



T.C.

**KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**MELEZ ATEŞBÖCEĞİ ALGORİTMASI İLE
ÇOK AMAÇLI ESNEK AKIŞ TİPİ
ÇİZELGELEME PROBLEMLERİNİN
ÇÖZÜMÜ**

Fatmagül TOPÇAM

YÜKSEK LİSANS TEZİ

Eylül-2019

KONYA

Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Fatmagül TOPÇAM tarafından hazırlanan “MELEZ ATEŞBÖCEĞİ ALGORİTMASI İLE ÇOK AMAÇLI ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN ÇÖZÜMÜ” adlı tez çalışması 11/09/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Danışman

Prof. Dr. Orhan ENGİN

Üye

Doç. Dr. Ahmet SARUCAN

Üye

Dr. Öğr. Üyesi Kadir BÜYÜKÖZKAN

İmza





Yukarıdaki sonucu onaylarım.

Prof. Dr. Hakan KARABÖRK
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Fatmagül TOPÇAM



ÖZET

YÜKSEK LİSANS TEZİ

MELEZ ATEŞBÖCEĞİ ALGORİTMASI İLE ÇOK AMAÇLI ESNEK AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN ÇÖZÜMÜ

Fatmagül TOPÇAM

**Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Endüstri Mühendisliği Anabilim Dalı**

Danışman: Prof. Dr. Orhan ENGİN

2019, 57 Sayfa

Jüri

Prof. Dr. Orhan ENGİN

Doç. Dr. Ahmet SARUCAN

Dr. Öğr. Üyesi Kadir BÜYÜKÖZKAN

Sürü davranışları, birbirlerinden etkilenen grupların hareketleri ve sergiledikleri eylemlerdir. Grupların hareketlerinden ve davranışlarından esinlenerek algoritmalar üretilmiştir. Üretilen algoritmalarından bir tanesi de ateş böceği algoritmasıdır. Ateş böceği algoritması, ateş böceklerinin ışık yayma ve ışık kaynağına doğru hareketlerinden esinlenerek çözüm üretmeyi amaçlayan bir yöntemdir. Diğer sezgisel algoritmalara kıyasla gerçek problemlere uygulanması kolaydır. Bu özelliğinden dolayı, ateş böceği algoritması, giderek daha önemli bir çözüm yöntemi haline gelmiştir. Araştırmada, ateş böceği algoritmasının çözüm adımları, çok amaçlı esnek akış tipi çizelgeleme problemlerinin çözümü için, genetik algoritmanın operatörleri yardımıyla melez hale getirilmiştir. Melez ateş böceği algoritması, literatürde yer alan esnek akış tipi çok amaçlı çizelgeleme probleminin çözümünde kullanılmıştır. Önerilen melez ateş böceği algoritmasının, çok amaçlı esnek akış tipi çizelgeleme problemlerinin çözümünde kullanılabilecek başarılı bir yöntem olduğu belirlenmiştir.

Anahtar Kelimeler: Çizelgeleme problemleri, çok amaçlı esnek akış tipi, melez ateş böceği algoritması.

ABSTRACT

MS THESIS

**SOLUTION OF MULTI-OBJECTIVE FLEXIBLE FLOW SHOP SCHEDULING
PROBLEMS WITH HYBRID FIREFLY ALGORITHM**

Fatmagül TOPÇAM

**Konya Technical University
Institute of Graduate Studies
Department of Industry Engineering**

Advisor: Prof. Dr. Orhan ENGİN

2019, 57 Pages

Jury

Prof. Dr. Orhan ENGİN

Assoc. Prof. Dr. Ahmet SARUCAN

Asst.Prof.Dr.Kadir BÜYÜKÖZKAN

Herd behaviors are the movements of the affected groups each other and their actions. Algorithms have been produced by inspiration from the movements and behaviors of the groups. One of the algorithms produced is the firefly algorithm. Firefly algorithm is a method that aims to produce solutions by firing insects based on light emission and movements towards the light source. Compared to other heuristic algorithms, it is easy to apply to real problems. Because of this feature, firefly algorithm has become an increasingly important solution method. In the study, the solution steps of the firefly algorithm are hybridized with the help of the operators of the genetic algorithm to solve the multi-objective flexible flow shop scheduling problems. The hybrid firefly algorithm is used to solve the flexible flow shop multipath scheduling problem in the literature. The proposed hybrid firefly algorithm has been determined to be a successful method for solving multi-objective flexible flow shop scheduling problems.

Keywords: Hybrid firefly algorithm, multi-objective flexible flow shop, scheduling problems.

ÖNSÖZ

Tez çalışmam sırasında kıymetli bilgi, birikim ve tecrübeleri ile bana yol gösteren ve destek olan değerli danışman hocam sayın Prof. Dr. Orhan Engin'e, sonsuz teşekkür ve saygılarımı sunarım.

Lisans ve yüksek lisans eğitimim boyunca yardım, bilgi ve tecrübeleri ile bana sürekli destek olan Endüstri Mühendisliği bölümündeki tüm hocalarıma teşekkür ederim.

Çalışmalarım boyunca yardımını hiç esirgemeyen değerli arkadaşım Fakher Tafakhori'ye teşekkürü bir borç bilirim.

Çalışmalarım boyunca maddi manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme, eşime ve biricik oğlum Kaan'a sonsuz teşekkür ederim.

Fatmagül TOPÇAM

KONYA-2019

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
KISALTMALAR	viii
ŞEKİL LİSTESİ.....	ix
ÇİZELGE LİSTESİ.....	ix
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	2
3. YÖNTEM VE MATERYAL.....	12
3.1. Yöntem.....	12
3.1.1. Ateş Böceği Algoritması.....	12
3.1.2. Melez Ateş Böceği Algoritması.....	15
3.2. Materyal.....	19
3.2.1. Çok Amaçlı Esnek Akış Tipi Çizelgeleme Problemlerinin Çözümü	19
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	21
4.1. Melez Ateş Böceği Algoritması ile Çok Amaçlı Esnek Akış Tipi Çizelgeleme Probleminin Çözümü	21
5. SONUÇLAR VE ÖNERİLER	32
KAYNAKLAR	33
EKLER	39
ÖZGEÇMİŞ	48

KISALTMALAR

ABA	: Ateş Böceği Algoritması
ASD	: Alt Sınır Değeri
C_{max}	: Tamamlanma zamanı
ÇAEATÇ	: Çok Amaçlı Esnek Akış Tipi Çizelgeleme
DSA	: Dal Sınır Algoritması
EATÇ	: Esnek Akış Tipi Çizelgeleme
GA	: Genetik Algoritma
İGİÇ	: İlk Giren İlk Çıkar
KDKAA	: Karma Değişken Komşuluk Arama Algoritması
KKK	: Karınca Kolonileri Algoritması
MABA	: Melez Ateş Böceği Algoritması
MA	: Memetik Algoritma
NEH	: Nawaz, Encor ve Ham Algoritması
NP	: Polinomial Olmayan (Non-polynomial)
PSO	: Parçacık Sürüsü Optimizasyonu
TA	: Tabu Arama
TB	: Tavlama Benzetimi
YBS	: Yapay Bağışıklık Sistemi

ŞEKİL LİSTESİ

- Şekil 3.1.** Genetik Algoritmaların Genel Akış Şeması
Şekil 3.2. Melez Ateş Böceği Algoritması Akış Şeması
Şekil 3.3. Paralel Makine Sistem Modeli
Şekil 3.4. Esnek Akış Tipi Sistem Modeli

ÇİZELGE LİSTESİ

- Çizelge 3.1.** Genetik Algoritmanın Özellikleri
Çizelge 4.1. Problemin İşlem Süreleri ve Aşamaları
Çizelge 4.2. Tamamlanma Süresi (C_{max}) ve Ortalama Akış Zamanı Değerleri
Çizelge 4.3. Motor Silindir Gömleği İşlem Süreleri
Çizelge 4.4. Algoritmalara Göre İş Sırası ve C_{max} Değerleri
Çizelge 4.5. EATÇ Problemleri İçin Geliştirilen MABA ile Elde Edilen Çözümler
Çizelge 4.6. EATÇ Problemlerinin İşlem Süreleri
Çizelge 4.7. ÇAEATÇ Problemleri İçin Elde Edilen Sonuçlar

1. GİRİŞ

Artan popülasyon sayısı ve gereksinimi ile birlikte azalan hammadde miktarı, insanları, malzemeleri en verimli şekilde kullanma yoluna itmiştir. Bunun sonucunda yapılan tasarımların, en az maliyete sahip olacak şekilde gerçekleştirilmesi hedeflenmiş ve bu amaçla çeşitli yöntemler üretilmiştir. Bu yöntemlerden biri olan optimizasyon, insanlar tarafından en uygun, en rahat, en ekonomik, en kısa, en sağlam gibi sorulara cevap verebilmek için yapılan çalışmalar sonucunda ortaya çıkmış bir çözümdür. Bir tasarım problemini çözmek için insanlar, doğayı gözlemlemiş ve bundan esinlenmiştir. Ateş böceklerinin parlaklığı, sezgisel algoritmanın amaç fonksiyonunu temsil etmektedir. Sürü içerisinde her ateş böceği, optimizasyon işlemi için uygun bir aday çözümünü ifade etmektedir.

Bu araştırmada, ateş böceği algoritmasının çözüm performansının artırılması için melez bir yöntem önerilmiştir. Önerilen yöntemde, ateş böceği algoritmasının işlem adımları, genetik algoritmanın operatörleri yardımıyla melez hale getirilmiştir. Geliştirilen bu yeni yöntem yardımıyla çok amaçlı esnek akış tipi çizelgeleme problemi için en uygun çözüm değerleri elde edilmeye çalışılmıştır.

Çalışmanın ikinci bölümünde, esnek akış tipi çizelgeleme problemi ve ateş böceği algoritması ile literatürde yapılan araştırmalar kısaca özetlenmiştir. Üçüncü bölümde, materyal ve yöntem açıklanmıştır. Dördüncü bölümde melez ateş böceği algoritması ile çok amaçlı esnek akış tipi çizelgeleme problemlerinin çözümlerine yer verilmiştir. Araştırmanın beşinci ve son bölümünde de elde edilen sonuçlar tartışılmış ve gelecekte yapılabilecek çalışmalar ile ilgili bilgiler sunulmuştur.

2. KAYNAK ARAŞTIRMASI

Esnek Akış Tipi Çizelgeleme (EATÇ) problemleri için en çok kullanılan analitik yöntem Dal Sınır Algoritması (DSA)'dır (Ruiz ve Vázquez-Rodríguez, 2010). EATÇ problemi ile DSA kullanılarak yapılan çalışmalar listelenmiştir.

Arthanary (1971), EATÇ problemleri ile ilgili ilk çalışmayı yaparak tamamlanma zamanı (C_{max}) minimizasyonu için bir DSA geliştirmiştir.

Brah ve Hunsucker (1991), maksimum tamamlanma süresini optimize etmek ve tesislerin çizelgeleme sorunlarını çözmek için bir DSA sunmuştur. Araştırmada geliştirilen alt sınırlar ve eleme kuralları, akış tipi probleminin genelleştirilmesine dayanmaktadır. Ayrıca, bir hesaplama algoritması, sonuçlarla birlikte sunulmuştur.

Rajendran ve Chaudhuri (1992), toplam akış süresini minimize etmek için, paralel çok amaçlı akış tipi çizelgeleme problemlerinin çözümünde bir DSA önermişlerdir. Akış tipi çizelgeleme problemi NP (Non-polynomial)-zor olduğundan, problem çözümleri için ayrıca iki aşamalı bir sezgisel algoritma geliştirmişlerdir. Sezgisel çözümleri, önerilen dal-sınır yöntemiyle verilen optimal çözümlerle karşılaştırmışlar, optimale yakın olduğunu belirlemişlerdir.

Dessouky ve ark. (1998), iki aşamalı paralel makine probleminin polinom zamanında da çözülebileceğini, fakat aşama sayısının ikiden fazla olduğu durumlarda problemin NP-zor olduğu göstermişlerdir. Üç aşamalı problemlere en uygun çözümü bulan bir DSA ve örnek problemlerde optimale yakın çözümler sağladığı ispatlanan hızlı bir sezgisel algoritma sunmuşlardır.

Portmann ve ark. (1998), esnek akış çizelgeleme problemini çözmek için en uygun yöntemleri araştırmışlardır. Problemin NP-zor olduğunu belirtmişlerdir. Önerdikleri yöntemde, GA'lar dal-sınır algoritması tarafından verilen ve üst sınırın iyileştirilmesi amacıyla bir dizi tam çözüm popülasyonu oluşturmuştur.

Moursli ve Pochet (2000), EATÇ probleminde C_{max} 'ı minimize etmek için bir DSA önermişlerdir. Yazarlar, üst sınırları hesaplamak için çeşitli sezgisel metotlar kullanmışlardır.

Carlier ve Neron (2000), EATÇ problemlerinin çözümünde hesaplama süresini minimize etmek için bir DSA geliştirmişlerdir.

Néron ve ark. (2001), EATÇ probleminin çözümü için bir DSA önermişlerdir. Önerilen yaklaşımların esnek akış tipi problemleri en iyi şekilde çözmek için bilinen algoritmalarından daha iyi performans gösterdiğini kanıtlamışlardır.

Azizoğlu ve ark. (2001), EATÇ 'de toplam akış zamanı problemini ele almışlardır. En uygun çözümü bulmak için DSA geliştirmişlerdir. Algoritmanın etkinliği bir baskınlık kriteri ile artırılmıştır. Hesaplanan deneyler, algoritmanın makul büyüklükteki problemleri ve orta büyüklükteki problemleri çözdüğünü göstermiştir.

Sawik (2005), sipariş tarihine göre değişen performans ölçümleriyle birlikte, sipariş üzerine üretim ortamında, üretim programlamasına tam sayılı bir yaklaşım sunmuştur. Önerilen tam sayılı programlama yöntemi için yeni kesme kısıtlamaları tanımlamıştır. Gerçek hayatta sipariş üzerine montaj işlemi gerçekleştirilen bir atölyede uygulama yapmıştır.

Allaoui ve Artiba (2006) çalışmalarında, çizelgeleme problemleri literatüründe makinelerin her zaman kullanılabilir olduğu varsayımından bahsetmişlerdir. Bununla birlikte, gerçek hayat endüstrisinde, makinelerin, arıza ve önleyici bakım faaliyetleri nedeniyle bazı kullanım dışı kalma sürelerine maruz kalabileceğini belirtmişlerdir. Çalışmalarında, ilk aşamada sadece bir makine ve ikinci aşamada m makineleri ile C_{max} 'ı en aza indirmek için iki aşamalı EATÇ problemini incelemişlerdir. Her makinenin en fazla bir kullanılmazlık süresine tabi olduğunu kabul etmişlerdir.

Haouari ve Hidri (2011), Çok Amaçlı Esnek Akış Tipi Çizelgeleme (ÇAEATÇ) problemi için yeni alt ve üst sınırları araştırmışlardır. ÇAEATÇ problemini çözmek için optimizasyon temelli bir sezgisel yöntem geliştirmişlerdir. Önerilen sınırlama prosedürlerinin mevcut en iyi olanları tutarlı bir şekilde iyileştirdiğine dair hesaplama deney sonuçlarını sunmuşlardır.

EATÇ problemlerinde sezgisel veya metasezgisel yaklaşımlar kullanılarak yapılan çalışmalar aşağıda listelenmiştir.

Kochhar ve Morris (1987), işleri sisteme sunma sırasına ve her bir makinede daha sonra hangi işe başlayacağına karar verme olmak üzere iki problemi incelemişlerdir. Literatürde yer alan test ve gerçekçi problemler üzerinde önerdikleri yöntemin performansını araştırmışlardır. Miyopik ve yerel arama yöntemleri dahil olmak üzere çeşitli optimizasyon tekniklerini araştırmışlardır.

Sriskandarajah ve Sethi (1989), belli sayıda makine merkezinden oluşan esnek akış tipi problemleri belirli bir üretim ortamı için zamanlama algoritmalarının performansını tartışmışlardır. Her merkez bir veya daha fazla aynı paralel makineye sahiptir. Her parça, her merkezde en fazla bir makine tarafından işlenir. Asgari bitiş zaman çizelgelerini bulma problemini incelemişlerdir. Sezgisel algoritmalar sunmuşlardır.

Wittrock (1992), işlerin bir üretim sistemine yüklenmesini çizelgelemek için sezgisel bir algoritma olan Orchard'ı önermiştir. Orchard baskılı devre kartı hattında kullanılmak üzere olarak tasarlanmış olsa da diğer üretim sistemlerinde de kullanılabilirliğini belirtmiştir.

Gupta ve Tunc (1994), her aşamada birkaç aynı paralel makineden oluşan ve aşamalardaki her işin hazırlık ve işlem sürelerinin ayrı olduğu, iki aşamalı akış tipi çizelgeleme problemini incelemişlerdir. Yazarlar, ilk aşamada sadece bir makine bulunan ve ikinci aşamada aynı paralel makine sayısının toplam iş sayısına eşit veya daha fazla olduğu özel durum için bir polinom optimizasyon algoritması geliştirmişlerdir. Önerilen sezgisel algoritmaları, test etmişlerdir.

Ding ve Kittichartphayak (1994), paralel makinelerin varsayıldığı akış tipi çizelgeleme problemlerini çözmek için yeni bir yöntem sunmuşlardır. Çalışmalarında üç yonteme yer vermişlerdir. Hesaplmalı sonuçlar, birleşik bir yaklaşımın problemi oldukça etkin bir şekilde çözdüğünü göstermiştir.

Guinet ve ark. (1994), aynı paralel makinelerden oluşan birkaç seri atölyede, bağımsız işlerin planlanması problemini incelemişlerdir. Yazarlar, her işin her atölyede bir makine tarafından işlendiğini ve her işin kendi bitiş tarihinin olduğunu ve amacın da azami gecikmeyi en aza indirecek çizelgeleme programını belirlemek olduğunu ifade etmişlerdir. Bu problemi çözmek için bir sezgisel algoritma önermişlerdir. Sezgisellerin kalitesini değerlendirmek için, optimal çözüme ilişkin birkaç Alt Sınır Değeri (ASD)'ni hesaplamışlardır.

Li (1997), hazırlık sürelerinin olduğu EATÇ problemini incelemiştir. Buna göre ileri ve geri tahsis politikalarının performansını değerlendirmiş ve geri tahsis politikasının, belirtilen problemlerin çözümünde daha iyi sonuçlar verdiğini gözlemlemiştir.

Oğuz ve Fikret Ercan (1997) iki aşamalı bir esnek akış tipi çok amaçlı çizelgeleme problemi için etkin sezgisel algoritmalar geliştirmişlerdir. Ayrıca problemler için iki etkili alt sınır türetmişlerdir. Daha sonra, her bir sezgisel çözümün ortalama bağıl farkını alt sınırdan hesaplayarak, sezgisel algoritmaların ortalama performansını analiz etmişlerdir. Önerilen sezgisel algoritmaların, ortalama performansını test etmek için yapılan rastgele oluşturulmuş bir dizi problemde, hesaplanan deney sonuçları, önerilen sezgisel algoritmaların iyi performans gösterdiğini belirlemişlerdir.

Haouari ve M'Hallah (1997), iki aşamalı EATÇ problemleri için sezgisel bir yöntem önermişlerdir. Önerdikleri yöntem, çözdükleri problemlerin %35'inde optimum sonuç elde etmiştir. Önerdikleri yöntemler Tabu Arama (TA) ve Tavlama Benzetimi (TB) algoritmasına dayanmaktadır. Elde ettikleri sonuçları mevcut sezgisel yöntemlerle bulunan çözümlerle ve yeni türetilmiş bir alt sınırla karşılaştırmışlardır. Önerilen sezgiselin etkin sonuçlara ulaştığını belirlemişlerdir.

Artiba ve Riane (1998), üç aşamalı EATÇ problemini incelemişlerdir. C_{max} 'ı en aza indirmek için iki sezgisel yaklaşım önermişlerdir. Hesaplama sonuçları ile önerilen sezgiselin, mükemmel performans sağladığını belirtmişlerdir.

Gupta ve Tunc (1998), iki aşamalı EATÇ problemini, toplam C_{max} 'ı en aza indirmek amacıyla yeni bir çözüm yöntemi geliştirmişlerdir.

Nowicki ve Smutnicki (1998), paralel makinalı akış tipi çizelgeleme problemlerinin çözümünde, yakın çevresini ekleme hareketleri ile keşfeden bir algoritma önermişlerdir. Önerdikleri algoritma, bir grafikte ve bir iş bloğunda kritik bir yol kavramlarını kullanan bir TA tekniğine dayanmaktadır.

Brah ve Loo (1999) çalışmalarında ÇAETAÇ probleminin çözümü için beş sezgisel yöntem önermişlerdir. Çalışmalarında, problem özelliklerinin (her bir işte iş sayısı, makine aşaması sayısı ve paralel işlemcilerin sayısı) etkileri ve regresyon analizi kullanılarak sezgisellerin performansını incelemişlerdir.

Gupta ve ark. (2002) çalışmalarında, zamanlama fonksiyonunu planlama aşaması ile birleştiren permütasyon akış tipi problemi incelemişlerdir. Araştırmada incelenen problemde her iş merkezi paralel aynı makinalardan oluşmaktadır. Her işin farklı bir teslim tarihi olduğu ve aynı sıradaki farklı makine merkezlerinden makineler üzerinde işlenmesi gereken sıralı işlemlerden oluştuğu belirtilmiştir. Yazarlar,

çalışmalarında 20 iş ve 10 makine merkezi olan problemler için hesaplamalı sonuçlarını vermişlerdir.

Jin ve ark. (2002), devre panosu üreten esnek akış tipi bir montaj hattında, üretim süresini minimize etmeye çalışmışlardır. Problemin çözümü için Genetik Algoritma (GA) ve üç alt problemi kullanan global bir yöntem önermişlerdir.

Alisantoso ve ark. (2003), esnek bir akış tipi çizelgeleme problemlerinin çözümü için bağışıklık algoritması önermişlerdir. Önerilen algoritma ile elde edilen sonuçları, iyi bilinen evrimsel algoritmalar olan GA tarafından hesaplanan sonuçlarla karşılaştırmışlardır.

Su (2003), birinci aşamada toplu işlemcili ve ikinci aşamada tek işlemcili esnek iki aşamalı bir akış tipi problemi incelemiştir. Sınırlı bekleme kısıtlamaları olan iki aşamalı akış tipi çizelgeleme probleminin NP-zor olduğunu göstermiştir. Sezgisel bir algoritma ve karma bir tamsayı programını, problemlerin çözümü için önermişlerdir.

Oğuz ve ark. (2003) çok amaçlı, iki aşamalı bir akış tipi çizelgeleme problemi için sezgisel algoritmalar önermişlerdir. Problem için bazı alt sınırlar, sezgisel algoritmaların performans analizinde kullanılmak üzere üretmişlerdir. Daha sonra, önerilen sezgisel algoritmaların ortalama performansı, rastgele oluşturulmuş problem örnekleri kullanılarak yapılan bir hesaplama deneyi ile analiz etmişlerdir. Sonuçlar bu sezgisel algoritmaların hem verimli hem de etkili olduğunu göstermiştir.

Wang ve ark. (2003), EATÇ problemini çözmek için bir sinir ağı tasarlamışlardır. Yazarlar, yapay sinir ağı tasarımının temel bir özelliğinin, problem yapısının ve sezgisel bilginin ağı yapısına ve çözümüne entegre edilmesi olduğunu belirtmişlerdir. Yapay sinir ağının performansını, çözüm kalitesi açısından bilinen güncel sezgisel yöntemlerle karşılaştırmışlardır. Elde ettikleri sonuçlara göre yaklaşımlarının, sezgiselden daha iyi performans sağladığını göstermişlerdir.

Engin ve Döylen (2004), EATÇ problemlerine etkin çözümler bulmak amacıyla, Yapay Bağışıklık Sistemi (YBS) algoritmasını önermişlerdir. Araştırmada, klonal seleksiyon prensibine ve bağışıklık tepkisinin afinite olgunlaşma mekanizmasına dayanan bir hesaplama yöntemi kullanmışlardır. Araştırmada, EATÇ problemleri için C_{max} 'ın en aza indirilmesini performans kriteri olarak dikkate almışlardır. Deneysel sonuçlara göre YBS algoritmasının EATÇ problemlerini çözmek için etkili ve verimli bir yöntem olduğunu belirtmişlerdir.

Wardono ve Fathi (2004), ara stok alanının kısıtlı olduğu çok aşamalı paralel makine problemini incelemişlerdir. Problemin NP-zor olduğunu göstermişlerdir. Problemin çözümü için bir TA algoritması önermişlerdir.

Ercan ve Oğuz (2005), rastgele işlem süreleri ve isteğe bağlı işlemci gereksinimlerini içeren boru hattı çok amaçlı çizelgeleme problemini araştırmışlardır. Problemin çözümü için yerel arama sezgisel algoritması kullanmışlardır. Bunlar TB, TA ve GA'dır.

Zhang ve ark. (2005) çalışmalarında, ilk aşamada m özdeş makine ve ikinci aşamada tek bir makine ile iki aşamalı esnek akış tipi çok amaçlı problemi incelemişlerdir.

Akrami ve ark. (2006), planlamanın sonlu olduğu ve yönetim tarafından sabitlendiği ve üretim aşamalarının seri halde olduğu, ÇAEATÇ problemini incelemişlerdir. Aşamalar arasındaki sınırlı ara tamponlar, tamamlanmış bir parçanın bir makine üzerinde kalabileceği problemini çözmeye çalışmışlardır. Problemin çözümü için bir karışık tam sayılı doğrusal olmayan bir yöntem önermişlerdir. Önerdikleri yöntem hem GA'yı hem de TA metodunu kullanmaktadır.

Belkadi ve ark. (2006) EATÇ probleminin çözümü için karma paralel GA önermişlerdir.

Jin ve ark. (2006), her kademenin paralel özdeş makinelerden oluştuğu çok amaçlı esnek akış tipi çizelgeleme problemini incelemişlerdir. Sınırlı bir planlama üzerinden belirli bir iş kümesi için C_{max} 'ı en aza indiren bir çözüm yöntemi önermişlerdir. Yöntem TB ve değişken derinlik araması metoduna göre çözüm bulmaktadır.

Logendran ve ark. (2006) çalışmalarında EATÇ problemlerinde hazırlık süreleri bağlamında grup planlamasını incelemişlerdir. Amaç olarak, atölyede serbest bırakılan tüm gruplardaki işleri işlemek için gereken C_{max} 'ı en aza indirmeyi seçmişlerdir. Problemin çözümü için TA'ya dayanan üç farklı algoritma geliştirmişlerdir. Farklı seviyelerde hesaplama zorluğuna sahip üç ayrı başlangıç çözümü bulma mekanizması önermişlerdir.

Tang ve ark. (2006) çalışmalarında her aşamada paralel özdeş makinelerden oluşan s aşamalı EATÇ problemini araştırmışlardır. Amaçlarının, işlerin ağırlıklı tamamlanma sürelerinin toplamını en aza indiren bir çözüm bulunması olduğunu

belirtmişlerdir. Problemin NP-zor olduğunu kanıtlamışlardır. Problemin çözümü için bir tam sayılı programlama önermişlerdir. Yaygın olarak kullanılan rahatlatıcı kapasite kısıtlama yönteminden farklı olarak, Lagrangian çarpanlarını tanıtarak, öncelik kısıtlamalarının amaç işlevine gevşetildiği yeni bir Lagrangian gevşeme algoritması sunmuşlardır. Dinamik programlama algoritmasını, işlerin negatif ağırlıklara sahip olabileceği paralel özdeş makine alt problemlerini çözmek için tasarlamışlardır.

Ying ve Lin (2006) çalışmalarında, ÇAEATÇ problemini incelemişlerdir. Problemi çözmek için yeni bir karınca kolonisi sistemi (KKS) sezgiseli önermişlerdir. Geliştirilen sezgisel yöntemi test etmek için, literatürde iyi bilinen iki kıyaslama problem setini çözmüşlerdir. Elde ettikleri sonuçları, literatürdeki GA ve TA ile karşılaştırmışlardır. Hesaplamalı sonuçlar, önerilen KKS sezgisel yaklaşımının mevcut problem için mevcut GA ve TA algoritmalarından daha iyi performans oluşturduğunu göstermiştir.

Zandieh ve ark. (2006) çalışmalarında, hazırlık zamanlarının olduğu EATÇ problemini incelemişlerdir. EATÇ problemlerinin çözümü için bir bağışıklık algoritması önermişlerdir.

Alaykýran ve ark. (2007) EATÇ problemlerini çözmek için Karınca Kolonileri Algoritmasını (KKA) önermişlerdir.

Janiak ve ark. (2007) her aşamada paralel makinelerin bulunduğu akış tipi çizelgeleme problemini incelemişlerdir. Problemin çözümünde üç amaç fonksiyonu tanımlamışlardır. Bunlar toplam ağırlıklı erken bitirme, toplam ağırlıklı gecikme ve toplam ağırlıklı bekleme süresi kriterleridir. Problemi çözmek için üç ayrı yöntem TA ve simüle edilmiş tavlama tekniğine dayalı metotları kullanmışlardır.

Tavakkoli-Moghaddam ve ark. (2009) çalışmalarında, işlemci blokajı ve ara tamponlar olmadan, EATÇ problemini çözmek için yeni bir yerel arama motoru, yani yuvalanmış değişken komşuluk araması ile birleştirilen etkili bir Memetik Algoritma (MA) sunmuşlardır. Önerilen MA, yukarıda belirtilen sorunu çözmek için yeni bir gösterim, operatörler ve yerel arama yöntemi kullanmıştır.

Kahraman ve ark. (2010) EATÇ problemlerinin çözümü için bir GA önermişlerdir. Önerilen algoritma ile literatürde yer alan kıyaslama problemleri için başarılı sonuçlar elde etmişlerdir.

Ruiz ve Vázquez-Rodríguez (2010) çalışmalarında, EATÇ problemlerinin çözümü için önerilen kesin, sezgisel ve metasezgisel yöntemler üzerine bir literatür taraması sunmuşlardır. Çalışmada esnek akış tipindeki araştırma fırsatlarını tartışmışlardır.

Engin ve ark. (2011) çalışmalarında NP-zor olarak bilinen ÇAEATÇ problemlerinin çözümü için GA yöntemini önermişlerdir. Problemlerin çözümü için yeni bir mutasyon operatörü geliştirmişlerdir. Önerdikleri yeni yöntem ile literatürde yer alan kıyaslama problemleri için başarılı sonuçlar elde etmişlerdir.

Li ve ark. (2014a), parçacık sürüsü optimizasyonu ve yerel aramayı birleştiren bir melez algoritmayı, önleyici bakım aktiviteli EATÇ probleminin çözümü için önermişlerdir.

Li ve ark. (2014b), EATÇ için melez bir komşuluk arama algoritması önermişlerdir. Carlier ve Neron'un literatürdeki kıyaslama problemlerini, önerdikleri bu algoritma ile çözmüşlerdir.

Marichelvam ve ark. (2014) ayırık ateş böceği algoritmasını, esnek çok amaçlı akış tipi çizelgeleme probleminin çözümü için önermişlerdir. Yazarlar, EATÇ probleminde, tamamlanma süresinin minimizasyonu için iyileştirilmiş bir guguklu arama algoritması önermişlerdir.

Mensendiek ve ark. (2015) çalışmalarında işlerin yalnızca belirli sabit teslimat tarihlerinde teslim edilebileceği aynı paralel makinelerde programlanacak bir iş kümesinin toplam gecikmesini en aza indirme problemini incelemişlerdir. Problemi çözmek için hem optimal hem de sezgisel prosedürleri geliştirmiş ve ampirik olarak değerlendirmişlerdir.

Karimi ve Davoudpour (2016), birbiriyle çelişen iki hedefin, C_{max} ve toplam ağırlıklı gecikmenin, eşzamanlı olarak minimize edilmeye çalışıldığı, esnek akış tipi problemleri analiz etmişlerdir. Rekabetçi koloni algoritmasının gerçek dünyadaki optimizasyon problemi için çok amaçlı versiyonunu, problemin çözümünde kullanmışlardır.

Chen ve ark. (2018) parti hacimli, EATÇ probleminde, enerji verimliliği için çok amaçlı GA önermişlerdir.

Engin ve Engin (2018) çalışmalarında, esnek akış tipi çok prosesli çizelgeleme problemini, ortak bir zaman penceresinde değerlendirmişlerdir. Bu pencerenin boyutu ve konumu, parametrelerle belirlenmiştir. Yazarlar çalışmalarında, işlerin erken ve gecikmesine katlanılan toplam ceza maliyetinin minimizasyon kriterini, amaç olarak belirlemişlerdir. Problemin çözümünde, global bir arama algoritması olan yeni bir MA kullanmışlardır.

Bu bölümde Ateş Böceği Algoritması (ABA)'nın çizelgeleme ve optimizasyon problemlerinin çözümü ile ilgili literatürde yer alan çalışmalar sunulmuştur.

Yang (2009), ateşböceklerinin hayat tarzlarının diğer canlılardan oldukça farklı olduğunu savunan Yang, ABA olarak adlandırılan bir algoritma geliştirmiş ve ABA'nın teknik özelliklerinden bahsetmiştir. Parçacık sürü optimizasyonu ve GA yöntemleriyle ABA'nı karşılaştırmıştır.

Yang (2010), doğrusal olmayan tasarım problemlerini çözmek için geliştirdiği ABA'nın nasıl kullanılacağını göstermiştir.

Gandomi ve ark. (2011), üst sezgisel bir algoritma olan ABA'nı, sürekli ve kesikli yapısal optimizasyon problemleri için kullanmışlardır. Literatürde yer alan altı yapısal optimizasyon problemleri üzerinde yapılan çalışmaların sonuçlarıyla ABA'nın geçerliliğini doğrulamışlardır.

Chakaravarthy ve ark. (2012), eşit ve değişken boyutlu, alt alanlara sahip bir akış tipi çizelgeleme probleminin toplam akış süresini en aza indirmek için bir çalışma yapmışlardır. Problemi çözmek için ABA ve YBS algoritmalarını kullanmışlardır. Önerilen algoritmalarla elde edilen sonuçlar, diğer geleneksel sezgisel çalışmaların performansıyla da kıyaslamışlardır.

Karthikeyan ve ark. (2012), sınırlı kaynak kısıtlamaları ile ÇAEATÇ problemi için melez ayrık ABA'nı araştırmışlardır. Çalışmalarında, sınırlı kaynak kısıtlamaları ile ÇAEATÇ problemini çözmek için bir karma ayrık ABA sunmuşlardır. Maksimum C_{max} , kritik makinenin iş yükü ve tüm makinelerin toplam iş yükü olmak üzere üç minimizasyon hedefini aynı anda gerçekleştirmeye çalışmışlardır.

Yang ve He (2013), ABA'nın temellerinden söz etmişlerdir. Kesikli arama yöntemleriyle karşılaştırınca, ABA'nın optimum kesikli arama yöntemlerinden daha iyi olduğu sonucuna varmışlardır. Ayrıca yüksek boyutlu optimizasyon problemleri için karşılaştırmalar yapmışlardır.

Vahedi-Nouri ve ark. (2013) ABA'nın, akış tipi problemi için öğrenme ve esnek bakım faaliyetleri üzerindeki etkisini araştırmışlardır. Öğrenme etkileri ve esnek bakım faaliyetleri ile akış tipi çizelgeleme problemini incelemiştir. Amaç olarak, gecikme maliyetleri ve bakım maliyetlerinin toplamını en aza indirmek için, işlerin sıralamasını ve bakım faaliyetlerinin bitiş sürelerini aynı anda belirlenmesini seçmişlerdir. Problemi formüle etmek için karışık tam sayılı bir programlama modeli önermişlerdir. Problemin yüksek karmaşıklığı nedeniyle, simüle edilmiş tavlama algoritmasına ve ABA'na dayanan iyileştirme sezgisel bir yöntem ve melez bir metasezgisel algoritma önermişlerdir.

Marichelvam ve ark. (2014), iki amaçlı EATÇ problemlerini çözmek için bir ABA geliştirmişlerdir.

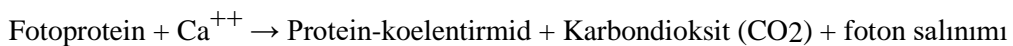
Lo ve ark. (2015) permütasyon akış tipi çizelgeleme probleminde zamanlama sorununun çözümünde, etkinliği artırmak için, bireylerin gerçek değerini önce ayrı iş dizilerine dönüştürmek için bir mekanizma geliştirmişlerdir. Özellikle, başlangıç popülasyonunu en iyi duruma getirmek için bir Kaos Tabanlı Ateş Böceği Algoritması kullanmışlar, bu üstün bir başlangıç ortamı sağlamış ve optimizasyonun kalitesini geliştirmiştir.

3. YÖNTEM VE MATERYAL

3.1. Yöntem

3.1.1. Ateş Böceği Algoritması

Ateş böceği (Lampyridae); böcekler sınıfının kınkanatlılar takımının bir familyasıdır. Kınkanatlıların çoğunun tersine kanat örtüleri (kalınlaşmış ön kanatları) yumuşaktır. Bedeni uzun, rengi gri ya da kahverengidir. Ateş böcekleri tek cinstir, bu yüzden erkek ve dişi ayrımı yoktur. Ateş böcekleri adını, yaydığı parlak sarı-yeşil ışıktan alırlar. Bu böceklerin, yaklaşık 45 metreden görülebilen ışıkları vardır. Kurtçukları etçildir; diğer böcekleri yiyerek beslendikleri için yararlı sayılırlar. Erginleri ise çiçekler ve küçük bitkiler üzerinde yaşarlar. Her böceğin kendine özgü sinyal şifresi vardır. Işık üretim organları karın bölgesinde bulunur. İç kısmı fotojenik hücrelere sahip, ışığı yansıtıcı bir tabakadan oluşur. Işık, organında üretilen yağa benzer lüsiferin maddesi lüsiferinaz enzimi ile kademeli olarak oksijenle yakılır. Bu kimyasal olayda ışık (foton) salınımı olur. Havadaki oksijenin kontrollü tüketimine bağlı olarak ışık zaman zaman yanıp söner. Bu yanıp sönmeler eşlerin birbirleri ile haberleşmesini sağlar. Ateş böceğinin çevreye yaydığı ışık, yavaş yavaş meydana gelen oksitlenme sonucu kimyasal enerjinin ışığa dönüşmesidir. Çıkan ışık tamamen soğuktur. Isı kaybı olmamaktadır. Bu reaksiyonlardaki kimyasal enerjiden ışık-foton oluşturulur. Genellikle soğuk kimyasal reaksiyonlardır. Bu reaksiyonlar denklem (3.1)' deki gibi gerçekleşirler (Yang, 2009):



Ateş böcekleri %100'lük bir verimle ışık üretirler. Ateş böceklerinin yaydıkları ışığın en önemli özelliği, ateşle ve sıcaklıkla ilgisinin bulunmamasıdır. Günümüzde aydınlatma teknolojisinin ulaşmaya çalıştığı amaç, ateş böceğindeki bu sistemdir. Normal bir ampul kendisine verilen enerjinin %4'ünü, floresan ampul ise %10'unu ışığa dönüştürür. Sıradan bir ampulde etkinlik derecesi ancak %3'dür ve %97'si kaybolur. Bu oran, ateş böceğinde %90'ı bulmaktadır. Ateş böceklerinin insanlar tarafından parlak bulunmasının nedeni, ışığı insan gözünün çok duyarlı olduğu bir dalga boyunda yaymalarıdır. Bazı türlerin ise bir insanın kitap okumasına yetecek kadar ışık yayabildiği bilinmektedir (Yang, 2009).

Ateş böcekleri çevrelerine kısa aralıklarla yanıp sönen bir ışık saçar. Bu ışığın yanıp sönmeye ritmi, ateş böceklerinin buluşmasını sağlayan işaret sisteminin bir parçasıdır. Bu özellik ateş böceklerini öbür ışık saçan böceklerden ayırt eden bir özelliktir. Işık saçmasının hızı, sıklığı ve ateş böceklerinin birbirlerine yanıt vermesinden önce geçen bekleme süresi özel anlamlar taşır. Haberleşmek ve çiftleşme mesajı verebilmek için bu ışıkları kullanan ateş böceklerinde, böceklerin türüne göre ışıldaama uzunluğu değişir. Çiftleşme gerçekleştikten sonra ateş böcekleri yumurtalarını bir kayanın ya da herhangi bir bitki örtüsünün altına bırakır. Beş hafta geçtikten sonra, yumurtalar kırılır ve larvalar çıkar. Ateş böceklerinin diğer bir özelliği de hayatlarının her safhasında gelişme halinde olmalarıdır. Ateşböcekleri ışıklarını kendilerini savunmak için kullanır, aynı zamanda saçtıkları ışık tatlara hakkında bilgi mesajı da iletir. Ateş böceklerinin yaşayış biçimlerini belirleyen bu mantık temel alınarak geliştirilmiş olan optimizasyon, ABA olarak adlandırılır (Zhang ve ark., 2007).

Stokastik sayısal optimizasyon arama yöntemlerinden biri olan ateş böceği yöntemi, K. N. Krishnanand ve D. Ghose tarafından 2008 yılında geliştirilmiştir (Krishnanand ve Ghose, 2008). Çok modelli fonksiyonları optimize etmek için önerilen popülasyon tabanlı sezgisel bir algoritmadır.

Ateş böceği sürü optimizasyon yöntemi, ateş böceklerinin sosyal davranışları gözlemlenerek ve taklit edilerek geliştirilmiştir. Bu yöntemin amacı, tüm yerel maksimumları yakalamayı sağlamaktır.

Her birey komşularından aldığı sinyal gücüyle hareketlerini belirlemektedir. Ateş böceğinin eşlerini veya avlarını çekmek için kullandığı ışığın yanıp sönmeye neden olan ışık üreten hücreler "lusiferin" adlı bir kimyasaldır. Bir ateş böceğinin parlaklığı ile diğer ateş böcekleri tarafından çekiciliği doğru orantılıdır (Yang, 2009).

ABA, Dr. Xin-She Yang tarafından geliştirilen ve tropikal iklim bölgelerindeki ateş böceklerinin sosyal davranışlarını temel alan bir meta sezgisel optimizasyon algoritmasıdır (Yang, 2009). Bu algoritmanın diğer algoritmalarla birçok benzerliği bulunmasına rağmen kavram ve uygulamada daha basittir. Bir ateş böceğinin ışıklarını yakıp söndürmesi, diğer ateş böceklerini çekmek için kullandığı bir sinyal sistemidir. Araştırmacılar, ışıklarını yakıp söndürmesinin ateş böceğine, arkadaşlarını bulmada, olası avlarını çekmede ve avcılarının kendilerini korumada yardımcı olduğunu düşünmektedirler. Algoritmanın adlandırılması, algoritmanın oluşturulmasında model

alınan ateş böceklerinden gelmektedir. Ateş böceklerinin tek cins olması ve birbirlerini çekmeleri algoritmanın temelini oluşturur.

Geliştirilmiş olan melez ateşböceği optimizasyon algoritmasının işleyiş adımlarını aşağıda gösterildiği şekilde özetlemek mümkündür (Yang, 2013).

Başlangıçta belli sayıdaki ateş böcekleri uygun arama uzayında yayılmak için rastgele üretilir. Her ateş böceği kendine ait bir pozisyona sahiptir. Ateş böceği algoritmasında, her bir ateş böceği ilk olarak rastgele konumlara yerleştirilir. Bu başlangıç konumu denklemi (3.2)'de verilmiştir.

$$x_i = X_{\min} + r(X_{\max} - X_{\min}) \quad (3.2)$$

Buradaki r değerinin tespiti için 0 ile 1 arasında değişen sayılar içinden rastgele sayı seçme uygulamasından faydalanılır. X_{\min} 1'e eşittir ve X_{\max} değişken seti içindeki toplam değer sayısıdır.

Tasarım uzayında her x_{it} konum değeri için $f(x_{it})$ amaç fonksiyonu değerleri belirlenir.

i numaralı ateş böceğinin t numaralı iterasyon adımına kadar elde edilen en iyi konumu a_{it} ve ateş böceğinin global optimum konumu agt güncellenir.

Her bir ateş böceğinin konumu güncellenir. Bir ateş böceğinin o ana kadar elde edilen en iyi konum değeri, o parçacığın en iyi konum değeri olarak tayin edilir ve her ateş böceği için ayrı bir vektör içinde depo edilir. Belirlenen bu konum, optimizasyon sürecinin başından bu yana küme içerisindeki herhangi bir ateş böceği tarafından elde edilen en iyi konum ise bütün ateş böceklerinin en iyi pozisyon değeri olarak adlandırılır ve yine bir vektör içine kaydedilir. Ateş böceklerinin $t+1$ 'deki konumu, denklem (3.3) (Yang, 2013) yardımıyla hesaplanır.

$$x_i^{t+1} = x_{it} + \beta_0 i^{-\pi r^2} ij(x_j^t - x_i^t) + a_t \varepsilon_i \quad (3.3)$$

Burada x_i^{t+1} ifadesi i . nci ateş böceğinin $t + 1$ 'nci adımdaki konumunu belirtmektedir.

Ateş böceği algoritmasında, iki önemli husus vardır. Bunlar; ışık şiddetinin değişimi ve çekiciliğin formülasyonudur. Kolaylık olması açısından ateş böceğinin kendi çekiciliğini kodlanmış amaç fonksiyonu ile ilişkilendirilmiş parlaklıkla belirlediği varsayılabilir(Gandomi ve ark., 2011).

Ateş böceklerinin aralarındaki mesafe, parlaklığı azalttığı için cazibeyi de azaltmaktadır. Bu değer birçok uygulamada Gaussian formu kullanılarak tahmin edilebilir (Yang, 2009).

$$I = I_0 e^{-\gamma r^2} \quad (3.4)$$

Burada I ışık yoğunluğunu (parlaklık), I_0 başlangıç ışık yoğunluğunu ve γ ışık yoğunluğu katsayısını ifade etmektedir. Işık yoğunluğu r mesafesine bağlı olarak değişir. Denklem (3.5)'de yer alan β çekicilik olarak tanımlanır (Yang, 2009).

$$\beta = \beta_0 e^{-\gamma r^2} \quad (3.5)$$

$r=0$ daki çekicilik β_0 ile gösterilmektedir.

i ve j noktalarındaki her iki ateş böceğinin arasındaki Kartezyen mesafe;

$$r_{ij} = \|x_i - x_j\| \quad (3.6)$$

ile hesaplanırken, her bir ateş böceğinin iterasyonlar içerisinde ulaşacağı yeni konumu aşağıdaki denkleme göre belirlenir.

$$x_i^{t+1} = x_i^t + \beta_0 i^{-\pi r^2} ij(x_j^t - x_i^t) + a_t \varepsilon_i^t \quad (3.7)$$

Denklem (3.7)'deki ikinci terim çekicilikle ilgilidir. İkinci terim ise rastgele sıralama yapmak için kullanılır.

Burada a , $[0,1]$ arasında değişen rastgele bir sayıdır. a ise rastgele sıralama parametresi olup, ε 'un ateş böceğinin yeni konumunun belirlenmesindeki ağırlığını ifade eden bir parametredir.

3.1.2. Melez Ateş Böceği Algoritması

Bu çalışmada, ABA ÇAEATÇ probleminin çözümü için GA operatörü olan mutasyon yöntemi yardımıyla melez hale getirilmiştir.

Genetik Algoritma

1970'li yıllarda GA konusunda ilk çalışmalar John HOLLAND (Holland, 1976) tarafından yapılmıştır. GA, yapay zekânın bir araştırma alanı olup, birçok alanda kullanılmaktadır. Darwin'in doğal seçim ve evrim teorisi ilkelerine dayanan optimizasyon yöntemidir. GA'nın avantajları ve dezavantajları Çizelge 3.1'de gösterilmiştir.

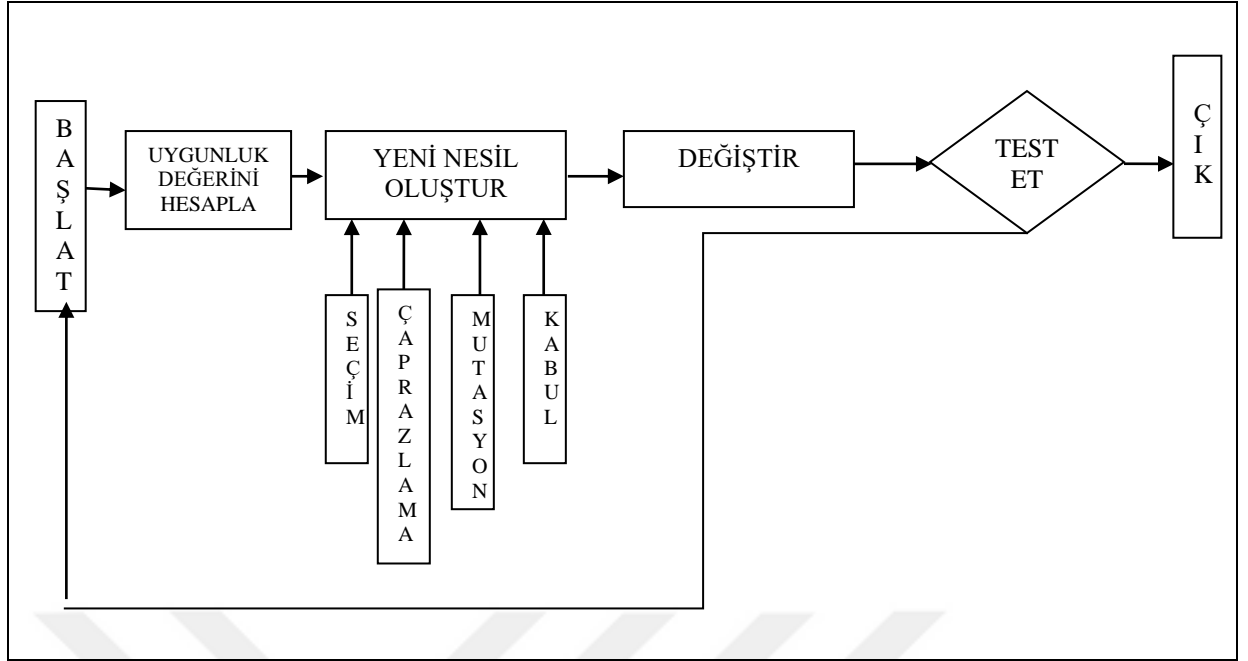
Çizelge 3.1. Genetik Algoritmanın Özellikleri

Genetik Algoritma Özellikleri	
Avantajları	Dezavantajları
Çok amaçlı optimizasyon yöntemleri ile kullanılabilirler.	Son kullanıcının modeli anlaması güçtür.
Çok karmaşık ortamlara uyarlanabilirler.	Problemi GA ile çözmeye uygun hale getirmek zordur.
Kısa sürelerde iyi sonuçlar verebilirler.	Uygunluk fonksiyonunu belirlemek zordur. Çaprazlama ve mutasyon tekniklerini belirlemek zordur.

Algoritmanın Adımları:

1. Uygunluk (fitness) fonksiyonunun tanımlanması
2. Kodlama (genetik kodlama)
3. Rastgele bireylerden oluşacak şekilde başlangıç popülasyonunun seçilmesi
4. Tekrarla (yeterince iyi bir çözüm bulunana kadar)
 - Popülasyondaki bütün bireylerin uygunluk fonksiyonunun hesaplanması
 - Yeni nesil için en iyi bireylerin seçilmesi
 - Çaprazlama ve mutasyon ile yeni neslin oluşturulması
 - Yeni nesli (kromozomları) popülasyona ekle
5. En iyi çözümü döndür.

GA'nın genel akış şeması Şekil 3.1.'de gösterildiği gibidir.



Şekil 3.1. Genetik Algoritmaların Genel Akış Şeması

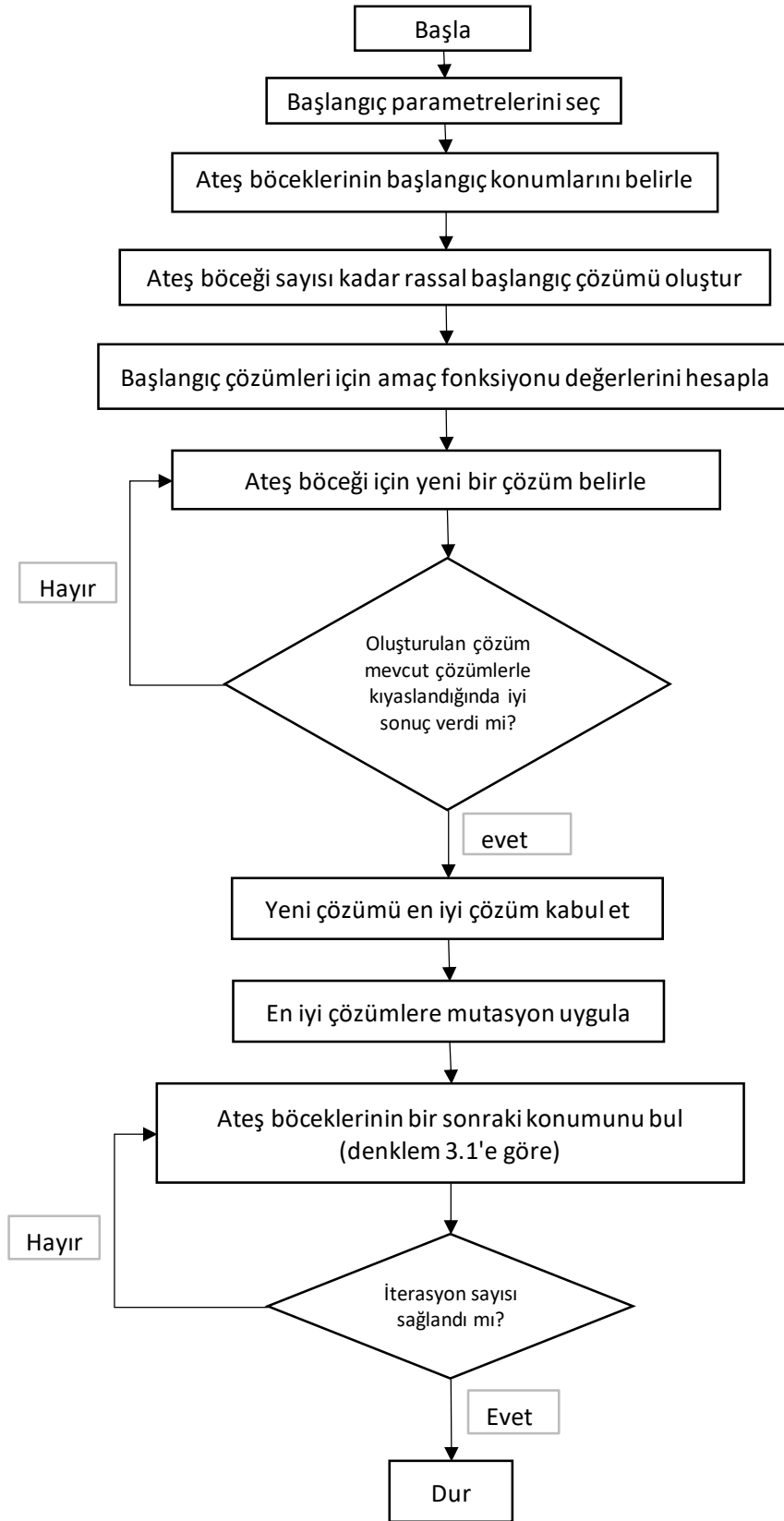
Mutasyon

Yeni nesil üretimi sonucunda, belli bir süre sonra nesildeki kromozomlar tekrar edebilir ve farklı kromozom üretimi azalabilir. Nesildeki kromozom çeşitliliğini artırmak için kromozomlardan bazılarında mutasyona işlemi uygulanır (Emel ve Taşkın, 2002).

Çaprazlama ve mutasyon işlemlerinden sonra, yeni nesil bireyler, anne ve babalarından daha iyi olabilirler. Çalışmada yerleştirme mutasyon yöntemi kullanılmıştır. Kullanılan bu yöntem aşağıda açıklanmıştır.:

- *Yerleştirme Yöntemi*: Dizi içinde rastgele bir genin seçilerek, yine aynı dizi içinde rastgele başka bir konuma yerleştirilmesidir.

Melez Ateş Böceği Algoritması Akış Şeması Şekil 3.2.'de gösterilmiştir.



Şekil 3.2. Melez Ateş Böceği Algoritması Akış Şeması

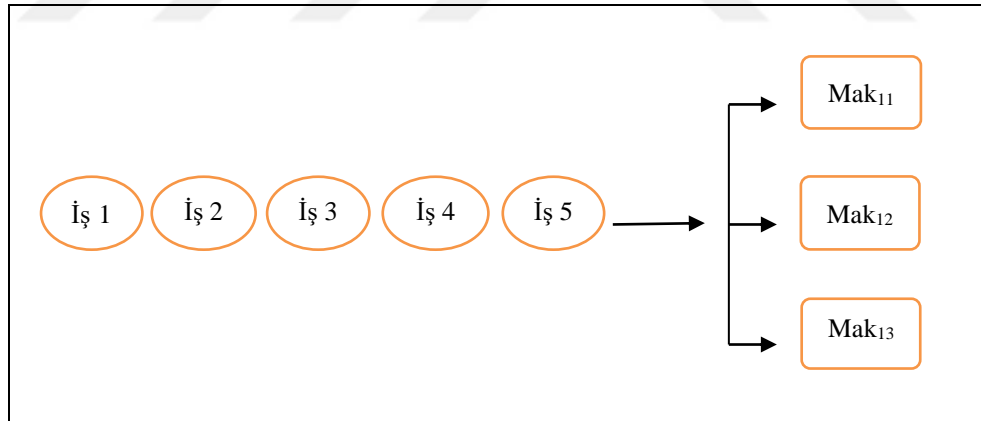
3.2. Materyal

3.2.1. Çok Amaçlı Esnek Akış Tipi Çizelgeleme Problemlerinin Çözümü

Kesikli-parça imalatının olduğu ortamlarda var olan üretim çizelgeleme, zaman düzleminde yapılması gereken görevler için bir veya birden fazla amacı eniyileyecek şekilde kısıt kaynakların kullanılmasıdır. Bu nedenle imalat endüstrilerinde önemli role sahip bir karar verme sürecidir. Kaynakların uygun olarak atanması ile firmanın hedeflerine en iyi şekilde ulaşması sağlanır. Çizelgeleme probleminde, makine kapasitesi kısıtları, teknolojik kısıtlar olmak üzere iki tür kısıt vardır. Çizelgeleme problemi bu iki tip kısıtın birbirine bağlı olması ve uygun çözümdür. Çoğu çizelgeleme problemi kısıtlara bağlı optimizasyon problemi olarak görülmektedir (Cowling ve Johansson, 2002).

Paralel Makine Problemleri

Paralel makine problemlerinde m adet eş makine paralel olarak yerleştirilmiştir. Bütün işler, tek bir operasyondan geçmekte ve işler paralel makinelerin herhangi birinde işlenebilir. Genellikle makine sayısı, iş sayısından azdır. Örnek bir paralel makine sistemi Şekil 3.3.'de gösterilmiştir (Engin ve Döyen, 2004).

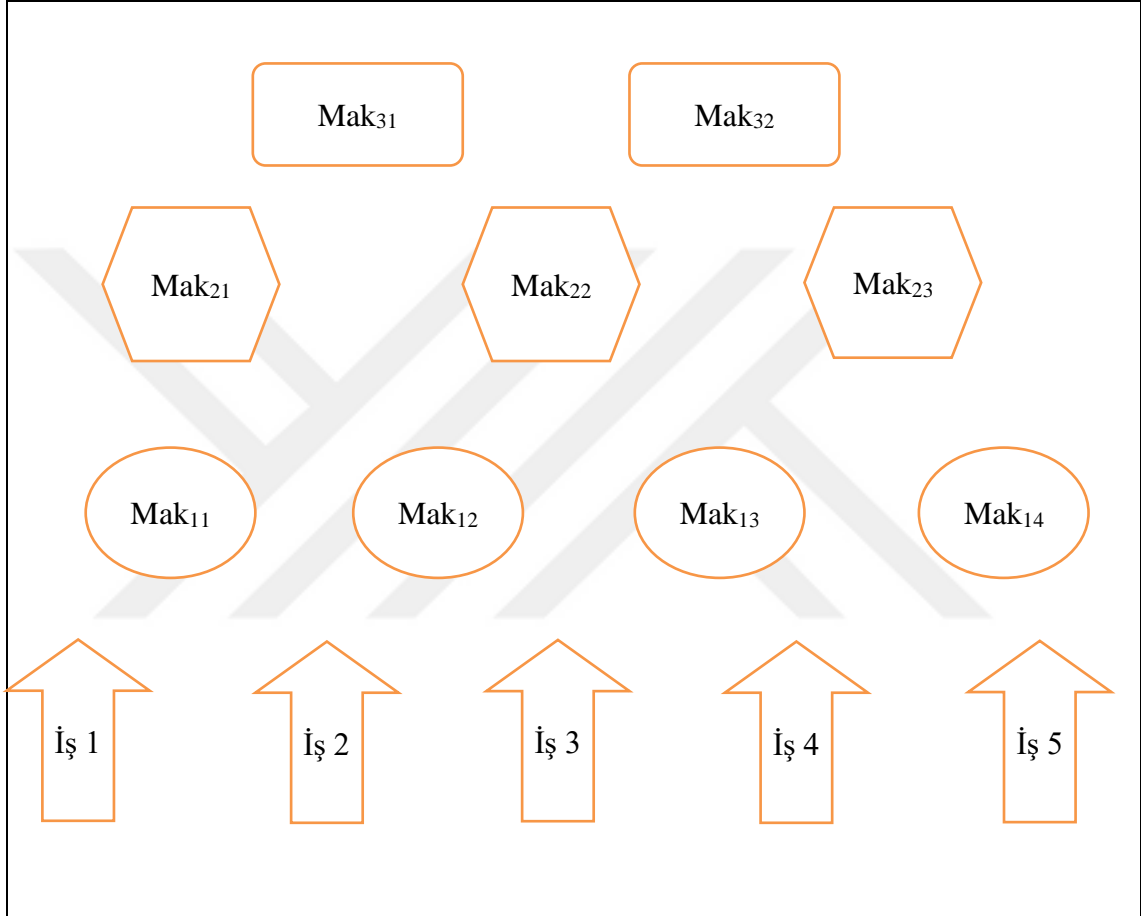


Şekil 3.3. Paralel Makine Sistem Modeli

Esnek Akış Tipi Sistemler

EATÇ problemi; paralel makine problemlerinin genelleştirilmiş şeklidir. Esnek akış tipi sistemde, makineler kademelere yerleştirilmiştir. İşlerin önceliği yoktur, bir makinede bir operasyon başladıktan sonra başka bir operasyonun benzer makinede işlem görmeye başlayabilmesi için öncekinin mutlaka bitirilmesi gerekir. Her makinede belirlenen bir anda maksimum bir işlem yapılabilir. Her kademedeki işlem bekleyen işler

için ayrılan stok alanı kısıtsız kabul edilir. Amaç, genellikle en son işin sistemden ayrılma zamanını (C_{max}), minimize etmektir. EATÇ problemler NP-zor problemlerdir(Gupta ve Kyparisis, 1987). Şekil 3.4.'de esnek akış tipi sistemin yapısı verilmiştir. Sistemde 5 iş vardır ve her birinde makine sayıları sırasıyla 2,3,3 olan 3 kademedeki çizelgenmiştir (Engin ve Döyen, 2004).



Şekil 3.4. Esnek Akış Tipi Sistem Modeli

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. Melez Ateş Böceği Algoritması ile Çok Amaçlı Esnek Akış Tipi Çizelgeleme Probleminin Çözümü

Bu çalışmada, Ateş Böceği Algoritması, ÇAEATÇ probleminin çözümü için GA operatörü olan yerleştirme mutasyon yöntemi yardımıyla melez hale getirilmiştir. Hesaplanan çözümlerin C_{max} , \bar{F} ve amaç değerine göre % sapma değerleri (4.1) denkleminde hesaplanmıştır.

$$\%Sapma = ((MABA \text{ Çözümü} - \text{En İyi Çözüm}) / \text{En İyi Çözüm}) * 100 \quad (4.1)$$

EATÇ ve ÇİEATÇ problemlerinin, MABA ile çözümü için Matlab R2015b dilinde program kullanılmıştır.

Test edilen EATÇ problemleri; Intel Pentium 2.5 GHz işlemcili, 4096 Mb ram ve Microsoft Windows 8.1 Pro with Media Center 64-bit işletim sistemi bulunan bir bilgisayarda hazırlanan program yardımıyla çözülmüştür.

Örnek 1

Marichelvam ve ark. (2014) tarafından bir mobilya üretim işletmesinde elde edilen 20-iş ve 5 aşamalı, EATÇ problemi, çok amaçlı olarak önerilen Melez Ateş Böceği algoritması (MABA) yardımıyla çözülmüştür. Amaç değeri olarak C_{max} ve ortalama akış zamanı (\bar{F}) kullanılmıştır. Amaç değeri denklem (4.2)'e göre hesaplanmıştır.

$$Z = \min (0,5 C_{max} + 0,5 \bar{F}) \quad (4.2)$$

Literatürden alınan 20-iş, 5 aşamalı EATÇ probleminin işlem süreleri ve aşamaları Çizelge 4.1.'de verilmiştir (Marichelvam ve ark., 2014).

Çizelge 4.1. Problemin İşlem Süreleri ve Aşamaları

İş numarası	Zımparalama	Bükme	Kaynak	Presleme	Delme
1	54	48	0	0	0
2	35	63	16	0	0
3	59	22	61	0	0
4	60	0	0	0	0
5	22	36	0	0	0
6	57	11	72	0	0
7	31	0	0	0	0
8	19	0	0	68	0
9	48	72	59	0	0
10	57	20	24	0	0
11	0	0	0	91	0
12	0	0	0	28	0
13	47	76	0	0	0
14	0	0	0	64	0
15	0	0	0	29	0
16	72	53	0	0	39
17	0	0	0	78	0
18	0	0	0	60	0
19	0	0	0	42	0
20	0	0	0	75	0
Makine Sayısı	5	8	5	3	1

Marichelvam ve ark. (2014) tarafından önerilen ve işlem süreleri ile aşamaları Çizelge 4.1.'de verilen problemde, işler İlk Giren İlk Çıkar (İGİÇ) kuralına göre sıralandığında ($J_1, J_2, J_3, \dots, J_{20}$) C_{max} değerinin 13640 saniye olduğu ayrıca kıyaslama problemi, Nawaz, Enscor ve Ham (NEH) (Nawaz ve ark., 1983) algoritması yardımıyla çözüldüğünde C_{max} değerinin, 11180 saniye olduğu, KKA ile çözüldüğünde ise C_{max} değerinin 10968 saniye olduğu literatürde belirtilmiştir (Marichelvam ve ark., 2014).

Marichelvam ve ark. (2014) tarafından önerilen, mobilya üretim işletmesine ait, 15-iş ve 5 aşamadan oluşan, EATÇ problemi, geliştirilen MABA ile çözüldüğünde, C_{max} 10800 saniye olarak hesaplanmıştır. Yeni geliştirilen MABA, İGİÇ, NEH algoritması ve KKA'na göre en küçük C_{max} değerini bulmuştur. C_{max} ve ortalama akış zamanı değerleri Çizelge 4.2.'de sunulmuştur.

Çizelge 4.2. Tamamlanma Süresi (C_{max}) ve Ortalama Akış Zamanı Değerleri

No	C_{max}	C_{max} % Sapma	\bar{F}	\bar{F} % Sapma
1	11400	5,56	2693	7,33
2	11400	5,56	2756	9,84
3	11340	5,00	2525	0,64
4	11160	3,33	2657	5,90
5	10980	1,67	2509	ASD
6	10800	ASD	2575	2,63

Çizelge 4.2. incelendiğinde, önerilen MABA'nın 10800 ile 11400 saniye arasında C_{max} değerlerine ulaşırken ortalama akış zamanı değerinin de 2509 ile 2756 saniye arasında değiştiği görülmektedir.

Örnek 2

Konya'da faaliyet gösteren, motor silindir gömleği üreten bir işletmeden elde edilen 15-iş ve 10 aşamalı, EATÇ problemi, çok amaçlı olarak önerilen MABA, NEH ve İGİÇ yöntemi yardımıyla çözülmüş ve elde edilen sonuçlar birbirleriyle karşılaştırılmıştır. 15-iş, 10 aşamalı EATÇ probleminin işlem süreleri ve aşamaları Çizelge 4.3.'de verilmiştir.

Çizelge 4.3. Motor Silindir Gömleği İşlem Süreleri

İş Numarası	Tepe Kesme	Kaba Dış Tornalama	Kaba İç Tornalama	Taşlama	Kaba Honlama	Son Honlama	Dış Yüzey Taşlama	Krom Kaplama	Son Kontrol	Markalama
1	5	10	15	12	0	0	12	0	7	5
2	8	17	22	25	21	8	18	0	8	6
3	12	28	32	35	38	18	28	0	10	5
4	22	32	36	30	28	15	22	45	6	4
5	3	6	8	10	0	0	12	0	5	3
6	7	10	12	16	12	8	15	0	6	5
7	7	12	14	15	15	9	13	0	3	5
8	10	19	21	23	18	8	18	0	6	5
9	16	32	36	38	0	0	40	0	10	6
10	3	6	8	10	0	0	12	0	3	3
11	6	8	10	13	15	8	17	25	5	6
12	9	12	15	17	20	12	16	0	5	4
13	11	14	16	20	0	0	23	0	5	6
14	16	20	23	28	29	15	28	0	5	4
15	12	15	18	20	22	12	26	20	4	5
Makine Sayısı	2	3	3	2	2	2	2	1	3	1

İşlem süreleri ile aşamaları Çizelge 4.3.'de verilen problemde, işler İGİÇ kuralına göre sıralandığında ($J_1, J_2, J_3, \dots, J_{20}$) C_{max} değerinin 300 saniye olduğu, NEH (Nawaz ve ark., 1983) algoritması yardımıyla çözüldüğünde C_{max} değerinin, 283 saniye olduğu, MABA ile çözüldüğünde ise C_{max} değerinin 282 saniye bulunmuştur. MABA, İGİÇ ve NEH algoritmasına göre kıyaslandığında en küçük C_{max} değerini bulmuştur. Çözüm kalitesi, MABA'nın elde ettiği en iyi çözüm (C_{max}) ile problemin ASD arasındaki yüzde sapma miktarı denklem (4.1) ile hesaplanmıştır.

Çizelge 4.4. Algoritmalara Göre İş Sırası ve C_{max} Değerleri

Metod	İş Sırası														C_{max}	
MABA	6	10	2	1	7	15	13	11	5	14	4	3	9	8	12	282
NEH	10	5	1	11	13	12	7	2	6	15	8	3	4	14	9	283
İGİÇ	13	11	7	12	15	1	6	2	4	14	5	3	9	8	10	300

Örnek 3

Carlier ve Neron'un (2000) $n(\text{iş}) \times s(\text{aşama})$ problemleri, 10x5, 10x10, 15x5 ve 15x10 olarak tanımlanmıştır. İşlem süreleri uniform dağılıma göre oluşturulmuştur. Toplam 77 adet problem vardır. Problemler; iş sayısı ve kademe sayıları özelliklerine göre gruplandırılabilir. Bir problemin yapısını bu iki özellik belirlemektedir. Örnek bir problem notasyonu: "j10c10a1" şeklindedir. Burada; j10, 10 iş bulunduğunu; c10, 10 kademe bulunduğunu; a, kademelerdeki makine yerleşimi yapısını; en sondaki 1 ise örnek indisini göstermektedir.

Yapılan çalışmada, Carlier ve Neron (2000) tarafından önerilen kıyaslama problemleri için MABA ile elde edilen çözüm değerleri; literatürdeki, YBS(Engin ve Döyen, 2004), DSA(Néron ve ark., 2001), KKA(Alaykýran ve ark., 2007), Karma Değişken Komşuluk Arama Algoritması (KDKAA)(Li ve ark., 2014b), Parçacık Sürüsü Optimizasyonu (PSO)(Liao ve ark., 2012) metotları ile karşılaştırılmıştır.

Yapılan analizlerde, MABA'nın çözüm kalitesi ve çözüme ulaşmada harcadığı süre (CPU) ölçülmüştür. Çözüm kalitesi, MABA'nın elde ettiği en iyi çözüm (C_{max}) ile problemin en iyi çözüm (alt sınır değeri) arasındaki yüzde sapma ile ölçülmektedir.

Carlier ve Neron (2000) tarafından önerilen kıyaslama problemleri için MABA ile elde edilen çözüm değerleri; % sapma olarak, literatürdeki YBS, DSA, KKA, KDKAA, PSO metotları ile bulunan sonuçlarla kıyaslanmış ve Çizelge 4.5.'de sunulmuştur.

Çizelge 4.5. EATÇ Problemleri İçin Geliştirilen MABA ile Elde Edilen Çözümler

Problem	C_{max}							% SAPMA					
	ASD	MABA	KDKAA	PSO	YBS	KKA	DSA	MABA	KDKAA	PSO	YBS	KKA	DSA
j10c5a2	88	88	88	88	88	88	88	0,00	0,00	0,00	0,00	0,00	0,00
j10c5a3	117	117	117	117	117	117	117	0,00	0,00	0,00	0,00	0,00	0,00
j10c5a4	121	121	121	121	121	121	121	0,00	0,00	0,00	0,00	0,00	0,00
j10c5a5	122	122	122	122	122	124	122	0,00	0,00	0,00	0,00	1,64	0,00
j10c5a6	110	110	110	110	110	110	110	0,00	0,00	0,00	0,00	0,00	0,00
j10c5b1	130	130	130	130	130	131	130	0,00	0,00	0,00	0,00	0,77	0,00
j10c5b2	107	107	107	107	107	107	107	0,00	0,00	0,00	0,00	0,00	0,00
j10c5b3	109	109	109	109	109	109	109	0,00	0,00	0,00	0,00	0,00	0,00
j10c5b4	122	122	122	122	122	124	122	0,00	0,00	0,00	0,00	1,64	0,00
j10c5b5	153	153	153	153	153	153	153	0,00	0,00	0,00	0,00	0,00	0,00
j10c5b6	115	115	115	115	115	115	115	0,00	0,00	0,00	0,00	0,00	0,00
j10c5c1	68	68	68	68	68	68	68	0,00	0,00	0,00	0,00	0,00	0,00
j10c5c2	74	74	74	74	74	76	74	0,00	0,00	0,00	0,00	2,70	0,00
j10c5c3	71	71	72	71	72	72	71	0,00	1,41	0,00	1,41	1,41	0,00
j10c5c4	66	66	66	66	66	66	66	0,00	0,00	0,00	0,00	0,00	0,00
j10c5c5	78	78	78	78	78	78	78	0,00	0,00	0,00	0,00	0,00	0,00
j10c5c6	69	69	69	69	69	69	69	0,00	0,00	0,00	0,00	0,00	0,00
j10c5d1	66	66	66	66	66	-	66	0,00	0,00	0,00	0,00	-	0,00
j10c5d2	73	73	73	73	73	-	73	0,00	0,00	0,00	0,00	-	0,00
j10c5d3	64	64	64	64	64	-	64	0,00	0,00	0,00	0,00	-	0,00
j10c5d4	70	70	70	70	70	-	70	0,00	0,00	0,00	0,00	-	0,00
j10c5d5	66	66	66	66	66	-	66	0,00	0,00	0,00	0,00	-	0,00
j10c5d6	62	62	62	62	62	-	62	0,00	0,00	0,00	0,00	-	0,00
j10c10a1	139	139	139	139	139	-	139	0,00	0,00	0,00	0,00	-	0,00
j10c10a2	158	158	158	158	158	-	158	0,00	0,00	0,00	0,00	-	0,00
j10c10a3	148	148	148	148	148	-	148	0,00	0,00	0,00	0,00	-	0,00
j10c10a4	149	149	149	149	149	-	149	0,00	0,00	0,00	0,00	-	0,00
j10c10a5	148	148	148	148	148	-	148	0,00	0,00	0,00	0,00	-	0,00
j10c10a6	146	146	146	146	146	-	146	0,00	0,00	0,00	0,00	-	0,00
j10c10b1	163	163	163	163	163	163	163	0,00	0,00	0,00	0,00	0,00	0,00
j10c10b2	157	157	157	157	157	157	157	0,00	0,00	0,00	0,00	0,00	0,00
j10c10b3	169	169	169	169	169	169	169	0,00	0,00	0,00	0,00	0,00	0,00
j10c10b4	159	159	159	159	159	159	159	0,00	0,00	0,00	0,00	0,00	0,00
j10c10b5	165	165	165	165	165	165	165	0,00	0,00	0,00	0,00	0,00	0,00
j10c10b6	165	165	165	165	165	165	165	0,00	0,00	0,00	0,00	0,00	0,00
j10c10c1	113	113	115	115	115	118	127	0,00	1,77	1,77	1,77	4,42	12,39
j10c10c2	116	116	117	117	119	117	116	0,00	0,86	0,86	2,59	0,86	0,00
j10c10c3	98	98	116	116	116	108	133	0,00	18,37	18,37	18,37	10,20	35,71
j10c10c4	103	103	120	120	120	112	135	0,00	16,50	16,50	16,50	8,74	31,07
j10c10c5	121	121	125	125	126	126	145	0,00	3,31	3,31	4,13	4,13	19,83
j10c10c6	97	97	106	106	106	102	112	0,00	9,28	9,28	9,28	5,15	15,46
Ortalama								0,00	1,26	1,22	1,32	1,44	2,79

Çizelge 4.5. (devamı)

Problem	C _{max}							% SAPMA					
	ASD	MABA	KDKAA	PSO	YBS	KKA	DSA	MABA	KDKAA	PSO	YBS	KKA	DSA
j15c5a1	178	178	178	178	178	178	178	0,00	0,00	0,00	0,00	0,00	0,00
j15c5a2	165	165	165	165	165	165	165	0,00	0,00	0,00	0,00	0,00	0,00
j15c5a3	130	130	130	130	130	132	130	0,00	0,00	0,00	0,00	1,54	0,00
j15c5a4	156	156	156	156	156	156	156	0,00	0,00	0,00	0,00	0,00	0,00
j15c5a5	164	164	164	164	164	166	164	0,00	0,00	0,00	0,00	1,22	0,00
j15c5a6	178	178	178	178	178	178	178	0,00	0,00	0,00	0,00	0,00	0,00
j15c5b1	170	170	170	170	170	170	170	0,00	0,00	0,00	0,00	0,00	0,00
j15c5b2	152	152	152	152	152	152	152	0,00	0,00	0,00	0,00	0,00	0,00
j15c5b3	157	157	157	157	157	157	157	0,00	0,00	0,00	0,00	0,00	0,00
j15c5b4	147	147	147	147	147	149	147	0,00	0,00	0,00	0,00	1,36	0,00
j15c5b5	166	166	166	166	166	166	166	0,00	0,00	0,00	0,00	0,00	0,00
j15c5b6	175	175	175	175	175	176	175	0,00	0,00	0,00	0,00	0,57	0,00
j15c5c1	85	85	85	85	85	85	85	0,00	0,00	0,00	0,00	0,00	0,00
j15c5c2	90	90	90	90	91	90	90	0,00	0,00	0,00	1,11	0,00	0,00
j15c5c3	87	87	87	87	87	87	87	0,00	0,00	0,00	0,00	0,00	0,00
j15c5c4	89	89	89	89	89	89	90	0,00	0,00	0,00	0,00	0,00	1,12
j15c5c5	73	73	74	74	74	73	84	0,00	1,37	1,37	1,37	0,00	15,07
j15c5c6	91	91	91	91	91	91	91	0,00	0,00	0,00	0,00	0,00	0,00
j15c5d1	167	167	167	167	167	167	167	0,00	0,00	0,00	0,00	0,00	0,00
j15c5d2	82	82	84	84	84	86	85	0,00	2,44	2,44	2,44	4,88	3,66
j15c5d3	77	77	82	82	83	83	96	0,00	6,49	6,49	7,79	7,79	24,68
j15c5d4	61	61	84	84	84	84	101	0,00	37,70	37,70	37,70	37,70	65,57
j15c5d5	67	67	79	79	80	80	97	0,00	17,91	17,91	19,40	19,40	44,78
j15c5d6	79	79	81	81	81	79	87	0,00	2,53	2,53	2,53	0,00	10,13
j15c10a1	236	236	236	236	236	236	236	0,00	0,00	0,00	0,00	0,00	0,00
j15c10a2	200	200	200	200	200	200	200	0,00	0,00	0,00	0,00	0,00	0,00
j15c10a3	198	198	198	198	198	198	198	0,00	0,00	0,00	0,00	0,00	0,00
j15c10a4	225	225	225	225	225	228	225	0,00	0,00	0,00	0,00	1,33	0,00
j15c10a5	182	182	182	182	182	182	183	0,00	0,00	0,00	0,00	0,00	0,55
j15c10a6	200	200	200	200	200	200	200	0,00	0,00	0,00	0,00	0,00	0,00
j15c10b1	222	222	222	222	222	222	222	0,00	0,00	0,00	0,00	0,00	0,00
j15c10b2	187	187	187	187	187	188	187	0,00	0,00	0,00	0,00	0,53	0,00
j15c10b3	222	222	222	222	222	224	222	0,00	0,00	0,00	0,00	0,90	0,00
j15c10b4	221	221	221	221	221	221	221	0,00	0,00	0,00	0,00	0,00	0,00
j15c10b5	200	200	200	200	200	-	200	0,00	0,00	0,00	0,00	-	0,00
j15c10b6	219	219	219	219	219	-	219	0,00	0,00	0,00	0,00	-	0,00
Ortalama								0,00	1,90	1,90	2,01	2,27	4,60

Yapılan çalışmada, Carlier ve Neron (2000) tarafından önerilen kıyaslama problemleri için MABA ile elde edilen çözüm değerleri literatürdeki YBS, DSA, KKA, KDCAA, PSO ile kıyaslanmıştır. Çizelge 4.5.'de görüldüğü gibi 77 problem için MABA ASD'ne ulaşmıştır. MABA'nın %sapma değeri 0 iken diğer yöntemlerin değerleri sırasıyla 1,58; 1,56; 1,66; 1,85 ve 3,70'e ulaşmıştır. MABA'nın daha iyi sonuçlar bulduğu belirlenmiştir. Carlier ve Neron (2000) tarafından önerilen kıyaslama problemleri için MABA ile literatürdeki metotların işlem süreleri Çizelge 4.7.'de karşılaştırılmıştır. İşlem süreleri "saniye" cinsindedir.



Çizelge 4.6. EATÇ Problemlerinin İşlem Süreleri(sn)

Problem	CPU					
	MABA	KDKAA	PSO	YBS	KKA	DSA
j10c5a2	0,365	0,004	0,002	1	-	13
j10c5a3	0,188	0,004	0,002	1	-	7
j10c5a4	0,162	0,002	0,003	1	-	6
j10c5a5	0,161	0,003	0,013	1	-	11
j10c5a6	0,088	0,002	0,174	4	-	6
j10c5b1	0,236	0,003	0,003	1	-	13
j10c5b2	0,221	0,003	0,003	1	-	6
j10c5b3	0,235	0,003	0,012	1	-	9
j10c5b4	0,305	0,004	0,025	2	-	6
j10c5b5	0,252	0,003	0,001	1	-	6
j10c5b6	0,242	0,002	0,001	1	-	11
j10c5c1	0,127	0,792	0,332	32	-	28
j10c5c2	0,176	0,752	0,535	4	-	19
j10c5c3	0,1	0,141	37	a	-	240
j10c5c4	0,195	0,631	0,215	3	-	1017
j10c5c5	0,191	0,129	0,122	14	-	42
j10c5c6	0,103	0,226	0,405	12	-	4865
j10c5d1	0,225	0,148	0,185	5	-	6490
j10c5d2	0,182	0,142	1,158	31	-	2617
j10c5d3	0,097	0,157	0,098	15	-	481
j10c5d4	0,079	0,675	0,337	5	-	393
j10c5d5	0,189	1,382	0,515	1446	-	1627
j10c5d6	0,179	0,135	0,383	8	-	6861
j10c10a1	0,361	0,227	0,055	1	-	41
j10c10a2	0,254	0,354	0,87	18	-	21
j10c10a3	0,096	0,225	0,017	1	-	58
j10c10a4	0,165	0,007	0,085	2	-	21
j10c10a5	0,266	0,006	0,102	1	-	36
j10c10a6	0,196	0,264	0,239	4	-	20
j10c10b1	0,102	0,006	0,013	1	-	36
j10c10b2	0,211	0,104	0,221	1	-	66
j10c10b3	0,143	0,006	0,014	1	-	19
j10c10b4	0,118	0,068	0,021	1	-	20
j10c10b5	0,201	0,005	0,037	1	-	33
j10c10b6	0,199	0,005	0,056	1	-	34
j10c10c1	0,293	0,571	a	a	-	a
j10c10c2	0,213	0,448	a	a	-	1100
j10c10c3	0,207	0,698	a	a	-	a
j10c10c4	0,178	0,461	a	a	-	a
j10c10c5	0,159	2,125	a	a	-	a
j10c10c6	0,221	0,343	a	a	-	a
Ortalama	0,192	0,275	1,236	47,735	-	729,972

Çizelge 4.6. (devamı)

Problem	CPU					
	MABA	KDKAA	PSO	YBS	KKA	DSA
j15c5a1	0,118	0,006	0,06	1	-	18
j15c5a2	0,205	0,005	0,005	1	-	35
j15c5a3	0,19	0,005	0,006	1	-	34
j15c5a4	0,146	0,005	0,013	2	-	21
j15c5a5	0,095	0,003	0,004	1	-	34
j15c5a6	0,122	0,005	0,006	1	-	38
j15c5b1	0,096	0,005	0,003	1	-	16
j15c5b2	0,191	0,003	0,005	1	-	25
j15c5b3	0,119	0,006	0,03	1	-	15
j15c5b4	0,161	0,007	0	1	-	37
j15c5b5	0,181	0,088	0,086	2	-	20
j15c5b6	0,179	0,004	0,016	1	-	23
j15c5c1	0,262	1,128	4,205	774	-	2131
j15c5c2	0,105	0,491	1,198	a	-	184
j15c5c3	0,214	0,502	2,398	16	-	202
j15c5c4	0,22	0,569	2,208	317	-	a
j15c5c5	0,222	1,968	a	a	-	a
j15c5c6	0,115	0,18	0,191	19	-	57
j15c5d1	0,186	0,004	0	1	-	24
j15c5d2	0,21	0,915	a	a	-	a
j15c5d3	0,211	1,356	a	a	-	a
j15c5d4	0,149	1,325	a	a	-	a
j15c5d5	0,223	0,585	a	a	-	a
j15c5d6	0,25	1,123	a	a	-	a
j15c10a1	0,21	0,008	0,018	1	-	40
j15c10a2	0,278	0,339	0,214	30	-	154
j15c10a3	0,241	0,313	0,171	4	-	45
j15c10a4	0,301	0,212	0,072	12	-	78
j15c10a5	0,189	0,008	0,509	2	-	a
j15c10a6	0,263	0,002	0,468	2	-	44
j15c10b1	0,313	0,009	0,017	3	-	70
j15c10b2	0,245	0,008	0,012	1	-	80
j15c10b3	0,29	0,008	0,007	1	-	80
j15c10b4	0,251	0,008	0,007	1	-	84
j15c10b5	0,338	0,045	0,135	1	-	84
j15c10b6	0,267	0,009	0,006	1	-	67
Ortalama	0,204	0,313	0,402	41,379	-	133,571

a: PSO, YBS ve DSA, toplam çalışma süresi içinde ASD'ye ulaşamamıştır.

Carlier ve Neron (2000) tarafından önerilen 10 işli EATÇ problemleri MABA ile en iyi çözüme ulaşmasında harcanan ortalama süre 0,192 sn iken 15 işli EATÇ problemleri için harcanan ortalama süre 0,204 sn'dir. Çizelge 4.6'daki değerlere bakıldığında literatürdeki YBS, DSA, KKA, KDKAA, PSO metotlarına göre MABA'nın ortalama işlem süresinin daha kısa olduğu görülmüştür.

Çok Amaçlı Esnek Akış Tipi Çizelgeleme Probleminin Çözümü

Performans kriterinde birden fazla amacın sistematik ve eş zamanlı olarak optimize edilmesi çok amaçlı optimizasyon olarak adlandırılır. Çok amaçlı optimizasyon problemlerinde, karar probleminin modeli kurulurken amaç fonksiyonunun oluşturulması zor olabilir. Karar problemlerinin birçoğunda çözümün kalitesini değerlendirmek için birden fazla kriter söz konusudur. Bu kriterleri tek bir amaç fonksiyonunda toplamak her zaman olanaklı olmayabilir. Birden fazla kriterin söz konusu olduğu, özellikle bu kriterlerin birbirleriyle çeliştiği problemlerde farklı çözüm alternatifleri söz konusu olur. Çok amaçlı optimizasyon problemlerinde birden fazla çözüm söz konusudur. Bu çalışmada Carlier ve Neron (2000) tarafından önerilen EATÇ problemleri içerisinde iş sayısı, kademe sayısı ve makine yerleşim yapısı farklı olan problemler seçilmiş ve çok amaçlı MABA ile çözülmüştür. Amaç değeri olarak C_{max} ve ortalama akış zamanı (\bar{F}) kullanılmıştır. Amaç değeri denkleminde denlem 4.2'ye göre C_{max} ve \bar{F} değerlerine farklı yüzde katsayıları sırası ile 10-90,20-80,30-70,40-60,50-50 verilerek sonuçlar üzerinde etkisi incelenmiştir.

Literatürden alınan EATÇ problemlerinin çok amaçlı olarak önerilen MABA yardımıyla çözümünde elde edilen sonuçlar Çizelge 4.7.'de verilmiştir.

Çizelge 4.7.ÇAEATÇ Problemleri İçin Elde Edilen Sonuçlar

Problem	Z	Problem	Z
	71,69		130,57
	79,17		134,94
j10c5b1	85,62	j15c5a1	140,66
	93,78		146,68
	100,12		151,46
	40,90		125,76
	43,71		131,22
j10c5c1	47,10	j15c5b1	135,93
	49,79		141,66
	53,86		145,62
	41,04		71,41
	44,35		73,69
j10c5d1	47,01	j15c5c1	75,85
	49,64		78,56
	53,73		81,14
	50,19		65,62
	61,46		68,04
j10c10a1	71,90	j15c5d1	69,16
	85,85		72,12
	93,95		74,77
	57,26		103,29
	69,43		118,06
j10c10b1	81,13	j15c10a1	132,90
	92,82		147,81
	104,52		162,63
	43,35		100,98
	52,23		110,34
j10c10c1	60,63	j15c10b1	119,85
	69,26		138,46
	77,72		153,63
	50,44		
	55,08		
j10c5a1	59,19		
	63,31		
	67,42		

Yapılan çalışmada, Carlier ve Neron (2000) tarafından önerilen EATÇ problemleri Çizelge 4.7.'de çözüm sonuçları verilmiştir. Sonuçlar yüzde ağırlıklara göre farklılık göstermiştir.

5. SONUÇLAR VE ÖNERİLER

Araştırmada, EATÇ ve ÇAEATÇ problemlerinin çözümünde MABA metodunun işlerliği ve performansı incelenmiştir. EATÇ ve ÇAEATÇ problemleri NP–zor problemler olarak bilinmektedir. Bu problemleri optimal şekilde çözmek için geliştirilmiş bir yöntem bulunmamakta ve daha çok sezgisel yöntemler kullanılmaktadır. Akış tipi çizelgeleme problemlerinde her aşama tek makineden oluşmakta ve işler atölyedeki aşamalarda makineleri aynı sırada ziyaret etmektedir. EATÇ problemi ise klasik akış tipi ve paralel makine problemlerinin bir birleşimi şeklindedir. Bu problemler gerçek hayatta karşılaşılabilecek çizelgeleme problemleridir. Özellikle EATÇ uygulamaları sıkça görülmektedir. Gerçek hayattaki esnek akış tipi sistemlerinde genellikle aynı kademedeki makineler eşittir. Her kademede sadece bir tip görev yapılmaktadır. Her görev sadece kendinden önceki görev tamamlandıktan sonra işlemine başlayabilir.

Bu araştırmada, literatürde optimizasyon problemlerinin çözümü için yoğun olarak kullanılan ABA, ÇAEATÇ probleminin çözümü için GA operatörlerinden olan mutasyon yöntemi yardımıyla melez hale getirilmiştir. Geliştirilen yöntem, literatürde daha önceden çözülmüş bir mobilya üretim işletmesinde elde edilmiş 20-iş ve 5 aşamalı EATÇ probleminin çözümünde çok amaçlı olarak kullanılmıştır. Amaç kriteri olarak, C_{max} ve ortalama akış süresi seçilmiştir. Geliştirilen melez yöntemin ÇAEATÇ problemlerinin çözümünde başarılı sonuç verdiği belirlenmiştir.

Konya’da faaliyet gösteren, motor silindir gömleği üreten bir işletmeden elde edilen 15-iş ve 10 aşamalı, EATÇ problemi, önerilen MABA, NEH ve İGİÇ yöntemi yardımıyla çözülmüş ve sonuçları karşılaştırılmıştır. MABA’nın bu problemin çözümünde de başarılı sonuç verdiği belirlenmiştir.

Carlier ve Neron (2000) ’un EATÇ problemlerinin, MABA yöntemi ile çözümünde bütün problemler için ASD’ye ulaşılmıştır. Carlier ve Neron (2000) ’un EATÇ problemleri için MABA metodu ile elde edilen çözümler, YBS, DSA, KKA, KDKAA, PSO metotları ile elde edilen çözümler ile kıyaslanmıştır. MABA’nın bu yöntemlerin hepsinden daha iyi sonuç verdiği belirlenmiştir.

Gelecekte, literatürde yer alan ve daha büyük ölçekli olan EATÇ problemlerinin çok amaçlı olarak geliştirilen MABA ile çözülebileceği önerilmektedir.

KAYNAKLAR

- Akrami, B., Karimi, B. ve Hosseini, S. M., 2006, Two metaheuristic methods for the common cycle economic lot sizing and scheduling in flexible flow shops with limited intermediate buffers: The finite horizon case, *Applied Mathematics and Computation*, 183 (1), 634-645.
- Alaykýran, K., Engin, O. ve Döyen, A., 2007, Using ant colony optimization to solve hybrid flow shop scheduling problems, *The international journal of advanced manufacturing technology*, 35 (5-6), 541-550.
- Alisantoso, D., Khoo, L. ve Jiang, P., 2003, An immune algorithm approach to the scheduling of a flexible PCB flow shop, *The International Journal of Advanced Manufacturing Technology*, 22 (11-12), 819-827.
- Allaoui, H. ve Artiba, A., 2006, Scheduling two-stage hybrid flow shop with availability constraints, *Computers & Operations Research*, 33 (5), 1399-1419.
- Arthanary, T., 1971, An extension of two machine sequencing problem, *Opsearch*, 8, 10-22.
- Artiba, A. ve Riane, F., 1998, An application of a planning and scheduling multi-model approach in the chemical industry, *Computers in Industry*, 36 (3), 209-229.
- Azizoğlu, M., Çakmak, E. ve Kondakci, S., 2001, A flexible flowshop problem with total flow time minimization, *European Journal of Operational Research*, 132 (3), 528-538.
- Belkadi, K., Gourgand, M. ve Benyettou, M., 2006, Parallel genetic algorithms with migration for the hybrid flow shop scheduling problem, *Advances in Decision Sciences*, 2006.
- Brah, S. A. ve Hunsucker, J. L., 1991, Branch and bound algorithm for the flow shop with multiple processors, *European journal of operational research*, 51 (1), 88-99.
- Brah, S. A. ve Loo, L. L., 1999, Heuristics for scheduling in a flow shop with multiple processors, *European Journal of Operational Research*, 113 (1), 113-122.
- Carlier, J. ve Neron, E., 2000, An exact method for solving the multi-processor flow-shop, *RAIRO-Operations Research*, 34 (1), 1-25.
- Chakaravarthy, G. V., Marimuthu, S. ve Sait, A. N., 2012, Comparison of firefly algorithm and artificial immune system algorithm for lot streaming in m-machine flow shop scheduling, *International Journal of Computational Intelligence Systems*, 5 (6), 1184-1199.
- Chen, T.-L., Cheng, C.-Y. ve Chou, Y.-H., 2018, Multi-objective genetic algorithm for energy-efficient hybrid flow shop scheduling with lot streaming, *Annals of Operations Research*, 1-24.

- Cowling, P. ve Johansson, M., 2002, Using real time information for effective dynamic scheduling, *European Journal of Operational Research*, 139 (2), 230-244.
- Dessouky, M. M., Dessouky, M. I. ve Verma, S. K., 1998, Flowshop scheduling with identical jobs and uniform parallel machines, *European Journal of Operational Research*, 109 (3), 620-631.
- Ding, F.-Y. ve Kittichartphayak, D., 1994, Heuristics for scheduling flexible flow lines, *Computers & Industrial Engineering*, 26 (1), 27-34.
- Emel, G. G. ve Taşkın, Ç., 2002, Genetik Algoritmalar ve Uygulama Alanları, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, 21 (1), 129-152.
- Engin, O. ve Döyen, A., 2004, A new approach to solve hybrid flow shop scheduling problems by artificial immune system, *Future Generation Computer Systems*, 20 (6), 1083-1095.
- Engin, O., Ceran, G. ve Yılmaz, M. K., 2011, An efficient genetic algorithm for hybrid flow shop scheduling with multiprocessor task problems, *Applied Soft Computing*, 11 (3), 3056-3065.
- Engin, O. ve Engin, B., 2018, Hybrid flow shop with multiprocessor task scheduling based on earliness and tardiness penalties, *Journal of Enterprise Information Management*, 31 (6), 925-936.
- Ercan, M. F. ve Oğuz, C., 2005, Performance of local search heuristics on scheduling a class of pipelined multiprocessor tasks, *Computers & Electrical Engineering*, 31 (8), 537-555.
- Gandomi, A. H., Yang, X.-S. ve Alavi, A. H., 2011, Mixed variable structural optimization using firefly algorithm, *Computers & Structures*, 89 (23-24), 2325-2336.
- Guinet, A., Foulon, R. ve Dussauchoy, A., 1994, Scheduling Jobs on Hybrid Flowshops to Minimize Maximum Tardiness, *IFAC Proceedings Volumes*, 27 (4), 245-249.
- Gupta, J. ve Tunc, E., 1998, Minimizing tardy jobs in a two-stage hybrid flowshop, *International Journal of Production Research*, 36 (9), 2397-2417.
- Gupta, J. N. D. ve Tunc, E. A., 1994, Scheduling a two-stage hybrid flowshop with separable setup and removal times, *European Journal of Operational Research*, 77 (3), 415-428.
- Gupta, J. N. D., Krüger, K., Lauff, V., Werner, F. ve Sotskov, Y. N., 2002, Heuristics for hybrid flow shops with controllable processing times and assignable due dates, *Computers & Operations Research*, 29 (10), 1417-1439.
- Gupta, S. K. ve Kyparisis, J., 1987, Single machine scheduling research, *Omega*, 15 (3), 207-227.
- Haouari, M. ve M'Hallah, R., 1997, Heuristic algorithms for the two-stage hybrid flowshop problem, *Operations Research Letters*, 21 (1), 43-53.

- Haouari, M. ve Hidri, L., 2011, Bounding strategies for the hybrid flow shop scheduling problem, *Applied Mathematics and Computation*, 217 (21), 8248-8263.
- Holland, J. H., 1976, Adaptation**Research reported in this article was supported in part by the National Science Foundation under grant DCR 71-01997, In: Progress in Theoretical Biology, Eds: Rosen, R. ve Snell, F. M.: Academic Press, p. 263-293.
- Janiak, A., Kozan, E., Lichtenstein, M. ve Oğuz, C., 2007, Metaheuristic approaches to the hybrid flow shop scheduling problem with a cost-related criterion, *International Journal of Production Economics*, 105 (2), 407-424.
- Jin, Z., Ohno, K., Ito, T. ve Elmaghraby, S., 2002, Scheduling hybrid flowshops in printed circuit board assembly lines, *Production and Operations Management*, 11 (2), 216-230.
- Jin, Z., Yang, Z. ve Ito, T., 2006, Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem, *International Journal of Production Economics*, 100 (2), 322-334.
- Kahraman, C., Engin, O., Kaya, İ. ve Öztürk, R. E., 2010, Multiprocessor task scheduling in multistage hybrid flow-shops: A parallel greedy algorithm approach, *Applied Soft Computing*, 10 (4), 1293-1300.
- Karimi, N. ve Davoudpour, H., 2016, Multi-objective colonial competitive algorithm for hybrid flowshop problem, *Applied Soft Computing*, 49, 725-733.
- Karthikeyan, S., Asokan, P., Nickolas, S. ve Page, T., 2012, Solving flexible job-shop scheduling problem using hybrid particle swarm optimisation algorithm and data mining, *International Journal of Manufacturing Technology and Management*, 26 (1-4), 81-103.
- Kochhar, S. ve Morris, R. J. T., 1987, Heuristic methods for flexible flow line scheduling, *Journal of Manufacturing Systems*, 6 (4), 299-314.
- Krishnanand, K. N. ve Ghose, D., 2008, Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations, *Robotics and Autonomous Systems*, 56 (7), 549-569.
- Li, J.-q., Pan, Q.-k. ve Mao, K., 2014a, Hybrid particle swarm optimization for hybrid flowshop scheduling problem with maintenance activities, *The Scientific World Journal*, 2014.
- Li, J.-q., Pan, Q.-k. ve Wang, F.-t., 2014b, A hybrid variable neighborhood search for solving the hybrid flow shop scheduling problem, *Applied Soft Computing*, 24, 63-77.
- Li, S., 1997, A hybrid two-stage flowshop with part family, batch production, major and minor set-ups, *European Journal of Operational Research*, 102 (1), 142-156.

- Liao, C.-J., Tjandradjaja, E. ve Chung, T.-P., 2012, An approach using particle swarm optimization and bottleneck heuristic to solve hybrid flow shop scheduling problem, *Applied Soft Computing*, 12 (6), 1755-1764.
- Lo, H.-l., Fong, S., Zhuang, Y., Wang, X. ve Hanne, T., 2015, Applying a chaos-based firefly algorithm to the permutation flow shop scheduling problem, *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, 51-57.
- Logendran, R., deSzoeko, P. ve Barnard, F., 2006, Sequence-dependent group scheduling problems in flexible flow shops, *International Journal of Production Economics*, 102 (1), 66-86.
- Marichelvam, M. K., Prabaharan, T. ve Yang, X. S., 2014, Improved cuckoo search algorithm for hybrid flow shop scheduling problems to minimize makespan, *Applied Soft Computing*, 19, 93-101.
- Mensendiek, A., Gupta, J. N. D. ve Herrmann, J., 2015, Scheduling identical parallel machines with fixed delivery dates to minimize total tardiness, *European Journal of Operational Research*, 243 (2), 514-522.
- Moursli, O. ve Pochet, Y., 2000, A branch-and-bound algorithm for the hybrid flowshop, *International Journal of Production Economics*, 64 (1), 113-125.
- Nawaz, M., Ensore, E. E. ve Ham, I., 1983, A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, *Omega*, 11 (1), 91-95.
- Néron, E., Baptiste, P. ve Gupta, J. N. D., 2001, Solving hybrid flow shop problem using energetic reasoning and global operations, *Omega*, 29 (6), 501-511.
- Nowicki, E. ve Smutnicki, C., 1998, The flow shop with parallel machines: A tabu search approach, *European Journal of Operational Research*, 106 (2), 226-253.
- Oğuz, C. ve Fikret Ercan, M., 1997, Scheduling multiprocessor tasks in a two-stage flow-shop environment, *Computers & Industrial Engineering*, 33 (1), 269-272.
- Oğuz, C., Fikret Ercan, M., Edwin Cheng, T. C. ve Fung, Y. F., 2003, Heuristic algorithms for multiprocessor task scheduling in a two-stage hybrid flow-shop, *European Journal of Operational Research*, 149 (2), 390-403.
- Portmann, M. C., Vignier, A., Dardilhac, D. ve Dezalay, D., 1998, Branch and bound crossed with GA to solve hybrid flowshops, *European Journal of Operational Research*, 107 (2), 389-400.
- Rajendran, C. ve Chaudhuri, D., 1992, A multi-stage parallel-processor flowshop problem with minimum flowtime, *European Journal of Operational Research*, 57 (1), 111-122.
- Ruiz, R. ve Vázquez-Rodríguez, J. A., 2010, The hybrid flow shop scheduling problem, *European Journal of Operational Research*, 205 (1), 1-18.

- Sawik, T., 2005, Integer programming approach to production scheduling for make-to-order manufacturing, *Mathematical and Computer Modelling*, 41 (1), 99-118.
- Sriskandarajah, C. ve Sethi, S. P., 1989, Scheduling algorithms for flexible flowshops: Worst and average case performance, *European Journal of Operational Research*, 43 (2), 143-160.
- Su, L.-H., 2003, A hybrid two-stage flowshop with limited waiting time constraints, *Computers & Industrial Engineering*, 44 (3), 409-424.
- Tang, L., Xuan, H. ve Liu, J., 2006, A new Lagrangian relaxation algorithm for hybrid flowshop scheduling to minimize total weighted completion time, *Computers & Operations Research*, 33 (11), 3344-3359.
- Tavakkoli-Moghaddam, R., Safaei, N. ve Sassani, F., 2009, A memetic algorithm for the flexible flow line scheduling problem with processor blocking, *Computers & Operations Research*, 36 (2), 402-414.
- Vahedi-Nouri, B., Fattahi, P. ve Ramezani, R., 2013, Minimizing total flow time for the non-permutation flow shop scheduling problem with learning effects and availability constraints, *Journal of Manufacturing Systems*, 32 (1), 167-173.
- Wang, H., Jacob, V. ve Rolland, E., 2003, Design of efficient hybrid neural networks for flexible flow shop scheduling, *Expert Systems*, 20 (4), 208-231.
- Wardono, B. ve Fathi, Y., 2004, A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities, *European Journal of Operational Research*, 155 (2), 380-401.
- Wittrock, R. J., 1992, The "Orchard" scheduler for manufacturing systems, *International Journal of Production Economics*, 28 (1), 47-70.
- Yang, X.-S., 2010, Firefly algorithm, Levy flights and global optimization, In: Research and development in intelligent systems XXVI, Eds: Springer, p. 209-218.
- Yang, X.-S. ve He, X., 2013, Firefly algorithm: recent advances and applications, *arXiv preprint arXiv:1308.3898*.
- Yang, X. S., 2009, Firefly algorithms for multimodal optimization, Proceedings of the Stochastic Algorithms: Foundations and Applications, *Lecture Notes in Computing Sciences*.
- Ying, K.-C. ve Lin, S.-W., 2006, Multiprocessor task scheduling in multistage hybrid flow-shops: an ant colony system approach, *International Journal of Production Research*, 44 (16), 3161-3177.
- Zandieh, M., Fatemi Ghomi, S. M. T. ve Moattar Hussein, S. M., 2006, An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times, *Applied Mathematics and Computation*, 180 (1), 111-127.

Zhang, C., Li, P., Guan, Z. ve Rao, Y., 2007, A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem, *Computers & Operations Research*, 34 (11), 3229-3242.

Zhang, W., Yin, C., Liu, J. ve Linn, R. J., 2005, Multi-job lot streaming to minimize the mean completion time in m-1 hybrid flowshops, *International Journal of Production Economics*, 96 (2), 189-200.



EKLER**EK-1 Programın Kaynak Kodları**

```

% ===== %
% This program from MATLAB is about using the Firefly
% Algorithm method to find a sequence of job with
% minimum operating time of machines.
% ===== %

% ===== %
% In this program, n is the number of different sequences
% of job and the function Cmax for each sequence of job
% minimum operating time of machines and the function
% improvement_cmax are to improve the operating
% time of the machines using the Firefly algorithm.
% ===== %
clear all
clc
n=input('please enter your initial population :');
[Sira, k, Avrage_Fmax, CMAX]=Cmax(n);
sira_rate=Sira(1:n,:);
impro_sira=zeros(n, k+1);
for i=1:1:n
    [fixed_sequence,fixed_sequence_z]=improvement_cmax(k, sira_rate(i,
:),...
    CMAX(i,1));
    impro_sira(i,1:k)=fixed_sequence;
    impro_sira(i,k+1)=fixed_sequence_z;
end

% Apply the Firefly algorithm to find the minimum Cmax.
function [y, y_z]=improvement_org(t, x_sira, Z1)
[row, column]=size(x_sira);
y=zeros(row, column);
k=zeros(1, t);
jj=0;
for j=1:1:row
    i=1;
    while jj~=1
        jj=0;
        y1=1+rand*(t-1);

```



```

p=round(y1);
if ismember(p, k)
    i=i-1;
else
    r=x_sira(j,p);
    x_sira(j,p)=x_sira(j,i);
    x_sira(j,i)=r;
    k(p)=p;
    z=Comparison_org(x_sira(j,:));
    %y(j, 21)=z;
    if z>Z1(j)
        r=x_sira(j,p);
        x_sira(j,p)=x_sira(j,i);
        x_sira(j,i)=r;
        k(p)=0;
        i=i-1;
    end
end
i=i+1;
if i==15
    i=1;
end
while ismember(i, k)
    i=i+1;
    if i==t
        i=1;
    end
end
for ii=1:1:t
    if k(ii)==0
        jj=jj+1;
    end
end
end
if z<Z1(j)
    y=x_sira;
    y_z=z;
else
    y=zeros(row, column);
    y_z=z;
end

```

```

    end
end

function [sira, Average, c_max]= Cmax(k, po)
S=xlsread('chart.xlsx');
sira=zeros(po,k);
total=zeros(po,1);
c_max=zeros(po,1);
for i=1:1:po
sira(i,:)=randperm(k);
end
c_max=zeros(po,1);
cc=S(1,:);
cc1=S(2,:);
cc2=S(3,:);
cc3=S(4,:);
cc4=S(5,:);
%sira=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15];
for ii=1:po
c=zeros(k,1); c1=zeros(k,1); c2=zeros(k,1); c3=zeros(k,1);
c4=zeros(k,1);
for i=1:1:k
    c(i,1)=cc(sira(ii,i),1);
    c1(i,1)=cc1(sira(ii,i),1);
    c2(i,1)=cc2(sira(ii,i),1);
    c3(i,1)=cc3(sira(ii,i),1);
    c4(i,1)=cc4(sira(ii,i),1);
end
m=zeros(13,k,2);
i=1;
while i<=k

    while (c(i)==0) && (i<k)
        i=i+1;
    end
    if (i==k) && (c(i)==0)
        break;
    end
end

```

```

if find((m(1:3,1,1))==0)
    [row, column]=find((m(1:3,1,1))==0);
    m(min(row),1,1)=c(i,1);
    m(min(row),1,2)=sira(ii,i);
elseif find((m(1:3,1,1))~=0)
    x=sum(m(1:3,1:k,1),2);
    [row, column]=find(x==min(x));
    x1=m(row(1,1),1:k);
    [row1, column1]=find(x1==min(x1));
    m(row(1,1),column1(1,1),1)=c(i,1);
    m(row(1,1),column1(1,1),2)=sira(ii,i);
end
i=i+1;
end
=====
j=1;
while j<=k
    while (c1(j)==0) && (j<k)
        j=j+1;
    end
    if (j==k) && (c1(j)==0)
        break;
    end
    [row2, column2]=find(m(1:3, :, 2)==sira(ii,j));
    if (isempty([row2, column2])) && (min(sum(m(4:6,1:k,1),2))==0)
        x=sum(m(4:6,1:k,1),2);
        [row, column]=find(x==min(x));
        m(min(row)+3,1,1)=c1(j,1);
        m(min(row)+3,1,2)=sira(ii,j);
    elseif min(sum(m(4:6,1:k,1),2))==0
        x=sum(m(4:6,1:k,1),2);
        [row, column]=find(x==min(x));
        m(min(row)+3,1,1)=sum(m(row2, 1:column2))+c1(j,1);
        m(min(row)+3,1,2)=sira(ii,j);
    elseif min(sum(m(4:6,1:k,1),2))~=0
        x=[max(m(4,1:k,1)),max(m(5,1:k,1)),max(m(6,1:k,1))];
        %x=sum(m(4:6,1:10,1),2);
        y1=min(x);
        y=sum(m(row2, 1:column2));
        if y1<=y

```

```

        [column, row]=find(x==min(x));
        %[row, column]=find(x==min(x));
        x1=m(row(1,1)+3,1:k);
        [row1, column1]=find(x1==min(x1));
        m(row(1,1)+3,column1(1,1),1)= y+c1(j,1);
        m(row(1,1)+3,column1(1,1),2)=sira(ii,j);
    else
        [column, row]=find(x==min(x));
        %[row, column]=find(x==min(x));
        x1=m(row(1,1)+3,1:k);
        [row1, column1]=find(x1==min(x1));
        m(row(1,1)+3,column1(1,1),1)= y1+c1(j,1);
        m(row(1,1)+3,column1(1,1),2)=sira(ii,j);
    end
end
j=j+1;
end
%=====
j=1;
while j<=k
    while (c2(j)==0) && (j<k)
        j=j+1;
    end
    if (j==k) && (c2(j)==0)
        break;
    end
    [row2, column2]=find(m(4:6, :, 2)==sira(ii,j));
    if (isempty([row2, column2])) && (min(sum(m(7,1:k,1), 2))==0)
        x=sum(m(7,1:k,1), 2);
        [row, column]=find(x==min(x));
        m(min(row)+6,1,1)=c2(j,1);
        m(min(row)+6,1,2)=sira(ii,j);
    elseif min(sum(m(7,1:k,1), 2))==0
        x=sum(m(7,1:k,1), 2);
        [row, column]=find(x==min(x));
        m(min(row)+6,1,1)=m(row2+3, 1:column2)+c2(j,1);
        m(min(row)+6,1,2)=sira(ii,j);
    elseif min(sum(m(7,1:k,1), 2))~=0
        x=max(m(7,1:k,1));
        %x=sum(m(4:6,1:k,1), 2);

```

```

y1=min(x);
y=m(row2+3, 1:column2);
if y1<=y
    [column, row]=find(x==min(x));
    %[row, column]=find(x==min(x));
    x1=m(row(1,1)+6,1:k);
    [row1, column1]=find(x1==min(x1));
    m(row(1,1)+6,column1(1,1),1)= y+c2(j,1);
    m(row(1,1)+6,column1(1,1),2)=sira(ii,j);
else
    [column, row]=find(x==min(x));
    %[row, column]=find(x==min(x));
    x1=m(row(1,1)+6,1:k);
    [row1, column1]=find(x1==min(x1));
    m(row(1,1)+6,column1(1,1),1)= y1+c2(j,1);
    m(row(1,1)+6,column1(1,1),2)=sira(ii,j);
end
end
j=j+1;
end
%=====
j=1;
while j<=k
    while (c3(j)==0) && (j<k)
        j=j+1;
    end
    if (j==k) && (c3(j)==0)
        break;
    end
    [row2, column2]=find(m(7, :,2)==sira(ii,j));
    if (isempty([row2, column2])) && (min(sum(m(8:10,1:k,1),2))==0)
        x=sum(m(8:10,1:k,1),2);
        [row, column]=find(x==min(x));
        m(min(row)+7,1,1)=c3(j,1);
        m(min(row)+7,1,2)=sira(ii,j);
    elseif min(sum(m(8:10,1:k,1),2))==0
        x=sum(m(8:10,1:k,1),2);
        [row, column]=find(x==min(x));
        m(min(row)+7,1,1)=m(row2+6, column2)+c3(j,1);
        m(min(row)+7,1,2)=sira(ii,j);

```

```

elseif min(sum(m(8:10,1:k,1),2))~=0
    x=[max(m(8,1:k,1)),max(m(9,1:k,1)),max(m(10,1:k,1))];
    %x=sum(m(4:6,1:k,1),2);
    y1=min(x);
    y=m(row2+6, column2);
    if y1<=y
        [column, row]=find(x==min(x));
        %[row, column]=find(x==min(x));
        x1=m(row(1,1)+7,1:k);
        [row1, column1]=find(x1==min(x1));
        m(row(1,1)+7,column1(1,1),1)= y+c3(j,1);
        m(row(1,1)+7,column1(1,1),2)=sira(ii,j);
    else
        [column, row]=find(x==min(x));
        %[row, column]=find(x==min(x));
        x1=m(row(1,1)+7,1:k);
        [row1, column1]=find(x1==min(x1));
        m(row(1,1)+7,column1(1,1),1)= y1+c3(j,1);
        m(row(1,1)+7,column1(1,1),2)=sira(ii,j);
    end
end
j=j+1;
end
=====
j=1;
while j<=k
    while (c4(j)==0) && (j<k)
        j=j+1;
    end
    if (j==k) && (c4(j)==0)
        break;
    end
    [row2, column2]=find(m(8:10, :, 2)==sira(ii,j));
    if (isempty([row2, column2])) && (min(sum(m(11:13,1:k,1),2))==0)
        x=sum(m(11:13,1:k,1),2);
        [row, column]=find(x==min(x));
        m(min(row)+10,1,1)=c4(j,1);
        m(min(row)+10,1,2)=sira(ii,j);
    elseif min(sum(m(11:13,1:k,1),2))==0
        x=sum(m(11:13,1:k,1),2);

```

```

[ row, column]=find(x==min(x));
m(min(row)+10,1,1)=m(row2+7, column2)+c4(j,1);
m(min(row)+10,1,2)=sira(ii,j);
elseif min(sum(m(11:13,1:k,1),2))~=0
x=[max(m(11,1:k,1)),max(m(12,1:k,1)),max(m(13,1:k,1))];
%x=sum(m(4:6,1:k,1),2);
y1=min(x);
y=m(row2+7, column2);
if y1<=y
[ column, row]=find(x==min(x));
%[ row, column]=find(x==min(x));
x1=m(row(1,1)+10,1:k);
[ row1, column1]=find(x1==min(x1));
m(row(1,1)+10,column1(1,1),1)= y+c4(j,1);
m(row(1,1)+10,column1(1,1),2)=sira(ii,j);
else
[ column, row]=find(x==min(x));
%[ row, column]=find(x==min(x));
x1=m(row(1,1)+10,1:k);
[ row1, column1]=find(x1==min(x1));
m(row(1,1)+10,column1(1,1),1)= y1+c4(j,1);
m(row(1,1)+10,column1(1,1),2)=sira(ii,j);
end
end
j=j+1;
end
for r=1:1:5
for r1=1:1:3
if m(r1,r,1)~=0
c_1(r1,r,1)=sum(m(r1,1:r));
end
end
end
m(1:3, :,1)=0;
[v1,w1]=size(c_1);
m(1:3,1:w1,1)=c_1(1:3,1:w1);
for r=1:1:k
[v,w]=find(m(11:13, :,2)==sira(ii,r));
c_ma(r)=max(m(v+10,w));
end

```

```
%f_max(ii,1)=sum(c_ma(1,:));  
%Average=f_max./13;  
c_max(ii,1)=max(c_ma);  
%z1=0.75.*c_max(ii,1)+0.25*Average;  
end
```



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Fatmagül TOPÇAM
Uyruğu : T.C.
Doğum Yeri ve Tarihi : KIRIKKALE-18.02.1991
Telefon : -
Faks : -
e-mail : fatmagulyiltas@gmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Çağrıbey Anadolu Lisesi, Mamak, ANKARA	2009
Üniversite	: Selçuk Üniversitesi, Selçuklu, KONYA	2014
Yüksek Lisans	: Konya Teknik Üniversitesi, Selçuklu, KONYA	-

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2017-...	Hava Kuvvetleri Komutanlığı	Mühendis Teğmen
2014-2017	AKONA Plastik	Üretim Planlama Sorumlusu

UZMANLIK ALANI

YABANCI DİLLER

İngilizce

BELİRTMEK İSTEĞİNİZ DİĞER ÖZELLİKLER

YAYINLAR