MELEZ AKIŞ TİPİ ÇİZELGELEME PROBLEMİNİN ÇÖZÜMÜ İÇİN ÖĞRENME ETKİSİ TABANLI ÇOK AMAÇLI GENETİK ALGORİTMA

(Yüksek Lisans Tezi)

Mahide TEKÇE

Kütahya - 2023

T.C.
KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
Endüstri Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

MELEZ AKIŞ TİPİ ÇİZELGELEME PROBLEMİNİN ÇÖZÜMÜ İÇİN ÖĞRENME ETKİSİ TABANLI ÇOK AMAÇLI GENETİK ALGORİTMA

Danışman:
Dr. Öğr. Üyesi Derya DELİKTAŞ

Hazırlayan:
Mahide TEKÇE

Kütahya – 2023

Kabul ve Onay

Endüstri Mühendisliği Ana bilim dalında, 801921110121 öğrenci numaralı, Mahide TEKÇE'nin hazırlamış olduğu "MELEZ AKIŞ TİPİ ÇİZELGELEME PROBLEMİNİN ÇÖZÜMÜ İÇİN ÖĞRENME ETKİSİ TABANLI ÇOK AMAÇLI GENETİK ALGORİTMA" başlıklı yüksek lisans tez çalışması ile ilgili tez savunma sınavı jüri tarafından yapılmış ve adayın tezinin OY BİRLİĞİ ile kabul edilmesine karar verilmiştir.

06/02/2023

Tez Jürisi	**İmza**	
	Kabul	**Red**
Dr. Öğr. Üyesi Derya DELİKTAŞ (Danışman)		
Prof. Dr. Özden ÜSTÜN		
Prof. Dr. Gürkan ÖZTÜRK		

Onay

Doç. Dr. Arif KOLAY

Enstitü Müdürü

Bilimsel Etik Bildirimi

Yüksek Lisans tezi olarak hazırladığım "*Melez Akış Tipi Çizelgeleme Probleminin Çözümü İçin Öğrenme Etkisi Tabanlı Çok Amaçlı Genetik Algoritma*" adlı çalışmamın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

06/02/2023

Mahide TEKÇE

Özgeçmiş

İlk, orta ve lise eğitimini Zonguldak'ta tamamladı. Lisans eğitimini Kütahya Dumlupınar Üniversitesi Endüstri Mühendisliği bölümünde tamamladı. 2012 yılında Eskişehir Arçelik A.Ş. fabrikasında üretim stajını tamamladı. 2014 yılında Dumlupınar Üniversitesi Bilgi İşlem Dairesinde bilgisayar stajını tamamladı. 2014 yılında Cilas Kauçuk A.Ş fabrikasında üretim sistemleri stajını tamamladı. 2020 yılında Dumlupınar Üniversitesi Endüstri Mühendisliği Bölümünde Yüksek Lisans eğitimine başladı.

ÖZET

MELEZ AKIŞ TİPİ ÇİZELGELEME PROBLEMİNİN ÇÖZÜMÜ İÇİN ÖĞRENME ETKİSİ TABANLI ÇOK AMAÇLI GENETİK ALGORİTMA

TEKÇE, Mahide
Yüksek Lisans Tezi, Endüstri Mühendisliği Ana Bilim Dalı
Tez Danışmanı: Dr. Öğr. Üyesi Derya DELİKTAŞ
Şubat, 2023, 106 sayfa

Teknolojinin hızla gelişmesi sebebiyle rekabetin iyice zorlaştığı üretim ortamlarında çizelgeleme, firmaların piyasada ayakta durabilmesini sağlayan en önemli faktörlerden biridir. Bu çalışmada, acil durum aydınlatma ürünleri üreten bir firmanın üretim sistemi incelenmiş ve çizelgeme türünün Melez Akış Tipi Çizelgeleme (MATÇ) problemi sınıfına girdiği belirlenmiştir. Ayrıca, bu imalat ortamında yapılan işlerin hızı ve kalitesi büyük oranda çalışanların el becerilerine dayalıdır. Bu çalışmada, en büyük tamamlanma zamanının ve toplam gecikme süresinin en küçüklenmesi amacıyla işler arası hazırlık sürelerini, taşıma sürelerini ve öğrenme etkisini dikkate alan çok amaçlı MATÇ probleminin çözümüne dair bir karma tamsayılı doğrusal matematiksel model önerilmiştir. Problemin karmaşık yapısı ve çok amaçlı olması nedeniyle, Ağırlıklı-Toplam (ATY), Konik (KSY) ve Tchebycheff (TSY) olmak üzere üç farklı skalerleştirme tabanlı çok amaçlı bir genetik algoritma (ÇAGA) çözüm yaklaşımı önerilmiştir. Skalerleştirme varyantlı GA'nın performansı, küçük, orta ve büyük boyutlu olan literatürden alınan 21 adet kıyaslama örneği ile ideal noktaya ağırlıklı uzaklık ve referans noktasına ağırlıklı uzaklık olmak üzere iki farklı performans metriği kullanılarak test edilmiştir. Önerilen ÇAGA'nın parametre ayarları, **irace** kullanılarak belirlenmiştir. Karşılaştırma sonuçları, ideal noktaya ağırlıklı uzaklık performans metriğine göre ATY'li GA'nın diğer algoritmalardan daha iyi sonuç verdiğini gösterirken referans noktasına ağırlıklı uzaklık performans metriğine göre TSY'li GA'nın KSY'li GA'dan daha iyi performans sergilediğini göstermektedir. Önerilen algoritma, ele alınan firmaya uygulanmıştır. Deneysel sonuçlara göre, önerilen algoritmanın etkili bir şekilde çalışabileceği ve firmanın kullanabileceği kullanıcı dostu bir arayüze sahip Karar Destek Sistemi tasarlanmıştır.

Anahtar Kelimeler: Genetik Algoritma, İrace, Karar Destek Sistemi, Karma Tamsayılı Doğrusal Programlama, Melez Akış Tipi Çizelgeleme, Öğrenme Etkisi

ABSTRACT

MULTI-OBJECTIVE GENETIC ALGORITHM BASED ON LEARNING EFFECT FOR SOLVING HYBRID FLOW SHOP SCHEDULING PROBLEM

TEKÇE, Mahide
Master's Thesis, Department of Industrial Engineering
Supervisor: Asst. Prof. Derya DELİKTAŞ
February, 2023, 106 pages

Scheduling is one of the most important issues that enable companies to survive in production environments where competition is very intense with the development of technology. In this study, the production system of a company producing emergency lighting products was discussed and it was determined that the scheduling type belonged to the Hybrid Flow Shop Scheduling (HFSS) problem class. In addition, the speed and quality of the work performed in this manufacturing environment are largely based on the manual skills of the operators. In this study, a mixed integer linear mathematical model for solving the multi-objective HFSS problem, which considers the setup times between jobs, removal times, and learning effect, is proposed to minimize the maximum completion time and total tardiness. Due to the complex structure of the problem and its multi-objective nature, a Multi-Objective Genetic Algorithm (MOGA) solution approach based on three different scalarization methods, Weighted-Sum (WSM), Conic (CSM) and Tchebycheff (TSM), has been proposed. The performance of the scalarization variant GA was tested using two different performance metrics: weighted distance to the ideal point and weighted distance to the reference point, with 21 benchmark instances taken from the literature, small, medium, and large. The parameter settings of the proposed MOGA were determined using **irace**. The comparison results show that GA with WSM performs better than other algorithms in terms of the weighted distance to the ideal point, while GA with TSM performs better than GA with CSM in accordance with the weighted distance to the reference point. The proposed algorithm was applied to the company. According to the experimental results, a Decision Support System with a user-friendly interface has been designed, through which the suggested algorithm can work effectively, and the company can use it.

Keywords: Decision Support System, Genetic Algorithm, Hybrid Flow Shop Scheduling, Irace, Learning Effect, Mixed Integer Linear Programming

ÖNSÖZ

Bu tezi hazırlarken bana gösterdiği sonsuz hoşgörüsü ve destekleri için sevgili eşim Ferzende TEKÇE'ye, akademik çalışma hayatı konusunda çok sınırlı bilgim olmasına karşın, bana bilimsel çalışma yürütme becerisi kazandıran, farkında olmadığım potansiyelimi açığa çıkaran, yüksek lisans eğitimim boyunca bana her gün yeni şeyler öğreten, öğrencisi olmaktan her zaman gurur duyacağım tez danışmanım çok değerli Dr. Öğr. Üyesi Derya DELİKTAŞ'a, bana kazandırdıkları bilgilerle bu tezin hazırlanmasını sağlayan Dumlupınar Üniversitesi Endüstri Mühendisliği Ana Bilim Dalındaki tüm değerli hocalarıma teşekkürlerimi sunarım.

Tez jürime katılma nezaketini gösteren değerli hocalarım Prof. Dr. Özden ÜSTÜN'e ve Prof. Dr. Gürkan ÖZTÜRK'e teşekkürlerimi sunarım.

Bu çalışmanın oluşturulması için ihtiyacımız olan her türlü bilgiyi ve desteği esirgemeyerek bize firmasının kapısını açan Sayın Altan EZER' e teşekkürlerimi sunarım.

Devamlı yenilik arayışında olan, kendisini ve yaptığı işi sürekli olarak geliştirme çabasında bulunarak beni de kendimi geliştirmeye teşvik ettiği için değerli Sebahattin TÜRKOĞLU'na teşekkürlerimi sunarım.

Son olarak bursiyer olarak yer aldığım bu çalışma, 222M366 proje numarası ile "TÜBİTAK ARDEB 1002/A Hızlı Destek Programı" tarafından desteklenmiştir. Maddi desteklerinden dolayı TÜBİTAK'a teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖZET ..... **v**
ABSTRACT ..... **vi**
TABLOLAR LİSTESİ ..... **x**
ŞEKİLLER LİSTESİ ..... **xi**
GÖRSELLER LİSTESİ ..... **xii**
KISALTMALAR VE SEMBOLLER ..... **xiii**
GİRİŞ **1**

BİRİNCİ BÖLÜM

LİTERATÜR ARAŞTIRMASI

1.1. LİTERATÜR ARAŞTIRMASI **4**

1.1.1. MATÇ'de Doğrusal Programlama Modeli ile İlgili Yapılan Çalışmalar 4
1.1.2. MATÇ Probleminin Çözümü için Önerilen Sezgisel Yöntemler 5
1.1.2.1. Belirsiz Süreler İçeren Çalışmalar 8
1.1.2.2. Gerçek Hayat Uygulamaları 9
1.1.2.3. İnsan Faktörü ve Öğrenme Etkisi ile İlgili Yapılan Çalışmalar 10

İKİNCİ BÖLÜM

METODOLOJİ

2.1. PROBLEMİN TANITIMI **19**

2.2. ÖĞRENME ETKİSİ **20**

2.3. KARMA TAMSAYILI DOĞRUSAL PROGRAMLAMA MODELİ **21**

2.4. GENETİK ALGORİTMA **28**

2.4.1. Başlangıç Popülasyonu 30
2.4.2. Uygunluk Değerinin Hesaplanması 31
2.4.3. Kromozom Yapısı 32
2.4.4. Seçim 34
2.4.5. Çaprazlama 36
2.4.6. Mutasyon 38
2.4.7. Elitizm 39
2.4.8. Durdurma Koşulu 40

ÜÇÜNCÜ BÖLÜM

MATÇ PROBLEMİNİN ÇÖZÜMÜ İÇİN GELİŞTİRİLEN ÖĞRENME ETKİSİ TABANLI ÇAGA

3.1. PARAMETRE AYARLAMA 43

3.1.1. Literatürde Kullanılan Parametre Ayarlama Yöntemleri 44
3.1.2. Tercih Edilen Parametre Ayarlama Yöntemi 44

3.2. PERFORMANS METRİKLERİ 45

3.3. KIYASLAMA ÖRNEKLERİ 47

3.3.1. Kıyaslama Örneklerinden Elde Edilen Algoritma Parametreleri 54

DÖRDÜNCÜ BÖLÜM

DENEY SONUÇLARI VE ANALİZ

4.1. SKALERLEŞTİRME YÖNTEMLERİNE DAYALI GA'NIN PERFORMANS KARŞILAŞTIRMASI 56

BEŞİNCİ BÖLÜM

KARAR DESTEK SİSTEMİ

5.1. KARAR DESTEK SİSTEMİ 62

5.1.1. KDS'nin Gerçek Verilerle Kullanımı ve Sonuçlar 71

SONUÇ VE ÖNERİLER 75

KAYNAKÇA 77

DİZİN 86

TABLOLAR LİSTESİ

Sayfa

Tablo 1.1: Literatür Araştırması Özeti 12
Tablo 2.1: Öğrenme Etkisinin Hesaplanması 21
Tablo 2.2: Örnek Problemin İşlem Süreleri 26
Tablo 2.3: Örnek Problemin Sıra Bağımlı Hazırlık Süreleri 27
Tablo 2.4: Örnek Problemin Aşamalar Arası Taşıma Süreleri 27
Tablo 2.5: Örnek Problemin Teslim Süreleri 27
Tablo 2.6: ÇAGA Sözde (Pseudo) Kodu 30
Tablo 2.7: ÇAGA Uygunluk Değeri Hesaplama Fonksiyonu Sözde (Pseudo) Kodu 32
Tablo 2.8: ÇAGA Seçim Fonksiyonu Sözde (Pseudo) Kodu 35
Tablo 2.9: ÇAGA Çaprazlama Fonksiyonu Sözde (Pseudo) Kodu 38
Tablo 2.10: ÇAGA Mutasyon Fonksiyonu Sözde (Pseudo) Kodu 39
Tablo 2.11: ÇAGA Elitizm Fonksiyonu Sözde (Pseudo) Kodu 40
Tablo 3.1: No#10 Örneği için Üç Farklı Referans Noktası Kombinasyonuna Göre Referans Noktaları 46
Tablo 3.2: MATÇ Problemi İçin Literatürde Çalışılan Kıyaslama (Benchmark) Örnekleri ve Örnek Boyutları 48
Tablo 3.3: Oluşturulan Örnek Seti ve Örnek Büyüklükleri 53
Tablo 3.4: Parametre Aralıkları ve Elde Edilen En İyi Parametre Değerleri 54
Tablo 4.1: 21 Kıyaslama Örneği İçin Performans Metriklerine Göre MATÇ Probleminin Skalerleştirme Yöntemleriyle GA'nın Performanslarının Karşılaştırılması 57
Tablo 5.1: Gerçek Hayat Verileri, Hazırlık Süreleri (Dakika) 72
Tablo 5.2: Gerçek Hayat Verileri, İşlem Süreleri (Dakika) 73
Tablo 5.3: Gerçek Hayat Verileri, Taşıma Süreleri (Dakika) 73

ŞEKİLLER LİSTESİ

Sayfa

Şekil 1.1: MAA Genel Yapısı 4
Şekil 2.1: İmalat Ortamının Yapısı 20
Şekil 2.2: Örnek Problemin Gantt Şeması 28
Şekil 2.3: Genetik Algoritma Akış Şeması 30
Şekil 2.4: Geliştirilen ÇAGA'nın Kromozom Yapısı 33
Şekil 2.5: Çaprazlama Operatörü 37
Şekil 2.6: Mutasyon Operatörü 39
Şekil 4.1: Üç Farklı Konveks Kombinasyon Vektörü ile Her Bir Boyuttan Alınan Üç Numune Örneği İçin Skalerleştirme Yöntemlerine Dayanan GA'nın 31 Çalıştırmasından Elde Edilen İdeal Noktalara Ağırlıklı Uzaklıkların Kutu Grafikleri 59
Şekil 4.2: Üç Farklı Konveks Kombinasyon Vektörü ile Her Bir Boyuttan Alınan Üç Numune Örneği İçin Skalerleştirme Yöntemlerine Dayanan GA'nın 31 Çalıştırmasından Elde Edilen Referans Noktalara Ağırlıklı Uzaklıkların Kutu Grafikleri 60

GÖRSELLER LİSTESİ

Sayfa

Görsel 5.1: KDS Giriş Sekmesi, "Üretim Hattı" Sayfası 64
Görsel 5.2: KDS Giriş Sekmesi, "Üretim Hattını Düzenle" Sayfası 65
Görsel 5.3: KDS Giriş Sekmesi, "Ürünler" Sayfası 65
Görsel 5.4: KDS Giriş Sekmesi, "İşlem Süreleri" Sayfası 66
Görsel 5.5: KDS Giriş Sekmesi, "Hazırlık Süreleri" sayfası 67
Görsel 5.6: KDS Giriş Sekmesi, "Taşıma Süreleri" Sayfası 67
Görsel 5.7: KDS Çizelge Sekmesi 68
Görsel 5.8: KDS Çizelge Sekmesi, "İş Emri Oluştur" Sayfası 69
Görsel 5.9: KDS Çizelge Sekmesi, "Gantt Şeması" 70
Görsel 5.10: KDS "Çizelge" Sekmesi, "Çizelge Sonuçları" 71
Görsel 5.11: KDS ile Elde Edilen Gerçek Hayat Örneği Gantt Şeması 74
Görsel 5.12: KDS ile Elde Edilen Gerçek Hayat Örneği Yakınsama Grafiği 74

KISALTMALAR VE SEMBOLLER

Kısaltmalar	Açıklama
C_{enb}	En Büyük Tamamlanma Zamanı
$\sum T$	Toplam Gecikme Zamanı
3BGA	Üç Boyutlu Genetik Algoritma
ABSA	Ayrık Balina Sürüsü Algoritması
ADEA	Ayrık Diferansiyel Evrim Algoritması
ADGÇAEA	Ayrışmaya Dayalı Geliştirilmiş Çok Amaçlı Evrimsel Algoritma
ADY	Ayrışmaya Dayalı Yaklaşım
AEKKTPM	Artırılmış Epsilon-Kısıt Karma Tamsayılı Programlama Metodu
ASÇES	Akış Süresi Çoklu Ekleme Sezgiseli
ATÇK	Açgözlü Tipi Çizelgeleme Kuralı
ATY	Ağırlıklı Toplam Yöntemi
AYAK	Ayrık Yapay Arı Kolonisi
BA	Bağışıklık Algoritması
BFS	Böl ve Fethet Sezgiseli
BMKBGADYÇY	Bulanık Mantık Kontrolü ile Birleştirilen veya Birleştirilmeyen Genetik Algoritmasına Dayanan Yaklaşık Çözünürlük Yöntemleri
BMKBPSODYÇY	Bulanık Mantık Kontrolü ile Birleştirilen veya Birleştirilmeyen Parçacık Sürüsü Optimizasyon Algoritmasına Dayanan Yaklaşık Çözünürlük Yöntemleri
BOKBA	Baskın Olmayan Komşu Bağışıklık Algoritması
BOSGA	Baskın Olmayan Sıralanmış Genetik Algoritma
BOSGAII	Baskın Olmayan Sıralama Genetik Algoritması
BOSHSABA	Baskın Olmayan Sıralama ile Hibrit Su Akışı Benzeri Algoritma
BSA	Balina Sürüsü Algoritması
BT	Benzetimli Tavlama
BYAKTAS	Birleştirilmiş Yapay Arı Kolonisi ve Tabu Araştırma Sezgiseli

CSM	Conic Scalarization Method
ÇAGA	Çok Amaçlı Genetik Algoritma
ÇATA	Çok Amaçlı Tabu Araştırması
ÇFGA	Çok Fazlı Genetik Algoritma
ÇK	Çıkarma Kuralı
ÇKM	Çift Kodlamalı Metasezgisel
DAİFSTT	Dağıtım Algoritmasının İki Fazlı Simülasyon Tabanlı Tahmini
DASİMDT	Dağıtım Algoritmasının Simülasyona Dayalı Tahmini
DASİPDT	Dağıtım Algoritmasının Siparişe Dayalı Tahmini
DAT	Dağıtım Algoritmasının Tahmini
DE	Diferansiyel Evrim
DKA	Değişken Komşuluk Araması
DKKZA	Dinamik Karma Kurbağa Zıplaması Algoritması
DNA	Dağıtılmış NEH Algoritması
DS	Döngüsel Sezgisel
DSA	Dal ve Sınır Algoritması
DSS	Decision Support System
EDGEKO	Etkili Düzenli Göç Eden Kuşlar Optimizasyonu
EEBT	En Erken Bitiş Tarihi
EEHZ	En Erken Hazırlık Zamanı
EİZDS	Enkısa İşlem Zamanı Döngüsel Sezgiseli
EKHZ	En Kısa Hazırlık Zamanı
EKİZ	En Kısa İşlem Zamanı
EUİZ	En Uzun İşlem Zamanı
EUKİS	En Uzun Kalan İşlem Süresi
GA	Genetik Algoritma
GA_OE	Oğuz ve Ercan Tarafından Önerilen Genetik Algoritma
GA_SU	Şerifoğlu ve Ulusoy Tarafından Önerilen Genetik Algoritma
GAYAK	Geliştirilmiş Ayrık Yapay Arı Kolonisi
GÇGA	Güçlü Çizelgeleme için Genetik Algoritma
GDÇGA	Geriye Dönük Çizelgeleme ile Genetik Algoritma
GDKA	Global Değişken Komşuluk Araması

GİAOATA	Geri İzleme Arama Optimizasyon Algoritması ve Tabu Araması
GKAA	Guguk Kuşu Arama Algoritması
GMSO	Geliştirilmiş Meyve Sineği Optimizasyonu
GPEA2	Güçlü Pareto Evrimsel Algoritma-2
GPSO	Geliştirilmiş Parçacık Sürü Optimizasyonu
HBT	Hibrit Benzetimli Tavlama
HBTYS	Hızlı Bellek Tabanlı Yapıcı Sezgisel
HDKA	Hibrit Değişken Komşuluk Araştırması
HFS	Hybrid Flow Shop
HFSS	Hybrid Flow Shop Scheduling
HUA	Hibrit Uyum Araştırması
HYAKA	Hibrit Yapay Arı Kolonisi Algoritması
HZDS	Hazır Zaman Döngüsel Sezgiseli
İAÇLAP	İki Aşamalı Çoklu Lot Lot Akış Problemi
İAYAA	İki Aşamalı Yinelemeli Açgözlü Algoritma
İBYBSA	İmmünoglobulin Bazlı Yapay Bağışıklık Sistemi Algoritması
İDÇGA	İleriye Dönük Çizelgeleme ile Genetik Algoritma
İPİMA	İki Populasyonlu İşbirlikçi Memetik Algoritması
JK	Johnson Kuralı
JKTS	Johnson Kuralı Tabanlı Sezgisel
KA	Kuantum Algoritması
KAA	Komşuluk Arama Algoritması
KBA	Kuantum Bağışıklık Algoritması
KDS	Karar Destek Sistemi
KEKİS	Kalan En Kısa İşlem Süresi
KKAYA	Kendi Kendini Ayarlayan Yinelemeli Açgözlü
KKO	Karınca Kolonisi Optimizasyonu
KKS	Karınca Koloni Sistemi
KKZA	Karma Kurbağa Zıplaması Algoritması
KP	Kısıt Programlama
KRO	Kimyasal Reaksiyon Optimizasyonu

KSY	Konik Skalerleştirme Yöntemi
KTDP	Karma Tamsayılı Doğrusal Programlama
KTPTA	Küçük Tohumlu Paralel Tabu Araştırması
LGA	Lagrange Gevşeme Algoritması
MA	Memetik Algoritma
MAA	Melez Akış Atölyesi
MATÇ	Melez Akış Tipi Çizelgeleme
MATÇP-İPM-SBHS-B	Melez Akış Tipi Çizelgeleme Problemi-İlgisiz Paralel Makineler-Sıra Bağımlı Hazırlık Süreleri-Beklemesiz
MATÇP-İPM-SBHS-E	Melez Akış Tipi Çizelgeleme Problemi-İlgisiz Paralel Makineler-Sıra Bağımlı Hazırlık Süreleri-Engellemeli
MÇES	Makespan Çoklu Ekleme Sezgiseli
MGS	Minimum Gevşeme Süresi
MOGA	Multi Objective Genetic Algorithm
NEH	NEH Algoritması
OBİBT	Optimal Bilgi İşlem Bütçesi Tahsisi
OYDDMA	Olası Bir Yaklaşıma Dayalı Değişken Mahalle Araması
PAA	Paralel Açgözlü Algoritma
PSO	Parçacık Sürü Optimizasyonu
PSOTL	Tseng ve Liao Tarafından Önerilen Parçacık Sürü Optimizasyonu
RAGA	Rastgele Anahtar Genetik Algoritma
RAJ	Rajendran Sezgiseli
RMGA	Ruiz ve Maroto Tarafından Önerilen Genetik Algoritma
RYSKA	Rastgele Yeniden Seçim Kuantum Algoritması
S1, S2	Sezgisel1, Sezgisel2
SABA	Su Akışı Benzeri Algoritma
SAYAA	Sınırlı Arama Yinelemeli Açgözlü Algoritma
SKÇÇAEA	Sezgisel Kod Çözme ile Çok Amaçlı Evrimsel Algoritmalar
SKSHSABA	Sıralama Kavramlarına Sahip Hibrit Su Akışı Benzeri Algoritma
STBÖ	Simüle Tavlama Blok Özellikli
TAA	Tabu Araştırması Algoritması

TADAİS	Tabu Araması Dağılım Aramasını İçeren Sezgisel
TDSBTA	Tırmanma Derinliği Sınırlamalı Bitişik Tutarsızlık Araması
TKGA	Tek Kodlamalı Genetik Algoritma
TSÇES	Tamamlanma Süresi Çoklu Ekleme Sezgiseli
TSM	Tchebycheff Scalarization Method
TSMODAİFSTT	Tavlama Seçim Mekanizması Olmadan Dağıtım Algoritmasının İki Fazlı Simülasyon Tabanlı Tahmini
TSY	Tchebycheff Skalerleştirme Yöntemi
TTSTA	Tepe Tırmanma Simüle Tavlama Algoritması
TTYAA	Tepe Tırmanma Yerel Arama Algoritması
UBA	Uyumsal Bağışıklık Algoritması
URYSKBA	Uyumsal Rastgele Yeniden Seçim Kuantum Bağışıklık Algoritması
YAA	Yinelemeli Açgözlü Algoritma
YAA_Ribas	Ribas Tarafından Önerilen Yinelemeli Açgözlü Algoritma
YAAwS	Yinelemeli Açgözlü Algoritma
YAÇAHGA	Yerel Arama ile Çok Amaçlı Hibrit Genetik Algoritma
YAK	Yapay Arı Kolonisi
YBS	Yapay Bağışıklık Sistemi
YG	Yinelemeli Gevşeme
YGA	Yaurima Tarafından Önerilen Genetik Algoritma
YS	Yapıcı Sezgisel
YSGA	Yön Seçimi ile Genetik Algoritma
YYA_Ribas	Ribas Tarafından Önerilen Yinelemeli Yerel Arama
YYAM	Yinelemeli Yerel Arama Metasezgiseli
WSM	Weighted-Sum Method

TEZ METNİ

GİRİŞ

Üretim planlama, hızla değişen küresel pazarda rekabetçi bir konumu sürdürmek için herhangi bir imalat endüstrisinde hayati önem taşımaktadır. Ürün fiyatını rekabet belirler ve karı artırmanın tek yolu üretim ve dağıtım maliyetlerini düşürmektir. Şirketlerin bu rekabet ortamında hayatta kalması ve kar elde edebilmesi için üretim kaynaklarını verimli şekilde kullanması gerekmektedir. Bu, organizasyonu verimli bir şekilde yönetmek ve işletmek anlamına gelmektedir. Bunun için etkili ve verimli çizelgeleme teknolojileri ve yaklaşımları geliştirilmektedir (Sule, 2007).

Üretim planlama terminolojisinde çizelgeleme modelleri, tek makine, akış atölyesi, paralel makineler, iş atölyesi, açık atölye, bağımlı atölye, seri işleme, sıra bağımlı hazırlık zamanı, işgücü planlama ve montaj hattı çizelgeleme olarak kategorilere ayrılmıştır. Bu kategorilerin oluşmasındaki en büyük etken imalatta kullanılan makinelerin türleri, makine sayıları ve ürün özellikleridir.

Firmalar, rekabet ortamına uyum sağlamak ve kar elde edebilmek için ürün hacmini, ürün çeşitliliğini ve üretim hızını artırmayı tercih edebilir. Bu tercihler, çok amaçlı makine kullanımına, makine sayılarını artırmaya veya üretim ortamını düzenlemeye yönelik yatırımlar yapmalarını gerektirmektedir. Üretim ortamındaki her bir değişiklik yapılacak çizelgeleme üzerinde de farklılıklar oluşturmaktadır. Örneğin, tüm işlerin aynı işlem rotasını takip ettiği bu nedenle farklı makine türlerinin sabit sıra ile dizildiği sistemlere akış atölyesi denilmektedir. Akış atölyesi sisteminde aynı işi yapabilecek yeni makineler eklenmesi durumunda sistem Melez Akış Atölyesi (MAA) problemi olarak adlandırılmaktadır.

Literatürde MAA çizelgeleme problemleri Melez Akış Tipi Çizelgeleme (MATÇ) olarak adlandırılır. MATÇ yapısında bulunan paralel makineler literatürde özdeş (identical) veya ilişkisiz (unrelated) olarak çalışılmıştır. Her işletme kendi içerisinde farklı sistem unsurlarına sahiptir, bazıları darboğaz bir makinenin bulunduğu aşamaya paralel özdeş bir makine daha ekleyerek bu problem ile baş edebilmektedir (Liao, Tjandradjaja ve Chung 2012). Bazı sistemler, benzer işleri farklı sürelerde yapabilen ilişkisiz makinelere sahiptir (Torna, Freze, CNC gibi), bu makinelerin işleri işleme hızları farklılık gösterse de yoğun sipariş ortamına hızlı cevap verebilmek amacıyla işler bu makinelere farklı kombinasyonlarla atanabilmektedir. Bu sayede hem işlerin teslim tarihine yetişmesi hem de daha fazla sipariş karşılanmış olur (Aqil ve Allali,

2020). Ayrıca MATÇ sistemlerinde stokastik ve deterministik olarak kullanılan süreler, bu problemin çözümü için geliştirilen algoritmalar için farklı yaklaşımlar sağlamaktadır. Buna ek olarak, bir sistemde sıra bağımlı hazırlık süresinin olması problemin çözümü için uygulanacak yöntemin belirlenmesinde önemli bir rol oynamaktadır. Her bir işin hazırlık süresi, işlerin sırasına bağlı olarak değişiyorsa bu sistem sıra bağımlı hazırlık süresine sahiptir.

Bu çalışmada ele alınan imalat ortamı akış tipi imalat yapısı ve özdeş paralel makine bulundurması sebebiyle MATÇ problemi olarak belirlenmiştir. Bu sistemde müşteri siparişlerinin farklı ürün boyutları, elektrik değerleri, renkler veya gövdeler içermesi nedeniyle ürün çeşitliliği fazladır, ayrıca proje bazlı üretim yapıldığı için sipariş miktarları da fazladır. Bu imalat ortamında etkili bir çizelgeleme yapılmasının üretim maliyetlerini azaltması ve teslim sürelerinde büyük bir iyileşme sağlayacağı açıktır. Ayrıca bu işletmede yapılan işin kalitesi ve hızı büyük oranda operatörlerin el becerilerine dayanmaktadır. Bu MATÇ problemi içerisinde sıra bağımlı hazırlık süreleri, operatör bazlı işlem süreleri ve aşamalar arasında taşıma işlemleri gerektirmektedir. Hazırlık ve işlem sürelerinin çalışanların öğrenme becerisi ile büyük oranda etkilendiği bu sistemde yapılacak çizelgelemede öğrenme etkisinin her iki zaman tanımına uygulanması gerektiği düşünülmüştür. Literatürde öğrenme etkisinin sıra bağımlı hazırlık süreleri üzerine uygulandığı çalışmalar bulunmaktadır. Ancak işlem süreleri üzerine herhangi bir çalışma rastlanılmamaktadır. Bu amaçla öncelikle ele alınan sistemin özelliklerine uygun matematiksel model belirlenmiştir. Matematiksel modelin doğruluğu Lingo 11.0 optimizasyon programı ile küçük boyutlu bir örnek üzerinde test edilmiştir. Problemin çok amaçlı ve NP-zor yapısı (Gupta, 1988) nedeniyle çözümü için bir sezgisel algoritma geliştirilmesine ihtiyaç duyulmuştur.

Çalışmanın devamında, birinci bölümde literatür özeti sunulacaktır, ikinci bölümde uygulanan metodoloji detaylandırılacaktır, üçüncü bölümde geliştirilen algoritmanın performans ölçümü anlatılacak, dördüncü bölümde deney sonuçları sunulacaktır. Beşinci bölümde geliştirilen Karar Destek Sistemi (KDS) tanıtılacak, firmadan alınan verilere KDS uygulanacaktır, son olarak bu çalışmadan elde edilen sonuçlar değerlendirilecek ve gelecek çalışmalar ile ilgili önerilerde bulunulacaktır.

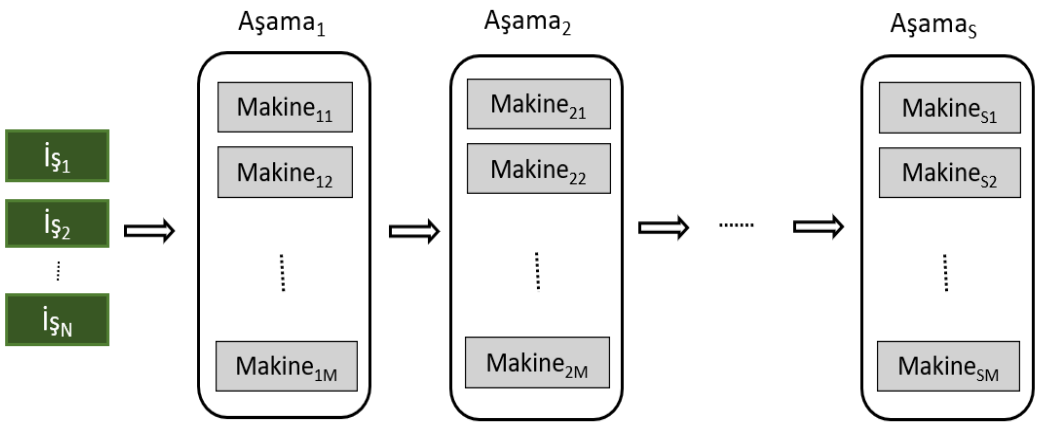
BİRİNCİ BÖLÜM
LİTERATÜR ARAŞTIRMASI

1.1. LİTERATÜR ARAŞTIRMASI

MATÇ problemi genel olarak, *n* adet işin aşamalardan geçtiği, her bir aşamada *m* adet makinenin bulunduğu (en az bir aşamada paralel makinelerin bulunduğu) MAA sistemlerini ele alır. Şekil (1.1)'de MAA'nın genel yapısı gösterilmiştir.

Gerçek hayat üretim sistemleri farklı üretim süreçlerine sahiptir. Bu üretim süreçleri bünyesinde, operatöre bağlı geleneksel veya modern makineler, robotlar, bilgisayarla bütünleşik sistemler vb., gerek üretim ortamı gerek üretilen ürünlerin spesifik özelliklerine bağlı olarak değişmektedir. MATÇ, farklı üretim ortamlarına uyarlanabilir olması nedeniyle literatürde en çok çalışılan konulardan biri olmuştur (Ozsoydan ve Sağir, 2021).

Şekil 1.1: MAA Genel Yapısı



MATÇ problemi ile ilgili yapılan çalışmalar aşağıda anlatılan özelliklerine göre başlıklandırılmış ve literatür incelemesi bu başlıklar altında toplanmıştır. Üretim ortamlarının çeşitliliği ve barındırdıkları birçok spesifik özellikleri nedeniyle MATÇ sisteminde matematiksel model üzerinde çalışılan araştırmalar "Doğrusal Programlama Modeli" başlığı altında incelenmiştir. MATÇ probleminin çözümü için geliştirilen sezgiseller ve bu sezgiseller ile yapılan çalışmalar "Sezgiseller" başlığı altında incelenmiştir. İncelenen sezgiseller de kendi içinde odaklandıkları problemlerin özelliklerine göre, "Belirsiz Süreler", "Gerçek Hayat Uygulamaları", "İnsan Faktörü ve Öğrenme Etkisini Baz Alan Sezgiseller" olarak alt başlıklar altında incelenmiştir.

1.1.1. MATÇ'de Doğrusal Programlama Modeli ile İlgili Yapılan Çalışmalar

MATÇ probleminin çözümü için en önemli basamaklardan biri problemin matematiksel modelinin oluşturulmasıdır. Üretim ortamlarında iş akış süreçlerinin

farklılığı veya kullanılan makinelerin spesifik özellikleri gibi durumlar nedeniyle bazı araştırmacılar problemin en iyi çözümünü bulmak için genel matematiksel modelinin geliştirilmesi üzerine çalışmışlardır.

Rathinasamy ve Raju (2009) bir otomobil imalat endüstrisinde farklı tipteki titreşim sönümleyicilerin işlenmesi problemini ele almışlardır. İki paralel ayrılmaz MAA aracılığıyla düzgün olmayan akış düzenine sahip nispeten farklı tipteki titreşim damperleri için en büyük tamamlanma zamanını enküçüklemeyi (C_{enb}) amaçlamışlardır. Bu amaçla MAA performansının simülasyonu için matematiksel bir model geliştirmişlerdir. Sarin (2013) lot akışı için bir KTDP Modeli (İAÇLAP) geliştirmiştir. 1. aşamada tek makinenin, 2. aşamada özdeş çoklu paralel makinelerin olduğu iki aşamalı MAA sisteminde her biri aynı öğelerden (ürünler) oluşan çoklu lotların işlendiği üretim ortamında C_{enb} ve bütün lotların toplam tamamlanma zamanını en küçüklemeyi amaçlamıştır. Laxmi, Mohan ve Madhusudanan, (2017) C_{enb} enküçüklemek için N aşamalı MAA lot akış problemini ele almışlardır. İlk $N-1$ aşamanın her birinde yalnızca bir makine; N aşamasında m tane makineden oluşan bir sistemi incelemişlerdir. Bu MAA için KTDP modeli geliştirmişlerdir. Meng vd. (2019) ilişkisiz paralel makinelerle MATÇ probleminde C_{enb} enküçüklemeyi amaçlamışlardır. 8 faklı karma tamsayılı model geliştirmiş ve bu modelleri sıra bağımlı hazırlık zamanları, beklemesiz (no-wait) ve engellemeli (blocking) MAA problemleri ile genişletmişlerdir. Elde ettikleri sonuçlara göre farklı modelleme fikirlerine bağlı KTDP modellerinin çok farklı performans gösterdiğini gözlemlemişlerdir. Fernandez-Viagas, Perez-Gonzalez ve M. Framinan (2019) klasik MATÇ problemine odaklanmışlar ve literatürde kullanılan farklı çözüm sunumlarını (ilk giren ilk çıkar, en kısa işlem süresi vb.) bazı sunumlar için yeniden belirlenen ana iş sıralama ve makine atama kuralları ile gözden geçirip sınıflandırmışlardır.

1.1.2. MATÇ Probleminin Çözümü için Önerilen Sezgisel Yöntemler

MATÇ problemi NP-zor yapıda olduğu için problem büyüdükçe çözümü hem uzun zaman almakta hem de optimal çözüme ulaşmak zorlaşmaktadır (Gupta, 1988). Bu nedenle literatürde, MATÇ probleminin çözümü için geliştirilen çok sayıda (meta)sezgisel algoritma bulunmaktadır.

Zandieh, FatemiGhomi ve MoattarHusseini (2006) sıra bağımlı hazırlık sürelerini kullanarak C_{enb} enküçüklemek amacıyla bir Bağışıklık Algoritması (BA)

önermişlerdir. Alaykýran, Engin ve Döyen (2007) Karınca Kolonisi Optimizasyonu (KKO) algoritması önermişlerdir. RezaAmin-Naseri ve AliBeheshti-Nia (2009) MAA'da C_{enb} enküçüklemek amacıyla paralel toplu iş çizelgeleme problemi üzerinde çalışmışlardır. Problemin çözümü için sırasıyla Johnson'ın kuralı, paralel makine çizelgeleme teknikleri ve kısıtlamalar teorisine dayanan üç algoritmanın birleşiminden oluşan bir 3 Boyutlu Genetik Algoritma (3BGA) sezgiselini geliştirmişlerdir.

Syam ve Al-Harkan (2010) literatürde klasik MATÇ probleminin çözümü için sıklıkla önerilen Genetik Algoritma (GA), Benzetim Tavlaması (BT) ve Tabu Araştırması (TAA) metasezgisellerinin karşılaştırmasını yapmışlardır. Xu ve Wang (2011) Diferansiyel Evrim (DE) algoritması önermişlerdir. Niu vd. (2012) MATÇ probleminde optimal iş sırasını bulmak ve toplam akış zamanını ($\sum F$) en aza indirmek için bir Kuantum Bağışıklık Algoritması (KBA) geliştirmişlerdir. Liao, Tjandradjaja ve Chung (2012) darboğaz bir aşamanın mevcut olduğu sistemde darboğaz aşamanın en yüksek seviyede kullanılmasını ve C_{enb} enküçüklemeyi amaçlamışlardır. Darboğaz aşamasından tam olarak yararlanmak için Parçacık Sürü Optimizasyonu (PSO) ve yerel optimumdan kaçınmak için BT kullanmışlardır. Yang J. (2013) iki aşamalı melez akış tipi üretim ortamında her aşamadaki işlem süresinin özelliklerine göre bazı özel durumlar için polinom zamanı çözüm prosedürleri geliştirmiştir. Johhnson's kuralı ve açgözlü tipi çizelgeleme kurallarını kullanarak 4 farklı özel durum için 4 farklı sezgisel önermiştir. Chung ve Liao (2013) IAIS algoritması olarak adlandırılan İmmünoglobulin Bazlı Yapay Bağışıklık Sistemi Algoritması (İBYBSA) geliştirmişlerdir. Elmi ve Topaloglu (2013) çoklu parça tiplerini, ilişkisiz paralel makineleri, çoklu robotları ve makine uygunluk kısıtlamalarını dikkate alan MAA hücrelerinin robotik çizelgeleme problemini ele almışlardır. Bu problemi çözmek için bir BT algoritması geliştirmişlerdir. Li, Pan ve Wang (2014) Kimyasal Reaksiyon Optimizasyonunu (KRO) ve Dağılım Algoritmasının Tahminini (DAT) birleştiren bir Hibrit Değişken Komşuluk Arama (HDKA) algoritması önermişlerdir. Marichelvam, Prabaharan ve Yang (2014) çok aşamalı MATÇ probleminde C_{enb} enküçüklemek amacıyla Guguk Kuşu Arama Algoritması (GKAA) önermişlerdir. Elmi ve Topaloglu (2014) birden fazla parça tipini, her aşamada farklı hıza sahip paralel makineleri, makine uygunluk kısıtlamalarını ve parçaları aşamalar arasında hareket ettirmek için tek bir taşıma robotunu dikkate alan engellemeli MATÇ'de robotik çizelgeleme problemini ele almışlardır. Problemin çözümü için BT algoritması geliştirmişlerdir. Lei ve Guo (2015) iki ajanlı MATÇ problemi üzerinde çalışmışlardır ve

problemin çözümü için KKZA önermişlerdir. Li ve Pan (2015) sınırlı depolama alanı bulunan MAA sisteminde C_{enb} enküçüklemek amacıyla Yapay Arı Kolonisi (YAK) ve TA sezgisellerini harmanlayarak yeni bir sezgisel (BYAKTAS) önermişlerdir.

Cui ve Gu (2015) MATÇ problemini C_{enb} enküçüklemek amacıyla vektör gösterimi ile modelleyip problemin çözümü için Geliştirilmiş Ayrık Yapay Arı Kolonisi (GAYAK) algoritması geliştirmişlerdir. Su ve Yi (2015) Genişletilmiş Giffler & Thompson algoritması ile Yön Seçimine Dayalı bir GA (YSGA) geliştirmişlerdir. Lei ve Zheng (2017) montaj operasyonlarında maksimum gecikmeyi en aza indirmeyi amaçlamışlardır. Problemin çözümünde, global arama yeteneğini geliştirmek için global arama ve komşuluk aramasını birleştirerek Komşuluk Araştırması Algoritması (GDKA) geliştirmişlerdir. Eskandari ve Hosseinzadeh (2014) yeniden girişli akışa sahip, ilişkisiz makinelerin bulunduğu sıra bağımlı hazırlık sürelerinin kullanıldığı MATÇ problemi üzerinde çalışmışlardır. Problemin çözümü için Değişken Komşuluk Araştırması (DKA) geliştirmişlerdir. Dios, Fernandez-Viagas ve Framinan (2018) bir plastik üreticisinden ilham alınan bazı aşamaların atlandığı MATÇ problemini ele almışlardır. Problemin çözüm yapısını etkileyecek iş sayısı, aşama sayısı vb. bir dizi durumu incelemişlerdir. Eksik operasyonların özel davranışlarından yararlanmak için, hem eksik operasyonları olan özel durum hem de genel MAA durumu için mevcut olanlarla karşılaştırılan bir dizi sevk kuralı ve iyileştirme sezgiselleri geliştirmişlerdir. Yu, Semeraro ve Matta (2018) ilişkisiz makineler ve makine uygunluk kısıtları ile MATÇ probleminde toplam gecikmeyi enküçüklemek ($\sum T$) amacıyla bir GA önermişlerdir. Li vd. (2021) ortak teslim tarihlerine sahip MATÇ probleminde $\sum T$ ve toplam erken bitirmeyi enküçüklemek için değiştirilmiş bir GA önermişlerdir. Wang, J. ve Wang, L. (2020) C_{enb} enküçüklemek amacıyla her bir MAA için iş sıralaması ve işlerin fabrikalara atanması gibi birbirine bağlı alt problemler içeren heterojen fabrikalarla dağıtılmış MATÇ problemini ele almışlardır. Problemin çözümü için İki Populasyonlu İşbirlikçi Memetik Algoritması (İPİMA) önermişlerdir. Costa, Fernandez-Viagas ve Framinan (2020) hazırlık işlemlerinin gerçekleşmesinden sorumlu sınırlı işgücünü ele almışlardır. Problemin çözümü için yeni bir kodlama yapısıyla çalışan bir TAA tarafından desteklenen bir Ayrık Geri İzleme Arama Algoritması (GİAOATA) önermişlerdir. Fernandez-Viagas, Costa ve Framinan (2020) birden fazla sunucu ve özdeş makinelere sahip MAA düzeninde C_{enb} enküçüklemeyi amaçlamışlardır.

Çok işlemcili MAA'da C_{enb} enküçüklemek için Wang, Chou ve Wu (2010) BT, (Jouglet, Oğuz ve Sevaux, 2009; Engin, Ceran ve Yilmaz, 2011) GA, Xu Y. vd. (2013) Karma Kurbağa Zıplaması Algoritması (KKZA), (Tseng ve Liao, 2008; Chou, 2013) Parçacık Sürü Optimizasyonu (PSO), Cai, Zhou ve Lei (2020) Dinamik Karma Kurbağa Zıplaması Algoritması (DDKZA) önermişlerdir. Harbaoui ve Khalfallah (2020) boşta kalma ve beklemenin olmadığı iki aşamalı MATÇ probleminin çözümü için TAA ve GA'yı karşılaştırmışlardır. Ozsoydan ve Sağir (2021) sıra bağımlı hazırlık süresi ile melez esnek akış atölyesi çizelgeleme probleminde C_{enb} enküçüklemek için YAA önermişlerdir. Önerdikleri algoritmayı üretim tesisinden alınan gerçek veriler ile test etmişlerdir. Daha sonra aynı verilerden yararlanılarak, geliştirilen algoritmayı literatürdeki sekiz farklı algoritma ile de karşılaştırmışlardır. Fernandez-Viagas (2022) amaç fonksiyonu hesaplama sürelerinin hızlandırılmasının çözüm algoritmalarının etkileri üzerinde önemli bir rolü olduğunu düşünmüşlerdir. Bu amaçla C_{enb} enküçüklemek için geleneksel Yinelemeli Açgözlü Algoritmaya (YAAwS) gömülü bir hızlandırma prosedürü önermişlerdir. Ayrıca problemde sıra bağımlı hazırlık sürelerini kullanmışlardır.

1.1.2.1. Belirsiz Süreler İçeren Çalışmalar

Gerçek hayat uygulamalarında üretim içerisinde meydana gelen makine arızaları, teslim tarihlerinin değişmesi, bir makinedeki bir işin işlem süresinin belirsizliği gibi durumlar teslim tarihinde belirsizliklere sebep olur. Chaari vd. (2011) işlem zamanlarının belirsiz olduğu işler ile MATÇ probleminin çözümüne odaklanmışlardır. Problemin çözümü için bir GA geliştirmişlerdir. Ebrahimi, FatemiGhomi ve Karimi (2014) belirsiz teslim süreleri ile sıra bağımlı hazırlık sürelerinin kullanıldığı MATÇ probleminde C_{enb} ve $\sum T$'yi enküçüklemek amacıyla Baskın Olmayan Sıralama Genetik Algoritması (BOSGAII) ve Çok Amaçlı Genetik Algoritma (ÇAGA) önermişlerdir. Wang, S vd. (2014) stokastik işlem zamanları altında MATÇ probleminde C_{enb} enküçüklemek için İki Fazlı Simülasyon Tabanlı Dağılım Algoritmasının Tahmini (DAİFSTT) ve Wang, Choi ve Qin (2014) Dağıtım Algoritmasının Siparişe Dayalı Tahmini (DASİPDT) algoritmalarını önermişlerdir. İki aşamalı beklemesiz MATÇ probleminde C_{enb} enküçüklemek amacıyla, Wang ve Liu (2013) GA önerirken, Hadda, Dridi ve Hajri-Gabouj (2014) Dal ve Sınır algoritması (DSA) önermişlerdir.

Wang , Liu ve Chu (2014) birinci aşamanın tek bir makine, ikinci aşamanın ise birkaç özdeş paralel makine içerdiği iki aşamalı beklemesiz MATÇ problemi için DSA

geliştirmişlerdir. Feng, Zheng ve Xu (2016) her bir işin işlem süresi ve olasılık dağılımının belirsiz olduğu, minimum maksimum pişmanlık modeli kullanılan iki aşamalı MATÇ probleminde C_{enb} enküçüklemeyi amaçlamışlardır. Problemin çözümü için Yinelemeli Gevşeme (YG) algoritması önermişlerdir. Shao, W, Shao, Z ve Pi (2020) işlerin birden fazla fabrikaya atandığı dağıtık MATÇ probleminde C_{enb} enküçüklemek amacıyla iki sezgisel önermişlerdir: Dağıtılmış NEH algoritması (DNA) ve Yinelemeli Açgözlü algoritma (YAA).

1.1.2.2. Gerçek Hayat Uygulamaları

Luo vd. (2010) müşteri siparişlerine göre her bir iş hattındaki üretim kaynaklarının sayısının farklı olduğu türünün tek örneği üretim şirketlerinde, tüm işlerin belirli bir zaman aralığında en az sayıda kaynakla tamamlanmasını sağlayan optimizasyon modeli üzerinde çalışmışlardır. Problemin çözümü için bir DSA önermişlerdir. Bożejko, Pempera ve Smutnicki (2013) otomatik üretim hatlarında oluşan gerçek bir problem üzerinde çalışmışlardır. Problemin çözümü için Küçük Tohumlu Paralel Tabu Arama (KTPTA) algoritması önermişlerdir. Costa, AntonioCappadonna ve Fichera (2013) ilişkisiz paralel makinelerin bulunduğu bir mikroelektronik üretim hattında, aynı türdeki işlerin çakışması, makinelerin kullanılamama süreleri ve iki hat arasındaki depolama alanlarında biriken işlerin bekleme süreleri kısıtları altında bir KTDP modeli geliştirmişlerdir. Problemin çözümü için Çift Kodlamalı Metasezgisel (ÇKM) ve Tek Kodlamalı GA (TKGA) önermişlerdir. Yalaoui vd. (2013) $\sum T$ enküçüklemek amacıyla önceden atanmış işler, özdeş olmayan paralel makineler ve belirli işler ile makineler arasındaki uyumsuzluk gibi belirli kısıtlamaları ele almışlardır. Problemin çözümü için bulanık mantık kontrolü ile birleştirilen veya birleştirilmeyen GA ve PSO algoritmalarına dayanan yaklaşık çözünürlük yöntemleri önermişlerdir (BMKBGADYÇ, BMKBPSODYÇY). Sun ve Yu (2015) demir-çelik üretiminin darboğazı olan çelik üretiminin sürekli döküm sürecinden kaynaklanan gerçek hayat MAA problemini incelemişlerdir. Problemin çözümü için üç gevşeme stratejisi (LGA) önermişlerdir. T.Lin ve Chen (2015) yarı iletken arka uç montaj tesisinde değişken talep ve arzdan kaynaklanan MATÇ probleminde $\sum F$ enküçüklemeyi amaçlamışlardır. Sorunun karmaşık ve stokastik yapısı nedeniyle, performans değerlendirmesi için bir simülasyon modeli, GA kullanan bir optimizasyon stratejisi ve Optimal Bilgi İşlem Bütçesi Tahsisi (OBİBT) yoluyla bir hızlandırma tekniği içeren bir simülasyon

optimizasyon yaklaşımı kullanmışlardır. Liu vd. (2017) tek ve seri üretimin olduğu, iki aşama arasında sınırlı depolama alanının bulunduğu ve makine uygunluk kısıtının olduğu temperli cam üretimi sisteminde 3 aşamalı MATÇ problemine odaklanmışlardır. Problemin çözümü için bir Yapıcı Sezgisel (YS), GA ve BT algoritmaları önermişlerdir. YiZhang ve Chen (2017) gerçek bir rotor atölyesinde makine uygunluk kısıtı ile yeniden girişli MATÇ problemi üzerinde odaklanmışlardır. Problemin çözümü için değiştirilmiş bir çaprazlama operatörü ile Ayrık Diferansiyel Evrim (ADEA) algoritması önermişlerdir. Wang S. vd. (2019) gerçek hayat cam-seramik üretim sürecinde iki aşamalı MATÇ problemine odaklanmışlardır. Hem paralel makine hem seri üretim makinesi içeren problemde C_{enb} ve toplam enerji tüketimini enküçüklemek amacıyla artırılmış bir Ɛ-kısıt (AEKKTPM) yönteminin benimsendiği KTDP modeli önermişlerdir. İki amaçlı TAA ve iki amaçlı KKO ile probleme uygun yapıcı sezgisel bir yöntem geliştirmişlerdir. Rolf vd. (2020) baskılı devre kartları üreten bir şirkette GA kullanarak sıra bağımlı hazırlık süreleriyle MATÇ problemini çözmüşlerdir. Aqil ve Allali (2020) döşeme ve kiriş üretiminde sıra bağımlı hazırlık süreleri ve ilişkisiz makinelerin olduğu MATÇ problemine odaklanmışlardır. Problemin çözümü için Yinelemeli Yerel Arama (YYAM) ve YAA metasezgisellerini önermişlerdir. Zhang vd. (2021) bir dizel motor montaj tesisinde sınırlı depolama alanı ve alternatif işlem rotaları ile MATÇ problemine odaklanmışlardır. Problemin çözümü için Ayrık Bir Balina Sürüsü Algoritması (ABSA) önermişlerdir. Lian vd. (2021) bir çelik üretim tesisinde enerji tüketimini de dikkate alarak ayrışmaya dayalı Geliştirilmiş Çok Amaçlı bir Evrimsel Algoritma (ADGÇAEA) önermişlerdir. Dabiri vd. (2021) bir fayans üretim tesisinde planlanan işlerin $\sum T$ enküçüklemek amacıyla iş reddini göz önünde bulundurarak MATÇ problemini ele almışlardır. Problemin çözümü için iki yenilikçi sezgisel önermişlerdir.

1.1.2.3. İnsan Faktörü ve Öğrenme Etkisi ile İlgili Yapılan Çalışmalar

Pargar ve Zandieh (2012) çalışmalarında sıra bağımlı hazırlık sürelerini dikkate alarak C_{enb} ve $\sum T$ enküçüklemek amacıyla Su Akışı Benzeri Algoritmasını (SABA) önermişlerdir. Behnamian ve Zandieh (2012) sıra bağımlı öğrenme etkisi altında $\sum T$ ve toplam teslim süresini enküçüklemek için PSO, BT ve DKA birleşimi ile oluşturulan bir Hibrit Benzetim Tavlaması (HBT) metasezgiseli geliştirmişlerdir. Pargar vd. (2018) sıra bağımlı hazırlık zamanlarının bulunduğu MATÇ probleminde C_{enb} ve $\sum T$ enküçüklemek amacıyla baskın olmayan sıralama ve BOSGA-II'den kalabalık mesafe

(crowding distance) gibi öğeleri SABA operatörleriyle birleştiren, Baskın Olmayan Sıralama ile Hibrit Su Akışı Benzeri Algoritması (BOSHSABA) ve Sıralama Kavramlarına Sahip Hibrit Su Akışı Benzeri Algoritması (SKSHSABA) metasezgisellerini önermişlerdir. Marichelvam, Geetha ve Tosun (2020) her aşamada özdeş paralel makineli çok aşamalı MATÇ probleminde insan faktörü etkisini dikkate almışlardır. Yetenek seviyesi, yaş faktörü, öğrenme ve unutma etkileri üzerinde çalışılmıştır. C_{enb} ve $\sum F$ enküçüklemeyi amaçlamışlardır. Problemin çözümü için Geliştirilmiş Parçacık Sürü Optimizasyonu (GPSO) algoritması önermişlerdir. Han vd. (2021) makinelere atanan işçilerin her bir makinedeki performansının farklı olduğu bir dökümhane tesisinde MATÇ problemine odaklanmışlardır. Problemin çözümü için yedi tane Sezgisel Kod Çözme ile Çok Amaçlı Evrimsel Algoritma (SKÇÇAEA) önermişlerdir. Literatür incelemesinin özeti Tablo (1.1)'de sunulmuştur.

Tablo 1.1: Literatür Araştırması Özeti

			Amaç Fonksiyonu				Kullanılan Süreler			MAA Çizelge Tipi						Makine Tipi					
Sn.	**Yıl**	**Yazarlar**	**Enk C_{enb}**	**Enk T_{enb}**	**Enk E_{enb}**	**Enk F**	**SBHS**	**BS**	**SİS**	**2MAA**	**3MAA**	**ÇAMAA**	**PMAA**	**MEAA**	**DMAA**	**ÖPM**	**İPM**	**ÖM**	**PPM**	**Önerilen Sezgisel**	**Problem Özellikleri ve Uygulama Alanı**
1	2006	Zandieh, FatemiGhomi ve MoattarHusseini	✓				✓													BA	
2	2007	Alaykýran, Engin ve Döyen	✓													✓				KKO	
3	2008	Tseng ve Liao	✓									✓				✓				PSO	Çok İşlemcili Görev
4	2009	Jouglet, Oğuz ve Sevaux	✓													✓				MA	Çok İşlemcili Görev
5	2009	Rathinasamy ve Raju	✓										✓							KTDP	Otomobil İmalat Sanayi
6	2009	RezaAmin-Naseri ve AliBeheshti-Nia	✓																	3BGA	
7	2010	Syam ve Al-Harkan	✓													✓				GA , BT , TAA	
8	2010	Wang, Chou ve Wu	✓													✓				BT	Çok İşlemcili Görev
9	2010	Luo vd.	✓			✓										✓				DSA	Türünün Tek Örneği Üretim (OKP), Müşteriye Özel Kapı ve Pencere İmalatı
10	2011	Chaari vd.	✓						✓							✓				GÇGA	
11	2011	Engin, Ceran ve Yilmaz	✓													✓				GA	Çok İşlemcili Görev
12	2011	Xu ve Wang	✓																	DE	
13	2012	Behnamian ve Zandieh		✓	✓		✓									✓				HBT	Pozisyon Bağımlı Öğrenme Etkisi

Tablo 1.1 (Devam): Literatür Araştırması Özeti

			Amaç Fonksiyonu				Kullanılan Süreler			MAA Çizelge Tipi						Makine Tipi					
Sn.	**Yıl**	**Yazarlar**	**Enk C_{enb}**	**Enk T_{enb}**	**Enk E_{enb}**	**Enk F**	**SBHS**	**BS**	**SİS**	**2MAA**	**3MAA**	**ÇAMAA**	**PMAA**	**MEAA**	**DMAA**	**ÖPM**	**İPM**	**ÖM**	**PPM**	**Önerilen Sezgisel**	**Problem Özellikleri ve Uygulama Alanı**
14	2012	Liao, Tjandradjaja, Chung	✓													✓				PSO	Darboğaz Aşama
15	2012	Pargar ve Zandieh	✓	✓			✓									✓				SABA	Öğrenme Etkisi
16	2012	Niu vd.	✓													✓				KA	
17	2013	Bożejko, Pempera ve Smutnicki	✓													✓				KTPTA	
18	2013	Chou	✓													✓				PSO	Çok İşlemcili Görev
19	2013	Chung ve Liao	✓																	İBYBSA	
20	2013	Costa, AntonioCappadonna, Fichera	✓					✓									✓			ÇKM, TKGA	Microelektronik İmalatı
21	2013	Elmi ve Topaloglu	✓														✓			BT	Birden Fazla Robot ve Makine Uygunluk Kısıtlı
22	2013	Sarin	✓													✓				İAÇLAP	Lot Akışı
23	2013	Wang ve Liu	✓							✓						✓				GA	
24	2013	Xu Y. vd.	✓													✓				KKZA	Çok İşlemcili Görev
25	2013	Yalaoui vd.		✓																BMKBGADYÇY , BMKBPSODYÇY	Önceden atanmış işler, Otomotiv Endüstrisi
26	2013	Yang J.	✓							✓								✓		S1	
27	2014	Elmi ve Topaloglu	✓																	BT	Robotik Çizelgeleme

Tablo 1.1 (Devam): Literatür Araştırması Özeti

			Amaç Fonksiyonu				Kullanılan Süreler			MAA Çizelge Tipi						Makine Tipi					
Sn.	**Yıl**	**Yazarlar**	**Enk C_{enb}**	**Enk T_{enb}**	**Enk E_{enb}**	**Enk F**	**SBHS**	**BS**	**SİS**	**2MAA**	**3MAA**	**ÇAMAA**	**PMAA**	**MEAA**	**DMAA**	**ÖPM**	**İPM**	**ÖM**	**PPM**	**Önerilen Sezgisel**	**Problem Özellikleri ve Uygulama Alanı**
28	2014	Ebrahimi, FatemiGhomi, Karimi	✓	✓			✓									✓				BOSGAII , ÇAGA	
29	2014	Hadda, Dridi ve Hajri-Gabouj	✓							✓								✓		DSA	
30	2014	Marichelvam, Prabaharan ve Yang	✓									✓				✓				GKAA	Mobilya İmalat Şirketi
31	2014	Eskandari ve Hosseinzadeh	✓				✓										✓			DKA	Yeniden Giriş Akışı
32	2014	Wang S. vd	✓						✓											DAİFSTT	
33	2014	Wang S. vd.	✓													✓				DSA	
34	2014	Wang, Choi ve Qin	✓						✓							✓				DASİPDT	
35	2015	Cui ve Gu	✓													✓				GAYAK	
36	2015	Lei ve Guo	✓	✓																KKZA	İki Ajanlı Sistem
37	2015	Li ve Pan	✓														✓			BYAKTAS	Sınırlı depolama alanı
38	2015	Su ve Yi	✓													✓				YSGA	
39	2015	Sun ve Yu	✓	✓																LGA	Demir ve Çelik Üretimi
40	2015	T.Lin ve Chen				✓										✓	✓			OBİBT	Yarı İletken Arka Uç Montaj Tesisi
41	2016	Feng, Zheng ve Xu	✓							✓						✓				YG	Güçlü Çizelgeleme

Tablo 1.1 (Devam): Literatür Araştırması Özeti

			Amaç Fonksiyonu				Kullanılan Süreler			MAA Çizelge Tipi						Makine Tipi					
Sn.	**Yıl**	**Yazarlar**	**Enk C_{enb}**	**Enk T_{enb}**	**Enk E_{enb}**	**Enk F**	**SBHS**	**BS**	**SİS**	**2MAA**	**3MAA**	**ÇAMAA**	**PMAA**	**MEAA**	**DMAA**	**ÖPM**	**İPM**	**ÖM**	**PPM**	**Önerilen Sezgisel**	**Problem Özellikleri ve Uygulama Alanı**
42	2017	Laxmi, Mohan ve Madhusudanan	✓													✓				KTDP	Lot Akışı
43	2017	Liu vd.	✓								✓									YS , GA , BT	Sınırlı depolama alanı ve makine uygunlulk kısıtı, Temperli Cam Üretim Tesisi
44	2017	Lei ve Zheng	✓	✓													✓			GDKA	Montaj Operasyonları
45	2018	Dios, Fernandez-Viagas ve Framinan	✓													✓				NEH , RAJ	Eksik İşlemler
46	2018	Pargar vd.	✓	✓			✓									✓				BOSHSABA , SKSHSABA	Öğrenme Etkisi
47	2018	Yu, Semeraro ve Matta		✓													✓			GA	
48	2018	YiZhang ve Chen		✓												✓				ADEA	Rotor Atölyesi
49	2019	Fernandez-Viagas, Perez-Gonzalez ve Framinan	✓													✓				KTDP	Yarıaktif Çizelgeleme
50	2019	Meng vd.	✓				✓										✓			KTDP	
51	2019	Wang S. vd.	✓							✓									✓	TAA , KKO	Cam-Seramik Üretim Tesisi
52	2020	Aqil ve Allali		✓			✓										✓			YYAM , YAA	Döşeme ve Kiriş Üretimi

Tablo 1.1 (Devam): Literatür Araştırması Özeti

			Amaç Fonksiyonu				Kullanılan Süreler			MAA Çizelge Tipi					Makine Tipi						
Sn.	**Yıl**	**Yazarlar**	**Enk C_{enb}**	**Enk T_{enb}**	**Enk E_{enb}**	**Enk F**	**SBHS**	**BS**	**SİS**	**2MAA**	**3MAA**	**ÇAMAA**	**PMAA**	**MEAA**	**DMAA**	**ÖPM**	**İPM**	**ÖM**	**PPM**	**Önerilen Sezgisel**	**Problem Özellikleri ve Uygulama Alanı**
53	2020	Cai, Zhou ve Lei	✓												✓	✓				DKKZA	Çok İşlemcili Görev
54	2020	Costa, Fernandez-Viagas ve Framinan	✓				✓			✓						✓				GİAOATA	Sınırlı İşgücü
55	2020	Fernandez-Viagas, Costa ve Framinan	✓													✓				BFS	Birden Çok Sunucu
56	2020	Harbaoui ve Khalfallah	✓							✓						✓				TAA	Boşta Kalma Yok, Bekleme Yok
57	2020	Marichelvam, Geetha ve Tosun	✓			✓						✓				✓				GPSO	Beceri, yaş, öğrenme ve unutkanlık faktörleri
58	2020	Rolf vd.	✓	✓			✓									✓				GA	PCB Montaj Şirketi
59	2020	Shao, W., Shao, Z., ve Pi	✓												✓	✓				DNA , YAA	
60	2020	Wang, J. ve Wang, L.	✓												✓	✓				İPİMA	Heterojen Fabrikalar
61	2021	Dabiri vd.		✓												✓				S1,S2	İş reddi, Fayans Üretim Tesisi
62	2021	Han vd.	✓	✓																SKÇÇAEA	İşgücü kısıtı, Dökümhane Tesisi
63	2021	Li vd.		✓	✓											✓				GA	Ortak Teslim Tarihi
64	2021	Lian vd.	✓																	ADGÇAEA	Çelik Üretim Tesisi

Tablo 1.1 (Devam): Literatür Araştırması Özeti

			Amaç Fonksiyonu				Kullanılan Süreler			MAA Çizelge Tipi						Makine Tipi					
Sn.	**Yıl**	**Yazarlar**	**Enk C_{enb}**	**Enk T_{enb}**	**Enk E_{enb}**	**Enk F**	**SBHS**	**BS**	**SİS**	**2MAA**	**3MAA**	**ÇAMAA**	**PMAA**	**MEAA**	**DMAA**	**ÖPM**	**İPM**	**ÖM**	**PPM**	**Önerilen Sezgisel**	**Problem Özellikleri ve Uygulama Alanı**
65	2021	Ozsoydan ve Sağir	✓				✓							✓		✓				YAA	Üretim Tesisi
66	2021	Zhang vd.	✓																	ABSA	Sınırlı depolama alanı ve alternatif işlem rotaları, Dizel Motor Montaj Tesisi
67	2022	Fernandez-Viagas	✓													✓				YAAwS	

SBHS: Sıra Bağımlı Hazırlık Süresi, **BS:** Bekleme Süresi, **SİS**: Stokastik İşlem Süresi

MAA: Melez Akış Atölyesi, **2MAA**: İki Aşamalı MAA, **3MAA**: Üç Aşamalı MAA, **ÇAMAA**: Çok Aşamalı MAA, **PMAA**: Paralel MAA, **DMAA**: Dağıtılmış MAA, **MEAA**: Melez Esnek Akış Atölyesi

ÖPM: Özdeş Paralel Makine, İPM: İlişkisiz Paralel Makine, **ÖM**: Özel Makine , **PPM**: Paralel Parti Makinesi

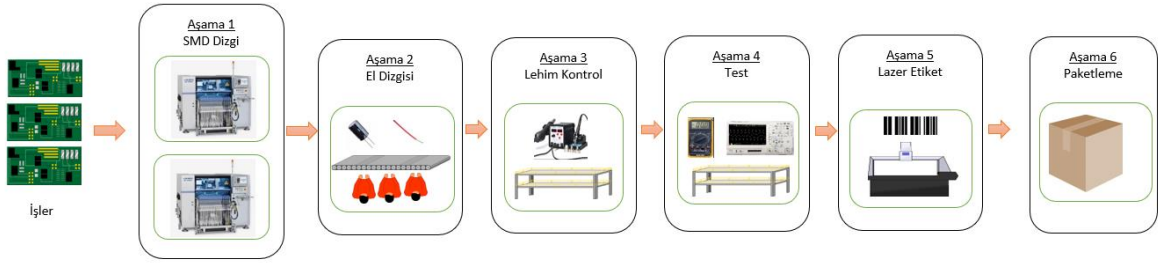
C_{enb}: En Büyük Toplam Tamamlanma Zamanı, **T_{enb}**=En Büyük Toplam Gecikme Süresi, **E_{enb}**: En Büyük Toplam Erken Bitiş Süresi, **F**: Toplam Akış Zamanı

İKİNCİ BÖLÜM
METODOLOJİ

2.1. PROBLEMİN TANITIMI

Bu çalışmada acil aydınlatma ürünleri üreten bir firmanın üretim sistemi ele alınmış ve bu sistemin ihtiyaçlarını karşılayacak etkili bir çizelgeleme yapılması amaçlanmıştır. Bu firmada imalatı yapılan ürünler gelen siparişlere göre farklı tasarım ve özelliklere sahiptir. Firma proje bazlı üretim yaptığı için gelen siparişler çoğunlukla farklı özelliklere sahiptir. Bir ürünün imalatı eğer aynı ürün daha önce üretilmediyse, elektronik devre kartının tasarım süreciyle başlamaktadır. Tasarım süreci bu devre kartlarına dizilecek elektronik komponentlerin özellikleri, sayısı ve yerleşimini kapsamaktadır. Tasarım süreci tamamlandıktan sonra ürün ilk aşamada bulunan SMD (Surface Mount Technology) dizgi makinasında işleme başlar. SMD teknolojisi temelde elektronik kartlar üzerine devre elemanlarının lehimleme işleminin çok hızlı bir şekilde yapılmasını sağlamaktadır. Ancak, kart tasarımı ve tasarımın makineye kodlanması süreçleri büyük bir hazırlık süresi gerektirmektedir. Firma üretimi hızlandırmak ve daha fazla ürün üretebilmek için ilk aşamada iki adet özdeş paralel SMD makinası bulunmasını tercih etmiştir. SMD makinasından çıkan ürünler ikinci aşamada el dizgisi işleminden geçmektedir. Bu aşamada SMD makinası ile lehimlenemeyen büyük devre parçaları, kablolar veya takılabilir komponentlerin devre kartına montajı yapılmaktadır. Üçüncü aşamada devre kartı üzerine montajı yapılan komponentlerin lehim kontrolü yapılmaktadır. Dördüncü aşama test aşamasıdır, burada devre kartlarının elektrik değerlerinin ölçümünün yapıldığı test cihazları bulunmaktadır. Beşinci aşama etiketleme aşamasıdır, lazer etiketleme makinası kullanılarak hazırlanan kartların üzerlerine ürün bilgileri basılmaktadır. Altıncı aşama, kartların üzerine gövdelerin takılması ve paketlenmesi ile ürünlerin tamamen hazır olduğu aşamadır. Bu atölye bütün işlerin aynı işlem rotasını izlemesi, ilk aşamada 2 özdeş paralel makine bulunması ve diğer aşamalarda tek makine bulunması sebebiyle MAA problem sınıfına girmektedir. Yukarıda bahsedilen imalat ortamının yapısı Şekil (2.1)'de sunulmuştur.

Şekil 2.1: İmalat Ortamının Yapısı



Ele alınan imalat ortamı yapısında bulundurduğu bazı özellikler sebebiyle Melez akış tipi çizelgeleme (MATÇ) problemine uyarlanırken kullanılan bazı varsayımlar aşağıda belirtilmiştir:

İkinci aşamadan itibaren tüm işler operatörlerin el becerisine dayalı olduğu için bu aşamalardaki operatörler farklı hızlara sahip esnek makinalar olarak varsayılmıştır ve her bir aşama kendi operatörü tarafından yönetilmektedir. Ayrıca bu atölyede sipariş miktarları fazladır, ürünler parti olarak üretilmektedir, fakat bu çalışmada ürün kavramı tek bir ürün olarak değerlendirilmiştir ve bir sipariş işleme alındığında siparişi oluşturan parçaların tamamı işlenmeden diğer bir siparişe geçilmediği varsayılmıştır. Bu nedenle çizelgelemede kullanılan süreler bir adetin süresi göz önüne alınarak sipariş miktarı ile çarpılmıştır. Örneğin A ürününün sipariş miktarı 100 adet ve bu ürünün 1 adetinin hazırlık süresi 5 birim ise, bu ürünün hazırlık süresi birim hazırlık süresi * sipariş miktarı yani 5*100=500 birim olarak çalışılmıştır.

2.2. ÖĞRENME ETKİSİ

TP Wright (1936) uçak endüstrisinde uçakları monte etmek için harcanan adam-saat sayısının daha fazla uçak üretildikçe azalması gözlemine dayanarak öğrenmenin gerçekleştiği hızın rastgele olmadığını ve dünyada uçak inşa etmek için ne kadar emek süresi gerekeceğini doğru bir şekilde tahmin etmenin aslında mümkün olduğunu ortaya koymuştur. Heizer ve Render (2001), işgücü tahmininde, maliyet ve bütçe hesaplarında, dışarıdan satın almalarda, şirket performansının stratejik gelişiminin tespiti vb. birçok uygulamalarda öğrenme eğrilerinin yararlı olduğunu ifade etmişlerdir.

Öğrenme etkisi çizelgelemede ilk kez Biskup (1999) tarafından kullanılmıştır. Son yıllarda literatürde çizelgeleme uygulamalarında öğrenme etkisi ve insan faktörü üzerine çalışmalar artmaya başlamıştır. Yapılan çalışmalarda çoğunlukla öğrenme etkisinin sıra bağımlı hazırlık süresi üzerindeki etkileri araştırılmıştır.

Wright, spesifik öğrenme eğrisi etkisini, kümülatif çıktı her iki katına çıktığında birim başına kümülatif ortalama sürenin sabit bir yüzde ile azalması şeklinde tanımlamıştır. Uçak endüstrisinde bu öğrenme oranının genellikle %80 civarında olduğu görülürken, farklı endüstrilerde başka oranlar da ortaya çıkmaktadır.

Literatürde sıklıkla kullanılan öğrenme oranları % 70'ten başlamaktadır. Üretim arttıkça çalışanların harcadıkları emek artacağından belirli bir zaman sonra öğrenme yavaşlar ve durur. Öğrenme oranının %100 olması artık öğrenmenin durduğu anlamına gelir. Öğrenme etkisinin %70-%90 ve %100 olduğu durumlarda, makinelerin 2., 3. ve 4. pozisyonlarındaki öğrenme etkisinin hesabı Tablo (2.1)'de sunulmuştur.

Tablo 2.1: Öğrenme Etkisinin Hesaplanması

Öğrenme Oranı	Öğrenme indisi (α)	Makinelerin farklı pozisyonları için öğrenme etkileri		
		r=2	r=3	r=4
%70	$\alpha=\log_2 0.70=-0.514$	$r^{\alpha}=2^{-0.514}=0.7$	0.56	0.49
%90	$\alpha=\log_2 0.90=-0.152$	$r^{\alpha}=2^{-0.152}=0.9$	0.84	0.81
%100	$\alpha=\log_2 1=0$	$r^{\alpha}=2^{0}=1$	1	1

Literatürdeki çizelgeleme problemlerinden farklı olarak bu çalışmada ele alınan imalat ortamının operatör bağımlı yapısı ve ürünlere uygulanan işlemlerin benzerlik göstermesi sebebiyle hazırlık sürelerine ek olarak işlem süreleri üzerinde de öğrenme etkisi olduğu düşünülmüştür. Ayrıca bu firmada sipariş miktarlarının fazla olması sebebiyle, her bir aşamada işlemi tamamlanan ürünler ara stok alanlarında tutulmakta ve bir sonraki aşamaya aktarılmaktadır. Eğer sipariş miktarı çok fazla ise ara stok alanlarında tutulan ürünlerin bir sonraki makineye taşınması için birden fazla taşıma işlemi gerçekleştirilebilir. Tüm bu nedenlerden dolayı bu hazırlık ve işlem sürelerine öğrenme etkisinin dâhil edildiği, taşıma süresinin ayrıca hesaplandığı bir çizelgeleme oluşturulması hedeflenmiştir. Bu çalışmada, öğrenme oranı, %80 olarak ele alınmıştır.

2.3. KARMA TAMSAYILI DOĞRUSAL PROGRAMLAMA MODELİ

Matematiksel modelleme, birçok uygulamada değerlidir ve problemin çözümü için hassasiyet ve strateji verir. Ek olarak, modellenen sistemin sistematik olarak anlaşılmasını sağlar. Ele alınan problemin verilen kısıtlar altında amaçları dikkate alınarak matematiksel olarak tanımlanması, problemin genel çerçevesini oluşturmak için önemlidir. Öncelikle bu çalışmada ele alınan sıra bağımlı hazırlık süreli ve taşıma süreli, öğrenme etkisi tabanlı MATÇ problemi için C_{enb} ve $\sum T$ enküçüklenmesi amaçlanmıştır.

Önerilen MATÇ probleminin matematiksel modeli, Pargar vd. (2018) çalışmasındaki modele, öğrenme etkili işlem süresi ve taşıma süresi kısıtları eklenerek yeniden düzenlenmiştir.

Gerçek hayat üretim sistemlerindeki makine arızaları, sipariş iptalleri, yetersiz/yanlış stok durumları, işgücü problemleri vb. aksaklılarını çizelgelemeye yansıtmak oldukça zor ve karmaşıktır bu nedenle birçok araştırmacı çizelge için model oluştururken bazı varsayımlar altında çalışmalarını yürütmektedir. Bu çalışmada ele alınan varsayımlar aşağıda listelenmiştir.

- Makineler çalışır durumdadır,
- Tüm işler bağımsızdır ve sıfır anında işlenmeye hazırdır,
- Çizelgeyi oluşturan işler için gerekli olan malzeme stokları her zaman mevcuttur,
- Tüm aşamalar arasında sınırsız ara stok alanı vardır,
- İşler tüm aşamaları ziyaret eder, eğer atlanan bir aşama varsa bu aşamadaki işlem süresi sıfıra eşittir,
- Bir işin işleneceği makine dolu ise iş makine boşalana kadar süresiz olarak bekler.

Bu varsayımlar altında oluşturulan matematiksel modelde kullanılan indisler, parametreler, karar değişkenleri ve açıklamaları aşağıdaki gibidir;

Notasyonlar:

I: İş kümesi, i=0,..,n,

J: İş kümesi, j=1,..,n,

M: Makine kümesi, m=1,…,m_t

R: Pozisyon kümesi, r=0,..,n,

T: Aşama (stage) kümesi, t=0,..,g,

Parametreler:

m_t**:** t aşamasındaki paralel makine sayısı

P_{jtm}**:** t aşamasındaki m makinesinde j işinin işlem süresi

PLE_{jrtm}**:** t aşamasındaki m makinesinde r pozisyonundaki j işinin işlem süresi

S_{ijtm}: *t* aşamasındaki *m* makinesinde *i* işinden hemen sonra başlayan *j* işinin hazırlık süresi

SLE_{ijrtm}: *t* aşamasındaki *m* makinesinde *r* pozisyonunda *i* işinden hemen sonra başlayan *j* işinin hazırlık süresi

RT_{jt}: *t-1* aşamasından *t* aşamasına *i* işinin taşıma süresi

d_j: *j* işinin teslim süresi

I: Öğrenme oranı

a: Öğrenme etkisi ($\alpha = log_2(l) \leq 0$)

r^a: *r* inci pozisyondaki işin öğrenme etkisi, r=1,….,n_m (n_m= aşamalarda makinelere atanan işlerin sayısı)

BN: Pozitif büyük bir sayı

Karar değişkenleri:

X_{ijrtm}: 1, eğer *i* işinden sonra *j* işi, *t* aşamasındaki *m* makinesinin *r* inci pozisyonuna atanırsa; 0, diğer durumlarda

C_{jt}: *t* aşamasındaki *j* işinin tamamlanma süresi

C_{enb}: En büyük tamamlanma süresi

T_j: *j* işinin gecikme süresi

$$enk\ C_{enb} \tag{2.1}$$

$$enk \sum T_j \tag{2.2}$$

$$\sum_{m=1}^{m_t} \sum_{r=1}^{n} \sum_{i=0, i \neq j}^{n} x_{ijrmt} = 1,\ \forall\ t, j \tag{2.3}$$

$$\sum_{i=0}^{n} \sum_{j=1, i \neq j}^{n} x_{ijrmt} \leq 1,\ \forall\ r,m,t \tag{2.4}$$

$$\sum_{i=0}^{n} \sum_{j=1, j \neq i}^{n} x_{ijrmt} \leq \sum_{i=0}^{n} \sum_{j=1, i \neq j}^{n} x_{ij(r-1)mt},\ \forall\ r \geq 2,m,t \tag{2.5}$$

$$\sum_{i=0}^{n} x_{ij(r-1)mt} \geq \sum_{k=1, k \neq j}^{n} x_{jkrmt},\ \forall\ j,r \geq 2,m,t \tag{2.6}$$

$$PLE_{jrtm} = P_{jtm} * r^a,\ \forall\ j,t,r \tag{2.7}$$

$$SLE_{ijrtm} = S_{ijtm} * r^{a},\ \forall\, i,j,i \neq j,t,r \succ 1 \tag{2.8}$$

$$C_{jt} - C_{it} + BN * (1 - x_{ijrmt}) \geq PLE_{jrtm} + SLE_{ijtm} + RT_{jt},\ \forall\, i,j,i \neq j,m,t,r \tag{2.9}$$

$$C_{jt} - C_{j(t-1)} + BN * (1 - x_{ijrmt}) \geq PLE_{jrtm} + SLE_{ijtm} + RT_{jt},\ \forall\, i,j,i \neq j,m,r,t \succ 1 \tag{2.10}$$

$$T_j \geq C_{jt} - d_j,\ \forall\, j,t \tag{2.11}$$

$$C_{enb} \geq C_{jt},\ \forall\, j,t \tag{2.12}$$

$$x_{ij1mt} = 0,\ \forall\, i \geq 1, j,i \neq j,m,t \tag{2.13}$$

$$x_{0jrmt} = 0,\ \forall\, r \geq 2, j,i \neq j,m,t \tag{2.14}$$

$$x_{ijrmt} \in \{0,1\},\ \forall\, i,j,i \neq j,r,m,t \tag{2.15}$$

$$C_{jt} \geq 0,\ \forall\, j,t \tag{2.16}$$

$$T_j \geq 0,\ \forall\, j \tag{2.17}$$

Bu modelde, Eşitlik (2.1), en büyük tamamlanma süresini enküçüklemeyi garantilerken, Eşitlik (2.2) toplam gecikmeyi enküçüklemeyi sağlamaktadır. Eşitlik (2.3), her bir işin her bir aşamada mevcut olan makinaların birinde sadece bir pozisyonda çizelgelenmesini sağlar. Eşitlik (2.4), her bir işin her bir aşamada başka bir işten sonra bir kez işlem görmesini temin eder. Eşitlik (2.5), makinalardaki pozisyonları sırayla doldurur. Eşitlik (2.5)'in tamamlayanı olan Eşitlik (2.6), bir makinada işlem gören her bir işin bir öncülünün ve ardılının olmasını sağlar ve işlerin iyi bir şekilde sıralanmasını garantileyen bir akış dengesi kısıtıdır. Eşitlik (2.7), her bir aşamada her bir makinadaki işlerin pozisyonlara göre öğrenme faktörünü dâhil ederek işlem sürelerini güncellemeyi garantilemektedir. Eşitlik (2.8) işler arasında pozisyona bağlı olarak hazırlık sürelerine öğrenme faktörünü dahil ederek hazırlık sürelerini günceller. Eşitlik (2.9)-(2.10), işlerin öğrenme etkili işlem süresini (güncellenmiş işlem süresi), güncellenmiş hazırlık süresini ve aşama geçişleri esnasında oluşan taşıma süresini dikkate alarak her bir aşamadaki her bir işin tamamlanma süresini hesaplamaktadır. Eşitlik (2.11)-(2.12), en büyük tamamlanma süresini ve her bir işin gecikmesini hesaplamaktadır. Eşitlik (2.13)-(2.14), her makinenin ilk pozisyonuna 0 işinin gelmesi ve böylece diğer pozisyonlara 0 işinin gelmemesini temin etmektedir. Eşitlik (2.15)-(2.17), karar değişkenlerini belirtir.

Çok amaçlı problemlerin çözümünde her bir amacı ayrı ayrı değerlendirmek çok fazla tercih edilen bir durum değildir. Karmaşık gerçek hayat problemlerinde her bir amaç tek bir sistemde aynı anda değerlendirilmesi gereken önemli problemlerdir, bu nedenle her bir amacı aynı anda makul ölçülerde sağlayan çözümler elde etmek daha önemli olmaktadır. Literatürde birden fazla amacı birleştirip tek bir amaç fonksiyonuna dönüştüren farklı yöntemler geliştirilmiştir. Bu yöntemlere genel olarak skalerleştirme yöntemleri adı verilmektedir, bunlardan birkaçı, Ağırlıklı-Toplam (ATY), ε- Kısıt, Konik Skalerleştirme (KSY), Pascoletti-Serafini Skalerleştirme, Benson Skalerleştirme ve Tchebycheff Skalerleştirme (TSY) yöntemleridir. Bu çalışmada ele alınan problem üç skalerleştirme yöntemi (ATY, KSY ve TSY) ile çözülerek elde edilen sonuçlar kıyaslanmıştır.

ATY, pareto etkin çözümleri elde etmede en bilinen yöntemlerden biridir (Miettinen, 2014; Ehrgott ve Schöbel, 2014). Eşitlik (2.18)'de verildiği gibi, amaç ağırlıkları amaç değerleriyle çarpılıp toplanarak çok amaçlı değer tek amaca dönüştürülür. Bu toplam problemin türüne göre enküçüklenmesi veya enbüyüklenmesi sağlanacak tek bir amaç fonksiyonunu verir. ATY'nin matematiksel modeli verilen probleme uyarlanması Eşitlik (2.18)'de formüle edilmiştir.

$$enk\ \mathrm{ATY}(x) = w_1 * C_{enb} + w_2 * \sum T_j \tag{2.18}$$

Burada w_1 ve w_2, C_{enb} ve $\sum T$ için önem ağırlıklarını temsil etmektedir. $w_1 + w_2 = 1$ olduğu varsayılarak $w_1, w_2 \geq 0$ olacaktır ve Eşitlik (2.3) - (2.17) arasındaki kısıtların tümü sağlanmalıdır.

KSY, Gasimov (2001) tarafından geliştirilen, pareto etkin değerlerin bulunması için, ağırlıklı skalerizasyonda kullanılan hiper-düzlemler yerine koniler kullanılarak problemin görüntü kümesinin desteklenmesine dayanmaktadır. Bu yöntem dışbükey olmayan çok amaçlı optimizasyon problemlerinde klasik optimizasyon yaklaşımlarına göre daha avantajlıdır ve kullanım alanı daha geniştir. KSY'nin matematiksel modeli verilen probleme uyarlanması Eşitlik (2.19)'da formüle edilmiştir.

$$enk\ \mathrm{KSY}(x) = \left[w_1 (C_{enb} - Rf_1) + w_2 \left(\sum T_j - Rf_2 \right) + \alpha \left(\left| C_{enb} - Rf_1 \right| + \left| \sum T_j - Rf_2 \right| \right) \right] \tag{2.19}$$

Burada, α destekleyici bir koninin apikal açısını değiştirmek için kullanılır ve $0 \leq \alpha \leq enk\{w_1, w_2\}$ olacak şekilde belirlenir. Bu çalışmada $\alpha = enk\{w_1, w_2\} - 0{,}01$

olarak hesaplanmıştır. Rf_1 ve Rf_2 , sırasıyla C_{enb} ve $\sum T$ için belirli bir aralıktan keyfi olarak bir karar verici tarafından belirlenen sabit referans noktalarıdır. Eşitlik (2.3)-(2.17) arasındaki kısıtların tümü sağlanmalıdır.

Çok amaçlı optimizasyon problemlerinde birden fazla amaç için uygun ağırlıkları belirlemek zordur. Uygun olmayan bir ağırlık çözüm kümesinin kalitesini düşürebilir ve Pareto optimal kümesinden sapabilir. Doğrusal olmayan bir skalerizasyon yöntemi olarak TSY, çözümleri daha geniş bir alanda etkili bir şekilde arar, dışbükey bölgelerden ziyade çözüm alanı ve yüzey şekli ne olursa olsun Pareto optimal çözümlerini elde eder. TSY'nin iki amaçlı probleme uyarlanmış formülü Eşitlik (2.20)'de gösterilmiştir.

$$\begin{aligned} enk\ \mathrm{TSY}(x) = \ enb\ & \left[w_1\left(C_{enb} - Rf_1\right), w_2\left(\sum T_j - Rf_2\right)\right] + \\ & \left[\rho\left(w_1\left(C_{enb} - Rf_1\right) + w_2\left(\sum T_j - Rf_2\right)\right)\right] \end{aligned} \tag{2.20}$$

Burada, C_{enb} ve $\sum T$ amaçları, Rf_1 ve Rf_2 amaç değerleri için bir karar verici tarafından tercih edilen referans noktalarını, $\rho > 0$ ise küçük bir sayıyı temsil etmektedir. Bu çalışmada ρ değeri, 0,01 olarak ele alınmıştır. Eşitlik (2.3)-(2.17) arasındaki kısıtların tümü sağlanmalıdır.

Modelin, istenen amaçlar doğrultusunda verilen kısıtlar altında nasıl çalıştığını ve çalışma sonucunda oluşan çizelgelemeyi göstermek için küçük-ölçekli bir problem üretilmiştir. (Tablo (2.2)-(2.5)). Örnek problem, üç iş (n=3), dört aşama (g=4) ve aşama 1'de iki makine ($t_1 \rightarrow [m_1, m_2]$), diğer aşamalarda birer makine ($t_2 \rightarrow [m_3]$); $t_3 \rightarrow [m_4]$; $t_4 \rightarrow [m_5]$) oluşmaktadır.

Tablo 2.2: Örnek Problemin İşlem Süreleri

P_{jtm}	T1		T2	T3	T4
	M1	M2	M3	M4	M5
i1	10	10	20	15	35
i2	15	15	25	40	45
i3	20	20	30	25	50

Tablo 2.3: Örnek Problemin Sıra Bağımlı Hazırlık Süreleri

S_{ijtm}	T1-M1			T1-M2			T2-M3			T3-M4			T4-M5		
	i1	i2	i3	i1	i2	i3	i1	i2	i3	i1	i2	i3	i1	i2	i3
i1	-	25	30	-	25	30	-	4	6	-	4	6	-	4	6
i2	20	-	30	20	-	30	2	-	6	2	-	6	2	-	6
i3	20	25	-	20	25	-	2	4	-	2	4	-	2	4	-

Tablo 2.4: Örnek Problemin Aşamalar Arası Taşıma Süreleri

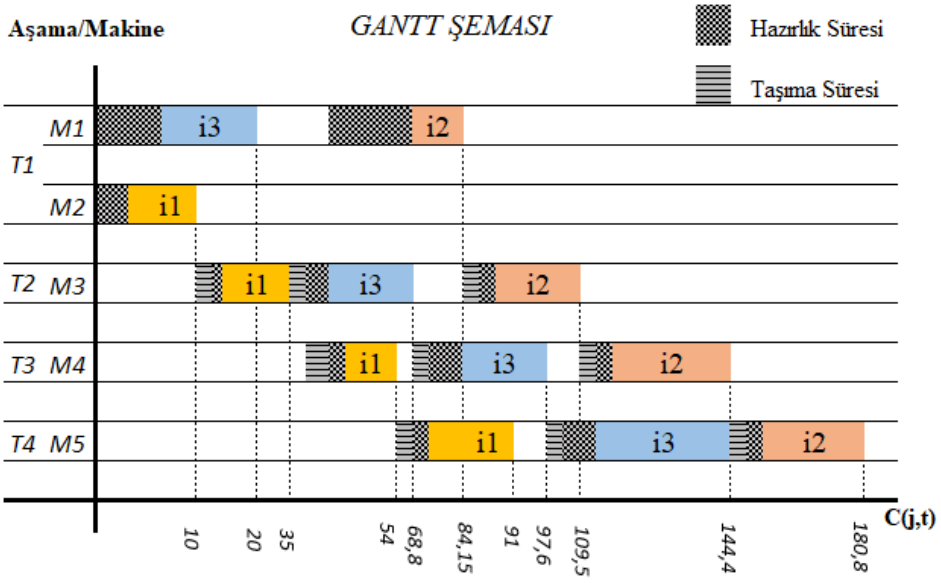
RT_{jt}	T1	T2	T3	T4
i1	0	5	4	2
i2	0	5	4	2
i3	0	5	4	2

Tablo 2.5: Örnek Problemin Teslim Süreleri

İşler	Teslim Süresi (d_j)
i1	23
i2	20
i3	19

Model ağırlıklı toplam yöntemi ile çözülmüştür ve her iki amacın ağırlıkları 0,5 olarak girilmiştir. Modelin çözümü için, Lingo 11.0 optimizasyon programı kullanılmıştır ve ortalama 4 saniyede global optimuma ulaşılmıştır. Verilen örnek problemin uygunluk (fitness) değeri, 267,493 br hesaplanmıştır. C_{enb} amacı 180,8 br olarak hesaplanırken, $\sum T$ amacı, 354,2 br bulunmuştur. İşlerin makinelerdeki çizelgelemesi, birinci aşamada m_1'de i3-i2 ve m_2'de i1 olarak çizelgelenmiştir. Diğer aşamalardaki makinelerde, işler i1-i3-i2 şeklinde çizelgelenmiştir. Sonuç olarak, Şekil (2.2)'de sunulan Gantt şeması oluşmuştur. Bu Gantt şemasına göre, birinci işin (i1) tamamlanma süresi, 91 br; ikinci işin (i2) tamamlanma süresi, 180,8 br ve son olarak üçüncü işin (i3) tamamlanma süresi 144,4 br'dir. Verilen teslim sürelerine göre (dj) birinci işin gecikme süresi 68 br, ikinci işin gecikme süresi 160,8 ve üçüncü işin gecikme süresi 125,4 br olarak hesaplanmıştır.

Şekil 2.2: Örnek Problemin Gantt Şeması



Küçük boyutlu örnek problemden elde edilen sonuçlar oluşturulan matematiksel modelin problemin özelliklerine uygun olduğunu ve doğru çalıştığını göstermiştir. Ancak bu çalışmada ele alınan problem ile ilgili oluşturulan farklı boyutlardaki örnekler Lingo optimizasyon programı ile çalıştırıldığında örnek boyutu arttığında optimum çözüm ve optimum çözüme ulaşmanın zorlaştığı gözlemlenmiştir. Bu nedenle, problemin çözümü için bir ÇAGA tasarlanmıştır.

2.4. GENETİK ALGORİTMA

Gerçek hayat optimizasyon problemlerinin çoğu matematiksel model çözüm yöntemleri ile çözülemeyecek kadar zordur. Klasik yöntemlerle sonuca ulaşmak uzun sürebilir veya uzun sürmesine rağmen beklenen sonucu vermeyebilir. Pek çok araştırmacı ve uygulayıcı, bu gibi durumlarda, makul bir sürede makul kalitede bir problem örneğini çözmek için meta/sezgisel optimizasyon yöntemlerine başvurur. Klasik optimizasyon yöntemleri matematiksel modelin amaç fonksiyonu ve kısıtların türüne (doğrusal, doğrusal olmayan vb.) ve karar değişkenlerinin türüne (0-1 tamsayı, gerçek sayı vb.) odaklanır. Sezgisel yöntemler karmaşık gerçek hayat problemlerine uyarlanabilir ve klasik optimizasyon yöntemlerinin uzun sürelerde ulaştığı çözümlere daha kısa sürelerde ulaşabilir. MATÇ problemi ile ilgili literatürde çok sayıda farklı sezgisel algoritma geliştirilmiştir. GA literatürde birçok farklı çalışma alanında uygulamaları mevcuttur bunun yanısıra çizelgeleme uygulamalarında da sık tercih edilen bir sezgisel algoritmadır.

GA doğadaki evrimsel mekanizmaları taklit ederek geniş alanlar üzerinde çalışan olasılıksal arama prosedürleridir. GA'nın ilkeleri ilk olarak John Holland (1975) tarafından ortaya atılmıştır. Literatürde birçok alanda başarılı uygulamaları bulunmaktadır. GA yalnızca amaç fonksiyonuna ihtiyaç duyar, çözümlere ulaşmak için parametre kümesi yerine probleme göre kodlanmış yapılar kullanır, çözüm uzayının tamamı yerine belirli bölümlerini tarar bu sayede geleneksel optimizasyon yöntemlerine göre daha etkin arama yaparak kısa sürede bir çözüme ulaşabilirler.

GA'nın çalışma adımları aşağıdaki gibidir;

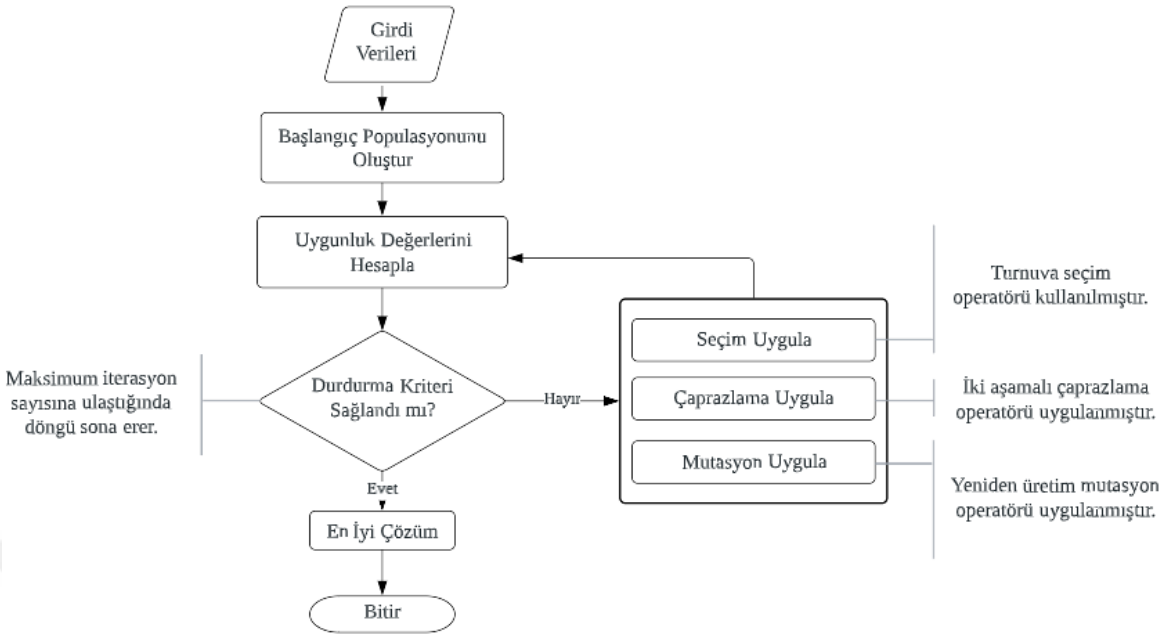
- Arama uzayındaki bütün olası çözümler kromozom olarak kodlanır,
- Genellikle rastgele oluşturulan bir çözüm seti (popülasyon) ile başlar, kromozomların her biri popülasyondaki bir bireyi oluşturur,
- Her bir bireyin uygunluk değeri hesaplanır,
- Popülasyon içerisinden belirli sayıda birey rassal olarak seçilir,
- Seçilen bireylere çaprazlama ve mutasyon operatörleri uygulanır ve yeni bir popülasyon elde edilir,
- Elde edilen yeni popülasyon tekrar döngüye girer,
- Önceden belirlenen iterasyon sayısı kadar döngü devam eder,
- İterasyon sayısına ulaşıldığında elde edilen en yüksek uygunluk değerine sahip birey en iyi çözümü verir.

GA'nın bilinmesi gereken en genel terimleri ve açıklamaları:

- **Kromozom**: Problemi oluşturan çözümlerin kodlandığı birim.
- **Gen:** Kromozomu oluşturan en küçük birim.
- **Alel**: Bir genin aldığı değer.
- **Birey**: Her bir kromozom bir bireyi temsil eder.
- **Popülasyon**: Birden fazla kromozomun oluşturduğu küme.
- **İterasyon:** GA'nın tek bir akışına verilen isim.
- **Nesil**: Her bir iterasyonun sonunda oluşan yeni popülasyona verilen isim.

Bu çalışmada ele alınan problemin çözümü için geliştirilen ÇAGA C# programlama dili kullanılarak, 64 bit işletim sistemi, Intel i5 işlemci, 8 GB Ram'e sahip bilgisayarda yazılmıştır. GA akış şeması Şekil (2.3)'te verilmiştir.

Şekil 2.3: Genetik Algoritma Akış Şeması



Oluşturulan ÇAGA'nın kapalı (Pseudo) kodu Tablo (2.6)'da belirtilmiştir:

Tablo 2.6: ÇAGA Sözde (Pseudo) Kodu

Algoritma Adımları:	**ÇAGA**
Girdi:	$n, t, m, m_t, P_{jtm}, S_{ijtm}, RT_{jt}, d_j, Turnuva_{büyüklük}, Ols_{çaprazlama}, Ols_{mutasyon}$
1:	Başlangıç popülasyonu (***Pop***$_{başlangıç}$) oluştur ()
2:	Uygunluk Değerlerini Hesapla() (Bkz., Eşitlik (9)-(11))
3:	***Pop***$_{mevcut}$ ← ***Pop***$_{başlangıç}$
4:	**while** durdurma kriteri sağlanana kadar **do**
5:	Seçim()
6:	Çaprazlama()
7:	Mutasyon()
8:	Uygunluk Değerlerini Hesapla()
9:	Elitizm()
10:	**end while**
11:	return ***Pop***$_{mevcut}$'daki en iyi birey (***Best***$_{kromozom}$)

2.4.1. Başlangıç Popülasyonu

Popülasyon kromozomlardan oluşan ve her bir kromozomun uygunluk değerine sahip olduğu, yani bir problemde olası birçok çözümü içerisinde barındıran çözümler kümesidir. GA iterasyon adı verilen belirli bir durdurma kriterine sahip döngülerden oluşmaktadır. Bu iterasyonlar probleme uygun en iyi çözümü bulmak için popülasyon üzerinde yapılan işlemlere dayanmaktadır. Bir iterasyon sonucunda oluşan yeni

popülasyonun bir öncekinden daha iyi olması beklenir, bu sayede her iterasyonda en iyi çözüme biraz daha yaklaşılmış olur. GA iterasyonları başlatmak için bir başlangıç popülasyonuna ihtiyaç duyar. Başlangıç popülasyonu oluşturmak için rastgele yöntemler veya sezgisel yöntemler tercih edilmektedir. Bununla beraber popülasyon büyüklüğü GA'nın performansını etkileyen önemli parametrelerden biridir. Popülasyon büyüklüğünün çok az olması olası çözüm alanını daraltır daha iyi sonuçlar elde etmenin önüne geçer, aynı şekilde çok fazla olması çözüm alanının çok genişlemesine ve en iyi çözümü bulma süresinin uzamasına neden olacaktır. Bu nedenle popülasyon büyüklüğünün belirlenmesi araştırmacılar için önemli bir konu olmuştur. Bu çalışmada GA'nın parametre ayarları Lopez-Ibanez vd. (2016) tarafından önerilen otomatik parametre yapılandırma aracı (Iterated F-Race algorithm (**irace**)) kullanılarak, popülasyon büyüklüğü U~[20,100] aralığından her bir örnek grubu için ayrı ayrı çalıştırılarak belirlendi.

2.4.2. Uygunluk Değerinin Hesaplanması

Uygunluk (fitness) değeri popülasyonu oluşturan her bir bireyin çözüm kalitesini belirler. Enküçükleme problemlerinde uygunluk değeri düşük olan bireylere, enbüyükleme problemlerinde de uygunluk değeri yüksek olan bireylere daha fazla hayatta kalma şansı verilir. Bu sayede her iterasyonda uygunluk değerleri iyi olan bireylerin genetik çeşitliliği artırılarak oluşturulan yeni popülasyonun bir öncekinden daha iyi olduğundan emin olunur. GA'da uygunluk değerinin hesaplanması kromozom yapısı ile ilişkilidir. Problem türüne uygun olarak oluşturulan bir kromozom uygunluk değerinin hesaplanmasında büyük kolaylık sağlamaktadır.

Bu çalışmada ele alınan problemde C_{enb} ve $\sum T$ amaçlarının enküçüklenmesi belirlenmiştir. Bu amaçlar işlerin sıralanmasına bağlıdır. Kromozom gösteriminden gelen iş sıralamaları kullanılarak amaçlar doğrultusunda uygunluk değeri hesaplanır. ÇAGA'da uygunluk değeri hesaplama fonksiyonunun sözde kodu Tablo (2.7)'de belirtilmiştir:

Tablo 2.7: ÇAGA Uygunluk Değeri Hesaplama Fonksiyonu Sözde (Pseudo) Kodu

Algoritma Adımları:	**Uygunluk Değeri Hesaplama**
Girdi:	$n, t, m, m_t, Pop_{mevcut}$
1:	**function** Uygunluk Değerlerini Hesapla ()
2:	**for** $i \leftarrow 0$ **to** $Pop_{büyüklük}$ **do**
3:	*İşleri_Sırala(* $\mathbf{Pop}_i$*)*
4:	*İşlem_Süresini_Hesapla (* $\mathbf{Pop}_i$*)*
5:	*Tamamlanma_Süresini_Hesapla (* $\mathbf{Pop}_i$*)*
6:	*Gecikme_Süresini_Hesapla (* $\mathbf{Pop}_i$*)*
7:	*Uygunluk_Değerini_Hesapla (* $\mathbf{Pop}_i$*)*
8:	**end for**
9:	**end function**

Önerilen ÇAGA'da uygunluk değerleri, ATY, KSY ve TSY olmak üzere 3 farklı skalerleştirme yöntemi kullanılarak hesaplanmıştır. Her bir yöntemden elde edilen sonuçlar kıyaslanmıştır.

2.4.3. Kromozom Yapısı

Kromozom yapısının belirlenmesi problemin performansını etkileyeceğinden iyi bir kromozom yapısı oluşturmak için problemi çok iyi analiz etmek önemlidir. İyi tasarlanmış bir kromozom problemin amaç fonksiyonları ile problemin spesifik özelliklerini de karşılamalıdır. Çözülmek istenen bir problemin spesifik özelliklerinin ve amaç fonksiyonlarının algoritmaya tanımlanması kromozom yapısı ile sağlanır. Kromozomu oluşturan en küçük birime gen denir. Gen yapısı problemden probleme farklılık gösterebilir. Literatürde farklı problem tipleri için geliştirilen birçok kromozom türü mevcuttur, bunlardan bazıları ikili kodlama (binary encoding), permütasyon kodlama (permutation encoding), değer kodlama (value encoding) ve ağaç kodlamadır (tree encoding). GA'da her bir kromozomun uygunluk değeri vardır. Yeni bireyler üretilirken uygunluk değeri iyi olan bireylerin seçim şansı artırılarak bu bireylerin çoğalmasına izin verilir, böylece iyi özellikler yeni nesle aktarılır ve her iterasyonda iyi özelliklerin yayılması sağlanmış olur.

Bu çalışmada tercih edilen genetik gösterim (kromozom yapısı), Pargar vd. (2018) çalışması baz alınarak oluşturulmuştur. MATÇ probleminin aşama sayısını, her bir aşamadaki makinelerini kapsayacak, hangi aşamada paralel makine olduğunu belirtecek ve aynı zamanda işlerin makinelere hangi sırada atandığını gösterecek bir kromozom yapısı Şekil (2.4)'te sunulan matris yapısıyla tanımlanmıştır. Bu yapıya göre

Pargar vd. (2018), çalışmalarında sunulan ondalıklı sayılardan oluşan gen gösterimi ile ondalıklı kısmın virgülden önceki rakamı makine numarasını, virgülden sonraki kısmı da küçükten büyüğe sıralandığında oluşan sıraya göre işlerin belirtilen makinelerde hangi sırada işlendiğini göstermektedir. Kromozomun matris yapısı satırlarda aşama sayısını, sütunlarda ise iş sayısını belirtmektedir. Her bir aşamada hangi makine(ler) var ise kromozomun gen değerleri bu makine numaraları dikkate alınarak oluşturulmaktadır. Örneğin iki adet paralel makinası bulunan ilk aşama için ($t_1 \rightarrow \{m_1\text{-}m_2\}$), 5 adet iş (i1-i2-i3-i4-i5) sayısı kadar rastgele 0-1 aralığında ondalıklı sayılar üretildikten sonra, makine numarası (1 veya 2) arasında da ayrıca rastgele bir seçim yapılıp bunların toplamı alınarak Şekil (2.4)'teki kromozom matrisinde ilk satırda belirtilen 1,12 - 2,31 - 2,9 - 1,78 - 1,56 değerleri elde edilmiştir. Bu satırın çözümlemesi; i1, i4 ve i5 virgülden önce 1 rakamı bulundurduğu için m_1'e atanmıştır. Ondalıklı kısım sıralandığında (i1=1,12 – i5=1,56 – i4=1,78) i1, birinci sırada; i5, ikinci sırada; i4 ise üçüncü sırada makine 1'de işlenecektir şeklinde yorumlanmaktadır. Aynı şekilde, m_2'ye atanan i2 ve i3 işleri de i2, birinci sırada ve i3, ikinci sırada işlenecektir.

Şekil 2.4: Geliştirilen ÇAGA'nın Kromozom Yapısı

Kromozom		İşler				
Aşamalar	**Makineler**	**i1**	**i2**	**i3**	**i4**	**i5**
T1	M1-M2	1,12	2,31	2,9	1,78	1,56
T2	M3	3,06	3,67	3,24	3,42	3,3
T3	M4	4,48	4,91	4,29	4,38	4,64
T4	M5-M6	5,69	6,14	5,68	5,59	5,02

Kromozom Çözümlemesi		i1	i2	i3	i4	i5
T1	M1	1	-	-	3	2
	M2	-	1	2	-	-
T2	M3	1	5	2	4	3
T3	M4	3	5	1	2	4
T4	M5	4	-	3	2	1
	M6	-	1	-	-	-

Yukarıda sunulan kromozom yapısı ile bu çalışmada işlenen problemin temel özellikleri büyük oranda algoritmaya aktarılmış olmaktadır. Bu problemin amaçlarını

oluşturan tamamlanma zamanı ve gecikme süresi, problemin girdilerini oluşturan iş, aşama, makine sayıları ile her bir işin teslim süresi parametreleri kullanılarak oluşturulan sıralamalar yardımı ile hesaplanacağından, bu kromozom yapısının problem için uygun olduğu görülmektedir.

2.4.4. Seçim

GA'nın nesilden nesile evrimi üç ana operatör tarafından gerçekleştirilir. Bunlar, seçim, çaprazlama ve mutasyon operatörleridir. Seçim operatörü, popülasyondan bireyleri seçmek ve onları bir çiftleşme havuzuna yerleştirmek için kullanılır. Çiftleşme havuzundaki bireyler, gelecek neslin temeli olan yeni yavruları oluşturmak için kullanılır. Bir sonraki neslin önceki nesilden daha iyi özelliklere sahip olması istenir, bu nedenle çiftleşme havuzuna alınacak bireylerin popülasyon içerisindeki iyi bireylerden seçilmesi gerekir. Seçim operatörü basit bir şekilde çiftleşme havuzu için popülasyondaki daha iyi bireylerin seçilmesini sağlayan bir süreçtir (Yang ve Soh, 1997).

GA'da popülasyon çeşitliliği, mevcut popülasyondaki iyi bireylerden gelen genlerin kullanılmasını sağlar. Seçim baskısı mevcut popülasyondaki daha iyi bireylerin tercih edilmesini sağlar. Popülasyon çeşitliliği ve seçim baskısı bir arada kullanıldığında GA'nın performansına iyileştirici etkisi olmaktadır. GA'nın çözüm uzayında arama yapabilme becerisi popülasyon çeşitliliği ile sınırlıdır, popülasyon ne kadar çeşitli olursa GA çözüm uzayında daha fazla alanda arama yapar. Buna karşın seçim operatörü bir önceki popülasyondan gelen iyi bireylerin yeni popülasyona artan sayıda katılmasını sağlar. Bu kısaca önceki popülasyonda bulunan bütün bireylerin yeni popülasyona katılmadığı anlamına gelir. Seçim operatörünün iyi olan bireylere tanıdığı öncelik ne kadar artarsa yeni popülasyonun çeşitliliği azalacaktır. Bu da GA'nın çözüm uzayındaki arama kapasitesinin sınırlanmasına sebep olacaktır. GA'nın performansının iyi olması için popülasyon çeşitliliği ve seçim baskısının dengeli olması gerekmektedir.

Goldberg ve Deb (1991), çalışmalarında 4 farklı seçim yönteminin, orantılı üreme, sıralama seçimi, turnuva seçimi, Genitor (veya "kararlı durum") seçimi, sabit ve özdeş amaç fonksiyonları ile farklı birey sınıflarının oranındaki değişimi tanımlayan deterministik sonlu fark denklemleri oluşturarak elde edilen sonuçların karşılaştırmasını yapmışlardır. Sıralama ve turnuva seçiminin normal koşullar altında yakınsama süresinde güçlü bir büyümeyi sürdürdüğü, ölçeklendirme olmaksızın orantılı seçimin yakınsamaya yönelik sabit bir baskıyı korumada daha az etkili olduğu gösterilmiştir. Genitor'un büyük

popülasyon büyüklükleri veya çoklu popülasyonlar gibi başka düzeltmeler gerektirmesinin nedenlerinden birinin, yüksek büyüme oranlarına ihtiyaç duymasının ve bu büyüme oranlarının neden olduğu erken yakınsama olduğunu öne sürmüşlerdir. Bu çalışmalarının esas amacı GA'da hangi seçim yönteminin kullanılacağına karar vermenin zor olmasıdır (A.g.e). Ancak herhangi bir yöntemi kullanmadaki sınırlı ampirik başarıya rağmen, hala kesin bir cevabı bulunmamaktadır.

Turnuva seçimi, uygunluk değerleri iyi olan kromozomlara daha yüksek seçilme şansı verirken aynı zamanda popülasyondaki tüm bireylere de seçilme şansı tanır, bu nedenle hızlı sonuçlar üretir ve uygulaması kolaydır. Bu nedenle bu çalışmada turnuva seçim yöntemi tercih edilmiştir.

Turnuva seçiminin adımları;

i. Popülasyon içerisinden rassal olarak turnuva büyüklüğü kadar kromozom seçilir.
ii. Seçilen kromozomlar uygunluk değerine göre yarışır, uygunluk değeri daha iyi olan birey yeni popülasyon listesine alınır.
iii. Bu işlem popülasyon büyüklüğü tamamlanana kadar tekrarlanır.
iv. Elde edilen yeni liste çaprazlama operatörüne gönderilir.

Turnuva seçimi popülasyondaki tüm bireylere seçilme şansı tanır. Kötü uygunluk değerine sahip bir birey turnuva seçiminden geçip çaprazlamaya girdiğinde her zaman kötü sonuçlar elde edilmeyebilir. Ayrıca bu sayede GA'nın yerel optimum noktalara takılmasının önüne geçer.

Turnuva seçim yönteminin sözde kodu Tablo (2.8)'de belirtilmiştir:

Tablo 2.8: ÇAGA Seçim Fonksiyonu Sözde (Pseudo) Kodu

Algoritma Adımları:	**Seçim**
Girdi:	$n, t, m, m_t, Pop_{mevcut}, Fitness, Turnuva_{büyüklük}$
1:	**function** Seçim()
2:	**for** $i \leftarrow 0$ **to** $Pop_{büyüklük}$ **do**
3:	**for** $j \leftarrow 0$ **to** $Turnuva_{büyüklük}$ **do**
4:	$(\mathbf{Rss_{kromozom}}) \leftarrow \boldsymbol{Pop}$
5:	if $Fitness_{onward} > Fitness_{best}$ **then**
6:	$\boldsymbol{Pop_{yeni}} \leftarrow \boldsymbol{Best}$
7:	**end for**
8:	$\boldsymbol{Pop_{mevcut}} \leftarrow \boldsymbol{Pop_{yeni}}$
9:	**end for**
10:	**end function**

2.4.5. Çaprazlama

Çaprazlama operatörü GA'nın bir diğer önemli operatörüdür. Seçim operatöründen gelen popülasyondaki bireylerden daha iyi olması beklenen yeni bireylerin oluşturulmasını amaçlar. Çaprazlamada genellikle iki ebeveyn kullanılır.

Literatürde çalışılan bazı çaprazlama operatörleri; Pozisyon Tabanlı Çaprazlama (Position Based Crossover-PBX), Sıra Tabanlı Çaprazlama (Order Based Crossover-OBX), Kısmi Planlı Çaprazlama (Partially Mapped Crossover -PMX), Dairesel Çaprazlama (Cycle Crossover-CX), Sıralı Çaprazlama (Order Crossover-OX), Modife Edilmiş Döngüsel Çaprazlama(Modified-Cycle Crossover-CX2). Çaprazlama operatörü problem türüne göre belirlenen kromozom yapısına bağlı olarak değişmektedir.

Bu çalışmada, Alizadeh vd. (2020) geliştirdikleri matris çaprazlama operatörü tercih edilmiştir. Nüfusu çeşitlendirmek ve yerel optimada sıkışmaktan kaçınmak için iki aşamalı çaprazlama operatörü kullanmışlardır. İlk çaprazlama işleminde, kromozomdaki her gene karşılık 0 veya 1 aralığında seçilen rassal sayılarla bir maske oluşturulur ($Maske_1$). Sayı 1'e eşitse, karşılık gelen genler yavru kromozomlarında değiştirilir. Aksi takdirde değişmeden kalırlar.

İkinci çaprazlama işleminde, kromozomdaki her gene karşılık 0 ve 1 arasında seçilen rassal sayılardan ikinci bir maske oluşturulur ($Maske_2$). İlk çaprazlama işleminden gelen yavru bireylerden yeni yavru bireylerin oluşturulması için aşağıdaki formülasyonlar kullanılır;

$$Y_1^{'} = Y_1 + r * (Y_2 - Y_1) \tag{2.21}$$

$$Y_2^{'} = Y_2 + r * (Y_1 - Y_2) \tag{2.22}$$

Burada Y_1 ve Y_2 ilk çaprazlama işleminden oluşturulan yavru bireyleri, r ifadesi $Maske_2$'deki ilgili gene karşılık gelen değeri göstermektedir. $Y_1^{'}$ ve $Y_2^{'}$ ifadeleri son oluşturulan yavru bireyleri temsil etmektedir. İki aşamalı çaprazlama operatörü Şekil (2.5)'te sunulmuştur.

Şekil 2.5: Çaprazlama Operatörü

	i1	i2	i3	i4	i5
T1	1,63	2,11	1,40	1,29	1,82
T2	3,04	3,94	3,02	3,20	3,46
T3	4,70	4,12	4,88	4,35	4,26
T4	5,98	6,21	5,35	5,65	6,00

Ebeveyn 1

	i1	i2	i3	i4	i5
T1	2,50	2,23	1,24	1,31	1,00
T2	3,79	3,46	3,70	3,09	3,49
T3	4,53	4,89	4,31	4,52	4,11
T4	5,54	5,81	5,53	6,84	5,99

Ebeveyn 2

Maske$_1$

1	1	1	1	0
0	0	1	0	1
0	1	1	1	1
0	0	0	1	0

	i1	i2	i3	i4	i5
T1	2,50	2,23	1,24	1,31	1,82
T2	3,04	3,94	3,70	3,20	3,49
T3	4,70	4,89	4,31	4,52	4,11
T4	5,98	6,21	5,35	6,84	6,00

Yavru 1

	i1	i2	i3	i4	i5
T1	1,63	2,11	1,40	1,29	1,00
T2	3,79	3,46	3,02	3,09	3,46
T3	4,53	4,12	4,88	4,35	4,26
T4	5,54	5,81	5,53	5,65	5,99

Yavru 2

Maske$_2$

0,58	0,29	0,13	0,95	0,35
0,81	0,67	0,98	0,88	0,80
0,23	0,85	0,84	0,74	0,61
0,52	0,23	0,80	0,13	0,10

	i1	i2	i3	i4	i5
T1	2,00	2,19	1,26	1,29	1,54
T2	3,64	3,62	3,03	3,10	3,47
T3	4,66	4,23	4,79	4,40	4,20
T4	5,75	6,12	5,50	6,68	6,00

Yavru 1'

	i1	i2	i3	i4	i5
T1	2,14	2,14	1,38	1,31	1,29
T2	3,18	3,79	3,69	3,19	3,49
T3	4,57	4,77	4,40	4,48	4,17
T4	5,77	5,91	5,39	5,81	5,99

Yavru 2'

İki aşamalı çaprazlama operatörünün avantajlarından biri MATÇ probleminin paralel makine yapısını bozmadan yavru bireyler oluşturmasıdır. Bir diğer avantajı da ilk aşama için rassal olarak oluşturulan maske$_1$'in her bir gen değeri 0 seçilirse çaprazlamaya giren bireyler aynı şekilde kalacaktır, tam tersinin olduğu durumda da (her gen değerinin 1 olması durumu) bireyler yer değiştirecektir. Bu da çaprazlamanın olmaması, ebeveyn bireylerin çaprazlamadan aynı şekilde çıkması anlamına gelir. Bu durum, çaprazlama operatörünün dolayısıyla GA'nın performansını olumsuz yönde etkileyecektir. Çaprazlama operatörünün ikinci aşaması çaprazlama işlevini garantilediği için algoritmanın performansı üzerinde olumlu bir etkisi olacaktır. Oluşturulan ÇAGA'da seçim işlemi ile elde edilen yeni popülasyon uygunluk değerlerine göre tekrar artan sırada sıralanır ve ardışık iki birey çaprazlamaya gönderilir. Çaprazlama olasılığı sağlanırsa çaprazlama işlemi uygulanır. Eğer çaprazlama olasılığı sağlanmazsa bireyler çaprazlama yapılmadan popülasyondaki yerlerini alırlar. Çaprazlama işleminin başında uygulanan sıralama işlemi birbirine yakın özellikler barındıran bireylerin çaprazlamaya girmesini sağlar. İyi bireylerden daha iyi özelliklerde yavru bireyler elde edilmesi beklendiğinden

bu sıralama işlemi de ÇAGA'nın performansına ek bir fayda sağlayacaktır. ÇAGA ile oluşturulan çaprazlama işleminin sözde kodu Tablo (2.9)'da verilmiştir:

Tablo 2.9: ÇAGA Çaprazlama Fonksiyonu Sözde (Pseudo) Kodu

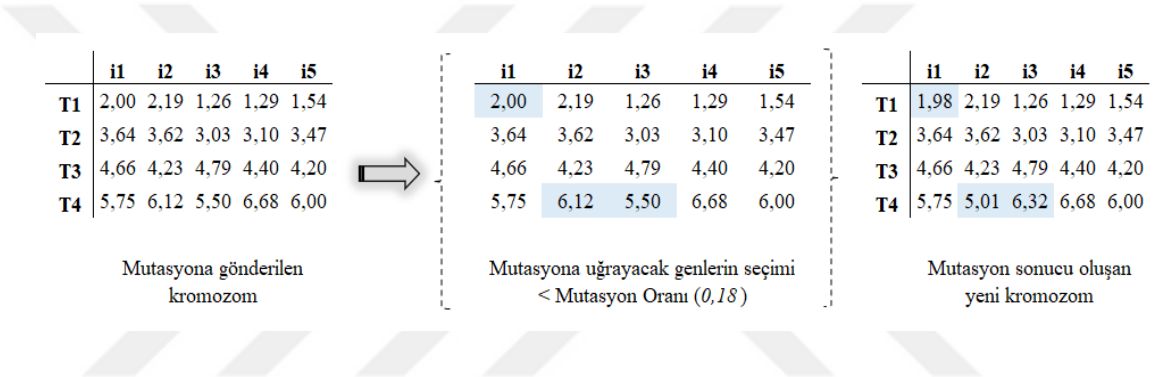
Algoritma Adımları:	**Çaprazlama**
Girdi:	$n, t, m, m_t, Pop_{mevcut}, Fitness, Ols_{çaprazlama}$
1:	**function** Çaprazlama()
2:	**for** $i \leftarrow 0$ **to** $Pop_{büyüklük}$ **do**
3:	ArtanSıralama($\boldsymbol{Pop_{mevcut}}$)
4:	**end for**
5:	**for** $i \leftarrow 0$ **to** $Pop_{büyüklük}/2$ **do**
6:	($\boldsymbol{Ebeveyn_1, Ebeveyn_2}$) ← Çaprazlanacak bireyleri seç
7:	($\mathbf{Rss_{sayı}}$) ← RassalSayıSeç[0,1]
8:	$\boldsymbol{maske_1[\ ]}$ ← RassalsayıSeç {0,1}
9:	$\boldsymbol{maske_2[\ ]}$ ← RassalsayıSeç [0,1]
10:	**if** ($\mathbf{Rss_{sayı}} < \boldsymbol{Ols_{çaprazlama}}$)
11:	**if** ($\boldsymbol{maske_1[\ ] = 1}$)
12:	$\boldsymbol{Yavru_1, Yavru_2}$ ← Çaprazlama uygula ($\boldsymbol{Ebeveyn_1, Ebeveyn_2}$)
13:	**end if**
14:	$\boldsymbol{Yavru_1'} = Yavru_1 + Maske_2[\ ] * (Yavru_2 - Yavru_1)$
15:	$\boldsymbol{Yavru_2'} = Yavru_2 + Maske_2[\ ] * (Yavru_1 - Yavru_2)$
16:	**end if**
17:	$\boldsymbol{Pop_{mevcut}} \leftarrow (\boldsymbol{Yavru_1', Yavru_2'})$
18:	**end for**
19:	**end function**

2.4.6. Mutasyon

Mutasyon işlemi GA'nın yerel optimuma erken yakınsamasını önleyen ve çözüm uzayındaki arama kabiliyetini artıran bir genetik operatördür. GA'da seçim ve çaprazlama daha iyi çözümler getirirken, mutasyon iyi çözümleri en iyi çözüme yaklaştırmak umuduyla kullanılır. Çaprazlama operatörü, iki kromozomun çiftleşme sürecini denetler. Her ebeveynden hangi genetik materyalin çocuk kromozomuna aktarıldığına karar verir. Mutasyon operatörü yavru havuzundaki her kromozomu alır ve içeriğinin bir kısmını rastgele değiştirir. Literatürde çalışılan bazı mutasyon operatörlerine; Ayna Mutasyonu ve İkili Bit Çevirme Mutasyonu (Mirror Mutation and Binary Bit-Flipping Mutation), Yönlendirilmiş Varyasyon Tekniklerine Dayalı Mutasyon (Mutation Based on Directed Variation Techniques), Yönlendirilmiş Mutasyon (Directed Mutation), Dalgacık Mutasyonu (Wavelet Mutation), Muhlenbein's Mutasyon (Muhlenbein's Mutation) örnek verilebilir (Lim vd., 2017).

Bu çalışmada uygulanan mutasyon operatörü, kromozomda bulunan her bir gene 0 ve 1 arasında rassal bir değer atayarak başlatılır. Eğer seçilen rassal değer mutasyon olasılığından küçük ise ilgili gen mutasyona uğrar. Mutasyona giren gen, başlangıç popülasyonu oluştururken tasarlanan kromozom üretme yöntemine göre yeniden üretilir. Çünkü başlangıç popülasyonunu oluşturmak için gerekli olan problemin makine özelliklerini kapsayan kromozom yapısının korunması gereklidir. Kromozomdaki genlerin ne kadarının mutasyona uğrayacağının belirlenmesi önemli bir konudur. Eğer mutasyon uygulanan gen sayısı fazla olursa kromozomun sahip olduğu iyi özellikler kaybedilebilir. Mutasyona uğrayan gen sayısı az olursa da GA'nın arama kabiliyeti düşer. Uygulanan mutasyon operatörünün yapısı Şekil (2.6)'da sunulmuştur.

Şekil 2.6: Mutasyon Operatörü



GA ile oluşturulan mutasyon işleminin sözde kodu Tablo (2.10)'da verilmiştir:

Tablo 2.10: ÇAGA Mutasyon Fonksiyonu Sözde (Pseudo) Kodu

Algoritma Adımları:	**Mutasyon**
Girdi:	$n, t, m, m_t, Pop_{mevcut}, Ols_{mutasyon}$
1:	**function** Mutasyon()
2:	**for** $i \leftarrow 0$ **to** $Pop_{büyüklük}$ **do**
3:	$(\mathbf{Rss_{sayı}}) \leftarrow$ RassalSayıSeç[0,1]
4:	**if** $(\mathbf{Rss_{sayı}} < \boldsymbol{Ols_{mutasyon}})$
5:	Üret($\boldsymbol{Yavru_{gen}'}$)
6:	$\boldsymbol{Ebeveyn_i} \leftarrow \boldsymbol{Yavru_{gen}'}$
7:	**end if**
8:	**end for**
9:	**end function**

2.4.7. Elitizm

GA'da her bir iterasyonun sonunda yeni bir popülasyon elde edilir. Oluşan her yeni popülasyon, içerisinde amaç fonksiyonlarını sağlayan en iyi değere sahip bir veya

daha fazla kromozom bulundurur. İterasyonlar devam ettikçe bulunan en iyi değerlerin kaybolmaması istenir. Çünkü GA'nın rassal yapısı nedeniyle her yeni popülasyonun bir öncekinden daha iyi olmama ihtimali vardır. Elitizm operatörü, mutasyon operatörü ile elde edilen yeni popülasyon içerisindeki en iyi uygunluk değerine sahip birey(lerin) bir sonraki nesle aktarılmasını sağlar. Bu sayede iyi özelliklere sahip birey(lerin) her nesil içerisinde bulunması sağlanmış olur. Elitizm sadece en iyi çözümün genetik operasyonlardan zarar görmesini önlemek için değil, aynı zamanda GA'nın küresel yakınsamasını sağlamak için de uygulanır (Sun ve Yu, 2015).

Bu çalışmada, hızlı bir yakınsamadan kaçınmak ve nesiller boyunca genetik çeşitliliği artırmak amacıyla sadece bir elit birey yeni nesle aktarılacaktır. Mutasyon operatöründen gelen yeni popülasyon içerisinde en iyi uygunluk değerine sahip birey seçilir ve bu popülasyonun ilk sırasına yerleştirilir. ÇAGA'da oluşturulan elitizm operatörünün sözde kodu Tablo (2.11)'de sunulmuştur:

Tablo 2.11: ÇAGA Elitizm Fonksiyonu Sözde (Pseudo) Kodu

Algoritma Adımları:	**Elitizm**
Girdi:	$n, t, m, Pop_{mevcut}, Fitness$
1:	**function** Elitizm()
2:	**for** $i \leftarrow 0$ **to** $Pop_{büyüklük}$ **do**
3:	EniyiyiBul $(\boldsymbol{Best_{kromozom}})$
4:	Güncelle $\boldsymbol{Pop_{mevcut}}$ $(\boldsymbol{Pop_0 \leftarrow Best_{kromozom}})$
5:	**end for**
6:	**end function**

2.4.8. Durdurma Koşulu

Durdurma koşulu bir GA'nın ne zaman sonlandırılacağını belirler. GA'da genellikle kullanılan durdurma kriterleri aşağıdaki gibidir;

- Belirli bir iterasyondan sonra popülasyonda herhangi bir gelişme olmadığında,
- Önceden belirlenen nesil sayısına ulaşıldığında,
- Amaç fonksiyonu değeri önceden tanımlanmış bir değere ulaştığında.

GA'da en sık kullanılan durdurma koşulu önceden belirlenen nesil sayısına ulaşılmasıdır. Uygunluk fonksiyonu değerlendirmelerinin süresi ve maliyeti, optimizasyon sürecinin uzunluğunu kısıtlayabilir. Bir başka faydalı sonlandırma koşulu, optimizasyon sürecinin yakınsamasıdır. Optimuma yaklaşırken, uygunluk değeri

iyileştirmelerinin ilerlemesi önemli ölçüde azalabilir. Önemli bir süreç gözlenmezse, evrimsel süreç durur. Örneğin, sürekli optimizasyon problemlerinin optimumuna yaklaşırken, ardışık birçok nesilde uygunluk farkının 10^{-8}'den düşük tekrarlanması durgunluğun tanımı için mantıklıdır. Tabii ki, durgunluk yalnızca aramanın yerel optimumda takılıp kaldığı ve dolayısıyla küresel olanı kaçırdığı anlamına gelebilir. Yeniden başlatma stratejileri, aynı yerel optimuma takılmaktan kaçınan yaklaşımlardır. GA, farklı alanlardan başlayarak çözüm uzayında tekrar tekrar aynı alana yaklaşırsa, yerel optimumun büyük bir çekici olma şansı yüksektir ve daha iyi bir yerel optimum bulma olasılığı düşüktür. Ayrıca bu yerel optimum global olabilir (Kramer, 2017).

Bu çalışmada durdurma koşulu, önceden belirlenen maksimum uygunluk değerlendirme sayısı (MaxFES) ile problemin parametrelerinden biri olan iş sayısının çarpımı şeklinde belirlenmiştir. Burada MaxFES, 10.000 olarak tercih edilmiş ve algoritmanın çözüm uzayında daha fazla arama yapması için belirlenen iterasyon sayısının iş sayısı parametresi ile çarpımı uygun görülmüştür. Çünkü iş sayısının artması problemin büyüdüğü anlamına gelmektedir, bu da algoritmanın daha kaliteli çözümler üretebilmesi için daha fazla iterasyona ihtiyaç duyabileceği şeklinde yorumlanmıştır.

ÜÇÜNCÜ BÖLÜM

MATÇ PROBLEMİNİN ÇÖZÜMÜ İÇİN GELİŞTİRİLEN ÖĞRENME ETKİSİ TABANLI ÇAGA

3.1. PARAMETRE AYARLAMA

GA popülasyona çeşitlilik getiren üreme mekanizmalarını kullanarak yavru üreten bireylerin popülasyonlarından oluştuğu için algoritmanın elde edeceği çözümlerin kalitesi ve çözüm hızı algoritmaya tanımlanan popülasyon büyüklüğü, çaprazlama ve mutasyon olasılığı gibi parametrelere bağlıdır. Parametrelerde yapılan değişiklikler algoritmanın davranışını büyük ölçüde değiştirmektedir. Algoritma tasarlanırken seçilen operatörlerin yapısı parametre sayısını etkiler. Örneğin, turnuva seçimi yarışa girecek olan bireylerin sayısını belirleyen turnuva büyüklüğü parametresini gerektirirken, rulet tekeri yöntemi herhangi bir parametre gerektirmez. Ayrıca elitizm operatörü, yapısına göre bir sonraki nesle aktarılacak iyi bireylerin sayısının belirtilmesini gerektiren elitizm oranı parametresini gerektirir. Parametre sayısının artması da algoritmanın performansını etkileyen bir diğer unsurdur. Diğer yandan aynı GA farklı şekillerde kodlanabilir ve bunlar performansı önemli ölçüde etkileyebilir. Örneğin, bazı GA'lar paralel programlama için kodlanabilirken diğerlerinin paralelleştirilmesi zordur. Önemli bir fark yaratabilecek bir başka sabit parametre de genom kodlamasıdır ve bu da bir tasarım parametresidir. Optimum çözümlere ulaşmak için, algoritma, algoritmanın farklı aşamalarında farklı parametre değerleri kullanabilir. Bu değişikliğin kontrol edildiği strateji, parametre kontrol problemi olarak bilinir. Kontrol problemi, parametre ayarlarını dinamik olarak oluşturmak için bir strateji arar. Kontrol parametresi prosedürleri, devam eden çalışma için parametre ayarlarını değiştirmek üzere gerçek zamanlı verileri kullanarak GA performansını iyileştirir. Özetle, parametre ayarı, parametre kontrolü ve parametre tasarımı GA'nın performansını etkileyen temel faktörlerdir.

Bu çalışmada tasarlanan ÇAGA'nın performans ölçümü için aşağıdaki parametreler gerekmektedir;

- Popülasyon büyüklüğü,
- Turnuva büyüklüğü,
- Çaprazlama olasılığı,
- Mutasyon olasılığı,
- Amaç ağırlıkları,
- Referans değerleri.

Bu çalışmada, turnuva büyüklüğü sabit 2 değeri olarak belirlenmiştir.

Literatürde her bir parametre için problem tiplerine göre hesaplanmış değer aralıkları bulunmaktadır. Ancak her bir parametrenin bir arada kullanıldığında (parametre kümesi) problemin en uygun çözümünü verecek değerlerinin belirlenmesi gerekmektedir. Bu işlem literatürde parametre ayarı olarak tanımlanmıştır.

3.1.1. Literatürde Kullanılan Parametre Ayarlama Yöntemleri

En uygun parametre kümesini bulmak oldukça karmaşık bir konudur. Tipik olarak, algoritma geliştiricisi önce birkaç parametre konfigürasyonu seçer, yani parametrelere tam değer atamaları yapar ve bunları test etmek için deneyler yürütür. Daha sonra sonuçları inceler ve farklı konfigürasyonları test etmeye, algoritmayı değiştirmeye veya süreci durdurmaya karar verir. Bu manuel ayarlama yaklaşımı hiç ayarlama yapmamaktan daha iyi olmasına ve yüksek performanslı algoritmalara yol açmasına rağmen, bir takım dezavantajları da vardır: (*i*) insan çabası açısından yoğun zaman alır; (*ii*) genellikle kişisel deneyim ve sezgi tarafından yönlendirilir ve bu nedenle taraflıdır ve tekrarlanamaz; (*iii*) algoritmalar tipik olarak yalnızca oldukça sınırlı sayıda örnek üzerinde test edilir; (*iv*) az sayıda tasarım alternatifi ve parametre ayarı araştırılmıştır; ve (*v*) genellikle tasarım ve parametre ayarlama aşamasında kullanılan aynı örnekler, nihai algoritmayı değerlendirmek için de kullanılır ve bu da yanlı bir performans değerlendirmesine yol açar (Lopez-Ibanez vd., 2016). Bu dezavantajlar nedeniyle, algoritma tasarımı ve parametre belirleme sürecini otomatikleştiren farklı yöntemler geliştirilmektedir. Korkmaztan ve Bora (2017) çalışmalarında ilk parametre setini belirleyen, model tabanlı, gözlem tabanlı, arama tabanlı, etmen tabanlı ve metasezgisel olmak üzere 6 farklı yöntem olduğunu ifade etmişlerdir.

3.1.2. Tercih Edilen Parametre Ayarlama Yöntemi

Bu çalışmada geliştirilen ÇAGA'nın parametre ayarı için Lopez-Ibanez vd. (2016) tarafından önerilen otomatik parametre yapılandırma aracı (Iterated F-Race algorithm (**irace**)) tercih edilmiştir. **irace** paketi, bir dizi otomatik yapılandırma prosedürünü uygulayan bir yazılım paketidir. Özellikle, çeşitli son teknoloji algoritmaları otomatik olarak yapılandırmak için başarıyla kullanılan yinelenen yarış prosedürleri sunar. **irace**'de uygulanan yinelenen yarış prosedürleri, yinelenen F-race algoritmasını ve bunun üzerinde çeşitli uzantıları ve iyileştirmeleri içerir. Bu iyileştirmelerin arasında, erken yakınsamayı önlemek için bir yeniden başlatma mekanizması, doğru parametre

sınırlarını işlemek için kesik örnekleme dağılımlarının kullanımı ve döndürülen en iyi konfigürasyonların aynı zamanda en fazla sayıda eğitim örneğinde değerlendirilen konfigürasyonlar olmasını sağlamak için elitist bir yarış prosedürü bulunmaktadır.

irace algoritması, R programı aracılığıyla çalıştırılır. Algoritma iterasyon prensibine göre çalışır. İlk olarak programa tanımlanan parametre değer aralıkları uygulanacak iterasyon sayısına göre bölümlere ayrılır. Bu bölümler her iterasyonda çalıştırılacak olan parametre kombinasyonlarını belirler.

İterasyondaki tüm parametre kombinasyonları örnek setleri üzerinde çalıştırılır, en iyi kombinasyonu bulmak için bir yarışa girerler. Her bir iterasyon sonucunda bir veya daha fazla sayıda en iyi çözümü veren kombinasyon seti seçilir. Ardından, sonuçlar üzerinde istatistiksel bir test yapılır. Test sonuçlarına göre, en kötü konfigürasyonlar yarıştan çıkarılırken, hayatta kalan konfigürasyonlar bir sonraki örnekte çalıştırılır. Program maksimum çalışma sayısına ulaştığında sonlandırılır ve elde edilen çözüm kümesi kullanılan örnek seti için uygun olan en iyi parametreleri verir. Görsel (3.1)'de **irace**'nin bir çalışma sonunda elde ettiği en iyi parametre kombinasyonları gösterilmiştir.

Görsel 3.1: İrace'de Bir Örnek Seti İçin Elde Edilen En İyi Sonuçlar

```
# Best configurations (first number is the configuration ID; listed from best
ng to the sum of ranks):
   popsize weight1 cross_rate mutation_rate
79     206    0.00       0.49          0.88
5      253    0.02       0.29          0.72
33     231    0.02       0.39          0.74
```

3.2. PERFORMANS METRİKLERİ

Bu çalışma, skalerleştirme yaklaşımlarının varyantlarına dayanan GA'nın performanslarını karşılaştırmak için performans ölçümlerini gerektirir. Ele alınan algoritmaların performans değerlendirme metrikleri, ideal noktalara ağırlıklı uzaklık ve referans noktalara ağırlıklı uzaklıklarıdır. Çok amaçlı optimizasyon problemlerinde, ideal bir nokta, o ana kadar incelenen tüm çözümler arasında en küçük amaç değerken (enküçükleme problemleri için), amaç uzayındaki bir referans noktası, karar vericinin istediği bölgeyi belirlemesi için daha odaklı bir arama yapmasına izin veren değerdir. Ek olarak, nadir nokta, o ana kadar incelenen tüm çözümler arasında en büyük hedef değeridir (enküçükleme problemleri için) (Deliktaş ve Ustun, 2022). Bu çalışmada, C_{enb}'nin ideal noktası, $w_1 = 1$ $(w_2 = 0)$ amaç ağırlığı için algoritmanın 31 kez çalıştırılması sonucu elde edilen sonuçlar arasından en küçük değer olarak hesaplanmıştır.

Benzer şekilde, $\sum T_j$ 'nin ideal noktası ise, $w_1 = 0 \quad (w_2 = 1)$ amaç ağırlığı için algoritmanın 31 kez çalıştırılması sonucu elde edilen sonuçlar arasından en küçük amaç değeridir. Nadir noktalar, aynı amaç ağırlıkları için elde edilen sonuçlar arasından her bir amacın en büyük değeri olarak hesaplanır.

Skalerleştirme yöntemlerinde (KSY ve TSY için) kullanılan referans noktaları, her bir amacın ideal ve nadir noktaları kullanılarak hesaplanmıştır. İyi dağıtılmış referans noktalarının hesabında, konveks kombinasyon tekniği kullanılmıştır (J. Kim ve Kim S.K., 2006). $x^k = \arg\min\{ f_k(x) \mid x \in S \}, \ \forall k$, k. amaç fonksiyonunun optimum ya da optimuma yakın çözümü olsun. Burada, $f_1(x)$ ve $f_2(x)$ sırasıyla C_{enb} ve $\sum T_j$ temsil ederken, k, amaç fonksiyonu sayısını ve S, uygun çözüm kümesini ifade etmektedir. $V_k = \left[f_1(x^k), f_2(x^k) \right], \ \forall k$, x^k'ya karşılık gelen amaç fonksiyonu vektörüdür. $\lambda = \{ \lambda_1 + \lambda_2 = 1; \ \lambda_1, \lambda_2 \geq 0 \}$, konveks kombinasyon vektörünü (KKV) temsil etmektedir. Daha sonrasında, referans noktası vektörleri kümesi, $Rf^{\lambda} = \lambda_1 V_1 + \lambda_2 V_2$ formülasyonu ile elde edilmiştir. Örneğin, orta boyutlu bir kıyaslama örneği (No#10) için üç farklı KKV'den elde edilen referans noktası Tablo (3.1)'de sunulmuştur. Diğer örnekler için referans noktaları, benzer şekilde hesaplanmıştır.

Tablo 3.1: No#10 Örneği için Üç Farklı Referans Noktası Kombinasyonuna Göre Referans Noktaları

	λ_1	λ_2	Rf_1	Rf_2
KKV1	0,50	0,50	540,080	16,194
KKV2	0,25	0,75	569,213	8,097
KKV3	0,75	0,25	510,947	24,290

Farklı amaç fonksiyonu değerleri, farklı büyüklüklere sahip olduğu için amaç değerleri ilgili amacın nadir değerine bölünerek normalize edilmiştir (Marler ve Arora, 2005). Bu çalışmada, skalerizasyon yöntemleri ile önerilen ÇAGA'da uygunluk değeri olarak normalize edilmiş amaç fonksiyonu değerleri kullanılmıştır. Bu eşitliklerde N_i, i. amaç fonksiyonunun nadir noktasını temsil etmektedir. ATY, KSY ve TSY tarafından skalerize edilen uygunluk değerleri, Eşitlik (3.1)-(3.3)'de verildiği gibi hesaplanmıştır.

$$enk \ \text{ATY}^{Norm}(x) = \left[w_1 \frac{C_{enb}}{N_1} + w_2 \frac{\sum T_j}{N_2} \right] \tag{3.1}$$

$$enk\ \mathrm{KSY}^{Norm}(x) = \left[w_1\left(\frac{C_{enb} - Rf_1}{N_1}\right) + w_2\left(\frac{\sum T_j - Rf_2}{N_2}\right) + \alpha\left(\left|\frac{C_{enb} - Rf_1}{N_1}\right| + \left|\frac{\sum T_j - Rf_2}{N_2}\right|\right)\right] \quad (3.2)$$

$$enk\ \mathrm{TSY}^{Norm}(x) = enb\left[w_1\left(\frac{C_{enb} - Rf_1}{N_1}\right), w_2\left(\frac{\sum T_j - Rf_2}{N_2}\right)\right] + \left[\rho\left(w_1\left(\frac{C_{enb} - Rf_1}{N_1}\right) + w_2\left(\frac{\sum T_j - Rf_2}{N_2}\right)\right)\right] \quad (3.3)$$

İdeal noktalara ağırlıklı uzaklık (D_{IP}) değeri, normalize edilmiş amaç fonksiyonu değerinden ideal noktaya öklid mesafesi ile amaç değeri ağırlığının çarpımı ile elde edilmiştir. Referans noktalara ağırlıklı uzaklık (D_{Ref}) değeri ise, referans noktaya öklid mesafesi ile amaç değeri ağırlığının çarpımı ile elde edilmiştir. D_{IP} ve D_{Ref} değerleri, Eşitlik (3.4)-(3.5) kullanılarak hesaplanmıştır. Bu eşitliklerde I_i, i. amaç fonksiyonunun ideal noktasını temsil etmektedir.

$$D_{IP} = \sqrt{w_1\left(\frac{C_{enb} - I_1}{N_1}\right)^2 + w_2\left(\frac{\sum T_j - I_2}{N_2}\right)^2} \quad (3.4)$$

$$D_{Ref} = \sqrt{w_1\left(\frac{C_{enb} - Rf_1}{N_1}\right)^2 + w_2\left(\frac{\sum T_j - Rf_2}{N_2}\right)^2} \quad (3.5)$$

3.3. KIYASLAMA ÖRNEKLERİ

MATÇ probleminin çözümü için literatürde birçok (meta)sezgisel algoritma geliştirilmiştir. Bu algoritmaların farklılıklarının ve etkisinin anlaşılabilmesi için geliştirilen kıyaslama problem setleri kullanılmaktadır. Burada amaç benzer sistem özellikleri ve değerleri altında farklı algoritmaların performanslarının ölçülmesidir. Literatürdeki MATÇ çalışmaları ile ilgili geliştirilen sezgiseller ve bu sezgisellerin kıyaslandığı diğer çalışmalar ile kullanılan kıyaslama örnekleri ve kullanılan örnek boyutlarını içeren özet tablosu Tablo (3.2)'de sunulmuştur.

Tablo 3.2: MATÇ Problemi İçin Literatürde Çalışılan Kıyaslama (Benchmark) Örnekleri ve Örnek Boyutları

Sn.	Yıl	Yazarlar	Önerilen Sezgisel	Kullanılan Kıyaslama Örnekleri	Kıyaslanan Sezgiseller	Örnek Boyutları
1	2006	Zandieh, FatemiGhomi ve MoattarHusseini	BA	Framinan (2020).	RAGA	$n \in \{6, 30, 100\}$, $s \in \{2, 4, 8\}$, $m \in \{1, 2, 10\}$
2	2007	Alaykýran, Engin ve Döyen	KKO	Carlier and Neron's	DSA	$n \in \{10, 15\}$, $s \in \{5, 10\}$
3	2008	Tseng ve Liao	PSO	Sivrikaya-Serifoğlu ve Ulusoy(2004)	GA, KKS	$n \in \{5, 10, 20, 50, 100\}$, $s \in \{2, 5, 8, 10\}$, $m \in [1,100]$
4	2009	Jouglet, Oğuz ve Sevaux	MA	Sivrikaya-Serifoğlu ve Ulusoy(2004)	GA, GA_OE	$n \in \{5, 10, 20, 50, 100\}$, $s \in \{2, 5, 8\}$
5	2009	RezaAmin-Naseri ve AliBeheshti-Nia	3BGA	RDÜ	3BGA, S1, S2, S3	$n \in \{10, 50, 100\}$, $s \in \{2, 10, 20\}$, $m \in [1,4\text{-}2,8]$
6	2010	Luo. vd.	DSA	RDÜ	CPLEX	$n \in \{10, 12, 15, 20\}$, $m \in \{2, 3, 5, 10\}$
7	2010	Syam ve Al-Harkan	GA , BT , TAA	Tailard's	GA, BT, TAA	$n \in \{20, 50, 100, 200, 500\}$, $s \in \{5, 10, 20\}$
8	2010	Wang, Chou ve Wu	BT	Oğuz and Ercan(2005)	GA, MA, KKS, PSO	$n \in \{5, 10, 20, 50, 100\}$, $s \in \{2, 5, 8\}$
9	2011	Chaari vd.	GÇGA	Carlier and Neron's		$n \in \{10, 15, 20, 50, 100\}$, $s \in \{2, 5\}$, $m \in [4, 15]$
10	2011	Engin, Ceran ve Yilmaz	GA	Oğuz and Ercan(2005)	GA, PAA	$n \in \{5, 10, 20, 50, 100\}$, $s \in \{2, 5, 8\}$, $m \in [1,5]$
11	2011	Xu ve Wang	DE	RDÜ	GA	$n \in \{12\}$, $s \in \{3, 4\}$, $m \in \{2, 3, 4\}$
12	2012	Behnamian ve Zandieh	HBT	RDÜ	PSO, BT, DKA	$n \in \{6, 30, 100\}$, $s \in \{2, 4, 8\}$, $m \in [1, 10]$
13	2012	Liao, Tjandradjaja, Chung	PSO	Carlier and Neron's	PSO, KBA, KKO, YBS, DSA	$n \in \{10, 15\}$, $s \in \{5, 10\}$
14	2012	Niu vd.	KA	RDÜ	BA, KA, GA, KBA, UBA, RYSKA, URYSKBA	$n \in \{10, 30, 40, 60\}$, $s \in \{3, 5, 10\}$, $m \in [2,6]$
15	2012	Pargar ve Zandieh	SABA	RDÜ	GA	$n \in \{20, 40, 50, 60, 80, 100\}$, $s \in \{4, 8\}$, $m \in [1,5\text{-}2,8]$
16	2013	Bożejko, Pempera ve Smutnicki	KTPTA	Tailard's		$n \in \{20, 50, 100, 200, 500\}$, $s \in \{5, 10, 20\}$, $m \in \{2, 3, 4\}$

Tablo 3.2 (Devam): MATÇ Problemi İçin Literatürde Çalışılan Kıyaslama (Benchmark) Örnekleri ve Örnek Boyutları

Sn.	Yıl	Yazarlar	Önerilen Sezgisel	Kullanılan Kıyaslama Örnekleri	Kıyaslanan Sezgiseller	Örnek Boyutları
17	2013	Chou	PSO	Oğuz and Ercan(2005) , Sivrikaya-Serifoğlu ve Ulusoy(2004)	TDSBTA, GA, KP, MA, GA_SU, PSOTL, BT	$n \in \{5, 10, 20, 50, 100\}$, s ∈ {2, 5, 8}
18	2013	Chung ve Liao	İBYBSA	Carlier and Neron's	GA, KKO	$n \in \{10, 30, 50, 70, 100\}$, s ∈ {5, 10}
19	2013	Costa, AntonioCappadonna, Fichera	ÇKM, TKGA	RDÜ	GA, YBS, PSO, BT	$n \in \{10, 30, 50, 70, 100\}$, s ∈ {3, 5}, m ∈ {3, 5}
20	2013	Elmi ve Topaloglu	BT	RDÜ	STBÖ	$n \in \{75, 80, 120, 200\}$, s ∈ {2, 4, 6, 8, 10}
21	2013	Sarin	İAÇLAP	RDÜ	CPLEX	Lotlar {5, 10, 15}
22	2013	Wang ve Liu	GA	RDÜ	GA	n ∈ {10, 30, 50, 100, 150, 200}, m ∈ {2, 3, 5, 10}
23	2013	Xu Y. vd.	KKZA	Sivrikaya-Serifoğlu ve Ulusoy(2004)	GA, KKS, PSO, BT1, BT2	$n \in \{5, 10, 20, 50, 100\}$, s ∈ {2, 5, 8, 10}, m ∈ [1,5]
24	2013	Yalaoui vd.	BMKBGADYÇY, BMKBPSODYÇY	Gerçek Veriler	GA, PSO	$n \in \{10, 12, 14, 16, 18\}$, s ∈ {2}, m ∈ {5}
25	2013	Yang J.	S1	RDÜ	S1, JK, ATÇK, ÇK	$n \in \{10, 100, 1000\}$, s ∈ {2}, m ∈ {2, 10, 20}
26	2014	Ebrahimi, FatemiGhomi, Karimi	BOSGAII , ÇAGA	RDÜ	ÇFGA	$n \in \{12, 15, 49, 64, 110, 144\}$, s ∈ {3, 4, 5 6, 7, 8}, m ∈ {2, 3, 4}
27	2014	Elmi ve Topaloglu	BT	RDÜ		$n \in \{75, 80, 120, 200\}$, s ∈ {2, 4, 6, 8, 10}
28	2014	Eskandari ve Hosseinzadeh	DKA	RDÜ	EEHZ, EKİZ, EUİZ, KEKİS, EUKİS	$n \in \{20, 40, 60, 80, 100\}$, s ∈ {3, 5, 8}, m ∈ [20, 120]
29	2014	Hadda, Dridi ve Hajri-Gabouj	DSA	RDÜ		$n \in \{50, 100, 150, 200, 250, 300\}$, s ∈ {2}, m ∈ {2, 5, 10}

Tablo 3.2 (Devam): MATÇ Problemi İçin Literatürde Çalışılan Kıyaslama (Benchmark) Örnekleri ve Örnek Boyutları

Sn.	Yıl	Yazarlar	Önerilen Sezgisel	Kullanılan Kıyaslama Örnekleri	Kıyaslanan Sezgiseller	Örnek Boyutları
30	2014	Marichelvam, Prabaharan ve Yang	GKAA	Gerçek Veriler	GA, BT, KKO, PSO, GKAA, NEH	$n \in \{10, 50, 100\}$, s ∈ {2, 5}, m ∈ {5, 10, 20}
31	2014	Wang S. vd.	DAİFSTT	RDÜ	DASİMDT, TSMODAİFSTT, ADY	$n \in \{20, 25, 30, 35, 40\}$, s ∈ {2, 4, 6, 8, 10, 12}, m ∈ {2, 3, 4, 5, 6}
32	2014	Wang, Choi ve Qin	DASİPDT	Chaari ve ark. (2011).	GA, DAT	$n \in \{10, 15, 20, 50, 100\}$, s ∈ {2, 5, }
33	2015	Cui ve Gu	GAYAK	Carlier and Neron's	DSA, YBS, KKO, GA, KBA, PSO	$n \in \{10, 15, 30\}$, s ∈ {5, 10}
34	2015	Lei ve Guo	KKZA	RDÜ	GA, BT	$n \in \{30, 40, 50, 60, 80, 100\}$, s ∈ {5, 10}, m ∈ {30}
35	2015	Li ve Pan	BYAKTAS	RDÜ	RMGA, YGA, TADAİS, TAA, AYAK	$n \in \{20, 40, 60, 80, 100, 200\}$, s ∈ {5, 10, 15, 20, 25, 30}, m ∈ {5, 10}
36	2015	Su ve Yi	YSGA	Carlier and Neron's	İDÇGA, GDÇGA	$n \in \{10, 15\}$, s ∈ {5, 10}
37	2015	Sun ve Yu	LGA	Gerçek Veriler		$n \in \{3, 4\}$, m ∈ {2, 3, 4, 5}
38	2016	Feng, Zheng ve Xu	YG	RDÜ	YG, TTYAA, TTSTA	$n \in \{4, 6, 10\}$, s ∈ {2}, m ∈ {2, 3, 4}
39	2017	Laxmi, Mohan ve Madhusudanan	KTDP	RDÜ		$n \in \{2, 3, 4, 5, 6, 7\}$, s ∈ {2, 3, 4}, m ∈ {2, 3}
40	2017	Lei ve Zheng	GDKA	RDÜ	GDKA, KAA, GDKA1, DKA, ÇATA, BOSGAII	$n \in \{30, 40, 50, 60, 80, 90, 100, 120\}$, s ∈ {5, 8, 10}
41	2017	Liu vd.	YS , GA , BT	RDÜ	YS, GA, BT	$n \in \{6, 8, 10\}$, s ∈ {2}, m ∈ {3, 4}
42	2018	Dios, Fernandez-Viagas ve Framinan	NEH , RAJ	Carlier and Neron's	NEH, RAJ	$n \in \{5, 6, 7, 8\}$, s ∈ {3, 5, 10, 15}, m ∈ [1,5]
43	2018	Pargar vd.	BOSHSABA ,SKSHSABA	RDÜ	BOSGAII, BOSGA	$n \in \{6, 10, 20, 40, 80, 100\}$, s ∈ {2, 4, 8}, m ∈ [1,5-2,8]

Tablo 3.2 (Devam): MATÇ Problemi İçin Literatürde Çalışılan Kıyaslama (Benchmark) Örnekleri ve Örnek Boyutları

Sn.	Yıl	Yazarlar	Önerilen Sezgisel	Kullanılan Kıyaslama Örnekleri	Kıyaslanan Sezgiseller	Örnek Boyutları
44	2018	YiZhang ve Chen	ADEA	Gerçek Veriler		$n \in \{10, 20, 30\}$, s ∈ {2, 3, 4}, m ∈ {3, 5, 7}
45	2018	Yu, Semeraro ve Matta	GA	RDÜ	BT, GA, YAK, NEH	n ∈ {20, 50, 100}, m ∈ {5, 10, 20}
46	2019	Fernandez-Viagas, Perez-Gonzalez ve Framinan	KTDP	RDÜ		$n \in \{6, 7, 8, 9, 10, 11\}$,s ∈ {2, 3, 4, 5}
47	2019	Meng vd.	KTDP	RDÜ	MATÇP-İPM-SBHS-E, MATÇP-İPM-SBHS-B	
48	2019	Wang S. vd.	TAA , KKO	RDÜ	AEKKTPM, YS, TAA, KKO	$n \in \{5, 8, 10, 15, 20, 40, 50, 80, 100, 120\}$, m ∈ {5, 6, 7, 30, 40, 50}
49	2020	Aqil ve Allali	YYAM , YAA	RDÜ	YYAM, YAA	$n \in \{10, 20, 30, \ldots, 200\}$, s ∈ [2, 6,10], m ∈ [2, 6,12]
50	2020	Cai, Zhou ve Lei	DKKZA	Ying and Lin (2018)	KKAYA, HYAKA, PSO	$n \in \{5, 10, 20, 50, 100\}$, s ∈ {2, 5, 8}, F ∈ {2, 3, 4, 5, 6}
51	2020	Costa, Fernandez-Viagas ve Framinan	GİAOATA	RDÜ	AYAK, HUA, GA, GPSO	$n \in \{20, 50, 100\}$, s ∈ {25, 100}, m ∈ {5, 10}
52	2020	Fernandez-Viagas, Costa ve Framinan	BFS	RDÜ	NEH, HBTYS	$n \in \{20, 50, 100\}$, s ∈ {5, 10, 20}, m ∈ {4, 5, 6}
53	2020	Harbaoui ve Khalfallah	TAA	RDÜ	TAA, GA	$n \in \{10, 15, 20, 30, 50, 100, 200\}$, s ∈ {2} m ∈ {2,10}
54	2020	Marichelvam, Geetha ve Tosun	GPSO	RDÜ	GA, TAA, SABA, HDKA, OYDDMA	$n \in \{20, 50, 100\}$, s ∈ {5, 10, 20}, m ∈ {2, 3, 5}
55	2020	Rolf vd.	GA	Gerçek Veriler	GA, EEBT, MGS, EKHZ, EKHZ	
56	2020	Shao, W., Shao, Z., ve Pi	DNA , YAA	RDÜ	SAYAA, İAYAA, YAA_Ribas, YYA_Ribas, TAA, KKAYA	$n \in \{40, 60, 80, 100\}$, s ∈ {5, 10, 15}, m ∈ [1,5]

Tablo 3.2 (Devam): MATÇ Problemi İçin Literatürde Çalışılan Kıyaslama (Benchmark) Örnekleri ve Örnek Boyutları

Sn.	Yıl	Yazarlar	Önerilen Sezgisel	Kullanılan Kıyaslama Örnekleri	Kıyaslanan Sezgiseller	Örnek Boyutları
57	2020	Wang, J. ve Wang, L.	İPİMA	RDÜ	GUROBİ, YAA	$n \in \{10, 15, 20, 50, 100\}$, s ∈ {2, 5, 8}, F ∈ {2, 3, 4}, m ∈ [1,5]
58	2021	Dabiri vd.	S1,S2	RDÜ	GA, EDGEKO, GMSO, AYAK, İBYBSA	$n \in \{6, 8, 10, 12, 20, 50, 80, 120\}$, s ∈ {2, 3, 4, 8}, m ∈ {2, U [1, 3]}
59	2021	Han vd.	SKÇÇAEA	Tailard's	BOSGAII, BOKBA	$n \in \{5, 10, 20, 50, 100\}$, s ∈ {2, 5, 8} m ∈ [2,3]
60	2021	Li vd.	GA	RDÜ	PSO, GA	
61	2021	Lian vd.	ADGÇAEA	RDÜ	BOSGAII, GPEA2, YAÇAHGA	$n \in \{15, 25, 40\}$
62	2021	Ozsoydan ve Sağir	YAA	RDÜ	EİZDS, DS, HZDS, ASÇES, TSÇES, MÇES, JKTS	$n \in \{20, 30, 50, 80, 100\}$
63	2021	Zhang vd.	ABSA	RDÜ	GA, PSO, 3BGA, BSA	$n \in \{3, 5, 10, 15, 20, 25\}$
64	2022	Fernandez-Viagas	YAAwS	Fernandez-Viagas ve Framinan (2020)	AYAK	$n \in \{40, 80, 120, 160, 200, 240\}$,$m \in \{5, 10, 15, 20\}$

RDÜ: Rasssal Dağılımla Üretilen

Bu çalışmada ele alınan MATÇ probleminin sıra bağımlı hazırlık süresi, taşıma süresi ve öğrenme etkili işlem süresi özelliklerini içermesi sebebiyle literatürde problemin bu parametrelerini bir arada karşılayan kıyaslama seti bulunamamıştır. Bu nedenle örnek setleri literatürde en sık kullanılan örnek boyutlarını dikkate alarak düzgün dağılımla üretilmiştir.

Tasarlanan ÇAGA'nın performans ölçümü için oluşturulan örnek setleri küçük, orta ve büyük olmak üzere 3 ana gruba ayrılmıştır. Bu setin oluşturulmasında, Naderi, Zandieh ve Shirazi (2009) çalışmasından, iş sayısı (n), aşama sayısı (t), her bir aşamadaki makine sayısı (m_t), hazırlık süresi (S_{ijtm}) ve işleme süreleri (P_{jtm}) için veri üretme yöntemi dikkate alınmıştır. Buna göre, üretilen test örnekleri, n ve t'nin kombinasyonlarından oluşturulmuştur: küçük örnek seti için, n = {4, 6, 8}, t = {2, 3, 4}, orta ve büyük örnek setleri için, n = {20, 50, 80, 120} ve t = {2, 4, 8}. İşlem süreleri, çizelgeleme literatüründe sıklıkla kullanılan [1, 99] arasında düzgün dağılım kullanılarak üretilmiştir. Hazırlık süresi, [1,75] arasında düzgün dağılımla üretilirken, her bir aşamadaki paralel makinaların sayısı küçük örnek seti için [1,2], orta ve büyük örnek seti için [1, 4] arasında düzgün dağılıma göre üretilmiştir. Taşıma süreleri de [1, 30] arasında düzgün dağılımla üretilmiştir. Tablo (3.3)'te oluşturulan örnek setinin elemanları ve elemanların aldıkları değerler/değer aralıkları özetlenmiştir.

Tablo 3.3: Oluşturulan Örnek Seti ve Örnek Büyüklükleri

		Küçük	Orta	Büyük
İş sayısı	:	4, 6, 8	20, 50	80, 120
Aşama sayısı	:	2, 3, 4	2, 4, 8	2, 4, 8
Makine dağılımı	:	U~[1,2]	U~[1,4]	U~[1,4]
İşlem zamanı	:	U~[1,99]		
Hazırlık zamanı	:	U~[1,75]		
Taşıma zamanı	:	U~[1,30]		

Teslim süresini oluşturmak için kullanılan adımlar:

1. Her bir işin bütün aşamalardaki işlem zamanı hesaplanır,

$$p_j = \sum_{i=1}^{m} P_{j,i}, \forall j$$

2. Olası tüm iş sıralamaları için hazırlık ve taşıma sürelerinin ortalaması hesaplanır ve bütün aşamalar için bu sürelerin toplamı alınır.

$$s_j = \sum_{i=1}^{m} \frac{(\Sigma_{k=0,k\neq i}^{n} s_{ijtm}) / n + RT_{it}}{2}, \forall j$$

3. Bütün işlerin teslim tarihi;

$$d_j = (p_j + s_j) \times (1 + rss \times 2), \forall j$$

formülüne göre belirlenir, burada *rss* ifadesi [0,1] aralığında düzgün dağılımla oluşturulan rassal sayıdır.

3.3.1. Kıyaslama Örneklerinden Elde Edilen Algoritma Parametreleri

Tasarlanan ÇAGA'nın parametre ayarı için tanımlanan parametre aralıkları Tablo (3.4)'te verilmiştir.

Tablo 3.4: Parametre Aralıkları ve Elde Edilen En İyi Parametre Değerleri

		Parametre Değer Aralıkları	Elde Edilen En iyi Parametre Değerleri		
			Küçük	**Orta**	**Büyük**
Popülasyon Büyüklüğü	:	U ~ [20 ; 100]	20	20	22
Çaprazlama Olasılığı	:	U ~ [0,7 ; 1]	0,92	0,85	0,86
Mutasyon Olasılığı	:	U ~ [0,01 ; 0,5]	0,26	0,18	0,19
Turnuva Büyüklüğü	:	2			
w_1	:	U ~ [0,09 ; 0,91]	0,43	0,09	0,19
w_2	:	U ~ [0,09 ; 0,91]	0,57	0,91	0,81

Algoritmanın uygunluk değeri hesaplama fonksiyonu ATY, KSY ve TSY kullanılarak 3 farklı şekilde tasarlanmıştır (Bkz., Eşitlik (2.18)-(2.20)). Algoritma her bir yöntem için, her örnek seti, 31 kez çalıştırılmıştır. Elde edilen sonuçlar bir sonraki bölümde analiz edilip yorumlanmıştır.

DÖRDÜNCÜ BÖLÜM
DENEY SONUÇLARI VE ANALİZ

4.1. SKALERLEŞTİRME YÖNTEMLERİNE DAYALI GA'NIN PERFORMANS KARŞILAŞTIRMASI

Skalerleştirme yöntemlerine dayalı GA'nın performansını değerlendirmek için, küçük, orta ve büyük boyutlu 21 örnek setini içeren (Tablo (4.1)'de No#1-#21) MATÇ modeli kullanılmıştır. KSY'li GA ve TSY'li GA, referans noktalarına sahip olduğu için, bu algoritmalar her bir kıyaslama örneği için 3 farklı KKV kullanılarak analiz edilmiştir. Elde edilen sonuçlar, her bir performans metriğine göre her bir kıyaslama test problemi için ATY'li GA ile karşılaştırılmıştır. Her bir deney, skalerleştirme yöntemlerine dayanan GA'nın üç farklı referans noktası dikkate alınarak her bir performans metriği için 31 kez tekrar edilmiştir.

Tablo (4.1), 21 kıyaslama örneği için performans metriklerine göre MATÇ probleminin skalerleştirme yöntemleriyle GA'nın performanslarının karşılaştırılmasını göstermektedir. Sonuçlar, ideal noktaya ağırlıklı uzaklık (D_{IP}) performans metriği açısından ATY'li GA'nın diğer algoritmalara göre daha iyi performans sergilediğini göstermektedir. Ayrıca, Tablo (4.1)'den görüldüğü gibi referans noktaya ağırlıklı uzaklık (D_{Ref}) performans metriği açısından TSY'li GA'nın, KSY'li GA'ya göre daha iyi performans sergilediğini göstermektedir. Skalerleştirme yöntemlerine dayalı GA'nın performansı, 21 test problemi için, her bir performans metriğine dayanan her bir algoritma çiftinin 31 çalışmasından elde edilen sonuçlara Wilcoxon signed-rank testi uygulanmıştır (Tablo (4.1)). Seçilen bir örnekte, D_{IP} performans metriği için ATY'ye vs. KSY, > veya < (≥ veya ≤), ATY'nin (KSY) KSY'den (ATY) (biraz) daha iyi veya (biraz) daha kötü performans gösterdiğini ifade etmektedir. Ek olarak, ≈, algoritmaların benzer performans gösterdiğini ifade etmektedir. Bu performans farkının %95 anlamlılık düzeyinde istatistiksel olarak anlamlı (ya da anlamlı olmadığını) olduğunu göstermektedir. Skalerizasyon yöntemlerine dayanan ÇAGA'nın farklı ikili karşılaştırmalarının toplam sayısı, D_{IP} performans metriği için 189 ikili karşılaştırma ve D_{Ref} performans göstergesi için 63 ikili karşılaştırma içeren toplam 252 (21×3×4) kadardır.

D_{IP}'de 189 ikili karşılaştırmanın 68'inde, ATY'li GA, diğer algoritmalardan daha iyi performans göstermektedir. Bu durumda, 68 ikili karşılaştırmanın 21'i, anlamlı derecede diğer algoritmalardan daha iyi performans gösterirken, geri kalanı biraz daha iyi performans göstermektedir. 189 ikili karşılaştırmanın 66'sında, KSY'li GA, diğer algoritmalardan daha iyi performans göstermektedir. Bu durumda, 66 ikili

karşılaştırmanın 18'i, anlamlı derecede diğer algoritmalardan daha iyi performans gösterirken, geri kalanı biraz daha iyi performans göstermektedir. Bu durumda, KSY'li GA az farkla da olsa MATÇ problem örnekleri üzerinde ikinci en iyi algoritmadır. Diğer taraftan, D_{Ref}'de 63 ikili karşılaştırmanın 33'ünde, TSY'li GA, KSY'li GA'dan daha iyi performans göstermektedir. Sonuç olarak, D_{IP}'de ATY'li GA'nın başarısı ve D_{Ref}'te TSY'li GA başarısı Tablo (4.1)'de görülmektedir.

Tablo 4.1: 21 Kıyaslama Örneği İçin Performans Metriklerine Göre MATÇ Probleminin Skalerleştirme Yöntemleriyle GA'nın Performanslarının Karşılaştırılması

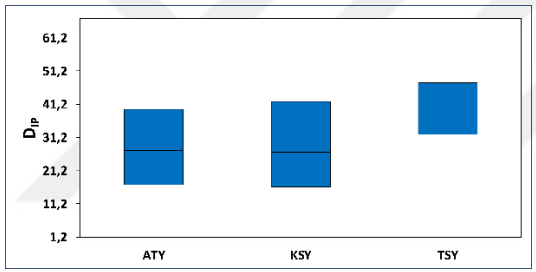
		KKV$_1$				**KKV$_2$**				**KKV$_3$**			
		D_{IP}			**D_{Ref}**	**D_{IP}**			**D_{Ref}**	**D_{IP}**			**D_{Ref}**
Boyut	**No#**	ATY vs. KSY	ATY vs. TSY	KSY vs. TSY	KSY vs. TSY	ATY vs. KSY	ATY vs. TSY	KSY vs. TSY	KSY vs. TSY	ATY vs. KSY	ATY vs. TSY	KSY vs. TSY	KSY vs. TSY
	1	≈	≥	≥	≥	≈	≥	≥	≤	≈	≥	≥	≥
	2	≥	≤	≥	≥	≤	≤	≥	≥	≤	≤	≥	≥
	3	≥	>	≥	≥	≥	>	≥	≤	≥	>	≥	≥
	4	≥	≥	≤	≤	≥	≥	≤	≤	≥	≤	≤	≤
Küçük	**5**	≤	>	>	≤	≤	>	>	<	≤	>	>	>
	6	≤	>	>	>	≤	>	>	≥	≤	>	>	>
	7	≤	>	>	≥	≤	>	>	<	≤	>	>	>
	8	≤	>	>	>	≤	>	>	>	≤	>	>	>
	9	≥	>	>	≥	≥	>	>	≤	≥	>	>	>
	10	≥	≤	≤	≤	≥	≤	≤	≤	≥	≤	≤	≤
	11	≥	≥	≤	≤	≥	≥	≤	≤	≥	≥	≤	≤
Orta	**12**	≥	≤	≤	≤	≥	≤	≤	≤	≥	≤	≤	≤
	13	≤	≥	>	>	≤	≥	>	>	≤	≥	>	>
	14	≤	<	≤	≤	≤	<	≤	≤	≤	<	≤	≤
	15	≥	≥	≤	≤	≥	≥	≤	≤	≥	≥	≤	≤
	16	≥	≤	≤	≤	≥	≤	≤	≤	≥	≤	≤	≤
	17	≤	≥	≥	≥	≤	≥	≥	≥	≤	≥	≥	≥
Büyük	**18**	≤	≤	≥	≥	≤	≤	≥	≥	≤	≤	≥	≥
	19	≥	≤	≤	≤	≥	≤	≤	≤	≥	≤	≤	≤
	20	≤	<	≤	≤	≤	<	≤	≤	≤	<	≤	≤
	21	≥	>	≥	≥	≥	>	≥	≥	≥	>	≥	≥

Bu analizlere ek olarak, küçük, orta, büyük boyutu içeren örnek büyüklüklerine göre her bir örnek değerlendirildiği zaman D_{IP} performans indikatöründe küçük boyutlu örneklerde 78 ikili karşılaştırmanın 39'unda KSY'li GA, diğer algoritmalardan daha iyi performans göstermektedir. Orta boyutlu örneklerde 54 ikili karşılaştırmanın 24'ünde,

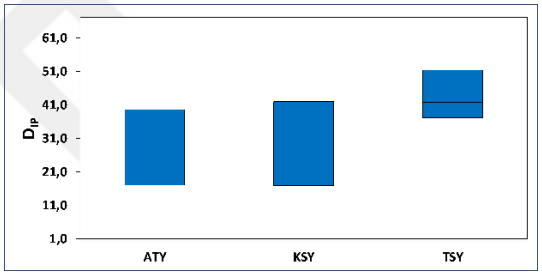
TSY'li GA, diğer algoritmalardan daha iyi performans göstermektedir. Büyük boyutlu örneklerde 54 ikili karşılaştırmanın 21'inde, TSY'li GA, diğer algoritmalardan daha iyi performans göstermektedir. D_{Ref} performans indikatörü dikkate alınarak örnek boyutları incelendiğinde, küçük boyutlu örneklerde 27 ikili karşılaştırmanın 18'inde KSY'li GA, TSY'li GA'dan; orta boyutlu örneklerde 18 ikili karşılaştırmanın 15'inde, TSY'li GA, KSY'li GA'dan; büyük boyutlu örneklerde 18 ikili karşılaştırmanın 9'unda, KSY'li GA, TSY'li GA'dan daha iyi performans göstermektedir.

Örnek seti içerisinden küçük, orta ve büyük örnek boyutlu rasgele seçilen örnekler üzerine skalerleştirme yöntemlerine dayanan GA'nın 31 kez çalıştırma sonucu elde edilen her bir performans metriğine ait kutu grafikleri Şekil (4.1)-(4.2)'de verilmiştir. Üretilen üç farklı konveks kombinasyonuna bağlı olarak, sırasıyla küçük, orta ve büyük boyutlu örnek setinden seçilen no#9 (D_{IP} için Şekil 4.1 (a)-(c) ve D_{Ref} için Şekil 4.2 (a)-(c)), no#15 (D_{IP} için Şekil 4.1 (d)-(f) ve D_{Ref} için Şekil 4.2 (d)-(f)) ve no#21 (D_{IP} için Şekil 4.1 (g)-(i) ve D_{Ref} için Şekil 4.2 (g)-(i)) analiz edilmiştir. D_{IP} performans metriği dikkate alındığında, ATY'li GA yaklaşımı küçük, orta ve büyük boyutlu örneklerde diğer algoritmalara göre daha iyi bir performans sergilemiştir. KSY'li GA, ATY'li GA ile hemen hemen benzer performans sergilese de ikinci en iyi performans sergileyen algoritmadır. Diğer taraftan, D_{Ref} performans metriğine göre, orta boyutlu örnek olan no#15'in tüm konveks kombinasyonlarında KSY'li GA ile TSY'li GA hemen hemen benzer performans göstermiştir (Bkz., Şekil 4.2 (d)-(f)). KKV_2 için no#9'da (Bkz., Şekil 4.2 (b)) TSY'li GA, KSY'li GA'dan daha iyi performans sergilerken KKV1 ve KKV3 için no#9 (Bkz., Şekil 4.2 (a) ve (c)) ve üç konveks kombinasyon için no#21 (Bkz., Şekil 4.2 (g)-(i)) örneklerinde KSY'li GA, TSY'li GA'dan daha iyi performansa sahip olduğu görülmektedir. Yapılan analiz sonucu, D_{IP} performans metriğine göre ATY'li GA'nın diğer algoritmalara göre daha iyi performansa sahip olduğu, D_{Ref} performans metriğine göre KSY'li GA'nın TSY'li GA'da daha iyi performans sergilediği görülmektedir.

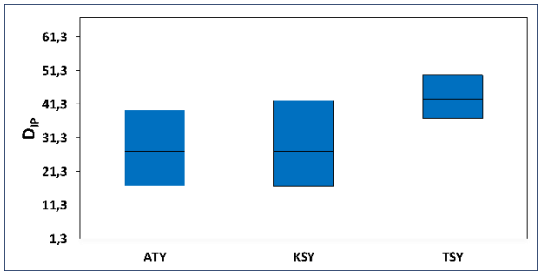
Şekil 4.1: Üç Farklı Konveks Kombinasyon Vektörü ile Her Bir Boyuttan Alınan Üç Numune Örneği İçin Skalerleştirme Yöntemlerine Dayanan GA'nın 31 Çalıştırmasından Elde Edilen İdeal Noktalara Ağırlıklı Uzaklıkların Kutu Grafikleri



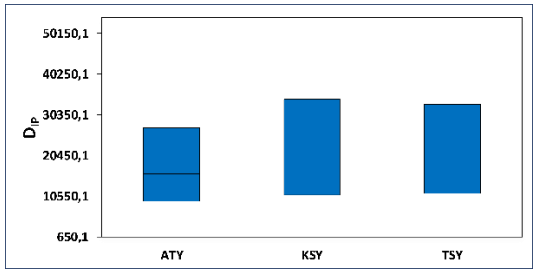
(a) KKV_1 için No#9



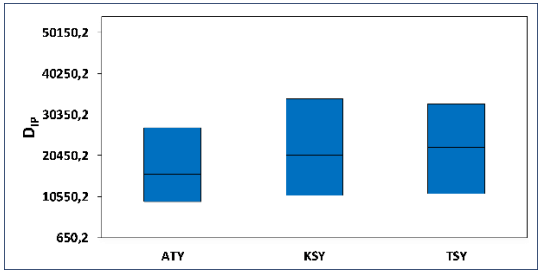
(b) KKV_2 için No#9



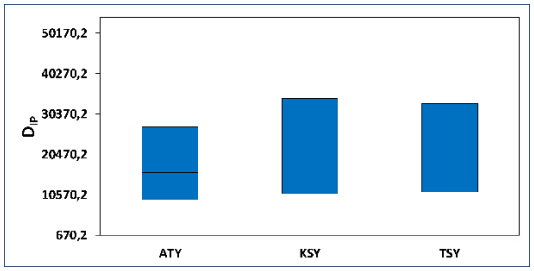
(c) KKV_3 için No#9



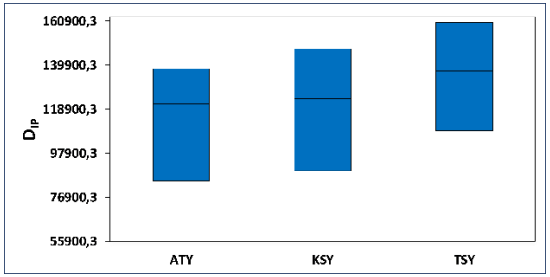
(d) KKV_1 için No#15



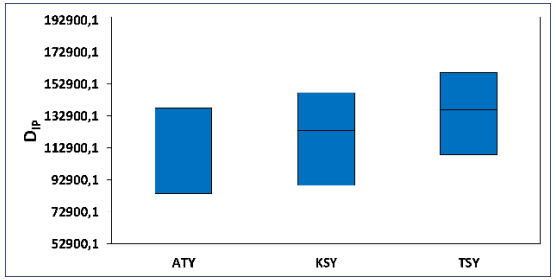
(e) KKV_2 için No#15



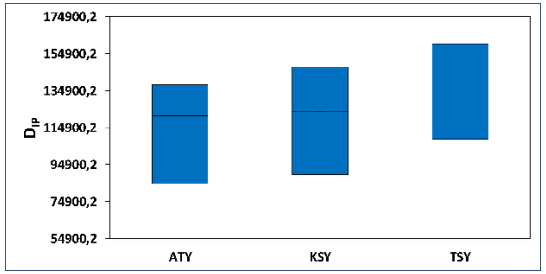
(f) KKV_3 için No#15



(g) KKV_1 için No#21

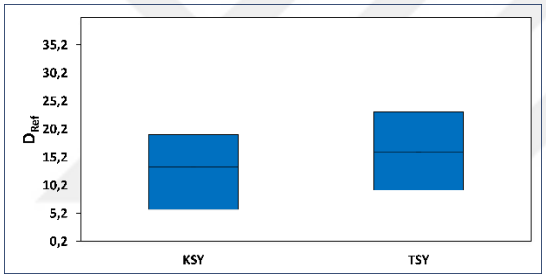


(h) KKV_2 için No#21

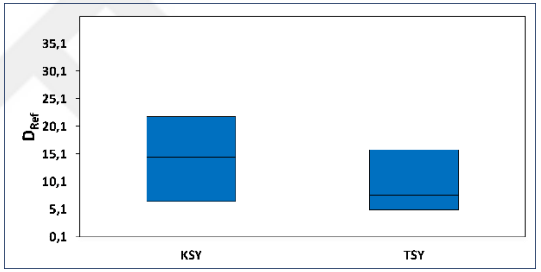


(i) KKV_3 için No#21

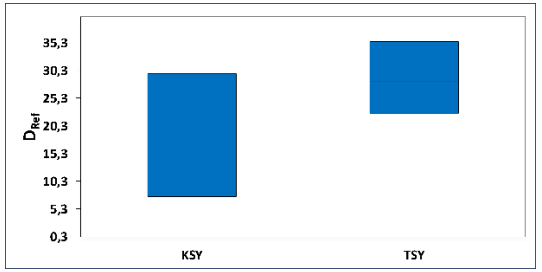
Şekil 4.2: Üç Farklı Konveks Kombinasyon Vektörü ile Her Bir Boyuttan Alınan Üç Numune Örneği İçin Skalerleştirme Yöntemlerine Dayanan GA'nın 31 Çalıştırmasından Elde Edilen Referans Noktalara Ağırlıklı Uzaklıkların Kutu Grafikleri



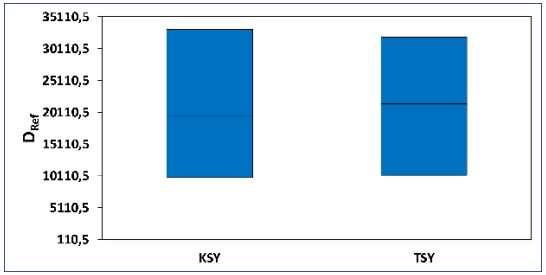
(a) KKV_1 için No#9



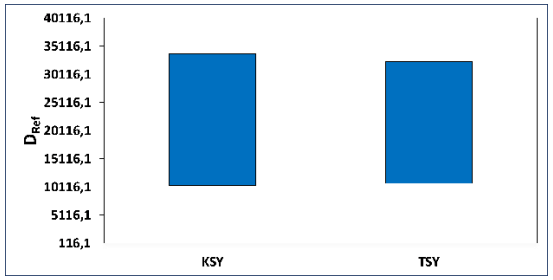
(b) KKV_2 için No#9



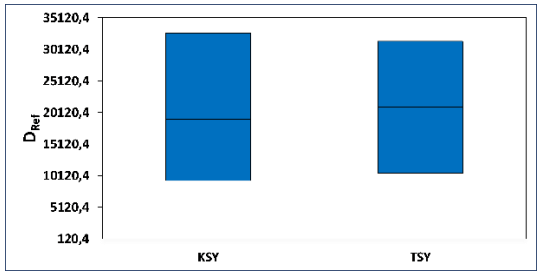
(c) KKV_3 için No#9



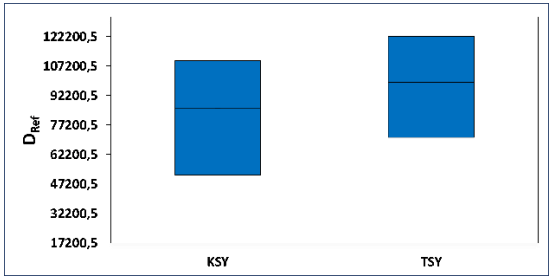
(d) KKV_1 için No#15



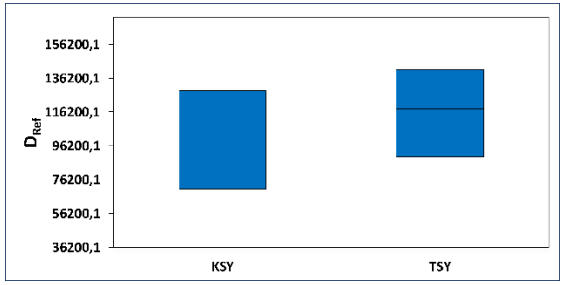
(e) KKV_2 için No#15



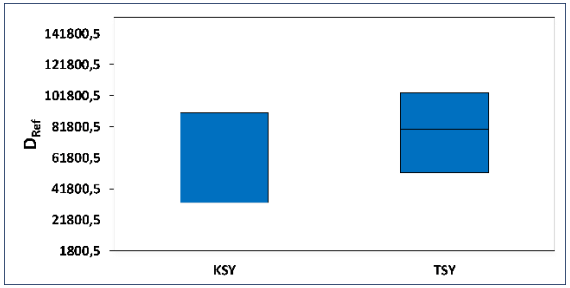
(f) KKV_3 için No#15



(g) KKV_1 için No#21



(h) KKV_2 için No#21



(i) KKV_3 için No#21

BEŞİNCİ BÖLÜM
KARAR DESTEK SİSTEMİ

5.1. KARAR DESTEK SİSTEMİ

Çizelgeleme problemleri temelde bir karar verme sürecini içermektedir. İşlerin hangi makinalarda hangi sırada işleneceği yöneticiler tarafından bir karar verme süreci anlamına gelmektedir. Günümüzün rekabet ortamı, imalat ve hizmet sistemlerinin hızlı karar verme süreçleri gerektirmesine ve gerçek hayat problemlerinin büyük boyutlu ve çok sayıda değişken içermesi nedeniyle bu karar verme sürecinin doğru yürütülmesi geleneksel yöntemlerle sağlanamamaktadır. Bu noktada problemin ihtiyaçlarını karşılayan algoritmaların bilgisayar ortamında geliştirilmesi ile bu yazılımların kullanıcı tarafından yönetilebilir bir sisteme dönüştürülmesi ihtiyacı doğmuştur. Bu amaçlar doğrultusunda geliştirilen yazılımlar literatürde Karar Destek Sistemi (KDS) olarak tanımlanmaktadır.

Sprague (1980) çalışmasında karar destek sistemlerinin gereksinimlerini, tasarım sürecinde dikkat edilmesi gereken noktalarını ve sistemin yapısını tanımlamışlardır. Sprague'ye göre;

- Bir KDS'nin doğası gereği, geleneksel toplu veya çevrimiçi işlem işleme sistemlerinden farklı bir tasarım tekniği gerektirir. Analiz ve tasarım için geleneksel yaklaşımların yetersiz olduğu kanıtlanmıştır, çünkü karar verme konusunda kapsamlı tek bir teori yoktur ve karar vericilerin karşı karşıya kaldıkları koşullardaki değişimin hızlı olması nedeniyle, tasarımcılar kelimenin tam anlamıyla "ilk aşamaya ulaşamazlar" çünkü hiç kimse, en azından karar verici veya kullanıcı, sistemin işlevsel gereksinimlerinin ne olması gerektiğini önceden tanımlayamaz. Geliştirmenin doğru bir şekilde ilerlediğinden emin olmak için kullanıcılardan kısa ve hızlı geri bildirim alan bir KDS oluşturulmalıdır. Hızlı ve kolay bir şekilde değişime izin verecek şekilde geliştirilmelidir.
- Bir KDS, her seviyedeki yöneticiler için karar verme desteği sağlamalı ve uygun olduğunda seviyeler arasındaki bütünleşmeye yardımcı olmalıdır. Bu gereklilik, tüm organizasyon düzeylerindeki yöneticilerin yukarıdaki ilk hedefte açıklandığı gibi "zor" sorunlarla karşı karşıya olduğunun farkına varılmasından gelişir. Ayrıca, yöneticiler tarafından dile getirilen önemli bir ihtiyaç, daha büyük bir sorunun ilgili bölümleriyle ilgilenen birkaç yöneticinin karar verme sürecinin entegrasyonu ve koordinasyonudur.

- Bir KDS'nin kullanımı kolay olmalıdır. Bu özelliği tanımlamak için esneklik, kullanıcı dostu, tehdit edici olmayan, vb. gibi çeşitli terimler kullanılmıştır. Bu özelliğin önemi, bir KDS müşterisinin takdir yetkisi ile vurgulanmaktadır. Ağır organizasyonel destek veya grup desteği gerektiren bazı sistemler takdir yetkisini bir şekilde sınırlayabilirse de bir KDS kullanıcısı, daha geleneksel bir işlem sistemi veya gerekli raporlama sistemi kullanıcısına göre daha fazla serbestliğe sahiptir. Bu nedenle, bir KDS, değerli ve kullanışlı olarak kullanıcılarının bağlılığını kazanmalıdır.

Ele alınan MATÇ problemi ile ilgili literatürde çok sayıda çalışma bulunmasına rağmen geliştirilen yöntemlerin kullanıcılar tarafından yönetilmesini sağlayacak kullanıcı dostu KDS tasarımı çok fazla çalışılmamıştır. Hasan, S.M.K. vd. (2011) ve Kim, J.W. (2015) çalışmalarında iş atölyesi sistemi için KDS tasarımı uygulamışlardır. MATÇ problemi gerçek hayat üretim ortamlarının birçoğunu kapsamaktadır ve bu ortamlarda çizelgeleme verimlilik açısından en önemli parametrelerden biridir. Kıyaslama örnekleri sonucunda ATY'li GA daha iyi performans gösterdiği için bu çalışmada, bir gerçek hayat üretim ortamındaki MATÇ probleminin çözümünde ATY'li GA'nın çalıştığı kullanıcı dostu bir arayüze sahip KDS tasarlanmıştır. Önerilen KDS'de, hazırlık, işlem ve taşıma sürelerini ayrı değerlendiren, girilen tüm verilerin veritabanı tablolarında kayıtlı tutulmasını sağlayan ve girdilerden elde edilen sonuçları kullanıcıya kolay anlaşılabilir bir şekilde sunacak bir yapı tasarlanmıştır.

Bu çalışmada önerilen KDS, ASUS 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz özelliklere sahip bilgisayarda Visual Studio 2022 programında C# programlama dili ve SQL veritabanı kullanılarak oluşturulmuştur. Aşağıda belirtilen tüm bilgiler SQL veritabanı tablolarına kaydedilmektedir.

- Aşamalar, makineler ve bunların birleşimiyle oluşturulan üretim ortamının aşama-makine kaydı,
- Ürün listesinin kaydı,
- İşlem, hazırlık ve taşıma sürelerinin kaydı,
- Veritabanında kayıtlı olan ürünler ile oluşturulan iş emirleri,
- GA tarafından oluşturulan çizelgeleme sonuçları.

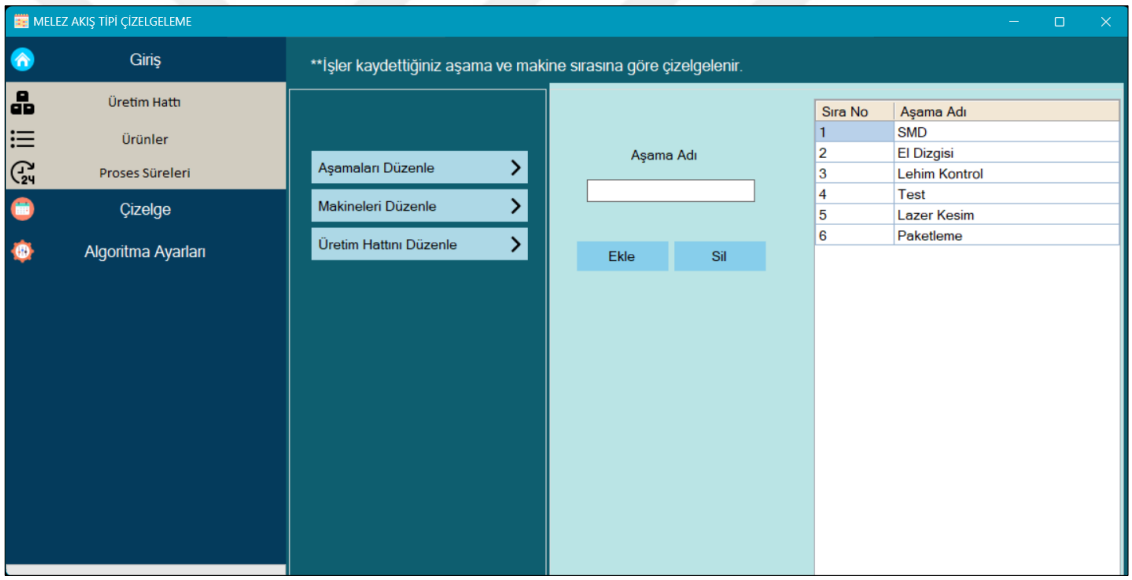
Üretim ortamlarında Excel dosyaları ile çalışmak yaygın ve kullanışlı olduğu için geliştirilen KDS'nin Excel ile bağlantılı olması kullanıcıya ayrı bir kolaylık

sağlayacaktır. Bu nedenle kullanıcılar KDS içerisinde bulunan veritabanı tablolarını .csv formatında Excel'e aktarabilir veya verileri Excel'den alabilirler.

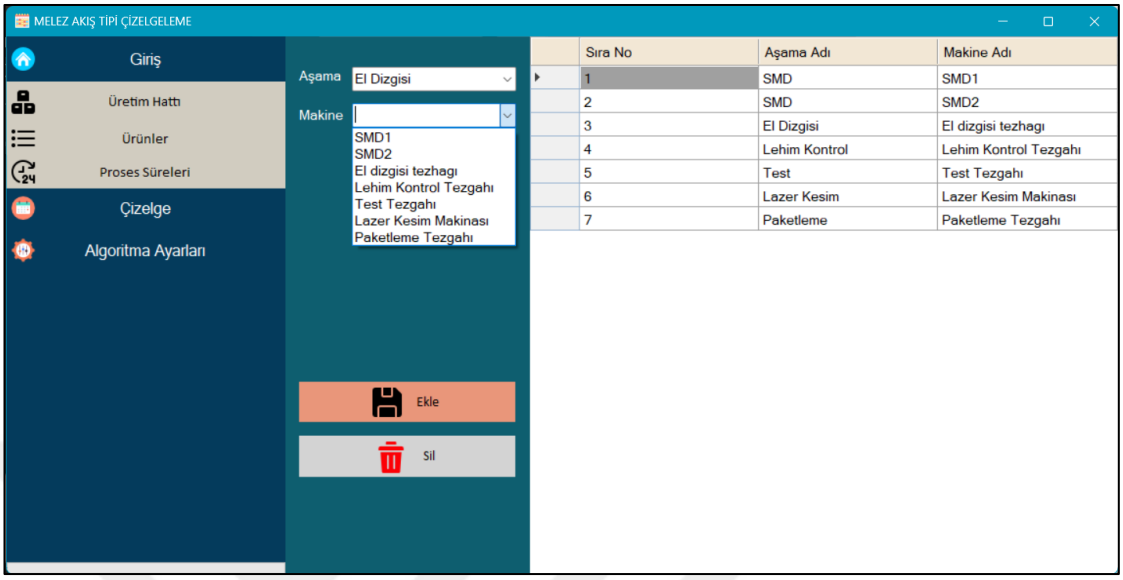
Tasarlanan KDS'nin arayüz görselleri ve açıklamaları aşağıda sunulmuştur.

Oluşturulan KDS'de öncelikle üretim hattının yapısını oluşturan aşama ve makinelerin kaydı ayrı ayrı yapılır (Bkz., Görsel (5.1)). Daha sonra bu veriler kullanılarak aşama-makine tablosu oluşturulur (Bkz., Görsel (5.2)). "Üretim Hattını Düzenle" sayfasında kullanıcı veritabanında kayıtlı olan aşamalara hangi makinelerin geleceğini belirler ve üretim hattının kaydını sağlar. Kaydedilen aşama, makine kombinasyonunun sırası üretim akışını verir. Çizelge çalıştırıldığında işler, kaydedilen aşama-makine sırasına göre çizelgelenir.

Görsel 5.1: KDS Giriş Sekmesi, "Üretim Hattı" Sayfası

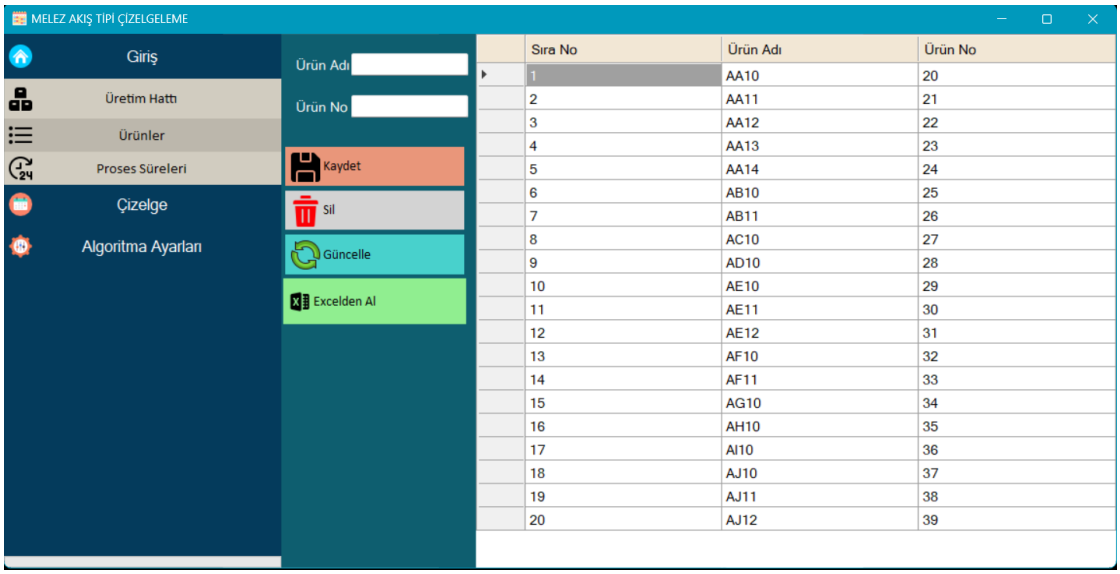


Görsel 5.2: KDS Giriş Sekmesi, “Üretim Hattını Düzenle” Sayfası



Görsel (5.3)’te sunulan “Ürünler” sayfasına üretimi yapılmakta olan ürünlerin listesi kaydedilir. Başlangıçta tüm ürünlerin listesini tek tek girmek vakit alacağı için ürünler toplu olarak, .csv uzantılı formatında oluşturulan bir Excel dosyasından veritabanına aktarılabilir.

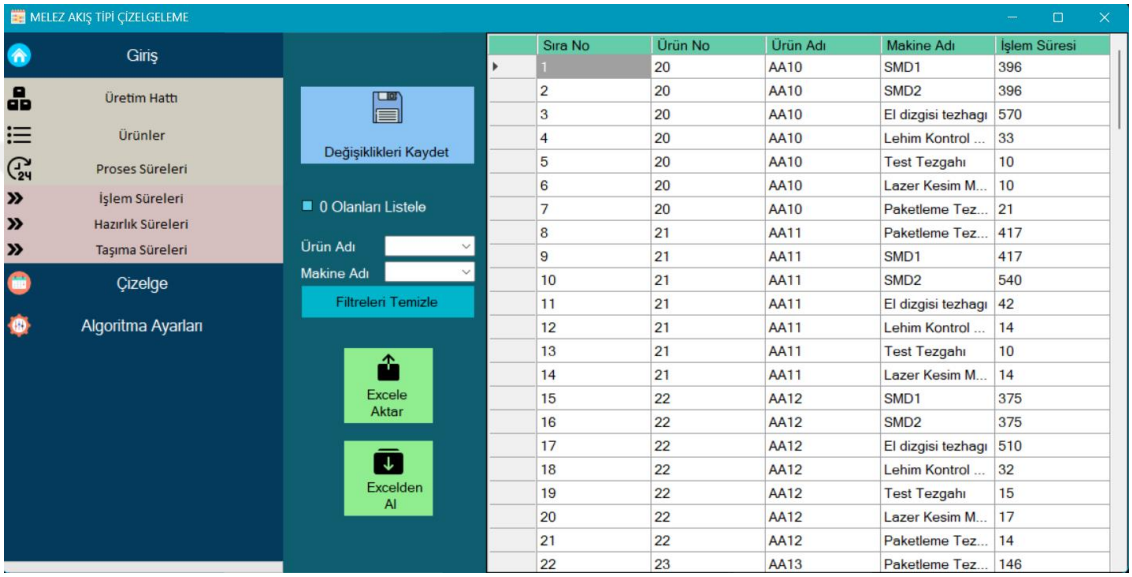
Görsel 5.3: KDS Giriş Sekmesi, “Ürünler” Sayfası



Görsel (5.4)’te sunulan “Proses Süreleri” sekmesinde, hazırlık, taşıma ve işlem süreleri için 3 farklı sekme altında proses sürelerinin kaydı yapılır, eğer kullanıcı isterse bu kayıtları Excel ortamına aktarabilir veya Excel’den alabilir. İşlem sürelerinin kaydının

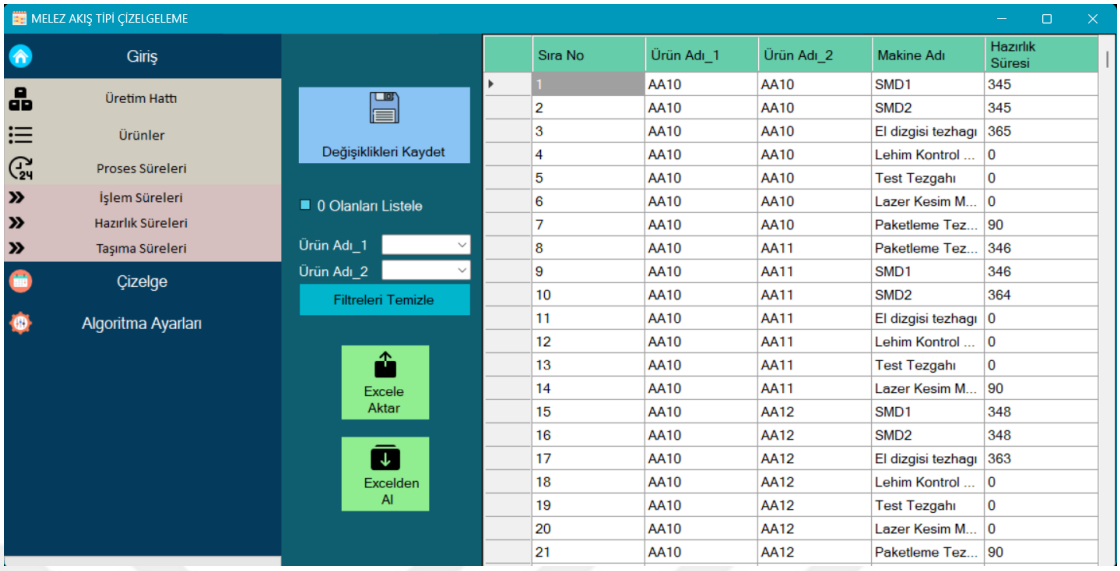
yapılabilmesi için, veritabanında kayıtlı olan ürünler ile makineler tek bir tabloda birleştirilir ve her bir ürünün sisteme tanımlanan makinelerdeki işlem sürelerinin kaydı yapılır. Kullanıcı yalnızca işlem sütunundaki satırlara veri girişi yapabilir. Eğer sisteme yeni ürünler eklenmişse bunların işlem süreleri, 0 olacağı için tablodan ayıklamak zor olabilir. Bu nedenle, sayfanın sol tarafında bulunan filtreleme seçenekleri kullanılarak, işlem süresi 0 olanlara göre, ürünlere veya makinelere göre tablo filtrelenip veri girişi hızlıca yapılabilir.

Görsel 5.4: KDS Giriş Sekmesi, "İşlem Süreleri" Sayfası



Görsel (5.5), aynı şekilde tüm ürünlerin hazırlık sürelerinin kaydının yapıldığı sayfadır. Bu sayfada da veritabanında kayıtlı olan tüm ürünlerin olası tüm kombinasyonları ile veritabanında kayıtlı olan makinelerin birleşimi tek tabloda sağlanır. Kullanıcı yalnızca "Hazırlık Süresi" sütunu altındaki satırlara veri girişi yapabilir. Eğer sisteme yeni bir ürün girilmişse kullanıcı filtreleme yaparak bu ürünler için hızlı bir şekilde hazırlık sürelerini girebilir.

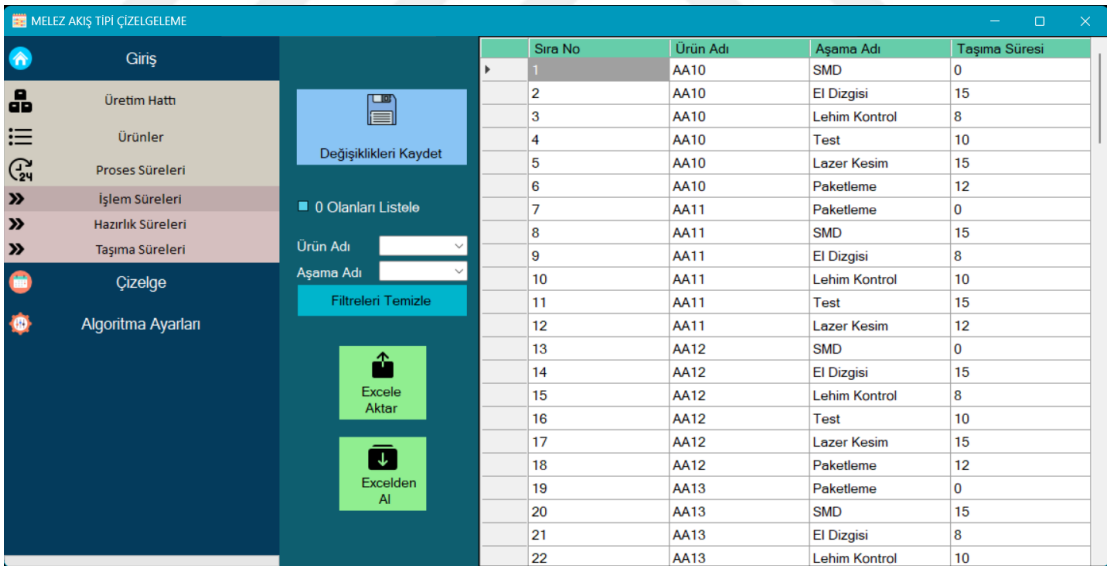
Görsel 5.5: KDS Giriş Sekmesi, “Hazırlık Süreleri” sayfası



Sıra No	Ürün Adı_1	Ürün Adı_2	Makine Adı	Hazırlık Süresi
1	AA10	AA10	SMD1	345
2	AA10	AA10	SMD2	345
3	AA10	AA10	El dizgisi tezhagı	365
4	AA10	AA10	Lehim Kontrol ...	0
5	AA10	AA10	Test Tezgahı	0
6	AA10	AA10	Lazer Kesim M...	0
7	AA10	AA10	Paketleme Tez...	90
8	AA10	AA11	Paketleme Tez...	346
9	AA10	AA11	SMD1	346
10	AA10	AA11	SMD2	364
11	AA10	AA11	El dizgisi tezhagı	0
12	AA10	AA11	Lehim Kontrol ...	0
13	AA10	AA11	Test Tezgahı	0
14	AA10	AA11	Lazer Kesim M...	90
15	AA10	AA12	SMD1	348
16	AA10	AA12	SMD2	348
17	AA10	AA12	El dizgisi tezhagı	363
18	AA10	AA12	Lehim Kontrol ...	0
19	AA10	AA12	Test Tezgahı	0
20	AA10	AA12	Lazer Kesim M...	0
21	AA10	AA12	Paketleme Tez...	90

Görsel (5.6)’da sunulduğu üzere taşıma süresi veritabanındaki tüm ürünler ile aşamaların tek tabloda birleştirildiği bir sayfadır. Bu sayfada taşıma süresi kaydı hazırlık ve işlem süresi ile aynı şekilde yapılmaktadır.

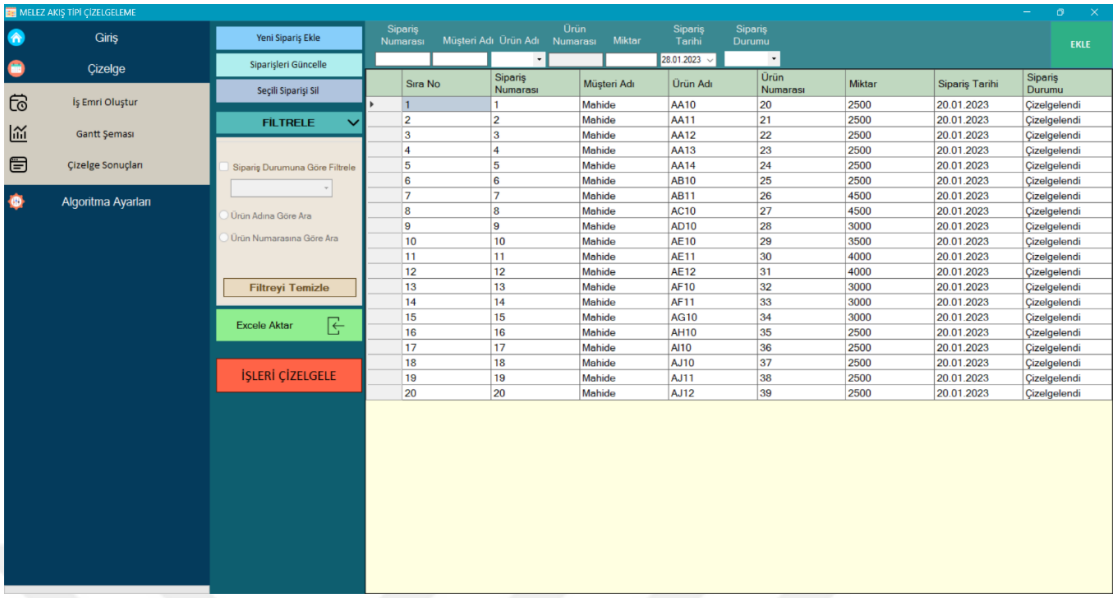
Görsel 5.6: KDS Giriş Sekmesi, “Taşıma Süreleri” Sayfası



Sıra No	Ürün Adı	Aşama Adı	Taşıma Süresi
1	AA10	SMD	0
2	AA10	El Dizgisi	15
3	AA10	Lehim Kontrol	8
4	AA10	Test	10
5	AA10	Lazer Kesim	15
6	AA10	Paketleme	12
7	AA11	Paketleme	0
8	AA11	SMD	15
9	AA11	El Dizgisi	8
10	AA11	Lehim Kontrol	10
11	AA11	Test	15
12	AA11	Lazer Kesim	12
13	AA12	SMD	0
14	AA12	El Dizgisi	15
15	AA12	Lehim Kontrol	8
16	AA12	Test	10
17	AA12	Lazer Kesim	15
18	AA12	Paketleme	12
19	AA13	Paketleme	0
20	AA13	SMD	15
21	AA13	El Dizgisi	8
22	AA13	Lehim Kontrol	10

Kullanıcı üretim hattı ile ilgili tüm düzenlemeleri yaptıktan sonra gelen siparişlerin kaydını ve iş çizelgelemesini yapmak için Görsel (5.7)’de sunulan “Çizelge” sekmesini kullanacaktır.

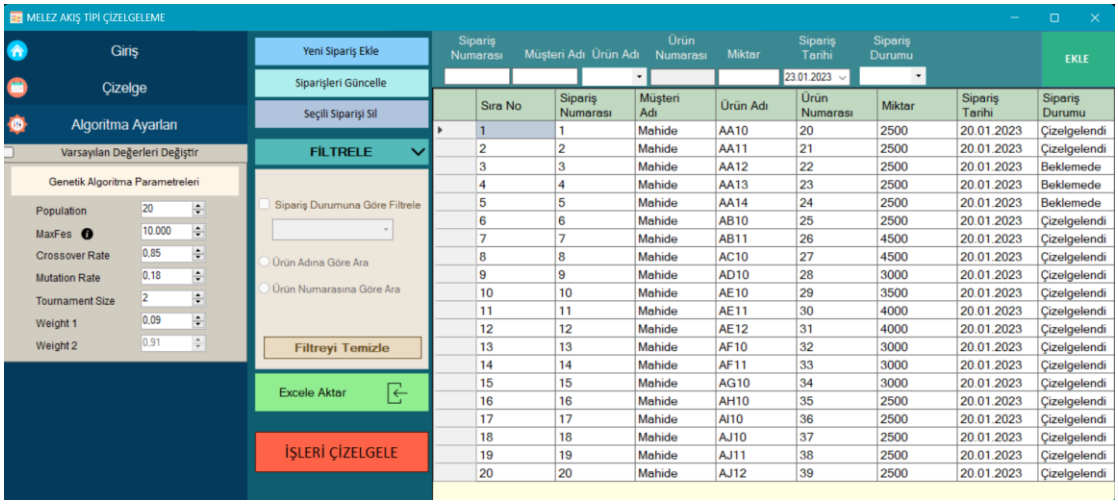
Görsel 5.7: KDS Çizelge Sekmesi



Algoritmanın çalıştırılacağı esas bölüm olan "İş Emri Oluştur" sayfasında kullanıcı siparişlerin kaydını tutar. Yeni bir sipariş alındığında bu siparişin kaydını yapabilir ve önceki siparişlerin durumunu güncelleyebilir.

Çizelgelemesi yapılmamış olan siparişler ilk kayıt aşamasında "Sipariş Durumu" bölümünde "Beklemede" olarak kaydedilir veya sonradan sipariş durumu değiştirilebilir. Sipariş kayıtları tamamlandıktan sonra "İşleri Çizelgele" butonuna tıklandığında durumu "Beklemede" olan siparişler için çizelge oluşturulur (Bkz., Görsel (5.8)). Sipariş listesine kaydedilen tüm işler veritabanında ayrı bir tabloda saklanır. Kullanıcı siparişler arasında arama yapacağı zaman "Filtrele" butonuna tıkladığında, veritabanındaki bu kayıtları, sipariş durumu, ürün adı ve ürün numarasına göre filtreleyebilir. Ayrıca kullanıcı "İş Emri Oluştur" sayfasına tıkladığında çizelgeyi çalıştıracağı için ATY'li GA parametrelerinin bulunduğu "Algoritma Ayarları" menüsü otomatik açılır. Buradan kullanıcı istediği şekilde değerleri değiştirebilir. Kullanıcı "Varsayılan Değerleri Değiştir" kutusundaki işareti kaldırdığında ayarlar eski haline gelir.

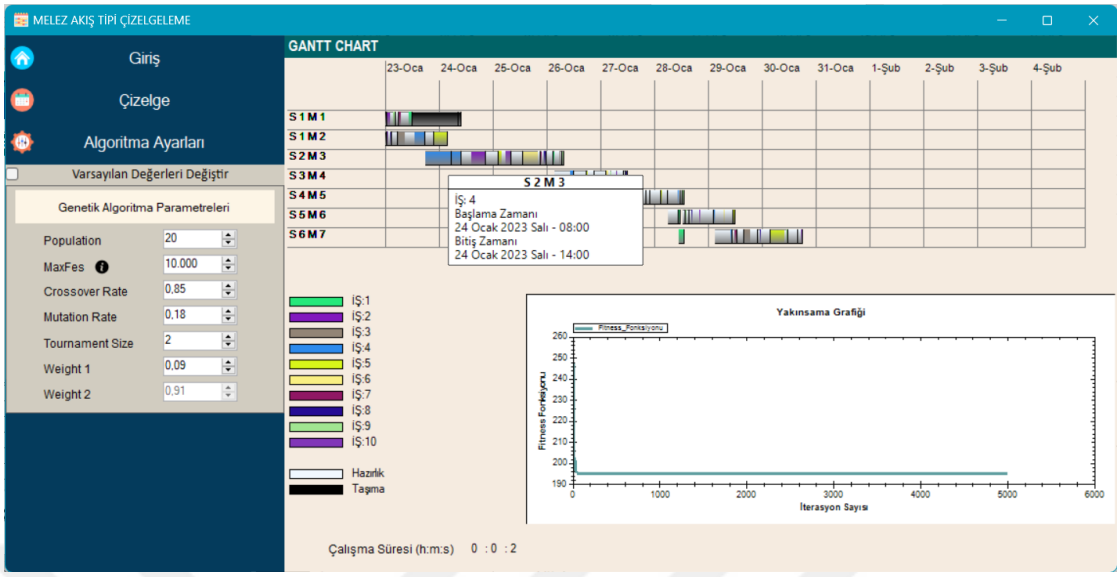
Görsel 5.8: KDS Çizelge Sekmesi, “İş Emri Oluştur” Sayfası



İşler çizelgelendikten sonra Gantt Şeması otomatik olarak açılır. Bu sayfada çizelgesi oluşturulan farklı işler farklı renklerde gösterilmektedir. Gantt şemasının her bir satırının başında hangi aşama ve hangi makine olduğu bilgisi verilir.

Şemanın en üstünde tarih aralıkları ve saatleri belirtilmektedir. Satır başları aşama ve makine bilgisini verir. (Görsel (5.9)’da “S” aşamayı, “M” makineyi ifade etmektedir.) Gantt şemasının üzerinde bir barın üstüne fare ile gelindiğinde ilgili iş ve o işin işlendiği zaman aralığı ile ilgili açılır bir bilgi kutucuğu çıkar. Şemanın hemen altında sol köşede şemadaki renklerin hangi işlere ait olduğu gösterilir. Ayrıca şemadaki beyaz renkli kısımlar hazırlık süresini, siyah renkli kısımlar taşıma süresini ifade etmektedir. Gantt grafiği oluşturulurken tüm işler taşıma süresi, hazırlık süresi ve işlem süresi sırasıyla çizelgelenir.

Görsel 5.9: KDS Çizelge Sekmesi, “Gantt Şeması”

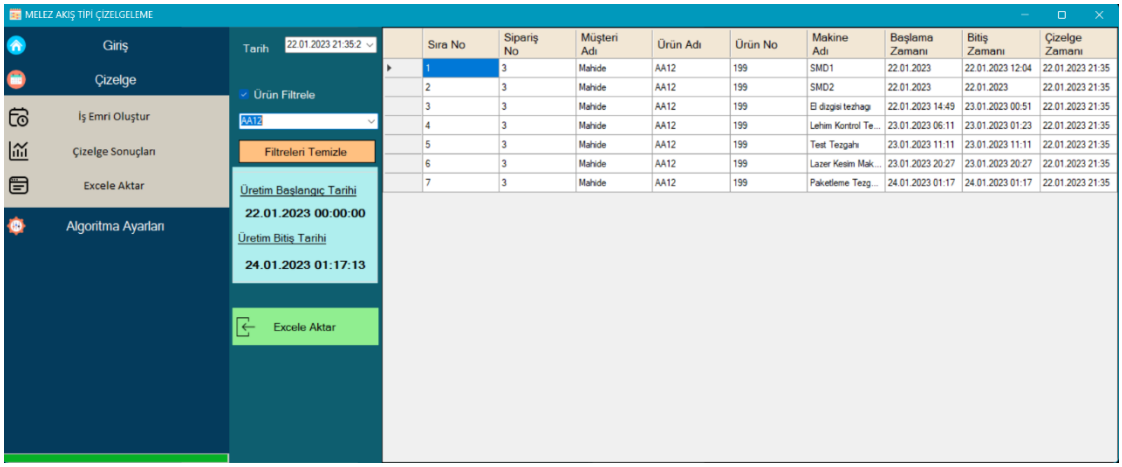


Şemanın altında bulunan yakınsama grafiği, algoritmanın çalışması süresince yaptığı iterasyonları ve bu iterasyonlardan elde edilen uygunluk değerlerinin değişimini göstermektedir. Sayfanın en altında da algoritmanın çalışma süresi (saat, dakika, saniye türünde) verilmektedir (Bkz., Görsel (5.9)).

Çizelgesi yapılan işler veritabanında ayrı bir tabloda tutulur, kullanıcı “Çizelge Sonuçları” sekmesine girdiğinde çizelgesi yapılan tüm işler listelenir. Kullanıcı isterse bu listede ürün adına göre filtreleme yapabilir, filtre yaptığında ilgili ürünün üretim hattında işleme alındığı zaman ile üretim hattından çıktığı zaman aralığı özet olarak gösterilir. Ve bu sayfada kullanıcı listelenen verileri Excel’e aktarabilir (Bkz., Görsel (5.10)).

Sonuç olarak, önerilen bu KDS ile firmaya yeni başlayan bir mühendisin (özellikle hiç tecrübesi olmayanların) tecrübeye bağlılık durumu en aza indirilmiş olacaktır ve Excel vb. araçlardaki insan faktörü kaynaklı hata oranı düşecektir.

Görsel 5.10: KDS “Çizelge” Sekmesi, “Çizelge Sonuçları”



5.1.1. KDS’nin Gerçek Verilerle Kullanımı ve Sonuçlar

Oluşturulan KDS firmadan alınan 20 adet ürünün işlem, hazırlık ve taşıma süreleri kullanılarak çalıştırılmıştır. Kullanılan süreler Tablo (5.1)-(5-3)’te sunulmuştur.

Bu atölyede üretilen ürünler, değişik ölçülerde ufak devre parçalarının devre kartları üzerine montajı yapılarak üretilmektedir. Bu parçaların miktarları çok fazla olduğu için sipariş miktarından ayrı tek bir birim olarak hazırlık ve işlem sürelerini düşünmek zor olacağından, elde edilen veriler sipariş miktarlarına göre dakika türünde uyarlanmıştır. Bu süreler, bir adet için harcanan sürenin sipariş miktarı ile çarpılmasıyla elde edilmiştir. Ayrıca ürünler SMD makinasına panel üzerinde çoklu adetlerde yüklenmektedir. 20 ürün için panel adetleri Tablo (5.2)’de sunulmuştur. SMD makinasındaki ürünlerin işlem süreleri, sipariş miktarları ve panel sayıları dikkate alınarak hesaplanmıştır. Panel büyüklüğüne göre SMD makinasındaki işlem süresinin, toplam sipariş miktarının panel sayısına bölümü ile elde edilen sayı ile çarpılmasıyla elde edilmiştir. Örneğin, panel miktarı 20, sipariş adeti 2000 olan bir işin bir adet panel için işlem süresi 2 dakika ise 2000 adet için gerekli olan süre (2000/20=100), 100*2=200 dakika olarak hesaplanmıştır. Bunun yanısıra paketleme için kullanılan koliler, farklı kapasitelere sahiptir ve tercih edilen siparişe göre farklı kapasitelerde koliler tercih edilmektedir. Bu çalışmada ele alınacak 20 örnek için, koli kapasiteleri sabit alınmıştır ve 500 ürün olarak düşünülmüştür. Paketleme işleminin hazırlık ve işlem süreleri, sipariş miktarına göre koli kapasitesi dikkate alınarak elde edilmiştir. Örneğin, koli kapasitesi 1000 için 3000 adetlik sipariş 3 koliye bölünerek paketlenmektedir, tek bir koli için gereken süreler 3 ile çarpılarak hesaplanmıştır. Taşıma işlemi için kullanılan paletlerin

ürün kapasitesi 1000 adetle sınırlı oluğu için aşamalar arası taşıma sayısı sipariş miktarına göre belirlenmiş (sipariş miktarı/1000) ve taşıma süresi ile çarpılarak elde edilmiştir.

Tablo 5.1: Gerçek Hayat Verileri, Hazırlık Süreleri (Dakika)

Ürün Kodu	Miktar	SMD	El dizgisi	Lazer Etiket	Paketleme
AA10	2500	345	365	0,333	90
AA11	2500	346	364	0,367	90
AA12	2500	348	363	0,25	90
AA13	2500	345	362	0,417	90
AA14	2500	345	365	0,417	90
AB10	2500	285	545	0,333	90
AB11	4500	540	93	0,933	162
AC10	4500	484	80,5	0,617	162
AD10	3000	514	158	0,583	108
AE10	3500	107,3	400	0,35	126
AE11	4000	107,3	400	0,2	144
AE12	4000	107,5	400	0,317	144
AF10	3000	167,3	0	0,267	108
AF11	3000	167,7	0	0,2	108
AG10	3000	107	400	0,4	108
AH10	2500	285	300	0,383	90
AI10	2500	238	28	0,3	90
AJ10	2500	273	28	0,467	90
AJ11	2500	283	28	0,267	90
AJ12	2500	243	28	0,3	90

Tablo 5.2: Gerçek Hayat Verileri, İşlem Süreleri (Dakika)

Ürün Kodu	Miktar	SMD	El dizgisi	Lehim Kontrol	Test	Lazer Etiket	Paketleme	SMD için panel miktarı
AA10	2500	395,83	570,00	32,64	10,42	10,42	20,83	4
AA11	2500	416,67	540,00	41,67	13,89	10,42	13,89	4
AA12	2500	375,00	510,00	31,94	15,28	17,36	13,89	10
AA13	2500	145,83	480,00	38,89	11,81	15,28	11,11	10
AA14	2500	145,83	480,00	37,50	9,03	17,36	17,36	10
AB10	2500	250,00	376,20	34,72	34,72	10,42	16,67	10
AB11	4500	487,50	959,85	127,50	12,50	31,25	47,50	10
AC10	4500	600,00	739,70	128,75	6,25	50,00	60,00	10
AD10	3000	1687,50	702,07	90,00	100,00	33,33	37,50	4
AE10	3500	88,67	214,60	54,44	9,72	14,58	20,42	25
AE11	4000	80,00	141,64	45,56	11,11	16,67	30,00	25
AE12	4000	80,00	322,50	44,44	11,11	16,67	33,33	25
AF10	3000	100,00	220,88	50,00	12,50	20,83	23,33	16
AF11	3000	84,38	221,32	35,83	12,50	20,83	25,00	16
AG10	3000	300,00	321,00	45,00	8,33	8,33	16,67	5
AH10	2500	231,48	376,20	34,72	14,58	10,42	14,58	9
AI10	2500	291,67	314,16	41,67	14,58	10,42	11,11	10
AJ10	2500	169,27	360,36	29,86	14,58	10,42	13,89	16
AJ11	2500	143,23	373,56	29,86	14,58	17,36	14,58	16
AJ12	2500	33,33	729,00	39,58	14,58	6,94	14,58	25

Tablo 5.3: Gerçek Hayat Verileri, Taşıma Süreleri (Dakika)

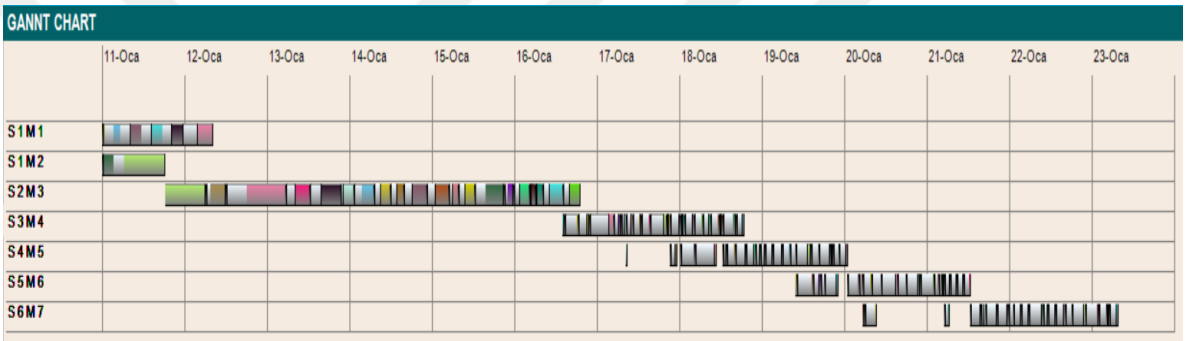
Ürün Kodu	Miktar	SMD	El dizgisi	Lehim Kontrol	Test	Lazer Etiket	Paketleme
AA10	2500	0	15	7,5	10	15	12,5
AA11	2500	0	15	7,5	10	15	12,5
AA12	2500	0	15	7,5	10	15	12,5
AA13	2500	0	15	7,5	10	15	12,5
AA14	2500	0	15	7,5	10	15	12,5
AB10	2500	0	15	7,5	10	15	12,5
AB11	4500	0	27	13,5	18	27	22,5
AC10	4500	0	27	13,5	18	27	22,5
AD10	3000	0	18	9	12	18	15
AE10	3500	0	21	10,5	14	21	17,5
AE11	4000	0	24	12	16	24	20
AE12	4000	0	24	12	16	24	20
AF10	3000	0	18	9	12	18	15
AF11	3000	0	18	9	12	18	15
AG10	3000	0	18	9	12	18	15

Tablo 5.3 (Devam): Gerçek Hayat Verileri, Taşıma Süreleri (Dakika)

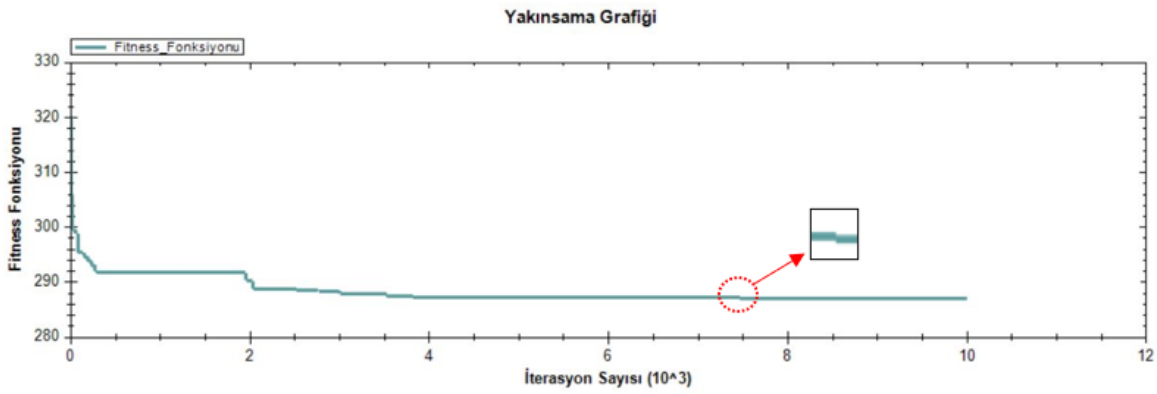
Ürün Kodu	Miktar	SMD	El dizgisi	Lehim Kontrol	Test	Lazer Etiket	Paketleme
AH10	2500	0	15	7,5	10	15	12,5
AI10	2500	0	15	7,5	10	15	12,5
AJ10	2500	0	15	7,5	10	15	12,5
AJ11	2500	0	15	7,5	10	15	12,5
AJ12	2500	0	15	7,5	10	15	12,5

Yukarıdaki süreler KDS'de çalıştırıldığında elde edilen Gantt şeması Görsel (5.11)'de, yakınsama grafiği ise Görsel (5.12)'de sunulmuştur.

Görsel 5.11: KDS ile Elde Edilen Gerçek Hayat Örneği Gantt Şeması



Görsel 5.12: KDS ile Elde Edilen Gerçek Hayat Örneği Yakınsama Grafiği



Algoritma 20 adet örnek için 10 saniye çalışmış ve yukarıdaki sonuçlar elde edilmiştir. Yakınsama grafiğine göre 4000 iterasyona kadar uygunluk değerinde bir iyileşme olmuştur. Görsel (5.12)'de görüldüğü üzere, 7500. iterasyonda çok düşük bir iyileşme daha olmuştur.

SONUÇ VE ÖNERİLER

Bu çalışmada İstanbul'da faaliyet gösteren acil aydınlatma ürünleri üreten bir firmanın üretim sistemi incelenmiş ve bu işletmede etkili bir çizelgeleme yapılması amaçlanmıştır. İncelenen üretim ortamı MAA sınıfına girmekte ve çalışanların el becerisine dayalı imalat yapılmakta olduğu için öğrenme etkisi tabanlı, sıra bağımlı hazırlık süresi ve taşıma süresi içeren MATÇ problemi için matematiksel model belirlenmiştir. Oluşturulan model Lingo 11.0 optimizasyon yazılımı ile farklı örnek boyutları için test edilmiş ve doğrulanmıştır. Örnek boyutu arttığında optimizasyon programının problemin çözümü için makul sürelerde çözüm üretememesi sebebiyle ele alınan problemin çözümü için bir skalerleştirme tabanlı GA tasarlanmıştır.

Tasarlanan ÇAGA'nın parametre ayarı için **irace** (Iterated F-Race algorithm) otomatik parametre yapılandırma algoritması tercih edilmiştir. ÇAGA'nın performans ölçümü için 3 adet deney seti grubu (küçük, orta, büyük) oluşturulmuş ve **irace** aracılığıyla her örnek grubu için uygun parametreler belirlenmiştir. ÇAGA'da amaçlar ATY, KSY ve TSY kullanılarak 3 farklı şekilde hesaplanmıştır (Bkz., Eşitlik (2.18)-(2.20)). Algoritmada her bir yöntem için, her örnek seti, 31 kez çalıştırılmıştır. Yapılan analizlere göre D_{IP} performans metriğine göre ATY'li GA diğer skalerleştirme tabanlı GA'lardan daha iyi performans gösterirken, D_{Ref} performans metriğine göre TSY'li GA KSY'li GA'dan daha iyi performans göstermiştir.

Son olarak geliştirilen algoritmanın işletmede kullanılabilmesi için bir KDS tasarlanmış ve gerçek hayat örnekleri kullanılarak elde edilen sonuçlar yorumlanmıştır.

Bu çalışmadan elde edilen veriler ve kullanılan varsayımlar altında MATÇ problemi ile ilgili gelecek çalışmalarda;

Ele alınan üretim ortamında el dizgisi ve test hatları kendi içinde birden fazla operatör çalıştırmaktadır, bu hatların durumu dikkate alınarak en uygun çizelgeyi oluşturacak optimum operatör sayısı belirlenebilir, SMD makinası için panel miktarları göz önüne alınarak bir çizelge oluşturulabilir. İşgücünün önemli olduğu bu tarz üretim ortamlarında çizelgeleme yaparken çalışanların ergonomik durumlarını dikkate alarak bir optimizasyon çalışması yapılabilir. Ayrıca bu çalışmada incelenen üretim ortamında her operatör bütün işleri yapabilme becerisine sahip ancak çalışma hızları farklılık göstermektedir, işlerin çizelgelemesi yapılırken aynı zamanda operatör çizelgelemesi de dikkate alınabilir veya operatörlerin yetenek seviyeleri dikkate alınarak bir çizelgeleme

çalışması yapılabilir. Önerilen skalerleştirme tabanlı GA'nın literatürdeki diğer algoritmalarla karşılaştırması yapılarak performansları karşılaştırılabilir. Çalışmada kullanılan parametre yapılandırma aracı **irace** ile literatürdeki diğer parametre ayarlama yöntemleri kıyaslanabilir.

KAYNAKÇA

Alaykýran, K., Engin, O., & Döyen, A. (2007). Using ant colony optimization to solve hybrid flow shop scheduling problems. *The International Journal of Advanced Manufacturing Technology, 35*, 541–550.

Alizadeh, R., vd. (2020). A modified genetic algorithm for non-emergency outpatient appointment scheduling with highly demanded medical services considering patient priorities. *Computers & Industrial Engineering, 5(*139*)*, 106-116.

Aqil, S., & Allali, K. (2020). Local search metaheuristic for solving hybrid flow shop problem in slabs and beams manufacturing. *Expert Systems with Applications, 162*, 716-733.

Behnamian, J., & Zandieh, M. (2012). Earliness and Tardiness Minimizing on a Realistic Hybrid Flowshop Scheduling with Learning Effect by Advanced Metaheuristic. *Arabian Journal for Science and Engineering, 38*, 1229–1242.

Biskup, D., (1999). Single-Machine Scheduling with Learning Considerations. *European Journal of Operational Research, 115*, 173-178.

Bożejko, W., Pempera, J., & Smutnicki, C. (2013). Parallel tabu search algorithm for the hybrid flow shop problem. *Computers & Industrial Engineering, 65*(3),466-474.

Cai, J., Zhou, R., & Lei, D. (2020). Dynamic shuffled frog-leaping algorithm for distributed hybrid flow shop scheduling with multiprocessor tasks. *Engineering Applications of Artificial Intelligence, 90*, 540-553.

Chaari, T., vd. (2011). A genetic algorithm for robust hybrid flow shop scheduling. *International Journal of Computer Integrated Manufacturing, 24*(9), 821-833.

Chou, F. D. (2013). Particle swarm optimization with cocktail decoding method for hybrid flow shop scheduling problems with multiprocessor tasks. *International Journal of Production Economics, 141*(1), 137-145.

Chung, T. P., & Liao, C. -J. (2013). An immunoglobulin-based artificial immune system for solving the hybrid flow shop problem. *Applied Soft Computing, 13*(8), 3729-3736.

Costa, A., AntonioCappadonna, F., & Fichera, S. (2013). A dual encoding-based meta-heuristic algorithm for solving a constrained hybrid flow shop scheduling problem. *Computers & Industrial Engineering, 64*(4), 937-958.

Costa, A., Fernandez-Viagas, V., & Framinan, J. M. (2020). Solving the hybrid flow shop scheduling problem with limited human resource constraint. *Computers & Industrial Engineering, 146,* 545-567.

Cui, Z., & Gu, X. (2015). An improved discrete artificial bee colony algorithm to minimize the makespan on hybrid flow shop problems. *Neurocomputing, 148,* 248-259.

Dabiri, M., Yazdani, M., Naderi, B., & Haleh, H. (2021). Modeling and solution methods for hybrid flow shop scheduling problem with job rejection. *Operational Research, 22,* 2721–2765.

Deliktaş, D. & Ustun, O. (basımda). Multi-objective genetic algorithm based on the fuzzy MULTIMOORA method for solving the cardinality constrained portfolio optimization. *Applied Intelligence.*

Dios, M., Fernandez-Viagas, V., & M. Framinan, J. (2018). Efficient heuristics for the hybrid flow shop scheduling problem with missing operations. *Computers & Industrial Engineering, 115,* 88-99.

Ebrahimi, M., FatemiGhomi, S. M. T., & Karimi, B. (2014). Hybrid flow shop scheduling with sequence dependent family setup time and uncertain due dates. *Applied Mathematical Modelling, 38*(9-10), 2490-2504.

Ehrgott M., Ide J., & Schöbel A. (2014). Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research, 239*(1), 17-31.

Elmi, A., & Topaloglu, S. (2013). A scheduling problem in blocking hybrid flow shop robotic cells with multiple robots. *Computers & Operations Research, 40*(10), 2543-2555.

Elmi, A., & Topaloglu, S. (2014). Scheduling multiple parts in hybrid flow shop robotic cells served by a single robot. *International Journal of Computer Integrated Manufacturing, 27*(12), 1144-1159.

Engin, O., Ceran, G., & K. Yilmaz, M. (2011). An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems. *Applied Soft Computing, 11*(3), 3056-3065.

Eskandari, H., & Hosseinzadeh, A. (2014). A variable neighbourhood search for hybrid flow-shop scheduling problem with rework and set-up times. *The Journal of the Operational Research Society, 65*, 1221–1231.

Feng, X., Zheng, F., & Xu, Y. (2016). Robust scheduling of a two-stage hybrid flow shop with uncertain interval processing times. *International Journal of Production Research, 54*(12), 3706-3717.

Fernandez-Viagas, V., Perez-Gonzalez, P., & M. Framinan, J. (2019). Efficiency of the solution representations for the hybrid flow shop scheduling problem with makespan objective. *Computers & Operations Research, 109*, 77-88.

Fernandez-Viagas, V., Costa, A., & M. Framinan, J. (2020). Hybrid flow shop with multiple servers: A computational evaluation and efficient divide-and-conquer heuristics. *Expert Systems with Applications, 153*(1), 462-477.

Fernandez-Viagas, V. (2022). A speed-up procedure for the hybrid flow shop scheduling problem. *Expert Systems with Applications, 187*, 903-914.

Gasimov, R. N. (2001). Characterization of the Benson Proper Efficiency and Scalarization in Nonconvex Vector Optimization. In: Köksalan, M., Zionts, S. (Eds.*), Multiple Criteria Decision Making in the New Millennium* (pp.189-198). Berlin, Germany: Springer.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms, I*, 69-93.

Gupta N. D., J. (1988). Two-Stage Hybrid Flowshop Scheduling Problem. *The Journal of the Operational Research Society, 39*(4), 359–64.

H. Sprague, R. (1980). A Framework for the Development of Decision Support Systems. *Management Information Systems Quarterly, 4*(4), 1-26.

Hadda, H., Dridi, N., & Hajri-Gabouj, S. (2014). Exact resolution of the two-stage hybrid flow shop with dedicated machines. *Optimization Letters, 8*, 2329–2339.

Han, W., vd. (2021). Multi-objective evolutionary algorithms with heuristic decoding for hybrid flow shop scheduling problem with worker constraint. *Expert Systems with Applications, 168,* 282-299.

Harbaoui, H., & Khalfallah, S. (2020). Tabu-search optimization approach for no-wait hybrid flow-shop scheduling with dedicated machines. *Procedia Computer Science, 176,* 706-712.

Hasan, S. M. K. vd. (2011). A DSS for job scheduling under process interruptions. *Flexible Services and Manufacturing Journal, 23,* 137–155.

Heizer, J., and Render, B., (2001). *Operations Management* (6th ed.). New Jersey, ABD: Prentice Hall.

Holland, J. H., (1975). *Adaption in Natural and Artificial Systems* (2nd ed.). Ann Arbor: MIT Press.

Jouglet, A., Oğuz, C., & Sevaux, M. (2009). Hybrid Flow-Shop: A Memetic Algorithm Using Constraint-Based Scheduling for Efficient Search. *Journal of Mathematical Modelling and Algorithms in Operations Research, 8,* 271-292.

Kim J. & Kim S. K. (2006). A chim-based interactive tchebycheff procedure for multiple objective decision making. *Computers & Operations Research, 33*(6), 1557–1574.

Kim, J. W. (2015). Developing a job shop scheduling system through integration of graphic user interface and genetic algorithm. *Multimedia Tools and Applications, 74,* 3329–3343.

Kramer, O. (2017). *Genetic Algorithm Essentials.* Studies in Computational Intelligence, (1st ed.). Berlin, Germany: Springer.

Korkmaztan, R., & Bora, Ş. (2017). Parameter Tuning Algorithms in Modeling and Simulation. *International Journal of Engineering Science and Application, 1*(2), 58-66.

Laxmi, L, J., Mohan, N., & Madhusudanan Pillai, V. (2017). Lot streaming in [N-1](1)+N(m) hybrid flow shop. *Journal of Manufacturing Systems, 44*(1), 12-21.

Lei, D., & Guo, X. (2015). A shuffled frog-leaping algorithm for hybrid flow shop scheduling with two agents. *Expert Systems with Applications, 42*(23), 9333-9339.

Lei, D., & Zheng, Y. (2017). Hybrid flow shop scheduling with assembly operations and key objectives: A novel neighborhood search. *Applied Soft Computing, 61*, 122-128.

Li, J., Pan, Q., Wang, F. (2014). A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem. *Applied Soft Computing, 24*, 63-77.

Li, J., & Pan, Q. (2015). Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. *Information Sciences, 316*, 487-502.

Li, Z., vd. (2021). Bi-objective hybrid flow shop scheduling with common due date. *Operational Research, 21*, 1153–1178.

Lian, X., vd. (2021). An energy-efficient hybrid flow shop scheduling problem in steelmaking plants. *Computers & Industrial Engineering, 162*, 683-698.

Liao, C. J., Tjandradjaja, E., & Chung, T. P. (2012). An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem. *Applied Soft Computing, 12*, 1755-1764.

Lim S. M., vd. (2017). Crossover and Mutation Operators of Genetic Algorithms. *International Journal of Machine Learning and Computing, 7*(1), 9-12.

Liu, M., vd. (2017). Scheduling a tempered glass manufacturing system: a three-stage hybrid flow shop model. *International Journal of Production Research, 55*(20), 6084-6107.

López-Ibáñez, M., vd. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives, 3*, 43-58.

Luo, X., vd. (2010). Optimal resource allocation for hybrid flow shop in one-of-a-kind production. *International Journal of Computer Integrated Manufacturing, 23*(2), 146-154.

Marichelvam, M. K., Prabaharan, T., & Yang, X. S. (2014). Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan. *Applied Soft Computing, 19,* 93-101.

Marichelvam, M. K., Geetha, M., & Tosun, Ö. (2020). An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors – A case study. *Computers & Operations Research, 114,* 812-821.

Meng, L., vd. (2019). More MILP models for hybrid flow shop scheduling problem and its extended problems. *International Journal of Production Research, 58*(13), 3905-3930.

Miettinen, K. (2014). Survey of methods to visualize alternatives in multiple criteria decision-making problems. *OR Spectrum, 36,* 3–37.

Naderi, B., Zandieh, M., & Shirazi, M. A. H. A. (2009). Modeling and scheduling a case of flexible flow shops: Total weighted tardiness minimization. *Computers & Industrial Engineering, 57*(4), 1258-1267.

Niu, Q., vd. (2012). An efficient quantum immune algorithm to minimize mean flow time for hybrid flow shop problems. *Mathematics and Computers in Simulation, 84,* 1-25.

Ozsoydan, F. B., & Sağir, M. (2021). Iterated greedy algorithms enhanced by hyper-heuristic-based learning for hybrid flexible flowshop scheduling problem with sequence dependent setup times: A case study at a manufacturing plant. *Computers & Operations Research, 125,* 44-59.

Pargar, F., & Zandieh, M. (2012). Bi-criteria SDST hybrid flow shop scheduling with learning effect of setup times: water flow-like algorithm approach. *International Journal of Production Research, 50,* 2609-2623.

Pargar, F., vd. (2018). The Effect of Worker Learning on Scheduling Jobs in a Hybrid Flow Shop: A Bi-Objective Approach. *Journal of Systems Science and Systems Engineering, 27,* 265–291.

R. T. Marler & J. S. Arora. (2005). Function-transformation methods for multi-objective optimization. *Engineering Optimization, 37*(6), 551-570.

Rathinasamy, S., & Raju, R. (2009). Sequencing and scheduling of nonuniform flow pattern in parallel hybrid flow shop. *The International Journal of Advanced Manufacturing Technology, 49*, 213–225.

RezaAmin-Naseri, M., & AliBeheshti-Nia, M. (2009). Hybrid flow shop scheduling with parallel batching. *International Journal of Production Economics, 117*(1), 185-196.

Rolf, B., vd. (2020). Assigning dispatching rules using a genetic algorithm to solve a hybrid flow shop scheduling problem. *Procedia Manufacturing, 42*, 442-449.

Sarin, S. (2013). Two-Stage, Multiple-Lot, Lot Streaming Problem for a 1 + 2 Hybrid Flow Shop. *IFAC Proceedings Volumes, 46*(9), 448-453.

Shao, W., Shao, Z., & Pi, D. (2020). Modeling and multi-neighborhood iterated greedy algorithm for distributed hybrid flow shop scheduling problem. *Knowledge-Based Systems, 194*, 527-544.

Su, Z., & Yi, J. (2015). Genetic Algorithm with Direction Selection for the Hybrid Flow Shop Scheduling Problem with makespan minimization [Sözlü bildiri]. *Proceedings of the 3rd International Conference on Mechatronics and Industrial Informatics, Atlantis Press, China*.

Sule D., (2007). *Production Planning and Industrial Scheduling* (2nd ed.). Florida, ABD: CRC Press.

Sun, L., & Yu, S. (2015). Scheduling a real-world hybrid flow shop with variable processing times using Lagrangian relaxation. *The International Journal of Advanced Manufacturing Technology, 78*, 1961–1970.

Syam, W., & Al-Harkan, I. (2010). Comparison of three meta heuristics to optimize hybrid flow shop scheduling problem with parallel machines. *World Academy of Science, Engineering and Technology, 38*, 919-926.

T. Lin, J., & Chen, C. M. (2015). Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. *Simulation Modelling Practice and Theory, 51*, 100-114.

Tseng, C. T., & Liao, C. J. (2008). A particle swarm optimization algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *International Journal of Production Research, 46*(17), 4655-4670.

Wang, H. M., Chou, F. D., & Wu, F. C. (2010). A simulated annealing for hybrid flow shop scheduling with multiprocessor tasks to minimize makespan. *The International Journal of Advanced Manufacturing Technology, 53*, 761–776.

Wang, J., & Wang, L. (2020). A Bi-Population Cooperative Memetic Algorithm for Distributed Hybrid Flow-Shop Scheduling. *IEEE Transactions on Emerging Topics in Computational Intelligence, 5*(6), 947-961.

Wang, K., Choi, S. H., & Qin, H. (2014). An estimation of distribution algorithm for hybrid flow shop scheduling under stochastic processing times. *International Journal of Production Research, 52*(24), 7360-7376.

Wang, S., & Liu, M. (2013). A genetic algorithm for two-stage no-wait hybrid flow shop scheduling problem. *Computers & Operations Research, 40*(4), 1064-1075.

Wang, S., vd. (2014). An order-based estimation of distribution algorithm for stochastic hybrid flow-shop scheduling problem. *International Journal of Computer Integrated Manufacturing, 28*(3), 307-320.

Wang, S., Liu, M., & Chu, C. (2014). A branch-and-bound algorithm for two-stage no-wait hybrid flow-shop scheduling. *International Journal of Production Research, 53*(4), 1143-1167.

Wang, S., vd. (2019). An energy-efficient two-stage hybrid flow shop scheduling problem in a glass production. *International Journal of Production Research, 58*(8), 2283-2314.

Wright, T. P., (1936). Factors Affecting the Cost of Airplanes. *Journal of The Aeronautical Sciences, 3*, 122-128.

Xu, Y., vd. (2013). An effective shuffled frog-leaping algorithm for hybrid flow-shop scheduling with multiprocessor tasks. *The International Journal of Advanced Manufacturing Technology, 68*, 1529–1537.

Xu, Y., & Wang, L. (2011). Differential evolution algorithm for hybrid flow-shop scheduling problems. *Journal of Systems Engineering and Electronics, 22*(5), 794-798.

Yalaoui, N., vd. (2013). Fuzzy-metaheuristic methods to solve a hybrid flow shop scheduling problem with pre-assignment. *International Journal of Production Research, 51*(12), 3609-3624.

Yang, J. (2013). A two-stage hybrid flow shop with dedicated machines at the first stage. *Computers & Operations Research, 40*(12), 2836-2843.

Yang J., & Soh C. K., (1997). Structural Optimization by Genetic Algorithms with Tournament Selection. *Journal of Computing in Civil Engineering, 11*(3), 195-200.

YiZhang, X., & Chen, L. (2017). A re-entrant hybrid flow shop scheduling problem with machine eligibility constraints. *International Journal of Production Research, 56*(16), 5293-5305.

Yu, C., Semeraro, Q., & Matta, A. (2018). A genetic algorithm for the hybrid flow shop scheduling with unrelated machines and machine eligibility. *Computers & Operations Research, 100*, 211-229.

Zandieh, M., FatemiGhomi, S. M. T., & MoattarHusseini, S. M. (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation, 180*(1), 111-127.

Zhang, C., vd. (2021). A discrete whale swarm algorithm for hybrid flow-shop scheduling problem with limited buffers. *Robotics and Computer-Integrated Manufacturing, 68*, 81-93.

Zhang, B., vd. (2021). An automatic multi-objective evolutionary algorithm for the hybrid flowshop scheduling problem with consistent sublots. *Knowledge-Based Systems, 238*, 819-840.

DİZİN

-A-

Alel, 29

-Ç-

Çaprazlama, viii, x, xi, 30, 36, 37, 38, 43, 54
Çizelgeleme, iii, v, xiii, xiv, xv, xvi, 1, 13, 14, 15, 62

-D-

Doğrusal Programlama, v, viii, xvi, 4
Durdurma Koşulu, viii, 40

-E-

Ebeveyn, 36, 37
Elitizm, viii, x, 30, 39, 40

-G-

GA, v, vi, ix, x, xi, xiv, 6, 7, 8, 9, 12, 13, 15, 16, 28, 29, 30, 31, 32, 34, 35, 36, 37, 38, 39, 40, 43, 45, 48, 49, 50, 51, 52, 56, 57, 58, 59, 60, 63, 68, 75, 76
Gantt, xi, xii, 27, 28, 69, 70, 74
Gecikme Süresi, 17
Gen, 29, 32, 63
Gerçek Hayat, viii, x, xii, 4, 9, 72, 73, 74

-H-

Hazırlık zamanı, 53

-İ-

İdeal Nokta, xi, 59
İrace, v, 45
İşlem Zamanı, xiv
İterasyon, 29

-K-

Karar Destek Sistemi, v, xv, 2, 62
Kıyaslama (Benchmark), x, 48, 49, 50, 51, 52
Konik Skalerleştirme, xvi, 25
Kromozom, viii, xi, 29, 31, 32, 33

-M-

MAA, xi, xvi, 1, 4, 5, 6, 7, 8, 9, 12, 13, 14, 15, 16, 17, 19, 75
MATÇ, v, viii, ix, x, xvi, 1, 2, 4, 5, 6, 7, 8, 9, 10, 20, 21, 22, 28, 32, 37, 42, 47, 48, 49, 50, 51, 52, 53, 56, 57, 63, 75
Mutasyon, viii, x, xi, 30, 38, 39, 40, 43, 54

-N-

Nesil, 29

-Ö-

Öğrenme Etkisi, iii, v, viii, x, 4, 10, 12, 13, 15, 21

-P-

Parametre, ix, x, 43, 44, 54
Parametre Ayarlama, ix, 44
Popülasyon, 29, 30, 34, 35, 43, 54

-R-

Referans Nokta, x, xi, 46, 60

-S-

Seçim, viii, x, xvi, xvii, 30, 34, 35, 36
Skalerleştirme, v, x, xi, 25, 46, 56, 57, 59, 60

-T-

Tchebycheff Skalerleştirme, xvii, 25

Turnuva Büyüklüğü, 54

-U-

Uygunluk Değeri, viii, x, 31, 32

-Y-

Yakınsama grafiği, 74
Yeni birey, 32