

**T.C.  
YILDIZ TEKNİK ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**İLİŞKİSİZ PARALEL MAKİNELİ HİBRİT AKIŞ TİPİ SIRALAMA PROBLEMİNDE  
ÜSTÜNLÜK ÖZELLİKLERİ VE GENETİK ALGORİTMA  
KULLANARAK İŞ SIRALAMA**

**PELİN ALCAN**

**DOKTORA TEZİ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI  
ENDÜSTRİ MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN  
PROF. DR. HÜSEYİN BAŞLIGİL**

**İSTANBUL, 2014**

**T.C.**  
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**İLİŞKİSİZ PARALEL MAKİNELİ HİBRİT AKIŞ TİPİ SIRALAMA PROBLEMİNDE**  
**ÜSTÜNLÜK ÖZELLİKLERİ VE GENETİK ALGORİTMA**  
**KULLANARAK İŞ SIRALAMA**

Pelin ALCAN tarafından hazırlanan tez çalışması 23.10.2014 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Prof. Dr. Hüseyin BAŞLIGİL  
Yıldız Teknik Üniversitesi

**Jüri Üyeleri**

Prof. Dr. Hüseyin BAŞLIGİL  
Yıldız Teknik Üniversitesi

Prof. Dr. İlker BİRBİL  
Sabancı Üniversitesi

Yard. Doç. Dr. Tufan DEMİREL  
Yıldız Teknik Üniversitesi

Prof. Dr. İsmail ADAK  
Yalova Üniversitesi

Doç. Dr. Semih ÖNÜT  
Yıldız Teknik Üniversitesi

---

---

---

---

---

## ÖNSÖZ

---

Bu tez çalışmasında, her aşamada konu ile ilgili bilgi ve deneyimini benimle paylaşan, ilgi ve önerilerini, desteği ile birlikte hiçbir zaman esirgemeyen, tez danışmanlığımı büyük bir özenle yürüten değerli hocam Prof. Dr. Hüseyin BAŞLIGİL' e, tez çalışmalarımın, raporlarımın izlenmesinde ve değerlendirilmesinde değerli katkılarını asla esirgemeyen değerli hocalarım Yard. Doç. Dr. Tufan Demirel ve Prof. Dr. İlker BİRBİL'e, ayrıca Sabancı Üniversitesi'nde çalışan ve Modelleme prosesinde tez çalışmama büyük katkı sağlayan arkadaşım Halil Şen ve manevi destekleri için, ailem ile bütün can dostlarıma teşekkür ederim.

Haziran, 2014

Pelin ALCAN

## İÇİNDEKİLER

---

	Sayfa
SİMGE LİSTESİ .....	viii
KISALTMA LİSTESİ .....	xi
ŞEKİL LİSTESİ.....	xiii
ÇİZELGE LİSTESİ .....	xv
ÖZET.....	xvi
ABSTRACT .....	xvii
<b>BÖLÜM 1</b>	
GİRİŞ.....	1
1.1 Literatür Araştırması .....	3
1.1.1 Paralel Makine Çizelgelenmesi Hakkındaki Literatür Taraması.....	3
1.1.2 Hibrit Akış Tipi Hakkındaki Literatür Taraması.....	11
1.2 Tezin Amacı .....	33
1.3 Orjinal Katkı.....	34
<b>BÖLÜM 2</b>	
İŞ SIRALAMA ve ÇİZELGELEME .....	35
2.1 İş Sıralama Kabulleri .....	37
2.2 Kısıtlar.....	39
2.3 Çalışma Ölçütleri .....	39
2.4 Job-Shop Tipi Sıralama Problemlerinin Parametreleri .....	43
2.5 Çizelgeleme Problemleri .....	44
2.5.1 İşlerin Geliş Şekli .....	44
2.5.2 İşlerin Görüleceği Tezgâh Sayısı.....	45
2.5.3 Yerleşim Tipleri .....	45
2.5.3.1 Akış tipi (Flow shop) Çizelgeleme .....	45
2.5.3.1.2 Hibrit Akış Tipi.....	47

2.5.3.2	Sipariş (Atölye) Tipi (Job Shop) Çizelgeleme.....	47
2.6	Çizelgelemenin Başarılı Olabilmesi İçin Dikkat Edilmesi Gerekenler .....	47
2.7	Birleşik Eniyileme Durumu .....	49

### BÖLÜM 3

PARALEL MAKİNELERİN ÇİZELGELENMESİ .....	51
3.1 Paralel Makine Çeşitleri.....	52
3.1.1 İlişkisiz Paralel Makine ile Problem Tanımı .....	53
3.2 Paralel Tezgâhlardan Oluşan Atölyede Sıralama Problemi .....	53
3.2.1 Yapım Süresinin Minimizasyonu .....	54
3.2.1.1 Öne geçmeli halin incelenmesi .....	54
3.2.1.2 Öne geçmesiz halin incelenmesi .....	56
3.3 Paralel Makine Çizelgeleme Probleminin İfadesi .....	58
3.3.1 Özdeş Paralel Makinelerde Sıralama .....	58
3.3.2 Özdeş Olmayan Paralel Makinelerde Sıralama.....	59

### BÖLÜM 4

HİBRİT AKIŞ TİPİ SIRALAMA PROBLEMİ.....	61
4.1 Hibrit Akış Tipi Sistem'in Notasyonu.....	63

### BÖLÜM 5

ÇİZELGELEME YAKLAŞIMLARI.....	67
5.1 Analitik Yaklaşım .....	68
5.2 Sezgisel Yaklaşım .....	68
5.3 Meta-Sezgisel Yaklaşım .....	70
5.3.1 Yerel Arama Algoritması .....	74

### BÖLÜM 6

GENETİK ALGORİTMA .....	78
6.1 Genetik Algoritmaların Sahip Olması Gereken Koşullar .....	81
6.1.1 Genetik Algoritmanın Diğer Yöntemlerden Farkı .....	81
6.2 Genetik Algoritmada Kullanılan En Temel Kavramlar .....	82
6.2.1 Gen .....	83
6.2.2 Kromozom .....	83
6.2.3 Popülasyon .....	83
6.2.4 Allel.....	83
6.2.5 Locus.....	84
6.2.6 Genotip.....	84
6.2.7 Fenotip.....	84
6.3 Genetik Algoritmanın Çalışma Prensibi .....	84

6.3.1	Kodlama yöntemleri .....	87
6.3.2	Uygunluk teknikleri .....	87
6.4	Genetik Algoritma Operatörleri.....	88
6.4.1	Başlangıç Popülasyonu .....	89
6.4.2	Çaprazlama .....	89
6.4.2.1	Tek Nokta Çaprazlama.....	90
6.4.2.2	Çift Nokta Çaprazlama.....	90
6.4.2.3	Uniform Çaprazlama.....	91
6.4.2.4	Sıraya Dayalı Çaprazlama.....	91
6.4.2.5	Pozisyona Dayalı Çaprazlama Yöntemi.....	92
6.4.2.6	Kısmi planlı çaprazlama .....	92
6.4.3	Mutasyon.....	93
6.4.3.1	Ters mutasyon .....	93
6.4.3.2	Komşu iki gen değiştirme .....	94
6.4.3.3	Keyfi iki gen değiştirme .....	94
6.4.3.4	Keyfi üç gen değiştirme .....	94
6.4.3.5	Araya yerleştirme .....	94
6.4.4	Uygunluk Fonksiyonu.....	95
6.4.5	Üretim Operatörü .....	95
6.4.6	Kodlama .....	96
6.4.6.1	İkili sayı sisteminde kodlama.....	96
6.4.6.2	Permütasyon kodlama .....	96
6.4.6.3	Değer kodlama .....	97
6.4.6.4	Ağaç Kodlama .....	98
6.4.7	Seçim Operatörü .....	98
6.4.7.1	Rulet Çarkı Seçimi.....	99
6.4.7.2	Turnuva Seçimi .....	99
6.4.7.3	Kararlı durum seçimi .....	100
6.4.7.4	Elitizm (Seçkinlik) seçimi .....	100
6.5	Genetik Algoritmada Parametre Seçimi .....	100
6.5.1	Popülasyon Büyüklüğü.....	101
6.5.2	Çaprazlama Oranı .....	101
6.5.3	Mutasyon Oranı .....	102
6.5.4	Seçim Stratejisi.....	102

## BÖLÜM 7

KURULUM ZAMANLI SIRALAMA PROBLEMLERİ İÇİN DP YARATIMI .....	103
7.1 Üstünlük Özelliklerinin Türetilmesi ve Kanıtları.....	105
7.1.1 Makineler İçi değişim.....	105
7.1.2 Makineler Arası Değişim .....	107
7.2 Üstünlük Özellikleri Adımları ve Bu Konuda Yapılan Bazı Örnekler.....	109
7.2.1 Tek Makineli Örnek.....	109
7.2.2 İki Makineli örnek sisteminin incelenmesi .....	113

## BÖLÜM 8

UYGULAMA.....	119
8.1 Matematiksel Model .....	120
8.2 Matematiksel Modelin Uygulanması .....	124
8.3 Genetik Algoritma ve Üstünlük Özellikleri Kullanımı .....	127
8.3.1 Parametre Seçimi .....	128

## BÖLÜM 9

DENEYSEL TASARIM SONUÇLARI .....	138
----------------------------------	-----

## BÖLÜM 10

SONUÇ VE ÖNERİLER .....	152
-------------------------	-----

KAYNAKLAR.....	155
----------------	-----

ÖZGEÇMİŞ.....	169
---------------	-----

## SİMGE LİSTESİ

---

C	Tamamlanma zamanı
F	Akış zamanı
T	Gecikme
E	Erken bitirme
C <sub>max</sub>	Maksimum tamamlanma zamanı
d <sub>i</sub>	Teslim zamanı
F <sub>max</sub>	Maksimum akış zamanı
w	Ağırlık
S <sub>t</sub>	Hazırlık zamanı
R	Kaynak
T	İş
a <sub>i</sub>	Müsaade edilen zaman
W <sub>ik</sub>	İ. işinin, k' ıncı işlemde önce geçen bekleme zamanıdır
W <sub>i</sub>	İ. işinin toplam bekleme zamanı
r <sub>i</sub>	Hazırlık zamanı
P <sub>ji</sub>	j makinesinde i işinin işlem zamanı
X	İşlerin ortalaması
N <sub>w</sub> (t)	t anında işlenmeye hazır olmayan işlerin sayısı.
N <sub>c</sub> (t)	t anında tamamlanmış olan işlerin sayısı
N <sub>p</sub> (t)	t anında işlenmekte olan işlerin sayısı.
N <sub>u</sub> (t)	t anında tamamlanacak olan işlerin sayısı
P <sub>m</sub>	Paralel benzer makine
Q <sub>m</sub>	Farklı hızlardaki paralel makine
R <sub>m</sub>	Birbirine paralel ilişkisiz makineler
F <sub>m</sub>	Flow shop hali
FF <sub>s</sub>	Esnek flow shop hali
O <sub>m</sub>	Open shop hali
J <sub>m</sub>	Job shop hali
prmp	Alıkoyma hali
prec	Öncelik zorlama hali



brkdown	Bozulma hali
$M_j$	Seçilebilir makine kısıtları.
prmu	Yer deęiřtirme
block	önleme
n	iřlerin sayısı
m	makine sayısı
A	Makine atölyesinde uygulanacak model veya akıř modeli.
G	Teknolojik sınırlamalar halinde hiřbir kısıtlamanın olmadıęı genel job-shop durumu
B	Programın deęerlendirilmesi ile performans ölçütü
$M_k$	k ařamasındaki makine sayısı
$\Omega$	çözüm uzayı
$r_j$	j iřinin bırakıř zamanı
$p_{ij}^t$	t ařamasında i makinesinde j iřinin proses zamanı
$v_{ij}^t$	t ařamasında i makinesinde j iřinin nispi hızı.
$M^*$	Yapım süresinin minimum deęeri
$X(i,j)$	mantıksal boolean deęiřkeni
$O_{jk}$	k ařamasında j iřinin iřlendięi operasyon zamanı
$\alpha$	HFS sisteminde atölye biçimi
$\beta$	HFS sisteminde sabitler ve varsayımlar
$\sigma$	HFS sisteminde dikkate alınan amaç fonksiyonu
unavail	makinaların her zaman hazır olmaması
no-wait	bekleme yok
$Z(\Pi)$	DP içerisinde alınan objektif fonksiyonu
$P_{[j][k]}$	k makinesinde j pozisyonundaki iřin proses zamanıdır.
$S_{[i][j][k]}$	k makinesinde, i iřinden sonra j pozisyonundaki iřin kurulum zamanı.
$AP_{[i][j][k]}$	k makinesinde i iřinden sonra j pozisyonundaki iřin düzeltilmiř proses zamanı
$C_{k1}$	k1 deki tamamlanma zamanı
$G_{1k}$	k makinesinde i iřinden önceki iř dizisi
$G_{2k}$	k makinesinde [i] iři ve [j+1] iřleri arasındaki iř dizisi
$G_{3k}$	k makinesinde [i] iřinden sonraki iř dizisi
$l, j, i$	iř indisi
k	ařama indisi
m	makine indisi
abe	iř sayısı
mc	makine sayısı
$b_k$	k ařamasında, eđer j iři m makinesine atanan ilk iř ise, j iřinin hazırlık zamanı
$r_{kmj}$	Müsaade edilen zaman
$R_{km}$	k ařamasında, m makinesinin hazır olduęu zaman
$O_{kj}$	k ařamasında, j iřinin operasyon zamanı
F	çok büyük bir sayı
$a_i$	Müsaade edilen zaman
$W_{ik}$	i. iřinin, k' ıncı iřlemeden önce geçen bekleme zamanıdır
$W_i$	i. iřinin toplam bekleme zamanı
$r_i$	Hazırlık zamanı
$P_{ji}$	j makinesinde i iřinin iřlem zamanı
X	iřlerin ortalaması

$N_w(t)$  t anında işlenmeye hazır olmayan işlerin sayısı.  
 $N_c(t)$  t anında tamamlanmış olan işlerin sayısı  
 $N_p(t)$  t anında işlenmekte olan işlerin sayısı.

## KISALTMA LİSTESİ

---

ACO	Karınca kolonisi Optimizasyonu
ANN	Yapay sinir ağları yaklaşımı
AP	Ayarlanmış proses zamanı
B&B	Dal sınır algoritması
BA	En iyi kabul stratejisi
BBFFL	Darboğaz tabanlı sezgisel
BFLPT	En uygun en uzun proses zamanı sezgiseli
DEDD	Dinamik en erken teslim tarihi algoritması
DLPT	Dinamik en uzun proses zamanı algoritması
DP	Üstünlük özellikleri
ES	Uzman sistemler yaklaşımı
FA	İlk kabul stratejisi
FR	Sık gevşet sezgiseli
GA	Genetik Algoritma
GRASP	Greedy Randomized Adaptive Search Procedure
HFS	Hibrit Flow Shop
HFSMT	Çok işlemcili iş sıralama problemi
IA	Immün algoritma
ICS	cuckoo araştırması
ILS	Tekrarlı lokal arama
LP	Lineer Programlama
LPT	En uzun işlem zamanı önce algoritması
MDD	Değiştirilmiş teslim tarihi kuralı
MEs	İkili kodlama tabanlı metasezgisel
MIP	Karma Tamsayı Programlama
MP	Matematiksel Programlama
MPGA	İki safhalı genetik algoritma
OBX	Sıraya dayalı çaprazlama
PBX	Pozisyona dayalı çaprazlama
PGA	Paralel greedy algoritması
PMX	Kısmi planlı çaprazlama
PTAS	Polinomial zaman yaklaştırma algoritması şeması

RH	Döner Horizon sezgiseli
SA	Benzetimli tavlama
SK	Sevkiyat Kuralları
TA	Tabu Arama
TSP	Gezgin satıcı problemi
YAA	Yerel arama algoritması

## ŞEKİL LİSTESİ

Sayfa

Şekil 1.1	Amaç fonksiyonlarının dağılımı; C/F = Toplam/Ortalama Tamamlanma zamanı veya akış zamanı, T=Toplam/Ortalama Gecikme, E/T= Erken bitirme veya gecikmenin toplam/ ortalama miktarı, Multi= Çoklu amaç değerleri.15	15
Şekil 1.2	B&B (Dal-Sınır), TS (Tabu arama), GA (Genetik Alg.), SA (Benzetimli Tavlama), MP (Matematiksel Programlama), SK (Sevkiyat Kuralları), Sezgiseller ve D. (Diğer yöntemler)' in HFS makalelerinde dağılımı .....	16
Şekil 2.1	Gantt diyagramı .....	43
Şekil 3.1	Paralel makine sistem modeli .....	52
Şekil 3.2	8 iş 3 makine halinin gantt diyagramında gösterimi.....	56
Şekil 3.3	LPT' ye göre oluşturulan Gantt diyagramı .....	57
Şekil 4.1	HFS yapısı .....	62
Şekil 4.2	HF2 (P3,P2) //Cmax yapısı.....	65
Şekil 5.1	Global ve Yerel Optimum.....	75
Şekil 6.1	Genetik algoritmanın akış diyagramı .....	86
Şekil 6.2	Tek noktalı çaprazlama .....	90
Şekil 6.3	Çift Nokta Çaprazlama .....	91
Şekil 6.4	Sıraya dayalı çaprazlama.....	91
Şekil 6.5	Pozisyona dayalı çaprazlama .....	92
Şekil 6.6	Komşu iki geni değiştirme .....	94
Şekil 6.7	Keyfi iki geni değiştirme .....	94
Şekil 6.8	Keyfi üç geni değiştirme.....	94
Şekil 6.9	Üretim operatörü.....	96
Şekil 6.10	Ağaç kodlamalı kromozomlar örneği .....	98
Şekil 6.11	Turnuva Seçimi yöntemi .....	100
Şekil 7.1	Bir paralel makine gösterimi .....	104
Şekil 7.2	Aynı makine üzerinde değişim .....	104
Şekil 7.3	Bitişik olmayan işlerin değişimi .....	107
Şekil 7.4	Farklı makinelerdeki i ve j işlerinin yer değişimi .....	109
Şekil 7.5	İşlemlerin sonundaki GANT Diyagramı .....	118
Şekil 7.6	Başlangıç aşamasındaki GANT Diyagramı .....	118
Şekil 8.1	UHFS-OPL örnek yaratımı.....	126
Şekil 8.2	OPL' de 15/5/2 test verisi için örnek ekranı.....	127
Şekil 8.3	Turnuva seçim mekanizması .....	129
Şekil 8.4	15 İş 2 Aşama 5 Makine (10 döngü).....	131
Şekil 8.5	15 İş 2 Aşama 5 Makine (20 döngü).....	132

Şekil 8.6	10 iş 5 makine 3 aşama (10 döngü) .....	132
Şekil 8.7	4 iş 2 aşama 2 makine (10 döngü) .....	133
Şekil 8.8	4 iş 2 aşama 4 makine (10 döngü) .....	133
Şekil 8.9	6 iş 3 aşama 3 makine (10 döngü) .....	134
Şekil 8.10	6 iş 2 aşama 3 makine (10 döngü) .....	134
Şekil 8.11	8 iş 3 aşama 4 makine (10 döngü) .....	135
Şekil 8.12	8 iş 4 aşama 4 makine (10 döngü) .....	135
Şekil 8.13	9 iş 3 aşama 3 makine (10 döngü) .....	136
Şekil 9.1	JMP programı açılış sayfası .....	139

## ÇİZELGE LİSTESİ

---

	Sayfa
Çizelge 1.1 Paralel Makinelerin HFS problemlerinin araştırıldığı makalelerde, çeşitlerine ve kademe sayılarına göre dağılımı.....	16
Çizelge 1.2 Paralel Makine Çizelgelenmesi Hakkındaki Literatür Taraması.....	17
Çizelge 1.3 Hibrit Akış Tipi Hakkındaki Literatür Taraması.....	25
Çizelge 2.1 Çizelgelemedeki Temel Notasyonlar.....	42
Çizelge 3.1 Öne geçmeli hal ile ilgili bir örnek.....	55
Çizelge 5.1 Metasezgisel ve sezgisel yaklaşım arasındaki farklar.....	74
Çizelge 6.1 İkili sayı sisteminde kodlanmış örnek kromozomlar.....	96
Çizelge 6.2 Permütasyon kodlama sistemi ile kodlanmış örnek kromozomlar.....	97
Çizelge 6.3 Değer kodlama sistemi ile kodlanmış örnek kromozomlar.....	97
Çizelge 7.1 Makine 1 için AP' ler.....	110
Çizelge 7.2 İki Makineli bir paralel makine örneğinde AP Zamanları.....	113
Çizelge 7.3 Makinelere atanan işlerin durumları.....	114
Çizelge 7.4 Makinelere atanan işlerin durumları.....	114
Çizelge 8.1 Başlangıç test aşamasında düşünülen parametreler.....	125
Çizelge 8.2 Başlangıç test aşamasında düşünülen faktörler.....	125
Çizelge 8.3 OPL sonucundaki Örnek Veriler Tablosu.....	126
Çizelge 8.4 Kullanılan Parametreler.....	130
Çizelge 8.5 Örnek Veriler Tablosu (Tüm sonuçlar için).....	136
Çizelge 9.1 4iş-2aşama-2makine için "genelleştirilmiş lineer analiz" sonuçları.....	140
Çizelge 9.2. Genelleştirilmiş Lineer Analiz içerisinde kodlanan değerler.....	141
Çizelge 9.3 4iş-2aşama-4makine için "genelleştirilmiş lineer analiz" sonuçları.....	142
Çizelge 9.4. 6iş-3aşama-3makine için "genelleştirilmiş lineer analiz" sonuçları.....	143
Çizelge 9.5. 6iş-2aşama-3makine için "genelleştirilmiş lineer analiz" sonuçları.....	144
Çizelge 9.6. 8iş-3aşama-4makine için "genelleştirilmiş lineer analiz" sonuçları.....	145
Çizelge 9.7. 8iş-4aşama-4makine için "genelleştirilmiş lineer analiz" sonuçları.....	146
Çizelge 9.8. 9iş-3aşama-3makine için "genelleştirilmiş lineer analiz" sonuçları.....	147
Çizelge 9.9. 10iş-3aşama-5makine için "genelleştirilmiş lineer analiz" sonuçları.....	148
Çizelge 9.10. 15iş-2aşama-5makine için "genelleştirilmiş lineer analiz" sonuçları.....	149
Çizelge 9.11. 18iş-4aşama-6makine için "genelleştirilmiş lineer analiz" sonuçları.....	150
Çizelge 9.12 Örnekler üzerindeki dik tasarım sonuçları.....	151

---

**İLİŞKİSİZ PARALEL MAKİNELİ HİBRİT AKIŞ TİPİ SIRALAMA PROBLEMİNDE  
ÜSTÜNLÜK ÖZELLİKLERİ VE GENETİK ALGORİTMA  
KULLANARAK İŞ SIRALAMA**

Pelin ALCAN

Endüstri Mühendisliği Anabilim Dalı  
Doktora Tezi

Tez Danışmanı: Prof. Dr. Hüseyin BAŞLIGİL

Günümüzde gelişen teknoloji, ürünlerle birlikte üretim sistemlerini de oldukça karmaşık hale getirmiştir. Sanayi devrimi ile ürünlerin daha kolay ulaşılabilir olması, müşterilerin taleplerini arttırmış, artan müşteri taleplerini karşılamak için de işletmeler, üretim kapasitelerini arttırmaya başlamışlardır. Literatür araştırması sonucunda “Hibrit Akış Tipi, ilişkisiz paralel makineli ve hazırlık zamanı kısıtlı” sistemler üzerinde yapılan çalışmaların az olduğu, hele “üstünlük özellikleri” gibi sezgisellerin işini kolaylaştıran yöntemlerin çok da kullanılmadığı görülmüştür. Çalışmaların çoğuna yakınında sezgisel ve meta sezgisel yöntemler denenmiştir. Problemlerin oldukça zor olması da yöntem seçimlerinde etkili rol oynamaktadır. Bu tez çalışması kapsamında sunulan problemin çözümünde, evrimsel algoritmalarından genetik algoritma, üstünlük özellikleri konusu ile birlikte denenmiştir.

**Anahtar Kelimeler:** Hibrit Akış Tipi, çizelgeleme, sezgisel, metasezgisel, genetik algoritma, üstünlük özellikleri.



---

**JOB SCHEDULING WITH THE HELP OF DOMINANCE PROPERTIES AND  
GENETIC ALGORITHM ON HYBRID FLOW SHOP PROBLEM  
WITH UNRELATED PARALLEL MACHINE**

Pelin ALCAN

Department of Industrial Engineering  
PhD. Thesis

Adviser: Prof. Dr. Hüseyin BAŞLIGİL

Today's developed technology makes not only products but also manufacturing systems complex. Products are accessed easier with the industrial revolution. Product accessibility increases the customer demand. Consequently, to satisfy increased customer demands companies expand their manufacturing capacities. After a literature review, we determined that there is hardly any study on "unrelated parallel machine and set up time constrained Hybrid Flow Shop" problems. Specifically, the techniques, i.e, dominance properties, that help heuristics methods are not used. In almost all studies, either heuristics or meta-heuristics methods are applied. The problem complexity plays an important role in selecting the solution methodologies. In this dissertation, genetic algorithm, which is an evolutionary algorithm, with dominance property is used to solve the proposed problem.

**Keywords:** Hybrid Flow Shop, Scheduling, heuristic, metaheuristic, Genetic Algorithm, Dominance Properties.

## BÖLÜM 1

---

### GİRİŞ

Günümüzde gelişen teknoloji ve bilim, ürünlerle birlikte üretim sistemlerini de oldukça karmaşık hale getirmiştir. Endüstriyel devrim ile ürünlerin daha kolay ulaşılabilir olması, müşterilerin taleplerini arttırmış, artan müşteri taleplerini karşılamak için işletmeler de, üretim kapasitelerini arttırmaya başlamışlardır. Sonuçta, büyüyen işletmelerde üretim kolayca koordine edilemez hale gelmiştir. Bu durum doğal olarak, üretim planlama ve kontrol sistemleri ihtiyacını doğurmuştur.

Herkesin bilebileceği gibi, hayatın büyük bir bölümü sıralama problemlerinden oluşmaktadır. Bu durum bilinçli ve/veya bilinçsizce olabilmektedir. Sıralama problemlerinde ise en temel felsefeler; zamanlama ve maliyettir. Çünkü sıralamada zaman kavramı, maliyetle beraber hareket eder. Minimum zaman içerisinde, bir işin yapılması önemli olmaktadır. Sıralama prosesi; üretim, yönetim, bilgisayar bilimleri ve bunun gibi daha pek çok alanda kullanılan önemli bir prosestir. Sıklıkla, ihtiyaç duyulan kısıtların altında, kaynakların yüksek zamanlı işlere atanmasıyla ilgilenmektedir. Uygun olan sıralama şekli, bütün zamanı küçültürken, ürün kalitesini arttırandır.

Çizelgeleme, belli bir takım işleri yapmak adına, hangi kaynakların, ne zaman ve nasıl kullanılacaklarının tespit edilmesi işlemidir. Sıralama problemleri, birçok değişken ihtiva eden ve çözümü zor olan problemlerdir. Akış tipi çizelgeleme problemleri, birleşik eniyileme problemi özelliğindedirler ve “NP-zor (NP-hard)” problem sınıfındadırlar.

Hibrit Akış Tipi (Hybrid Flow Shops-HFS) ise,  $m$  aşamalı seri bir sistemde, bir grup işin olduğu, genel bir üretim sistemidir. HFS sistemi, bir dizi aşamadan ve her aşamada da bir ya da daha

çok paralel makinelerden oluşmaktadır. HFS probleminin “standart” formunda bütün işler ve makineler zamanın sıfır anında hazırdir. Verilen bir aşamadaki makineler özdeştir. Herhangi bir makine bir zamanda sadece bir işi işleme tabi tutabilmekte ve herhangi bir iş, belli bir zamanda bir makine ile işleme girebilmektedir. Hazırlık zamanları ihmal edilebilir. Aşamalar arasında bulunan ara stok kapasiteleri sınırsızdır. Problem bilgisi deterministik yani kesindir ve önceden bilinmektedir.

Bu çalışma içerisinde, öncelikle “iş sıralama” problemleri içerisinde “Hibrit Akış Tipi (HFS)” sistemler incelenmiş ve ardından bu konu üzerinde bir model düşünülmüştür. HFS yapısı içerisinde, her aşamada ise, literatürde bu tip sistemler içinde oldukça az bulunan “ilişkisiz paralel makineler” kullanılmıştır. Yapılan modellemede görülmüştür ki, örnek boyutu büyüdükçe model yavaşlamaktadır. Daha sonra modelin çözümüne yardımcı olması açısından “sezgisel” teknikler denenmiştir. Sezgisel tekniklerin en çok kullanılanlarından birisi olan “genetik algoritma” yöntemi uygulanmıştır.

Genetik algoritmalar problemlerin çözümünü parametrelerin değerleriyle değil, kodlarıyla arayan “meta sezgisel” yöntemlerdir. Yani parametreler kodlanabildiği sürece çözüm üretilebilmektedirler. Genetik algoritmalar aramaya tek bir noktadan değil, noktalar kümesinden başlarlar. Bu nedenle çoğunlukla “yerel en iyi” çözümde sıkışıp kalmadıkları görülür. En önemli özelliklerinden birisi de budur. Parametrelerin kodları önemlidir. Diğer bir önemli nokta ise, kromozomların eşlenmesi durumudur. Bu işlem, Kromozomların uygunluk değerlerine göre yapılır. Bir sonraki nesil için anne ve baba olacak kromozomlar, seçim metotlarından herhangi birisi ile belirlenmektedir.

Genetik algoritmanın ise, kullanılabilirliğinin ve etkinliğinin daha arttırılması amacıyla, bu çalışma içerisinde ayrıca “üstünlük/baskınlık özellikleri” kullanılmıştır. Verilen bir sıralama işlemi için “baskınlık özellikleri (Dominance Properties-DP)” kullanılarak, optimale yakın bir çözüm elde edilebilmektedir. Büyük boyutlu problemlerin çözümü zor olmaktadır. Bunun için, çözüm kalitesini arttırmaya yardımcı olarak, baskınlık özelliklerinin genetik algoritma ile birleştirildiği, yeni bir meta sezgisel ortaya konulabilir.

Tezin ilerleyen bölümleri şu şekilde organize edilmiştir: Bölüm 1’de İlk olarak, paralel makine çizelgelenmesi hakkında bir literatür taraması yapılmıştır. Daha sonra ise, Hibrit Akış Tipi sistemlerde bir literatür taraması yapılarak, bu sistemler incelenmiştir. Bölüm 2’de İş Sıralama ve Çizelgeleme kısmı anlatılmıştır. 3. Bölüm’de Paralel Makinelerin Çizelgelenmesi, 4. Bölüm’de Hibrit Akış Tipi Sıralama Problemi, 5. Bölüm’de Çizelgeleme Yaklaşımları, 6. Bölüm’de ise Genetik Algoritma genel hatları ile çizilmiştir. Bölüm 7’ de Kurulum Zamanlı Sıralama Problemleri İçin Üstünlük Özellikleri Yaratımı irdelenirken, Bölüm 8’de yapılan Uygulama anlatılmıştır. 9. Bölüm’de Deneysel Tasarım Sonuçları verilmiştir. 10. Bölüm’de sonuçlar ve öneriler anlatılmıştır.

## 1.1 Literatür Araştırması

### 1.1.1. Paralel Makine Çizelgelenmesi Hakkındaki Literatür Taraması

Çizelgeleme ve çok ölçütlü çizelgeleme konuları, son yıllarda araştırmacıların en çok ilgisini çeken konulardandır. Bu bağlamda “tek” makineli ve “akış tipi” sistemler üzerinde literatürde oldukça çalışma yapılmış olmasına karşın, paralel makineli sistemler konusunda çok daha az çalışma yapılmıştır.

Kravchenko ve Werner (1997) [1]’ in çalışmasında, tamamlanma zamanı problemi, iki benzer paralel makine için polinomial zamanda çözülemeyen bir durum olmuştur. Söz konusu problem, keyfi makinelerle genelleştirilen, polinomial zamanda çözülemeyen (NP-hard) ve analizi sezgisel yöntemlerle yapılan bir problemdir. Makalede birli NP-hard (unary NP-hard) yöntemi hazırlık zamanı kısıtları için geliştirilmiştir.

Park ve Kim (1997) [2]’in makalelerinde “benzer” paralel makinelerdeki bir sıralama problemi için, iki ayrı sezgisel önerilmiştir: bunlar tavlama benzetimi (Simulated Annealing) ve tabu arama (Tabu Search) sezgiselleridir.

Torres ve arkadaşları (1997) [3] makalelerinde  $M$  adet benzer paralel makinede,  $N$  işli sıralama probleminin çok kriterli halini tartışmışlardır. Amaç, bekleyen işlerin sayılarını ve ortalama akış zamanını eş zamanlı olarak minimize etmektir. Dört sezgisel; tavlama benzetimi ve komşuluk yaklaşımı (neighborhood search) tabanlı olarak gösterilmiştir.

Chiu ve arkadaşları (1999) [4] karma bir tamsayı programlama modelini, sıralı paralel makine operasyonları içinde göstermişlerdir. Ardından genetik tabanlı bir algoritmayı optimal paralel operasyon sırasının bulunması için önermişlerdir.

Min ve Cheng (1999) [5] çalışmalarında, benzer paralel makine sıralama problemi için, tamamlanma zamanını minimize etmek amaçlı bir çeşit genetik algoritma önermişlerdir.

Gupta ve Torres (2000) [6] makalelerinde  $n$  iş  $m$  benzer makine sıralama problemi için yeni bir sezgisel önermişlerdir. Amaç; optimal toplam akış zamanını elde etmek için maksimum tamamlanma zamanını minimize etmektir.

Laguna ve arkadaşları (2000) [7] ise çalışmalarında dağılım araştırmasına (scatter search) örnek bir yöntem kullanmıştır. Dağılım araştırması; “popülasyon” tabanlı bir metodoloji içermektedir. Burada “sözde evrimsel metotlar” (so-called evolutionary methods) ile özellikler paylaşılmaktadır.

Gupta ve Ho (2001) [8] iki paralel benzer makinede sıralanan işlerin problemi için, polinomial hesaplama zamanı içerisinde, problemin tahmini olarak çözümünü veren, bir “çoklu değiştirilmiş algoritma” (modified multifit algorithm) tanımlamıştır.

Bank ve Werner (2001) [9] çalışmalarında, ilişkisiz paralel makineler üzerinde, toplam ağırlıklandırılmış gecikme ve erken başlama cezalarını minimize eden bir sıralama problemi için, bazı “yapısal algoritmalar” ile “lokal yaklaşım” sezgisellerini öne sürmüşlerdir.

Mokotoff ve Chretienne (2002) [10] ile Liaw (2003) [11] da çalışmalarında sezgisellerden faydalanmışlardır. Dal-sınır algoritması ve tabu arama yöntemlerini ilişkisiz paralel makine sıralama problemi için, tamamlanma zamanı minimizasyonunda önermişlerdir.

Alagöz ve Azizoğlu (2002) [12] çalışmalarında, “makine seçilebilirlik” kısıtları altında, paralel makine üzerinde yeniden sıralama problemini tanımlarken, bir “Lineer Programlama (LP)

modeli" önermiştir. Ayrıca hiyerarşik probleme uygulanmak üzere bir dal-sınır algoritması da geliştirmişlerdir.

Meyr (2002) [13] benzer olmayan paralel üretim hatlarında, meta-stratejilerden; basamak kabulü, tavlama benzetimi ve "çift yeniden optimize etme" metodlarının kombinasyonunu çalışmıştır.

Cochran ve arkadaşları (2003) [14] ise makalelerinde çok amaçlı paralel makine çizelgeleme probleminin çözümü için iki safhalı genetik algoritma tekniğini (MPGA) önermiştir.

Liu ve Wu (2003) [15] makalelerinde, "evrimsel programlama" metodunu, benzer paralel makine üretim hattı çizelgeleme probleminde geciken işlerin minimize edilmesi için uygulamışlardır.

Lin ve Liao (2003) [16] ise makalelerinde minimum akış zamanı durumu için tamamlanma zamanı mimizasyonu amaçlı benzer paralel makine probleminde "Lexicographic araştırma tabanlı" bir algoritmayı ve "alt sınır hesabı" (the lower bound calculation) ile "işlerin yer değiştirmesi" kurallarını (the job replacement) kullanmışlardır.

Peng ve Liu (2004) [17] çalışmalarında bulanık proses zamanlarıyla modellenen paralel makine sıralama problemi için bir metodoloji geliştirmiştir. Çalışma içerisinde, 3 adet yeni bulanık sıralama modeli gösterilmiştir. Hibrit bir "intelligent algoritma (zeki algoritmalar)" bütün bu modellerin çözümü için tasarlanmıştır. Sonuçta, bazı nümerik örnekler önerilen algoritmanın hesapsal verimliliğini kanıtlamak için gösterilmiştir.

Chen ve Wu (2006) [18] makalelerinde "basamak-kabulü metodu" (threshold-accepting method), tabu listeleri ve prosedür geliştirme tabanlı bir sezgiseli, toplam gecikmeyi minimize etmesi için önermişlerdir.

Gupta ve Torres (2005) [19] makalelerinde ortalama akış zamanı ve geciken iş sayılarını hesaba katan ve paralel makinelerde sıralanan iş problemlerini tanımlarken, genetik

algoritma, tabu arama, benzetim tavlama gibi sezgiselleri kıyaslamış ve çözüm için önermiştir.

Cao ve arkadaşları (2005) [20] çalışmalarında tabu arama mekanizmasını temel alan sezgisel bir algoritmayı, optimal veya optimale yakın çözümlerin bulunması için geliştirmişlerdir.

Ayrıca Nessah ve arkadaşları (2006) [21] ile Dunstall ve Wirth (2004) [22] de, benzer paralel makine sıralama problemi için, bir dal sınır algoritması önermişlerdir. Nessah ve arkadaşları (2006) [21] makalelerinde benzer paralel makine sıralama problemini incelemişlerdir. Toplam tamamlanma zamanının minimize edilmesi için, “boşta kalma süresi” ve “hazırlık zamanı tabanlı sıralama” tartışılmıştır. Sezgisel yöntem olarak, burada da dal-sınır algoritması kullanılmıştır. Böylece daha düşük sınırlar ve geçerli şartlar birleştirilmiştir.

Ghirardi ve Potts (2004) [23] makalelerinde tamamlanma zamanını minimize etmek amaçlı ilişkisiz paralel makinelerde, “beam search (taban arama)” algoritmasını denemişlerdir. Çalışma içerisinde, “geleneksel beam search algoritması”, kesikli dal-sınır algoritmasının bir parçası olmuştur.

Chang ve arkadaşları (2005) [24] ise çalışmalarında çok amaçlı sıralama probleminin çözümünde değişik bir genetik algoritma tekniğini (TPSPGA) denemişlerdir.

Toledo ve Armentano’ nun (2006) [25] makalesinde ilişkisiz paralel makinelerde çok parçalı üretim için “Lagrangian gevşetmesi” (Lagrangian relaxation) sezgiseli önerilmiştir.

Rocha ve arkadaşları (2006) [26] ise çalışmalarında ilişkisiz paralel makineler için bir dal-sınır algoritması geliştirmiş, *metaheuristic GRASP* yöntemi ile bir üst sınır olarak kullanılacak bir çözüm üretmişlerdir.

Torres ve arkadaşları (2007) [27] makalelerinde üniform paralel makinelerdeki işlerin sıralama problemini incelemişlerdir. Burada makine hızı ikinci bir kaynağın tahsisiyle kontrol edilmiştir. Geç kalan iş sayılarının minimize edilmesi önemlidir. 5 tip sezgisel önerilmiştir.

Li, Sivakumar ve Ganesan (2008) [28] ise çalışmalarında paralel makine montaj sıralama problemini “karma bir tamsayı programlama” modeli olarak formüle etmişlerdir. Sezgisel algoritma tabanlı benzetimli tavlama yöntemi, sıralama problemi çözümü için geliştirilmiştir.

Low (2005) [29] ise çalışmasında, ilişkisiz paralel makinelerle ilgili çok safhalı akış atölye tipi sıralama probleminde, benzetim tavlama (SA) tekniğini önermiştir.

Blazewicz v.d (2008) [30] iki makineli Akış Tipi sıralama probleminde, ağırlıklandırılmış gecikme ile teslim tarihini incelemişlerdir. Bunun için tabu arama ve benzetimli tavlama sezgisellerini kullanmışlardır.

Beraldi ve arkadaşları (2008) [31] yine aynı yıl içerisinde, “döner horizon” (rolling-horizon - RH) ile “sık-gevşet” (FR) sezgisellerini benzer paralel makine parti boyutu ve sıralama problemi (kurulum maliyetleri tabanlı sıralama) üzerinde geliştirmişlerdir. Bu durum, önerilen yaklaşımı daha geniş boyutlar için uygulanabilir hale sokmuştur. Makaledeki sezgiseller “dağılım yaklaşımı” (decomposition approach) tabanlıdır. RH sezgiselleri, genelde dinamik parti boyutu ve sıralama problemlerinde kullanılmaktadır. FR sezgiseli ise geniş boyutlu bir MIP (karma tamsayılı programlama) problemine ayrılmaktadır.

Biskup ve arkadaşları (2008) [32] çalışmalarında, toplam gecikmeyi azaltmak için benzer paralel makine sıralamasını incelemişlerdir. NP-hard’ın doğası gereği ise yeni bir sezgisel yaklaşım önerilmiştir. MDD (değiştirilmiş teslim tarihi kuralı) sezgiseli ve KPM sezgiseli anlatılmıştır.

Chen ve Chen (2009) [33] makalelerinde, tamamlanma zamanını minimize etmek için, ilişkisiz paralel makinelerde, esnek bir akış hattı üstünde, darboğaz tabanlı (bottleneck-based heuristic) (BBFFL) bir sezgisel denemişlerdir. Buradaki temel fikir; darboğaz aşamasında sıralanan işlerin, bütün aşamalarda sıralanan işlerin sezgisel performansını etkileyebilir olmasıdır.



Chen ve Chen (2009) [34] aynı yıl çıkarttıkları bir başka makalelerinde, ilişkisiz paralel makinelerle esnek akış hattını, “darboğaz tabanlı” sezgisellerle incelemişlerdir. Amaç maksimum gecikmeyi minimize etmektir. 3 makine seçim kuralı ile 2 darboğaz tabanlı sezgisel, problemi çözmek için önerilmiştir. Darboğaz tabanlı sezgisellerin performansını değerlendirmek için, çeşitli “dağıtım kuralları” ve bir tabu arama algoritması kullanılmıştır.

Rossi ve Boschi (2009) [35] benzer paralel makine atölye tipi sıralama probleminde, çözüm için bir hibrit sezgisel denemişlerdir. Burada birleşik optimizasyon problemlerinin geniş bir sınıfı için, en çok önerilen 2 adet sezgisel kullanılmıştır. Bunlar genetik algoritma ve “karınca kolonisi” (Ant Colony) optimizasyon tekniğidir.

Yang (2009) [36] benzer paralel makine sıralama probleminde, evrimsel bir simülasyon-optimizasyon yaklaşımı denemiştir. Burada anlatılan “wire bonding (tel sarma)” sıralama problemi, benzer paralel makine sıralama problemi üzerinde denenmiştir. Problem, kesikli-olay simülasyonu ile modellenmiştir.

Agnetis vd. (2009) [37] çalışmalarında, min-max problemleri için paralel makinelerde sezgisel çözümler üretmişlerdir. Aslında sezgiselin kalitesini ölçmek için bir araç önermişlerdir. Çalışmanın özünde, “sınırlandırılmış bir şema (bounding scheme)” göstermişlerdir. Buradaki min-max durumu, gecikmenin veya tamamlanma zamanının minimizasyonu veya maksimumu olmaktadır. Bu amaçlarla kullanılan algoritmalar; dinamik-en erken-teslim-tarihi (DEDD) ve dinamik-en uzun-proses-zamanı (DLPT) algoritmaları olmuştur.

Wang ve Cheng (2009) [38] çalışmalarında, müşterilere giden işlerin son ulaşım zamanının ağırlıklandırılmış toplamının ve dağıtım maliyetinin minimizasyonu problemlerini ele almışlardır. Problem benzer paralel makineler üzerinde ilerlerken bazı sezgiseller önerilmiştir. Tamamlanma zamanını minimize eden klasik benzer paralel makine sıralama problemi NP-hard’tır. Kullanılan ilk sezgisel, “en geniş dağıtım zamanı ilk” kuralı ile bir liste sıralamasının kullanıldığı bir sezgisel olmuştur. İkinci sezgisel ise, “en küçük makineye ilk yükleme” kuralıdır.

Xu ve arkadaşları (2009) [39] çalışmalarında iki paralel makine sıralaması için tamamlanma zamanı minimizasyonunu incelemişlerdir. Klasik LPT algoritması ve LS algoritması kullanılmıştır.

Li ve Yang (2009) [40] makalelerinde benzer olmayan paralel makine sıralamasında toplam ağırlıklandırılmış tamamlanma zamanı minimizasyonunu incelemişlerdir. Burada sezgiseller 3 aşamada incelenmiştir; “yaklaştırma algoritması” (approximation algorithms), meta-sezgiseller ve diğer sezgisellerdir. Yaklaştırma algoritması bir polinomial zaman algoritmasıdır. Bir “polinomial zaman yaklaştırma algoritması şeması” (PTAS) da kullanılmıştır. Meta-sezgisellerde ise “tekrarlanan” gelişim olarak çağrılan, temel bir lokal araştırma metodu kullanılmıştır. Çalışmada, Tabu arama, genetik algoritma ve benzetimli tavlama 3 tip önemli meta-sezgisellerden olmuştur. Tabu Arama (TA), Glover tarafından kombinatoriyal problemlerin çözümü için önerilmiş yüksek seviyeli bir sezgisel programlama tekniğidir. Diğer sezgisellerden ise SWPT (En küçük ağırlıklandırılmış proses zamanı ilk) kuralı kullanılmıştır.

Behnamian ve arkadaşları (2010) [41] çalışmalarında kurulum zamanı tabanlı sıralama ile benzer paralel makine sıralama problemini incelemişlerdir. Hibrit bir metasezgisel tamamlanma zamanı minimizasyonu için kullanılmıştır. Çözüm yaklaşımı, Karınca kolonisi yaklaşımı, benzetimli tavlama ve değişken bir komşuluk yaklaşımından oluşmuştur.

Kim ve arkadaşları (2009) [42] ise makalelerinde çoklu benzer paralel makineler üzerinde, deterministik bir sıralama problemini incelemişlerdir. Lineer Programlama (LP) tabanlı bir “liste sıralama” sezgiseli kullanılmıştır.

Kaynak tabanlı proses zamanı ile paralel makine sıralaması, pek çok üretim yönetimi uygulamasında görülen bir yöntemdir. Özellikle, kısıtlar teorisi ve yalın üretimde olduğu gibi. Su ve Lien (2009) [43] de bunun üzerine eğilmişlerdir. Bu tip bir sıralamada kaynakların işlere işlerin de makinelere atanması tanımlanırken, tamamlanma zamanı minimizasyonu önemli olmaktadır. Çalışmada kendi geliştirdikleri bir sezgisel önermişler, daha sonra bu sezgiseli Gupta- Torres yaklaşımlarından olan LISTFIT sezgiseli ile karşılaştırmışlardır.

Ying ve Cheng (2010) [44] ise “tekrarlı bir sezgisel” (iterated greedy heuristic-IG) kullanmışlardır. Set-up zamanı tabanlı sıralama ile dinamik paralel makine sıralaması incelenmiştir. Burada ki sezgiselde bir başlangıç çözümü, ardından bir yıkım fazı ve oluşum fazı bulunmaktadır.

Paula ve arkadaşları (2010) [45] ise çalışmalarında ilişkisiz paralel makineleri incelemişlerdir. Amaç fonksiyonu, işlerin toplam ağırlıklandırılmış gecikmesinin minimizasyonudur. Bir “Lagrangean sezgiseli” gene önerilmiştir.

James ve Lobo (2011) [46] çalışmalarında, tekli ve paralel makineli durumlarda, genel amaçlı bir sezgisel yaklaşımı metasezgiseller ile birleştirerek önermişlerdir. Sıralama tabanlı hazırlık zamanları amaç fonksiyonunu oluşturmuştur.

Dalfarda ve Mohammadi (2012) [47] ise, çok amaçlı esnek akış tipi çizelgeleme probleminde, paralel makinelerin kullanımı ile, bakım maliyeti konusuna yönelmişlerdir. Yeni bir matematiksel modelleme problem için geliştirilmiştir. İki metasezgisel algoritma, bir hibrit genetik algoritma ve bir benzetimli tavlama algoritması probleme modellemenin ardından uygulanmıştır.

Liao ve arkadaşları (2012) [48] makalelerinde, aile-hazırlık zamanları ve toplam ağırlıklandırılmış tamamlanma zamanı minimizasyonu tabanlı, benzer paralel makine problemini incelemişlerdir. Yazarların kendilerinin geliştirdiği bir sezgisel kullanılmıştır daha sonrasında.

Li ve arkadaşları (2013) [49] ise ilişkisiz paralel makinelerde yük proseslerini incelerken, tamamlanma zamanı minimizasyonu üzerinde çalışma yapmışlardır. En uygun en uzun proses zamanı (BFLPT) sezgiseli tabanlı sezgiseller 2 grup problem üzerinde, çözüm için önerilmişlerdir.

Nogueira ve arkadaşları (2014) [50] çalışmalarında, ilişkisiz paralel makine problemini, toplam erken ve geç bitirme cezalarını minimize ederek incelemişlerdir. Bunun için GRASP

(Greedy Randomized Adaptive Search Procedure- GRASP), tekrarlı lokal arama (Iterated Local Search-ILS) ve rota birleřtirme (path relinking) sezgiselleri denenmiřtir.

### **1.1.2 Hibrit Akıř Tipi Hakkındaki Literatür Taraması**

Linn ve Zhang (1999) [51] alıřmalarında, Hibrit Akıř Tipi (HFS) 'nin genel bir izimini yapmıř ve gelecek arařtırmaları iin ynlendirmelerde bulunmuřlardır.

Genoulaz (2000) [52] maksimum gecikmeyi minimize etmek iin “zaman aralıęı” ve “ncelik kısıtları” ile HFS problemini incelemiřtir. Bunun iin, “problem bazlı” sezgiseller nerilmiřtir.

Oguz vd. (2003) [53] ok iřlemcili 2 ařamalı HFS probleminde, tamamlanma zamanını minimize etmek iin “sezgisel algoritmalar” nermiřlerdir. Sezgiseller doęası gereęi yapısaldirler.

Pek ok alıřma (sıralama problemleri zerinde) ya hazırlık zamanlarını grmemezlikten gelmekte ya da iř sıralamasının olduęu her makinenin baęımsız olduęunu varsaymaktadır. Bunun sonucunda, Zandieh vd. (2006) [54] alıřmalarında “sıralama” baęımlı hazırlık zamanlarının olduęu bir HFS problemini (SDST) incelemiřlerdir. Sezgisel olarak “immun (immn) algoritma (IA)” dřnlmřtir.

Vob ve Witt (2007) [55] gerek bir dnya multi mod'lu/multi proje sıralama problemini incelemiřlerdir. İerisinde, 16 adet retim ařaması ieren bir HFS sistemi bulunmaktadır. Ayrıca “sıralama tabanlı, kurulum zamanı esaslı (2 ařamada devreye giren)” yntemi ile daęıtım kuralları nerilmiřtir.

Low vd. (2008) [56] iliřkisiz makineler ile 2 ařamalı HFS problemini incelemiřlerdir. Ama, sistemde tamamlanma zamanını minimize etmektir. Kullanılan sezgiseller, modifiye edilmiř Johnson kuralı ve daęıtım kurallarıdır.

Tseng vd. (2008) [57] ok ařamalı HFS problemini incelemiřlerdir. Fabrikada retim sistemi, iki ařamada ve seri olarak dřnlmřtir. İlk ařama, bir makine ierirken, ikinci ařama iki

benzer paralel makine içermektedir. Sistemde bazı işler her iki aşamada da prosese girebilirken, bazıları ise sadece ikinci aşamada prosese girebilmektedir. Tamamlanma zamanını minimize etmek amaçlı, kaçan iş operasyonları ile ilgili olarak bir sezgisel önerilmiştir.

Choi ve Lee (2009) [58] ise geciken işleri minimize etmek amacıyla 2 aşamalı HFS problemini incelemişlerdir. Her aşamada çoklu benzer paralel makine mevcut olmuştur. Çözüm için, Dal-Sınır Algoritması ve sezgiseller kullanılmıştır.

Cihanlı (2010) [59] çalışmasında iki aşamalı hibrit akış ipi üretim (HFS) ortamında çizelgeleme problemini ele almıştır. Özdeş olmayan paralel tezgahları içeren problemde, işler sadece belirlenen tezgahlarda işlenebilmekte ve sıralamaya bağlı hazırlık sürelerini içermektedir. Problemin amacı yine en büyük tamamlanma zamanının en küçüklenmesi olmuştur. Bunun için bir matematiksel model geliştirilmiştir.

Behnamian vd.(2010) [41] HFS ortamında, sıralama hazırlık zamanları ile işleri sıralamış ve hibrit bir sezgisel kullanmışlardır. Uygulamada ise, “değişken komşuluk yaklaşımı”, ACO (Karınca kolonisi Optimizasyonu) ve benzetimli tavlama yöntemleri uygulanmıştır.

Khalouli vd. (2010) [60] çalışmalarında toplam erken bitirme ve geciktirme cezalarının toplam minimizasyonunu ele alarak bir HFS sıralama probleminde, sezgisel ve metasezgisel (ACO) metotları incelemişlerdir.

Engin vd. (2011) [61] ise makalelerinde “çok işlemcili” sıralama problemi ile HFS sıralaması içeren bir problem düşünmüştür. Tamamlanma zamanı minimizasyonu esastır ve geniş boyutlu problemler çözümünde bazı zorlukları mevcuttur. Çözüm için Genetik Algoritma önerilmiştir.

Ruiz ve Vázquez-Rodríguez (2010) [62] değişik HFS problemlerini incelemişler ve her birisinde değişik kısıtlar, amaç fonksiyonları ve varsayımları ele almışlardır.

Syam ve Al-Harkan (2010) [63] çalışmalarında HFS (benzer paralel makineli) probleminde Cmax'ı minimize etmek için 3 metasezgiseli (Genetik Algoritma, Tavlama Benzetimi ve Tabu Arama) karşılaştırmıştır. Tabu aramanın HFS problemleri için etkili ve verimli olduğu ortaya çıkmıştır.

Amin-Naseri ve Beheshti-Nia (2009) [64] Cmax minimizasyonu esaslı bir HFS (benzer paralel makineli) sisteminde "paralel yığın sıralama" problemini incelemişlerdir. 3 sezgisel algoritma önerilmiştir.

Kahraman vd. (2010) [65] çok aşamalı HFS (benzer paralel makineli) sisteminde, çok işlemcili iş sıralama problemi incelenmiş (HFSMT) ve "paralel greedy algoritması (PGA)" önerilmiştir.

Ribas vd. (2010) [66] HFS sıralaması ile alakalı olarak en son çıkan makaleleri incelemişler ve ortaya bir literatür taraması çalışması çıkartmışlardır.

Wang ve Liu (2013) [67] tahsis edilmiş/bağlanmış makinelerin olduğu iki aşamalı HFS probleminde, Dal-Sınır algoritması tabanlı bir sezgisel önermişlerdir. Ardından Hadda (2013) [68] yayınladığı makale ile, uyguladıkları eliminasyon kurallarından birisinin doğru olmadığını ve diğerinin de bir başka eliminasyon kuralının hükmü altında olduğunu belirterek Wang ve Liu (2013) [67]'ya cevap vermiştir. Yang (2013) [69] çalışmasında, 2 aşamalı HFS probleminde sadece ilk aşamada tahsis edilmiş/bağlanmış makinelerin olduğu hali incelemiştir. Sezgisel olarak, Johnson kuralı ve "doymaz-tipli (greedy-type)" sıralama kuralı denenmiştir.

Wang ve Liu (2013) [70] aynı sene bir başka çalışmalarında, 2 aşamalı beklemez HFS (benzer paralel makineli) probleminde Genetik Algoritma'yı kullanmışlardır. Costa vd. (2013) [71] ise çalışmalarında, suni bir HFS (ilişkisiz paralel makineli) sıralama problemi çözümünde "ikili kodlama" tabanlı (two-phase metaheuristics-MEs) bir meta sezgisel denemişlerdir.

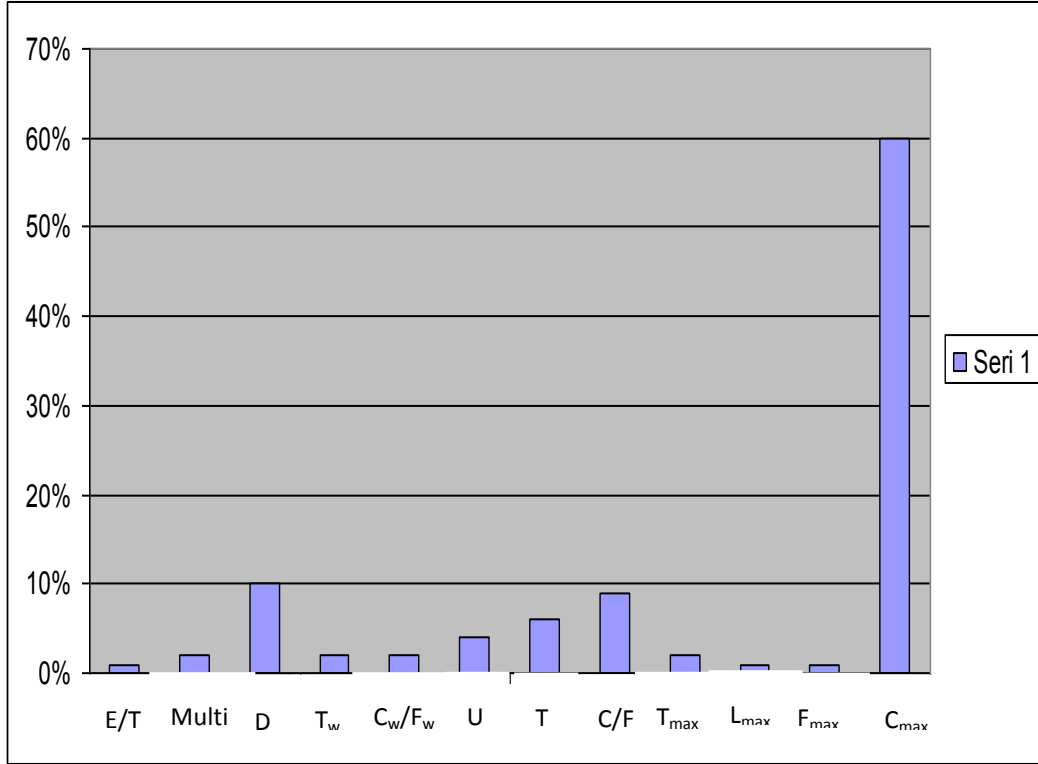
Chung ve Liao (2013) [72] çalışmalarında HFS (benzer paralel makineli) sisteminde tamamlanma zamanı minimizasyonu "immunoglobulin" tabanlı bir yapay immün sistem (IAIS) ile çözmeye çalışmıştır.

Chou (2013) [73] ise, çok işlemcili işlerde, “kokteyl kodlama metodu” ile parçacık sürü optimizasyon yöntemini HFS (benzer paralel makineli) sıralama problemleri için uygulamışlardır. Lahimer vd. (2013) [74] da gene çok işlemcili işler için tamamlanma zamanı minimizasyonu esaslı bir HFS sistemini (her aşamada benzer paralel makineler vardır) incelemişlerdir. Yeni bir “tutarsızlık (discrepancy) arama” metodu denenmiştir.

Marichelvam vd. (2014) [75] çalışmalarında, tamamlanma zamanını minimize etmeyi amaç eden, çok aşamalı bir HFS sıralama probleminde, son zamanlarda geliştirilen “cuckoo araştırması (ICS)” metasezgiselini denemişlerdir. ICS içerisinde başlangıç çözümlerinin optimal veya optimale yakın bulunabilmesi için ise, NEH sezgiseli ile birleştirme yapılmıştır.

Fattahi vd. (2014) [76] ise hazırlık zamanlı ve montaj operasyonlarının olduğu bir HFS probleminde Dal-Sınır Algoritması kullanmışlardır.

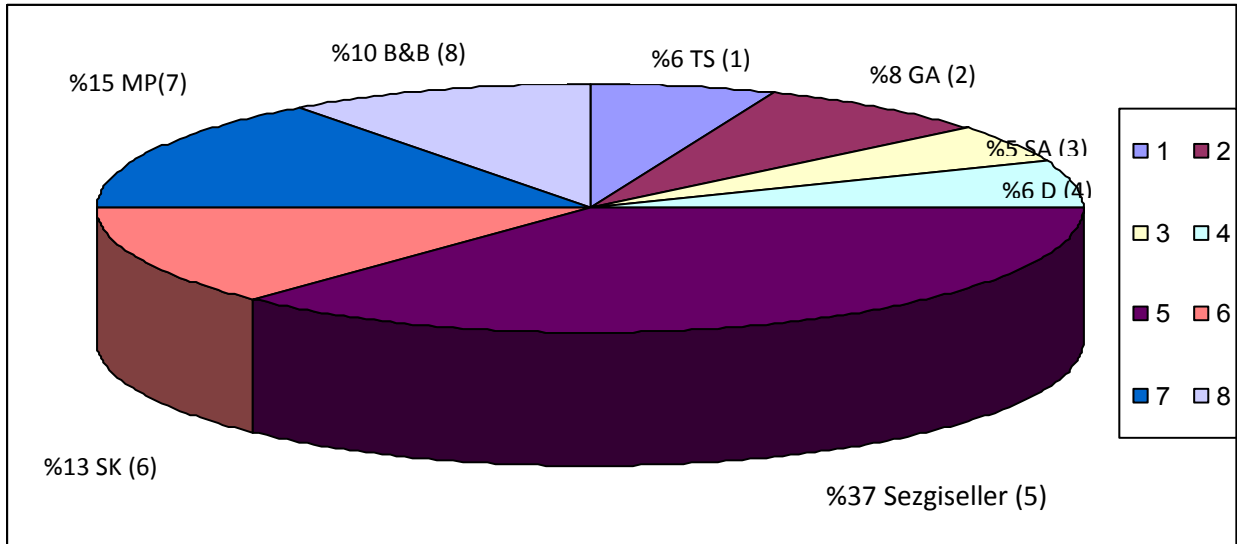
HFS problemi dikkate değer bir şekilde karmaşıktır ve çalışmalara bakıldığında, özellikleri ile ilgili bilgi toplayan ve bunları kesin-olmayan algoritmalar içinde kullananların oldukça az sayıda olduğu görülür. Başarılı meta-sezgiseller, iki esaslı fikirden birini veya diğerini takip etmiştir; aramayı küçültmek için problemin graf gösteriminin özelliklerini kullanmak veya iş indislerinin permütasyon uzayının aramasını sınırlandırmak.



Şekil 1.1 Amaç fonksiyonlarının dağılımı; C/F = Toplam/Ortalama Tamamlanma zamanı veya akış zamanı, T=Toplam/Ortalama Gecikme, E/T= Erken bitirme veya gecikmenin toplam/ortalama miktarı, Multi= Çoklu amaç değerleri

Şekil 1.1'den de görüldüğü üzere, Literatür ağırlıklı olarak C<sub>max</sub> kriteri doğrultusunda ilerlemiştir. Şekil 1.2, 2012 senesine kadar incelenen iş sıralama (HFS ve/veya sadece paralel makineli iş sıralama) makalelerinde kullanılan metodların yüzde oranlarını göstermektedir.





Şekil 1.2 B&B (Dal-Sınır), TS (Tabu arama), GA (Genetik Alg.), SA (Benzetimli Tavlama), MP (Matematiksel Programlama), SK (Sevkiyat/Dağıtım Kuralları), Sezgiseller ve D. (Diğer yöntemler)' in HFS makalelerinde dağılımı

Şekil 1.2' de görüldüğü gibi; matematiksel programlama ve B&B yöntemleri toplamda %25' i bulmaktadır. Sezgiseller ise pastanın % 37'sini oluşturmaktadır. GA, TS, SA dışında ki kısım ise %6' lık bir kısım ile "diğer yöntemler" gelmektedir. Çizelge 1.1 ise, Paralel Makinelerin HFS problemleri içinde araştırıldığı makalelerde, yine 2012 senesine kadar olan, çeşitlerine ve kademe sayılarına göre dağılımını vermektedir. Daha sonra gelen, Çizelge 1.2, 1997-2014 seneleri arasında basılan makaleleri referans alarak, Paralel Makine Çizelgenmesi Hakkındaki Literatür Taramasını verirken, Çizelge 1.3. ise, 1999-2014 tarihleri arasında yayımlanan makaleleri inceleyerek, HFS hakkındaki literatür taramasını vermektedir.

Çizelge 1.1 Paralel Makinelerin HFS problemlerinin araştırıldığı makalelerde, çeşitlerine ve kademe sayılarına göre dağılımı

Aşama Sayıları	Paralel Makine Çeşitleri			Toplam
	Benzer	Uniform	İlişkisiz	
2	26	2	5	32
3	4	1	0	5
m	54	1	7	63
<b>Toplam</b>	<b>84</b>	<b>4</b>	<b>12</b>	<b>100</b>

Çizelge 1.2 Paralel Makine Çizelgelenmesi Hakkındaki Literatür Taraması

No	Yayın Yılı	Yazar	Paralel Makine Yapısı			Başlık	Kullanılan Teknikler	Amaç Fonksiyonu									
			Ident.	Non-ident.	Related			Unrelated	$C_{max}$	$F_{max}$	$S_t$	W	D	T	L	Diğer	
1	1997	Kravchenko ve Werner	+				Parallel machine scheduling problems with a single server", Mathematical Computational Modelling	Sergisel yöntemler.		+							
2	1997	Park ve Kim	+				Search Heuristics for a Parallel Machine Scheduling Problem with Ready Times and Due Dates	Tavlama benzetimi (Simulated Annealing) ve tabu arama (Tabu Search)					+				
3	1997	Torres ve arkadaşları	+				Simulated annealing heuristics for the average flow-time and the number of tardy jobs bi-criteria identical parallel machine problem	Tavlama benzetimi ve komşuluk yaklaşımı (neighborhood search)		Fort	+		+				

Çizelge 1.2 Paralel Makine Çizelgelenmesi Hakkındaki Literatür Taraması (devamı)

4	1999	Chiu, Fang, Lee	+				Sequencing parallel machining operations by genetic Algorithms	Genetik algoritma	+											
5	1999	Min ve Cheng	+				A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines	Genetik algoritma	+											
6	2000	Gupta ve Torres	+				Minimizing makespan subject to minimum total flow-time on identical parallel machines	Deneysel sezgisel	+											
7	2000	Laguna ve arkadaşları	+				Minimizing weighted tardiness of jobs with stochastic interruptions in parallel machines	Yayılım (Scatter) araştırması, sözde evrimsel metotlar	+											
8	2001	Gupta ve Ho	+				Minimizing makespan subject to minimum flowtime on two identical parallel machines	Çoklu değiştirilmiş algoritma (modified multifit algorithm)	+											
9	2001	Bank ve Werner					Heuristic Algorithms for Unrelated Parallel Machine Scheduling with a Common Due Date, Release Dates,	Yapısal algoritmalar ile lokal yaklaşım sezgiselleri.												+

















Çizelge 1.2 Paralel Makine Çizelgelenmesi Hakkındaki Literatür Taraması (devamı)

43	2010	Behnamian ve arkadaşları	+						Hibrit bir metasezgisel. Karınca kolonisi yaklaşımı, benzetimli tavlama ve değişken bir komşuluk yaklaşımı	+									+	
44	2010	Paula ve arkadaşları							A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times		+								Sezgisel yöntemler (Lagrangean sezgiseli).	
45	2011	James ve Lobo	+						Single and parallel machine capacitated lot sizing and scheduling: New iterative MIP-based neighborhood search heuristics									Genel amaçlı, MIP tabanlı bir komşuluk yaklaşımı sezgiseli.	+	
46	2012	Dalfarda ve Mohammadi	+						Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance									Metasezgisel algoritma, Hibrit genetik algoritma, Benzetimli tavlama algoritması.		

Çizelge 1.2 Paralel Makine Çizelgelenmesi Hakkındaki Literatür Taraması (devamı)

47	2012	Liao ve arkadaşları	+					An improved heuristic for parallel machine weighted flowtime scheduling with family set-up times	Probleme özgü oluşturulan sezgisel.	+	+							E
48	2013	Li ve arkadaşları						Scheduling unrelated parallel batch processing machines with non-identical job sizes	En uygun en uzun proses zamanı (BELPT) sezgiseli tabanlı sezgiseller	+								
49	2014	Nogueira ve arkadaşları						Hybrid GRASP Heuristics to Solve an Unrelated Parallel Machine Scheduling Problem with Earliness and Tardiness Penalties	Hibrit sezgiseller (GRASP).									

Çizelge 1.3 Hibrit Akış Tipi Hakkındaki Literatür Taraması

No	Yayın Yılı	Yazar	Paralel Makine Yapısı				Başlık	Kullanılan Teknikler	Amaç Fonksiyonu									
			Ident.	Non-ident.	Related	Unrelated			$C_{max}$	$F_{max}$	$S_t$	W	D	T	L	Diğer		
1	1999	Linn ve Zhang					HYBRID FLOW SHOP SCHEDULING: A SURVEY	Literatür Taraması.										
2	2000	Genoulaz	+				Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness	Problem bazlı sezgiseller								+		
3	2003	Oğuz ve arkadaşları	+				Heuristic algorithms for multiprocessor task scheduling in a two-stage hybrid flow-shop	Sezgisel algoritmalar										+
4	2006	Zandieh ve arkadaşları	+				An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times	immün algoritmalar (IA)										

Çizelge 1.3 Hibrit Akış Tipi Hakkındaki Literatür Taraması (devam)

5	2007	Vob ve Witt	+					Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application	Dağıtım kuralları	+					
6	2008	Low ve arkadaşları			+			A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines.	Modifiye edilmiş Johnson kuralı, dağıtım kuralları	+					
7	2008	Tseng ve arkadaşları	+					A note on two-stage hybrid flowshop scheduling with missing operations	Yapısal sezgiseller	+					
8	2009	Choi ve Lee	+					Scheduling algorithms to minimize the number of tardy jobs in two-stage hybrid flow shops	Dal-Sınır Algoritması, Sezgiseller.					+	
9	2009	Ruiz ve Vázquez-Rodríguez						The hybrid flow shop scheduling problem	Literatür taraması						
10	2009	Amin-Naseri ve Beheshti-	+					Hybrid flow shop scheduling with	Sezgisel yöntemler.	+					



Çizelge 1.3 Hibrit Akış Tipi Hakkındaki Literatür Taraması (devam)

16	2011	Engin, Ceran ve Yilmaz	+											Genetik algoritma	+				
17	2013	Wang ve Liu	+											Dal-Sınır algoritması	+				
18	2013	Hadda												Sezgisel metodlar.					
19	2013	Yang	+											Johnson kuralı ve doymaz-tipli (greedy-type)	+				
20	2013	Wang ve Liu	+ 2 aşama											Genetik algoritma.					
21	2013	Costa ve arkadaşları	+											ikili kodlama tabanlı (two-phase metaheuristics-MEs)					



Çizelge 1.3 Hibrit Akış Tipi Hakkındaki Literatür Taraması (devam)

22	2013	Chung ve Liao	+	An immunoglobulin-based artificial immune system for solving the hybrid flow shop problem	İmmunoglobulin tabanlı bir yapay immün sistem (IAIS) algoritması								
23	2013	Chou	+	Particle swarm optimization with cocktail decoding method for hybrid flow shop scheduling problems with multiprocessor tasks	Kokteyl kodlama metodu, Parçacık sürü optimizasyonu								
24	2013	Lahimer ve arkadaşları	+	Improved bounds for hybrid flow shop scheduling with multiprocessor tasks	Tutarlılık (discrepancy) arama metodu	+							
25	2014	Marichelvam ve arkadaşları	+	Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan	Cuckoo araştırması (ICS) metasegiseli.	+							
26	2014	Fattahi ve arkadaşları	+	A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations	Dal-Sınır Algoritması							+	

## 1.2 Tezin Amacı

Bu çalışma içerisinde, öncelikle “iş sıralama” problemleri içerisinde önemli ve görece olarak daha az ilgilenilen bir konu olan “Hibrit Akış Tipi (HFS)” sistemler incelenmiş ve ardından bu konu üzerinde bir model düşünülmüştür. HFS yapısı içerisinde her aşamada ise, literatürde bu tip sistemler içinde oldukça az yer bulan “ilişkisiz paralel makineler” kullanılmıştır.

Hibrit Akış Tipi, genel olarak, verilen bir amaç fonksiyonunu en iyileyen seri m aşamalı durumda prosese giren bir dizi n işin sıralanması durumudur. HFS sıralama problemleri hem akış tipi hem de paralel makine sıralama problemlerini içeren, özel bir yapıyı arz eder.

Yapılan modelleme prosesinde görülmüştür ki, örnek boyutu büyüdükçe, model yavaşlamaktadır. Sıralama problemleri zaten doğası gereği oldukça zor yapıda olan ve çözümü de komplike bir yapı arz eden problemlerdir. Bu nedenle, modellemenin ardından modelin çözümüne yardımcı olması açısından, “sezgisel” ve hatta “metasezgisel” teknikler denenmiştir. Sezgisel tekniklerin en çok kullanılanlarından birisi olan “genetik algoritma” yöntemi uygulanmıştır. Genetik algoritma yöntemi içerisinde, en başta ise, yöntem her anlamda yardımcı olmak adına “baskınlık özellikleri (Dominance Properties-DP)” isimli ve sıklıkla az denenilen bir yöntem eklenmiştir.

Verilen bir sıralama işlemi için “baskınlık özellikleri” kullanılarak, optimale yakın bir çözüm elde edilebilmektedir. Daha önce de söylendiği gibi, büyük boyutlu problemlerin çözümü zor olmaktadır. Bunun için, çözüm kalitesini arttırmaya yardımcı olarak, baskınlık özelliklerinin genetik algoritma ile birleştirildiği, yeni bir meta sezgisel ortaya konulabileceği bu çalışma içerisinde verilen en temel noktalardan birisidir. Genetik Algoritmalar, en iyinin korunumu ve doğal seçim ilkesinin benzetim yoluyla bilgisayarlara uygulanması ile elde edilen bir arama yöntemidir. Sonuçta, görülmüştür ki Üstünlük Özellikleri ile birlikte kullanılan Genetik Algoritma yöntemi Hibrit Akış Tipi bir sistemde oldukça iyi ve başarılı sonuçlar vermektedir.

### **1.3 Orijinal Katkı**

Bu tez çalışması içerisindeki orijinal katkı, HFS yapısı içerisinde birden çok aşama üzerinde sıralama işlemlerinin düşünülmesi ve giderek zorlaşan sıralama problemi için “Üstünlük Özellikleri” kavramının Genetik Algoritma Metasezgisel yöntemi’ne uygulanarak, bu tip bir yöntemin denenmesi olmuştur.

Çalışmada, öncelikle “iş sıralama” problemleri içerisinde önemli olan ancak diğer sistemlere göre, daha az ilgilenilen bir konu olan “Hibrit Akış Tipi (HFS)” sistemler incelenmiş ve ardından bu konu üzerinde bir model düşünülmüştür.

Verilen bir sıralama prosesi için “baskınlık/üstünlük özellikleri” kullanılarak, optimal sonuca yakın olan bir çözüm elde edilebilmektedir. Bu nedenle, çözüm kalitesini arttırmaya yardımcı olmak adına, üstünlük özelliklerinin genetik algoritma metasezgiseli ile birleştirildiği yeni bir yöntem bu tez çalışmasındaki en temel noktayı oluşturmaktadır.

## BÖLÜM 2

---

### İŞ SIRALAMA ve ÇİZELGELEME

Hayatın büyük bir bölümü sıralama problemlerinden oluşmaktadır. Bu durum bilinçli ve/veya bilinçsizce olabilmektedir. Medeni olma, Adalet duygusu ve daha birçok faktör bu bağlamda ön plana çıkmaktadır. Sıralama problemlerinde temel felsefeler zamanlama ve maliyettir. Çünkü sıralamada zaman maliyetle beraber hareket etmektedir. Minimum zamanda işin yapılması önemlidir. İşlere de bu şekilde yaklaşılmaktadır. Yani maksimum kar minimum maliyettir.

Sıralama işlemi üretim, imalat, yönetim, bilgisayar bilimi ve bunun gibi pek çok alanda kullanılan önemli bir procestir. Genel olarak ihtiyaç duyulan kısıtların altında, kaynakların yüksek zamanlı işlere atanmasıyla ilgilenir. Uygun olan sıralama şekli, bütün zamanı küçültürken, ürün kalitesini arttırır. Kaynaklar ve işler pek çok formlarda olabilmektedir. Sıralama amacı, sonuncu işin tamamlanma zamanı minimizasyonu olabileceği gibi, her işin teslim tarihinden önce tamamlanması da olabilmektedir [77].

Çizelgeleme, belirli bir takım görevleri yerine getirmek için kaynakların zaman içerisinde tahsisi olarak tanımlanabilmektedir. Bu tanım içerisinde görevler, içinde bulunulan ortama bağlı olarak, değişik şekillerde adlandırılabilirler. Sıklıkla, belirli ürünleri ya da hizmetleri meydana getirmek için gerekli olan faaliyetler olmaktadır.

Çizelgeleme içerisinde genel olarak, üç unsur söz konusu olmaktadır. Bunlar; üretim, kaynak ve zamandır. Bu unsurları göz önüne alarak çizelgeleme şöyle tanımlanabilir: Çizelgeleme, belli bir takım işleri yapmak adına, hangi kaynakların, ne zaman ve nasıl kullanılacaklarının tespit edilmesi işlemidir. Etkin bir çizelgeleme işlemi sayesinde belirli faaliyetlerin daha az kaynak kullanımı ve/veya daha kısa zamanda yapılabilme olanağı ortaya çıkabilmektedir.

Çizelgeleme, birçok üretim ve servis endüstrisinde kullanılan temel bir karar verme prensibidir. Verilen zaman süresi (periyodu) içinde kaynakları işlere paylaştırır ve bir ya da daha fazla amacı optimize etmeyi amaçlar [78]. Üretim çizelgeleme, imalat işletmelerinin üretim planlamasının önemli bir bölümünü oluşturur ve prosesler açısından önemli bir karar verme problemidir [79].

Bir üretim sisteminde, tamamlanma süresinin minimuma indirilmesi üretim kontrolü açısından arzulanan bir durum iken, maksimum gecikmenin minimize edilmesi ise müşteriler tarafından talep edilir [80]. Üretim birimi ve müşteri tarafından gelen bu iki talebin de karşılanması üretimi yapan işletmenin sorumluluğudur.

Wight (1984) [81]' e göre; Üretim çizelgelemenin iki temel problemi “öncelikler” ve “kapasite” dir. Diğer bir ifade ile hangi işi önce yapmalıyım ve söz konusu iş hangi kaynak tarafından yapılmalı, hangi operasyon ne zaman başlamalı ve ne zaman tamamlanmalı sorularına cevap aranmaktadır.

Üretim Çizelgeleme hikâyesi, Henry Gantt tarafından geliştirilen ilk çizelgelenmelerden, gelişmiş/karmaşık algoritmalar içeren gelişmiş çizelgelere kadar uzanır. Henry L. Gantt, üretim kontrolü için şemalar geliştirmiştir. Cox vd.'ye göre (1992) [82] Gantt şeması; planlanan performans ile gerçekleşen performans arasındaki ilişkiyi grafik olarak göstermek üzere dizayn edilen ilk ve en çok bilinen kontrol şemasıdır [79].

## 2.1 İş Sıralama Kabulleri

Sıralama problemleri, birçok değişken ihtiva eden çözümü zor problemlerdir. Sıralama problemlerinin yapısı gereği pek çok kabuller yapılmaktadır.

Bu kabuller şunlardır [83]:

1. Her iş bir varlıktır; her iş belirli operasyonlardan oluşmuştur. Aynı işin iki işlemi aynı anda işlenemez. Böylece bitmiş bir ürünün, montaj önceliğine göre imal edilmiş parçalarının ortaya çıkardığı problemleri ihmal ediyoruz.

2. Kesinti yoktur: Bir operasyon o makinede başlamadan önce, daha önce başlamış olan operasyon tamamlanmalıdır.

3. Her makinede bir adet olmak üzere her bir işin m sayıda belirli operasyonları vardır. bir işin aynı makinede iki işlem gerektireceği olasılığına izin verilmemektedir. Aynı şekilde her bir işin tüm makinelerde işlenmesine ısrar edilmektedir. Bir iş, bir veya daha fazla makineyi atlayamaz.

4. Erteleme yoktur. Her iş tamamlanıncaya kadar işlenmelidir.

5. İşlem zamanları sıralamadan bağımsızdır. Burada özellikle iki şey kabul edilir. Birincisi her işi tesis etme zamanı sıralamadan bağımsızdır. Yani bir işin yapılması için makinenin ayarlanması son işten bağımsızdır. İkinci olarak işlerin makineler arasında taşınması için geçen zaman göz ardı edilebilir.

6. İşlem sırasında ara stoklara müsaade edilir; yani işler bir sonraki makinenin boşalmasını bekleyebilir. Bu yersiz bir varsayım değildir. Bazı problemlerde işlerin yapılması operasyondan operasyona devamlı olmalıdır.

7. Her makineden sadece bir tane vardır. Bir işin yapılmasında seçilebilecek makineler olmasına izin verilmemektedir. Bu varsayım diğerlerinin arasında belli makinelerin yaratacağı darboğazlardan kurtulmak için bu makinelerin çoğaltılmasını bertaraf eder.

8. Makineler işsiz (boş) olabilir.

9. Hiçbir makine aynı anda birden fazla iş yapamaz.

10. Makineler hiçbir zaman bozulmamaktadır ve sıralama periyodu boyunca hazır bulunmaktadır.

11. Teknolojik kısıtlar önceden bilinmekte ve sabittirler.

12. Hiçbir rastgelelik yoktur. Özellikle; a) işlerin sayısı biliniyor ve sabit, b) makinelerin sayısı biliniyor ve sabit, c) işlem zamanları biliniyor ve sabit, d) hazırlık zamanları biliniyor ve sabit, e) bir problemi belirtmek için gerekli bütün nicel değerler biliniyor ve sabit.

Bazen belirli örneklerde bu kabullerin bir veya ikisi göz ardı edilebilmektedir.

İş sıralamada erteleme yoktur. Sıralama optimize edildikten sonra erteleme yapılmamaktadır. Zaten stoklara müsaade edilmezse geciken işin gecikmesi bütün sıraya etki etmektedir. Diğer taraftan makineler işi, işler de makineyi bekleyebilmektedir. Arzu edilen ise, makina ve iş hazır ise hemen işin başlayabilmesidir. Öyle bir planlama yapılmalıdır ki, minimum ara stoklara izin verilebilmelidir.

Sıralamanın amaçları firmadan firmaya ve günden güne değişecektir. Belki de amaç bütün bölümlerde pahalı işgücü ve makinelerin nadiren boş kalacağı eşit bir aktivite seviyesi sağlamaktır. Amaç mümkün olduğu kadar kısa zamanda işleri bitirmek veya işi belirli kontrat tarihinde başarmış olmaktır.

Pek çok kaynak üzerinde pek çok işin sıralanması genel haliyle NP-tam (NP- complete) problemi olarak bilinmektedir. Sıralama problemlerinin yüksek karmaşıklığı nedeniyle pek çok sezgisel yaklaşım farklı sıralama problemi çözümünde geliştirilmiştir [84].

Genel bir sıralama modeli bir dizi kaynak (R) üzerinde sıralanan bir grup iş (T) sistemini içermektedir. Bu noktada bir dizi maliyet fonksiyonu performansı ve bir dizi kısıtlar önem kazanmaktadır. Kaynaklar pek çok durumda makineler veya işlemciler olacaktır [85] :

$$R = \{R_1, R_2, \dots, R_m\} \quad (2.1)$$

Kaynaklar benzer veya farklı işlem hızlarına sahip olabilirler. Pek çok durumda, ek kaynaklar da devreye girebilmektedir. Sistem işleri şöyle tanımlanabilir:

$T = \{T_1, T_2, \dots, T_n\}$  ifadesi bir küme işi tanımlamaktadır. Burada bir iş, bir program aktivitesi veya bir programın yürütülebilir bir kısmını temsil etmektedir.

Sıralama genelde iki sınıfa ayrılmaktadır: öne almalı ve öne almasız sıralama. Öne almalı sıralama aktivitelerin, bazı diğer önemli aktivitelerin daha hızlı işlem görmesi için kesikli olmasına izin vermektedir. Diğer taraftan, öne almasız sıralama sistemleri ise, aktivitelerin hiçbir zaman kesikli olmasına izin vermemektedir [77].

## 2.2 Kısıtlar

Bu noktada iş sıralamada iki önemli kısıt devreye girmektedir. Bunlar şöyle sıralanmaktadır:

a) Öncelik Kısıtı: Bir işin yapılabilmesi için o işten önceki işlerin mutlaka yapılması gerekliliğidir.

b) Teknolojik Kısıt: O işin teknolojik nedenlerden dolayı yapılabilmesi özelliğidir.

İş görenler makinelerin onlara sağladıkları teknolojik koşullar ile ancak o işi yapabilmektedir.

## 2.3 Çalışma Ölçütleri

Literatürde geçen en önemli "iş sıralama ve çizelgeleme" ölçütleri şunlardır [83]:

$d_i$  : Teslim zamanı. Bir işin başlangıcından bitip müşteriye teslim edilmesine kadar geçen zaman.

$a_i$  : Müsaade edilen zaman.



$W_{ik}$  :  $J_i$  işinin,  $k$ ' ıncı işlemden önce geçen bekleme zamanıdır.  $k$ ' ıncı iş, işlem sırasına göre gelen iştir.

$W_i$  :  $J_i$ ' nin toplam bekleme zamanı.  $W_i = \sum_{k=1}^m W_{ik}$

$C_i$  : Tamamlanma zamanı.  $C_i = r_i + \sum_{k=1}^m (W_{ik} + P_{ij(k)})$

$F_i$  : Akış zamanı.  $F_i = C_i - r_i$

$r_i$  : Hazırlık zamanı.

$P_{ji}$  :  $j$  makinesinde  $i$  işinin işlem zamanı.

$L_i$  :  $C_i$ ' nin gecikme zamanı. Basitçe söz verilen tarih ile bitirme (tamamlanma) zamanı arasındaki fark.  $L_i = C_i - d_i$  dir.

$T_i$  :  $J_i$  nin gecikmesi;  $j$  makinesindeki  $i$  işinin gecikmesi  $T_i$ ' dir.  $T_i = \max\{L_i, 0\}$

$E_i$  :  $J_i$ ' nin erken bitirilmesi  $E_i = \max\{-L_i, 0\}$  dir.

$\bar{X}$  : İşlerin ortalaması .  $\bar{X} = 1/n \sum_{i=1}^n X_i$  .

$N_w(t)$  :  $t$  anında işlenmeye hazır olmayan veya makineler arasında bekleyen işlerin sayısı.

$N_c(t)$  :  $t$  anında tamamlanmış olan işlerin sayısı.

$N_u(t)$  :  $t$  anında tamamlanacak olan işlerin sayısı. Bitmek üzere olan işlerin sayısı.

$N_p(t)$  :  $t$  anında işlenmekte olan işlerin sayısı.

$N_w(t) + N_p(t) + N_c(t) = n$        $n$  : Yapılacak işlerin sayısı.

$N_w(t) + N_p(t) = N_u(t)$

$N_u(0) = n$  ve  $N_u(C_{\max}) = 0$

$P_m$  : Paralel benzer makine.

$Q_m$  : Farklı hızlardaki paralel makine.

$R_m$  : Birbirine paralel ilişkisiz makineler.

$F_m$  : Flow shop hali.

$FF_s$  : Esnek flow shop hali.

$O_m$  : Open shop hali.

$J_m$  : Job shop hali.

$prmp$  : Alıkoyma hali.

$prec$  : Öncelik zorlama hali.

$brkdown$  : Bozulma hali.

$M_j$  : Seçilebilir makine kısıtlamaları.

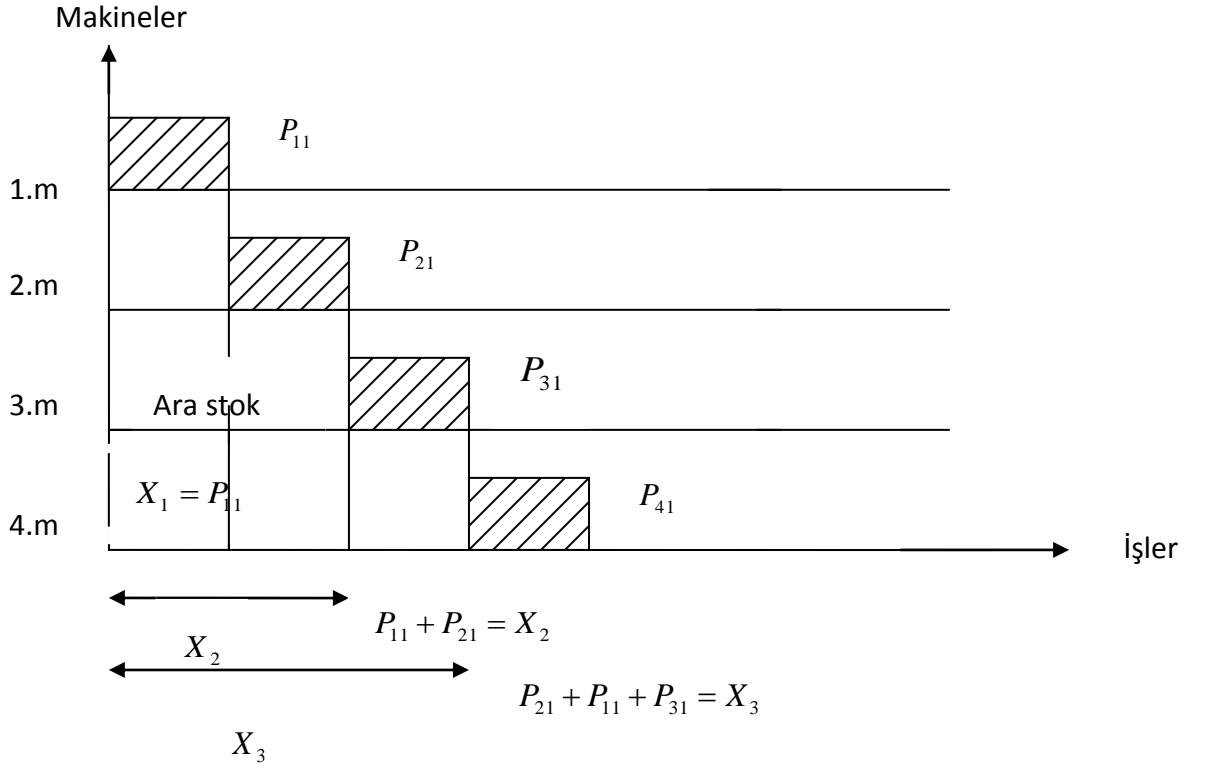
$prmu$  : Yer deęiřtirme (permütasyon).

$block$  : Önleme

Çizelge 2.1'de, Çizelgeleme'de kullanılan en temel notasyonlar görölmektedir.

Çizelge 2.1 Çizelgelemedeki Temel Notasyonlar [77]

<b>Veriler</b>	$n$ $p_i$ $d_i$ $s_i$ $r_i$ $F_i$	<p><b>İş sayısı</b></p> <p><b>i.işin işlem zamanı</b></p> <p><b>i.işin teslim tarihi</b></p> <p><b>i işinin istenilen başlama zamanı</b></p> <p><b>i işinin işlem için hazır olma zamanı</b></p> <p><b>i işinin akış zamanı</b></p>
<b>Kısıtlar</b>	$pmtn$ $nmit$	<p><b>İşler kesintiye uğrayabilir ve sonra kaldığı yerden tekrar başlar.</b></p> <p><b>Makine boş tutulmasına izin veriliyor</b></p>
<b>Değişkenler</b>	$C_i$ $E_i$ $L_i$ $T_i$ $U_i$ $n_T$	<p><b>i.işin tamamlanma zamanı</b></p> <p><b>i.işin erken bitmesi</b>      <math>E_i = \max(0, d_i - C_i)</math></p> <p><b>i.işin gecikmesi</b>      <math>L_i = C_i - d_i</math></p> <p><b>i.işin geç bitmesi</b>      <math>T_i = \max(0, C_i - d_i)</math></p> <p><b>i işinin gecikme durumu</b>      <math>U_i = \{1 \ C_i &gt; d_i, 0 \ dd</math></p> <p><b>toplam geciken iş sayısı</b>      <math>n_T = \sum_{i=1}^n U_i</math></p>
<b>Ölçütler</b>	$f_{\max}$ $C_{\max}$ $L_{\max}$ $L_{\min}$ $T_{\max}$ $E_{\max}$ $\bar{F}$ $(w\bar{F})$ $\bar{T}$ $(w\bar{T})$ $\bar{E}$ $(w\bar{E})$ $\bar{U}$ $(w\bar{U})$	<p><b>Maliyet fonksiyonu</b></p> <p><b>Maksimum tamamlanma Zamanı</b>      <math>C_{\max} = \max_{i=1, \dots, n}(C_i)</math></p> <p><b>Maksimum gecikme</b>      <math>L_{\max} = \max_{i=1, \dots, n}(L_i)</math></p> <p><b>Minimum gecikme</b>      <math>L_{\min} = \min_{i=1, \dots, n}(L_i)</math></p> <p><b>Maksimum geç bitirme</b>      <math>T_{\max} = \max_{i=1, \dots, n}(T_i)</math></p> <p><b>Maksimum erken bitirme</b>      <math>E_{\max} = \max_{i=1, \dots, n}(E_i)</math></p> <p><b>Ortalama akış zamanı</b></p> <p><b>(Ort ağırlıklı akış zamanı)</b>      <math>\bar{F} = \sum_{i=1}^n F_i/n, (w\bar{F} = \sum_{i=1}^n wF_i/n)</math></p> <p><b>Ortalama geç bitirme</b></p> <p><b>(Ortalama ağırlıklı geç bitirme)</b>      <math>\bar{T} = \sum_{i=1}^n T_i/n, (w\bar{T} = \sum_{i=1}^n wT_i/n)</math></p> <p><b>Ortalama erken bitirme</b></p> <p><b>(Ortalama ağırlıklı erken bitirme)</b>      <math>\bar{E} = \sum_{i=1}^n E_i/n, (w\bar{E} = \sum_{i=1}^n wE_i/n)</math></p> <p><b>Ortalama geciken iş</b></p> <p><b>(Ortalama ağırlıklı geciken iş)</b>      <math>\bar{U} = \sum_{i=1}^n U_i/n, (w\bar{U} = \sum_{i=1}^n wU_i/n)</math></p>



Şekil 2.1 Gantt diyagramı.

1.Hal:  $r_i = 0$  ise  $C_i = F_i$  olur.  $C_i = \sum_{j=1}^n (P_{ji} + W_{ji})$  olmaktadır.

2. Hal (İdeal Hal):  $\sum_{j=1}^n W_{ji} = 0$ ,  $r_i = 0$ ,  $C_i = \sum_{j=1}^n P_{ji} = F_i$  dir.

3. Hal:  $r \neq 0$ ,  $\sum_{j=1}^n W_{ji} = 0$ ,  $C_i = r_i + F_i$ ,  $C_i = r_i + \sum_{j=1}^n p_{ji}$

Üretim hatları, sıralama problemlerinin temel uygulama alanlarıdır. Üretim hatlarının özel hali, montaj hatlarıdır. Montaj hattında beklemler olmamaktadır. Beklemeler sıfır kabul edilir. Bu durum en ideal haldir. Montaj hattında taşımalar da yoktur. İşlem sürelerine dahildir. Bu durum 2. HAL 'i tanımlamaktadır.

#### 2.4 Job-Shop Tipi Sıralama Problemlerinin Parametreleri

Job-shop tipi sıralama problemlerinin parametreleri literatürde genel olarak şöyle sıralanmaktadır:

$n$  : işlerin sayısı

$m$  : Makinelerin sayısı.

$A$  : Makine atölyesinde uygulanacak modeli veya akış modelini tanımlar.  $A$  şunlar olabilir:

$F$  : Flow-shop durumu içindir. Yani bütün işler için makine sırası aynıdır.

$P$  : Flow-shop durumu permütasyonu içindir.

$G$  : Teknolojik sınırlamalar halinde hiçbir kısıtlamanın olmadığı genel job-shop durumu.

$B$  : Programın değerlendirilmesi ile performans ölçütünü tanımlar.

## 2.5 Çizelgeleme Problemleri

Genel olarak çizelgeleme problemlerini üç sınıfta toplamak mümkündür:

- Atölye çizelgeleme problem
- Montaj hattı dengeleme problemi
- Seri akışlı çizelgeleme problemi

İşlerin hesaplanan öncelik değerlerine göre sıraya konulması çalışmalarına “iş sıralama” denir [86]. Böylece bir tezgâh boşaldığı zaman tezgâha yüklenecek iş, önceden yapılan iş sıralamasına göre seçilir. Klasik sıralama problemleri için faktörler şöyle sıralanabilir:

### 2.5.1 İşlerin Geliş Şekli

“İşlerin geliş şekli” 2 şık altında incelenebilmektedir [83]:

**a) Statik problemler:** Bu tip problemlerde, belirli dönem için iş listesi bilinmektedir ve bu liste genellikle değişmez. İşler, boş olan atölyeye hemen işlenmek üzere düzenli olarak gelirler. Bu tip problemlerde genel amaç, işlerin tezgâhlardan geçiş sırasını bulmaktır.

**b) Dinamik Problemler:** Bu tip problemlerde ise iş listesi sürekli ve rastgele değişmekte, işler düzensiz aralıklarla atölyeye gelmekte olup; herhangi bir zamanda gelebilecek için özellikleri, sıralamanın sürekli değişmesini gerektirebilir. Bu tip problemlerde genel amaç, tezgah boş bekleme süresini veya bekleyen iş sayısını enazlamak, istenen teslim tarihlerini gerçekleştirmek şeklinde olabilmektedir.

## 2.5.2 İşlerin Görüleceği Tezgâh Sayısı

Atölyede yer alan tezgâh sayısına göre, tek ve çok tezgâhlı olarak ikiye ayrılır [86]:

1. Paralel tezgâhlar → işler aynı işlemi yapan tezgâhlara dağıtılır
2. Seri tezgâhlar → İşler farklı işlem gören ilgili tezgâhlarda yapılır.

*Performans Ölçütü:* Atölye performansını değerlendirmek için kullanılan bir ölçüttür. Çizelgeleme problemlerinde bu ölçüt önemli rol oynamaktadır. Etkinliğin en iyilenmesinde seçilebilecek kimi ölçütler ise şunlardır:

- Bir siparişin tamamlanma zamanının (iş tamamlanma zamanlarının en büyüğünün veya en son sırada yapılan işin bitiş zamanının) enküçüklenmesi
- Ortalama tamamlanma zamanının enküçüklenmesi
- Ortalama ağırlıklı tamamlanma zamanının enküçüklenmesi
- İş sırasının ortaya çıkardığı ortalama bekleme süresinin enküçüklenmesi
- Ortalama gecikme süresinin enküçüklenmesi
- Ortalama ağırlıklı gecikme süresinin enküçüklenmesi
- En büyük gecikme süresinin enküçüklenmesi
- Kuyrukta bekleyen ortalama iş sayısının yansıttığı üretim içi stok düzeyi (geciken iş sayısı, yarı işlenmiş ürün miktarı) değerinin enküçüklenmesi
- Ağırlıklı geciken iş sayısının enküçüklenmesi
- Tezgâh ve işgücü kullanım oranlarının enbüyüklenmesi

## 2.5.3 Yerleşim Tipleri

### 2.5.3.1 Akış tipi (Flow shop) Çizelgeleme

Bütün işlerin rotası aynıdır. Tüm işler aynı tezgâh sırasını izlemektedir. Sürekli üretim söz konusudur. Bu tip yerleşimde amaç, iş sıralarını belirlemektir.

Akış tipi çizelgeleme, birbirinden farklı,  $m$  makine ve  $n$  işin bulunduğu; her bir işin  $m$  operasyondan oluştuğu, her bir operasyonun farklı makinelerde yapıldığı ve bütün işlerin operasyonlarının aynı sıra ile yapıldığı problemlere denir. Akış tipi çizelgeleme probleminde amaç fonksiyonu, toplam akış süresinin ağırlığını ve en son işin tamamlanma zamanını (makespan) minimize etmektir [87].

Günümüzde birçok endüstri alanında akış tipi üretim yaygın şekilde kullanılmaktadır. Bu nedenle, akış tipi çizelgeleme problemi, üzerinde dikkatle durulan bir problem olmuştur.

Akış tipi çizelgeleme problemleri birleşik eniyileme problemi özelliğindedir ve NP-zor problem sınıfındadır. Mesela, “permütasyon akış tipi” çizelgeleme problemi, tüm makinelerde bir işin işlem sırasının aynı olduğu,  $m$  makine ( $j=1,2,\dots,m$ ) üzerinde belli işlem sürelerine sahip  $n$  işin ( $i=1,2,\dots,n$ ) çizelgelenmesinden oluşmaktadır.

Genelde akış tipi çizelgeleme problemleri aşağıdaki temel varsayımlara sahiptir [88]:

1.  $n$  adet çok işlemlili iş kümesi, işlenmek üzere sıfır anında hazırdır.
2. İşlerin hazırlık süreleri sıraya bağlı değildir ve işlem sürelerine dahildir.
3. İşlem süreleri önceden biliniyordur.
4.  $m$  farklı makine sürekli çalışabilir.
5. İşler kesilemez.

Eğer  $j$  makinesindeki  $i$  işinin işlem süresi  $t(i,j)$  ve permütasyon iş kümesi  $\{\pi_1,\pi_2,\dots,\pi_n\}$  ise, tamamlanma zamanı  $C(\pi_i, j)$ :

$$C(\pi_i, 1) = t(\pi_i, 1) \quad (2.2)$$

$$C(\pi_i, 1) = C(\pi_{i-1}, 1) + t(\pi_i, 1) \quad (2.3)$$

$$C(\pi_1, j) = C(\pi_1, j-1) + t(\pi_1, j) \quad (2.4)$$

$$C(\pi_i, j) = \max \{ C(\pi_{i-1}, 1), C(\pi_1, j-1) \} + C(\pi_i, j) \quad (2.5)$$

$$\text{Toplam tamamlanma zamanı: } C_{\max} = C(\pi_n, m) \quad (2.6)$$

Akış tipi çizelgeleme problemleri ile ilgili ilk çalışma, Johnson tarafından 1954 senesinde yapıldı. Johnson,  $n$ -işli ve 2-makineli akış tipi çizelgeleme problemleri üzerinde toplam tamamlanma zamanını minimize etmek için bir eniyileme algoritması tanımladı. Diğer taraftan, göz önüne alınan problem için daha sonra kesin çözümler oluşturan algoritmalar (dal-sınır algoritmaları vb. gibi) önerildi.

Unutulmamalıdır ki, çok sayıda iş ve makine içeren akış tipi çizelgeleme problemleri, birleşik eniyileme problemleri özelliğindedir ve “NP-zor tipi problemler” sınıfındadır. Bu nedenle bu tip problemler için, yaklaşık çözümler tercih edilmektedir [88].

Literatürde akış tipi problemleri için pek çok sayıda sezgisel yaklaşım bulunmaktadır. Son yıllarda birleşik eniyileme problemlerine çözüm sağlamak üzere tavlama benzetimi (Simulated Annealing-SA), tabu arama (Tabu Search-TS), genetik algoritmalar gibi metasezgisel yaklaşımların kullanımı, gösterdikleri performans açısından tercih edilmiş olmuştur.

#### **2.5.3.1.2 Hibrit Akış Tipi**

Hibrit Akış Tipi (Hybrid Flow Shops-HFS), m aşamalı seri bir sistemde, bir grup işin olduğu, genel bir üretim sistemidir. Bir grup değişkenin olduğu ve hepsinin de sıklıkla, şu karakteristiklerde olduğu durum şöyledir [62]:

\*Proses aşamaları sayısı en azından 2 dir.

\*Her aşama  $k$ 'dır ve  $M(k) \geq 1$  olarak, paralel makinelere sahiptir.

\*Bütün işler, takip eden aynı üretim aşamasında prosese girer: aşama 1, aşama 2,..., aşama m, gibi. Bir iş, aşamalardan herhangi birini (prosesse girdiği aşamalardan en azından birinde) atlayamaz.

#### **2.5.3.2 Sipariş (Atölye) Tipi (Job Shop) Çizelgeleme**

Tüm işler, farklı tezgâh sıralarını izlemektedir. Her işin kendine göre bir rotası vardır. Bu tip yerleşimde amaç, işlerin makineye atanmasıdır.

### **2.6 Çizelgelemenin Başarılı Olabilmesi İçin Dikkat Edilmesi Gerekenler**

Çizelgelemenin başarılı olabilmesi için dikkat edilmesi gereken noktalar şunlardır:

Kapasite: Kapasitenin ne olduğu bilinmelidir. Uygulanması mümkün olmayan ya da oldukça pahalıya mal olabilecek çizelgelerin herhangi bir faydası yoktur. Ayrıca kapasite sabit bir kavram değildir ve üretilen ürün çeşitlerinin miktarına göre değişir. (elektrik motoru ile jeneratör üretimi kapasitesi eşit değildir.)



Yeterlilik: bir makine veya işçi diğerlerine göre daha kaliteli veya hızlı çalışabiliyor olabilir. Bu kavramın hangi kaynağın hangi işe atanacağına karar verirken kullanılması gerekir.

İşin Gereksinimleri: Hangi kalite ve maliyet standartları istendiği, işin ne zaman bitirilmesi gerektiği veya operasyonların sırası gibi gereksinimler bilinmelidir.

Ölçüm Standartları: Zaman, maliyet, kalite ve kapasite ile ilgili bilinen her şey için ve bunların tahsisi ile ilgili standartlar oluşturulmalıdır.

Özetle üretim çizelgeleme, iş emirlerinde bulunan işlemlerin, atölyelerin üretim olanaklarına en rasyonel şekilde yüklenmesidir.

Çizelgelemenin temel amaçları şu şekilde ifade edilebilir [89]:

- 1-Üretim olanaklarının en etkin şekilde kullanımı
- 2-Müşteri taleplerine olabildiğince çabuk cevap verilmesi
- 3-İşlerin, teslim tarihlerinde gecikmeye neden olunmadan tamamlanması
- 4-Yarı mamul envanterinin en küçüklenmesi
- 5-Fazla mesai çalışmalarının en küçüklenmesi

Çizelgelemede başka önemli bir konu atölye yapısıdır. Akış Atölyesi tipinde sürekli ve hücreli üretim sistemi mevcuttur. Burada bütün işler aynı rota ve sabit bir sıralamaya göre gerçekleştirilir. Ancak İş atölyesinde siparişe göre üretim vardır ve her işin kendine göre bir rotası vardır. Hücreli üretim sistemleri akış sürelerinin işlem stoklarının, üretim hazırlık sürelerinin, malzeme taşıma maliyetlerinin azaltılmasını sağlamaktadır. Hücrenin şekli, yapısı (emek/sermaye yoğunluğu) çeşidi (makine, fabrikasyon, montaj, hibrid) çizelgelemeyi etkilemektedir. Çizelgeler çok basit veya kompleks olabilir. Basit bir dolum ünitesinin çizelgenmesi ne kadar kolaysa; bir otomobil montaj hattı için gerekli çizelge o ölçüde karmaşık ve zordur [89].

## 2.7 Birleşik Eniyileme Durumu

Optimizasyon problemleri, karar değişkenlerinin sürekli ve kesikli olmalarına göre iki kısımdan oluşur. Karar değişkenleri kesikli olanlara “birleşik eniyileme (Combinatorial Optimizations)” problemleri denir [87].

Birleşik Eniyileme Durumu, optimal bir düzenleme, gruplama, sıralama veya genelde sonlu sayının kesikli amaçlarının seçiminin bulunması için yapılan bir matematiksel çalışmadır. Tipik bir birleşik minimizasyon problemi bir dizi örnek ile belirtilebilir. Her örnek bir çözüm uzayı  $\Omega$ , bir fizibil uzay  $X$  ( $X \subseteq \Omega$ ) ile ilişkilendirilir. Bunlar problem kısıtları tarafından tanımlanır. İnfizibil bölge  $\Omega / X$  ve amaç fonksiyonu  $C$  dir. Bu her  $S \in \Omega$  için, gerçek bir maliyet değerini atamaktadır.  $C : \Omega \rightarrow R'$  dir.  $S$ , çözüm'dür.  $R$ , ise reel sayılardır [90].

Genelde Birleşik Eniyileme Problemi şöyle gösterilecektir:

P: Minimize (or Maximize)  $C(S)$

Subject to  $S \in X \subseteq \Omega$

$X$  ve  $\Omega$  kümeleri parçalıdır ve bir dizi karar değişkenleri ile tanımlanabilir. Bu değişkenler farklı tamsayı değerleri (problem formülasyonu içindeki rollerine göre) alabilirler. Değerleri açıkça verilmez ama kesin alan içinde tanımlanır. Optimizasyon probleminin amacı, optimal fizibil çözümün  $S^* \in X$  in bulunmasıdır.  $C(S^*) \leq C(S')$  tüm  $S' \in X \subseteq \Omega$  olmaktadır. Birleşik Eniyileme problemleri örnekleri; lokasyon problemleri, gezgin satıcı problemi, atama ve sıralama problemleri, araç rotalama problemleridir.

Birleşik Eniyileme Problemleri normalde kolay tanımlanırken, çözümü zordur. Osman ve Kelly'nin (1996) [90] kitabında “Hesapsal karmaşıklık” kavramının ilk olarak Cook tarafından 1971'de gösterildiği belirtilmiştir.

Bir problemin “kolay” olarak çağrılması için, problem örneğinin boyutu ile sınırlandırılmış bir “polynomial zaman karmaşıklığı” içinde, problemin her örneğinde optimalliği çözen bir algoritmanın geliştirilmesi gerekir. Bir problem “zor” ise, etken

algoritmalar çözüm için var olmamaktadır. Hesapsal karmaşıklık teorisinin sonuçları, pek çok birleşik eniyileme probleminin zor olduğu ve NP-tam (non-deterministik polinomial zamanda çözülen) sınıfına ait olan durumlara açıklık getirmiştir. Optimal bir algoritma, bu sınıftaki bir problem için, problem boyutu ile katlanarak büyüyen, bir dizi hesapsal aşamayı gerektirecektir. Bu nedenle sadece küçük boyutlu örnekler çözülebilmektedir.

Genel kanı; bu tip nitelikte “zor” problemlerin çözümünü sağlayan etkili algoritmaların her zaman bulunamadığıdır. Dolayısı ile, sezgiseller (veya approximate – tahmini/yakınsayan- algoritmalar) zor birleşik eniyileme problemleri çözümü için tek pratik araçlar yığını olarak düşünülmektedir [90].

Polinomial zamanlı algoritmalar, pratikteki problemlerin çözümünde iyi performans göstermektedirler. Büyük algoritmalar genellikle, üstel algoritma olarak adlandırılırlar. Üstel algoritmalarda, polinomial olmayan artış hızları geçerlidir ( $2^n$ ,  $n!$ ,  $n^2$ ,  $n^{\log n}$  gibi). Kombinatoriyel optimizasyon problemlerinin çoğunluğu, NP-Complete, polinomial zaman sınırı olmayan problemler sınıfına girmektedir. NP problemlerine, polinomial algoritma geliştirilememiştir.

NP kapsamında olan problemler için, asıl optimum çözüm yerine, yakın çözümler tercih edilir. Bu tip problemlerin kesin çözümlerine makul sürelerde ulaşılamadığından, yerel arama ve stokastik arama yöntemleri ile yaklaşık çözümler elde edilir [87].

### PARALEL MAKİNELERİN ÇİZELGELENMESİ

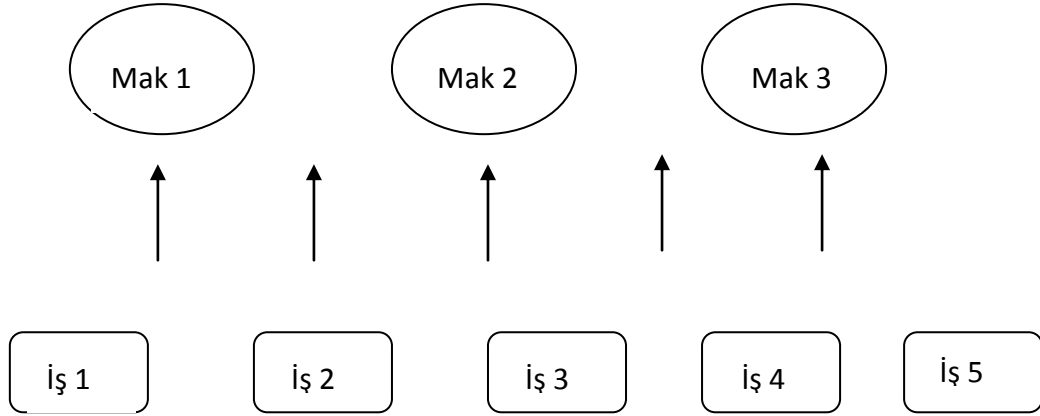
Paralel sıralanmış bir dizi makine hem teorik açıdan hem de pratik açıdan önemli olmaktadır. Teorik bakış açısında, tek makine durumunun genelleştirilmiş durumu vardır ve “esnek kayan bant” halinin özel bir şekli vuku bulmaktadır. Pratik bakış açısı da oldukça önemlidir. Gerçekte karşımıza en sık çıkan durum “paralel kaynaklar” durumudur. Paralel makine durumları, çok aşamalı sistemlerin araştırılması prosedürlerinde de kullanılmaktadır.

Paralel makine çizelgelemede  $n$  sayıda işin  $m$  sayıda makineye atanması söz konusudur. Paralel durumdaki makinelerle ilgilenirken, “yapım zamanı” kayda değer bir amaç haline gelmektedir. Pratikte paralel durumdaki makinelerde yük dengelemesi problemi ile sık sık karşılaşılır ve yapım zamanı minimize edilmeye çalışılır. Böylece sistem içerisinde, iyi bir denge sağlanacaktır [2].

Paralel makinelerde sıralama iki aşamalı bir işlem olarak kabul edilmektedir. Birinci aşamada, hangi işin hangi makinede işleneceği saptanmalı, ikinci aşamada ise her makinedeki iş yükü için “sıralama” tespit edilmelidir. İşlem zamanı amacına göre ise, sadece birinci aşama önemli olacaktır.

Paralel makinelerin olduğu sistemlerde, öncelikler tek makine durumuna göre çok daha büyük önem taşımaktadır. Tek makine koşulunda, öncelikler sıklıkla işlerin “teslim” zamanlarının farklı olması durumunda önem taşıyacaktır. Fakat paralel makine hallerinde, öncelikli işler aynı zamanda teslim edilecekse dahi, önemli olmaktadır [91].

Paralel makine problemleri,  $m$  tane eş makinenin paralel olarak yerleştiği sistemlerdir. Her iş yalnız bir operasyona sahiptir ve bu operasyonlar  $m$  makinenin herhangi birinde işlenebilmektedir. Genellikle makine sayısı, iş sayısından az olmaktadır. Örnek bir paralel makine sistemi Şekil 3.1’de gösterilmiştir [92].



Şekil 3.1 Paralel makine sistem modeli

### 3.1 Paralel Makine Çeşitleri

Literatürde yapılan araştırmalardan paralel makine çeşitleri dört grup altında toplanabilmektedir. Bunlar şöyledir:

- Aynı tip olup, hızları da aynı olan paralel makineler.
- Aynı tip olup, hızları farklı olan paralel makineler.
- Farklı tip olup, aynı hızda olan paralel makineler.
- Farklı tipte olup, farklı hızda olan paralel makineler.

Paralel makineli bir üretim sisteminde özdeş, benzer ve birbirinden bağımsız makinelerden söz edilebilmektedir. Üretim sisteminde yer alan makineler aynı işi yapan, aynı hıza sahip makinelerden oluştuğunda, “özdeş makine çizelgeleme” probleminde bahsedilmektedir. İşlemlerden her biri, sistemde yer alan herhangi bir makinede işlem görebilmekte ve işlem süresi değişmemektedir. Yani bir işlem  $m$  adet özdeş makineden hangisinde işlem görürse görsün, aynı işlem süresine sahip olmaktadır. Genel olarak, literatürde, birbirinden farklı hızlara sahip makineler; “benzer

makinelere" olarak adlandırılırken, makine hızlarının işlere bağlı olarak değiştiği makineler ise "birbirinden bağımsız paralel makineler" adını almaktadır [79].

### 3.1.1 İlişkisiz Paralel Makine ile Problem Tanımı

Esnek atölye akış tipi sistemi bir dizi  $k$  proses aşamalarından oluşan bir diziyi  $O = \{1, \dots, t, \dots, k\}$  tanımlar. Her aşama  $t$  dir. Ve  $t \in O$  dur.  $m^t$  ilişkisiz paralel makinelerinden oluşan dizi ise şöyledir;  $M^t = \{1, \dots, i, \dots, m^t\}$ .  $n$  bağımsız işin dizisi  $J = \{1, \dots, j, \dots, n\}$ ,  $M_1, \dots, M_k$  ile tanımlanan bir makine dizisi üzerinde prosese girmektedir. Her iş  $j$  ve  $j \in J$  dir. Bırakış zamanı (release date)  $r_j$  dir ve  $r_j \geq 0$ 'dır. Ayrıca teslim tarihi  $d_j \geq 0$ 'dır. Her  $t$ ,  $t \in O$  aşaması için sabit standart bir proses zamanı bulunmaktadır. İlişkisiz paralel makinelerin kabulü ile,  $t$  aşamasında  $i$  makinesindeki  $j$  işinin  $p_{ij}^t$  proses zamanı şöyledir;  $ps_j^t/v_{ij}^t$  dir. Burada,  $ps_j^t$   $t$  aşamasındaki  $j$  işinin standart proses zamanıdır.  $v_{ij}^t$  ise  $t$  aşamasındaki  $i$  makinesindeki  $j$  işinin nispi hızıdır [93].

İşlerin proses kısıtlamaları şöyledir;

1. herhangi bir makinede işler öne almasız prosese girmektedir.
2. belli bir zamanda her makinede sadece bir operasyon işlem görmektedir.
3. bir işin operasyonları aşamalar arasında, sırası ile bindirme olmaksızın işlem görmelidir.
4. işlerin bölünmesine izin verilmez.

Paralel şekilde  $m$  sayıda birbirinden farklı makine vardır.  $j$  işi  $i$  makinesinde  $v_{ij}$  hızı ile işlem görebilmekte ve işlem süresi  $p_j/v_{ij}$  olarak gerçekleşmektedir. Makine hızlarının tüm işlerden bağımsız olması durumunda makine hızı  $v_{ij}$  yerine  $v_i$  gelmektedir.

### 3.2 Paralel Tezgâhlardan Oluşan Atölyede Sıralama Problemi

Atölyede birden çok tezgâhın paralel olarak çalışması, sıralama problemi için değişik modeller kurmayı gerektirmektedir. Bu tür bir atölyede bulunan tüm tezgâhlar, mevcut  $n$  adet işi yapabilecek kapasite ve yetenektedirler.

Bu kabulden başka tezgâh ve işler için şu kısıtlar konulmaktadır:

1. m adet tezgâh sürekli olarak çalışabilirler ve bir tezgâh aynı anda birden çok işi işleyemez.
2. Başlangıç anında mevcut, birbirinden bağımsız n adet iş, yalnız bir tezgâh tarafından işlenerek atölyeyi terk ederler.
3. İş tarifleri önceden bilinmektedir, işlem süreleri belirli ve sabittir.

Paralel tezgâhlar hali diğer problemlerden farklı olarak 2 boyutlu bir kararı geliştirmek için incelenmektedir. Kararın birinci boyutu işlerin tezgâhlara dağıtılması, ikinci boyutu ise, bu tezgâhlardaki işlenme sırası ile ilgilidir. Karar prosesi bu iki boyutu kapsayacak şekilde ele alınarak aşağıdaki etkinlik ölçütlerine göre problem incelenmektedir [94].

### 3.2.1 Yapım Süresinin Minimizasyonu

Yapım süresinin minimizasyonu problemi, öne geçmeli ve geçmesiz halde bazı farklılıklar göstermektedir.

#### 3.2.1.1 Öne geçmeli halin incelenmesi

Bir iş, bir tezgâhta bitmeden başka bir tezgâha aktarılabilirse, problem oldukça basit bir şekilde çözülebilir. Bu halde yapım süresinin minimum değeri;

$$M^* = \frac{1}{m} \sum_{j=1}^n P_j \quad (3.1)$$

Şeklinde ifade edilebilir.

$M^*$  değeri hesaplandıktan sonra işler aşağıda verilen algoritma uygulamasında belirtildiği şekilde sıralanır:

Adım 1: Herhangi bir iş, 0 anında başlamak üzere 1. tezgâhta programlanır.

Adım 2: Programlanmamış işlerden biri bu işten sonra gelecek şekilde aynı tezgâhta programlanır. Bu tezgâhtaki işlem süreleri  $M^*$  oluncaya kadar aynı işlem tekrarlanır.

Adım 3: tezgâhta en son programlanan iş  $M^*$  süresi içinde bitirilemiyorsa, bu işe ait geri kalan işlem süresi bir sonraki tezgâhta programlanır ve adım 2 ye dönlür. Tezgâhta en son programlanan iş  $M^*$  süresi sonunda bitiriliyorsa, bir sonraki tezgâha programlanmamış işlerden biri tahsis edilerek adım 2 ye dönlür.

Adım 2 ve adım 3 tüm işler programlanıncaya kadar sürdürölür [83]. Çizelge 3.1'de öne geçmeli hal için yapılan bir örnek görölmektedir.

Öne geçmeli hal içerisinde verilen bu örnekte, 8 iş ele alınmaktadır. 8 işin ayrı ayrı proses zamanları ise aşağıdaki şekilde verildiği gibidir.

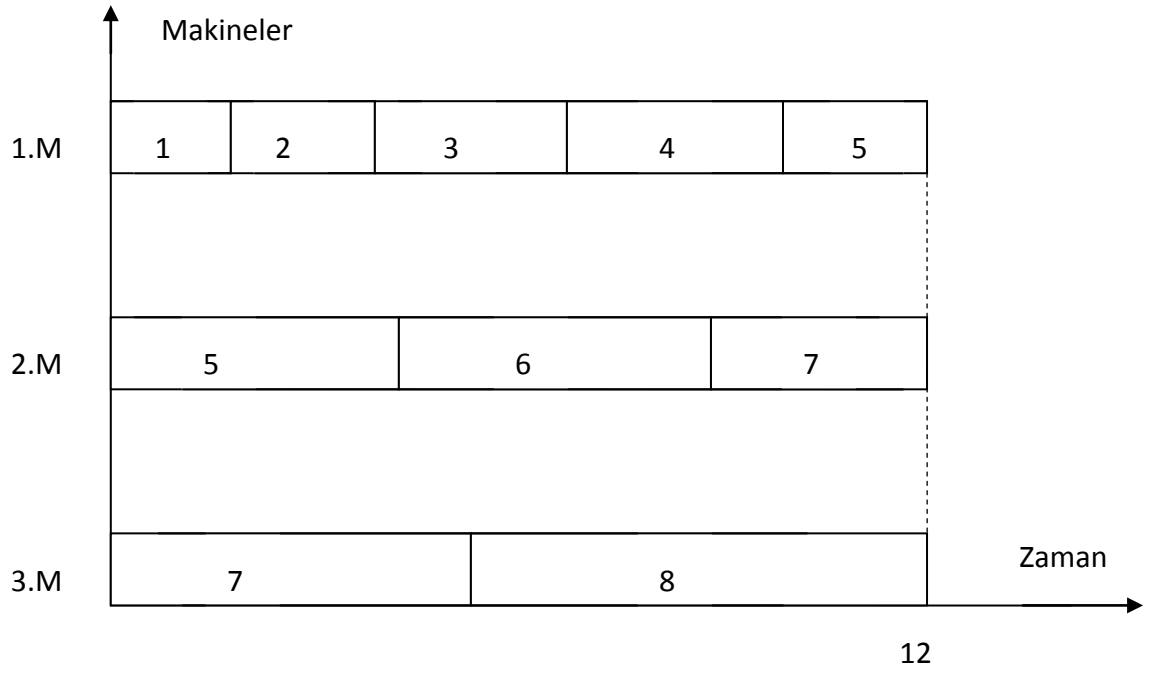
Çizelge 3.1 Öne geçmeli hal ile ilgili bir örnek

J işi	1	2	3	4	5	6	7	8
$P_j$	1	2	3	4	5	6	7	8

$$M^* = \frac{1}{m} \sum_{j=1}^n P_j = \frac{1}{3} \sum_{j=1}^n P_j = 12 \text{ dir.}$$

O halde işler Şekil 3.2'deki gibi gösterilir:





Şekil 3.2 8 iş 3 makine halinin gantt diyagramında gösterimi

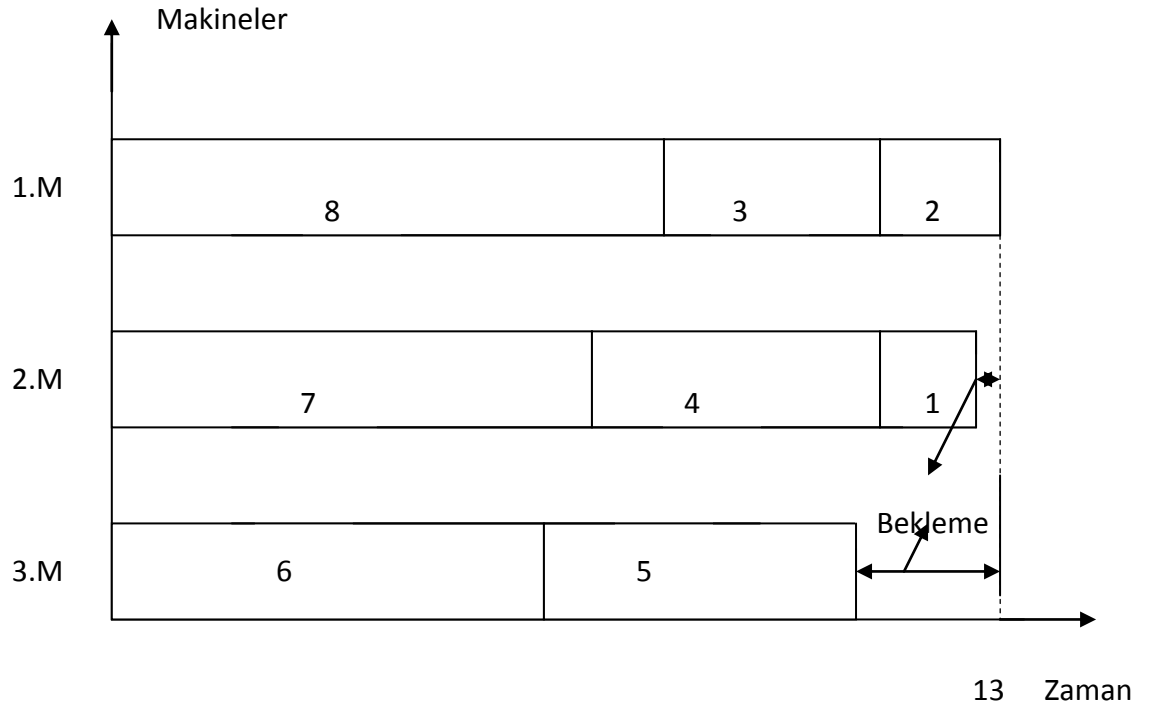
### 3.2.1.2 Öne geçmesiz halin incelenmesi

Öne geçmesiz halde, yapım süresinin minimizasyonu probleminde bazı güçlükler ortaya çıkmaktadır. Tam çözüm için tamsayı programlama yöntemine dayanan bazı modeller kurulmuştur, ancak bunların kullanışlı çözümlerini bulmak mümkün olamamıştır. Öne geçmesiz hal için yeterli hassasiyetle çözüm verebilen bazı sezgisel algoritmalar geliştirilmiştir. Bunlardan biri şöyledir [83]:

Adım 1: İşler LPT (Longiest processing time) sırasına göre dizilir (En uzun işlem süreli iş, en başa gelecek şekilde).

Adım 2: İşler bu sıraya göre tezgahlara tahsis edilir. Tahsis şöyle yapılır. Programlanmamış işlerden ilki en az yüklü tezgaha tahsis edilir. Bu işlem tüm işler programlanıncaya kadar sürdürülür. Örnek problem için bu yöntemin verdiği çözüm şöyledir.

LPT sırası: 8,7,6,5,4,3,2,1



Şekil 3.3 LPT' ye göre oluşturulan Gantt diyagramı

Paralel makinelerin çizelgelenme problemi literatürde genel olarak birbirinden bağımsız işlerin, birden fazla paralel makineye, işlerin toplam akış süresinin (flow time) ya da en son işin tamamlanma süresinin (makespan) en küçüklenmesi olarak ele alınmaktadır (sırasıyla  $P // F_{\max}$  ya da  $P // C_{\max}$  olarak not edilir) [95].

Çizelgeleme problemlerinin karmaşıklık derecelerini konu eden bir çok çalışma yapmış ve çoğunun NP-Tam (NP-Complete) sınıfına ait olduğu sonucu ortaya konmuştur. Paralel makinelerin çizelgelenme problemlerinin çoğu da NP sınıfına aittir. Her ne kadar en uygun çizelgelemeyi bulmak için Matematiksel Programlama uygulama imkânı olsa da, bu problemlerin çözüm süreleri önceden belli olmayan polinomial bir yapı arz ettiğinden, boyutları arttığında etkin çözüm alınamaz. Bu yüzden araştırmacılar, daha etkin ve en uygun sonuca yakın sonuç üretebilen sezgisel algoritmalara yönelmişlerdir. Bunlardan bazıları Tabu Search (Tabu Arama), Yapar Sinir Ağları ve Genetik Algoritma (GA) olarak sayılabilir

Bu algoritmalar arasında GA sonuca hızlı yakınsaması, kolay tatbik edilebilmesi ve en uygun sonuca yaklaşık sonuçlar verebilmesi nedeniyle, özellikle sanayide uygulanan üretim çizelgeleme problemlerine uygulamada, bir adım öne çıkmaktadır [96].

### 3.3 Paralel Makine Çizelgeleme Probleminin İfadesi

Genel görüşe göre paralel makine problemleri tek makine problemlerine göre oldukça zordur. Çünkü hem her bir makinedeki işlerin kendi aralarında sıralanması gerekir hem de işlerin birden fazla makineye paylaşılması söz konusudur.

Paralel makine çizelgelemede temelde ele alınan üç amaç; üretim süresinin, toplam tamamlanma süresinin ve maksimum gecikmenin minimize edilmesidir. Tek makine çizelgelemede üretim süresinin minimize edilmesi amacı genellikle yalnızca hazırlık zamanlarına bağlı bir sıralama oluşturulması ile sağlanıyordu. Diğer taraftan üretim süresi işlem sürelerinin toplamına eşitti ve sıralamadan bağımsız idi. Paralel makine çizelgeleme ile birlikte üretim süresi kayda değer bir amaç haline geldi [78].

Paralel makine çizelgeleme problemi sıklıkla, " $P_m // C_{max}$ " olarak ele alınmaktadır. Makinelerin iş yükü dengesini etkileyeceği için, "üretim süresi minimizasyonu" ile ilgilenilmektedir.  $P_2 // C_{max}$  ve  $P_2 // \sum w_i C_i$  problemleri ise NP-hard olarak nitelendirilmektedir [79].

#### 3.3.1 Özdeş Paralel Makinelerde Sıralama

Özdeş (identical) paralel makinelerde işlerin tamamlanma süresini en küçükleyen (makespan) çizelgeleme problemi şu şekilde tanımlanabilir:  $n$  tane bağımsız iş,  $m$  tane özdeş paralel makineye atanacaktır, her işin işlem süresi sabittir ve her iş, her makinede işlenebilir. En uygun çizelge, tüm işlerin tamamlanması için gereken süreyi en küçükleyen çizelgedir.  $j$  işinin işlem süresi  $t(j)$  olarak not edilmektedir.  $j$  işinin,  $i$  makinesinde işlem görüp görmediği  $x(i, j)$  mantıksal (boolean) değişkeni tarafından belirlenir.  $x(i, j)$  değişkenleri tarafından oluşturulan  $X$  matrisi şu özelliklere sahiptir [97]:

- her elemanın değeri 1 ya da 0'a eşittir,
- her sütununda sadece bir tane 1 vardır,
- toplam  $n$  tane 1 değerine sahip elemanı vardır.

Tüm işlerin tamamlanması için gereken süre de şu şekilde ifade edilebilir [96]:

$$C_{\max} = \max_{i=1}^m \left\{ \sum_{j=1}^n x(i, j) \times t(j) \right\} \quad (3.2)$$

Bu takdirde problemin amaç fonksiyonu ise şu şekilde yazılır:

$$\min C_{\max} = \max_{i=1}^m \left\{ \sum_{j=1}^n x(i, j) \times t(j) \right\} \quad (3.3)$$

### 3.3.2 Özdeş Olmayan Paralel Makinelerde Sıralama

Özdeş olmayan (non-identical) paralel makinelerde ise,  $m$  adet özdeş olmayan paralel makinede  $n$  adet iş prosese sokulmaktadır.  $j$  işi  $i$  makinesiyle de işleme sokulabilmektedir. Her makinenin farklı hızı vardır.

Ayrıca belli bir zamanda sadece bir iş prosese girebilmektedir. Her iş kesilme olmaksızın işlem görmektedir.  $i$  makinesindeki  $j$  işinin proses zamanı  $t(i, j)$  ile ifade edilmektedir. Maksimum tamamlanma zamanını minimize eden bir sıralama bulunmaya çalışılmaktadır.

$x(i, j)$  boolean değerleridir. Burada  $j$  işi  $i$  makinesiyle (eğer  $x(i, j) = 1$ ) işleme sokulmakta, (eğer  $x(i, j) = 0$ ) ise sokulmamaktadır.  $x$  matrisi  $x(i, j)$  değişkenlerinden oluşmaktadır ve aşağıdaki özelliklere sahiptir [96]:

- Bütün elemanlar "0" veya "1" e eşittir,  $x(i, j) \in \{0,1\}$
- Her sütun değeri "1" olan bir tek elemana sahiptir,  $\sum_{i=1}^m x(i, j) = 1, j = 1, \dots, n$
- "1" olarak değerlendirilen eleman sayıları  $n$ ' dir,  $\sum_{j=1}^n \sum_{i=1}^m x(i, j) = n$

Farklı makinelerde işlem gören  $j$  işi sıradaki eşitlikle açıklanabilir:

$$t(i_1, j) \times V_1 = t(i_2, j) \times V_2$$

Maksimum tamamlanma zamanı ise şuna eşittir:

$$C_{\max} = \max_{i=1}^m \left\{ \sum_{j=1}^n x(i, j) \times t(i, j) \right\} \quad (3.4)$$

Böylece, amaç fonksiyonu şu şekilde formüle edilmektedir:

$$\min C_{\max} = \max_{i=1}^m \left\{ \sum_{j=1}^n x(i, j) \times t(i, j) \right\} \quad (3.5)$$

Burada amaç fonksiyonu,  $C_{\max}$  değeri üzerinden ilerlemektedir. Maksimum tamamlanma zamanının minimize edilmesi esastır. Daha önce de söylendiği gibi,  $i$  makinesindeki  $j$  işinin proses zamanı  $t(i, j)$  ile ifade edilmektedir.

$x(i, j)$  boolean değişkenleridir. Burada  $j$  işi  $i$  makinesiyle eğer  $x(i, j) = 1$  oluyor ise, işleme sokuluyor,  $x(i, j) = 0$  oluyor ise, sokulmamaktadır.

$X$  matrisi  $x(i, j)$  değişkenlerinden oluşmakta ve yukarıda da belirtildiği gibi bir takım özelliklere sahip olmaktadır.

Farklı makinelerde işlem görme durumu için, "hızlar" devreye girmekte ve buna göre paralel makine durumları ortaya çıkmaktadır. Farklı makinelerde işlem gören  $j$  işi için bazı eşitlikler tanımlanmaktadır.

Bu eşitliklerden birisi;

$$t(i_1, j) \times V_1 = t(i_2, j) \times V_2 \text{ eşitliğidir.}$$

Bütün bu açıklamalara bağlı olarak da, amaç fonksiyonu (5.5)'de olduğu gibi formüle edilmektedir.

### HİBRİT AKIŞ TİPİ SIRALAMA PROBLEMİ

Her kademe için birden çok paralel makineler ile akış tipi zaman çizelgelemesinin oluşturulması durumu genel olarak “hibrit akış tipi (HFS) atölye sıralama” problemi olarak adlandırılmaktadır. Bu durum birçok gerçek dünya uygulamasında karşımıza çıkmaktadır [62].

Hibrit Akış Tipi, genel olarak, verilen bir amaç fonksiyonunu en iyileyen seri m aşamalı durumda prosese giren bir dizi n işin sıralanması durumudur. Farklı işler aşamaları aynı sırada ziyaret etmektedirler. Her aşamada, bir iş sadece bir makine tarafından işlem görmektedir [98].

Bir dizi değişken ise şu şekilde olmaktadır;

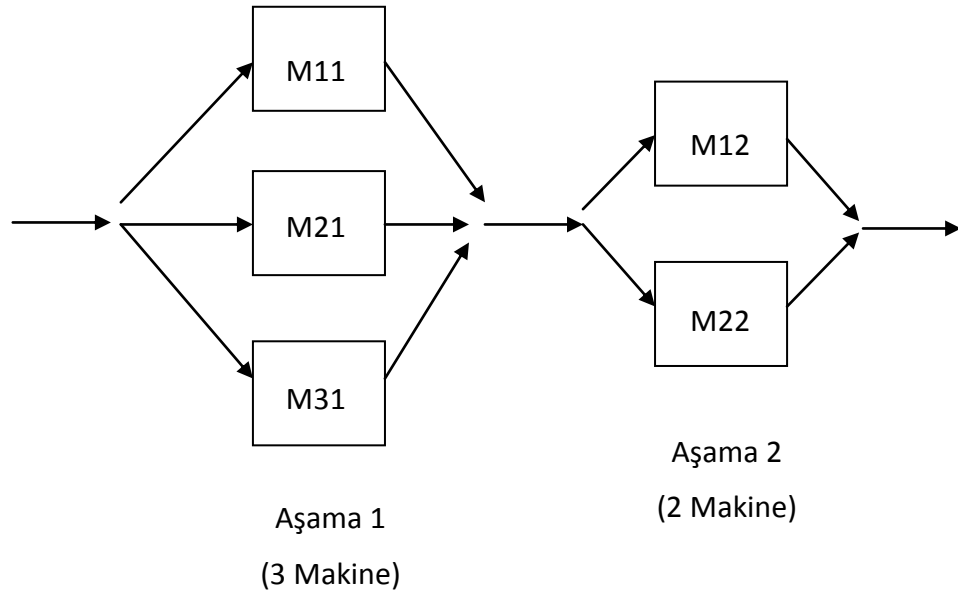
1. Proses aşamalarının sayısı en azından 2'dir.
2. Her kademe en az bir kademe  $M_{(k)} > 1$  olmak üzere  $M_{(k)} \geq 1$  paralel makine içerir.
3. Bütün işler takip eden aynı üretim aşamasında (aşama 1, aşama 2,...,aşama m) işlem görmektedir. İş herhangi bir kademeyi en azından birinde işleme tabi tutulmak şartı ile geçebilir.
4. Aşama k' da, her j işi, bir  $p_{jk}$  proses zamanına sahiptir. Biz k aşamasında j işinin işlenmesini  $O_{jk}$  operasyonu ile göstermeliyiz.

HFS probleminin “standart” formunda bütün işler ve makineler zamanın sıfır anında hazırdır. Verilen bir aşamadaki makineler özdeştir. Herhangi bir makine bir zamanda

sadece bir işi işleme tabi tutabilir ve herhangi bir iş, belli bir zamanda bir makine ile işleme girebilir. Hazırlık zamanları ihmal edilebilir. Aşamalar arasında ki ara stok kapasiteleri sınırsızdır. Problem bilgisi deterministiktir ve önceden bilinmektedir [62].

İşlerin önceliği yoktur, örneğin, bir makinede bir operasyon başladıktan sonra başka bir operasyonun aynı makinede işlem görmeye başlayabilmesi için öncekinin mutlaka bitirilmesi gerekir [92]. Amaç, genellikle en son işin sistemden ayrılma zamanını ( $C_{max}$ ), minimize etmektir.

Esnek akış tipi problemler NP-Zor'dur [99]. Örneğin; Gupta' nın sonuçlarından sonra bir kademenin 2 makine içerdiği ve diğer kademenin tek makine içerdiği, iki kademeli HFS, NP-zor yapıdadır.



Şekil 4.1 HFS yapısı

HFS, elektronik, kağıt ve tekstil sanayileri dahil, her çeşit gerçek dünya durumlarında bulunabilmektedir. Bu örneklere ayrıca beton üretimi, fotografik film imalatı gibi durumlarda eklenmektedir.

imalatin olmadığı durumlarda da Hibrit Akış Tipi Sistem (HFS)'e rastlanmaktadır; internet hizmetleri mimarisi, konteynır kaldırma vs... gibi [62].

#### 4.1 Hibrit Akış Tipi Sistem'in Notasyonu

Problemler, atölye biçimine ( $\alpha$ ), sabitler ve varsayımlara ( $\beta$ ) ve dikkate alınan amaç fonksiyonuna ( $\gamma$ ) göre sınıflandırılmıştır. Her problem  $\alpha$ ,  $\beta$ ,  $\gamma$  üçlüsü ile tanımlanır.  $\alpha$  parametresi kademe sayısını ve her kademedeki makinelerin sayısını ve karakteristiğini içeren atölye yapısını belirtir.  $\alpha$ ;  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$  ve  $\alpha_4$  parametrelerinin birleşiminden oluşur.

$\alpha_1$ , genel akış tipini, misal "melez" akış tipini gösterir.  $\alpha_2$ , atölyedeki kademeleri,  $\alpha_3$  ve  $\alpha_4$  ise birlikte, her kademedeki makinelerin özelliklerini tanımlar.  $(\alpha_3 \alpha_4)^k$  yaklaşımı, k kademesinde  $\alpha_3$  tipinde  $\alpha_4$  adet paralel makine olduğu anlamına gelmektedir [100].

P özdeş paralel makinelerin, Q farklı hızlarda paralel makinelerin, R birbirinden bağımsız paralel makinelerin göstergesi olmak üzere  $\alpha_3 \in \{\emptyset, P, Q, R\}$  dir. Tek makine durumunda  $\alpha_3 = \emptyset$  olmaktadır. İkinci eleman  $\beta$ , problemleri karakterize eden sabitler ve varsayımları listelemektedir. En yaygın olanları ise şöyledir [62]:

- $r_j$ , j işinin  $r_j$  bırakış süresinden (release date) önce başlayamayacağını gösterir.
- prmu; her kademedeki işleme tabi tutulan işlerin yer değiştirmesi ile ilgilidir (permütasyon).
- prec; farklı işlerdeki operasyonlarda öncelik zorlamaları olduğunu göstermektedir.
- $M_j$ ; işlem gören j işinin, k kademesinde  $M_j$  makine kümesi ile sınırlı olduğunu gösterir. Uygunluk durumu olarak bilinmektedir.
- $S_{sd}$ ; hazırlık zamanlarının ardışık operasyonlara bağımlı olduğunu göstermektedir.
- prmp; öne alımlara izin verildiğini göstermektedir.
- block; kademeler arasında ki ara stok kapasitesinin sınırlı olduğunu belirtir. iş yeterli "boşluk" çıkana kadar bir önceki kademedeki bekletilmelidir.
- recrc; işlerin aynı kademedeki birden çok kez işlenmesi için izin verildiğini ve bu durumun gerekli olduğunu göstermektedir.



- unavail; makinelerin her zaman hazır olmadığına işaret eder.
- no-wait; işlerin iki ardı ardına gelen kademeler arasında beklemesine izin vermez. Bu aslında atölyenin “ilk giren ilk çıkar-FIFO” kuralına göre işlem yaptığına dikkat eder.
- $p_j = p$  ; bütün işlem zamanlarının  $p'$  ye eşit olduğunu göstermektedir.
- $size_{jk}$ ;  $o_{jk}'$  nın  $size_{jk}$  makinelerinde aynı anda işlem görmesi gerektiğini vurgulamaktadır.

$C_{jk}$ ; k kademesindeki j işinin tamamlanma zamanını belirtir.  $c_j$  ise atölyenin son kademesindeki (m kademe) j işinin tamamlanma zamanını göstermektedir.

$j$  işinin akış zamanı,  $F_j$ , sistemde harcanan zamandır ve  $F_j = c_j - r_j'$  dir.

$L_j$  j işinin gecikme zamanı ve  $C_j - d_j'$  dir.  $L_j$  , negatif olabilmektedir.

Gecikme,  $T_j = \max\{C_j - d_j, 0\}$  olmaktadır. Erken bitme ise;  $E_j = \max\{d_j - C_j, 0\}$  olurken, negatif olmamaktadır.

$U_j \in \{1,0\}$  olarak, her geç kalmış iş için bir ceza ölçüsünü tanımlanmaktadır.

$C_j - d_j > 0$  olursa, 1 değerini, aksi takdirde 0 değerini almaktadır.

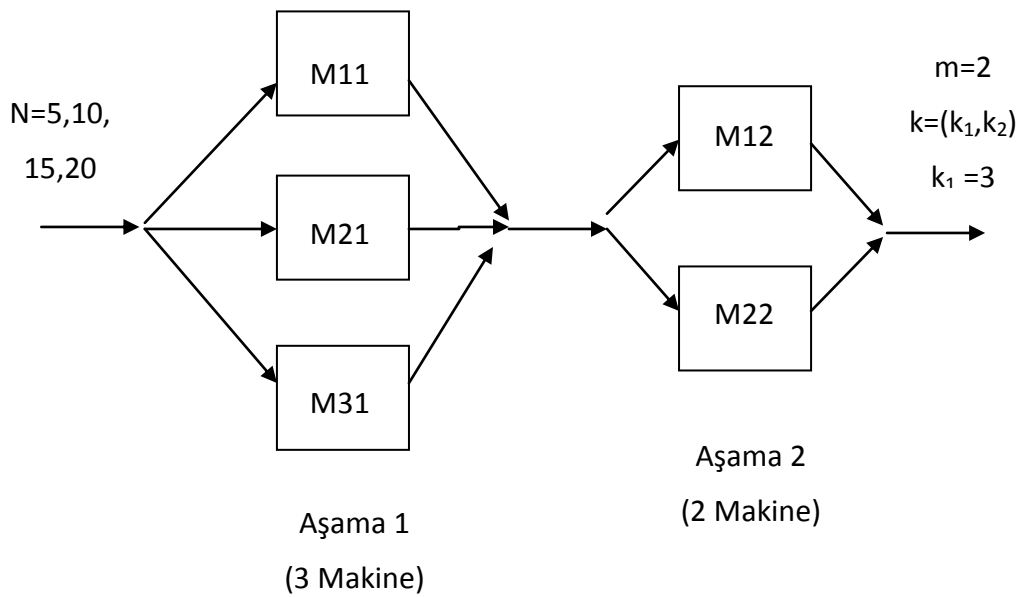
Modelleme aşamasında, j işini  $w_j$  ağırlığı ile birleştirmek önemlidir. Böyle bir ağırlık, işin maliyetini, hacmini, ilişkisel önceliğini temsil edebilmektedir.

$C_j$ ,  $F_j$  göstergeleri ve onların ağırlık eşlenikleri,  $w_j C_j$ ,  $w_j F_j$ , çoğu kez amaç fonksiyonunu tanımlamak için kullanılmaktadır [62].

İşlerin tamamlanma zamanlarını ( $C_j$ ) içeren fonksiyonlar, kapasite kullanımı ile ilgilidir. Fonksiyonlarla ilişkili teslim tarihleri,  $T$ ,  $L$  ve  $U$  ceza hataları, müşteri talepleri gibi

durumları zamanında karşılamaktadır. E'yi içeren fonksiyonlar ise, genelde “envanter” veya “bitmiş” ürünlerin maliyeti ile bağlantılı olmaktadır.

HFS problemi için, tamamlanma zamanı minimizasyonu ve  $m$ -aşamalı “benzer” paralel makineler ile çalışılması, doğal bir başlangıç noktası olmaktadır. Ruiz ve Rodrigues (2010) [62] makalelerinde, “araştırmacılar ister akademiden, ister endüstriden olsun, HFS’ nin bu standart formunu çalışmaları onlar için daha doğru bir nokta olacaktır” demişlerdir.



Şekil 4.2 HF2 (P3,P2) //C<sub>max</sub> yapısı

Özetle; HFS, bir grup aşamadan ve her aşamada 1 ya da daha çok paralel makineden oluşan bir sistemdir. Farklı işler, aşamaları aynı sırada ziyaret ederler. Her aşamada 1 iş sadece 1 makine tarafından işlenmektedir. Her aşama arasında, işler sınırlı ya da sınırsız tampon stoklarda bekleyebilir ya da bekleyemez.

Şekil 4.2’deki sistemde, 2 aşama ve her aşamada makineler görülmektedir. 1.aşamada 3 makine, 2.aşamada ise 2 makine vardır. Bir sınırsız kapasitenin tampon stoğu, sistemin aşamaları arasında birleştirilmektedir. Ayrıca tüm işlerin sıfır değerinde “bırakiş zamanı (release date)” olarak sisteme giriş yaptığı varsayılmaktadır.

Makinelere atanan işler ve sistemin girişinde sıralanan “n” adet iş bulunmaktadır. Makinelerde sıralanan işlerin ve her aşamada her makineye atanan her işin bir amaç (bir performans kriterini minimize etmek gibi) içerisine dahil edilmesi gerekmektedir.  $C_{max}$  , son aşamada son işin tamamlanma zamanı'dır. Bu şekildeki notasyon ise; HF2 (P3,P2) //  $C_{max}$  şeklinde olmaktadır.

## BÖLÜM 5

---

### ÇİZELGELEME YAKLAŞIMLARI

Üretim sistemlerinde hangi tezgâhta (iş istasyonunda) hangi işin, hangi sıra ile yapılması gerektiğini bulmamıza yarayan işlem, iş sıralamadır. İş sıralama ve çizelgeleme ise, işlerin makinelere en uygun şekilde atanıp, makinelerde sıralanması ve gerekli kriterler/kısıtlar ile programlanmasıdır.

Çizelgeleme yapılırken ilk önce problem tipine karar verilmelidir. Çizelgelemeye temel modeller açısından yaklaşıldığında, kaynakların düzeni ve görevlerin tabii yapısı dikkate alınarak bir problem türü gruplandırması yapılabilir. Örneğin; atölye tipi/akış tipi çizelgeleme problemleri şeklinde ya da tek makine/paralel makine, statik/dinamik çizelgeleme problemi şeklinde gruplandırma olabilir [101].

Ayrıca, literatürde, çok ölçütlü çizelgeleme problemleri tek makineli ve akış tipli sistemler üzerinde yoğunlaşmış olmasına rağmen, paralel makineli sistemler için yapılan çalışma sayısı oldukça sınırlıdır.

Bunun tam tersi olarak, yukarıdaki problem türlerini çözecek teknikler açısından yaklaşıldığında ise, bu tekniklerin yapısı gereği bir gruplandırmaya gidilebilir.

Bu teknikler ise, analitik yaklaşımlar ve sezgisel yaklaşımlardır.

## 5.1 Analitik Yaklaşım

Analitik yaklaşımlar; bağımsız değişkenler tarafından kısıtlanan bir fonksiyonu optimize etmeye yarayan kuyruk teorisi, lineer programlama, Langrange açılımı, dal-sınır algoritması, dinamik programlama v.b. gibi teknikleri içine almaktadır. Bu teknikler kesin (tüm detayları bilinen) çözümler üretir. Ancak bu teknikler, çizelgeleme probleminin doğası gereği, büyük hacimli problemleri çözmede çok yararlı olamamaktadır.

Makine sayısının arttığı durumlarda, çizelgeleme problemlerinin çözümü zorlaşmaktadır. Dinamik programlama, tamsayılı programlama gibi yöntemler, çizelgeleme problemlerine uygulayarak optimal çözümler elde edilmektedir.

Bir tamsayılı programlama tekniği olan dal-sınır algoritması tekniği çeşitli bilim adamları tarafından çizelgeleme problemlerine uygulanmıştır. Bu uygulama küçük boyutlu problemlere optimum sonucu veren bir teknik olmasına rağmen büyük boyutlu problemler genel olarak uygulanmamaktadır. Çünkü çizelgeleme problemi, özel bazı durumlar hariç, polinomial olmayan (üstel) türdendir. Aynı zamanda dinamik atölye ortamında, analitik yaklaşımlar ile çözüme ulaşılamamaktadır.

Dinamik programlamanın da küçük boyutlu problemler dışında uygulama imkânı olmadığından, büyük boyutlu problemler için sezgisel (heuristic) yaklaşım geliştirilmiştir [101].

## 5.2 Sezgisel Yaklaşım

Günümüzün karmaşık ve zor koşulları problemlere hızlı ve kolay çözüm veren yeni çözüm yöntemleri arayışına neden olmuştur. Bu nedenle 1980'li yıllardan bu yana, sezgisel ve analitik çizelgeleme yaklaşımlarına ek olarak meta-sezgisel yaklaşımlar gelişme göstermeye başlamıştır. Meta sezgisel yaklaşımlar, sezgisel yaklaşımların etkili ve verimli çözüme ulaşamadığı zor kombinatoryel problemler için çözüm olarak geliştirilmiştir.

Deterministik sonuçlar veren ve klasik yöntemler olarak anılan analitik yaklaşımlar, kaynaklar nedeniyle her durum için, çözüm sağlayamamaktadır. Analitik yaklaşımla çözümün mümkün olmadığı çizelgeleme problemlerinde sezgisel yaklaşımlar kullanılmaktadır.

Sezgisel algoritmalar optimal sonucu garanti etmeyen, yaklaşık çözümler üreten ancak uygulaması kolay ve hesaplama zamanı kısa olan yaklaşımlardır. Sezgisel algoritmalar büyük hacimli problemlerde (yüz binlerce değişkenin olduğu) ve/veya dinamik atölye ortamında, karmaşık problemlerin çözümünde kullanılan deterministik olmayan yöntemlerdir. Optimal sonucu değil; optimele en yakın olan sonucu, en kısa sürede üretmeye çalışırlar.

Sezgisel yaklaşımda uygulanan yöntemin doğruluğunun ispat edilmesi gerekmez, tek istenen karmaşık bir problemi daha basit hale getirmesi veya algoritmanın tatmin edici bir sonuç bulabilmesidir.

Bu sezgisel algoritmalar; gerçek zamanlı çizelgeleme yaklaşımı, benzetim tabanlı çizelgeleme yaklaşımı, çeşitli komşuluk ve tabu araştırma metotları, uzman sistemler yaklaşımı (ES), yapay sinir ağları yaklaşımı (ANN), genetik algoritma yaklaşımı (GA) ve melez yaklaşımlar olarak sınıflandırılabilir.

Unutulmaması gereken, sezgisel yaklaşımların problemler için özelleşmiş olduğudur. Problem alanı bilgilerini kullanırlar ve probleme göre iyileştirilirler. Sezgisel yaklaşımların probleme özgü tasarlanması onun uygulama alanını sınırlamaktadır. Başka bir problem sahasına geçildiğinde genellikle aynı sezgiseller kullanılamamaktadır. Bu nedenle sezgisel yöntemlerin düşük performans gösterdiği ya da yetersiz kaldığı, denenmesi gereken ihtimalin çok fazla olduğu durumlarda, daha az deneme ile “iyi” çözümü önermek amacıyla sezgisel yöntemlerin bir alt grubu olan meta sezgisel yaklaşımlar kullanılmaktadır [102].

Sezgiseller 3 sınıfa ayrılabilir: indeks-geliştirme (index-development), çözüm-oluşturma (solution-construction) ve çözüm geliştirme (solution-improvement). Bununla beraber

bazı sezgisellerin bu kısımlardan bir veya daha fazlasını içermesi de mümkündür. Bir gönderme kuralı (dispatching rule) indeks geliştirme içinde yer alır. Çoklu ekleme sezgiseli (multiple- insertion heuristic) ise bir çözüm-oluşturma sezgiselidir. Tabu arama ve benzetimli tavlama gibi meta sezgiseller ise çözüm geliştirme sezgiseli gibi düşünülebilir. Açıkça, çözüm geliştirme sezgisellerinin, çözüm için en uzun hesaplama zamanını gerektirdiği görülmektedir [33].

### 5.3 Meta-Sezgisel Yaklaşım

Meta sezgiseller (Meta - Heuristik), yakınsama (approximate – tahmini / yakınsayan-) metodlarının bir sınıfıdır. Bunlar, klasik sezgisellerin etkili ve verimli olmadığı durumlarda “zor” birleşik eniyileme problemlerine etki edecek şekilde tasarlanmışlardır.

Metasezgisel yaklaşım, klasik sezgisel yaklaşımın etkili ve verimli bir çözüm bulmada başarısız olduğu zor kombinatoriyal problemlerin optimizasyonu için dizayn edilmiştir ve 1980’lerdeki başlangıcından bu yana önemli ölçüde gelişme göstermiştir. Metasezgisel yaklaşım, klasik sezgisel yaklaşımlardan gelen farklı kavramları birleştirerek, çözüm uzayında etkin bir aramayı sağlayan kapsamlı bir yaklaşımdır. Glover (1986) [103] çalışması içerisinde, literatüre kazandırılan metaheuristik kelimesinde “heuristik” bulma, keşfetme anlamına gelirken “meta” ise üst seviye anlamına gelmektedir. Günümüzde meta-heuristik yaklaşımlar modern sezgiseller olarak da anılabilmektedir. Literatürde meta-sezgiseller için farklı tanımlar yapılmıştır: Metasezgisel, arama uzayında araştırma (exploring) ve işletme(exploiting) için farklı kavramları uygun bir şekilde birleştirerek alt seviye sezgisellere rehberlik eden iteratif üretim sürecidir [102].

Meta sezgiseller; klasik sezgiseller, yapay zeka, biyolojik evrim, sinir sistemleri ve statistik mekanik gibi konulardan sağlanan farklı konseptlerin bir araya gelmesi ile yeni “hibrit” durumların yaratılmasına izin veren, genel bir çerçeveyi oluşturmaktadır. Bu yaklaşım aileleri; genetik algoritma, tekrarlı rasgele adaptif yaklaşım, sinir ağları, benzetimli tavlama, tabu arama, basamak kabulü ve hibritlerini içermektedir.

Metasezgisel, tekrarlı oluřum prosesidir. Burada, ikincil bir sezgisel, etkili bir yakın optimal çözümlerin bulunması için, yapısal bilgiye götüren “öğrenme” stratejileri kullanılarak, araştırma uzayını keşfetmek ve kullanmak için yetenekli-farklı konseptlerin birleşimi sağlanmaktadır. Bazı kabuller altında, bir kısım metasezgiseller için, kuramsal yakınsamaların olduđu unutulmamalıdır. Bununla beraber, bu kabullerin pek çok pratik uygulamada, tahmini veya hoşnut edici durumları sağlamadığı da bilinmelidir. Bu yüzden optimallik, kanıtlanma yeteneđi düşük ve “yakın” çözümler olarak sağlanmaktadır. Bu kusura rağmen, meta sezgiseller, optimal veya yakın optimal çözümlerin bulunmasında, pek çok alan içinde, kendi ikincil sezgisellerine oranla oldukça yüksek başarılarla sahiptirler [90].

Metasezgisel, yüksek kaliteli çözümleri etkin bir şekilde üretmek için alt seviye sezgisellere rehberlik eden bir iteratif üst seviye prosesidir. Her iterasyonda, bir çözümleri ya da çözümlerin bir topluluđunu kullanır. Alt seviye sezgiseller, basit yerel arama algoritması ya da çözüm kurucu bir yöntem olabilir [104].

Metasezgiseller, arama uzayında yüksek kaliteli çözümlere ulaşmak için probleme özgü sezgisellere rehberlik eden “yüksek seviye” stratejileridir. Amaç, yerel en iyi tuzaklara yakalanmayı engellemektir. Bu amaca, arama sırasında kötü hareketlere izin vererek ya da rastsal başlangıç çözümleri yerine, çok daha uygun bir yolla üretilen çözümleri kullanarak ulaşılabilir [105].

Metasezgisel yöntemler zaten, arama uzayının yüksek kaliteli çözümlerini kapsayan bölgelerinde aramayı gerçekleştirmek için, probleme özgü sezgisellere rehberlik etmek amacıyla tasarlanan genel amaçlı sezgisel bir yöntemdir [106].

Metasezgisel yöntemlerin özetlenecek özellikleri şöyle sıralanır:

- Metasezgiseller, arama sürecine rehberlik eden stratejilerdir.
- Amaç, en iyi ya da en iyiye yakın çözümleri bulmak için arama uzayını hızlı bir şekilde arařtırmaktır.
- Metasezgiseller, basit yerel arama algoritmalarından, karmaşık öğrenme proseslerine kadar geniş bir yelpazeyi içermektedir.



- Metasezgiseller, yaklaşık algoritmalarıdır ve genellikle deterministik değildir.
- Arama uzayındaki yerel en iyi tuzaklardan kurtulmak için çeşitli mekanizmaları kullanırlar.
- Metasezgiseller, probleme özgü değildirler.
- Metasezgiseller, üst seviye stratejiler tarafından kontrol edilen sezgiseller ile probleme özgü bilgi kullanımına izin verirler.
- İleri seviye metasezgiseller, aramaya rehberlik etmesi amacıyla arama sırasında elde edilen bilgiyi (hafızayı) kullanırlar.
- Metasezgisellerin en önemli özelliği: çeşitlendirme (diversification) ve yoğunlaşma (intensification) arasındaki dinamik dengeyi oluşturmasıdır.

*Çeşitlendirme*; arama uzayında araştırmayı,

*Yoğunlaşma*; arama sırasında elde edilen tecrübenin (bilginin) işletilmesini ifade eder.

Metasezgisel yöntemler sezgisel yöntemlerin aksine probleme bağımlı değildir; problem alanındaki bilgilerle ilgilenmezler. Problemden bağımsız olarak geliştirildiklerinden her problem alanında kullanılabilirler.

En sık kullanılan meta-sezgisel yaklaşımlar şunlardır:

- Genetik algoritmalar
- Tabu araştırması
- Tavlama Benzetimi
- Karınca algoritması
- Yapay sinir ağları
- Değişken Komşu Arama
- Kuş Sürüsü Optimizasyonu

Metasezgiseller;

- Esinlendikleri kaynaklara,
- Aramada kullandıkları çözüm sayısına,
- Kullanılan amaç fonksiyonuna,

- Kullanılan komşuluk yapısına,
- Hafıza kullanımına,

göre sınıflandırılmaktadırlar [107].

Metasezgisel ve sezgisel yaklaşımlar arasındaki farklar, Çizelge 5.1’de yazıldığı gibi olmaktadır.

Meta sezgisel yaklaşımlar, probleme bağlı değildir ve problem alanı bilgilerini kullanmazlar. Her problem sahasında kullanılabilirler. Yerel en iyi sonuca takılmayı engelleyip, global optimumu araştırırlar. Arama uzayının yüksek kaliteli çözümlerini kapsayan kısımlarını hızlı ve etkin bir şekilde tarayıp, optimal sonuca ulaşırlar.

Sezgisel yaklaşımlar ise, problemler için özelleşmişlerdir. Problem alanı bilgilerini kullanıp, probleme göre iyileştirilebilme özellikleri bulunmaktadır. Probleme özgü tasarlandıkları için, başka bir problem sahasına geçildiğinde kullanılamayabilirler.

Yerel arama algoritması ya da “çözüm kurucu” olduğunda başarılı sonuçlar verirler. Ayrıca arama uzayının tamamını tararlar. Bu nedenle optimal sonuca, meta-sezgisellere göre daha uzun zamanda ulaşırlar.

Çizelge 5.1 Metasezgisel ve sezgisel yaklaşım arasındaki farklar

Metasezgisel Yaklaşım	Sezgisel yaklaşım
Problem alanı bilgilerini kullanmaz.	Problem alanı bilgilerini kullanır, probleme göre iyileştirilebilir.
Probleme bağlı değildir.	Problem için özelleşmişlerdir.
Yerel en iyi sonuca takılmayı engeller, global optimumu araştırır.	Yerel arama algoritması ya da çözüm kurucu olduğunda başarılı sonuç verirler.
Her problem sahasında kullanılabilirler.	Probleme özgü tasarlandıklarından başka bir problem sahasına geçildiğinde kullanılamazlar.
Arama uzayının yüksek kaliteli çözümlerini kapsayan kısımlarını hızlı ve etkin bir şekilde tarayarak optimal sonuca ulaşır.	Arama uzayının tamamını tarar. Bu nedenle optimal sonuca meta-sezgisellere göre daha uzun zamanda ulaşır.

### 5.3.1 Yerel Arama Algoritması

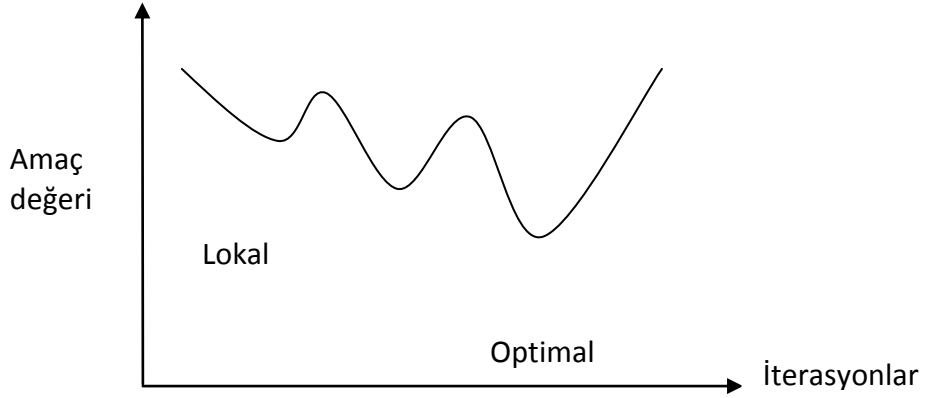
Eğer  $(\hat{c}, f)$  bir minimizasyon problemi olarak kabul edilirse,  $\hat{c}$  çözüm kümesi ve  $f(\hat{c})$  maliyet fonksiyonu olacaktır. Bu şekilde bütün  $\hat{c}$  kümesi araştırılmak zorundadır. Fakat gerçek hayat problemlerinde bu çok büyük bir küme olduğu için, bir alt küme seçilerek bu alt küme içinde çözümün araştırılması sonucunda “yerel arama algoritması (YAA)” ortaya çıkmaktadır.

YAA rastlantı olarak bir  $i \in \hat{c}$  noktasını belirler ve bu noktanın komşularını aramaya başlar. Bu noktanın komşuları,  $\Psi(i)$  fonksiyonu ile belirlenir. Böylece  $i$ 'nin komşusu  $j = \Psi(i)$  olur ve eğer  $f(j) < f(i)$  ise  $i=j$  olmaktadır. Yani rastlantı olarak başlanan aramada seçilen komşu çözümün maliyeti mevcut çözümden daha düşük ise, artık komşu çözüm mevcut çözüm olacaktır ve onun komşuları araştırılacaktır [108]. Yani,  $i \in \hat{c}$  başlangıç çözümü rastlantı olarak belirlenmektedir.

Örnek bir YAA şu şekilde olmaktadır;

```
while (tüm  $\Psi(i)$  araştırılmadı) {
j=  $\Psi(i)$ 
if ( $f(j) < f(i)$ ) i=j
}
```

YAA' nın en önemli dezavantajı ise global en iyi yerine, çoğunlukla yerel en iyi noktaya takılıp kalması olmaktadır.



Şekil 5.1 Global ve Yerel Optimum

Klasik Komşuluk (veya yerel arama) metotları, yakınsama/tahmini (approximate) sezgisellerinin genel bir sınıfını oluşturmaktadırlar. Yakınsama sezgiselleri, geçerli çözümün komşuluğunu (vicinity) keşfeden bir yapı temellidir. Komşuluk çözümleri genel bir hareket mekanizması (move generation) ile oluşturulurlar. Bu çözümler, bazı ön-tanımlanmış kriterlere bağlı olarak seçilir ve kabul edilirler.

Verilen bir birleşik eniyileme probleminde, tanımlanmış üçlü gösterim ise şöyle olmaktadır:

$(X, \Omega, C) \rightarrow$  Fizibil ve infizibil çözümler ile amaç fonksiyonu minimize edilmektedir. Sonraki tanımlamalar ise şöyledir [90];

Bir komşuluk yapısı bir harita gibidir.

$N : \Omega \rightarrow \Omega$  , çözümün her  $S \in \Omega$  bir dizi  $N(S) \subseteq \Omega$  için (ki bunlar S nin komşuluğudur) tanımlanır.

$N(S)$  dizisi S nin komşuluğunu tanımlar ve her  $S' \in N(S)$  ;  $S'$  nin bir komşusu olarak çağrılmaktadır.

Genel bir hareket mekanizması, verilen S örneğinin bir özelliği veya özelliklerinin bir kombinasyonunun değişen komşuluk dizilerinden oluşmaktadır. Bir hareket (move)

yapısı, bir  $S$  çözümünden, diğer bir  $S' \in N(S)$  çözümüne tek bir adımda (veya iterasyon) geçişi göstermektedir.

$\overline{N(S)}$  çözümlerinin bir aday listesi  $S'$  çözümlerinin bir dizisidir. Ki bunlar daha önce tanımlanmış kabul ve giriş kriterlerini sağlamaktadırlar.  $\overline{N(S)} \subseteq N(S)$

Bir çözümün, bir özelliği sık sık,  $P$  veya amaç değeri içinde ikili bir değişken ile ilişkilendirilmektedir. Lokal arama algoritmasının tasarımında şu durumlara ihtiyaç duyulmaktadır [90]:

- i) Başlangıç Aşaması (initialization): Bir başlangıç fizibil çözümleri olarak,  $S$  oluşturulur.
- ii) Mekanizmanın Oluşumu: Fizibil  $S'$  komşuları,  $N(S)$  den sağlanır ve "hareket" oluşum mekanizması kullanılır.
- iii) Kabul ve seçim stratejileri: Eğer kabul edilebilecek çok komşu varsa,  $S'$ ,  $N(S)$  den seçilmelidir. 2 tip seçim kriteri vardır.
  - a- Bir ilk-kabul (FA) stratejisi, ilk  $S'$  komşusunu seçer ki bu durum  $\overline{N(S)}$  yi kaydeder.
  - b- Bir en iyi kabul (BA) stratejisi "en iyi"  $S'$  nü  $\overline{N(S)}$  içinde seçer. Bir kere  $S'$  tanımlanır, sonra o başlangıç çözümleri olan  $S$  ile yer değiştirir. Aksi takdirde  $S$  tutulur ve arama devam eder.
- iv) Durdurma Testi: Durdurma kriterleri, seçim stratejilerine ve lokal arama algoritma yapısına bağlı olmaktadır.

Klasik bir lokal arama prosedürü adımları ise şöyle olmaktadır [90]:

Adım 1: Bir  $S$  başlangıç çözümleri alınır ve onun  $C(S)$  amacı hesaplanır.

Adım 2: Test edilmemiş bir  $S' \in \overline{N(S)}$  komşusu varken, takip eden durumlar şöyledir;

- Sırayla bir  $S' \in \overline{N(S)}$  deneyi oluştur ve  $C(S')$  oluşturup, hesapları yap.
- Eğer  $C(S') < C(S)$  ise, sonrasında  $S'$  nü bir mevcut çözümleri olarak,  $S$  ile yer değiştirir. Aksi takdirde,  $S'$  i tut ve Adım 2' yi tekrarla.

Adım 3: Aramayı sonlandır ve lokal optimum çözüm olarak  $S'$  i getir.

Herhangi bir lokal arama türetim prosedürü ile üretilen lokal optimum, optimallikten çok uzak olabilir. Lokal arama metodları sadece sıralanan çözümleri (ki bunlar amaç fonksiyon değerinde azalmalar oluşturur) kabul ettiği müddetçe, kör bir aramayı gerçekleştirecektir.

Bu yöntemler sadece, algoritmanın icrası sırasında toplanan bilgilerin bazısını kullanmayı amaçlamamaktadırlar. Bu durum, büyük oranda başlangıç çözümlerine ve komşuluk oluşum mekanizmasına bağlı olmaktadır.

Bazı “yerel arama tüetimi” dezavantajlarından sakınmak için ise, yaklaşımın genelliğini ve basitliğini sürdürmek amaçlı, bazı deterministik ve olasılıklı yol gösterici stratejiler geliştirilmektedir [90].

## BÖLÜM 6

---

### GENETİK ALGORİTMA

Genetik Algoritmalar (GA), en iyinin korunumu ve “doğal seçim” ilkesinin benzetim yoluyla bilgisayarlara uygulanması sonucu elde edilen bir arama yöntemidir. Uyarlanabilir, sezgisel algoritmalardandır. Bunlar genetik ve doğal seleksiyonu temel alan evrimsel algoritmaların başını çekmektedirler.

GA temel olarak, evrimsel sistemin doğal işleyişini canlandırabilecek şekilde biçimlendirilmiştir. Bunların başında da ilk kez Charles Darwin tarafından ortaya atılan en uygun olanın hayatta kalabilmesi prensibi gelmektedir. GA da tüm sezgisel yaklaşımlar gibi “rastlantı” aramaya dayanmaktadır.

GA’ nın öncülüğünü 1960’larda John Holland yapmıştır. O günlerden bu yana bir çok araştırmacı tarafından çalışılmış, geliştirilmiş ve mühendislik sahalarında pek çok uygulama alanı bulmuştur. Yalnızca alternatif bir metot olmakla kalmamış, literatürdeki pek çok çalışmada, diğer sezgisel tekniklere üstün geldiği de gözlemlenmiştir [108].

Evrimsel algoritmalar; genetik algoritma, genetik programlama ve evrim stratejilerini içermektedir. Evrimsel algoritma prensiplerinden biri, popülasyondaki en güçlü bireylerin kurtarılma için daha çok şansa sahip olması olmaktadır [36].

Standart bir GA’ da aday sonuçlar “eşit boyutlu” vektörler olarak ifade edilmektedir. Başlangıçta, bu vektörlerden bir grup “rastlantısal” olarak seçilerek, belirli büyüklükte bir popülasyon (toplum) oluşturulmaktadır. Kromozom adı verilen bu vektörler, yeni

nesiller oluşturarak, değişikliklere uğramaktadır. Bir kromozomun üzerindeki genler, n boyutlu vektörlerin bir boyutuna karşılık gelmektedir.

Her yeni nesilde kromozomların iyiliği ölçülebilmektedir, yani her vektör (kromozom) amaç fonksiyonuna yerleştirilerek vermiş olduğu sonuç hesaplanabilmektedir. Bir sonraki nesil oluşturulurken, bazı kromozomlar yeniden üretilebilir, çaprazlanabilir ve mutasyona uğratılabilir. Tekrarlama, çaprazlama ve mutasyondan geçecek olan kromozomlar, amaç fonksiyonu değerlerine göre rastlantısal olarak seçilebilmektedir.

Örneğin, amaç fonksiyonu değeri daha iyi olan kromozomun “çaprazlanma” ya da “tekrar üretilme olasılığı”, değeri kötü olana göre daha fazla olmaktadır. Bunun yanı sıra, popülasyon içinde iyiliği düşük olan kromozomların bir sonraki nesilde yer almamaları mümkün olabilmekte, yani yok olabilmektedirler [36].

Standart bir GA yordamı ise şöyle olmaktadır [109]:

- Olası çözümlerin kodlandığı bir çözüm grubu oluşturulur. Toplumda, yani popülasyonda bulunacak birey sayısı için bir standart yoktur. Genel olarak önerilen 100-300 aralığında bir büyüklüktür. Birey sayısı oluşturulur ve ardından probleme bağlı olarak kromozomların kodlanması gerekir.
- Toplumdaki her kromozomun ne kadar iyi olduğu bulunur. Bu amaçla kullanılan fonksiyona “uygunluk fonksiyonu (fitness function)” denir. Bu fonksiyon yardımı ile kromozomların uygunluklarının bulunmasına ise “evrimleşme” denilir. Uygunluk fonksiyonu, GA' nın beynidir. GA' da probleme özel çalışan tek fonksiyon, uygunluk fonksiyonudur. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirerek, onların bir bakıma şifresini çözmekte (decoding) ve sonra bu parametrelere göre hesaplamayı yaparak, kromozomların uygunluğunu bulmaktadır.
- Seçilen kromozomları eşleyerek “yeniden kopyalama” ve “değiştirme operatörleri” uygulanır. Sonuçta yeni bir toplum oluşturulur. Bu eşleme kromozomların uygunluk değerlerine göre yapılır. Bu seçim için, rulet tekerleği seçimi, turnuva seçimi gibi seçme yöntemleri bulunmaktadır. Yeniden



kopyalama, genlerdeki genetik bilginin birinden diğerine geçmesi işlemine benzediği için çaprazlama olarak adlandırılır. Bu işlem toplumda çeşitliliği sağlar. Değişirme yani mutasyonun etkisi sadece bir çözüm üzerinde olmaktadır.

- Yeni kromozomlara yer açmak için eski kromozomlar çıkartılarak sabit büyüklükte bir toplum sağlanır.
- Tüm kromozomların uygunlukları tekrar hesaplanır ve yeni toplumun başarısı bulunur.
- İşlemler tekrarlanarak verilmiş zaman içinde daha iyi olan yeni nesillerin oluşturulması sağlanır.
- Sonuç olarak, toplumların hesaplanması sırasında en iyi bireyler bulunduğu çözüm elde edilmiş olur.

GA' nın aşamalarından, bu yöntemin "tekrarlı" karakter taşıdığı görülmektedir. Basit bir GA' da üretme sayısı 50-500 arasında değişmektedir.

Yöntemde karşılaşılan ilk sorun, kromozomların oluşturulması ve şifreleme türünün seçimi ile ilgilidir. Bunun için şifreleme (encoding) işlemi ile birlikte çaprazlama (crossover) ve mutasyon gibi GA' nın temel operatörleri açıklanmalıdır. Diğer bir önemli soru ise, çaprazlama için ebeveynlerin nasıl seçileceğidir. Ana düşünce, daha iyi olan ebeveynlerin seçilmesidir. Böylece daha iyi döl üretilebilir. Yeni dölle oluşturulan popülasyonda, son popülasyondaki en iyi kromozomun kaybolacağı da düşünülebilir. Bunun önlenmesi için seçkincilik (elitism) kullanılır. Böylelikle, en azından iyi bir sonuç, hiçbir değişiklik olmadan yeni popülasyona kopyalanır. Bulunan en iyi sonuç korunmuş olunur [109].

Son on yılda yapılan çalışmalarda , m farklı makine ve n işten oluşan ve NP-Zor ( $m > 2$  için) olarak kabul edilen (Chen ve diğ., 1995) [110] akış tipi çizelgeleme problemleri için Genetik Algoritma yine başarılı sonuçlar vermiştir.

Birbirinden farklı m-makine ve n-işden oluşan, her bir işin aynı sıra ile m farklı operasyondan geçtiği, akış tipi çizelgeleme problemleri ile ilgili ilk çalışma Johnson tarafından yapılmıştır. Çok makine ( $m > 2$ ) problemleri için son yıllarda çeşitli sezgisel (heuristic) yöntemler geliştirilmiştir. Bu yöntemler içinde en iyi çözüm veren arama metotlarından birisi Genetik Algoritmadır. Akış tipi çizelgeleme problemlerinin GA ile çözümünde ilk model Cleveland ve Smith (1989) [111] tarafından gerçekleştirilmiştir.

Reeves (1995) [112] n/m/P/Cmax, akış tipi çizelgeleme problemlerini genetik algoritmalar ile çözerek elde ettiği sonuçları tavlama benzetimi (Simulated Annealing-SA) ve komşuluk aralığı tekniği (Neighbourhood Search) ile bulunan sonuçlarla karşılaştırarak GA'nın akış tipi çizelgeleme problemleri için daha iyi sonuç verdiğini göstermiştir.

### **6.1 Genetik Algoritmaların Sahip Olması Gereken Koşullar**

Genetik algoritmaların etkin bir şekilde kullanılabilmesi için, bazı koşulların sağlanması gerekmektedir. Araştırma uzayı parametreleri, ikili (0-1) sistemde veya alfabetik bir şekilde kodlanarak belirli bir uzunlukta kromozom-gen düzeni oluşturulmalıdır. Uygun çözüm sayısını azaltmak için problemin amacı doğrultusunda bir "uygunluk fonksiyonu" kurulmalıdır.

Paralel ve global araştırmaya imkan sağlanması açısından, "araştırma uzayı" içindeki her nokta taranmalıdır. Bir araştırma uzayından diğerine bir gelişme sağlanabilmesi için, olasılıklı geçiş (değişim) kuralları kullanılmalıdır [113].

#### **6.1.1 Genetik Algoritmanın Diğer Yöntemlerden Farkı**

Genetik algoritmalar problemlerin çözümünü parametrelerin değerleriyle değil, kodlarıyla ararlar. Yani parametreler kodlanabildiği sürece çözüm üretilebilmektedirler. Bu sebeple genetik algoritmalar ne yaptığı konusundan öte, nasıl yaptığını bilen bir algoritma içerirler [109]:

- Genetik algoritmalar aramaya tek bir noktadan değil, noktalar kümesinden başlar. Bu nedenle çoğunlukla yerel en iyi çözümde sıkışıp kalmazlar.

- Genetik algoritmalar türev yerine uygunluk fonksiyonunun değerini kullanır. Bu değer kullanılması ayrıca yardımcı bir bilginin kullanılmasını gerektirmez.
- Genetik algoritmalar gerekirci kuralları değil olasılıksal kuralları kullanır.

Genetik Algoritmaları (GA) diğer metodlardan ayıran noktalar şu şekilde sıralanabilir:

- GA, sadece bir arama noktası değil, bir grup arama noktası (adaylar ) üzerinde çalışır. Yani arama uzayında, yerel değil global arama yaparak sonuca ulaşmaya çalışır. Bir tek yerden değil bir grup çözüm içinden arama yapar.
- GA, arama uzayında bireylerin uygunluk değerini bulmak için sadece “amaç-uygunluk fonksiyonu (objective-fitness function)” ister. Böylelikle sonuca ulaşmak için türev ve diferansiyel işlemler gibi başka bilgi ve kabul kullanmaya gerek duymaz.
- Bireyleri seçme ve birleştirme aşamalarında deterministik kurallar değil “olasılık kuralları” kullanır. Diğer metodlarda olduğu gibi doğrudan parametreler üzerinde çalışmaz .
- Genetik Algoritmalar, optimize edilecek parametreleri kodlar ve parametreler üzerinde değil, bu kodlar üzerinde işlem yapar. Parametrelerin kodlarıyla uğraşır. Bu kodlamanın amacı, orjinal optimizasyon problemini kombinezonsal bir probleme çevirmektir.
- Genetik algoritma ne yaptığı konusunda bilgi içermez, nasıl yaptığını bilir. Bu nedenle kör bir arama metotudur.
- Olasılık kurallarına göre çalışırlar.
- Programın ne kadar iyi çalıştığı önceden kesin olarak belirlenemez. Ama olasılıkla hesaplanabilir.
- GA, kombinezonsal bir atama mekanizmasıdır.

## 6.2 Genetik Algoritmada Kullanılan En Temel Kavramlar

Çeşitli problemlerin çözümünde kullanılan genetik algoritmalar, aşağıdaki özellikleri taşırlar [108]:

- Uygun çözümler için bir veya daha fazla “popülasyon” mümkündür.

- Önceden bilinen çoklu çözümlerin özelliklerini bir araya getirerek, yeni uygun çözümler üreten bir mekanizmaya sahiptir.
- Önceden bilinen bir çözümün düzenini rastgele bir şekilde değiştirerek, yeni uygun çözüm üreten bir mekanizmaya sahiptir.
- Popülasyon içinden, öncelik vererek, ayrı çözümleri seçen bir mekanizmaya sahiptir.
- Popülasyon içinden bazı çözümleri dışarı çıkaran bir mekanizmaya sahiptir.

Genetik Algoritmada Kullanılan Temel Kavramlar ise şu şekilde sıralanabilir [114] :

### **6.2.1 Gen**

Kendi başına anlamlı genetik bilgi taşıyan en küçük genetik yapıya “gen” denir. Genetik algoritmanın kullanıldığı “programlama-kod” yapısında gen yapıları, programcının tanımlamasına bağlı olmaktadır.

### **6.2.2 Kromozom**

Birden fazla genin bir araya gelerek oluşturduğu diziye “kromozom” denilir. Kromozomlar, alternatif aday çözümleri gösterebilmektedirler. Kromozomlar genetik algoritma yaklaşımında üzerinde durulan en önemli birim olduğu için bilgisayar ortamında iyi ifade edilmeleri gerekecektir.

### **6.2.3 Popülasyon**

Kromozomlardan oluşan topluluğa “popülasyon” adı verilir. Popülasyon, geçerli alternatif çözüm kümesidir. Popülasyondaki birey sayısı (kromozom) sıklıkla sabit tutulmaktadır. Genetik algoritma’da popülasyondaki birey sayısı ile ilgili genel bir kural bulunmamaktadır. Popülasyondaki kromozom sayısı arttıkça, çözüme ulaşma süresi azalmaktadır. Problemin özelliğine göre seçilecek olan popülasyon sayısı programcı tarafından iyi belirlenmek zorundadır.

### **6.2.4 Allel**

Bir özelliği temsil eden bir genin alabileceği değişik değerlere “Allel (Allele)” denilmektedir.

### **6.2.5 Locus**

Kromozom üzerindeki her bit'in bulunduğu yere verilen isimdir.

### **6.2.6. Genotip**

Kodlanmış çözümden eski haline dönüştürülen çözüm "Genotip (Genotype)" olarak adlandırılmaktadır.

### **6.2.7 Fenotip**

Kodlanan çözüm, "Fenotip (Phenotype)" olarak isimlendirilmektedir.

## **6.3 Genetik Algoritmanın Çalışma Prensipleri**

İlk olarak toplumda bulunacak birey sayısı belirlenmek zorundadır. Kullanılacak sayı için ise belirli bir standart yoktur. Genel olarak önerilen, 100–300 aralığında bir büyüklüktür. Büyüklük seçiminde yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemli olmaktadır. Toplum, bu işlemten sonra belirlenen uzay içerisinde rastgele sayılardan oluşturulacaktır.

Kromozomların ne kadar iyi olduğunu bulan fonksiyona "uygunluk fonksiyonu" denilmektedir. Bu fonksiyon işletilerek kromozomların uygunluklarının bulunmasına ise "hesaplama (evaluation)" adı verilir. Bu fonksiyon genetik algoritmanın beynini oluşturmaktadır. Genetik algoritmada probleme özel çalışan tek kısım bu fonksiyondur. Uygunluk fonksiyonu, kromozomları problemin parametreleri haline getirerek, onların bir bakıma şifresini çözmektedir. Daha sonra bu parametrelere göre hesaplama yapılarak kromozomların uygunluğu bulunacaktır.

Kromozomların eşlenmesi, kromozomların uygunluk değerlerine göre yapılır. Bir sonraki nesil için anne ve baba olacak kromozomlar, seçim metotlarından herhangi birisi ile belirlenir. Anne ve baba olarak belirlenen kromozomlar, yine daha önceden belirlenen çaprazlama metotlarından birisi ile çaprazlanırlar.

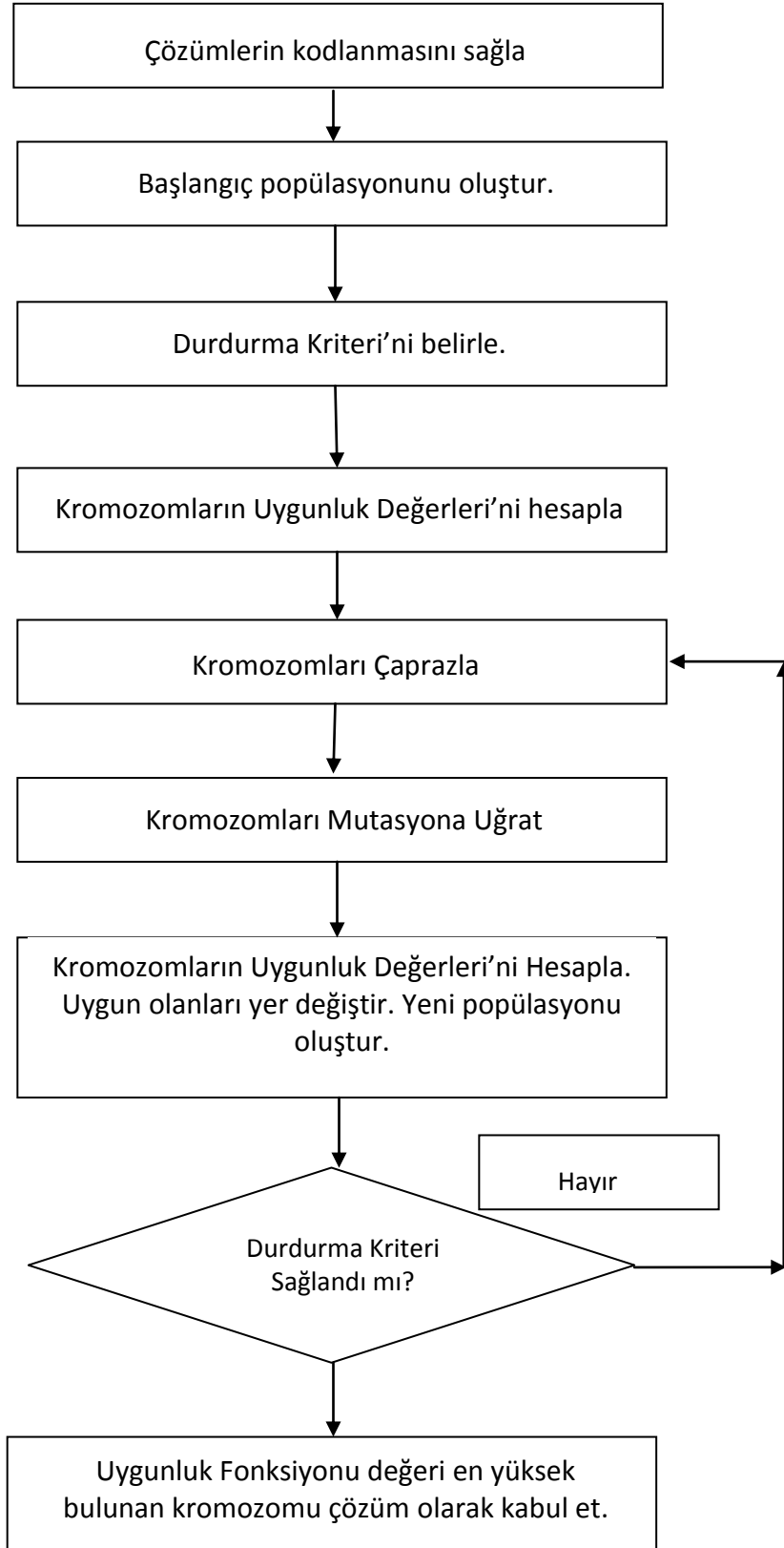
Çaprazlama, genetik algoritmanın en temel operatörüdür. Çaprazlama, toplumda çeşitliliği sağlar. İyi özelliklerin bir araya gelmesini kolaylaştırarak en iyiye yaklaşmayı sağlar.

Mutasyon, kromozomun bir parçasının dışarıdan değiştirilmesi şeklinde tanımlanır. Değiştirme görünüşte genetik algoritmanın dayanak noktasıdır, ancak etkisi bir çözüm üzerindedir. Bu da yalnız başına başarılı olmasını zorlaştırır. Çaprazlama ve mutasyonun olasılıkları için kesin bir sayı yoktur. Mutasyon olasılığı 0.01–0.001, çaprazlama olasılığı 0.5–1.0 aralığında tavsiye edilir.

Çaprazlama ve mutasyon sonucu elde edilmiş yeni kromozomlar ile eski kromozomlar uygunluk değerlerine bağlı olarak en iyiden en kötüye doğru sıralanırlar. Uygunluk değeri kötü olan kromozomlar çıkartılarak sabit büyüklükte bir toplum sağlanır.

Genetik algoritma defalarca çalıştırılarak çok sayıda toplum oluşturulup hesaplanır. Toplumların hesaplanması sırasında en iyi bireyler saklandığı için, o ana kadar bulunmuş en iyi çözüm, gerçek çözümdür. Aşağıdaki şekilde, genetik algoritmanın akış diyagramı verilmektedir [114].

Şekil 6.1’de genel olarak bir Genetik Algoritma akışı görülmektedir.



Şekil 6.1 Genetik algoritmanın akış diyagramı

### **6.3.1 Kodlama yöntemleri**

Kodlama planı Genetik algoritmanın en önemli kısımlarından birisini teşkil etmektedir. Böylelikle, probleme özgü bilginin kromozomlar ile temsili sağlanmaktadır. Kromozom sıklıkla, problemdeki değişkenlerin belli bir düzende sıralanması olarak tanımlanmaktadır. Kromozomu oluşturmak için sıralanmış her bir değişkene ise “gen” adı verildiği daha önce bu çalışma içerisinde açıklanmıştı. Yani, bir gen kendi başına “anlamli” genetik bilgiyi taşıyan en küçük genetik yapı olmaktadır.

Mesela; 101 bit dizisi bir noktanın x-koordinatının ikilik düzende kodlandığı gen olabilmektedir. Aynı şekilde bir kromozom ise, bir ya da daha fazla genin bir araya gelmesiyle oluşan ve problemin çözümü için gerekli tüm bilgiyi üzerinde taşıyan genetik yapı olarak tanımlanabilmektedir. Örnek vermek gerekirse; 100011101111; (x1, y1, x2, y2) koordinatlarından oluşan iki noktanın konumu hakkında bize bilgi verecektir.

Bu parametreleri kodlarken dikkat edilmesi gereken en önemli noktalardan biri ise kodlamanın nasıl yapıldığıdır. Misal, bazen bir parametrenin doğrusal ya da logaritmik kodlanması GA performansında önemli farklara yol açabilmektedir.

Kodlamanın diğer önemli bir hususu ise, kodlama gösteriminin nasıl yapıldığıdır. Bu durum da, yeterince açık olmamakla birlikte, GA performansını etkileyen bir nokta olmaktadır [115].

GA’da kodlamalar, “binary kod (ikili kod)” ve “real kod (gerçek kod)” olarak 2 tip gösterime tabi tutulmaktadır. Binary Kod’lu GA içerisinde kromozomlardaki her bit “mutasyon oranı” olarak bilinen bir “olasılık” ile değiştirilmektedir.

### **6.3.2 Uygunluk teknikleri**

Başlangıç topluluğu yani popülasyonu bir kez oluşturulduktan sonra evrim işlemleri başlayacaktır. Genetik algoritma, bireylerin uygunluklarına göre ayrılıp, fark edilmesine gerek duymaktadır. Uygunluk, topluluktaki bazı bireylerin problemi nasıl çözeceğine



dair iyi bir ölçü olmaktadır. Yüksek ihtimalle uygun olan bu üyeler tekrar üreme, çaprazlama ve mutasyon operatörleriyle seçileceklerdir.

Bazı problemler için bireyin uygunluğu, bireyden elde edilen sonuç ile tahmin edilen sonuç arasındaki hatadan bulunabilmektedir. Daha iyi bireylerde bu hata sifıra yakın olmaktadır. Bu hata genellikle, tekrar sunulacak kombinezonlarının ortalaması veya toplamıyla hesaplanmaktadır. Beklenen ve üretilen değer arasındaki korelasyon etkeni, uygunluk değerini hesaplamak için kullanılabilir [116].

Genetik algoritmaların etkin bir şekilde kullanılabilmesi için, aşağıdaki koşulların sağlanması gerekmektedir. Araştırma uzayı parametreleri, ikili (0-1) sistemde veya alfabetik bir şekilde kodlanarak belirli bir uzunlukta kromozom-gen düzeni oluşturulmalıdır. Uygun çözüm sayısını azaltmak için problemin amacı doğrultusunda bir uygunluk fonksiyonu kurulmalıdır. Paralel ve global araştırmaya imkan sağlanması açısından, araştırma uzayı içindeki her nokta taranmalıdır. Bir araştırma uzayından diğerine bir gelişme sağlanabilmesi için, olasılıklı geçiş (değişim) kuralları kullanılmalıdır [113].

#### **6.4 Genetik Algoritma Operatörleri**

Holland tarafından 1975’de geliştirilen Genetik Algoritma (GA), pek çok kombinatoriyel optimizasyon problemleri içerisinde başarılı olarak uygulanmıştır. Genetik algoritma rassal arama tekniklerini kullanarak çözüm bulmaya çalışan, “parametre kodlama” esasına dayalı bir çözüm arama tekniğidir [117].

Genetik Algoritmalar, yeni bir nesil oluşturabilmek için 3 aşamadan geçmektedirler [117]:

- Eski nesildeki her bir bireyin uygunluk değerini hesaplamak.
- Bireyleri, uygunluk değerini göz önüne alarak (uygunluk fonksiyonu ) seçmek.
- Seçilen bireyleri, çaprazlama (crossover), mutasyon (mutation) gibi genetik operatörler kullanarak eşleştirmek.

Genetik algoritmanın etkinliđi kullanılan parametrelere bađlı olarak deđiřmektedir [110]. Akıř tipi izelgeleme problemlerinin genetik algoritma ile özümünde özüm kalitesi üzerinde etkisi olduđu bilinen bazı parametreler kullanılarak deneyler yapılmıřtır.

aprazlama ve mutasyon, Genetik Algoritma yönteminin en önemli kısımlarıdır. Yöntemin performansı bu iki operatörden ciddi manada etkilenmektedir. Bu operatörlerin her ikisi de, bir popülasyondan yeni bir popülasyon oluştururken kullanılmaktadırlar.

#### **6.4.1 Bařlangı Popülasyonu**

Problemlerin test edilmesinde, literatürdeki alıřmalara benzer olarak, bařlangı popülasyonu büyüklüđu problem büyüklüđüne bađlı olarak önemli olmaktadır. Bařlangı popülasyonu, genellikle 10-30 aralıđında seilmektedir [118]. Bu operatör, “popülasyon büyüklüđu” olarak da ele alınabilmektedir.

#### **6.4.2 aprazlama**

aprazlama, genetik algoritma ierisinde, ebeveynlerden bazı genleri alarak yeni bireyler oluřturma iřlemine verilen isimdir. aprazlama iřlemi sırasında konum rastgele seilir. Oluřan yeni birey, ebeveynlerinin bazı özelliklerini almıř ve bir bakıma ikisinin kopyası olmuřtur.

aprazlama iřlemi, literatürde bařka řekillerde de yapılabilir. Birden fazla aprazlama noktalarında daha iyi performans almak amacıyla deđiřik aprazlamalar kullanılabilir [119].

Popülasyonda ne oranda bir aprazlamaya izin verildiđi aprazlama oranı ( $pc$ ) ile gösterilmektedir. Literatürde daha önce eřitli “aprazlama operatörleri” önerilmiřtir. Bunlardan en yaygın kullanılanları “tek noktalı aprazlama” ve “ok noktalı aprazlama” operatörleridir [117].

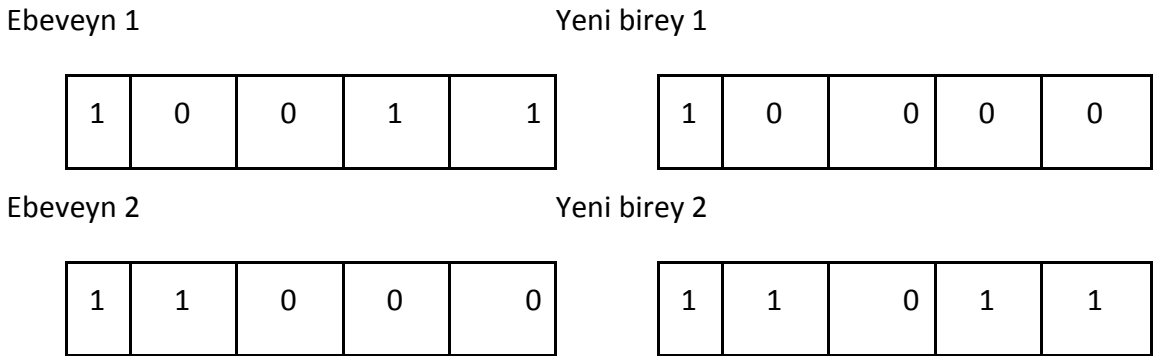
Çaprazlama operatörleri 6 temel başlık altında incelenebilmektedir:

- \* Tek- Nokta Çaprazlama
- \* Çift- Nokta Çaprazlama
- \* Uniform Çaprazlama
- \* Sıraya Dayalı Çaprazlama
- \* Pozisyona Dayalı Çaprazlama Yöntemi
- \* Kısmi planlı çaprazlama

#### 6.4.2.1 Tek Nokta Çaprazlama

Tek nokta çaprazlamada, rassal olarak seçilen iki genin yerleri karşılıklı olarak değiştirilmektedir. Genetik algoritmanın kullandığı en basit çaprazlamadır. Burada çaprazlama işlemi için, kromozomda çaprazlama yapılacak bölge kullanıcı tarafından rastgele seçilebilir (Bolat vd., 2004) [120]. Seçilen noktanın sağında kalan ve “eşleme bölümü” adı verilen bölümlerin yerleri değiştirilerek, iki yeni birey oluşturulacaktır. Şekil 12’de tek nokta çaprazlama şematik olarak verilmiştir. Çok noktalı çaprazlamada ise, kesme noktası sayısı birden fazla olmaktadır [117].

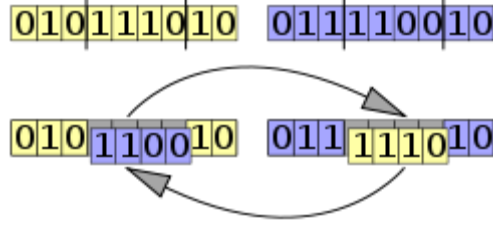
Oluşan yeni birey ebeveynlerin bazı özelliklerini alarak her ikisinin kopyası olacaktır.



Şekil 6.2 Tek noktalı çaprazlama

#### 6.4.2.2 Çift Nokta Çaprazlama

İki nokta çaprazlamada iki nokta arasında kalan alt dizilerin değiştirilmesiyle iki yeni birey elde edilmektedir. Bu yöntemde dizi üzerinde ilk ve son genler hariç iki tane rastgele nokta seçilir. Çaprazlama işlemi seçilen bu iki nokta arasındaki genlerin yer değişimidir. Ve bu noktalar arasındaki genler değiştirilir.



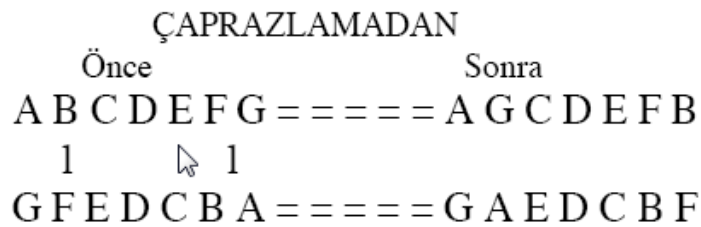
Şekil 6.3 Çift Nokta Çaprazlama

#### 6.4.2.3 Uniform Çaprazlama

Uniform çaprazlamada rastgele olarak çaprazlama maskesi oluşturulmaktadır. Bu durum, birinci ve ikinci kromozoma karşılık gelen genlerin kopyalanmasıyla yaratılmaktadır. Çaprazlama maskesinde “1” o genin birinci kromozomundan, “0” ise o genin ikinci kromozomundan kopyalandığı anlamına gelmektedir [120].

#### 6.4.2.4 Sıraya Dayalı Çaprazlama

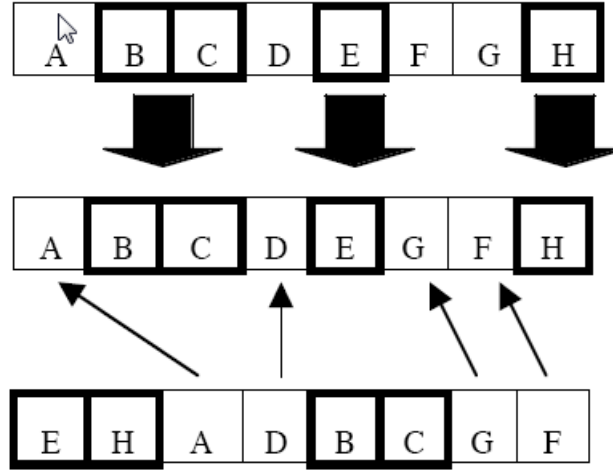
Sıraya dayalı çaprazlama (OBX) yönteminde, bir grup nokta rastgele seçilmektedir. Birinci kromozomun seçilen noktalara karşılık gelen karakterleri aynen yerlerini korumaktadır. İkinci kromozomun seçilen noktalara ait karakterleri, birinci kromozomun aynı noktalarındaki karakterlerinin arkasına getirilmektedir. Geriye kalan boş pozisyonlara ikinci kromozomdan aktarılan yeni karakterler de göz önünde bulundurularak, ilk kromozomun kullanılmayan karakterleri sıra ile (soldan sağa) yerleştirilerek yeni bir kromozom elde edilir. Bu tür çaprazlama, kromozomu oluşturan karakterlerin sayı ve sıralarının önem taşıdığı durumlarda kullanılmaktadır [121]. Bu çaprazlama işlemine ait birer çaprazlama örneği Şekil 6.4’de verilmiştir.



Şekil 6.4 Sıraya dayalı çaprazlama

#### 6.4.2.5 Pozisyona Dayalı Çaprazlama Yöntemi

Pozisyona Dayalı Çaprazlama yöntemi (PBX) içerisinde rassal olarak seçilmiş pozisyondaki genler, bir ebeveynden çocuğa aktarılmaktadır. Diğer işler diğer ebeveynde buldukları sıra ile yerleştirilecektir. Öncelikle pozisyondaki sayılar  $[1,n]$  rassal tamsayılar şeklinde düzenlenir ve daha sonra bu pozisyonlar rassal olarak seçilir. Her pozisyonun çaprazlama olasılığı %50 dir [123]. Şekil 6.5' de sekiz gen içeren iki kromozomdan yapılan pozisyona dayalı çaprazlama örneği görülmektedir.



Şekil 6.5 Pozisyona dayalı çaprazlama

#### 6.4.2.6 Kısmi planlı çaprazlama

Goldberg (1989) [117] tarafından geliştirilen Kısmi planlı çaprazlama (PMX) ilk olarak gezgin satıcı probleminde (TSP) kullanılmıştır. Bu yöntemde iki ayrı gen (iş) sırasında rassal olarak aralıklar belirlenir ve bu aralıkta yer alan genlerin (işlerin) yeri, karşılıklı olarak değiştirilmektedir [117]. Bu yöntem aşağıda bir örnek üzerinde açıklanmaktadır [122]:

Çaprazlama için seçilen ebeveyn yapılar, A ve B olarak adlandırılmıştır ve sekiz iş içermektedir. Yapıdaki elementler aşağıdaki biçimde verilmiştir.

A = 2 8 6 4 5 7 1 3

B = 8 7 2 1 3 4 6 5

A ve B'ye PMX operatörü uygulanır ise, A ve B'den ilk olarak ortak bir aralık, rassal şekilde seçilir. Daha sonra, seçilmiş iki aralıktaki elementlerin değişim planları

belirlenecektir. Bu örnekte, seçilmiş aralıklar arasındaki plan 6'ya 2, 4'e 1 ve 5'e 3 olmaktadır.

İkinci olarak, A ve B' deki iki aralık karşılıklı değiştirilecektir. Bir dizide bir iş tekrarlı olarak yer aldığından, her iki yapının da uygun olmadığı görülmektedir. Buradan elde edilen yapılar geçici sonuçlar olarak değerlendirilecektir.

A = 2 8 2 1 3 7 1 3

B = 8 7 6 4 5 4 6 5

Bundan dolayı, yeni yapıları uygun olmayan A ve B' de, değişim planının uygulanması gerekecektir. Bu örnekte, A yapısının 1, 7 ve 8 pozisyonlarında yer alan; 2, 1 ve 3 değerleri sırasıyla 6, 4 ve 5 değerleriyle değiştirilecektir. B yapısında ise 6, 7, ve 8 pozisyonlarındaki 4, 6 ve 5 değerleri, sırasıyla 1, 2 ve 3 tarafından değiştirilecektir. Yeni yapı ise, aşağıdaki gibi oluşmaktadır:

A = 6 8 2 1 3 7 4 5

B = 8 7 6 4 5 1 2 3

### 6.4.3 Mutasyon

Canlılarda “gen yeniden kombinasyonları” dışındaki diğer nedenlerle ve ani olarak meydana gelen kalıtsal değişimlere “mutasyon” denmektedir. Mutasyon işlemi esnasında kromozomdaki gen sayısı değişmemekte, sabit kalmaktadır. Mutasyon frekansının büyüklüğü GA'nın performansını etkilemektedir [122].

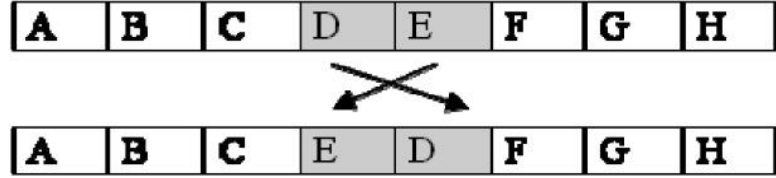
Akış tipi iş sıralama örnekleri literatüründe kullanılan, beş ayrı mutasyon operatörünün test edilmesi sonucu, en uygun olanın “Keyfi üç iş değiştirme mutasyon yöntemi” olduğu söylenmektedir [123] [87].

#### 6.4.3.1 Ters mutasyon

Bu yöntemde, bir kromozomda rassal olarak iki pozisyon seçilmekte ve bu iki pozisyonadaki alt diziler ters çevrilmektedir [123].

#### 6.4.3.2 Komşu iki gen deęiştirme

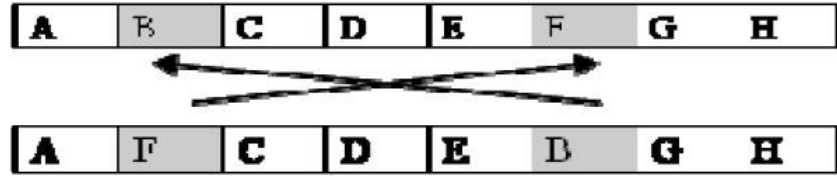
Komşu iki gen deęiştirme yönteminde, rassal olarak seçilen iki komşu gen deęiştirilmektedir [123]. Şekil 6.6'da komşu iki gen deęiştirme mutasyon işlemi gösterilmektedir.



Şekil 6.6 Komşu iki geni deęiştirme

#### 6.4.3.3 Keyfi iki gen deęiştirme

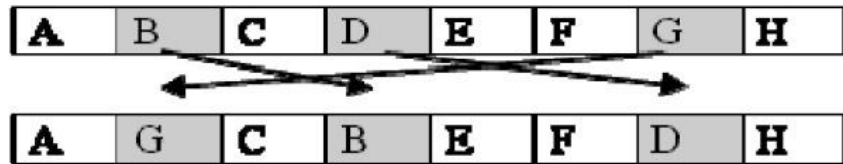
Keyfi iki geni deęiştirme mutasyon yönteminde, rassal seçilen iki gen deęiştirilmektedir [123]. Şekil 6.7'de keyfi olarak iki genin deęiştirildięi mutasyon işlemi gösterilmiştir.



Şekil 6.7 Keyfi iki geni deęiştirme

#### 6.4.3.4 Keyfi üç gen deęiştirme

Burada, rassal olarak seçilen üç gen keyfi olarak deęiştirilmektedir [123]. Şekil da keyfi üç geni deęiştirme mutasyon işlemi gösterilmektedir.



Şekil 6.8 Keyfi üç geni deęiştirme

#### 6.4.3.5 Araya yerleştirme

Bu mutasyon işleminde, bir operasyondaki bir gen kaydırılır ve dięer bir pozisyona konulur. Kaydırma noktası rassal olarak seçilmektedir [123].

#### 6.4.4 Uygunluk Fonksiyonu

Uygunluk fonksiyonu tanım olarak, kromozomların çözümde gösterdikleri başarı derecesini belirleyen bir değerlendirme işlevidir. Hangi kromozomların bir sonraki nesle aktarılacağına ve hangi kromozomların yok olacağına, uygunluk değerlerinin büyüklüğü ile karar verilir. Zaten Genetik Algoritma' da kullanılan değerlendirme işlevi/uygunluk fonksiyonu, problemin amaç işlevini oluşturmaktadır. Uygunluk fonksiyonun formül olarak gösterimi ise genel olarak şöyledir [124]:

Tüm kısıtlar için bir popülasyondaki ceza puanları toplamı:

$$\sum_{k=0}^t \sum_{i=0}^n \sum_{j=0}^l P_k * C_{ij} \quad (6.1)$$

Uygunluk fonksiyonu:

$$(f) = \frac{1}{1 + (\sum_{k=0}^t \sum_{i=0}^n \sum_{j=0}^l P_k * C_{ij})} \quad (6.2)$$

l= kromozomdaki gen sayısı,

n= popülasyondaki kromozom sayısı,

t= kısıt sayısı,

C<sub>ij</sub> = i. Kromozomdaki ceza puanına sahip j. gen,

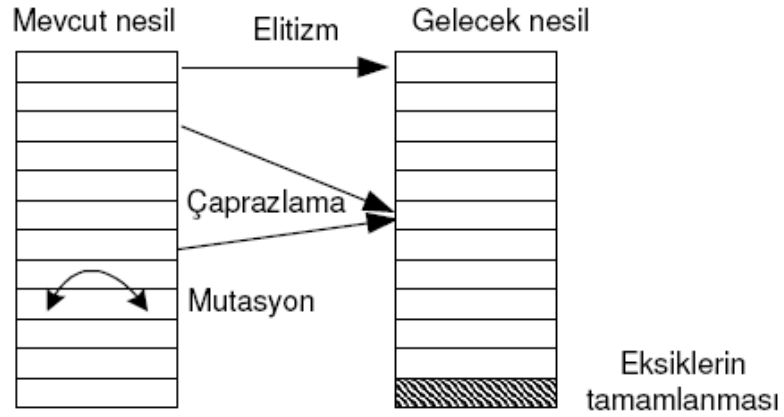
P<sub>k</sub> = k. kısıta ait ceza puanı.

#### 6.4.5 Üretim Operatörü

Üretim operatörü, mevcut yığından bir sonraki nesle aktarılacak olan dizilerin seçilme işlemidir. Taşınan diziler, genetik olarak mevcut nesilde (yığında) en uygun yapıya sahip olan diziler olacaklardır. Bu işlem, belirlenen uygunluk değerlerine sahip "iyi" bireylerin bir sonraki nesle aktarılmasını sağlamaktadır.

Sıklıkla, bu seçimi yapacak olan en basit ve en kullanışlı seçim mekanizması rulet tekerleği seçimi olmaktadır. Her nesilde en iyi bireyin saklanarak, bir sonraki nesle aktarılmasıyla bir sonraki neslin kötüye gitmesi engellenmektedir. Bu işleme "elitizm" denmekte ve üreme operatörlerinin sonucunda mevcut yığında kötüleşme olsa bile, en iyi bireyin bir sonraki nesle aktarımı sağlanmaktadır [124].





Şekil 6.9 Üretim operatörü

#### 6.4.6 Kodlama

Her bir kromozom bir diziden oluşur. Bu dizideki her bir bit, çözümün bir karakterine sahip olabilir [126]. Çözümlerin kodlanması, probleme özgü bilgilerin genetik algoritmanın kullanacağı şekle çevrilmesidir. Her problem kendine özgü farklı kodlamalara ihtiyaç duyabilir. Tüm problemler için geçerli olabilecek genel bir kodlama tekniği yoktur [125].

##### 6.4.6.1 İkili sayı sisteminde kodlama

İkili sayı sisteminde (Binary) kodlama, en sık kullanılan kodlama çeşididir. Binary kodlama' da kromozomlar 1 ve 0 sayılarından oluşur [126].

Çizelge 6.1 İkili sayı sisteminde kodlanmış örnek kromozomlar

Kromozom 1	1101100100110110
Kromozom 2	1101111000011110

##### 6.4.6.2 Permütasyon kodlama

Permütasyon kodlama çeşidi, gezgin satıcı problemlerinde kullanılabilir. Permütasyon kodlamasında, her kromozom, her biri art arda gelen sayı dizisinin bir elemanı olan çok basamaklı bir sayı ile gösterilmektedir. Bu tip kodlama, sadece sıralama problemlerinin çözümü için kullanılabilir. Bazı çaprazlama ve

mutasyon tipleri için düzeltmeler, kromozomları uygun hale getirecek şekilde yapılmak zorundadır [126].

Çizelge 6.2 Permütasyon kodlama sistemi ile kodlanmış örnek kromozomlar

Kromozom A	1 5 3 2 6 4 7 9 8
Kromozom B	8 5 6 7 2 3 1 4 9

#### 6.4.6.3 Değer kodlama

Reel sayılar gibi bazı karmaşık değerlerin kullanıldığı durumda, problemlerin çözümünde direkt değer şifrelemesi kullanılabilir. Bu tip problemlerde “0-1” kodlamasının kullanılması oldukça zor olmaktadır.

Değer kodlaması işleminde, her kromozom, değerlerin oluşturduğu bir dizi olarak düşünülmektedir. Değerler, probleme bağlı olarak reel sayılardan, karmaşık objelere kadar birçok şey olabilmektedir. Değer kodlaması, bazı özel problemler için çok uygundur. Ayrıca, bu kodlama işleminde, probleme özgü bazı çaprazlama ve mutasyonlar geliştirilmelidir [124].

Çizelge 6.3 Değer kodlama sistemi ile kodlanmış örnek kromozomlar

Kromozom A	1.2324 5.3243 0.4556 2.3293 2.4545
Kromozom B	ABDJEIFJDHDIERJFDLDFLFEGT
Kromozom C	(siyah), (arka), (sağ), (ön), (sol)

#### 6.4.6.4 Ağaç Kodlama

Bu yöntem, genetik programlama gibi değişen/gelişen programlar ve/veya ifadeler için kullanılmaktadır. Ağaç kodlama işleminde, her kromozom bazı nesnelerin, mesela fonksiyonlar ya da programlama dilindeki komutlar gibi, bir ağacı olmaktadır [126].



Şekil 6.10 Ağaç kodlamalı kromozomlar örneği

Özetle, bu yöntemde her bir dizi nesnelere oluşan bir “ağaç yapısı” şeklinde ifade edilmektedir. Bu nesnelere fonksiyonlar ya da programlama dillerinde geçen komutlar olabilmektedir. Bu yöntem, verilen değerlere uygun bir fonksiyon bulmak için kullanılabilir [125].

#### 6.4.7 Seçim Operatörü

Bir nesildeki kromozomlardan bir kısmı bir sonraki nesle aktarılırken bir kısmı yok olabilmektedir. İşte bu aşamada, hangi kromozomların bir sonraki nesle aktarılacağı sorunsalı, kurulan seçim mekanizmaları ile çözülmektedir.

Kromozomların eşlenmesi, kromozomların uygunluk değerlerine göre yapılır. Bu seçimi yapmak için Rulet Çarkı (Rulet tekeri-Roulette Wheel), Turnuva Seçimi (tournament selection), Düzenli Durum Seçimi ve Elitizm Seçimi gibi seçim yöntemleri bulunmaktadır.

Burada asıl amaç, uygunlukları küçük olan kromozomların elenip, uygunlukları büyük olan kromozomların ile yeni bir nesil oluşturmaktır. Seçim işlemi, uygunlukları büyük kromozomların uygunlukları küçük kromozomlarınkiler üzerine tekrar yazılarak yapılmaktadır. Kromozomların karakterleri, sahip oldukları uygunluk değerlerine göre

saptanmaktadır. Seçim işlemi, çeşitli kriterlere göre yapılabilmektedir (Demir ve Karcı, 2013) [127].

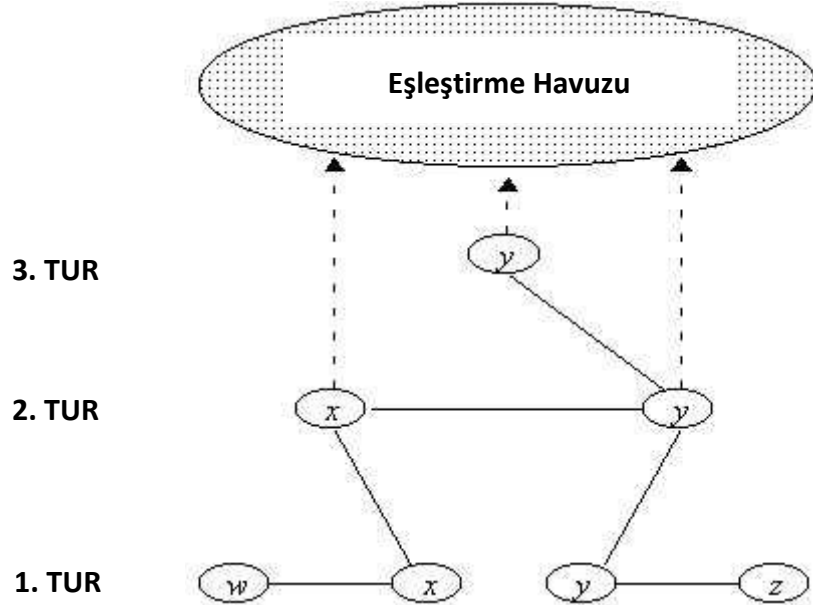
#### **6.4.7.1 Rulet Çarkı Seçimi**

Bu yöntemde her dizi uyum değeriyle orantılı bir olasılık değeri ile seçilmektedir. Rulet tekerleğinin yüzeyi dizilerin uyum değerleri ile orantılı olarak işaretlenir. Tekerlek kaç defa döndürülürse, her seferinde bir dizi eşleme havuzuna atılacaktır. Daha iyi uyum değerine sahip diziler, tekerlekte daha fazla yer aldıklarından onların seçilme şansları daha yüksektir [125].

Uygunluk fonksiyonu, her hangi bir kritere uyan bireylerin seçilmesi için kullanılmaktadır. Buradaki çalışma prensibi, bu çarkın döndürülerek herhangi bir dilimde durmasını beklemektir. Doğal olarak okun, en fazla dilime sahip olan kromozomu gösterme olasılığı en büyüktür.

#### **6.4.7.2 Turnuva Seçimi**

Tüm nesil içinden her seferinde rastgele dizi seçilir. Bu diziler ise, turnuvaya katılmış olur. Bu diziler arasında yapılan turnuvada, hangisi daha iyiyse o kazanacaktır. Yani, bu yöntemde popülasyondan rassal olarak belirli sayıda birey seçilip, kendi aralarında uygunluklarına göre turnuvaya sokulmaktadır [128].



Şekil 6.11 Turnuva Seçimi yöntemi

#### 6.4.7.3 Kararlı durum seçimi

Bu seçim, anne baba belirlenmesinde özel bir metot değildir. “Kromozomların büyük bir bölümü bir sonraki nesilde hayatını devam ettirecektir” düşüncesi, kararlı durum seçim yönteminin ana fikrini oluşturmaktadır. Bu seçimde öncelikle, her bir nesilde uyumlulukları yüksek olan bazı kromozomlar, çocukları oluşturmak için belirlenirler. Daha sonra uyumluluk değerleri düşük olan kromozomlar atılır. Kalan bireyler yeni nesli devam ettireceklerdir [129].

#### 6.4.7.4 Elitizm (Seçkinlik) seçimi

Çaprazlama ve mutasyon ile yeni nesil oluşturulduğunda en iyi kromozomun kaybedileceği büyük bir değişim meydana gelmiştir. Elitizm adı verilen bu metot ile, ilk olarak yeni nesil için en iyi kromozomun bir kopyası alınacaktır. Geriye kalan adımlar ise, diğer metotlara benzer şekilde gerçekleştirilecektir. Kaybolan en iyi çözümü koruduğu için, bu metot genetik algoritmanın performansını çok hızlı bir şekilde arttırabilmektedir [129].

### 6.5 Genetik Algoritmada Parametre Seçimi

Genetik algoritmanın performansını, üreme, çaprazlama, mutasyon operatörleri ve çaprazlama/mutasyon oranları önemli ölçüde etkilemektedir [117]. Akış tipi

çizelgeleme problemleri için genetik algoritma çözümlerinde, kısa işlem süreleriyle ve optimum/optimuma yakın çözümlere ulaşmak için bu parametrelerin seçimi ile ilgili herhangi bir kural bulunmamaktadır. Bir problem türü için kullanılan ve optimum veya optimuma yakın çözüm veren bir kontrol parametresi seti, başka bir GA uygulaması için genelleştirilememektedir [130].

Bu nedenlerden dolayı optimum çözümü bulunamayan ( $m>2$ ) akış tipi çizelgeleme problemlerinin GA ile çözüm performansının artırılması için, en etkin çaprazlama yönteminin belirlenmesi, çözüm süresini kısaltmak ve daha iyi (optimum veya optimuma yakın) çözümler elde edilebilmesi açısından önem taşımaktadır [122].

GA'ların parametreleri; çaprazlama oranı, mutasyon oranı, popülasyon büyüklüğü, seçim, kodlama (encoding), çaprazlama ve mutasyon tipi gibi genel parametreler olmaktadır. Aşağıda "parametre seçimi" içerisinde üzerinde durulan kısımlar verilmektedir.

#### **6.5.1 Popülasyon Büyüklüğü**

Genetik algoritma kullanıcısı tarafından verilen en önemli kararlardan birisi, popülasyon büyüklüğüdür. Bu değer çok küçük olduğunda, genetik algoritma yerel bir optimuma takılabilmektedir. Çok büyük olduğunda ise, çözüme ulaşma zamanı artmaktadır.

Bu konuda Goldberg 1985 [131]'de, yalnızca kromozom uzunluğuna bağlı bir popülasyon büyüklüğü hesaplama yöntemi önermiştir. Ayrıca Schaffer ve arkadaşları 1989 [132]'da çok sayıda test fonksiyonları üzerinde yaptıkları araştırmalar sonucunda, 20-30 arası bir popülasyon büyüklüğünün iyi sonuçlar verdiğini belirtmişlerdir.

#### **6.5.2 Çaprazlama Oranı**

Çaprazlama işleminin amacı, mevcut iyi kromozomların özelliklerini birleştirerek daha uygun kromozomlar yaratmaktır. Kromozom çiftleri  $P(c)$  olasılığı ile çaprazlamaya

uğramak üzere seçilirler. Çaprazlamanın artması, bazı iyi kromozomların da bozulma olasılığını arttırabilmektedir.

Çaprazlama oranı, döngü sayısı ile çarpılarak, çaprazlama sayısının bulunmasına yardım etmektedir:

$$\text{Çaprazlama sayısı} = \text{Döngü Sayısı} * \text{Çaprazlama Oranı} \quad (6.3)$$

### 6.5.3 Mutasyon Oranı

Mutasyonun amacı, genetik popülasyondaki genetik çeşitliliğin korunmasıdır. Eğer mutasyon oranı artarsa, genetik arama rastsal bir aramaya dönüşür. Fakat bu aynı zamanda kayıp genetik kromozomların da tekrar bulunmasına yardım edecektir.

Çaprazlama sayısında çaprazlama oranı gibi, mutasyon oranı da, döngü sayısı ile çarpılarak, mutasyon sayısının bulunmasına yardım etmektedir:

$$\text{Mutasyon sayısı} = \text{Döngü Sayısı} * \text{Mutasyon Oranı} \quad (6.4)$$

### 6.5.4 Seçim Stratejisi

Eski kuşağı yenilemenin çeşitli yöntemleri bulunmaktadır. Kuşaklara bağlı bir evrimsel stratejide, mevcut popülasyondaki kromozomlar tamamen yavrular ile yer değiştirebilmektedir. Seçim işlemi için genellikle, rulet tekerleği, turnuva seçimi gibi yöntemler kullanılmaktadır. Bunun yan sıra rank seçimi, kararlı durum (steady state) ve elitizm gibi seçim yöntemleri de kullanılabilir.

### KURULUM ZAMANLI SIRALAMA PROBLEMLERİ İÇİN ÜSTÜNLÜK ÖZELLİKLERİ YARATIMI

Verilen bir sıralama için “baskınlık özellikleri (Dominance Properties-DP)” kullanılarak, optimale yakın bir çözüm elde edilebilmektedir. Büyük boyutlu problemlerin çözümü zordur. Bunun için, çözüm kalitesini arttırmaya yardımcı olarak, baskınlık özelliklerinin genetik algoritma ile birleştirildiği, yeni bir meta sezgisel ortaya konulabilir.

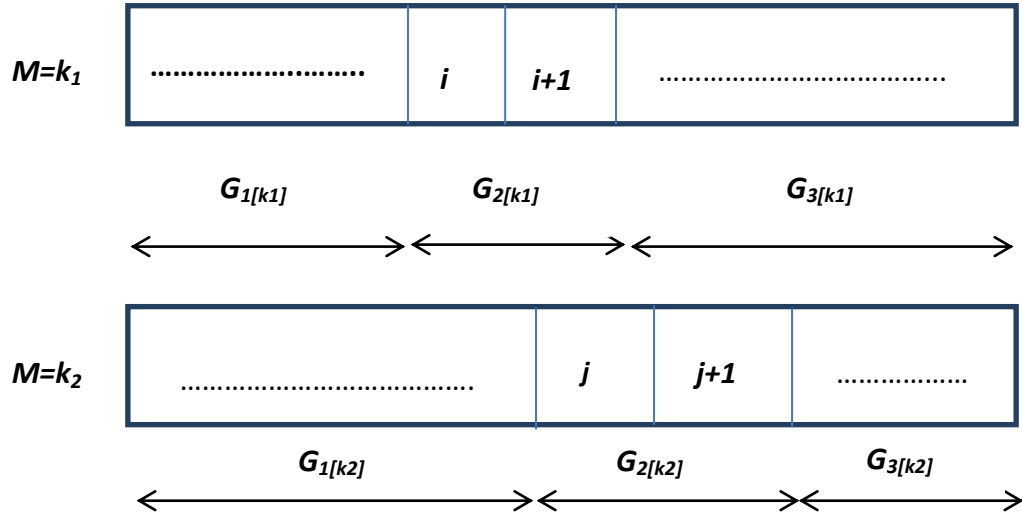
İlişkisiz paralel makine (unrelated paralel machine) sıralama problemi, sıfır anında  $m$  adet ilişkisiz paralel makinede ( $R_m$ ),  $C_{max}$ 'ı yani “tamamlanma zamanı”nı minimize etmek için, mümkün görülen şekilde,  $i$  işinin sıralanması problemi. Eğer işlerin proses zamanları, makine atamalarına bağımlı ve makineler arasında herhangi bir ilişki yok ise, bu makineler “ilişkisiz paralel makine” olarak düşünülmektedir.

Chang ve Chen (2011) [133] makalelerinde, “genetik algoritma” gibi meta sezgisellerin sıralama problemlerine uyumunun yavaş olmasından bahsederler. Bu uyumu geliştirmek ve ilerletmek için ise, üstünlük özelliklerinin (dominance properties) meta sezgisellere eklenmesi önemli olmuştur.

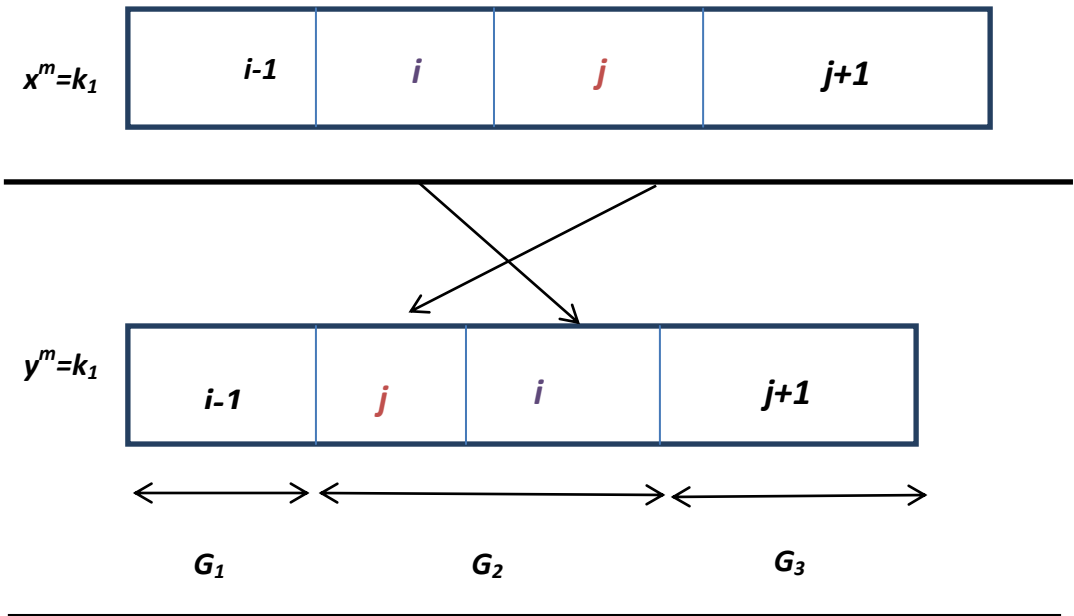
Optimal sıralamanın üstünlük özellikleri, iki bitişik işin,  $i$  ve  $j$  nin yer değişimi temeline dayanmaktadır. Üstünlük özellikleri, meta sezgiseller uygulanmadan önce, “ilk” verimli çözümleri elde ederler. Böylelikle, DP “ilk iyi” çözümleri ürettiği zaman, metasezgiseller optimuma daha hızlı yakınsayabilecektir. Elbette DP çalışması esnasında, “ek



hesaplama" çabasına ihtiyaç duyacaktır. Bu da ek bir zaman anlamına gelmektedir. Ancak unutulmamalıdır ki, "Üstünlük Özellikleri", Genetik Algoritma ve benzeri sezgisel yöntemlere, çözüme "daha hızlı" yakınsaması adına yardım etmektedir.



Şekil 7.1'de bir paralel makine gösterimi görülmektedir. Şekil 7.2'de ise Aynı makine üzerindeki değişim gözükmemektedir.



Burada optimal bir sıralama için, makine içi (intra machine) ve makineler arası (inter machine) iş değişimlerinin özellikleri düşünülerek üstünlük özellikleri (DP) geliştirilmiştir.

### 7.1 Üstünlük Özelliklerinin Türetilmesi ve Kanıtları

Bu kısımda, üstünlük özelliklerinin nasıl türetildiği, kanıtları ile beraber verilecektir.  $\Pi$  çizelgesi tamamlanma zamanı için objektif fonksiyonu ( $Z(\Pi)$ ) olarak alınmaktadır. Optimal sıralamanın üstünlük özelliklerini sağlayan ve ilişkisiz paralel makineli  $n$  iş sıralama problemi ayrıca incelenecektir.  $\Pi$  çizelgesi için üstünlük özelliklerini sağlamak adına, aynı makinede iki işin veya farklı makinelerde iki işin değişimi, bazı ara sonuçları geliştirmek adına düşünülmektedir.

Notasyonlar şöyle olacaktır;

$[j]$ :  $[j]$  konumundaki iştir.

$P_{[j][k]}$ :  $[k]$  makinesinde  $[j]$  pozisyonunda ki işin proses zamanıdır.

$S_{[i][j][k]}$ :  $k$  makinesinde,  $i$  işinden sonra  $j$  pozisyonundaki işin kurulum zamanı.

$AP_{[i][j][k]}$ :  $k$  makinesinde  $i$  işinden sonra  $j$  pozisyonundaki işin düzeltilmiş proses zamanı (adjusted).

Böylece  $AP_{[i][j][k]} = P_{[j][k]} + S_{[i][j][k]}$  'dır.

$C_{k1}$ :  $k1$  deki tamamlanma zamanı.

$G_{1k}$ :  $k$  makinesinde  $i$  işinden önceki iş dizisi.

$G_{2k}$ :  $k$  makinesinde  $[i]$  işi ve  $[j+1]$  işleri arasındaki iş dizisi.

$G_{3k}$ :  $k$  makinesinde  $[i]$  işinden sonraki iş dizisi.

Daha önce de belirtildiği gibi, iş değişimleri ile alakalı 2 şart bulunmaktadır; bunlar makineler arası ve makineler içindeki değişimlerdir.

#### 7.1.1 Makineler İçi Değişim

Makineler arası değişim ile ilgili olarak düşünülen 2 durum söz konusudur; bitişik değişim (adjacent exchange) ve bitişik olmayan değişim (non- adjacent exchange). Aşağıdaki formülasyon "adjacent exchange" ile ilgili olmaktadır:

## **BITİŞİK OLAN DEĞİŞİM**

### **FORMÜL 1a.**

Aşağıdaki şart sağlanıyorsa, değiştirilen çizelge asıl çizelgeden daha iyidir;

$$(AP_{[i-1][j][k]} - AP_{[i-1][i][k]}) + (AP_{[j][i][k]} - AP_{[i][j][k]}) + (AP_{[i][j+1][k]} - AP_{[j][j+1][k]}) < 0$$

### **Kanıt 1a.**

$\Pi_x$  çizelgesinde  $i$  işi ile  $j$  işinin yerlerini değiştirelim.

Bu işler aynı makine üzerinde birbirlerine bitişik sıralanmaktadır. Değişimin ardından, bu iki komşu iş, yani  $\Pi_x$  çizelgesi,  $\Pi_y'$  ye dönüşür.

$G_{1[k]}$  ve  $G_{1[k]}$ 'nin tamamlanma zamanları;

$$G_{1[k]} = \sum_{a=1}^{i-1} AP_{[a-1][a][k]} = G'_{1[k]}$$

$$G_{2[k]} = G_{1[k]} + AP_{[i-1][i][k]} + AP_{[i][j][k]} + AP_{[j][j+1][k]}$$

$$G_{3[k]} = G_{2[k]} + \sum_{a=j+2}^n AP_{[a-1][a][k]}$$

$$G'_{2[k]} = G_{1[k]} + AP_{[i-1][j][k]} + AP_{[j][i][k]} + AP_{[i][j+1][k]}$$

$$G'_{3[k]} = G'_{2[k]} + \sum_{a=j+2}^n AP_{[a-1][a][k]}$$

$\Delta$  farklılık değeri;

$$\Delta = \Pi_y - \Pi_x = G_{2[k]}' - G_{2[k]}$$

$$= (AP_{[i-1][j][k]} - AP_{[i-1][i][k]}) + (AP_{[j][i][k]} - AP_{[i][j][k]}) + (AP_{[i][j+1][k]} - AP_{[j][j+1][k]})$$

Sonuç olarak; eğer  $\Delta$  sıfırdan küçük ise,  $i$  işi ve  $j$  işi yer değiştirir. Bitişik olmayan değişimde de metodoloji aynıdır.

## **BITİŞİK OLMAYAN DEĞİŞİM**

### **FORMÜL 1b.**

Aşağıdaki durumlar sağlanırsa,  $i$  işi ve  $j$  işi yer değiştirecektir.

$$(AP_{[i-1][j][k]} - AP_{[i-1][i][k]}) + (AP_{[j][i+1][k]} - AP_{[i][i+1][k]}) + (AP_{[j-1][i][k]} - AP_{[j-1][j][k]}) \\ + (AP_{[i][j+1][k]} - AP_{[j][j+1][k]}) < 0$$

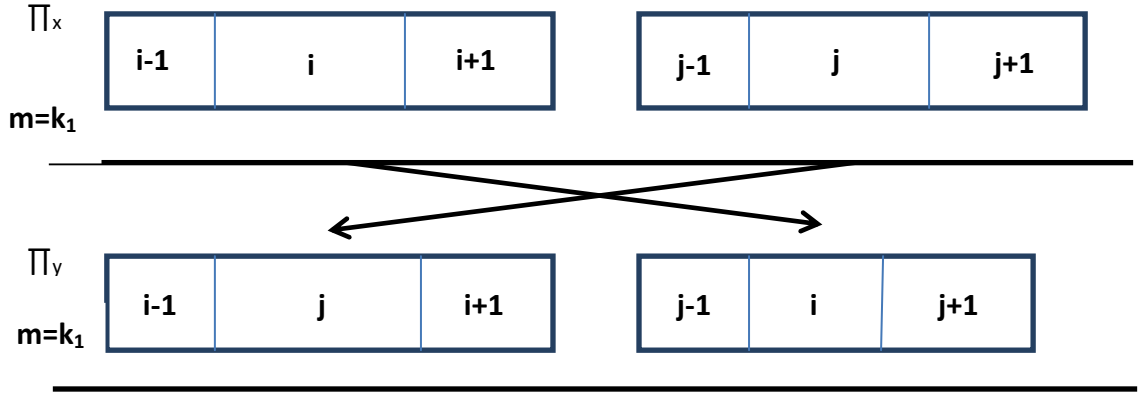
$$G_{1[k]} = \sum_{a=1}^{i-1} AP_{[a-1][a][k]} = G'_{1[k]}$$

$$G_{2[k]} = G_{1[k]} + AP_{[i-1][i][k]} + AP_{[i][i+1][k]} + AP_{[j-1][j][k]} + AP_{[j][j+1][k]}$$

$$G_{3[k]} = G_{2[k]} + \sum_{a=j+2}^n AP_{[a-1][a][k]}$$

$$G'_{2[k]} = G_{1[k]} + AP_{[i-1][j][k]} + AP_{[j][i+1][k]} + AP_{[j-1][i][k]} + AP_{[i][j+1][k]}$$

$$G'_{3[k]} = G'_{2[k]} + \sum_{a=j+2}^n AP_{[a-1][a][k]}$$



Şekil 7.3 Bitişik olmayan işlerin değişimi

$\Delta$  farklılık değeri ise şöyledir;

$$\begin{aligned} \Delta &= \Pi_y - \Pi_x = G_{2[k]}' - G_{2[k]} \\ &= (AP_{[i-1][j][k]} - AP_{[i-1][i][k]}) + (AP_{[j][i+1][k]} - AP_{[i][i+1][k]}) + (AP_{[j-1][i][k]} - AP_{[j-1][j][k]}) + (AP_{[i][j+1][k]} - AP_{[j][j+1][k]}). \end{aligned}$$

$\Delta$  sıfırdan küçük ise yer değiştirme yapılmaktadır.

### 7.1.2 Makineler Arası Değişim

Makineler arası değişim (Intra-machine exchanging) kısmı, her hangi iki farklı makinede bulunan  $i$  ve  $j$  işlerinin birbiri arasındaki değişimini tartışmaktadır. Paralel makine sayıları 2'ye eşit ya da daha fazladır,  $k_1$  ve  $k_2$  gösterimleri  $i$  işi ve  $j$  işi için makine sayılarını göstermektedir.

#### **FORMÜL 2.**

$$\text{Max} \{ (G_{3[k_1]} + (AP_{[i-1][j][k_1]} - AP_{[i-1][i][k_1]}) + (AP_{[j][i+1][k_1]} - AP_{[i][i+1][k_1]}), G_{3[k_2]} + (AP_{[j-1][i][k_2]} - AP_{[j-1][j][k_2]}) + (AP_{[i][j+1][k_2]} - AP_{[j][j+1][k_2]}) \} < C_{\max}$$

### **Kanıt.**

Orijinal sıralamanın  $C_{\max}'$  ı ile,  $\text{Max} (G_{3[k_1]}, G_{3[k_2]})$ , ile değiştirilen sıralamanın  $C_{\max}'$  ı arasında bir kıyaslama söz konusudur. Eğer  $C_{\max}' < C_{\max}$  tan az ise, i ve j işi yer değiştirir. İki makineye ait olan amaç ilişkisinin doğruluğunu kanıtlamak adına,  $k_1$  in amaç farklılığı  $k_2$  den önce hesaplanır.

### **$k_1$ ' in amaç farklılığı**

$k_1$  deki her iş dizisinin tamamlanma zamanı şöyledir;

$$G_{1[k_1]} = \sum_{a=1}^{i-1} AP_{[a-1][a][k_1]} = G'_{1[k_1]}$$

$$G_{2[k_1]} = G_{1[k_1]} + AP_{[i-1][i][k_1]} + AP_{[i][i+1][k_1]}$$

$$G_{3[k_1]} = G_{2[k_1]} + \sum_{a=j+2}^n AP_{[a-1][a][k_1]}$$

$$G'_{2[k_1]} = G'_{1[k_1]} + AP_{[i-1][j][k_1]} + AP_{[j][i+1][k_1]}$$

$$G'_{3[k_1]} = G'_{2[k_1]} + \sum_{a=j+2}^n AP_{[a-1][a][k_1]}$$

Buna göre, makine  $k_1$  in amaç farklılığı şöyledir;

$$\Delta_{k_1} = \prod'_x - \prod_x = (AP_{[i-1][j][k_1]} - AP_{[i-1][i][k_1]}) + (AP_{[j][i+1][k_1]} - AP_{[i][i+1][k_1]})$$

### **$k_2$ ' nin amaç farklılığı**

Makine  $k_2$ ' deki tamamlanma zamanı şöyledir;

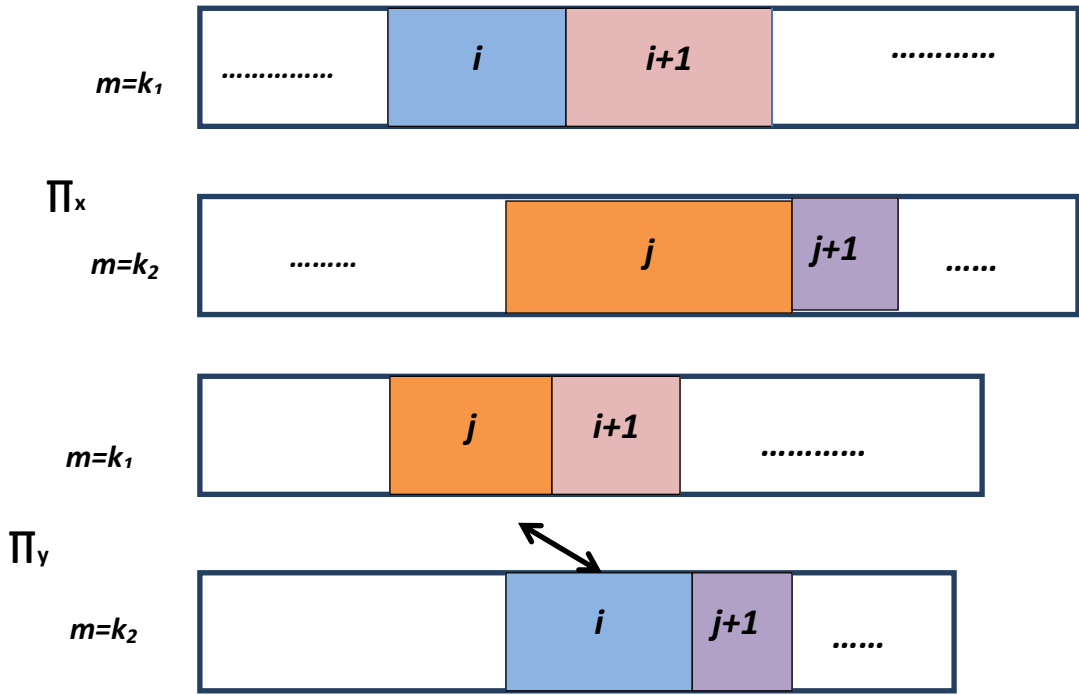
$$G_{1[k_2]} = \sum_{a=1}^{i-1} AP_{[a-1][a][k_2]} = G'_{1[k_2]}$$

$$G_{2[k_2]} = G_{1[k_2]} + AP_{[j-1][j][k_2]} + AP_{[j][j+1][k_2]}$$

$$G_{3[k_2]} = G_{2[k_2]} + \sum_{a=j+2}^n AP_{[a-1][a][k_2]}$$

$$G'_{2[k_2]} = G'_{1[k_2]} + AP_{[j-1][i][k_2]} + AP_{[i][j+1][k_2]}$$

$$G'_{3[k_2]} = G'_{2[k_2]} + \sum_{a=j+2}^n AP_{[a-1][a][k_2]}$$



Şekil 7.4 Farklı makinelerdeki i ve j işlerinin yer değişimi

Amaç farklılığını  $\Pi'_y$  yi  $\Pi_y$  den çıkararak bulabiliriz.

$$\Delta_{k1} = \Pi'_y - \Pi_y = (AP_{[j-1][i][k2]} - AP_{[j-1][j][k2]}) + (AP_{[j][j+1][k2]} - AP_{[j][j+1][k2]})$$

Sonuç olarak, değişen sıralamanın  $C'_{\max}$  değeri,  $\text{Max}(G_{3[k1]} + \Delta_{k1}, G_{3[k2]} + \Delta_{k2})$  olmaktadır. Bu nedenle, eğer  $C'_{\max}$ ,  $C_{\max}$  dan daha az ise, i ve j işi yer değiştirir.

## 7.2 Üstünlük Özellikleri Adımları ve Bu Konuda Yapılan Bazı Örnekler

Bu kısımda, üstünlük özellikleri hakkında yapılan örneklendirmeler prosesin adımları üzerinden gidilerek verilecektir.

### 7.2.1 Tek Makineli Örnek

Tek makineli problemler için verilen algoritma adımları şöyledir:

**Adım 1:** İşlerin “proses” zamanları ile “hazırlık” zamanlarını toplayarak AP’leri bul.

$$AP_{ijk} = P_{jk} + S_{ijk}$$

**Adım 2:** AP’leri her makine için sıralayarak tabloya dök. Bizim örneğimizde AP’ler Excel dosyasından hesaplanacaktır. Burada,

Çizelge 7.1 Makine 1 için AP' ler

AP1	1	2	3	4	5	6
0	116	142	130	109	157	152
1	-	167	166	122	179	141
2	127	-	120	145	170	180
3	143	165	-	150	143	145
4	124	165	157	-	173	137
5	118	162	158	108	-	137
6	132	170	136	151	181	-

Mesela AP<sub>25</sub> dediğimiz zaman, 5 işi için 2 işinden sonra, Makine 1' deki ayarlanmış proses zamanı verilmektedir. Bunun değeri, 170' dir.

**Adım 3:** 1. Sıradaki bütün işlerin AP'lerini sırala. Minimum değerini bul. Makine 1' e atanan ilk işi göster.

**Adım 4:** Makine 2' deki ilk işleri sıralayıp, minimum değerini bul. Ardından bu değeri Makine 2' ye atanan ilk iş olarak göster.

**Adım 5:** Makine 3' deki ilk işleri sıralayıp, minimum değerini bul. Ardından bu değeri Makine 3' e atanan ilk iş olarak göster.

**Adım 6:** Makine 4' deki ilk işleri sıralayıp, minimum değerini bul. Ardından bu değeri Makine 4' e atanan ilk iş olarak göster.

**Adım 7:** Tekrardan Makine 1' deki ikinci iş, 2.gelen minimum değer olacaktır. (Misal, bu örnekte olduğu gibi, Makine 1' de 109' dan sonra gelen 116 değeri 2.en küçük değerdir. Bu da Makine 1' e atanan 2.iş olacaktır.)

Makine 2' deki ikinci iş ise, Makine 2' nin 0.kısımındaki 2.gelen minimum değer olacaktır. Aynı durumlar Makine 3 ve Makine 4 için de geçerli olacaktır.

**Adım 8:** Makine 1' deki 3. iş minimum gelen 3. Değer olacaktır (Misal yukarıdaki tabloda minimum gelen 3. iş 130 değeridir. Bu da 3. iş demektir. 3.kısma 3. iş atanmaktadır.)

Makine 2' deki üçüncü iş ise, Makine 2' nin 0. kısmındaki 3. gelen minimum değer olacaktır. Aynı durumlar Makine 3 ve Makine 4 için de geçerli olacaktır.

**Adım 9:** Makine 1' deki 4. iş minimum gelen 4. Değer olacaktır. Makine 2' deki 4. iş ise, Makine 2' nin 0. kısmındaki 4. gelen minimum değer olacaktır. Aynı durumlar Makine 3 ve Makine 4 için de geçerli olacaktır.

**Adım 10:** Makine 1' deki 5. iş minimum gelen 5. Değer olacaktır. Makine 2' deki 5. iş ise, Makine 2' nin 0. kısmındaki 5. gelen minimum değer olacaktır. Aynı durumlar Makine 3 ve Makine 4 için de geçerli olacaktır.

**Adım 11:** Bu şekilde, Makine 1, Makine 2, Makine 3 ve Makine 4 için 5' er iş sıralanacaktır. Makine 1, Makine 2, Makine 3 ve Makine 4 için Maksimum Tamamlanma Zamanları hesaplanır.

**Adım 12:** Üstünlük Özelliği (Dominance Properties) Kuralı uygulanmaya başlanır. Aynı makine üzerinde değişim yapılır. **Her makine için ayrı ayrı  $\Delta$  değerleri bulunur. Buna göre;**

- Eğer i ve j işleri komşu işler ise;

Örneğin; [4,1,3] Makine 1 deki sıralamamız olsun. 4 ve 1 numaralı işler bitişik işlerdir. Bu işler değiştirildiği takdirde yeni sıralama [1,4,3] olacaktır. Her iki sıralama içinde AP değerleri Tablodaki değerlerden bulunacaktır.

$$\Delta = \begin{pmatrix} 109 & 124 & 166 \\ 116 & 122 & 157 \end{pmatrix} = (116-109)+(122-124)+(157-166) = -4$$

-4<0 olmaktadır. Dolayısı ile değişim yapılacaktır. Yeni sıralama; [1,4,3] olacaktır.

$C_{m1} = 116 + 122 + 157 = 395$  olmaktadır.



$C_{max} = 411$  idi.

[1,4,3] sıralaması alınarak, buradaki 4 ve 3 numaralı işler değiş tokuş kuralına sokulur.

[1,3,4] olacaktır. Bunların arasındaki  $\Delta = \begin{pmatrix} 116 & 122 & 157 \\ 116 & 166 & 150 \end{pmatrix}$  değeri hesaplanır.

$(116-116)+(166-122)+(150-157) = 37$  değeri 0'dan büyüktür.

Dolayısı ile 4 ve 3 işleri değiştirilmeyecektir.

Bu kural bütün makineler üzerinde tek tek bütün işlere uygulanacaktır.

- Eğer i ve j işleri komşu olmayan işler ise;

Örneğin; [1,4,3] Makine 1 deki sıralamamız olsun. 1 ve 3 numaralı işler bitişik olmayan işlerdir. Bu işler değiştirildiği takdirde yeni sıralama [3,4,1] olacaktır. Her iki sıralama içinde AP değerleri Tablodaki değerlerden bulunacaktır.

$$\Delta = \begin{pmatrix} 116 & 122 & 157 \\ 130 & 150 & 124 \end{pmatrix} = (130-116)+(150-122)+(124-157) = 9 \text{ değeri gelmektedir.}$$

**9 > 0** olmaktadır. Sıralama değiştirilmeyecektir.

Aynı durum Makine 2, Makine 3 ve Makine 4 için de yapılır.

**Adım 13:** Makineler arasında değişim yapılacaktır. Eğer i ve j işi komşu işler değilse;

Misal; Makine 1 sıralaması : [1,4,3]  $C_{m1} = 395$

Makine 2 sıralaması : [5,2,6] olsun.  $C_{m2} = 379$  Buradan  $C_{max} = 395$  olmaktadır.

1 ve 5 işlerinin yerleri değiştirilsin.

Yani  $\Pi_x = \text{Makine 1} = [1,4,3]$  ve  $\text{Makine 2} = [5,2,6]$

$\Pi_y = \text{Makine 1} = [5,4,3]$  ve  $\text{Makine 2} = [1,2,6]$

$$\Delta_{m1} = \begin{pmatrix} 116 & 122 & 157 \\ 157 & 108 & 157 \end{pmatrix} = (157-116) + (108-122) + (157-157) = 27$$

$$\Delta_{m2} = \begin{pmatrix} 127 & 142 & 110 \\ 163 & 135 & 110 \end{pmatrix} = (163-127) + (135-142) + (110-110) = 29$$

$C'_{max} = \text{Max}\{395+27, 379+29\} = 422$  (> 395) dolayısı ile değişim olmaz.

**Aynı işlemi makineler arasında diğer bitişik olmayan işler için de gerçekleştiririz.**

$\Pi x = \text{Makine 1} = [1,4,3]$  ve  $\text{Makine 2} = [5,2,6]$

$\Pi y = \text{Makine 1} = [2,4,3]$  ve  $\text{Makine 2} = [5,1,6]$

444 çıkar.  $C'_{max}$ .  $444 > 395$  olduğundan değişim olmaz.

$\Pi x = \text{Makine 1} = [1,4,3]$  ve  $\text{Makine 2} = [5,2,6]$

$\Pi y = \text{Makine 1} = [6,4,3]$  ve  $\text{Makine 2} = [5,2,1]$

460 çıkar.  $C'_{max}$ .  $460 > 395$  olduğundan değişim olmaz.

**Bütün bu işlemler yapıldıktan sonra en uygun sıralama yazılır. Makine 1 = [1,4,3],**

**$C_{m1}=395$**

**Makine 2 = [5,2,6],  $C_{m1}=379$**

**Adım 14:** Maksimum tamamlanma zamanı kaydedilir. Gantt diyagramına aktarılır.

### 7.2.2 İki Makineli örnek sisteminin incelenmesi

Bu kısımda, iki paralel makineli örnek bir sistem üzerinde üstünlük özellikleri incelenecektir.

Çizelge 7.2 İki Makineli bir paralel makine örneğinde AP Zamanları

AP1	1	2	3	4	5	6
0	116	142	130	109	157	152
1	-	167	166	122	179	141
2	127	-	120	145	170	180
3	143	165	-	150	143	145
4	124	165	157	-	173	137
5	118	162	158	108	-	137
6	132	170	136	151	181	-

AP2	1	2	3	4	5	6
0	163	135	149	136	127	131
1	-	135	171	126	139	156
2	144	-	128	158	165	110
3	15	154	-	145	127	136
4	147	152	159	-	170	140
5	144	142	168	171	-	127
6	166	157	172	172	173	-

Daha önceki kısımlarda anlatıldığı gibi, üstünlük özelliklerini uygulamak için öncelikle, işlem zamanları ve hazırlık zamanları toplanarak, çıkan sayılar AP olarak kaydedilecektir.

Bu örnekte, 6 adet iş ve 2 tane paralel makine bulunmaktadır. Yukarıdaki Çizelge 7.2, Makine1 ve Makine2 için ayrı ayrı “ayarlanmış proses (AP)” zamanlarını göstermektedir. Misal, AP<sub>25</sub> dendiği zaman, AP1 tablosunda, 5 işi için 2 işinden sonra Makine 1’deki “ayarlanmış proses” zamanları verilmektedir. Bunun değeri 170’dir. Örnek için başlangıç sıralaması şöyledir:

Aşama1:  $\text{Min}\{116, 142, 130, \mathbf{109}, 157, 152, 163, 135, 149, 136, 127, 131\} = 109$

Bu durumda, Makine1’de atanan ilk iş 4 işidir.

Aşama2:  $\text{Min}\{163, 135, 149, 136, 127, 131\} = 127$

Bu durumda, Makine2’de atanan ilk iş 5 işidir.

Aşama3: Tekrardan, Makine1’e atanan işlere bakılır. Bunlardan “en az” olan ikinci iş yani 1 işidir. Makine2’de atanan 2. iş ise 6 işidir.

Çizelge 7.3 Makinelere atanan işlerin durumları

AP1	1	2	3	4	5	6
0	<b>116</b>	142	130	Atandı	157	152

AP2	1	2	3	4	5	6
0	163	135	149	136	Atandı	<b>131</b>

Aşama 4: Aynı mantık ile, Makine1’de atanan 3. iş 3 işi ve Makine2’ye atanan ise 2 işidir.

Çizelge 7.4 Makinelere atanan işlerin durumları

AP1	1	2	3	4	5	6
0	Atandı	142	<b>130</b>	Atandı	157	152

AP2	1	2	3	4	5	6
0	163	<b>135</b>	149	136	Atandı	Atandı

Bu aşamalara göre; Makineler için en uygun sıralamalar:

Makine 1 : [4, 1, 3]  $C_{m1} = 109 + 124 + 166 = 399$  (Tablodaki AP'lere göre)

Makine 2 : [5, 6, 2] olmaktadır.  $C_{m2} = 127 + 127 + 157 = 411$  (Tablodaki AP'lere göre)

\*\*\*  $C_{max} = 411$  olmaktadır.

Aşama5: Artık makine içi ve makineler arası iş değiştirmelere geçilecektir.

### **AYNI MAKİNEDE YAN YANA OLAN İŞ DEĞİŞTİRMELER:**

Aynı makine üzerinde yan yana olan işlerin iş değiştirmeleri şöyle olmaktadır.

**(i) Makine 1 üzerindeki değişim:** 1.ve 2. yan yana olan işler yer değiştirir.

$$\Pi_x = [4,1,3] \quad \Pi_y = [1,4,3]$$

$$\Delta = \begin{pmatrix} 109 & 124 & 166 \\ 116 & 122 & 157 \end{pmatrix} = (116-109)+(122-124)+(157-166) = -4 < 0$$

Yani  $\Pi_y$ ,  $\Pi_x'$  ten daha iyidir. 1 işi, 4 işi ile değiştirilir. Yeni sıralama;

[1,4,3] olmaktadır.

$C_{m1} = 116 + 122 + 157 = 395$  olurken

$C_{max} = 411$  olmaktadır.

Değişen sıralama üzerinden bu sefer son 2 yan yana olan iş yer değiştirilir.

$$\Pi_x = [1,4,3] \quad \Pi_y = [1,3,4]$$

$$\Delta = \begin{pmatrix} 116 & 122 & 157 \\ 116 & 166 & 150 \end{pmatrix} = (116-116)+(166-122)+(150-157) = 37 \text{ değeri } 0 \text{ 'dan}$$

**büyüktür.**

Yani  $\Pi_y$ ,  $\Pi_x'$  ten daha iyi değildir. Çünkü  $\Delta > 0$  dır. 2.ve 3.sıradaki işler yer değiştirmez.

Dolayısı ile bütün işlere bakılmıştır (yan yana olan). Şimdi Makine 2 üzerindeki yan yana işlere bakılacaktır.

**(ii) Makine 2 üzerindeki değişim:** 1.ve 2. yan yana olan işler yer değiştirir.

$$\Pi_x = [5, 6, 2] \quad \Pi_y = [6, 5, 2]$$

$$\Delta = \begin{pmatrix} 127 & 127 & 157 \\ 131 & 173 & 142 \end{pmatrix} = 35 > 0$$

1.ve 2. sıradaki işler yer değiştirmez.

$$\Pi_x = [5, 6, 2] \quad \Pi_y = [5, 2, 6]$$

$$\Delta = \begin{pmatrix} 127 & 127 & 157 \\ 127 & 142 & 110 \end{pmatrix} = -32 < 0$$

2.ve 3.sıradaki işler yer değiştirir.

Yeni sıralama, [5, 2, 6] olur.

$$C_{m2} = 127 + 142 + 110 = 379$$

\*\*\*  $C_{max} = 395$  olmaktadır.

Dolayısı ile bütün işlere bakılmıştır (yan yana olan).

#### **AYNI MAKİNEDE YAN YANA OLMAYAN İŞ DEĞİŞTİRMELER:**

Aynı makine üzerinde yan yana olmayan işlerin iş değiştirmeleri şöyle olmaktadır.

##### **(i) Makine 1 üzerindeki değişim:**

$$\Pi_x = [1, 4, 3] \quad \Pi_y = [3, 4, 1]$$

$$\Delta = \begin{pmatrix} 116 & 122 & 157 \\ 130 & 150 & 124 \end{pmatrix} = (130-116)+(150-122)+(124-157) = 9 > 0 \text{ olmaktadır. Sıralama}$$

değiştirilmeyecektir.

##### **(ii) Makine 2 üzerindeki değişim:**

$$\Pi_x = [5, 2, 6] \quad \Pi_y = [6, 2, 5]$$

$$\Delta = \begin{pmatrix} 127 & 142 & 110 \\ 131 & 157 & 165 \end{pmatrix} = 74 > 0. \text{ Sıralama değiştirilmeyecektir.}$$

#### **FARKLI MAKİNELERDE VAR OLAN İŞ DEĞİŞTİRMELER:**

**Adım 1:**

$$\Pi_x = \text{Makine 1} = [1, 4, 3] \quad \text{ve} \quad \text{Makine 2} = [5, 2, 6]$$

$$\Pi_y = \text{Makine 1} = [5, 4, 3] \quad \text{ve} \quad \text{Makine 2} = [1, 2, 6]$$

$$\Delta_{m1} = \begin{pmatrix} 116 & 122 & 157 \\ 157 & 108 & 157 \end{pmatrix} = (157-116) + (108-122) + (157-157) = 27$$

$$\Delta_{m2} = \begin{pmatrix} 127 & 142 & 110 \\ 163 & 135 & 110 \end{pmatrix} = (163-127) + (135-142) + (110-110) = 29$$

$$C'_{max} = \text{Max}\{C_{m1} + \Delta_{m1}, C_{m2} + \Delta_{m2}\}$$

$C'_{max} = \text{Max}\{395+27, 379+29\} = 422 (> 395)$  dolayısı ile deęişim olmaz. Yani  $\Pi_y, \Pi_x'$  ten daha iyi deęildir.

Aynı işlemleri makineler arasında dięer işler için de gerçekleştiririz.

### Adım 2:

$$\Pi_x = \text{Makine 1} = [1,4,3] \quad \text{ve} \quad \text{Makine 2} = [5,2,6]$$

$$\Pi_y = \text{Makine 1} = [2,4,3] \quad \text{ve} \quad \text{Makine 2} = [5,1,6]$$

444 çıkar  $C'_{max}$ .  $444 > 395$  olduğundan deęişim olmaz.

$$\Pi_x = \text{Makine 1} = [1,4,3] \quad \text{ve} \quad \text{Makine 2} = [5,2,6]$$

$$\Pi_y = \text{Makine 1} = [6,4,3] \quad \text{ve} \quad \text{Makine 2} = [5,2,1]$$

460 çıkar  $C'_{max}$ .  $460 > 395$  olduğundan deęişim olmaz.

Yani  $\Pi_y, \Pi_x'$  ten daha iyi olmadığından, çizelge deęiştirilmez.

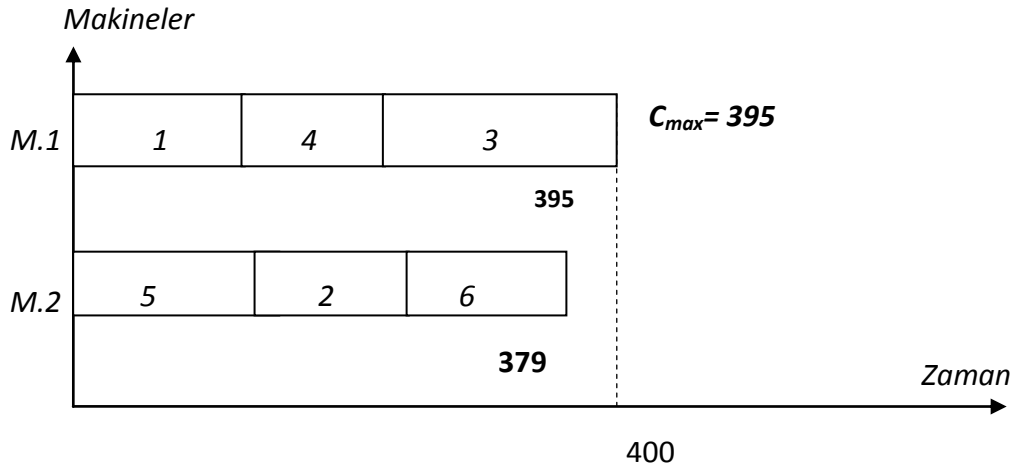
Bütün bu işlemler yapıldıktan sonra ise en uygun sıralama yazılacaktır

$$\text{Makine 1} = [1,4,3], C_{m1}=395$$

$$\text{Makine 2} = [5,2,6], C_{m1}=379$$

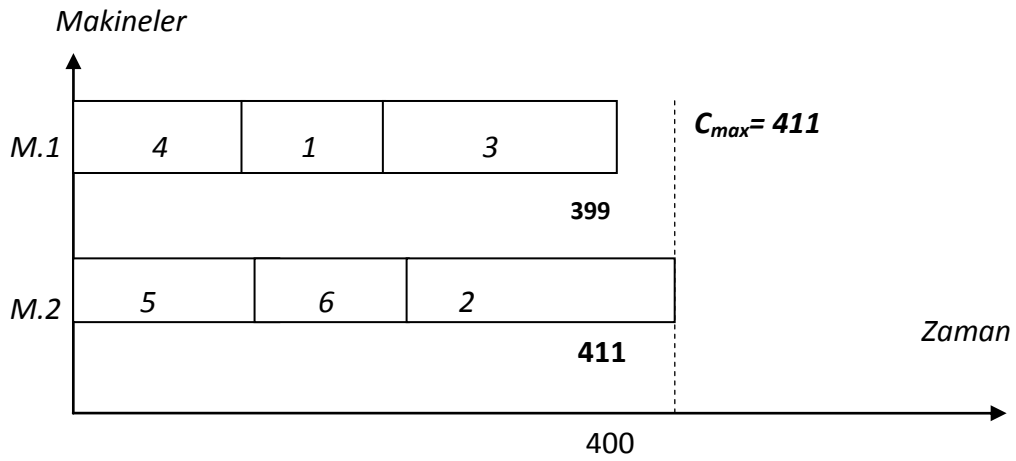
Bu duruma göre, Makine 1'deki tamamlanma zamanı 395 ve Makine 2'deki deęer ise 379 olmaktadır.

Sonrasında ise, işlemler ve makineler Şekil 7.5'deki gibi GANT diyagramına dökülür.



Şekil 7.5 İşlemlerin sonundaki GANT Diyagramı

Başlangıç aşamasındaki GANT diyagramı ise şöyle idi:



Şekil 7.6 Başlangıç aşamasındaki GANT Diyagramı

Yukarıdaki iki Gant diyagramı incelendiğinde, C<sub>max</sub> değerlerinin düştüğü ve bu bağlamda da sonuçların iyileştirildiği görülmektedir. Baskınlık özellikleri, meta sezgisel algoritmalar ile entegre edilebilmekte ve sıralama problemlerinin çözümünde oldukça etkili sonuçlar verebilmektedir.

### KURULAN MODEL VE YAPILAN UYGULAMANIN AÇIKLANMASI

Bu çalışmada hibrit akış tipi (HFS) bir sistem incelenmektedir. İşlemler, operasyonlardan oluşmaktadır. Bir işlem bir veya daha çok makinede yapılmaktadır. Farklı işler, aşamaları aynı sırada ziyaret etmektedirler. Her aşamada, bir iş sadece bir makine tarafından işlem görür. Proseste bir dizi değişken ve kabuller şu şekildedir;

1. Bütün işler birbirinden farklıdır. Sisteme bir kez gelen iş bir daha gelmemektedir.
2. Proses aşamalarının sayısı birden fazladır.
3. Her kademe en az bir kademe  $M_{(k)} > 1$  olmak üzere  $M_{(k)} \geq 1$  paralel makine içermektedir.
4. Bütün işler, takip eden aynı üretim aşamasında (aşama 1, aşama 2,...) işlem görmektedir.
5. Aşama  $k'$  da, her  $j$  işi, bir  $p_{jk}$  proses zamanına sahiptir. Biz  $k$  aşamasında  $j$  işinin işlenmesini  $O_{jk}$  operasyonu ile göstermeliyiz.
6. Verilen her aşamadaki makineler "ilişkisiz (unrelated)" paralel makinelerdir.
7. Herhangi bir makine, bir zamanda sadece bir işi işleme tabi tutabilmektedir ve herhangi bir iş, belli bir zamanda bir makine ile işleme girebilmektedir.
8. Her iş bütün aşamalara uğramaktadır.
9. Problem bilgisi deterministiktir ve önceden bilinmektedir.
10. İşlerin önceliği yoktur. Örneğin, bir makinede bir operasyon başladıktan sonra başka bir operasyonun aynı makinede işlem görmeye başlayabilmesi için öncekinin mutlaka bitirilmesi gerekmektedir.



11. Amaç, en son işin sistemden ayrılma zamanını ( $C_{max}$ ) minimize etmektir.
12. Her bir işlemin bir öncülü ve bir ardılı mevcuttur.
13. Malzeme eksikliği, makine arızası, operatör sorunları dikkate alınmamıştır.
14. Bir makinede bir işlemde başka bir işleme geçilirken hazırlık süresine ihtiyaç duyulmaktadır. Sıraya bağlı ve makineye bağlı hazırlık zamanlarının her ikisi de düşünülmüştür.
15. Her aşamadaki makinede, her iş sıralı operasyonlar halinde prosese girmektedir.
16. İşlerin bırakış süreleri ve teslim tarihleri bilinmektedir.

### 8.1 Matematiksel Modelin Açıklanması

#### İndisler:

k : aşama indisi  $k=1,2,3,\dots,a$

m : makine indisi  $m=1,2,3,\dots,mc$

l, j, i : iş indisi  $l, j, i = 1,2,3,\dots,abe$

g : zaman

abe+1 : T

#### Parametreler:

abe: iş sayısı

mc : makine sayısı

a : aşama sayısı

$b_k$  : k aşamasındaki paralel makine sayısı

$d_j$  : j işinin teslim tarihi.

$s_{kji}$  : k aşamasında, j işi ile i işi arasındaki hazırlık zamanı.

$r_{kmj}$  : k aşamasında, eğer j işi m makinesine atanan ilk iş ise, j işinin hazırlık zamanı.

$R_{km}$  : k aşamasında, m makinesinin hazır olduğu zaman.

$O_{kj}$  : k aşamasında, j işinin operasyon zamanı.

$C_{kj}$  : k aşamasında, j işinin tamamlanma zamanı.

$C_{0j}$  : eğer j işi ilk iş ise j işinin tamamlanma zamanı.

$C_{max}$  : tamamlanma zamanı.

F : çok büyük bir sayı.

### Karar Değişkenleri:

$X_{kmji}$  : {1, eğer j işi k aşamasındaki m makinesinde i işinden hemen önce prosese giriyorsa  
0, aksi takdirde }

### Amaç Fonksiyonu:

$$\text{Min } C_{max} \quad (1)$$

### Kısıtlar :

$$\text{s. t. } \sum_{m \in M^k} \sum_{\substack{j \in J^k \cup \{0\} \\ j \neq i}} X_{kmji} = 1, \quad \forall k, i \in J^k, \quad (2)$$

$$\sum_{m \in M^k} \sum_{\substack{i \in J^k \cup \{T\} \\ i \neq j}} X_{kmji} = 1, \quad \forall k, j \in J^k, \quad (3)$$

$$\sum_{i \in J^k \cup \{T\}} X_{km0i} = 1, \quad \forall k, m \in M^k \quad (4)$$

$$\sum_{j \in J^k \cup \{0\}} X_{kmjT} = 1, \quad \forall k, m \in M^k, \quad (5)$$

$$\sum_{\substack{i \in J^k \cup \{T\} \\ i \neq j}} X_{kmji} - \sum_{\substack{l \in J^k \cup \{0\} \\ l \neq j}} X_{kmlj} = 0, \quad \forall k, m \in M^k, j \in J^k \quad (6)$$

$$O_{kj} = \sum_{m \in M^k} \sum_{i \in J^k \cup \{T\}} X_{kmji} p_{kmj}, \quad \forall k, j \in J^k \quad (7)$$

$$C_{0j} = 0 \quad \forall j, \quad (8)$$

$$C_{ki} - C_{kj} \geq s_{kji} + O_{ki} + \left( \sum_{m \in M^k} X_{kmji} \right) - 1)F, \quad \forall k, j, i, \quad (9)$$

$$C_{ki} - C_{(k-1)i} \geq \sum_{m \in M^k} \sum_{j \in J^k} X_{kmji} \cdot s_{kji} + \sum_{m \in M^k} r_{kmi} \cdot X_{km0i} + O_{ki}, \quad \forall k, i \in J^k \quad (10)$$

$$C_{kj} \geq \sum_{m \in M^k} X_{km0j} (R_{km} + r_{kmj}) + O_{kj}, \quad \forall k, j \in J^k, \quad (11)$$

$$C_{max} \geq C_{aj} \quad \forall j, \quad (12)$$

$$X_{kmji} \in \{0,1\}, \quad \forall k, m, j, i, \quad (13)$$

Burada,

$$p_{kmj} = \left\{ \begin{array}{ll} \frac{p_{kj}}{v_{kmj}} & \text{eğer } v_{kmj} \neq 0, \\ \infty & \frac{o}{w}, \end{array} \right\} \quad \forall k, m \in M^k, i \in J^k \quad (14)$$

F, yani “çok büyük bir sayı”, (9). eşitlikteki gibi, geniş bir aralıkta seçilmelidir. Ancak, problem içerisinde bu durumu makul sınırlar altına almak adına, F’ nin maksimum  $J^k / (O_{kj} + s_{kji})$  değerinden çok daha büyük olması yeterli olacaktır.

Problemimiz bu yaklaşımla şöyle ilerleyecektir:

k aşamasında, m makinesinde, j işinin proses zamanı ( $p_{kmj}$ ) ve k aşamasında, j ile i işi arasındaki hazırlık zamanı ( $s_{kji}$ ) değeri olmaktadır. Eğer iş, k aşamasında ya da m makinesinde prosese girmiyorsa, sonsuz bir değere ilerleyecektir. Böylece ( $O_{kj} + s_{kji}$ ) değeri sınırsız görünecektir ve F “ $\infty$ ” olmaya ihtiyaç duyacaktır.

Ancak biz bu sonuçları kesmek yani sınırlamak istediğimizden, sadece ( $O_{kj} + s_{kji}$ ) kombinasyonlarının mümkün olanlarını düşünebiliriz ve düşüneceğiz. Yani bu durumda F değerini (16). eşitlikteki gibi yapacağız.

Söylenildiği gibi, **“bazı işler ve bazı makineler, bazı aşamalarda uygun değiller”** sözü, modelde en önemli kısımımızdır. Böylelikle model çok daha küçük olacaktır. Ayrıca çözülebilirliği de daha etkin olacaktır.

( $p_{kmj} = \infty$ ) ve ( $s_{kji} = \infty$ )’ un, “negatif performans etkisi”ni elimine etmek için,  $M^k$  (k aşamasındaki) makine seti,  $M_j^k$  ile yer değiştirir. Burada, “k aşamasında j işini prosese sokan “mevcut/uygun” durumda olan makine” kastedilmektedir.

$$M_j^k = \{m \mid p_{kmj} < \infty, \quad m \in M^k \} \quad (15)$$

$$F = \max_{k,i \in J^k, j \in J^k} \{card(J^k) * (O_{kj} + s_{kji}) \mid M_i^k \cap M_j^k \neq \emptyset\} \quad (16)$$

1. kısıt ile amaç fonksiyonu tanımlanmıştır. Yani maksimum tamamlanma zamanının minimize edilmesi bizim amaç kısıtımızdır.

Kısıt 2 ve 3, her aşamada her sıralı pozisyona (sadece) tek bir işin atanacağını göstermektedir. Ayrıca belirli işler, bu aşamalarda uygun (available) makinelere uğramaktadır.  $J^k$ , aşamalardaki (k aşamalarındaki) iş sayılarını belirtmektedir.

Kısıt 4 ve 5 sadece bir işin, her aşamada ve belirli/uygun (available) makinede, sırası ile ilk ve son pozisyonlara atanabileceğini garanti eder.

Kısıt 6, her aşamada uygun bir sıralama yapılmasını mecbur etmektedir. Bu kısıt, işlerin "consecutive (ard arda)" olmasını sağlamaktadır. Her iş, makinelerde, işin prosese girdiği yerde, bir öncüle ve bir ardıla sahiptir.  $l, j$  ve  $i$  işleri birbirlerinden farklıdır. Her aşamada, belirli makinelerde, belirli işler işlem görmektedir.

Kısıt 7, makineye bağlı olan her işin operasyon zamanını tanımlamaktadır. Burada  $p_{kmj}$ , k aşamasında, m makinesinde bulunan j işinin proses zamanını anlatmaktadır.  $v_{kmj}$  ise nispi hızıdır. Bu değer, sıfır'dan farklı bir değer aldığı anda (ki böyle olmalıdır zaten), " $\frac{p_{kj}}{v_{kmj}}$ " olmaktadır. Burada anlatılan paralel makine, ilişkisiz paralel makinedir.

Kısıt 8, başlangıç aşamasında her işin tamamlanma zamanının sıfır olduğunu söyler.

Kısıt 9' dan 12' ye kadar ise her işin tamamlanma zamanı bulunmaktadır. Kısıt 9' daki durum şöyledir; eğer j ve i işi, i işinden önce j' nin sıralanması ile, her aşamada, belirli

(available) makinede sıralanırsa; ardından j işi, i işi başlamadan önce prosesi tamamlamak zorundadır.

Kısıt 10 ile söylenen şey, bir işin k aşamasında bitmesinden önce “k+1” aşamasında prosese asla başlayamayacağıdır.

Kısıt 11, her makinede ilk sıralamada atanan işlere uygulanacaktır. Aşamalardaki makinelerin hazır bulunmaları önemlidir.

Kısıt 12, maksimum tamamlanma zamanının, son işin tamamlanma zamanından büyük veya eşit olacağını göstermektedir.

13. Kısıt,  $X_{kmji}$  karar değişkenini, ikili değişken olarak belirtmektedir.

## **8.2 Matematiksel Modelin Uygulaması**

Veriler, Excel üzerinde, “UHFS OPL örnek yaratımı” ile oluşturulmuştur. Burada, iş, makine ve aşama sayıları tek tek girilmekte, ardından “create instance for UHFS OPL model” sekmesi tıklanmaktadır. Çizelge 8.1 ‘de modelin, başlangıç test aşamasında iken, düşünülen parametreler ile tanımlamaları tek tek gösterilmektedir. Çizelge 8.2’de ise başlangıç test aşamasında düşünülen faktörler verilmektedir.

Excel’de iş, makine, aşama sayıları, k aşamasında hazır olan makineler, k aşamasında yapılması gereken işler, k aşamasında m makinesindeki ayrılış zamanı, j işi ile k aşamasında m makinesinin ilk hazırlık zamanı, m makinesinde k aşamasında j işinin proses zamanı, k aşamasında i ve j işi arasındaki hazırlık zamanı ve k aşamasında j işini prosese sokan uygun makinelerin tanımlamaları girilmiştir.

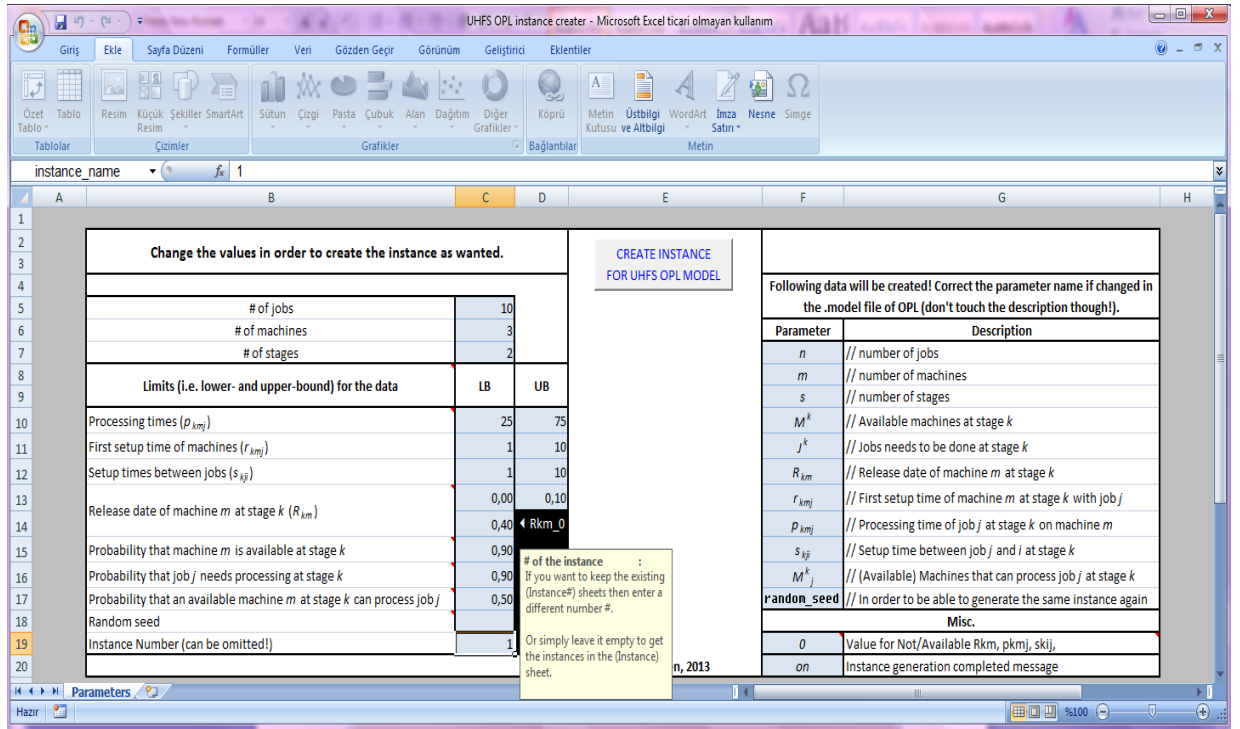
Çizelge 8.1 Başlangıç test aşamasında düşünülen parametreler

Tanımlamalar	Parametreler
İş sayıları	N
Makine sayıları	M
Aşama sayıları	S
k aşamasında hazır olan makineler	Mk
k aşamasında yapılması gereken işler	Jk
k aşamasında m makinesindeki ayrılış zamanı	Rkm
j işi ile k aşamasında m makinesinin ilk hazırlık zamanı	Rkmj
m makinesinde k aşamasında j işinin proses zamanı	Pkmj
k aşamasında i ve j işi arasındaki hazırlık zamanı	Skji
k aşamasında j işini prosese sokan uygun makineler	Mkj

Çizelge 8.2 Başlangıç test aşamasında düşünülen faktörler

Faktör	LB	UB
Proses zamanları (pkmj)	25	75
Makinelerin ilk hazırlık zamanları (rkmj)	1	10
İşler arasındaki hazırlık zamanları (skji)	1	10
k aşamasında m makinesinde ayrılış zamanı (Rkm)	0	0,1
k aşamasında uygun olan m makinesinin olasılığı	0,4	
k aşamasında prosese ihtiyaç duyan j işinin olasılığı	0,9	
k aşamasında j işini prosese sokan uygun olan m makinesinin olasılığı	0,9	

Şekil 8.1'de "UHFS-OPL" için EXCEL'de örnek yaratımına dair Print screen ekranı görülmektedir.



Şekil 8.1 UHFS-OPL örnek yaratımı

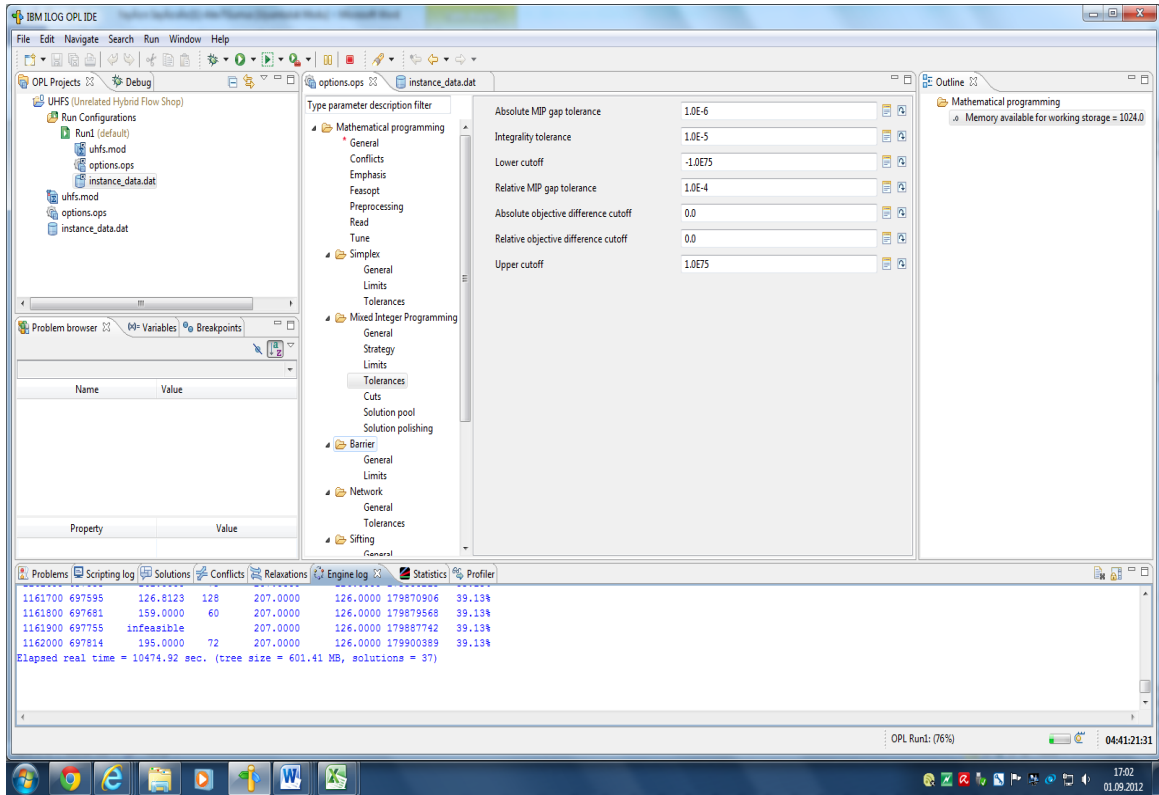
OPL (optimization programming language/optimizasyon programlama dili), mühendisler tarafından sıklıkla kullanılan bir programlama dilidir. Çizelge 8.3'de oluşturulan Modelin, OPL'deki ayrı ayrı veri setleri için sonuçları görülmektedir.

Çizelge 8.3 OPL sonucundaki Örnek Veriler Tablosu

ÖRNEK VERİLER	OPL Cmax
4 iş / 2 aşama / 2 makine	205
4 iş / 2 aşama / 4 makine	155
6 iş / 3 aşama / 3 makine	152
6 iş / 2 aşama / 3 makine	227
8 iş / 4 aşama / 4 makine	287
8 iş / 3 aşama / 4 makine	296
9 iş / 3 aşama / 3 makine	Çözemedi
10 iş / 3 aşama / 5 makine	Çözemedi
12 iş / 3 aşama / 6 makine	Çözemedi
12 iş / 4 aşama / 6 makine	Çözemedi

Buna göre, verilerin boyutu büyüdükçe, modelin çalışma süresi de uzamaya başlamaktadır. Örnek verilerin tanımlamalarını, "iş/makine/aşama" olarak düşünürsek, model, 4/2/2, 6/3/3, 8/4/4 örnekleri için çalışıyor ve sonuç veriyorken, MAKİNE ve AŞAMA sayıları arttıkça çok uzun sürede çözmeye başlamaktadır. Örneğin,

15 iş, 5 makine, 2 aşama (15/5/2) örneği için OPL oldukça uzun zaman harcamaktadır. Şekil 8.2.'de bu örnek için olan “print screen” ekranı görülmektedir.



Şekil 8.2 OPL' de 15/5/2 test verisi için örnek ekranı

### 8.3 Genetik Algoritma ve Üstünlük Özellikleri Kullanımı

Model çok uzun zaman harcamaya ve çözümü zorlaşmaya başladıktan sonra, bir sezgisel kullanımına karar verilmiştir. NP-zor sınıftaki problemlerin çözümü için polinomial zamanlı algoritma yoktur. NP-zor sınıftaki problemler için dal-sınır ya da kesme-düzlemi gibi etkin yöntemlerin de başarısız olma nedeni, bu yöntemlerde üstel sınırların olmayışıdır.

Sezgisel algoritmalar, çözüm zamanlarını azaltan algoritmalarlardır. Bu algoritmalar en iyi sonucu bulacaklarını garanti etmezler ancak makul bir süre içerisinde bir çözüm elde edebileceklerini garanti ederler. Sıklıkla en iyiye yakın olan çözüm yoluna hızlı ve kolay bir şekilde ulaşırlar. Model, OPL programlama dilinde uzun zaman harcamaya başladıktan sonra bir sezgisel/metasezgisel yöntemin kullanılmasının iyi olacağına karar verilmiştir.



Bütün bu nedenlerden dolayı, çalışmanın 2. Bölümü'nde de verilen literatür taramalarının ardından bu çalışma içerisinde "sezgisel" olarak kullanılabilir en uygun yöntemin bir metasezgisel olan "genetik algoritma" olacağı düşünülmüştür. Genetik Algoritma'da iyi en iyi ilk çözümlerin bulunması için ise, üstünlük özellikleri uygulanmıştır.

Sezgiselin çözümü ve kodlanması için, programlama dili olarak "C# (Csharp) Programlama Dili" kullanılmıştır. C#, Microsoft'un geliştirmiş olduğu yeni nesil programlama dillerinden birisidir.

### **8.3.1 Parametre Seçimi**

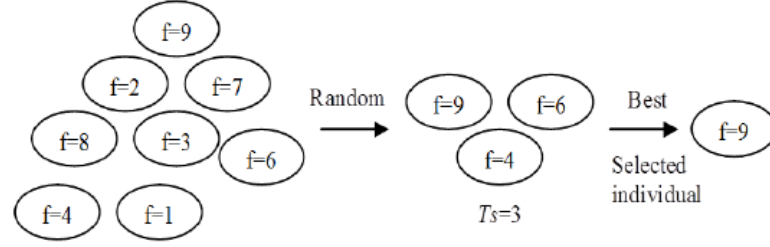
Daha önce de anlatıldığı gibi, genetik algoritma kullanımı içerisinde parametre seçimleri önemli olmaktadır. Başlangıç popülasyonu 30 olarak alınmıştır. Seçim yöntemi olarak ise, "turnuva seçimi" kullanılmıştır. Döngü sayısı, ilk etapta 20 olarak belirlenmiştir. Her kromozomun tek tek uygunluk fonksiyonları bulunup, ardından "elitist" seçim kuralı olarak "turnuva seçimi kuralı" uygulanacaktır.

Turnuva seçimi, popülasyondan rassal olarak seçilen birkaç bireyin arasından yapılan seçimi gösterir. Burada turnuva koşumu önemli olmaktadır. Her turnuvada kazanan en iyi uygunluktaki birey çaprazlamaya gönderilmektedir.

Turnuva seçimi, seçim zorluğunu turnuva boyutu olan "s" ile s'ye rakip olanlar arasından bir turnuva tutarak sağlamaktadır. Turnuvanın kazananı "s" turnuva rakibi arasından en yüksek uygunluk değerli olan bireydir. Kazanan daha sonra eşleştirme havuzuna eklenecektir. Eşleştirme havuzu, ortalama popülasyon uygunluğundan daha yüksek uygunluğa sahip turnuva kazananlarını içermektedir. Bu uygunluk farklılığı, her oluşturulan jenerasyonun uygunluğunu geliştirmek için GA'yı ilerletmektedir.

N adet birey, popülasyondan rassal olarak seçilmektedir. Seçilen bireyler, kendi aralarında kıyaslanır. En yüksek uygunluk değerine sahip olan birey kazanır ve sonraki

jenerasyon popülasyonunun bireyi olarak seçilir. Her turnuvada rekabet eden birey sayısı “turnuva boyutu” olarak addedilir. Şekil 8.3, turnuva seçim mekanizmasını göstermektedir.



Şekil 8.3 Turnuva seçim mekanizması

Burada  $T_s$ , yani turnuva boyutu 3' tür. Anlamı 3 kromozom birbiri ile rekabet halindedir. Bunların arasından yalnızca en iyi kromozom “yeniden üreme (reproduction)” için seçilecektir. 30 birey ilk etapta seçilecektir. Her bireyin uygunluk değerleri bulunacaktır. Uygunluk değerleri, her bireyin tek tek  $C_{max}$  değerleri olmaktadır.

Buna göre, yüksek olan fitness'lı kromozom alınıp, “eşleşme havuzu”na konulacaktır. Eşleşme havuzu, çaprazlama ve mutasyon operatörlerinin uygulandığı yer demektir. Sonra bu eşleşme havuzundaki seçilenlere tek tek, çaprazlama ve mutasyon işlemleri uygulanacaktır. Bu şekilde gelen bireyler de alınacaktır. Bu iyi bireyler de eşleşme havuzuna konulacaktır. İlk etaptaki 30 birey ile birleştirilmeyecektir.

Çaprazlama ve mutasyon işlemleri bitince, havuzdakiler yeniden başlangıç popülasyonuna eklenecek ve tekrar sıralanacaktır. En iyi 30 birey alınacak, diğerleri atılacak ve sıradaki işlem tekrarlanacaktır.

Çizelge 8.4 Kullanılan Parametreler

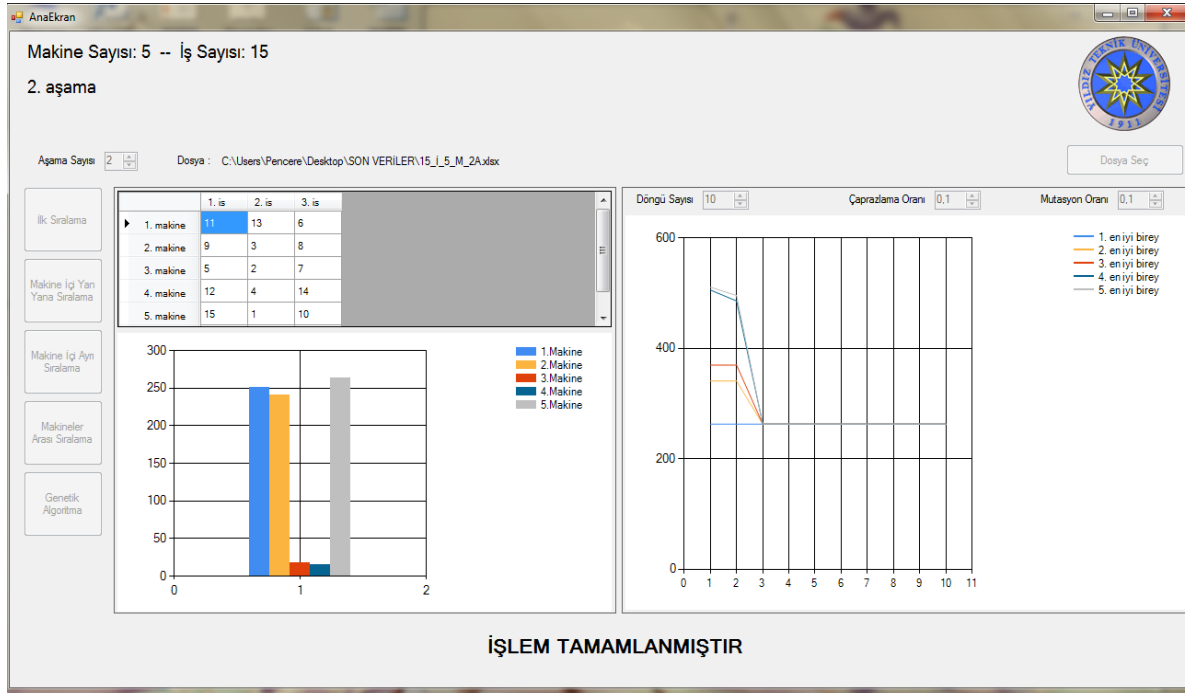
Çaprazlama oranı	0.1
Mutasyon oranı	0.1
Döngü sayısı	20
Başlangıç popülasyon sayısı	30
Mutasyon şekli	Keyfi üç iş değiştirme
Çaprazlama şekli	Pozisyona dayalı çaprazlama
Seçim Kuralı	Turnuva seçimi

Örneğin, 15/5/2 örneği OPL programında çok uzun zaman harcarken, C# ile çözülmeye çalışılacaktır. Aşağıda bu örnek için olan “print screen” ekranları vardır. Bu verilere göre oluşturulan Csharp ekran çıktısı Şekil 8.4’deki gibi olmaktadır. Ayrıca Döngü sayısı ilk etapta 10 olarak alınmıştır. Çaprazlama/mutasyon oranları, döngü sayısı, başlangıç popülasyon sayısı, mutasyon şekli, çaprazlama şekli ve seçim kuralı, Çizelge 8.3’de anlatıldığı gibi alınmaktadır. Bu sonuçlar **genetik algoritma** ile **baskınlık özellikleri** birleştirilerek alınmıştır.

Verilen bir sıralama için “baskınlık özellikleri” kullanılarak, optimale yakın bir çözüm elde edilebileceği daha önceki kısımlarda anlatılmıştı. Büyük boyutlu problemler için, çözüm kalitesini arttırmaya yardımcı olmak adına, baskınlık özelliklerinin genetik algoritma ile birleştirildiği, yeni bir meta sezgisel ortaya konulabiliyordu. Yani, iyi ilk çözümler baskınlık özellikleri ile elde edilip ardından genetik algoritma metasezgiseline sokulmaktadır.

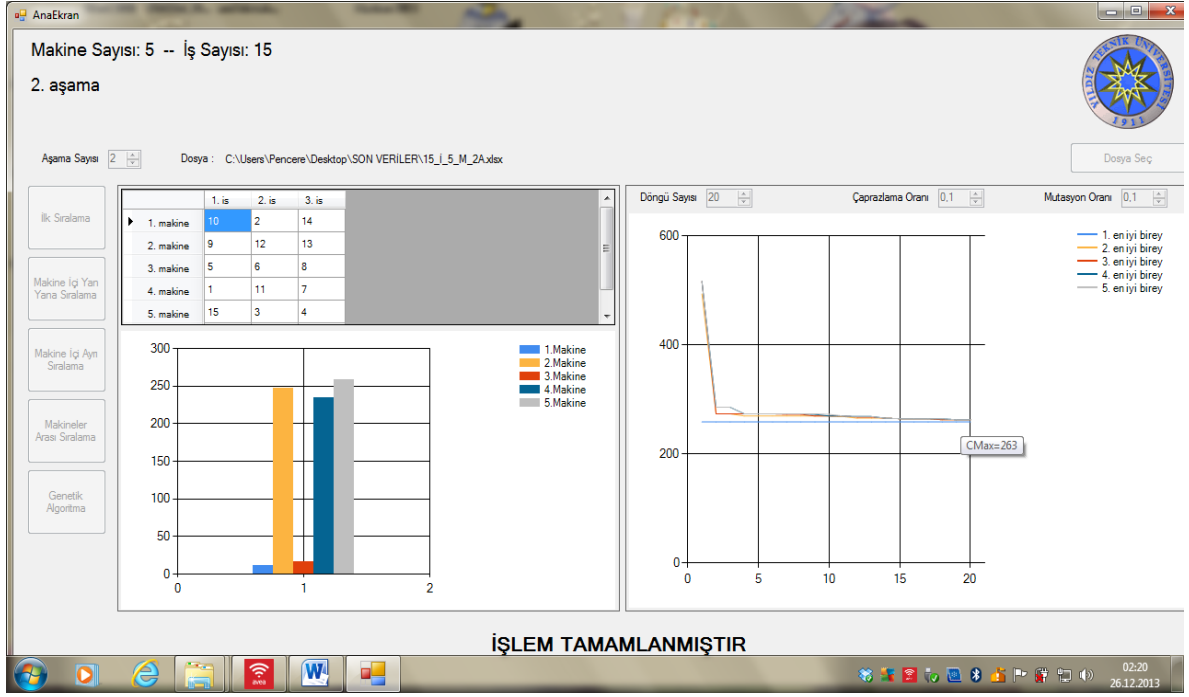
Optimal sıralamanın üstünlük özellikleri, iki bitişik işin,  $i$  ve  $j'$  nin yer değişimi temeline dayanmakta idi. Üstünlük özellikleri (DP), meta sezgiseller uygulanmadan önce, “ilk” verimli çözümleri elde etmekteydiler. Böylelikle, DP iyi ilk çözümleri ürettiği zaman, metasezgiseller daha hızlı yakınsayabilecektir.

Bütün bu verilerin ışığında, 15/5/2 örneği için  $C_{max}$  değeri 267 çıkmaktadır. Daha sonraki örnekte ise döngü sayısının artırıldığı görülmektedir. Buna göre, Şekil 8.5’de döngü sayısı 10’dan 20’ye çıkartılır. Genetik Kodlama, C sharp programlama dili ile yapılmıştır. C# dili, basit, modern, genel amaçlı ve nesneye yönelik bir programlama dili olarak, çizelgeleme problemlerinde Matlab’tan sonra sıklıkla kullanılan bir programlama dilidir.



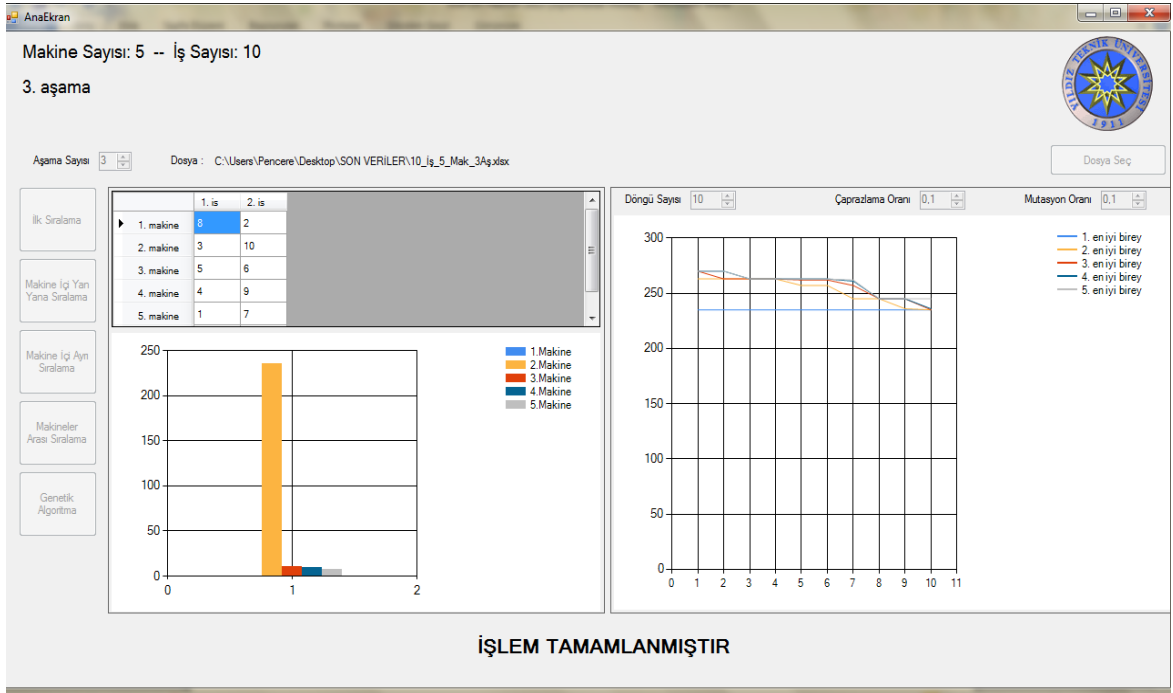
Şekil 8.4 15 İş 2 Aşama 5 Makine (10 döngü)

Eğer döngü sayısı 20, çaprazlama ve mutasyon oranları da yine 0.1 olarak alınırsa,  $C_{max}$  263 olmakta ve sonuç ekranı Şekil 8.5’deki gibi olmaktadır:

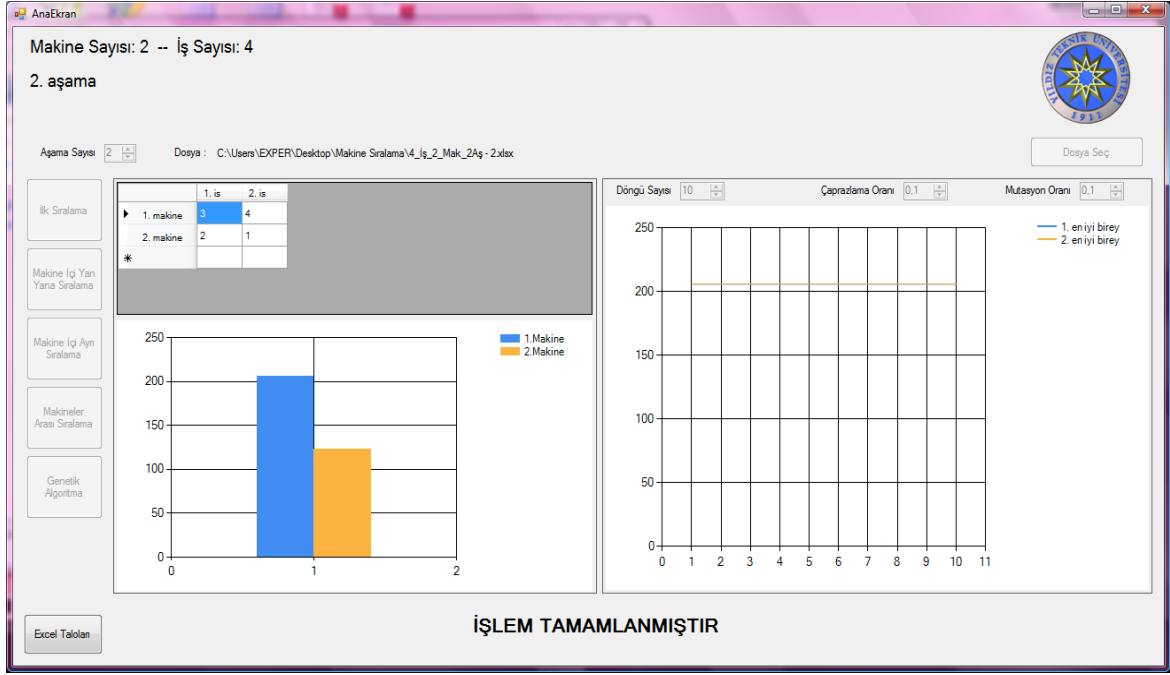


Şekil 8.5 15 İş 2 Aşama 5 Makine (20 döngü)

Aynı şekilde 18 iş 9 makine 3 aşama için ise Cmax 272 olmuştur. 10 iş 5 makine 3 aşama (10 döngü) için ise sonuçlar aşağıdadır ve Cmax değeri Şekil 8.6' da da görüldüğü gibi, 236 olmaktadır.

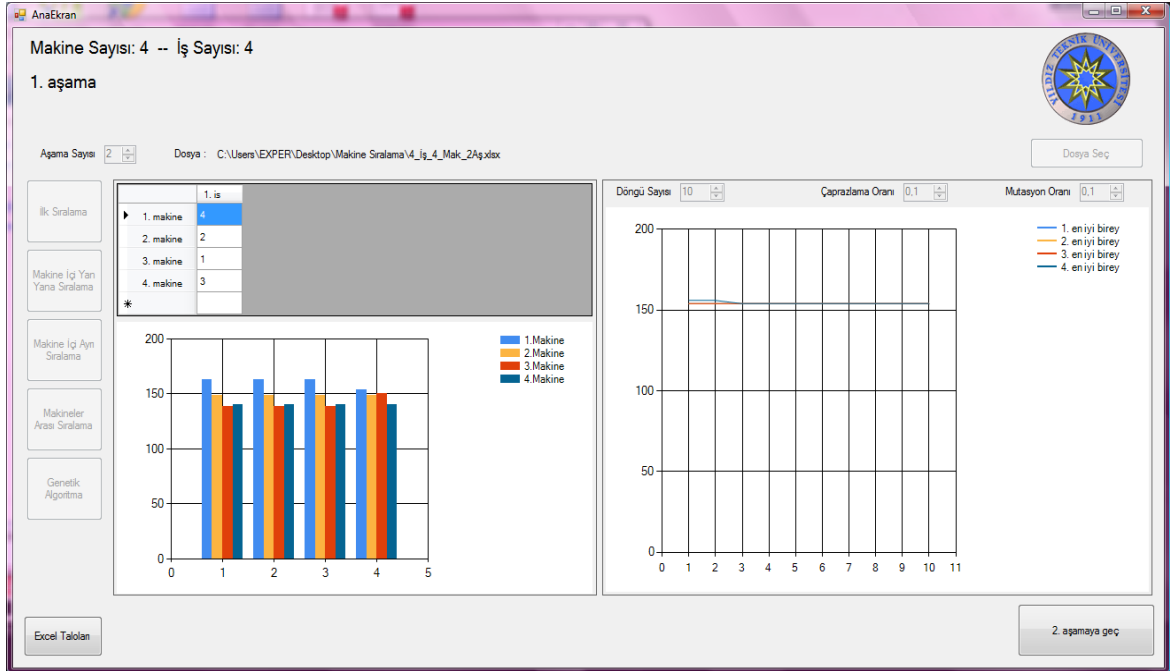


Şekil 8.6 10 İş 5 makine 3 aşama (10 döngü)



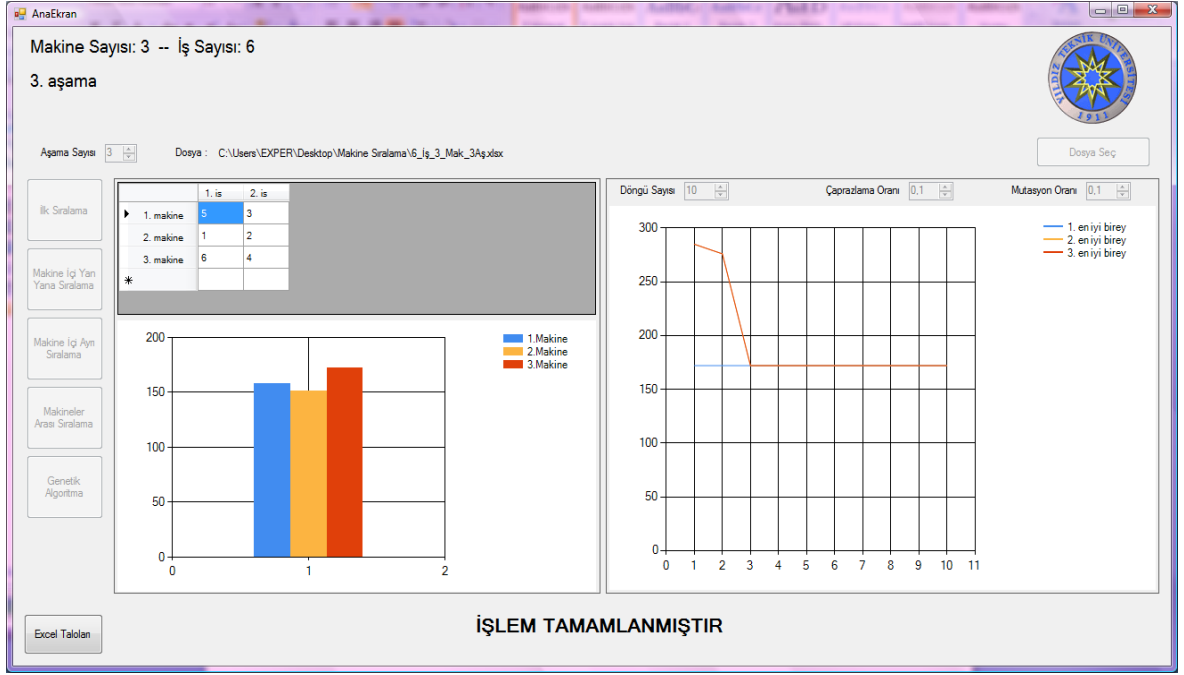
Şekil 8.7 4 iş 2 aşama 2 makine (10 döngü)

4 iş 2 aşama 2 makine örnek verileri için ise (Şekil 8.7), Döngü sayısı gene 10, Çaprazlama ve Mutasyon oranları yine 0.1 alındığında ise Cmax değeri; 206'yı göstermektedir.



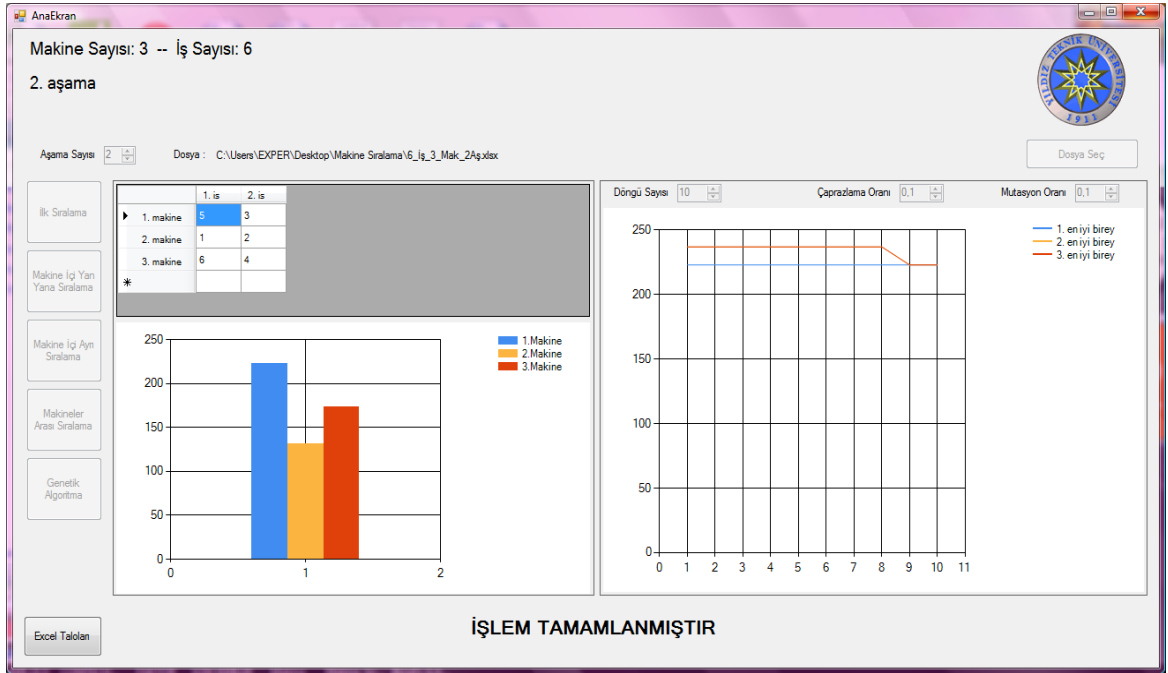
Şekil 8.8 4 iş 2 aşama 4 makine (10 döngü)

4 iş 2 aşama 4 makine örnek verileri için ise (Şekil 8.8), Döngü sayısı gene 10, Çaprazlama ve Mutasyon oranları yine 0.1 alındığında ise Cmax değeri; 156 olmaktadır.

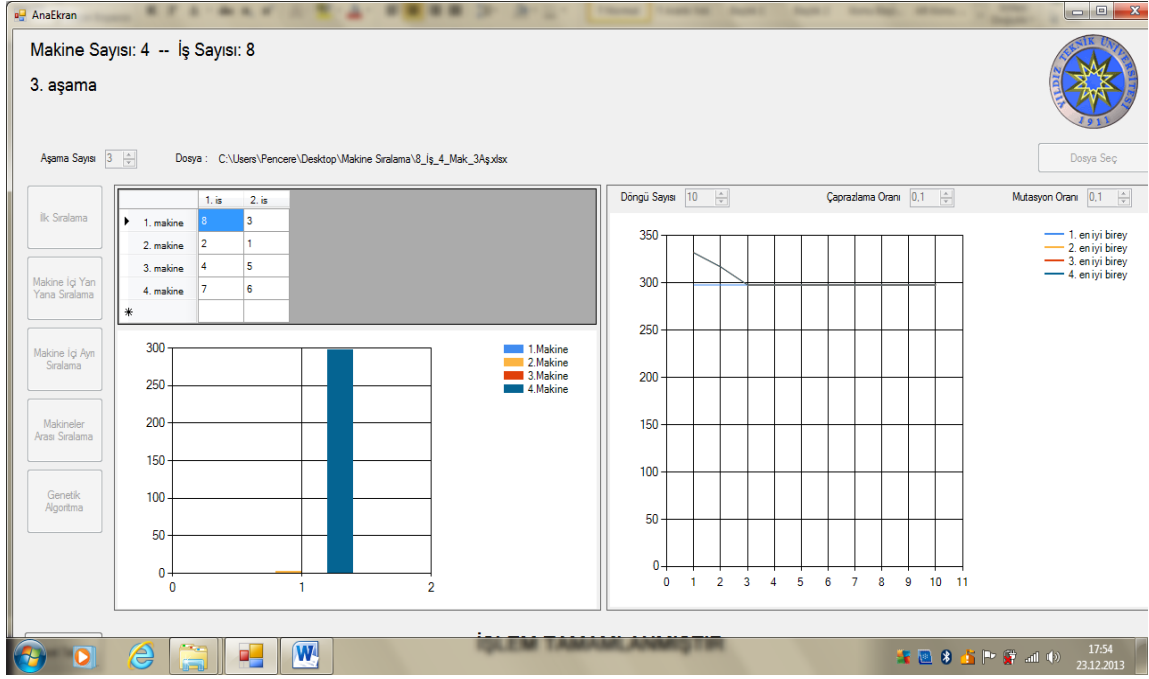


Şekil 8.9 6 iş 3 aşama 3 makine (10 döngü)

6 iş 3 aşama 3 makine örnek verileri için ise (Şekil 8.9), Döngü sayısı gene 10, Çaprazlama ve Mutasyon oranları yine 0.1 alındığında ise Cmax değeri; 175 olmaktadır. 6 iş 2 aşama 3 makine örnek verileri için ise (Şekil 8.10), Döngü sayısı gene 10, Çaprazlama ve Mutasyon oranları yine 0.1 alındığında ise Cmax değeri; 227 olmaktadır.

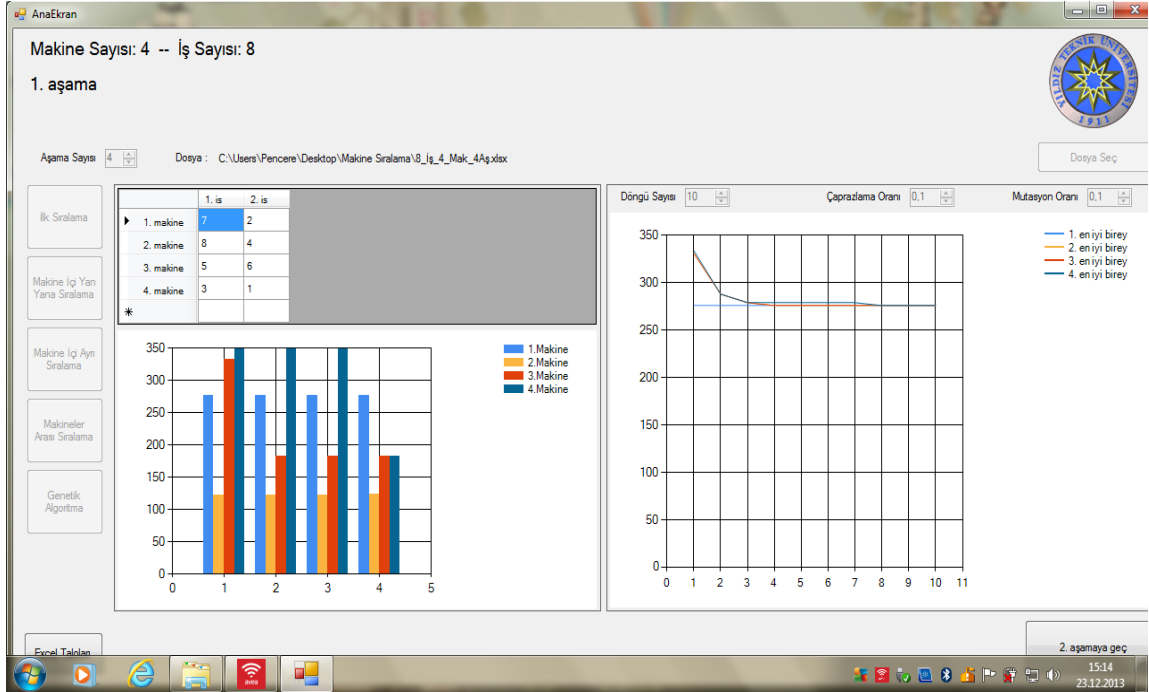


Şekil 8.10 6 iş 2 aşama 3 makine (10 döngü)



Şekil 8.11 8 iş 3 aşama 4 makine (10 döngü)

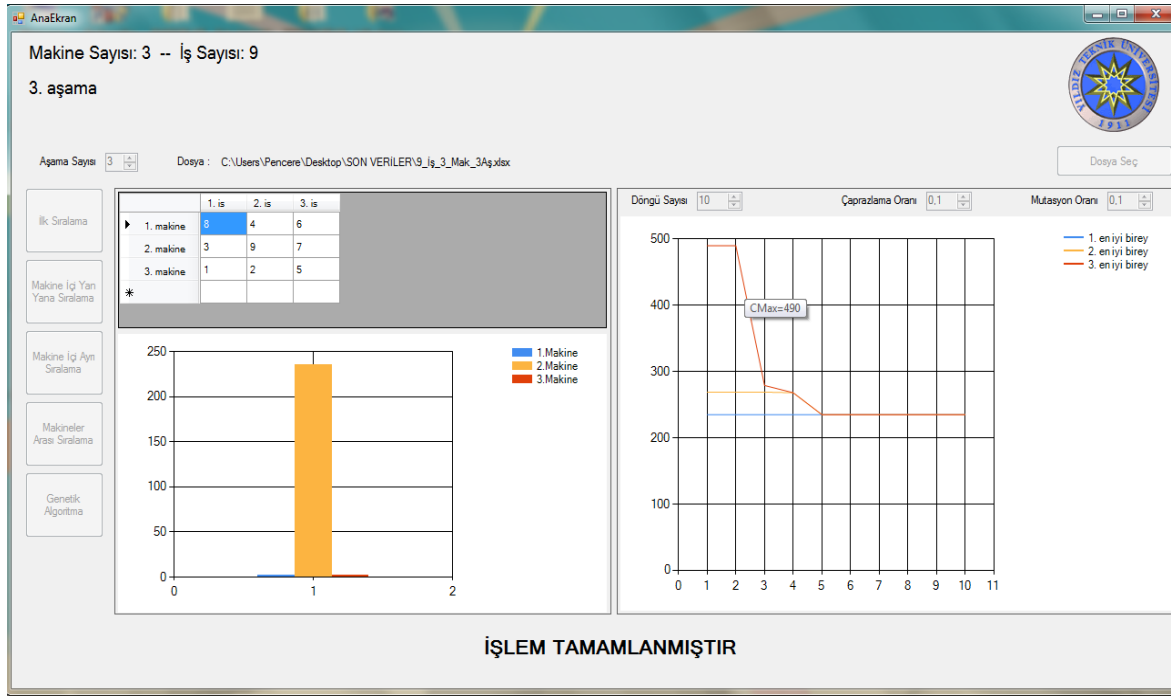
8 iş, 3 aşama, 4 makine için Cmax değeri Şekil 8.11’de görüldüğü gibi, 298 olmaktadır.



Şekil 8.12 8 iş 4 aşama 4 makine (10 döngü)

8 iş, 4 aşama, 4 makine için ise Cmax değeri 280’i vermektedir. 9 iş 3 aşama 3 makine için (Şekil 8.13) ise Cmax değeri 245 olmaktadır.





Şekil 8.13 9 iş 3 aşama 3 makine (10 döngü)

$$GAP = (Sezgisel - OPL) / OPL$$

Bütün bu veriler ışığında, bulunan GAP değerleri Çizelge 8.3' deki gibi olmaktadır. GAP yani yüzde oranları da Çizelge 8.5' de son sütunda görülmektedir.

Çizelge 8.5 Örnek Veriler Tablosu (Tüm sonuçlar için)

ÖRNEK VERİLER	GA-DP $C_{max}$	OPL $C_{max}$	GAP
4 iş / 2 aşama / 2 makine	206	205	0.0048
4 iş / 2 aşama / 4 makine	156	155	0.0064
6 iş / 3 aşama / 3 makine	175	152	0.1513
6 iş / 2 aşama / 3 makine	228	227	0.0044
8 iş / 4 aşama / 4 makine	287	285	0.007
8 iş / 3 aşama / 4 makine	298	296	0.0067
9 iş / 3 aşama / 3 makine	245	Çözemedi	-
10 iş / 3 aşama / 5 makine	236	Çözemedi	-
15 iş / 2 aşama / 5 makine	267	Çözemedi	-
18 iş / 4 aşama / 6 makine	272	Çözemedi	-

Verileri küçükten büyüğe doğru sıraladığımızda, Genetik Algoritma (GA) ve Dominance Properties (Üstünlük özellikleri-DP) yöntemi ile 9 iş / 3 aşama / 3 makine'den sonraki örneklerin çözülebildiği görülmektedir. Oysaki 9 iş / 3 aşama / 3 makine, 10 iş / 3 aşama / 5 makine, 15 iş / 2 aşama / 5 makine ve 18 iş / 4 aşama / 6 makine için OPL ile

herhangi bir sonuç alınamamış, yani OPL çok uzun zaman harcamıştır. Aynı örnekler için ise, GA-DP sezgiselinde ise sonuçlar 245, 236, 267 ve 272 olmaktadır. Bu sonuçlara göre, Hibrit Akış Tipi-ilişkısız paralel makineler örneklerinde görölmektedirki, GA-DP oldukça verimli sonuçlar vermektedir.

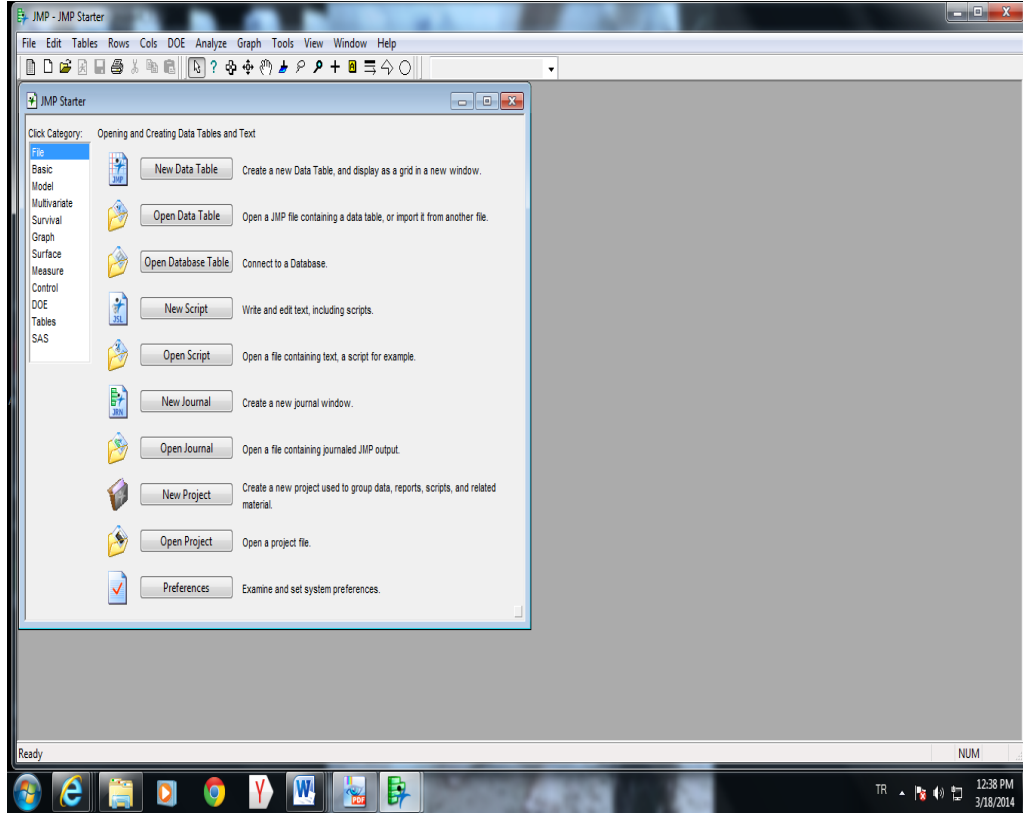
### DENEYSEL TASARIM SONUÇLARI

Lineer ve lineer olmayan regresyon modellerinde normal dağılım önemli bir rol oynamaktadır. Hem lineer hem de lineer olmayan modellerde sonuç çıkarımı için Y yanıt değişkeninin normal dağılıma sahip olduğu varsayılmaktadır. Bazı durumlarda bu varsayım gerçekçi olmamaktadır. Yanıt değişkeninin sayı (count) gibi kesikli değişken olması durumu, yani kusurların sayısı, trafik kazaları belirli hastalığa yakalanan insanların sayısı veya deprem ve kasırgaları içeren doğal olayların meydana gelme sayısı gibi “nadir olaylar” örnek olarak verilebilir [134].

Kritik kararların alınmasına destek veren “Genelleştirilmiş Lineer Modeller”, karşı karşıya kaldığımız normal olmayan verinin analizinde kullanılmaktadır. Genelleştirilmiş Lineer Modeller yazılım yardımıyla oluşturulmaktadır.

Bu çalışma içerisinde, deneysel tasarım kısmında, “Genelleştirilmiş Lineer Modeller” kullanılmıştır. “Orthogonal Design (ortogonal tasarım)” kavramı ile birlikte deneysel tasarım yürütülmüştür. Kısaltması POOD olan ortogonal tasarım, veritabanı araştırmacıları David McGoveran ve Christopher J. Date tarafından 1990’ların başında geliştirilmiştir. Veritabanı tasarımının 2 ilkesinden biri, veritabanlarını çok karmaşık ya da gerekenden fazla olmasını önlemeye yönelik olmuştur. Veritabanı normalizasyonu olarak POOD kontrol edilemeyen bellek ve anlamlı belirsizliği elimine etmek için hizmet etmektedir [135].

Bu çalışmada deneysel tasarım için kullanılan istatistiksel program ise, JMP programı’dır. JMP programı açılış sayfası Şekil 9.1’deki gibidir:



Şekil 9.1 JMP programı açılış sayfası

Sırası ile 4iş-2aşama-2makine, 4iş-2aşama-4makine, 6iş-3aşama-3makine, 6iş-2aşama-3makine, 8iş-4aşama-4makine, 8iş-3aşama-4makine, 9iş-3aşama-3makine, 10iş-3aşama-5makine, 15iş-2aşama-5makine, 18iş-4aşama-6makine örnekleri için “deneysel tasarım” yapılmıştır. Döngü sayısı, Mutasyon ve Çaprazlama oranları değiştirilerek, aralarında anlamlı farklar olup olmadığı araştırılmıştır. 4iş-2aşama-2makine için çıkan “genelleştirilmiş lineer analiz” sonuçları Çizelge 9.1’de görüldüğü gibidir. Dik tasarım çalışması içerisinde genetik algoritma parametrelerinden; 5, 10, 15 döngü sayısı, 0.1, 0.2, 0.3, 0.4 mutasyon ve çaprazlama oranları tek tek denenerek yukarıdaki örneklerin analizleri yapılmıştır. Bütün bu anlatılanların yardımı ile Çizelge 9.1’de 4iş-2aşama-2makine için dik tasarım sonuçları görülmektedir.

Bu tasarım içerisinde, parametre değerleri kodlanmıştır. Örneğin, Mutasyon’un 4 alınması demek, aslında mutasyon değerinin 0.4 olduğunu göstermektedir. Aynı şekilde Döngü değerleri olan 1-2-3 değerleri de, 5-10-15 değerlerine tekabül etmektedir. Çizelge 9.1’de Cmax değeri 206.8375’dir. Bu değer altında ya da üstünde çıkan değerler o değer anlamlı olup olmadığını göstermektedir. Çizelge 9.1’e bakıldığında,

döngü[2]\* mutasyon[2]\* çaprazlama[4] (Döngü, 10, Mutasyon oranı 0.2 ve çaprazlama oranı ise 0.4) olarak alındığında, en büyük negatif “tahmin (estimate)” değerinin sağlandığı görülmektedir. Tahmin değerleri “sıfır” çıktığında ise o döngü, mutasyon oranı ve çaprazlama oranı üzerinde herhangi bir anlamlı farkın olmadığı görülmektedir.

Çizelge 9.1 4iř-2ařama-2makine için “genelleřtirilmiř lineer analiz” sonuřları

Term	Tahmin Deęerleri
Intercept	Cmax = 206.8375
dongu[1]*mutasyon[1]*caprazlama[1]	-0.826
dongu[1]*mutasyon[1]*caprazlama[2]	1.3822
dongu[1]*mutasyon[1]*caprazlama[3]	1.6542
dongu[1]*mutasyon[1]*caprazlama[4]	2.5648
dongu[1]*mutasyon[2]*caprazlama[1]	0.234
dongu[1]*mutasyon[2]*caprazlama[2]	0.1895
dongu[1]*mutasyon[2]*caprazlama[3]	3.6135
dongu[1]*mutasyon[2]*caprazlama[4]	-2.1115
dongu[1]*mutasyon[3]*caprazlama[1]	0.926
dongu[1]*mutasyon[3]*caprazlama[2]	0.725
dongu[1]*mutasyon[3]*caprazlama[3]	2.772
dongu[1]*mutasyon[3]*caprazlama[4]	-0.543
dongu[1]*mutasyon[4]*caprazlama[1]	-1.487
dongu[1]*mutasyon[4]*caprazlama[2]	0
dongu[1]*mutasyon[4]*caprazlama[3]	2.5702
dongu[1]*mutasyon[4]*caprazlama[4]	0.7415
dongu[2]*mutasyon[1]*caprazlama[1]	0
dongu[2]*mutasyon[1]*caprazlama[2]	-2.6872
dongu[2]*mutasyon[1]*caprazlama[3]	1.113
dongu[2]*mutasyon[1]*caprazlama[4]	2.786
dongu[2]*mutasyon[2]*caprazlama[1]	-1.2235
dongu[2]*mutasyon[2]*caprazlama[2]	2.286
dongu[2]*mutasyon[2]*caprazlama[3]	0.221
dongu[2]*mutasyon[2]*caprazlama[4] ****	***-5.308
dongu[2]*mutasyon[3]*caprazlama[1]	2.6603
dongu[2]*mutasyon[3]*caprazlama[2]	1.5102
dongu[2]*mutasyon[3]*caprazlama[3]	1.784
dongu[2]*mutasyon[3]*caprazlama[4]	0.536
dongu[2]*mutasyon[4]*caprazlama[1]	1.482
dongu[2]*mutasyon[4]*caprazlama[2]	0
dongu[2]*mutasyon[4]*caprazlama[3]	0.8815
dongu[2]*mutasyon[4]*caprazlama[4]	2.6323
dongu[3]*mutasyon[1]*caprazlama[1]	0.8139
dongu[3]*mutasyon[1]*caprazlama[2]	2.5133
dongu[3]*mutasyon[1]*caprazlama[3]	3.1415
dongu[3]*mutasyon[1]*caprazlama[4]	-0.116
dongu[3]*mutasyon[2]*caprazlama[1]	-0.756
dongu[3]*mutasyon[2]*caprazlama[2]	0
dongu[3]*mutasyon[2]*caprazlama[3]	0
dongu[3]*mutasyon[2]*caprazlama[4]	2.7544
dongu[3]*mutasyon[3]*caprazlama[1]	0.147
dongu[3]*mutasyon[3]*caprazlama[2]	-0.1278
dongu[3]*mutasyon[3]*caprazlama[3]	2.1135
dongu[3]*mutasyon[3]*caprazlama[4]	2.674
dongu[3]*mutasyon[4]*caprazlama[1]	0.536
dongu[3]*mutasyon[4]*caprazlama[2]	1.394
dongu[3]*mutasyon[4]*caprazlama[3]	-0.113
dongu[3]*mutasyon[4]*caprazlama[4]	0.3615

Kodlanan deęerler Çizelge 9.2’de “Genelleřtirilmiř Lineer Analiz ierisinde kodlanan deęerler” tablosu ierisinde grlmektedir.

Çizelge 9.2. Genelleştirilmiş Lineer Analiz içerisinde kodlanan değerler

<b>Döngü Değerleri</b>	<b>Mutasyon Oranları</b>	<b>Çaprazlama Oranları</b>
1 (5 döngü)	1 (0.1)	1 (0.1)
2 (10 döngü)	2 (0.2)	2 (0.2)
3 (15 döngü)	3 (0.3)	3 (0.3)
	4 (0.4)	4 (0.4)

Çalışma içerisinde amaç fonksiyonu,  $C_{max}$  minimizasyonudur. Bu nedenle “negatif” değerli olan “estimate” değerleri, amaç fonksiyonu olan  $C_{max}$  değeri üzerindeki anlamlı farklılıkların olduğunu ya da olacağını göstermektedir. 4iş-2aşama-4makine için çıkan “genelleştirilmiş lineer analiz” sonuçları ise Çizelge 9.3’ deki gibidir.

Çizelge 9.3 4iş-2aşama-4makine için “genelleştirilmiş lineer analiz” sonuçları

Term	Tahmin Değerleri
Intercept	156.207
dongu[1]*mutasyon[1]*caprazlama[1]	-0.8471
dongu[1]*mutasyon[1]*caprazlama[2]	0.2356
dongu[1]*mutasyon[1]*caprazlama[3]	-0.8621
dongu[1]*mutasyon[1]*caprazlama[4]	-0.3024
dongu[1]*mutasyon[2]*caprazlama[1]	0.9302
dongu[1]*mutasyon[2]*caprazlama[2]	3.1587
dongu[1]*mutasyon[2]*caprazlama[3]	2.794
dongu[1]*mutasyon[2]*caprazlama[4]	-0.0268
dongu[1]*mutasyon[3]*caprazlama[1]	5.587
dongu[1]*mutasyon[3]*caprazlama[2]	-5.1224
dongu[1]*mutasyon[3]*caprazlama[3]	4.5703
dongu[1]*mutasyon[3]*caprazlama[4]	-0.4231
dongu[1]*mutasyon[4]*caprazlama[1]	0
dongu[1]*mutasyon[4]*caprazlama[2]	4.223
dongu[1]*mutasyon[4]*caprazlama[3]	0
dongu[1]*mutasyon[4]*caprazlama[4]	-0.167
dongu[2]*mutasyon[1]*caprazlama[1]	-4.1568
dongu[2]*mutasyon[1]*caprazlama[2]	0.4329
dongu[2]*mutasyon[1]*caprazlama[3]	-0.9822
dongu[2]*mutasyon[1]*caprazlama[4]	0.2478
dongu[2]*mutasyon[2]*caprazlama[1]	0.798
dongu[2]*mutasyon[2]*caprazlama[2]	5.6853
dongu[2]*mutasyon[2]*caprazlama[3]	-0.318
dongu[2]*mutasyon[2]*caprazlama[4]	0.7631
dongu[2]*mutasyon[3]*caprazlama[1]	-0.4328
dongu[2]*mutasyon[3]*caprazlama[2]	3.4584
dongu[2]*mutasyon[3]*caprazlama[3]	1.7516
dongu[2]*mutasyon[3]*caprazlama[4]	-0.289
dongu[2]*mutasyon[4]*caprazlama[1]	4.875
dongu[2]*mutasyon[4]*caprazlama[2]	***-8.1215
dongu[2]*mutasyon[4]*caprazlama[3]	3.6507
dongu[2]*mutasyon[4]*caprazlama[4]	-0.7123
dongu[3]*mutasyon[1]*caprazlama[1]	0.5169
dongu[3]*mutasyon[1]*caprazlama[2]	6.7162
dongu[3]*mutasyon[1]*caprazlama[3]	0.6743
dongu[3]*mutasyon[1]*caprazlama[4]	0
dongu[3]*mutasyon[2]*caprazlama[1]	2.1926
dongu[3]*mutasyon[2]*caprazlama[2]	0.8423
dongu[3]*mutasyon[2]*caprazlama[3]	-0.6472
dongu[3]*mutasyon[2]*caprazlama[4]	0.4358
dongu[3]*mutasyon[3]*caprazlama[1]	-1.9626
dongu[3]*mutasyon[3]*caprazlama[2]	-0.3055
dongu[3]*mutasyon[3]*caprazlama[3]	0.9218
dongu[3]*mutasyon[3]*caprazlama[4]	2.1589
dongu[3]*mutasyon[4]*caprazlama[1]	2.894
dongu[3]*mutasyon[4]*caprazlama[2]	0
dongu[3]*mutasyon[4]*caprazlama[3]	4.586
dongu[3]*mutasyon[4]*caprazlama[4]	-5.3218

Çizelge 9.3’de de görüldüğü gibi, döngü[2]\* mutasyon[4]\* çaprazlama[2] (Döngü, 10, Mutasyon oranı 0.4 ve çaprazlama oranı ise 0.2) olarak alındığında, en büyük negatif “estimate” değerinin sağlandığı görülmektedir. Yani  $C_{max}$  üzerindeki en anlamlı fark bu parametre değerleri ile meydana gelmektedir.

6iş-3aşama-3makine için çıkan “genelleştirilmiş lineer analiz” sonuçları ise Çizelge 9.4’deki gibidir.

Çizelge 9.4. 6iş-3aşama-3makine için “genelleştirilmiş lineer analiz” sonuçları

Term	Tahmin Değerleri
Intercept	176.742
dongu[1]*mutasyon[1]*caprazlama[1]	-6.348
dongu[1]*mutasyon[1]*caprazlama[2]	***-7.912
dongu[1]*mutasyon[1]*caprazlama[3]	-6.4341
dongu[1]*mutasyon[1]*caprazlama[4]	1.325
dongu[1]*mutasyon[2]*caprazlama[1]	3.3192
dongu[1]*mutasyon[2]*caprazlama[2]	2.6772
dongu[1]*mutasyon[2]*caprazlama[3]	-0.4381
dongu[1]*mutasyon[2]*caprazlama[4]	-1.6523
dongu[1]*mutasyon[3]*caprazlama[1]	0
dongu[1]*mutasyon[3]*caprazlama[2]	3.478
dongu[1]*mutasyon[3]*caprazlama[3]	-0.6319
dongu[1]*mutasyon[3]*caprazlama[4]	-2.2677
dongu[1]*mutasyon[4]*caprazlama[1]	-1.435
dongu[1]*mutasyon[4]*caprazlama[2]	0.1236
dongu[1]*mutasyon[4]*caprazlama[3]	-1.321
dongu[1]*mutasyon[4]*caprazlama[4]	0
dongu[2]*mutasyon[1]*caprazlama[1]	0.2175
dongu[2]*mutasyon[1]*caprazlama[2]	4.1348
dongu[2]*mutasyon[1]*caprazlama[3]	-6.516
dongu[2]*mutasyon[1]*caprazlama[4]	5.3243
dongu[2]*mutasyon[2]*caprazlama[1]	-1.1255
dongu[2]*mutasyon[2]*caprazlama[2]	2.732
dongu[2]*mutasyon[2]*caprazlama[3]	-0.5281
dongu[2]*mutasyon[2]*caprazlama[4]	-1.4576
dongu[2]*mutasyon[3]*caprazlama[1]	-0.224
dongu[2]*mutasyon[3]*caprazlama[2]	4.577
dongu[2]*mutasyon[3]*caprazlama[3]	-1.8516
dongu[2]*mutasyon[3]*caprazlama[4]	-3.5679
dongu[2]*mutasyon[4]*caprazlama[1]	-4.433
dongu[2]*mutasyon[4]*caprazlama[2]	0
dongu[2]*mutasyon[4]*caprazlama[3]	-2.451
dongu[2]*mutasyon[4]*caprazlama[4]	2.2176
dongu[3]*mutasyon[1]*caprazlama[1]	4.246
dongu[3]*mutasyon[1]*caprazlama[2]	-5.719
dongu[3]*mutasyon[1]*caprazlama[3]	4.543
dongu[3]*mutasyon[1]*caprazlama[4]	-1.2754
dongu[3]*mutasyon[2]*caprazlama[1]	3.7324
dongu[3]*mutasyon[2]*caprazlama[2]	-0.858
dongu[3]*mutasyon[2]*caprazlama[3]	-3.4386
dongu[3]*mutasyon[2]*caprazlama[4]	-0.3414
dongu[3]*mutasyon[3]*caprazlama[1]	4.7897
dongu[3]*mutasyon[3]*caprazlama[2]	0
dongu[3]*mutasyon[3]*caprazlama[3]	-3.5857
dongu[3]*mutasyon[3]*caprazlama[4]	-4.5138
dongu[3]*mutasyon[4]*caprazlama[1]	0.322
dongu[3]*mutasyon[4]*caprazlama[2]	-5.4559
dongu[3]*mutasyon[4]*caprazlama[3]	1.4176
dongu[3]*mutasyon[4]*caprazlama[4]	5.1236

Çizelge 9.4’de de görüldüğü gibi, döngü[1]\* mutasyon[1]\* çaprazlama[2] (Döngü, 5, Mutasyon oranı 0.1 ve çaprazlama oranı ise 0.2) olarak alındığında, en büyük negatif “estimate” değerinin sağlandığı görülmektedir. Daha önce de bahsettiğimiz gibi,  $C_{max}$  üzerindeki en anlamlı fark bu parametre değerleri ile oluşmaktadır.

6iş-2aşama-3makine için çıkan “genelleştirilmiş lineer analiz” sonuçları ise Çizelge 9.5’deki gibidir.



Çizelge 9.5. 6iř-2ařama-3makine için “genelleřtirilmiř lineer analiz” sonuřları

Term	Tahmin Deęerleri
Intercept	230.42
dongu[1]*mutasyon[1]*caprazlama[1]	0.415
dongu[1]*mutasyon[1]*caprazlama[2]	-4.723
dongu[1]*mutasyon[1]*caprazlama[3]	2.517
dongu[1]*mutasyon[1]*caprazlama[4]	1.3887
dongu[1]*mutasyon[2]*caprazlama[1]	-1.4323
dongu[1]*mutasyon[2]*caprazlama[2]	5.2628
dongu[1]*mutasyon[2]*caprazlama[3]	0
dongu[1]*mutasyon[2]*caprazlama[4]	2.546
dongu[1]*mutasyon[3]*caprazlama[1]	2.561
dongu[1]*mutasyon[3]*caprazlama[2]	2.384
dongu[1]*mutasyon[3]*caprazlama[3]	-3.489
dongu[1]*mutasyon[3]*caprazlama[4]	1.2738
dongu[1]*mutasyon[4]*caprazlama[1]	-2.69
dongu[1]*mutasyon[4]*caprazlama[2]	3.145
dongu[1]*mutasyon[4]*caprazlama[3]	3.124
dongu[1]*mutasyon[4]*caprazlama[4]	-2.766
dongu[2]*mutasyon[1]*caprazlama[1]	1.1275
dongu[2]*mutasyon[1]*caprazlama[2]	-2.462
dongu[2]*mutasyon[1]*caprazlama[3]	0
dongu[2]*mutasyon[1]*caprazlama[4]	4.428
dongu[2]*mutasyon[2]*caprazlama[1]	-1.8454
dongu[2]*mutasyon[2]*caprazlama[2]	-2.243
dongu[2]*mutasyon[2]*caprazlama[3]	0.9352
dongu[2]*mutasyon[2]*caprazlama[4]	***-5.124
dongu[2]*mutasyon[3]*caprazlama[1]	-1.0389
dongu[2]*mutasyon[3]*caprazlama[2]	2.5451
dongu[2]*mutasyon[3]*caprazlama[3]	2.472
dongu[2]*mutasyon[3]*caprazlama[4]	3.858
dongu[2]*mutasyon[4]*caprazlama[1]	-3.125
dongu[2]*mutasyon[4]*caprazlama[2]	1.473
dongu[2]*mutasyon[4]*caprazlama[3]	-4.5672
dongu[2]*mutasyon[4]*caprazlama[4]	2.141
dongu[3]*mutasyon[1]*caprazlama[1]	1.1239
dongu[3]*mutasyon[1]*caprazlama[2]	-1.856
dongu[3]*mutasyon[1]*caprazlama[3]	3.2789
dongu[3]*mutasyon[1]*caprazlama[4]	-2.4638
dongu[3]*mutasyon[2]*caprazlama[1]	0
dongu[3]*mutasyon[2]*caprazlama[2]	4.342
dongu[3]*mutasyon[2]*caprazlama[3]	-2.3845
dongu[3]*mutasyon[2]*caprazlama[4]	2.437
dongu[3]*mutasyon[3]*caprazlama[1]	0.535
dongu[3]*mutasyon[3]*caprazlama[2]	4.1262
dongu[3]*mutasyon[3]*caprazlama[3]	0
dongu[3]*mutasyon[3]*caprazlama[4]	2.145
dongu[3]*mutasyon[4]*caprazlama[1]	1.561
dongu[3]*mutasyon[4]*caprazlama[2]	3.384
dongu[3]*mutasyon[4]*caprazlama[3]	-4.4576
dongu[3]*mutasyon[4]*caprazlama[4]	1.7324

Çizelge 9.5’te görüldüęü gibi, döngü[2]\* mutasyon[2]\* çaprazlama[4] (Döngü, 10, Mutasyon oranı 0.2 ve çaprazlama oranı ise 0.4) olarak alındığında, en büyük negatif “estimate” deęerinin saęlandıęı görülmektedir.  $C_{max}$  üzerindeki en anlamlı fark bu parametre deęerleri ile oluřmaktadır.

8iř-3ařama-4makine iin ıkan “genelleřtirilmiř lineer analiz” sonuları ise izelge 9.6’ daki gibidir.

izelge 9.6. 8iř-3ařama-4makine iin “genelleřtirilmiř lineer analiz” sonuları

Term	Tahmin Deęerleri
Intercept	300.28
dongu[1]*mutasyon[1]*caprazlama[1]	-1.579
dongu[1]*mutasyon[1]*caprazlama[2]	0
dongu[1]*mutasyon[1]*caprazlama[3]	2.581
dongu[1]*mutasyon[1]*caprazlama[4]	-1.493
dongu[1]*mutasyon[2]*caprazlama[1]	2.308
dongu[1]*mutasyon[2]*caprazlama[2]	-1.387
dongu[1]*mutasyon[2]*caprazlama[3]	1.424
dongu[1]*mutasyon[2]*caprazlama[4]	2.124
dongu[1]*mutasyon[3]*caprazlama[1]	0.275
dongu[1]*mutasyon[3]*caprazlama[2]	-0.523
dongu[1]*mutasyon[3]*caprazlama[3]	3.073
dongu[1]*mutasyon[3]*caprazlama[4]	***-5.527
dongu[1]*mutasyon[4]*caprazlama[1]	1.238
dongu[1]*mutasyon[4]*caprazlama[2]	-2.837
dongu[1]*mutasyon[4]*caprazlama[3]	-4.49
dongu[1]*mutasyon[4]*caprazlama[4]	-3.534
dongu[2]*mutasyon[1]*caprazlama[1]	1.265
dongu[2]*mutasyon[1]*caprazlama[2]	-2.341
dongu[2]*mutasyon[1]*caprazlama[3]	3.987
dongu[2]*mutasyon[1]*caprazlama[4]	1.352
dongu[2]*mutasyon[2]*caprazlama[1]	0.631
dongu[2]*mutasyon[2]*caprazlama[2]	0.325
dongu[2]*mutasyon[2]*caprazlama[3]	1.123
dongu[2]*mutasyon[2]*caprazlama[4]	-2.072
dongu[2]*mutasyon[3]*caprazlama[1]	-1.457
dongu[2]*mutasyon[3]*caprazlama[2]	0
dongu[2]*mutasyon[3]*caprazlama[3]	2.158
dongu[2]*mutasyon[3]*caprazlama[4]	-1.749
dongu[2]*mutasyon[4]*caprazlama[1]	2.93
dongu[2]*mutasyon[4]*caprazlama[2]	-1.2238
dongu[2]*mutasyon[4]*caprazlama[3]	1.742
dongu[2]*mutasyon[4]*caprazlama[4]	2.861
dongu[3]*mutasyon[1]*caprazlama[1]	0.527
dongu[3]*mutasyon[1]*caprazlama[2]	-1.152
dongu[3]*mutasyon[1]*caprazlama[3]	3.237
dongu[3]*mutasyon[1]*caprazlama[4]	3.12
dongu[3]*mutasyon[2]*caprazlama[1]	2.489
dongu[3]*mutasyon[2]*caprazlama[2]	-3.131
dongu[3]*mutasyon[2]*caprazlama[3]	-4.522
dongu[3]*mutasyon[2]*caprazlama[4]	-4.143
dongu[3]*mutasyon[3]*caprazlama[1]	1.335
dongu[3]*mutasyon[3]*caprazlama[2]	-1.1541
dongu[3]*mutasyon[3]*caprazlama[3]	-3.871
dongu[3]*mutasyon[3]*caprazlama[4]	0
dongu[3]*mutasyon[4]*caprazlama[1]	0
dongu[3]*mutasyon[4]*caprazlama[2]	0.127
dongu[3]*mutasyon[4]*caprazlama[3]	1.2423
dongu[3]*mutasyon[4]*caprazlama[4]	-0.537

izelge 9.6’da grldę gibi, dng[1]\* mutasyon[3]\* aprazlama[4] (Dng 5, Mutasyon oranı 0.3 ve aprazlama oranı ise 0.4) olarak alındıęında, en byk negatif “estimate” deęerinin saęlandıęı grlmektedir.  $C_{max}$  zerindeki en anlamlı fark bu parametre deęerleri ile oluřmaktadır. Fark deęeri ise, -5.527 olarak kaydedilmektedir.

8iř-4ařama-4makine iin ıkan “genelleřtirilmiř lineer analiz” sonuları ise izelge 9.7’deki gibidir.

izelge 9.7. 8iř-4ařama-4makine iin “genelleřtirilmiř lineer analiz” sonuları

Term	Tahmin Deęerleri
Intercept	290.591
dongu[1]*mutasyon[1]*caprazlama[1]	5.644
dongu[1]*mutasyon[1]*caprazlama[2]	-0.5578
dongu[1]*mutasyon[1]*caprazlama[3]	-0.211
dongu[1]*mutasyon[1]*caprazlama[4]	-2.757
dongu[1]*mutasyon[2]*caprazlama[1]	-2.389
dongu[1]*mutasyon[2]*caprazlama[2]	-1.532
dongu[1]*mutasyon[2]*caprazlama[3]	-6.835
dongu[1]*mutasyon[2]*caprazlama[4]	0.818
dongu[1]*mutasyon[3]*caprazlama[1]	7.238
dongu[1]*mutasyon[3]*caprazlama[2]	-0.912
dongu[1]*mutasyon[3]*caprazlama[3]	1.267
dongu[1]*mutasyon[3]*caprazlama[4]	-2.338
dongu[1]*mutasyon[4]*caprazlama[1]	2.135
dongu[1]*mutasyon[4]*caprazlama[2]	-0.445
dongu[1]*mutasyon[4]*caprazlama[3]	-1.887
dongu[1]*mutasyon[4]*caprazlama[4]	0
dongu[2]*mutasyon[1]*caprazlama[1]	5.984
dongu[2]*mutasyon[1]*caprazlama[2]	-0.127
dongu[2]*mutasyon[1]*caprazlama[3]	-2.451
dongu[2]*mutasyon[1]*caprazlama[4]	-2.686
dongu[2]*mutasyon[2]*caprazlama[1]	0
dongu[2]*mutasyon[2]*caprazlama[2]	5.564
dongu[2]*mutasyon[2]*caprazlama[3]	-1.378
dongu[2]*mutasyon[2]*caprazlama[4]	-0.811
dongu[2]*mutasyon[3]*caprazlama[1]	-2.457
dongu[2]*mutasyon[3]*caprazlama[2]	-2.679
dongu[2]*mutasyon[3]*caprazlama[3]	0.551
dongu[2]*mutasyon[3]*caprazlama[4]	*** -7.123
dongu[2]*mutasyon[4]*caprazlama[1]	-0.891
dongu[2]*mutasyon[4]*caprazlama[2]	1.236
dongu[2]*mutasyon[4]*caprazlama[3]	-2.33
dongu[2]*mutasyon[4]*caprazlama[4]	2.913
dongu[3]*mutasyon[1]*caprazlama[1]	0
dongu[3]*mutasyon[1]*caprazlama[2]	-1.818
dongu[3]*mutasyon[1]*caprazlama[3]	-0.445
dongu[3]*mutasyon[1]*caprazlama[4]	4.128
dongu[3]*mutasyon[2]*caprazlama[1]	-0.812
dongu[3]*mutasyon[2]*caprazlama[2]	-3.445
dongu[3]*mutasyon[2]*caprazlama[3]	-1.438
dongu[3]*mutasyon[2]*caprazlama[4]	-0.143
dongu[3]*mutasyon[3]*caprazlama[1]	4.336
dongu[3]*mutasyon[3]*caprazlama[2]	-1.257
dongu[3]*mutasyon[3]*caprazlama[3]	0
dongu[3]*mutasyon[3]*caprazlama[4]	-2.435
dongu[3]*mutasyon[4]*caprazlama[1]	2.567
dongu[3]*mutasyon[4]*caprazlama[2]	2.431
dongu[3]*mutasyon[4]*caprazlama[3]	-3.123
dongu[3]*mutasyon[4]*caprazlama[4]	0.231

Yukarıda izelge 9.7’de grldę gibi, dng[2]\* mutasyon[3]\* aprazlama[4] (Dng 10, Mutasyon oranı 0.3 ve aprazlama oranı ise 0.4) olarak alındıęında, en byk negatif “estimate” deęerinin saęlandıęı grlmektedir.  $C_{max}$  zerindeki en anlamlı fark bu parametre deęerleri ile oluřmaktadır. Burada ise, fark deęeri, -7.123 olarak

kaydedilmektedir. 9iř-3ařama-3makine için çıkan “genelleřtirilmiř lineer analiz” sonuçları ise izelge 9.8’deki gibidir.

izelge 9.8. 9iř-3ařama-3makine için “genelleřtirilmiř lineer analiz” sonuçları

Term	Tahmin Deęerleri
Intercept	249.4185
dongu[1]*mutasyon[1]*caprazlama[1]	-3.548
dongu[1]*mutasyon[1]*caprazlama[2]	5.2149
dongu[1]*mutasyon[1]*caprazlama[3]	-2.992
dongu[1]*mutasyon[1]*caprazlama[4]	-4.129
dongu[1]*mutasyon[2]*caprazlama[1]	-1.3354
dongu[1]*mutasyon[2]*caprazlama[2]	*** -7.4332
dongu[1]*mutasyon[2]*caprazlama[3]	2.303
dongu[1]*mutasyon[2]*caprazlama[4]	0
dongu[1]*mutasyon[3]*caprazlama[1]	0.2375
dongu[1]*mutasyon[3]*caprazlama[2]	0.192
dongu[1]*mutasyon[3]*caprazlama[3]	0.2167
dongu[1]*mutasyon[3]*caprazlama[4]	0.232
dongu[1]*mutasyon[4]*caprazlama[1]	0.109
dongu[1]*mutasyon[4]*caprazlama[2]	0.6762
dongu[1]*mutasyon[4]*caprazlama[3]	1.812
dongu[1]*mutasyon[4]*caprazlama[4]	2.541
dongu[2]*mutasyon[1]*caprazlama[1]	0
dongu[2]*mutasyon[1]*caprazlama[2]	-3.982
dongu[2]*mutasyon[1]*caprazlama[3]	4.278
dongu[2]*mutasyon[1]*caprazlama[4]	-2.281
dongu[2]*mutasyon[2]*caprazlama[1]	-2.334
dongu[2]*mutasyon[2]*caprazlama[2]	-1.459
dongu[2]*mutasyon[2]*caprazlama[3]	3.243
dongu[2]*mutasyon[2]*caprazlama[4]	-0.662
dongu[2]*mutasyon[3]*caprazlama[1]	-1.7742
dongu[2]*mutasyon[3]*caprazlama[2]	1.328
dongu[2]*mutasyon[3]*caprazlama[3]	-4.998
dongu[2]*mutasyon[3]*caprazlama[4]	2.549
dongu[2]*mutasyon[4]*caprazlama[1]	-1.389
dongu[2]*mutasyon[4]*caprazlama[2]	-2.322
dongu[2]*mutasyon[4]*caprazlama[3]	-2.534
dongu[2]*mutasyon[4]*caprazlama[4]	-1.378
dongu[3]*mutasyon[1]*caprazlama[1]	1.303
dongu[3]*mutasyon[1]*caprazlama[2]	1.67
dongu[3]*mutasyon[1]*caprazlama[3]	0.374
dongu[3]*mutasyon[1]*caprazlama[4]	0
dongu[3]*mutasyon[2]*caprazlama[1]	0
dongu[3]*mutasyon[2]*caprazlama[2]	1.134
dongu[3]*mutasyon[2]*caprazlama[3]	1.341
dongu[3]*mutasyon[2]*caprazlama[4]	2.295
dongu[3]*mutasyon[3]*caprazlama[1]	-3.6278
dongu[3]*mutasyon[3]*caprazlama[2]	4.7832
dongu[3]*mutasyon[3]*caprazlama[3]	-2.815
dongu[3]*mutasyon[3]*caprazlama[4]	-1.549
dongu[3]*mutasyon[4]*caprazlama[1]	-2.875
dongu[3]*mutasyon[4]*caprazlama[2]	2.307
dongu[3]*mutasyon[4]*caprazlama[3]	-1.221
dongu[3]*mutasyon[4]*caprazlama[4]	0.422

izelge 9.8’de grldę gibi, dng[1]\* mutasyon[2]\* aprazlama[2] (Dng 5, Mutasyon oranı 0.2 ve aprazlama oranı ise 0.2) olarak alındıęında, en byk negatif “estimate” deęerinin saęlandıęı grlmektedir.  $C_{max}$  zerindeki en anlamlı fark bu parametre deęerleri ile oluřmaktadır. Burada ise, fark deęeri, -7.4332 olarak

kaydedilmektedir. 10iř-3ařama-5makine için genelleřtirilmiř analiz sonuları izelge 9.9’daki gibidir.

izelge 9.9. 10iř-3ařama-5makine için “genelleřtirilmiř lineer analiz” sonuları

Term	Tahmin Deęerleri
Intercept	240.549
dongu[1]*mutasyon[1]*caprazlama[1]	-1.125
dongu[1]*mutasyon[1]*caprazlama[2]	-2.896
dongu[1]*mutasyon[1]*caprazlama[3]	2.143
dongu[1]*mutasyon[1]*caprazlama[4]	1.236
dongu[1]*mutasyon[2]*caprazlama[1]	0.241
dongu[1]*mutasyon[2]*caprazlama[2]	2.314
dongu[1]*mutasyon[2]*caprazlama[3]	-2.297
dongu[1]*mutasyon[2]*caprazlama[4]	5.636
dongu[1]*mutasyon[3]*caprazlama[1]	0.138
dongu[1]*mutasyon[3]*caprazlama[2]	*** -6.745
dongu[1]*mutasyon[3]*caprazlama[3]	0.37
dongu[1]*mutasyon[3]*caprazlama[4]	-2.218
dongu[1]*mutasyon[4]*caprazlama[1]	0.384
dongu[1]*mutasyon[4]*caprazlama[2]	-4.224
dongu[1]*mutasyon[4]*caprazlama[3]	0
dongu[1]*mutasyon[4]*caprazlama[4]	-3.126
dongu[2]*mutasyon[1]*caprazlama[1]	-0.275
dongu[2]*mutasyon[1]*caprazlama[2]	3.252
dongu[2]*mutasyon[1]*caprazlama[3]	0
dongu[2]*mutasyon[1]*caprazlama[4]	-5.240
dongu[2]*mutasyon[2]*caprazlama[1]	0.875
dongu[2]*mutasyon[2]*caprazlama[2]	-3.1226
dongu[2]*mutasyon[2]*caprazlama[3]	-1.275
dongu[2]*mutasyon[2]*caprazlama[4]	3.552
dongu[2]*mutasyon[3]*caprazlama[1]	-0.786
dongu[2]*mutasyon[3]*caprazlama[2]	8.79
dongu[2]*mutasyon[3]*caprazlama[3]	-3.053
dongu[2]*mutasyon[3]*caprazlama[4]	1.127
dongu[2]*mutasyon[4]*caprazlama[1]	-3.312
dongu[2]*mutasyon[4]*caprazlama[2]	0.265
dongu[2]*mutasyon[4]*caprazlama[3]	-0.213
dongu[2]*mutasyon[4]*caprazlama[4]	-4.837
dongu[3]*mutasyon[1]*caprazlama[1]	8.525
dongu[3]*mutasyon[1]*caprazlama[2]	-0.2325
dongu[3]*mutasyon[1]*caprazlama[3]	-2.136
dongu[3]*mutasyon[1]*caprazlama[4]	0.143
dongu[3]*mutasyon[2]*caprazlama[1]	1.456
dongu[3]*mutasyon[2]*caprazlama[2]	0.341
dongu[3]*mutasyon[2]*caprazlama[3]	2.114
dongu[3]*mutasyon[2]*caprazlama[4]	-2.297
dongu[3]*mutasyon[3]*caprazlama[1]	5.236
dongu[3]*mutasyon[3]*caprazlama[2]	0.038
dongu[3]*mutasyon[3]*caprazlama[3]	-4.245
dongu[3]*mutasyon[3]*caprazlama[4]	1.477
dongu[3]*mutasyon[4]*caprazlama[1]	-4.218
dongu[3]*mutasyon[4]*caprazlama[2]	0.284
dongu[3]*mutasyon[4]*caprazlama[3]	-5.224
dongu[3]*mutasyon[4]*caprazlama[4]	0.755

izelge 9.9’da grldę gibi, dng[1]\* mutasyon[3]\* aprazlama[2] (Dng 5, Mutasyon oranı 0.3 ve aprazlama oranı ise 0.3) olarak alındıęında, en byk negatif “estimate” deęerinin saęlandıęı grlmektedir.  $C_{max}$  zerindeki en anlamlı fark bu parametre deęerleri ile oluřmaktadır. Burada ise, fark deęeri, -6.745 olarak

kaydedilmektedir. 15iř-2ařama-5makine için genelleřtirilmiř analiz sonuları izelge 9.10'daki gibidir.

izelge 9.10. 15iř-2ařama-5makine için "genelleřtirilmiř lineer analiz" sonuları

Term	Tahmin Deęerleri
Intercept	271.2475
dongu[1]*mutasyon[1]*caprazlama[1]	4.173
dongu[1]*mutasyon[1]*caprazlama[2]	-0.229
dongu[1]*mutasyon[1]*caprazlama[3]	-1.656
dongu[1]*mutasyon[1]*caprazlama[4]	0.867
dongu[1]*mutasyon[2]*caprazlama[1]	2.952
dongu[1]*mutasyon[2]*caprazlama[2]	-0.454
dongu[1]*mutasyon[2]*caprazlama[3]	-1.297
dongu[1]*mutasyon[2]*caprazlama[4]	-2.43
dongu[1]*mutasyon[3]*caprazlama[1]	3.894
dongu[1]*mutasyon[3]*caprazlama[2]	-1.236
dongu[1]*mutasyon[3]*caprazlama[3]	-2.386
dongu[1]*mutasyon[3]*caprazlama[4]	4.672
dongu[1]*mutasyon[4]*caprazlama[1]	-5.94
dongu[1]*mutasyon[4]*caprazlama[2]	2.06
dongu[1]*mutasyon[4]*caprazlama[3]	0
dongu[1]*mutasyon[4]*caprazlama[4]	2.183
dongu[2]*mutasyon[1]*caprazlama[1]	4.273
dongu[2]*mutasyon[1]*caprazlama[2]	0
dongu[2]*mutasyon[1]*caprazlama[3]	-1.656
dongu[2]*mutasyon[1]*caprazlama[4]	0.2867
dongu[2]*mutasyon[2]*caprazlama[1]	2.152
dongu[2]*mutasyon[2]*caprazlama[2]	0
dongu[2]*mutasyon[2]*caprazlama[3]	-0.797
dongu[2]*mutasyon[2]*caprazlama[4]	-4.13
dongu[2]*mutasyon[3]*caprazlama[1]	3.765
dongu[2]*mutasyon[3]*caprazlama[2]	-1.775
dongu[2]*mutasyon[3]*caprazlama[3]	-1.96
dongu[2]*mutasyon[3]*caprazlama[4]	4.324
dongu[2]*mutasyon[4]*caprazlama[1]	*** -6.294
dongu[2]*mutasyon[4]*caprazlama[2]	3.486
dongu[2]*mutasyon[4]*caprazlama[3]	0.144
dongu[2]*mutasyon[4]*caprazlama[4]	2.883
dongu[3]*mutasyon[1]*caprazlama[1]	4.573
dongu[3]*mutasyon[1]*caprazlama[2]	-0.195
dongu[3]*mutasyon[1]*caprazlama[3]	-1.256
dongu[3]*mutasyon[1]*caprazlama[4]	0
dongu[3]*mutasyon[2]*caprazlama[1]	2.452
dongu[3]*mutasyon[2]*caprazlama[2]	-0.754
dongu[3]*mutasyon[2]*caprazlama[3]	-1.2097
dongu[3]*mutasyon[2]*caprazlama[4]	-2.513
dongu[3]*mutasyon[3]*caprazlama[1]	3.245
dongu[3]*mutasyon[3]*caprazlama[2]	-1.116
dongu[3]*mutasyon[3]*caprazlama[3]	-2.26
dongu[3]*mutasyon[3]*caprazlama[4]	4.372
dongu[3]*mutasyon[4]*caprazlama[1]	-5.294
dongu[3]*mutasyon[4]*caprazlama[2]	2.906
dongu[3]*mutasyon[4]*caprazlama[3]	0.144
dongu[3]*mutasyon[4]*caprazlama[4]	2.5583

izelge 9.10'da grldęi gibi, dng[2]\* mutasyon[4]\* aprazlama[1] (Dng 10, Mutasyon oranı 0.4 ve aprazlama oranı ise 0.1) olarak alındıęında, en byk negatif "estimate" deęerinin saęlandıęı grlmektedir.  $C_{max}$  zerindeki en anlamlı fark bu parametre deęerleri ile oluřmaktadır. Burada ise, fark deęeri, -6.294 olarak

kaydedilmektedir. 18iř-4ařama-6makine için genelleřtirilmiř analiz sonuları izelge 9.11'deki gibidir.

izelge 9.11. 18iř-4ařama-6makine için "genelleřtirilmiř lineer analiz" sonuları

Term	Tahmin Deęerleri
Intercept	276.847
dongu[1]*mutasyon[1]*caprazlama[1]	-1.113
dongu[1]*mutasyon[1]*caprazlama[2]	-1.453
dongu[1]*mutasyon[1]*caprazlama[3]	0.1016
dongu[1]*mutasyon[1]*caprazlama[4]	-4.827
dongu[1]*mutasyon[2]*caprazlama[1]	-2.2276
dongu[1]*mutasyon[2]*caprazlama[2]	1.232
dongu[1]*mutasyon[2]*caprazlama[3]	0
dongu[1]*mutasyon[2]*caprazlama[4]	-1.266
dongu[1]*mutasyon[3]*caprazlama[1]	-0.986
dongu[1]*mutasyon[3]*caprazlama[2]	-0.1322
dongu[1]*mutasyon[3]*caprazlama[3]	-1.437
dongu[1]*mutasyon[3]*caprazlama[4]	-2.786
dongu[1]*mutasyon[4]*caprazlama[1]	-0.5622
dongu[1]*mutasyon[4]*caprazlama[2]	-1.137
dongu[1]*mutasyon[4]*caprazlama[3]	-0.931
dongu[1]*mutasyon[4]*caprazlama[4]	1.2295
dongu[2]*mutasyon[1]*caprazlama[1]	-2.902
dongu[2]*mutasyon[1]*caprazlama[2]	-1.059
dongu[2]*mutasyon[1]*caprazlama[3]	0
dongu[2]*mutasyon[1]*caprazlama[4]	-2.135
dongu[2]*mutasyon[2]*caprazlama[1]	0.2353
dongu[2]*mutasyon[2]*caprazlama[2]	-1.015
dongu[2]*mutasyon[2]*caprazlama[3]	4.4415
dongu[2]*mutasyon[2]*caprazlama[4]	-2.2347
dongu[2]*mutasyon[3]*caprazlama[1]	-3.2701
dongu[2]*mutasyon[3]*caprazlama[2]	2.2247
dongu[2]*mutasyon[3]*caprazlama[3]	-1.4453
dongu[2]*mutasyon[3]*caprazlama[4]	3.455
dongu[2]*mutasyon[4]*caprazlama[1]	2.901
dongu[2]*mutasyon[4]*caprazlama[2]	*** -5.867
dongu[2]*mutasyon[4]*caprazlama[3]	-2.3076
dongu[2]*mutasyon[4]*caprazlama[4]	2.8732
dongu[3]*mutasyon[1]*caprazlama[1]	0
dongu[3]*mutasyon[1]*caprazlama[2]	-1.2266
dongu[3]*mutasyon[1]*caprazlama[3]	-0.286
dongu[3]*mutasyon[1]*caprazlama[4]	1.982
dongu[3]*mutasyon[2]*caprazlama[1]	-1.0037
dongu[3]*mutasyon[2]*caprazlama[2]	0.611
dongu[3]*mutasyon[2]*caprazlama[3]	1.885
dongu[3]*mutasyon[2]*caprazlama[4]	-0.279
dongu[3]*mutasyon[3]*caprazlama[1]	1.872
dongu[3]*mutasyon[3]*caprazlama[2]	0.2343
dongu[3]*mutasyon[3]*caprazlama[3]	-1.3435
dongu[3]*mutasyon[3]*caprazlama[4]	0.453
dongu[3]*mutasyon[4]*caprazlama[1]	1.115
dongu[3]*mutasyon[4]*caprazlama[2]	4.261
dongu[3]*mutasyon[4]*caprazlama[3]	-0.094
dongu[3]*mutasyon[4]*caprazlama[4]	-4.706

izelge 9.11'de grldęi gibi, dng[2]\* mutasyon[4]\* aprazlama[2] (Dng 10, Mutasyon oranı 0.4 ve aprazlama oranı ise 0.2) olarak alındıęında, en byk negatif "tahmin" deęerinin saęlandıęı grlmektedir.  $C_{max}$  zerindeki en anlamlı fark bu parametre deęerleri ile oluřmaktadır. Burada ise, fark deęeri, -5.867 olarak kaydedilmektedir.

Sonuçta, deneysel tasarıma tabi tutulan bu örnekler üzerindeki en anlamlı farklılıklar çizelge 9.12'deki gibi olmaktadır.

Çizelge 9.12 Örnekler üzerindeki dik tasarım sonuçları

<b>İş/ aşama/ makine</b>	<b>Döngü/ Mutasyon Oranı/ Çaprazlama Oranı</b>	<b>Anlamlı Farklılıklar</b>
4iş-2aşama-2makine	10, 0.2, 0.4	-5.308
4iş-2aşama-4makine	10, 0.4, 0.2	-8.1215
6iş-3aşama-3makine	5, 0.1, 0.2	-7.912
6iş-2aşama-3makine	10, 0.2, 0.4	-5.124
8iş-3aşama-4makine	5, 0.3, 0.4	-5.527
8iş-4aşama-4makine	10, 0.3, 0.4	-7.123
9iş-3aşama-3makine	5, 0.2, 0.2	-7.4332
10iş-3aşama-5makine	5, 0.3, 0.3	-6.745
15iş-2aşama-5makine	10, 0.4, 0.1	-6.294
18iş-4aşama-6makine	10, 0.4, 0.2	-5.867



## BÖLÜM 10

---

### SONUÇLAR VE ÖNERİLER

Sıralamanın amaçları firmadan firmaya ve günden güne değişmektedir. İş sıralama ve çizelgeleme içerisinde amaç mümkün olduğu kadar kısa zamanda işleri bitirmek ve/veya işi belirli kontrat tarihleri içerisinde başarmış olmaktır.

Yapılan literatür taraması sonucunda görülmüştür ki, çizelgeleme problemleri sıklıkla çok tezgahlı durumlarla da ilgilenmektedir. Bu tip sistemlerde, tezgahlar paralel ya da seri konumda olabilmektedirler. Paralel tezgahlı ortam, tek tezgah ortamının bir genellemesi olup, esnek akış tipi çizelgeleme ortamının spesifik bir hali olarak incelenmektedir.

Bu çalışma içerisinde, öncelikle “iş sıralama” problemleri içerisinde önemli ve görece olarak daha az ilgilenilen bir konu olan “Hibrit Akış Tipi (HFS)” sistemler incelenmiş ve ardından bu konu üzerinde bir model düşünülmüştür. HFS yapısı içerisinde her aşamada ise, literatürde bu tip sistemler içinde oldukça az yer bulan “ilişkisiz paralel makineler” kullanılmıştır.

Hibrit Akış Tipi, genel olarak, verilen bir amaç fonksiyonunu en iyileyen seri m aşamalı durumda prosese giren bir dizi n işin sıralanması durumudur. HFS sıralama problemleri hem akış tipi hem de paralel makine sıralama problemlerini içeren, özel bir yapıyı arz etmektedirler.

Yapılan modelleme prosesinde görülmüştür ki, örnek boyutu büyüdükçe, model yavaşlamaktadır ve çözümünü de zorlaşmaktadır. Sıralama problemleri zaten doğası gereği oldukça zor yapıda olan ve çözümünü de komplike bir yapı arz eden problemlerdir. Bu nedenle, modellemenin ardından modelin çözümüne yardımcı olması açısından, C sharp programı üzerinde yazılıp kodlanan “metasezgisel” bir teknik denenmiştir. Çizelgeleme Literatüründe, Metasezgisel çalışmalar içerisinde en çok kullanılan yöntemlerden birisi olan “genetik algoritma” uygulanmıştır. Metasezgiseller, daha önce de söylendiği gibi; yakınsama (approximate–tahmini/yakınsayan-) metodlarının bir sınıfıdır. Bunlar, klasik sezgisellerin etkili ve verimli olmadığı durumlarda “zor” birleşik eniyileme problemlerine hamle edecek şekilde tasarlanmışlardır.

Genetik Algoritmalar, en iyinin korunumu ve doğal seçim ilkesinin benzetim yoluyla bilgisayarlara uygulanması ile elde edilen bir arama yöntemidir. Genetik algoritma yöntemi içerisinde, en başta ise, yöntemine her anlamda yardımcı olması adına “baskınlık özellikleri (Dominance Properties-DP)” isimli ve sıklıkla az denenilen bir yöntem eklenmiştir.

Verilen bir sıralama işlemi için “baskınlık özellikleri” kullanılarak, optimale yakın bir çözüm elde edilebilmektedir. Daha önce de söylendiği gibi, büyük boyutlu problemlerin çözümü zor olmaktadır. Bunun için, çözüm kalitesini arttırmaya yardımcı olarak, baskınlık özelliklerinin genetik algoritma ile birleştirildiği, yeni bir meta sezgisel ortaya konulabileceği bu çalışma içerisinde verilen en temel ve en önemli noktalardan birisi olmaktadır.

Sonuçta, Genetik algoritmanın kullanılabilirliği ile etkinliğinin daha da artırılması adına, “üstünlük/baskınlık özellikleri” metodolojiye eklenerek, oldukça başarılı sonuçlar elde edilmiştir. Görülmüştür ki Üstünlük Özellikleri ile birlikte kullanılan Genetik Algoritma yöntemi Hibrit Akış Tipi bir sistemde oldukça iyi ve başarılı sonuçlar vermektedir.

İlerleyen senelerde, çalışmayı geliştirmek adına, farklı meta sezgisel ya da sezgisel yöntemlerle “baskınlık özellikleri” birleştirilebilir. Ya da HFS sistemi içerisinde

baskınlık özelliklerinin eklendiđi ve bu şekilde denenen bir meta sezgisel yöntem içerisinde, farklı amaç fonksiyonları ele alınabilir. Ayrıca diđer meta sezgisel yöntemler de bu tip bir HFS sistemine dahil edilebilir.

## KAYNAKLAR

---

- [1] Kravchenko, S.A. ve Werner, F., (1997). "Parallel machine scheduling problems with a single server", *Mathematical Computational Modelling*, 26:111.
- [2] Park M. ve Kim Y., (1997). "Search Heuristics for a Parallel Machine Scheduling Problem with Ready Times and Due Dates", *Computers ind. Engng*, 33: 2-3-4.
- [3] Torres A.J.R., Ensore E.E. ve Barton R.R., (1997). "Simulated annealing heuristics for the average flow-time and the number of tardy jobs bi-criteria identical parallel machine problem", 33: 257-260.
- [4] Chiu N.C., Fang S.C. ve Lee Y.S., (1999). "Sequencing parallel machining operations by genetic algorithms", *Computers & Industrial Engineering* 36: 259-280.
- [5] Min L. ve Cheng W., (1999). "A genetic algorithm for minimizing the makespan in the case of scheduling identical parallel machines", *Artificial Intelligence in Engineering* 13: 399-403.
- [6] Gupta J.N.D. ve Torres A.J.R., (2000). "Minimizing makespan subject to minimum total flow-time on identical parallel machines", *European Journal of Operational Research*, 125: 370-380.
- [7] Laguna M., Lino P., Perez A., Quintanilla S. ve Valls V., (2000). "Minimizing weighted tardiness of jobs with stochastic interruptions in parallel machines", *European Journal of Operational Research*, 127, 1.

- [8] Gupta J.N.D. ve Ho J.C., (2001). "Minimizing makespan subject to minimum flowtime on two identical parallel machines", *Computers & Operations Research*, 28: 1-4.
- [9] Bank J. ve Werner F., (2001). "Heuristic Algorithms for Unrelated Parallel Machine Scheduling with a Common Due Date, Release Dates, and Linear Earliness and Tardiness Penalties", *Mathematical and Computer Modelling*, 33: 14-19.
- [10] Mokotoff E. ve Chretienne P., (2002). "A cutting plane algorithm for the unrelated parallel machine scheduling problem", *European Journal of Operational Research*, 141-345.
- [11] Liaw C.F., (2003). An efficient tabu search approach for the two machine preemptive open shop scheduling problem. *Computers & Operations Research* 30, 14: 2081-2095.
- [12] Alagöz O. ve Azizoğlu M., (2002). "Rescheduling of identical parallel machines under machine eligibility constraints", *European Journal of Operational Research*, 149: 2.
- [13] Meyr, H., (2002). "Simultaneous lot sizing and scheduling on parallel machines", *European Journal of Operational Research*, 139, 1.
- [14] Cochran, J. K., Horng S.M. ve Fowler J.W., (2003). "A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines", *Computers & Operations Research* 30: 1087–1102.
- [15] Liu M. ve Wu C., (2003). "Scheduling algorithm based on evolutionary computing in identical parallel machine production line", *Robotics and Computer Integrated Manufacturing*, 19: 6-7.
- [16] Liao C.J. ve Lin C.H., (2003). "Makespan minimization for two uniform parallel machines. *Int. J. Production Economics*", 84, 4.
- [17] Peng J. ve Liu B., (2004). "Parallel machine scheduling models with fuzzy processing times", *Information Sciences*, 166, 1.
- [18] Chen J.F. ve Wu T.H., (2006). "Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints", *Omega*, 34, 2.

- [19] Gupta J.N.D. ve Torres A., (2005). "Generating efficient schedules for identical paralel machines involving flow-time and tardy jobs", *European Journal of Operational Research*, 167: 1-4.
- [20] Dong C., Mingyuan C. ve Guohua W., (2005). "Parallel machine selection and job scheduling to minimize machine cost and job tardiness", *Computers & Operations Research*, 32 (8): 1995–2012
- [21] Nessah R., Yalaoui F. ve Chu C., (2006). "A branch-and-bound algorithm to minimize total weighted completion time on identical parallel machines with job release dates", *Computers & Operations Research*, 1.
- [22] Dunstall S. ve Wirth A., (2005). "Heuristic methods for the identical parallel machine flowtime problem with set-up times", *Computers & Operations Research*, 32, 11.
- [23] Ghirardi M. ve Potts C.N., (2005). "Makespan minimization for scheduling unrelated parallel machines: A recovering beam search approach", *European Journal of Operational Research*, 165, 1.
- [24] Chang P., Chen S. ve Lin K., (2005). "Two-phase sub population genetic algorithm for paralel machine-scheduling problem". *Expert Systems with Applications*, 29: 1-7.
- [25] Toledo F.M.B. ve Armentano V.A., (2006). "A Lagrangian-based heuristic for the capacitated lot-sizing problem in parallel machines", *European Journal of Operational Research*, 175, 1.
- [26] Rocha P., Ravettia M., Mateus G. ve Pardalos P., (2006). "Exact algorithms for a scheduling problem with unrelated paralel machines and sequence and machine-dependent setup times", *Computers & Operations Research*, 154-234.
- [27] Torres A., Lo'pez F. ve Ho J., (2007). Scheduling uniform parallel machines subject to a secondary resource to minimize the number of tardy jobs. *European Journal of Operational Research*, 179:1-13.
- [28] Li K., Sivakumar A.I. ve Ganesan V.K., (2008). "Complexities and algorithms for synchronized scheduling of parallel machine assembly and air transportation in consumer electronics supply chain", *European Journal of Operational Research*, 187: 442–455.

- [29] Low C., (2005). "Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines", *Computers & Operations Research*, 32: 2013-2025.
- [30] Blazewicz J., Pesch E., Sterna M. ve Werner F., (2008). "Metaheuristic approaches for the two-machine flow-shop problem with weighted late work criterion and common due date", *Computers & Operations Research*, 35( 2): 574-599.
- [31] Beraldi P., Ghiani G., Grieco A. ve Guerriero E., (2008). "Rolling-horizon and fix-and-relax heuristics for the paralel machine lot-sizing and scheduling problem with sequence-dependent set-up costs", *Computers & Operations Research*, 35: 3644 – 3656.
- [32] Biskup D., Herrmann J. ve Gupta J.N.D., (2008). "Scheduling identical parallel machines to minimize total tardiness", *Int. J. Production Economics*, 115: 134– 142.
- [33] Chen C.L. ve Chen C.L., (2009). "Bottleneck-based heuristics to minimize total tardiness for the flexible flow line with unrelated parallel machines", *Computers & Industrial Engineering* 56: 1393–1401.
- [34] Chen C.L. ve Chen C.L., (2009). "A bottleneck-based heuristic for minimizing makespan in a flexible flow line with unrelated paralel machines", *Computers & Operations Research*, 36: 3073–3081.
- [35] Rossi A. ve Boschi E., (2009). "A hybrid heuristic to solve the parallel machines job-shop scheduling problem", *Advances in Engineering Software* 40:118–127
- [36] Yang T., (2009). "An evolutionary simulation–optimization approach in solving paralel machine scheduling problems – A case study", *Computers & Industrial Engineering* 56:1126–1136
- [37] Agnetis A., Alfieri A. ve Nicosia G., (2009). "Assessing the quality of heuristic solutions to parallel machines min–max scheduling problems", *Int. J. Production Economics* 122: 755–762.
- [38] Wang X. ve Cheng T.C.E., (2009). "Heuristics for parallel-machine scheduling with job class setups and delivery to multiple customers", *Int. J. Production Economics* 119: 199–206.

[39] Xu D., Cheng Z., Yin Y. ve Li H., (2009). "Makespan minimization for two parallel machines scheduling with a periodic availability constraint", *Computers & Operations Research* 36: 1809–1812.

[40] Li K. ve Yang S., (2009). "Non-identical parallel-machine scheduling research with minimizing total weighted completion times: Models, relaxations and algorithms", *Applied Mathematical Modelling* 33: 2145–2158.

[41] Behnamian J., Fatemi Ghomi S.M.T. ve Zandieh M. (2010). Development of a hybrid metaheuristic to minimise earliness and tardiness in a hybrid flowshop with sequence dependent setup times, *International Journal of Production Research*, 48 (5): 1415-1438.

[42] Kim E.S., Sung C.S. ve Lee I.S., (2009). "Scheduling of parallel machines to minimize total completion time subject to s-precedence constraints", *Computers & Operations Research* 36: 698 – 710.

[43] Su L.H. ve Lien C.Y., (2009). "Scheduling parallel machines with resource-dependent processing times", *Int. J. Production Economics*, 117: 256–266.

[44] Ying K.C. ve Cheng H.M.,(2010). "Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic", *Expert Systems with Applications* 37: 2848–2852.

[45] Paula M.R., Mateus G.R. ve Ravetti M.G., (2010). "A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times", *Computers & Operations Research* 37: 938–949.

[46] Ross J.W. J. ve Bernardo A.L, (2011). Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics, *Computers & Operations Research*, 38 (12): 1816–1825.

[47] Dalfarda V.M. ve Mohammadi G.,( 2012). "Two meta-heuristic algorithms for solving multi-objective flexible job-shop scheduling with parallel machine and maintenance constraints", *Computers & Mathematics with Applications*, 64 (6): 2111–2117.

[48] Liao, C.J., Chao, C.W. ve Liang-Chuan Chen, (2012). "An improved heuristic for parallel machine weighted flowtime scheduling with family set-up times", *Computers & Mathematics with Applications*, 63 (1): 110–117.



[49] Li, X., Huang, Y., Tan, Q. ve Chen, H., (2013). "Scheduling unrelated parallel batch processing machines with non-identical job sizes", *Computers & Operations Research*, 40 (12): 2983–2990.

[50] Nogueira J.P., Arroyo J.E.C., Villadiego H.M.M. ve Gonçalves L.B., (2014). "Hybrid GRASP Heuristics to Solve an Unrelated Parallel Machine Scheduling Problem with Earliness and Tardiness Penalties", *Electronic Notes in Theoretical Computer Science*, 302, 53–72.

[51] Linn R. ve Zhang W., (1999). "HYBRID FLOW SHOP SCHEDULING: A SURVEY", *Computers & Industrial Engineering* 37: 57-61.

[52] Genoulaz, V.B., (2000). "Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness", *Int. J. Production Economics* 64: 101-111.

[53] Oguz C., Ercan M.F., Cheng T.C.E. ve Fung Y.F., (2003). "Heuristic algorithms for multiprocessor task scheduling in a two-stage hybrid flow-shop", *European Journal of Operational Research* 149: 390–403.

[54] Zandieh M., Fatemi Ghomi S.M.T. ve Moattar Hussein S.M., (2006). "An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times", *Applied Mathematics and Computation* 180: 111–127.

[55] Vob S. ve Witt A., (2007). "Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements: A real-world application", *Int. J. Production Economics* 105: 445–458.

[56] Low, C.Y., Hsu, C.J. ve Su, C.T., (2008). "A two-stage hybrid flowshop scheduling problem with a function constraint and unrelated alternative machines", *Computers & Operations Research* 35 (3): 845–853.

[57] Tseng, C.T., Liao, C.J. ve Liao, T., (2008). "A note on two-stage hybrid flowshop scheduling with missing operations", *Computers & Industrial Engineering*, 54 (3): 695–704.

[58] Choi H.S. ve Lee D.H., (2009). "Scheduling algorithms to minimize the number of tardy jobs in two-stage hybrid flow shops", *Computers & Industrial Engineering* 56: 113–120.

- [59] Cihanlı Ö., (2010). "Hibrit Akış Tipi Atölyede Çizelgeleme", Uludağ Üniversitesi, Yüksek Lisans Tezi.
- [60] Khalouli S., Ghedjati F. ve Hamzaoui A., (2010). "A meta-heuristic approach to solve a JIT scheduling problem in hybrid flow shop", *Engineering Applications of Artificial Intelligence* 23: 765–771.
- [61] Engin O., Ceran G. ve Yilmaz M.K., (2011). "An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems", *Applied Soft Computing* 11: 3056–3065.
- [62] Ruiz R. ve Rodríguez J.A.V., (2010). "The hybrid flow shop scheduling problem", *European Journal of Operational Research*, 205: 1-18.
- [63] Syam W.P. ve Al-Harkan I.M., (2010). "Comparison of Three Meta Heuristics to Optimize Hybrid Flow Shop Scheduling Problem with Parallel Machines", *World Academy of Science, Engineering and Technology*, 62.
- [64] Amin-Naseri M.R. ve Beheshti-Nia M.A., (2009). "Hybrid flow shop scheduling with paralel batching", *Int. J. Production Economics*, 117: 185–196.
- [65] Kahraman C., Engin O., Kaya İ. ve Öztürk R.E., (2010). "Multiprocessor task scheduling in multistage hybrid flow-shops: A parallel greedy algorithm approach", *Applied Soft Computing* 10: 1293–1300.
- [66] Ribas I., Leisten R. ve Framinan J.M., (2010). "Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective", *Computers & Operations Research* 37: 1439–1454.
- [67] Wang, S. ve Liu, M., (2013). "A heuristic method for two-stage hybrid flow shop with dedicated machines", *Computers & Operations Research*, 40 (1): 438–450.
- [68] Hadda, H., (2013). "A note on "A heuristic method for two-stage hybrid flow shop with dedicated machines", *Computers & Operations Research*, 40 (10): 2283.
- [69] Yang, J. (2013). "A two-stage hybrid flow shop with dedicated machines at the first stage", *Computers & Operations Research*, 40 (12): 2836–2843.

- [70] Wang, S. ve Liu, M., (2013). "A genetic algorithm for two-stage no-wait hybrid flow shop scheduling problem", *Computers & Operations Research*, 40 (4): 1064–1075.
- [71] Costa, A., Cappadonna, F.A. ve Fichera, S., (2013). "A dual encoding-based meta-heuristic algorithm for solving a constrained hybrid flow shop scheduling problem", *Computers & Industrial Engineering*, 64 (4): 937-958.
- [72] Chung, T.P. ve Liao, C.J., (2013). "An immunoglobulin-based artificial immune system for solving the hybrid flow shop problem", *Applied Soft Computing*, 13 (8): 3729–3736.
- [73] Chou (2013). "Particle swarm optimization with cocktail decoding method for hybrid flow shop scheduling problems with multiprocessor tasks", *International Journal of Production Economics*, 141 (1): 137–145.
- [74] Lahimer, A., Lopez, P. ve Haouari, M., (2013). "Improved bounds for hybrid flow shop scheduling with multiprocessor tasks", *Computers & Industrial Engineering*, 66 (4): 1106–1114.
- [75] Marichelvam, M.K., Prabakaran, T. ve Yang, X.S., (2014), "Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan", *Applied Soft Computing*, 19: 93-101.
- [76] Fattahi, P., Hosseini, S.M.H., Jolai, F. ve Tavakkoli-Moghaddam, R., (2014). "A branch and bound algorithm for hybrid flow shop scheduling problem with setup time and assembly operations", *Applied Mathematical Modelling*, 38 (1): 119–134.
- [77] Hong T., Huang C.M. ve Yu K.M., (1998). "LPT scheduling for fuzzy tasks", *Fuzzy Sets and Systems* 97: 277-286.
- [78] Pinedo M.L, (2008). "Scheduling Theory, Algorithms, and Systems", Springer.
- [79] Küçük, B. (2010). "Özdeş Paralel Makineli Bir Üretim Sisteminin Karınca Koloni Algoritması İle Çözülmesi", Doktora Tezi, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul.
- [80] Mohri S., Masuda T. ve Ishii H, (1999). "Bi-criteria scheduling problem on three identical parallel machines", *International Journal of Production Economics*, 60 (61): 529-536.

[81] Wight, O.W. (1984). "Production and Inventory Management in the Compute Age", Van Nostrand Reinhold Company, Inc., New York.

[82] Cox J.F., Blackstone J.H. ve Spencer M.S., (1992). APICS Dictionary, Amerikan Production and Inventory Control Society, Falls Church, Virginia.

[83] Başlıgil, H., (1988). "Sıralama ve Programlama", Yıldız Üniversitesi, 1. Baskı, İstanbul.

[84] Litoiu M. ve Tadei R., (2001). "Fuzzy schedulingwith application to real-time systems", Fuzzy Sets and Systems 121: 523–535.

[85] Anglani A., Grieco A., Guerriero E. ve Musmanno R., (2005). "Robust scheduling of parallel machines with sequence-dependent set-up costs", European Journal of Operational Research 161: 704–720.

[86] Volkan Ekinci, Üretim Planlama DERS notları,

<http://tr.scribd.com/doc/6569690/Uretim-Planlama-ve-Cizelgeleme> , 10 Ekim 2013.

[87] Engin, O., (2001). "Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma ile Çözüm Performansının Arttırılmasında Parametre Optimizasyonu", İTÜ, Doktora Tezi.

[88] Yağmahan B. ve Yenisey M. M., (2008). "Ant Colony Optimization for Multi – Objective Flow Shop Scheduling Problem", Computers & Industrial Engineering, 54 (3): 411-420.

[89] Tanyaş M. ve Başkak M., (2012). "Üretim Planlama ve Kontrol", İrfan Yayıncılık - Yönetim Bilimleri, İstanbul, ISBN : 9789753711098.

[90] Osman İ. ve Kelly J.P., (1996). "Meta-Heuristics: Theory and Applications", Kluwer Academic Publishers, ISBN:0792397002.

[91] Moukrim A. ve Quilliot A., (2005), "Optimal preemptive scheduling on a fixed number of identical parallel machines", Operations Research Letters, 33, 1.

[92] Ceren, G., (2006). "Esnek Akış Tipi Çizelgeleme Problemlerinin Veri Madenciliği Ve Genetik Algoritma Kullanılarak Çözülmesi", Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Y.Lisans Tezi.

- [93] Jungwattanakit J., Reodech M., Chaovalitwongse P. ve Werner F., (2009). "A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria". *Computers & Operations Research*, 36 (2): 358-378.
- [94] Koulamas C. ve Kyparisis G.J., (2000). "Scheduling on uniform parallel machines to minimize maximum lateness", *Operations Research Letters*, 26: 1.
- [95] Shabtay D. ve Kaspı M., (2006). "Parallel machine scheduling with a convex resource consumption function. *European Journal of Operational Research*", 173: 3-13.
- [96] Balın, S., Başlıgil, H. ve Alcan, P., (2007), "Paralel Makinelerin Genetik Algoritma Kullanılarak Çizelgelenmesi", 2. Ulusal Sistem Mühendisliği Sempozyumu, Yıldız Teknik Üniversitesi, Makine Fakültesi, Endüstri Müh. Bölümü, Yıldız-İstanbul.
- [97] Sun H. ve Wang G., (2003), "Parallel machine earliness and tardiness scheduling with proportional weights", *Computers & Operations Research*, 30: 1.
- [98] Belkadi K., Gourgand M. ve Benyettou M., (2006). "Parallel Genetic Algorithms With Migration For The Hybrid Flow Shop Scheduling Problem". *Journal of Applied Mathematics and Decision Sciences*. Article ID 65746: 1–17.
- [99] Gupta, J.N.D., (1988). "Two-stage, hybrid flow shop scheduling problem", *Journal of the Operational Research Society* 39 (4): 359–364.
- [100] Pinedo, M., (2002). "Scheduling Theory, Algorithms, and Systems", Prentice-Hall.
- [101] Graves, S.C., (1981). "A Review of Production Scheduling", *OPERATIONS RESEARCH*, 29: 646-675.
- [102] Osman, I.H. ve Laporte, G. (1996), "Metaheuristics: A bibliography". *Ann. Oper. Res.* 63: 513–623.
- [103] Glover, F., (1986). "Future paths for integer programming and links to artificial intelligence", *Computers & Operations Research*, Elsevier.

[104] Voss S., Osman IH., ve Roucairol C., (1999). "Meta-heuristics: Advances and trends in local search paradigms for optimization", Kluwer Academic Publishers Norwell, MA, USA.

[105] Stützle, T., (1999). "An ant approach to the flow shop problem", Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98), 3: 1560-1564.

[106] Dorigo M. ve Stützle T., (2003). "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances", International Series in Operations Research & Management Science, 57: 250-285.

[107] Başkent Üniversitesi, Metasezgisel Yöntemler Ders Notları,  
<http://www.baskent.edu.tr/~ayyuce/END407%20Ders3a.pdf> , 9 Eylül 2013.

[108] Cura T. (2008). Modern Sezgisel Teknikler ve Uygulamaları, Papatya Yayıncılık.

[109] Nabiyev, V.V., (2005). "Yapay Zeka", Seçkin Yayıncılık. 2.Baskı. Ankara.

[110] CHEN, C.L., VEMPATI, V.S., ALJABER, N. (1995). "An Application of Genetic Algorithms for Flowshop Problems", European Journal of Operational Research, 80: 389-396.

[111] Cleveland, G.A. ve Smith, F.S. (1989). "Using Genetic Algorithm to Schedule Flow Shop Release", Proc. 3rd Int. Conf. On Genetic Algorithms Applications, 160-169.

[112] Reeves, C.R. (1995). "A Genetic Algorithms for Flowshop Sequencing", Computers and Operations Research, 22 (1): 5-13.

[113] Paksoy S., (2007). "Genetik Algoritma ile Proje Çizelgeleme, Çukurova Üniversitesi", Doktora tezi, Sosyal Bilimler Enstitüsü, İşletme Anabilim Dalı, Adana, 2007.

[114] Cengiz, Y., (2004). "Optimum Performanslı Mikrodalga Kuvvetlendirici Tasarımı", Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 224s.

- [115] Pozitif Bilimler, Biyoloji, Genetik Algoritmalar çalışması;  
<http://www.pdrhizmetleri.com/archive/index.php/t-51553.html> , 7 Kasım 2013.
- [116] Biyoloji forumu, Genetik Algoritmalar Notları;  
<http://www.msxlabs.org/forum/biyoloji/155472-genetik-algoritmalar.html> , 20 Ekim 2012
- [117] Goldberg, D.E. (1989). "Genetic Algorithms in Search Optimization and Machine Learning", Addison Wesley Publishing Company, USA.
- [118] GHEDJATI, F. (1999). "Genetic Algorithms for the Job-Shop Scheduling Problem with Unrelated Parallel Constraints: Heuristic Mixing Method Machines and Precedence", Computers and Industrial Engineering, 37: 39-42.
- [119] Kurt, M. ve Semetay, C., (2001). "Genetik Algoritma ve Uygulama Alanları", M.M.O. Makine Mühendis Dergisi, 42 (501): 19-24.
- [120] Bolat, B., Erol, K.O ve İmrak, C.E., (2004), Genetic Algorithms in Engineering Applications and the Function of Operators, Journal of Engineering and Natural Sciences Mühendislik ve Fen Bilimleri Dergisi.
- [121] Kaya, İ., (2006), Çok Aşamalı Proseslerde Örnek Hacminin Belirlenmesi Üzerine Bir Model ve Genetik Algoritmalar Yardımıyla Çözüm Önerisi, Selçuk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 01.
- [122] Engin O. ve Fırlalı A., (2002), Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma Yardımı İle Çözümünde Uygun Çaprazlama Operatörünün Belirlenmesi, Doğu Üniversitesi Dergisi, 6, 27-35.
- [123] Murata, T., Ishibuchi, H., Tanaka, H., (1996), Genetic Algorithms for Flow Shop Scheduling Problems., Computers and Industrial Engineering, 30, (4): 1061-1071
- [124] Biroğlu, S., Güvenç, U., (2004). Genetik Algoritma Çözümü Gerçekleştiren Atölye Çizelgeleme Probleminde Ürün Sayısının Etkisi. Gazi Üniversitesi Teknik Eğitim Fakültesi Elektrik Eğitimi Bölümü, Beşevler, Ankara.

[125] Karabük Üniversitesi, Mühendislik Fakültesi İbrahim Çayıroğlu çalışması, <http://www.ibrahimcayiroglu.com/Dokumanlar/IleriAlgoritmaAnalizi/IleriAlgoritmaAnalizi-7.Hafta-GenetikAlgoritma.pdf> , 10 Mayıs 2014

[126] Çivril, H. (2009). “Hemşire Çizelgeleme Problemlerinin Genetik Algoritma ile Çözümü”. Yüksek Lisans Tezi. Süleyman Demirel Üniversitesi, Elektronik Bilgisayar Eğitimi Anabilim Dalı.

[127] Demir, M. ve Karcı, A., (2013). Biyoinformatik Verilerin Uygulamalı Analizi, Otomatik Kontrol Ulusal Toplantısı, TOK2013, 26-28 Eylül 2013, Malatya.

[128] Yönerge/Blogspot, Genetik Algoritmalar Seçim Yöntemleri Notları, <http://yonerge.blogspot.com.tr/2012/04/genetik-algoritmalar-secim-yontemleri.html>, 10 Mart 2014.

[129] Obitko, M., (1998). Genetic Algorithms, Hochschule für Technik und Wirtschaft Dresden (FD), <http://cs.felk.cvut.cz/~xobitko/ga/> , 10 Mayıs 2014.

[130] CICIRELLO, V.A. ve SMITH, S.F.. (2000). Modeling GA Performance for Control Parameter Optimization, Genetic and Evolutionary Computation, Conference, GECCO 2000, Las Vegas, Nevada, USA.

[131] Goldberg, D.E., (1985). “Computer-Aided Gas Pipeline Operation Using Genetic Algorithms And Rule Learning Part I: Genetic Algorithms In Pipeline Optimization”, Society of Petroleum Engineers, SPE-14590-MS, <http://dx.doi.org/NA> .

[132] Schaffer, J. D., Caruana, R. A. , Eshelman L. J. ve Das R., (1989). “A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization”, Proceedings of the Third International Conference on Genetic Algorithms, ed. J. D. Schaffer, San Mateo, CA: Morgan Kaufmann, 51-60.

[133] Chang, P.C. ve Chen, S.H. (2011), “Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times”. Appl. Soft Comput., 11 (1): 1263-1274.

[134] Çodur, M.Y., Tortum, A. ve Çodur, M., (2013), “Erzurum North Ring Road Accident Prediction Model with Generalized Linear Regression”, Iğdır Üni. Fen Bilimleri Enst. Der. /Iğdır Univ. J. Inst. Sci. & Tech., 3 (1): 79-84.



[135] Wikipedia Ortogonal Tasarım Notları,

[http://en.wikipedia.org/wiki/Principle\\_of\\_orthogonal\\_design](http://en.wikipedia.org/wiki/Principle_of_orthogonal_design) , 10 Mayıs 2014.

## ÖZGEÇMİŞ

---

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Pelin ALCAN  
**Doğum Tarihi ve Yeri** : 12.05.1981 / İstanbul  
**Yabancı Dili** : İngilizce  
**E-posta** : [pelinalcan@gmail.com](mailto:pelinalcan@gmail.com)

### ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Endüstri Müh.	Y.T.Ü	2008
Lisans	Endüstri Müh.	Y.T.Ü	2006
Lise		Haydarpaşa Süper Lisesi	1999

### YAYINLARI

#### Makale

1. Hüseyin Başlıgil, Selin Kara, Pelin Alcan, Betül Özkan, E. Gözde Çağlar, A distribution network optimization problem for third party logistics service providers, Expert Systems with Applications, Issue 38, Vol. 10, pp. 12730-127, (2011) (SCI).
2. Pelin Alcan, Hüseyin Başlıgil, A genetic algorithm application using fuzzy processing times in non-identical parallel machine scheduling problem, Advances in Engineering Software, Issue 1, Vol. 45, pp. 272-280, (2012) (SCI).

3. Pelin Alcan, Yavuz Özdemir, Berk Çınar, Hüseyin Başlıgil, An Implementation with Genetic Algorithm Approach in Non-Identical Parallel Machines, Journal of Multiple-Valued Logic and Soft Computing, (2013), Volume 21, p. 373-389 (SCI).
4. Yavuz Özdemir, Pelin Alcan, Huseyin Basligil, Cagri Dokuz, "Just-In Time Production System Using Fuzzy Logic Approach and A Simulation Application", Advanced Materials Research Vol. 445, (2012) , pp. 1029-1034 (SCI KAPSAMI DIŞINDAKİ İNDEKSLİ DERGİ).
5. Erman Yetiz, Pelin Alcan, Vildan Özkır, Hüseyin Başlıgil, "Application of Fuzzy AHP and ANP Methods for Chemical Reactions in Nitrochlorobenzen Formation", Sigma, Volume 27, Issue 3, ISSN: 1304-7191, 2009 (ULUSAL DERGİ).
6. Pelin Alcan, Yavuz Özdemir, Hüseyin Başlıgil, Emine Sevinçtekin, "Mobbing Davranışlarının Etkilerinin Araştırılması", Sigma, 2013 (ULUSAL DERGİ).

### **Bildiri**

1. Savaş Balın, Hüseyin Basligil, Pelin Alcan, "Paralel Makinelerin Genetik Algoritma Kullanılarak Çizelgelenmesi", Ulusal Sistem Mühendisliği Kongresi, Yıldız Teknik Üniversitesi, 6-8 Şubat 2008, Yıldız, İstanbul (ULUSAL KONGRE).
2. Umut İnan, Pelin Alcan, Hüseyin Basligil, "Kalite Yönetim Sistem Tetkiklerinin Önemi Ve Sıkça Karşılaşılan Problemler -Türk Karton Ambalaj Endüstrisinde Bir Uygulama", Ulusal Sistem Mühendisliği Kongresi, Yıldız Teknik Üniversitesi, 6-8 Şubat 2008, Yıldız, İstanbul (ULUSAL KONGRE).
3. E. Gözde Çağlar, Didem Gökcel, Pelin Alcan, "Genetik Algoritma Ve Hibrit Genetik Algoritmalar Arasından Paralel Makine Erken/Geç Bitirme Sıralama Problemleri İçin En Uygun Metodun Analitik Network Prosesi İle Belirlenmesi", Ulusal Sistem Mühendisliği Kongresi, Yıldız Teknik Üniversitesi, 6-8 Şubat 2008, Yıldız, İstanbul (ULUSAL KONGRE).
4. Pelin Alcan, Hüseyin Basligil, "Benzetimli Tavlama Yönteminin Paralel Makinelere Uygulanması Hakkında Literatür Araştırması", Ulusal Sistem Mühendisliği Kongresi, Yıldız Teknik Üniversitesi, 6-8 Şubat 2008, Yıldız, İstanbul (ULUSAL KONGRE).
5. Pelin Alcan, Hüseyin Basligil, "Bir Mermer Fabrikası İçin, Kuruluş Yeri Seçiminde Bulanık Topsis Uygulaması", Yöneylem Araştırması ve Endüstri Mühendisliği 29. Ulusal Kongresi, Bilkent-Ankara, 22-24 Haziran 2009 (ULUSAL KONGRE).
6. Pelin Alcan, Hüseyin Başlıgil, "A Study of selecting the best cogeneration system by fuzzy TOPSIS", 13th International Research/Expert Conference "Trends in the Development of Machinery and Associated Technology", TMT 2009, Hammamet-Tunisia, 16-21 October 2009 (ULUSLAR ARASI KONGRE).
7. E. Gözde Çağlar, Pelin Alcan, Hüseyin Başlıgil, "A Distribution Network Optimization Study in Third Party Logistics System", VII. International Logistics and Supply Chain Congress Program, İstanbul-Turkey, 5th-6th November 2009 (ULUSLAR ARASI KONGRE).
8. Pelin Alcan, Hüseyin Başlıgil, "A Literature Review for the Simulated Annealing Method with the Implementation on the Parallel Machines Since 2000", 24th European Conference on Operational Research, Lisbon-Portugal, 2010 (ULUSLAR ARASI KONGRE).

9. Yavuz Özdemir, Pelin Alcan, Hüseyin Başlıgil, Çağrı Dokuz, “Just-In Time Production System Using Fuzzy Logic Approach and A Simulation Application”, 14th International Conference on Advances in Materials and Processing Technologies, AMPT, İstanbul – Turkey, 2011 (ULUSLAR ARASI KONGRE).
10. Pelin Alcan, Hüseyin Başlıgil, “A Facility Location Selection Problem By Fuzzy Topsis”, 15th International Research/Expert Conference” Trends in the Development of Machinery and Associated Technology” TMT 2011, Prague, Czech Republic, 12-18 September 2011 (ULUSLAR ARASI KONGRE).
11. Pelin Alcan, Abit Balin, Hüseyin Başlıgil, “THE BEST ENERGY POLICY SELECTION USING VIKOR METHODOLOGY”, The Energy and Materials Research Conference 2012 (EMR 2012), Spain Malaga (ULUSLAR ARASI KONGRE).
12. Abit Balin, Pelin Alcan, Hüseyin Başlıgil, “Performance comparison of CCHP systems using different fuzzy multi criteria decision making models for energy sources”, The Energy and Materials Research Conference 2012 (EMR 2012), Spain Malaga (ULUSLAR ARASI KONGRE).
14. Pelin Alcan, Hüseyin Başlıgil, Tuğçe Deniz, “A Simulation Practice Of A Beverage Manufacturing Plant Using Simul 8”, World Conference on Business, Economics and Management (WC-BEM 2012), 3-6 Mayıs 2012, Antalya, Turkey (ULUSLAR ARASI KONGRE).
15. Pelin Alcan, Hüseyin Başlıgil, “Genetic Algorithms and Parallel Machine Applications”, InfoTech, June 20-22, 2013, Dalian-China (ULUSLAR ARASI KONGRE).

### **Kitap**

1. Fuelling the future: Advances in Science and Technologies for Energy Generation, Transmission and Storage, Edited by: A. Mendez-Vilas, BrownWalker Press, Boca Raton, Florida-USA, 2012, ISBN-10: 1-61233-558-6, ISBN-13: 978-1-61233-558-2.

### **Hakemlik Yaptığı Çalışma Ve Dergiler**

1. SRE-11-2148, “Use of Neural Fuzzy network with parallel genetic algorithm in Parallel Job scheduling”, Scientific Research and Essays, 2012.
2. IJMSEM\_01\_111019, “Two hybrid algorithms for solving the multi objective single machine batch scheduling problem based on simulated annealing and clustering methods”, International Journal of Management Science and Engineering Management, 2012.
3. SRE-12-448, “Certain Hybrid Optimization techniques for the optimization of the fuzzy controller of a parallel Job scheduling”, Scientific Research and Essays, 2012.
4. JCEIT-13-64, “Development of Neuro-Fuzzy-Genetic Framework for Oil Spillage Risk Management”, Journal of Computer Engineering & Information, 2013.
5. TPRS-2012-IJPR-1029, “Challenges in design and production of customized tracheal stents”, International Journal of Production Research, Aralık 2012.

- 6.** IJCIS-FLINS2012 Special Issue on Computational Intelligence in Decision Making, "A fuzzy multi-criteria decision making approach to assess building energy performance", 26-29 Ağustos 2012.
- 7.** AJBM/29.12.12/1593, "Strategies to schedule jobs in identical parallel machine production environment", African Journal of Business Management, Ocak 2013.
- 8.** TPRS-2012-IJPR-1378, "Exploration of Quality Function Deployment and Service quality for Fashion Boutique in Taiwan", International Journal of Production Research, Şubat 2013.
- 9.** SRE-15.03.13-5383, "Optimization of Power Distribution Networks using Parallel Genetic Algorithms", Scientific Research and Essays, Mart 2013.
- 10.** JCEIT-13-64, "Development of Neuro-Fuzzy-Genetic Framework for Oil Spillage Risk Management", Journal of Computer Engineering and Information Technology, Temmuz 2013.
- 11.** TPRS-2014-IJPR-0176, "Multi-criteria assessment of the Smart Grid efficiency using the fuzzy analytical hierarchy process", International Journal of Production Research, Nisan 2014.
- 12.** JOSH-D-13-00027, "Local search techniques for the Unrelated Parallel Machine Scheduling Problem with Sequence Dependent Setup Times", Journal of Scheduling, Mayıs 2014.
- 13.** 106209, "Sustainable Deforestation Evaluation Model and System Dynamics Analysis", Hindawi Publishing Corporation, Haziran 2014.