



**SIRA BAĞIMLI HAZIRLIK ZAMANLARINI DİKKATE ALAN
PARALEL ÇOK İŞÇİLİ MONTAJ HATLARININ DENGELENMESİ**

Funda GÜNER

**DOKTORA TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ŞUBAT 2019

Funda GÜNER tarafından hazırlanan “SIRA BAĞIMLI HAZIRLIK ZAMANLARINI DİKKATE ALAN PARALEL ÇOK İŞÇİLİ MONTAJ HATLARININ DENGELENMESİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ / OY ÇOKLUĞU ile Gazi Üniversitesi Endüstri Mühendisliği Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Bilal TOKLU
Endüstri Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum/onaylamıyorum.

.....

Başkan:
Ana Bilim Dalı, Üniversite Adı (Örnek: Fizik Ana Bilim Dalı, Gazi Üniversitesi)

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum/onaylamıyorum.

.....

Üye:
Endüstri Mühendisliği Ana Bilim Dalı,

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum/onaylamıyorum.

.....

Üye:
Endüstri Mühendisliği Ana Bilim Dalı

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum/onaylamıyorum.

.....

Üye:
Ana Bilim Dalı, Üniversite Adı (Örnek: Fizik Ana Bilim Dalı, Gazi Üniversitesi)

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum/onaylamıyorum.

.....

Tez Savunma Tarihi: 22/02/2019

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Prof. Dr. Sena YAŞYERLİ
Fen Bilimleri Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Funda GÜNER

22/02/2019

SIRA BAĞIMLI HAZIRLIK ZAMANLARINI DİKKATE ALAN PARALEL ÇOK İŞÇİLİ MONTAJ HATLARININ DENGELENMESİ

(Doktora Tezi)

Funda GÜNER

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Şubat 2019

ÖZET

Paralel çok işçili montaj hatları genellikle büyük hacimli ürünlerin (otomobil, otobüs, kamyon, vb.) üretildiği sistemler tarafından tercih edilmektedir. Söz konusu hatları geleneksel montaj hatlarından farklı kılan özellik, bir istasyonda birden fazla işçinin eşzamanlı olarak aynı ara ürün üzerinde çalışabilmesidir. Bu tip büyük hacimli ürünlerin üretildiği sistemlerde, montaj hatlarının geleneksel montaj hatları olarak tasarlanması halinde, üretim için gerekli görev sayısının fazlalığı ve görev süresi uzunlukları da dikkate alındığında gerek üretim sisteminin fiziksel kapasitesini aşan gerekse yatırım maliyetlerini yukarı çeken sonuçlar ortaya çıkabilmektedir. Anılan üretim sistemlerinin paralel çok işçili montaj hattı ile tasarlanması, fiziksel alan kullanımı ve yatırım maliyetleri açısından ileri seviyede bir iyileştirmeye imkân tanıyabilecektir. Bahse konu montaj hatlarında genellikle, aynı işçiye atanan ardışık görevler arasındaki geçişlerde, ekipman değişikliğinden kaynaklanan sıra bağımlı hazırlık zamanları söz konusudur. Bu durum, geleneksel paralel çok işçili montaj hattı dengeleme problemini daha karmaşık hale getirmektedir. Bu kapsamda, işbu tez çalışmasında, sıra bağımlı hazırlık zamanlarını göz önünde bulunduran paralel çok işçili montaj hattı dengeleme problemleri için çözüm yaklaşımları sunulmaktadır. İlk olarak, problemin formal tanımı yapılmış olup, çevrim zamanı kısıdı altında hattaki toplam işçi sayısının birincil amaç, hattaki toplam istasyon sayısının ise ikincil amaç olarak en küçüklendiği iki adet karma tam sayılı programlama modeli sunulmuştur. Sonrasında, problem için hem klasik kısıt programlama yaklaşımı hem de aralık yaklaşımı kullanılarak iki adet kısıt programlama modeli kurulmuştur. Orta ve büyük boyutlu problemlere sonuç elde edebilmek için meta sezgisel algoritmalar (genetik algoritma ve tavlama benzetimi algoritması) geliştirilmiş; ardından problem için bir Benders ayrıştırma metodu önerilmiştir. Tüm bu çözüm yöntemleri literatürde bulunan test problemleri kullanılarak karşılaştırılmış olup, yöntemlere ilişkin performans sonuçları ve ileri çalışma önerileri sunulmuştur.

Bilim Kodu : 90610

Anahtar Kelimeler : Montaj hattı dengeleme, paralel çok işçili montaj hatları, sıra bağımlı hazırlık zamanları, karma tam sayılı programlama, meta-sezgisel algoritmalar, kısıt programlama, Benders ayrıştırması

Sayfa Adedi : 129

Danışman : Prof. Dr. Bilal TOKLU

BALANCING PARALLEL MULTI-MANNED ASSEMBLY LINES WITH SEQUENCE
DEPENDENT SETUP TIMES

(Ph. D. Thesis)

Funda GÜNER

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

February 2019

ABSTRACT

Parallel multi-manned assembly lines are preferred by systems where large size products (cars, buses, trucks, etc.) are produced. The difference from traditional assembly lines is that more than one worker can work simultaneously in a station on the same workpiece. When assembly lines are designed as conventional lines in systems where such large size products are produced, results may exceed physical capacity and may increase investment costs, due to the excessing number and durations of tasks required to produce the final product. Designing these production systems as parallel multi-manned assembly lines may lead a considerable improvement on capacity usage and investment costs. In general, there are sequence-dependent setup times due to equipment changes while the transitions between successive tasks assigned to the same worker. This makes the traditional parallel multi-manned assembly line balancing problem more complicated. The problem involves not only the assignment issue of tasks to a specific station but also scheduling these tasks in that station. In this thesis, solution approaches for parallel multi-manned assembly line balancing problems with sequence-dependent setup times are presented. After a formal definition of the problem, two mixed integer programming models are proposed, where the primary objective is to minimize the total number of workers in the line and the second one is the minimization of the total number of stations. Two constraint programming models are then developed using traditional and the interval variable approaches. In order to obtain results for medium and large-scale problems, two meta-heuristic algorithms (genetic algorithm, simulated annealing algorithm) are developed, and then a Benders Decomposition Method is proposed. By using benchmark instances from the literature, all of these solution methods are tested and their performances are compared. Lastly, results and further study suggestions are presented.

Science Code : 90610

Key Words : Assembly line balancing, parallel multi-manned assembly lines, sequence dependent setup times, mixed integer programming, meta-heuristic algorithms, constraint programming, Benders decomposition

Page Number : 129

Supervisor : Prof. Dr. Bilal TOKLU

TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde, değerli bilgilerini benimle paylaşan, desteğini hiçbir zaman esirgemeyen ve tecrübesiyle bana ışık tutan tez danışmanım Prof. Dr. Bilal TOKLU'ya derin şükranlarımı sunarım. Gerek akademik kariyerimde gerekse sosyal yaşantımda her daim örnek alacağım saygıdeğer hocamla çalışmış olmaktan büyük bir onur duymaktayım. Ayrıca, kendisine ne zaman danışsam kıymetli zamanını ayırıp sabırla ve büyük bir ilgiyle yardımcı olan, yanına çekinmeden gidebildiğim, güler yüzünü ve samimiyetini benden esirgemeyen Doç. Dr. Talip KELLEĞÖZ'e teşekkür ederim. Yine çalışmamda konu, kaynak ve yöntem açısından bana sürekli yardımda bulunarak yol gösteren kıymetli tez izleme kurulu üyeleri Prof. Dr. Burak BİRGÖREN ve Doç. Dr. Selçuk Kürşat İŞLEYEN hocalarıma teşekkür ederim. Kıymetli zamanını ayırıp, değerli bilgilerini benimle paylaşan Prof. Dr. Yakup KARA'ya teşekkür ederim.

Manevi desteği ile beni hiçbir zaman yalnız bırakmayan ve her türlü fedakârlığı gösteren sevgili eşim Doğan ve bana hayat enerji veren biricik oğlum Furkan'a sonsuz şükranlarımı sunarım.

Bu çalışma süresince en yoğun olduğum zamanlarda bile sıkılmadan, yorulmadan yanımda olarak en büyük fedakârlığı gösteren anneme, babama, kardeşime ve eşimin ailesine teşekkürü bir borç bilirim.

Bu çalışma boyunca desteklerini esirgemeyen Çankaya Üniversitesi Endüstri Mühendisliği Bölümü'ndeki mesai arkadaşlarıma teşekkür ederim.

Bu çalışma TUBİTAK BİDEB 2211-C Öncelikli Alanlar Doktora Tezi Bursu Programı kapsamında tamamlanmış olup, TUBİTAK'a teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR.....	xiv
1. GİRİŞ.....	1
2. MONTAJ HATTI Dengeleme Problemleri	5
2.1. Montaj Hattı Dengeleme Problemlerine Ait Temel Terimler	5
2.2. Montaj Hattı Dengeleme Problemlerinin Sınıflandırılması	7
2.2.1. Öncelik diyagramı karakteristikleri	8
2.2.2. Görev (Operasyon) süreleri.....	8
2.2.3. Gecikmeli/Gecikmesiz hatlar	9
2.2.4. Atama kisitlari.....	9
2.2.5. Hat ve istasyon karakteristikleri.....	10
2.2.6. Amaçlar	14
3. LİTERATÜR ARAŞTIRMASI	17
3.1. Çift Taraflı Montaj Hattı Dengeleme Problemleri (ÇMHDP)	18
3.2. Paralel Çok İşçili Montaj Hattı Dengeleme Problemleri	22
3.3. Görevler Arasında Sıra Bağımlı Hazırlık Zamanlarının Bulunduğu Montaj Hattı Dengeleme Problemleri.....	27
4. PROBLEM TANIMI.....	33

	Sayfa
5. ÖNERİLEN MATEMATİKSEL PROGRAMLAMA MODELİ (ÖMAT-1)	37
5.1. Hesaplama Sonuçları ve Değerlendirme	40
6. MONTAJ HATTI DENGELEME PROBLEMLERİ İÇİN ÖNERİLEN KISIT PROGRAMLAMA MODELLERİ	45
6.1. Kısıt Programlama	45
6.2. Karma Tamsayı Programlama ve Kısıt Programlama Karşılaştırması	46
6.3. Önerilen Kısıt Programlama Modelleri	48
6.3.1. Klasik yaklaşım ile KP modeli (KP1)	48
6.3.2. Aralık yaklaşımı ile KP modeli (KP2)	52
6.4. Hesaplama Sonuçları ve Değerlendirme	58
7. ÖNERİLEN METASEZGİSEL YÖNTEMLER	61
7.1. Genetik Algoritma	61
7.1.1. Genetik algoritmaların genel yapısı	62
7.1.2. Önerilen genetik algoritma (GA)	67
7.2. Tavlama Benzetimi Algoritması	80
7.2.1. Fiziksel tavlama tavlama süreci	80
7.2.2. Tavlama benzetimi algoritmasının genel yapısı	81
7.2.3. Önerilen tavlama benzetimi algoritması	82
7.3. Parametre Analizi	84
7.3.1. Tavlama benzetimi parametre seçimi	84
7.3.2. Genetik algoritma parametre seçimi	84
7.4. Pİ-MHDP/SB için Geçerli Bir Alt Sınır	85
7.5. Hesaplama Sonuçları ve Değerlendirme	87

	Sayfa
8. MONTAJ HATTI Dengeleme Problemi İin Benders Ayırıştırması	97
8.1. Benders Ayırıştırmasının Genel Yapısı.....	97
8.2. Geleneksel Benders Ayırıştırması	98
8.3. Mantık Tabanlı Benders Ayırıştırması (MTBA).....	100
8.4. Önerilen Mantık Tabanlı Benders Ayırıştırması Algoritması (ÖMTBA).....	103
8.4.1. Önerilen matematiksel programlama modeli-2 (ÖMAT-2).....	105
8.4.2. Gevşetilmiş ana problem (GAP)	109
8.4.3. Alt problem (AP)	111
8.4.4. Benders kesmeleri	113
8.5. Hesaplama Sonuçları ve Değerlendirme	113
9. SONUÇ ve ÖNERİLER	119
KAYNAKLAR	121
ÖZGEÇMİŞ	128

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Öncelik matrisi.....	6
Çizelge 3.1. Çift taraflı montaj hattı dengeleme problemleri literatür araştırması	22
Çizelge 3.2. Paralel çok işçili montaj hattı dengeleme problemleri	23
Çizelge 3.3. Görevler arası sıra bağımlı hazırlık zamanlarının bulunduğu montaj hattı dengeleme problemleri	32
Çizelge 5.1. Test problemleri.....	41
Çizelge 5.2. Matematiksel programlama modeli sonuçları/ÖMAT-1	42
Çizelge 5.3. Farklı hazırlık zamanı yoğunlukları için ÖMAT-1 sonuçları.....	43
Çizelge 6.1. Optimizasyon programlama dili (OPL) kısıtları.....	55
Çizelge 6.2. Kısıt programlama (KP1) modeli sonuçları.....	59
Çizelge 6.3. Kısıt programlama (KP2) modeli sonuçları.....	59
Çizelge 6.4. Farklı hazırlık zamanı yoğunlukları için KP2 sonuçları.....	60
Çizelge 7.1. İkili kodlama örnek.....	62
Çizelge 7.2. Gerçek değer kodlama örnek	63
Çizelge 7.4. Permütasyon kodlama örnek.....	63
Çizelge 7.5. Tavlama süreci ile tavlama benzetimi algoritması arasındaki benzerlik	81
Çizelge 7.6. Tavlama benzetimi parametre seçimi	84
Çizelge 7.7. Genetik algoritma parametre seçimi	84
Çizelge 7.8. ÖMAT-1 –GA ve TB karşılaştırması-küçük boyut ($\alpha = 0.25$).....	88
Çizelge 7.9. ÖMAT-1 –GA ve TB karşılaştırması-küçük boyut ($\alpha = 0.50$).....	88
Çizelge 7.10. ÖMAT-1 –GA ve TB karşılaştırması-küçük boyut ($\alpha = 0.75$).....	89
Çizelge 7.11. ÖMAT-1 –GA ve TB karşılaştırması-küçük boyut ($\alpha = 1.00$).....	89
Çizelge 7.12. Tavlama benzetimi ve genetik algoritma karşılaştırması-orta boyut ($\alpha = 0.25$)	90

Çizelge	Sayfa
Çizelge 7.13. Tavlama benzetimi ve genetik algoritma karşılaştırması-orta boyut ($\alpha = 0.50$)	91
Çizelge 7.14. Tavlama benzetimi ve genetik algoritma karşılaştırması-orta boyut ($\alpha = 0.75$)	92
Çizelge 7.15. Tavlama benzetimi ve genetik algoritma karşılaştırması-orta boyut ($\alpha = 1.00$)	93
Çizelge 7.16. Tavlama benzetimi ve genetik algoritma karşılaştırması-büyük boyut ($\alpha = 0.25$)	94
Çizelge 7.17. Tavlama benzetimi ve genetik algoritma karşılaştırması-büyük boyut ($\alpha = 0.50$)	94
Çizelge 7.18. Tavlama benzetimi ve genetik algoritma karşılaştırması-büyük boyut ($\alpha = 0.75$)	95
Çizelge 7.19. Tavlama benzetimi ve genetik algoritma karşılaştırması-büyük boyut ($\alpha = 1.00$)	95
Çizelge 8.1. Matematiksel programlama modeli sonuçları/ÖMAT-2	114
Çizelge 8.2. Farklı hazırlık zamanı yoğunlukları için ÖMAT-2 sonuçları.....	114
Çizelge 8.3. ÖMTBA test problemi sonuçları	115
Çizelge 8.4. Farklı hazırlık zamanı yoğunlukları için ÖMTBA sonuçları	116
Çizelge 8.5. ÖMAT-2 ve ÖMTBA karşılaştırması.....	117

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Öncelik diyagramı	6
Şekil 2.2. Montaj hattı dengeleme problemlerinin sınıflandırılması	7
Şekil 2.3. Model bazlı montaj hattı sıralaması.....	8
Şekil 2.4. Düz montaj hattı	10
Şekil 2.5. U-Tipi Montaj Hattı.....	11
Şekil 2.6. Çift taraflı montaj hattı	12
Şekil 2.7. Paralel çok işçili montaj hattı	13
Şekil 4.1. İleri ve geri hazırlık zamanları.....	34
Şekil 7.1 Ağaç kodlama örnek	63
Şekil 7.2. Genetik algoritma genel yapısı	64
Şekil 7.3. Rulet Tekerleği Seçimi Örnek	66
Şekil 7.4. Önerilen genetik algoritma genel yapısı	68
Şekil 7.5. Örnek problem	70
Şekil 7.6. Örnek problem çözüm gösterimi	70
Şekil 7.7. Örnek problem uygun çözüm hat konfigürasyonu	70
Şekil 7.8. Başlangıç popülasyonu algoritması	71
Şekil 7.9. Uygun çözüm algoritması.....	72
Şekil 7.10. Ekleme komşu üretme mekanizması	75
Şekil 7.11. Uygun ekleme hareketleri.....	75
Şekil 7.12. İkili değiştirme komşu üretme mekanizması.....	77
Şekil 7.13. İkili yer değiştirme uygun hareketleri.....	77
Şekil 7.14. Tek noktalı OBX çaprazlama	78
Şekil 7.15. Düzeltme fonksiyonu.....	79

Şekil	Sayfa
Şekil 7.16. Önerilen tavlama benzetimi algoritması	83
Şekil 8.1. Benders ayrıştırması genel yapısı	97
Şekil 8.2. ÖMTBA genel yapısı.....	104
Şekil 8.3. ÖMTBA akış şeması.....	105

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

nt	Toplam görev sayısı.
$M_{\max}$	Bir istasyonda çalışabilecek en fazla işçi sayısı
CT	Çevrim zamanı
t_i	i görevine ait görev süresi
τ_{ih}	i görevi ile h görevi arasındaki ileri sıra bağımlı hazırlık zamanı matrisi
μ_{hi}	h görevi ile i görevi arasındaki geri sıra bağımlı hazırlık zamanı matrisi
$P(i)$	i görevinin hemen (komşu) öncül seti
$P_a(i)$	i görevinin tüm öncül görevleri seti
$S(i)$	i görevinin hemen (komşu) ardıl seti
$S_a(i)$	i görevinin tüm ardıl görevleri seti
φ	Oldukça büyük bir sayı

Kısaltmalar

Açıklamalar

AP	Alt problemler
Bcut	Benders kesmeleri
BMHDP	Basit montaj hattı dengeleme problemi
BMHDP/SB	Sıra bağımlı hazırlık zamanlarını içeren basit montaj hattı dengeleme problemi
ÇMHDP	Çift taraflı montaj hattı dengeleme problemleri
GA	Önerilen genetik algoritma

Kısaltmalar**Açıklamalar****GAP**

Gevşetilmiş ana problem

KP1

Önerilen kısıt programlama modeli-1

KP2

Önerilen kısıt programlama modeli-2

MHDP

Montaj hattı dengeleme problemi

MTBA

Mantık tabanlı Benders ayrıştırması

ÖMAT-1

Önerilen karma tam sayılı programlama modeli-1

ÖMAT-2

Önerilen karma tam sayılı programlama modeli-2

ÖMTBA

Önerilen mantık tabanlı Benders ayrıştırması

Pİ-MHDP

Paralel çok işçili montaj hattı dengeleme problemi

Pi-MHDP/SB

Sıra bağımlı hazırlık zamanlarını içeren paralel çok işçili montaj hattı dengeleme problemi

TB

Önerilen tavlama benzetimi algoritması

1. GİRİŞ

Gelişen üretim teknolojileri, artan müşteri talepleri ve taleplerdeki çeşitlilik üretici firmaların pazardaki paylarını korumak için çok sayıda kaliteli ürünü düşük maliyetle sunabileceği farklı üretim süreçleri geliştirmelerine sebep olmuştur. Bu doğrultuda en çok kullanılan üretim sistemlerinden biri olan montaj hatları, ilk olarak 1915 yılında Henry Ford tarafından Ford Model T arabalarının üretiminde uygulanmıştır. Ancak, gerek farklı ürün özellikleri gerekse de değişen piyasa şartları (*müşteri istekleri, rakip firmaların varlığı, vb.*) beraberinde farklı montaj hattı uygulamalarını (*Düz hatlar, U-tipi hatlar, tek işçili veya çok işçili istasyonlar, karışık veya çoklu modelli hatlar vb.*) ortaya çıkarmıştır.

Montaj hattı dengeleme problemleri (MHDP) bir üretim sisteminde üretilcek ürünün tüm en küçük iş parçalarının (görev) istasyonlara görevler arasındaki öncelik ilişkileri ve diğer bazı kısıtları sağlayarak belirlenen performans ölçütlerini en iyilemesi olarak tanımlanmaktadır (Ağpak ve diğerleri, 2002). MHDP, üretim sistemlerinde karşılaşılan çeşitliliğe cevap vermek için gerek kesin gerekse sezgisel yöntemlerle çözüm aranan bir konu olmuştur. Literatürde NP-Zor sınıfı problemlerden biri olarak bilinmektedir (Erel ve Sarin,1998). MHDP, ilk olarak Helgeson (1954) tarafından çalışılmış ancak matematiksel olarak Salveson (1955) tarafından ifade edilmiştir. İlk çalışmaların yapıldığı bu tarihlerden günümüze kadar birçok araştırmacı problemin farklı varyasyonlarını çalışmıştır.

Son yüzyılda meydana gelen teknolojik gelişmeler sonucu dünya genelinde tüketici taleplerinde olağanüstü artış görülen bilgisayar, cep telefonu, televizyon gibi elektronik ürünler ile motorlu araçlar gibi (*otomobil, kamyon, tır, otobüs, traktör vb.*) endüstriyel ürünlerin üretimlerinde, bahsedilen yüksek talebi karşılamayı teminen montaj hatları daha fazla tercih edilir olmuştur. Bunlardan, büyük hacimli ürünlerin (*kamyon, traktör, otobüs vb.*) istasyonlarında tek bir işçinin bulunduğu geleneksel montaj hattı tasarımı ile üretilmesi halinde çok fazla sayıda istasyona, dolayısıyla da çok daha büyük yatırıma ihtiyaç duyulacaktır. Böylelikle değişen müşteri taleplerine, pazar ihtiyaçlarına etkin cevap vermek imkânsız hale gelecektir. Her ne kadar büyük hacimli ürünlerin üretildiği sistemlerde istasyonlarda birden fazla işçinin çalıştırıldığı uygulamalar görülsede, literatürde bu tip montaj hattı tasarımları ile ilgili çok fazla çalışmaya rastlanılmamaktadır.

Son yıllardaki çalışmalara bakıldığında ise, çok işçili ve çok taraflı montaj hatlarındaki temel problem tanımlarının yapılmaya başlandığı ve çözüm metodolojilerinin geliştirildiği görülmektedir. Ancak, literatürdeki çalışmalar halen endüstrinin ihtiyaçlarına tam anlamıyla cevap verememekte, teorik düzeyde kalmakta ve uygulanabilirlik açısından yeterli aşamaya ulaşamamaktadır. Ek olarak, çok işçili bir montaj hattında aynı istasyon ve aynı işçiye atanan görevler arasında, bir görevden diğer bir göreve geçerken görevin başlangıç süresini etkileyen bir hazırlık zamanı gerekebilmektedir. Sıra bağımlı olan bu hazırlık zamanları, ileri ve geri hazırlık zamanı olarak tanımlanabilir. Bu hazırlık zamanı aynı istasyondaki aynı işçiye atanan ardışık görevler arasında ise ileri; aynı istasyondaki aynı işçiye atanan ilk ve son görev arasında ise geri olarak adlandırılmaktadır (Scholl, 2013). Bu tez kapsamında, büyük hacimli ürünlerin montaj hatlarında sıklıkla görülen tasarım özelliklerinden yola çıkılarak; bir istasyonda, aynı ürün üzerinde eşzamanlı farklı görevler yapabilen birden fazla işçinin bulunmasına izin verilen, görevler arası sıra bağımlı hazırlık zamanlarının bulunduğu paralel çok işçili montaj hattı dengeleme problemi (Pİ-MHDB/SB) çalışılarak, literatürdeki boşluğun olabildiğince giderilmesi amaçlanmıştır.

Tezin bölümleri şu şekilde planlanmıştır. İkinci bölümde, montaj hatları ile montaj hattı dengeleme problemlerine ilişkin genel bilgiler sunulmuş ve montaj hatlarında kullanılan terminolojiden bahsedilmiştir. Ardından, montaj hatları için bir sınıflandırma yapılmıştır. Sınıflandırma; öncelik diyagramı karakteristikleri, görev zamanları, atama kısıtları ve hat ve istasyon karakteristikleri temelinde yapılmıştır. Özellikle, bu tez kapsamında çalışılan hat konfigürasyonunun paralel çok işçili montaj hattı (Pİ-MHDP) olarak tasarlandığı ve görevler arasında sıra bağımlı hazırlık zamanlarının bulunduğu paralel çok işçili montaj hattı dengeleme problemine (Pİ-MHDP/SB) ait temel bilgiler verilmiştir.

Üçüncü bölümde, tez kapsamında çalışılan görevler arasında sıra bağımlı hazırlık zamanlarının bulunduğu paralel çok işçili montaj hattı dengeleme (Pİ-MHDP/SB) problemleri ve benzer problemler için geniş bir literatür taraması sunulmuştur.

Dördüncü bölümde, Pİ-MHDP/SB için detaylı problem tanımı, varsayımları ile beraber yapılmıştır.

Beşinci bölümde, Pİ-MHDP/SB için toplam işçi sayısının birincil ve toplam istasyon sayısının ikincil amaç olarak en küçüklenmesini performans ölçütü olarak ele alan karışık

tam sayılı programlama modeli (ÖMAT-1) kurulmuştur. Literatürdeki test problemleri üzerinde modelin performansı test edilip, sonuçları sunulmuştur.

Altıncı bölümde, matematiksel modelin (ÖMAT-1) sonuçlarından yola çıkarak Pİ-MHDP/SB için iki adet kısıt programlama modeli (KP1, KP2) sunulmuştur. İlk olarak kısıt programlama ile ilgili genel bilgiler verilmektedir. Ardından Pİ-MHDP/SB ve benzeri çalışmalar için bir literatür taraması yapılmıştır. Kurulan model detayları ile birlikte anlatılmıştır. KP' nin etkinliğini ölçmek için ÖMAT-1'de kullanılan test problemlerinin aynısı kullanılmış ve sonuçları raporlanmıştır.

Yedinci bölümde, matematiksel modelin ve kısıt programlama modelinin etkisiz kaldığı problemler için geliştirilen meta sezgisel algoritmalar anlatılmıştır. Tavlama benzetimi (TB) ve genetik algoritma (GA) hakkında genel bilgi verilmiştir. Algoritmaların tarihçelerinden ve literatürdeki kullanımlarından bahsedilmiştir. Ardından Pİ-MHDP/SB' ye özgü tasarlanan algoritmalar tüm detayları ile anlatılmıştır. Algoritmaların çalıştırılmasında ihtiyaç duyulan tüm parametreler için parametre optimizasyonu yapılmıştır. TB ve GA için test problemi sonuçları karşılaştırmalı olarak sunulmuştur.

Sekizinci bölümde, Pİ-MHDP/SB için bir mantık tabanlı Benders Ayırıştırması (MTBA) sunulmuştur. İlk olarak, geleneksel Benders ayırıştırması ile mantık tabanlı Benders ayırıştırmasına ait genel bilgi ilgili literatür taraması ile birlikte verilmiştir. Problemin ayırıştırılabilmesi için ÖMAT-1 den farklı bir karışık tam sayılı programlama modeline ihtiyaç duyulmuştur. Bu sebeple, ÖMAT-2 modeli geliştirilmiştir. Ardından, önerilen mantık tabanlı Benders Ayırıştırması algoritması (ÖMTBA), gevşetilmiş ana problem (GAP), alt problemler (AP) ve bu problemlerin birbirleriyle iletişimini sağlayan Benders kesmeleri detayları ile birlikte anlatılmıştır. Önerilen ÖMTBA algoritmasının test problemi sonuçları sunulmuştur.

Dokuzuncu ve son bölümde ise Pİ-MHDP/SB için yapılan tüm çalışmalar değerlendirilmiş ve elde edilen sonuçlar ışığında ileride yapılabilecek çalışmalardan bahsedilmiştir.

2. MONTAJ HATTI Dengeleme Problemleri

Bu bölümde ilk olarak montaj hatlarında kullanılan temel kavramlar aktarılmış olup, ardından montaj hattı dengeleme problemlerine ait bir sınıflandırma sunulmuştur. Montaj hattı dengeleme problemlerinin sınıflandırılması öncelik diyagramı özellikleri, görev zamanları, problemde ele alınan amaçlar ve hat/istasyon yerleşimi temel alınarak yapılmıştır.

2.1. Montaj Hattı Dengeleme Problemlerine Ait Temel Terimler

Montaj hatlarında kullanılan önemli terimlerden bazıları sunulmuştur (Scholl, 1999; Scholl ve Becker, 2006).

Montaj: Birçok alt parçanın bir araya getirilerek nihai bir ürün oluşturma işlemine montaj denir. Montaj hatları, seri üretim yöntemlerinden biri olarak, standartlaşmış endüstriyel ürünlerin yüksek miktarlarda üretilmesini sağlamaktadır (Scholl ve Boysen, 2009).

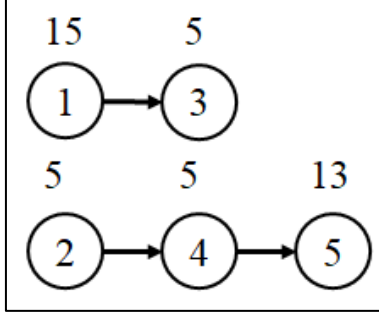
Görev: Montaj hatlarında toplam iş yükünü oluşturan parçaların her birine denir. Görevi tamamlamak için gereken süreye ise *görev süresi* denmektedir. Görev süresi deterministik, stokastik ya da dinamik olabilir (Scholl, 1999).

İstasyon: Montaj hattında nihai ürünü elde etmek için yapılması gereken görevlerin bir kısmının gerçekleştirildiği montaj hattının bölümlerine istasyon denir. Bu bölümlerde yapılan görevlerin görev sürelerinin toplamı *istasyon zamanı* olarak adlandırılmaktadır.

Çevrim zamanı: Hattın her bir istasyonundaki iş parçalarının işlenmesi için verilen maksimum zamanı belirtmektedir (Scholl ve Becker, 2006).

Öncelik diyagramı: Teknolojik kısıtlar sebebiyle montaj hatlarında bazı görevlerin yapılabilmesi için başka görevlerin tamamlanmış olması gerekmektedir. Görevlerin birbirini takip etmesi gerekliliği ile kısmi bir sıralama oluşmaktadır. Bu kısmi sıralama öncelik diyagramı ile gösterilmektedir. Öncelik diyagramında düğümler görevleri, düğümler arası oklar ise görevler arası öncüllük/ardıllık ilişkisini ifade etmektedir.

Düğümlerin üzerinde bulunan rakamlar ise görevin gerçekleştirilebilmesi için gereken görev süresini göstermektedir. Şekil 2.1.'de 5 görevli öncelik diyagramına ait bir örnek verilmiştir. Örneğe göre, görev 3'ün yapılabilmesi için görev 1'in yapılmış olması gerekmektedir. Bir diğer anlatımla, görev 3, görev 1'in ardıl görevidir.



Şekil 2.1. Öncelik diyagramı

Görevler arası öncüllük/ardıllık ilişkisi öncelik diyagramı dışında *öncelik matrisi* ile de gösterilebilmektedir. Çizelge 2.1.'de hemen (komşu) öncelik matrisine ait bir örnek sunulmuştur (Scholl, 1999).

Çizelge 2.1. Öncelik matrisi

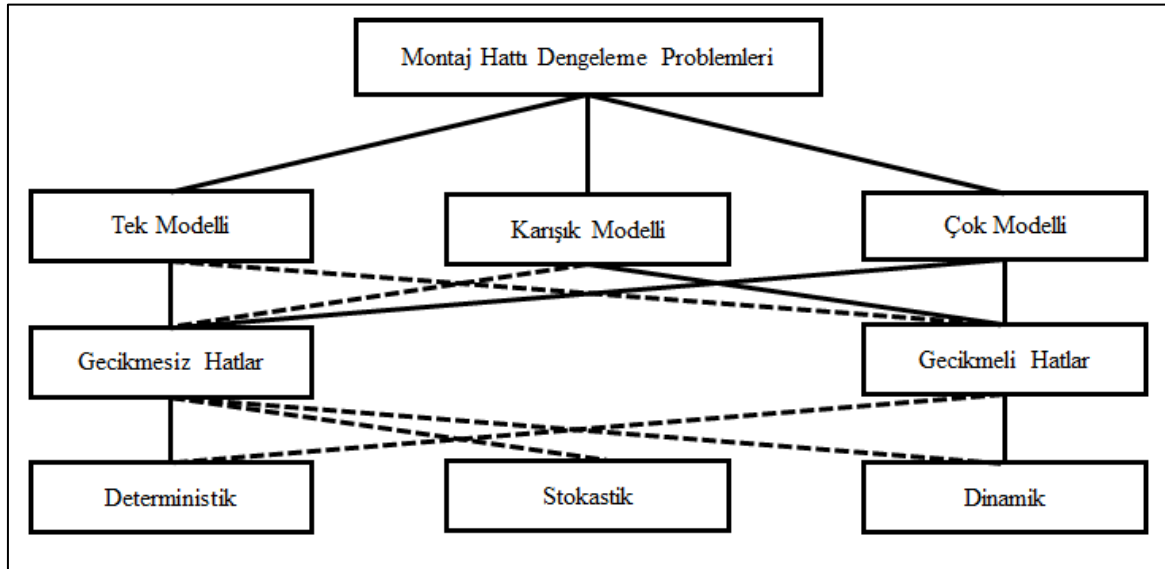
Görevler	1	2	3	4	5
1	0	0	1	0	0
2	0	0	0	1	0
3	0	0	0	0	0
4	0	0	0	0	1
5	0	0	0	0	0

Atıl zaman: Bir istasyonun çevrim zamanı ile istasyon zamanı arasındaki farktır (Scholl, 1999).

Dengeleme problemi: Nihai ürünün üretilmesi için gereken tüm montaj görevlerinin öncelik ilişkileri sağlanarak belirlenen performans ölçütünü eniyileyen sıralı istasyonlara atanması montaj hattı dengeleme problemi olarak tanımlanmaktadır (Ağpak ve diğerleri, 2002).

2.2. Montaj Hattı Dengeleme Problemlerinin Sınıflandırılması

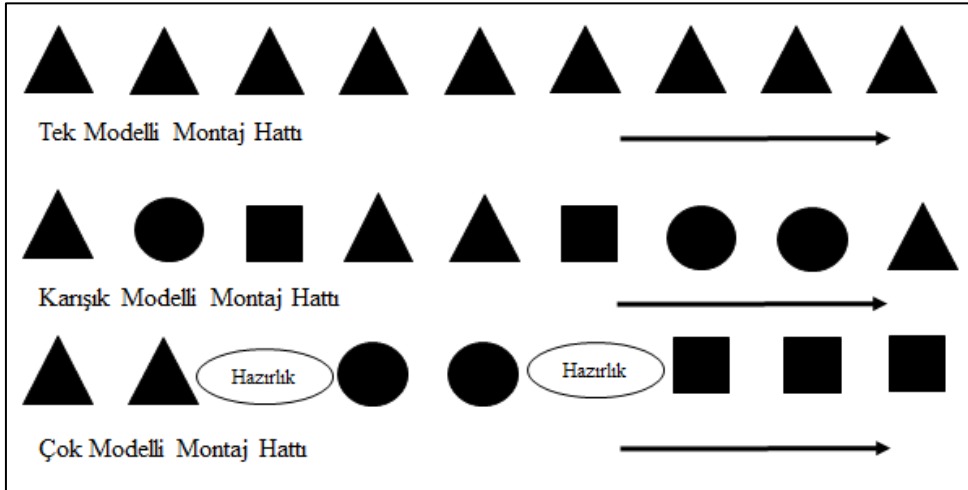
Günümüzde montaj hatları birçok farklı üretim sistemlerinde kullanılmaktadır. Cep telefonu, bilgisayar gibi ürünlerin seri imalatından otomobil, otobüs, kamyon, traktör gibi büyük hacimli ürünlerin üretiminde montaj hatları büyük öneme sahiptir. Ürün çeşitliği ve teknolojik gereksinimler göz önüne alındığında montaj hatlarının sınıflandırması problemin doğru yorumlanması açısından büyük önem arz etmektedir. Şekil 2.2.'de, montaj hatları sınıflandırmasının şematik olarak ifadesi gösterilmektedir (Scholl, 1999). Bu şekil montaj hattı dengeleme problemlerinin temel özelliklerini ve aralarındaki ilişkiyi ifade eder. Bu bağlamda montaj hatları; hatta üretilen ürün tiplerine ait öncelik diyagramı karakteristiklerine (Tek modelli, Karışık modelli ve Çok modelli), hattın gecikmeli ya da gecikmesiz hat olmasına ve görev sürelerinin çeşitliliğine göre ayrılmaktadır. Şemadaki düz çizgilerin kombinasyonu tipik montaj hattı dengeleme problemlerini, kesikli çizgilerin kombinasyonu ise sık karşılaşılmayan montaj hattı dengeleme problemlerini göstermektedir. Şema üzerinde bulunmayan ancak montaj hattı dengeleme problemlerinin sınıflandırılmasında önemli yeri olan atama kısıtları, hat ve istasyon karakteristikleri ve amaçlar ayrıca anlatılmıştır.



Şekil 2.2. Montaj hattı dengeleme problemlerinin sınıflandırılması

2.2.1. Öncelik diyagramı karakteristikleri

Öncelik diyagramı karakteristiklerine göre montaj hatları tek modelli, karışık modelli ve çok modelli hatlar olmak üzere üçe ayrılmaktadır. Şekil 2.3’ te tek modelli, karışık modelli ve çok modelli hatlar gösterilmiştir. Tek modelli hatlarda tek bir ürünün tek bir öncelik diyagramı bulunmaktadır. Hatta üretilen ürünler arasında farklılık görülmez. Karışık modelli hatlarda temel ürünün varyasyonları üretilmektedir. Üretim aşamaları birbirine benzer olduğundan ya çok az hazırlık zamanı gerektirir ya da hiç gerektirmez. Çok modelli hatlarda üretilen ürünler arasında farklılıklar söz konusu olduğundan bir üründen diğer ürüne geçerken hattaki ekipmanların yeniden düzenlenmesi gerekebilir. Bu yüzden çok modelli hatlarda ürünler parti olarak üretilmektedir (Scholl, 1999).



Şekil 2.3. Model bazlı montaj hattı sıralaması

2.2.2. Görev (Operasyon) süreleri

Görev süreleri stokastik, deterministik ve dinamik olabilir. Stokastik görev süresi bulunan montaj hatlarında, görev sürelerinin değişkenlik göstermesi özellikle montaj hattındaki insan (işçi) faktöründen kaynaklanmaktadır. Bu tip hatlarda görev süreleri rassal değişkenler ile ifade edilir. Deterministik görev zamanları bulunan hatlarda görev süreleri sabit olarak alınmaktadır.

2.2.3. Gecikmeli/Gecikmesiz hatlar

Gecikmeli ve gecikmesiz hatlara ilişkin tanımlar aşağıda detaylandırılmıştır.

Gecikmeli hatlar

Bu tip hatlarda ürünün iş parçalarının istasyonlardaki işlemleri tamamlanır tamamlanmaz bu iş parçaları ya bir sonraki istasyona ya da ara stok bölümüne gönderilir. Bu hatlar senkronizasyonun olmadığı hatlardır. Ya da hatta bulunan tüm istasyonlardaki işlemler tamamlandıktan sonra istasyonlarda bulunan parçalar aynı anda bir sonraki istasyona gönderilir. Bu durum ise senkronize hatlar olarak ifade edilir.

Gecikmesiz hatlar

Nihai ürünün iş parçalarının tamamlanması için hatta bulunan istasyonlara verilen süre (çevrim zamanı) birbirine eşittir. Bir diğer ifadeyle, çevrim süresi bitiminde her istasyon iş parçasını bir sonraki istasyona ya sabit hızlı bir konveyör yardımı ile ya da farklı bir yöntem ile bir sonraki istasyona aktarır (Scholl, 1999).

2.2.4. Atama kısıtları

Montaj hatlarında bazı görev grupları aynı istasyona atanmak zorunda iken bazı görev gruplarının aynı istasyona atanmamaları gerekmektedir. Görevlerin aynı istasyona atanmak zorunda olma kısıdına pozitif bölgeleme kısıdı denir. Montaj hattında görevlerin aynı istasyona atanmaması gerekliliğine ise negatif bölgeleme kısıdı adı verilir. Bunlar görev temelli atama kısıtlarıdır. Montaj hattında bazı görevlerin hangi istasyona atanması gerekliliği kısıtları da bulunabilir. Bunlar istasyon temelli atama kısıtlarıdır. Görev ve istasyon temelli atama kısıtlarına ek olarak, büyük hacimli ürünlerin montaj sürecinde karşılaşılan bir istasyonda birden fazla bölgede eşzamanlı olarak işçilerin çalışabildiği durumlar için bazı görevlerin belirli istasyonda belirli bir bölgede yapılabilmesi bölgeleme kısıtlarını ortaya çıkarmaktadır.

2.2.5. Hat ve istasyon karakteristikleri

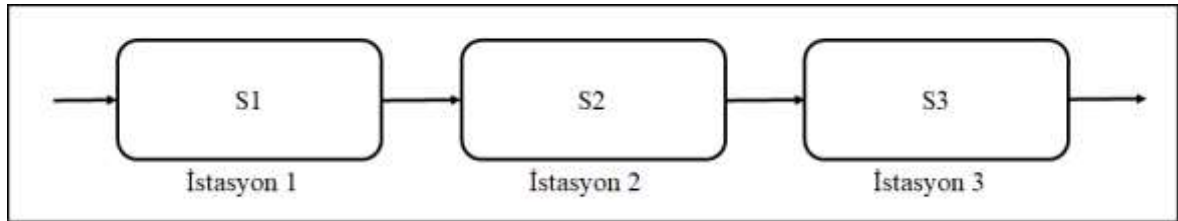
Hat ve istasyon karakteristiklerine göre montaj hatlarının sınıflandırılması aşağıda detaylı bir şekilde anlatılmıştır.

Hat yerleşimine göre montaj hatları

Montaj hattı, düz, paralel ya da U-tipi olabilir.

Düz montaj hatları

Montaj hattı tasarımının en eski ve geleneksel halidir. Düz bir şekle sahip olan hat üzerindeki istasyonlarda bulunan işçiler sadece bulundukları istasyondaki görevlerden sorumludurlar. Literatürdeki birçok montaj hattı dengeleme çalışmasının düz hat planları dikkate alınarak yapıldığı görülmektedir. Düz hatlar Şekil 2.4.'de görselleştirilmiştir.

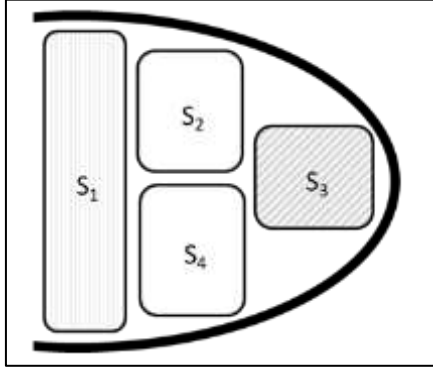


Şekil 2.4. Düz montaj hattı

U-tipi montaj hatları

Düz montaj hatlarının bazı dezavantajlarını (monoton ve kalifiye işçi gerektirmeyen sıkıcı işler, talep değişikliklerine ve üretim sistemi değişikliklerine karşı esnek olmaması) ortadan kaldırmak için kullanılmaya başlanmıştır. Hattın en bariz özelliği bir işçinin aynı çevrim içerisinde farklı istasyonda görev yapmasına izin verilmesidir. U-Tipi montaj hatlarında Şekil 2.5.'ten de anlaşılacağı gibi, işçi ilk istasyona ait bazı görevleri ve son istasyona ait diğer bazı görevleri gerçekleştirebilir. İstasyonlar birbirine oldukça yakındır, böylelikle işçiler birbirleri ile gerek olduğunda kolayca haberleşebilmektedirler. Bu imkanlar, hattın gerçekleştirebilecek değişikliklere karşı esnek olmasını sağlamaktadır (Scholl, 1999). Literatür incelendiğinde hatırı sayılır miktarda çalışmanın da U-tipi montaj hatları üzerinde yapıldığı görülmektedir (Gökçen ve Agpak, 2006; Baykasoğlu ve Özbakir,

2007; Kara ve diğerkleri, 2007; Özcan ve Toklu, 2009; Özcan ve diğerkleri, 2011; Kara ve diğerkleri, 2011; Çelik ve diğerkleri, 2014). U-tipi montaj hatlarında geniş bir literatür taraması için Sivasankaran ve Shahabudeen (2014) çalışmalarına göz atılabilir.



Şekil 2.5. U-Tipi Montaj Hattı

Paralel montaj hatları

Müşteri taleplerine anında cevap verebilmek, hat üzerinde doğabilecek sıkıntıları biraz olsun elimine edebilmek için işletmeler üretimi tek bir montaj hattı üzerinden değil, birden fazla montaj hattı ile gerçekleştirebilirler. Bu sistemlere paralel montaj hatları denmektedir. Sistemdeki bir işçinin aynı çevrim içerisinde farklı hatlardaki görevleri gerçekleştirmesine izin verilmektedir.

Paralel istasyon bulunduran montaj hatları

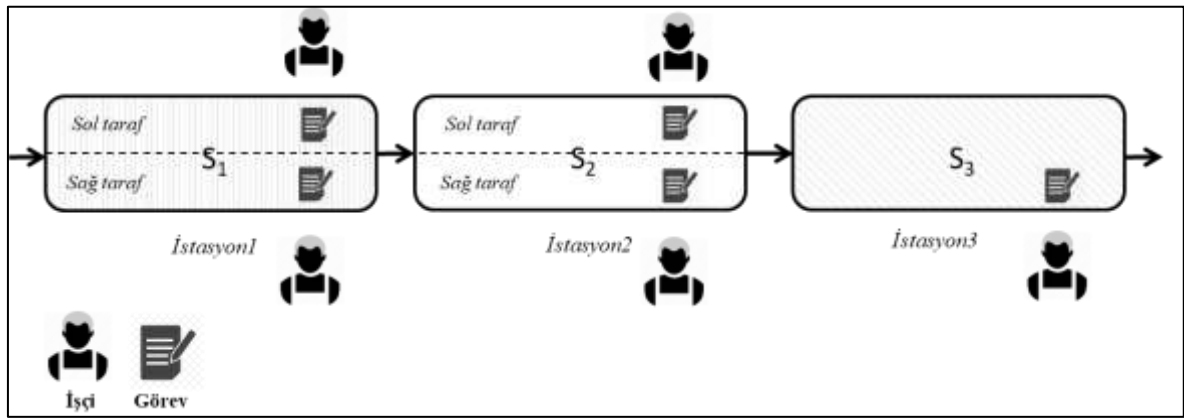
Birden fazla sayıda montaj hattı kurmak yerine var olan montaj hattı üzerindeki bazı istasyonlar için paralel istasyonlar oluşturulabilmektedir.

İstasyon Karakteristiklerine Göre Montaj Hatları

Montaj hatları düz, paralel ya da U-tipi olarak tasarlandıktan sonra, hat üzerinde bulunan istasyonlar geleneksel (basit) olabileceği gibi, çift taraflı ya da çok taraflı (paralel çok işçili) olarak tasarlanabilmektedir.

Çift taraflı montaj hatları

Montaj hattı üzerinde bulunan istasyonların sağ ve sol taraf olarak bölünmesine imkân vermektedir. Çift taraflı olarak tasarlanan montaj hatları aynı istasyon üzerinde iki işçinin eşzamanlı olarak çalışmasına olanak tanımaktadır. Çift taraflı montaj hatlarında görevler, sağ taraf görevleri, sol taraf görevleri, her iki tarafta da yapılabilen görevler ve aynı anda her iki tarafta da yapılması gereken görevler olarak ayrılmaktadır. Çift taraflı montaj hatları Şekil 2.6.'da görselleştirilmiştir.



Şekil 2.6. Çift taraflı montaj hattı

Paralel çok işçili montaj hatları

Büyük hacimli ürünlerin (otomobil, otobüs, kamyon vb.) üretim sürecinde istasyonlarda eşzamanlı olarak birden fazla işçi aynı ara ürün üzerinde çalışabilmektedir. Bu yolla çok sayıda işçiye ihtiyaç duyan montaj hatlarının uzunluğu geleneksel hatlara kıyasla daha kısa olabilmektedir. Böylelikle işletmelerde montaj hattı için gereken fiziksel alan ihtiyacı da azalmaktadır. Aynı istasyonda birden fazla işçinin eşzamanlı olarak kendilerine atanan görevleri gerçekleştirdikleri bu hat tasarımına paralel çok işçili montaj hatları denmektedir. İlk olarak Dimitriadis (2006) tarafından ortaya atılmıştır. Paralel çok işçili montaj hatlarının kullanılabilmesi için hat üzerindeki istasyonların işçilerin eşzamanlı çalışabilmelerine olanak verecek kadar büyük olması gerekmektedir (Dimitriadis, 2006). Paralel çok işçili montaj hatları çift taraflı montaj hatlarına benzemektedir ancak çift taraflı montaj hatlarında hattın üzerindeki istasyonlar sağ taraf ve sol taraf olarak ikiye bölünmüştür ve bu sebep ile istasyonlarda çalışabilecek işçi sayısı en fazla ikidir. Aynı zamanda görevler de sağ taraf görevleri (R), sol taraf görevler (L) ve her iki tarafta

arasında güçlü bir ilişki bulunmamaktadır. Bir diğer ifadeyle geleneksel montaj hatlarında işçi sayısı istasyon sayısına eşit olduğundan (her bir istasyonda bir işçi) işçi sayısını en küçüklemek hat uzunluğunu da en küçüklemektedir. Bu sebeple, paralel çok işçili montaj hatlarında işçi sayısı en küçüklenirken hat uzunluğunu da en küçükleyecek bir yaklaşım gerekmektedir.

Paralel çok işçili montaj hatları günümüz büyük hacimli ürünlerin üretim sistemlerinde hali hazırda yoğun kullanılmasına rağmen, akademik olarak dengeleme problemi çok fazla incelenmemiştir.

Bu tez kapsamında, birden fazla işçinin aynı istasyonda aynı ürün üzerinde eşzamanlı çalışmasına olanak veren, görevler arasında sıra bağımlı hazırlık zamanlarının bulunduğu paralel çok işçili montaj hattı dengeleme problemi göz önünde bulundurulmuştur.

2.2.6. Amaçlar

Üretim sistemlerinde gelen taleplere cevap verebilmek için montaj hatları iyi tasarlanmalıdır. Üretilen ürüne ait görevler istasyonlara etkin bir şekilde dağıtılmalıdır (atanmalıdır). Montaj hatlarında görevlerin istasyonlara atanmasında dikkate alınan amaçlar aşağıda sınıflandırılmaktadır.

İstasyon sayısının en küçüklenmesi (Tip 1)

Montaj hattı tasarımında istasyonlar için belirlenen çevrim süresi kısıdı ile görevler istasyonlara atanırken açılacak olan istasyon sayısının en küçüklenmesi problemidir.

Çevrim zamanının en küçüklenmesi (Tip 2)

Montaj hattındaki istasyon sayısı bilinirken görevlerin istasyonlara çevrim süresinin en küçüklenecek şekilde atanması problemidir.

Hat etkinliđinin en büyüklenmesi (Tip E)

Montaj hattı tasarımında hat etkinliđini en büyükleyecek istasyon sayısı ve çevrim süresinin bulunması problemidir.

Yukarıda belirtilen yaygın kullanılan amaçlar dışında, ana maliyetin en küçüklenmesi, kârın en büyüklenmesi, istasyon zamanlarının düzgünleştirilmesi amaçları montaj hatlarında göz önünde bulundurulmaktadır. Ayrıca, tek bir amaç yerine birden fazla amaç aynı anda montaj hattı tasarımında düşünölebilmektedir.

3. LİTERATÜR ARAŞTIRMASI

Montaj hatları özel bir akış tipi üretim sistemidir ve yüksek miktarda standart ürünlerin endüstriyel üretiminde büyük öneme sahiptir. Son zamanlarda, özelleştirilmiş ve az miktarda üretilen ürünlerin üretiminde de montaj hatlarının kullanımına rastlanmaktadır.

Geleneksel montaj hattı dengeleme araştırmaları, basit montaj hattı dengeleme problemlerini modellemeye ve çözmeye yöneliktir. Basit montaj hattı dengeleme problemlerinin ana karakteristikleri,

- Tek bir modelin (ürünün) seri üretimi,
 - Sabit çevrim zamanlı gecikmesiz hatlar,
 - Deterministik görev zamanları,
 - Öncelik kısıtları dışında atama kısıdının olmaması,
 - Tek taraflı düz hat,
 - Her istasyonun eşit sayıda ekipman, donanıma ve işçiye sahip olması,
- olarak sıralanabilir (Scholl, 1999).

Geleneksel basit montaj hattı dengeleme problemleri ilk olarak Helgeson (1954) tarafından formüle edilmiştir. İlk matematiksel model formülasyonun yayımlanması ve doğrusal programlama sonuçlarının sunulması Salveson (1955) tarafından yapılmıştır. Daha sonra Jackson (1956) söz konusu probleme uygulanabilir ilk dinamik programlama algoritmasını geliştirmiştir. İlk karışık tam sayılı programlama modeli ise Bowman (1960) tarafından sunulmuştur. White (1961) Bowman tarafından sunulan bu modeli iyileştirmiştir. Bu çalışmalardan günümüze montaj hattı dengeleme konusunda araştırmacıların yeni problem tanımları, geliştirilen modeller, sunulan çözüm yöntemleri ile ilgili pek çok çalışma yapmıştır. Baybars (1986), Scholl ve Becker (2006), Becker ve Scholl (2006), Sivasankaran ve Shahabudeen (2014) tarafından sunulan literatür araştırmalarında bu çalışmalara detaylı bir şekilde ulaşılabilmektedir.

Bu çalışma kapsamında incelenen Pİ-MHDP/SB probleminin en bariz özelliği; aynı istasyonda bulunan aynı ara ürün üzerinde birden fazla sayıda işçinin eş zamanlı olarak çalışmasına izin vermesidir. Bu özelliği itibarıyla çift taraflı montaj hattı dengeleme

problemlerine benzerlik arz etmektedir. Bu nedenle, sıra bağımlı hazırlık zamanları içeren hat dengeleme problemleri ve Pİ-MHDP' in yanı sıra çift taraflı montaj hattı dengeleme problemleri konusunda yapılan çalışmalar da literatür taraması kapsamına alınmıştır.

3.1. Çift Taraflı Montaj Hattı Dengeleme Problemleri (ÇMHDP)

Çift taraflı montaj hatları ile ilgili literatür incelendiğinde ilk çalışmanın Bartholdi (1993) tarafından yapıldığı görülmektedir. Söz konusu çalışmada, montaj hattı, sağ ve sol olmak üzere iki tarafa ayrılmış olup, işçiler hattın sağ ve sol tarafında kendilerine atanan görevleri eşzamanlı olarak yapmaktadır. Problemin çözümü için basit bir atama kuralına dayanan sezgisel algoritma geliştirilmiş ve geliştirilen algoritma ile çalışmada sunulan gerçek hayat problemine çözüm aranmıştır. Ayrıca, çift taraflı montaj hatları ile ilgili birtakım teorik özellikler sunulmuş, bu özellikler göz önünde bulundurularak istasyonlara atanan bazı görevleri sabitlemesine izin veren bir bilgisayar programı oluşturulmuştur. Bartholdi' nin bu çalışmasını takiben çift taraflı montaj hattı dengeleme problemleri birçok araştırmacının ilgisini çekmiştir.

Kim ve diğerleri (2000) çevrim zamanı sabit olduğunda istasyon sayısının en küçüklenmesi problemi için genetik algoritma tabanlı bir sezgisel sunmuşlardır. Görevlere ait pozisyon kısıtlarını da göz önünde bulundurmışlardır. Yapılan deneysel çalışmalar sonucunda önerilen algoritma ile kabul edilebilir süreler içerisinde her zaman olurlu çözümlere ulaşmışlardır.

Hu ve diğerleri (2008) istasyon tabanlı atamaya dayalı birerleme yöntemi sunmuşlardır. Tüm görevler için en erken ve en geç istasyon zamanları belirlenerek sundukları yaklaşımla istasyondaki sol tarafı öncelikli tutarak görev atamasını gerçekleştirmişlerdir. Belirtilen yöntem Hoffman (1963) tarafından geliştirilen sezgiselle birleştirilmiş ve çift taraflı montaj hattı dengeleme problemlerinde kullanılmıştır. Deneysel çalışmalar sonucunda küçük boyutlu problemler için sunulan yaklaşımın hesapladıkları alt sınır değerine oldukça yakın olduğunu raporlamışlardır.

Baykasaoğlu ve Dereli (2008), karınca kolonisi tabanlı bir sezgisel geliştirerek bölgeleme kısıtlarının bulunduğu çift taraflı montaj hattı dengeleme problemlerine çözüm aramışlardır.

Wu ve diğerleri (2008), ilk amacın hat uzunluğunun en küçüklenmesi, ikinci amacın istasyon sayısının en küçüklenmesi olan çok amaçlı çift taraflı montaj hattı dengeleme problemleri için bir dal-sınır algoritması geliştirmişlerdir. Gerçekleştirilen deneysel çalışmalar ile daha etkili baskınlık kuralları ve alt sınır bulunmasının gerekliliğini önerdikleri yöntemin daha iyi bir performans sergilemesi için vurgulamışlardır.

Kim ve diğerleri (2009), çift taraflı istasyon sayısının bilindiği, çevrim zamanının en küçüklenmesi hedeflenen, çift taraflı montaj hattı dengeleme problemleri için bir matematiksel model ve genetik algoritma sunmuşlardır. Geliştirilen algoritmayı Bartholdi (1993) ve Kim ve diğerleri (2000) tarafından geliştirilen sezgisellerden daha iyi sonuç verdiği görülmüştür.

Özcan ve Toklu (2009) karışık modelli çift taraflı montaj hattı dengeleme problemi için birincil amacın hat uzunluğunun en küçüklenmesi, ikincil amaç olarak istasyon sayısının en küçüklenmesi olacak şekilde karma tam sayılı programlama modelini sunmuşlardır. Önerilen model temel olarak Kim ve diğerleri (2009) tarafından önerilen modeli almaktadır. Problemin çözümü için tavlama benzetimi tabanlı bir sezgisel yaklaşım geliştirmişlerdir. Önerdikleri sezgisel yaklaşımda çözüm bir görevlere ait öncelik değerleri ile ifade edilmiştir. Görevlerin ataması çözümde bulunan öncelik değerlerine göre gerçekleştirilmektedir. Ayrıca, probleme yönelik bir alt sınır Hu ve diğerleri (2008) tarafından önerilen alt sınır temel alınarak geliştirilmiştir. Deneysel çalışmalar sonucunda küçük boyutlu problemlerde önerilen sezgisel yaklaşımın matematiksel model ile elde edilen optimal sonuçlar ile aynı olduğu, daha büyük boyutlu problemler için önerilen alt sınır değerlerine yakın olduğu raporlanmıştır.

Özcan ve Toklu (2009), çift taraflı montaj hattı dengelem problemi için pozitif ve negatif bölgeleme kısıtlarını da göz önünde bulundurarak 3 farklı matematiksel model geliştirmişlerdir. Bu modeller için Kim ve diğerleri (2009) tarafından çift taraflı montaj hattı dengelem problemlerinde çevrim zamanını en küçüklemek için önerilen model temel alınmıştır. Birinci model birincil amaç olarak hat uzunluğunu ikincil amaç olarak istasyon sayısını en küçüklerken, ikinci model öncelikli hedef programlama yapısında olup göz önünde bulundurulmuş hedefler sırasıyla hat uzunluğu, çevrim zamanı ve istasyonlara atanan görev sayılarının belirlenen hedef değerinden küçük olması ile ilgili modellerdir. Üçüncü

model ise ikinci modeldeki hedeflerin bulanık olması durumunda geliştirilen bulanık karma tam sayılı doğrusal programlama modelidir.

Özcan ve Toklu (2009), çok amaçlı çift taraflı montaj hattı dengeleme problemleri için tabu arama algoritması sunmuşlardır. Performans ölçütü olarak hat etkinliği ve düzgünlük indeksini dikkate almışlardır. Elde edilen deneysel sonuçlar literatürde hâlihazırda bulunan diğer yöntemler ile karşılaştırılmış ve daha iyi sonuç verdiği gözlemlenmiştir.

Simaria ve Vilarinho (2009), birincil amacın verilen çevrim zamanında istasyon sayısını en küçüklemek, ikincil amacın ise modeller arasındaki iş yükü dengesinin sağlanması olan çift taraflı karışık modellenli montaj hattı dengeleme problemi için ilk olarak karma tam sayılı doğrusal olmayan matematiksel model geliştirmişlerdir. Problemin çözümü için 2-ANTBAL adında karınca kolonisi optimizasyonu algoritmasını sunmuşlardır. Yapılan deneysel çalışmalar ile yöntem literatürdeki farklı yöntemler ile karşılaştırılmış ve daha iyi sonuç elde ettiği raporlanmıştır.

Özcan (2010), stokastik görev zamanlı tek modellenli çift taraflı montaj hattı dengeleme problemleri için, ilk aşamada şans kısıtlı parçalı doğrusal karma tam sayılı programlama modeli ve ikinci aşamada tavlama benzetimi sunmuştur. Bahse konu çalışmada, birincil amaç, çift taraflı istasyonların bazı kısıtlar altında en küçüklenmesi; ikincil amaç ise toplam istasyon sayısının en küçüklenmesidir. Daha sonra tavlama benzetimi algoritmasının sonuçları COMSOAL ile karşılaştırmış ve önerilen algoritmanın iyi sonuç verdiği görülmüştür.

Özcan ve Toklu (2010) görevler arası sıra bağımlı hazırlık zamanlarının bulunduğu çift taraflı montaj hattı dengeleme problemleri için birincil amacın hat uzunluğunun en küçüklenmesi, ikincil amacın istasyon sayısının en küçüklenmesi olan çok amaçlı karma tam sayılı matematiksel formülasyonu geliştirmişlerdir. Problemin çözümü için COMSOAL tabanlı bir sezgisel algoritma önermişlerdir. Deneysel çalışmalar ile önerilen metodun performansının oldukça iyi olduğu sonucuna varılmıştır.

Özcan ve diğerleri (2010), paralel çift taraflı karışık modellenli montaj hattı dengeleme problemlerinde istasyon sayısının en küçüklenmesini amaçlayan tabu araması tabanlı bir

sezgisel algoritma sunmuşlardır. Literatürde bulunan veri setleri kullanılarak yapılan analizlerde sunulan algoritmanın performansının iyi olduğunu göstermişlerdir.

Özcan ve diğerleri (2010), paralel karışık modellenli montaj hatlarında dengeleme ve sıralama problemlerinde amaç fonksiyonunun istasyon sayısını ve düzgünlük indeksini dikkate aldığı bir tavlama benzetimi algoritması geliştirmişlerdir. Yapılan deneysel çalışmalar elde edilen sonuçların performans ölçütleri açısından yeterli olduğunu ifade etmiştir.

Yegül ve diğerleri (2010), U tipi çift taraflı montaj hattı dengeleme problemleri için verilen çevrim zamanında istasyon sayısını en küçükleyen çok geçişli rastgele atama algoritması geliştirmişlerdir. Söz konusu çalışmada, montaj hattının bir tarafı U tipi olarak tasarlanırken, diğer tarafı geleneksel düz hat olarak dengelenmektedir.

Özbakır ve Tapkan (2011), alan kısıtlı çift taraflı montaj hattı dengeleme problemleri için arı kolonisi algoritması sunmuşlardır.

Roshani ve diğerleri (2012), çift taraflı montaj hattı dengeleme problemleri için maliyet tabanlı tavlama benzetimi algoritması geliştirmişlerdir.

Tapkan ve diğerleri (2012), tek modellenli deterministik zamanlı çift taraflı montaj hattı dengeleme problemleri için arı algoritması ve yapay arı algoritması tabanlı 2 adet sezgisel algoritma geliştirmişlerdir. Geliştirilen bu sezgiselleri kendi aralarında karşılaştırmışlar ve pek farklı olmadıklarına karar vermişlerdir.

Tapkan ve diğerleri (2012), kısıtlı çift taraflı montaj hattı dengeleme problemini modelleyip ardından arı kolonisi algoritmalarıyla probleme çözüm aramışlardır.

Aghajani ve diğerleri (2014) robotik çift taraflı montaj hattı dengeleme problemlerinde çevrim zamanını en küçüklemeyi amaçlayan karma tam sayılı programlama modelini önermişlerdir. Problemin çözümü için tavlama benzetimi algoritması geliştirmişlerdir.

Küçükkoç ve Zhang (2014) karışık modellenli paralel çift taraflı montaj hatlarında eşzamanlı dengeleme ve sıralama problemleri için ajan tabanlı karınca kolonisi algoritması geliştirmişlerdir.

Çizelge 3.1. Çift taraflı montaj hattı dengeleme problemleri literatür araştırması

Yazar(lar)	Yöntem	Problem Tipi
Bartholdi (1993)	Kullanıcı etkileşimli bilgisayar programı ve atama kuralına dayalı bir sezgisel algoritma	ÇMHDP-I
Kim ve diğerleri (2000)	Genetik algoritma	ÇMHDP-I (pozisyon kısıtlı)
Hu ve diğerleri (2008)	Sezgisel yöntem	ÇMHDP-I
Baykasaoğlu ve Dereli (2008)	Karınca kolonisi algoritması	ÇMHDP-I(bölgeleme kısıtlı)
Wu ve diğerleri (2008)	Dal-sınır algoritması	ÇMHDP-I
Kim ve diğerleri (2009)	Matematiksel model, Genetik algoritma	ÇMHDP-II
Özcan ve Toklu (2009)	Matematiksel model, Tavlama benzetimi	ÇMDP-I
Özcan ve Toklu (2009)	Amaç programlama, Bulanık programlama	ÇMHDP-I(bölgeleme kısıtlı)
Özcan ve Toklu (2009)	Tabu arama algoritması	ÇMDP-I
Simaria ve Vilarinho (2009)	Matematiksel model ve karınca kolonisi algoritması	ÇMDP-II
Özcan (2010)	Matematiksel model ve tavlama benzetimi	ÇMDP-I (Stokastik görev zamanları)
Özcan ve Toklu (2010)	Matematiksel model ve sezgisel algoritma	ÇMDP-I (sıra bağımlı hazırlık zamanları)
Özcan ve diğerleri (2010)	Tabu araması algoritması	PKÇMHDP-I
Özcan ve diğerleri (2010)	Tavlama benzetimi	PKÇMHDP-I ve sıralama
Yegül ve diğerleri (2010)	Çok geçişli rastgele atama algoritması	UÇMHDP-II
Özbakır ve Tapkan (2011)	Arı kolonisi algoritması	ÇMDP-I (alan kısıtlı)
Roshani ve diğerleri (2012)	Tavlama benzetimi algoritması	ÇMDP-I
Tapkan ve diğerleri (2012)	Arı kolonisi algoritması ve yapay arı kolonisi algoritması	ÇMDP-I
Aghajani ve diğerleri (2014)	Matematiksel model ve tavlama benzetimi algoritması	RÇMDP-I
Küçükkoç ve Zhang (2014)	Ajan tabanlı karınca kolonisi algoritması	Karışık ÇMHDP ve sıralama

3.2. Paralel Çok İşçili Montaj Hattı Dengeleme Problemleri

Montaj istasyonlarında, birden fazla işçinin aynı anda çalışmalarına izin verilen hatların paralel çok işçili montaj hatları olarak adlandırıldığından bir önceki bölümlerde bahsedilmiştir. Paralel çok işçili montaj hattı dengeleme problemleri, genelleştirilmiş montaj hattı dengeleme problemleri altında yeni bir araştırma konusudur. Bu problemlerde, her istasyona birden fazla işçinin atanması söz konusudur. Bu tip problemler genel olarak

büyük hacimdeki ürünlerin (*otomobil, kamyon, traktör, otobüs vb.*) üretildiği üretim sistemlerinde görülmektedir. Problem, ilk olarak Dimitriadis (2006) tarafından ortaya atılmıştır. Çok işçili montaj hatları günümüz üretim sistemlerinde hali hazırda kullanılmasına rağmen, akademik camiada çok fazla çalışma ile karşılaşmamaktadır.

Çizelge 3.2. Paralel çok işçili montaj hattı dengeleme problemleri

Yazar(lar)	Yöntem	Problem Tipi
Dimitriadis (2006)	Sezgisel algoritma	Pİ-MHDP1
Cevikcan ve diğerleri (2009)	Karma tam sayılı programlama modeli ve sezgisel algoritma	Pİ-MHDP1 ve Model Sıralama
Becker ve Scholl (2009)	Karma tam sayılı programlama modeli ve dal-dınır algoritması	DA-MHDP (atama ve bölgeleme kısıtlı)
Chang ve Chang (2010)	Karma tam sayılı programlama modeli	Karışık modellenli Pİ-MHDP1
Yazdanparast ve Hajhosseini (2011)	Karma tam sayılı programlama modeli	Pİ-MHDP1
Fattahi ve Roshani (2011)	Karma tam sayılı programlama modeli ve karınca kolonisi algoritması	Pİ-MHDP1
Kellegöz ve Toklu (2012)	Karma tam sayılı programlama modeli ve dal-sınır algoritması	Pİ-MHDP1
Kazemi ve Sedighi (2013)	Karma tam sayılı programlama modeli ve genetik algoritma ve karınca kolonisi algoritması	Pİ-MHDP
Kellegöz ve Toklu (2014)	Karma tam sayılı programlama modeli, yapıcı sezgisel algoritma ve genetik algoritma	Pİ-MHDP1
Kellegöz (2016)	Hedef programlama modeli, Değişken komşu arama	Pİ-MHDP1
Kellegöz (2017)	Karma tam sayılı programlama modeli, gantt temelli sezgisel metot, komşuluk üretme mekanizması	Pİ-MHDP1
Roshani ve Giglio (2016)	Karma tam sayılı programlama modeli, tavlama benzetimi	Pİ-MHDP2
Roshani ve Nazemi (2017)	Karma tam sayılı programlama modeli, tavlama benzetimi	Pİ-MHDP1
Giglio ve diğerleri (2017)	Karma tam sayılı programlama modeli	Pİ-MHDP maliyet minimizasyonu
Naderi ve diğerleri (2018)	Karma tam sayılı programlama modeli, Benders ayrıştırması	Pİ-MHDP1

Dimitriadis (2006), istasyonlarda birden fazla işçiye izin veren ilk araştırmacıdır. Tek modellenli paralel çok işçili istasyonlarda toplam alan ihtiyacını azaltmayı toplam istasyon sayısını en küçükleterek yapmaktadır. Problemin çözümü için Hoffman (1963) tarafından önerilen sezgisel metodu temel alan bir yaklaşım önermiştir. Hoffman (1963) tarafından önerilen sezgisel mevcut istasyon için atanabilecek tüm görevleri birerlemektedir. Bu durum çok fazla hesaplama zamanı getirmektedir. Ancak söz konusu çalışmada önerilen

sezgisel yaklaşım mevcut istasyonda çalışan işçiler için kabul edilebilir bir atıl zamanı sağladığında yeni istasyon açılmaktadır. İşçilere görevler Hu (1961) algoritması ile atanmaktadır. Eğer işçiler için kabul edilebilir bir atıl zamana ulaşıldı ise yeni istasyona geçilir, diğer durumda işçi sayısı bir azaltılarak işlemler tekrarlanır. Deneysel çalışmalar sonucunda hattatın işçi sayısı değişmeden daha az istasyon ile dengelendiği raporlanmıştır.

Cevikcan ve diğerleri (2009), karışık modelli her model için farklı çevrim zamanlarının ve bölgeleme (pozitif ve negatif) kısıtlarının bulunduğu montaj hatlarında hem dengeleme hem de model sıralama problemi için beş aşamalı bir sezgisel yaklaşım sunmuşlardır. İlk aşama yatay dengeleme aşamasıdır. Her bir model ait görevler öncelik kısıtlarını ve çevrim zamanını sağlayacak şekilde sıralı gruplara ayrılır. İkinci aşama dikey dengelemedir. Bu aşamada iş yükü ortalama iş yükünün üzerinde olan model grupları için bu model gruplarına ait bazı görevler iş yükü ortalamasının altında olan gruplara atanır. Üçüncü safha ise fiziksel istasyonların oluşturulmasıdır. Bu aşamada bir önceki aşamalarda elde edilen sıralı görev gruplarından bazıları bir istasyonda birden fazla işçi çalışacak şekilde birleştirilir. Dördüncü aşama model sıralamadır. Hatta girecek olan modellerin sıraları bu aşama ile belirlenmektedir. Son aşama olan beşinci aşama ise talep değişikliği ya da yeni bir modelin hatta eklenmesi durumunda hat üzerinde işçi transferinin yapılabileceğini göstermektedir. Yapılan deneysel çalışmalar sonucunda Dimitriadis (2006) tarafından önerilen sezgiselden daha iyi sonuç verdiği gözlemlenmiştir. Ek olarak, geliştirilen yöntem traktör üretimi yapan bir firmada uygulanmış ve sonuçları sunulmuştur.

Becker ve Scholl (2009), tek modelli, çevrim zamanı ve istasyon sayısının bilinip, işçi sayısının en küçüklendiği montaj hattı dengeleme probleminde, istasyonu çalışma alanlarına bölmüş ve her bir iş alanında bir işçi çalışacağını varsayarak değişken çalışma alanlı montaj hattı dengeleme problemi (DA-MHDP) tanımını yapmıştır. DA-MHDP Pİ-MHDP' nin özel bir halidir. İstasyonlar belirli sayıda pozisyona ayrılmaktadır ve bu pozisyonlarda çalışacak işçi sayısı bilinmektedir. Problemin karma tam sayılı matematiksel modeli çalışma alanlarına ait atama kısıtları ile bölgeleme kısıtları dikkate alarak yapmışlardır. Basit montaj hattı dengeleme problemleri için geliştirilen alt sınır yöntemleri bu problem için uyarlanmıştır. Ardından, problemin çözümüne yönelik Scholl ve Klein (1997) tarafından geliştirilen SALOME isimli metot temel alınarak bir dal sınır algoritması önermişlerdir. Geliştirilen yöntemin Kim ve diğerleri (2000) ve Baykasaoğlu ve Dereli (2008) tarafından çift taraflı montaj hattı dengeleme problemleri için geliştirilen

yöntemlerden daha etkin olduğu yapılan deneysel çalışmalar ile gösterilmiştir. Önerilen dal sınır algoritması küçük ve orta büyüklükteki problemlerde etkin sonuç verirken, büyük boyutlu problemlerde önerilen yaklaşıma bir süre sınırlaması vererek sezgisel metot olarak kullanılmasının uygun olacağı ifade edilmiştir.

Chang ve Chang (2010), karışık modelli çok işçili montaj hattı dengeleme problemlerinde çok işçili istasyonların sayısını en küçükleyen matematiksel modeli sunmuşlardır.

Yazdanparast ve Hajihosseini (2011), çok işçili montaj hatlarındaki istasyonlardaki çok fazla işçi yoğunluğunun görev zamanları üzerindeki negatif etkisine dikkat çekerek, ilgili problemde birincil amaç olarak toplam işçi sayısının azaltılması ikincil amaç olarak istasyon sayısının azaltılması amaçlayan matematiksel model geliştirmişlerdir.

Fattahi ve diğerleri (2011), birincil amacın toplam işçi sayısının en küçüklenmesi ikincil amacın ise toplam istasyon sayısının en küçüklenmesi olan çok işçili montaj hattı dengeleme problemi için Kim ve diğerleri (2009) tarafından çift taraflı montaj hattı dengeleme problemleri için kurulan karışık tam sayılı matematiksel modelini modifiye ederek yeni bir matematiksel bir model sunmuşlardır. Çözüm için, karınca kolonisi algoritması tabanlı sezgisel bir yaklaşım geliştirmişlerdir. Önerilen sezgisel algoritma, Dimitriadis (2006) tarafından sunulan sezgisel algoritma sonuçları ile karşılaştırılmış ve daha iyi sonuç verdiği gözlemlenmiştir.

Kellegöz ve Toklu (2012), paralel çok işçili montaj hattı dengeleme problemlerinde toplam işçi sayısını en küçükleyen JUMPER adında dal sınır algoritması geliştirmişlerdir. Geliştirilen algoritma Becker ve Scholl (2009) tarafından geliştirilen algoritmayla karşılaştırılmış ve JUMPER CPU zamanı ve olurlu çözümlerin kalitesi bakımından daha iyi sonuç verdiğini göstermişlerdir.

Kazemi ve Sedighi (2013), üretilen ürün başına toplam maliyeti en küçükleyen çok işçili karışık modelli montaj hattı dengeleme problemine ait matematiksel model geliştirmişlerdir. Ardından genetik algoritma ve parçacık sürüsü optimizasyonu algoritmalarını sunmuşlardır.

Kellegöz ve Toklu (2014), paralel çok işçili montaj hattı dengeleme problemini ele almışlardır. Tanımlanan problem için karma tam sayılı programlama modeli sunmuşlardır. Sunulan model sadece küçük boyutlu problem örnekleri için optimal sonuca ulaşabildiğinden, orta ve büyük boyutlu problemler için öncelik tabanlı yapıcı sezgisel yaklaşım ve genetik algoritma tabanlı iyileştirme algoritması geliştirmişlerdir. Yapılan deney tasarımı ile birlikte karşılaştırmalı sonuçları sunmuşlardır. Önerilen yaklaşımın büyük orta ve büyük boyutlu problemlerde kabul edilebilir zamanlarda kabul edilebilir sonuçlar elde edildiği raporlanmıştır.

Kellegöz (2016) paralel çok işçili montaj hattı dengeleme problemi için bir hedef programlama modeli önermiştir. Önerilen modelde hedefler; hatta açılan istasyon sayısının en küçüklenmesi (birincil), hatta kullanılan işçi sayısının en küçüklenmesi (ikincil) ve her istasyonda kullanılan işçi sayılarının düzgünleştirilmesi (üçüncül) olarak sıralanmıştır. Problemin çözümü için geliştirilen yaklaşımlar değişken komşu arama algoritmasına dayanmaktadır. Yapılan deneysel çalışmalar ile önerilen yaklaşımlar karşılaştırılmıştır.

Kellegöz (2017) çalışmasında birincil amacın toplam işçi sayısının, ikincil amacın toplam istasyon sayısının minimizasyonu olan paralel çok işçili montaj hattı dengeleme problemi için yeni bir karma tam sayılı programlama modeli geliştirmiştir. Modelin problem için daha önceden geliştirilen modellerin sahip olduğu karar değişkeni ve kısıt sayısından daha azına sahip olduğunu ifade etmiştir. Problemin çözümü için geliştirilen sezgisel metodun özelliği problemin çözümünün Gantt diyagramı üzerindeki gösterimine dayanmasıdır. Buna ek olarak çalışmada önemli komşuluk üretim mekanizmaları sunulmuştur. Deneysel çalışmalar sonucunda elde edilen sonuçların kabul edilebilir olduğu raporlanmıştır.

Roshani ve Giglio (2016) çalışmalarında paralel çok işçili montaj hatlarında çevrim zamanını en küçükleyen karma tam sayılı programlama modeli geliştirmişlerdir. Problemin çözümü için tavlama benzetimi tabanlı 2 algoritma önermişlerdir. Yapılan deneysel çalışmalarda kendi önerdikleri algoritmaları birbirleri ile karşılaştırmışlardır.

Roshani ve Nazemi (2017) karışık modelli çok işçili montaj hattı dengeleme problemi için karma tam sayılı programlama modeli önermişlerdir. Çalışmanın amacı karışık modelli çok işçili montaj hatlarında kullanılan işçi sayısının ve çok işçili açılan istasyon sayısının minimizasyonu olarak belirlenmiştir. Çözüm olarak tavlama benzetimi algoritması

geliştirmişlerdir. Yapılan deneysel çalışmalar sonunda önerilen algoritmanın oldukça iyi sonuçlar verdiği gözlemlenmiştir.

Giglio ve diğerleri (2017) çok işçili montaj hattı dengelem problemi için hattın operasyonel maliyetini en küçükleyen bir formülasyon sunmuşlardır. Ayrıca dikkate alınan problemde, istasyondaki işçilerin kalifikasyonları birbirinden farklıdır. Sunulan formülasyonun hattın toplam maliyetini iyileştirirken kullanılan işçi sayısını ve açılan istasyon sayısını da azalttığı görülmüştür.

Naderi ve diğerleri (2018) birden fazla modelli otomobil üretimi yapan bir firmada gerçek hayat problemi olan karışık modelli işçilerin hareket ettiği, istasyon sayısı ve çevrim zamanı bilinirken hatta kullanılacak toplam işçi sayısını en küçükleyen paralel çok işçili montaj hattı dengeleme probleminin karma tam sayılı programlaması sunmuş ve çözüm yöntemi olarak mantık tabanlı Benders ayrıştırması algoritması önermişlerdir. Önerdikleri Benders ayrıştırması için ana problem görevlerin işçilere atanması ile ilgilenirken, birincil alt problem görevlerin çizelgelenmesi ile ilgilenmektedir ve atamaların uygunluğunu kontrol eder. Uygun olamayan birincil alt problemler için ikincil alt problemler yeniden atama ve çizelgeleme yapmaktadır. Elde edilen çözümün uygunluğu tekrar kontrol edilir ve tüm istasyonlar için uygun çözüm elde edildiğinde algoritma sonlanır, en iyi çözüme ulaşılmıştır. Yazarlar iki aşamalı bir alt problem belirlemenin çözüm zamanı kalitesi açısından daha etkin olduğunu yapılan deneysel çalışmalar sonunda raporlamışlardır.

3.3. Görevler Arasında Sıra Bağımlı Hazırlık Zamanlarının Bulunduğu Montaj Hattı Dengeleme Problemleri

Geleneksel montaj hattı dengeleme araştırmaları çoğunlukla basit montaj hattı dengeleme problemleri üzerine yapılmıştır. Ancak son zamanlarda daha gerçekçi ve genelleştirilmiş montaj hattı dengeleme problemlerinin de hatırı sayılır bir miktarda çalışıldığı görülmektedir. Genelleştirilmiş montaj hattı dengeleme problemleri temel montaj hattı dengeleme problemlerinin varsayımlarından bir ya da birkaçının gevşetilmiş halidir. Bunlara maliyet fonksiyonları, paralelleştirme, U tipi montaj hatları (düz montaj hatlarının özel bir hali olarak), karışık modelli montaj hatları vb. örnek verilebilir. Bu özelliklerin yanında, daha gerçekçi modellerin kurulmasına ve etkin çözüm algoritmalarının geliştirilmesine ihtiyaç vardır. Bu bağlamda, çoğu üretim hattında, istasyonlarda

gerçekleştirilen görevler arasında sıra bağımlı hazırlık zamanları mevcuttur. Bu tez kapsamında, daha önceden de bahsedildiği üzere, basit montaj hattı dengeleme problemlerinin genelleştirilmiş hali olan paralel çok işçili montaj hatları sıra bağımlı hazırlık zamanlarına sahip görevlerin de bulunduğu hali ile çalışılmaktadır. Böylelikle sadece istasyonlar arası görev ataması değil istasyonlar içi görev çizelgelemesi problemi de ortaya çıkmaktadır.

Birçok montaj hattında hazırlık zamanı bulunmasına rağmen, söz konusu hazırlık zamanları görev zamanları ile karşılaştırıldığında göz ardı edilmekte veya hatta bulunan görevlerden bağımsız oldukları düşünülerek görev zamanına eklenmektedir.

Montaj hattı dengeleme literatürüne bakıldığında paralel çok işçili montaj hattı dengeleme problemlerinin göreceli olarak araştırmacıların henüz ilgisini çekmiş yeni bir konu olduğu söylenebilir. Bu kapsamda, sıra bağımlı görev zamanlarının bulunduğu paralel çok işçili montaj hattı dengeleme problemine ilişkin olarak, bilindiği kadarı ile literatürde herhangi bir araştırma bulunmamaktadır. Ancak, tek bir ürünün üretildiği hat üzerinde bulunan istasyonlarda tek bir işçinin çalıştığı basit montaj hattı dengeleme problemlerinde sıra bağımlı hazırlık zamanları ile ilgili birkaç çalışma görülmektedir. Bu bağlamda, basit montaj hattı dengeleme problemlerinde ilk defa sıra bağımlı hazırlık zamanlarını dikkate alan Andres ve diğerleri (2008) olmuştur. Söz konusu problem için 0-1 tam sayılı programlama modeli geliştirmişlerdir. Önerilen matematiksel modelin test problemleri sonuçlarına bakıldığında problemin NP-Zor yapısı ortaya çıkmaktadır. Bu sebeple çözüm için 8 adet farklı sezgisel ile GRASP tabanlı bir sezgisel algoritma sunulmuştur. Çalışmada tüm test problemleri için kesin çözümleri elde etmek mümkün olmadığından elde edilen sonuçların karşılaştırılması için alt sınır hesaplamalarına yer verilmiştir. Sonuç olarak literatürde montaj hatlarında sıra bağımlı hazırlık zamanlarını içeren problemler için daha fazla çözüm yöntemlerine ihtiyaç duyulduğu gözlemlenmiştir.

Martino ve Pastor (2009), öncelik kurallarına dayalı bir sezgisel algoritmayı literatüre kazandırmışlardır.

Scholl ve diğerleri (2008) tarafından yapılan çalışmada, sıra bağımlı hazırlık zamanlı görevler herhangi bir görev herhangi bir görevden sonra yapılıyor ise o görev zamanına bir

miktar zaman artışı eklenir şeklinde tanımlanmıştır. Bu çalışma Andres ve diğerleri (2008) ve Martino ve Pastor (2009) yaptıkları çalışmalardan tanım olarak farklıdır.

Özcan ve Toklu (2010), çift taraflı montaj hattı dengeleme problemlerinde sıra bağımlı hazırlık zamanlı görevlerin olduğu problemi ilk defa tanımlamışlardır. Belirtilen probleme matematiksel bir model geliştirebilmek için ilk olarak Kim ve diğerleri (2009) tarafından çift taraflı montaj hattı dengeleme problemleri için geliştirilmiş olan ilk karma tam sayılı programlama modelini Özcan ve Toklu (2009) çalışmalarında kesin çözüm elde etmek için üzerinde değişiklik yapmışlardır. Daha sonra, değişiklik yapılan bu model ile Andres ve diğerleri (2008)'ne ait olan modeli kullanarak sıra bağımlı hazırlık zamanlı görevlerin bulunduğu çift taraflı montaj hattı dengeleme problemleri için matematiksel model sunmuş ardından, COMSOAL tabanlı 2-COMSOAL/S olarak adlandırdıkları sezgisel yaklaşımı geliştirmişlerdir.

Yazdanparast ve diğerleri (2011), Andres ve diğerleri (2008) tarafından tanımlanan sıra bağımlı hazırlık zamanları ile montaj hattı dengeleme problemleri için matematiksel bir model sunup, maliyet tabanlı bir sezgisel yaklaşım geliştirmişlerdir.

Seyed-Alagheband ve diğerleri (2011), görevler arası sıra bağımlı hazırlık zamanlarının bulunduğu montaj hattı dengeleme problemlerine çözüm getirmek amacıyla tavlama benzetimi yaklaşımı geliştirmişlerdir.

Yolmeh ve Kianfar (2012), sıra bağımlı hazırlık zamanlarının bulunduğu basit montaj hattı dengeleme problemlerinde çevrim zamanını en küçükleme için etkili bir hibrit genetik algoritma önermişlerdir. Sıra bağımlı hazırlık zamanları ileri hazırlık ve geri hazırlık zamanı olmak üzere ikiye ayrılmıştır. Önerilen algoritmanın sonuçlarının tanımlanan problemle ilgili bugüne kadar sunulan algoritmalarından daha iyi sonuç verdiğini göstermişlerdir.

Hamta ve diğerleri (2012), esnek görev zamanları, sıra bağımlı hazırlık zamanları ile öğrenme eğrisini dikkate alan çok amaçlı montaj hattı dengeleme problemi için hibrit parçacık sürüsü optimizasyonu algoritması geliştirmişlerdir.

Scholl ve diğerleri (2013), sıra bağımlı hazırlık zamanlarının (ileri hazırlık zamanı ve geri hazırlık zamanı) bulunduğu montaj hattı dengeleme ve çizelgeleme problemlerinde toplam

istasyon sayısını en küçükleyen karma tam sayılı programlama modeli önermişlerdir. Ardından Andres ve diğerleri (2008) tarafından problem için geliştirilen alt sınır formülasyonunu genişletmişlerdir. Problemin çözümü için etkin sezgisel algoritmalar önermişlerdir. Deneysel çalışmalar sonucunda önerilen metotların problem için daha önceden geliştirilen metotlardan daha etkin sonuç verdiği raporlanmıştır.

Montaj hatlarında sıra bağımlı hazırlık zamanlarının dışında sıra bağımlı olmayan hazırlık zamanlarının da son zamanlarda literatürde çalışıldığı görülmektedir. Akpınar ve diğerleri (2013), görevler arası sıra bağımlı hazırlık zamanlarının bulunduğu karışık modelli montaj hattı dengeleme problemleri için karınca kolonisi algoritmasını genetik algoritma ile hibritleştirerek sezgisel bir yaklaşım sunmuşlardır.

Akpınar ve Baykasoğlu (2014a) ile Akpınar ve Baykasoğlu (2014b), çalışmalarında ilk olarak hazırlık zamanlarının bulunduğu karışık modelli montaj hattı dengeleme problemi için matematiksel model sunmuş, ardından çoklu kolonili hibrit arı algoritması geliştirmişlerdir. Belirtilen algoritmayı tek kolonili arı algoritmasıyla karşılaştırmışlar ve belirtilen algoritmanın daha iyi sonuç verdiğini gözlemlemişlerdir.

Esmailbeigi ve diğerleri (2016) görevler arasında sıra bağımlı hazırlık zamanlarının bulunduğu basit montaj hattı dengeleme problemlerinde belirli bir çevrim zamanı içerisinde toplam istasyon sayısını en küçükleyen üç farklı tam sayılı programlama modeli sunmuşlardır. Ayrıca, literatürde var olan modellerin eksikliklerini ortaya çıkarmışlardır. Sunulan modellerin ikisi istasyon tabanlı modellenmiş olup, bir tanesi çizelgeleme tabanlı (karar değişkenleri) istasyondan bağımsız olarak modellenmiştir. Deneysel çalışmalarda bu modellerin birbirlerine olan üstünlükleri gösterilmiştir.

Akpınar ve diğerleri (2016) çalışmalarında hazırlık zamanlarını göz önünde bulunduran basit ve karışık modelli montaj hattı dengeleme problemleri için toplam istasyon sayısını en küçükleyen bir kesin çözüm yöntemi önermişlerdir. Problemde verilmesi gereken iki kararın olduğunu belirtmişlerdir. Bunlar, görevlerin istasyonlara atanması ve istasyonlara atanan görevlerin çizelgelenmesidir. Problemin bu şekilde ayrıştırılabilmesi sebebiyle, yazarlar Benders ayrıştırmasına dayanan bir kesin çözüm metodu geliştirmişlerdir. Ana problem olarak görevlerin istasyonlara atanması, alt problem olarak görevlerin istasyonlarda çizelgelenmesi belirlenmiştir. Ana problem ile alt problemlerin birbirleri ile

arasındaki iletişimi uygunluk kesmeleri ile sağlamışlardır. Yapılan deneysel çalışmalar sonucunda önerilen Benders ayrıştırmasının önerilen tam sayılı programlama modelinden daha fazla sayıda en iyi sonucu elde ettiği raporlanmıştır.

Şahin ve Kellegöz (2017) U-tipi montaj hatlarında çevrim zamanının en küçüklendiği dengeleme problemi için Scholl ve diğerleri (2013) tarafından ile Dong ve diğerleri (2014) tarafından önerilen matematiksel formülasyonlar temel alınarak bir karma tam sayılı programlama modeli önerilmiştir. Çözüm için tavlama benzetimi algoritması ve genetik algoritma tabanlı iki adet sezgisel geliştirmişlerdir. Alt sınır hesaplamaları için gereken minimum hazırlık zamanlarını bir atama problemi olarak Macar algoritması ile elde etmektedirler. Deneysel sonuçlar problem için matematiksel model ile kabul edilebilir bir zamanda sonuç elde etmenin mümkün olmadığını, önerilen sezgisel metotların hesaplanan alt sınır değerlerinde kabul edilebilir uzaklıkta olduğunu göstermiştir.

Delice ve diğerleri (2018) görevler arasında sıra bağımlı hazırlık zamanlarının bulunan U-tipi çift taraflı montaj hattı dengeleme problemine birincil amacın açılan çift taraflı istasyonların en küçüklenmesi, ikincil amacın ise toplam istasyon sayısının en küçüklenmesi için karınca kolonisi algoritması önermişlerdir. U-tipi çift taraflı montaj hatlarında hazırlık zamanlarını göz önünde bulunduran ilk çalışmadır. Ayrıca sıra bağımlı hazırlık zamanları üç gruba ayrılmıştır. Bunlar; ileri hazırlık zamanı, geri hazırlık zamanı ve karşıdan karşıya geçirme hazırlık zamanıdır. Deneysel çalışmalar sonucunda elde literatürdeki test problemleri için elde edilen sonuçların ümit verici olduğunu göstermişlerdir.

Çizelge 3.3. Görevler arası sıra bağımlı hazırlık zamanlarının bulunduğu montaj hattı dengeleme problemleri

Yazar(lar)	Yöntem	Problem Tipi
Andres ve diğerleri (2008)	0-1 Tam sayılı programlama modeli, sezgisel algoritmalar	SB-BMHDP-1
Scholl ve diğerleri (2008)	Matematiksel model, arama yöntemleri	SB-BMHDP-1
Martino ve Pastor (2009)	Öncelik kuralına dayalı sezgisel algoritma	SB-BMHDP-1
Özcan ve Toklu (2010)	Matematiksel model ve sezgisel algoritma	SB-ÇMHDP-1
Yazdanparast ve diğerleri (2011)	Matematiksel model ve sezgisel algoritma	SB-BMHDP-1
Seyed-Alagheband ve diğerleri (2011)	Tavlama benzetimi	SB-BMHDP-1
Yolmeh ve Kianfar (2012)	Hibrit genetik algoritma	SB-BMHDP-1
Hamta ve diğerleri (2012)	Hibrit parçacık sürüsü optimizasyonu algoritması	SB-BMHDP-1 (esnek görev zamanları)
Scholl ve diğerleri (2013)	Matematiksel model ve sezgisel algoritmalar	SB-BMHDP-1
Akpınar ve diğerleri (2013)	Hibrit karınca kolonisi algoritması ve genetik algoritma	SB-KMHDP-1
Akpınar ve Baykasoğlu (2014a)	Matematiksel model	SB-KMHDP-1
Akpınar ve Baykasoğlu (2014b)	Çoklu kolonili hibrit arı algoritması	SB-KMHDP-1
Esmailbeigi ve diğerleri (2016)	Karma tam sayılı programlama modeli	SB-BMHDP-1
Akpınar ve diğerleri (2016)	Karma tam sayılı programlama modeli, Benders ayrıştırması	SB-KMHDP-1
Şahin ve Kellegöz (2017)	Karma tam sayılı programlama modeli, genetik algoritma, tavlama benzetimim algoritması	SB-UMHDP-2
Delice ve diğerleri (2018)	Karınca kolonisi algoritması	SB-UMHDP-1

4. PROBLEM TANIMI

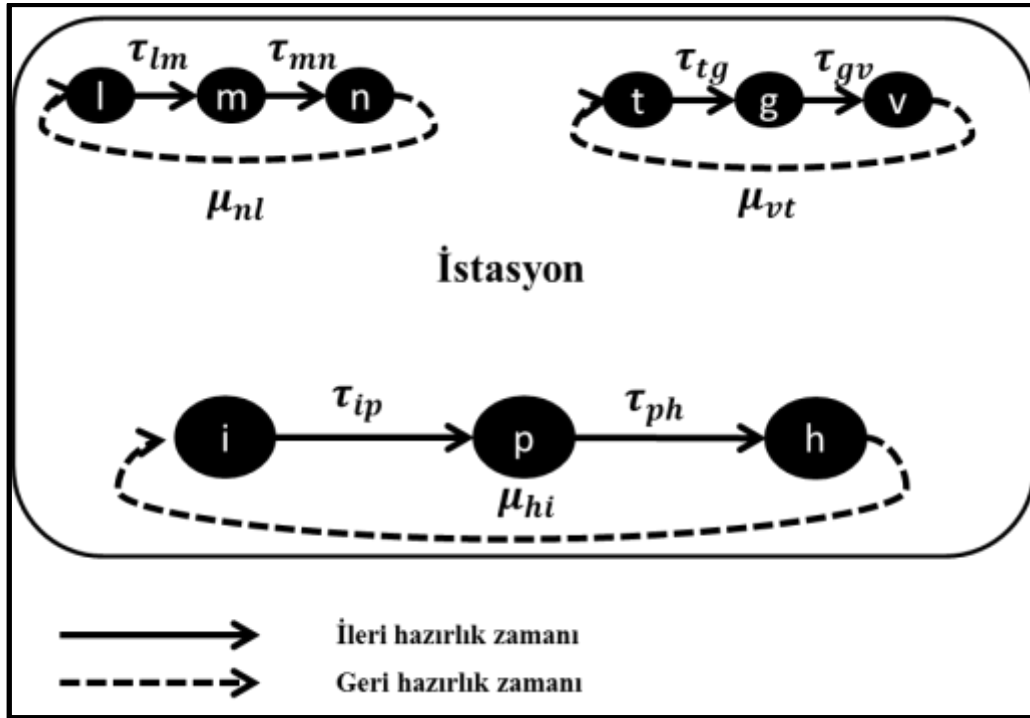
Bu tez kapsamında görevler arası sıra bağımlı hazırlık zamanlarının bulunduğu paralel çok işçili montaj hattı dengeleme problemleri (Pİ-MHDP/SB) göz önünde bulundurulmuştur. Görevler arası sıra bağımlı hazırlık zamanlarının bulunduğu paralel çok işçili montaj hattı dengeleme problemleri, paralel çok işçili montaj hattı dengeleme problemlerinin bir uzantısıdır. Tipik bir paralel çok işçili montaj hattı dengeleme probleminde, iş parçaları her çok işçili istasyonda çevrim zamanı boyunca beklerken, işçiler kendilerine atanan farklı görevleri eşzamanlı olarak gerçekleştirirler. Her işçi kendine atanan görevleri önceden belirlenmiş sıraya göre gerçekleştirmektedir. Herhangi bir görev ancak o göreve ait tüm öncül görevler tamamlandığında başlayabilmektedir. İstasyonlarda olurlu bir görev atamasının önemli bazı özellikleri vardır. Bu özellikler;

- Her bir görev tek bir istasyona atanır.
- Görevler arası öncelik ilişkileri sağlanır.
- Görevler çevrim süresi içerisinde tamamlanır.

Gerçek hayat uygulamalarında, paralel çok işçili montaj hatlarında üretilen ürünün hacimsel büyüklüğü düşünüldüğünde, bir işçi için kendisine atanan bir görevden diğer bir göreve geçişinde kullanılacak donanım değişikliği sebebi ile görevler arası sıra bağımlı hazırlık zamanları bulunmaktadır (Scholl ve Boysen, 2008). Aynı istasyon içerisinde herhangi bir görevi başka bir görevden hemen önce gerçekleştirmek sonrasında yapılacak görevi etkileyebilir. Sonradan yapılacak görev için bir hazırlık zamanı gerekebilir. Eğer aynı istasyonda aynı işçiye bir görev diğer bir görevden sonra atandıysa ve hazırlık zamanı gerekli ise, hazırlık zamanı o görevin başlama zamanına eklenmelidir. Ayrıca, bir görev atandığı istasyondaki işçinin son görevi ise, o istasyonda o işçiye atanan ilk görev için hazırlık zamanı gerekebilir, çünkü görevler istasyonlarda işçilerde döngüsel olarak gerçekleştirilmektedir. Daha formal bir ifade ile; Pİ-MHDP/SB' inde herhangi istasyon j 'de işçi k ' ya atanan görevler sırasıyla i , p ve h olsun. Görev p görev i 'den hemen sonra gerçekleştiriliyor ise görev p 'ye başlamak için bir hazırlık gerekebilir. Bu hazırlık zamanına *ileri hazırlık zamanı* denir ve τ_{ip} olarak gösterilir. Benzer ileri zaman ilişkisi verilen örnekteki görev p ve görev h arasında da bulunmaktadır. Ek olarak, eğer görev h istasyondaki işçi k 'ya atanan son görev ise bir diğer çevrimde istasyondaki ilk görev olan

görev i 'ye başlanabilmesi için bir hazırlık zamanına ihtiyaç vardır. Bu hazırlık zamanı *geri hazırlık zamanı* olarak adlandırılır ve μ_{hi} ile gösterilir. Benzer ileri hazırlık zamanı ve geri hazırlık zamanı ilişkileri istasyonda bulunan tüm işçiler için geçerlidir. Şekil 4.1. bir istasyondaki işçilere atanan görevler arasında ardışık çevrimlerde ileri ve geri hazırlık zamanlarını göstermektedir (Scholl ve diğerleri, 2013). Örnekten de anlaşılacağı gibi hazırlıkla ilişkili ve atıl olarak değerlendirilmesi gereken zamanlar;

- Ardışık iki görev arasında,
- İlk görevden önce,
- Son görevden sonra olarak görülebilir.



Şekil 4.1. İleri ve geri hazırlık zamanları

Bu tez çalışması kapsamında yukarıda bahsedilen sıra bağımlı ileri ve geri hazırlık zamanlarını dikkate alan paralel çok işçili montaj hattı dengeleme problemi çalışılmaktadır. Dengelemenin amacı ise, belirli bir çevrim zamanı için birincil amacın toplam işçi sayısını, ikincil amacın toplam istasyon sayısının en küçüklenmesidir. Bu amaca hizmet eden çözüm yöntemleri sunulmuştur. Probleme ait varsayımlar aşağıda belirtilmiştir.

Varsayımlar

- Görev zamanları ve görevler arası sıra bağımlı hazırlık zamanları önceden biliniyor ve atandıkları istasyondan bağımsızdır.
- Görevler arası öncelik ilişkileri bilinmektedir.
- Her bir görev işçilerden sadece birisine atanabilir.
- Hatta ait istasyonlarda birden fazla işçi bulunabilir ve istasyonlarda eşit sayıda işçi bulunma zorunluluğu yoktur.
- Herhangi bir işçi birim zamanda tek bir görev yapabilir.
- Hatta bir ürüne ait tek bir model üretilmektedir.
- İşçiler kendilerine atanan görevleri verilen sabit çevrim zamanı içerisinde istasyonlarda eşzamanlı olarak gerçekleştirir.
- Hattaki tüm istasyonlar eşit donanımına sahiptir ve herhangi bir görev herhangi bir istasyona atanabilir.
- Herhangi bir istasyonda bulunabilecek en büyük işçi sayısı bilinmektedir.
- İşçilerin yürüme zamanları göz ardı edilmiştir.
- Hatta paralel görevlere ve paralel istasyonlara izin verilmez.
- Yarı mamul envanterine izin verilmez.
- Pİ-MHDP/SB için amaç, sunulan kısıtlar altında ihtiyaç duyulan işçi sayısının birincil ve istasyon sayısının ikincil olarak en küçüklenmesidir.

5. ÖNERİLEN MATEMATİKSEL PROGRAMLAMA MODELİ (ÖMAT-1)

Bu bölümde önerilen karma tam sayılı programlama ile modellenmiş matematiksel model ÖMAT-1 anlatılmıştır. ÖMAT-1 için Fattahi ve Roshani (2011) tarafından paralel çok işçili montaj hattı dengeleme problemleri için geliştirilen karma tam sayılı programlama modeli, Özcan ve Toklu (2010) tarafından çift taraflı montaj hattı dengeleme problemleri için sıra bağımlı hazırlık zamanları göz önünde bulundurularak oluşturulan matematiksel model ve Scholl ve diğerleri (2013) tarafından SB-BMHDP-1 için önerilen modeller temel alınmıştır. Pİ-MHDP/SB için ÖMAT-1 karma tam sayılı matematiksel formülasyonun detayları aşağıda verilmiştir.

Parametreler ve setler

nt	Toplam görev sayısı
$M_{\max}$	Bir istasyonda çalışabilecek en fazla işçi sayısı
CT	Çevrim zamanı.
t_i	i görevine ait görev zamanı
τ_{ih}	i görevi ile h görevi arasındaki ileri sıra bağımlı hazırlık zamanı matrisi
μ_{hi}	h görevi ile i görevi arasındaki geri sıra bağımlı hazırlık zamanı matrisi
I	Görev Seti; $I = \{1, 2, \dots, i, \dots, nt\}$
J	İstasyon seti; $J = \{1, 2, \dots, j, \dots, nt\}$
K	İşçi seti; $K = \{1, 2, \dots, k, \dots, M_{\max}\}$
S	Pozisyon seti; $S = \{1, 2, \dots, s, \dots, nt\}$
$P(i)$	i görevinin hemen (komşu) öncül seti
$P_a(i)$	i görevinin tüm öncül görevleri seti
$SU(i)$	i görevinin hemen (komşu) ardıl seti
$SU_a(i)$	i görevinin tüm ardıl görevleri seti
φ	Oldukça büyük bir sayı

Karar değişkenleri

$$x_{ijks} = \begin{cases} 1, & \text{eğer } i \text{ görevi } j \text{ istasyonundaki } k \text{ işçisinin } s \text{ pozisyonuna atandı ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$st_i = i$ görevinin başlangıç zamanı

$ft_i = i$ görevinin tamamlanma süresi

$$w_{jk} = \begin{cases} 1, & k \text{ işçisi } j \text{ istasyonunda kullanıldı ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$ws_{jk} = \begin{cases} 1, & j \text{ istasyonunda } k \text{ adet işçi kullanıldı ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$y_j = \begin{cases} 1, & j \text{ istasyonu açıldı ise} \\ 0, & \text{diğer durumda} \end{cases}$$

Amaç fonksiyonu

$$\text{Min} \sum_{j \in J} \sum_{k \in K} w_{jk} + (1/(nt+1)) \sum_{j \in J} y_j \quad (5.1)$$

Kısıtlar

$$\sum_{j \in J} \sum_{k \in K} \sum_{s \in S} x_{ijks} = 1, \forall i \in I \quad (5.2)$$

$$\sum_{i \in I} x_{ijks} \leq 1, j \in J, k \in K, s \in S \quad (5.3)$$

$$\sum_{i \in I} x_{ijks+1} - \sum_{i \in I} x_{ijks} \leq 0, j \in J, k \in K, s \in S \wedge s < nt \quad (5.4)$$

$$\sum_{j \in J} \sum_{k \in K} \sum_{s \in S} jx_{hjks} \leq \sum_{j \in J} \sum_{k \in K} \sum_{s \in S} jx_{ijks}, \forall i, h \in I, h \in P(i) \quad (5.5)$$

$$st_i + t_i \leq CT, \forall i \in I \quad (5.6)$$

$$st_i + t_i = ft_i, \forall i \in I \quad (5.7)$$

$$st_i - st_h + \varphi\left(1 - \sum_{k \in K} \sum_{s \in S} x_{h j k s}\right) + \varphi\left(1 - \sum_{k \in K} \sum_{s \in S} x_{i j k s}\right) \geq t_h, \forall i, h \in I, h \in P(i), j \in J \quad (5.8)$$

$$st_h - st_i + \varphi(1 - x_{i j k s}) + \varphi(1 - x_{h j k s+1}) \geq t_i + \tau_{ih}, \forall i, h \in I \wedge i \neq h, j \in J, k \in K, s \in S \wedge s < nt \quad (5.9)$$

$$st_i + (CT - ft_h) + \varphi(1 - x_{i j k 1}) + \varphi(1 - x_{h j k s-1}) + \varphi\left(\sum_{r \in I} x_{r j k s}\right) \geq \mu_{hi}, \\ \forall i, h \in I \wedge i \neq h, j \in J, k \in K, s \in S \wedge s > 1 \quad (5.10)$$

$$\sum_{k \in K} \sum_{s \in S} x_{i j k s} - nt \cdot w_{jk} \leq 0, \forall i \in I, j \in J \quad (5.11)$$

$$\sum_{k \in K} k \cdot ws_{jk} - \sum_{k \in K} w_{jk} = 0, \forall j \in J \quad (5.12)$$

$$w_{j k+1} \leq w_{jk}, \forall j \in J, k \in K \wedge k < M_{\max} \quad (5.13)$$

$$\sum_{i \in I} \sum_{k \in K} \sum_{s \in S} x_{i j+1 k s} \leq nt \sum_{i \in I} \sum_{k \in K} \sum_{s \in S} x_{i j k s}, \forall j \in J \wedge j < nt \quad (5.14)$$

$$\sum_k w_{jk} \leq M_{\max} y_j, \forall j \in J \quad (5.15)$$

$$\sum_k ws_{jk} \leq 1, \forall j \in J \quad (5.16)$$

$$x_{i j k s} \in \{0, 1\}, \forall i \in I, j \in J, k \in K, s \in S \quad (5.17)$$

$$y_j \in \{0, 1\}, \forall j \in J \quad (5.18)$$

$$ws_{jk} \in \{0, 1\}, \forall j \in J, k \in K \quad (5.19)$$

$$w_{jk} \in \{0, 1\}, \forall j \in J, k \in K \quad (5.20)$$

$$st_i \geq 0, \forall i \in I, \quad (5.21)$$

$$ft_i \geq 0, \forall i \in I, \quad (5.22)$$

Amaç fonksiyonu Eş. 5.1 birincil amaç olarak istasyonlarda bulunan toplam işçi sayısı ve ikincil amaç olarak açılan istasyon sayısını en küçüklemektedir. Eş 5.2 atama kısıt grubudur. Her bir görevin tek bir istasyonda tek bir işçinin tek bir pozisyonuna atanmasını garantiler. Eş. 5.3, her bir istasyondaki her bir işçinin her bir pozisyonuna en fazla bir görev atamaktadır. Eş. 5.4 her bir istasyondaki işçilerin pozisyonlarına atanan görevlerin

atanma sıralarını düzenler. İşçiye atanan görevlerin pozisyonları arasında boşluk olmasını önler. Eş. 5.5 bütün görevler arasındaki öncelik ilişkilerinin sağlanmasını garantiler. Eş. 5.6 çevrim zamanı kısıt kümesidir. Bir görevin başlangıç zamanı ile o görevin görev zamanının toplamı çevrim süresini aşamaz. Eş. 5.7 aralarında öncelik ilişkisi bulunan görev çiftleri için başlangıç zamanlarını vermektedir. h görevi i görevinin hemen (komşu) öncülü ise, ve aynı istasyona atandıysa, i görevinin başlangıç zamanı en az h görevinin başlangıç zamanı ile h görevinin görev zamanı toplamı kadar olmalıdır. Eş. 5.8 tüm i ve h görevler çiftleri için ileri hazırlık zamanını göz önünde bulundurarak başlangıç zamanı hesaplamaktadır. Eğer bu görevler aynı istasyonda aynı işçiye atandılar ve i görevi h görevinden hemen önce atandı ise, h görevinin başlangıç zamanı, en az i görevinin başlangıç zamanı, görev zamanı ve görevler arası ileri sıra bağımlı hazırlık zamanı toplamı kadar olmalıdır. Eş. 5.9 tüm i ve h görevleri için geri hazırlık zamanlarını dikkate alarak başlangıç zamanı hesaplamaktadır. i görevi istasyona atanan ilk görev, h görevi istasyona atanan son görev, burada aynı istasyonda aynı işçide ardışık çevrimlerde bu iki görev arasında oluşabilecek geri sıra bağımlı hazırlık zamanını görevlerin başlangıç zamanına eklenmesi görülmektedir. Eş. 5.10, k işçisi j istasyonunda kullanıldıysa w_{jk} 1 değerini alacaktır. Eş. 5.11 istasyonlara atanan işçi sayılarını doğrulamaktadır. Eğer j istasyonunda k tane işçi kullanıldıysa ws_{jk} 1 değerini alır. Eş. 5.12 işçilerin kullanım sıralarını düzenlemektedir. Eş. 5.13 istasyonlarının kullanım sıralarını düzenler. Eş. 5.14 eğer k işçisi j istasyonuna atandı ise j istasyonu açılmıştır. Eş. 5.15'e göre bir istasyonda kullanılan işçi sayısı o istasyona atanabilecek en büyük işçi sayısından fazla olamaz. Eş. 5.16'ya göre bir istasyonda en fazla k adet işçi kullanılabilir. Eş. 5.17 -5.22 karar değişkenleri için işaret kısıt kümeleridir.

5.1. Hesaplama Sonuçları ve Değerlendirme

Bu tez kapsamında ele alınan Pİ-MHDP/SB için literatürde henüz yayınlanmış karşılaştırma problemleri bulunmamaktadır. Karşılaştırma problemlerinin üretilmesi için literatürde Scholl ve diğerleri (2013) tarafından sıra bağımlı hazırlık zamanlarını göz önünde bulunduran basit montaj hattı dengeleme tip-1 problemi (SB-BMHDP-1) için sunulan SBF veri setlerinden yararlanılmıştır. SBF veri setleri altında 4 farklı sıra bağımlı hazırlık zamanı seviyeleri (α) için dört grup veri seti bulunmaktadır. Bu dört farklı grupta aynı olan problem parametreleri ise görev sayıları, görevlerin operasyon zamanları, çevrim

zamanı ve görevlerin öncelik ilişkileridir. Görevler arasında bulunan sıra bağımlı hazırlık zamanları hazırlık zamanı seviyesine bağlı olarak değişkenlik göstermektedir. Bu bağlamda toplam 236 adet problem verisi üretilmiştir. Hesaplamalarda kullanılan problemler Çizelge 5.1’de verilmektedir. Çizelge 5.1.’deki ilk kolon kullanılan veri setinin literatürdeki adını göstermektedir. İkinci kolonda veri setinde kaç adet görev bulunduğu görülmektedir. Diğer kolonlar sırasıyla, veri setindeki en küçük görev zamanını (t_{enk}), en büyük görev zamanını (t_{enb}) ve toplam görev zamanını (t_{toplam}) ifade etmektedir. Çizelgedeki son iki kolon veri setine ait çevrim zamanı ile o veri setine ait toplam kaç adet problem olduğunu ifade etmektedir. Bu veri setlerine <http://www.assembly-line-balancing.de/> adresinden ulaşılabilir. Karşılaştırma için kullanılacak problem veri setleri üç seviyeye ayrılmıştır. Bunlar küçük, orta ve büyük boyutlu problemlerin oluşturduğu setlerdir. Küçük boyutlu problemlerin bulunduğu set, literatürde bilen adları ile yazmak gerekirse; Bowman, Jackson, Jaeschke, ve Mertens problemlerinden oluşmaktadır. Orta boyutlu problemler ise; Buxey, Heskia, Gunther, Lutz1 ve Sawyer’ dir. Arcus, Bartholdi ve Mukherje problemleri ise büyük boyutlu problem setini oluşturmaktadır.

Çizelge 5.1. Test problemleri

Problem	nt	t_{enk}	t_{enb}	t_{toplam}	Literatürde kullanılan çevrim zamanları (CT)	Toplam Problem Sayısı
Bowman	8	3	17	75	20	4
Jackson	11	1	7	46	7, 9, 10, 13, 14, 21	24
Jaeschke	9	1	6	37	6, 7, 8, 10, 18	20
Mertens	7	1	6	29	6, 7, 8, 10, 15, 18	24
Heskia	28	1	108	1024	138, 205, 324	12
Gunther	35	1	40	483	41,44,49,54,61,69,81	28
Buxey	29	1	25	324	27,30,33,36,41,47,54	28
Lutz1	32	100	1400	14140	1414,1572,	8
Sawyer	30	1	25	324	25, 27, 30, 33, 36, 41, 47, 57, 75	40
Arcus1	83	233	3691	75707	3786,3985, 4206	12
Arcus2	111	10	5689	150399	5755, 5785, 6016	12
Bartholdi	148	3	383	5634	84, 85, 87	12
Mukherje	94	8	171	4208	176,192,211	12

Karşılaştırma işlemlerinde Intel Core i5 2.53 GHz 8 GB dahili bellek bulunan kişisel bilgisayar kullanılmıştır. Pİ-MHDP/SB için önerilen karma tam sayılı programlama modeli olan ÖMAT-1 için GAMS 24.1 yazılımı aracılığı ile Cplex çözücüsü kullanılmış ve her bir

veri seti için 3600 CPU saniyesi çalıştırılmıştır. Elde edilen sonuçlar Çizelge-5.2.'de sunulmuştur.

Çizelge 5.2. ÖMAT-1 ile 3600 CPU saniyesi kısıdı ile çalıştırılan problem veri setlerinden elde edilen sonuçları özetlemektedir. ÖMAT-1 ile 236 problem veri setinden sadece 56 adet problem veri setine ait en iyi sonuçlar elde edilmiştir. Geri kalan veri setlerinden yalnızca 9 adet veri setinde verilen CPU saniye kısıdı içerisinde uygun çözüm elde edilmiştir. Toplam problem veri setlerine oranı (%73) hayli büyük olan 173 adet veri setinde belirlenen CPU saniye kısıdı altında hiçbir çözüme ulaşılamamıştır. Çizelgeye bakıldığında problem büyüklüğü arttıkça ÖMAT-1 performansının düştüğü görülmektedir. Elde edilen en iyi çözümlerin küçük boyutlu problemler olduğu gözlemlenmiştir. Orta boyutlu ve büyük boyutlu olarak adlandırılacak problem veri setlerinde istenilen performans çalışılan problemin NP-Zor yapısı sebebiyle elde edilememiştir.

Çizelge 5.2. Matematiksel programlama modeli sonuçları/ÖMAT-1

Problem	<i>nt</i>	Optimali Bulunan	Uygun Çözüm Bulunan	Uygun Çözüm Bulunamayan	Toplam
Bowman	8	4			4
Jackson	11	8	9	7	24
Jaeschke	9	20			20
Mertens	7	24			24
Heskia	28			12	12
Gunther	35			28	28
Buxey	29			28	28
Lutz1	32			8	8
Sawyer	30			40	40
Arcus1	83			12	12
Arcus2	111			12	12
Bartholdi	148			12	12
Mukherje	94			12	12
		56	9	173	236

Farklı hazırlık zamanı yoğunlukları için ÖMAT-1 detaylı sonuçları sırasıyla Çizelge-5.3.'te sunulmuştur. Çizelge incelendiğinde sıra bağımlı hazırlık zamanları yoğunlukları artarken ÖMAT-1'e ait CPU zamanının arttığı gözlemlenmektedir. Bowman, Jaeschke ve Mertens problemleri için her ne kadar CPU zamanı artsa da modeli durdurma koşulu olarak verilen CPU zamanı içerisinde en iyi sonuçlara ulaşılmıştır. Ancak, Jackson problem seti için hazırlık zamanı yoğunluğu artarken bulunan en iyi çözüm sayısı, uygun

çözüm sayısı ve çözüm bulunamayan problem seti sayısı değişkenlik göstermektedir. Çizelge 5.3.'ten görüldüğü üzere, ÖMAT-1 $\alpha=0.25$ yoğunlukta 4 adet problemin en iyi sonucuna ulaşmışken $\alpha=0.50 - 0.75$ yoğunluklarında sadece 2 adet problem seti en iyi çözüme ulaşmıştır. $\alpha=1.00$ yoğunluğunda ise söz konusu problem seti için en iyi çözüm belirlenen süre içerisinde elde edilememiştir.

Çizelge 5.2. ve Çizelge 5.3. ÖMAT-1'in orta boyutlu ve büyük boyutlu problem setlerinde kabul edilebilir bir CPU zamanı içerisinde herhangi bir çözüme ulaşmadığını göstermiştir. Bu sebeple, Pİ-MHDP/SB problemi için tez çalışmasının bir sonraki bölümü olan 6. bölümde iki farklı yaklaşımla (klasik ve aralık yaklaşımı) kısıt programlama modeli sunulmuştur. Kısıt programlama ile ÖMAT-1'in dezavantajlarının bertaraf edilmesi amaçlanmıştır.

Çizelge 5.3. Farklı hazırlık zamanı yoğunlukları için ÖMAT-1 sonuçları

α	Problem	Optimali Bulunan	Uygun Çözüm Bulunan	Uygun Çözüm Bulunamayan	CPU (saniye)
0.25	Bowman	1			8,733
	Jackson	4	1	1	2772
	Jaeschke	5			83,24
	Mertens	6			32,,52
0.50	Bowman	1			14,73
	Jackson	2	2	2	3150,15
	Jaeschke	5			149,52
	Mertens	6			537,81
0.75	Bowman	1			19,25
	Jackson	2	2	2	3452,87
	Jaeschke	5			268,35
	Mertens	6			781,45
1.00	Bowman	1			24,46
	Jackson	0	4	2	3600,2
	Jaeschke	5			461,82
	Mertens	6			927,54

6. MONTAJ HATTI Dengeleme Problemleri İin nerilen Kısıt Programlama Modelleri

Bu blmde, sıra baėımlı hazırlık zamanlarını dikkate alan paralel ok iřili montaj hattı dengeleme problemleri iin (PI-MHDP/SB) hem klasik yntem hem de aralık (interval) yaklařımı ile kısıt programlama modelleri (KP1 ve KP2) geliřtirilmiřtir. Blm, kısıt programlama hakkında genel bilgi ile bařlamaktadır. Ardından ilgili literatr taraması verilmiřtir. nerilen kısıt programlama modelleri detaylı bir řekilde anlatılmıřtır. Yapılan deneysel alıřmaların sonuları raporlanmıřtır.

6.1. Kısıt Programlama

Bu tez alıřmasında NP-Zor sınıfında olan PMHDP/SB problemi iin kısıt programlama yaklařımını kullanarak bir zm sunulmaktadır. Kısıt programlama doėrusal programlamanın optimal zme ulařmasını bilgisayar programlamanın zelliklerini birleřtirerek, matematiksel programlamaya alternatif bir modelleme ve zm metodudur. Kısıt programlama lineer cebirin kullanıldıėı matematiksel modelin aksine graf teori ve mantık programlama gibi bilgisayar bilimi temellerine dayanmaktadır. Birok karmařık izelgeleme ve sıralama problemlerine zm aramak iin olduka deėerlidir. Bu tr problemlerdeki karar deėiřkenleri, aktiviteler ve kaynaklar arasındaki aritmetik iliřkiler sz konusu ise kısıt programlamaya ihtiya vardır. Dal-sınır ve dinamik programlama gibi diėer kombinatoryal zm metotları arasında yer alan tam bir zm tekniėidir (Talbi, 2009: 18-19). Kısıtlar kısıt programlamada temel kavramdır. Deėiřkenler, deėiřkenlerin alabileceėi olası deėerler seti (domain (zm kmesi)), ve kısıtlar bir KP modelini oluřturur. Karar deėiřkenlerine atanabilecek muhtemel deėerlere zm kmesi denir. Kısıtlar birtakım deėiřkenler ve bu deėiřkenlere ait zm kmesinin iliřkisi olarak ifade edilir. Kısıtlar doėrusal, doėrusal olmayan, mantıksal ve global kısıtlar olabilir.

Genel olarak kısıt programlama algoritması řu řekilde iřler. Deėiřkenler, deėiřkenlerin zm kmeleri ve kısıt setleri tanımlanır. Daha sonra kısıt tretme ve zm kmesi zel filtreleme algoritmaları kullanılarak daraltılır. Bunun sonucunda ilgili probleme zg zm bulunmuř olabilir. Bulunmadı ise tutarsızlık kontrol yapılır. Tutarsızlık, deėiřkenlerden herhangi birinin zm kmesinde eleman olmaması olarak tanımlanır.

Tutarsızlık kontrolü başarısız olabilir ancak arama stratejileri kullanılarak çözüm oluşturulabilir bu çözüme dal denir. Bir diğer ifade ile ana probleme geçici kısıtlar eklenir ve yeni alt problemler oluşturulur. Bu dal oluşturma olarak adlandırılır. Bu işlemin sonucunda bu çözüm ile kısıtlar tekrar oluşturulur. Yukarıdaki işlemler tekrarlanır. Oluşturulan tüm dallar budanmış ise algoritma sonlanır, diğer durumda geriye dönülür ve yeni bir çözüm üretilir.

Probleme etkin bir çözüm aranıyorsa KP modelinin çok iyi ifade edilmesi gerekmektedir (Rossi, Beek ve Walsh, 2006). Bu sebeple KP modeli özenle oluşturulmalıdır ve iyi bir KP çözücüsü seçilmelidir.

KP ikiye ayrılır. Bunlardan birincisi modelde amaç fonksiyonunun olmadığı kısıt sağlama problemleri (KSP), ikincisi ise amaç fonksiyonu bulunan kısıt optimizasyonu problemleridir (KOP). Kısıt sağlama problemleri bir amaca bağlı olmadan problemin bütün uygun çözümlerinin bulunmasını sağlayan yöntemdir. Bu çözümler tüm problemdeki kısıtları sağlamışsa çözüm tutarlıdır, eğer bir veya bir kaç kısıt sağlanmadıysa çözüm tutarsızdır. Tüm değişkenlerin kendi çözüm kümelerinden bir değer aldığı çözümler tam çözümler, bazı değişkenlerin değer alamadığı çözümler ise parçalı çözümler olarak adlandırılır (Solnon, 2010: 27-50).

Kısıt programlama ile optimum çözümün bulunması amaçlanan problemlerde verilen bir amaç fonksiyonuna göre her bir çözüm değerlendirilir ve bu yolla verilen amaç fonksiyonu tipine uygun şekilde maksimum veya minimum amaç fonksiyonu değerine sahip çözüm veya çözümler bulunur. Bu problem tarzı kısıt optimizasyonu problemi olarak tanımlanır. Kısıt optimizasyon problemleri, kısıt sağlama problemlerine amaç fonksiyonu eklenerek geliştirilmiş bir problem çözme yöntemidir. Bu çalışma kapsamında tanımlanan montaj hattı dengeleme problemi için kısıt optimizasyonu modelleri kurulacaktır.

6.2. Karma Tamsayılı Programlama ve Kısıt Programlama Karşılaştırması

Kısıt programlama (KP) ile karma tam sayılı programlama (KTP) kombinatoryal optimizasyon problemlerine çözüm aramak için farklı yaklaşımlar kullanan iki metottur. Puget ve Lustig (2001)'in çalışmalarında KP ve KTP' nin üç açıdan karşılaştırıldığı görülmektedir. Bunlar, modelleme yeteneği, düğüm işleme ve arama stratejisidir. Her

ikisinde de modellemenin elemanları: karar değişkenleri, kısıtlar, amaç fonksiyonu ve arama ağacıdır. Ancak arama stratejileri ne kadar benzer olsa da, düğüm işleme iki metot için de farklılık göstermektedir. Lustig ve Puget (2013)' in çalışmalarında bu iki kavram arasında iki temel farklılıktan bahsedilmiştir. Birincisi, arama uzayı KTP de karar değişkenlerinin kesirli olmayan değerleri ile bölünürken, KP ise arama uzayını herhangi bir kısıt setini kullanarak ürettiği noktalar ile bölmektedir. KTP alt optimal çözümleri elemek için, mevcut bulunan her bir düğüme bir alt sınır hesaplamaktadır. KP da ise elemeye uygun olmayan çözümlerin elenmesi olarak yapılır. Bir sonraki farklılık ise, dallanma değişken seçimindedir. KP de her bir problem için dallanma stratejileri belirlenebilir. Ancak KTP de geleneksel dal sınır yöntemini kullanır.

KP, KTP' ye göre uygulama dilinde avantajlara sahiptir. KP' de birçok kısıt tipi; özel programlanmış kısıtlar, mantıksal kısıtlar ve doğrusal olmayan kısıtlar kolaylıkla ifade edilebilirken, KTP sadece doğrusallaştırılmış kısıtlar veya karesel konveks kısıtları kullanmaktadır. KP' nin en temel dezavantajları hafıza kullanımı ve değişken tanımlama olarak söylenebilir. Çünkü KP' de her bir değişken için ayrı çözüm kümesi tanımlanmaktadır. Bu durum, daha fazla hafızaya ihtiyaç duyar. Ek olarak, KP birçok kombinatorial optimizasyon problemlerinde KTP ye alternatif olarak kullanılabilir (ILOG, 2013a)

Kısıt programlama bahsedildiği gibi birçok kombinatorial NP-Zor problemin çözümünde yaygın olarak kullanılmaktadır. Az hesaplama zamanı ile tutarlı analizler yapılabilir. Özellikle çizelgeleme problemlerinde sıklıkla kullanıldığını görülmektedir. Bu çalışmalardan bazıları Zeballos (2010), Terekhov, Doğru, Özen ve Beck (2012) olarak verilebilir. Çizelgelemeden farklı olarak tedarik zinciri konfigürasyonu için Li ve Womer (2012) matematiksel programlama ile KP' nin birleşiminden oluşan bir algoritma önermişlerdir. Siala, Hebrard ve Huguet (2015) çalışmalarında araç rotalama problemi için önerilen KP içeren bir budama sezgiseli bulunmaktadır.

Üretimin değişik alanlarında yaygın bir şekilde kullanıldığı görülmesine rağmen MHDP' de uygulaması oldukça azdır. İlk olarak Bockmayr ve Pissaruk (2001) MHDP' de KP' yi incelemiş ve tam sayılı programlama (TP) ile KP'nin birleşiminden oluşan dal kesme yöntemine dayalı hibrit bir çözüm yöntemi önerilmiştir. Sayısal deneyler sonucunda

MHDP' nin çözümünde KP' nin alternatif bir modelleme ve çözüm yöntemi olduğunu göstermişlerdir.

Montaj operasyonlarının seçilmesi ve sıralanması problemi için Valle, Marquez, Gasca ve Toro (2003) toplam montaj zamanını en küçükleyen bir KP modeli önermişlerdir.

KP modeli ile iki farklı TP modelini hesaplama açısından kıyaslayan bir çalışmayı Pastor, Ferrer ve Garcia (2007) MHDP için yaptıkları görülmektedir. Hem tip-1 hem de tip-2 problemleri için modeller geliştirilmiştir. KP' nin özellikle büyük boyutlu problemlerde KTP'den daha iyi ve hızlı olduğunu göstermişlerdir.

Topaloğlu, Salum ve Supçiller (2012) MHDP çözümü için kural tabanlı bir KP modeli önermişlerdir. KP ve TP modelleri için modelleme kapasitesi, çözüm kalitesi ve CPU zamanı açısından karşılaştırmalı sonuçlar sunmuşlardır. KP' nin çözümde TP'den daha etkin olduğunu göstermişlerdir.

6.3. Önerilen Kısıt Programlama Modelleri

Sıra bağımlı hazırlık zamanlarını dikkate alan paralel çok işçili montaj hattı dengeleme problemi için hem klasik yöntem hem de aralık (interval) yaklaşımı ile geliştirilmiş kısıt programlama modelleri (KP1 ve KP2) sunulmuştur.

6.3.1. Klasik yaklaşım ile KP modeli (KP1)

Klasik yaklaşımla bir diğer ifade ile tam sayılı karar değişkenlerinin kullanımı yoluyla PİMHP/SB' nin kısıt programlama modeli oluşturulmuştur. Modelin bileşenleri aşağıda sunulmaktadır.

Sabitler

Model için gerekli olan sabit tanımları ÖMAT-1 için kullanılan tanımlamalar ile birebir aynı kullanılmıştır.

Karar değişkenleri (X seti)

$Sıra_i$: Göreli görev sırası

S_i : i görevinin atandığı istasyon indeksi

w_i : i görevinin atandığı S_i istasyonunda bulunan ve istasyon içerisinde atandığı işçi indeksi

$start_i$: Göreli başlama zamanı

tw_j : j istasyonunda kullanılan toplam işçi sayısı

$start_i$ değişkeni Kellegöz (2016) tarafından önerilen ve aynı problemin hazırlık zamanı içermeyen versiyonu için yeni bir tam sayılı programlama modelinin oluşturulması amacıyla kullanılan yaklaşım temel alınarak tasarlanmıştır. Bu yaklaşıma göre istasyon 1'in işçi 1'inin başlama zamanı sıfır, işçi 2'sinin başlama zamanı CT , işçi k 'nin başlama zamanı $(k-1)CT$ şeklindedir. İstasyon 2'in işçi 1'i için bu zaman $M_{\max}CT$ 'den, işçi 2'si için $M_{\max}CT + CT$ şeklinde devam eder. Genel olarak j . istasyonun k . işçisi için göreli zaman $((j-1)M_{\max} + (k-1)CT)$ 'den başlar. Spiral şeklinde olan göreli zaman ekseninin her hangi bir noktasında sadece tek bir görev yer alabilir. Bu tez çalışmasında dikkate alınan problemde hazırlık zamanları da yer aldığından, hazırlık zamanlarının modellenmesi amacıyla Kellegöz (2016) tarafından önerilen göreli başlama zamanı yaklaşımından esinlenilerek göreli görev sırası $Sıra_i$ değişkeni tanımlanmıştır. Bu değişken görev i 'nin göreli zaman eksenini üzerindeki sırasını göstermektedir.

Alanlar-tanım kümeleri (D seti)

Tamsayı $Sıra_i \in \{0, \dots, nt-1\}$

Tamsayı $S_i \in \{0, \dots, nt\}$

Tamsayı $w_i \in \{0, \dots, M_{\max}\}$

Tamsayı $start_i \in \{0, \dots, nt \times M_{\max} \times CT - t_i\}$

Tamsayı $tw_i \in \{0, \dots, M_{\max}\}$

Kısıtlamalar (C seti)

KP modelinde yer alan kısıtlar aşağıdaki gibidir.

$$allDifferent(Sıra) \tag{6.1}$$

$$\begin{aligned} start_f &\geq (s_f M_{\max} + w_f)CT + \mu_{lf}, \\ \forall i, h, f, l \in I \wedge (s_f M_{\max} + w_f = s_l M_{\max} + w_l) \wedge \\ \{ (sira_f == 0) \vee (sira_f == sira_i + 1) \wedge (s_f M_{\max} + w_f \neq s_i M_{\max} + w_i) \} \wedge \\ \{ (sira_l == n - 1) \vee (sira_l == sira_j - 1) \wedge (s_l M_{\max} + w_l \neq s_h M_{\max} + w_h) \} \end{aligned} \tag{6.2}$$

$$tw_j \geq w_i + 1, \forall i \in I \wedge \forall j \in J \wedge s_i = j \tag{6.3}$$

$$start_h \geq start_i + t_i, \forall i, h \in I \wedge sira_h = sira_i + 1 \tag{6.4}$$

$$start_h \geq start_i + t_i + \tau_{ih}, \forall i, h \in I \wedge sira_h = sira_i + 1 \wedge s_i = s_h \wedge w_i = w_h \tag{6.5}$$

$$(s_i M_{\max} + w_i)CT \leq start_i, \forall i \in I \tag{6.6}$$

$$(s_i M_{\max} + w_i + 1)CT \geq start_i + t_i, \forall i \in I \tag{6.7}$$

$$\begin{aligned} start_h - (s_h (M_{\max} - 1) + w_h)CT &\leq start_i + t_i - (s_i (M_{\max} - 1) + w_i)CT \\ \forall i, h \in I \wedge i \in P(h) \end{aligned} \tag{6.8}$$

Sıkılaştırıcı Kısıtlar

$$s_h \leq s_i + 1, \forall i, h \in I \wedge (sira_h = sira_i + 1) \tag{6.9}$$

$$w_h \leq w_i + 1, \forall i, h \in I \wedge (sira_h = sira_i + 1) \tag{6.10}$$

$$\sum_{k \in K} tw_k \leq k \tag{6.11}$$

Eş. (6.1), sıra değişkenlerinin birbirinden farklı değerler almasını sağlar. Diğer bir ifadeyle, sıra değişken değerlerinin kendi içerisinde permütasyon oluşturacağını ifade eder. allDifferent ifadesi KP kapsamında kullanılan Global kısıtlar arasında yer alır. Global kısıtlar değişkenlerin bir temele göre sıralanması amacıyla kullanılan kısıtlardır (Rossi, Beek ve Walsh, 2006). Eş. (6.2) geriye doğru hazırlık zamanının etkin olacağı durumları ifade etmektedir. Görev f ve görev l arasında geriye doğru hazırlık zamanının aktif olması yani görev f 'in görelî başlama zamanının geriye doğru hazırlık zamanı kadar ötelenmesi

$(start_f \geq (s_f M_{\max} + w_f)CT + \mu_f)$ için: Görev f ve görev l aynı istasyonun aynı işçisine atanmış olmalı $(s_f M_{\max} + w_f = s_l M_{\max} + w_l)$. Görev f atandığı işçinin ilk görevi olmalı (birinci istasyonun ilk işçisine ilk sırada atanmış olması $(sıra_f = 0)$ veya f'den bir öndeki görelî pozisyona yerleşmiş görev i'nin $(sıra_f = sıra_i + 1)$ farklı işçide yer alması $(s_f M_{\max} + w_f \neq s_l M_{\max} + w_l)$). Görev l'nin atandığı işçinin son görevi olması (aktif olan son istasyonun son işçisine son sırada atanmış olması $(sıra_l = nt - 1)$ veya l'den bir sonraki görelî pozisyona yerleşmiş h görevinin $(sıra_l = sıra_h - 1)$ farklı işçide yer alması $(s_l M_{\max} + w_l \neq s_h M_{\max} + w_h)$) koşullarının tamamının aynı anda sağlanması gerekir. Eş. (6.3) amaç fonksiyonunda kullanılan tw_k değişkenlerine uygun değerlerin atanmasını sağlar. Bir i görevi k istasyonuna atanmışsa $(s_i = j)$, j istasyonunun içerdiği işçisi sayısı i görevinin atandığı işçi indisinin bir büyüğünden küçük olamaz $(tw_k \geq w_i + 1)$. Bu kısıt kapsamında dikkat edilecek durum, daha önce de ifade edildiği gibi, işçi indisleri sıfırdan başlamakta olup, indisi k olan işçi istasyondaki $(k+1)$. işçiye karşılık gelmektedir. Eş. (6.4), görelî pozisyon için birbirini takip eden görevlerin başlama zamanlarını ayarlamaktadır. Eş. (6.5) ise ileriye doğru hazırlık zamanının uygulanması amacıyla oluşturulmuştur. Görev j, görelî sıralamada görev i'den hemen sonra geliyorsa $(sıra_h = sıra_i + 1)$, bu iki görev aynı istasyona $(s_i = s_h)$ ve aynı işçiye $(w_i = w_h)$ atanmışsa, görev j' in başlaması için görev i' in bitişinden sonra en az ileriye doğru hazırlık zamanının geçmesi gerekir $(start_h \geq start_i + t_i + \tau_{ih})$. Eş. (6.6) ve (6.7), görelî başlama zamanı bazında bir görevin istasyon başlama ve bitiş zamanları arasında gerçekleştirilmesi gerektiğini ifade eder. Eş. (6.6)'ya göre görev i' in başlama zamanı atandığı istasyon s_i ve işçi indeksine w_i göre hesaplanan görelî istasyon başlama zamanından küçük olamaz $(s_i M_{\max} + w_i)CT \leq start_i$. Benzer yaklaşımla Eş. (6.7), görevin bitiş zamanının atandığı istasyonun görelî istasyon zamanı bitişinden büyük olmayacağını ifade etmektedir $(s_i M_{\max} + w_i + 1)CT \geq start_i + t_i$. Eş. (6.8), görevler arasındaki öncelik ilişkilerinin sağlanması amacıyla kullanılmaktadır. Eğer görev i, görev h'nin öncülüysen, görev h' in başlayabilmesi için görev i' in tamamlanması gerekir $(start_j - (s_h (M_{\max} - 1 + w_h)CT) \geq start_i + t_i - s_i (M_{\max} - 1 + w_i)CT)$. Daha önce de ifade edildiği gibi $start_i$ değişkenleri görelî başlama zamanlarını tutmaktadır. Teknolojik öncelikler ise Pİ-MHDP/SB' de saat zamanı bazında (clock time) başlama zamanlarına göre sağlanmalıdır. Saat zamanı yaklaşımı Scholl (1999) tarafından

kullanılmış olup ilk istasyon için saat zamanı sıfırdan başlayıp çevrim zamanında biterken ikinci istasyon için çevrim zamanında başlayıp 2 çevrim zamanında biter. Genel olarak j . istasyon için $(j-1)CT$ 'de başlayıp $(j)CT$ 'de biter. Bu kısıt kapsamında i görevinin görelî bitiş ve h görevinin görelî başlama zamanları $(start_j$ ve $start_i + t_i)$ saat bazında zamanlara çevrilmektedir. $(start_i + t_i - (s_i(M_{\max} - 1) + w_i)CT)$ ve $(start_i + t_i - (s_i(M_{\max} - 1) + w_i)CT)$ saat zamanları bazında öncelik ilişkisi ifade edilmektedir. $(start_h - (s_h(M_{\max} - 1) + w_h)CT \leq start_i + t_i - (s_i(M_{\max} - 1) + w_i)CT)$.

Eş. (6.9)-(6.11) sıkılaştırıcı kısıtları ifade etmektedir. Sıkılaştırıcı kısıtlar, problemin tanımı kapsamında gerekli olmamasına rağmen alternatif çözümleri elimine etmek veya uygun çözüme daha hızlı bir şekilde ulaşmak amacıyla yazılmıştır. Eş. (6.9), Görev j , görev i 'den hemen sonraki görelî pozisyona atanmışsa $(sira_h = sıra_i + 1)$, görev h ' in ya aynı istasyona veya bir sonraki istasyona atanması gerektiğini $(s_h \leq s_i + 1)$ ifade eder. Diğer bir deyişle, görev ataması yapılan iki istasyon arasında kullanılmamış istasyon olmaması gerektiğini ifade etmektedir. Eş. (6.10) ise aynı yaklaşımı işçilere uygular, aynı istasyonda bulunan ve görev atanmış iki işçi arasında kullanılmamış işçi olmayacağını ifade eder. Son olarak Eş. (6.11), kullanılacak toplam işçi sayısının görev sayısından büyük olmayacağını ifade etmektedir. Dikkat edilirse, model kapsamında amaç fonksiyonu toplam işçi sayısının en küçüklenmesidir.

Amaç fonksiyonu

$$\min \sum_{k \in K} tw_k + 1 / (nt + 1) \max(s_i) \quad (6.12)$$

Problemdede amaç fonksiyonu ÖMAT-1'in amaç fonksiyonu ile aynıdır. Eş. 6.12 de verilmiştir. Amaç, birincil olarak kullanılan toplam işçi sayısının en küçüklenmesi, ikincil olarak açılan istasyon sayısının en küçüklenmesidir.

6.3.2. Aralık değişkeni yaklaşımı ile KP modeli (KP2)

Bu bölümde IBM ILOG tarafından sunulan değişken tanımları ve kısıtları ile PMHDP/SB ifade edilerek modellenmiştir. İlk olarak bu değişkenler ve fonksiyonlar ile ilgili gerekli

tanımlamalara yer verilmiştir. Ardından önerilen kısıt programlama modeli tüm detayları ile sunulmuştur.

IBM ILOG Aralık ve Sıralama Değişkenleri ve İlgili Fonksiyonlar

Önerilen kısıt programlama modeli kurulurken IBM ILOG' un aralık ve sıralama değişkenleri ile bunlara ait tanımladığı ilgili fonksiyonlardan yararlanılmıştır.

Aralık değişkenleri

Bir aralık başlangıç, bitiş, boyut, uzunluk değerlerine sahiptir. Aralık değişkenleri bu değerlerin modelde değişken olmalarına izin vermektedir.

Aralık değişkenlerinin uzunluğu değişkenin başlangıç ve bitiş zamanı arasındaki fark kadardır. Boyutu ise görevin işlenmesi için gereken süre kadardır. Aralık değişkenleri için varsayılan boyut değeri uzunluğa eşittir. Genel olarak, boyut uzunluğun alt sınırıdır.

Aralık değişkeni opsiyonel olabilmektedir. Aralık çözümde görünmeyecek ise, bu aralıktaki kısıtlar yokmuş gibi davranacaklardır.

Aralık değişkenleri özellikle çizelgeleme ve sıralama problemlerinin çözümündeki performansın artırılması amacıyla tasarlanmıştır. MHDP' de çizelgeleme problemleri kapsamında yer aldığı için, bu çalışmada, incelenen problem için klasik yaklaşıma ek olarak aralık değişkenlerinin kullanıldığı bir KP modeli (KP2) oluşturulmuştur.

Sıralama Değişkenleri

Birçok problem aktivitelerin tek bir zamanda tek bir aktivite yapabilen kaynaklarda çizelgelenmesini içermektedir. Kaynak açısından bakılırsa, çözüm aktivitelerin sıralanmasıdır. Tek bir zamanda aktivitelerin üst üste gelmesinin yanında, hazırlık zamanları ile ilgili kısıtlar veya sıralamadaki aktivitelerin göreceli pozisyonları üzerindeki kısıtlamalar oldukça yaygındır. Sıralama değişkenleri bu fikri kapsamaktadır. Ve bu değişkenin değeri aralık değişkenlerinin bir permütasyonudur.

İlgili Fonksiyonlar

Kısıt programlama için OPL (Optimizasyon Programlama Dili) kısıtları şu şekilde tanımlanmıştır.

Aritmetik Kısıtlar ve İfadeler

Toplama, çıkarma gibi aritmetik operasyonları, bir sete ait en büyük en küçük değerleri bulma gibi aritmetik ifadeleri ve eşitlik, eşit olmama gibi aritmetik kısıtları içermektedir.

KP için Mantıksal Kısıtlar

Mantıksal ve/veya gibi mantıksal operatörleri içermektedir.

Çizelgeleme Kısıtları

Çizelgelemede kullanılabilecek kısıtları içermektedir. Bunlar öncelik kısıtları, aralık değişkenleri için kısıtlar, sıralama kısıtları ve durum fonksiyon kısıtları olarak sınıflandırılabilir.

Özelleşmiş Kısıtlar

Optimizasyon programlama dili kısıt programlama iyileştiricisi (OPL CP Optimizer) bazı güçlü kombinatorial kısıtları özelleşmiş kısıtlar olarak kabul etmektedir. Bazı güçlü türetme algoritmaları kullanılarak karar değişkeni sayısı azaltılmaya çalışılır. Bu çalışmada kullanılan aralık ve sıralama değişkenlerinin kontrolü amacıyla IBM ILOG kapsamında yer alan önemli bazı fonksiyonlara ilişkin kısa bilgiler Çizelge 6.1.'de sunulmuştur.

Çizelge 6.1. Optimizasyon programlama dili (OPL) kısıtları

OPL Kısıt	Gösterim	Kullanım Amacı
presenceOf	presenceOf(<i>a</i>)	opsiyonel olarak tanımlanan <i>a</i> aralık değişkeni seçilmiş yani aktifse true (1), seçilmemişse false (0) değerini çeviren bir fonksiyondur.
Alternative	Alternative(<i>a</i> , <i>B</i>)	<i>a</i> aralık değişkeni için <i>B</i> setinde bulunan opsiyonel aralık değişkenlerinden birisinin aktif yapılacağını (seçileceğini) ifade eder.
noOverlap	noOverlap(<i>M</i> [, <i>next</i>])	Aralıkların sıra değişkeni üzerinde çakışmamasını sağlar. Burada <i>M</i> sıra değişkeni ve <i>next</i> eklenecek beklemeledir.
startofPrev	startofPrev (<i>p</i> , <i>a</i> , <i>pVal</i> , [, <i>absVal</i>])	Tamsayı bir değer verir. <i>p</i> sıra değişkenindeki <i>a</i> aralık değişkeninden önceki aralık değişkeninin başlangıcını verir.
startofNext	startofNext (<i>p</i> , <i>a</i> , <i>pVal</i> , [, <i>absVal</i>])	Verilen sıra değişkenindeki bir sonraki aralık değişkeninin başlangıcını verir.
startOf	startOf(<i>a</i> , [, <i>dual</i>])	<i>A</i> aralığına ait başlangıç değerini verir.

KP2 için belirlenen sabitlerin tanımı, karar değişkenleri, alan-tanım kümeleri, kısıtlamalar ve sıkılaştırıcı kısıtlar bu bölümde detaylı bir şekilde anlatılmıştır.

Sabitler

Modelde kullanılan sabitler aşağıdaki gibidir.

Workers : Montaj hattında potansiyel olarak kullanılabilecek bütün işçilerin seti,
 $Workers = \{ 0, 1, \dots, nt \cdot M_{max} - 1 \}$

Stations : Montaj hattında potansiyel olarak kullanılabilecek bütün istasyonların seti,
 $Stations = \{ 0, 1, \dots, nt - 1 \}$

Karar değişkenleri (X seti)

$WorkerIsUsed_k$: $\begin{cases} 1, k \in Workers \text{ işçisi kullanıldı ise} \\ 0, \text{diğer durumda} \end{cases}$

$StationIsUsed_j$: $\begin{cases} 1, j \in Stations \text{ istasyonu kullanıldı ise} \\ 0, \text{diğer durumda} \end{cases}$

$Operation_i$: *i* görevi için aralık değişken

$altOperation_{ik}$: i görevinin k işçisine atanması durumunda kullanılacak opsiyonel aralık değişken

$sira_k$: $k \in Workers$ işçisinde aktif olacak (yani $operation_i$ değişkeni için seçilecek) $altOperation_{ik}$ aralık değişken değerlerinin sıralanmasını sağlayan sıra (sequence) değişkeni

Aralık değişkenler, belirli bir tamsayı noktasından başlayıp, belirli bir uzunluk boyunca devam ederek başka bir tamsayı noktasında sonlanan kararları temsil etmektedir. Örneğin görev zamanı 3 olan bir görevin bir işçiye görev operasyonunun 5 zamanında başlayacak şekilde atanması, uzunluğu 3 olan ve 5 zamanında başlayıp 8 zamanında biten bir aralık değişken değeriyle temsil edilecektir. Görevin atanabileceği her bir alternatif işçi için birer adet opsiyonel aralık değişkeni tanımlanır. İlgili görev için tanımlanan opsiyonel aralık değişkenlerden sadece birisi aktif olacak, diğerleri tamamı pasif yapılacaktır. Aktif olan aralık değişkenin ait olduğu işçi seçilen işçi olacaktır. Sıra değişkeni ise işçilerde aktif olan opsiyonel aralık değişkenlerinin sıralanması amacıyla kullanılmaktadır.

Alanlar-tanım kümeleri (D seti)

D-seti aşağıdaki gibi tanımlanmıştır.

$$WorkerIsUsed_k \in \{0,1\}$$

$$StationIsUsed_k \in \{0,1\}$$

$$Aralık\ Operation_i \in \{altOperation_{ik} \text{ değişkenleri}\}$$

$$altOperation_{ik} \in \{(k / M_{max})CT..((k / (M_{max} + 1))CT)\text{with size } t_i$$

$$Sequence\ sirak \in \{altOperation_{ik} \text{ değişkenleri}\}$$

Kısıtlamalar (C seti)

C-seti aşağıdaki gibi tanımlanmıştır.

$$workerIsUsed_k \geq \text{presenceOf}(altOperation_{ik}), \quad \forall i \in I \wedge \forall k \in Workers \quad (6.13)$$

$$stationIsUsed_{(k \text{ div } M_{max})} \geq 1, \\ \forall k \in Workers \wedge ((k \bmod M_{max} == 0) \wedge (workerIsUsed_k == 1)) \quad (6.14)$$

$$operation_i = \mathbf{alternative}\{altOperation_{i1}, altOperation_{i2}, \dots, altOperation_{ik}\}, \forall i \in I \quad (6.15)$$

$$\mathbf{noOverlap} (sira_k, ForwardSetup), \forall k \in Workers \quad (6.16)$$

$$\mathbf{startOf} (operation_h) \geq \mathbf{startOfPrev} (operation_i), \forall i, h \in I \text{ ve } i \in P(h) \quad (6.17)$$

$$\begin{aligned} \mathbf{startOfNext} (operation_i) &\geq (k / M_{\max})CT + \mu_{hi}, \\ &\forall i, h \in I \wedge \forall k \in Workers \wedge \{i \text{ görevi işçi } k' \text{ de ilk operasyon} \} \wedge \\ &\{h \text{ görevi işçi } k' \text{ de son operasyon} \} \end{aligned} \quad (6.18)$$

Sıkılaştırıcı kısıtlar

Sıkılaştırıcı kısıtlar aşağıdaki gibi tanımlanmıştır.

$$\begin{aligned} workerIsUsed_{k1} &\geq workerIsUsed_{k2}, \forall k1, k2 \in Workers \wedge (k1 / M_{\max} = k2 / M_{\max}) \wedge \\ &(k1 < k2) \end{aligned} \quad (6.19)$$

$$\begin{aligned} workerIsUsed_{k1} &\geq workerIsUsed_{k2}, \forall k1, k2 \in Workers \wedge (k1 \bmod M_{\max} = 0) \wedge \\ &(k2 \bmod M_{\max} = 0) \wedge (k1 < k2) \end{aligned} \quad (6.20)$$

Eş (6.13) kapsamında $workerIsUsed_k$ değişkenlerinin değerleri ayarlanır. Eğer m işçisinde tanımlı i aralık değişkeni seçilirse ($Seçildi(altOperation_{ik})=1$) değişkeninin değeri 1'den büyük olur ($workerIsUsed_k \geq Seçildi(altOperation_{ik})$). Eş. (6.14) yoluyla görev i için tanımlanan aralık değişkeni $operation_i$ için yine aynı göreve yönelik tanımlanmış opsiyonel aralık değişkenleri olan $altOperation_{i1}, altOperation_{i2}, \dots, altOperation_{ik}$ 'den birisinin seçilmesini sağlarken diğer değişkenler pasif yapılır. Eş. (6.15), işçi k 'da aktif yapılacak aralık değişkenlerinin örtüşmesini engelleyip sıralamaya bağlı olarak ileriye doğru hazırlık zamanlarının eklenmesi amacıyla kullanılmıştır. Eş. (6.16), aralarında öncelik ilişkisi olan görevler için tanımlanmış aralık değişkenlerinin başlangıç ve bitiş zamanlarını ayarlar. Eş. (6.17) ise aynı k işçisine ilk sırada atanmış (yani, opsiyonel aralık değişkeni seçilmiş) i aralık değişkeni ve son sırada atanmış j aralık değişkeni için i değişkeninin başlangıç zamanının $(k / M_{\max})CT + \mu_{hi}$ değerinden büyük olmasını sağlar. Burada $(k / M_{\max})CT$ değeri işçi başlangıç zamanını, μ_{hi} geriye doğru hazırlık zamanını göstermektedir. Dolayısıyla bu kısıt, k işçisine ilk sırada atanmış $operation_i$ aralık değişkeninin başlangıç zamanını en az hazırlık zamanı kadar geciktirir. Eş. (6.19) ve Eş. (6.20) model tanımı için gerekli olmayıp sıkılaştırıcı kısıtlar olarak oluşturulmuştur. Eş. (6.19), aynı istasyonda

bulunan ($k1/M_{\max} = k2/M_{\max}$) $k1$ ve $k2$ işçileri için eğer $k1 < k2$ ise $k1$ işçisi kullanılmadan $k2$ işçisinin kullanılmasını engeller ($workerIsUsed_{k1} \geq workerIsUsed_{k2}$). Diğer bir ifadeyle, kullanılan işçiler arasında kullanılmamış işçi olmadığından emin olmaktadır. Benzer şekilde Eş (6.20), istasyonlardaki ilk işçiler olan $k1$ ve $k2$ işçileri için ($k1 \bmod M_{\max} = k2 \bmod M_{\max} = 0$), eğer $k1 < k2$ ise (diğer bir ifadeyle $k1$ işçisi daha erken bir istasyonda bulunan ilk işçi ise) $k1$ kullanılmadan $k2$ 'in kullanılmasını önlemektedir. Yani, kullanılmış istasyonlar arasında kullanılmayan istasyon kalmadığından emin olmaktadır.

6.4. Hesaplama Sonuçları ve Değerlendirme

Pİ-MHDP/SB için önerilen çözüm yaklaşımlarının koşturulduğu problem setlerinden Bölüm 5'te bahsedilmiştir. Önerilen yaklaşımların birbirleri ile karşılaştırılabilmesi için ÖMAT-1 için kullanılan problem seti KP1 ve KP2 için de kullanılmıştır. Modeller için IBM ILOG Cplex CP Optimizer 12.6 çözücüsü kullanılmıştır. Çizelge 6.2. ve Çizelge 6.3'de Pİ/MHDP-SB için önerilen kısıt programlama modellerinin sonuçları özetlenmiştir. KP1 modeli ile problemlerin hiç biri için 3600 CPU zamanı içerisinde hiçbir çözüme (en iyi ya da uygun) ulaşamamıştır. Bu durum Çizelge 6.2'de görülmektedir. Çizelge 6.3.'e göre, KP2 modeli ile 72 adet problem setine ait en iyi çözümler elde edilmiştir. Bunun yanı sıra 38 adet problem setine ait sadece uygun çözümlere ulaşılmıştır. Çizelge 6.4.'te farklı hazırlık zamanları yoğunlukları için önerilen kısıt programlama modeli (KP2) performansını göstermektedir. KP1 modeli için hiçbir çözüme ulaşamadığından farklı hazırlık zamanları ile elde edilen sonuçlar tablolastırılmamıştır. KP2 modeli için CPU zamanı açısından hazırlık zamanı yoğunluklarının problem setleri üzerinde bir etkisi bulunmamaktadır. Buna bağlı olarak modelin çalıştırıldığı problem setlerinde elde edilen en iyi sonuçlar değişmemiştir. Gerek elde edilen en iyi çözüm sayısı gerekse kullanılan CPU zamanı açısından önerilen KP2 modelinin performansının ÖMAT-1'in performansından daha iyi olduğu açıkça görülmektedir. Ancak yine önerilen KP2 modelinin orta ve büyük boyutlardaki problem setlerinde istenilen verim alınamamıştır. Bu sebeple, Pİ-MHDP/SB için problemin NP-Zor yapısı düşünüldüğünde, orta ve büyük boyuttaki problem setlerinde kabul edilebilir bir zamanda iyi bir çözüme ulaşabilmek için meta sezgisel algoritmalarından yığın tabanlı arama yapan Genetik Algoritma (GA) ile, tek noktadan arama yapan Tavlama Benzetimi algoritması (TB) önerilmiştir.

Çizelge 6.2. Kısıt programlama (KP1) modeli sonuçları

Problem	<i>nt</i>	Optimali Bulunan	Uygun Çözüm Bulunan	Uygun Çözüm Bulunamayan	Toplam
Bowman	8			4	4
Jackson	11			24	24
Jaeschke	9			20	20
Mertens	7			24	24
Heskia	28			12	12
Gunther	35			16	28
Buxey	29			21	28
Lutz1	32			8	8
Sawyer	30			28	28
Arcus1	83			12	12
Arcus2	111			12	12
Bartholdi	148			12	12
Mukherje	94			12	12
Toplam				236	236

Çizelge 6.3. Kısıt programlama (KP2) modeli sonuçları

Problem	<i>nt</i>	Optimali Bulunan	Uygun Çözüm Bulunan	Uygun Çözüm Bulunamayan	Toplam
Bowman	8	4			4
Jackson	11	24			24
Jaeschke	9	20			20
Mertens	7	24			24
Heskia	28		4	8	12
Gunther	35		12	16	28
Buxey	29		7	21	28
Lutz1	32		4	4	8
Sawyer	30		11	17	28
Arcus1	83			12	12
Arcus2	111			12	12
Bartholdi	148			12	12
Mukherje	94			12	12
Toplam		72	38	126	236

Çizelge 6.4. Farklı hazırlık zamanı yoğunlukları için KP2 sonuçları

α	Problem	Optimali Bulunan	Uygun Çözüm Bulunan	Uygun Çözüm Bulunamayan	CPU
0.25	Bowman	1			1,59
	Jackson	6			4,55
	Jaeschke	5			2,71
	Mertens	6			2,02
0.50	Bowman	1			1,47
	Jackson	6			6,02
	Jaeschke	5			3,15
	Mertens	6			1,94
0.75	Bowman	1			1,63
	Jackson	6			5,47
	Jaeschke	5			4,73
	Mertens	6			2,16
1.00	Bowman	1			1,87
	Jackson	6			7,85
	Jaeschke	5			6,73
	Mertens	6			2,44

7. ÖNERİLEN METASEZGİSEL YÖNTEMLER

Bu bölümde Pİ-MHDP/SB için iki farklı meta sezgisel algoritma geliştirilmiştir. Bunlar, Genetik Algoritma ve Tavlama Benzetimi tabanlı algoritmalar. Bu algoritmaların detayları aşağıda anlatılmıştır.

7.1. Genetik Algoritma

Genetik Algoritmalar optimizasyon problemlerinin çözümünde kullanılabilen uyarlamalı metotlardır. Organizmaların genetik biyolojik süreçlerine dayanır. Jenerasyonlar boyunca doğal yığınlar doğal seçim kurallarına ve en iyinin hayatta kalmasına göre evrimleşmişlerdir. Genetik algoritmalar bu süreci taklit ederek gerçek hayat problemlerin çözümüne uyarlanabilirler (Goldberg, 1989).

Doğada, hayatta kalmayı en iyi başarabilen bireylerin daha fazla çocukları bulunmaktadır. Başarısız bireyler daha az çocuk üretirken bir süre sonra hiç üretemezler. Bunun anlamı, iyi adapte olmuş uyum sağlamış bireylerin genleri bir sonraki nesillerde bireylere aktarılacaktır. Farklı atalardan gelen güçlü karakteristikler bazen süper uygun çocuğu oluşturabilir. Bu süper uyumlu çocuğun uygunluk değeri kendi ebeveynlerinden de büyüktür. Böylelikle, türler çevrelerine daha uyumlu hale gelmektedir (Nabiyev, 2003).

Genetik Algoritmalar doğal davranışı direk olarak kullanırlar. Bireylerden oluşan yığınlarla çalışırlar. Yığınlardaki her birey tanımlanan problem için olası çözümleri ifade etmektedir. Bireyler kromozomlar ile ifade edilirler. Her kromozomun karakteristik özellikleri üzerlerindeki genlerin sıralanışı ile ifade edilir. Her bir kromozoma ait uygunluk değeri bulunmaktadır. Bu uygunluk değeri çözümün kalitesini belirtmektedir. Uygunluk değeri yüksek kromozomlar yığındaki diğer bireylerle birleşerek yeni kromozomlar üretirler ve yeni nesil bir önceki neslin en iyi bireylerinin karakteristiklerini içerirler. Böylelikle, en iyi özellikler sonraki nesillerde korunmuş olur. Eğer Genetik Algoritma iyi kurgulanmış ise, yığın sonunda optimal ya da optimale yakın çözüme ulaşır. İlk olarak 1975’ te Holland bilgisayar üzerinde optimizasyon problemlerinin çözümü için bu doğal sürecin simülasyonunu göstermiştir. 1989 ‘da Goldberg Genetik Algoritma uygulamalarının arama, optimizasyon ve makine öğrenmesi, teorisine detaylı bir giriş yapmıştır.

Genetik algoritmaların gücü yöntemin sağlam oluşundan ve büyük boyutlu problemler ile uğraşabilmesinden gelmektedir. Genetik Algoritmalar optimal sonucu bulmayı garantilemez ancak genellikle kabul edilebilir zamanlarda kabul edilebilir iyi sonuçlar vermektedir.

7.1.1. Genetik algoritmaların genel yapısı

Genetik Algoritmanın ilk aşamasında, tüm olası çözümlerin alt kümesinden oluşan bir başlangıç yığını oluşturulmaktadır. Yığının her ögesi bir dizi olarak kodlanır. Her dizi biyolojik olarak bir kromozomu ifade eder. Yığındaki her dizi uygunluk değerine sahiptir. Uygunluk değerine bakılarak hangi bireyin bir sonraki nesle taşınacağına karar verilmektedir. Problemin yapısına göre dizinin uygunluk değerinin yüksek veya düşük değer alması istenebilir. GA' nın her yeniden üretim işleminde seçim, çaprazlama ve mutasyon operatörleri işlev görmektedir (Karaoğlu ve Altıparmak, 2005).

Kodlama

GA ilgili problemi çözmedeki başarısı probleme uygun kodlama yapılmasına bağlıdır. İlgili probleme olası çözümün parametreler (genler) kümesi ile tanımlandığı varsayılır. Bu parametreler (genler) bir dizi (kromozom) oluşturmak için birleşirler. Literatürde farklı kodlama yapıları bulunmaktadır. Bunlar;

İkili kodlama

En yaygın kullanılan kodlama şeklidir. Kromozom ikili bit dizisi olarak kodlanır. Çözüme ait karakteristik özellikler dizideki her bit ile ifade edilir. Her bit dizisi çözümdür.

Çizelge 7.1. İkili kodlama örnek

Kromozom 1	1 0 0 1 1 1 0 0 0 1 0 0 1 1 1
Kromozom 2	1 0 0 0 0 1 1 1 0 1 0 0 0 1 0

Gerçek değer kodlama

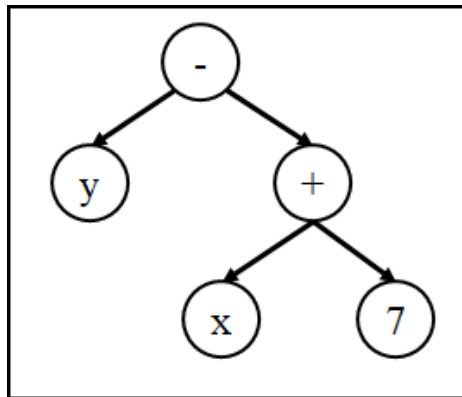
Kromozom içerisindeki genlerin, gerçek değerler, semboller ya da dizinler kullanılarak kodlanmasıdır.

Çizelge 7.2. Gerçek değer kodlama örnekleri

Kromozom 1	1 10.2 25 4 65 25.4
Kromozom 2	F F D G U O
Kromozom 3	(sağ), (sol), (geri), (ileri), (ön), (arka)

Ağaç kodlama

Kromozomları ağaç formunda ifade etmektedir. Şekil-7.1’de görselleştirilmiştir.



Şekil 7.1 Ağaç kodlama örnek

Permütasyon kodlama

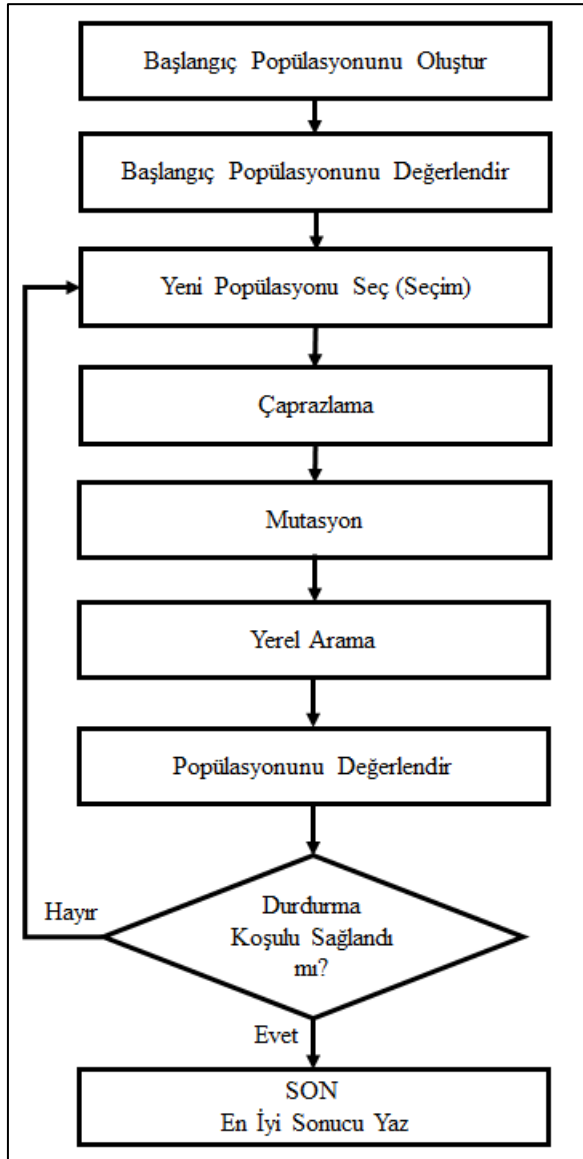
Kromozom, ardışık sayılar kullanılarak ifade edilmektedir. Her kromozom tam sayılı ve gerçek değerlerle gösterilir.

Çizelge 7.3. Permütasyon kodlama örnek

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

Genetik algoritma adımları

Şekil 7.2.'de Genetik Algoritmanın (GA) genel yapısı akış diyagramı ile anlatılmaktadır. Algoritma ilk olarak başlangıç popülasyonunun oluşturulması ile başlamaktadır. Ardından başlangıç popülasyonundaki bireyler için uygunluk değerleri hesaplanır. Uygunlukların hesaplanması ile yeni neslin oluşturulması için kullanılacak bireylerin belirlenmesi (seçim) gerçekleştirilir. Daha sonra seçilen bireylere çaprazlama ve mutasyon uygulanır. Belirlenen durdurma koşulu sağlanana kadar bu işlemler sırası ile tekrarlanır. Durdurma koşulu sağlandığında geliştirilen çözüm uzayından en uygun çözüm sonuç olarak sunulur (Mittchell, 1999). GA' nın bu adımları detaylı olarak alt bölümlerde anlatılmaktadır.



Şekil 7.2. Genetik algoritma genel yapısı

Başlangıç popülasyonunun oluşturulması

Genetik Algoritma’ da ilk olarak başlangıç popülasyonu oluşturulur. Popülasyonun her ögesi kromozom olarak ifade edilir. Popülasyondaki kromozom sayısı kullanıcı tarafından belirlenir ve bu sayının büyüklüğünün doğru belirlenmesi çözümün kalitesi ve hızı açısından önem arz etmektedir. Çok büyük olur ise en iyi çözüme yaklaşma olasılığı artabilir ancak, çözüme ulaşma süresi de artacaktır. Bu sebeple, popülasyon büyüklüğünün iyi belirlenmesi gerekmektedir. (Diaz-Gomez ve Hougen, 2007). Başlangıç popülasyonu kullanılarak, optimal çözüme ulaşmak için yeni popülasyon üretir. Üretilen bu yeni popülasyona genetik operatörler (yeniden üretim, çaprazlama ve mutasyon) uygulanır. Böylelikle daha yüksek uygunluk değerine sahip olan yeni nesil oluşturulmaya çalışır. (Toğan ve Daloğlu, 2008).

Uygunluk değerinin hesaplanması

Genetik algoritmalarda popülasyondaki kromozomların her birine ait uygunluk değeri bulunmaktadır. Bu değer, kromozomların çözüme ne kadar yaklaştığı ile ilgili bilgi vermektedir. Bu sebeple oldukça önem arz etmektedir. Uygunluk değerinin bir fonksiyon yardımı ile hesaplanması gerekmektedir. Bu fonksiyon problemin optimal çözüme ulaşma performansında önemli bir yere sahip olduğundan iyi tanımlanması gerekmektedir (Keskintürk ve Küçük, 2006).

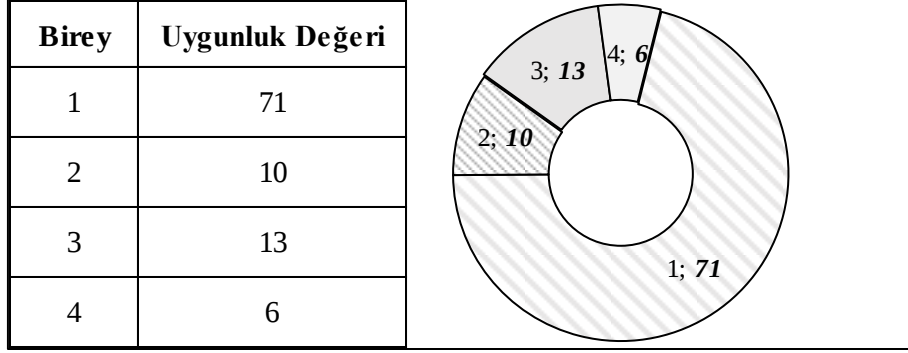
Seçim

Seçim operatörü popülasyondan yeni popülasyon oluşturmak için kullanılır. Seçilen bireylere çaprazlama ve mutasyon operatörleri uygulanıp yeni popülasyon elde edilir.

Seçim operatörü, bir sonraki popülasyonu oluştururken, kromozomların uygunluk değerlerini dikkate alır (Mccall, 2005). Bu operatör ile oluşturulacak yeni popülasyon içerisinde uygunluğu yüksek bireylerin bulunması amaçlanmaktadır (Goldberg, 1989). Literatürde geliştirilmiş birçok seçim yöntemi bulunmaktadır.

Rulet tekerleđi

En sık kullanılan seçim yöntemidir. Her kromozomun seçilme olasılığı uygunluk değeriyle orantılıdır (Miller ve Goldberg, 1995). Şekil 7.3.'de gösterilmiştir.



Şekil 7.3. Rulet Tekerleđi Seçimi Örnek

Sıralı seçim

Rulet tekerleđi seçim operatörü, kromozomlar arasında uygunluk değerlerindeki farkın fazla olduđu durumlarda uygulanması bazı kromozomların seçilme olasılığını azalttığı için problem yaratmaktadır. Sıralı seçim operatöründe kromozomun seçilme olasılığı, popülasyon içerisindeki uygunluk değerinin yerine göre belirlenir (Razali ve Geraghty, 2011). Sıralı seçim, kromozomları uygunluk değeri en büyükten başlayarak en küçüğe doğru sıralar. Böylece her bireyin seçilme olasılığı, sıralamadaki yerine göre belirlenmektedir.

Turnuva seçimi

Turnuva seçimi her ne kadar sıralı seçime benzese de hesaplama etkinliği açısından daha uygun bir seçim operatörüdür (Mitchell, 1999). Bireyler arasında turnuva gerçekleştirerek, eşleşme havuzunu doldurmaya çalışmaktadır. Genellikle iki birey arasında yapılan bu turnuva, ikiden fazla birey arasında da yapılabilir.

Eşleşme havuzu doluncaya kadar, operatör işleme devam etmektedir. Bu sayede eşleşme havuzu, uygunluk düzeyi yüksek bireylerle dolmaktadır. Uygunluk düzeyleri arasında

oluşan bu fark seçim baskısı yaratmakta, bu da genetik algoritmayı başarılı genlerin uygunluğunu arttırmaya zorlamaktadır (Sivanandam ve Deepa,2008).

Çaprazlama

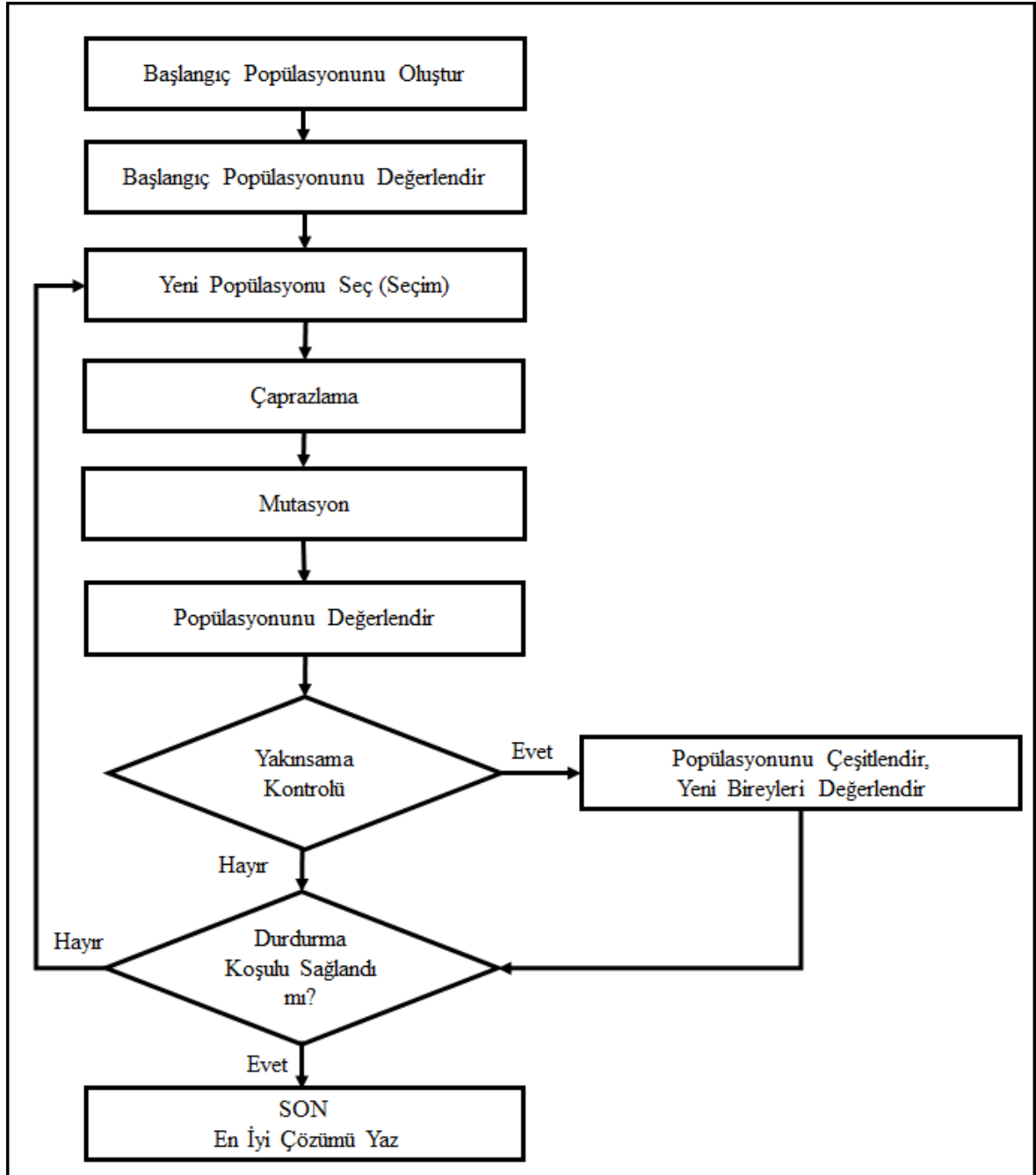
Çaprazlama operatörü ile ebeveyn kromozomlarının belirlenen bölgelerinin karşılıklı değişir. Böylelikle genetik özellikleri birbirinden farklı bir şekilde birleştirilmiş yavru kromozomlar oluşturulur. (Michalewicz, 1992). Literatürde çok sayıda çaprazlama operatörleri olup, en bilinenleri; Tek Noktalı, İki Noktalı ve N Noktalı Çaprazlama operatörleridir.

Mutasyon

Lokal minimum noktalarda sıkışmayı önlemek için, popülasyon çeşitliliğini yüksek tutmak gerekmektedir. Bunu gerçekleştirebilmek için, genetik algoritmalar mutasyonu önermektedir. Mutasyon; bir kromozom içerisinde seçilmiş gen veya genlerde rassal küçük değişiklikler yaparak, yeni birey oluşturma sürecidir (Razali ve Geraghty, 2011a).

7.1.2. Önerilen genetik algoritma

Bu bölümde önerilen genetik algoritma (GA) yaklaşımı tüm detayları ile sunulmuştur. GA, Kellegöz (2016) tarafından paralel çok işçili montaj hattı dengeleme problemi için tasarlanan algoritma göz önünde bulundurularak tasarlanmıştır. Şekil 7.4.'deki akış diyagramı önerilen genetik algoritmanın adımlarını gösterilmektedir. Algoritma ilk olarak başlangıç popülasyonunun oluşturulması ile başlar. Başlangıç popülasyonu değerlendirilir ve çaprazlama ve mutasyon operatörleri uygulanacak kromozomlar seçilir. Tekrar popülasyon değerlendirilir ve yakınsama kontrolü yapılır. Popülasyondaki bireyler arasındaki yakınsama belirlenen eşik değerinden büyük ise popülasyon çeşitlendirilir. Durdurma koşulu sağlanana kadar bu adımlar devam eder.

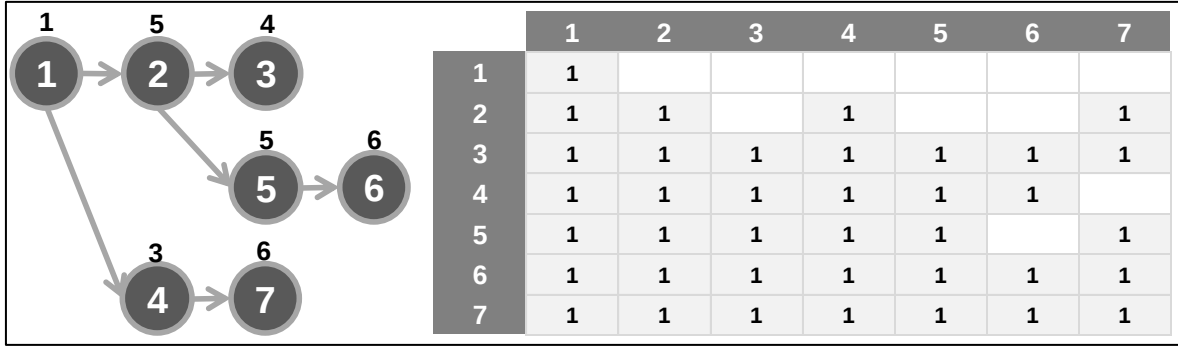


Şekil 7.4. Önerilen genetik algoritma genel yapısı

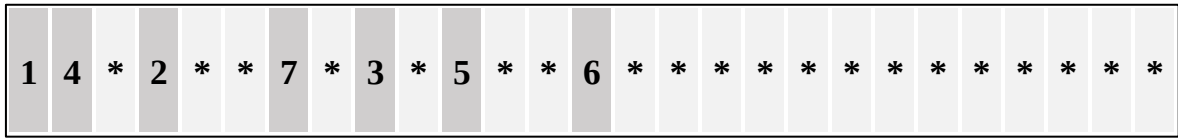
Çözümün kodlanması

Probleme etkin bir çözüm elde etmek isteniyorsa çözüm gösteriminin büyük önemi bulunmaktadır. Bu bağlamda, montaj hattı dengeleme problemlerinde en çok kullanılan çözüm gösterimi yöntemleri istasyon tabanlı ve görev tabanlı olmak üzere ikiye ayrılır (Scholl ve Becker, 2006). İstasyon tabanlı çözüm gösteriminde indeksler görevleri, indekslere denk gelen değerler istasyon bilgisini vermektedir. Görev tabanlı çözüm

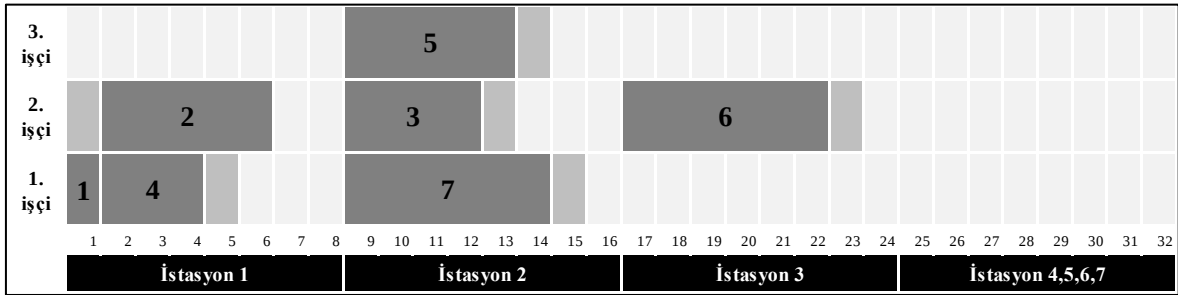
gösteriminde uygun görev sıralaması çözüm olarak ifade edilir. İstasyon tabanlı çözüm gösteriminde görevlerin sıralama bilgisi bulunmadığından bu gösterimin incelenen probleme uygun olmadığı gözlemlenmiştir. Bu sebeple, burada görev tabanlı çözüm gösterimi yaklaşımı kullanılmaktadır. Ancak çok işçili montaj hattı dengeleme problemlerini görev tabanlı çözüm gösterimi yaklaşımı ile ifade etmek için çözümde istasyonlarda bulunan işçilerin birbirinden bir ayraç yardımı ile ayrılması gerekmektedir. Sonuç olarak çözüm gösteriminde kullanılacak olan kromozom büyüklüğü (kromozomdaki gen sayısı) $nt + K_{max} \cdot M_{max} - 1$ (*maksimum istasyon sayısı ile maksimum işçi sayısı çarpımının görev sayısı ile toplamından yalnızca bir eksik*) kadar olmalıdır. nt atanacak görev sayısını gösterirken, $K_{max} \cdot M_{max} - 1$ kromozomda kullanılacak ayraç sayısını göstermektedir. Şekil 7.3.'teki örnek problem literatürde Mertens problemi olarak geçmektedir. Problemde bulunan 7 göreve ait öncelik diyagramı verilmiş olup görev zamanları diyagramdaki düğümlerin üzerinde gösterilmiştir. Şekil 7.5'te bulunan matris ise görevler arası geri hazırlık zamanlarını göstermektedir. Bu veri setinde görevler arası ileri hazırlık zamanı bulunmamaktadır. Çevrim zamanı 8 olarak belirlenmiştir. K_{max} (*izin verilen maksimum istasyon sayısı*) ve M_{max} (*bir istasyonda bulunabilecek maksimum işçi sayısı*) değerleri sırasıyla 7 ve 3 olarak tanımlanmıştır. Şekil 7.6.'te bulunan kromozom çok işçili montaj hattındaki uygun görev atamasını göstermektedir. Bu uygun kromozomu detaylandırmak gerekirse; 20 adet yıldız (*) ayraç olarak kullanılmıştır. Bu yirmi adet ayraç hatta yer alabilecek maksimum toplam işçi sayısından ($7 \cdot 3 = 21$) yalnızca 1 eksiktir. Ayraçlar işçi yüklerinin birbirinden ayırmaktadır. Sıralamaya bakıldığında ilk işçinin görevlerinin sırası ile 1 ile 4, 2. İşçinin görevlerinin 2, 3. İşçinin ise atanan hiçbir görevinin almadığı anlaşılmaktadır. Bu kodlamaya karşılık gelen çözümün montaj hattı üzerindeki gösterimi Şekil 7.7.'teki gibidir. Montaj hattında maksimum 7 adet istasyon ve her istasyonun maksimum 3 işçi bulundurduğu görülmektedir. İşçilere atanan görevler başlangıç ve bitiş zamanları ile birlikte gösterilmiştir. Toplam kullanılan işçi sayısının 6, istasyon sayısının ise 3 olduğu görülmektedir. Şekil 7.7.'te işçiler üzerinde bulunan siyah renkler işçilere atanan görevleri, koyu gri renkler ise görevler arasındaki geri hazırlık zamanlarını ve açık gri renkler boş zamanları göstermektedir. Örneğin, 1. istasyondaki 1. işçide 1 ve 4 görevleri ardışık sıra ile yapılacaktır ve bir sonraki çevrimde 4 görevinden 1 görevine geçerken 1 birimlik geri hazırlık zamanına ihtiyaç duyacaktır.



Şekil 7.5. Örnek problem



Şekil 7.6. Örnek problem çözüm gösterimi



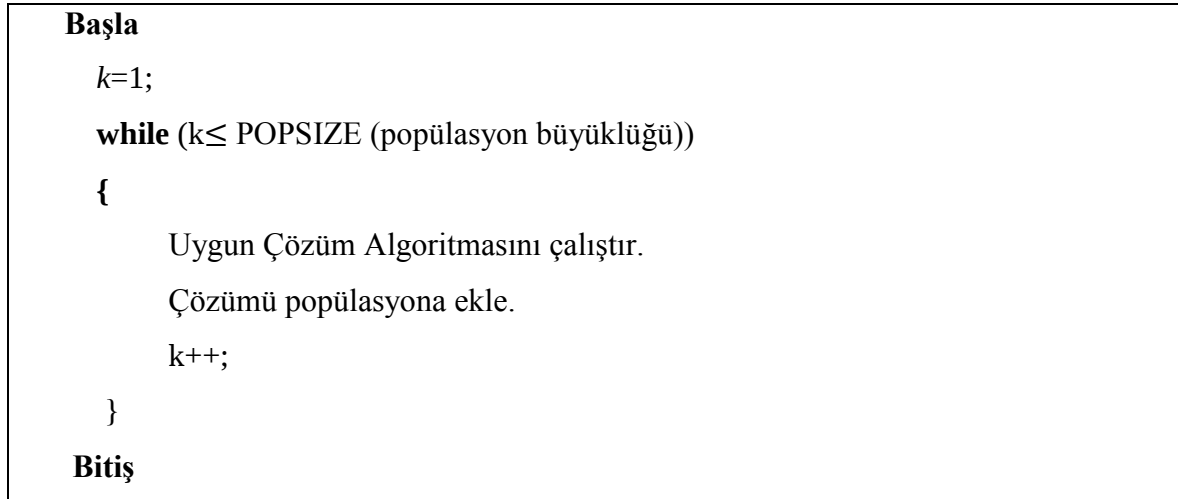
Şekil 7.7. Örnek problem uygun çözüm hat konfigürasyonu

Önerilen genetik algoritma adımları

Önerilen genetik algoritmaya ait adımlar aşağıda tüm detayları ile anlatılmıştır.

Başlangıç popülasyonunun oluşturulması

Başlangıç popülasyonunun oluşturulabilmesi için Şekil 7.8'deki algoritma uygulanır. Algoritmaya göre, popülasyon büyüklüğü (POPSIZE) belirlendikten sonra, popülasyon büyüklüğüne ulaşıncaya kadar incelenen problem için tasarlanan uygun çözüm algoritması çalıştırılır.



Şekil 7.8. Başlangıç popülasyonu algoritması

Başlangıç popülasyonundaki her kromozom problem için uygun bir çözümü ifade eder. Uygun çözümler Şekil 7.9.'de sunulan algoritmaya göre rassal olarak elde edilir. Bu algoritmaya göre, ilk olarak atanabilecek görevler listesi (AGL) belirlenir. AGL öncül görevleri atanmış görevlerden oluşmaktadır. Bu listeden bir görev rassal olarak seçilir ve işçi sayısı 1 ile Mmax değeri arasında rassal olarak belirlenmiş bir istasyon açılır. AGL listesinden rassal olarak seçilen her görev için çevrim zamanı kontrol edilerek açılan istasyondan rassal olarak atanabilecek işçi seçilir. İstasyona atanabilecek görev kalmadığında, yeni bir istasyon açılır ve tüm görevler atanana kadar bu izlek devam ettirilir.

Başla

Atanabilecek görevler listesini hazırla (AGL) (*öncülleri atanmış görevler*)

İşçi sayısı 1 ile M_{max} arasından rassal olarak belirlenen bir istasyon aç.

$i=1$;

$t \leftarrow$ AGL'den rassal görev seç

while (AGL boş değil)

{

if (t görevi istasyondaki işçilerden birine (*işçiler rassal seçilir*)
 atanabiliyor ise)

 {

t görevini seçilen işçiye ata.

 Atanabilecek görevler listesini hazırla (AGL)

 }

else

 {

$i++$;

 Rassal başka bir t görevi seç.

 }

if ($i > AGL'nin\ boyutu$)

 {

 İşçi sayısı 1 ile M_{max} arasında rassal belirlenen yeni bir istasyon aç.

$i=1$;

$t \leftarrow$ AGL'den rassal görev seç

 }

Bitiş

Şekil 7.9. Uygun çözüm algoritması

Uygunluk değerinin hesaplanması

Bu tez kapsamında çalışılan montaj hattı dengeleme problemi en küçükleme problemi olduğu için problemin amaç fonksiyonu, uygunluk değeri olarak doğrudan kullanılamaz. Öyle ki, uygunluk değeri GA tarafından maksimum yapılmaktadır. Amaç fonksiyonu, birincil amaç olarak istasyonlarda bulunan toplam işçi sayısı ve ikincil amaç olarak açılan

istasyon sayısını en küçüklemektedir. Bu çalışmada uygunluk değeri amaç fonksiyonunun tersi olarak alınmıştır. Eş.7.1’de gösterilmiştir.

$$1 / \left(\sum_{j \in J} \sum_{k \in K} w_{jk} + 1 / (nt + 1) \sum_{j \in J} y_j \right) \quad (7.1)$$

Seçim

Başlangıç popülasyonundan yeni popülasyon oluşturmak için Seçim operatörü olarak İkili Turnuva Seçim operatörü kullanılmıştır. Popülasyondaki kromozomlar arasından rastgele seçilen iki kromozom uygunluk değerlerine göre turnuvaya sokulur ve iyi olan çaprazlama ve mutasyon operatörlerinin uygulanması için yeni popülasyona aktarılır.

Mutasyon

Eğer mutasyon olasılığı sağlanıyor ise popülasyondan rassal seçilen bir kromozoma 0,5 olasılığı ile ya *ekleme* ya da *ikili değiştirme* komşu üretme mekanizması kromozomda rassal seçilen pozisyonlara uygulanır.

Ekleme komşu üretme mekanizması

Güncel çözümden rassal olarak bir görevi seçer ve o görevi aynı işçinin farklı bir pozisyonuna ya da farklı bir istasyonda farklı bir işçinin bir pozisyonuna atayabilir. Burada belirtilmesi gereken önemli husus, ekleme mekanizması sonrasında oluşan komşu çözüm her zaman uygun bir çözümdür. Güncel çözümden rassal olarak seçilen görevin çözümün uygunluğunu bozmadan nereye eklenebileceği yine rassal olarak uygun alternatifler arasından seçilir. Elde edilen komşu çözümde uygun çözümü yakalamak için aşağıdaki kuralların sağlanması gerekmektedir.

Kural 1: Ekleme mekanizması uygulanacak görev için rassal olarak seçilen alternatif pozisyon farklı bir istasyonda ise o görevin hemen (komşu) öncülleri o istasyona ve daha önceki istasyonlara atanmış değildir.

Kural 2: Ekleme mekanizması uygulanacak görev için rassal olarak seçilen alternatif pozisyon farklı bir istasyonda ise o görevin hemen (komşu) ardılları o istasyona ve daha sonraki istasyonlara atanmış olmalıdır.

Kural 3: Ekleme mekanizması uygulanacak görev atanmış işçinin iş yükü çevrim zamanı kısıdını aşmamalıdır.

Kural 4: Ekleme mekanizması uygulanacak görev o istasyona eklendiğinde istasyondaki hiçbir işçinin iş yükü çevrim zamanı kısıdını aşmamalıdır. Bir diğer deyişle istasyonda bulunan hiçbir görevin bitiş zamanı çevrim zamanından büyük olamaz.

Kural 5: Görevin ekleme yapılacağı pozisyonundan sonra atanmış bir öncülü bulunmamalıdır.

Kural 6: Görevin ekleme yapılacağı pozisyonundan önce atanmış bir ardılı bulunmamalıdır.

Ancak bu kurallar sağlandığında uygun bir komşu çözüm elde edilmektedir. Ekleme komşu üretme hareket mekanizmasına ait sözde kod Şekil 7.10.'da sunulmuştur. İlk olarak rassal görev listesi (RL) oluşturulur. Rassal görev listesindeki her bir görev için rassal uygun istasyon listesi oluşturulur (RS). Bu liste oluşturulurken Kural 1 ve Kural 2 uygulanır. RS de bulunan her bir istasyon için rassal işçi (Rİ) listesi hazırlanır. Bu liste için uygulanan kural Kural 3 tür. Rİ deki her bir işçi için Kural 5 ve Kural 6 uygulanarak işçideki uygun rassal pozisyon listesi (RP) hazırlanır ve rassal seçilen görev Kural 4 sağlanıyorsa ilgili atama yapılır ve algoritma sonlanır. Diğer durumda algoritma RL den başka bir görevle aynı adımları izler. Ekleme komşu üretme hareket mekanizması için Şekil-7.5' te verilen örnek probleme uygun bazı hareketler yukarıda belirtilen kurallar uygulandığında Şekil 7.11'deki gibidir. Örneğin görev 4'ün ekleme hareketi ile atanmasının uygun olduğu işçiler birinci istasyonun sonuncu işçisi, ikinci istasyonun ilk ve ikinci işçileridir. Görev 2 için uygun olan ekleme hareketleri ise birinci istasyonun sonuncu işçisidir.

BaşlaRassal görev listesi oluştur(*RL*)**Foreach** (*görev in RL*)

{

Rassal uygun istasyon listesi oluştur (*RS*) *Kural 1 ve Kural 2***Foreach** (*istasyon in RS*)

{

Rassal işçi listesi oluştur (*RI*) *Kural 3***Foreach** (*işçi in RI*)

{

Rassal uygun pozisyon listesi oluştur (*RP*) *Kural 5 ve Kural 6***Foreach** (*pozisyon in RP*)

{

if (*görev atanabiliyorsa*) *Kural 4*

{

Eklemeyi yap.

Algoritmayı sonlandır.

}

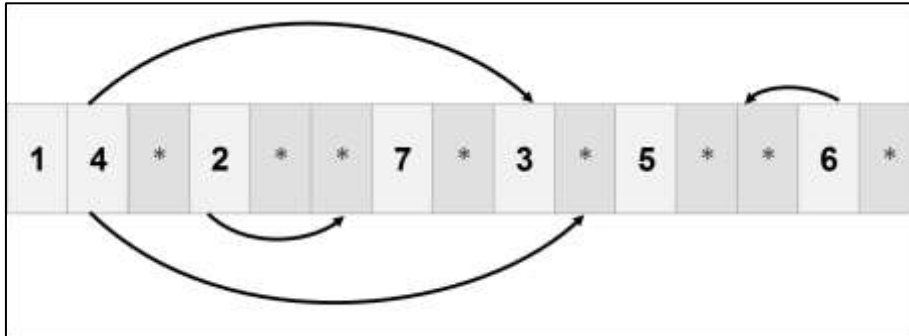
}

}

}

Bitiş

Şekil 7.10. Ekleme komşu üretme mekanizması



Şekil 7.11. Uygun ekleme hareketleri

İkili değiştirme hareket mekanizması

İkili Değiştirme hareket mekanizması ile güncel çözümden rassal olarak seçilen görev için kendi indis numarasından büyük olan görevler arasından rassal olarak bir görev seçilir. Bu iki görev arasında ikili yer değiştirme algoritması uygulanır. Şekil 7.12.'de algoritmanın sözde kodu bulunmaktadır. Burada da ekleme komşu üretme mekanizmasında olduğu gibi belirtilmesi gereken önemli husus, İkili değiştirme sonrasında oluşan komşu çözüm her

zaman uygun bir çözümdür. Elde edilen komşu çözümde uygun çözümü yakalamak için aşağıdaki kuralların sağlanması gerekmektedir.

Kural 1: Eğer seçilen bu iki görev aynı istasyonda aynı işçinin farklı pozisyonlarına atanmışlar ise algoritmanın uygulanabilmesi için aralarında öncelik ilişkisi bulunmamalıdır.

Kural 2: Eğer seçilen bu iki görev aynı istasyonda farklı işçilere atanmışlar ise işçilerden en az birine i atanan görev sayısı birden büyük olmalıdır.

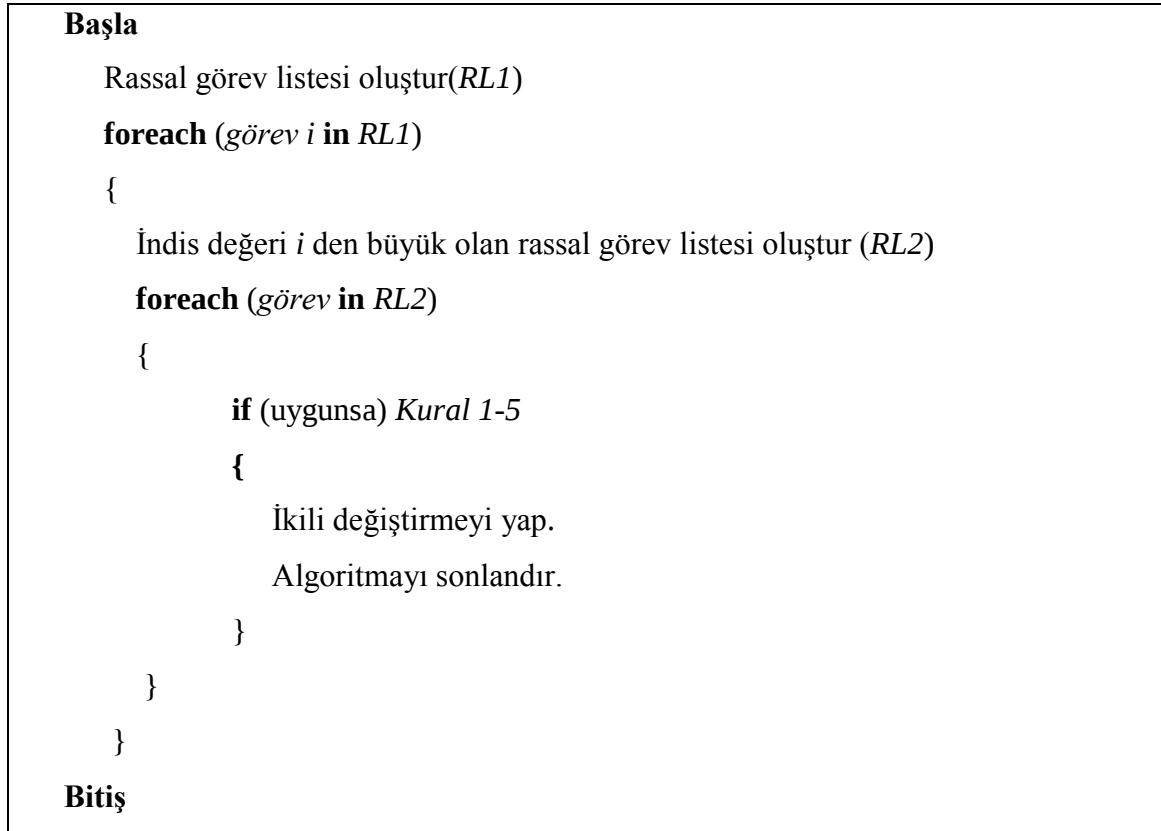
Kural 3: Eğer seçilen görevler farklı istasyonlarda ise aralarında öncelik ilişkisi bulunmamalıdır. Ayrıca her iki görev için istasyonlar atanabilir olmalıdır. (Görevlerin öncülleri ve ardıları göz önünde bulundurularak)

Kural 4: Seçilen görevlerin atanacağı pozisyonların öncülleri ve ardıları dikkate alındıktan sonra atanabilir pozisyon olmaları gerekmektedir.

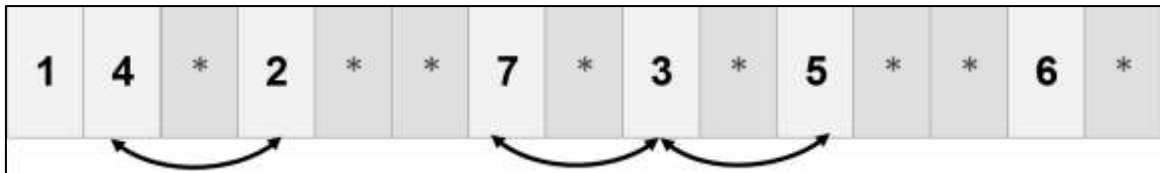
Kural 5: İkili yer değiştirme gerçekleştikten sonra her işçilerin iş yükleri belirlenen çevrim zamanını aşmamalıdır.

Kural 6: İkili yer değiştirme gerçekleştikten sonra hiçbir görevin bitiş süresi çevrim zamanını aşmamalıdır.

Ancak bu kurallar sağlandığında uygun bir komşu çözüm elde edilmektedir. Şekil 7.13.'e bakıldığında uygun olan bazı yer değiştirme hareketleri görülmektedir. Örneğin görev 2 ile görev 4 ün yukarıdaki kurallar uygulandığında yer değiştirilmelerinin uygun olduğu görülmektedir.



Şekil 7.12. İkili değiştirme komşu üretme mekanizması



Şekil 7.13. İkili yer değiştirme uygun hareketleri

Çaprazlama

Permütasyon kodlama için uygun çaprazlama metotlarından biri olan OBX tek noktalı çaprazlama operatörü modifiye edilmiş permütasyon kodlamada kullanılmıştır. Seçim operatörü sonrası oluşturulan popülasyondan rastgele seçilen iki adet kromozom eğer çaprazlama olasılığı sağlanıyor ise rastgele seçilen bir noktadan iki çocuk oluşturmak için çaprazlanır. Çaprazlama sonucunda ortaya çıkan çocuk uygun bir çözüm olmayabilir. Bu yüzden bir düzeltme fonksiyonu yardımı ile elde edilen çocuk uygun bir çözüm haline getirilir. Örnek çaprazlama Şekil-7.14'te gösterilmiştir. Örnekteki ebeveyn kromozomlarının uzunluğu 27 (7 (görev sayısı) + 7×3 (istasyon sayısı ve işçi sayısı) -1) olmaktadır. Örnekte çaprazlama noktası rassal olarak 9 olarak belirlenmiştir. Çaprazlama

sonucunda oluşturulacak çocuk kromozomun ilk dokuz geni, ebeveyn 1 kromozomunun ilk dokuz geni ile aynıdır. Ebeveyn 2 de çocuk kromozoma ait ilk dokuz genin değerleri çıkartılır ve geriye kalan değerler sırasıyla çocuk kromozomun genlerini tamamlamak için kullanılır. Böylelikle permütasyon sıralama yapısı korunmuş olur. Aynı işlem ebeveynlerin yerleri değiştirilerek ikinci çocuk kromozomu oluşturmak için kullanılır.

Ebeveyn 1																											
No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Kromozom Değeri	1	4	*	*	*	2	*	*	7	*	5	*	*	3	*	*	*	6	*	*	*	*	*	*	*	*	*
Çocuk																											
No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Kromozom Değeri	1	4	*	*	*	2	*	*	7	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
Ebeveyn 2																											
No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Kromozom Değeri	1	2	*	*	*	5	*	3	4	*	*	6	*	*	7	*	*	*	*	*	*	*	*	*	*	*	*
Çocuk																											
No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Kromozom Değeri	1	4	*	*	*	2	*	*	7	?	?	*	6	*	*	*	*	*	*	*	*	*	*	*	*	*	*

Şekil 7.14. Tek noktalı OBX çaprazlama

Çaprazlama operasyonu sonucunda elde edilen kromozomların bir kısmı uygun olmayabilir. Bu kromozomlar Şekil 7.15.'teki düzeltme fonksiyonu yardımı ile uygun hale getirilir.

Elitizm

Popülasyondaki en iyi uygunluk değerine sahip bireye elit birey denir. Elit birey bir sonraki nesle çaprazlama, mutasyon operatörlerine maruz kalmadan aktarılır. Böylelikle bir sonraki nesle en iyi bireyin özellikleri taşınmış olur.

Başla

```

cp=çaprazlama noktası
çaprazlama sonrası elde edilen kromozom parçasını al.
k=cp+1;
While ( $k \leq (nt + (M_{max} \cdot K_{max} - 1))$ )
{
    If (Kromozom[k] != '*')
    {
        Kromozom[k] daki görevin hangi işçiye atandığını tespit et.
        If (görevin atandığı işçi uygunsa (Ekleme komşu üretme kuralları sağlanıyorsa)
        {
            Kromozom [k] nın değerini koru.
            k++;
        }
        Else
        {
            Kromozom[k]=(Kromozom[k+1] ile Kromozom[nt+(Mmax.Kmax-1)]
            arasındaki ilk *, ekleme metodu ile Kromozom[k]ya eklenir.)
            k++;
        }
    }
    Else
    {
        k++;
    }
}

```

Bitiş

Şekil 7.15. Düzeltme fonksiyonu

Yakınsama kontrolü

Her yeni nesil oluşturulduktan sonra bir benzerlik değeri (SV) hesaplanır. SV değeri Eş. 7.2’de verilmiştir. SV değeri, kromozomların aynı pozisyonlarının değerlerinin birbirleriyle olan benzerliğine bakılır (Özcan, Kellegöz, Toklu, 2011). Geliştirilen algorithmada SV değeri popülasyondaki ilk birey baz alınarak hesaplanmaktadır.

$$R_{ijp} = \begin{cases} 1, & \text{eğer } i \text{ kromozomunun ve } j \text{ kromozomunun } p \text{ pozisyonundaki değeri aynı ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$\sum_{j=2}^{POPSIZE} \sum_{p=1}^{nt+nt*Mmax-1} R_{ijp} / ((POPSIZE - 1) \times (nt + nt * Mmax - 1)) \quad (7.2)$$

Eğer SV değeri belirlenen bir eşik değerine (TRE) eşit ya da büyük ise popülasyondaki kromozomlar elit kromozom hariç rassal olarak yeniden üretilir.

7.2. Tavlama Benzetimi Algoritması

Bu bölümde tez kapsamında çalışılan montaj hattı dengeleme problemine tek noktadan arama yapan bir metot ile bir çözüm önerisi getirmek için sunulan Tavlama Benzetimi Algoritması (TB) anlatılmaktadır. İlk olarak, algoritmaya adını veren fiziksel bir süreç olan tavlama benzetiminden bahsedilmektedir. Ardından, algoritma genel yapısı ile tanıtılmaktadır. Son olarak önerilen tavlama benzetimi algoritması detayları ile anlatılmaktadır.

7.2.1. Fiziksel tavlama süreci

Tavlama benzetimi fizik biliminde, bir katının ısı banyosunda düşük enerji durumlarının elde edilmesi için bir ısıl süreç olarak tanımlanmaktadır. İlk olarak ısı banyosunun sıcaklığı katının eriyebileceği sıcaklığa yükseltilir ardından ısı banyosunun sıcaklığı dikkatli bir biçimde azaltılır. Azaltma işlemi katının yer durumundaki parçacıkları kendini düzenleyene kadar yapılır. Katı sıvı halde iken tüm parçacıkları kendini rassal olarak düzenler. Yer durumuna geçtiğinde bu parçacıklar bir kafes şeklinde düzenlenir. Bu durumda sistemin sahip olduğu enerji en düşük seviyededir. Eğer sıcaklık yeteri kadar yükseltilmemiş ve soğutma yeterince yavaş yapılmamış ise katı yer durumuna geçemez. Fizik bilimindeki bu sürecin kombinatoryal optimizasyon problemleri ile benzerliği Çizelge 7.4.'te özetlenmiştir. Katı maddenin farklı durumları kombinatoryal problemdeki olası farklı çözümleri, sistemin enerjisi problemin sahip olduğu amaç fonksiyonunu ifade etmektedir. Global optimum değer katının yer durumu ile yerel optimum değer yarı kararlı durum ile benzerlik gösterir.

Çizelge 7.4. Tavlama süreci ile tavlama benzetimi algoritması arasındaki benzerlik

Termodinamik Benzetimi	Kombinatoryal Optimizasyon
Sistemin durumları	Uygun Çözümler
Enerji	Durumun değişimi
Amaç fonksiyonu değeri	Komsu çözüm
Sıcaklık	Parametre
Kristalleşme	Sezgisel Çözüm

Katının mevcut durumu i enerjisi E_i olsun, bir sonraki j durumuna bir hareket mekanizması ile geçiş yapılır ve enerjisi E_j olur. İki durum arasındaki enerji farkı ($E_j - E_i$) sıfır veya daha küçük ise j durumu yeni mevcut durum olarak kabul edilir. Eğer fark sıfırdan büyükse bir olasılık fonksiyonuyla j durumu kabul edilir. Bu kabul fonksiyonuna Metropolis Kriteri denir.

7.2.2. Tavlama benzetimi algoritmasının genel yapısı

Tavlama Benzetimi Algoritması yerel arama tabanlı meta sezgisel metotlardan biridir. İlk olarak kombinatoryal optimizasyon problemleri için Kirkpatrick ve diğerleri (1983) tarafından kullanılmıştır. Algoritma bir önceki bölümde anlatılan fiziksel bir süreç olan tavlama sürecini taklit etmektedir. Tavlama Benzetimi Algoritması yerel en iyide takılmayı engelleyen oldukça etkin bir yapıya sahip olan komşu arama metotlarından biridir. Rassal olarak ya da bir sezgisel metot ile belirlenen tek bir başlangıç çözümü ile başlar. Her bir iterasyonda bir hareket mekanizması ile güncel çözümün etrafında gezinir (komşu çözümler oluşturulur). Elde edilen komşu çözüm bir öncekinden daha kötü değil ise, komşu çözüm güncel çözüm olur. Yerel en iyiden sakınmak ve arama sürecindeki çeşitliliği sağlamak için kötü çözümler de $e^{-\frac{\Delta f}{T}}$ olasılığı ile kabul edilir. Bu kabul fonksiyonundaki Δf güncel çözüm ile güncel çözümün amaç fonksiyonları arasındaki farkı ifade ederken, T_k sıcaklığı göstermektedir. Mevcut sıcaklık değeri T_k ne kadar yüksek ise kötü bir çözümü kabul etme olasılığı da o kadar yükselir. Bu yüzden, T_k sıcaklığı süreç boyunca sistematik olarak azaltılmaktadır. Böylelikle, arama süreci boyunca en iyi çözümün bulunması amaçlanmaktadır.

7.2.3. Önerilen tavlama benzetimi algoritması

Önerilen Tavlama benzetimi algoritması Şekil-7.16'daki gösterilmiştir. İlk olarak algoritma, rassal elde edilen bir başlangıç çözümü ile başlamaktadır. Bu çözüm bir önceki bölümde anlatılan uygun çözüm bulma algoritması ile elde edilmektedir. Başlangıç çözümü, hem güncel çözüm hem de ilk en iyi sonuç olarak belirlenir ve güncel T_k sıcaklık değeri ilk sıcaklığa (T_{ilk}) eşitlenir. T_k sıcaklığı α nın soğutma katsayısı olduğu durumda $T_{k+1} = \alpha \cdot T_k$ soğutma fonksiyonu ile yavaşça T_{son} sıcaklığına ulaşmaya kadar soğutulur. Her sıcaklık değerinde tavlama benzetimi algoritmasının adımları M iterasyon sayısı kadar tekrar eder. Güncel çözümden belirli bir olasılık değerine göre (0,5) komşu çözüm üretilir. Komşu çözümler ekle ve ikili yer değiştirme tabanlı hareket mekanizmaları kullanılarak üretilmektedirler. Komşu çözüm üretme mekanizmaları olan ekle ve ikili yer değiştirme, GA başlığı altında bir önceki bölümlerde detaylı anlatılmıştır. Komşu üretildikten sonra, güncel çözüm ile komşu çözüm arasındaki değişim Δf hesaplanır. Eğer komşu çözümün amaç fonksiyonu güncel çözümün amaç fonksiyonundan daha iyi ise, komşu çözüm yeni güncel çözüm olarak kabul edilir. Diğer taraftan, komşu çözüm güncel çözümden daha kötü ise, komşu $e^{-\frac{\Delta f}{T_k}}$ olasılığı değeri ile güncel çözüm olarak kabul edilir. Olasılık değeri sağlanmıyor ise komşu reddedilir. Güncel çözüm o ana kadar bulunan en iyi çözümden daha iyi ise, en iyi çözüm olarak güncellenir. Bu süreç T_{son} sıcaklık değerine (durdurma koşulu sağlanana kadar) ulaşılanaya kadar devam eder. Algoritma sonlandığında en iyi çözüm gösterilir.

Başla

Başlangıç çözümünü oluştur (S_0) ;

$S_c = S_0, S^{best} = S_0$;

$T_k = T_{ilk}$;

Başlangıç çözümünün amaç fonksiyonu değerini hesapla ($f(S_0)$);

$f(S_c) = f(S_0), f(S^{best}) = f(S_0)$;

while ($T_k > T_{son}$)

{

$k=1$;

while ($k \leq M$)

 {

if (rassal sayı < 0.5)

 {

 Güncel çözümünden (S_c) *ekle* ile komşu çözüm (S_n) üret;

 }

else

 {

 Güncel çözümünden (S_c) *ikili değiştirme* ile komşu çözüm (S_n) üret;

 }

$\Delta f = f(S_n) - f(S_c)$;

if ($\Delta f \leq 0$ veya rassal sayı $\leq e^{(-\Delta f/T_k)}$)

 {

$S_c = S_n$;

$f(S_c) = f(S_n)$

 }

if ($f(S_c) < f(S^{best})$)

 {

$S^{best} = S_c$;

$f(S^{best}) = f(S_n)$;

 }

$k = k + 1$;

 }

$T_{k+1} = \alpha \cdot T_k$;

}

Yaz S^{best}

Bitiş

Şekil 7.16. Önerilen tavlama benzetimi algoritması

7.3. Parametre Analizi

Bu bölümde önerilen algoritmalar için gerekli parametre analizleri anlatılmaktadır.

7.3.1. Tavlama benzetimi parametre seçimi

Tavlama Benzetimi Algoritmasının parametreleri Çizelge 7.5'deki gibidir.

Çizelge 7.5. Tavlama benzetimi parametre seçimi

Parametre	Tanımı	Seviye	Seçilen Değer
T_{ilk}	İlk Sıcaklık	300, 500, 750	500
T_{son}	Son sıcaklık	0,05; 0,15; 0,25	0,15
α	Soğutma Oranı	0,80; 0,90; 0,99	0,99
M	İterasyon Sayısı	10, 20, 40	20

Parametre analizi için küçük, orta ve büyük boyutlardaki problem setlerinden problemler alınarak bir problemler seti oluşturulmuştur. Bu problemlerin her biri belirtilen parametre seviyelerinde oluşturulan deney tasarımı ile karşılaştırılmıştır. Her bir problem seti beşer kere koşturulmuştur. Koşturma sonucunda elde edilen sonuçların ortalaması alınmıştır. Parametrelerin birbirine olan üstünlüğü bu ortalama değerlerin birbirine olan üstünlüğü ile belirlenmiştir. Sonuç olarak Tavlama Benzetimi algoritması için en iyi parametre değerleri ilk sıcaklık için (T_{ilk}) 500, soğutma oranı için (α) 0.99, iterasyon sayısı için (M) 20 olarak belirlenmiştir.

7.3.2. Genetik algoritma parametre seçimi

Genetik Algoritma parametreleri Çizelge 7.6.'da gösterilmiştir.

Çizelge 7.6. Genetik algoritma parametre seçimi

Parametre	Tanımı	Seviye	Seçilen Değer
POPSIZE	Popülasyon Boyutu	10, 20, 40	20
MUTOL	Mutasyon Oranı	0.01, 0.05, 0.1	0.05
CAPOL	Çaprazlama Oranı	0.75, 0.85, 0.95	0.75
TRE	Benzerlik Eşik Değeri	0.70, 0.85, 0.95	0.95

Tavlama Benzetimi algoritması parametre analizi için kullanılan problemler seti genetik algoritma parametrelerinin belirlenmesi için de kullanılmıştır. Problemler beşer kere koşturulmuş ve elde edilen sonuçların ortalaması alınarak parametre değerleri seçilmiştir. Bu bağlamda elde edilen en iyi sonuçlar; popülasyon boyutu (POPSIZE) için 20, mutasyon oranı (MUTOL) için 0,05, çaprazlama oranı (CAPOL) için 0,75 ve benzerlik eşik değeri (TRE) için 95 seçildiğinde gözlemlenmiştir.

7.4. Pİ-MHDP/SB için Geçerli Bir Alt Sınır

Optimal çözümü kabul edilebilir zamanlarda elde edilemeyen problemler için önerilen metotların performansı hesaplanan alt sınır değerleri ile değerlendirilebilmektedir. En küçükleme problemleri için bir alt sınır hesaplanırken, problemin çözümünü zorlaştıran kısıt ya da kısıtlar dikkate alınmaz. Böylece, problemin amaç fonksiyonu değeri değişmez ya da iyileşir. Bir diğer ifadeyle, Pİ-MHDP/SB için dikkate alınmayan kısıt (kısıtlar) ile elde edilen amaç fonksiyonu değeri orijinal problemin amaç fonksiyonu değerinden daha büyük olamaz. Bu durum, ihmal edilen kısıt ile elde edilen sonucun, orijinal problem için bir alt sınır değeridir.

MHDP literatürü incelendiğinde hatta bulunan toplam istasyon sayısının minimizasyonu için Scholl ve diğerleri (2013) tarafından BMHDP/SB için Andres ve diğerleri (2008) tarafından geliştirilen Pastor ve diğerleri (2010) tarafından düzeltilen alt sınır çalışmasının temel alınarak Andres ve diğerleri (2008)'nin çalışmasında göz önünde bulundurulmayan ileri ve geri hazırlık zamanları dikkate alınarak Eş. 7.4'te sunulan bir alt sınır (AS-big) hesaplanmıştır. AS-big BMHDP'nde istasyon sayısının bulunması için kullanılan alt sınır hesabını (AS-b) temel alınarak hesaplanmıştır. Ek olarak U tipi MHDP/SB için Şahin ve Kellegöz tarafından (2016) aşağıda Eş. 7.5 'te formülasyonu sunulan AS-bu geliştirilmiştir. AS-bu temelde BMHD-Tip 2 probleminin alt sınırına benzese de ileri ve geri hazırlık zamanlarını, formülasyondaki WUT değişkeni ile hesaplamaktadır. WUT değişkeni hesaplanırken, incelenen montaj hattında görev sayısı kadar hazırlık zamanı (ileri + geri) olacağını öngörülmüştür. Buna bağlı olarak, görevlere ait ileri ve geri hazırlık zamanı matrisleri öncelik ilişkileri göz önünde bulundurularak tek bir matris üzerinde birleştirilmiştir. Bir atama algoritması olan Macar yöntemiyle görev sayısı kadar hazırlık zamanı toplamaları en az olacak şekilde elde edilmektedir.

$$AS - b = \left\lceil \frac{\sum_{i \in I} t_i}{CT} \right\rceil \quad (7.3)$$

$$AS - big = \left\lceil \frac{\sum_{i \in I} t_i + \tau_{toplaml}^{nt-m} + \mu_{toplaml}^m}{CT} \right\rceil \quad (7.4)$$

$$AS - bu = \left\lceil \frac{\sum_{i \in I} t_i + WUT}{CT} \right\rceil \quad (7.5)$$

Eş.7.3’ de formülasyonu sunulan AS-b incelendiğinde bu tez çalışmasındaki Pİ-MHDP/SB için yetersiz kaldığı görülmektedir. AS-b’nin Pİ-MHDP’de hem işçi sayısı (AS-pi) hem de istasyon sayısı (AS-ps) için uyarlanması aşağıda sunulan Eş. 7.6 ve Eş. 7.7’ de verilmiştir (Kellegöz ve Toklu, 2012).

$$AS - pi = \left\lceil \frac{\sum_{i \in I} t_i}{CT} \right\rceil \quad (7.6)$$

$$AS - ps = \left\lceil \frac{\sum_{i \in I} t_i}{CT \times M_{\max}} \right\rceil \quad (7.7)$$

AS-pi ve AS-ps formülasyonları ile bu tez kapsamında çalışılan Pİ-MHDP/SB için alt sınır değerleri elde edilebilmektedir. Ancak, incelenen problem için ileri ve geri hazırlık zamanlarını göz önünde bulunduran bir alt sınır hesaplamasına ihtiyaç duyulmuştur. Bu sebeple, Kellegöz ve Toklu (2012) tarafından önerilen alt sınır hesaplama yöntemi ile Şahin ve Kellegöz (2016) tarafından önerilen alt sınır yöntemi bütünleştirilmiştir. Hatta kullanılan toplam işçi sayısı ve hatta açılan toplam istasyon sayısı için alt sınır değerleri sırası ile Eş. 7.8 ve Eş. 7.9’daki eşitlikler ile ifade edilmiştir. IGS miktarı Şahin ve Kellegöz (2016) çalışmasındaki WUT’ un elde edilmesi için kullanılan yöntem olan Macar Yöntemi ile elde edilmiştir.

$$AS - pis = \left\lceil \frac{\sum_{i \in I} t_i + IGS}{CT} \right\rceil \quad (7.8)$$

$$AS - pss = \left\lceil \frac{\sum_{i \in I} t_i + IGS}{CT \times M_{\max}} \right\rceil \quad (7.9)$$

7.5. Hesaplama Sonuçları ve Değerlendirme

Çizelge 7.7' den Çizelge 7.18.'e kadar olan tüm çizelgelerde problem için sunulan meta sezgisel algoritmaların etkinliği gösterilmiştir. Önerilen iki farklı meta sezgisel algoritmanın etkinliği ilk olarak matematiksel model sonuçları (ÖMAT-1) ile karşılaştırılmıştır. Çizelgelerde ilk kolon problem adını ve problemin kaç görev içerdiğini, ikinci kolon çevrim zamanlarını, diğer kolonlar sunulan çözüm yöntemlerinin (ÖMAT-1, TB, ve GA) koşturulması sonucu elde edilen ortalama ve en iyi işçi sayısı (NW), ortalama ve en iyi istasyon sayısı (NS) ve bu koşturmalar için geçen CPU zamanını ifade etmektedir. GA ve TB çözüm yaklaşımları her bir veri seti için 5 kere koşturulmuştur. Koşturma sonucu elde edilen ortalama toplam işçi ve istasyon sayısı ile en iyi toplam işçi sayısı ve toplam istasyon sayısı raporlanmıştır. ÖMAT-1, TB ve GA farklı yoğunlukta hazırlık zamanlarında amaç fonksiyonu (hattaki toplam işçi sayısı ve toplam istasyon sayısı) ve CPU zamanı performans ölçütleri olarak alınıp karşılaştırılmıştır.

Çizelge 7.7., 7.8., 7.9., 7.10.,'ye bakıldığında önerilen meta sezgisel yaklaşımların küçük boyutlu problemlerde ÖMAT-1'in vermiş olduğu en iyi sonuçları yakaladığı görülmektedir. Ayrıca farklı yoğunluktaki sıra bağımlı hazırlık zamanlarının küçük boyutlu problemler için çözüm kalitesine bir etkisi bulunmamaktadır. Bu tabloya bakıldığında küçük boyutlu problemlerde ne Tavlama Benzetimi Algoritmasının Genetik Algoritmaya ne de Genetik Algoritmanın Tavlama Benzetimi algoritmasına CPU zamanı baz alındığında bir üstünlüğü bulunmamaktadır.

Çizelge 7.7. ÖMAT-1 –GA ve TB karşılaştırması-küçük boyut ($\alpha = 0.25$)

Problem(nt)	Çevrim zamanı	ÖMAT-1		GA			TB		
		NW/NS	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Bowman(8)	20	5/4	8,733	5,2/4,4	5/4	0,161	6,2/4,2	5/4	0,546
Jackson(11)	7	9/5	2598,2	10,4/6,2	9/5	0,317	10/6,6	9/5	0,421
	9	7/4	2535,6	7,6/4,4	7/4	0,146	8,6/4,8	7/4	0,157
	10	6/3	1030,8	6,4/4,4	6/3	0,275	7,4/4	6/3	0,418
	13	4/3	1467,1	4,8/4,2	4/3	0,159	5/5,2	4/3	0,398
	14	4/4*	3600	4,6/3,8	4/3	0,486	4,4/3,8	4/3	0,842
	21	3/2	1758,6	4,2/3,6	3/2	0,235	5,2/3,8	3/2	0,183
Jaeschke(9)	6	8/6	17,071	9,4/7,2	8/6	0,379	10,4/7,4	8/6	0,288
	7	8/6	17,980	8,8/7,8	8/6	0,321	9,6/8,2	8/6	0,404
	8	7/5	20,911	8,6/6,4	7/5	0,390	10,8/6,6	7/5	0,605
	10	6/4	441,6	6,8/5	6/4	0,328	8,2/5,8	6/4	0,571
	18	3/2	487,2	3,6/3,4	3/2	0,421	4,6/3,2	3/2	0,321
Mertens(7)	6	6/3	2,491	6,8/4,2	6/3	0,312	8,4/4,4	6/3	0,126
	7	6/3	3,556	6,4/3,2	6/3	0,328	6,6/3,2	6/3	0,229
	8	5/3	4,885	5,6/4,2	5/3	0,344	6,8/4,4	5/3	0,502
	10	4/3	11,016	4,4/4	4/3	0,359	5,2/4,2	4/3	0,370
	15	3/2	5,746	4/2,8	3/2	0,360	4,2/3	3/2	0,193
	18	2/1	3,370	3,2/2,2	2/1	0,391	3,8/2,6	2/1	0,248

* 3600 CPU saniyesi içinde bulunan en iyi değer

- 3600 CPU saniyesi içinde bir çözüm elde edilemedi

Çizelge 7.8. ÖMAT-1 –GA ve TB karşılaştırması-küçük boyut ($\alpha = 0.50$)

Problem (nt)	Çevrim zamanı	ÖMAT-1		NW/NS (Ortalama)	GA		NW/NS (Ortalama)	TB	
		NW/NS	CPU		NW/NS (En İyi)	CPU		NW/NS (En İyi)	CPU
Bowman(8)	20	5/4	6,941	5,4/4,4	5/4	0,125	5,8/6,4	5/4	0,429
Jackson (11)	7	9/5	2754,75	9,6/6,6	9/5	0,164	9,8/6,4	9/5	0,446
	9	7/4	3067,50	8,6/5	7/4	0,262	9,2/4,4	7/4	0,209
	10	6/3	1670,3	6,6/4,8	6/3	0,331	7,2/5	6/3	0,381
	13	4/3	1509,4	5,2/4,6	4/3	0,175	6,2/4,2	4/3	0,492
	14	-	3600	5,6/4,2	4/3	0,194	5,4/4,8	4/3	0,541
	21	3/3*	3600	4,2/3,2	3/2	0,327	4,8/3,8	3/2	0,354
Jaeschke(9)	6	8/6	19,21	10,6/7,4	8/6	0,416	9,8/7,2	8/6	0,309
	7	8/6	22,07	8,4/8	8/6	0,119	10,2/7,8	8/6	0,427
	8	7/5	31,14	7,6/6	7/5	0,221	9,6/6,2	7/5	0,617
	10	6/4	556,4	7,8/5,2	6/4	0,324	8,8/5,8	6/4	0,551
	18	3/2	678,7	4,6/4,4	3/2	0,186	5,6/3,4	3/2	0,724
Mertens(7)	6	6/3	4,17	7,8/4,4	6/3	0,319	8,4/4,2	6/3	0,229
	7	6/3	7,83	8,4/3,8	6/3	0,276	6,6/3,4	6/3	0,378
	8	5/3	6,71	6,8/4,2	5/3	0,295	7,8/4,2	5/3	0,514
	10	4/3	9,16	6,4/5	4/3	0,317	6,2/3,4	4/3	0,343
	15	3/2	6,79	4,4/3	3/2	0,268	5,2/3,2	3/2	0,186
	18	2/1	5,62	3/2,8	2/1	0,329	4,2/2,6	2/1	0,308

* 3600 CPU saniyesi içinde bulunan en iyi değer

- 3600 CPU saniyesi içinde bir çözüm elde edilemedi

Çizelge 7.9. ÖMAT-1 –GA ve TB karşılaştırması-küçük boyut ($\alpha = 0.75$)

Problem(nt)	Çevrim zamanı	ÖMAT-1		NW/NS (Ortalama)	GA		NW/NS (Ortalama)	TB	
		NW/NS	CPU		NW/NS (En İyi)	CPU		NW/NS (En İyi)	CPU
Bowman(8)	20	5/4	12,627	5,6/4,6	5/4	0,261	6/6,2	5/4	0,419
Jackson(11)	7	10/6*	3600	10/5,8	9/5	0,192	10/6	9/5	0,436
	9	7/5*	3600	7,8/5,2	7/4	0,165	8,8/4,6	7/4	0,472
	10	6/3	1592,3	6,2/4	6/3	0,317	8,2/4,8	6/3	0,338
	13	4/3	1763,7	5,4/4,4	4/3	0,206	6,2/3,8	4/3	0,390
	14	-	3600	4,6/4	4/3	0,186	5,4/4,4	4/3	0,843
	21	-	3600	5,2/3,4	3/2	0,362	4/4,2	3/2	0,561
Jaeschke(9)	6	8/6	23,75	9,6/7	8/6	0,347	8,8/7	8/6	0,419
	7	8/6	29,14	8,8/7,8	8/6	0,409	10/7,2	8/6	0,397
	8	7/5	34,16	7,4/5,8	7/5	0,428	9,2/6	7/5	0,664
	10	6/4	711,40	6,8/5	6/4	0,513	7,8/5,2	6/4	0,917
	18	3/2	617,72	4,2/3,6	3/2	0,460	4,6/3	3/2	0,671
Mertens(7)	6	6/3	6,31	7,2/4,2	6/3	0,329	8,4/4,4	6/3	0,446
	7	6/3	5,92	7,4/3,4	6/3	0,190	6,2/3,6	6/3	0,320
	8	5/3	7,36	6,2/4	5/3	0,229	7,6/4	5/3	0,660
	10	4/3	11,19	5,2/4,6	4/3	0,196	6,8/4,2	4/3	0,426
	15	3/2	10,78	4,8/3,2	3/2	0,231	4,4/3,6	3/2	0,209
	18	2/1	8,93	3,2/2	2/1	0,285	4,2/2,2	2/1	0,310

* 3600 CPU saniyesi içinde bulunan en iyi değer

- 3600 CPU saniyesi içinde bir çözüm elde edilemedi

Çizelge 7.10. ÖMAT-1 –GA ve TB karşılaştırması-küçük boyut ($\alpha = 1.00$)

Problem(nt)	Çevrim zamanı	ÖMAT-1		NW/NS (Ortalama)	GA		NW/NS (Ortalama)	TB	
		NW/NS	CPU		NW/NS (En İyi)	CPU		NW/NS (En İyi)	CPU
Bowman(8)	20	5/4	12,627	6/4,4	5/4	0,199	5,8/5,2	5/4	0,472
Jackson(11)	7	10/6*	3600	10,8/6,2	9/5	0,257	11/6	9/5	0,446
	9	8/5*	3600	7,8/4,4	7/4	0,306	8/4,2	7/4	0,392
	10	6/4*	3600	7,2/4,4	6/3	0,429	6,8/4,2	6/3	0,346
	13	5/4*	3600	5,4/3,8	4/3	0,254	5,2/4	4/3	0,447
	14	-	3600	4,8/3,2	4/3	0,305	5,8/4	4/3	0,745
	21	-	3600	4,2/2,8	3/2	0,471	3,8/3	3/2	0,606
Jaeschke(9)	6	8/6	23,75	9,4/7,2	8/6	0,289	9/7,2	8/6	0,504
	7	8/6	29,14	8,4/7	8/6	0,330	8,8/6,2	8/6	1,674
	8	7/5	34,16	7,8/6,2	7/5	0,527	8,6/5,8	7/5	0,668
	10	6/4	711,40	6,6/6	6/4	0,438	6,4/4,4	6/4	1,207
	18	3/2	617,72	4,2/2,6	3/2	0,379	3,6/2,2	3/2	0,672
Mertens(7)	6	6/3	6,31	6,8/3,4	6/3	0,419	6,8/4	6/3	0,329
	7	6/3	5,92	7/3,2	6/3	0,351	7/3,4	6/3	0,516
	8	5/3	7,36	6/4,2	5/3	0,197	5,2/3,6	5/3	0,746
	10	4/3	11,19	4,4/3,4	4/3	0,226	4,4/3,6	4/3	0,394
	15	3/2	10,78	4,2/3	3/2	0,341	3,4/2,6	3/2	0,223
	18	2/1	8,93	3/1,8	2/1	0,283	3/2	2/1	0,352

* 3600 CPU saniyesi içinde bulunan en iyi değer

- 3600 CPU saniyesi içinde bir çözüm elde edilemedi

Çizelge 7.11., 7.12, 7.13 ve 7.14. orta boyutlu problemlerde önerilen yaklaşımların sonuçlarını özetlemektedir. Karşılaştırma elde edilen toplam işçi sayısı ile toplam istasyon sayısı üzerinden yapılmaktadır. CPU zamanı her iki algoritmada aynı kullanılmıştır. İlk olarak TB koşturulmuş ve TB'nin kullandığı CPU zamanı GA' nın koşturulması için kullanılmıştır. Bu tabloya bakıldığında, önerilen Genetik Algoritma nispeten önerilen Tavlama Benzetimi Algoritmasından daha iyi sonuç vermektedir. Ek olarak farklı yoğunluktaki hazırlık zamanlarının algoritmanın kalitesine etki ettiği görülmektedir. Yoğunluk arttıkça elde bazı problem setlerinde çözüm kalitesi (hattaki toplam işçi sayısı ve toplam istasyon sayısı) düşmektedir.

Çizelge 7.11. Tavlama benzetimi ve genetik algoritma karşılaştırması-orta boyut
($\alpha = 0.25$)

Problem(nt)	Çevrim zamanı	Alt Sınır		Tavlama Benzetimi			Genetik Algoritma		
		AS-pis	AS-pss	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Buxey(29)	27	12	4	14/8,8	13/8	1,897	13/9	12/8	1,897
	30	11	4	14/7,6	13/7	2,102	12,8/8	12/7	2,102
	33	10	4	14,2/8	13/7	2,540	13/7,8	12/7	2,540
	36	9	3	12,2/7,2	11/6	2,114	12/6,2	11/6	2,114
	41	8	3	11,8/6,2	11/6	1,075	11,6/6,4	10/6	1,075
	47	7	3	11,4/6,2	10/5	3,010	10/6	9/5	3,010
	54	6	2	8,2/4,6	7/4	1,006	8,2/4,4	7/4	1,006
Heskia(28)	138	8	3	11/6	10/5	1,578	9,8/5,6	9/5	1,578
	205	5	2	8,2/4,6	7/4	1,694	8,8/4,4	7/4	1,694
	216	5	2	8/4,4	7/4	1,413	8/5	7/4	1,413
	256	4	2	6/3,6	5/3	2,129	6/4,2	5/3	2,129
	342	3	1	4,4/3,4	4/3	1,272	4,4/3,6	3/3	1,272
Lutz1(32)	1414	10	4	13,2/9,8	12/9	1,042	13,2/9,8	12/9	1,042
	1572	9	3	13,6/10	13/9	1,711	12,8/10	12/9	1,711
	1768	8	3	13/9	12/8	1,687	12/9,2	11/8	1,687
	2020	7	3	11,2/7	10/6	1,533	9,8/7	9/6	1,533
	2357	6	2	8/5,6	7/5	2,130	8/6	7/5	2,130
	2828	5	2	7,8/6	7/5	2,569	6,8/5,2	6/5	2,569
Sawyer(30)	25	13	5	16,4/8,4	15/8	3,974	16/8,4	15/8	3,974
	27	12	4	15/9,2	14/8	1,036	14/9	13/8	1,036
	30	11	4	13,6/8	12/7	1,794	14/8	12/7	1,794
	33	10	4	13,4/8,2	12/7	2,656	12,2/8,2	11/7	2,656
	36	9	3	11/6,2	10/6	3,124	11/6,4	10/6	3,124
	41	8	3	10,8/5,6	10/4	1,334	11/5	9/4	1,334
	47	7	3	10,4/5	9/4	2,462	10/4,4	9/4	2,462
	54	6	2	9/5	7/4	2,247	8,8/5,2	7/4	2,247
	75	5	2	5,8/5	5/4	2,083	6/4	5/3	2,083

Çizelge 7.12. Tavlama benzetimi ve genetik algoritma karşılaştırması-orta boyut
($\alpha = 0.50$)

Problem (nt)	Çevrim zamanı	Alt Sınır		Tavlama Benzetimi			Genetik Algoritma		
		AS- pis	AS-pss	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Buxey (29)	27	12	4	15/9	14/8	2,226	14/8,8	13/8	2,226
	30	11	4	14/8	13/7	1,753	13,2/7,4	12/7	1,753
	33	10	4	13,8/8,2	13/7	1,476	13/8	12/7	1,476
	36	9	3	13/6,4	12/6	2,338	13,4/6,8	12/6	2,338
	41	8	3	12/6,8	11/6	3,745	11/6,2	10/6	3,745
	47	7	3	12,2/6	11/5	3,411	11,2/6	10/5	3,411
	54	6	2	8/4,6	7/4	2,128	8,2/4,6	7/4	2,128
Heskia (28)	138	8	3	10,8/5,6	10/5	1,346	9,8/6	9/5	1,346
	205	5	2	8,4/5	8/4	1,798	8,6/5,2	8/4	1,798
	216	5	2	7,8/5,2	7/4	2,034	8/5	7/4	2,034
	256	4	2	6/3,4	5/3	3,751	5,2/3,4	5/3	3,751
	342	3	1	4,4/3,6	4/3	2,119	3,4/3,6	3/3	2,119
Lutz1 (32)	1414	10	4	14/9,4	13/9	2,027	13,8/9,2	13/9	2,027
	1572	9	3	14,6/10	13/9	1,335	13/10	12/9	1,335
	1768	8	3	12,8/9	12/8	1,358	12/8,2	11/8	1,358
	2020	7	3	12/7	11/6	2,449	10,4/6,2	10/6	2,449
	2357	6	2	9/5,6	8/5	3,765	7,8/5,2	7/5	3,765
	2828	5	2	8,2/6	7/5	4,967	7,6/5,4	6/5	4,967
Sawyer (30)	25	13	5	17,6/9	16/8	4,705	16,2/8,8	15/8	4,705
	27	12	4	16,2/9	15/8	3,429	15,2/8,4	14/8	3,429
	30	11	4	13/7,2	12/7	1,291	12,4/7,2	12/7	1,291
	33	10	4	14/8	13/7	4,751	12/7,2	11/7	4,751
	36	9	3	10,8/7	10/6	3,069	11/6,2	10/6	3,069
	41	8	3	11/5	10/4	2,116	11/4,2	10/4	2,116
	47	7	3	11,2/5,2	10/4	2,453	9,8/5	9/4	2,453
	54	6	2	8/5,2	7/4	1,994	8/4,2	7/4	1,994
	75	5	2	6/4,2	5/4	1,335	5,2/4	5/3	1,335

Çizelge 7.13. Tavlama benzetimi ve genetik algoritma karşılaştırması-orta boyut
($\alpha = 0.75$)

Problem(nt)	Çevrim zamanı	Alt Sınır		Tavlama Benzetimi			Genetik Algoritma		
		AS-pis	AS-pss	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Buxey(29)	27	12	4	15,2/8,4	14/8	1,562	14/8,4	13/8	1,562
	30	11	4	14,8/7,2	14/7	2,113	13/8	12/7	2,113
	33	10	4	13,4/8	13/7	2,451	13,2/7,6	12/7	2,451
	36	9	3	13,6/6,8	12/6	3,157	12,8/6,4	12/6	3,157
	41	8	3	12,4/7	11/6	3,045	11,6/7	11/6	3,045
	47	7	3	12/6	11/5	2,336	10,6/5,4	10/5	2,336
	54	6	2	8,8/5	8/4	4,687	9,2/4,4	8/4	4,687
Heskia(28)	138	8	3	11,2/5,4	10/5	2,691	9,6/5,8	9/5	2,691
	205	5	2	9,2/4,6	8/4	3,068	8/4,6	7/4	3,068
	216	5	2	8,4/4,8	8/4	3,145	7,8/4,8	7/4	3,145
	256	4	2	7/4,6	6/4	4,088	7/5	6/4	4,088
	342	3	1	5,2/3,4	4/3	2,046	4,6/3,4	4/3	2,046
Lutz1(32)	1414	10	4	15,4/9,2	14/9	3,117	14/9,2	13/9	3,117
	1572	9	3	14,4/10	13/9	2,996	13,8/9,4	13/9	2,996
	1768	8	3	13,4/9	12/8	1,934	12/9	11/8	1,934
	2020	7	3	13,2/7,2	12/7	1,672	10,2/6,4	9/6	1,672
	2357	6	2	8,4/5,6	8/5	1,537	8,8/6	8/5	1,537
	2828	5	2	9,2/6,4	8/6	2,734	8/5,4	7/5	2,734
Sawyer(30)	25	13	5	18/9	16/8	3,486	16,4/8,8	15/8	3,486
	27	12	4	17,2/8,8	16/8	3,421	15/9	14/8	3,421
	30	11	4	14/8,6	13/8	4,756	12,4/7,6	12/7	4,756
	33	10	4	13/7,8	12/7	4,067	12/7,6	11/7	4,067
	36	9	3	11/6,2	10/6	4,875	11,2/6,4	10/6	4,875
	41	8	3	11/4,4	10/4	3,450	10/5,6	9/4	3,450
	47	7	3	10,2/4,6	9/4	4,721	10,6/5	10/4	4,721
	54	6	2	8,2/5	7/4	4,306	8/4,2	7/4	4,306
	75	5	2	5,4/4,6	5/4	3,590	6,8/5	6/4	3,590

Çizelge 7.14. Tavlama benzetimi ve genetik algoritma karşılaştırması-orta boyut
($\alpha = 1.00$)

Problem (nt)	Çevrim zamanı	Alt Sınır		Tavlama Benzetimi			Genetik Algoritma		
		AS-pis	AS-pss	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Buxey (29)	27	12	4	15,2/8,6	14/8	1,892	14,8/9	13/8	1,562
	30	11	4	15,4/8	14/7	1,364	12,4/7,2	12/7	2,113
	33	10	4	13,8/7,6	13/7	2,627	12,8/7,6	12/7	2,451
	36	9	3	12,6/7	12/6	2,943	13/6,2	12/6	3,157
	41	8	3	14/7	13/6	2,874	13,2/6,4	12/6	3,045
	47	7	3	12,4/6,2	11/5	3,411	11/6,2	10/5	2,336
	54	6	2	9,2/5	9/4	5,791	10/5	9/4	4,687
Heskia (28)	138	8	3	10,8/5,4	10/5	3,419	9,8/5,6	9/5	2,691
	205	5	2	9,2/4,2	8/4	2,576	7,8/4,6	7/4	3,068
	216	5	2	9/4,6	8/4	3,751	8/5	7/4	3,145
	256	4	2	6,8/4,6	6/4	3,624	7,2/5,2	6/4	4,088
	342	3	1	4,8/4,6	4/4	2,443	4,4/3,2	4/3	2,046
Lutz1 (32)	1414	10	4	15,6/9,4	14/9	4,820	14/10	13/9	3,117
	1572	9	3	14,6/9,4	13/9	1,786	14,2/9,4	13/9	2,996
	1768	8	3	14,6/8,8	13/8	2,452	13,2/8,8	12/8	1,934
	2020	7	3	12,4/7,8	12/7	2,347	11,6/6,4	10/6	1,672
	2357	6	2	10/6	9/5	1,961	9/5,8	8/5	1,537
	2828	5	2	9/6,2	8/6	3,426	7,8/5,6	7/5	2,734
Sawyer (30)	25	13	5	17/8,8	16/8	4,321	16/8,4	15/8	3,486
	27	12	4	17,8/9	16/8	2,904	15/8,2	14/8	3,421
	30	11	4	14/8,2	13/8	4,137	13/7,2	12/7	4,756
	33	10	4	13/8	12/7	3,870	12,4/8	11/7	4,067
	36	9	3	12/7,8	11/7	4,849	12/6,2	11/6	4,875
	41	8	3	11/4,6	10/4	4,155	10/4,4	9/4	3,450
	47	7	3	10/4,6	9/4	5,307	11/4,8	10/4	4,721
	54	6	2	8/4,6	7/4	3,618	7,8/5	7/4	4,306
	75	5	2	7/5	6/4	4,228	6,6/5	6/4	3,590

Çizelge 7.16., 7.17., 7.18., ve 7.19. ise büyük boyutlu problemlerin farklı yoğunluktaki hazırlık zamanları için koşturma sonuçlarını vermektedir. Algoritmalar kendi aralarında karşılaştırılacak olursa her ne kadar Tavlama Benzetimi Algoritması Genetik Algoritma ile bazı veri setlerinde aynı sonucu yakalamış olsa bile Genetik algoritma ile elde edilen sonuçların daha iyi olduğu görülmektedir. Buna ek olarak orta boyutlu problemlerde karşılaşıldığı gibi farklı hazırlık zamanı yoğunluklarının büyük boyutlu problem setlerindeki çözümler üzerinde olumsuz bir etkisi olduğu gözlemlenmiştir.

Çizelge 7.15. Tavlama benzetimi ve genetik algoritma karşılaştırması-büyük boyut
($\alpha = 0.25$)

Problem(nt)	Çevrim zamanı	Alt Sınır		Tavlama Benzetimi			Genetik Algoritma		
		AS-pis	AS-pss	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Arcus(83)	3786	20	7	27/15,2	25/14	42,153	26,8/15	25/14	42,153
	3985	19	7	26,4/14,4	25/14	38,642	24,2/14	23/13	38,642
	4206	18	6	25,6/14	24/13	45,782	24,2/13	23/12	45,782
Arcus(111)	5755	27	9	33,4/16,4	32/16	58,961	31/16	30/15	58,961
	5785	26	9	33/15,6	32/15	61,783	33/15,8	32/15	61,783
	6016	25	9	36,2/15	35/14	55,447	32/14	31/13	55,447
Bartholdi (148)	84	51	17	57/21	56/20	67,254	55/20,4	54/19	67,254
	85	50	17	56,8	56/19	72,795	56,2/20,4	54/19	72,795
	87	49	20	20	54/19	62,573	54,2/20,4	53/19	62,573
Mukherje (94)	176	24	8	30,2/15,8	29/15	46,542	27,8/14,8	27/14	46,542
	192	22	8	28,4/15,2	27/14	39,284	26,8/15	26/14	39,284
	211	20	7	26/14	25/12	51,478	25,4/13,4	24/12	51,478

Çizelge 7.16. Tavlama benzetimi ve genetik algoritma karşılaştırması-büyük boyut
($\alpha = 0.50$)

Problem(nt)	Çevrim zamanı	Alt Sınır		Tavlama Benzetimi			Genetik Algoritma		
		AS-pis	AS-pss	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Arcus(83)	3786	20	7	27,8/16	26/15	35,614	26,4/16,2	25/15	35,614
	3985	19	7	26/14,8	25/14	44,238	25,4/13,8	24/13	44,238
	4206	18	6	25/14	24/13	38,462	24,4/13	23/12	38,462
Arcus(111)	5755	27	9	33/16,4	32/16	46,781	32,2/16	31/15	46,781
	5785	26	9	34/16,2	33/15	42,553	33,4/16	32/15	42,553
	6016	25	9	36,2/15,8	35/15	32,462	33,6/15	32/14	32,462
Bartholdi (148)	84	51	17	59,8/22	57/20	73,540	58/21,2	56/19	73,540
	85	50	17	57,4/20,4	56/19	78,634	55,4/20,2	54/19	78,634
	87	49	20	55,4/20,8	54/19	65,721	56,4/21,6	54/19	65,721
Mukherje (94)	176	24	8	30,8/16,2	29/15	41,734	30,6/16,4	28/15	41,734
	192	22	8	28,4/16,4	27/15	60,083	27,8/15,2	26/14	60,083
	211	20	7	27,4/13	26/12	58,325	26,4/15	25/14	58,325

Çizelge 7.17. Tavlama benzetimi ve genetik algoritma karşılaştırması-büyük boyut
($\alpha = 0.75$)

Problem(nt)	Çevrim zamanı	Alt Sınır		Tavlama Benzetimi			Genetik Algoritma		
		AS-pis	AS-pss	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Arcus(83)	3786	20	7	27,2/16	26/15	39,548	26,8/16	25/15	39,548
	3985	19	7	26/15	25/14	45,231	25,2/14	24/13	45,231
	4206	18	6	24,8/14,2	24/13	46,224	24,8/13	24/12	46,224
Arcus(111)	5755	27	9	33,8/17	32/16	64,069	32,4/16,2	31/15	64,069
	5785	26	9	34/16,8	33/16	62,387	33,4/16,8	32/15	62,387
	6016	25	9	36,8/16,2	35/15	53,746	34/15,4	32/14	53,746
Bartholdi (148)	84	51	17	59/21	57/20	59,796	58,8/21,6	56/20	59,796
	85	50	17	58,2/21,2	57/19	67,892	55,6/20,2	54/19	67,892
	87	49	20	55,6/20,8	54/19	56,734	54,6/20,8	53/19	56,734
Mukherje (94)	176	24	8	31,2/16,8	29/15	40,643	29,8/16,2	28/15	40,643
	192	22	8	28,4/16,4	27/15	54,226	27,2/15,4	26/14	54,226
	211	20	7	26,8/13	26/12	49,577	26,4/14,2	25/13	49,577

Çizelge 7.18. Tavlama benzetimi ve genetik algoritma karşılaştırması-büyük boyut
($\alpha = 1.00$)

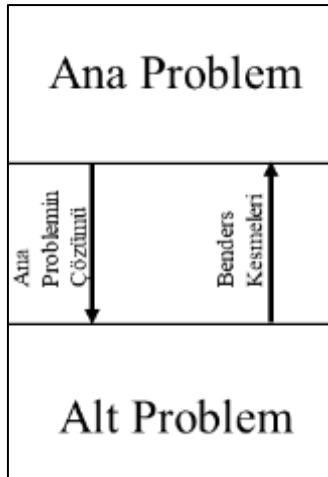
Problem(nt)	Çevrim zamanı	Alt Sınır		Tavlama Benzetimi			Genetik Algoritma		
		AS-pis	AS-pss	NW/NS (Ortalama)	NW/NS (En İyi)	CPU	NW/NS (Ortalama)	NW/NS (En İyi)	CPU
Arcus(83)	3786	20	7	26,4/15	25/14	36,752	27/15,6	25/14	36,752
	3985	19	7	27,2/15,4	25/14	43,593	24,2/13,8	23/13	43,593
	4206	18	6	25,8/14	24/13	40,769	24,8/14	23/12	40,769
Arcus(111)	5755	27	9	33,8/17,2	32/16	61,951	31,6/16,2	30/15	61,951
	5785	26	9	33,4/16,2	32/15	56,478	33,8/16,2	32/15	56,478
	6016	25	9	36,2/15,4	35/14	42,294	32,6/14	31/13	42,294
Bartholdi (148)	84	51	17	59,4/21,4	57/20	71,382	56,8/21,2	55/20	71,382
	85	50	17	56,8/20,2	56/19	63,667	56,8/20,1	55/19	63,667
	87	49	20	55,8/20,2	55/19	68,375	55,8/21,6	54/19	68,375
Mukherje (94)	176	24	8	31,2/16,4	29/15	44,242	28,8/15,4	27/14	44,242
	192	22	8	30,4/16	28/15	50,854	27,6/15	26/14	50,854
	211	20	7	27,8/14,6	26/14	45,337	26,2/14,4	25/13	45,337

8. MONTAJ HATTI DENGELEME PROBLEMİ İÇİN BENDERS AYRIŞTIRMASI

Bu bölümde, Pİ-MHDP/SB için geliştirilen bir ayrıştırma metodu olan Benders Ayrıştırması detayları ile birlikte anlatılmıştır. İlk olarak Benders Ayrıştırması hakkında temel bilgiler verilmiştir. Daha sonra bu tez çalışması kapsamındaki Pİ-MHDP/SB için Benders Ayrıştırmasının özel bir durumu olan mantık tabanlı Benders ayrıştırması algoritması adımları verilmiştir. Algoritmanın tasarlanması için ayrıştırmaya uygun bir yapısı olan yeni bir matematiksel programlama modeli (ÖMAT-2) önerilmiştir. ÖMAT-2 ye bağlı olarak gevşetilmiş ana problem (GAP), alt problemler (AP) ve bu ikisi arasında iletişimi sağlayan Benders kesmeleri detaylı bir şekilde anlatılmıştır. Son olarak önerilen algoritmanın problem setlerindeki etkinliği değerlendirilmiştir.

8.1. Benders Ayrıştırmasının Genel Yapısı

Benders ayrıştırması, optimizasyon problemlerini çözmek için kullanılan yöntemlerden biridir. Yöntem, 1962 yılında J. F. Benders tarafından önerilmiştir. Yaklaşım “*birinin hatalarından öğrenme*” kavramına dayanarak problemi akıllı bir şekilde bölüştürerek (ayrıştırarak) çözüm elde eder. Bu yaklaşım, karar değişkenlerinin birincil (karmaşık) ve ikincil kararlar şeklinde bölünebileceği problemlere çok uygundur. Bir diğer deyişle problem ana ve alt problemler olmak üzere ayrıştırılır ve çözüm için gerekli hesaplama yükü dağıtılmış olur.



Şekil 8.1. Benders ayrıştırması genel yapısı

Problem ana ve alt problemlere ayrıştırıldıktan sonra, Benders algoritması bu iki problem arasında yineleme yapar. Her bir problemi sırası ile çözer ve problemin en önemli olursuzluğunun nerede olduğunu aşama aşama öğrenmek için bu çözümleri kullanır. Benders ayrıştırma yönteminin problemler arası yinelemesinin şematik anlatımı Şekil 8.1.’de gösterilmektedir. Ayrıştırma sonrası elde edilen ana problem ikincil değişkenleri içermediğinden gevşetilmiş ana problem olarak ifade edilebilir. En küçükleme probleminde, gevşetilmiş ana problemin sonucu, optimal değer için bir alt sınır oluşturur. Gevşetilmiş ana problemin bu çözümü alt problemler için çözüm bulmak için kullanılır ve mümkünse alt problemler optimal çözülür. Eğer alt problem olursuz bir çözüme sahipse, bu bilgi gevşetilmiş ana probleme aktarılır ve ana problem bu bilgi ışığında tekrar çözdürülür. Bu bilgiye *Benders kesmeleri* denir. Benders kesmelerinin sadece uygun olmayan çözümü ana problemin çözüm uzayından çıkarmaya yönelik değil de, ayrıca benzer sebeplerle uygunsuz çözüm elde edecek diğer çözümleri de çözüm uzayından çıkarmaya yönelik tasarlanması tercih edilmektedir.

8.2. Geleneksel Benders Ayrıştırması

Benders tarafından önerilen ilk ayrıştırma yöntemi, ilk olarak karma tam sayılı doğrusal programlama problemlerine uygulanmıştır. Tam sayılı değişkenler sabitlenip, geriye kalan doğrusal değişkenler için doğrusal dualite teoreminden yararlanılarak kesmeler oluşturulur. Rahmaniani ve diğerleri (2017). Bu ayrıştırmanın genel bir minimizasyon formunda doğrusal programlama problemine uygulanışı aşağıdaki gibidir. Örnek Young (2017)’ in çalışması göz önünde bulundurularak hazırlanmıştır.

Bir problemin karar değişkenleri x ve y olsun, y zorlaştırıcı değişkenleri ifade etsin.

$$\text{Min } c^T x + d^T y \quad (8.1)$$

Kısıtlar

$$Ax + By \leq b \quad (8.2)$$

$$Ey \leq h \quad (8.3)$$

Yukarıdaki probleme Benders ayrıştırması uygulandığında iki basit problem elde edilmektedir. Ana problem y zorlaştırmacı değişkenlerine karar verirken alt problem x değişkenlerine karar vermektedir. Böylelikle örnek bir $y=\bar{y}$ çözümü için , bu bilgi alt probleme aşağıdaki gibi iletilmektedir.

$$\text{Min } c^T x + d^T \bar{y} \quad (8.4)$$

Kısıtlar

$$Ax \leq b - B\bar{y} \quad (8.5)$$

Doğrusal programlamanın dualite özelliği kullanılarak u dual değişkenleri ile alt problem aşağıdaki gibi ifade edilir. Alt problemin amaç fonksiyonunda y zorlaştırmacı değişkenleri bulunur ancak kısıtlar y değişkenlerini içermez. Çözüldüğünde ya optimalite ya da uygunsuz çözüm elde edilir.

$$\text{Max}(b - B\bar{y})u \quad (8.6)$$

Kısıtlar

$$A^T u \leq c \quad (8.7)$$

Eğer ana problemin çözümü uygun bir çözüm ise, alt probleme ait dual problem uç noktalarda optimaliteyi yakalar, ana problem çözümü uygunsuz ise dual alt problem sınırsızdır. Her iki durum için, güncel çözümü ana problemden çıkaracak bir Benders kesmesi formüle edilebilir.

Alt problemin çözümü sınırlı ve optimal ise, ana probleme aşağıdaki gibi Benders kesmeleri eklenebilir. V_j extreme noktayı göstermektedir.

$$z \geq (b - B\bar{y})V_j^{-T} + d^T \bar{y}, \quad j = 1, 2, \dots \quad (8.8)$$

Alt problemin çözümü sınırsız ise aşağıdaki gibi Benders kesmeleri eklenebilir. U_j uç rayı göstermektedir.

$$(b - By)u_i^{-T} \leq 0, \quad i = 1, 2, \dots \quad (8.9)$$

Kesmeler oluşturulduktan sonra, ana probleme eklenir ve yeni ana problem yeni y değişkenleri bulmak için aşağıdaki gibi düzenlenir.

$$\text{Min } z \quad (8.10)$$

Kısıtlar

$$Ey \leq h \quad (8.11)$$

$$z \geq (b - By)v_j^{-T} + d^T y, \quad j = 1, 2, \dots \quad (8.12)$$

$$(b - By)u_i^{-T} \leq 0, \quad i = 1, 2, \dots \quad (8.13)$$

Klasik yaklaşım daha geniş çaptaki optimizasyon problemlerinde kullanılabilmesi için Geoffrion (1972) tarafından genelleştirilmiştir. Bir diğer deyişle, alt problemlerin doğrusal program olarak modellenmesi zorunluluğu ortadan kalkmıştır. Doğrusal olmayan bir şekilde de formüle edilebilir. Daha sonra Hooker ve Ottoson (2003) tanımlanan alt problemin karma tam sayılı programlama modeli olarak kurulmasına izin veren Mantık tabanlı Benders ayrıştırmasını literatüre kazandırmışlardır. Benders ayrıştırması algoritmasını farklı problem türlerine uygulayabilmek için birçok genişletme metotları sunulmuştur. Farklı problem türlerinde uygulanabilen bu metotlar için Rahmaniani ve diğerleri (2017) çalışmasına bakılabilir.

8.3. Mantık Tabanlı Benders Ayrıştırması (MTBA)

Geleneksel Benders ayrıştırması, alt problem ya da problemlerin doğrusal modellenmesini beklemektedir. Ancak bu durum yaklaşımın başka kombinatoriyal problemlere özellikle doğrusal şekilde modellenmeleri pratikte uygun olmayan problemlere örneğin çizelgeleme problemlerine uygulanmasını kısıtlamaktadır.

Mantık tabanlı Benders ayrıştırması (MTBA) Hooker tarafından klasik Benders ayrıştırmasının bir genelleştirilmesi olarak sunulmuştur. Klasik yaklaşımdaki gibi kombinatoriyal optimizasyon problemlerini bir ana problem ve alt probleme bölüştürerek bir çözüm arar. Klasik yaklaşımdan farkı ise ayrıştırmanın farklı bileşenleri üzerinde yapısal bir kısıtlama yoktur. Elde edilen alt problemler her zaman doğrusal değişkenleri içermeyebilir. Alt probleme ait değişkenler tam sayılı ya da doğrusal olmayan değişkenler olabilir. Bu durumda Benders kesmeleri klasik doğrusal dualite ile elde edilemez. Bu düşünce ile optimizasyon literatürüne inference dual kavramını tanıttırmışlardır. MTBA, ana problem ile alt problem arasında iletişimi sağlar. Bu iletişim, ana problemin çözümü kullanılarak alt problemlerin oluşturulması sağlanır ve alt problemlerin çözümü ile ana probleme eklenecek kesmeler belirlenerek sağlanır. Aşağıdaki gibi formüle edilebilir.

$$\text{Min } f(x, y) \quad (8.14)$$

Kısıtlar

$$S(x, y) \quad (8.15)$$

$$x \in X \quad (8.16)$$

$$y \in Y \quad (8.17)$$

$S(x, y)$ x ve y değişkenlerine bağlı kısıt kümesidir. X ve Y x ve y değişkenlerine ait tanım kümelerini (domain) ifade eder. Eğer y değişkenlerinin değeri $\bar{y}$ olarak sabitlenirse, alt problem aşağıdaki gibi şekillenir.

$$\text{Min } f(x, \bar{y}) \quad (8.18)$$

Kısıtlar

$$S(x, \bar{y}) \quad (8.19)$$

$$x \in X \quad (8.20)$$

$S(x, \bar{y})$ kümesi, $y = \bar{y}$ olarak sabitlendiği kısıt kümesini göstermektedir. Eğer ana problemin amaç fonksiyonu z ile gösterilirse, Benders kesmeleri eklenmiş ana problem aşağıdaki gibidir. $y^1, y^2, \dots, y^{t-1}$ bir önceki ana problemin çözümlerini göstermektedir.

$$\text{Min } z \quad (8.21)$$

Kısıtlar

$$z \geq B_{y^i}(y), \quad i = 1, 2, \dots, t-1 \quad (8.22)$$

$$z \in Z \quad (8.23)$$

$$y \in Y \quad (8.24)$$

MTBA ana problem ile alt problemi yakınsayana kadar yinelemeli olarak çözer. Ana problem optimal çözülür ve üretilen çözüm alt problem formülasyonunda kullanılır. Daha sonra alt problemlerden Benders kesmeleri elde edilir. Eğer i . ana problem 1 den i ye kadar üretilen tüm Benders kısıtlarını sağlıyorsa süreç global optimum a ulaşmıştır. Bu gerçekleşmez ise ana problem tekrar çözülür ve yeni benders kesmeleri üretilir. Chu ve Xia (2004) çalışmalarında MTBA'sının belirli sayıdaki iterasyon sonunda global optimuma ulaşabilmesi öncelikle karar değişkenlerinin sonlu bir domaine sahip olması gerektiğini ve kesmelerin aşağıdaki özelliklere sahip olması gerektiğini öne sürmüşlerdir.

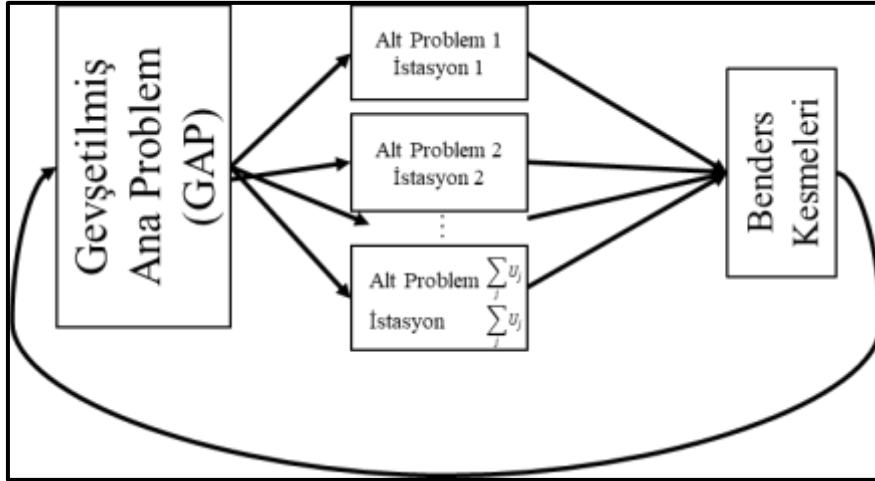
- 1) Alt problemin çözümü olurlu değil ise Benders kesmeleri bu çözümü hariç tutmalı,
- 2) Eklenen Benders kesmeleri tüm global olurlu çözümleri çözüm kümesinden çıkarmamalı.

Özetlemek gerekirse, MTBA hem gevşetme hem de kısıtlama kullanan bir ayrıştırma yöntemidir. Ana problem gevşetme problemidir. Çünkü bazı kısıtlar ana problemden ya kaldırılmıştır ya da gevşetilmiştir. Bu yüzden ana problemden elde edilen çözüm geçerli bir çözüm olmayacaktır. Çıkan çözümün geçerliliğinin mutlaka alt problemlerde test edilmesi gerekmektedir. Ayrıştırmanın bir diğer parçası olan alt problem ise kısıtlama problemidir. Ana problem ile zorlaştırmacı karar değişkenleri atanır ve alt problem bu bilgiyi kullanarak diğer değişkenleri elde eder.

8.4. Önerilen Mantık Tabanlı Benders Ayırıştırması Algoritması (ÖMTBA)

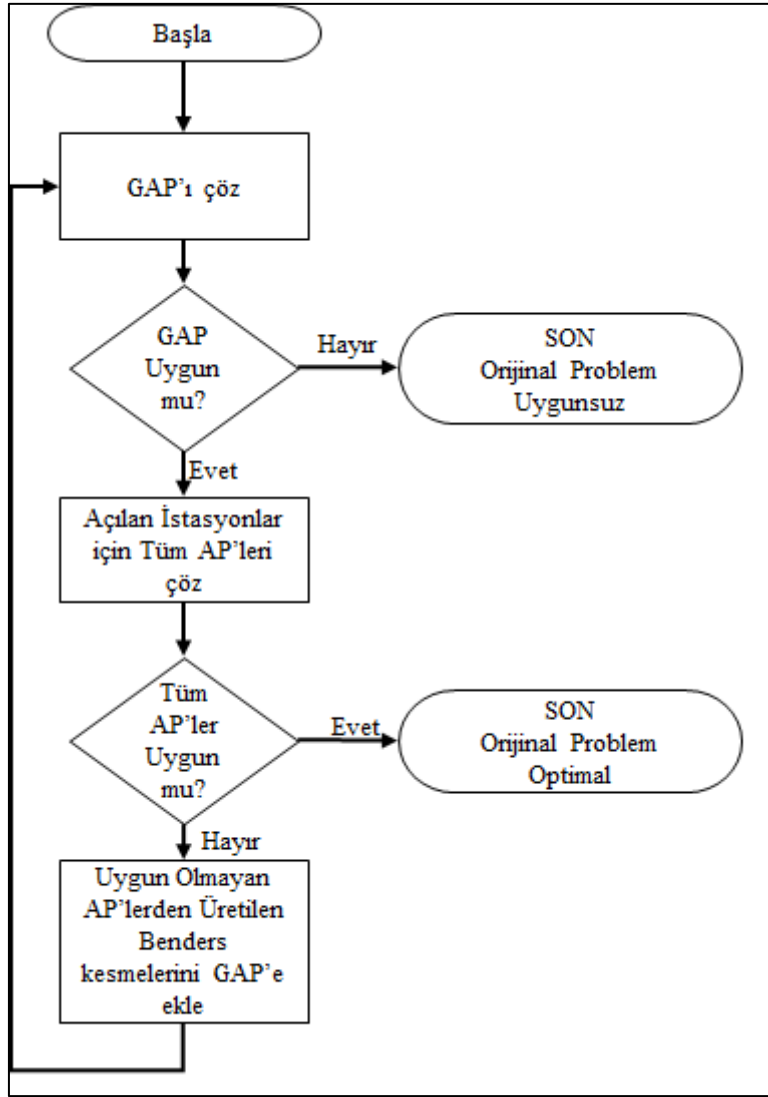
Bu bölümde sıra bağımlı hazırlık zamanlarını göz önünde bulunduran çok işçili montaj hattı dengeleme problemi için sunulan Mantık Tabanlı Benders Ayırıştırmasının (ÖMTBA) detayları anlatılmıştır. Ardından Benders Ayırıştırmasının parçaları olan gevşetilmiş ana problem ile alt problem formülasyonları yapılmıştır. Ana problem ile alt problem arasında iletişimi sağlayan Benders kesmeleri sunulmuştur. Son olarak algoritmanın problem setleri üzerindeki etkinliği tartışılmıştır.

İlk olarak, bu tez kapsamında çalışılan sıra bağımlı hazırlık zamanlarını içeren çok işçili montaj hattı dengeleme problemin nasıl ayrıştırılacağına karar vermek gerekmektedir. Problem’de verilmesi gereken 3 tip karar bulunmaktadır. Bunlar, görevlerin istasyonlara ve işçilere atanması, istasyondaki işçi konfigürasyonunun belirlenmesi ve görevlerin atandıkları istasyon ve işçilerde çizelgelenmesidir. Bu kararlardan ilk ikisi birincil kararlar olarak adlandırılmıştır. İkincil kararlar ise görevlerin çizelgelenmesidir. Yapılan bu ayırım ile beraber gevşetilmiş ana problem (GAP) geleneksel paralel çok işçili montaj hattı dengeleme problemlerine benzemektedir. Hatırlamak gerekirse paralel çok işçili montaj hattı problemleri görevler arasındaki sıra bağımlı hazırlık zamanlarını yok sayarak yalnızca görevlerin öncelik kısıtları altında istasyonlara ve işçilere atanması ile ilgilenmektedir. Gevşetilmiş ana problem çözüldüğü zaman; açılan istasyon sayısı kadar alt problem ortaya çıkmaktadır. Her bir alt problem kendisine atanan işçi ve görevler ile birlikte bir istasyonu ifade etmektedir. Alt problemlerin amacı, ilgili istasyona atanan görevleri, bu görevler hangi işçilere atanmış ise o işçinin toplam yükünü görevler arasındaki öncelik ilişkilerini de göz önünde bulundurarak en küçükleme. Daha öncede bahsedildiği gibi alt problemler asimetrik gezgin satıcı problemine benzemektedir. Alt problemler için en iyi çözüm ya da uygunsuz çözümler (istasyondaki herhangi bir işçinin yükünün belirlenen çevrim zamanından büyük olması) elde edildikten sonra bu bilgi ana probleme geri gönderilir. Bilgi aktarımı Benders kesmeleri ile yapılmaktadır. GAP ile AP arasındaki ilişki Şekil 8.2.’de görselleştirilmiştir. Burada U_j , istasyon j açıldı ise 1, değilse 0 değerini almaktadır. Bu değerler ana problemin çözümünden gelmektedir.



Şekil 8.2. ÖMTBA genel yapısı

Önerilen Benders algoritmasının akış şeması Şekil 8.3.'de sunulmaktadır. Algoritma ilk olarak Gevşetilmiş Ana Probleme ait girdi parametrelerinin oluşturulması ile başlar. Bu parametreler ile GAP optimal çözülür. Çözümün uygunluğu kontrol edilir. Uygun değil ise problem uygunsuzdur denir ve algoritma sonlanır. Eğer çözüm uygun ise tüm AP'ler çözülür. Eğer tüm AP çözümleri uygun ise orijinal problem için optimal çözüm rapor edilir ve algoritma sonlanır. Diğer durumda, uygunsuz alt çözümlerden gelen Benders kesmeleri GAP'a eklenir ve GAP tekrar çözülür. Bu döngü durdurma koşulu sağlanana kadar devam eder.



Şekil 8.3. ÖMTBA akış şeması

8.4.1. Önerilen matematiksel programlama modeli-2 (ÖMAT-2)

Yukarıda bahsedilen mantık tabanlı Benders ayrıştırması metodunun Pİ-MHDP/SB' ye uygulanabilmesi için problemin verdiği kararların (görevlerin ve işçilerin istasyonlara atanması, görevlerin işçilere atanması ve görevlerin çizelgelenmesi) ayrıştırılabilir olduğu bir formülasyona ihtiyaç duyulmuştur. Bu sebeple, yeni bir karışık tam sayılı programlama modeli (ÖMAT-2) oluşturulmuştur. ÖMAT-2 oluşturulurken Esmailbeigi ve diğerleri (2016) sıra bağımlı hazırlık zamanlarını içeren basit montaj hattı dengeleme Tip-1 problemi için sunmuş oldukları istasyon tabanlı formülasyon temel alınmıştır. Pİ-MHDP/SB için geliştirilen formülasyondan çıkarılan gevşetilmiş ana problem ve alt problem formülasyonları da tüm detayları ile tanıtılmıştır. ÖMAT-2 için kullanılan

notasyon ve tanımlar ÖMAT-1 için kullanılan notasyon ve tanımlar ile mümkün mertebe benzer tutulmaya çalışılmıştır. Farklılıkları aşağıdaki gibidir.

Karar değişkenleri

$$x_{ijk} = \begin{cases} 1, & i \text{ görevi } j \text{ istasyonunda } k \text{ işçisine atandı ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$w_{jk} = \begin{cases} 1, & j \text{ istasyonunda } k \text{ işçisi kullanıldı ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$U_j = \begin{cases} 1, & j \text{ istasyonu açıldı ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$O_{ijk} = \begin{cases} 1, & i \text{ görevi } j \text{ istasyonunda } k \text{ işçisinin son görevi ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$Y_{ih} = \begin{cases} 1, & i \text{ görevi } h \text{ görevinden hemen önce yapılıyor ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$Z_{ih} = \begin{cases} 1, & i \text{ görevi son görev } h \text{ görevi ilk görev ise} \\ 0, & \text{diğer durumda} \end{cases}$$

$$St_i = i \text{ görevinin başlangıç zamanı } (0 \leq St_i \leq CT)$$

$$S_i = i \text{ görevinin atandığı istasyon indisi}$$

$$M_i = i \text{ görevinin atandığı işçi indisi}$$

$$Ys_{ih}, Ym_{ih}, Zs_{ih}, Zm_{ih} \text{ model için gerekli yardımcı 0-1 değişkenleridir.}$$

Amaç Fonksiyonu

$$\text{Min} \sum_{j \in J} \sum_{k \in K} w_{jk} + (1/(nt+1)) \sum_{j \in J} U_j \quad (8.25)$$

Amaç fonksiyonu Eş. 3.1 ÖMAT-1 modelinin amaç fonksiyonu ile aynıdır. Birincil amaç olarak istasyonlarda bulunan toplam işçi sayısını ve ikincil amaç olarak açılan istasyon sayısını en küçüklemektedir.

Kısıtlar

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} = 1, \forall i \in I \quad (8.26)$$

$$\sum_{j \in J} \sum_{k \in K} jx_{ijk} = S_i, \forall i \in I \quad (8.27)$$

$$\sum_{j \in J} \sum_{k \in K} kx_{ijk} = M_i, \forall i \in I \quad (8.28)$$

$$\sum_{h \in I} y_{ih} + \sum_{h \in I} z_{ih} = 1, \forall i \in I \quad (8.29)$$

$$\sum_{i \in I} y_{ih} + \sum_{i \in I} z_{ih} = 1, \forall h \in I \quad (8.30)$$

$$O_{ijk} \leq x_{ijk}, \forall i \in I, \forall j \in J, \forall k \in K \quad (8.31)$$

$$\sum_{h \in I} z_{ih} \leq \sum_{j \in J} \sum_{k \in K} O_{ijk}, \forall i \in I \quad (8.32)$$

$$\sum_{i \in I} O_{ijk} \leq 1, \forall j \in J, \forall k \in K \quad (8.33)$$

$$S_i - S_h \leq \psi(1 - Ys_{ih}), \forall i, h \in I \quad (8.34)$$

$$S_h - S_i \leq \psi(1 - Ys_{ih}), \forall i, h \in I \quad (8.35)$$

$$M_i - M_h \leq \psi(1 - Ym_{ih}), \forall i, h \in I \quad (8.36)$$

$$M_h - M_i \leq \psi(1 - Ym_{ih}), \forall i, h \in I \quad (8.37)$$

$$Ys_{ih} + Ym_{ih} \geq 2Y_{ih}, \forall i, h \in I \quad (8.38)$$

$$S_i - S_h \leq \psi(1 - Zs_{ih}), \forall i, h \in I \quad (8.39)$$

$$S_h - S_i \leq \psi(1 - Zs_{ih}), \forall i, h \in I \quad (8.40)$$

$$M_i - M_h \leq \psi(1 - Zm_{ih}), \forall i, h \in I \quad (8.41)$$

$$M_h - M_i \leq \psi(1 - Zm_{ih}), \forall i, h \in I \quad (8.42)$$

$$Zs_{ih} + Zm_{ih} \geq 2Z_{ih}, \forall i, h \in I \quad (8.43)$$

$$U_{j+1} \leq U_j, \forall j \in J \wedge j < nt \quad (8.44)$$

$$\sum_{i \in I} t_i x_{ijk} \leq CT.U_j, \forall j \in J, \forall k \in K \quad (8.45)$$

$$St_i + t_i + \sum_{h \in I} \mu_{ih} Z_{ih} \leq CT, \forall i \in I \quad (8.46)$$

$$St_i + \psi(S_i - S_h) + t_i + \tau_{ih} Y_{ih} \leq St_h, \forall i, h \in I \wedge i \in P(h) \quad (8.47)$$

$$St_i + (t_i + \tau_{ih}) + (\psi + \tau_{ih})(Y_{ih} - 1) \leq St_h, \forall i, h \in I \quad (8.48)$$

$$\sum_{i \in I} x_{ijk} \leq nt.w_{jk}, \forall j \in J, \forall k \in K \quad (8.49)$$

$$w_{jk+1} \leq w_{jk}, \forall j \in J, \forall k \in K \wedge k < M_{\max} \quad (8.50)$$

$$Y_{ih} + Y_{hi} \leq 1, \forall i, h \in I \quad (8.51)$$

$$x_{ijk} \in \{0, 1\}, \forall i \in I, \forall j \in J, \forall k \in K \quad (8.52)$$

$$O_{ijk} \in \{0, 1\}, \forall i \in I, \forall j \in J, \forall k \in K \quad (8.53)$$

$$St_i \geq 0, \forall i \in I \quad (8.54)$$

$$w_{jk} \in \{0, 1\}, \forall j \in J, \forall k \in K \quad (8.55)$$

$$U_j \in \{0, 1\}, \forall j \in J \quad (8.56)$$

$$S_i \geq 0, \forall i \in I \quad (8.57)$$

$$M_i \geq 0, \forall i \in I \quad (8.58)$$

$$Y_{ih} \in \{0, 1\}, \forall i, h \in I \quad (8.59)$$

$$Z_{ih} \in \{0, 1\}, \forall i, h \in I \quad (8.60)$$

$$Ys_{ih} \in \{0, 1\}, \forall i, h \in I \quad (8.61)$$

$$Ym_{ih} \in \{0, 1\}, \forall i, h \in I \quad (8.62)$$

$$Zs_{ih} \in \{0, 1\}, \forall i, h \in I \quad (8.63)$$

$$Zm_{ih} \in \{0, 1\}, \forall i, h \in I \quad (8.64)$$

Eş 8.26 kısıt grubu atama kısıdıdır ve her bir görevin tek bir istasyonda tek bir işçiye atanabilmesini sağlamaktadır. Eş. 8.27 kısıt grubu her bir görev için atandığı istasyon bilgisini vermektedir. Eş. 8.28 ise her bir görevin atandığı işçi bilgisini vermektedir. Eş. 8.27 ve Eş. 8.28 kısıt grupları dikkate alınmayabilir. Modelde S_i ve M_i görünen yerlere

sırasıyla $\sum_{j \in J} \sum_{k \in K} j \cdot x_{ijk}$ ve $\sum_{j \in J} \sum_{k \in K} k \cdot x_{ijk}$ yazılmalıdır. Eş. 8.29 ve Eş. 8.30'ya göre, her bir görevin sadece bir takip eden görevi ve bir takip ettiği görev olabilmektedir. Eş. 8.31, bir görevin istasyondaki bir işçide son iş olabilmesi için o görevin o istasyondaki o işçisine atanmış olması gerektiğini anlatmaktadır. Eş. 8.32, toplam geri hazırlık zamanı, hattaki toplam son iş sayısından fazla olamayacağını belirtmektedir. Eş. 8.33'a göre, istasyonda bulunan işçilerin en fazla bir adet son işi bulunabilmektedir. Eş. 8.34, Eş. 8.35-8.36-8.37, ve Eş. 4.38, bir görevin başka bir görevi hemen öncesinde takip edebilmesi için her iki görevin de hem aynı istasyona hem de aynı işçiye atanması gerektiğini söylemektedir. Eş. 8.39-8.40-8.41-8.42-8.43, bir görevin son görev başka bir görevin ilk görev olabilmesi için her iki görevin de hem aynı istasyona hem de aynı işçiye atanması gerektiğini ifade etmektedir. Eş. 8.44 açılan istasyonları düzenlemektedir. Eş. 8.45, her istasyondaki her işçi için o işçiye atanan görevlerin toplam sürelerinin belirlenen çevrim zamanını aşmaması gerektiğini söylemektedir. Bu kısıt sunulan formülasyonu güçlendirmeyi amaçlamaktadır. Eş. 8.46, görevlerin bitiş sürelerinin belirlenen çevrim zamanını aşamayacağını anlatmaktadır. Eş. 8.47 aralarında öncelik ilişkisi bulunan görevler için istasyondaki ve istasyonlar arasındaki başlangıç zamanlarını düzenlemektedir. Eş. 8.48, görevler arasındaki ileri hazırlık zamanlarını ayarlamaktadır. Eş. 8.49, istasyonda kullanılmayan işçiye görev ataması yapılmasını engellemektedir. Eş. 8.50 istasyonlardaki işçi atamalarını düzenlemektedir. Eş. 8.51, h görevi i görevinin ileri yönde takip edeni iken aynı zamanda, i görevi h görevinin ileri yönde takip edeni olamayacağını ifade etmektedir. Eş. 8.52-8.64, ilgili değişkenlere ait işaret kısıtlarıdır.

8.4.2. Gevşetilmiş ana problem (GAP)

Gevşetilmiş ana problem görevlerin istasyon ve işçilere atanması ve istasyonlarda çalışacak işçilerin belirlenmesi ile ilgilenmektedir. Görevleri çizelgelerken hazırlık zamanlarının dikkate alınması ile ortaya çıkan karmaşıklık gevşetilmiş ana problemden çıkarılmıştır. Bir diğer deyişle görevler arasında bulunan ileri ve geri hazırlık zamanları yok sayılmıştır. Gevşetilmiş ana problem ÖMAT-2 karma tam sayılı programlama modeli kullanılarak tasarlanmıştır. Model Kellegöz (2016) tarafından paralel çok işçili montaj hattı dengeleme problemi için önerilen karma tam sayılı programlama modeline benzemektedir. Önerilen Benders algoritmasının iterasyonları ρ ile ifade edilsin. GAP için ρ iterasyonundaki formülasyon aşağıdaki gibidir. Kullanılan tanım ve notasyonlar ÖMAT-2

ile aynıdır. Ek olarak istasyonlardaki işçilere atanan görevleri takip edebilmek için YY_{ih} 0-1 tamsayı değişkeni kullanılmıştır. Eğer görev i görev h den önce yapılıyor ise değişken 1, diğer durumda 0 değerini almaktadır.

$$GAP(\rho) : \text{Min} \sum_{j \in J} \sum_{k \in K} w_{jk}^\rho + (1/(nt+1)) \sum_{j \in J} U_j^\rho \quad (8.65)$$

$$\sum_{j \in J} \sum_{k \in K} x_{ijk}^\rho = 1, \forall i \in I \quad (8.66)$$

$$\sum_{j \in J} jx_{ijk}^\rho \leq \sum_{j \in J} jx_{hjp}^\rho, \forall i, h \in I \wedge i \in P(h), \forall k \in K \quad (8.67)$$

$$\sum_{i \in I} t_i x_{ijk}^\rho \leq CT.U_j^\rho, \forall j \in J, \forall k \in K \quad (8.68)$$

$$St_i^\rho + t_i \leq CT, \forall i \in I \quad (8.69)$$

$$\sum_{i \in I} x_{ijk}^\rho \leq nt.w_{jk}^\rho, \forall j \in J, \forall k \in K \quad (8.70)$$

$$St_i^\rho - St_h^\rho + \psi(1 - \sum_{k \in K} x_{ijk}^\rho) + \psi(1 - \sum_{k \in K} x_{hjk}^\rho) \geq t_h, \forall i, h \in I \wedge h \in P(i), \forall j \in J \quad (8.71)$$

$$St_h^\rho - St_i^\rho + \psi(1 - x_{ijk}^\rho) + \psi(1 - x_{hjk}^\rho) + \psi(1 - YY_{ih}^\rho) \geq t_i, \forall i, h \in I, \forall j \in J, \forall k \in K \quad (8.72)$$

$$St_i^\rho - St_h^\rho + \psi(1 - x_{ijk}^\rho) + \psi(1 - x_{hjk}^\rho) + \psi(YY_{ih}^\rho) \geq t_i, \forall i, h \in I, \forall j \in J, \forall k \in K \quad (8.73)$$

$$U_{j+1}^\rho \leq U_j^\rho, \forall j \in J \quad (8.74)$$

$$w_{jk+1}^\rho \leq w_{jk}^\rho, \forall j \in J, \forall k \in K \quad (8.75)$$

$$YY_{ih}^\rho + YY_{hi}^\rho \leq 1, \forall i, h \in I \quad (8.76)$$

$$Bcut^e \quad e \in \{1, 2, \dots, \rho-1\} \quad (8.77)$$

$$x_{ijk}^\rho \in \{0, 1\}, \forall i \in I, \forall j \in J, \forall k \in K \quad (8.78)$$

$$St_i^\rho \geq 0, \forall i \in I \quad (8.79)$$

$$w_{jk}^\rho \in \{0, 1\}, \forall j \in J, \forall k \in K \quad (8.80)$$

$$U_j^\rho \in \{0, 1\}, \forall j \in J \quad (8.81)$$

$$YY_{ih}^\rho \in \{0, 1\}, \forall i, h \in I \quad (8.82)$$

Gevşetilmiş ana probleme ait amaç fonksiyonu (kısıt grubu Eş.8.65) ÖMAT-1 ve ÖMAT-2 formülasyonlarına ait amaç fonksiyonu ile aynıdır. Birincil amaç olarak istasyonlarda bulunan toplam işçi sayısı ve ikincil amaç olarak açılan istasyon sayısını en küçüklemektedir. Eş. 8.66, bir görevin tek bir istasyondaki tek bir işçiye atanmasını garantiler. Eş. 8.67, görevler arasındaki öncelik ilişkilerini sağlar. Eş. 8.68, eğer bir istasyondaki bir işçiye bir görev atandı ise, o istasyonun açılmış olduğunu belirtir. Eş. 8.69'e göre, herhangi görev belirlenen çevrim zamanı içerisinde tamamlanmalıdır. Eş 8.70 herhangi bir istasyondaki herhangi bir işçiye en az bir görev atandı ise o istasyonda o işçisi kullanılmış olduğunu gösterir. Eş. 8.71, aralarında öncelik ilişkisi bulunan görevler için aynı istasyonun aynı işçisinde ya da farklı işçisinde başlangıç zamanlarını düzenler. Eş 8.72 ve Eş. 8.73, aralarında öncelik ilişkisi bulunmayan görevlerin başlangıç zamanlarını düzenler. Eş 8.74 ve Eş. 8.75 istasyon ve işçi kullanımlarını düzenler. Eş. 8.76'e göre bir i görevi başka bir h görevinin ileri yönde takip edeni ise, h görevi i görevinin ileri yönde takip edeni olamaz. Eş. 8.77 önceki iterasyonlarda elde edilen Benders kesmelerini ifade etmektedir. Burada $Bcut^e$ bir set olup, Eş. 8.78-Eş. 5.82 kısıt grupları işaret kısıtlarıdır.

8.4.3. Alt problem (AP)

İlk olarak GAP bir çözüm elde eder. Bu çözüm $(\bar{X}, \bar{U}, \bar{W})$ olarak ifade edilebilir. Alt problemler bu çözümü, görevlerin atanan istasyon ve işçiler için çizelgelemesini yaparak verilen çevrim zamanına uygunluğunu kontrol eder. Önceki bölümlerde de bahsedildiği üzere alt problemler literatürdeki asimetrik gezgin satıcı problemine benzemektedir. Görevler gezgin satıcı problemindeki şehirleri, hazırlık zamanları ise şehirlerarasındaki mesafeyi ifade etmektedir. GAP çözümünde açılan her bir istasyon için bir adet alt problem oluşturulmaktadır. Örneğin görev sayısı 10 olan bir montaj hattı dengeleme problemi verilsin. GAP ile görevlerin 1. istasyon için 1, 7, 6, 2 ve 2. İstasyon için 3, 4, 5 ve 3. İstasyon için 8, 9, 10 olarak atandığı varsayılınsın. Bu durumda ayrı ayrı 3 adet AP çözümü yapılacaktır. İlk alt problem sadece 1. istasyona atanan görevleri, ikinci alt problem 2. İstasyona atanan görevleri ve 3. Alt problem 3. İstasyona atanan görevleri içerecektir. Spesifik bir istasyon j için AP formülasyonu aşağıdaki gibi tanımlanmıştır. Formülasyon için gerekli ek notasyonlar aşağıdaki gibidir.

A	: Tüm öncüllük ilişkileri kümesi
I_j	: İstasyon j 'ye atanan görevler kümesi

A_j : İstasyon j 'ye atanan görevlerin öncüllük ilişkileri kümesi

$$\text{Min } Cmax \quad (8.83)$$

Kısıtlar

$$\sum_{h \in I_j} y_{ih} + \sum_{h \in I_j} z_{ih} = 1, \forall i \in I_j \quad (8.84)$$

$$\sum_{i \in I_j} y_{ih} + \sum_{i \in I_j} z_{ih} = 1, \forall h \in I_j \quad (8.85)$$

$$O_{ijk} \leq \bar{x}_{ijk} \quad \forall i \in I_j, \forall k \in K \wedge \bar{w}_{jk} = 1 \quad (8.86)$$

$$\sum_{h \in I_j} z_{ih} \leq \sum_{k \in K} O_{ijk}, \quad \forall i \in I_j \quad (8.87)$$

$$\sum_{i \in I_j} O_{ijk} \leq 1, \quad \forall k \in K \wedge \bar{w}_{jk} = 1 \quad (8.88)$$

$$St_i + t_i + \sum_{h \in I_j \wedge x_{hjk}=1} \mu_{ih} z_{ih} \leq Cmax, \forall i \in I_j \wedge \bar{x}_{ijk} = 1 \quad (8.89)$$

$$St_i + t_i + \tau_{ih} y_{ih} \leq St_h, \forall i, h \in A_j \quad (8.90)$$

$$St_i + (\tau_{ih} + t_i) + \psi(y_{ih} - 1) \leq St_h, \forall i, h \in I_j \quad (8.91)$$

$$y_{ih} + y_{hi} \leq 1, \forall i, h \in I_j \quad (8.92)$$

$$St_i \geq 0, \forall i \in I_j \quad (8.93)$$

$$y_{ih} \in \{0,1\}, \forall i, h \in I_j \quad (8.94)$$

$$z_{ih} \in \{0,1\}, \forall i, h \in I_j \quad (8.95)$$

$$Cmax \geq 0 \quad (8.96)$$

Amaç fonksiyonu Eş. 8.83 istasyondaki en son tamamlanan görevin tamamlanma zamanını minimum yapar. Eş. 8.84 ve Eş. 8.85'e göre, her bir görevin sadece bir takip eden görevi ve bir takip ettiği görev olabilmektedir. Eş. 8.86, bir görevin istasyondaki bir işçide son iş olabilmesi için o görevin o istasyondaki o işçisine atanmış olması gerektiğini anlatmaktadır. Eş. 8.87, toplam geri hazırlık zamanı, hattaki toplam son iş sayısından fazla olamayacağını belirtmektedir. Eş. 8.88'e göre, istasyonda bulunan işçilerin en fazla bir adet son işi bulunabilmektedir. Eş. 8.89 işçilerin yüklerini hesaplamaktadır. Eş. 8.90-8.91,

görevlerin başlangıç zamanlarını hazırlık zamanlarını göz önünde bulundurarak hesaplamaktadır. Eş. 8.92, h görevi i görevini ileri yönde takip ederken aynı anda, i görevi h görevini ileri yönde takip edemeyeceğini ifade etmektedir. Eş. 8.93-8.96, ilgili değişkenlere ait işaret kısıtlarıdır.

8.4.4. Benders kesmeleri

Önceki bölümlerde bahsedildiği üzere önerilen Benders algoritması ÖMTBA, GAP ile elde edilen $(\bar{X}, \bar{U}, \bar{W})$ çözümün uygunluğunu AP ile kontrol etmektedir. AP $(\bar{X}, \bar{U}, \bar{W})$ her bir istasyon için uygun çözüm elde edebilir ise orijinal problem için en iyi çözüm bulunmuş demektir. Diğer taraftan AP en az bir istasyon için uygun çözüm elde edememiş ise ρ . iterasyondaki gevşetilmiş ana probleme e. iterasyondaki Benders kesmesi (*Bcut*) eklenir ve GAP tekrar çözülür. *Bcut* aşağıdaki Eş. 8.97'deki gibi formüle edilmektedir. Bu kesme, uygun çözüm elde edilmesine engel olan kısmi çözümü çözüm uzayından kaldırmaktadır.

L : Uygunsuz çözüme sahip istasyonların seti

$$\sum_{k \in K} \sum_{i \in I_j} x_{ijk} \leq \left(\sum_{k \in K} \sum_i \bar{x}_{ijk} \right) - 1, \quad \forall j \in L \quad (8.97)$$

8.5. Hesaplama Sonuçları ve Değerlendirme

Bu bölümde ÖMAT-2 ve ÖMTBA yaklaşımlarının daha önceden tanımlanan test problemleri için elde edilen sonuçları raporlanmıştır.

Çizelge 8.1. ÖMAT-2 ile 3600 CPU saniyesi kısıdı ile çalıştırılan problem veri setlerinden elde edilen sonuçları özetlemektedir. ÖMAT-2 ile 236 problem veri setinden sadece 72 adet problem veri setine ait en iyi sonuçlar elde edilmiştir. Bu değer ÖMAT-1 ile elde edilen değerden fazla olduğu görülmektedir. Geri kalan veri setlerinden 52 adet veri setinde ise verilen CPU saniye kısıdı içerisinde uygun çözüm elde edilmiştir. 112 adet veri setinde ise belirlenen CPU saniye kısıdı altında hiçbir çözüme ulaşamamıştır. ÖMAT-1 ve ÖMAT-2 performansları karşılaştırılacak olursa, ÖMAT-2 ye ait sonuçların daha iyi olduğu gözlemlenmiştir. Her ne kadar ÖMAT-2 sonuçlarının ÖMAT-1 sonuçlarından daha

iyi olduğu görülse de problem büyüklüğü arttıkça önerilen modelin performansı düşmektedir.

Çizelge 8.1. Matematiksel programlama modeli sonuçları/ÖMAT-2

Problem	nt	Optimali Bulunan	Uygun Çözüm Bulunan	Uygun Çözüm Bulunamayan	Toplam
Bowman	8	4			4
Jackson	11	24			24
Jaeschke	9	20			20
Mertens	7	24			24
Heskia	28		9	3	12
Gunther	35		18	10	28
Buxey	29		7	21	28
Lutz1	32		4	4	8
Sawyer	30		18	22	40
Arcus1	83			12	12
Arcus2	11			12	12
Bartholdi	14			12	12
Mukherje	94			12	12
Toplam		72	52	112	236

Çizelge 8.2. ÖMAT-2 için farklı hazırlık zamanı yoğunluklarında modelin davranışını göstermektedir. CPU zamanlarının artan hazırlık zamanı yoğunluklarından etkilenmediğinin gözlemlendiği daha önceden belirtilmiştir. Bu sebeple, her bir yoğunlukta elde edilen en iyi çözüm sayılarında bir değişiklik görülmemektedir.

Çizelge 8.2. Farklı hazırlık zamanı yoğunlukları için ÖMAT-2 sonuçları

α	Problem	Optimali Bulunan	Uygun Çözüm Bulunan	Uygun Çözüm Bulunamayan	CPU
0.25	Bowman	1			0,421
	Jackson	6			4,254
	Jaeschke	5			1,736
	Mertens	6			0,941
0.50	Bowman	1			0,764
	Jackson	6			9,348
	Jaeschke	5			2,045
	Mertens	6			1,581
0.75	Bowman	1			1,392
	Jackson	6			11,873
	Jaeschke	5			3,961
	Mertens	6			2,112
1.00	Bowman	1			1,871
	Jackson	6			16,413
	Jaeschke	5			5,257
	Mertens	6			3,746

Pİ-MHDP/SB için önerilen mantık tabanlı Benders ayrıştırması (ÖMTBA) algoritması ve matematiksel modelleri Gurobi 8 yardımı ile çözdürülmüş ve algoritma için Phyton 3.6 programlama dili kullanılmıştır. Her bir veri seti için 3600 CPU saniyesi çalıştırılmıştır. Çizelge 8.3. ÖMTBA' sının problem veri setlerinde elde ettiği sonuçları özetlemektedir. Toplam 79 problemde en iyi çözümler elde edilmiştir. 82 problem seti için hiçbir çözüm bulunamazken 75 adet problem seti için verilen CPU zamanı kısıdı altında uygun çözümlere ulaşılmıştır. Bu durumun Pİ/MHDP-SB' nin ayrıştırılmasından kaynaklandığı söylenebilir. Sonuç olarak bu tez kapsamında çalışılan problemin iki farklı probleme (GAP ve AP) ayrıştırılıp bu problemlerin birbirileri ile Benders kesmeleri ile iletişime geçmelerini sağlamak, koşturulan problem setlerinde elde edilen en iyi çözüm sayıları biraz olsa da artırabildiği gözlemlenmiştir. Ayrıca farklı yoğunluğa sahip hazırlık zamanları için önerilen MTBA algoritmasının elde edilen çözümler ve CPU zamanı açısından değerlendirilmesi Çizelge 8.4.'te sunulmuştur. Farklı yoğunluktaki hazırlık zamanlarının elde edilen en iyi çözüm sayıları ve CPU zamanı üzerinde bir etkisi olmadığı görülmüştür.

Çizelge 8.3. ÖMTBA test problemi sonuçları

Problem	N	Optimali Bulunan	Alt Sınırı Bulunan	Uygun Çözüm Bulunamayan	Toplam
Bowman	8	4			4
Jackson	11	24			24
Jaeschke	9	20			20
Mertens	7	24			24
Heskia	28	3	5*	4	12
Gunther	35		22*	6	28
Buxey	29		15*	13	28
Lutz1	32		6*	2	8
Sawyer	30	4	27*	9	40
Arcus1	83			12	12
Arcus2	111			12	12
Bartholdi	148			12	12
Mukherje	94			12	12
Toplam		79	75	82	236
* 3600 CPU süresi sonunda GAP ile elde edilen en iyi değer.					

Çizelge 8.4. Farklı hazırlık zamanı yoğunlukları için ÖMTBA sonuçları

α	Problem	Optimali Bulunan	CPU
0.25	Bowman	1	0,254
	Jackson	6	2,710
	Jaeschke	5	0,867
	Mertens	6	0,330
	Heskia	2	65,44
	Sawyer	1	46,12
0.50	Bowman	1	0,226
	Jackson	6	3,091
	Jaeschke	5	0,681
	Mertens	6	0,317
	Heskia	1	37,62
	Sawyer	1	114,45
0.75	Bowman	1	0,283
	Jackson	6	2,573
	Jaeschke	5	0,790
	Mertens	6	0,267
	Sawyer	1	75,40
1.00	Bowman	1	0,194
	Jackson	6	3,462
	Jaeschke	5	0,819
	Mertens	6	0,229
	Sawyer	1	52,07

Çizelge 8.5.'te ÖMAT-2 ve ÖMTBA algoritması karşılaştırılmıştır. Çizelgeden görüleceği üzere koşturulan problem setlerinde her iki çözüm yönteminin elde ettiği sonuçlar aynıdır. Ancak, CPU zamanı açısından ÖMTBA algoritması daha iyi bir performans sergilemektedir. Bunun sebebi Pİ/MHDP-SB' nin Gevşetilmiş Ana Problem (GAP) ve Alt Problemler (AP) olarak ayrıştırılmasıdır. GAP sadece görevlerin istasyonlara ve istasyondaki işçilere atanması ile ilgilenmektedir. Görevler arasındaki sıra bağımlı hazırlık zamanlarının GAP içerisinde bulunmaması algoritmanın kullandığı CPU zamanını azaltmaktadır.

9. SONUÇ ve ÖNERİLER

İstasyonlarda birden fazla işçinin eşzamanlı olarak kendilerine atanan görevleri gerçekleştirmelerine olanak sağlayan paralel çok işçili montaj hatları tasarımı, özellikle hacimsel olarak büyük ürünlerin (*otomobil, otobüs, kamyon, vb.*) üretildiği sistemlerde kullanılmaktadır. Ancak ilgili literatür incelendiğinde paralel çok işçili montaj hatları ile ilgili çalışmalara çok fazla rastlanılmamıştır. Bu tez çalışması kapsamında, bilinen daha önceden yapılmış bir çalışma olmayan, görevler arasında sıra bağımlı hazırlık zamanlarının bulunduğu, birincil amacın hatta kullanılan işçi sayısının, ikincil amacın ise hatta kullanılan istasyon sayısının en küçüklendiği paralel çok işçili montaj hatlarının dengelenmesi için çözüm yaklaşımları sunulmuştur. İlk olarak karma tam sayılı programlama modeli olan ÖMAT-1 geliştirilmiştir. Literatürden uyarlanan test problemleri üzerinde modelin performansı test edilmiştir. Küçük boyutlu problemlerin çoğunda modelin koşturulması için verilen süre içerisinde en iyi çözümlere ulaşılmıştır. Ancak orta ve büyük boyutlu problem setlerinde verilen CPU zamanı içerisinde uygun çözüme dahi ulaşamamıştır. Bu durum farklı çözüm yöntemi arayışlarını beraberinde getirmiştir. Bu sebeple, kısıt programlama literatürü montaj hattı dengeleme problemleri için irdelenmiş ve Pİ-MHDP/SB için literatürde görülen bu boşluk üzerine kısıt programlama modelleri (KP1 ve KP2) önerilmiştir. Problem setleri için KP2'den elde edilen sonuçlar, ÖMAT-1 ile karşılaştırıldığında hem bulunan en iyi çözüm sayılarının hem de kullanılan CPU zamanlarının ÖMAT-1 tarafından sunulan sonuçlardan daha iyi olduğu görülmüştür. Buna rağmen, orta ve büyük boyutlu problem setlerinde model istenilen etkinliği yakalayamamıştır. KP1'in hem ÖMAT-1 hem de KP2'nin gösterdiği performansı yakalayamadığı görülmüştür. Bu durumu aşabilmek için Pİ-MHDP/SB için iki adet meta sezgisel algoritma önerilmiştir. Bu algoritmalarından biri yığın tabanlı arama yapan Genetik algoritma olarak, bir diğeri ise tek noktadan araba yapan Tavlama Benzetimi algoritması olarak tasarlanmıştır. Algoritmaların küçük boyutlu problemlerde bilinen en iyi sonuçları yakaladığı görülmüştür. Önerilen meta sezgisel algoritmalar küçük boyutlu problemler için kendi aralarında kıyaslandıklarında CPU zamanı açısından birbirlerine üstünlükleri görülmemiştir. Orta ve büyük boyutlu problem setlerinden elde edilen sonuçlar incelendiğinde GA'nın etkinliğinin TB'den daha iyi olduğu gözlemlenmiştir.

Pİ-MHDP/SB' de verilen 3 tip karar bulunmaktadır. Bunlar görevlerin atanması, istasyon işçi konfigürasyonu ve görevlerin istasyonlarda çizelgelenmesidir. Problemin verilmesi

gereken bu kararlar ışığında ayrıştırılabilmesi, problem setleri için daha fazla en iyi çözüme ulaşılmasına olanak sağlayabilecektir. Bu doğrultuda, montaj hattı dengeleme problemleri dışında sıklıkla kullanılan Benders ayrıştırması metodu mantık tabanlı Benders ayrıştırması ile Pİ-MHDP-SB' ye uyarlanmıştır. Ancak, klasik Benders ayrıştırması yaklaşımının Pİ-MHDP/SP için uyarlanması mümkün olmamıştır. Tanımlanan alt problemlerin karma tam sayılı programlama modeli olarak modellenemediği Mantık Tabanlı Benders Ayrıştırması algoritmasına dayalı bir algoritma geliştirilmiştir. Bilinen üzere, Pİ-MHDP/SB için ÖMTBA algoritması bu tez çalışması ile ilk kez uygulanmıştır. Elde edilen problem setlerinin sonuçları daha önceki modelleme yaklaşımları (*matematiksel ve kısıt programlama*) ile elde edilememiş problem setlerinde en iyi çözümlere ulaşmayı başarmışlardır. Ancak, ayrıştırılan problemin parçaları (GAP ve AP) kombinatoriyal açıdan değerlendirdiklerinde halen NP-Zor sınıfındadırlar. Bu sebeple, bir kısım orta boyut ve büyük boyutlu problemlerde kabul edilebilir bir zaman içerisinde çözüm elde etmek zorlaşmaktadır.

Bu tez çalışması ile Pİ-MHDP/SB için kesin çözüm metodu olarak iki farklı karma tam sayılı matematiksel model (ÖMAT-1 ve ÖMAT-2), kısıt programlama modelleri (KP1 ve KP2) ve mantık tabanlı Benders ayrıştırması algoritması (ÖMTBA) geliştirilmiştir. Büyük boyutlu problem setleri için meta sezgisel algoritmalarından yığın tabanlı arama yapan bir Genetik Algoritma (GA) ve tek noktadan arama yapan bir Tavlama Benzetimi Algoritması (TB) geliştirilmiştir. Pİ-MHDP/SB için önerilen kesin çözüm metotlarının hibrit bir şekilde tasarlanması; örneğin, ÖMTBA'da ki alt problemlerin yapısı kısıt programlama için oldukça uygundur. Karma tam sayılı programlamanın dezavantajları alt problem için kısıt programlama ile giderilebilir. Bu hibritleştirme, Pİ-MHDP/SB için bir ileri çalışma önerisidir. ÖMTBA algoritmasında kullanılan Benders kesmelerinin kalitesi önerilen algoritmanın kalitesini etkileyen başlıca etmenlerdendir. Bu tez çalışmasında olduğu gibi uygunluk kesmeleri yerine daha etkin kesmeler tanımlanabilir. Ek olarak, problemin komşu problemleri (daha fazla gerçek hayat kısıdı olan problemler; *belirli bölgeleme kısıtları, işçi yetkinlikleri vb.*) üzerinde benzer metotlar geliştirilebilir. Böylelikle endüstrinin ihtiyacına cevap verilmesine çalışılabilir.

KAYNAKLAR

- Aghajani M., Ghodsi R., Javadi B.(2014), Balancing of robotic mixed-model two-sided assembly line with robot setup times. *International Journal of Advanced Manufacturing Technology*, 74(5-8),1005-1016.
- Akpınar S., Bayhan G.M., Baykasoğlu A. (2013). Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks, *Applied Soft Computing*, 13(1), 574–589.
- Akpınar S., Baykasoğlu A. (2014). Modeling and solving mixed-model assembly line balancing problem with setups-Part I: a mixed integer linear programming model, *Journal of Manufacturing Systems*, 33, 177–187.
- Akpınar S., Baykasoğlu A. (2014). Modeling and solving mixed-model assembly line balancing problem with setups. Part II: A multiple colony hybrid bees algorithm, *Journal of Manufacturing Systems*, 33, 445–461.
- Andre's, C., Miralles, C., Pastor, R., (2008). Balancing and scheduling tasks in assembly lines with sequence-dependent setup times. *European Journal of Operational Research*, 187 (3), 1212–1223.
- Bartholdi, J.J. (1993) Balancing two-sided assembly lines: A case study. *International Journal of Production Research*, 31, 2447-2461.
- Baybars, I. (1981). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, 32, 909–932.
- Baykasoglu A., Ozbakir L. (2007). Stochastic U-line balancing using genetic algorithms. *International Journal of Advanced Manufacturing Technologies*, 32,139–147.
- Baykasoglu, A. and Dereli, T., (2008). Two-sided assembly line balancing using an ant-colony-based heuristic. *International Journal of Advanced Manufacturing Technology*, 36 (5–6), 582–588.
- Becker, C., A. Scholl. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research*, 168, 694–715.
- Becker, C., Scholl, A. (2009). Balancing assembly lines with variable parallel workplaces: Problem definition and effective solution procedure, *European Journal of Operational Research*, 199, 359-374.
- Bockmayr, A., Pizaruk, N. (2001). Solving an assembly line balancing problem by combining IP and CP. *6th Annual Workshop of ERCIM Working Group on Constraints*, Prag, Çek Cumhuriyeti.
- Bowman, E. H. (1960). Assembly line balancing by linear programming, *Operations Research*, 8, 385-389.

- Bukchin, J., Dar-El, E.M., Rubinovitz, J. (1997). Team oriented assembly system design: A new approach, *International Journal of Production Economics*, 51, 47–57.
- Cevikcan, E., Durmusoglu, M.B., Unal, M.E. (2009). A team-oriented design methodology for mixed model assembly systems, *Computers & Industrial Engineering*, 56: 576-599.
- Chang, H.J., Chang, T.M., (2010). Simultaneous Perspective-Based Mixed-Model Assembly Line Balancing Problem, *Tamkang Journal of Science and Engineering*, 13, 327-336.
- Chu Y., Xia Q. (2004). Generating Benders Cuts for a General Class of Integer Programming Problems. *Springer CPAIOR Lecture Notes in Computer Science*, 3011.
- Celik E., Kara Y., Atasagun Y. (2014). A new approach for rebalancing of U-lines with stochastic task times using ant colony optimisation algorithm. *International Journal of Production Research*, 52:24, 7262-7275.
- Coley, D. A. (1999). An introduction to genetic algorithms for scientists and engineers. *World Scientific, Singapore*, 227.
- Dimitriadis, S.G. (2006). Assembly line balancing and group working: A heuristic procedure for workers' groups operating on the same product and workstation, *Computers & Operations Research*, 33: 2757-2774.
- Erel, E., Sarin, S.C. (1998). A survey of the assembly line balancing procedures. *Production Planning & Control*, 9, 414-434.
- Erel E., Sabuncuoglu I., Sekerci H. (2005). Stochastic assembly line balancing using beam search. *International Journal of Production Research*, 43(7),1411–1426.
- Fattahi, P., Roshani, A., Roshani A. (2011). A mathematical model and ant colony algorithm for multi-manned assembly line balancing problem, *International Journal of Advanced Manufacturing Technologies*, 53, 363-378.
- Gen, M., Cheng, R. (2000). Genetic algorithms and engineering optimization. *John Wiley & Sons Inc.*, New York, USA, 495 .
- Geofrion A. M. (1972). *Generalized benders decomposition*. *Journal of Optimization Theory and Applications*, 10(4),237-260.
- Goldberg, D. E. (1989). Genetic algorithms in search, *Optimization and Machine Learning* (1st bs.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.
- Gökçen H., Agpak K. (2006). A goal programming approach to simple U-line balancing problem. *European Journal of Operations Research*, 171, 577–585.

- Hamta N., Fatemi Ghomi S.M.T., Jolai F., Shirazi M.A.,. (2012). A hybrid PSO algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect, *International Journal of Production Economics*, 141 (1), 99–111.
- Helgeson W.B., Brine D.P. (1961). *Assembly line balancing using ranked positional weight technique*, *Journal of Industrial Engineering*, 12, 394-398.
- Hoffmann, T.R. (1963). Assembly line balancing with a precedence matrix. *Management Science*, 9, 551-62.
- Hooker J.N., Ottosson G.. (2003). *Logic-based benders decomposition*. *Mathematical Programming*, 96(1):33-60.
- Hooker J. N. (2005). A hybrid method for the planning and scheduling, *Constraints*, 10(4),385-401.
- Hooker J. N. (2007). Planning and scheduling by logic-based benders decomposition. *Operations Research*. 55(3),588-602.
- Hu, XF., Wu, EF., Jin, Y. (2008). A station-oriented enumerative algorithm for two-sided assembly line balancing. *European Journal of Operational Research*, 186 (1), 435–440.
- ILOG. (2013a). IBM ILOG OPL 12.6 Language Manual. *ILOG S. A., USA*.
- Jackson, J. R. (1956). A computing procedure for a line balancing problem, *Management Science*, 2 (3), 261-271.
- Kara Y., Paksoy T., Chang C.(2009). Binary fuzzy goal programming approach to single model straight and U-shaped assembly line balancing. *European Journal of Operational Research*, 195(2), 335-347.
- Kara Y., Özgüven C., Yalçın N., & Yakup Atasagun (2011). Balancing straight and U-shaped assembly lines with resource dependent task times. *International Journal of Production Research*, 49:21, 6387-6405.
- Kazemi, A., Sedighi, A. (2013). *A cost-oriented model for balancing mixed-model assembly lines with multi-manned workstations*, *International Journal of Services and Operations Management*, 16: 289-309.
- Kellegöz, T., Toklu, B. (2012). An efficient branch and bound algorithm for assembly line balancing problems with parallel multi-manned workstations, *Computers & Operations Research*, 39, 3344-3360.
- Kellegöz, T., Toklu, B. (2014). A priority rule based constructive heuristic and an improvement method for balancing assembly lines with parallel multi-manned workstations, *International Journal of Production Research*, 53(3), 736-756.

- Keskintürk T., Küçük B., Karışık modellenli montaj hatlarının genetik algoritma kullanılarak dengelenmesi, *Yönetim Dergisi*, 17 (53), 52-63.
- Kim, Y.K., Kim, Y., Kim, Y.J., (2000). Two-sided assembly line balancing: a genetic algorithm approach. *Production Planning & Control*, 11 (1), 44–53.
- Kim, Y.K., Song, W.S., Kim, J.H., (2009). A mathematical model and a genetic algorithm for two-sided assembly line balancing. *Computers and Operations Research*, 36 (3), 853–865.
- Küçükkoç İ., Zhang D. (2014). Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. *International Journal of Production Research*, 52(12), 3665-3687.
- Li, H., Womer, K. (2012). Optimizing the supply chain configuration for make-to-order manufacturing. *European Journal of Operational Research*, 221(1), 118-128.
- Lustig I., Puget J. F. (2013). Constraint Programming. *Encyclopedia of Operations Research and Management Science*, New York: Springer, 267-273.
- Martino, L., Pastor, R. (2009). Heuristic procedures for solving the general assembly line balancing problem with setups. *International Journal of Production Research*, 48(6), 1787-1804.
- McMullen P.R., Frazier G.V. (1998). Using simulated annealing to solve a multi-objective assembly line balancing problem with parallel workstations. *International Journal of Production Research*, 36(10), 2717–2741.
- McMullen PR., Tarasewich P. (2006). Multi-objective assembly line balancing via a modified ant colony technique. *International Journal of Production Research*, 44(1), 27–42.
- Miller, B. L., Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3), 193-212.
- Michalewicz, Z. (1996). Genetic algorithm + data structure = evolution programs. *Springer-Verlag*, New York, USA, 387.
- Mitchell, M. (1996). An introduction to genetic algorithms. *MIT Press*, Massachusetts, 158.
- Nabiyev VV. (2003), Yapay Zeka: Problemler Yöntemler Algoritmalar. *1. Baskı, Ankara: Seçkin Yayınevi*, s.606.
- Özbakir L., Baykasoglu A., Gorkemli B., Gorkemli L. (2011). Multiple-colony ant algorithm for parallel line balancing problem. *Applied Soft Computing*.
- Özcan, U., Toklu, B., (2009). Multiple-criteria decision-making in two-sided assembly line balancing: a goal programming and a fuzzy goal programming models. *Computers and Operations Research*, 36 (6), 1955–1965

- Özcan U., Toklu B. (2009). A tabu search algorithm for two-sided assembly line balancing. *International Journal of Advanced Manufacturing Technology*, 43:822–829.
- Özcan U., Toklu B. (2009). A new hybrid improvement heuristic approach to simple straight and U-type assembly line balancing problems. *Journal of Intelligent Manufacturing*, 20,123–136.
- Özcan U. (2010). Balancing stochastic two-sided assembly lines: a chance-constrained piecewise-linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research*, 205, 81–97.
- Özcan U., Cercioglu H., Gokcen H., Toklu B. (2010). Balancing and sequencing of parallel mixed-model assembly lines. *International Journal of Production Research*, 48(17), 5089–5113.
- Özcan U., Toklu B. (2010). Balancing two-sided assembly lines with sequence-dependent setup times. *International Journal of Production Research*, 48 (18), 5363– 5383.
- Özcan U., Kellegoz T., Toklu B. (2011). A genetic algorithm for the stochastic mixed model U-line balancing and sequencing problem. *International Journal of Production Research*, 49(6), 1605–1626.
- Pastor, R., Ferrer, L. (2009). An improved mathematical program to solve the simple assembly line balancing problem. *International Journal of Production Research*, 47(11), 2943–2959.
- Pastor, R., Ferrer, L., and García, A. (2007). Evaluating optimization models to solve SALBP. *Lecture Notes in Computer Science*, 4705, 791-803.
- Puget, J. F., Lustig, I. (2001). Constraint programming and maths programming. *The Knowledge Engineering Review*, 16(1), 5-23.
- Rahmaniani R., Crainic T.G., Gendreau M., and Rei W. (2017). The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3), 801-817.
- Roshani A., Fattahi P., Roshani A., Salehi M., Roshani A. (2012). Cost oriented two-sided assembly line balancing problem: a simulated annealing approach. *International Journal of Computing Integrated Manufacturing*, 25, 680–715.
- Roshani A., Nezami F.F. (2017). Mixed-model multi-manned assembly line balancing problem: a mathematical model and a simulated annealing approach. *Assembly Automation*, 37(1), 34-50.
- Rossi, F., Beek, P., Walsh, T. (2006). Handbook of Constraint Programming. Amsterdam: Elsevier Science, 3-10.
- Salveson, M.E. (1955). The assembly line balancing problem. *Journal of Industrial Engineering*, 6, 18-25.

- Scholl, A. (1999). Balancing and sequencing of assembly lines, *Heidelberg, Germany: Physica*.
- Scholl, A., Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168(3), 666–693
- Scholl, A., Boysen, N., Fliedner, M. (2008). The sequence-dependent assembly line balancing problem. *OR Spectrum*, 30 (3), 579–609.
- Scholl A., Boysen N., Fliedner M. (2013). The assembly line balancing and scheduling problem with sequence-dependent setup times: problem extension, model formulation and efficient heuristics, *OR Spectrum*, 35 (1), 291–320.
- Seyed-Alagheband S.A., Ghomi S.M.T.F., Zandieh M. (2011). A simulated annealing algorithm for balancing the assembly line type II problem with sequence-dependent setup times between tasks. *International Journal of Production Research*, 49(3), 805–825.
- Simaria, A.S. ,Vilarinho, P.M. (2009). 2-ANTBAL: An ant colony optimization algorithm for balancing two-sided assembly lines, *Computers & Industrial Engineering*, 56: 489-506.
- Sivasankaran, P., Shahabudeen, P. (2014). Literature review of assembly line balancing problems, *International Journal of Advanced Manufacturing Technologies*, 73, 1665-1694.
- Solnon, C. (2010). Ant colony optimization and constraint programming, *London: ISTE Ltd*, 27-50.
- Talbi, E. G. (2009). Metaheuristics: from design to implementation. *New Jersey: John Wiley & Sons*, 18-19, 418.
- Tapkan P., Özbakir L., Baykasoglu A. (2012). Modeling and solving constrained two-sided assembly line balancing problem via bee algorithms. *Applied Soft Computing*, 12, 3343–3355.
- Terekhov, D., Doğru, M. K., Özen, U., Beck J. C., (2012). Solving two-machine assembly scheduling problems with inventory constraints. *Computers & Industrial Engineering*, 63(1), 120-134.
- Topaloğlu, S., Salum, L., Supciller A. A. (2012). Rule-based modeling and constraint programming based solution of the assembly line balancing problem. *Expert Systems with Applications*, 39(3), 3484–3493.
- Valle, C.D., Marquez, A.A., Gasca, R.M., Toro, M. (2003). On selecting an scheduling assembly plans using constraint programming. *Lecture Notes in Computer Science*, 2774, 1329-1336.

- Yazdanparast, V.A., Hajihosseini, H.B., Bahalke, A.C. (2011). Cost oriented assembly line balancing problem with sequence dependent setup times, *Australian Journal of Basic and Applied Sciences*, 5(9), 878-884.
- Yegul MF., Agpak K., Yavuz M. (2010). A new algorithm for U-shaped two-sided assembly line balancing. *Trans Can Soc Mech Eng*, 34(2), 225–241.
- Yolmeh A., Kianfar F. (2012). An efficient hybrid genetic algorithm to solve assembly line balancing problem with sequence-dependent setup times, *Computers and Industiral Engineering*, 62, 936–945.
- White, W.W. (1961). Comments on a paper of Bowman. *Operations Research*. 9(2), 274–276.
- Zeballos, L. J. (2010). A constraint programming approach to tool allocation and production scheduling in flexible manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 26(6), 725-743.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : GÜNER, Funda
 Uyruğu : T.C.
 Doğum tarihi ve yeri : 04.01.1987, Zonguldak
 Medeni hali : Evli
 Telefon : 0 (312) 233 13 59
 Faks : 0 (312) 233 10 26
 e-mail : fkarabak@cankaya.edu.tr



Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Endüstri Mühendisliği	2019
Yüksek lisans	Çankaya Üniversitesi / Endüstri Mühendisliği	2012
Lisans	Çankaya Üniversitesi / İktisat (Çift Anadal)	2011
Lisans	Çankaya Üniversitesi / Endüstri Mühendisliği	2009
Lise	Zonguldak Atatürk Anadolu Lisesi	2004

İş Deneyimi

Yıl	Yer	Görev
2018-Halen	Çankaya Üniversitesi	Öğretim Görevlisi
2009-2018	Çankaya Üniversitesi	Uzman

Yabancı Dil

İngilizce

Yayınlar

Güner, F., Kellegöz, T., Toklu, B., *Multi-Manned Assembly Line Balancing Problems With Sequence-Dependent Setup Times*, 47. International Conference on Computers & Industrial Engineering, Lisbon, 11-13 October 2017, Portugal

- Karabak, F., Güner, N.D., Satır, B., Kandiller, L., Gürsoy, İ., *Montaj Hattı İstasyonlarına İşgücü Atanma Problemine Bir Yaklaşım ve Bir Uygulama*, Yöneylem Araştırması ve Endüstri Mühendisliği 31. Ulusal Kongresi, 5-6-7 July 2011, Sakarya, TÜRKİYE
- Karabak, F., Güner, N.D., Satır, B., Kandiller, L., Gürsoy, İ., *An Optimization Model For Worker Assignment of a Mixed Model Vehicle Production Assembly Line Under Worker Mobility*, 41. International Conference on Computers & Industrial Engineering, Los Angeles, 23-25 October 2011, CA, USA
- Karabak, F., Güner, N.D., Satır, B., Kandiller, L., Gürsoy, İ., *Man Türkiye A.Ş. Montaj Hattında İş Gücü Dengeleme Karar Destek Sistemi OTEKON'6*, 4-5 Haziran 2012, Bursa, TÜRKİYE



GAZİ GELECEKTİR...