

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**MELEZ AKIŞ TİPİ ÇİZELGELEME PROBLEMİ İÇİN TEPKİSEL BİR
ALGORİTMA**

DOKTORA TEZİ

Abdullah AKTEL

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

AĞUSTOS 2015

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**MELEZ AKIŞ TİPİ ÇİZELGELEME PROBLEMİ İÇİN TEPKİSEL BİR
ALGORİTMA**

DOKTORA TEZİ

**Abdullah AKTEL
(507082110)**

Endüstri Mühendisliği Anabilim Dalı

Endüstri Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Mehmet Mutlu YENİSEY

AĞUSTOS 2015

İTÜ, Fen Bilimleri Enstitüsü'nün 507082110 numaralı Doktora Öğrencisi **Abdullah AKTEL**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “**MELEZ AKIŞ TİPİ ÇİZELGELEME PROBLEMİ İÇİN TEPKİSEL BİR ALGORİTMA**” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Mehmet Mutlu YENİSEY**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Vehbi Tufan KOÇ**
İstanbul Teknik Üniversitesi

Prof. Dr. Seçkin POLAT
İstanbul Teknik Üniversitesi

Prof. Dr. Şakir ESNAF
İstanbul Üniversitesi

Yrd. Doç. Dr. Betül YAĞMAHAN
Uludağ Üniversitesi

Teslim Tarihi : **30 Haziran 2015**
Savunma Tarihi : **05 Ağustos 2015**

Aileme,

ÖNSÖZ

Bu tez çalışmasında melez akış tipi bir üretim sisteminin dinamik olarak çizelgelenebilmesi araştırılmıştır. Çizelgeleme algoritması olarak bir genetik algoritma önerilmesi ve sistemde olabilecek değişkenliklere tepkisel olarak cevap verilmesi amaçlanmıştır. Bu çalışmada saygıdeğer hocalarımın, arkadaşlarımla ve ailemin büyük bir payı vardır. Tez çalışmam süresince engin bilgi ve birikimiyle yol gösteren değerli hocam Prof. Dr. Mehmet Mutlu YENİSEY'e her konudaki desteği ve güveni için çok teşekkür ederim. Yaptığı öneriler ve değerlendirmeler ile beni yönlendiren değerli hocam Doç. Dr. Dilay ÇELEBİ'ye, tez için yaptığımız toplantılarda beni sabırla dinleyip, önerileri ile tezin gelişmesine ve bu hale gelmesine katkıda bulunan Sayın Yrd. Doç. Dr. Betül YAĞMAHAN'a çok teşekkür ederim. Tezimin istatistiksel analizler ve performans değerlendirmesi kısmında yaptığı değerlendirmeler ve önerileri ile yol gösteren değerli hocam Yrd. Doç. Dr. Fahrettin ELDEMİR'e müteşekkirim. Tezimin her aşamasında bana verdiği destek için Sayın Engin SANSARCI'ya en derin saygı ve şükranlarımı sunarım.

Ayrıca hayatım boyunca maddi ve manevi hiçbir yardımı esirgemeyen çok değerli aileme sonsuz teşekkür ederim.

Ağustos 2015

Abdullah AKTEL

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
ÇİZELGE LİSTESİ.....	xiii
ŞEKİL LİSTESİ.....	xv
SEMBOL LİSTESİ	xvii
ÖZET.....	xix
SUMMARY	xxiii
1. GİRİŞ	1
2. LİTERATÜR TARAMASI	7
2.1 Melez Akış Tipi Çizelgeleme Problemi	7
2.1.1 Analitik yöntemler	7
2.1.2 Sezgisel yöntemler	9
2.1.3 Metasezgisel yöntemler.....	11
2.2 Tepkisel Çizelgeleme ve Dinamiklik	13
2.2.1 Analitik yaklaşımlar	13
2.2.2 Simülasyon tabanlı yaklaşımlar	15
2.2.3 Yapay zeka ve bilgi tabanlı yaklaşımlar	16
2.2.4 Melez yaklaşımlar	16
2.2.4.1 Sezgisellere simülasyonu entegre eden yaklaşımlar	19
2.3 Literatürün Analizi	21
3. ÇİZELGELEME ORTAMI.....	27
3.1 Standart Melez Akış Tipi Çizelgeleme Problemi Matematiksel Modeli	28
3.2 Problem Karmaşıklığı	29
3.3 Performans Ölçütü.....	30
3.4 Çizelgeleme Süreci.....	31
4. ÖNERİLEN METODOLOJİ.....	33
4.1 Çizelgeleme Algoritması.....	33
4.2 Dinamik Melez Akış Tipi Çizelgeleme Problemine Yönelik Olarak Önerilen Genetik Algoritma (GA) Yaklaşımı	34
4.2.1 Çözüm gösterimi	36
4.2.2 Başlangıç popülasyonunun üretilmesi	40
4.2.3 Bir çözümün uygunluk değerinin belirlenmesi.....	43
4.2.3.1 İşlem zamanlarının olasılıksal (stokastik) olduğu durumda uygunluk değerinin belirlenmesi.....	46
4.2.4 Seçim.....	48
4.2.4.1 Rulet tekerlek seçim metodu	48
4.2.4.2 Turnuva seçim metodu	50
4.2.5 Çaprazlama.....	51
4.2.6 Mutasyon.....	52
4.2.7 Bir sonraki popülasyonun belirlenmesi.....	54

4.2.8 Popülasyondaki bireylerin güncellenmesi.....	56
4.2.8.1 Çözüm uzayı değişimi sonrası permutasyonların oluşturulması	58
4.2.9 GA evrilme koşulu (durma koşulu).....	61
4.3 Tasarlanan Çizelgeleme Sistemi.....	63
4.3.1 GA tarafından önerilen bir çözümün atölyede gerçekleşmesi	64
5. DENEYLER VE SONUÇLARIN ANALİZİ	69
5.1 Çizelgeleme Ortamının Düzenlenmesi	69
5.2 GA Stratejilerinin Seçimi	71
5.3 GA Parametre Optimizasyonu.....	79
5.4 GA'nın Davranışının Gözlemlenmesi	81
5.5 Stokastik Durumda GA Uygunluk Değeri Koşum Sayısının Belirlenmesi.....	83
5.6 Önerilen GA Yaklaşımının Performansının Analizi	89
5.6.1 Dinamik deterministik durumda önerilen GA yaklaşımının performansının analizi	91
5.6.2 Dinamik stokastik durumda önerilen GA yaklaşımının performansının analizi	97
6. SONUÇ VE ÖNERİLER.....	105
KAYNAKLAR.....	109
ÖZGEÇMİŞ.....	119

KISALTMALAR

D	: Düzgün Dağılım
EBU	: Popülasyondaki En Büyük Uygunluk Değeri
EKÖ	: En Kısa İşlem Zamanı Önce
GA	: Genetik Algoritma
KT	: Kareler Toplamı
KO	: Kareler Ortalaması
NP	: Deterministik Olmayan Polinomial Zamanlı
R	: Rassal Sayı
SD	: Serbestlik Derecesi
SPT	: Shortest Processing Time
TB	: Tavlama Benzetim (SA-Simulated Annealing)
U	: Uygunluk değeri
US	: Üssel Dağılım

ÇİZELGE LİSTESİ

Sayfa

Çizelge 2.1 : Literatürün analizi.	22
Çizelge 4.1 : Örnek bir probleme ait işlerin sistemdeki işlem zamanları.	44
Çizelge 5.1 : Test problemleri.	71
Çizelge 5.2 : GA parametre ilk değerleri.	74
Çizelge 5.3 : Kolay problem sınıfı 1. Aşama varyans çözümleme analizi sonuçları.	74
Çizelge 5.4 : Kolay problem sınıfı 1. aşama faktör düzeyleri ortalama akış zamanları.	74
Çizelge 5.5 : Analiz prosedürü sonucu seçilen GA stratejileri.	78
Çizelge 5.6 : Çaprazlama oranı alternatifleri.	79
Çizelge 5.7 : Mutasyon oranı alternatifleri.	79
Çizelge 5.8 : 5 replikasyon için ortalama akış zamanı sonuçlarının ortalamaları.	80
Çizelge 5.9 : Alternatif koşum sayıları ve problem tipleri için hesaplanan hassasiyet oranları.	87
Çizelge 5.10 : Alternatif koşum sayıları ve problem boyutları için 5 deneme sonunda hesaplanan ortalama akış zamanı değerleri.	88
Çizelge 5.11 : Alternatif koşum sayıları ve problem sınıfları için hesaplanan değerlerin en iyi ortalama sonuçlara oransal uzaklıkları.	88
Çizelge 5.12 : Statik deterministik durum için geliştirilen test problemleri parametreleri.	89
Çizelge 5.13 : Statik durumda önerilen GA yaklaşımı ortalama sonuçları ile CPLEX sonuçlarının kıyaslanması.	90
Çizelge 5.14 : Statik durumda önerilen GA yaklaşımı ve CPLEX'in çözüm sürelerinin kıyaslanması.	90
Çizelge 5.15 : Dinamik durumda önerilen GA yaklaşımı ortalama sonuçları ile alternatif algoritmaların ortalama sonuçlarının kıyaslanması.	96
Çizelge 5.16 : Dinamik durumda önerilen GA yaklaşımı çözüm süreleri ile alternatif algoritmaların çözüm sürelerinin kıyaslanması.	96
Çizelge 5.17 : Dinamik stokastik durumda önerilen GA yaklaşımı ortalama sonuçları.	99
Çizelge 5.18 : Dinamik stokastik durumda önerilen GA yaklaşımı çözüm süreleri.	99
Çizelge 5.19 : Dinamik stokastik durumda önerilen GA yaklaşımı ortalama sonuçları ile dinamik deterministik durumda önerilen GA yaklaşımı ortalama sonuçlarının kıyaslanması.	100
Çizelge 5.20 : Dinamik stokastik durumda önerilen GA ile alternatif popülasyon büyüklükleri ve koşum sayıları hesaplanan ortalama akış zamanı değerleri ortalama sonuçları.	101
Çizelge 5.21 : Dinamik stokastik durumda önerilen GA'nın alternatif popülasyon büyüklükleri ve koşum sayıları için çözüm süreleri (saniye).	102
Çizelge 5.22 : Alternatif popülasyon büyüklükleri ve koşum sayıları için hesaplanan değerlerin en iyi ortalama sonuçlara oransal uzaklıkları.	102

Çizelge 5.23 : Popülasyon büyüklüğü 2000 ve koşum sayısı 1 için stokastik durumda önerilen GA yaklaşımı ortalama sonuçları ile deterministik durumda önerilen GA yaklaşımı ortalama sonuçlarının kıyaslanması..... 103

ŞEKİL LİSTESİ

Sayfa

Şekil 3.1 : Melez akış tipi çizelgeleme problemi.....	28
Şekil 3.2 : Standart melez akış tipi çizelgeleme problemi matematiksel model notasyonu.....	28
Şekil 3.3 : Dinamik çizelgeleme süreci.....	31
Şekil 4.1 : Önerilen GA'nın sözde kodu.....	36
Şekil 4.2 : Bir çözüm adayının permutasyon temsili.....	37
Şekil 4.3 : 9 ışık bir koromozomun gösterimi.....	37
Şekil 4.4 : 9 ışık bir çözüm adayı.....	37
Şekil 4.5 : En hızlı boş makineye atama kuralı sözde kodu.....	38
Şekil 4.6 : En küçük tamamlanma zamanı makine atama kuralı sözde kodu.....	39
Şekil 4.7 : Başlangıç popülasyonu üretim algoritması sözde kodu.....	41
Şekil 4.8 : Rastgele birey üretim algoritması sözde kodu.....	42
Şekil 4.9 : En Kısa İşlem Zamanı Önce Kuralı ile birey üretim algoritması sözde kodu.....	43
Şekil 4.10 : En Uzun İşlem Zamanı Önce Kuralı ile birey üretim algoritması sözde kodu.....	43
Şekil 4.11 : Sistemde 9 işin bulunduğu probleme ait kromozom.....	44
Şekil 4.12 : Bir çözüm adayına ait çizelge örneği.....	45
Şekil 4.13 : Rulet tekerlek seçim.....	49
Şekil 4.14 : Rulet tekerlek seçim metodu sözde kodu.....	50
Şekil 4.15 : Turnuva seçim metodu sözde kodu.....	51
Şekil 4.16 : Örnek bir çaprazlama işlemi.....	52
Şekil 4.17 : Çaprazlama operatörü sözde kodu.....	52
Şekil 4.18 : Örnek bir mutasyon işlemi.....	53
Şekil 4.19 : Mutasyon operatörü sözde kodu.....	53
Şekil 4.20 : Elitist strateji sözde kodu.....	55
Şekil 4.21 : Uygunluk tabanlı strateji sözde kodu.....	56
Şekil 4.22 : Dinamik bir üretim sisteminin t_1 anında yeni iş geldiğindeki durumu.....	57
Şekil 4.23 : Yeniden çizelgeleme öncesi durumun çözüm gösterimi.....	59
Şekil 4.24 : Yeniden çizelgeleme sonrası durumun çözüm gösterimi.....	59
Şekil 4.25 : Popülasyonun güncellenmesi sözde kodu.....	60
Şekil 4.26 : Dinamik bir üretim sisteminin herhangi bir andaki durumu.....	61
Şekil 4.27 : Tasarlanan çizelgeleme sisteminin genel çerçevesi.....	63
Şekil 4.28 : Çizelgenin atölyede gerçekleşmesi.....	66
Şekil 4.29 : Ele alınan dinamik melez akış problemine yönelik olarak önerilen GA yaklaşımının akış diyagramı.....	67
Şekil 5.1 : GA strateji alternatifleri.....	71
Şekil 5.2 : Her problem sınıfı için Varyans Çözümlemesi veri seti planı.....	72
Şekil 5.3 : t-testi veri seti planı.....	73
Şekil 5.4 : Kolay problem sınıfı analiz prosedürü aşamaları.....	75
Şekil 5.5 : Orta düzey problem sınıfı analiz prosedürü aşamaları.....	76

Şekil 5.6 : Zor problem sınıfı analiz prosedürü aşamaları.....	77
Şekil 5.7 : Kolay, orta ve zor problem sınıfları için F faktörü ortalama sonuçları. ...	78
Şekil 5.8 : Analiz prosedürü özet sonuçları.	78
Şekil 5.9 : Belirlenen stratejiler ve parametreler doğrultusunda GA'nın kolay problem sınıfı üzerindeki davranışı.	81
Şekil 5.10 : Belirlenen stratejiler ve parametreler doğrultusunda GA'nın orta düzey problem sınıfı üzerindeki davranışı.	82
Şekil 5.11 : Belirlenen stratejiler ve parametreler doğrultusunda GA'nın zor problem sınıfı.	82
Şekil 5.12 : Stokastik durumda kolay problem sınıfı için herhangi bir çözüm adayının uygunluk değerleri.....	84
Şekil 5.13 : Stokastik durumda orta düzey problem sınıfı için herhangi bir çözüm adayının uygunluk değerleri.....	85
Şekil 5.14 : Stokastik durumda zor problem sınıfı için herhangi bir çözüm adayının uygunluk değerleri.	86
Şekil 5.15 : Tavlama benzetim algoritması sözde kodu.	92
Şekil 5.16 : Tavlama benzetim algoritmasının kolay problem sınıfı üzerindeki davranışı.	93
Şekil 5.17 : Tavlama benzetim algoritmasının orta düzey problem sınıfı üzerindeki davranışı.	93
Şekil 5.18 : Tavlama benzetim algoritmasının zor problem sınıfı üzerindeki davranışı.	94
Şekil 5.19 : En kısa işlem zamanı önce algoritması sözde kodu.	94

SEMBOL LİSTESİ

j	: iş indeksi
r_j	: j işinin geliş zamanı
J_t	: t anına kadar sisteme gelen işler kümesi
p	: Bir işin işlem zamanı
\hat{p}_j	: Bir işin beklenen işlem zamanı
p_j^G	: Bir işinin gerçekleşen işlem zamanı
ψ	: Bir çizelge
$s_j(\psi)$: j işinin ψ çizelgesindeki başlangıç zamanı
t	: Zaman
$C_j(\psi)$: j işinin ψ çizelgesindeki beklenen tamamlanma zamanı
s_j^G	: j işinin gerçekleşen işleme başlama zamanı
C_j^G	: j işinin gerçekleşen tamamlanma zamanı
F_j^G	: j işinin gerçekleşen akış zamanı
\bar{F}	: Ortalama akış zamanı
Π_i	: i aşamasındaki işlerin sırası (permutasyonu)
$\Pi_i(h)$: h . pozisyondaki iş
$E(\bar{F})$: Beklenen ortalama akış zamanı
E_i	: i aşamasında işlem gören işler
n	: Koşum sayısı
α	: Güven düzeyi
ϵ	: Hata değeri
λ	: İş geliş hızı
U	: Makine kullanım oranı
N	: İş sayısı
k	: Üretim aşama sayısı
m	: Aşama makine sayısı
μ	: Servis hızı
ρ	: Sistem yoğunluğu
P_c	: Çaprazlama oranı
P_m	: Mutasyon oranı
σ	: Standart sapma
h	: Yarı güven aralığı
T	: Sıcaklık değeri

MELEZ AKIŞ TİPİ ÇİZELGELEME PROBLEMİ İÇİN TEPKİSEL BİR ALGORİTMA

ÖZET

Üretim sistemlerinin performansını artırmak yöneylem araştırmasının başlıca araştırma alanları içerisinde yer almaktadır. Gelişmiş üretim sistemlerinde birçok farklı ürün, müşterilerin istekleri doğrultusunda farklı zamanlarda ve farklı miktarlarda üretilmekte ve farklı tarihlerde teslim edilmektedir. Bu kadar karmaşık bir üretim sisteminin başarılı bir şekilde işletilebilmesi için, kaynakların etkin kullanılması gerekmektedir. Bu doğrultuda kompleks üretim sistemlerinin planlanması ve etkin bir şekilde işletilmesi için çeşitli modeller ve yaklaşımlar geliştirilmiştir.

Artan rekabet ortamı, müşteri isteklerinin farklılaşması ve beklentilerin yükselmesi, işletmeleri üretim sistemlerini daha etkin yönetmeye mecbur kılmaktadır. Sürekli değişen koşullar ve ortamdaki belirsizlikler dinamik karar verme mekanizmalarının geliştirilmesini gerektirmektedir.

Birçok üretim ve hizmet sistemi incelendiğinde yapılan işlerin bir dizi seri operasyondan geçirilerek hazır hale getirildiği görülmektedir. Sıklıkla bu operasyonlar her iş için aynı sırayı izlemek zorundadır. Makinelerin seri olarak sıralandığı ve işlerin de bu sırayı izlediği üretim ortamı, akış tipi olarak adlandırılır. Bir m - akış tipi üretim ortamında, m işlem aşaması vardır. Eğer her aşamada sadece bir işlem birimi var ise akış tipi üretim sistemi, birden fazla işlem birimi var ise melez akış tipi üretim sistemi olarak adlandırılır.

Çizelgeleme sınırlı kaynakların etkin bir şekilde tahsis edilmesine yönelik bir karar verme sürecidir. Melez akış tipi çizelgeleme problemine yönelik olarak mevcut literatür incelendiğinde araştırmacıların çoğunlukla problemin en genel hali ile ilgilendikleri ve bu durum için optimum çizelgeyi üretmeye çalıştıkları görülmektedir. Standart melez akış tipi çizelgeleme probleminde bütün işler ve makineler sıfır anında mevcuttur. Seçilen bir aşama için o aşamada işlem yapan makineler özdeştir. Sistemde herhangi bir stokastik unsur yoktur.

Bununla birlikte, gerçek hayatta önceden ön görülemeyen birçok belirsizlik kaynağı vardır. Makine bozulmaları, yeni gelen siparişler, sipariş iptalleri, teslim tarihi değişiklikleri, işlem zamanı dalgalanmaları, malzeme eksikliği gibi beklenmedik durumlarla her an karşılaşılabilir. Üretim sisteminde meydana gelen değişiklikler ve belirsizlikler önceden planlanmış üretim çizelgesini bozarak değişiklik yapmayı gerektirebilir. Bu gibi durumlar karşısında, üretim çizelgesi tepkisel algoritmalarla değişen koşullara göre dinamik olarak adapte edilebilir.

Bu çalışmada rassal olarak yeni iş gelişlerinin olduğu melez akış tipi bir üretim sisteminin dinamik olarak çizelgelenmesi hedeflenmiştir. Ele alınan dinamik melez akış tipi çizelgeleme problemine yönelik olarak tepkisel bir algoritma geliştirilmesi amaçlanmıştır. Bu doğrultuda ilk olarak kapsamlı bir literatür araştırması

gerçekleştirilerek, sürekli olarak çalışan dinamik bir üretim sistemi için efektif bir çizelgeleme/yeniden çizelgeleme sisteminin nasıl tasarlanabileceği incelenmiştir. Basit liste çizelgeleme kurallarından daha kompleks algoritmalara kadar pek çok alternatif dinamik çizelgeleme yaklaşımı olduğu görülmüştür. Bu yaklaşımların pek çoğunda çizelge üretimi için harcanan süre göz ardı edilmekte, dinamik çizelgeleme problemi statik alt problemlere bölünerek çözülmektedir.

Bu tez çalışması kapsamında önerilen yaklaşım, literatürdeki çalışmalardan çizelgeleme sürecini bir bütün olarak ele alma ve bu sürece yönelik olarak önerdiği dinamik çizelgeleme sistemi ile ayrılmaktadır. Çizelgeleme süreci, çizelge üretme ve üretilen iş emirlerinin gerçekleştirildiği atölye olmak üzere bir bütün olarak ele alınmıştır. Önerilen yaklaşım çözüm üretilmesi için geçen zamanı da çizelgeleme sürecine yansıtarak çalışan, tamamlanma zamanını temel alan bir yapıdadır.

Bu tez çalışması kapsamında dinamik stokastik ortamda melez akış tipi çizelgeleme problemine tepkisel olarak çözüm üretebilecek bir algoritma geliştirilmesi hedeflenmiştir. Ele alınan probleme yönelik olarak algoritma geliştirilirken problemin dinamik bir problem olduğu, algoritmanın çalışırken sistemin her an değişebileceği, çözüm süresi ve çözümlerin kalitesi göz önüne alınmıştır.

Bu doğrultuda öncelikle melez akış tipi çizelgeleme probleminin statik deterministik durumu için alternatif çözüm algoritmaları incelenmiştir. Daha sonra dinamik durumda büyük problem boyutlarında da iyi çözümleri kabul edilebilir sürede üretebilecek çözüm yaklaşımları analiz edilmiştir.

Dinamik melez akış tipi çizelgeleme problemine yönelik olarak evrimsel algoritmaların değişen ortamlara uyum sağlayabilme yeteneğinden yararlanılmaya çalışılmıştır. Bu kapsamda dinamik melez akış tipi çizelgeleme problemine yönelik olarak çizelgeleme ortamını anlık olarak ele alabilen değişken kromozom yapısına sahip ve hızlı bir şekilde çözüm üretebilen bir genetik algoritma (GA) geliştirilmiştir.

GA'nın çeşitli problem boyutları için statik durumdaki çözüm kalitesi test edildikten sonra genetik algoritma popülasyonlarının nasıl değişen problem koşullarına entegre edilebileceği araştırılmıştır. Değişkenliğin olduğu ana kadar ki arama süreci kazanımlarını değişkenlik sonrası ortama aktarabilen değişken kromozom yapıları bir yaklaşımla algoritmanın çok hızlı bir şekilde yeni durum için çizelge üretebildiği görülmüştür.

Geliştirilen GA, tasarlanan çizelgeleme sistemine uyarlanmıştır. Olay tabanlı bir yeniden çizelgeleme politikası tercih edilmiştir. Sisteme yeni iş gelişi ya da herhangi bir üretim aşamasından bir işin ayrılışı yeniden çizelgeleme noktası olarak kabul edilmiştir. Sürekli olarak çalışan bir üretim sistemine dinamik olarak çizelge üretebilmek için GA dinamik çizelgeleme sürecine entegre edilmiştir.

Bu doğrultuda klasik GA yaklaşımından bağımsız olarak problem değişimi sonrası o anki mevcut popülasyon yeni probleme değişken kromozom yaklaşımı ile uyarlanmış ayrıca GA'nın evrilmesi için beklenmemiştir. Bunun yerine arama süreci kazanımlarını değişkenlik sonrası güncellenen ortama aktarabilen küçük ama sürekli iyileştirmelere güvenen bir yaklaşım benimsenmiştir.

Dinamik çizelgeleme sistemi tasarlandıktan sonra, stokastik ortamın çizelgeleme sürecine nasıl yansıtılabileceği üzerinde durulmuştur. Önerilen genetik algoritmanın çözüm (çizelge) değerlendirme aşamasının simülasyon tabanlı sonuç üretmesi durumu incelenmiştir. Genetik algoritmanın çözüm değerlendirme modülü,

simülasyon entegrasyonu ile stokastik bir yapıya getirilerek çizelgeleme sürecine stokastik ortam yansıtılmaya çalışılmıştır.

Ele alınan dinamik melez akış tipi çizelgeleme problemine yönelik olarak GA strateji seçimi ve parametre optimizasyonu gerçekleştirilerek algoritmanın çözüm kalitesi iyileştirilmiştir. Dinamik deterministik ortamda geliştirilen GA'nın performansını analiz edebilmek için tavlama benzetim (TB) ve en kısa işlem zamanı önce (EKÖ) algoritmaları kıyaslama algoritması olarak kullanılmıştır. Öncelikle TB algoritmasında parametre optimizasyonu gerçekleştirilmiştir. Daha sonra hem TB algoritmasının hem de EKÖ algoritmasının GA için tasarlanan çözüm üretme ve çizelge gerçekleştirme sistemine entegrasyonu gerçekleştirilmiştir. Aynı çizelgeleme ortamında üç algoritma çözüm kalitesi ve çözüm süresi açısından kıyaslanmıştır. Önerilen GA yaklaşımı TB ve EKÖ algoritmalarına göre daha iyi sonuç vermiştir.

Dinamik stokastik ortam için önerilen GA'nın, uygunluk hesaplama simülasyon modülü koşum sayısı incelenmiştir. Çeşitli boyutlardaki test problemlerinde seçilen koşum sayıları için yarı güven aralığı değerleri incelenmiştir. Daha sonra dinamik stokastik ortamdaki GA'nın performansını analiz edebilmek için bir analiz prosedürü öne sürülmüştür. Buna göre dinamik deterministik ortam için iyi bir çözüm adayının dinamik stokastik ortam için de iyi bir çözüm adayı olabileceği varsayımında bulunulmuştur. Dinamik stokastik durum ile dinamik deterministik durum için önerilen GA çözümleri arasındaki fark performans kriteri olarak kabul edilmiştir. Alternatif test problemleri üzerinden dinamik stokastik durum ve dinamik deterministik durum kıyaslanmıştır.

Dinamik stokastik durumda GA tarafından önerilen çözümlerin kalitesini artırabilmek için herhangi bir nesildeki toplam uygunluk hesaplama bütçesini değiştirmeden alternatif popülasyon büyüklükleri ve koşum sayıları için elde edilen sonuçlar analiz edilmiştir. Koşum sayısının mümkün olduğunca küçük tutulduğu büyük popülasyonlu bir GA, ele alınan dinamik stokastik melez akış tipi çizelgeleme problemine yönelik olarak daha iyi çözümler üretmiştir. Ayrıca iş geliş hızının daha düşük olduğu, sistemdeki dinamikliğin kabul edilebilir bir seviyede bulunduğu çizelgeleme ortamları için dinamik deterministik durumdaki bir çözüm adayının dinamik stokastik durum için iyi bir çözüm olabileceği varsayımı doğrulanmıştır.

A REACTIVE ALGORITHM FOR THE HYBRID FLOW SHOP SCHEDULING PROBLEM

SUMMARY

Increasing the performance of manufacturing systems is one of the major application areas of operational research. Developed manufacturing systems can produce different products according to the different customer requirements at different times and deliver them with different quantities at different due dates. It is necessary to use resources effectively to run such a complex production system successfully. To plan and operate complex production systems effectively various models and approaches have been developed.

Increasing competition, differentiation of customer demands and rising of expectations oblige companies to manage production systems more effectively. Continuously changing conditions and uncertainties in the environment requires developing dynamic decision-making mechanisms.

In many production and service systems, operations are made ready after a series of processing units. Often these operations follow the same sequence. The processing environment which the processing units are ordered in series and the operations follow this sequence called as flow shop. m stages flow shop production system is composed of m processing units. If there are more than one processor units at least one stage, then this processing environment is called as a hybrid flow shop production system.

Scheduling is a decision-making process which deals with allocation of limited resources. Researchers are usually interested in the standard hybrid flow shop scheduling problem and try to produce the optimum schedule for the generic case of the problem. In the standard form of the hybrid flow shop scheduling problem all jobs and machines are available at time zero. Machines at a given stage are identical. There is no stochasticity in the system, problem data is deterministic and known in advance.

However, there are so many sources of uncertainty in real life. Unexpected situations such as machine breakdowns, new incoming orders, order cancellations, due date changes, fluctuations in processing time, material shortage can be encountered at any time. Occurred changes in the production and uncertainty in the system disrupt the planned schedule and may require making changes. In those circumstances, production schedule can be adapted changing conditions dynamically by using reactive algorithms.

In this study, a hybrid flow shop production system where jobs arrive randomly is investigated. It is aimed to develop a reactive algorithm to schedule dynamically the hybrid flow shop scheduling problem. Firstly, an extensive literature review is conducted to design an effective scheduling/rescheduling system for a continuously running production environment. It has been seen that there are so many alternative dynamic scheduling approaches such as dispatching rules, exact algorithms, and

heuristics in the literature. In many cases the time needed for schedule generation is neglected and the dynamic scheduling problem is solved by dividing the problem static sub-problems.

However, in real life, the schedule generation takes time. Depending on the size of the scheduling data, the scheduling method which is applied, the time it takes to perform the scheduling process may be very significant. Our study differs from the ones in the literature in terms of the taking the scheduling process as a whole and the proposed dynamic scheduling system. Scheduling process is considered as a schedule generation and job order realization on shop floor. The proposed approach works completion time based and reflects the elapsed time while schedule generation to the shop floor.

In this study, we aimed to develop a reactive algorithm that can produce solutions to the dynamic stochastic hybrid flow shop scheduling problem. It is considered that the scheduling environment is stochastic, and while the algorithm is running, the production system and also the solution quality can be changed at any moment.

Alternative solution algorithms for the hybrid flow shop problem in the static and deterministic environment are analyzed. After that, the solution approaches which are generate feasible solutions in a reasonable time for the bigger problem sizes and dynamic environment are investigated. The conducted extensive review revealed that the fact that genetic algorithms (GA) used in dynamic scheduling mostly because of their adaptation skill to the changing environments. So we decided to use a GA to generate schedules for the considered hybrid flow shop scheduling problem. It is aimed to develop a GA to cover the changing production environment instantaneously by using changeable chromosome structure in its population.

Firstly the GA solution quality is checked in static environment by using alternative test problems in different sizes. It is investigated that how the GA population is integrated the changing conditions. It is found that using an approach that considers the search process gains which already done before the dynamic situation happened improve the efficiency. In other words, instead of restarting the GA when the dynamic situation happened, the population of the newly started GA is obtained from the final population of the previous GA by adjusting it to the requirements of the new problem.

The developed GA is adapted to the designed scheduling system by using an event driven rescheduling policy. New job arrivals or job completions at any stage are chosen to the rescheduling point. To generate schedules dynamically on a continuously running environment the GA is integrated the scheduling process.

Separate from the classical GA approaches after the problem changed, it is not needed to wait until the algorithm evolved. Instead of this, the algorithm trusts little but continuous improvements and transfer of search process gains to the new problem after the dynamic situation happened.

After the dynamic scheduling system is designed, we focused on how the stochastic environment can be reflected in the scheduling process. Integration of simulation to the solution evaluation module of the proposed GA is investigated. The stochasticity in the environment is reflected to the GA by using random variables as jobs processing times. And fitness calculation of solutions is done by using simulation.

To increase the solution quality of the proposed GA, strategy selection and parameter optimization is performed. The performance of the proposed GA in the static

environment is compared with CPLEX optimum solutions for small size problems. To analyze the performance of the proposed GA in the deterministic and dynamic environment, simulated annealing (SA) and shortest processing time (SPT) algorithms are used as a benchmark.

Parameter optimization is also performed for SA. Both SA and SPT algorithm is integrated to the designed solution generation and schedule realization process. In the same conditions, the three algorithm is compared according to the solution quality and run time. The proposed GA approach yielded better results than others.

Replication number of the simulation module while evaluating fitness of solutions in the dynamic stochastic environment is investigated. By using various test problems at different sizes, confidence interval is analyzed for chosen replication numbers. To analyze the proposed GA performance in the dynamic stochastic environment an analysis procedure is proposed. In this procedure a good solution in the dynamic deterministic environment is assumed to be also a good solution for the dynamic stochastic environment.

By using alternative test problems, the proposed algorithm performance in the dynamic stochastic environment and in the dynamic deterministic environment is compared. To increase solution quality in the dynamic stochastic environment, replication number and population size are analyzed together without changing total evaluation budget in the algorithm. Increasing the population size and decreasing the replication number strategy gives better solutions. Test results revealed that a good solution in the dynamic deterministic environment is also good solution for the dynamic stochastic environment assumption is verified especially when there is low level stochasticity in the system.

1. GİRİŞ

Çizelgeleme, birbiri ile rekabet halindeki bir veya daha fazla hedefi optimize etmek amacı ile sınırlı kaynakların tahsis edilmesine yönelik bir karar verme sürecidir. Klasik çizelgeleme teorisinde problemler öncelikle matematiksel olarak modellenir. Daha sonra bu modeller optimum sonuç veren mevcut algoritmalar ile ya da optimuma yakın iyi bir sonuç veren sezgisel algoritmalar yardımı ile çözülür.

Birçok çizelgeleme problemi NP-zor olduğu için araştırmacılar çoğunlukla bu problemlerin statik ya da deterministik versiyonları ile ilgilenmişlerdir. Çizelgeleme literatürü dikkate alındığında yapılan çalışmaların büyük çoğunluğu belirsizliğin olmadığı verilerin kesin olarak bilindiği durumları göz önüne almıştır. Bununla birlikte, gerçek hayatta üretim planını etkileyen bir çok belirsizlik kaynağı vardır. Makine bozulmaları, yeni gelen siparişler, sipariş iptalleri, teslim tarihi değişiklikleri, işlem zamanı dalgalanmaları, malzeme eksikliği gibi beklenmedik durumlar üretim ortamında her an karşımıza çıkabilir. Üretim ortamındaki bu değişkenlikler ve belirsizlikler önceden planlanmış olan üretim çizelgesini bozarak değişiklik yapmayı gerektirebilir.

Bozulmuş bir çizelge, kaçırılmış teslim tarihleri, makine beklemeleri, atıl kapasite, yüksek süreç içi envanteri gibi yüksek maliyetli sonuçlar doğurabilir. Bütün bu nedenler, dinamik üretim ortamında çizelgeleme yaparken sadece optimum ya da optimuma yakın bir çözüm bulmak yerine belirsizlik ortamında değişkenliklere adapte olabilen bir çizelgeme metodolojisini gerekli kılmaktadır.

Dinamik üretim sistemlerinde optimum çizelge kolaylıkla bozulabilir ya da uygulanamaz bir hale gelebilir. Bu gibi durumlar, üretim çizelgesini bozan bu faktörlere cevap veren tepkisel algoritmalar ile giderilebilir. Bu sayede üretim çizelgesi yeniden uygun bir hale getirilebilir.

Dinamik ve stokastik karakter gösteren bir çizelgeleme ortamı ile bu ortamlarda üretim yapan firmaların nasıl başa çıktığına bakıldığında ise, birçok çizelgeleme

uygulamasında ve yapılan arařtırmalarda izelgeleme sreci adımlarının ařađıdaki gibi gerekleřtiđi grlmřtr (zgn, 2012):

1. Bekleyen sipariřler, atlyenin durumu, rn zellikleri (iřlem zamanları, hazırlık zamanları, vb.) retim kapasitesi, rn ve hammadde gibi izelgeleme verisinin sisteme yklenmesi.
2. Birinci adımda fotođrafı ekilen sistemin durumu temel alınarak, bu sistemin optimum izelgesinin elde edilmesi iin statik ve deterministik bir izelgeleme algoritmasının kořturulması.
3. izelgeleme sonucunun deđerlendirilmesi ve iř emirlerinin oluřturulması.
4. retilen izelgenin atlyeye iletilmesi.

Birinci adımla veri tabanından yklenen verinin sistemin o anki durumunu yansıttıđı dřnlr. Bu yzden ikinci adımda ele alınan izelgeleme problemi statik ve deterministik bir karakterdedir. Sistemin stokastik karakteri ile her izelgeleme sreci ncesinde o andaki verinin izelgeleme srecine alınması yolu ile bařa ıkılmaya alıřılır.

Literatrdeki birok alıřmaya bakıldıđında 1. ve 4. adımların anlık olarak gerekleřtiđi dřnlr. Fakat gerek dnyada btn adımlar iin belli bir sre harcanır. izelgeleme yapılan verinin byklđ, ele alınan problemin kompleksitesi ve kullanılan izelgeleme yntemi, izelgelemenin ne kadar sreceđinde kritik rol oynar. Bu dođrultudan bakıldıđında gerek bir retim sisteminin izelgelenmesi saatler srebilir.

1. ve 4. adım bu alıřmanın konusu deđerildir. Bu yzden 1 ve 4 adımlarının anlık olarak gerekleřtiđi ve herhangi bir zaman almadıđı dřnlecektir. 2. ve 3. adımları ise ayrı ayrı ele almanın bu alıřma aısından herhangi bir ek getirisi olmayacađı dřnlr. Bu iki adım bir arada deđerlendirilecektir. Bu alıřmaya gre sadece 2. adım belli bir sre almakta ve diđer btn adımlar anlık olarak gerekleřmektedir.

izelgeleme sreci sistemin belli bir andaki durumunu ele alır (1. adım) ve o an iin bir izelgeleme algoritması kořturur. Daha nce belirtildiđi zere izelgeleme algoritmasının kořturulması sırasında belli bir sre geer. izelgeleme algoritması kořturulurken sistemde stokastiklik ve dinamiklik olduđu iin sistemin bu sırada deđerleřtiđi dřnlr. Sistemin durumu, nihai izelge ve iř emirleri atlyeye

iletildiğinde (4. adım) çizelgeleme algoritmasında kullanılan veriye göre çok farklılık gösterebilir. Elde edilen sonuç mevcut atölye koşulları ile örtüşmeyen bir sonuç olabilir (Özgün, 2012).

Genel bir çizelgeleme sisteminde iki temel mekanizma vardır: çizelge üretimi ve kontrolü. Çizelge üretimi, bu sistemin planlama işlevini yerini getirir. İşlerin ön görülen başlangıç ve bitiş zamanlarının belirlenmesi ve atanması çizelge üretimi aşamasında belirlenir. Kontrol mekanizması ise daha ziyade mevcut çizelgenin yürütülmesi ve beklenmedik olaylar karşısında çözüm üretilmesi yönünde çalışır. Kontrol sistemi bu anlamda makine bozulmaları, acil işler, malzeme azlığı gibi değişkenliklere karşı cevap veren tepkisel bir mekanizmadır (Sabuncuoğlu ve Kızılışık, 2003).

Önceden planlanmış olan çizelgenin performansı, değişken durumlar karşısında sistemin bu durumlara karşı ürettiği çözümlere bağlıdır. Dinamik ortamda, sistemi önceden planlanmış haline ya da istenen bir seviyeye getirmek için hızlı ve etkili stratejiler geliştirmek gerekmektedir.

Dinamik çizelgeleme literatürü incelendiğinde periyodik ve olay tabanlı olmak üzere farklı formlarda çizelgeleme süreci başlatma mekanizmalarının analiz edildiği görülür (Church ve Uzsoy, 1992). Periyodik çizelgeleme yaklaşımında bir çizelgeleme frekansı belirlenir ve bu frekans doğrultusunda yeniden çizelgeleme gerçekleştirilir. Olay tabanlı çizelgeleme yaklaşımında ise, çizelgeleme yapılan sisteme yeni bir iş gelişi, sistemdeki makinelerden birinin bozulması gibi yeniden çizelgeleme gerektiren dinamik olaylar gerçekleştiğinde yeniden çizelgeleme yapılır (Sabuncuoğlu ve Kızılışık, 2003).

Bu çalışmada gerçek üretim sistemlerinde pek çok uygulama alanı olan dinamik melez akış tipi üretim sistemi ele alınarak tepkisel bir çizelgeleme algoritması geliştirilmiştir. Bu kapsamda ilk olarak, sürekli olarak çalışan dinamik bir üretim ortamı için efektif bir çizelgeleme/yeniden çizelgeleme sisteminin nasıl tasarlanabileceği incelenmiştir. Daha sonra melez akış tipi çizelgeleme problemine yönelik olarak bir algoritma önerilmiştir.

Çizelgeleme süreci, çizelge üretme ve üretilen iş emirlerinin gerçekleştiği atölye olmak üzere bir bütün olarak ele alınmıştır. Dinamik melez akış tipi üretim sistemi için geliştirilen algoritmanın sürekli olarak çalışan bu sistemde çizelgeleme

yapabilecek şekilde çözüm üretmesi (çizelge) sağlanmıştır. Çalışmanın son kısmında ise stokastik ortamın çizelgeleme sürecine yansıtılması için simülasyon tabanlı bir çözüm önerilmiştir.

Ele alınan probleme yönelik olarak algoritma geliştirilirken problemin dinamik bir problem olduğu, algoritmanın çalışırken sistemin her an değişebileceği, çözüm süresi ve çözümlerin kalitesi göz önüne alınmıştır. Bu doğrultuda literatürdeki melez akış tipi çizelgeleme algoritmaları, dinamik çizelgeleme yaklaşımları ve dinamik melez akış tipi çizelgeleme çalışmaları incelenmiştir.

Dinamik melez akış tipi çizelgeleme problemine yönelik olarak evrimsel algoritmaların değişen ortamlara uyum sağlayabilme yeteneğinden yararlanılmaya çalışılmıştır. Bu kapsamda dinamik melez akış tipi çizelgeleme problemine yönelik olarak çizelgeleme ortamını anlık olarak ele alabilen değişken kromozom yapısına sahip ve hızlı bir şekilde çözüm üretebilen bir genetik algoritma geliştirilmiştir.

Geliştirilen genetik algoritma, tasarlanan çizelgeleme sistemine uyarlanmıştır. Olay tabanlı bir yeniden çizelgeleme politikası tercih edilmiştir. Sisteme yeni iş gelişi ya da sistemden bir işin ayrılışı yeniden çizelgeleme noktası olarak kabul edilmiştir. Çizelgeleme sistemi tasarlanırken çizelgeleme süreci göz önünde bulundurulmuş, algoritmanın önerdiği çizelgenin atölyede gerçekleşmesi sağlanmıştır. Atölye ile çizelgeleme algoritmasının bir biri ile iletişim halinde olduğu, gerekli değişiklik ve güncellemelerin anlık olarak yapıldığı bir çizelgeleme süreci tasarlanmıştır.

Dinamik çizelgeleme sistemi tasarlandıktan sonra, stokastik ortamın çizelgeleme sürecine nasıl yansıtılabileceği üzerinde durulmuştur. Önerilen genetik algoritmanın çözüm (çizelge) değerlendirme aşamasının simülasyon tabanlı sonuç üretmesi durumu incelenmiştir. Genetik algoritmanın çözüm değerlendirme modülü, simülasyon entegrasyonu ile stokastik bir yapıya getirilerek çizelgeleme sürecine stokastik ortam yansıtılmaya çalışılmıştır.

Bu çalışmanın geri kalan kısmı ise şöyle devam etmektedir: İkinci bölümde bu çalışma kapsamında ele alınan konu ile ilgili literatür verilmiştir. Üçüncü bölümde melez akış tipi çizelgeleme problemi ve çizelgeleme ortamı analiz edilmiştir. Dördüncü bölümde ele alınan probleme yönelik olarak önerilen metodoloji açıklanmıştır. Beşinci bölümde yapılan deneyler ve sonuçları analiz edilmiştir.

Altıncı ve son bölümde ise elde edilen sonuçlar değerlendirilmiş ve ilerideki çalışmaların hangi yönlerde ilerleyebileceği tartışılmıştır.

2. LİTERATÜR TARAMASI

Tez konusunun bir çok alanla ilişkili olmasından dolayı literatür taraması alt ana başlıklara bölünerek yürütülmüştür. Bu alt ana başlıklarda temel olabilecek ve güncel yayınlara yer verilmiştir. Literatür taramasında ilk önce melez akış tipi çizelgeleme problemi genel olarak ele alınacak ve bu problemle ilgili yayınlar özetlenecek, ardından dinamik ortamlarda çizelgeleme ve tepkisel algoritmalarla ilgili yayınlara değinilecektir.

2.1 Melez Akış Tipi Çizelgeleme Problemi

Birçok üretim ve montaj tesisinde işler bir dizi seri operasyondan geçirilerek hazır hale getirilir. Sıklıkla bu operasyonlar her iş için aynı sırayı izlemek zorundadır. Makinelerin seri olarak sıralandığı ve işlerin de bu sırayı izlediği üretim ortamı akış tipi üretim olarak adlandırılır. Bir m -makine akış tipi üretim ortamında, m -üretim aşaması vardır. İşler mutlaka her aşamada işlem görmeli ve aynı sırayı takip etmelidirler. İşlerin farklı aşamalardaki işlem süreleri farklı olabilir. Eğer her aşamada sadece bir makine var ise akış tipi üretim sistemi, birden fazla makine var ise melez (esnek) akış tipi üretim sistemi olarak adlandırılır (Pinedo, 2008).

Melez akış tipi çizelgeleme probleminin çözümü için uygulanan yaklaşımlar üç ana başlıkta sınıflandırılabilir. Bunlar: analitik yöntemler, sezgisel yöntemler, metasezgisel yöntemlerdir.

2.1.1 Analitik yöntemler

Melez akış tipi çizelgeleme problemleri için en çok kullanılan analitik yöntem dal sınır algoritmasıdır (Ruiz ve Vazquez-Rodriguez, 2010). Gupta ve diğ. (1997) ilk aşamasında birden çok paralel makine olan iki aşamalı bir melez akış tipi çizelgeleme problemi için bir dal sınır algoritması önermişlerdir. Daha düşük alt sınır bularak arama uzayını sınırlandırmışlardır. Kim ve diğ. (1997) Johnson kuralı benzeri bir optimal prosedür geliştirerek iki aşamalı melez akış tipi çizelgeleme probleminin üretim süresini minimize etmeye çalışmışlardır. Brockmann ve diğ.

(1997) paralel bir dal sınır algoritması öne sürmüşler. Öne sürdükleri algoritma birçok durumda diğer algoritmalara oranla daha hızlı sonuç verme açısından lineer bir iyileşme sağlamıştır. Moursli ve Pochet (2000) esnek akış tipi çizelgeleme probleminin en genel hali için (m -aşama ve her aşama için k -makine durumu) yeni bir dal sınır algoritması ile yeni bir dallanma metodu önermişlerdir. Öncelikle iş bağımlı kuyruk kavramını tanımlamışlardır, buna göre bir işin son iş son aşamadan geçip tamamlanana kadar üretim hattında geçirdiği minimum süre iş bağımlı kuyruk olarak tanımlanmıştır. Geliştirdikleri algoritmada her dallanma düğümü iki ayrı problemden oluşmakta ve iş ayrılış zamanı ve iş bağımlı kuyruk değişebilmektedir. Alt sınırı, ana problemi tek aşamadan oluşan alt problemlere esneterek hesaplamaya çalışmışlardır.

Carlier ve Neron (2000) m -aşama ve her aşama için k -makine durumu için yeni bir dal sınır algoritması önermişlerdir. Hesaplama süresini azaltmak için enerjik düşünme ve küresel operasyonlar kavramlarını kullanmışlardır. Haouari ve diğ. (2006) önerdikleri dal sınır algoritması ile iki aşamalı esnek akış tipi çizelgeleme problemini büyük problem boyutları için çözebilmeyi amaçlamışlardır. Allaoui ve Artiba (2006)'nın iki aşamalı esnek akış tipi çizelgeleme problemi için geliştirdiği dal sınır algoritması ile yeni bir dallanma planı önermişlerdir. Bu yeni algoritmayı makinelerin üretim için müsait olmadığı durumların önceden bilindiği durumlar için uygulamışlardır buna göre her makine en fazla bir periyot müsait olmamaktadır.

Guirchoun ve diğ. (2005) iki aşamalı melez akış tipi çizelgeleme probleminin bekleme olmaksızın ve ilk aşamada her işin aynı işlem süresinde olduğu durum için bir polinomial zaman algoritması önermişlerdir. Bu algoritma öncelikle işleri 3 kümeye böler ve daha sonra en kısa işlem süresi kuralına göre sıralar ve her işi ilk müsait makine kuralına göre atar. Sawik (2000) esnek akış tipi çizelgeleme probleminin en genel halinin optimum çözümünü bulmak için bir karma tam sayılı programlama önermiştir. Siparişe göre üretim ve limitli ara stok durumu için esnek akış tipi çizelgeleme probleminin üretim süresini optimum sonuç veren yöntemler ile minimize etmeye çalışmıştır. İki amaçlı esnek akış tipi çizelgeleme probleminin özdeş makineler ve limitli ara stok durumu için bir leksikografik yaklaşım önermiştir.

Dessouky ve diğ. (1998) üç aşamalı melez akış tipi çizelgeleme problemi özdeş makineler durumu ile ilgilenmişlerdir. Üretim süresini minimize ederek optimum

sonucu bulan bir dal sınır algoritması önermişlerdir. Sawik (2005) ve Sawik (2006) melez akış tipi çizelgeleme probleminin en genel halinde makinelerin birden çok işi aynı anda işleyebildiği durum için optimum çözüm veren bir yöntem önermiştir. He ve diğ. (2007) iki aşamalı melez akış tipi çizelgeleme problemi için ilk aşamada m -özdeş makine ikinci aşamada ise parti işleyici bulunduğu durum için n -işli çizelgeleyen bir dinamik programlama algoritması önermişlerdir.

2.1.2 Sezgisel yöntemler

Literatürde melez akış tipi çizelgeleme problemi için birçok sezgisel yöntem önerilmiştir, bu sezgiseller arasındaki en basit tip sezgiseller, sevk kuralları ya da diğer bir deyişle çizelgeleme politikaları veya liste çizelgeleme algoritmalarıdır. Bu kurallar temel olarak işlerin sıralanması ve makinelere atanmasını içerirler. Guinet ve diğ. (1996) iki aşamalı melez akış tipi çizelgeleme problemi için üretim süresini minimize eden iki aşamalı bir algoritma önermişlerdir. Problemi öncelikle parçalara ayırmışlar daha sonra bir öncelik listesine göre işleri sıralamışlar ve işleri bu öncelik sırasına göre müsait olan makinelere atamışlardır. İşleri derecelendirmek için birçok sıralama kuralını gözden geçirmişlerdir. Johnson kuralının en iyi sonuç verdiğini göstermişlerdir.

Guinet ve Solomon (1996) m -aşamalı bir melez akış tipi çizelgeleme problemi için maksimum gecikmeyi veya maksimum tamamlanma zamanını minimize etmeye çalışmışlardır. İşleri akış tipi çizelgeleme algoritması kullanarak sıralayıp önceliklendirmişlerdir. Bu algoritmalar içinde Nawaz ve diğ. (1983)'nin sezgiseli (NEH) bütün kriterlere göre diğerlerinden daha üstün sonuç vermektedir. Gupta ve diğ. (1997) iki aşamalı esnek akış tipi çizelgeleme problemi için (ilk aşamada birden çok makine ikinci aşamada tek makine durumu için) birçok yapıcı sezgiseli incelemiş ve üç aşamalı bir sezgisel önermişlerdir. Buna göre; ilk aşamada işler Johnson kuralı uyarlamasına göre sıralanır. Daha sonra işler sanki ilk aşamada tek makine ikinci aşamada birden çok makine var gibi kabul edilerek atanır. Son olarak orjinal problemde varsayılan probleme göre yapılan atama sonucunda boş zaman dilimi kalacağından ikinci aşamadaki işlerin tekrardan çizelgelenmesi ilk aşamadaki tamamlanma zamanlarına göre yapılır.

Li (1997) iki aşamalı melez akış tipi çizelgeleme problemini hazırlık sürelerinin olduğu durum için incelemişlerdir. Benzer tipteki işleri, iş aileleri şeklinde

gruplamışlar ve siparişler hazırlanırken aile tabanlı hazırlık sürelerini göz önünde bulundurmışlardır. Buna göre ileriye doğru ve geriye doğru atama politikalarının performansını sekiz sıralama kuralı ile kombine ederek değerlendirmişler ve geriye doğru tahsis politikasının daha iyi sonuçlar verdiğini gözlemlemişlerdir. Zhang ve diğ. (2005) iki aşamalı melez akış tipi çizelgeleme problemini hazırlık sürelerinin var olma durumu için ele almışlardır. Her iş için o işin alt parti büyüklüklerini ve sayısını tespit ederek üretim süresini minimize etmeye çalışmışlardır. Önerdikleri sezgisel öncelikle her iş için parti büyüklüklerini tespit edip birinci aşamadaki toplam iş yükünü her iki aşamaya yaymaya dayanmaktadır.

Oguz ve Ercan (1997) iki aşamalı melez akış tipi çizelgeleme problemi için geliştirdikleri sezgisel ile problemi aşamalara bölüp her aşamadaki işleri atamak için çizelgeleme kuralları önermişlerdir. 5 çizelgeleme kuralının beraber çalıştığı kombine bir yöntem geliştirmişlerdir. Oguz ve diğ. (2003a) iki aşamalı melez akış tipi çizelgeleme problemi için önerdikleri sezgisel, önce işleri literatürdeki bir öncelik kuralına göre sıralamak daha sonra her iş için çizelgeyi bulmak şeklindedir. Vandeveld ve diğ. (2005) önerdikleri ileri aşama ve geri aşama algoritması ile her aşamayı paralel makine çizelgeleme problemi gibi ele alıp çözmüşlerdir. Gupta ve Tunc (1998) iki aşamalı melez akış tipi çizelgeleme problemi için iki sezgisel önermişlerdir.

Gupta ve diğ. (2002) melez akış tipi çizelgeleme problemini kontrol edilebilir işlem zamanları ve teslim tarihleri durumu için ele almışlar ve bu probleme yönelik olarak ekleme tabanlı bir algoritma önermişlerdir. Thornton ve Hunsucker (2004) standart melez akış tipi çizelgeleme problemini sınırlı ara stok alanları durumu için ele almışlardır. Önerdikleri yaklaşım birçok alternatif çizelge üretip bunların arasından üretim süresini minimum yapacak olanı Johnson kuralını kullanarak seçen bir sezgiseldir. Soewandi ve Elmaghraby (2001) üç aşamalı esnek akış tipi çizelgeleme problemi için dört sezgisel önermişlerdir. Bunlardan üç tanesi işleri bölmeye dayalı bir yaklaşım sergilemektedir. Öncelikle ilk iki aşama ele alınır ve daha sonra üçüncü aşama değerlendirilir. Dördüncü sezgisel ise bu üç aşamalı problem için en doğru iş sırasını bulmak ve makinelere atamak için kullanaktadır. Cheng ve diğ. (2001) darboğazların yer değiştirmesini esas alan bir yaklaşım önermişlerdir. Buna göre öncelikle k -aşamalı melez akış tipi çizelgeleme problemi alt probleme ayrılmakta daha sonra ise her aşama kendi içinde tek tek çözülmektedir. En sonunda ise her

aşama, şimdiye kadar bulunmuş en iyi çizelge ile kıyaslanarak yeniden çizelgelenmektedir.

Kurz ve Askin (2003) melez akış tipi çizelgeleme problemini hazırlık zamanları var olma durumu için ele almışlardır. Üç sezgiseli kıyaslamışlardır: işlerin makinelere basit bir şekilde atanması prensibine dayalı periyodik sezgisel, NEH sezgiselinin bir uyarlaması olan ekleme tabanlı sezgisel ve Johnson kuralı tabanlı sezgisel. Sonuç olarak Johnson kuralı sezgiselinin gerçek zamanlı karar verme durumu söz konusu olduğunda iyi bir alternatif olduğu sonucunu ortaya koymuşlardır. Soewandi ve Elmaghraby (2003) iki aşamalı melez akış tipi çizelgeleme problemi için 3 prosedür önermişlerdir. Birinci sezgisel, simetrik işlem zamanları söz konusu olduğu durumda diğer iki sezgisel ise simetrik olmayan işlem zamanlarının söz konusu olduğu durumlar için iyi sonuç vermektedir. Suresh (1997) iki aşamalı melez akış tipi çizelgeleme problemi için üç aşamalı bir prosedür önermiştir. İlk olarak aşama birdeki işler çizelgelenir, ikinci olarak işlerin ayrılış zamanları hesaplanır, üçüncü olarak ise ikinci aşamadaki işler ayrılış zamanlarına göre çizelgelenir.

2.1.3 Metasezgisel yöntemler

Basit deterministik sezgisellere rassallığın eklenmesi ile metasezgisel yaklaşımlar elde edilmiştir. Melez akış tipi çizelgeleme probleminin çözümünde sıkça kullanılan metasezgiseller Tavlama Benzetim, Tabu Arama, Genetik Algoritmalar, Karınca Kolonisi Optimizasyonu, Yapay Sinir Ağları ve Yapay Bağışıklık Sistemi sezgiselleridir. Bu sezgiseller genel olarak arama uzayındaki alternatif permutasyonların daraltılması prensibine dayanır. Temel fikir n işi makineler atayacak en doğru permutasyonu bulmaktır (Ruiz ve Vazquez-Rodriguez, 2010). Haouari ve M'Hallah (1997) iki aşamalı esnek akış tipi çizelgeleme problemi için bir tabu arama ve tavlama benzetim algoritması önermişlerdir. Önerdikleri tabu arama algoritması çözdükleri problemlerin %35'inde optimum sonuç vermiştir.

Nowicki ve Smutnicki (1998) melez akış tipi çizelgeleme problemi için mevcut çözümün yakın çevresini ekleme hareketleri ile keşfeden bir tabu arama algoritması önermişlerdir. Wardono ve Fathi (2004) melez akış tipi çizelgeleme problemi ve ara stok alanının kısıtlı olduğu durum için bir tabu arama algoritması önermişlerdir. Bu algoritmanın en önemli özelliği arama uzayındaki olası permutasyon vektörlerini sınırlandırabilmesidir. Önerdikleri algoritma çizelgenin son halini oluşturmak için ise

yapıcı bir prosedür kullanmaktadır. Garcia ve Lozano (2005) iki aşamalı bekleme olmayan melez akış tipi çizelgeleme problemi için bir tabu arama algoritması önermişlerdir. Logendran ve diğ. (2006) hazırlık zamanlarının bulunduğu durum için melez akış tipi çizelgeleme problemine önerdikleri tabu arama algoritması iki arama düzeyinden oluşmaktadır. Buna göre ilk olarak en iyi iş ailesi sırası belirlenmekte daha sonra her bir iş ailesi için en iyi iş sırası bulunmaktadır.

Akrami ve diğ. (2006) melez akış tipi üretim ortamında çok ürünlü parti büyüklüğü belirleme ve çizelgeleme problemi için bir tabu arama algoritması önermişlerdir. Jin ve diğ. (2006) melez akış tipi çizelgeleme problemi için iki tavlama benzetim algoritması önermişlerdir. Bu algoritmalar birbirlerinden işleri atamak için kullandıkları kurallarla ayrılırlar. Janiak ve diğ. (2007) melez akış tipi çizelgeleme problemi için tavlama benzetim; tabu arama ve bu iki metodu beraber kullanan melez bir algoritma önermişlerdir. Melez algoritma diğerlerine göre daha iyi sonuç vermektedir. Tavlama benzetim algoritması; özellikle melez akış tipi çizelgeleme probleminde hazırlık sürelerinin olduğu durum için kullanılmıştır (Ruiz ve Vazquez-Rodriguez, 2010). Low (2005) hazırlık sürelerinin bulunduğu durum için toplam akış süresini minimize eden bir tavlama benzetim algoritması önermiştir. Allahverdi ve Al-Anzi (2006) melez akış tipi çizelgeleme problemi için birçok sezgiselle beraber tavlama benzetim algoritmasını da önermişlerdir.

Jin ve diğ. (2002) devre panosu üreten bir montaj hattında üretim süresini bir genetik algoritma kullanarak minimize etmeye çalışmışlardır. Sivrikaya Serifoglu ve Ulusoy (2004) ara stok alanı kısıtlı durumda melez akış tipi çizelgeleme problemi için bir genetik algoritma önermişlerdir. Önerdikleri genetik algoritma ilk aşama için işlerin sıralarının olduğu bir permutasyon üretmekte, sonraki aşamalar için işler artan üretim sürelerine göre sıralanmaktadır. Oguz ve Ercan (2005) ise aynı problem için geliştirdikleri genetik algoritmada bu problemi daha iyi açıklayabilmek için yeni bir çaprazlama operatörü kullanmışlardır. Oguz ve diğ. (2003b) iki aşamalı melez akış tipi çizelgeleme problemi için yer değiştirmeli bir paralel genetik algoritma önermiştir. Bu genetik algoritmada popülasyon alt gruplara bölünmüş ve her alt grubun bir ardışık genetik algoritma ile çözülmesi amaçlanmıştır. Daha sonra alt popülasyonlar arasında bireylerin yer değiştirmesi ile çeşitlilik sağlanmıştır. Paralel genetik algoritma ile hesaplama süresi düşürülmüş ve çözüm kalitesi artırılmıştır.

Belkadi ve diğ. (2006) de melez akış tipi çizelgeleme problemi için yer değiştirmeli bir paralel genetik algoritma önermişlerdir. Bu algoritmanın ardışık uygulamalı versiyondan daha iyi sonuç bulduğunu göstermişlerdir. Ying and Lin (2006) melez akış tipi çizelgeleme problemi için bir karınca kolonisi sistemi sezgiseli önermişlerdir. Alisantoso ve diğ. (2003) ve Engin ve Doyen (2004) melez akış tipi çizelgeleme problemini çözmek için yapay bağışıklık sistemi algoritmasını önermişlerdir. Gupta ve diğ. (2000) iki aşamalı melez akış tipi çizelgeleme problemini çözmekte kullanılacak en iyi sezgiseli seçmek için yapay sinir ağı kullanmışlardır. Wang ve diğ. (2003) melez akış tipi çizelgeleme problemi için bir yapay sinir ağı önermişlerdir. Buna göre her iterasyonda mevcut alt sınır ile yeni bulunan çözüm kıyaslanmaktadır. Ağırlıkların modifikasyonu ise alt sınır ile mevcut çözüm arasındaki farkın bir fonksiyonu şeklinde yapılmaktadır. Tavakkoli-Moghaddam ve diğ. (2009) aynı problem için iç içe geçmiş değişken komşu aramalı bir memetik algoritma önermişlerdir. Alaykyran ve diğ. (2007) melez akış tipi çizelgeleme problemi için bir karınca kolonisi optimizasyonu algoritması önermişlerdir. Kurz ve Askin (2004) melez akış tipi çizelgeleme problemi için bir rassal anahtar genetik algoritması önermişlerdir. Bu algoritma rassal sayıları çözümü oluşturacak anahtarları sıralamak için kullanmıştır. Zandieh ve diğ. (2006) aynı problem için önerdikleri yapay bağışıklık sistemi algoritması rassal anahtar genetik algoritmasından daha iyi çözüm vermektedir

2.2 Tepkisel Çizelgeleme ve Dinamiklik

Tepkisel çizelgeleme giderek önem kazanmakta ve her geçen gün dinamik ortamda çizelgeleme metotları ve stratejileri ile ilgili yayınlar literatürde yer almaktadır. Belirsizliğin olduğu dinamik ve stokastik ortamlarda çizelgeleme ile ilgili çalışmalar geliştirdikleri yöntemlere göre; analitik yaklaşımlar, simülasyon tabanlı yaklaşımlar, yapay zeka ve bilgi tabanlı yaklaşımlar ve melez yaklaşımlar olarak dört ana başlıkta gruplanabilir.

2.2.1 Analitik yaklaşımlar

Dinamik üretim ortamı ile başa çıkabilmek için geliştirilmiş tepkisel çizelgeleme ile ilgili ilk çalışmalar, analitik yaklaşımlardan yararlanmışlardır. Analitik yaklaşım, problemin bir dizi statik ya da deterministik alt probleme bölünerek bu problemlerin

çözülmesi ve daha sonra toplam çözümün elde edilmesi şeklindedir. Bu alandaki ilk çalışma Nelson ve Holloway (1977), tarafından atölye tipi üretim yapan bir kesikli üretim ortamı için önerilmiş olan bir çizelgeleme sistemidir. Bu çalışmayı Muhlemann ve diğ. (1982) ve Yamamoto ve Nof (1985) çalışmaları takip etmiştir. Çalışmalarında rastgele makine bozulmalarının olduğu durum için bir yeniden çizelgeleme algoritması araştırmışlardır. Daha sonra, Bean ve diğ. (1991) örtüşen (matchup scheduling) yaklaşımı önermişlerdir. Buna göre; dinamik ortamda ilk çizelgeyi bozucu bir durum oluştuğunda, çizelge yeniden düzenlenerek ilk çizelgedeki durum ileriki bir zamanda yeniden sağlanmaya çalışılmaktadır.

Wu ve diğ. (1993), makine bozulması olduğu durumun hemen ertesinde yapılmamış bütün işler için bir yerel arama sezgiseli geliştirmişlerdir. Church ve Uzsoy (1992) dinamik bir üretim ortamında tek makine için yeniden çizelgeleme durumunu ele almışlardır. Periyodik ve olay güdümlü çizelgeleme politikalarını geliştirmişlerdir. Ovacık ve Uzsoy (1994) 'da çizelgeleme problemini bir dizi alt probleme bölerek ayrı ayrı çözen ve daha sonra birleştiren yaklaşımlar üzerinde durmuşlardır. Sabuncuoğlu ve Karabuk (1999) esnek üretim sistemlerinde çizelgeleme ve yeniden çizelgeleme problemi üzerinde durmuşlardır. Makine bozulmalarının ve işlem zamanı değişkenliklerinin olduğu durum için bir filtrelenmiş yayılım algoritması (filtered beam search) önermişler ve birçok tepkisel çizelgeleme politikasını test etmişlerdir. Daha sonra, Sabuncuoğlu ve Bayiz (2000) tepkisel çizelgeleme problemlerini birçok farklı üretim ortamı durumu için incelemişlerdir. Geliştirdikleri çizelgenin etkinliği toplam gecikme ile ölçülmüştür.

Guo ve Nonaka (1999) melez akış tipi çizelgeleme problemini makine bozulmalarının olduğu durum için ele almıştır. Maksimum üretim süresinin minimizasyonunu sağlayacak bir analitik yaklaşım geliştirmişlerdir. Buna göre çizelgeyi zaman aralıklarına bölerek her zaman aralığı olası bozulmalar karşısında yeniden çizelgelenmiştir.

Literatürde dinamik ve belirsiz ortamlar için bir diğer çözüm önerisi güçlü ya da sağlam bir çizelgeleme metodunun geliştirilebilmesidir (Leon ve diğ. 1994; Daniels ve Kouvelis, 1995; Wu ve Stroter, 1994). Amaç geliştirilen çizelgeleme sisteminin performansının olası bozulmalar karşısında çok fazla dalgalanmasını önlemektir. Mehta ve Uzsoy (1998) ve Mehta ve Uzsoy (1999) değişken dinamik üretim

ortamında beklenmedik olaylar karşısında ilk çizelgenin çok fazla bozulmaya uğramadığı sabit bir çizelgeleme sistemi önermişlerdir.

2.2.2 Simülasyon tabanlı yaklaşımlar

Belirsizliğin olduğu dinamik üretim ortamları için geliştirilen tepkisel çizelgeleme yaklaşımlarından bir diğeri kesikli olay simülasyonudur. Kim ve Kim (1994) dinamik üretim ortamları için simülasyon tabanlı bir yaklaşım önermiştir. Önerdikleri sistem iki temel öge içermektedir: simülasyon mekanizması ve tepkisel kontrol mekanizması. Simülasyon mekanizması alternatif çözüm önerilerini performans kriterlerine göre değerlendirmekte ve en iyi olanı seçmektedir. Jeong ve Kim (1998) dinamik ortamlar için daha önce önerdikleri yöntemi geliştirmişlerdir. Ekdikleri modüllerle birçok yeniden çizelgeleme politikasını test edebilmektedirler. Kutanoglu ve Sabuncuoglu (2001a) dinamik ortamlara önerdikleri simülasyon tabanlı çizelgeleme sistemi ile yeni tepkisel algoritmaları deneyebilmişlerdir. Önerdikleri sistem iteratif simülasyonun uygulanması açısından önemli bir örnektir. Bütün bu simülasyon sistemlerinin ortak özelliği, çizelgeleme mekanizmasının dağıtım kuralları tabanlı olmasıdır. En iyi çizelgeleme kuralı belirtilen kritere göre simülasyonla seçilir ve bir sonraki çizelgeleme noktasından itibaren uygulanır. Seçim mekanizması ya belli zaman periyotlarında düzenli olarak ya da makine bozulmaları gibi mevcut durumu bozan beklenmedik durumlarda gerçekleşir.

Sabuncuoglu ve Kizilisik (2003), dinamik ve stokastik bir esnek üretim sistemi için simülasyon tabanlı bir tepkisel çizelgeleme sistemi önermişlerdir. Rassallık altında üretim çizelgesini gözden geçirip değişkenliklere bozulmalara cevap verecek bir simülasyon tabanlı çizelgeleme sistemi geliştirmişlerdir. Dinamik ve stokastik esnek üretim ortamı simülasyonla elde edilmiştir. Kutanoglu ve Sabuncuoglu (2001b) atölye tipi bir üretim ortamında makine bozulmalarının olduğu durum için dört tepkisel stratejiyi incelemişlerdir. Sonuçlar, önerilen bütün tepkisel politikaların makine bozulmaları olduğu durumlarda iyi rotalar üreterek tutarlı çizelgeler oluşturduklarını göstermiştir. He ve diğ. (1994), Lawrence ve Sewell (1997), Svetska ve Abumaziar (1997), Lejmi ve Sabuncuoglu (2002) stokastik olayların ve değişkenliklerin çizelgeleme kuralları üzerine olan etkilerini incelemişlerdir.

2.2.3 Yapay zeka ve bilgi tabanlı yaklaşımlar

Yapay zeka ve bilgi tabanlı sistemlerin dinamik ortamlarda çizelgeleme için kullanılması literatürde önemli bir yer tutmaktadır. Bu yaklaşımların temel motivasyon kaynağı, her çizelgeleme sistemi kendine özgü spesifik şartlar taşır ve bu yüzden bu tür sistemleri etkili bir şekilde çizelgeleyebilmek için geniş teknik ve uzmanlık gerektiren bilgi, sistem spesifik bakış açısı gerekir. Fox ve Smith (1984), Smith ve diğ. (1990), Shaw ve diğ. (1992) ve Dutta (1994) dinamik ortamlarda çizelgeleme için yapay zeka ve bilgi tabanlı yaklaşımları kullanmışlardır. Szelke ve Kerr (1994), yapay zekanın tepkisel çizelgeleme ile kullanılmasına yönelik daha detaylı bir çalışma yapmışlardır. Sabuncuoglu (1998) yapay sinir ağlarının kullanımı ile ilgili olan çalışmaları özetlemiştir.

2.2.4 Melez yaklaşımlar

Literatürde tepkisel çizelgeleme ile ilgili birden çok yaklaşımın bir arada kullanıldığı ya da sezgisel, metasezgisel yöntemler, proaktif-reaktif yaklaşımlar gibi dinamik üretim ortamı için alternatif çizelge geliştirmeye yönelik yayınların sayısı giderek artmaktadır. Mendez ve Cerda (2004) yığın üretim yapan, çok aşamalı bir üretim sistemi için makine bozulmaları, yeni gelen işler ve malzeme azlığı gibi belirsizliklerin olduğu durum için ortalama gecikmeyi minimize edecek karışık tamsayı tabanlı bir sistem önermiştir. Önerdikleri sistem mevcut çizelgeyi beklenmedik durumlar karşısında gözden geçirip güncellemektedir. Bu yaklaşım bütün çizelgeyi en baştan çizelgelemeyi önleyecek zaman kaydırma, kaynakların yeniden tahsisi ve yerel yığınların yeniden sıralanarak kaynaklardan daha iyi yararlanılabilmesi gibi parçalı çizelge uygulamaları yapabilmektedir.

Sawik (2007) melez akış tipi üretim sistemi için ortalama gecikmeyi minimum yapacak karışık tamsayı tabanlı bir sezgisel önermiştir. Üretim çizelgesi yeni bir sipariş geldiğinde ya da sipariş değiştiğinde hizmet seviyesi ve envanter kısıtları dikkate alınarak güncellenmektedir. Duenas ve Petrovic (2007) paralel makine çizelgeleme problemine malzeme azlığı durumu için tahminsel-tepkisel bir yöntem önermişlerdir. Tahminsel yaklaşım olası bozulmaların etkisini giderebilmek için üretim çizelgesinde bir miktar boşluk bırakmayı gerektirir. Bazı parametre değerlerinin belirlenmesinde bulanık kümeler teorisinden yararlanmışlardır. Yang ve Geunas (2008) tek makine çizelgeleme problemi için beklenmedik işlerin olduğu

durumda ağırlıklandırılmış gecikmeyi minimize edecek bir tepkisel algoritma önermişlerdir. Önerdikleri yöntem çizelgeye yerleştirilecek olan boşluğun boyutunu optimize edecek bir dinamik programlama tabanlı sezgiseldir. Algoritma çizelgedeki bozulmaları önleyecek şekilde çalışarak gecikmeyi minimize eder.

Morandin ve diğ. (2008) üretim miktarı dalgalanmalarının olduğu bir melez üretim sistemi için maksimum üretim süresini minimum yapan bir yöntem önermişlerdir. Önerdikleri yöntem genetik algoritma tabanlıdır, değişken durumlar karşısında algoritma bu durumlara göre tepkiler vererek uyarlanabilmektedir. Turkcan ve diğ. (2009), makine bozulmaları, yeni gelen işler ve malzeme azlığının olduğu bir dinamik ortamda paralel makine çizelgeleme problemini ele almışlardır. Toplam üretim maliyetini ve toplam ağırlıklandırılmış erken gelme ve gecikmeyi minimize etmeyi amaçlamışlardır. Bu problem için zaman indeksli bir tamsayı programlama modeli önermişlerdir. Bu modeli çözmek için lineer programlamanın gevşetilmesine dayanan iki aşamalı bir algoritma önermişlerdir.

Gürel ve diğ. (2010) makine bozulmalarının olduğu dinamik bir ortamda paralel makine çizelgeleme problemi için üretim maliyetini minimize etmeyi amaçlayan örtüşen strateji tabanlı bir sezgisel algoritma önermişlerdir. Gomes ve diğ. (2010) siparişe göre üretim yapan atölye tipi bir üretim ortamını ele almışlardır. Yeni gelen işlerin olduğu durum için tepkisel bir çizelgeleme sistemi önermişlerdir. Toplam ağırlıklı gecikme ve erken gelmeyi bir tamsayı programlama modeli ile minimize etmeye çalışmışlardır. Çizelgeleme dönemlerini kaydırarak modeli iteratif bir şekilde çözen ve yeni gelen siparişleri çizelgeye ekleyen tepkisel bir çizelgeleme algoritması önermişlerdir. Tanimizu ve diğ. (2010) makine bozulmaları acil işler gibi üretim sürecini geciktiren etkenlerin bulunduğu atölye tipi bir üretim sistemi için maksimum üretim sistemini minimize eden evrimsel algoritmalarından yararlanan bir tepkisel algoritma önermişlerdir.

Wu ve diğ. (2010) işlem zamanı değişkenlikleri, yeni gelen işler, acil işler gibi durumların olduğu dinamik bir atölye tipi üretim çizelgeleme problemi için tepkisel bir algoritma geliştirmişlerdir. Bu problem için, planlanan durumla mevcut durum arasındaki farkı minimize etmeye çalışan bir genetik algoritma önermişlerdir. Lou ve diğ. (2011) işlem zamanlarının belirsiz olduğu bir üretim ortamı için maksimum üretim süresini minimize eden ajan tabanlı bir proaktif-reaktif çizelgeleme algoritması önermişlerdir. Proaktif aşamada olası belirsizliklere hazırlıklı bir stabil

izelge retilmesi amalanmıř, reaktif ařamada ise mevcut izelge zerinde oluřabilecek olası bozulmalara karřı eřitli tepkisel mekanizmalarla özm retmeye alıřmıřlardır. Stuart ve Kozan (2011) bir ameliyathane sistemini tek makine izelgeleme problemi gibi ele alıp dinamik bir ortamda izelgelemeye alıřmıřlardır. Ekipman eksiklięi ya da bozulması, ani vaka gelmesi gibi dinamik bir ortam iin belirlenen tarihten nce tamamlanan ameliyat sayısını maksimize eden bir dal sınır algoritması ile tepkisel izelgeleme problemine özm aramıřlardır. Vandevonder ve dię. (2007) iřlem zamanı deęiřkenliklerinin bulunduęu proje izelgeleme problemi iin tepkisel bir izelgeleme yntemi nermiřlerdir. Bu yntem, eřitli izelgeleme ve nceliklendirme kurallarını birlikte kullanarak mevcut durumdan sapmaları minimize etmeyi amalamıřtır.

Gomes ve dię. (2006) ok amalı yıęın retim yapan bir retim ortamı iin yeni gelen sipariřlerin olduęu durum iin karma tamsayı tabanlı bir model nermiřlerdir. Erken gelme ve gecikmeyi minimize etmeyi amalamıřlardır. Wang ve dię. (2008) makine bozulmalarının olduęu yıęın retim yapan bir retim sistemi iin gecikmeyi minimize edecek bir tepkisel algoritma nermiřlerdir. nerdikleri algoritma proses planlaması ve yeniden izelgelemenin entegrasyonu ařamalarından oluřmaktadır. Kato ve dię. (2009) makine bozulmaları ve malzeme azlıęı gibi belirsizliklerin olduęu bir atlye tipi retim ortamı iin tepkisel bir izelgeleme algoritması nermiřlerdir. Buna gre nerdikleri yntem, karınca kolonisi optimizasyonu algoritmasından yararlanarak maksimum retim sresini minimize etmeye alıřmaktadır.

Al-Fawzan ve Haouari (2005) kısıtlı kaynakların olduęu iřlem zamanı deęiřkenlikleri durumu iin proje izelgeleme problemini ele almıřlardır. Bu problemin özm iin tabu arama algoritmasından yararlanmıřlardır. Artigues ve dię. (2005) atlye tipi bir retim ortamı iin olası bozulmalar karřısında oklu bir izelgeleme stratejisi nermiřtir. nerdikleri yaklařım esneklięin artırılarak alternatif özmlerin bozulmalar karřısında kullanılmasına imkan vermektedir. Kayan darboęaz algoritması sezgiselinden yararlanarak özm olarak nerilebilecek alternatif izelgelerin artırılması amalanmıřtır.

Kasperski (2005) iřlem zamanlarının deęiřken ve teslim tarihlerinin belirsiz olduęu durumdaki bir tek makine izelgeleme problemi iin polinomial zaman algoritması nermiřlerdir. Sevaux and Srensen (2004) tek makine izelgeleme problemini

stokastik teslim tarihleri için ele almıştır. Toplam gecikmeyi minimum yapacak olan bir genetik algoritma önermişlerdir. O'Donovan ve diğ. (1999) reaktif ve proaktif yaklaşımları bir arada kullanan tek makine çizelgeleme problemi için çizelgeleme/yeniden çizelgeleme politikası önermişlerdir.

2.2.4.1 Sezgisellere simülasyonu entegre eden yaklaşımlar

Literatürde stokastik ortamlarda yapılan çizelgeleme çalışmalarına bakıldığında, araştırmacıların problem kompleksitesini metasezgisel yaklaşımlarla, fonksiyonel kompleksiteyi ise, simülasyonu metasezgisellere entegre ederek aşmaya çalıştıkları görülmüştür. Sezgiseller yardımı ile oluşturulan alternatif çözümler, sezgiselle entegre edilmiş simülasyon modülü yolu ile değerlendirilmektedir.

Wang ve Choi (2014) stokastik işlem zamanlarının olduğu melez akış tipi çizelgeleme problemi için ayrışma tabanlı holonik bir yaklaşım önermişlerdir. Algoritma işbirlikçi holonların çözüm üretmesine dayanmaktadır. Öncelikle K-means sınıflama algoritması ile üretim sistemindeki makineler sınıflandırılmakta, sınıfların stokastikliğine göre seçilen çizelgeleme politikası ile ilgili sınıflar çizelgelenmektedir. Düşük stokastiklikteki sınıflar Genetik Algoritma Kontrol merkezli bir şekilde çizelgelenirken, yüksek stokastisiteye sahip sınıflar En Kısa İşlem Zamanı Önce tabanlı bir protokol yardımı ile çizelgelenir. 10 değişik beklenen işlem zamanı durumunun her biri için 50 simülasyon çalışması ile deneylerini gerçekleştirmişlerdir.

Zandieh ve Gholami (2009) rassal makine bozulmalarının olduğu melez akış tipi çizelgeleme problemi için bir bağışıklık algoritması önermişlerdir. Algoritmaya entegre edilmiş simülasyon modülleri yardımı ile makine bozulmaları oluşturulurken aynı zamanda bağışıklık algoritmasının önerdiği çözümlerin de beklenen değerleri hesaplanmaktadır. Almeder ve Hartl (2013) stokastikliğin bulunduğu bir melez akış tipi çizelgeleme problemi için değişen komşuluk arama algoritması önermişlerdir. Önerdikleri algoritma alternatif durumlara ait senaryoları simüle ederek çalışmaktadır.

Hornig ve diğ (2012), rassal işlem zamanlarının olduğu atölye tipi çizelgeleme problemi için sırasal optimizasyona evrimsel stratejilerini entegre eden bir evrimsel algoritma önermişlerdir. Önerdikleri algoritma stokastik simülasyonlar ile çalışmaktadır. Önerdikleri algoritmanın performansını kritik oran, erken teslim tarihi

önce, en kısa işlem zamanı önce, en uzun işlem zamanı önce, ilk gelen ilk işlem görür kuralları ile kıyaslamışlardır. Gholami ve Zandieh (2009) stokastik makine bozulmalarının olduğu melez akış tipi çizelgeleme problemi için bir genetik algoritma önermişlerdir. Algoritma alternatif çözümleri simülasyon yolu ile değerlendirmektedir. Her bir çözümün 20 deneme sonrasında beklenen uygunluk değeri bulunmaktadır.

Dinamik çizelgeleme problemlerinin kompleksitesinin yüksekliği birçok araştırmacıyı bu alanda çalışmalar yapmaya itmiştir. Yapılan yayınlara bakıldığında araştırmacıların, dinamik ortamlarla baş edebilmek için evrimsel algoritmalara başvurduğu görülmüştür. Louis ve Xu (1996), makine bozulmalarının olduğu açık atölye tipi problemini çizelgelemek için genetik algoritma kullanmışlardır. Her makine bozulması durumunda yeniden çizelgeleme yapmışlardır.

Ramsey ve Grefenstette (1993) vaka bazlı düşünme felsefesini evrimsel algoritmalara entegre etmeyi önermişlerdir. Çevresel şartların ölçülebileceğini varsaymışlardır. Popülasyonda bulunan en iyi çözüm, bilgi tabanında indekslenerek o andaki şartları karakterize eder. Ne zaman yeni bir ortama girilse evrimsel algoritmalara yeniden başlatılır.

Bierwirth ve diğ. (1995) dinamik atölye tipi çizelgeleme problemi için geliştirdikleri genetik algoritmada, üretim ortamının her değişmesi sonrasında popülasyonun yeni ortama adapte edilebilmesi için popülasyonda bulunan bütün kromozomların modifiye edilmesini önermişlerdir. Buna göre sisteme yeni bir iş geldiğinde üretim ortamı değişmekte ve popülasyon modifiye edilmektedir. Bu önerilerini de dinamik atölye tipi çizelgeleme probleminde permutasyon temsil yöntemi kullanan bir genetik algoritma ile test etmişlerdir. Yeni bir iş geldiğinde, gelen yeni iş popülasyonda o anda bulunan bütün bireylerin kromozomlarına rassal olarak eklenmektedir. Yeni gelen işle ortam değişmekte fakat popülasyondaki kromozomlar sıfırlanmak yerine önceki popülasyonun bireyelerine yeni gelen iş olarak eklenerek aramaya devam edilmektedir. Böylece arama süreci hızlandırılabilir.

Al-Hinai ve ElMekkawy (2011) rassal makine bozulmalarının olduğu melez akış tipi çizelgeleme problemi için güçlü ve sağlam bir çizelgeleme için iki aşamalı melez bir genetik algoritma önermişlerdir. Kainfar ve diğ. (2012) dinamik iş gelişlerinin olduğu melez akış tipi çizelgeleme problemine yönelik melez bir genetik algoritma

önermişlerdir. Önerdikleri algoritma komşuluk tabanlı aramalı sevk kuralı ile entegre çalışmaktadır. Ele aldıkları sisteme stokastik ve dinamik bir şekilde iş gelişi olmakta ve gelecek olay listesine kaydedilmektedir. Yeniden çizelgeleme noktasına gelindiğinde o ana kadar gelecek olay listesine kaydedilen işler popülasyonda o anda bulunan bireylerin kromozomlarına eklenerek arama sürecine devam edilmektedir.

Zhang ve diğ. (2013) rassal iş gelişlerinin ve makine bozulmalarının olduğu durum için dinamik atölye tipi çizelgeleme problemini incelemişlerdir. Bu probleme yönelik olarak genetik algoritma ve tabu arama algoritmasına dayalı melez bir yeniden çizelgeleme metodu önermişlerdir. Yeni iş gelişleri ve makine bozulmalarını simülasyon yolu ile önerdikleri algoritmaya entegre edilmiştir. Çeşitli iş gelişleri frekansları için çizelge etkinliğini ve çizelge stabilitesini analiz etmişlerdir.

Juan ve diğ. (2014) stokastik işlem zamanlarının olduğu permutasyon akış tipi çizelgeleme problemi için bir simülasyon optimizasyon algoritması önermişlerdir. Önerdikleri algoritma problemin deterministik versiyonu için önerilen iyi bir çözümün aynı zamanda problemin stokastik versiyonu için de iyi bir çözüm olabileceği varsayımına dayanmaktadır. Önerilen algoritma Monte Carlo simülasyonunu iteratif yerel arama algoritmasına entegre ederek çalışmaktadır.

2.3 Literatürün Analizi

Çizelge 2.1, dinamik ortamlarda çizelgelemeye yönelik yapılmış bazı çalışmalarını özetlemektedir.

Çizelge 2.1 : Literatürün analizi.

Referans	Atölye	Dinamiklik/Stokastiklik	Metot	Amaç Fonksiyonu
O' Donovan ve diğ. (1999)	Tek Makina	Makina Bozulmaları, İşlem Zamanı Değişkenliği	ATC türevleri ve kombinasyonları	Toplam Gecikmeyi Düşük Seviyede Tutarak Toplam Tamamlanma Zamanını Minimize Etmek
Guo ve Nanoka (1999)	Akış Tipi (3 Makina)	Tek Makina Bozulması	Analitik Analiz	Tamamlanma Zamanı Değişkenliğini Minimize Etmek
Sabuncuoğlu ve Kızılışık (2003)	Esnek İmalat Sistemi	Makina Bozulmaları, İşlem Zamanı Değişkenliği	Simülasyon Tabanlı	Sapmaları Azaltmak
Al-Fawzan ve Haouari (2005)	Proje Çizelgeleme	İşlem Zamanı Değişkenliği	Tabu Arama	En Büyük İşlem Zamanını Azaltmak Toplam Serbest Zamanı Arttırmak
Sevaux ve Sörensen (2004)	Tek Makina	Hazır Olma Zamanı	Genetik Algoritma	Beklenen Performansı ve Gerçekleşen Performansın Standart Sapmasını Minimize Etmek
Mendez ve Cerda (2004)	Çok Aşamalı Toplu Planlama	Makina Bozulmaları, Gelen Yeni İşler , Malzeme Kıtlığı	Karmaşık Tam Sayılı Doğrusal Programlama	Ortalama Gecikmeyi Minimize Etmek
Artigues ve diğ. (2005)	Atölye Tipi	Bozulma Durumlarında Uygulanan Çoklu Çizelgeleme	Kayan Darboğaz Algoritmasına Benzer Bir Sezgisel	Maksimum Esneklik

Çizelge 2.1 (devam) : Literatürün analizi.

Referans	Atölye	Dinamiklik/Stokastiklik	Metot	Amaç Fonksiyonu
Kasperski (2005)	Tek Makina	İşlem Zamanı Değişkenliği, Teslim Zamanı Değişkenliği	İteratif Olarak Lawler Algoritmasının Özel Bir Durum İçin En Kötü Vaka Durumu Analizi	Mutlak Kötü Durumları Minimize Etmek Maksimum Gecikme Farkını Minimize Etmek
Gomes ve diğ. (2006)	Çok Amaçlı Toplu Planlama	Yeni Sipariş Gelişi	Karmaşık Tam Sayılı Doğrusal Programlama	Erken Gelme ve Gecikmeyi Minimize Etmek
Sawik (2007)	Esnek Akış Tipi	Yeni Sipariş Gelişi	Karmaşık Tam Sayı Tabanlı Sezgizel Algoritma	Gecikmeyi Minimize Etmek
Duenas ve Petrovic (2007)	Paralel Makina	Malzeme Kıtlığı	Modellemede Bulanık Kümelerin Kullanılması	En Büyük İşlem Zamanını Minimize Etmek
Van de Vonder ve diğ. (2007)	Proje Çizelgeleme	İşlem Zamanı Değişkenliği	Sezgizel Yaklaşım	Tamamlanma Zamanının Sapmasını Minimize Etmek
Yang ve Geunes (2008)	Tek Makina	Belirsiz İşler	Sezgisel Yaklaşım	Boş Zaman Optimizasyonu
Morandin ve diğ. (2008)	Esnek İmalat Sistemi	Üretim Hacim Değişkenliği	Genetik Algoritma	En Büyük İşlem Zamanını Minimize Etmek
Wang ve diğ. (2008)	Yığın Üretim	Makina Bozulmaları	Süreç Planlama ve Entegrasyonun Yeniden Çizelgelenmesi	Gecikmeyi Minimize Etmek

Çizelge 2.1 (devam) : Literatürün analizi.

Referans	Atölye	Dinamiklik/Stokastiklik	Metot	Amaç Fonksiyonu
Kato ve diğ. (2009)	Atölye Tipi	Makina Bozulmaları, Malzeme Kıtlığı	Karınca Kolonisi Optimizasyonu	En Büyük İşlem Zamanını Minimize Etmek
Türkcan ve diğ. (2009)	Paralel Makina	Makina Bozulmaları, Gelen Yeni İşler , Malzeme Kıtlığı	DP-Gevşetme Tabanlı İki Aşamalı Algoritma	Toplam Ağırlıklı Erken Gelme ve Gecikmeyi Minimize Etmek
Gürel ve diğ. (2010)	Paralel Özdeş Olmayan Makinalar	Makina Bozulmaları	Sezgisel Yaklaşım (Örtüşme Stratejisi)	Üretim Maliyetini Minimize Etmek
Gomes ve diğ. (2010)	Atölye Tipi	Makina Bozulmaları, Gelen Yeni İşler , Malzeme Kıtlığı	Tamsayılı Programlama Tabanlı Algoritma	Toplam Ağırlıklı Erken Gelme ve Gecikmeyi Minimize Etmek
Tanimizu ve diğ. (2010)	Atölye Tipi	Üretim Süreci Gecikmeleri, Acil İş İlaveleri, Makina Bozulmaları	Beraber Evrimleşen Genetik Algoritmalar	En Büyük İşlem Zamanını Minimize Etmek
Wu ve diğ. (2010)	Atölye Tipi	İşlem Zamanı Değişkenliği, İş Revizyonları, Acil İş Gelişleri	Genetik Algoritma	Gerçekleşen ve Planlanan Operasyonlar Arasındaki Farkı Minimize Etmek
Lou ve diğ. (2011)	Pratik Üretim Ortamı	Belirsiz İşlem Süreleri	Çoklu Ajan Temelli	En Büyük İşlem Zamanını Minimize Etmek
Stuart ve Kozan (2011)	Tek Makina Problemi	Rastgele Hasta Gelişi (İş Gelişi)	Dal ve Sınır	Zamanında Tamamlanması Beklenenlerin Maksimizasyonu

Literatür analiz edildiğinde belirsizliğin olduğu dinamik ortamlarda melez akış tipi çizelgeleme problemi çalışmalarının sınırlı sayıda olduğu ve yeterince üzerinde durulmadığı görülmüştür.

Bu çalışma kapsamında önerilen yaklaşım literatürdeki diğer çalışmalardan sürekli iyileştirme ve çizelgeleme sürecini bir bütün olarak ele alma noktaları ile farklılaşmaktadır. Önerilen algoritma sürekli arama yapan bir yapıdadır ve değişkenliğin olduğu ana kadar ki arama süreci kazanımlarını değişkenlik sonrası güncellenen ortama aktararak arama sürecini hızlandırabilmektedir. Önerilen algoritma küçük küçük ama sürekli iyileştirmelere güvenmekte, dinamik ve sürekli değişen bir üretim sistemine yönelik olarak hızlı ve efektif çözümler üretmektedir.

Ayrıca önerilen çalışma, literatürdeki çalışmalardan çizelgeleme sürecini bir bütün olarak ele alma ve bu sürece yönelik olarak önerdiği dinamik çizelgeleme sistemi ile ayrılmaktadır. Önerilen algoritma çözüm üretilmesi için geçen zamanı da çizelgeleme sürecine yansıtarak çalışan, tamamlanma zamanını temel alan bir yapıdadır. Bu açıdan bakıldığında önerilen çizelgeleme sistemi, geliştirilen algoritmanın simülasyonundan ibaret olmayıp, gerçek hayattaki herhangi bir dinamik üretim sistemine entegre edilerek çizelgeleme yapabilecek yapıdadır.

3. ÇİZELGELEME ORTAMI

Bu tez çalışması kapsamında dinamik bir üretim sisteminin incelenmesi amaçlanmıştır. Dinamik sistemlerde bir işin sisteme ne zaman geleceği ya da bu işin işlenmesi için ne kadar süre gerektiği önceden kesin olarak bilinmez. Bu çalışma kapsamında da ele alınan üretim sistemlerinde işlerin sisteme geliş zamanları önceden belli değildir, işlerin sisteme rassal olarak geldiği düşünülecektir. Ayrıca işlerin işlem zamanı değişkenlik gösterebilmektedir, işlem zamanları deterministik değil stokastiktir.

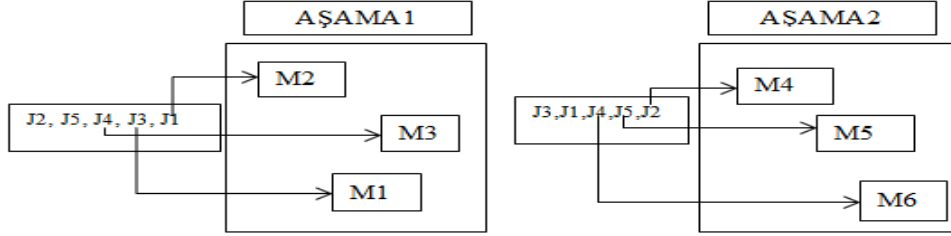
Bu açıdan bakıldığında dinamik üretim sistemlerinde çizelgeleme yapmak ucu açık bir zaman aralığında deterministik olmayan bir problemle baş etmeye çalışmaya benzemektedir. Beklenmedik olaylar ve yeni iş gelişleri sebebi ile üretim ortamı değişmekte ve dolayısı ile sık sık çizelge revizyonları gerekmektedir.

Bu tez çalışması kapsamında ele alınması düşünülen üretim sistemi gerçek hayatta birçok uygulaması olan melez akış tipi üretimdir. Melez Akış Tipi Üretim, n işin birbirini takip eden m aşamada işlem görmesini esas alır. Melez Akış Tipi Üretim sistemlerinin birçok farklı versiyonu olmakla beraber Melez Akış Tipi üretim sistemlerinin büyük bir çoğunluğu için aşağıdaki karakteristik özellikler ortaktır (Ruiz ve Vazquez-Rodriguez, 2010) :

- İşlem yapılan aşama sayısı m , en az ikidir.
- Her k işlem aşamasında en az 1 tane işlem yapan makine vardır, ve aşamaların en az bir tanesi birden çok paralel makineden oluşur.
- Bütün işler aynı üretim akışını takip ederek işlem görür: aşama 1, aşama 2, ... aşama m . Herhangi bir iş, en az bir aşamada işlem görmek şartı ile bir veya birden çok aşamayı atlayabilir.

Problemin en genel hali olan Standart Melez Akış Tipi Çizelgeleme probleminde bütün işler ve makineler sıfır anında mevcuttur. Seçilen bir aşama için o aşamada işlem yapan makineler özdeştir. Herhangi bir makine herhangi bir anda sadece bir iş

işleyebilir. Herhangi bir iş herhangi bir zamanda sadece bir makine tarafından işlenebilir. İşlerin hazırlık ve transfer zamanları ihmal edilir. İşlerin yarıda kesilmesine izin verilmez. Aşamalar arasındaki ara stok alanlarının limitsiz olduğu varsayılır. Probleme herhangi bir stokastik unsur yoktur ve problem deterministiktir. Şekil 3.1’de standart bir melez akış tipi üretim sistemi gösterilmiştir.



Şekil 3.1 : Melez akış tipi çizelgeleme problemi.

3.1 Standart Melez Akış Tipi Çizelgeleme Problemi Matematiksel Modeli

Standart Melez Akış Tipi Çizelgeleme problemi karma tamsayılı matematiksel model olarak modellenenir (Naderi ve diğ, 2014):

İndisler:		
İşler	j, k	$= 1, 2, \dots, n$
Aşamalar	i	$= 1, 2, \dots, m$
Makineler	l	$= 1, 2, \dots, m_i$
Parametreler:		
n	: Toplam iş sayısı.	
m	: Toplam aşama sayısı.	
m_i	: i aşamasındaki makine sayısı.	
$p_{j,i}$: j işinin i . aşamadaki işlem zamanı.	
M	: Çok büyük bir sayı.	
Değişkenler:		
$X_{j,i,k}$: j işi i aşamasında k işinden sonra işlenirse 1, değilse 0.	
$Y_{j,i,l}$: j işi i aşamasında l makinesinde işlenirse 1, değilse 0.	
$C_{j,i}$: j işinin i aşamasındaki tamamlanma zamanı	
Kümeler:		
E_i	: i aşamasında işlem gören işler	

Şekil 3.2 : Standart melez akış tipi çizelgeleme problemi matematiksel model notasyonu.

$$Z = \text{enk } C_{max} \quad (3.1)$$

öyle ki:

$$\sum_{l=1}^{m_i} Y_{j,i,l} = 1, \quad \forall_{i,j \in E_i} \quad (3.2)$$

$$C_{j,i} \geq C_{j,i-1} + p_{j,i}, \quad \forall_{j,i} \quad (3.3)$$

$$C_{j,i} \geq C_{k,i} + p_{j,i} - M \cdot (3 - X_{j,i,k} - Y_{j,i,l} - Y_{k,i,l}), \quad \forall_{i,l,(j,k) \in E_i} \quad (3.4)$$

$$C_{k,i} \geq C_{j,i} + p_{k,i} - M \cdot X_{j,i,k} - M \cdot (2 - Y_{j,i,l} - Y_{k,i,l}), \quad \forall_{i,l,(j,k) \in E_i} \quad (3.5)$$

$$C_{max} \geq C_{j,m}, \quad \forall_j \quad (3.6)$$

$$C_{j,i} \geq 0, \quad \forall_{j,i} \quad (3.7)$$

$$X_{j,i,k} \in \{0,1\}, \quad \forall_{i,(j,k) \in E_i} \quad (3.8)$$

$$Y_{j,i,l} \in \{0,1\}, \quad \forall_{i,l,j \in E_i} \quad (3.9)$$

(3.1) sistemdeki herhangi bir işe ait en büyük tamamlanma zamanını en küçüklemeyi ifade eder ve amaç fonksiyonudur. (3.2) nolu ifade bir işin herhangi bir aşamada yalnızca bir makinede işlem görmesini sağlar. (3.3) nolu ifade herhangi bir aşamadaki bir işin o aşamadaki asgari tamamlanma zamanının önceki aşamadaki o işin bitiş zamanından büyük ya da eşit olmasını garanti eder. (3.4) ve (3.5) nolu ifadeler herhangi bir aşamadaki bir işin asgari tamamlanma zamanının o işin işlendiği makinede daha önce işlenen işlere bağlı olarak belirlenmesini sağlar. (3.6) nolu ifade işlem gören işlere ait en büyük tamamlanma zamanını belirler. (3.7), (3.8) ve (3.9) nolu ifadeler karar değişkenlerinin tanım kümelerini gösterir.

3.2 Problem Karmaşıklığı

Melez akış tipi çizelgeleme probleminin birçok durumda NP-zor olduğu görülmüştür (Ruiz ve Vazquez-Rodriguez, 2010). Gupta (1988)'nin yaptığı çalışmanın sonuçları bir aşamasında iki makine diğer aşamasında tek makine olan iki aşamalı bir melez

akış tipi üretim sisteminin NP-zor olduğunu göstermektedir. Benzer şekilde Hoogeveen ve diğ. (1996)'nin yaptığı çalışmanın sonuçları işlerin yarıda kesilmesine izin verilen iki aşamalı bir melez akış tipi çizelgeleme probleminin NP-zor olduğunu göstermiştir.

Bu tez çalışması kapsamında işlerin sisteme rassal olarak geldiği, aşamalarda işlem yapan makinelerin özdeş olmadığı, işlerin yarıda kesilmesine izin verilmeyen, işlem zamanlarının önceden bilinmediği (stokastik işlem zamanları) dinamik bir melez akış tipi üretim sistemi incelenmiştir. Sadece iki aşamalı ve deterministik karakterdeki birçok melez akış tipi çizelgeleme probleminin dahi NP-zor olduğu birçok çalışmada gösterilmiştir. Bu tez çalışması kapsamında incelenecek olan üretim sistemleri dinamik stokastik karakter gösteren, ikiden fazla aşamalı, özdeş olmayan makinelerden oluşan melez akış tipi çizelgeleme problemleridir. Bu açıdan bakıldığında bu tez çalışmasında ele alınacak olan problemlerin NP-zor problemler olduğu söylenebilir.

3.3 Performans Ölçütü

Bu tez çalışması kapsamında ele alınan melez akış tipi çizelgeleme probleminin dinamik yapısı gereği, sisteme farklı zamanlarda iş gelişi olmaktadır. Bu sistem içerisinde geçirilen ortalama süre dinamik karaktere sahip bu problemin performansının değerlendirilmesinde uygun bir ölçüt olabilir.

Sisteme gelen bir j işinin geliş zamanı r_j , t anına kadar sisteme gelen işler kümesi ise J_t ile gösterilebilir. Bu sistemdeki işlerin, sisteme geliş zamanlarına göre artan bir şekilde indekslendiği varsayılmıştır. Sisteme gelen iş sayısı sonsuz kabul edilmiştir. Sisteme bir j işi geldiğinde bu işe bir beklenen işlem zamanı \hat{p}_j atanır. ψ bir çizelgeyi temsil eder. j işinin ψ çizelgesindeki başlangıç zamanı $s_j(\psi)$ ile gösterilir. Bir j işinin ψ çizelgesindeki beklenen tamamlanma zamanı $C_j(\psi) = s_j(\psi) + \hat{p}_j$ dir.

s_j^G , bir j işinin gerçekleşen işleme başlama zamanını, C_j^G gerçekleşen tamamlanma zamanını temsil eder. p_j^G , bir j işinin gerçekleşen işlem zamanını gösterir ve işlem zamanları deterministik olmadığı için $p_j^G \neq \hat{p}_j$ dir.

Sistemdeki işlerin geliş zamanları birbirlerinden farklı olduğu için, bu sistem için ortalama akış zamanı iyi bir performans ölçütü olabilir. Örneğin bu sisteme yönelik olarak en büyük tamamlanma zamanı (C_{max}) performans ölçütü olarak seçilirse,

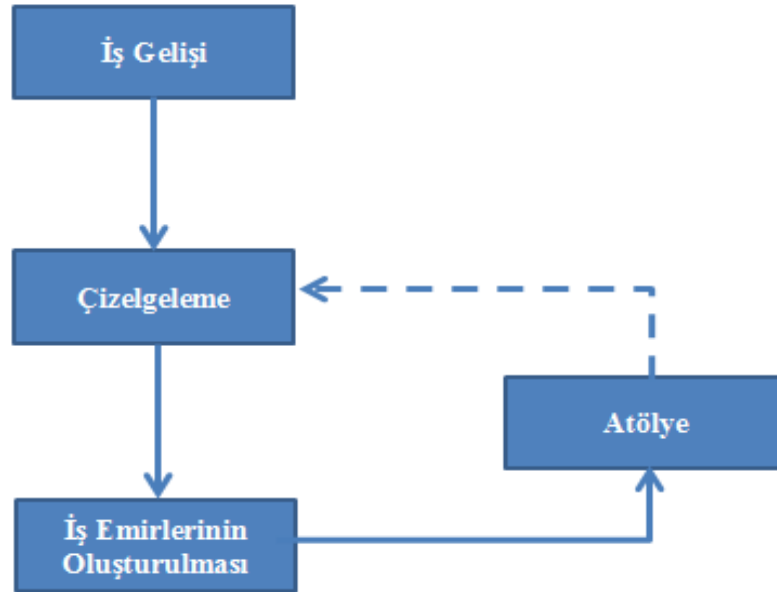
sistemin performansı büyük ölçüde sisteme gelen son işin geliş zamanına bağlı olacaktır. Çözüm olarak önerilen alternatif iki çizelgenin performansı arasındaki fark yeterince iyi ölçülemeyebilir. Her iki çizelgenin performansı da sisteme gelen son işin sistemden çıkış zamanına bağlı olacak, daha önce sistemde bulunan işlerin çizelge performansı üzerindeki etkisi yeterince iyi değerlendirilemeyecektir. Bir j işine ait gerçekleşen akış zamanı, $F_j^G = C_j^G - r_j$ ile tanımlanabilir. Bu sistemin gerçekleşen performansı,

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n F_j^G}{n} \quad (3.10)$$

ile ölçülebilir. Bu tez çalışması kapsamında yukarıda karakteristik özellikleri tanımlanan bir üretim sisteminin çizelgelenmesi amaçlanmaktadır. Buna göre üretilen sonuçların performansı, ortalama akış zamanı ile değerlendirilecektir. Bu kapsamda çizelgeleme sürecinin analiz edilmesi uygun olacaktır.

3.4 Çizelgeleme Süreci

Bu tez çalışmasında, dinamik bir üretim sistemi için efektif bir çizelgeleme algoritması tasarlanması amaçlanmaktadır. Dinamik üretim sistemlerinde çizelgeleme süreci Şekil 3.3'teki gibi özetlenebilir.



Şekil 3.3 : Dinamik çizelgeleme süreci.

Sisteme iş gelişi ile birlikte çizelgeleme başlar. Çizelgeleme sonrasında oluşturulan iş emirleri atölyeye iletilir. Aşamalarda bulunan makineler, iş emirleri doğrultusunda o makinelere atanan işleri işlerler. İşi biten işlerin bilgisi çizelgeleme süreci ile paylaşılır. Sisteme yeni gelen işlerin bekleyen işler arasına eklenmesi, tamamlanan işlerin ise bekleyen işler listesinden silinmesi ile çizelgeleme sistemi güncellenir ve yeni çizelge üretilir.

Dinamik çizelgeleme literatürü incelendiğinde yeniden çizelgeleme zamanı için temelde periyodik ve olay tabanlı olmak üzere iki yaklaşım önerildiği görülür. Periyodik çizelgeleme yaklaşımı, belirlenen bir çizelgeleme frekansı doğrultusunda yeniden çizelgeleme yapmayı gerektirir. Olay tabanlı çizelgelemede ise üretim sisteminde çizelgeyi değiştiren durumlar oluştuğunda, yeniden çizelgeleme yapılır.

Bu tez çalışması kapsamında olay tabanlı yeniden çizelgeleme sistemi temel alınmıştır. Buna göre sisteme yeni bir iş gelişi veya sistemin herhangi bir aşamasındaki bir işin o aşamadan ayrılışı yeniden çizelgeleme noktası olarak kabul edilmiştir.

4. ÖNERİLEN METODOLOJİ

Bu kısımda ele alınan dinamik melez akış tipi çizelgeleme problemine yönelik olarak önerilen çizelgeleme algoritması (genetik algoritma) ve tasarlanan çizelgeleme sistemi açıklanacaktır. Dinamik çizelgeleme mekanizması açıklanarak, değişken ortamlarla nasıl başa çıkıldığına değinilecektir.

4.1 Çizelgeleme Algoritması

Ele alınan melez akış tipi çizelgeleme problemi, NP-zor bir problem olması nedeni ile algoritmik bir karmaşıklığa sahiptir. Çizelgeleme yapılan üretim sistemindeki işlerin işlem zamanlarının değişken olması ve sisteme işlerin rassal olarak gelmesi, probleme algoritmik karmaşıklığın yanında yapısal ve fonksiyonel bir zorlukta getirmektedir.

Literatürde dinamik ortamlarda yapılan çizelgeleme çalışmalarına bakıldığında, araştırmacıların problem karmaşıklığını metasezgisel yaklaşımlarla, fonksiyonel karmaşıklığı ise, simülasyonu metasezgisellere entegre ederek aşmaya çalıştıkları görülmüştür.

Bu çalışma kapsamında tasarlanması planlanan çizelgeleme sisteminin dinamik ortamlarda kaliteli çözümler elde etmesi amaçlanmıştır. Bu doğrultuda çözüm algoritmasının dinamik ve değişken ortamlara uyum sağlarken hızlı ve olurlu çözümler üretebilecek bir algoritma olması hedeflenmiştir. Algoritmanın, değişkenliğin olduğu ana kadar olan arama süreci kazanımlarını değişkenlik sonrasında oluşan yeni ortamın gereklerine göre adapte ederek yeni ortamda da efektif bir şekilde olurlu çizelgeler oluşturacak bir yapıya imkanı tanıması gerekmektedir. Literatür incelendiğinde dinamik ortamlarda, genetik algoritma yaklaşımının atölye tipi ve akış tipi çizelgeleme problemi üzerinde pek çok başarılı uygulaması görülmektedir. Bu çalışma kapsamında ele alınan dinamik melez akış tipi çizelgeleme problemine yönelik çizelgeleme algoritması olarak bir genetik algoritma geliştirilmesine karar verilmiştir.

4.2 Dinamik Melez Akış Tipi Çizelgeleme Problemine Yönelik Olarak Önerilen Genetik Algoritma (GA) Yaklaşımı

Genetik algoritmalar biyolojik sistemlerdeki adaptif süreçleri temel alırlar (Güçlü, 2006). Biyolojik sistemler kendilerini zaman içindeki değişimlere uyumlandırma eğilimindedirler. Biyolojik sistemler bu adaptasyonu, organizmaların genlerindeki değişim (varyasyon) ile başarırlar. Kalıtım ve değişim genlerde gerçekleşir ve organizmaların ya da popülasyonların zaman içindeki değişimlere adaptasyonunu sağlar. Kalıtım, anne ve babadan çocuk bireylere geçen özellikler olarak nitelendirilebilir.

Bununla birlikte genler zaman içinde değişim gösterme eğilimindedirler. Bu ise çaprazlama ve mutasyon gibi genlerde gerçekleşen değişikliklerle olur. Genlerde meydana gelen bu değişiklikler kimi zaman yararlı, kimi zaman etkisiz ya da zararlı olabilir. Yararlı değişiklikler, organizmaların değişimlere ayak uydurabilmesini sağlar. Değişime ayak uyduran bireyler hayatta kalırken, diğerleri en sonunda yok olurlar. Buna doğal seleksiyon denir.

Popülasyondaki uygun (fit) bireyler avantajlı bir genetik yapıya sahiptirler ve diğer bireylere göre popülasyonda daha uzun yaşayarak üreme gerçekleştirebilirler. Uygun bireylerin kromozomları üreme yolu ile bir sonraki nesle aktarılır. Kalıtım yolu ile bir önceki nesildeki bireylerin özellikleri sonraki nesillere taşınırken çaprazlama ve mutasyon yolu ile sonraki nesillerin genlerinin önceki nesillerden farklılaşması sağlanır.

Genetik algoritmalar Holland (1975) ve onun araştırma ekibince geliştirilmiştir. Genetik algoritmalar, biyolojik süreçleri örnek alır. Başlangıçta bulunan genetik özelliklerin nesilden nesle geçerken değişimini temel prensip olarak kabul eden genetik algoritmalar görece güçlü ve amaca daha uygun özelliklerin korunması, zayıf özelliklerin ise elimine edilmesini basit formda temel alırlar.

Genetik algoritmalar birden çok noktadan başlanarak arama yapan popülasyon tabanlı bir arama algoritmasıdır. En sonunda amaçlanan en iyi bireye ya da bir başka deyişle en iyi çözüme ulaşmaktır. Popülasyonu oluşturan aday çözümlerin soyut bir tanımlaması olan ve kromozom, genotip, gen gibi isimlerle ifade edilen değerler, çeşitli evrimsel tabanlı işlemler sonucu daha iyi sonuçları temsil eden çözüm adaylarına dönüşürler (evrimleşirler). Bu işlem kabul edilebilir bir uygunluk değerine

erişinciye kadar ya da daha önceden belirlenen bir nesil sayısı karşılanıncaya kadar sürdürülür (Kevran, 2009).

Bu çalışma kapsamında geliştirilmesi düşünülen genetik algoritmanın dinamik bir yapıya sahip olması ve dinamik ortamlarda karşılaşılan çözüm uzayı değişimlerinde, popülasyonu oluşturan bireylerin algoritma tarafından güncellenmesi hedeflenmiştir. Genetik algoritma dinamik olarak değişen bir yapıdaki üretim ortamında sürekli olarak daha iyi çözümleri daha iyi çizelgeleri arayacaktır. Bir bireye ait uygunluk (fitness) değeri alternatif çizelgelerin ortalama akış zamanı hesaplanarak bulunacaktır.

Standart bir genetik algortmada bulunması gereken adımlar:

- Çözüm Gösterimi
- Başlangıç Popülasyonunun Üretilmesi
- Bir Çözümün Uygunluk Değerinin Belirlenmesi
- Seçim
- Çaprazlama
- Mutasyon
- Bir Sonraki Popülasyonun Belirlenmesi

Dinamik melez akış tipi çizelgeleme problemi GA uygulamasında yukarıdaki adımların hepsinin olması düşünülmüştür. Literatürde birçok GA yaklaşımı ve uygulamasını görmek mümkündür. Bu çalışma kapsamında ele alınan probleme yönelik olarak geliştirilmesi düşünülen GA'nın adımları Şekil 4.1'de verilmiştir.

1. *Başlangıç popülasyonunu üret*
2. *Popülasyondaki bireylerin uygunluk değerlerini belirle*
3. *Eğer son iş üretim sistemden çıkmışsa, DUR*
4. *Üretim sisteminde dinamik bir durum meydana gelinceye kadar*
 - 4.1. *Ebeveyn seçimi*
 - 4.2. *Çaprazlama*
 - 4.3. *Mutasyon*
 - 4.4. *Bir sonraki popülasyonu belirle*
5. *Popülasyondaki bireyleri güncelle*
6. *Adım 2'e git*

Şekil 4.1 : Önerilen GA'nın sözde kodu.

Önerilen GA'nın çözüm gösterimi bir sonraki bölümde açıklanmıştır.

4.2.1 Çözüm gösterimi

Popülasyondaki her çözümün genetik gösterimi (semboller dizilimi) algoritma açısından kritik bir konudur. Dizilim, kromozom ve semboller ise gen olarak adlandırılırlar (Eroğlu, 2014). Uygulanması düşünülen herhangi bir probleme yönelik olarak bir çözümün gösterimi bir çeşit kodlama olarak düşünülebilir.

Birçok GA uygulamasında çözüm gösteriminin genellikle sayılardan ya da harflerden oluşan probleme özgü karakter dizileri olduğu görülür. Kodlamanın kromozom dizilimleri arasında etkin ve verimli olarak bilgi alışverişini sağlayacak şekilde tasarlanması gerekmektedir.

Literatürde akış tipi çizelgeleme problemine yönelik olarak geliştirilen GA'nın kromozom yapılarına bakıldığında pek çoğunda permutasyon tabanlı temsilin kullanıldığı görülmektedir (Wang ve Choi, 2014), (Jungwattanakit ve diğ., 2008) (Bierwirth ve Mattfeld, 1999). Buna göre sistemde o anda bulunan işlerin permutasyonu bir çözüm adayının kromozomunu temsil eder. Herhangi bir t anında sistemde 10 iş varsa bu işlerin herhangi bir permutasyonu alternatif bir çözüm adayını oluşturur. Bir çözüm adayının permutasyon temsili Şekil 4.2'de gösterilmiştir.

İş5	İş2	İş1	İş3	İş4	İş9	İş7	İş6	İş8	İş10
-----	-----	-----	-----	-----	-----	-----	-----	-----	------

Şekil 4.2 : Bir çözüm adayının permutasyon temsili.

Bir çözüm adayının, kromozom yapısının oluşturulmasında ise 0 ile 1 arasındaki reel sayılardan yararlanılmasına karar verilmiştir. Çözüm gösteriminde 0 ile 1 arasındaki sayıların kullanımı rassal anahtar temsil yöntemi diye adlandırılır. Her bir rassal sayı farklı bir işi simgeler. Örneğin 9 işlik bir çözüme ait kromozom Şekil 4.3'te gösterilmiştir.

0.35	0.62	0.18	0.42	0.83	0.76	0.39	0.51	0.36
------	------	------	------	------	------	------	------	------

Şekil 4.3 : 9 işlik bir kromozomun gösterimi.

Bu vektördeki reel sayıları küçükten büyüğe doğru sıraladığımızda 0.18 1. işi, 0.83 ise 9. işi temsil eder. Buna göre bu 9 işlik çözüm adayı Şekil 4.4'teki gibi oluşur.

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

Şekil 4.4 : 9 işlik bir çözüm adayı.

Rassal anahtar temsil yöntemi hem algoritmik açıdan hem de kodlama açısından pek çok avantaj getirmektedir. Rassal anahtar temsil yöntemi ile popülasyonu oluşturan çözüm adayları (permutasyonlar) çok hızlı bir şekilde oluşturulabilmektedir. Ayrıca GA'nın en önemli operatörlerinden olan çaprazlama operatörü gerçekleştirilirken ayrı bir modifikasyon ya da düzeltmeye gerek kalmamaktadır. Ele alınan problemin melez akış tipi çizelgeleme problemi olduğu dikkate alınarak, üretim sisteminin ilk aşamasında işlem görmeyi bekleyen işlerin permutasyonu bir çözüm adayı olarak kabul edilebilir. Bununla birlikte sadece ilk aşamadaki işlerin permutasyonu tam bir çizelge oluşturmaz. Dinamik melez akış tipi çizelgeleme probleminin yapısı gereği sisteme gelen bir iş çeşitli aşamalarda makinelerde işlem görerek sistemi terk etmektedir. Buna göre bir çizelge, ilk aşamada oluşturulan iş sırası ve seçilecek bir iş-makine atama politikasına göre işlerin makinelere atanması ile oluşturulabilir. İşler ilk aşamadaki iş sırası ve seçilen makine atama politikasına göre ilk aşamadaki makinelere atanırlar.

Birinci aşamada işlem gören işler o aşamadaki tamamlanma zamanlarına göre küçükten büyüğe doğru sıralanırlar. İşler ikinci aşamada bir önceki aşamada oluşturulan sıra ve aynı makine atama politikasına göre ikinci aşamadaki makinelere atanırlar. Bu şekilde sonraki aşamalar için de aynı yol izlenerek çizelge oluşturulur (Oğuz ve Ercan, 2005).

İşlerin makinelere atanmasında farklı politikalar izlenebilir. Ele alınan melez akış tipi çizelgeleme probleminde üretim aşamaları özdeş olmayan paralel makinelerden oluşmaktadır. İşlerin herhangi bir aşama için o aşamadaki en hızlı boş makineye atanarak işlenmesi melez akış tipi çizelgeleme problemi için bir iş-makine atama kuralı olarak düşünülebilir. En hızlı boş makineye atama kuralı temelde makinelerin sürekli çalışması prensibine dayanır ve makine boş kalmasına izin vermez. En hızlı boş makine atama kuralı algoritması Şekil 4.5'te verilmiştir.

Π_i : i aşamasındaki işlerin sırası (permutasyonu)

$\Pi_i(h)$: h . pozisyondaki iş

olmak üzere;

1. $\Pi_1 =$ Birinci aşamadaki iş sırası olarak GA'nın önerdiği permutasyonu al
2. $i=1$ 'den $i= m$ 'ye kadar
 - 2.1. $h=1$ 'den $h=n$ 'ye kadar
 - 2.1.1. İş $\Pi_i(h)$ 'yi, i aşamadaki makineler içinden (m_i) en hızlı boş makineye ata. Eğer i . aşamadaki bütün makineler meşgul ise, $\Pi_i(h)$ 'yi ilk boşalan makineye ata.
 - 2.2. Eğer $i < (Aşama\ sayısı)$ ise;
 - 2.2.1. i . aşamadaki işleri i aşamadaki tamamlanma zamanlarına göre küçükten büyüğe doğru sırala ve bu sırayı $(i+1)$. aşamadaki iş sırası (Π_{i+1}) olarak belirle.

Şekil 4.5 : En hızlı boş makineye atama kuralı sözde kodu.

Melez akış tipi çizelgeleme problemi için aşamalardaki işlerin makinelerde işlenmesine yönelik olarak diğer bir iş-makine atama kuralında ise makine boş kalmasına izin verilir. Bu politikada temel olarak herhangi bir iş o aşamadaki tamamlanma zamanı en küçük olacak şekilde bulunduğu aşamadaki makinelere atanır. En küçük tamamlanma zamanı makine atama kuralı algoritması sözde kodu Şekil 4.6'da verilmiştir.

1. $\Pi_1 =$ Birinci aşamadaki iş sırası olarak GA'nın önerdiği permutasyonu al
2. $i=1$ 'den $i=m$ 'ye kadar
 - 2.1. $h=1$ 'den $h=n$ 'ye kadar
 - 2.1.1. İş $\Pi_i(h)$ 'yi, o aşamadaki makineler içinden (m_i) i. aşamadaki tamamlanma zamanı en küçük olacak makineye ata
 - 2.2. Eğer $i < (\text{Aşama sayısı})$ ise;
 - 2.2.1. i. aşamadaki işleri o aşamadaki tamamlanma zamanlarına göre küçükten büyüğe doğru sırala ve bu sırayı $(i+1)$. aşamadaki iş sırası (Π_{i+1}) olarak belirle.

Şekil 4.6 : En küçük tamamlanma zamanı makine atama kuralı sözde kodu.

Bu tez çalışması kapsamında herhangi bir çözüm adayı (permutasyonu) için çizelge oluşturulurken, ilk aşama için GA'nın önerdiği en iyi iş sırası ve seçilen makine atama kuralı dikkate alınarak işlerin makinelere atanması gerçekleştirilir. Bir sonraki aşamadaki iş sırası, bir önceki aşamadaki işlerin tamamlanma zamanlarının küçükten büyüğe göre sıralanması ile oluşturulur ve aynı makine atama politikası ile işlerin atanması gerçekleştirilir ve diğer aşamalar için de aynı yöntem izlenir. Makine atama politikası olarak, en hızlı boş makine atama kuralı ve en küçük tamamlanma zamanı makine atama kuralları seçilen örnek problemler üzerinde test edilmiştir. Beşinci bölümde, GA stratejilerinin belirlenmesi kısmında hangi iş-makine atama politikasının nasıl seçildiği ve performansları detaylı olarak açıklanmıştır.

4.2.2 Başlangıç popülasyonunun üretilmesi

GA uygulamalarında ilk popülasyonun üretilmesi birçok açıdan kritik bir öneme sahiptir. Başlangıç popülasyonunda yeterince farklı çözümün olması, ele alınan problemin çözüm uzayının dengeli bir şekilde kapsanabilmesine imkan tanır. Burada dikkat edilmesi gereken husus, GA'nın erken yakınsamasının önüne geçilmesidir (Gen ve Cheng, 2000).

Başlangıç popülasyonunda yeterince farklı çözümün olmaması algoritmanın erken yakınsayarak yerel optimuma takılmasına sebep olabilir. Bierwirth ve Mattfeld (1999) ve Gholami ve Zandieh (2009) dinamik atölye tipi çizelgeleme problemi için başlangıç popülasyonunun tamamen rassal permutasyonlardan oluşturulmasını önermişler, böylece rassallığın sağlanarak yerel optimuma takılmanın önüne geçilebileceğini savunmuşlardır.

Reeves (2003), ele alınan probleme özgü iyi çözümlerle başlangıç popülasyonunu oluşturmanın daha hızlı bir şekilde iyi çözümler bulmayı kolaylaştıracağını vurgulamıştır. Bu çalışma kapsamında ele alınan dinamik melez akış tipi çizelgeleme probleminde ortalama akış zamanının minimize edilmesi amaçlanmaktadır. En Kısa İşlem Zamanı Önce liste çizelgeleme kuralının çoğu zaman en büyük işlem zamanı ve ortalama akış zamanı amaç fonksiyonları için atölye tipi ve melez akış tipi çizelgeleme problemlerinde, dinamik çizelgelemede kullanıldığı görülmektedir. Pinedo (2008) makinelerin özdeş olduğu ve her aşamada aynı sayıda makinenin olduğu sınırsız ara stok alanlı melez akış tipi çizelgeleme problemi için En Kısa İşlem Zamanı Önce kuralının işlerin toplam tamamlanma zamanını minimize ettiğini göstermiştir. Pinedo (2008) paralel makine üretim ortamı için En Uzun İşlem Zamanı Önce liste çizelgeleme kuralının iyi çözümler verdiğini göstermiştir.

Bu çalışma kapsamında başlangıç popülasyonunun oluşturulmasında liste çizelgeleme yöntemlerinden faydalanılmasına karar verilmiştir. Buna göre, başlangıç popülasyonunu oluşturan çözümlerin %80'ninin rastgele permutasyonlardan oluşturulmasına, %10'unun En Kısa İşlem Zamanı Önce liste çizelgeleme kuralı ve diğer kalan %10'unun da En Uzun İşlem Zamanı Önce liste çizelgeleme kuralı ile üretilmesine karar verilmiştir. Başlangıç popülasyonu üretim algoritması sözde programı Şekil 4.7'de özetlenmiştir.

1. *Başlangıç popülasyonunu üret*
 - 1.1. *Üretilen birey sayısı popülasyon büyüklüğünün %80'ine ulaşıncaya kadar*
 - 1.1.1. *Rastgele bir permutasyon (iş sırası) ile bir bireyi (çözüm adayını) üret*
 - 1.1.2. *Üretilen bireyi popülasyona ekle*
 - 1.2. *Üretilen birey sayısı popülasyon büyüklüğünün %90'nına ulaşıncaya kadar*
 - 1.2.1. *En Kısa İşlem Zamanı Önce kuralı ile bir bireye ait permutasyonu üret*
 - 1.2.2. *Üretilen bireyi popülasyona ekle*
 - 1.3. *Üretilen birey sayısı popülasyon büyüklüğünün %100'üne ulaşıncaya kadar*
 - 1.3.1. *En Uzun İşlem Zamanı Önce kuralı ile bir bireye ait permutasyonu üret*
 - 1.3.2. *Üretilen bireyi popülasyona ekle*

Şekil 4.7 : Başlangıç popülasyonu üretim algoritması sözde kodu.

Bir çözüm adayına ait kromozomun üretilmesinde rassal sayıların özelliklerinden yararlanılır. Bir bireye ait rastgele bir permutasyon oluşturulurken iş sayısı kadar rassal sayı üretilerek bir vektör oluşturulur, ve bu vektörün üyelerinin küçükten büyüğe sıralanması ile rastgele permutasyona sahip bir birey üretilebilir. Rastgele birey üretilmesi algoritması detayları Şekil 4.8'de verilmiştir.

1. Rastgele bir birey üret

1.1. Üretim sisteminin ilk aşamasında işlem görmeyi bekleyen iş sayısı uzunluğunda bir rassal sayı vektörü tanımla

1.2. Üretim sisteminin ilk aşamasında işlem görmeyi bekleyen iş sayısı uzunluğunda bir tam sayı vektörü tanımla

1.3. Üretim sisteminin ilk aşamasında işlem görmeyi bekleyen iş sayısı kadar

1.3.1. 0 ile 1 arasında bir rassal sayı üret

1.3.2. Üretilen rassal sayıları rassal sayı vektörüne ekle

1.4. Rassal sayıların rassal sayı vektöründeki pozisyonlarını değiştirmeden, rassal sayıları kendi içinde küçükten büyüğe doğru sırala, ve bu sıra doğrultusunda her rassal sayıya farklı bir indis ata

1.5. Rassal sayı indislerini rassal sayıların rassal sayı vektöründeki pozisyonlarına ekleyerek tam sayı vektörünü oluştur.

Şekil 4.8 : Rastgele birey üretim algoritması sözde kodu.

En Kısa İşlem Zamanı Önce ve En Uzun İşlem Zamanı Önce kuralları ile bir bireye ait iş sırası oluşturulurken, ortalama işlem zamanlarından yararlanılmasının uygun olacağı düşünülmüştür. Ele alınan melez akış tipi çizelgeleme problemi birden çok aşamada işlem yapılan özdeş olmayan paralel makinelerden oluşmaktadır. Buna göre bir iş birden çok aşamada farklı makinelerde farklı işlem zamanları ile işlem görebilmektedir. En Kısa İşlem Zamanı Önce ve En Uzun İşlem Zamanı Önce kuralları ile birey üretim algoritmaları Şekil 4.9 ve Şekil 4.10'da açıklanmıştır.

1. *En Kısa İşlem Zamanı Önce Kuralı ile bir birey üret*
 - 1.1. *Üretim sisteminin ilk aşamasında işlem görmeyi bekleyen iş sayısı uzunluğunda bir tam sayı vektörü tanımla*
 - 1.2. *Üretim sisteminin ilk aşamasında işlem görmeyi bekleyen işlerden bir iş vektörü oluştur*
 - 1.3. *İş vektöründeki işlerin pozisyonlarını değiştirmeden, işleri kendi içinde ortalama işlem zamanına göre küçükten büyüğe doğru sırala, ve bu sıra doğrultusunda her işe farklı bir indis ata*
 - 1.4. *İş indislerini, işlerin iş vektöründeki pozisyonlarına ekleyerek tam sayı vektörünü oluştur.*

Şekil 4.9 : En Kısa İşlem Zamanı Önce Kuralı ile birey üretim algoritması sözde kodu.

1. *En Uzun İşlem Zamanı Önce Kuralı ile bir birey üret*
 - 1.1. *Üretim sisteminin ilk aşamasında işlem görmeyi bekleyen iş sayısı uzunluğunda bir tam sayı vektörü tanımla*
 - 1.2. *Üretim sisteminin ilk aşamasında işlem görmeyi bekleyen işlerden bir iş vektörü oluştur*
 - 1.3. *İş vektöründeki işlerin pozisyonlarını değiştirmeden, işleri kendi içinde ortalama işlem zamanına göre büyükten küçüğe doğru sırala, ve bu sıra doğrultusunda her işe farklı bir indis ata*
 - 1.4. *İş indislerini, işlerin iş vektöründeki pozisyonlarına ekleyerek tam sayı vektörünü oluştur.*

Şekil 4.10 : En Uzun İşlem Zamanı Önce Kuralı ile birey üretim algoritması sözde kodu.

4.2.3 Bir çözümün uygunluk değerinin belirlenmesi

Bir bireyin uygunluk değeri o bireyin genotipinin ileride oluşturacağı etki olarak düşünülebilir. Uygun bireylerin hayatta kalacağı yaklaşımı, popülasyondaki bireylerin uygunluk değerlerine göre sınıflandırılmasını temel alır. Popülasyondaki her bireyin bir uygunluk değeri ve uygunluk seviyesi vardır. Uygunluk değerlerinin düzeyi hem bireyin hem de o bireyden üretilen çocukların popülasyondaki geleceğini belirler.

Bir GA'da, her bir çözüm gösterimi bir uygunluk değerine karşılık gelir. Uygunluk değerinin hesaplanması prosedürü bir miktar çaba isteyebilir. Bu prosedür belki basit birkaç operasyon olabileceği gibi, karmaşık bir algoritmanın koşturulması da olabilir.

Bir çözümün uygunluk değerini hesaplayabilmek için, bir takım performans göstergeleri tanımlanmalıdır. Bu tez çalışması kapsamında ele alınan dinamik melez akış tipi çizelgeleme problemi için anahtar performans ölçütü ortalama akış zamanı, \bar{F} olarak belirlenmiştir.

Bu doğrultuda bir çözüme ait uygunluk değeri, o çözüme ait çizelgenin ortalama akış zamanı hesaplanarak bulunur. Popülasyondaki bir bireyin çözüm gösterimi üretim sisteminin ilk aşamasındaki işlerin sırası olarak tanımlanmıştır. Bir bireyin uygunluk değeri, birinci aşamadaki iş sırası ve seçilen iş-makine atama politikası dikkate alınarak işlerin üretim aşamalarındaki makinelere atanması ile oluşturulan çizelgenin ortalama akış zamanı hesaplanarak bulunur.

Örnek bir problemle, bir çözüme ait uygunluk değerinin nasıl hesaplandığı pekiştirilebilir. Başlangıçta sistemde dokuz işin işlenmek üzere beklediği, iki aşamalı ve her aşamada beş özdeş olmayan paralel makinenin bulunduğu bir üretim sistemine ait işlem zamanları (p_{ijk}) Çizelge 4.1'de verilmiştir.

- p_{ijk} : i işinin j aşamasında k makinesindeki işlem süresi

Çizelge 4.1 : Örnek bir probleme ait işlerin sistemdeki işlem zamanları.

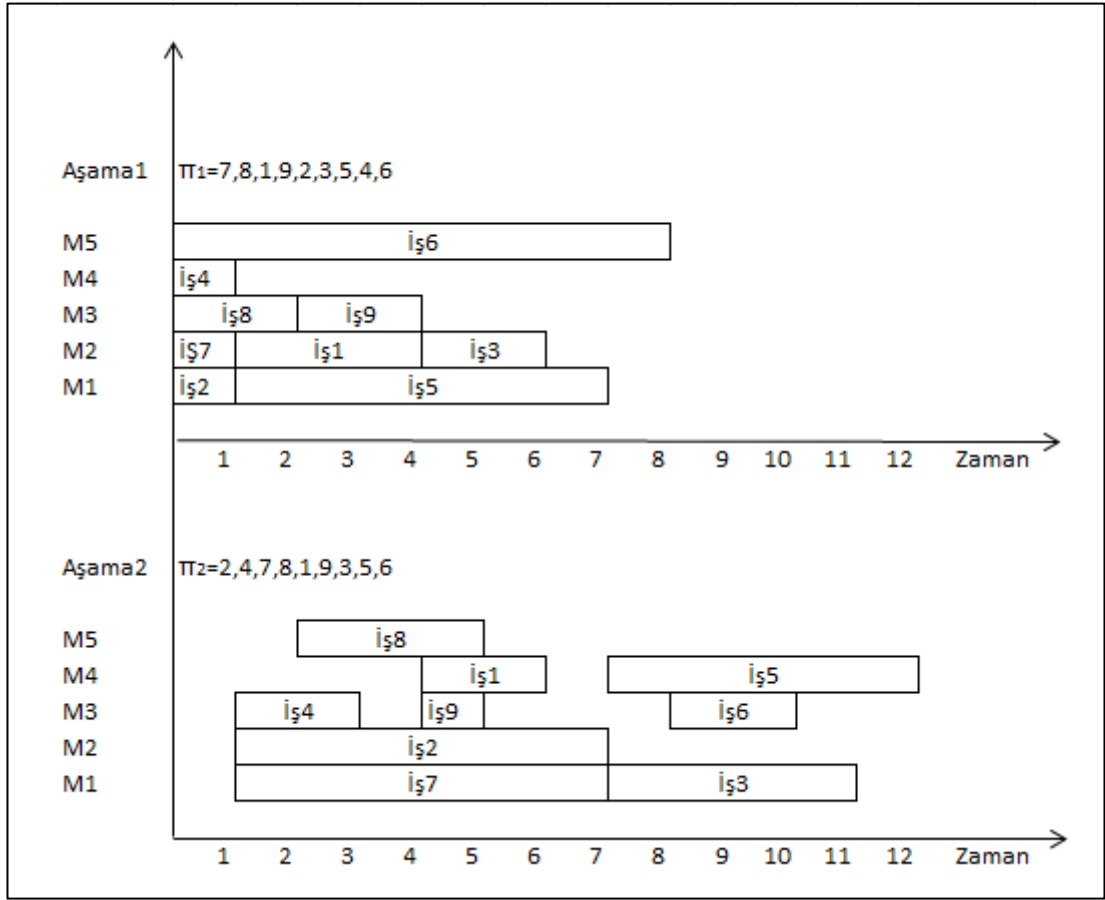
	1. Aşama						2. Aşama				
	Mak1	Mak2	Mak3	Mak4	Mak5		Mak1	Mak2	Mak3	Mak4	Mak5
İş1	5	3	9	15	13	9	7	18	2	13	
İş2	1	5	7	19	13	13	6	12	14	14	
İş3	19	2	4	8	16	4	16	6	16	17	
İş4	11	14	14	1	19	16	17	2	10	10	
İş5	6	19	14	8	9	13	13	9	5	7	
İş6	10	4	7	14	8	3	19	2	19	10	
İş7	13	1	3	5	2	6	11	6	19	17	
İş8	16	5	2	4	9	19	8	15	19	3	
İş9	10	7	2	9	7	8	8	1	12	16	

Üretim sisteminde başlangıçta dokuz işin olduğu bu probleme yönelik olarak GA'nın ürettiği en iyi çözüme ait kromozom Şekil 4.11'deki gibidir.

0.15	0.44	0.66	0.97	0.72	0.98	0.04	0.05	0.37
------	------	------	------	------	------	------	------	------

Şekil 4.11 : Sistemde 9 işin bulunduğu probleme ait kromozom.

Çizelge 4.1'deki işlem zamanları ve Şekil 4.11'de verilen çözüme ait kromozom doğrultusunda oluşturulan çizelge, Şekil 4.12'te verilmiştir.



Şekil 4.12 : Bir çözüm adayına ait çizelge örneği.

Başlangıçta üretim sisteminin 1. Aşamasında sistemde 9 iş vardır. Bu işlere ait olarak genetik algoritmanın önerdiği en iyi çözüme ait kromozom Şekil 4.11'de verilmiştir. Bu kromozomdaki rassal sayılar küçükten büyüğe doğru sıralandığında işlerin sistemde hangi sırada işlem göreceğinin sırasını verecek olan permutasyon ya da çözüm üretilir. Buna göre kromozomdaki en küçük rassal sayı 0.04'tür ve İş7'yi göstermektedir. 0.04'ten sonra gelen en küçük rassal sayı 0.05'tir ve sistemdeki 8. İş'i temsil eder. Bu doğrultuda 1.Aşama'da sistemdeki işlerin hangi sırada işlem göreceğini belirleyen iş permutasyonu $\pi_1 = 7,8,1,9,2,3,5,4,6$ olarak belirlenir. İşlerin üretim sistemindeki işlem sırası belirlendikten sonra ikinci olarak iş-makine atama politikasına karar verilmesi gerekmektedir. En küçük tamamlanma zamanı kuralı iş-makine atama politikası olarak seçilmiştir. Bu doğrultuda 1.Aşama'ya ait çizelge Şekil 4.12'deki gibi oluşmuştur. Buna göre sistemde ilk sırada işlem görmeyi bekleyen 7. İş, Çizelge 4.1'deki işlem zamanlarına göre sistemdeki en küçük

tamamlanma zamanı 1'i verecek olan Makine2'de işlem görecektir. Daha sonra gelen 8. İş, en küçük tamamlanma zamanı 2'yi verecek olan Makine3'te işlem görecektir. İş8'den sonra işlem sırasında olan 1. İş, sistemde en küçük tamamlanma zamanını verecek olan Makine2'de işlem görecektir. Burada dikkat edilmesi gereken İş1 makinelerine atanmak istendiğinde o anda sistemde boşta bulunan Makine1, Makine4 ya da Makine5' te işlem görebilecek iken bu makinelerdeki işlem zamanları 5, 15 ve 13 dikkate alınarak, Makine2'deki işin bitmesi beklenmiştir. Buna göre t=0 anında Makine2'ye atanan İş7, t=1 anında bitecektir ve İş1'in bu makinedeki işlem süresi 3'tür. Buna göre İş1, 1. Aşamadaki en küçük tamamlanma zamanı 1+3 = 4'ü verecek olan Makine2'de işlem görecektir. 1. Aşamada iş sırasında bulunan diğer işler de aynı şekilde en küçük tamamlanma zamanını verecek şekilde makinelerine atanırlar.

1.Aşamadaki işler, 1.Aşama'dan çıkış zamanlarına göre küçükten büyüğe doğru sıralanarak, işlerin 2.Aşama'daki işlem sırası, $\pi_2 = 2,4,7,8,1,9,3,5,6$ şeklinde oluşturulur. İşler, 1.Aşama'da olduğu gibi 2.Aşama'da da makinelerine en küçük tamamlanma zamanı politikasına göre atanırlar.

Sistemdeki bütün işler tamamlandıktan sonra bu çözüm adayına ait uygunluk değerini belirleyecek olan ortalama akış zamanı, \bar{F} hesaplanır. Herhangi bir işe ait akış zamanı o işin tamamlanma zamanından en erken hazır olma zamanı çıkarılarak, $(F_j = C_j - r_j)$ hesaplanır. Örneğin İş1'in tamamlanma zamanı 6, İş1'in sistemde işleme en erken hazır olma zamanı 0'dır. Buna göre İş1'in akış zamanı $6-0 = 6$ olarak hesaplanır. Aynı şekilde diğer işlerin de akış zamanları hesaplanır ve toplam akış zamanı $(\sum F)$, 66 olarak bulunur. Ortalama akış zamanı ise toplam akış zamanının (66) iş sayısına (9) bölünmesi ile $\bar{F} = \frac{66}{9} = 7.33$ olarak bulunur. Sonuç olarak bu çözüm adayına ait uygunluk değeri 7.33 olarak belirlenir.

4.2.3.1 İşlem zamanlarının olasılıksal (stokastik) olduğu durumda uygunluk değerinin belirlenmesi

Ele alınan dinamik melez akış tipi çizelgeleme probleminde işlem zamanları deterministik değildirler. Sistemdeki işlerin işlem zamanları olasılıksaldır. Buna göre işlem zamanları 1 ile 19 arasındaki düzgün dağılıma uymaktadır. Sistemdeki stokastikliğin GA'ya yansıtılması gerekmektedir. Bunun ise, uygunluk değerlerinin simülasyon ile hesaplanmasıyla yapılabileceği düşünülmüştür. Bu bağlamda, GA tarafından önerilen bir çözüm adayına ait işlem zamanları sabit değildir. İşlem

zamanları ortalaması 10 olan 1 ile 19 arasındaki düzgün rassal değişkenlerdir. Her bir çözüme ait beklenen ortalama akış zamanı simülasyon yolu ile hesaplanarak her bir çözümün beklenen uygunluk değeri ya da bir başka deyişle beklenen ortalama akış zamanı, $E(\bar{F})$ bulunur.

Çözümlerin beklenen uygunluk değerleri hesaplanırken her bir çözüm için simülasyon gerçekleştirilir. Öncelikle bir çözüm adayını oluşturan işlerin üretim sisteminde dolaşması (işlem görmesi) sonucu ortaya çıkan akış zamanı değeri bulunur ve o çözüm adayı için kaydedilir. Daha sonra aynı çözüm adayı n-1 defa daha üretim sisteminde dolaştırılır ve n farklı akış zamanı değeri bulunur. Bulunan değerlerinin ortalaması alınarak o çözüm adayının beklenen uygunluk değeri hesaplanır.

Koşum sayısı (n değeri) algoritmanın performansı açısından kritik bir değerdir. Belirlenen bir anlamlılık düzeyi (α) ve hata değeri (ϵ) için koşum sayısı, $n \geq \left(\frac{z_{\alpha/2} \cdot \sigma}{\epsilon}\right)^2$ formülü ile bulunabilir (Banks ve diğ., 2010).

Evrimsel algoritmalarla değişkenliğin olduğu ortamlarda yapılan optimizasyon çalışmalarına bakıldığında koşum sayısı ve popülasyon büyüklüğünün beraber analiz edildiği görülmektedir (Arnold ve Beyer, 2000, 2001).

Branke (2002) farklı örneklem sayıları ve popülasyon büyüklükleri için 5 boyutlu bir deney düzeneği hazırlamıştır. Örneklem sayıları 1, 2, 4, 8, ve 16 olarak değişkenlik gösterirken, popülasyon büyüklükleri 50, 100, 200, 400 ve 800 arasında değişmektedir. Her bir alternatifin, bir jenerasyondaki toplam uygunluk hesaplama sayısı birbirine eşittir ($800 \cdot 1 = 400 \cdot 2 = 200 \cdot 4 = 100 \cdot 8 = 50 \cdot 16$). Seçtikleri iki farklı probleme göre (f1, f2) popülasyon büyüklüğü ve örneklem sayısının algoritma başarımına etkisini incelemişlerdir. Büyük popülasyonlarla çalışan evrimsel algoritmaların arama sürecinin başında daha fazla olasılığı bir arada değerlendirme avantajı ile daha hızlı bir şekilde global optimuma yaklaşabileceğini vurgulamışlardır. Bununla birlikte örneklem sayısının artırılmasının ise popülasyon yakınsadıktan sonra yerel optimuma takılmanın önlenmesi açısından yararlı olabileceğine dikkat çekmişlerdir. Beşinci bölümde, GA parametrelerinin optimizasyonu kısmında koşum sayısının nasıl belirlendiği seçilen test problemleri üzerinden detaylı bir şekilde açıklanmıştır.

4.2.4 Seçim

Çözüm havuzu (popülasyon) içinden hangi çözümlerin bir sonraki nesle aktarılacağı, seçim operatörü yardımı ile gerçekleştirilir. Evrimsel algoritmalar, uygun bireylerin hayatta kalma ve çoğalma şansının daha yüksek olduğu prensibine dayanır. Bu sebeple, dinamik melez akış tipi çizelgeleme problemine yönelik olarak geliştirilmesi düşünülen GA'nın seçim prosedürünün, popülasyondaki çözümlerin uygunluk değerlerine göre çalışması amaçlanmıştır.

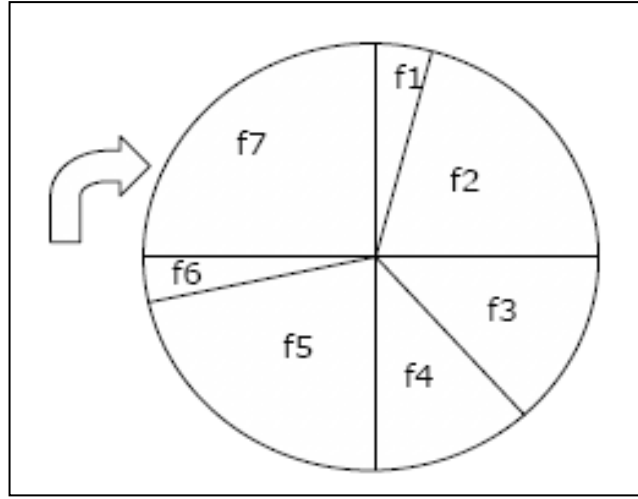
Seçilen bir çözüm, ebeveyn olarak adlandırılır ve bu çözüm avantajlı bir genetik yapıya sahip olduğu için muhtemelen sonraki nesle aktarılacaktır. Ebeveynlerin kromozomları genetik operatörlerin olasılıksal yapısı gereği kısmen ya da tamamen bir sonraki nesle geçer. GA'larda ebeveyn seçimi için uygulanabilecek farklı yöntemler mevcuttur.

Literatürde pek çok GA uygulamasında Rulet Tekerlek Seçim metodunun ebeveyn seçiminde kullanıldığı görülmektedir. Rulet Tekerlek Seçim metodunda bir çözümün ebeveyn olarak seçilmesi popülasyondaki çözümlerin uygunluk değerlerine bağlı olarak üretilen olasılıklara göre değişir.

Branke (2002), dinamik ortamlarda seçim metodu olarak sıra bazlı seçim metodlarının, uygunluk tabanlı seçim metodlarına göre daha iyi sonuçlar verdiği dikkat çekmiştir. Turnuva Seçim metodu sıra bazlı seçim metodları içinde gösterilmektedir.

4.2.4.1 Rulet tekerlek seçim metodu

Rulet Tekerlek Seçim prosedürü, popülasyondaki bireylerin uygunluk değerlerine göre dilimlenmiş bir rulet tekerleğinin döndürülmesine benzetilmektedir. Bir referans noktasına göre fırıladağın 0 ile 2π arasında rastgele bir açı kadar hareket ettirilmesi ile bir birey seçilebilir.



Şekil 4.13 : Rulet tekerlek seçim.

Şekil 4.13'te 7 bireylik bir popülasyona ait çözümlerin uygunluk değerlerine göre nasıl rulet tekerleği üzerinde konumlandırıldığı görülebilir. Uygunluk değerine göre en fazla alana sahip çözüm 7.çözümdür, bu yüzden 7.bireyin ebeveyn olarak seçilme olasılığı daha yüksektir.

Popülasyondaki bir i bireyinin seçim olasılığı, bireylerin uygunluk değerlerinin popülasyondaki tüm bireylerin uygunluk değerlerinin toplamına oranlanarak aşağıdaki formülle belirlenebilir:

$$olasılık_i = \frac{uygunluk_i}{\sum_j uygunluk_j}, \quad j = 1, \dots, pop.büyüküğü \quad (4.1)$$

Her zaman ham uygunluk değerlerini direkt olarak seçim olasılıklarını belirlemede kullanmak uygun olmayabilir. Değerler birbirine çok yakın olabilir, ya da popülasyondaki en küçük değer ile en büyük değer arasında çok az bir fark olabilir. Uygunluk değerleri uygunluk olasılıklarını belirlemede yeterince etkili olmayabilir. Bu gibi durumlarda uygunluk değerleri üzerinde doğrusal bir transformasyon yapılabilir. Burada amaçlanan uygunluk değerlerine göre bireylerin seçilme olasılıklarını bir normalizasyon ile düzeltmektir.

Popülasyon içindeki iyi uygunluk değerine (minimizasyon problemi için düşük değerlere) sahip bireylere daha yüksek şans tanımak için, popülasyon içindeki en kötü uygunluk değerine (minimizasyon problemi için en büyük değer) sahip birey belirlenir ve bireylerin en kötü bireye olan uzaklığı dikkate alınarak seçim olasılıkları

belirlenir. Şekil 4.14'te seçim olasılıkları doğrusal olarak ölçeklendirilmiş Rulet Tekerlek Seçim metodu algoritması sözde programı verilmiştir.

```
EBU      : Popülasyondaki en büyük uygunluk değeri
Pop.Büy  : Popülasyon büyüklüğü
Top      : Popülasyondaki bütün bireylerin uygunluk değerlerinin
toplamaı
Ui     : i.bireyin uygunluk değeri
R       : 0 ve ( Pop.Büy*EBU – Top ) değerleri arasında üretilen bir rassal
sayı

1. Popülasyondan iki ebeveyn seç
    1.1. mevcut_toplam=0
    1.2. i=1
        1.2.1. while ( doğru )
            1.2.1.1. mevcut_toplam=mevcut_toplam + ( EBU - Ui )
            1.2.1.2. eğer(mevcut_toplam>R) i.bireyi ebeveyn olarak
                belirle
            1.2.1.3. değilse, i=i+1
```

Şekil 4.14 : Rulet tekerlek seçim metodu sözde kodu.

4.2.4.2 Turnuva seçim metodu

Turnuva Seçim metodunda popülasyon içinden seçilen iki ya da daha fazla birey uygunluk değerlerine göre kıyaslanır ve en iyi birey ebeveyn olarak seçilir. Turnuva seçim metodu sıra tabanlı bir seçim metodudur. Popülasyon içindeki bireylerden birbirlerine kıyasla daha iyi sıraya sahip olanlar ebeveyn olarak seçilir. Bir defasında kaç bireyin kıyaslanacağını belirleyen turnuva büyüklüğü değeri algoritma açısından kritik bir öneme sahiptir. Turnuva Seçim metodu algoritması sözde kodu Şekil 4.15'te verilmiştir.

k =Turnuva büyüklüğü

1. *Popülasyondan iki ebeveyn seç*

1.1. *Rastgele k birey seç*

1.2. *k birey içinden uygunluk değerine göre en iyisini seç*

1.3. *Seçilen bireyi i . ebeveyn olarak belirle*

Şekil 4.15 : Turnuva seçim metodu sözde kodu.

Ebeveyn seçim politikası olarak, rulet tekerlek seçim metodu ve turnuva seçim metodu seçilen örnek problemler üzerinde test edilmiştir. Beşinci bölümde, GA stratejilerinin belirlenmesi kısmında hangi ebeveyn seçim politikasının algoritmada kullanılmak üzere nasıl seçildiği ve performansları detaylı olarak açıklanmıştır.

4.2.5 Çaprazlama

GA uygulamalarında anne ve baba bireylerin birbirleri ile eşleştirilerek yeni çözüm adaylarını, çocuk bireyleri meydana getirmesi çaprazlama operatörü ile gerçekleştirilir. Çaprazlama operatörü ile iyi genetik özelliklerin çocuk bireylere aktarılarak bir sonraki nesilde daha iyi çözümlerin üretilmesi amaçlanır. Çaprazlama en basit anlamda iki ebeveyn birey arasında gen değişimi gibi düşünülebilir. Belirlenen bir “çaprazlama oranına” göre ebeveyn bireyler eşleştirilirler. 0 ile 1 arasında üretilen bir rassal sayı çaprazlama oranından küçük ise çaprazlama koşulu sağlanmıştır. Çaprazlama oranı algoritmaya rassallık katar ve probleme özgü deneylerle belirlenir. Çaprazlama oranına bağlı olarak iki ebeveyn üzerine çaprazlama uygulanırsa ebeveynlerin genlerinin değişimi sonucu çocuk bireyler meydana gelir. Eğer çaprazlama koşulu sağlanmaz ise, çocuk birey anne ya da babanın bir kopyası olarak üretilir.

Literatürde melez akış tipi çizelgeleme problemi için geliştirilen GA uygulamalarında birçok farklı çaprazlama operatörü ile karşılaşılmaktadır. Ele alınan dinamik melez akış tipi çizelgeleme problemine yönelik geliştirilen çözüm adaylarının kromozomları rassal sayılardan oluşmaktadır. Rassal anahtar temsil yönteminde bir kromozomdaki herhangi iki gen değerinin birbirine eşit olması çok düşük bir olasılıktır. Rassal anahtar temsil yönteminin bu özelliğinden yararlanılarak

bir çaprazlama operatörü geliştirilebilir. Buna göre bir çocuk birey anne ve baba bireylerin genlerinin değerinin ortalaması alınarak bulunabilir.

Örneğin sistemde 9 işin bulunduğu bir durumda anne, baba çözümler ve çaprazlama sonucu oluşan çocuk birey Şekil 4.16'daki gibi oluşur.

Baba	0.12	0.27	0.33	0.4	0.59	0.64	0.71	0.88	0.95
Anne	0.34	0.22	0.36	0.28	0.81	0.02	0.16	0.89	0.11
Çocuk	0.23	0.245	0.345	0.34	0.7	0.33	0.435	0.885	0.53

Şekil 4.16 : Örnek bir çaprazlama işlemi.

olarak bulunur. Anne ve babalardan oluşan çocuk bireyler de olurlu çözümler vereceğinden çaprazlama sonrası çocuk bireyler üzerinde ek bir düzeltme yapmaya gerek kalmayacaktır. Önerilen çaprazlama operatörü anne ve babanın özelliklerini eşit oranlarda çocuk bireye aktarması ve ek bir kontrol gerektirmemesi ile efektif bir çaprazlama operatörüdür. Şekil 4.17'de ele alınan dinamik melez akış tipi çözümlenme problemine yönelik olarak önerilen çaprazlama operatörü algoritması sözde kodu verilmiştir.

1. Rastgele seçilen iki ebeveyninden bir çocuk çözüm oluşturacak şekilde çaprazlamayı gerçekleştir

1.1. Çaprazlama koşulu sağlandı mı

1.1.1. Seçilen iki ebeveynin genlerinin ortalamasını hesapla ve bu ortalama değerleri çocuk bireyin genleri olarak belirle

Şekil 4.17 : Çaprazlama operatörü sözde kodu.

4.2.6 Mutasyon

Mutasyon operatörü ile genetik çeşitliliğin sağlanması amaçlanır. Mutasyon oranına göre çocuk bireylerin kromozomları üzerinde mutasyon işlemi gerçekleştirilir. Mutasyon oranının düzeyi ele alınan probleme özgü olarak deneylerle belirlenir. Mutasyon, kromozomların genlerindeki bir takım bozma ve karışımlar yolu ile

sağlanır. Mutasyon ile genetik çeşitlilik sağlanarak yerel optimuma takılmanın önüne geçilebilir.

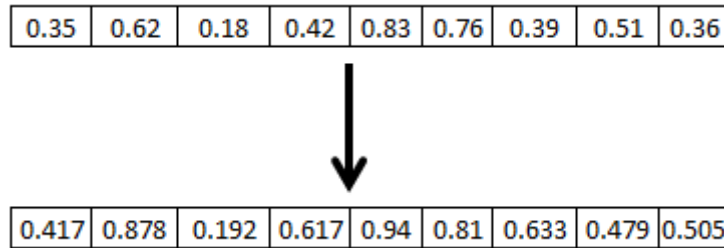
Ele alınan melez akış tipi probleminin yapısını ve önerilen rassal anahtar temsil yöntemini dikkate alarak bir mutasyon operatörünün geliştirilmesi gerekmektedir. Önerilen GA uygulamasında, kromozomların genlerini oluşturan rassal sayılar değiştirilerek mutasyon sağlanabilir. Geliştirilmesi düşünülen mutasyon operatöründe çocuk bireylerin tüm gen değerlerine ortalaması ve standart sapması belli bir dağılıma uyan bir rassal değer eklenir. Bu rassal değer mutasyon oranı (p_m) olarak tanımlanır. Buna göre;

$$\text{Çocuk}=[0.35, 0.62, 0.18, 0.42, 0.83, 0.76, 0.39, 0.51, 0.36]$$

çocuk çözümüne ortalaması 0 standart sapması 0.2 olan normal dağılıma uyan değerlerin eklenmesi ile mutasyona uğratılan birey;

$$\text{Çocuk}=[0.417, 0.878, 0.192, 0.617, 0.939, 0.813, 0.633, 0.479, 0.505]$$

şeklinde oluşur.



Şekil 4.18 : Örnek bir mutasyon işlemi.

Şekil 4.19’da melez akış tipi çizelgeleme problemine yönelik olarak önerilen mutasyon operatörünün sözde programı verilmiştir.

1. Her çocuk bireye mutasyon uygula

1.1. Çocuk bireyin kromozomundaki her gene belirlenen mutasyon oranına göre bir rassal sayı ekle

Şekil 4.19 : Mutasyon operatörü sözde kodu.

4.2.7 Bir sonraki popülasyonun belirlenmesi

GA operatörleri; seçim, çaprazlama ve mutasyonun uygulanması ile elde edilen çocuk bireylerden hangilerinin bir sonraki nesle aktarılacağı belirlenmesi gerekmektedir. GA uygulamalarında bir sonraki popülasyon belirlenirken çeşitli yaklaşımların temel alındığı görülmektedir. Bir sonraki popülasyon nesil tabanlı ya da uygunluk tabanlı yaklaşımlarla belirlenebilir.

Nesil tabanlı yaklaşım bir sonraki neslin tamamının çocuk bireylerden oluşması prensibine dayanır. Nesil tabanlı yaklaşımda çocuk bireylerin uygunluk değerleri ne olursa olsun sonraki popülasyon çocuk bireylerden oluşturulur. Uygunluk tabanlı yaklaşımda ise çocuk bireyler bir sonraki nesle aktarılırken ebeveynlerin seçildiği çözüm havuzu içindeki çözümlerle çocuk çözümler uygunluk değerlerine göre kıyaslanırlar. Bir sonraki popülasyon, uygunluk değerlerine göre hem çocuk bireylerden hem de ebeveynlerden en iyi bireyler seçilerek oluşturulur.

Bir sonraki nesil belirlenirken çocuk bireylerin bir kerede mi yoksa teker teker mi sonraki nesle aktarılacağı da belirlenmesi gerekmektedir. GA operatörlerinin ebeveylere uygulanması sonucu oluşturulan çocuk bireyler bir defasında nesil tabanlı ya da uygunluk tabanlı stratejiye göre sonraki nesle aktarılabilir. Başlangıçta m bireyin olduğu bir popülasyondan yine m tane çocuk birey oluşturulur. Oluşturulan m çocuk bireyin tamamı bir sonraki nesle bir kerede aktarılır.

Bir diğer stratejide ise GA operatörleri, çaprazlama ve mutasyon iteratif bir şekilde uygulanarak bir sonraki nesil oluşturulur. Seçilen iki ebeveyne önce çaprazlama operatörü sonra mutasyon uygulanarak bir çocuk birey oluşturulur. Bu çocuk bireyin, başlangıçta varolan m çözüme eklenmesi ile $(m+1)$ tane çözüm oluşturulur. Oluşturulan $(m+1)$ tane çözümden m tanesi nesil tabanlı stratejiye göre ya da uygunluk tabanlı stratejiye göre bir sonraki nesle aktarılır.

Branke (2002) dinamik ortamlardaki GA uygulamalarında elitist stratejiden yararlanılabileceğini belirtmiştir. Elitist stratejide ebeveynleri oluşturan popülasyondaki en iyi bireyin kaybedilmemesi amaçlanır. Popülasyonun en iyi bireyi çocuk bireylerle kıyaslanır ve bu birey eğer en kötü uygunluğa sahip çocuk bireyden daha iyi ise o çocuk bireyle değiştirilerek bir sonraki nesil belirlenir.

Bu tez çalışması kapsamında elitist stratejiyi ve uygunluk tabanlı stratejiyi temel alan iki farklı sonraki nesil belirleme yönteminin uygulanmasına karar verilmiştir. Elitist

stratejide ebeveynleri oluşturan m bireyin en iyisi, GA operatörleri ile üretilen m çocuk bireyin en kötüsü ile kıyaslanır eğer daha iyi ise çocuk bireyle yer değiştirilir değilse değiştirilmez. Kıyaslama sonucunda son hali verilen m çocuk birey bir kerede sonraki nesle aktarılarak sonraki nesil belirlenir.

Uygunluk tabanlı stratejide ise GA operatörleri çaprazlama ve mutasyon popülasyonun bütün üyelerine (m tane bireye) bir kerede uygulanıp m çocuk birey üretilmek yerine, her defasında bir çocuk birey üretilir ve bu çocuk bireyle beraber toplamda $(m+1)$ tane bireye ulaşılır. $(m+1)$ tane bireyden oluşan genişletilmiş birey havuzunun m tanesi uygunluk tabanlı strateji doğrultusunda bir sonraki nesli oluşturmak üzere seçilir.

Bu tez çalışması kapsamında önerilen elitist sonraki nesil belirleme stratejisi ve uygunluk tabanlı sonraki nesil belirleme stratejisine ait sözde kodlar Şekil 4.20 ve Şekil 4.21’de verilmiştir.

1. *Sonraki popülasyonu belirle*
 - 1.1. *m popülasyon üyesinden GA operatörleri (çaprazlama ve mutasyon) ile m çocuk birey üret*
 - 1.2. *Popülasyonun en iyi uygunluk değerine sahip üyesini belirle*
 - 1.3. *En kötü uygunluk değerine sahip çocuk bireyi belirle*
 - 1.4. *En iyi popülasyon üyesi en kötü çocuk bireyden daha iyi ise en kötü çocuk bireyle en iyi popülasyon üyesini değiştir ve bir sonraki nesli $(m-1)$ çocuk birey ve en iyi popülasyon üyesi olarak belirle*
 - 1.5. *En iyi popülasyon üyesi en kötü çocuk bireyden daha iyi değil ise bir sonraki nesli m çocuk birey olarak belirle*

Şekil 4.20 : Elitist strateji sözde kodu.

1. *Sonraki popülasyonu belirle*

1.1 *m popülasyon üyesinden GA operatörleri (çaprazlama ve mutasyon) ile 1 çocuk birey üret*

1.2 *m popülasyon üyesi ve 1 çocuktan oluşan havuzdaki en iyi m üyeyi uygunluk değerlerine göre değerlendir ve uygunluk değerine göre en iyi m üyeyi bir sonraki nesil olarak belirle*

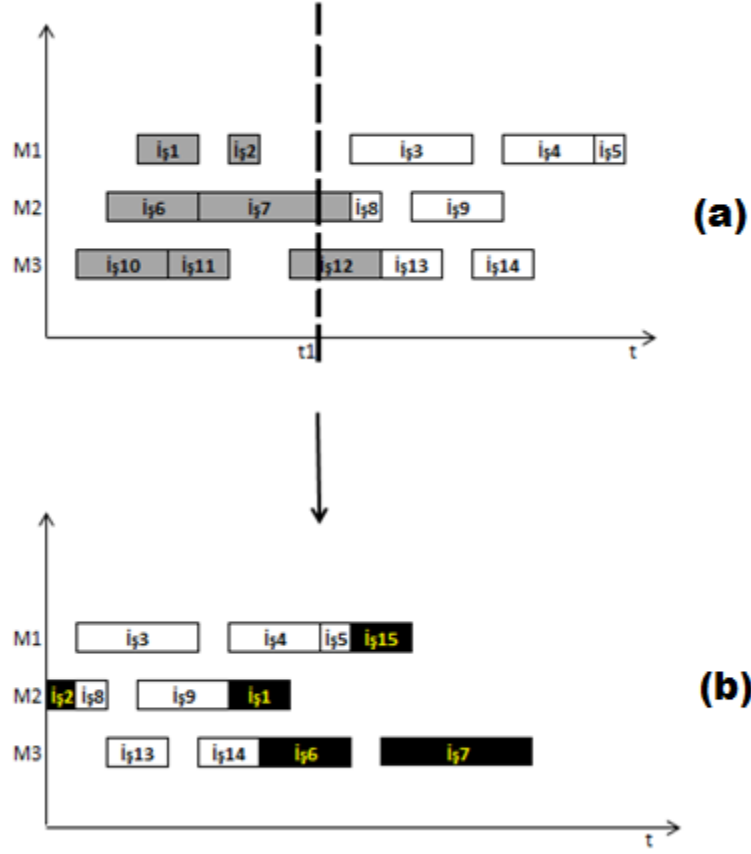
Şekil 4.21 : Uygunluk tabanlı strateji sözde kodu.

Bir sonraki neslin belirlenmesi politikası olarak, elitist strateji ve uygunluk tabanlı strateji örnek problemler üzerinde test edilmiştir. Beşinci bölümde, GA stratejilerinin belirlenmesi kısmında hangi politikanın algoritmada kullanılmak üzere nasıl seçildiği detaylı olarak açıklanmıştır.

4.2.8 Popülasyondaki bireylerin güncellenmesi

Bu tez çalışması kapsamında incelenen melez akış tipi çizelgeleme probleminin dinamik yapısı gereği problem zaman içinde değişmektedir. Sisteme yeni bir iş gelişi ya da sistemdeki bir işin tamamlanması ile ele alınan problem değişebilmektedir.

Geliştirilmesi düşünülen GA'nın ortamdaki değişimlere uyum sağlayabilen tepkisel bir karakterde olması planlanmaktadır. Buna göre üretim sistemindeki herhangi bir değişim problemi değiştirmekte, problem değişiminin ise GA'ya yansıtılması gerekmektedir. Sisteme yeni bir iş gelişi ya da sistemdeki üretim aşamalarından herhangi birindeki bir işin o aşamadan ayrılışı, popülasyondaki bireylerin güncellenmesini gerektirmektedir. Bu ise, herhangi bir üretim aşamasına yeni gelen işlerin GA kromozomlarına yeni iş olarak eklenmesi, o aşamadan ayrılan işlerin ise kromozomlardan silinmesi yolu ile gerçekleştirilebilir.



Şekil 4.22 : Dinamik bir üretim sisteminin t_1 anında yeni iş geldiğindeki durumu.

Şekil 4.22 (a)'da her aşamasında bir tane makine olan üç aşamalı bir üretim sistemi verilmiştir. Aşamalardaki makineler sırası ile M1, M2 ve M3 makineleridir. Sistemdeki herhangi bir iş, sırası ile bütün aşamalardan geçerek işlem görmektedir. Sistemin t_1 anındaki fotoğrafı çekilmiştir. Gri boyalı işler t_1 anından önce o aşamadaki makinede tamamlanmış ya da işleme başlanmış işleri, beyaz boyalı işler t_1 anından sonra o aşamadaki makinede işlenmesi planlanmış işleri, siyah boyalı işler ise t_1 anında ya da t_1 anından sonra o aşamaya yeni gelmiş veya transfer olmuş işleri göstermektedir. Bu problemde referans noktası olarak t_1 anının seçilmesinin nedeni, t_1 anında 1.Aşama'da bulunan M1 makinesine yeni bir iş gelmiştir. Şekil 4.22 (b)'de sisteme yeni gelen iş, M1 makinesinde en son sıradaki siyah boyalı 15 no'lu iş olarak gösterilmiştir.

Bu tez çalışması kapsamında olay tabanlı bir yeniden çizelgeleme yaklaşımı benimsenmiştir. Sisteme yeni bir iş gelişi ya da herhangi bir aşamadaki bir işin tamamlanması yeniden çizelgeleme noktası olarak tanımlanmıştır. Buna göre her yeniden çizelgeleme noktasında GA'nın çözümlerinin güncellenmesi gerekmektedir.

Şekil 4.22’de verilen sistemin yeni iş gelişi öncesi durum (Şekil 4.22 (a)) ve yeni iş gelişi sonrası durum (Şekil 4.22 (b)) olarak ele alındığı görülmektedir. t_1 anı öncesinde 1. Aşama’da M1 makinesinde işlenmiş ya da işlemeye devam eden İş1 ve İş2, t_1 anı sonrası durumda M1 makinesinde gösterilmemiştir. t_1 anı öncesinde 2.Aşama’da M2 makinesinde işlenmiş ve işlenmeye başlamış İş6 ve İş7, t_1 anı sonrası durumda M2 makinesinde gösterilmemiştir. Yine aynı şekilde 3. Aşamada tamamlanmış ya da işlenmeye devam eden İş10, İş11 ve İş12, t_1 anı sonrası durumda M3 makinesinde gösterilmemiştir. t_1 anı sonrası durum işlenmesi planlanan, yeni gelen ya da önceki aşamalardan transfer edilen işlerden oluşturulmuştur. 1. Aşama’daki işler t_1 anından sonra M1 makinesinde işlenmesi planlanan İş3, İş4, İş5 ve yeni gelen İş15’ten oluşturulmuştur. 2. Aşama’daki işler t_1 anından sonra M2 makinesinde işlenmesi planlanan İş8, İş9 ve 1. Aşama’daki işlemleri tamamlanmış olan İş1 ve İş2’den oluşturulmuştur. Yine aynı şekilde 3. Aşama’daki işler M3 makinesinde t_1 anından sonra işlenmesi planlanan İş13, İş14 ve 2. Aşama’daki işlemleri tamamlanmış ya da t_1 anında devam eden İş6 ve İş7’den oluşturulmuştur.

GA açısından bakıldığında yeni iş geldiğinde ya da herhangi bir aşamadaki bir iş tamamlandığında sistemde yaşanan değişimin popülasyondaki kromozomlara yansıtılması gerekir. Güncellenen işlere göre GA operatörleri uygulanarak algoritma çözüm aramaya devam eder.

4.2.8.1 Çözüm uzayı değişimi sonrası permutasyonların oluşturulması

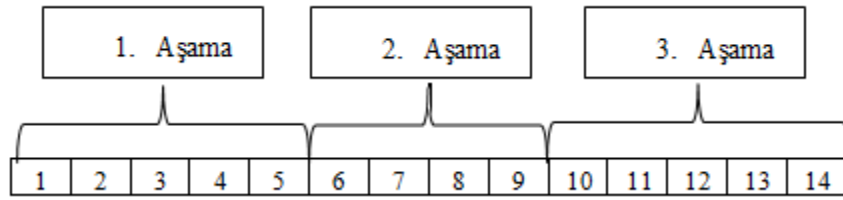
Sisteme yeni bir iş gelmesi veya herhangi bir aşamadaki bir işin tamamlanması gibi çözüm uzayı değişimleri sonrasında güncellenen üretim aşamalarına göre iş permutasyonlarının yeniden oluşturulması gerekmektedir. Üretim sisteminin herhangi bir aşamasına yeni bir işin eklenmesi ya da transfer edilmesi veya bir aşamadaki herhangi bir işin tamamlanması sonrası o aşamadan silinmesi sonrasında oluşan yeni duruma göre popülasyondaki çözümlerin oluşturulması gerekmektedir.

Burada üzerinde durulması gereken konu çözüm uzayı değişimi sonrasında iş permutasyonlarının nasıl oluşturulacağıdır. Bilinen en basit yaklaşımda her çözüm uzayı değişimi sonrasında problem değiştiği için güncellenen koşullara göre rassal permutasyonlardan oluşan yeni bir ilk popülasyon oluşturulur. Buna göre sisteme her yeni iş gelişi ya da herhangi bir aşamadaki bir işin bir sonraki aşamaya transfer olması gibi dinamik durum oluşması sonucu permutasyonlar yeniden oluşturulur.

Dinamik durum oluşması öncesindeki permutasyonlar silinir ve yeni duruma göre sıfırdan rassal permutasyonlar oluşturulur. Bu yaklaşımın en zayıf yanı GA'nın değişim öncesi duruma kadar olan yaptığı arama süreci kazanımlarının değişim sonrası permutasyonların rastgele oluşturulması ile kaybedilmesidir.

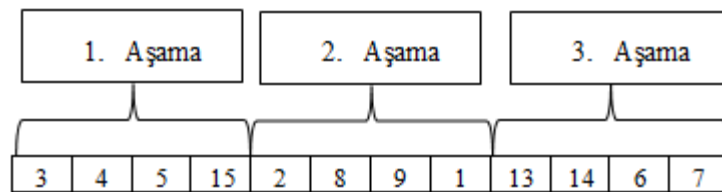
Bir diğer yaklaşımda ise GA'nın dinamik durumun olduğu ana kadar olan arama süreci kazanımlarının muhaza edilmesini amaçlanır. Önceki yaklaşımda olduğu gibi bu yaklaşımda da dinamik durum olduğunda çözüm uzayı ve problem değişimi gerçekleştirilir. Önceki yaklaşımdan farklı olarak değişim sonrası durumda yeni popülasyon, değişim öncesi durumun final popülasyonu üzerinden gerekli güncellemeler yapılarak oluşturulur. Bu yaklaşımda arama süreci kazanımlarının kaybedilmemesi amaçlanır. Şekil 4.22'de verilen örnek ele alındığında değişim öncesi duruma ait GA'nın atölyeye verdiği en iyi çözüm:

$\pi_1 = 1,2,3,4,5$; $\pi_2 = 6,7,8,9$; $\pi_3 = 10,11,12,13,14$ 'tür.



Şekil 4.23 : Yeniden çizelgeleme öncesi durumun çözüm gösterimi.

t_1 anından sonra atölye'nin durumuna göre oluşan çözüm ise $\pi_1 = 3,4,5,15$; $\pi_2 = 2,8,9,1$; $\pi_3 = 13,14,6,7$ şeklinde güncellenir.



Şekil 4.24 : Yeniden çizelgeleme sonrası durumun çözüm gösterimi.

Şekil 4.23 ve 4.24'de görüldüğü üzere çözüm uzayı değişimi sonrasında tamamen rastgele yeni bir permutasyon oluşturulmak yerine mevcut en iyi permutasyon üzerinde yeni durumun gereklerine göre güncellemeler yapılarak dinamik durum sonrası yeni çözüm permutasyonu oluşturulmuştur. Bu yaklaşımda mevcut kromozomlar üzerinde değişim sonrası problemde olmayan işler silinir yeni gelen

işler ise mevcut popülasyondaki kromozomlara rassal bir pozisyonda eklenerek arama sürecine devam edilir.

Bu tez çalışması kapsamında ortam değişimi sonrasında permutasyonların sıfırdan rassal olarak oluşturulması yerine ikinci yaklaşımda olduğu gibi değişim öncesi durumun final popülasyonu üzerinde gerekli değişiklikler yapılarak popülasyonun güncellenmesine karar verilmiştir. Buna göre sisteme yeni gelen bir iş çözümlere ekleneceği zaman o işe karşılık olarak bir rassal sayı kromozomlara eklenir. Eğer bir işin bulunduğu aşamadaki işlemi tamamlanmışsa o işe ait rassal sayı, o işin işlendiği aşamayı temsil eden kromozomlardan silinir ve bir sonraki aşamanın kromozomlarına transfer edilen iş olarak eklenir. Rassal sayılar küçükten büyüğe doğru sıralanarak permutasyonlar, iş sıraları oluşturulur. Ekleme ve silmeler haricinde rassal sayılar üzerinde başka bir işlem yapılmaz. Bu yüzden kromozomlarda bulunan mevcut işlere ait rassal sayıların değerleri değişmez. Dolayısı ile mevcut işlere ait sıralama değişmez, sadece yeni gelen iş ona atanan rassal sayı doğrultusunda önceki iş sırası içinde kendine bir yer bulur. Şekil 4.25'te üretim sisteminde değişiklik olduğunda popülasyonun nasıl güncellendiğini gösteren algoritmanın sözde kodu verilmiştir.

1. Popülasyondaki bireyleri güncelle

1.1. Üretim sisteminde dinamik bir durum oluştu mu

1.1.1. Üretim aşamalarında tamamlanan işleri belirle

1.1.2. Üretim aşamalarına yeni gelen ya da bir önceki aşamadan transfer olan işleri belirle

1.1.3. Tamamlanan işleri mevcut popülasyondaki kromozomlardan sil

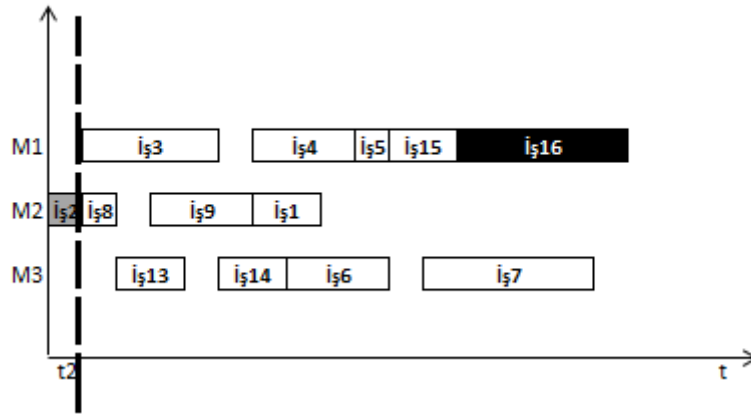
1.1.4. Aşamalara yeni gelene ya da bir önceki aşamadan transfer edilen işleri kromozomlara her bir iş için bir rassal sayı vererek ekle

1.1.5. Mevcut popülasyondaki en uygun bireyi belirle ve atölyeye en iyi çizelge olarak öner

Şekil 4.25 : Popülasyonun güncellenmesi sözde kodu.

4.2.9 GA evrilme koşulu (durma koşulu)

Tasarlanan çizelgeleme sisteminde, GA'nın sürekli arama yapması ortam değişimi olan ana kadar ki kazanımları değişim sonrasında oluşan probleme aktararak aramaya devam etmesi amaçlanmıştır. Sistemin o anki durumuna göre GA'nın ürettiği en iyi çizelge atölye ile paylaşılır. Atölye bu çizelgeye göre işlemlerini gerçekleştirir. Sisteme yeni iş geldiğinde ya da sistemin herhangi bir aşamasında işlem gören bir işin o aşamadaki işi bittiğinde bu durum GA'ya çözüm uzayı değişimi olarak yansır. GA mevcut popülasyonu atölyeden gelen bilgiler doğrultusunda güncelleyerek arama sürecine devam eder. Bu sayede sistemde olabilecek küçük değişimlerin bile anında GA'ya iletilmesi ve algoritmanın en güncel duruma göre arama sürecine devam etmesi sağlanır.



Şekil 4.26 : Dinamik bir üretim sisteminin herhangi bir andaki durumu.

Şekil 4.26'da, Şekil 4.22'de verilen üretim sisteminde t_2 anına gelindiğinde sistemin durumu gösterilmiştir. t_2 anında 1. Aşamaya yeni bir iş (İş16) gelmiş ve 2. Aşamada bir iş (İş2) tamamlanmıştır. Buna göre t_2 anında güncellenen çözüm:

$\pi_1 = 3,4,5,15,16$; $\pi_2 = 8,9,1$; $\pi_3 = 13,14,6,7,2$ şeklinde oluşur. Atölyedeki en son duruma göre popülasyondaki kromozomlar güncellenir. GA, çözüm uzayını yani işleri güncellemek için atölyeyi referans alır. Popülasyondaki çözümlere yeni gelen işler rassal olarak yerleştirilir, tamamlanan ya da işi biten işler ise çözümlerden silinirler. Atölyedeki aşamalarda işlem göreceği olan yeni işler o aşamaları temsil eden kromozomlara rastgele bir pozisyonla eklenir, herhangi bir aşamada işi biten işler ise o aşamaları temsil eden kromozomlardan silinirler.

Sistemdeki deęişimlerin algoritmaya entegrasyonunun yanında üzerinde durulması gereken dięer bir konu GA'nın evrilme ya da durma koşulunun ne olması gerektięidir. Literatürde GA'ların durma koşulu olarak birkaç farklı yöntem önerilmiştir. Buna göre GA'nın ürettięi çözümler birbirine yakınsayıp sürekli aynı çözümlerin üretilmeye başlanması aynı çözümlerin nesiller boyunca tekrarlanması artık algoritmanın daha fazla iyileştirme yapmasına gerek kalmadığını gösterir ve bu noktada durulur. Bu doğrultuda aynı çözümlerin tekrarlanma sayısı bir durma koşulu olarak gösterilebilir. Literatürde sıkça durma koşulu olarak kullanılan bir dięer koşul ise maksimum nesil sayısı olarak gösterilebilir. GA önceden belirlenmiş bir nesil sayısı kadar çalıştırılır ve bu sayıya ulaşırsa durulur.

Literatürde dinamik ortamlardaki GA çizelgeleme uygulamalarına bakıldığında araştırmacıların çoęu zaman durma koşulu olarak maksimum nesil sayısını seçtięi görülmektedir (Zhang ve dię, 2013). Her yeniden çizelgeleme noktasında çözüm uzayının büyüklüğüne göre maksimum nesil sayısını dinamik olarak belirleyen çalışmalarla da karşılaşmaktadır (Bierwirth ve Mattfeld, 1999).

Araştırmacılar yeniden çizelgeleme öncesi durum ve deęişim sonrası durumları ayrı ayrı ele alıp GA'yı belirlenen bir nesil sayısı kadar çalıştırıp bu şekilde yeniden çizelgelemeyi gerçekleştirmektedirler. Bu yaklaşımlarda yeniden çizelgeleme noktasında zaman durdurulmakta ve GA belirli bir süre çalıştırıp bulunan çözüm atölyeye verilmektedir. Dinamik problem statik alt problemlere bölünmekte ve her bir statik probleme GA ile çözüm üretilmektedir.

Bu tez çalışmasında önerilen dinamik çizelgeleme yaklaşımı, dinamik problemi statik alt problemlere bölerek çözüm üretmek yerine, üretim ortamındaki en küçük deęişikliklerin dahi hızlı bir şekilde GA kromozomlarına yansıtıldığı sürekli ve küçük küçük iyileştirmelere güvenen dinamik bir yapıdadır. Buna göre GA'nın durma koşulu olarak önceden belirlenmiş bir nesil sayısı ya da üst sınır yoktur. GA dinamik durum ya da çizelgede deęişkenlik gerektiren olay gerçekleşinceye kadar çalışır, dinamik durum oluştuğunda popülasyondaki en iyi çözüm atölyedeki deęişen koşullara göre uyarlanır ve bu çözüm atölyeye en iyi çizelge olarak verilir.

Bizim yaklaşımımızda dinamik durum gerçekleştiğinde zaman durdurulmaz. GA'nın çözüm üretmesi sırasında geçen süre ihmal edilmez ve bu zaman çok küçük bir süre dahi olsa atölyeye yansıtılır. Atölyede gerçekleştirilecek en küçük deęişimler dahi

anlık olarak kromozomlara entegre edilir ve arama sürecine hiç ara verilmez. Popülasyonlar üzerinde güncellemeler yapılarak arama sürecine devam edilir.

4.3 Tasarlanan Çizelgeleme Sistemi

Tasarlanan çizelgeleme sistemi Şekil 3.1’de verilen gerçek hayattaki gibi bir dinamik çizelgeleme sürecini temel almaktadır. Tasarlanan çizelgeleme sisteminde başlangıçtaki durum için GA tarafından üretilen en iyi çözüm atölyeye çizelge olarak verilir. Atölye, çizelgeye göre iş emirlerini gerçekleştirir. Sistemde herhangi bir dinamik durumun oluşması sonucu değişen atölye ve sistemin durumu GA’ya bildirilir. GA’nın çözümleri atölyenin durumuna göre güncellenerek popülasyondaki en iyi çözüm atölyeye en iyi çizelge olarak verilir ve çizelgeleme sistemi bu şekilde çalışmaya devam eder. Tasarlanan çizelgeleme sisteminin genel çerçevesi Şekil 4.27’de verilmiştir.

1. *GA ile başlangıçtaki durum için çözüm üret ve bu çözümü çizelge olarak belirle*
2. *Çizelgeyi atölyede gerçekleştir*
3. *Sistemde dinamik bir durum meydana geldi mi*
 - 3.1. *GA popülasyonunu güncelle ve en iyi çözümü çizelge olarak belirle*
 - 3.2. *Adım 2’ye git*
4. *GA’yı bir nesil koştur*
 - 4.1. *Adım 3’e git*

Şekil 4.27 : Tasarlanan çizelgeleme sisteminin genel çerçevesi.

Tasarlanan çizelgeleme sistemi temelde çözüm üretme, çözüm gerçekleştirme ve çözüm güncelleme şeklindedir. Tasarlanan çizelgeleme sisteminde Adım 3’e gelindiğinde eğer sistemde dinamik bir durum oluşmamış ise Adım 4’e geçilir ve mevcut popülasyon üzerinden GA bir nesil daha çalıştırılır. Adım 3’e dönülür atölyeye bakılır. Eğer sisteme yeni bir iş gelmiş ya da üretim aşamalarında herhangi birindeki bir iş tamamlanmış ise GA popülasyonu güncellenir ve en iyi çözüm çizelge olarak atölyeye verilir. Eğer dinamik bir durum meydana gelmemiş ise GA bir nesil daha

koşturulur. Böylece dinamik durum meydana gelinceye kadar GA'nın sürekli olarak çalışması ve daha iyi çözümlere evrilmesi amaçlanır.

Tasarlanan çizelgeleme sisteminde çözüm gerçekleştirme aşaması yer almaktadır. Gerçek hayattaki durumu simüle edebilmek için böyle bir aşamaya ihtiyaç duyulmuştur. Literatürdeki dinamik çizelgeleme GA uygulamalarına bakıldığında, GA tarafından önerilen en iyi çözümün anlık olarak atölyede gerçekleştirildiğinin varsayıldığı görülmektedir. Çözüm üretimi için geçen süre atölyeye yansıtılmamakta, her şey anlık olarak gerçekleştirilmektedir. Daha önceden planlanan işlem süreleri üzerinden hesaplamalar gerçekleştirilmektedir. Bu yüzden ayrıca bir çözüm gerçekleştirme aşamasına ihtiyaç duyulmamaktadır.

Bu tez çalışması kapsamında gerçek hayattakine benzer bir çizelgeleme sisteminin tasarlanması amaçlanmıştır. Ayrıca işlem zamanları gerçek hayattakine benzer olarak değişkenlik göstermektedir. Bu doğrultuda bir çizelge gerçekleştirme mekanizması geliştirilmiş ve tasarlanan çizelgeleme sistemine entegre edilmiştir.

4.3.1 GA tarafından önerilen bir çözümün atölyede gerçekleştirilmesi

Tasarlanan çizelgeleme sistemi çözüm üretme, çizelge gerçekleştirme ve çözüm güncelleme adımlarından oluşmaktadır. GA tarafından önerilen en iyi çözüm atölyeye çizelge olarak iletilir. Çözüm üretme, çizelge gerçekleştirme ve çözüm güncelleme esnasında geçen zaman kaydedilir. GA, dinamik bir durum oluşuncaya kadar çalıştırılır. Eğer sistemde dinamik bir durum oluşmuş ise, GA çalışmayı durdurur ve en iyi çözümünü atölyeye çizelge olarak verir. Eğer sistemde dinamik bir durum oluşmamış ise GA bir nesil daha çalıştırılır ve yine dinamik bir durum oluşup oluşmadığı kontrol edilir. Sistemde dinamik bir durum oluşmuş ise atölye o anki sistem zamanını referans olarak GA tarafından önerilen çizelgeyi gerçekleştirir.

Çizelge gerçekleştirme ile yapılan, GA tarafından önerilen çizelgenin atölyede ilerletilmesidir. Çizelge doğrultusunda işler, makineler atanırlar. Atölyedeki makineler temelde müsait ve meşgul olmak üzere iki durumda bulunabilirler. Başlangıça, atölyedeki bütün makineler müsaittir. Sistemde dinamik bir durum oluştuğunda önerilen çizelge, seçilen en küçük zaman birimi kadar ilerletilir. Makinelerin müsaitlik durumu o anki sistem zamanı ve makineler atanmış işlerin işlem zamanları doğrultusunda güncellenir. Buna göre müsait yani boşta bulunan bir makine, o makineye atanmış bir iş olması durumunda yeniden çizelgeleme anından

itibaren o işin işlem süresi boyunca meşgul olacaktır. Bir başka deyişle çizelge gerçekleşmesi, atölyedeki makinelerin müsaitlik durumlarının, o makinelerde işlem gören işlerin ve makinelerin işleme başlama ve bitiş zamanlarının güncellenmesidir.

Atölye gerçekleşmesi sonrasında GA bir nesil daha çalıştırılır. GA'nın çalışması sırasında geçen zaman çok küçük bir süre dahi olsa hesaplanır ve toplam sistem zamanına eklenir. Sistemde dinamik bir durum oluşup oluşmadığı kontrol edilir. Sistemde dinamik bir durum oluştuğunda GA'nın mevcut popülasyonu o anki atölye şartlarına göre güncellenir. GA güncellenirken atölyedeki makinelerin o anki durumu, işlem gören işler ve bu işlerin tamamlanma zamanları dikkate alınır. Güncellenen popülasyondaki en iyi çözüm adayı yeni çizelge olarak atölyeye önerilir. Seçilen en küçük zaman birimi kadar önerilen çizelge atölyede ilerletilir ve bu şekilde bir sonraki dinamik durum oluşuncaya kadar GA'nın çalıştırılması, çizelgenin ilerletilmesi ve çözüm popülasyonunun güncellenmesine devam edilir.

Sisteme yeni iş gelişi ya da herhangi bir aşamadaki bir işin tamamlanması olmak üzere temelde iki dinamik durum oluşabilir. Dinamik durum oluştuğunda, atölyedeki aşamalarda işlem görmesi planlanmış kuyrukta bekleyen, işlem gören, işi biten ve atölyeye yeni gelen işlerin durumu dikkate alınır. Atölyenin o anki durumu doğrultusunda mevcut çözüm popülasyonu güncellenir ve popülasyondaki en uygun ya da en iyi diye nitelendirilebilecek çözüm atölyeye yeni çizelge olarak verilir.

GA planlanan işlem zamanlarına göre çözüm permutasyonlarını ve çizelgeyi üretir. Üretilen en iyi çözüm atölyede ilerletilirken fiziksel dünya koşulları da dikkate alınarak çizelge gerçekleşir. Çözüm üretilmesi için geçen zaman atölyeye yansıtılır. Örneğin GA'nın önerdiği en iyi çözümde birinci sıradaki işin işlenmeye başlanması için öngörülen zaman $t=0$ olabilir. Bununla birlikte GA'nın çözüm üretmesi için 0.01 birim zaman harcanmış olabilir. Dolayısı ile birinci sıradaki iş atölyede en erken $t=0.01$ zamanında işlenmeye başlayabilir ve birinci sıradaki işin atölyede işleme başlama zamanı $t=0.01$ olarak değiştirilir. Atölyedeki çizelge sonuçlarının GA'nın önerdiği planlanan çizelgeden farklı olmasının tek sebebi çözüm üretilmesi için geçen zaman değildir. Ele alınan dinamik melez akış tipi çizelgeleme probleminin stokastik yapısı gereği işlem zamanları kesin olarak belli değildir, belli bir olasılık dağılımından gelmektedir.

GA planlanan işlem zamanlarına göre çizelgeyi oluştursa da, atölyede çizelge gerçekleşmesi sırasında problemin olasılıksal yapısı gereği belirlenen dağılımın ortalamasına ve standart sapmasına bağlı olarak işlerin işlem zamanları realize edilir. Buna göre örneğin GA'nın önerdiği çizelgede bir işin işlem zamanı 5 birim olabilir, ama bu işin atölyede işlenmesi 5.2 birim sürede gerçekleşebilir.

Şekil 4.28'de önerilen bir çözümün çizelge olarak atölyede gerçekleşmesine ilişkin sözde kod verilmiştir.

1. Çizelgeyi atölyede gerçekleştir

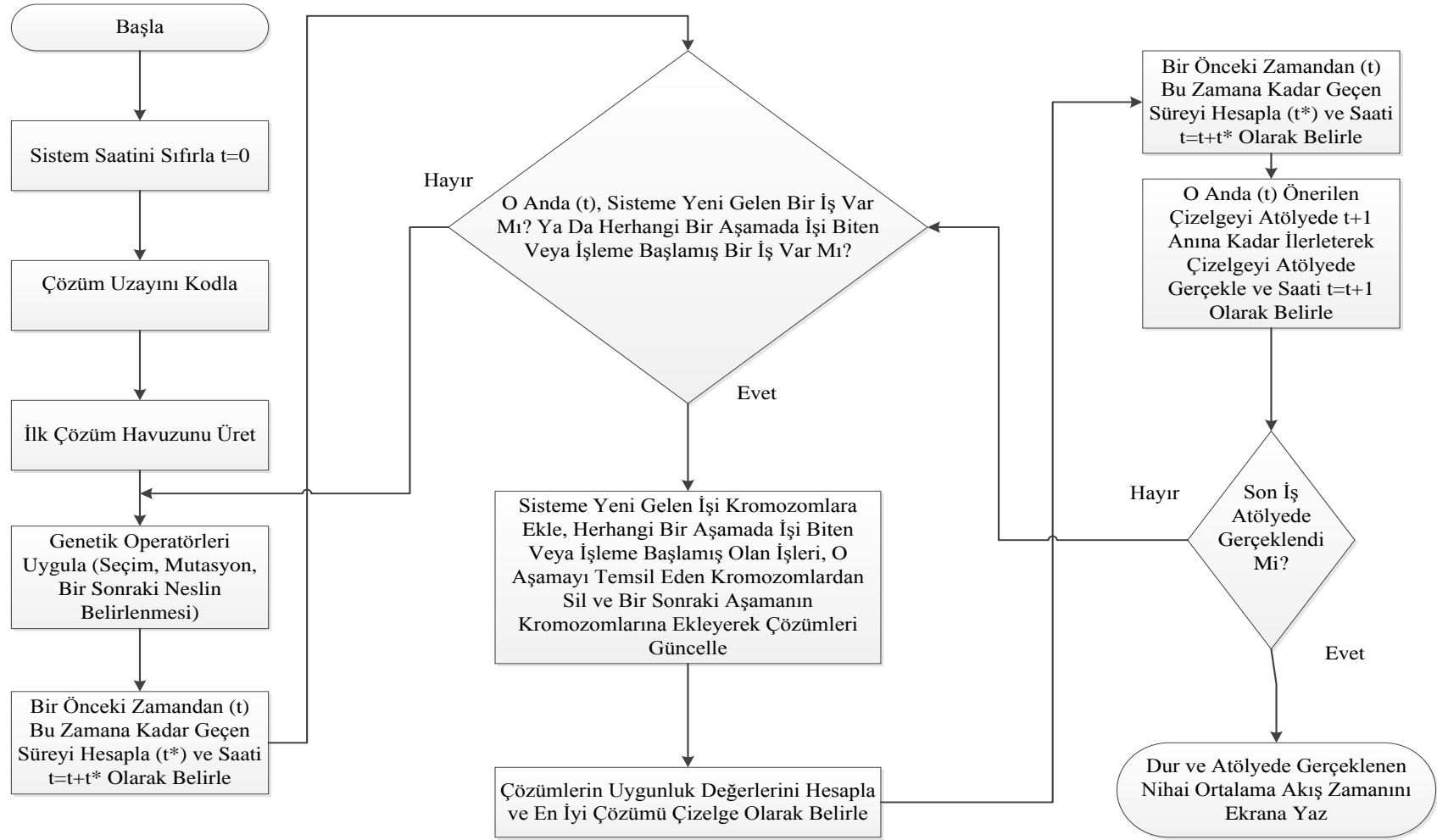
1.1. Güncel sistem zamanını kontrol et, ve o anki zamanı tespit et

1.2. Çizelgeyi atölyede bir birimlik zaman dilimi kadar ilerlet

1.3. Sistem zamanını bir birim ilerlet

Şekil 4.28 : Çizelgenin atölyede gerçekleşmesi.

Bu tez çalışması kapsamında ele alınan dinamik melez akış tipi çizelgeleme problemine önerilen GA yaklaşımı Şekil 4.29'da verilen akış şeması ile özetlenebilir.



Şekil 4.29 : Ele alınan dinamik melez akış problemine yönelik olarak önerilen GA yaklaşımının akış diyagramı.

5. DENEYLER VE SONUÇLARIN ANALİZİ

Bu kısımda ele alınan problem için geliştirilen GA'nın operatör ve strateji alternatiflerinin analizi gerçekleştirilmiştir. Stratejiler test problemleri ile performanslarına göre kıyaslanmış ve en iyi alternatifler seçilmiştir. Parametre optimizasyonu yapılarak ele alınan problem için en iyi parametre seti belirlenmiştir.

Önerilen GA'nın performansını değerlendirebilmek için, öncelikle statik ve deterministik durumda küçük problem boyutları için CPLEX çözücü ile önerilen GA'nın performansı kıyaslanmıştır. Daha sonra dinamik ve deterministik durumda önerilen GA ile literatürdeki alternatif algoritmaların performansları çeşitli test problemleri üzerinde test edilmiştir. Son olarak dinamik ve stokastik durumda önerilen GA'nın performansını değerlendirebilmek için bir prosedür geliştirilmiş ve önerilen yaklaşımın performansı analiz edilmiştir.

5.1 Çizelgeleme Ortamının Düzenlenmesi

GA stratejilerinin belirlenmesi ve önerilen yaklaşımın performansının analiz edilmesi öncesinde deney ortamının parametrelerinin belirlenmesi gerekmektedir. Çizelgeleme ortamı belirlenirken aşağıdaki faktörlerin dikkate alınmasına karar verilmiştir:

- İş geliş hızı λ ,
- Makine kullanım oranı U ,
- İş sayısı N ,
- Üretim aşaması sayısı k ,
- Aşama makine sayısı m ,

Literatürdeki dinamik çizelgeleme çalışmalarında iş geliş hızının ve makine kullanım oranının çizelgeleme ortamının belirlenmesinde üstünde durulan en kritik faktörlerden olduğu görülmüştür. Sisteme gelen işlerin gelişler arası süresi üretim sistemindeki iş yükünü etkileyen başlıca etmenlerdendir.

Herhangi bir üretim sisteminde uzun vadede kuyruk uzunluğunun sonsuza gitmemesi için, iş geliş hızının (λ), servis hızından (μ) küçük ya da eşit olması, $\lambda \leq \mu$ gerekmektedir. Sistem yoğunluğu ($\rho = \frac{\lambda}{\mu}$), $\rho < 1$ için uzun vadede bu sisteminin kuyruk uzunluğunun sonsuza ulaşmayacağı söylenebilir. Buna göre dinamik bir üretim sistemi için kabul edilebilir bir ortalama iş gelişler arası süresi $1/\lambda$, ortalama işlem zamanının sistemdeki makine sayısına oranlanması ile bulunabilir.

Ele alınan dinamik melez akış tipi çizelgeleme problemi birden çok aşamadan ve aşamalar ise özdeş olmayan paralel makinelerden oluşan bir yapıdadır. Bir i işinin üretim aşamalarındaki (j, l, \dots, k) ortalama işlem zamanları birbirine eşittir ($\bar{P}_{ij} = \bar{P}_{il} = \dots = \bar{P}_{ik}$) ve her aşamada eşit sayıda m özdeş olmayan paralel makine vardır. Buna göre herhangi bir aşamadaki bir işin ortalama işlem zamanının o aşamadaki makine sayısına oranı, ortalama gelişler arası sürenin belirlenmesinde bir limit değer olarak alınabilir.

Herhangi bir üretim sisteminde %100 oranında makine kullanımı çok olası bir durum değildir. Sistemin ortalama gelişler arası süresi istenilen makine kullanım oranına göre belirlenebilir. Buna göre ortalama gelişler arası süre, aşamalar için ortalama işlem zamanının, aşamalardaki makine sayısı ve makine kullanım oranının çarpımına bölünmesi ($\frac{1}{\lambda} = \frac{\bar{P}_j}{m \cdot U}$) ile bulunabilir.

Ele alınan üretim sistemindeki işlem zamanları 1 ile 19 arasındaki düzgün dağılımdan, $p_j \sim D(1,19)$ gelmektedir. Sistemdeki işlerin gelişler arası süresi ise ortalaması $1/\lambda$ olan üstel dağılımdan, $US(\frac{1}{\lambda})$ gelmektedir.

Çizelgeleme ortamındaki test problemleri aşağıdaki faktör seviyelerine göre belirlenebilir:

- İş sayısı: 50, 100, 250
- Aşama sayısı: 3, 5, 10
- Makine sayısı: 3, 6, 10
- Makine kullanım oranı: 0.6, 0.75, 0.9

Çizelge 5.1 : Test problemleri.

Problem Sınıfı	İş Sayısı	Aşama Sayısı	Makine Sayısı	Makine Kullanım Oranı
Kolay	50	3	3	0.6
Orta Düzey	100	5	6	0.75
Zor	250	10	10	0.9

Faktör seviyelerine göre test problemleri Çizelge 5.1'deki gibi Kolay, Orta Düzey ve Zor olmak üzere sınıflandırılabilir. Kolay problem sınıfı diğer problemlere göre görece daha az iş içermekte ve işler daha seyrek bir sıklıkta üretim sistemine gelmektedir. Zor iş sınıfı ise diğer problem sınıflarına göre görece daha fazla iş içermekte ve işler üretim sistemine daha sık gelmektedir. Alternatif test problemleri ile önerilen yaklaşımın farklı durumlardaki performansının gözlemlenebilmesi amaçlanmıştır. Bütün testlerde çizelgelenmesi planlanan toplam iş sayısının %10'un başlangıçta (t=0) anında sistemde mevcut olduğu farz edilmiştir.

5.2 GA Stratejilerinin Seçimi

Bölüm 4.2.1'de bir çözüm adayına ait çizelge oluşturulurken iki farklı makine atama kuralından bahsedilmiştir. Bölüm 4.2.4 ve bölüm 4.2.7'de geliştirilmesi düşünülen GA için, alternatif ebeveyn seçim metotları ve sonraki nesil belirleme stratejileri verilmiştir. Önerilen GA yaklaşımı için en iyi strateji kombinasyonunu belirleyebilmek için bir dizi deney çalışması yapılmıştır. Kolay, orta düzey ve zor sınıf test problemleri bu deneylerde en iyi strateji alternatiflerinin belirlenmesi için kullanılmıştır. GA strateji alternatifleri Şekil 5.1'de verilmiştir.

Faktör Düzeyleri	Karar Adımı: Çizelge Oluşturma (I)
I1	En Hızlı Boş Makineye Atama Kuralı
I2	En Küçük Tamamlanma Zamanı Makine Atama Kuralı
Faktör Düzeyleri	Karar Adımı: Ebeveyn Seçimi (F)
F1	Turnuva Seçim
F2	Rulet Tekerlek Seçim
Faktör Düzeyleri	Karar Adımı: Bir Sonraki Popülasyonun Belirlenmesi (S)
S1	Elitist Strateji
S2	Uygunluk Tabanlı Strateji

Şekil 5.1 : GA strateji alternatifleri.

Strateji alternatiflerinin performanslarını değerlendirebilmek için sistematik bir yaklaşım uygulanmıştır (Güçlü, 2006). Strateji alternatifleri farklı problem sınıfları üzerinde uygulanarak daha iyi sonuç veren faktör düzeyleri tespit edilmiştir.

Strateji alternatiflerinin performansları atölyede gerçekleşen ortalama akış zamanına göre değerlendirilmiştir. Her strateji kombinasyonu için 5 koşum yapılarak o strateji alternatifi için ortalama değer kaydedilmiştir. Bulunan ortalama akış zamanları ise gerçekleştirilecek olan analize girdi olarak alınmıştır. Analiz, en etkili karar adımlarının tespiti ve her karar adımı için en iyi performanstaki faktör düzeylerinin belirlenmesi olarak özetlenebilir. Toplamda 8 ($=2*2*2$) tane farklı strateji kombinasyonu mevcuttur. Ayrıca her alternatif kolay, orta ve zor problem sınıflarının her birinde beşer replikasyon ile uygulanmıştır. Toplamda 120 ($=8*3*5$) deney gerçekleştirilmiştir.

En etkili faktörleri bulabilmek için Varyans Çözümlemesi (ANOVA) yönteminden yararlanılmıştır. Varyans Çözümlemesi gerçekleştirilerek, en etkili karar adımı belirlenmeye çalışılmıştır. En etkili karar adımı belirlendikten sonra, o karar adımının en iyi faktör düzeyi t-testi ile belirlenmiştir.

Varyans Çözümlemesini gerçekleştirebilmek için test verisi Şekil 5.2'deki gibi düzenlenmiştir.

Kombinasyon	Rep1	Rep2	...	Rep5	5 Replikasyonun Ortalama Akış Zamanları Ortalaması
I1-F1-S1	$Y_{1,1}$	$Y_{1,2}$...	$Y_{1,5}$	\bar{Y}_{1j}
I1-F1-S2	$Y_{2,1}$	$Y_{2,2}$...	$Y_{2,5}$	\bar{Y}_{2j}
...
I2-F2-S2	$Y_{8,1}$	$Y_{8,1}$...	$Y_{8,5}$	\bar{Y}_{8j}

Şekil 5.2 : Her problem sınıfı için Varyans Çözümlemesi veri seti planı.

Her üç problem sınıfı için de Varyans Çözümlemesi gerçekleştirilmiştir. Bir karar adımının faktör düzeyleri arasında anlamlı derecede bir fark olup olmadığını analiz edebilmek için ise t-testi gerçekleştirilmiştir. t-testini gerçekleştirebilmek için veri seti Şekil 5.3'teki gibi düzenlenmiştir.

Kombinasyon	Rep1	Rep2	...	Rep5
F_{S1}	$\bar{Y}_{S1,1}$	$\bar{Y}_{S1,2}$...	$\bar{Y}_{S1,5}$
F_{S2}	$\bar{Y}_{S2,1}$	$\bar{Y}_{S2,2}$...	$\bar{Y}_{S2,5}$

Şekil 5.3 : t-testi veri seti planı.

F_{SL} , veri seti içinde S faktörünün L düzeyine ait kombinasyonlarını simgelemektedir. Şekil 5.3'te verilen satırlar, herhangi bir faktör için daha iyi faktör düzeyine sahip olduğu düşünülen durum ve eşleniği arasında anlamlı bir fark var mı yok mu diye test edilmesi öncesinde verinin nasıl düzenlendiğini göstermektedir.

Tüm analiz prosedürü aşağıdaki gibi özetlenebilir:

1. Varyans Çözümlemesi veri setine uygulanarak en etkili faktör ve bu faktöre ait en düşük ortalama değere sahip en iyi faktör düzeyi belirlenir.
2. Eşlenik t-testi uygulanarak 1. adımda seçilen faktör düzeyinin eşlenik faktör düzeyinden anlamlı bir derecede farklı olup olmadığı test edilir.
3. Eğer seçilen faktör düzeyi anlamlı derecede diğer faktör düzeyinden farklı ise, seçilen faktör düzeyi sabitlenir ve veri seti eşlenik faktör düzeyi hariç tutularak güncellenir.

Yukarıdaki prosedür, Varyans Çözümlemesi ile etkili faktör bulunamayana kadar ya da t-testi ile faktör düzeyleri arasında anlamlı derecede fark bulunamayana kadar uygulanmaya devam edilir. Üç problem sınıfı için de belirlenen 8 ($=2^3$) farklı strateji alternatifi, önerilen GA yaklaşımı doğrultusunda 5 kere uygulanarak ortalama akış zamanları üretilir ve faktör düzeyi kombinasyonları test edilir. Analiz prosedürünün test problemlerine uygulanması öncesinde GA parametre ilk değerlerinin belirlenmesi gerekir. Analiz prosedürü aşamaları için GA parametre ilk değerleri Çizelge 5.2'deki gibi seçilmiştir. Bununla birlikte GA parametre optimizasyonuna bir sonraki bölümde ayrıca değinilecektir.

Çizelge 5.2 : GA parametre ilk değerleri.

GA Parametreleri	İlk Değer
Popülasyon Büyüklüğü	100
Çaprazlama Olasılığı	1
Mutasyon Olasılığı	0.1

Varyans Çözümlemesi ilk olarak “kolay” problem sınıfına uygulandığında bulunan analiz sonuçları Çizelge 5.3’te verilmiştir. I karar adımı (Çizelge Oluşturma) en etkili faktör olarak görülmektedir.

Çizelge 5.3 : Kolay problem sınıfı 1. Aşama varyans çözümleme analizi sonuçları.

Değişimin Kaynağı	Kareler Toplamı(KT)	Serbestlik Derecesi(SD)	Kareler Ortalaması(KO)	F Oranı
KT _I	93.1626	1	93.1626	2113.02
KT _F	0.0001	1	0.0001	0.00
KT _S	0.3350	1	0.3350	7.60
KT _E	1.5872	36	0.0441	
KT _T	95.0849			

Çizelge 5.4’e göre I2 (En Küçük Tamamlanma Zamanı Makine Atama Kuralı) daha düşük ortalama akış zamanına sahiptir.

Çizelge 5.4 : Kolay problem sınıfı 1. aşama faktör düzeyleri ortalama akış zamanları.

Ortalama Akış Zamanı	
I1	19.94
I2	16.88

Eğer t-testi sonucu I2’nin anlamlı derecede I1’den daha iyi olduğunu gösterirse, I2 faktör düzeyi veri seti için sabitlenecek ve I1 faktör düzeyi hariç tutularak analize devam edilecektir.

H_0 : I2 faktör düzeyi anlamlı derecede I1 faktör düzeyinden farklı değildir

H_1 : I2 faktör düzeyi anlamlı derecede I1 faktör düzeyinden farklıdır.

Eşlenik t-testi p-değeri <0.001 bulunmuştur. Bulunan p-değeri=0.001 < 0.025 olduğu için, %95 güven düzeyinde H_0 hipotezi reddedilir. Buna göre I2 daha iyi bir performansa sahiptir, faktör düzeyi olarak I2 seçilir. Kolay problem sınıfı 1. aşama analiz prosedürü böylelikle tamamlanmış olur. Analiz prosedürüne I1 faktörü veri setinden hariç tutulup, I2 faktör düzeyi sabitlenerek Varyans Çözümlemesi ve t-testi ile etkili bir faktör tespit edilemeyinceye kadar kalan F ve S faktörlerine de aynı şekilde analiz prosedürü uygulanarak devam edilir. Kolay, orta düzey ve zor problem sınıfları için analiz prosedürü sonuçları Şekil 5.4, Şekil 5.5 ve Şekil 5.6'da verilmiştir.

Kolay Problem Sınıfı					
1. Aşama					
Değişimin Kaynağı	KT	SD	KO	F	p-Değeri
KT _I	93.1626	1	93.1626	2113.02	<0.001
KT _F	0.0001	1	0.0001	0.00	0.95
KT _S	0.335	1	0.335	7.60	0.01
KT _E	1.5872	36	0.0441		
KT _T	95.0849				
Ortalama Akış Zamanları		eşlenik t-testi p-değeri			
I1	19.94	I1			
I2	16.88	I2	<0.001		
I2 anlamlı derecede I1 den daha iyidir, bu yüzden I2 veri setinde sabitlenir					
2. Aşama					
Değişimin Kaynağı	KT	SD	KO	F	p-Değeri
KT _F	0.009786	1	0.009786	0.31	0.58
KT _S	0.212098	1	0.212098	6.72	0.01
KT _E	0.536447	17	0.031556		
KT _T	0.758331				
Ortalama Akış Zamanları		eşlenik t-testi p-değeri			
S1	16.98	S1			
S2	16.78	S2	0.07		
S2 anlamlı derecede S1 den daha iyidir denilemez					

Şekil 5.4 : Kolay problem sınıfı analiz prosedürü aşamaları.

Orta Düzey
Problem Sınıfı
1. Aşama

Değişimin Kaynağı	KT	SD	KO	F	p-Değeri
KT _I	328.69	1	328.69	8548.55	<0.001
KT _F	0.085	1	0.085	2.2	0.15
KT _S	4.09	1	4.09	106.36	0.00
KT _E	1.384	36	0.038		

KT_T 334.248

Ortalama Akış Zamanları

I1	23.76
I2	18.03

eşlenik t-testi p-değeri

I1	
I2	<0.001

I2 anlamlı derecede I1 den daha iyidir, bu yüzden I2 veri setinde sabitlenir.

2.Aşama

Değişimin Kaynağı	KT	SD	KO	F	p-Değeri
KT _F	0.00026	1	0.00026	0.07	0.78
KT _S	1.87652	1	1.87652	530.35	<0.001
KT _E	0.06015	17	0.00354		

KT_T 1.93693

Ortalama Akış Zamanları

S1	18.33
S2	17.72

eşlenik t-testi p-değeri

S1	
S2	<0.001

S2 anlamlı derecede S1 den daha iyidir, bu yüzden veri setinde S2 sabitlenir.

Şekil 5.5 : Orta düzey problem sınıfı analiz prosedürü aşamaları.

Zor Problem Sınıfı					
1. Aşama					
Değişimin Kaynağı	KT	SD	KO	F	p-Değeri
KT _I	248.889	1	248.889	3137.04	<0.001
KT _F	0.079	1	0.079	1.00	0.33
KT _S	6.844	1	6.844	86.26	<0.001
KT _E	2.856	36	0.079		
KT _T	258.668				
Ortalama Akış Zamanları		eşlenik t-testi p-değeri			
I1	29.39	I1			
I2	24.40	I2	<0.001		
I2 anlamlı derecede I1 den daha iyidir, bu yüzden I2 veri setinde sabitlenir.					
2. Aşama					
Değişimin Kaynağı	KT	SD	KO	F	p-Değeri
KT _F	0.00356	1	0.00356	0.28	0.61
KT _S	1.08667	1	1.08667	85.06	<0.001
KT _E	0.21717	17	0.01277		
KT _T	1.30740				
Ortalama Akış Zamanları		eşlenik t-testi p-değeri			
S1	24.63	S1			
S2	24.16	S2	0.001		
S2 anlamlı derecede S1 den daha iyidir, bu yüzden veri setinde S2 sabitlenir.					

Şekil 5.6 : Zor problem sınıfı analiz prosedürü aşamaları.

Kolay, orta ve zor problemler için analiz prosedürünün uygulanması sonucunda I (Çizelge Oluşturma) ve S (Bir Sonraki Popülasyonun Belirlenmesi) faktörleri, I2 ve S2 düzeyleri için en etkili faktörler olarak belirlenmiştir. F (Ebeveyn Seçimi) faktörü analiz sonuçlarına göre etkili bir faktör olarak gözükmesine de, F1 faktör düzeyi ortalama akış zamanı sonuçları ortalaması her üç problem sınıfı için de F2 faktör düzeyi ortalama akış zamanı sonuçları ortalamasından daha düşük çıkmıştır. Şekil 5.7’de F faktörünün her iki düzeyi için ortalama sonuçlar verilmiştir.

F1 Faktör Düzeyi Ortalama Akış Zamanları		F2 Faktör Düzeyi Ortalama Akış Zamanları	
Kolay	16.77	Kolay	16.79
Orta	17.71	Orta	17.73
Zor	24.16	Zor	24.17

Şekil 5.7 : Kolay, orta ve zor problem sınıfları için F faktörü ortalama sonuçları.

Bu doğrultuda, F1(Turnuva Seçim) faktör düzeyi, geliştirilecek olan GA'nın ebeveyn seçim metodu olarak belirlenmiştir. Analiz prosedürü sonuçları ve seçilen strateji alternatifleri Şekil 5.8 ve Çizelge 5.5'te özetlenmiştir.

1. Aşama			
Problem Tipi	En Etkili Faktör	En İyi Faktör Düzeyi	t-testi Sonucu
Kolay	I	I2	Anlamlı derecede farklı
Orta	I	I2	Anlamlı derecede farklı
Zor	I	I2	Anlamlı derecede farklı
2. Aşama			
Problem Tipi	En Etkili Faktör	En İyi Faktör Düzeyi	t-testi Sonucu
Kolay	S	S2	Anlamlı derecede farklı değil
Orta	S	S2	Anlamlı derecede farklı
Zor	S	S2	Anlamlı derecede farklı

Şekil 5.8 : Analiz prosedürü özet sonuçları.

Çizelge 5.5 : Analiz prosedürü sonucu seçilen GA stratejileri.

Karar Adımı	Seçilen Faktör Düzeyi	Strateji Alternatifi
Çizelge Oluşturma (I)	I2	En Küçük Tamamlanma Zamanı Makine Atama Kuralı
Bir Sonraki Popülasyonun Belirlenmesi (S)	S2	Uygunluk Tabanlı Strateji
Ebeveyn Seçimi (F)	F1	Turnuva Seçim

5.3 GA Parametre Optimizasyonu

Önerilen GA'nın parametre optimizasyonunun gerçekleştirilmesi gerekmektedir. Bu kısımda optimizasyonu gerçekleştirilecek olan parametreler:

- 1) Çaprazlama Oranı (P_c)
- 2) Mutasyon Oranı (P_m)

şeklindedir. Literatürde akış tipi ve atölye tipi çizelgeleme problemi GA uygulamalarında kullanılan çaprazlama oranı ve mutasyon oranı değerleri dikkate alınarak çaprazlama ve mutasyon oranı düzeylerine karar verilmiştir. Bu çalışma kapsamında kullanılan çaprazlama oranı ve mutasyon oranı düzeyleri Çizelge 5.6 ve Çizelge 5.7'de verilmiştir.

Çizelge 5.6 : Çaprazlama oranı alternatifleri.

Faktör Düzeyi	Çaprazlama Oranı (P_c)
X1	1
X2	0.9
X3	0.8
X4	0.7
X5	0.6

Çizelge 5.7 : Mutasyon oranı alternatifleri.

Faktör Düzeyi	Mutasyon Oranı (P_m)
M1	0.001
M2	0.01
M3	0.05
M4	0.1
M5	0.2

25 (=5*5) kombinasyon alternatifinin her biri için 5 kere olmak üzere, önerilen GA yaklaşımı toplamda 125 kere zor problem sınıfına uygulanarak ve her bir durum için atölyede gerçekleşen ortalama akış zamanları bulunmuştur. GA parametre kombinasyonlarına ait bulunan ortalama değerlere ait sonuçlar Çizelge 5.8'de verilmiştir.

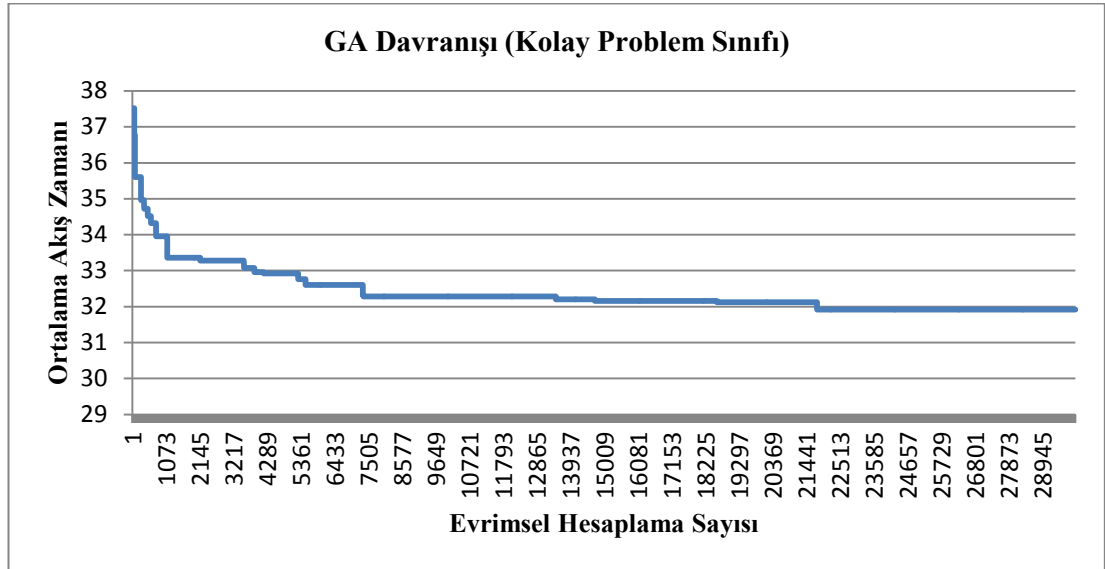
Çizelge 5.8 : 5 replikasyon için ortalama akış zamanı sonuçlarının ortalamaları.

No	Kombinasyon	Ortalama Akış Zamanı
1	X1-M1	24.157
2	X1-M2	24.165
3	X1-M3	24.227
4	X1-M4	24.225
5	X1-M5	24.180
6	X2-M1	24.173
7	X2-M2	24.191
8	X2-M3	24.183
9	X2-M4	24.166
10	X2-M5	24.179
11	X3-M1	24.202
12	X3-M2	24.195
13	X3-M3	24.185
14	X3-M4	24.159
15	X3-M5	24.219
16	X4-M1	24.193
17	X4-M2	24.176
18	X4-M3	24.197
19	X4-M4	24.122
20	X4-M5	24.180
21	X5-M1	24.202
22	X5-M2	24.181
23	X5-M3	24.219
24	X5-M4	24.216
25	X5-M5	24.204

Sonuç olarak X4-M4 24.122 en düşük ortalama sonuç ile en iyi parametre kombinasyonu olarak bulunmuştur. Bu doğrultuda geliştirilecek olan GA'da çaprazlama oranının, $P_c = 0.7$ ve mutasyon oranının, $P_m = 0.1$ olarak belirlenmesine karar verilmiştir.

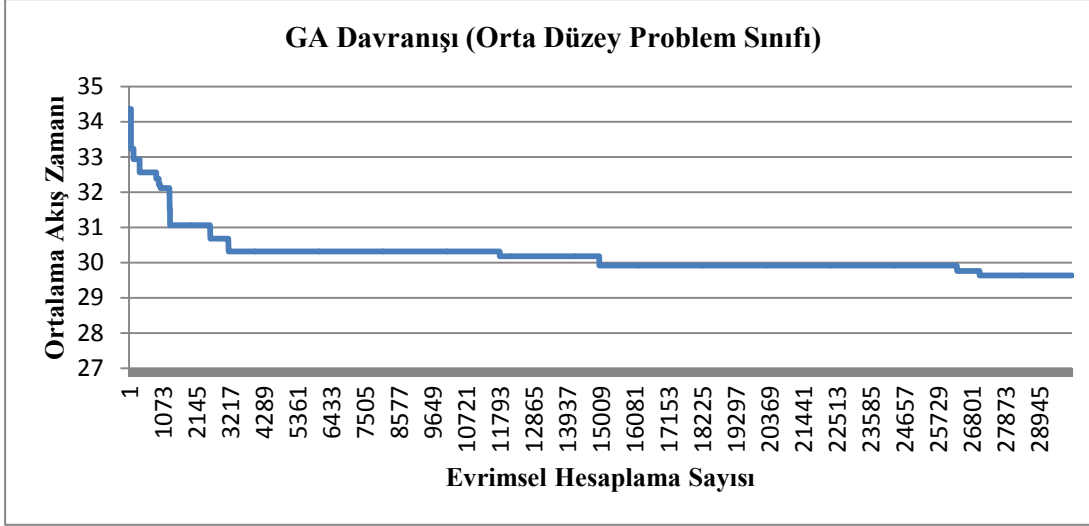
5.4 GA'nın Davranışının Gözlemlenmesi

Strateji seçimi ve parametre optimizasyonunun gerçekleştirilmesinden sonra, geliştirilmesi amaçlanan GA'nın tasarlanan test problemleri üzerindeki davranışı gözlemlenmiştir. Buna göre öncelikle kolay problem sınıfındaki bir probleme, sistemde herhangi bir anda ortalama 25 iş varken GA uygulandığında algoritmanın davranışı Şekil 5.9'daki gibi gözlemlenmiştir.



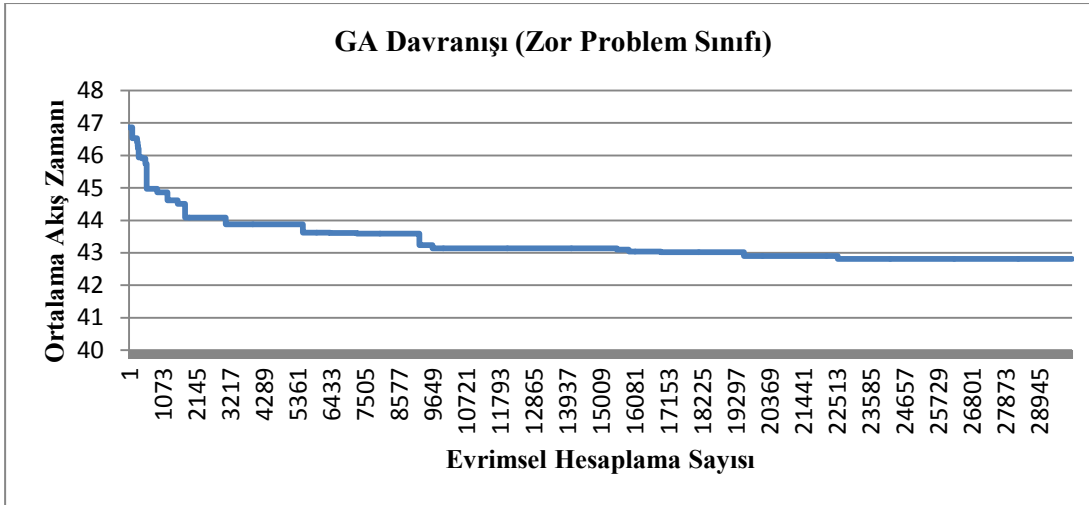
Şekil 5.9 : Belirlenen stratejiler ve parametreler doğrultusunda GA'nın kolay problem sınıfı üzerindeki davranışı.

Orta düzey problem sınıfındaki bir probleme, sistemde herhangi bir anda ortalama 50 iş varken GA uygulandığında algoritmanın davranışı Şekil 5.10'daki gibi gözlemlenmiştir.



Şekil 5.10 : Belirlenen stratejiler ve parametreler doğrultusunda GA'nın orta düzey problem sınıfı üzerindeki davranışı.

Zor problem sınıfındaki bir probleme, sistemde herhangi bir anda ortalama 125 iş varken GA uygulandığında algoritmanın davranışı Şekil 5.11'deki gibi gözlemlenmiştir.



Şekil 5.11 : Belirlenen stratejiler ve parametreler doğrultusunda GA'nın zor problem sınıfı.

Her üç durumda da GA, minimizasyon probleminin karakterine uygun olarak başlangıçta hesapladığı ortalama akış zamanı değerlerinden daha düşük değerler bulma eğilimindedir. Kolay, orta düzey ve zor problem sınıfları için GA'nın bulduğu ortalama akış zamanı değerleri nesilden nesle daha iyi çözümlere doğru evrilerek belli bir değere yakınsama eğilimindedir. Bu doğrultuda seçilen stratejiler ve belirlenen parametreler doğrultusunda geliştirilen GA'nın nesilden nesle daha iyi

çözümlere üretebilme yeteneğine sahip olduğu söylenebilir. GA hızlı bir şekilde önceki nesilden daha iyi çözümler bularak yakınsamaktadır. Buna göre seçilen stratejiler ve belirlenen parametreler doğrultusunda geliştirilen GA'nın tutarlı sonuçlar ürettiği söylenebilir.

5.5 Stokastik Durumda GA Uygunluk Değeri Koşum Sayısının Belirlenmesi

Bölüm 4.2.3.1'de işlem zamanlarının olasılıksal olduğu durumda uygunluk değeri hesaplanırken simülasyondan yararlanılacağına değinilmiştir. Stokastik durumda, bir çözüme ait beklenen uygunluk değeri, o çözüm için birden fazla kere hesaplatılıp bulunan değerlerin ortalaması alınarak belirlenir. Herhangi bir çözüm adayı için kaç kere uygunluk değeri hesaplatılacağına tespit edilmesi gerekmektedir. Başka bir deyişle bir çözümün simülasyonunda gerçekleştirilmesi gereken koşum sayısına karar verilmesi gerekmektedir.

α anlamlılık düzeyi ve ϵ hata değeri için koşum sayısı n ,

$$n \geq \left(\frac{Z_{\alpha/2} \cdot \sigma}{\epsilon} \right)^2 \quad (5.1)$$

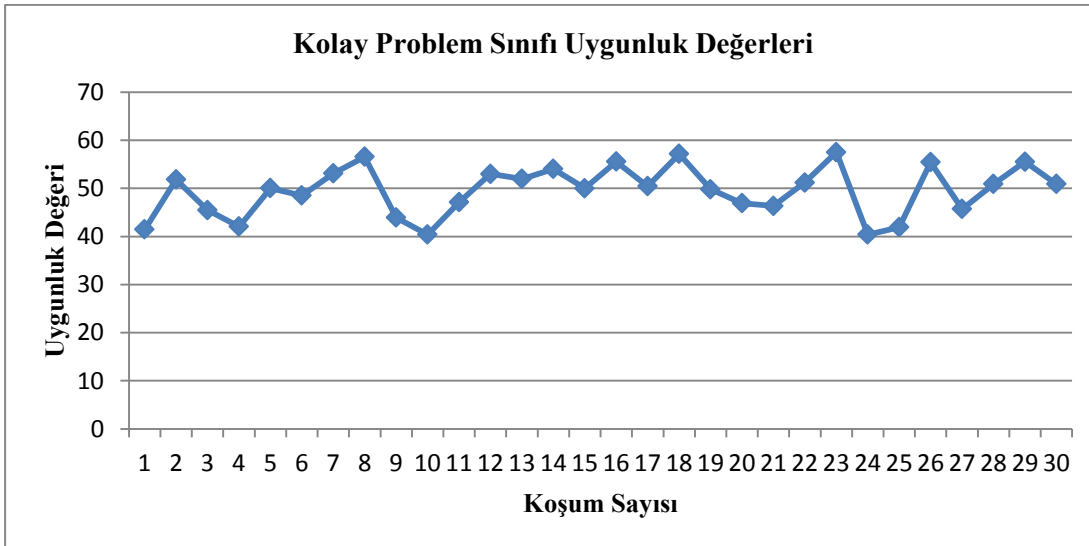
formülü ile bulunabilir.

Rastgele seçilen bir n değeri için bir başlangıç güven aralığı bulunabilir ve daha sonra n değeri artırılıp ya da azaltılarak istenilen hassasiyete ulaşılabilir. Bu tez çalışması kapsamında koşum sayısının karar verilmesi için sistematik bir yaklaşım uygulanmıştır. İlk olarak bir başlangıç koşum sayısı belirlenerek stokastik durumda çözüm adayları için beklenen uygunluk değerleri hesaplanmıştır. Kolay, orta düzey ve zor problem sınıfları için sistemde herhangi bir anda bulunabilecek ortalama iş sayısı dikkate alınarak seçilen başlangıç koşum sayısına göre herhangi bir çözüm için uygunluk değerleri üretilmiştir. Üretilen uygunluk değerlerinin standart sapmaları hesaplanarak seçilen koşum sayısı ve problem tipi için güven aralığı belirlenmiştir. Üç problem sınıfı için de istenilen hassasiyet düzeyi göz önüne alınarak alternatif koşum sayıları belirlenmiştir. Daha sonra kolay, orta ve zor problem sınıfları için alternatif koşum sayıları ile önerilen GA yaklaşımı ile her alternatif için 5 kere ortalama akış zamanı bulunmuş ve o alternatif için atölyede gerçekleşen ortalama akış zamanı değerlerinin ortalaması alınarak nihai sonuç bulunmuştur. Bütün problem sınıfları ve bütün koşum sayısı alternatifleri için aynı prosedür tekrarlanarak

ortalama sonuçlar bulunmuştur. Son olarak en iyi koşum sayısına karar vermek için, her problem sınıfı ve alternatif koşum sayıları için hesaplanan değerlerin kendi problem sınıfındaki en iyi sonuca yüzde olarak ne kadar uzak olduğu hesaplanmıştır. Koşum sayısı karar adımları aşağıdaki gibi özetlenebilir:

1. Bir başlangıç koşum sayısı belirlenerek kolay, orta düzey ve zor problem sınıfları için seçilen koşum sayısınca uygunluk değerleri hesaplanır ve her üç problem sınıfı için de standart sapma ve güven aralığı değerleri bulunur.
2. Üç problem sınıfı için de istenilen hassasiyet düzeyi göz önüne alınarak alternatif koşum sayıları belirlenir.
3. Önerilen GA yaklaşımı alternatif koşum sayılarında her üç problem sınıfına da uygulanarak ortalama akış zamanı değerleri bulunur.

Koşum sayısının başlangıç değeri 30 olarak seçilmiştir. Kolay problem sınıfında herhangi bir anda sistemde ortalama 25 ($=50/2$) iş bulunabilir. Buna göre kolay problem sınıfında sistemde herhangi bir anda 25 işin bulunduğu durum için, GA tarafından önerilen bir çözüm adayının beklenen uygunluk değeri o çözüm adayı için 30 kere hesaplanarak bulunmuştur. Hesaplanan uygunluk değerlerinin grafiği Şekil 5.12'de verilmiştir.



Şekil 5.12 : Stokastik durumda kolay problem sınıfı için herhangi bir çözüm adayının uygunluk değerleri.

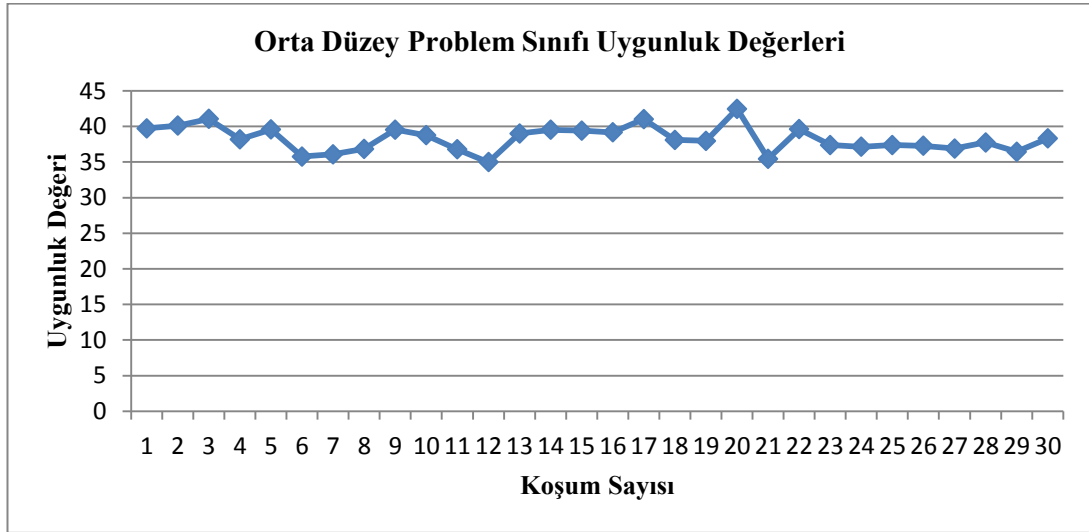
Buna göre beklenen uygunluk değeri, üretilen $n = 30$ uygunluk değerinin ortalaması alınarak $49.51 (= \frac{41,44+51,84+\dots+50,92}{30})$ bulunmuştur. Uygunluk değerlerinin standart

sapması $\sigma = 5.17$ bulunmuştur. Bu durumda kolay problem sınıfı için $\alpha = 0.05$ anlamlılık düzeyinde yarı güven aralığı h ;

$$h = \frac{Z_{\alpha/2} * \sigma}{\sqrt{n}} \quad (5.2)$$

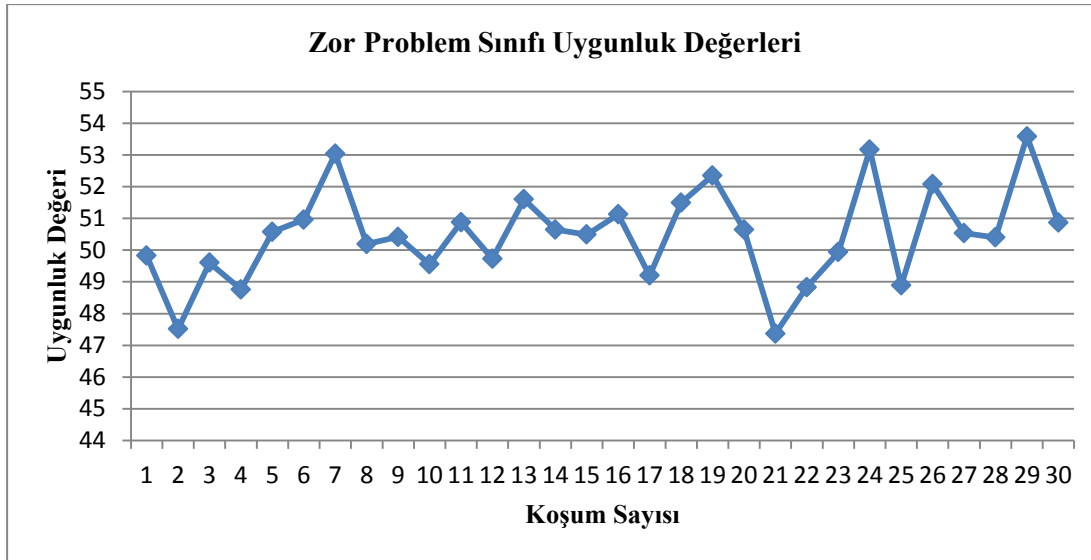
formülü ile 1,85 olarak ($h_{kolay}=1.85$) bulunmuştur.

Orta düzey problem sınıfı için de koşum sayısı $n = 30$ seçilerek, sistemde herhangi bir anda ortalama 50 ($=100/2$) işin bulunduğu düşünülerek herhangi bir çözüm adayı için beklenen uygunluk değeri bulunmuştur. Şekil 5.13'de seçilen çözüm adayı için üretilen uygunluk değerlerinin grafiği verilmiştir.



Şekil 5.13 : Stokastik durumda orta düzey problem sınıfı için herhangi bir çözüm adayının uygunluk değerleri.

Seçilen çözüm adayının beklenen ortalama uygunluk değeri 38.25 standart sapması ise 1.77 olarak bulunmuştur. Buna göre seçilen çözüm adayı için orta düzey problem sınıfındaki yarı güven aralığı $h_{orta} = 0.64$ olarak hesaplanmıştır. Aynı şekilde zor problem sınıfı için herhangi bir anda sistemde ortalama 125 ($=250/2$) işin bulunduğu bir durum için $n = 30$ koşum sayısı ile seçilen bir çözüm adayı için uygunluk değerleri Şekil 5.14'deki gibi bulunmuştur.



Şekil 5.14 : Stokastik durumda zor problem sınıfı için herhangi bir çözüm adayının uygunluk değerleri.

Seçilen çözüm adayının beklenen uygunluk değeri 50.47 standart sapması ise 1.49 bulunmuştur. Buna göre yarı güven aralığı $h_{zor} = 0.53$ olarak hesaplanmıştır.

Alternatif koşum sayılarına karar verilirken bulunan yarı güven aralıkları ve beklenen uygunluk değerlerinden yararlanılmıştır. Örneğin kolay problem sınıfı için bulunan yarı güven aralığının beklenen uygunluk değerine oranlanması ($1.85/49.51=0.037$) ile seçilen koşum sayısında hesaplanan uygunluk değerlerinin hassasiyet yüzdesi belirlenebilir. Kolay problem sınıfı için koşum sayısı 50'ye çıkarıldığında hesaplanan yarı güven aralığı 1.43'ün ($=\frac{1.96*5.17}{\sqrt{50}}$) yine beklenen uygunluk değeri 49.51'e oranlanması ile 50 koşum sayısı için bulunan oran ise 0.028'dir. Bu şekilde 10 ile 50 arasındaki koşum sayıları için her üç problem sınıfındaki hassasiyet yüzdeleri hesaplanarak alternatif koşum sayılarının ne olması gerektiğine karar verilmiştir. Alternatif koşum sayıları ve problem tipleri için hesaplanan hassasiyet değerleri Çizelge 5.9'da verilmiştir.

Çizelge 5.9 : Alternatif koşum sayıları ve problem tipleri için hesaplanan hassasiyet oranları.

Koşum Sayısı	Kolay Problem Sınıfı	Orta Düzey Problem Sınıfı	Zor Problem Sınıfı
10	0.064	0.028	0.012
15	0.052	0.023	0.010
20	0.045	0.020	0.009
25	0.040	0.018	0.008
30	0.037	0.016	0.007
35	0.034	0.015	0.007
40	0.032	0.014	0.006
45	0.030	0.013	0.006
50	0.028	0.012	0.005

Çizelge 5.9'da görüldüğü üzere koşum sayısı arttıkça hassasiyet oranları düşmektedir. Bu tez çalışması kapsamında %5 hassasiyet düzeyinin üst sınır olarak alınmasına karar verilmiştir. Çizelge 5.7'da görüleceği üzere 20, 25, 30, 35, 40, 45 ve 50 koşum sayıları için hesaplanan hassasiyet düzeyleri belirlenen üst sınır olan %5 hassasiyet düzeyinin altındadır. Bununla birlikte 15 koşum sayısı için kolay problem sınıfında ancak %5.2 hassasiyet düzeyi ile uygunluk değerleri hesaplanabilmektedir ve belirlenen üst sınır olan %5'in üzerindedir. Bu doğrultuda bu tez çalışması kapsamında alternatif koşum sayılarının 20, 25, 30, 35, 40, 45 ve 50 olarak belirlenmesine karar verilmiştir.

Koşum sayısı alternatifleri belirlendikten sonra, kolay, orta ve zor problem sınıfları için alternatif koşum sayıları ile önerilen GA yaklaşımı ile her alternatif için 5 kere ortalama akış zamanı bulunarak nihai ortalama sonuçlar elde edilmiştir. Çizelge 5.10'da alternatif koşum sayıları için hesaplanan ortalama akış zamanı değerleri verilmiştir.

Çizelge 5.10 : Alternatif koşum sayıları ve problem boyutları için 5 deneme sonunda hesaplanan ortalama akış zamanı değerleri.

Koşum Sayısı	Kolay Problem Sınıfı	Orta Düzey Problem Sınıfı	Zor Problem Sınıfı
20	25.87	36.06	79.20
25	26.44	36.99	79.84
30	26.73	38.14	82.75
35	26.03	35.91	84.03
40	26.52	38.40	82.96
45	26.29	37.18	89.78
50	26.79	36.22	88.78
En İyi Ortalama Sonuç	25.87	35.91	79.20

En iyi koşum sayısına karar vermek için, her problem sınıfı ve alternatif koşum sayıları için hesaplanan değerlerin kendi problem sınıfındaki en iyi sonuca yüzde olarak ne kadar uzak olduğu tespit edilir ve bu değerlerin ortalamaları alınarak oransal ortalama uzaklıklar hesaplanır. Buna göre alternatif koşum sayıları için üretilen ortalama akış zamanlarının o problem sınıfındaki en iyi ortalama sonuçla arasındaki fark hesaplanır ve bu fark o problem sınıfındaki en iyi ortalama sonuca oranlanır. Her problem sınıfı için bulunan oransal uzaklıkların ortalaması alınarak oransal ortalama uzaklıklar bulunur. Çizelge 5.11’de ortalama akış zamanı değerlerinin en iyi ortalama sonuçlara yüzde olarak uzaklıkları verilmiştir.

Çizelge 5.11 : Alternatif koşum sayıları ve problem sınıfları için hesaplanan değerlerin en iyi ortalama sonuçlara oransal uzaklıkları.

Koşum Sayısı	Kolay Problem Sınıfı	Orta Düzey Problem Sınıfı	Zor Problem Sınıfı	Oransal Ortalama Uzaklık
20	%0	%0.42	%0	%0.14
25	%0.80	%3.01	%2.20	%2.00
30	%4.48	%6.20	%3.30	%4.66
35	%6.10	%0	%0.62	%2.24
40	%4.73	%6.93	%2.48	%4.72
45	%13.35	%3.53	%1.62	%6.17
50	%12.08	%0.86	%3.56	%5.50

Çizelge 5.11'deki oransal ortalama uzaklık sütununa bakıldığında koşum sayısı 20 için hesaplanan oransal uzaklık değeri 0.14'ün diğer koşum sayısı alternatiflerinin oransal ortalama uzaklık değerlerinden daha iyi olduğu görülmektedir. Koşum sayısı arttıkça atölyede gerçekleşen nihai ortalama akış zamanı değerleri büyümektedir. Bu beklenmedik bir durum değildir çünkü bir çözüm adayına ait beklenen uygunluk değerini ne kadar çok deneme ile bulursanız o kadar zaman geçmekte bu zaman da atölyede gerçekleşen ortalama akış zamanına yansımaktadır.

Sonuç olarak bu tez çalışması kapsamında stokastik durumda bir çözüm adayının beklenen uygunluk değeri bulunurken koşum sayısının 20 olmasına karar verilmiştir.

5.6 Önerilen GA Yaklaşımının Performansının Analizi

Bu tez çalışması kapsamında ele alınan melez akış tipi çizelgeleme problemine yönelik olarak önerilen GA yaklaşımının performansı öncelikle en basit durum diyebileceğimiz statik ve deterministik durumda küçük problem boyutları için optimum çözümlerle kıyaslanmıştır. GA'nın bir anlamda iyi çözümler bulabilme yeteneği test edilmeye çalışılmış ayrıca atölyede çözüm gerçekleştirme bu aşamada yapılmamıştır. Önerilen GA yaklaşımı C# programlama dili ile kodlanmış ve Intel Core i5 1.70 GHz işlemcili bir bilgisayarda çalıştırılmıştır.

Bu doğrultuda işlem zamanlarının $D(1,19)$ dağılımına uyduğu 5 yeni test problemi geliştirilmiştir. Çizelge 5.12'de test problemlerinin parametreleri verilmiştir.

Çizelge 5.12 : Statik deterministik durum için geliştirilen test problemleri parametreleri.

Problem	İş Sayısı	Aşama Sayısı	Makine Sayısı
P1	8	2	2
P2	9	2	3
P3	10	2	4
P4	12	2	5
P5	13	2	5

Bu test problemlerinin optimum çözümüne ulaşmak için IBM ILOG programı CPLEX çözücüsünden yararlanılmıştır. Çizelge 5.13'te, önerilen GA yaklaşımının her problem için 5 deneme sonunda bulunduğu ortalama sonuçları ile CPLEX

çözücünün bulduğu optimum sonuçlar verilmiştir. Çizelge 5.14'te ise önerilen GA yaklaşımı ile CPLEX, çözüm süreleri açısından kıyaslanmıştır.

Çizelge 5.13 : Statik durumda önerilen GA yaklaşımı ortalama sonuçları ile CPLEX sonuçlarının kıyaslanması.

Problem	CPLEX Ortalama Akış Zamanı	Önerilen GA Ortalama Akış Zamanı
P1	40.24	40.25
P2	30	30.66
P3	25	25.2
P4	22.08	22.58
P5	22.23	22.47

Çizelge 5.14 : Statik durumda önerilen GA yaklaşımı ve CPLEX'in çözüm sürelerinin kıyaslanması.

Problem	CPLEX Çözüm Süresi (Saniye)	Önerilen GA Çözüm Süresi (Saniye)
P1	3.62	1.87
P2	6.02	2.45
P3	10.95	2.92
P4	229	3.86
P5	7523	4.27

P1, P2, P3, P4 ve P5 problemleri için önerilen GA yaklaşımının ürettiği sonuçlar ile CPLEX Solver'ın ürettiği optimum sonuçları kıyasladığımızda önerilen metodolojiye ait sonuçların optimum sonuçlardan ortalama %1 daha kötü olduğu görülmüştür. Her iki yaklaşım çözüm süreleri açısından kıyaslandığında CPLEX çözücünün çözüm süresinin problem boyutuna bağlı olarak üssel olarak arttığı görülmüştür. P5 problemi için önerilen GA yaklaşımı optimum sonucu 4.27 saniyede bulabilmekte CPLEX solver ise ancak 7523 saniyede sonuç üretebilmektedir. Buradan hareketle statik deterministik durumda önerilen GA yaklaşımının kabul edilebilir düzeyde iyi sonuçlara hızlı bir şekilde ulaştığı sonucuna varılmıştır. Bu aşamadan sonra önerilen GA yaklaşımının performansı dinamik ve deterministik ortamda test edilmiştir.

5.6.1 Dinamik deterministik durumda önerilen GA yaklaşımının performansının analizi

Dinamik durumda problem boyutları değişkenlik gösterebilmektedir. Melez akış tipi çizelgeleme problemine yönelik olarak optimum sonuç üretebilen CPLEX gibi çözümlerin çözüm süresi problem boyutlarına bağlı olarak üssel olarak artabilmektedir. Bu yüzden literatürde dinamik çizelgeleme için araştırmacıların sıkça başvurduğu ele alınan melez akış tipi çizelgeleme problemine uygun sezgisel yöntemler kıyaslama algoritmaları olarak tercih edilmiştir. Bu doğrultuda önerilen GA yaklaşımının performansı, alternatif sezgisel algoritma sonuçları ile kıyaslanarak analiz edilmiştir.

Ele alınan problemin dinamik yapısı gereği büyük problem boyutlarına ulaşabilmesi ve optimum sonuç veren algoritmaların büyük problemler için çok uzun sürede sonuç üretebilecek olması, kıyaslama algoritması olarak sezgisellerden yararlanmayı zorunlu kılmaktadır. Literatürde bir çok araştırmacının dinamik üretim sistemlerinde çizelgeleme için Tavlama Benzetim algoritmasından yararlandığı görülmektedir (Ruiz ve Vázquez-Rodríguez, 2010; Haouari ve M'Hallah, 1997; Jin ve diğ., 2006). Dinamik çizelgeleme problemleri için önerilen algoritmaların performanslarının liste çizelgeleme kuralları ile kıyaslanması, araştırmacıların sıkça başvurduğu bir diğer sezgisel yöntemdir. En Kısa İşlem Zamanı Önce liste çizelgeleme kuralı dinamik ortamlarda çizelgeleme çalışmalarında pek çok kere kıyaslama algoritması olarak kullanılmıştır (Bierwirth ve Mattfeld, 1999).

Bu tez çalışması kapsamında dinamik durum için önerilen GA yaklaşımının performansının, Tavlama Benzetim (TB) meta sezgiseli ve En Kısa İşlem Zamanı Önce (EKÖ) sezgiseli ile kıyaslanarak analiz edilmesi kararlaştırılmıştır.

Şekil 5.15'te kıyaslama algoritması olarak düşünülen TB algoritmasının sözde kodu verilmiştir.

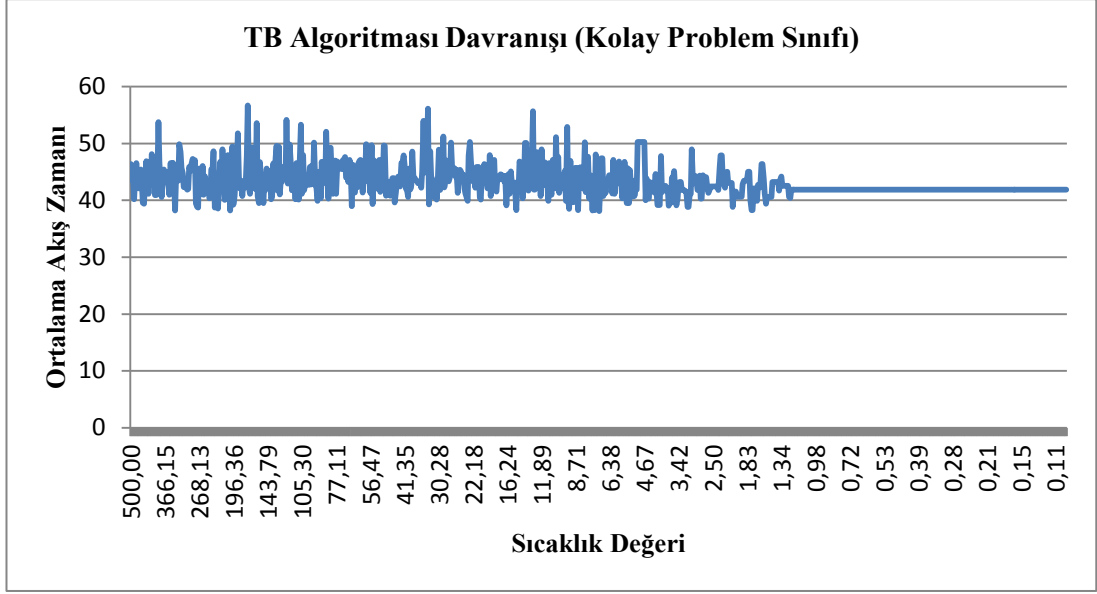
1. İlk çözümü üret ve en iyi çözüm (s_0) olarak belirle;
2. İlk sıcaklık değerini (T) belirle;
3. Sıcaklık indirgeme fonksiyonunu (α) belirle;
4. $T > 0,1$ olduğu sürece;
 - 4.1. Rastgele komşu bir çözüm (s) belirle
 - 4.2. $\delta = f(s) - f(s_0)$
 - 4.3. Eğer $\delta < 0$
 - 4.3.1. $s_0 = s$
 - 4.4. Eğer $\delta \geq 0$
 - 4.4.1. 0 ile 1 arasında bir rassal sayı üret (RS)
 - 4.4.2. Eğer $RS < \exp(-\delta/T)$
 - 4.4.2.1. $s_0 = s$
 - 4.5. $T = \alpha(T)$

Şekil 5.15 : Tavlama benzetim algoritması sözde kodu.

Kıyaslama algoritması olarak geliştirilen TB algoritmasına ilişkin olarak ilk sıcaklık değeri (T) ve sıcaklık indirgeme fonksiyonunun (α) belirlenmesi gerekmektedir. Ele alınan kolay, orta ve zor problem sınıfları göz önünde bulundurularak en iyi parametre değerlerine karar verilmeye çalışılmıştır. Sonuç olarak başlangıç sıcaklık değeri $T = 500$ ve sıcaklık indirgeme fonksiyonu:

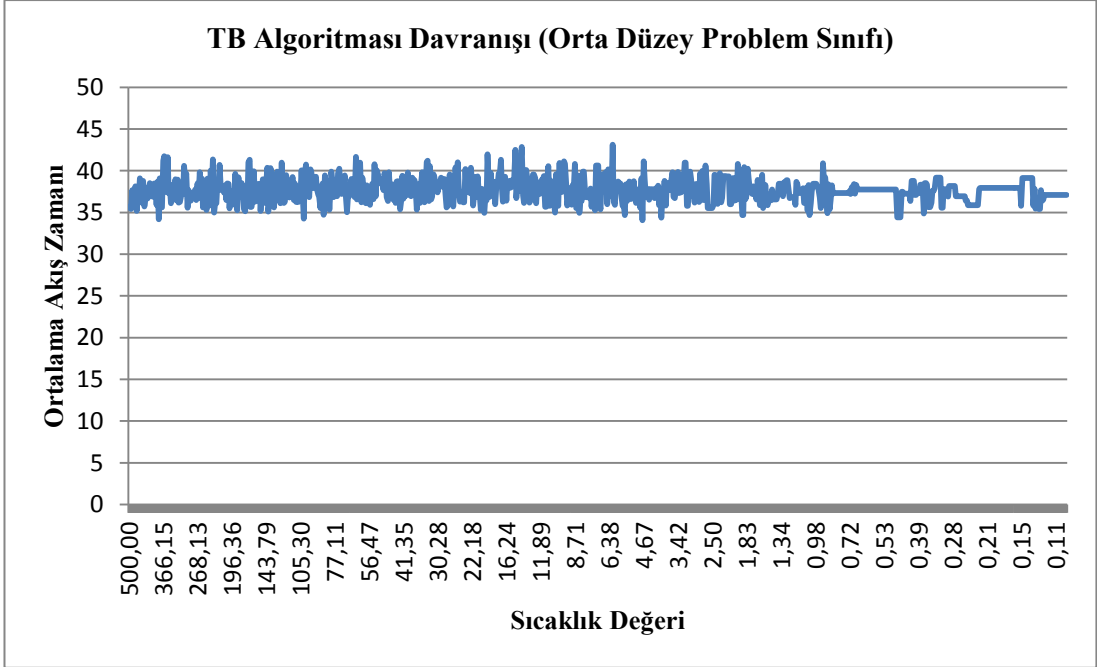
$$T = 0.99 * T \quad (5.2)$$

olarak belirlenmiştir. Bu doğrultuda belirlenen ilk sıcaklık değeri ve sıcaklık indirgeme fonksiyonuna göre TB algoritması kolay, orta ve zor problem sınıflarına uygulanarak algoritmanın davranışı gözlemlenmiştir. Kolay problem sınıfındaki bir probleme, sistemde herhangi bir anda ortalama 25 iş varken TB algoritması uygulandığında algoritmanın davranışı Şekil 5.16'daki gibi gözlemlenmiştir.



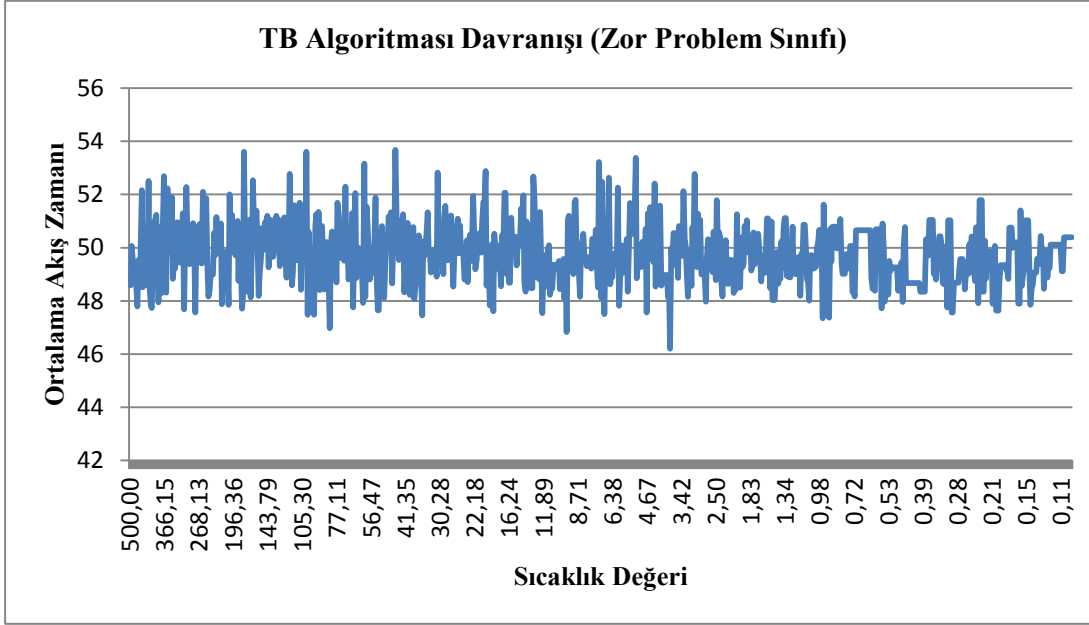
Şekil 5.16 : Tavlama benzetim algoritmasının kolay problem sınıfı üzerindeki davranışı.

Orta düzey problem sınıfındaki bir probleme, sistemde herhangi bir anda ortalama 50 iş varken TB algoritması uygulandığında algoritmanın davranışı Şekil 5.17'deki gibi gözlemlenmiştir.



Şekil 5.17 : Tavlama benzetim algoritmasının orta düzey problem sınıfı üzerindeki davranışı.

Zor problem sınıfındaki bir probleme, sistemde herhangi bir anda ortalama 125 iş varken TB algoritması uygulandığında algoritmanın davranışı Şekil 5.18'deki gibi gözlemlenmiştir.



Şekil 5.18 : Tavlama benzetim algoritmasının zor problem sınıfı üzerindeki davranışı.

Her üç durumda da TB algoritması belirli bir değere yakınsama eğilimi göstermektedir. En hızlı yakınsama durumu TB algoritması kolay problem sınıfına uygulandığında, en yavaş yakınsama durumu ise TB algoritması zor problem sınıfına uygulandığında gözlemlenmektedir. Kolay problem sınıfında iş geliş hızı ve çizelgelenen iş sayısı zor problem sınıfına göre daha düşük olduğu için, kolay problem sınıfının zor problem sınıfına göre daha hızlı yakınsaması beklenen bir durumdur. Bu doğrultuda seçilen ilk sıcaklık değeri ve sıcaklık indirgeme fonksiyonu ile geliştirilen TB algoritmasının tutarlı sonuçlar ürettiği söylenebilir. Şekil 5.19'da kıyaslama algoritması olarak düşünülen EKÖ algoritmasının sözde kodu verilmiştir.

1. Üretim sisteminde işlem görmeyi bekleyen işlerden bir iş vektörü oluştur.
2. İş vektöründeki işleri ortalama işlem zamanına göre küçükten büyüğe doğru sırala.

Şekil 5.19 : En kısa işlem zamanı önce algoritması sözde kodu.

Önerilen GA yaklaşımının, TB ve EKÖ algoritmalarının performanslarını dinamik ortamda kıyaslayabilmek için her üç metod için de bütün özellikleri aynı olan bir dinamik iş gelişi süreci simüle edilmiştir. Bölüm 5.1' deki çizelgeleme ortamı temel alınarak daha büyük problem boyutları ve farklı makine kullanım oranları için ilave alternatif test problemleri geliştirilmiştir. 1000 tane işin, her aşamada özdeş olmayan 6 paralel makinenin olduğu 5 aşamalı bir melez akış tipi üretim sisteminde 0.60, 0.65, 0.70, 0.75, 0.80, 0.85 ve 0.90 makine kullanım oranları (U) için çizelgelenmesi planlanmıştır. Başlangıçta ($t=0$ anında) çizelgelenmesi planlanan toplam işlerin %1'nin sistemde bulunduğu düşünülmüştür. Buna göre başlangıçta üretim sisteminde 10 işin bulunduğu farz edilir, geri kalan 990 iş ise zaman içinde sisteme gelecektir.

Dinamik deterministik ortama özgü olarak ortalama akış zamanının (\bar{F}) yanında toplam ağırlıklı tamamlanma zamanı ($\sum w_j C_j$) performans kriterine göre de önerilen GA yaklaşımının, TB ve EKÖ algoritmaları ile kıyaslanması planlanmıştır. Sistemdeki işlere ait ağırlık (w_j) değerleri, işlerin aşamalarındaki makinelerde planlanan işlem zamanlarının ortalaması alınarak üretilmiştir.

Bu doğrultuda önerilen GA yaklaşımı ve alternatif kıyaslama algoritmaları TB ve EKÖ, her çizelgeleme ortamı için 5 kere tekrarlanarak atölyede gerçekleşen ortalama akış zamanı ve toplam ağırlıklı tamamlanma zamanı sonuçlarının ortalaması bulunmuştur. Çizelge 5.15'te, önerilen GA yaklaşımının TB ve EKÖ algoritmalarının her çizelgeleme ortamı için 5 kere uygulanması sonucunda bulunan ortalama akış zamanı sonuçlarının ve toplam ağırlıklı tamamlanma zamanı sonuçlarının ortalama değerleri verilmiştir.

Çizelge 5.15 : Dinamik durumda önerilen GA yaklaşımı ortalama sonuçları ile alternatif algoritmaların ortalama sonuçlarının kıyaslanması.

U	Önerilen GA Ortalama \bar{F}	TB Ortalama \bar{F}	EKÖ Ortalama \bar{F}	Önerilen GA Ortalama $\sum w_j C_j$	TB Ortalama $\sum w_j C_j$	EKÖ Ortalama $\sum w_j C_j$
0.60	17.31	21.18	21.23	408407	409551	409577
0.65	17.55	21.20	21.31	377460	378597	378590
0.70	17.70	21.30	21.35	350925	351975	352029
0.75	17.73	21.24	21.26	327901	328947	328954
0.80	17.98	21.46	21.50	307796	308865	308871
0.85	18.20	21.43	21.54	290059	291066	291103
0.90	18.33	21.70	21.68	274315	275316	275321

Her iki performans kriteri, ortalama akış zamanı (\bar{F}) ve toplam ağırlıklı tamamlanma zamanı ($\sum w_j C_j$), için hesaplanan ortalama sonuçlar incelendiğinde, bütün makine kullanım oranları için önerilen GA yaklaşımının kıyaslama algoritmaları TB ve EKÖ algoritmalarından daha düşük ortalama sonuçlar bulduğu gözlemlenmiştir. Buna göre ele alınan dinamik deterministik melez akış sisteminde farklı iş geliş hızları için önerilen GA yaklaşımı bütün durumlarda kıyaslama algoritmalarından daha iyi sonuç vermektedir.

Çizelge 5.16 : Dinamik durumda önerilen GA yaklaşımı çözüm süreleri ile alternatif algoritmaların çözüm sürelerinin kıyaslanması.

Makine Kullanım Oranı (U)	Önerilen GA Yaklaşımı Çözüm Süresi (Saniye)	TB Çözüm Süresi (Saniye)	EKÖ Çözüm Süresi (Saniye)
0.60	2755.83	2735.83	2735.82
0.65	2543.05	2525.38	2525.38
0.70	2363.25	2344.99	2344.99
0.75	2207.02	2188.66	2188.65
0.80	2070.35	2051.86	2051.87
0.85	1949.72	1931.17	1931.16
0.90	1841.52	1823.88	1823.88

Çizelge 5.16’da verilen algoritma çözüm süreleri dikkate alındığında önerilen GA’nın, TB ve EKÖ algoritmalarında yaklaşık 20 saniye sonra çözüm verdiği görülmektedir. 1000 tane işin dinamik ortamda çizelgelendiği ve önerilen GA ile daha iyi çözümlere ulaşıldığı düşünüldüğünde, bu sürenin katlanılabilir bir süre olduğu rahatlıkla söylenilebilir. Sonuç olarak dinamik deterministik durumda önerilen GA ile bulunan ortalama sonuçlar, dinamik çizelgelemede bir çok araştırmacının faydalandığı TB meta sezgiseli ve dinamik çizelgelemede bir çok durumda kıyaslama algoritması olarak kullanılan EKÖ algoritmaları ile kıyaslanmış ve önerilen GA yaklaşımı açık bir şekilde daha iyi sonuç üretmiştir.

5.6.2 Dinamik stokastik durumda önerilen GA yaklaşımının performansının analizi

Dinamik deterministik durumda bir işin işleme başlamadan önce o aşamadaki makinelerde göreceği işlem zamanı sistem tarafından önceden bilinir ve bu değer 1 ile 19 arasında düzgün dağılıma $D(1,19)$ uyan bir değerdir. Deterministik durumda GA tarafından atölyeye önerilen bir çizelge, yine deterministik durum için o çizelgedeki işlerin önceden planlanan işlem zamanlarına göre gerçekleşir.

Dinamik stokastik durumda ise bir işin işlem zamanı önceden kesin olarak bilinmez, bir işe ait işlem zamanı ortalaması 10 olan 1 ile 19 arasındaki düzgün rassal değişkenlerden üretilir. Deterministik durumda, GA ile bir çözüm adayının uygunluk değerinin nasıl bulunduğu bölüm 4.2.3’te örnek bir durum için gösterilmiştir. İşlem görmesi planlanan işlerin işlem zamanları Çizelge 4.1’de görüldüğü üzere daha önceden bellidir. Buna göre deterministik durumda bir çözüm adayının uygunluk değeri önceden belli olan işlem zamanlarına göre bulunur ve bu işlem zamanları doğrultusunda atölyede gerçekleştirilerek nihai ortalama akış zamanı değerine ulaşılır.

Stokastik durumda ise işlem zamanları belirlenen dağılım gereğince 1 ile 19 arasındaki rassal değişkenlerdir. Bu yüzden stokastik durumda uygunluk değeri hesaplanırken direkt işlem zamanları kullanılmak yerine, işlem zamanlarının geldiği rassal değişkenler kullanılır ve her defasında bu rassal değişkenler ile işlem zamanları tekrardan üretilir. Buna göre bir çözüm adayına ait bir işin işlem zamanı herhangi bir aşamadaki bir makine için bir koşumdan diğer koşuma farklılık gösterebilir. Stokastik durumda belirlenen koşum sayısınca o çözüm adayı için

uygunluk deęerleri 1 ile 19 arasındaki düzgün rassal deęişkenlerden üretilerek her defasında yeniden hesaplanır ve bulunan uygunluk deęerlerinin ortalaması alınarak beklenen uygunluk deęeri bulunur.

İşlem zamanlarının önceden belli olmadığı stokastik durumda, önerilen GA yaklaşımının performansını deęerlendirebilmek için deterministik durum için önerilen iyi bir çözümün stokastik durum için de iyi bir çözüm olabileceęi varsayımından yola çıkılmıştır. Dinamik stokastik ortamda GA'nın bulduęu sonuçlar, aynı problem için dinamik deterministik durumda GA tarafından bulunan sonuçlar ile kıyaslanmış ve iki durum arasındaki farkın performans kriteri olarak belirlenmesine karar verilmiştir. Stokastik durumda GA tarafından atölyeye önerilen çizelgenin gerçekleşmesinde ise deterministik durumda bir iş için belirlenmiş işlem zamanları kullanılmasına karar verilmiştir.

Stokastik durumda GA atölyeye çizelge önerirken her defasında $D(1,19)$ düzgün rassal deęişkenlerine göre uygunluk deęerlerini ve en iyi çözümü yani çizelgeyi belirler. Bununla birlikte planlanan çizelgedeki işlem zamanları ile, stokastisitenin olduęu gerçek hayatta atölyede gerçekleşen işlem zamanları arasında farklılık olması büyük ihtimaldir. Bu doğrultuda önerilen GA ile en iyi çözüm yani çizelge üretilirken işlem zamanları için $D(1,19)$ arasındaki rassal deęişkenlerden yararlanılmış, atölyeye önerilen çizelgenin gerçekleşmesi ise deterministik durum için önceden belirlenmiş işlem zamanları kullanılmıştır. Sonuçta deterministik durum içinde bir işin işlem zamanı $D(1,19)$ arasındaki dağılıma uymaktadır, dolayısı ile stokastik durum için önerilen çizelgedeki işlem zamanları ile aynı dağılımdan gelmektedirler, bu anlamda herhangi bir tutarsızlık oluşmamaktadır. İlave olarak çizelgenin atölyede gerçekleşmesinde deterministik ve stokastik durum için aynı işlem zamanlarını kullanmak bize bir anlamda deterministik durumdaki sonuçları alt sınır olarak kullanma imkanı da tanımaktadır.

Bölüm 5.4.1'de önerilen test problemleri ele alınmıştır. Ele alınan problemler için, dinamik deterministik durumda önerilen GA'nın farklı makine kullanım oranlarında 1000 işi çizelgelemesi sonucu atölyede bulunan ortalama akış zamanı sonuçları alt sınır olarak belirlenmiştir. Daha sonra dinamik stokastik durum için işlem zamanları 1 ile 19 arasındaki rassal deęişkenlerden üretilmiş 1000 iş, her aşamada 6 özdeş olmayan paralel makinenin bulunduęu 5 aşamalı bir üretim sisteminde belirlenen makine kullanım oranlarının her biri için 5 kere uygulanarak ortalama akış zamanı

sonuçları bulunmuştur. Çizelge 5.17’de dinamik stokastik durumda, önerilen GA ile bulunan ortalama akış zamanı sonuçları verilmiştir. Çizelge 5.18’de ise dinamik stokastik durumda önerilen GA’nın çözüm üretme süreleri verilmiştir.

Çizelge 5.17 : Dinamik stokastik durumda önerilen GA yaklaşımı ortalama sonuçları.

Makine Kullanım Oranı	Önerilen GA Yaklaşımı Ortalama Akış Zamanı
0.60	29.45
0.65	31.65
0.70	33.11
0.75	35.68
0.80	38.41
0.85	46.98
0.90	53.62

Çizelge 5.18 : Dinamik stokastik durumda önerilen GA yaklaşımı çözüm süreleri.

Makine Kullanım Oranı	Önerilen GA Yaklaşımı Çözüm Süresi (Saniye)
0.60	2761.67
0.65	2558.93
0.70	2372.71
0.75	2219.97
0.80	2085.72
0.85	1972.88
0.90	1877.43

Dinamik stokastik durumda önerilen GA ile bulunan sonuçlar ile dinamik deterministik durumda önerilen GA ile bulunan sonuçlar Çizelge 5.19’da kıyaslanmıştır.

Çizelge 5.19 : Dinamik stokastik durumda önerilen GA yaklaşımı ortalama sonuçları ile dinamik deterministik durumda önerilen GA yaklaşımı ortalama sonuçlarının kıyaslanması.

Makine Kullanım Oranı	Stokastik Durumda Önerilen GA Yaklaşımı Ortalama Akış Zamanı	Deterministik Durumda Önerilen GA Yaklaşımı Ortalama Akış Zamanı	Oransal Ortalama Uzaklık (%)
0.60	29.45	17.31	70.13
0.65	31.65	17.55	80.34
0.70	33.11	17.70	87.06
0.75	35.68	17.73	101.24
0.80	38.41	17.98	113.63
0.85	46.98	18.20	158.13
0.90	53.62	18.33	192.53

Çizelge 5.19'daki stokastik durum ve deterministik durum için bulunan sonuçlar incelendiğinde, deterministik durum için atölyede gerçekleşen ortalama akış zamanı sonuçlarının bütün makine kullanım oranları için stokastik durum için bulunan sonuçlardan daha iyi olduğu görülmektedir. Deterministik durumda atölyede gerçekleşen işlem zamanları, GA tarafından önerilen çizelgedeki işlem zamanları ile aynı olduğu için deterministik durumdaki sonuçlar stokastik durumdaki sonuçlardan daha iyidir.

Stokastik durumda, atölyeye çizelge önerilirken çizelgedeki bir işe ait işlem zamanı önceden bilinemediğinden atölyede gerçekleşen sonuçlar ile önerilen çizelgedeki sonuçlar arasında sapma olması çok büyük ihtimaldir. Buna göre önerilen GA tarafından deterministik durum için hesaplanan ortalama akış zamanı sonuçlarının stokastik durum için hesaplanan ortalama akış zamanı sonuçlarından daha iyi olması beklenen bir durumdur.

Literatürde stokastikliğin olduğu dinamik ortamlardaki GA uygulamalarında koşum sayısı ve popülasyon büyüklüğünün beraber analiz edildiği görülmektedir. (Arnold ve Beyer, 2000, 2001). Büyük popülasyonlarla çalışan evrimsel algoritmalar arama sürecinin başında daha fazla olasılığı bir arada değerlendirme avantajı ile daha hızlı bir şekilde global optimuma yaklaşabilmekte, bununla birlikte koşum sayısının artırılması ile popülasyon yakınsadıktan sonra yerel optimuma takılma önlenmektedir.

Bu tez çalışması kapsamında dinamik stokastik durumda popülasyon büyüklüğü: 100, koşum sayısı: 20 olarak belirlenmiştir. Bir popülasyondaki toplam uygunluk hesaplama bütçesi $100 \times 20 = 2000$ 'dir. Toplam uygunluk hesaplama bütçesi sabit tutularak, popülasyon büyüklüğü ve koşum sayısı için alternatif durumların olduğu bir test düzeneği oluşturulabilir. Burada amaçlanan dinamik stokastik durumda GA tarafından önerilen sonuçlar ile dinamik deterministik durumda GA tarafından önerilen sonuçlar arasındaki uzaklığı azaltabilecek daha iyi bir kombinasyonun bulunmasıdır. Buna göre koşum sayıları 20, 10, 5 ve 1 olarak değişen, popülasyon büyüklükleri ise 100, 200, 400 ve 2000 arasında değişebilen bir deney düzeneği oluşturulmuştur. Her bir kombinasyonun, bir jenerasyondaki toplam uygunluk hesaplama sayısı birbirine eşittir ($2000 \times 1 = 400 \times 5 = 200 \times 10 = 100 \times 20$). Bu doğrultuda bütün kombinasyon alternatifleri için ele alınan dinamik stokastik melez akış tipi çizelgeleme problemi 1000 iş için atölyede gerçekleşmiş ve ortalama akış zamanı sonuçları Çizelge 5.20'de ve çözüm süreleri Çizelge 5.21'de verilmiştir.

Çizelge 5.20 : Dinamik stokastik durumda önerilen GA ile alternatif popülasyon büyüklükleri ve koşum sayıları hesaplanan ortalama akış zamanı değerleri ortalama sonuçları.

Popülasyon Büyüküğü X Koşum Sayısı	Makine Kullanım Oranı						
	0.6	0.65	0.7	0.75	0.8	0.85	0.9
2000X1	21.42	23.53	24.54	28.13	32.66	38.49	50.93
400X5	27.36	28.42	30.80	33.19	37.15	42.45	49.35
200X10	29.05	39.56	32.19	34.33	38.86	44.53	50.97
100X20	29.45	31.65	33.11	35.68	38.41	46.98	53.62
En İyi Ortalama Sonuç	21.42	23.53	24.54	28.13	32.66	38.49	49.35

Çizelge 5.21 : Dinamik stokastik durumda önerilen GA'nın alternatif popülasyon büyüklükleri ve koşum sayıları için çözüm süreleri (saniye).

Popülasyon Büyüküğü X	Makine Kullanım Oranı							
	Koşum Sayısı	0.6	0.65	0.7	0.75	0.8	0.85	0.9
2000X1		2754.34	2547.07	2369.13	2212.35	2075.16	1964.47	1862.12
400X5		2759.95	2554.93	2371.37	2215.85	2080.65	1965.86	1866.88
200X10		2756.86	2552.42	2375.12	2216.57	2090.32	1977.04	1875.17
100X20		2761.67	2558.93	2372.71	2219.97	2085.72	1972.88	1877.43

Çizelge 5.22'de ise alternatif popülasyon büyüklükleri ve koşum sayıları için bulunan ortalama akış zamanı sonuçlarının en iyi ortalama sonuçlara oransal uzaklıkları verilmiştir.

Çizelge 5.22 : Alternatif popülasyon büyüklükleri ve koşum sayıları için hesaplanan değerlerin en iyi ortalama sonuçlara oransal uzaklıkları.

Popülasyon Büyüküğü X	Makine Kullanım Oranı							Oransal Ortalama Uzaklık	
	Koşum Sayısı	0.6	0.65	0.7	0.75	0.8	0.85		0.9
2000X1		%0	%0	%0	%0	%0	%0	%3.20	%0.46
400X5		%27.73	%20.78	%25.51	%17.99	%13.75	%10.29	%0	%16.58
200X10		%35.62	%68.13	%31.17	%22.04	%18.98	%15.69	%3.28	%27.85
100X20		%37.49	%34.51	%34.92	%26.84	%17.61	%22.06	%8.65	%26.01

Çizelge 5.22'de görüldüğü üzere popülasyon büyüklüğü 2000 ve koşum sayısı 1 alternatifi için en düşük oransal ortalama uzaklık değeri %0.46 elde edilmiştir. Çizelge 5.21'de verilen algoritma çözüm süreleri incelendiğinde popülasyon büyüklüğü 2000 ve koşum sayısı 1 durumu için çözüm sürelerinin popülasyon büyüklüğü 100 ve koşum sayısı 20 durumundan daha iyi olduğu gözlemlenmiştir. GA'nın başlangıçtaki popülasyon büyüklüğünü 100'den 2000'e çıkarmak ve koşum sayısını 20'den 1'e düşürmek algoritmaya ek herhangi bir yük getirmezken daha iyi ortalama akış zamanı sonuçlarına ulaşılmasını sağlamıştır.

Dinamik stokastik durumda önerilen GA için popülasyon büyüklüğü 2000 ve koşum sayısı 1 seçildiğinde elde edilen sonuçlar ile dinamik deterministik durumdaki ortalama akış zamanı sonuçları Çizelge 5.23'te kıyaslanmıştır.

Çizelge 5.23 : Popülasyon büyüklüğü 2000 ve koşum sayısı 1 için stokastik durumda önerilen GA yaklaşımı ortalama sonuçları ile deterministik durumda önerilen GA yaklaşımı ortalama sonuçlarının kıyaslanması.

Makine Kullanım Oranı	Stokastik Durumda Önerilen GA Yaklaşımı Ortalama Akış Zamanı	Deterministik Durumda Önerilen GA Yaklaşımı Ortalama Akış Zamanı	Oransal Ortalama Uzaklık (%)
0.60	21.42	17.31	23.74
0.65	23.53	17.55	34.07
0.70	24.54	17.70	38.64
0.75	28.13	17.73	58.66
0.80	32.66	17.98	81.65
0.85	38.49	18.20	111.48
0.90	50.93	18.33	177.85

Makine kullanım oranı arttıkça iş geliş hızı da artmaktadır, dolayısı ile sistemdeki dinamiklik de artmaktadır. 0.60, 0.65 ve 0.70 makine kullanım oranları için stokastik durumda önerilen sonuçlar ile deterministik durumda önerilen sonuçlar arasındaki fark kabul edilebilir bir seviyededir. Bu doğrultuda sistemdeki dinamikliğin daha düşük olduğu durumlar için deterministik durumdaki iyi bir çözümün stokastik durum için de iyi bir çözüm olduğu söylenebilir. Buna göre popülasyon büyüklüğü 100 ve koşum sayısı 20 durumu için 0.60, 0.65 ve 0.70 makine kullanım oranlarının olduğu çizelgeleme ortamlarında dinamik deterministik durum ile dinamik stokastik durum arasındaki ortalama uzaklık değerleri sırası ile %70.13, %80.34 ve %87.06 olarak bulunmuştur. Popülasyon büyüklüğü 2000 koşum sayısı 1 durumu için aynı makine kullanım oranlarında ortalama uzaklık değerleri %23.74, %34.07 ve %38.64 olarak bulunmuştur. Popülasyon büyüklüğünün 100'den 2000'e çıkarılması ve koşum sayısının 20'den 1'e düşürülmesi sonucunda stokastik durumda bulunan ortalama akış zamanı sonuçlarının deterministik durumdaki ortalama akış zamanı sonuçlarına oransal ortalama uzaklıklarında kayda değer bir iyileşme gözlemlenmiştir.

Sonuç olarak bu çalışma kapsamında ele alınan dinamik stokastik melez akış tipi çizelgeleme problemine önerilen GA yaklaşımında, 2000 çözüm adayından oluşan bir popülasyon ile evrimsel hesaplamaları gerçekleştirmek ve koşum sayısını mümkün olduğunca küçük tutarak uygunluk değerlerini belirlemek, daha küçük bir

popülasyonla çok koşum yapmaya göre daha iyi ortalama sonuçlar bulunmasını sağlamıştır.

6. SONUÇ VE ÖNERİLER

Bu çalışmada, sürekli olarak çalışan bir üretim sisteminin dinamik olarak çizelgelenebilmesi için bir yaklaşım geliştirilmesi amaçlanmıştır. Bu amaçla özdeş olmayan paralel makinelerden oluşan melez akış tipi bir üretim sistemi ele alınmıştır. Dinamik ve stokastik ortamda melez akış tipi çizelgeleme problemine tepkisel olarak çözüm üretebilecek bir algoritma geliştirilmesi hedeflenmiştir.

Literatürdeki dinamik çizelgeleme çalışmaları incelendiğinde basit liste çizelgeleme kuralları ile ya da daha kompleks algoritmalarla çizelgeleme yapılabilmektedir. Çizelge üretimi için geçen süre çoğu zaman göz ardı edilmekte, dinamik problem statik alt problemlere bölünerek çözülmektedir. Bu çalışmada önerilen algoritmanın sürekli değişen bir üretim sisteminde dinamik olarak çizelgeleme yapabilmesi amaçlanmıştır.

Öncelikle melez akış tipi çizelgeleme probleminin statik deterministik durumu için alternatif çözüm algoritmaları incelenmiştir. Daha sonra dinamik durumda büyük problem boyutlarında da iyi çözümleri kabul edilebilir sürede üretebilecek çözüm yaklaşımları analiz edilmiştir. Bu doğrultuda melez akış tipi çizelgeleme problemi için bir genetik algoritma geliştirilmesine karar verilmiştir. GA'nın çeşitli problem boyutları için statik durumdaki çözüm kalitesi test edildikten sonra genetik algoritma popülasyonlarının nasıl değişen problem koşullarına entegre edilebileceği araştırılmıştır. Değişkenliğin olduğu ana kadar ki arama süreci kazanımlarını değişkenlik sonrası ortama aktarabilen değişken kromozom yapılı bir yaklaşımla algoritmanın çok hızlı bir şekilde yeni durum için çizelge üretebildiği görülmüştür.

Olay tabanlı bir yeniden çizelgeleme yaklaşımı temel alınmıştır. Buna göre sisteme yeni iş gelişi ya da üretim aşamalarındaki herhangi bir işin o aşamadaki işleminin tamamlanması yeniden çizelgeleme noktası olarak seçilmiştir. Algoritmanın temel hedefi olan sürekli olarak çalışan bir üretim sistemine dinamik olarak çizelge üretebilmek için GA dinamik çizelgeleme sürecine entegre edilmiştir. Bu doğrultuda klasik GA yaklaşımından bağımsız olarak problem değişimi sonrası o anki mevcut

popülasyon yeni probleme deęişken kromozom yaklaşımı ile uyarlanmış ayrıca GA'nın evrilmesi için beklenmemiştir. Bunun yerine arama süreci kazanımlarını deęişkenlik sonrası güncellenen ortama aktarabilen küçük ama sürekli iyileştirmelere güvenen bir yaklaşım benimsenmiştir.

Dinamik deterministik duruma yönelik olarak GA'nın uyarlanması gerçekleştirildikten sonra, dinamik stokastik ortamlarla nasıl başa çıkılacağı analiz edilmiştir. Dinamik stokastik ortamın GA popülasyonlarına yansıtılabilmesi için çözüm adaylarının uygunluk hesaplama işleminin simülasyon yolu ile yapılmasına karar verilmiştir.

GA'nın çözüm üretme için harcadığı sürenin atölyedeki etkisini gözlemleyebilmek için algoritmanın çalıştırıldığı ortamın sistem saatini temel alan bir çözüm üretme ve çizelge gerçekleştirme süreci tasarlanmıştır. Sisteme yeni işlerin gelmesi, herhangi bir aşamada tamamlanan bir işin bir sonraki aşamaya transfer edilmesi, GA'nın çözüm bulma için geçirdiğinin sürenin sisteme yansıtılarak önerilen çizelgenin atölyede gerçekleştirilmesi sağlanmıştır.

Ele alınan dinamik melez akış tipi çizelgeleme problemine yönelik olarak GA strateji seçimi ve parametre optimizasyonu gerçekleştirilerek algoritmanın çözüm kalitesi iyileştirilmiştir. Dinamik deterministik ortamda geliştirilen GA'nın performansını analiz edebilmek için TB ve EKÖ algoritmaları kıyaslama algoritması olarak kullanılmıştır. Öncelikle TB algoritmasında parametre optimizasyonu gerçekleştirilmiştir. Daha sonra hem TB algoritmasının hem de EKÖ algoritmasının GA için tasarlanan çözüm üretme ve çizelge gerçekleştirme sistemine entegrasyonu gerçekleştirilmiştir. Aynı çizelgeleme ortamında üç algoritma çözüm kalitesi ve çözüm süresi açısından kıyaslanmıştır. Önerilen GA, TB ve EKÖ algoritmalarına göre daha iyi sonuç vermiştir.

Dinamik stokastik ortam için önerilen GA'nın, uygunluk hesaplama simülasyon modülü koşum sayısı incelenmiştir. %5 hata değeri alt sınır alınarak çeşitli boyutlardaki test problemlerinde seçilen koşum sayıları için yarı güven aralığı değerleri analiz edilmiştir. Dinamik stokastik ortamdaki GA'nın performansını analiz edebilmek için bir analiz prosedürü öne sürülmüştür. Buna göre dinamik deterministik ortam için iyi bir çözüm adayının dinamik stokastik ortam için de iyi bir çözüm adayı olabileceği varsayımında bulunulmuştur. Dinamik stokastik durum

ile dinamik deterministik durum için önerilen GA çözümleri arasındaki fark performans kriteri olarak kabul edilmiştir. Alternatif makine kullanım oranları için dinamik stokastik durum ve dinamik deterministik durum kıyaslanmıştır.

Dinamik stokastik durumda GA tarafından önerilen çözümlerin kalitesini artırabilmek için herhangi bir nesildeki toplam uygunluk hesaplama bütçesini değiştirmeden alternatif popülasyon büyüklükleri ve koşum sayıları için elde edilen sonuçlar analiz edilmiştir. Koşum sayısının mümkün olduğunca küçük tutulduğu büyük popülasyonlu bir GA, ele alınan dinamik stokastik melez akış tipi çizelgeleme problemine yönelik olarak daha iyi çözümler üretmiştir. Ayrıca iş geliş hızının daha düşük olduğu, sistemdeki dinamikliğin kabul edilebilir bir seviyede bulunduğu çizelgeleme ortamları için dinamik deterministik durumdaki bir çözüm adayının dinamik stokastik durum için de iyi bir çözüm olabileceği varsayımı doğrulanmıştır.

Sonuç olarak sürekli olarak çalışan değişkenliğin olduğu melez akış tipi üretim ortamında dinamik olarak çizelgeleme yapabilen tepkisel bir algoritma geliştirilmiştir. Geliştirilen algoritma çeşitli test problemleri ile statik, dinamik ve stokastik üretim ortamlarında alternatif yöntemler ile kıyaslanarak analiz edilmiştir. Önerilen GA yaklaşımının hızlı ve efektif bir şekilde dinamik çizelgeleme yapabildiği gösterilmiştir.

Gelecek araştırmalarda dinamik bir üretim sisteminin çizelgelenmesinde stabilitenin ve sağlamlığın bir arada ele alındığı bir amaç fonksiyonu önerilebilir. Dinamik stokastik durumda değişen problemin yapısına uygun olarak popülasyon büyüklüğü ve koşum sayısının anlık olarak uyarlanabildiği bir çalışma yapılabilir. Stokastik durumda bir sonraki neslin belirlenmesinde bireylerin sadece uygunluk değerlerine göre değil sistemde geçirdikleri süreye göre bir sonraki popülasyon oluşturulabilir. Dinamik çizelgeleme sürecinde meydana gelen anlık değişimlere, ele alınan problemin karakterine uygun olarak basit ama zeki bir sezgisel metot entegre edilerek anlık değişimlere anlık cevaplar veren bir çalışma yapılabilir.

KAYNAKLAR

- Akrami, B., Karimi, B. ve Hosseini, S. M. M.** (2006). Two metaheuristic methods for the common cycle economic lot sizing and scheduling in flexible flow shops with limited intermediate buffers: The finite horizon case. *Applied Mathematics and Computation*, 183, 634-645.
- Alaykyran, K., Engin, O. ve Doyen, A.** (2007). Using ant colony optimization to solve hybrid flow shop scheduling problems. *The International Journal of Advance Manufacturing Technology*, 35, 541-550.
- Al-Fawzan., M. A. ve Haouari, M.** (2005). A bi-objective modal for robust resource constrained project scheduling. *International Journal of Production Economics*, 96, 175-187.
- Al-Hinai, N. ve ElMekawy, T. Y.** (2011). Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *International Journal of Production Economics*, 132, 279-291.
- Allaoui, H. ve Artiba, A.** (2006). Scheduling two-stage hybrid flow shop with availability constraints. *Computers & Operations Research*, 33, 1399-1419.
- Alisantoso, D., Khoo L. P. ve Jiang P. Y.** (2003). An immune algorithm approach to the scheduling of a flexible PCB flow shop. *The International Journal of Advance Manufacturing Technology*, 22, 819-827.
- Allahverdi, A. ve Al-Anzi, F. S.** (2006). Scheduling multi-stage parallel-processor services to minimize average response time. *Journal of the Operational Research Society*, 57, 101-110.
- Almeder, C. ve Hartl, R.** (2013). A metaheuristic optimization approach for a real-world stochastic flexible flow shop problem with limited buffer. *International Journal of Production Economics*, 145, 88-95.
- Arnold, D. ve Beyer, H.** (2000). Efficiency and mutation strength adaptation of the $(\mu/\mu, \lambda)$ -ES in a noisy environment. *Lecture Notes in Computer Science*, 1917, 39-48.
- Arnold, D. ve Beyer, H.** (2001). Local performance of the $(\mu/\mu, \lambda)$ -ES in a noisy environment. *Foundations of Genetic Algorithms*, 6, 127-141.
- Artigues, C., Billaut, J. C., ve Esswein, C.** (2005). Maximization of solution flexibility for robust shop scheduling. *European Journal of Operational Research*, 165, 314-328.
- Banks, J., Carson, J. S., Nelson, B. L. ve Nicol, D. M.** (2010). *Discrete-Event System Simulation*. Pearson Education, New Jersey.

- Bean, J. C., Birge, J., Mintenhal, J. ve Noon, C.** (1991). Matchup scheduling with multiple resources, release dates, and disruptions. *Operations Research*, 39, 470-483.
- 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*, 2006, 1-17.
- Bierwirth, C., Kopfer, H., Mattfeld, D. C., ve Rixen, I.** (1995). Genetic algorithm based scheduling in a dynamic manufacturing environment. Proceedings of the Second Conference on Evolutionary Computation.
- Bierwirth, C. ve Mattfeld, D.** (1999). Production scheduling and rescheduling with genetic algorithms. *Evolutionary Computation*, 7, 1-18.
- Branke, J.** (2002). *Evolutionary Optimization In Dynamic Environments*. University of Karlsruhe.
- Brockmann, K., Dangelmaier, W. ve Holthöfer, N.** (1997). Parallel branch & bound algorithm for makespan optimal scheduling in flow shops with multiple processor. *Operations Research Proceedings 1997, 1998*, 428-433.
- Cheng, J., Yoshiyuki, K. ve Kise, H.** (2001). A shifting bottleneck approach for parallel-machine flowshop scheduling problem. *Journal of the Operation Research Society of Japan*, 44, 141-154.
- Carlier, J. ve Neron, E.** (2000). An exact method for solving the multi-processor flow- shop. *RAIRO - Operations Research*, 34, 1-25.
- Church, L. ve Uzsoy, R.** (1992). Analysis of periodic and event driven rescheduling policies in dynamic shops. *International Journal of Computer Integrated Manufacturing*, 5, 153-163.
- Daniels, R. ve Kouvelis, P.** (1995). Robust scheduling to hedge against processing time uncertainty in single stage production. *Management Science*, 41, 363-376.
- Dessouky, M. M., Dessouky, M. I. ve Verma S. K.** (1998). Flowshop scheduling with identical jobs and uniform parallel machines. *European Journal of Operational Research*, 109, 620-631.
- Duenas, A. ve Petrovic, D.** (2007). An approach to predictive-reactive scheduling of parallel machines subject to disruptions. *Annals of Operations Research*, 159, 65-82.
- Dutta, A.** (1994). Reacting to scheduling exceptions in FMS environments. *IIE Transactions*, 22, 300-314.
- Engin, O. ve Doyen A.** (2004). A new approach to solve hybrid flow shop scheduling problems by artificial immune system. *Future Generation Computer Systems*, 20, 1083-1095.
- Eroğlu, D.** (2014). *İşlerin bölünerek çizelgelenmesi için geliştirilen genetik algoritma ve uygulama*, (doktora tezi), Uludağ Üniversitesi, Bursa.
- Fox, M. S. and Smith, S. F.** (1984). ISIS-knowledge based system for factory scheduling. *Expert Systems*, 1, 25-49.

- Garcia, J. M. ve Lozano, S.** (2005). Production and delivery scheduling problem with time windows. *Computers & Industrial Engineering*, 48, 733-742.
- Gen, M. ve Cheng, R.** (2000). *Genetic Algorithm and Engineering Optimization*. John Wiley & Sons, New York, NY.
- Gholami, M. ve Zandieh, M.** (2009). Integrating simulation and genetic algorithm to schedule a dynamic flexible job shop. *Journal of Intelligent Manufacturing*, 20, 481-498.
- Gomes, M. C., Barbosa-Póvoa, A. P. ve Novais, A. Q.** (2006). Optimal reactive scheduling of multipurpose, make-to-order industries. *Computer Aided Chemical Engineering*, 21, 1587-1592.
- Gomes, M. C., Barbosa-Póvoa, A. P. ve Novais, A. Q.** (2010). A discrete time reactive scheduling model for new order insertion in job shop, make-to-order industries. *International Journal of Production Research*, 48, 7395-7422.
- Guinet, A. ve Solomon, M. M.** (1996). Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time. *International Journal of Production Research*, 34, 1643-1654.
- Guinet, A., Solomon, M. M., Kedia P. K. ve Dussauchoy A.** (1996). A computational study of heuristics for two-stage flexible flowshops. *International Journal of Production Research*, 34, 1399-1415.
- Guirchoun, S., Martineau, P. ve Billaut, J.** (2005). Total completion time minimization in a computer system with a server and two parallel processors. *Computers & Operations Research*, 32, 599-611.
- Guo, B., ve Nonaka, Y.** (1999). Rescheduling and optimization of schedules considering machine failures. *International Journal of Production Economics*, 60-61, 503-513.
- Gupta, J. N. D.** (1988). Two-stage, hybrid flow shop scheduling problem. *Journal of the Operational Research Society*, 39, 359-364.
- Gupta, J. N. D., Hariri, A. M. A. ve Potts C. N.** (1997). Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. *Annals of Operations Research*, 69, 171-191.
- Gupta, J. N. D. ve Tunc, E. A.** (1998). Minimizing tardy jobs in a two-stage hybrid flowshop. *International Journal of Production Research*, 36, 2397-2417.
- Gupta, J. N. D., Sexton, R. S. ve Tunc, E. A.** (2000). Selecting scheduling heuristics using neural networks. *Inform Journal on Computing*, 12, 150-162.
- Gupta, J. N. D., Krüger, K., Lauff, V., Werner, F. ve Sotskov, Y. N.** (2002). Heuristics for hybrid flow shops with controllable processing times and assignable due dates. *Computers & Operations Research*, 29, 1417-1439.

- Güçlü, İ.** (2006). *A genetic algorithm approach to the tool allocation problem in flexible manufacturing systems*, (yüksek lisans tezi), Boğaziçi Üniversitesi, İstanbul.
- Gürel, S., Körpeoğlu, E., ve Aktürk, M. S.** (2010). An anticipative scheduling approach with controllable processing times. *Computers & Operations Research*, 37, 1002-1013.
- Haouari, M. ve M'Hallah, R.** (1997). Heuristic algorithms for the two-stage hybrid flowshop problem. *Operations Research Letters*, 21, 43-53
- Haouari, M., Hidri, L. ve Gharbi, A.** (2006). Optimal scheduling of a two-stage hybrid flow shop. *Mathematical Methods of Operations Research*, 64, 107-124.
- He, L. M., Sun S. J. ve Luo R. Z.** (2007). A hybrid two-stage flowshop scheduling problem. *Asia-Pacific Journal of Operational Research*, 24, 45-56.
- He, Y., Smith, M. ve Dudek, R.** (1994). Effects of inaccuracy of processing time estimation on effectiveness of dispatch rules, *Third Industrial Engineering Research Conference* (Sf. 308-313).
- Holland, J.** (1975). Adaptation in natural and artificial systems. *University of Michigan Press*.
- Hoogeveen, J. A., Lenstra J. K. ve Veltman, B.** (1996). Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. *European Journal of Operational Research*, 89, 172-175.
- Horng, S. C., Lin, S. S. ve Yang, F. Y.** (2012). Evolutionary algorithm for stochastic job shop scheduling random processing time. *Expert Systems with Applications*, 39, 3603-3610.
- Janiak, A., Kozan, E., Lichtenstein, M. ve Oğuz, C.** (2007). Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion. *International Journal of Production Economics*, 105, 407-424.
- Jeong, K. J. ve Kim, Y. D.** (1998). A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules. *International Journal of Production Research*, 36, 2609-2626.
- Jin, Z. H., Ohno, K., Ito, T. ve Elmaghraby, S. E.** (2002). Scheduling hybrid flowshops in printed circuit board assembly lines. *Production and Operations Management*, 11, 216-230.
- Jin, Z. H., Yang Z. ve Ito, T.** (2006). Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem. *International Journal of Production Economics*, 100, 322-334.
- Juan, A. A., Barrios, B. B., Vallada, E., Daniel, R. ve Jorba, J.** (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory*, 46, 101-117.
- Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P. ve Werner. F.** (2008). Algorithms for flexible flow shop problems with unrelated parallel

- machines, setup times, and dual criteria. *The International Journal of Advanced Manufacturing Technology*, 37, 354-370.
- Kainfar, K., Fatemi, G., ve Oroojlooy, J.** (2012). Study of stochastic sequence-dependent flexible flow shop via developing a dispatching rule and a hybrid GA. *Engineering Applications of Artificial Intelligence*, 25, 494-506.
- Kasperski, A.** (2005). Minimizing maximal regret in the single machine sequencing problem with maximum lateness criterion. *Operations Research Letters*, 33, 431-436.
- Kato, E. R. R., Morandin, O. ve Fonseca, M. A. S.** (2009). Ant colony optimization algorithm for reactive production scheduling problem in the job shop system, *IEEE International Conference on Systems, Man and Cybernetics* (Sf. 2199-2204).
- Kevran, M. M.** (2009). *Çoklu sensör konumlandırma probleminin genetik algoritmalar ile çözülmesi*, (yüksek lisans tezi), Hava Harp Okulu, İstanbul.
- Kim, J. S., Kang, S. H. ve Lee, S. M.** (1997). Transfer batch scheduling for a two-stage flowshop with identical parallel machines at each stage. *Omega*, 25, 547-555.
- Kim, M. ve Kim, Y.** (1994). Simulation based real-time scheduling in a flexible manufacturing system. *Journal of Manufacturing System*, 13, 85-93.
- Kurz, M. E. ve Askin, R. G.** (2003). Comparing scheduling rules for flexible flow lines. *International Journal of Production Economics*, 85, 371-388.
- Kurz, M. E. ve Askin, R. G.** (2004). Scheduling flexible flow lines with sequence-dependent setup times. *European Journal of Operational Research*, 159, 66-82.
- Kutanoglu, E. ve Sabuncuoglu, I.** (2001a). Experimental investigation of iterative simulation-based scheduling in dynamic and stochastic job shop. *Journal of Manufacturing Systems*, 20, 264-278.
- Kutanoglu, E. ve Sabuncuoglu, I.** (2001b). Routing-based reactive scheduling policies for machine failures in job shops. *International Journal of Production Research*, 39, 3141-3158.
- Lawrence, S. ve Sewell, E.** (1997). Heuristic, optimal, static, and dynamic schedules when processing times are uncertain. *Journal of Operations Management*, 15, 71-82.
- Lejmi, T. ve Sabuncuoglu, I.** (2002). Effect of load and processing time variation on the effectiveness of scheduling rules. *International Journal of Production Research*, 40, 945-974.
- Leon, V. J., Wu, S. D. ve Stroer, R. H.** (1994). Robustness measures and robust scheduling for job shops. *IIE Transactions*, 26, 32-43.
- Li, S. L.** (1997). A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. *European Journal of Operational Research*, 102, 142-156.

- Logendran, R., deSzoek, P. ve Barnard, F.** (2006). Sequence-dependent group scheduling problems in flexible flow shops. *International Journal of Production Economics*, 102, 66-86.
- Lou, P., Liu, Q., Zhou, Z., Wang, H. ve Sun, S. X.** (2011). Multi-agent-based proactive–reactive scheduling for a job shop. *The International Journal of Advanced Manufacturing Technology*, 59, 311-324.
- Louis, S. J. ve Xu, Z.** (1996). Genetic algorithms for open shop scheduling and re-scheduling. *ISCA Eleventh International Conference on Computers and their Applications* (Sf. 99-102).
- Low, C.** (2005). Simulated annealing heuristic for flow shop scheduling problems with unrelated parallel machines. *Computers & Operations Research*, 32, 2013-2025.
- Mehta, S. ve Uzsoy, R.** (1998). Predictable scheduling of a job shop subject to breakdowns. *IEEE Transactions on Robotics and Automation*, 14, 365-378.
- Mehta, S. ve Uzsoy, R.** (1999). Predictable scheduling of a single machine subject to breakdowns. *International Journal Computer Integrated Manufacturing*, 12, 15-38.
- Mendez, C. ve Cerda, J.** (2004). An MILP framework for batch reactive scheduling with limited discrete resources. *Computers & Chemical Engineering*, 28, 1059-1068.
- Morandin, O., Kato, E. R. R., Deriz, A. C. ve Sanches, D. S.** (2008). A search method using Genetic Algorithm for production reactive scheduling of manufacturing systems. *IEEE International Symposium on Industrial Electronics* (Sf. 1843-1848).
- Moursli, O. ve Pochet, Y.** (2000). A branch-and-bound algorithm for the hybrid flowshop. *International Journal of Production Economics*, 64, 113-125.
- Muhlemann, A. P., Lockett, A. G. ve Farn, C. K.** (1982). Job shop scheduling heuristics and frequency of scheduling. *International Journal of Production Research*, 20, 227-241.
- Naderi, B., Gohari, S. ve Yazdani, M.** (2014). Hybrid flexible flowshop problems: Models and solution methods. *Applied Mathematical Modelling*, 38, 5767-5780.
- Nawaz, M., Enscore, J. E. E. ve Ham, I.** (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega*, 11, 91-95.
- Nelson, R. T. ve Holloway, C. A.** (1977). Centralized scheduling and priority implementation heuristics for a dynamic job shop model with due dates and variable processing times. *AIIE Transactions*, 19, 95-102.
- Nowicki, E. ve Smutnicki, C.** (1998). The flow shop with parallel machines: a tabu search approach. *European Journal of Operational Research*, 106, 226-253.

- O'Donovan, R., Uzsoy, R. ve McKay, K. N.** (1999). Predictable Scheduling of a Single Machine with Breakdowns and Sensitive Jobs. *International Journal of Production Research*, 37, 4217-4233.
- Oguz, C. ve Ercan, M. F.** (1997). Scheduling multiprocessor tasks in a two-stage flow-shop environment. *Computers & Industrial Engineering*, 33, 269-272.
- Oguz, C., Ercan, M. F., Cheng, T. C. E. ve Fung, Y. F.** (2003a). Heuristic algorithms for multi-processor task scheduling in a two-stage hybrid flow-shop. *European Journal of Operational Research*, 149, 390-403.
- Oguz, C., Fung Y. F., Ercan, M. F. ve Qi, X. T.** (2003b). Parallel genetic algorithm for a flow-shop problem with multiprocessor Task. *ICCSA, Lecture notes in computer sciences*, 2667, 987-997.
- Oguz, C. ve Ercan, M.** (2005). A genetic algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *Journal of Scheduling*, 8, 323-351.
- Ovacık, I. M. ve Uzsoy, R.** (1994). Rolling horizon algorithms for a single machine dynamic scheduling problem with sequence-dependent set-up times. *International Journal of Production Research*, 32, 1243-1263.
- Özgün, K.** (2012). *Analysis of the impact of decesion time on the system performance in distributed systems*, (doktora tezi), Boğaziçi Üniversitesi, İstanbul.
- Pinedo, M.** (2008). *Scheduling Theory Algorithms, and Systems*. Springer, New York, NY.
- Ramsey, C. L. ve Grefenstette, J. J.** (1993). Case-based initialization of genetic algorithms. In S. Forrest (Ed.), *Fifth International Conference on Genetic Algorithms* (Sf. 84-91). Morgan Kaufmann.
- Reeves, C.** (2003). Genetic Algorithms. In: Glover F, Kochenberger GA (Ed.), *Handbook of Metaheuristics* (Sf. 55-82). Kluwer Academic Publishers, London.
- Ruiz, R. ve Vázquez-Rodríguez, J. A.** (2010). The hybrid flow shop scheduling problem. *European Journal of Operational Research*, 205, 1-18.
- Sabuncuoglu, I.** (1998). Scheduling with neural networks: A review of the literature and new research directions. *Production and Inventory Control*, 9, 2-12.
- Sabuncuoglu, I. ve Karabuk, S.** (1999). Rescheduling frequency in an FMS with uncertain processing times and unreliable machines. *Journal of Manufacturing Systems*, 18, 268-283.
- Sabuncuoglu, I. ve Bayiz, M.** (2000). Analysis of reactive scheduling problems in a job shop environment. *European Journal of Operational Research*, 126, 567-586.
- Sabuncuoglu, I. ve Kizilisik, O. B.** (2003). Reactive scheduling in a dynamic and stochastic FMS environment. *International Journal of Production Research*, 41, 4211-4231.

- Sawik, T.** (2000). Mixed integer programming for scheduling flexible flow lines with limited intermediate buffers. *Mathematical and Computer Modelling*, 31, 39-52.
- Sawik, T.** (2005). Integer programming approach to production scheduling for make-to-order manufacturing. *Mathematical and Computer Modelling*, 41, 99-118.
- Sawik, T.** (2006). Hierarchical approach to production scheduling in make-to-order assembly. *International Journal of Production Research*, 44, 801-830.
- Sawik, T.** (2007). Integer programming approach to reactive scheduling in make-to-order manufacturing. *Mathematical and Computer Modelling*, 46, 1373-1387.
- Sevaux, M. ve Sörensen, K.** (2004). A genetic algorithm for robust schedules in a one-machine environment with ready times and due dates. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 2, 129-147.
- Shaw, M., Park, S. ve Raman, N.** (1992). Intelligent scheduling with machine learning capabilities: the induction of scheduling knowledge. *IIE Transactions*, 24, 156-168.
- Sivrikaya Serifoglu, F. ve Ulusoy, G.** (2004). Multiprocessor task scheduling in hybrid flowshops: a genetic algorithm approach. *Journal of Operational Research Society*, 55, 504-512.
- Smith, S. F., Ow, P., Potwin, J., Muscettola, N. ve Matthys, D. C.** (1990). An integrated framework for generating and revising factory schedules. *Journal of Operational Research Society*, 41, 539-552.
- Soewandi, H. ve Elmaghraby, S.** (2001). Sequencing three-stage flexible flowshops with identical machines to minimize makespan. *IIE Transactions*, 2001, 33, 985-994.
- Soewandi, H. ve Elmaghraby, S.** (2003). Sequencing on two stage hybrid flowshops with uniform machines to minimize makespan. *IIE Transactions*, 35, 467-477.
- Stuart, K. ve Kozan, E.** (2011). Reactive scheduling model for the operating theatre. *Flexible Services and Manufacturing Journal*, 24, 400-421.
- Suresh, V.** (1997). A note on scheduling of two-stage flow shop with multiple processors. *International Journal of Production Economics*, 49, 77-82.
- Svetska, A. J. ve Abumaziar, R. J.** (1997). Rescheduling job shops under random disruptions. *International Journal of Production Research*, 35, 2065-2082.
- Szelke, E. ve Kerr, R.** (1994). Knowledge-based reactive scheduling. *Production Planning and Control*, 5, 124-145.
- Tanimizu, Y., Komatsu, Y., Ozawa, C., Iwamura, K., ve Sugimura, N.** (2010). Co-evolutionary genetic algorithms for reactive scheduling. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, 4, 569-577.

- Tavakkoli-Moghaddam, R., Safaei, N. ve Sassani, F.** (2009). A memetic algorithm for the flexible flow line scheduling problem with processor blocking. *Computers & Operations Research*, 36, 402-414.
- Thornton, H. W. ve Hunsucker, J. L.** (2004). A new heuristic for minimal makespan in flow shops with multiple processors and no intermediate storage. *European Journal of Operational Research*, 152, 96-114.
- Turkcan, A., Akturk, M. S. ve Storer, R. H.** (2009). Predictive/reactive scheduling with controllable processing times and earliness-tardiness penalties. *IIE Transactions*, 41, 1080-1095.
- Vandevelde, A., Hoogeveen, H., Hurkens, C. ve Lenstra, J. K.** (2005). Lower bounds for the head-body-tail problem on parallel machines: a computational study of the multiprocessor flow shop. *Informatics Journal on Computing*, 17, 305-320.
- Vandevonder, S., Ballestin, F., Demeulemeester, E. ve Herroelen, W.** (2007). Heuristic procedures for reactive project scheduling. *Computers & Industrial Engineering*, 52, 11-28.
- Wang, H., Jacob, V. ve Rolland, E.** (2003). Design of efficient hybrid neural networks for flexible flow shop scheduling. *Expert Systems*, 20, 208-231.
- Wang, Y. F., Zhang, Y. F., Fuh, J. Y. H., Zhou, Z. D., Lou, P. ve Xue, L. G.** (2008). An integrated approach to reactive scheduling subject to machine breakdown. *IEEE International Conference on Automation and Logistics* (Sf. 542-547).
- Wang, K. ve Choi S. H.** (2014). A holonic approach to flexible flow shop scheduling under stochastic processing times. *Computers & Operations Research*, 43, 157-168.
- Wardono, B. ve Fathi, Y.** (2004). A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. *European Journal of Operational Research*, 155, 380-401.
- Wu, D., Storer, R. H. ve Chang, P.** (1993). One-machine rescheduling heuristics with efficiency and stability as criteria. *Computers & Operations Research*, 20, 1-13.
- Wu, L. ve Storer, R. H.** (1994). Robustness scheduling for job shop. *IIE Transactions*, 26, 32-41.
- Wu, L. H., Chen, X. D., Chen, X. ve Chen, Q. X.** (2010). A genetic algorithm for reactive scheduling based on real-time manufacturing information. *5th International Conference on Responsive Manufacturing – Green Manufacturing* (Sf. 375-381), 11-13 Haziran.
- Yamamoto, M. ve Nof, S. Y.** (1985). Scheduling/rescheduling in the manufacturing operating system environment. *International Journal of Production Research*, 23, 705-722.
- Yang, B. ve Geunes, J.** (2008). Predictive-reactive scheduling on a single resource with uncertain future jobs. *European Journal of Operational Research*, 189, 1267-1283.

- Ying, K. C. ve Lin S. W.** (2006). Multiprocessor task scheduling in multistage hybrid flow-shops: an ant colony system approach. *International Journal of Production Research*, 44, 3161-3177.
- 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.
- Zandieh, M. ve Gholami, M.** (2009). An immune algorithm for scheduling a hybrid flow shop with sequence dependent setup times and with breakdowns. *International Journal of Production Research*, 47, 6999-7027.
- Zhang, W., Yin C. Y., Liu J. Y. ve Linn R. J.** (2005). Multi-job lot streaming to minimize the mean completion time in hybrid flowsops. *International Journal of Production Research*, 96, 189-200.
- Zhang, L., Gao, L. ve Li, X.** (2013). A hybrid genetic algorithm and tabu search for multi-objective dynamic job shop scheduling problem. *International Journal of Production Research*, 51, 3516-3531.

ÖZGEÇMİŞ



Ad Soyad: Abdullah AKTEL
Doğum Yeri ve Tarihi: İstanbul, 05.11.1981
E-Posta: abduhahaktel@gmail.com
Lisans: Yıldız Teknik Üniversitesi
Yüksek Lisans: Boğaziçi Üniversitesi

Mesleki Deneyim ve Ödüller:

Araştırma Görevlisi; Eylül 2009 - Haziran 2011, Okan Üniversitesi, İstanbul.
Öğretim Görevlisi; Haziran 2011 - Aralık 2012, Okan Üniversitesi, İstanbul.
Araştırmacı; Aralık 2012 - ... , TÜBİTAK Türkiye Sanayi Sevk ve İdare Enstitüsü.

Yayın Listesi:

Aktel, A., ve Guresen, E. 2013. Automatically Forecasting Magazine Demands and Modeling Seasonality, *AWERProcedia Information Technology & Computer Science*, 4, 370-374.

Aktel, A., ve Guresen, E. 2012. Automatically Forecasting Magazine Demands and Modeling Seasonality, *3rd World Conference on Information Technology (WCIT-2012)*, November 14-16, 2012 Barcelona, Spain.

Sansarcı, E., ve **Aktel, A.** 2011. Genetic Algorithm with Geographical Isolation on a Unit Sphere, *2nd World Conference on Information Technology (WCIT-2011)*, November 23-27, 2011 Antalya, Turkey.

Sansarcı, E., **Aktel, A.**, Çelebi, D., ve Bayraktar, D. 2009. Gezin Satıcı Problemi İçin Bir Memetik Algoritma Önerisi, *Yöneylem Araştırması ve Endüstri Mühendisliği 29. Ulusal Kongresi*, Haziran 22-24, 2009 Ankara (CD-ROM), Türkiye.

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

▪ **Aktel, A.**, ve Yenisey, M. M. 2014. Melez Akış Tipi Çizelgeleme Problemi İçin Bir Genetik Algoritma, *KHO Bilim Dergisi*, 24(2), 61-82.

▪ **Aktel, A.**, ve Yenisey, M. M. 2014. A Genetic Algorithm for Dynamic Single Machine Scheduling Problem, *Yöneylem Araştırması ve Endüstri Mühendisliği 34. Ulusal Kongresi*, Haziran 25-27, 2014 Bursa, Türkiye.