modelinin süresidir.

Problemin gerçek hayat koşullarını daha çok yansıtması adına paralel istasyon oluşumu ve bölgeleme kısıtları da modele dahil edilmiştir.

Tablo 5.1’den de görüldüğü gibi 30 numaralı görevin süresi (11.3 br zaman), çevrim süresini (10 br zaman) aşmaktadır. Bu durumda paralel istasyon oluşumu söz konusu olacaktır.

**Tablo 5.1** Gerçekleştirilmesi gereken görevlerin modellere göre süreleri

<b>İş</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>	<b>15</b>
<b>t<sub>1</sub></b>	3.4	1.0	2.1	3.4	7.8	6.2	8.5	6.3	9.7	3.0	7.6	8.2	0.0	6.1	2.2
<b>t<sub>2</sub></b>	3.2	1.2	0.0	6.4	5.5	0.0	7.9	2.3	3.7	3.0	9.1	3.5	5.7	5.6	1.8
<b>t<sub>ort</sub></b>	3.3	1.1	0.8	5.2	6.4	2.5	8.1	3.9	6.1	3.0	8.5	5.4	3.4	5.8	2.0
<b>İş</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>	<b>29</b>	<b>30</b>
<b>t<sub>1</sub></b>	1.5	2.9	0.4	7.6	4.9	8.1	1.6	2.1	9.9	4.5	0.6	7.7	2.5	6.6	7.0
<b>t<sub>2</sub></b>	2.9	2.3	0.8	9.3	8.2	4.5	9.9	3.6	1.6	4.6	1.9	0.0	8.7	9.1	14.2
<b>t<sub>ort</sub></b>	2.3	2.5	0.6	8.6	6.9	5.9	6.6	3.0	4.9	4.6	1.4	3.1	6.2	8.1	11.3
<b>İş</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>										
<b>t<sub>1</sub></b>	1.2	5.0	6.8	0.0	2.0										
<b>t<sub>2</sub></b>	10.0	4.8	2.3	8.6	2.9										
<b>t<sub>ort</sub></b>	6.5	4.9	4.1	5.2	2.5										

Bölgeleme kısıtı olarak ise 12 ve 13 numaralı görevlerin farklı istasyonlara atanması gerekliliği, negatif bölgeleme kısıtı şeklinde dikkate alınacaktır.

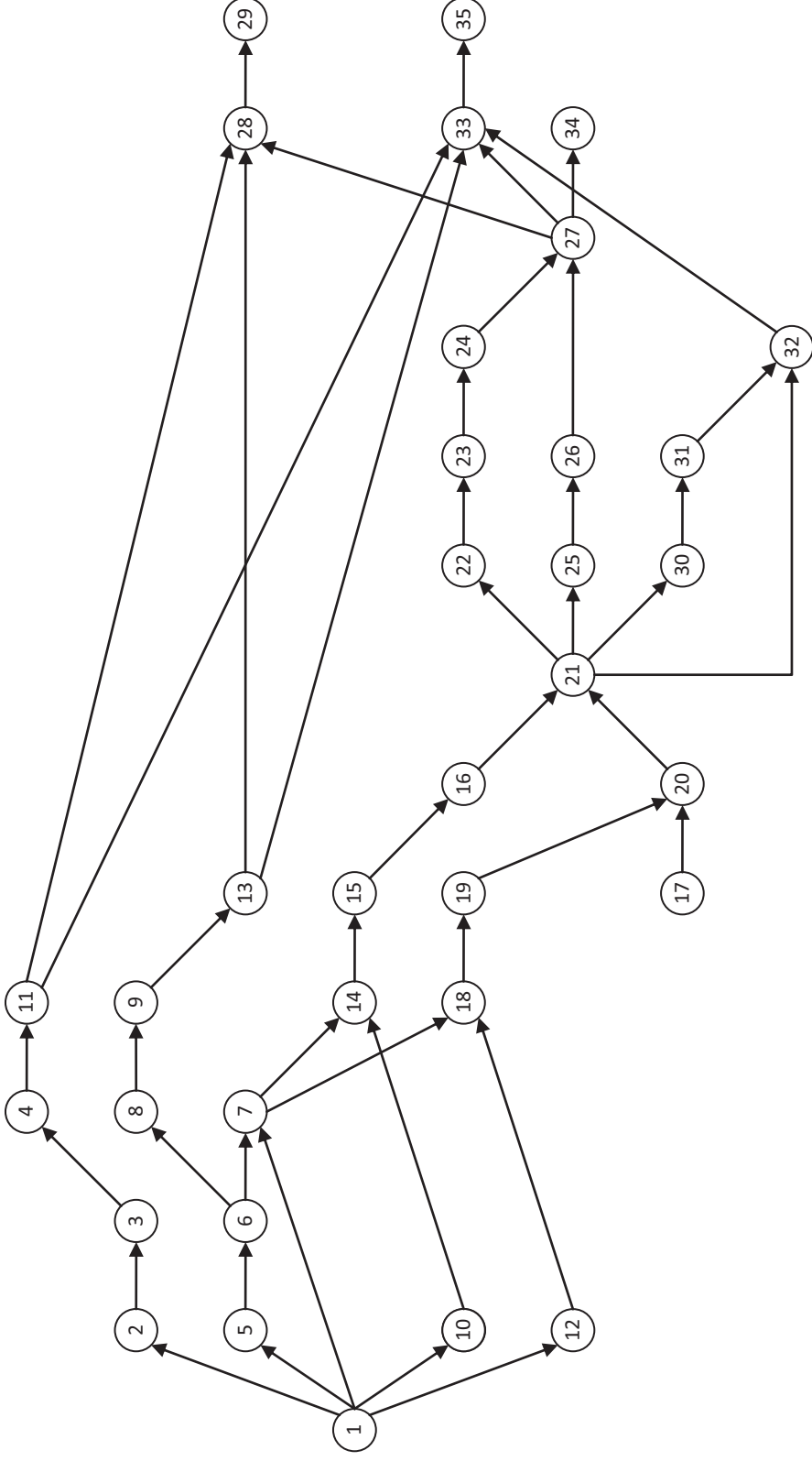
Şekil 5.4’teki öncelik ilişkileri diyagramından da görüldüğü gibi dengelenmesi gereken görev sayısı 35’tir ( $n = 35$ ). Görevlerin dengelenmesi sırasında amacımız, istasyon sayısının ve dolayısıyla atıl zamanın minimize edilmesidir. Bunun yanında ikincil amaç ise istasyonlar arası iş yükü dağılımının düzgün olmasıdır. Bu amaçla algoritmada kullanılan amaç fonksiyonu (5.2) eşitliğinde verilmektedir [21].

$$Min Z = \sum_{k=1}^S \frac{(C - W_k)^2}{S} \quad (5.2)$$

Burada  $S$ , toplam istasyon sayısını;  $C$ , çevrim zamanını;  $W_k$ ,  $k$  istasyonunun toplam iş yükünü temsil etmektedir. İstasyonlarda oluşan atıl zamanın karesinin

alınması ve bunların toplanarak istasyon sayısına bölünmesi yoluyla, istasyonlar arası düzgün iş yükü dağılımının sağlanması amaçlanmaktadır.

Tek noktalı çaprazlama yapılmakta ve kullanılan çaprazlama oranı % 10, mutasyon oranı ise % 5'tir. Bu değerler GA'da genelde kullanılan değerlere yakın olarak seçilmişlerdir. Popülasyon büyüklüğü 20, iterasyon sayısı ise 50'dir.



Şekil 5.4 Uygulama örneğine ait Gunther'dan alınan öncelik ilişkileri diyagramı (Scholl 1993)

Çaprazlama ve mutasyon sonucu elde edilen bireyler uygun çözüme sahip ve uygunluk değerleri popülasyondaki bazı bireylerden daha iyi ise popülasyondaki kötü bireylerin yerine, yeni oluşan iyi bireyler geçmektedir. Böylelikle optimum çözüme daha çok yaklaşma imkanı sağlanmaktadır.

Algoritma on kez çalıştırılmış ve elde edilen en iyi çözüm izleyen tabloda verilmiştir. Bu durumda 19 adet iş istasyonuna gerek duyulduğu görülmektedir. En fazla oluşan atıl zaman 1 numaralı istasyondadır ve 3.1 br zaman kullanılamamaktadır. Tam kapasiteye en yakın durumdaki iş istasyonu ise 12 numaralı iş istasyonudur. Bu istasyonun iş yükü 9.6 br zamandır.

13 numaralı istasyonda, 30 numaralı göreve ait sürenin çevrim süresinden büyük olması sebebiyle paralel istasyon oluşumu söz konusudur. Dolayısıyla bu istasyonun kapasitesi diğer istasyonlara göre iki katına çıkmaktadır.

Negatif bölgeleme kısıtı olan 12 ve 13 numaralı görevler, farklı istasyonlara atanarak bölgeleme kısıtını sağlamaktadırlar.

**Tablo 5. 2** Elde edilen en iyi çözüm

İst	Görev	Görev Süresi	İst. Süresi	Atıl Zaman	R	İst	Görev	Görev Süresi	İst. Süresi	Atıl Zaman	R
1	1	3.3	6.9	3.1	1	11	16	2.3	8.2	1.8	1
	17	2.5					21	5.9			
	2	1.1				12	22	6.6	9.6	0.4	1
5	6.4	23	3								
2	6	2.5	8.9	1.1	1	13	30	11.3	17.8	2.2	2
	12	5.4					31	6.5			
3	10	3	8.4	1.6	1	14	24	4.9	9.5	0.5	1
	7	8.1					25	4.6			
4	14	5.8	8.4	1.6	1	15	32	4.9	9.4	0.6	1
	15	2					26	1.4			
	18	0.6					27	3.1			
6	19	8.6	8.6	1.4	1	16	34	5.2	9.3	0.7	1
7	20	6.9	7.7	2.3	1		33	4.1			
	3	0.8				17	28	6.2	8.7	1.3	1
8	4	5.2	9.1	0.9	1		35	2.5			
	8	3.9				29	8.1				
9	11	8.5	8.5	1.5	1	<b>Toplam</b>			<b>164.7</b>	<b>25.3</b>	<b>19</b>
10	9	6.1	9.5	0.5	1						
	13	3,4									

Elde edilen çözümün uygunluk değeri (fitness value) 2.36'dır. Dengeleme sonucu oluşan toplam atıl zaman (5.3) eşitliği ile denge kaybı ise (5.4) eşitliği yardımıyla hesaplanabilir.

$$SC - \sum_{i=1}^n t_i = 19 \times 10 - 164.7 = 25.3 \quad (5.3)$$

$$Denge kaybı = \frac{S(C) - \sum t}{S(C)} \times 100 = \frac{19(10) - 164.7}{19(10)} \times 100 = \% 13 \quad (5.4)$$

### 5.3 Kabul Edilen Test Problemleri ve Sonuçlarının Kıyaslanması

Algoritmanın performansını ölçmek amacıyla 12 test problemi [14] üzerinde denemeler yapılmıştır. Her problemde model 10 defa çalıştırılmış ve sonuçlar alınmıştır. Daha sonra bu sonuçlar içerisinde her problem için minimum istasyon sayısı (S') ile minimum, ortalama ve maksimum uygunluk değerleri hesaplanarak tabloda verilmiştir. Elde edilen çözümlerin kıyaslanması amacıyla problem için çözüm olabilecek istasyon sayısının alt sınır değerini ifade eden  $LB_{pmix}$  [22] ve problemlerin yalnız GA [10] ile elde edilmiş çözümlerine göre gerekli istasyon sayıları verilmiştir.

Ayrıca N değeri problemin görev sayısını, M değeri model sayısını, C değeri ise çevrim süresini ifade etmektedir.



**Tablo 5. 3** Test problemlerinin sonuçlarının kıyaslanması

#	Problem	N	M	C	$LB_{pmix}$	Yalın GA	Akpınar & Bayhan	Hibrid GA Min S'	Sapma (%D)	Uygunluk Değeri		
										Min	Ort	Max
1	Vilarinho and Simaria	25	2	10	14	16	16	<b>14</b>	0.00	1.91	2.48	4.59
2	Vilarinho and Simaria	25	3	10	14	15	<b>14</b>	<b>14</b>	0.00	2.17	2.62	4.30
3	Heskiaoff	28	2	10	19	21	<b>20</b>	<b>20</b>	5.26	1.55	2.10	3.55
4	Heskiaoff	28	3	10	18	20	20	<b>18</b>	0.00	1.10	2.77	3.31
5	Sawyer	30	2	10	15	<b>16</b>	<b>16</b>	<b>16</b>	6.67	2.06	2.90	4.48
6	Sawyer	30	3	10	17	19	19	<b>18</b>	5.88	2.20	4.54	5.18
7	Lutzl	32	2	10	15	19	19	<b>16</b>	6.67	0.95	3.12	5.04
8	Lutzl	32	3	10	17	19	19	<b>18</b>	5.88	3.18	4.5	4.93
9	Kilbridge and Wester	45	2	10	23	25	25	<b>24</b>	4.35	1.61	2.67	3.60
10	Kilbridge and Wester	45	3	10	24	28	27	<b>26</b>	8.33	3.00	3.65	3.98
11	Tonge	70	2	10	41	45	<b>44</b>	45	9.76	4.12	5.31	5.95
12	Tonge	70	3	10	39	45	<b>44</b>	<b>44</b>	12.82	3.18	4.24	5.03

Burada ele alınan test problemleri için yalın GA ile elde edilen değerler Akpınar ve Bayhan (2011)'den alınmıştır. Daha sonra bu değerler yeni geliştirilen hibrid GA ile karşılaştırılmıştır.

Tablo 5.3'ten de görüldüğü üzere geliştirilen hibrid GA, 5 ve 11 numaralı problem hariç diğer tüm problemlerde yalın GA'dan daha iyi sonuçlar vermiştir. 5 ve 11 numaralı problemlerde ise yalın GA ve hibrid GA ile elde edilen çözümler aynıdır. 1, 2 ve 4 numaralı problemlerde hibrid GA ile elde edilen çözümde, mümkün olan en küçük istasyon sayısına erişilmiştir.

1, 4, 6, 7, 8, 9 ve 10 numaralı problemlerde önerilen hibrid GA ile elde edilen çözümler, Akpınar ve Bayhan (2011) tarafından önerilen hibrid GA ile elde edilen çözümlerden daha iyidir. 2, 3, 5 ve 12 numaralı problemlerde durum eşit iken yalnızca 11 numaralı problemde durum Akpınar ve Bayhan (2011) lehinedir.

Sapma değeri (% D) ise Vilarinho ve Simaria tarafından geliştirilen, problemin çözümü için teorik minimum istasyon sayısı alt sınırını ifade eden  $LB_{pmix}$  ile bulunan minimum istasyon arasındaki sapmayı ifade etmektedir.

## 6. SONUÇ VE TARTIŞMA

Bu çalışmada karışık modelli düz montaj hatlarının dengelenmesine yönelik hibrid GA geliştirilmiştir. GA'da kullanılmak üzere uygun başlangıç çözümleri üretmesi için algoritmaya entegre edilen COMSOAL yöntemi, Arcus (1965)'ten biraz farklı olarak iteratif bir yapıyla kullanıcının belirlediği iterasyon sayısına kadar çalışmakta, daha sonra üretilen sonuçlar içerisinde GA'nın başlangıç popülasyonu için gerekli olan sayıda çözümü GA'ya sunmaktadır. Ardından, GA operatörleri kullanılarak yine belirlenen iterasyon sayısına ulaşılan dek algoritma çalıştırılmakta ve popülasyon içinden içlerinden en iyi uygunluk değerine (Leu et al. 1994) sahip çözüm, problemin çözümü olarak alınmaktadır.

Model kurulurken, problemin gerçek hayat koşullarını yansıtması amacıyla çevrim süresini aşan görevlerin gerçekleştirilmesinde paralel istasyon oluşumuna gidilmiştir. Ayrıca pozitif ve negatif bölgeleme kısıtları da dikkate alınmaktadır. Bu kısıtlar zaten karmaşık olan problemin karmaşıklığını daha da artırmaktadır.

Geliştirilen hibrid GA ile elde edilen sonuçların yalnız GA ile elde edilen sonuçlara kıyasla görece üstünlüğe sahip olduğu Tablo 5.3'te görülmektedir.

Ayrıca geliştirilen hibrid GA ile yapılan çözümlerde popülasyon büyüklüğü ve iterasyon sayısı gibi parametrelerin daha küçük değerler seçilebilmesi mümkündür. Çünkü başlangıç popülasyonu COMSOAL ile üretildiği için popülasyondaki tüm bireyler uygun ve iyi çözümlerdir. Bu uygun ve iyi çözümleri başlangıç popülasyonu olarak alan algoritmanın optimum çözüme yaklaşması daha da kolaylaşmakta ve algoritmanın çalışma süresi de kısalmaktadır.

Bu alıřma ile hem GA hakkında hem de montaj hattı dengeleme problemleri konusunda detaylı bilgi edinme fırsatı bulunmuřtur. Bu bilgi birikiminin ileriki alıřmalara temel oluřturması ve yeni alıřmalara ıřık tutması ümit edilmektedir.

## 7. KAYNAKÇA

- [1] Gen, Mitsuo, Cheng, Runwei ve Lin, Lin. "Network Models and Optimization, Multiobjective Genetic Algorithm Approach", Springer, London, (2008).
- [2] Bukchin, J., Dar-el, Ezey M. ve Rubinovitz, J., "Mixed model assembly line balancing in a make-to-order environment", *Computers&Industrial Engineering*, **41**, (2002) 405.
- [3] Baybars, İ., "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem", *Management Science*, **32/8**, (1986) 909.
- [4] Thomopoulos, Nick T., "Line Balancing-Sequencing for Mixed-Model Assembly", *Management Science*, **14/2**, (1967) 59.
- [5] Boysen, N., Fliedner, M. ve Scholl, A., "A classification of assembly line balancing problems", *European Journal of Operational Research*, **183**, (2007) 674.
- [6] Vilarinho, P. M. ve Simaria, A. S., "A two-stage heuristic method for balancing mixed-model assembly lines with parallel workstations", *International Journal of Production Research*, **40/6**, (2002) 1405.
- [7] Yagmahan, B., "Mixed-model assembly line balancing using a multi-objective ant colony optimization approach", *Expert Systems with Applications*, (2011) doi:10.1016/j.eswa.2011.04.026.
- [8] Chong, K. E., Omar, M. K. ve Bakar, N. A., "Solving Assembly Line Balancing Problem using Genetic Algorithm with Heuristics-Treated Initial Population" Proceedings of the World Congress on Engineering: WCE, London, (2008).
- [9] Gokcen, H. ve Erel, E., "A goal programming approach to mixed-model assembly line balancing problem", *International Journal of Production Economics*, **48**, (1997) 175.
- [10] Akpınar, S. ve Bayhan, G.M., "A hybrid genetic algorithm for mixed model assembly line balancing problem with parallel workstations and zoning constraints", *Engineering Applications of Artificial Intelligence*, **24**, (2011) 449.
- [11] Battini, D., Faccio, M., Ferrari, E., Persona A. and Sgarbossa F., "Design configuration for a mixed-model assembly system in case of low product demand", *International Journal of Advanced Manufacturing Technology*, **34**, (2007) 188.
- [12] Scholl, A., "Balancing and sequencing of assembly lines", *Darmstadt: Physica-Verlag*, (1995).

- [13] Bukchin, Y. ve Rabinowitch, I., "A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs", *European Journal of Operational Research*, **174**, (2006) 492.
- [14] Simaria, Ana Sofia de A., Assembly line balancing - new perspectives and procedures, PhD Thesis, Universidade de Aveiro, Portugal, (2006).
- [15] Sule, D. R., Production Planning and Industrial Scheduling: Examples, Case Studies and Applications, Second Edition, Taylor & Francis Group, Boca Raton, (2008).
- [16] Wild, R., Production and Operations Management, Fourth Edition, Cassell Educational Limited, Londra, (1989).
- [17] Cura, T., Modern Sezgisel Teknikler ve Uygulamaları, Papatya Yayıncılık Eğitim, (2008).
- [18] Elmas, Ç., Yapay Zeka Uygulamaları, Seçkin Yayıncılık, Ankara (2007).
- [19] Nabiyev, V. V., Yapay Zeka, Seçkin Yayıncılık, (2010).
- [20] Rardin, Ronald L., Optimization in operations research, Prentice Hall, (1998).
- [21] Leu, Y. Yuh., Matheson, L. A. ve Rees, L. P., "Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria", *Decision Sciences*, **30**, (1994) 581. doi: 10.1111/j.1540-5915.1994.tb00822.x.
- [22] Vilarinho, P. M. ve Simaria, A. S. "ANTBAL: an ant colony optimization algorithm for balancing mixed-model", *International Journal of Production Research*, **44/2**, (2006) 291.
- [23] Yaman, R., Üretim Planlama: Kontrol ve Bütünleştirme, Nobel Yayınevi, Ankara (2011).
- [24] Kim, Y. K., Kim, J. Y. ve Kim, Y., "An endosymbiotic evolutionary algorithm for the integration of balancing and sequencing in mixed-model U-lines", *European Journal of Operational Research*, **168**, (2006) 838.
- [25] Yaman, R. ve Küçükkoç, İ., "A New COMSOAL Based Heuristic Approach to Mixed Model Assembly Line Balancing with Parallel Stations and Zoning Constraints", Yöneylem Araştırması ve Endüstri Mühendisliği 31. Ulusal Kongresi, Sakarya, 5-7 Temmuz, (2011).
- [26] Kim, Y. K., Kim, J. Y. ve Kim, Y., "A Coevolutionary Algorithm for Balancing and Sequencing in Mixed Model Assembly Lines", *Applied Intelligence*, **13**, (2000) 247.
- [27] Askin, R. G. ve Zhou, M. "A parallel station heuristic for the mixed-model production line balancing problem", *International Journal of Production Research*, **35/11**, (1997) 3095.

- [28] Simaria, A. S. ve Vilarinho, P. M., "2-ANTBAL: An ant colony optimisation algorithm for balancing two-sided assembly lines", *Computers & Industrial Engineering*, **56**, (2009) 489.
- [29] Rekiek, B. ve Delchambre, A., *Assembly Line Design: The Balancing of Mixed-Model Hybrid Assembly Lines with Genetic Algorithms* (Springer Series in Advanced Manufacturing), Springer, (2005).

## 8. EKLER

### EK-A. Karışık Modelli Montaj Hatları İçin Test Problemleri ve Süreleri

Tablo A. 1 Vilarinho&Simaria (2002) ve Heskiaoff test problemlerine ait süreler

Görev	Vilarinho&Simaria (2002)		Vilarinho&Simaria (2002)			Heskiaoff		Heskiaoff		
	t <sub>A</sub>	t <sub>B</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>C</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>C</sub>
1	0	2	4.1	4.1	4.1	1.0	1.0	3.4	3.4	0.0
2	7.7	7.7	2.7	2.7	2.7	3.7	3.7	10.0	10.0	10.0
3	7.3	7.3	4.6	4.6	4.6	7.2	7.2	8.0	8.0	8.0
4	15	15	4.1	4.1	4.1	12.6	12.6	12.7	11.8	12.7
5	8.8	8.8	2.0	2.0	2.0	4.2	4.2	9.3	9.3	9.3
6	6.2	0	0.0	2.0	2.0	3.4	3.4	3.3	3.3	3.3
7	3.6	0	11.3	11.3	11.3	3.4	3.4	4.7	4.7	4.7
8	0	2	7.8	7.8	7.8	10.7	10.7	8.6	8.6	8.6
9	6.6	6.6	0.0	10.0	10.0	9.9	0.0	8.2	0.0	8.2
10	2.5	2.5	3.5	3.5	3.3	7.6	7.6	8.4	8.4	8.4
11	5.5	5.5	3.9	4.2	3.9	9.8	9.8	0.0	0.0	9.0
12	7.1	7.1	1.0	1.0	1.0	2.4	2.4	8.9	8.3	8.9
13	5.9	5.9	2.5	2.3	2.5	9.8	8.7	4.6	4.6	0.0
14	1.3	0	5.1	5.1	5.1	10.0	10.0	0.0	0.0	5.0
15	5.5	5.5	3.5	3.5	3.5	0.0	9.0	9.2	9.3	9.2
16	1.9	2	3.5	3.5	3.4	0.0	9.0	1.9	2.1	2.0
17	3.7	0	6.8	6.8	6.8	2.7	2.7	4.8	4.8	4.8
18	9.4	9.4	8.5	8.5	9.6	3.6	3.7	10.2	11.5	10.2
19	1.3	1.3	9.9	9.9	9.9	7.7	7.7	4.4	4.6	4.4
20	0	9	7.2	7.2	7.2	12.4	12.4	0.0	3.0	0.0
21	2	2	4.8	4.8	4.8	6.2	6.2	1.4	1.4	1.4
22	4.7	4.7	3.8	3.8	3.9	6.0	6.6	4.1	4.1	0.0
23	9.6	8.2	2.9	2.8	2.6	5.5	5.5	4.1	4.1	4.1
24	4.1	3.7	3.5	3.5	3.5	1.9	1.9	0.0	9.0	9.0
25	12.5	0	7.8	7.8	7.8	4.4	4.4	9.5	9.5	10.0
26						12.1	12.1	9.2	9.2	9.2
27						9.2	9.2	9.9	9.9	9.9
28						9.5	9.9	3.7	3.7	3.1

**Tablo A. 2** Sawyer ve Lutz1 test problemlerine ait süreler

Görev	Sawyer		Sawyer			Lutz1		Lutz1		
	t <sub>A</sub>	t <sub>B</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>C</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>C</sub>
1	9.5	9.5	2.1	2.1	2.1	2.4	2.4	0.0	4.0	0.0
2	1.3	1.3	3.2	3.2	3.6	5.3	5.3	6.6	5.7	6.6
3	4.8	4.8	4.3	0.0	4.3	2.0	2.0	2.0	2.0	2.0
4	3.3	3.3	0.0	5.0	5.0	4.0	4.3	4.3	4.0	4.3
5	1.5	1.7	3.7	3.7	3.7	4.0	4.0	4.0	4.0	4.0
6	4.5	4.1	13.4	13.4	13.4	4.4	4.4	4.4	4.4	4.4
7	3.6	3.6	2.1	0.0	2.1	1.3	1.5	1.3	1.3	1.5
8	0.0	2.0	7.8	0.0	7.8	1.4	1.4	1.4	1.4	1.4
9	12.3	12.3	4.2	4.2	4.2	4.9	4.5	4.9	4.9	4.5
10	0.0	8.0	9.9	9.9	9.9	6.6	6.6	4.5	4.4	4.5
11	2.5	2.5	5.8	5.8	6.0	6.9	0.0	3.8	3.8	3.8
12	4.3	4.3	3.0	2.9	3.0	2.4	2.4	2.4	2.4	2.4
13	6.5	0.0	0.0	2.0	2.0	5.3	5.3	5.3	5.3	5.3
14	1.7	1.7	2.4	2.4	2.4	12.2	12.2	12.2	12.2	12.2
15	7.0	7.0	7.0	6.5	7.0	2.7	2.7	2.7	2.7	2.7
16	1.4	1.4	3.8	0.0	3.8	5.5	5.5	5.5	5.5	5.5
17	7.8	7.8	0.0	2.0	2.0	6.6	6.6	6.6	6.6	6.6
18	2.9	2.9	5.7	5.7	5.7	6.8	6.5	6.8	6.8	6.5
19	1.6	1.6	1.4	1.4	1.4	9.5	9.5	9.5	9.5	9.5
20	7.0	7.0	13.7	13.7	13.7	0.0	14.9	14.9	0.0	14.9
21	8.7	8.7	6.4	6.4	6.9	2.2	2.2	2.8	2.8	2.8
22	3.9	4.1	5.0	5.0	5.3	4.8	4.8	6.4	6.4	6.4
23	6.4	6.4	12.8	12.8	12.8	5.1	5.8	8.6	8.6	8.6
24	2.8	2.7	0.0	2.0	2.0	0.0	10.0	9.7	8.2	9.7
25	8.5	8.5	8.2	8.2	0.0	1.1	1.0	5.8	5.5	5.8
26	6.7	6.7	9.1	9.1	9.1	2.4	2.4	2.4	2.4	2.4
27	1.9	1.9	9.7	9.7	9.7	1.7	1.7	0.0	2.0	2.0
28	9.9	9.9	7.2	7.2	7.2	12.3	13.5	3.9	3.9	4.5
29	4.6	0.0	10.5	10.5	10.5	2.5	2.5	7.1	7.1	7.1
30	4.0	4.2	2.4	2.4	2.4	0.0	0.0	2.0	0.0	0.0
31						5.1	5.1	5.1	5.1	5.1
32						4.1	4.0	4.0	4.1	4.0



**Tablo A. 3** Kilbridge&Wester ve Tonge test problemlerine ait süreler

Görev	Kilbridge&Wester		Kilbridge&Wester			Tonge		Tonge		
	t <sub>A</sub>	t <sub>B</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>C</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>A</sub>	t <sub>B</sub>	t <sub>C</sub>
1	1.0	1.0	7.5	8.0	7.5	5.8	5.8	3.2	3.5	3.2
2	4.4	5.1	6.0	0.0	6.0	5.7	5.7	11.7	11.7	11.7
3	14.3	0.0	5.6	5.6	5.6	0.0	10.0	6.0	5.6	6.0
4	2.2	2.2	3.8	0.0	3.8	5.0	5.1	1.9	0.0	1.9
5	4.8	4.8	3.1	2.9	3.1	5.1	5.1	0.0	15.0	0.0
6	5.1	5.8	4.2	4.2	4.2	0.0	8.0	7.4	7.4	7.4
7	0.0	10.0	10.6	10.6	10.6	6.8	6.8	7.9	8.4	7.9
8	5.1	5.1	0.0	8.0	8.0	4.5	4.5	5.2	5.4	5.2
9	9.4	9.4	3.6	3.6	3.6	5.1	5.1	2.3	2.3	2.3
10	5.0	5.0	2.4	2.4	2.4	9.6	9.6	5.8	4.9	5.8
11	3.5	3.5	9.3	9.3	0.0	1.5	1.5	2.6	2.6	0.0
12	0.0	4.0	1.1	1.2	1.1	3.9	3.9	4.3	4.6	4.3
13	7.0	0.0	2.7	0.0	0.0	9.6	9.6	2.2	2.2	2.2
14	2.7	0.0	5.6	5.6	5.0	2.7	2.7	3.5	3.5	3.7
15	5.3	5.3	9.3	9.3	9.3	1.5	1.5	8.7	0.0	8.7
16	0.0	3.0	0.0	8.0	7.2	4.2	0.0	8.4	8.4	8.4
17	2.2	2.2	10.4	10.4	10.4	5.3	5.3	2.1	2.1	2.1
18	0.0	3.0	9.0	9.0	9.0	3.7	3.7	8.4	8.4	8.4
19	8.3	8.3	12.0	12.0	12.1	8.1	8.1	8.1	8.1	8.1
20	2.6	2.6	6.6	6.6	0.0	13.4	13.4	5.8	5.8	5.8
21	2.5	2.5	8.7	0.0	8.7	13.4	13.4	0.0	6.0	6.0
22	5.7	5.7	2.9	2.9	2.8	1.1	0.0	6.6	7.3	5.7
23	9.7	8.8	3.3	0.0	3.3	1.3	1.3	5.8	5.8	5.8
24	3.7	3.7	4.0	4.0	4.0	9.3	9.3	9.2	9.2	9.2
25	9.6	9.6	1.2	1.2	1.2	6.7	6.4	3.1	3.1	3.1
26	8.8	8.8	9.0	9.0	0.0	5.3	0.0	4.7	4.7	4.7
27	4.8	4.8	5.6	5.6	5.6	9.9	9.9	9.7	9.7	0.0
28	8.0	0.0	2.0	2.0	2.0	0.0	3.0	6.7	6.7	6.7
29	5.6	5.6	7.5	7.5	7.5	3.7	3.7	1.8	1.8	1.8
30	4.0	4.0	2.5	2.5	2.5	3.2	3.2	4.3	4.3	4.3
31	4.8	4.4	8.8	8.8	8.8	1.1	1.1	8.4	8.4	8.3
32	8.6	8.6	9.7	10.1	9.7	8.8	8.8	3.1	3.1	3.1
33	10.0	8.9	3.6	3.6	3.6	6.4	6.4	8.0	8.0	8.0
34	5.4	5.4	6.3	5.9	6.3	7.4	7.4	3.4	3.4	3.4
35	4.7	5.4	0.0	13.0	0.0	7.1	7.0	3.9	3.7	3.9
36	9.4	9.4	4.7	4.7	4.7	6.2	6.9	9.7	0.0	0.0
37	1.0	1.0	7.9	7.9	7.9	0.0	14.0	4.9	4.9	4.9
38	7.3	6.9	5.6	6.1	5.6	6.5	6.5	4.4	4.4	4.4
39	4.1	4.1	2.2	2.2	0.0	9.2	8.4	2.1	2.1	2.1
40	1.2	1.4	8.7	8.7	8.7	4.3	4.3	14.0	14.6	14.0
41	1.1	1.0	3.6	3.6	3.6	0.0	6.0	9.4	0.0	9.4
42	2.4	2.4	1.6	1.5	0.0	6.4	6.4	5.0	5.0	5.0
43	1.7	1.7	1.2	1.2	0.0	7.9	7.9	5.2	5.2	5.2
44	12.3	13.5	1.7	1.7	1.7	3.8	4.0	0.0	9.0	9.0
45	2.5	2.5	4.9	4.9	4.9	4.8	0.0	9.6	9.6	9.6
46						2.3	0.0	1.3	1.3	1.3
47						8.5	8.5	3.0	3.0	3.0
48						4.6	4.8	7.6	7.6	7.6
49						13.6	13.6	0.0	2.0	2.0
50						3.6	3.6	8.4	8.4	8.4
51						9.2	9.2	3.8	3.8	3.8

**Tablo A. 4** Kilbridge&Wester ve Tonge test problemlerine ait süreler (devam)

52						1.5	1.5	3.3	3.3	3.3
53						5.1	0.0	7.9	7.9	7.9
54						4.4	4.4	4.6	4.2	5.1
55						3.8	3.8	13.6	13.6	13.6
56						6.7	6.7	5.9	5.9	6.0
57						11.3	11.3	9.5	9.5	9.5
58						7.0	0.0	7.7	0.0	7.7
59						2.2	2.2	4.4	4.4	4.7
60						15.0	15.0	3.9	3.9	3.9
61						8.4	8.4	10.0	11.2	10.0
62						1.3	1.3	8.8	8.8	8.8
63						1.7	1.7	0.0	6.0	6.0
64						7.5	7.5	0.0	7.0	0.0
65						5.1	5.1	4.5	4.5	4.5
66						0.0	5.0	3.7	3.7	3.7
67						1.4	1.6	1.1	0.0	1.0
68						6.5	7.1	2.9	2.9	2.9
69						3.9	3.9	8.7	8.4	8.5
70						2.9	2.9	2.7	3.1	2.7