



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DAĞITIK PERMÜTASYON AKIŞ TİPİ
ÇİZELGELEME PROBLEMLERİNİN YAPAY
ARI KOLONİ ALGORİTMASI İLE ÇÖZÜMÜ

İpek ARSEVEN

YÜKSEK LİSANS

Endüstri Mühendisliği Anabilim Dalı

Mayıs-2019
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

İpek ARSEVEN tarafından hazırlanan “DAĞITIK PERMÜTASYON AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN YAPAY ARI KOLONİ ALGORİTMASI İLE ÇÖZÜMÜ” adlı tez çalışması 31/05/2019 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Endüstri Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Danışman

Doç. Dr. Ahmet SARUCAN


Üye

Prof. Dr. Orhan ENGİN

Üye

Dr. Öğr. Üyesi Kemal ALAYKIRAN

İmza



Yukarıdaki sonucu onaylarım.

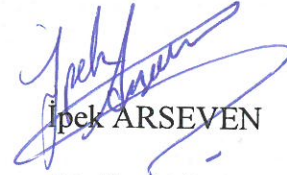
Prof. Dr. Yakup KARA
Enstitü Müdürü

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.



İpek ARSEVEN

Tarih: 31/05/2019

ÖZET

YÜKSEK LİSANS TEZİ

DAĞITIK PERMÜTASYON AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN YAPAY ARI KOLONİ ALGORİTMASI İLE ÇÖZÜMÜ

İpek ARSEVEN

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Ahmet SARUCAN

2019, 34 Sayfa

Jüri

Doç. Dr. Ahmet SARUCAN
Prof. Dr. Orhan ENGİN
Dr. Öğr. Üyesi Kemal ALAYKIRAN

Bu çalışmada amaç fonksiyonu minimum tamamlanma süresi olan dağıtık permütasyon akış tipi çizelgeleme problemleri ele alınmıştır. Bu problemin klasik akış tipi çizelgeleme probleminden farkı, işlerin birden fazla fabrikaya dağıtılmasıdır.

Problemin çözümünde doğadaki arıların besin arama davranışını temel alan yapay arı koloni algoritması kullanılmıştır. Algoritmanın başlangıç çözümleri üretmesinde NEH sezgiselinden yararlanılmıştır. Algoritmanın evrelerinde, (işçi, gözlemci ve kâşif arı evresi) komşu çözümler için yer değiştirme metodu kullanılmıştır. Bu metotta rastgele seçilen iki işin yerleri değiştirilerek farklı iş sıraları elde edilmiştir.

Algoritmanın problem üzerindeki başarısı literatürde iyi bilinen Taillard'ın küçük ve büyük boyutlu test problemleri kullanılarak gösterilmiştir. Algoritma, 14 adet sezgisel ile karşılaştırılmıştır ve en iyi sonuçları sunmuştur.

Anahtar kelimeler: Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemi, Yapay Arı Kolonisi Algoritması

ABSTRACT

MS THESIS

**SOLUTION OF DISTRIBUTED PERMUTATED FLOW SHOP SCHEDULING
PROBLEMS AND ARTIFICIAL BEE COLONY ALGORITHM**

İpek ARSEVEN

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

Advisor: Assoc. Prof. Dr. Ahmet SARUCAN

2019, 34 Pages

Jury

Assoc. Prof. Dr. Ahmet SARUCAN

Prof. Dr. Orhan ENGİN

Asst. Prof. Dr. Kemal ALAYKIRAN

In this study, distributed permutation flow shop scheduling problems with minimum function completion time are discussed. The difference between this problem and the classical flow shop scheduling problem is that the works are distributed to more than one factory.

In order to solve the problem, artificial bee colony algorithm based on the nutrient search behavior of bees in nature was used. The NEH intuition was used to generate the initial solutions of the algorithm. In the phases of the algorithm, the method of displacement for neighboring solutions (employed, onlooker and scout bee phases) was used. In this method, different jobs were obtained by changing the positions of two randomly selected jobs.

The success of the algorithm on the problem has been demonstrated in the literature by using well-known Taillard's small and large sized test problems. The algorithm was compared with 14 intuitive and presented best results.

Keywords: Distributed Permutation Flow Shop Scheduling Problem, Artificial Bee Colony Algorithm

ÖNSÖZ

Akış tipi çizelgeleme problemleri, son zamanlarda literatürde üzerinde çok durulan konu haline gelmiş ve çeşitli yöntem ve sezgiseller ile çözümler geliştirilmeye çalışılmıştır.

Akış tipi çizelgeleme problemlerin farklı bir türü olan dağıtık permütasyon akış tipi çizelgeleme problemleri çalışmada ele alınmıştır. Problemin incelenmesinde yapay arı koloni algoritmasının kullanılabilceği gösterilmiştir.

Bu çalışmada bana destek ve yardımlarını esirgemeyen değerli meslektaşım A. Sezer KÜPELİ'ye, Doç. Dr. Ahmet SARUCAN hocama, ve diğer tüm hocalarımın teşekkürü borç bilirim.

İpek ARSEVEN
KONYA-2019

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	v
ÖNSÖZ	vi
İÇİNDEKİLER.....	vii
SİMGELER VE KISALTMALAR.....	viii
1. GİRİŞ	1
2. LİTERATÜR ARAŞTIRMASI	3
2.1. Dağıtık Permütasyon Akış Tipi Çizelgeleme	3
2.2. Yapay Arı Koloni Algoritması	8
3. MATERYAL VE METOT.....	12
3.1. Akış Tipi Çizelgeleme Problemleri	12
3.1.1. Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemi.....	12
3.2. Yapay Arı Koloni Algoritması	13
3.2.1. Başlangıç Parametreleri	14
3.2.2. Besin Kaynaklarının Oluşturulması.....	14
3.2.3. İşçi Arı Aşaması.....	14
3.2.4. Gözlemci Arı Aşaması.....	15
3.2.5. Kâşif Arı Aşaması.....	15
3.2.6. Temel YAKA Adımları	15
3.3. NEH Sezgiseli.....	16
3.4. Algoritmanın Probleme Uyarlanması	16
3.4.1. Kullanılan Parametreler	18
3.5. YAKA Örneği.....	19
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	23
4.1. Bulgular	23
4.1.1. Küçük Boyutlu Test Problemleri	23
4.1.2. Büyük Boyutlu Test Problemleri	24
5. SONUÇ	30
KAYNAKLAR.....	31
ÖZGEÇMİŞ	34

SİMGELER VE KISALTMALAR

Kısaltmalar

ARUAP	:	Açgözlü Rastgele Uyarlanmış Araştırma Prosedürü
BM-DA	:	Bulanık Mantık Temelli Dağıtım Algoritması
BPAATÇP	:	Blokeli Parti Akışlı Akış Tipi Çizelgeleme Problemi
BT	:	Benzetim Tavlama
DBATÇP	:	Dağıtık Beklemesiz Akış Tipi Çizelgeleme Problemi
DBPATÇP	:	Dağıtık Boşluksuz Permütasyon Akış Tipi Çizelgeleme Problemi
DMPATÇP	:	Dağıtık Montaj Permütasyon Akış Tipi Çizelgeleme Problemi
DPATÇP	:	Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemi
EDA	:	Etkin Dağıtım Algoritması Tahmini
GA	:	Genetik Algoritma
GAYAK	:	Geliştirilmiş Ayrık Yapay Arı Kolonisi
HAGKA	:	Hibrit Ayrık Guguk Kuşu Arama
HYAK	:	Hibrit Yapay Arı Kolonisi
KKA	:	Karıncı Kolonisi Algoritması
KRO	:	Kimyasal Reaksiyon Optimizasyonu
MA	:	Memetik Algoritma
PATÇP	:	Permütasyon Akış Tipi Çizelgeleme Problemi
PSA	:	Parçacık Sürü Algoritması
TA	:	Tabu Arama
TAA	:	Tekrarlı Açgözlü Algoritma
TRAA	:	Tekrarlı Referanslı Açgözlü Algoritması
YAKA	:	Yapay Arı Kolonisi Algoritması

1. GİRİŞ

Günümüzde rekabetin artması ile müşterilerin istek ve beklentilerinin hızlı bir şekilde karşılanması önem kazanmıştır. Fabrika yönetimi, üretim sürecinde verimliliği ve hizmet kalitesinin iyileştirilmesinde önemli bir rol oynar. Özellikle, çok tesisli çizelgeleme problemleri, fabrika otomasyonunda öncelikli bir faktör olarak kabul edilmektedir. Uygulayıcıların ve araştırmacıların çok fabrikalı bir ortamda programlamaya olan ilgisi arttıkça, işlerin farklı fabrikalara / atölyelere ve daha sonradan bu atölyelere atanmalarının eşzamanlı olarak ele alınması ile karakterize olan bu tip karar problemlerin çizelgenmesine olan ihtiyaç giderek artmaktadır.

Çizelgeleme problemlerine ilk çözüm Johnson tarafından geliştirilmiştir. Bu problemlere olan ilgi giderek artmış ve pek çok çözüm yöntemi ortaya konulmuştur. Johnson algoritması n iş-2 makine ve n iş-3 makine çizelgeleme problemlerine kesin çözümler sunmaktadır. Ancak makine sayısı arttıkça problem karmaşık hale gelmekte ve çözüm bulmak giderek zorlaşmaktadır. Bu problemlere literatürde NP-Zor problemler denilmektedir. Problem çözümünün zorluğu, sezgisel ve metasezgisel algoritmaların geliştirilmesinde rol oynamış, bu yöntemler, kısa sürede yakın optimal çözümler bulmayı sağlamıştır.

Metasezgisel ve sezgisel algoritmalar, çizelgeleme problemlerinin çözümünde oldukça yaygın olarak kullanılmaktadır. Problemlerin çok boyutlu olması ve hesaplama karmaşıklığı, bu algoritmaların tercih edilmesinde önemli bir etkidir. Metasezgisel algoritmaların birçoğu doğadan ilham alınarak oluşturulmuştur. Bunlardan bazıları; Genetik Algoritma (GA), Karınca Kolonisi Algoritması (KKA) Parçacık Sürü Algoritması (PSA), Yapay Arı Kolonisi Algoritması (YAKA), Benzetim Tavlama (BT), Tabu Arama (TA) vb.'dir.

Çizelgeleme problemlerinin önemli bir alt dalı akış tipi çizelgeleme problemleridir. Akış tipi çizelgeleme, tüm işlerin aynı sıra ile makineler arasında ilerlediği üretim süreçleri için geliştirilmiş olan bir çizelgeleme dalıdır. Bu özelliği ile akış tipi çizelgeleme; birbirinden farklı, m makine ve n işin bulunduğu; her bir işin m operasyondan oluştuğu, her bir operasyonun farklı makinelerde yapıldığı ve bütün işlerin operasyonlarının aynı sıra ile yapıldığı problemler olarak tanımlanabilir. Literatürdeki çalışmalarda, akış tipi çizelgeleme problemleri için en çok kullanılan amaç fonksiyonu toplam tamamlanma zamanını minimize etmektir (Engin ve Fiğlalı, 2002).

Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemleri (DPATÇP) güçlü bir mühendislik geçmişi olan bir çizelgeleme problemidir. Geçmiş yıllarda çizelgeleme problemleri üzerinde çalışmalar olmasına karşın, DPATÇP ilgili çalışmalar literatürde yetersizdir ve başlangıç aşamasındadır. Dağıtık çizelgeleme konusunda araştırmalar arttıkça, DPATÇP olan ilgi son zamanlarda giderek dikkat çekmeye başlamıştır. Bu çalışmada DPATÇP'nin çözümünde, bal arılarının yiyecek arama davranışlarından esinlenerek geliştirilmiş, YAKA'dan yararlanılmıştır.

Çalışmanın ilerleyen bölümlerinde, akış tipi çizelgeleme problemlerinin bir türü olan dağıtık akış tipi çizelgeleme problemi üzerinde durulmuştur. Çalışmanın ikinci bölümünde problemle ve kullanılan algoritma ile ilgili literatür araştırması yer almaktadır. Çalışmanın üçüncü ve dördüncü bölümünde materyal ve metot tanıtımı yapılmıştır. Son bölümde ise sonuç değerlendirmeleri sunulmuştur.

2. LİTERATÜR ARAŞTIRMASI

Bu bölümde literatür araştırması, iki başlık altında ele alınacaktır. İlk olarak ele alınan DPATÇP'ye ait literatürde yer almış çalışmalar irdelenecektir. İkinci olarak, problem çözümünde kullanılacak olan YAKA ile ilgili literatürde yer almış çalışmalar incelenecektir.

2.1. Dağıtık Permütasyon Akış Tipi Çizelgeleme

Ruiz ve ark. (2019), minimum tamamlanma süresi kısıtlı DPATÇP'nin çözümünde başlama, inşa ve imha prosedürlerini içeren yerel bir arama olan Tekrarlı Açgözlü Algoritmayı (TAA) önermişlerdir.

Ling-Fang ve ark. (2018), dağıtık boşluksuz permütasyon akış tipi çizelgeleme probleminin (DBPATÇP) tamamlanma süresi minimizasyonu için iki aşamalı bir Memetik Algoritma (MA) önermişlerdir. DBPATÇP'nin boşluksuz özelliğine göre, ilk popülasyonu üretmek için hibrit bir yöntem, aramayı ve kullanımı dengelemek için de iki arama aşaması tasarlamışlardır. Her aşamada, fabrika görevlerini ve iş sırasını ayarlamak için özel operatör kullanmışlardır. 420 küçük ve 660 büyük ölçekli kıyaslama örneği ile sayısal testler yapılarak etkinliğini test etmişlerdir.

Pan ve ark. (2018), DPATÇP'nin toplam akış süresi kriteri ile çözülmesi için ayrık YAKA'yı önermişlerdir. Ekleme operatörü, arama sürecinde komşu çözümler üretmek üzere uyarlamışlardır. İşçi, gözlemci ve kâşif arıları içeren arama aşamaları probleme göre tasarlamışlardır. DPATÇP'nin toplam akış süresi kriteri ile çözülmesi için önerilen ayrık YAKA'nın etkinliğini, 720 test problemi kullanılarak göstermişlerdir.

Fernandez-Viagas ve ark. (2018) toplam akış süresini en aza indirmek için DPATÇP'yi ele almışlardır. CPU sürelerinde yüksek kaliteli çözümler elde etmek için on sekiz yapıcı sezgisel ve elde edilen çözümleri daha da geliştirmek için yinelemeli bir geliştirme algoritması önermişlerdir.

Wang ve ark. (2017a) tamamlanma süresi kriterine göre makine arızası altındaki DPATÇP'leri ele almak için Dağıtım Algoritmasının bulanık mantık temelli melez bir algoritma (BM-DA) önermişlerdir. Daha umut verici bir arama alanı keşfetmek için BM-DA, dağıtım algoritması tahmin modelini, yeni birey üretmek için GA'nın çaprazlama ve mutasyon operatörleri ile melezleştirir. BM-DA'da, olasılıksal model

tarafından üretilen bireylerin oranını dinamik olarak ayarlayarak popülasyon çeşitliliğini korumak için yeni bir bulanık mantık temelli uyarlamalı evrim stratejisini benimsemişlerdir.

Dağıtık Montaj Permütasyon Akış Tipi Çizelgeleme Problemi (DMPATÇP), iki göreceli bağımsız evreden, yani operasyon ve montaj aşaması tarafından oluşturulan, yaygın olarak uygulanan bir imalat modelini temsil eder. (Wang ve ark., 2017b) maksimum ağırlık/gecikme maliyetini en aza indirmek üzere, tam zamanında üretimli DMPATÇP'yi oluşturmak için geleneksel DMPATÇP'nin iki aşaması arasındaki tam zamanında üretim kısıtını eklemiştir. Değişken komşu arama tabanlı MA, birkaç yeni komşu çözüm dahil edilerek önerilmiştir. Farklı operatörlerin ve algoritmaların sonuçlarını karşılaştırmak için 10 küçük ve 5 büyük ölçekli kıyaslama problemi üzerinde hesaplama testleri gerçekleştirmişlerdir.

Deng ve Wang (2017) çok hedefli DPATÇP'yi tamamlanma süresi ve toplam gecikme kriterleri ile çözmek için rekabetçi MA önermişlerdir. Önerilen algoritmada iki farklı hedefe karşılık gelen iki popülasyon kullanmışlardır. Her popülasyon için hedefe özgü bazı operatörler ve iki popülasyon arasında özel bir etkileşim mekanizması tasarlamışlardır.

Shao ve ark. (2017) TAA kullanarak, Dağıtık Beklemesiz Akış Tipi Çizelgeleme Problemini (DBATÇP) tamamlanma süresi optimizasyonu kriterine göre ele almışlardır. İlk olarak komşu çözüm bulma süresini azaltmaya yönelik hızlandırma yöntemlerini araştırmışlardır. İkincisi başlangıç çözümü üretmek için NEH sezgiseli uygulamışlardır. Üçüncü olarak, yerel optimumdan kaçınmak için fabrika ataması ve iş sırası düzenlemesine dayanan dört komşu yapısı kullanmışlardır. Son olarak, komşu hareketlerini ve yerel arama yöntemlerini etkin bir şekilde düzenlemek için bunları değişken komşu arama, değişken komşu nesli ve rasgele komşu yapısı çerçevesine dahil etmişlerdir. Sonuçta, DBATÇP için iyi bilinen Naderi ve Ruiz test veri setinden 720 örnek için en iyi çözümler önermişlerdir.

Ying ve ark. (2017) tamamlanma süresini minimize etmek amacıyla DBPATÇP'yi araştırmışlardır. Bu problemi etkin bir şekilde çözmek için Tekrarlı Referanslı Açgözlü Algoritmasını (TRAA) önermişlerdir. TRAA'nın performansı, TAA, ve Karışık Tamsayılı Doğrusal Programlama modeliyle kıyaslamışlardır. Hesaplama sonuçları, önerilen TRAA'nın TAA'yı geride bıraktığını göstermişlerdir.

Olasılıklı memetik çerçeve, her bireyin öğrenme yoğunluğunu kontrol ederek keşif ve sömürüyü dengeleyen yeni bir MA çerçevesidir. Duan ve ark. (2017), hem

makine hem de iş sırasına bağlı hazırlık süresi kısıtlamalarını DPATÇP 'de ele almışlardır.

Bargaoui ve ark. (2017) DPATÇP maksimum tamamlanma süresini en aza indirgemeyi amaçlayan bir meta sezgisel olan Kimyasal Reaksiyon Optimizasyonu (KRO) önermişlerdir. KRO'da, etkili NEH sezgiseli ilk molekül popülasyonunu oluşturmak için uyarlamışlardır. Önerilen algoritmanın performansını doğrulamak için, Taillard'ın 720 büyük örnek üzerinde test etmişlerdir.

Li ve ark. (2016) her fabrika için farklı taşıma süreleri ve yükleme kapasitesinin dikkate alındığı yeni bir tür olan DPATÇP'yi önermişlerdir. Ayrıca, farklı fabrikalara olan mesafelerin eşit olmadığı, nakliye programının ve yükleme kapasitelerinin de dikkate alındığı varsayılmıştır. Maksimum tamamlanma süresini en aza indirmek amacıyla problemi çözmek için BT tabanlı yerel arama kullanmışlardır.

Lin ve Ying (2016) minimum tamamlanma süresi kriterli DBATÇP için karma tamsayılı programlama matematiksel modeli ve Yinelenen Kokteyl Açgözlü Algoritması kullanarak çözüm geliştirmişlerdir.

Deng ve ark. (2016a) Dağıtık İki Makineli Akış Tipi Çizelgeleme Problemini tamamlanma süresi kriteri ile çözmek için geliştirilmiş bir Uyum Arama algoritması önermişlerdir. Parametre ayarlarının etkisi, Taguchi deney tasarımı yöntemine dayanarak incelenmiş ve sayısal deneyler gerçekleştirmişlerdir.

Deng ve ark. (2016b) tamamlanma süresi ve toplam karbon emisyonu kriterli çok amaçlı DPATÇP'nin çözümünde rekabetçi MA önermişlerdir. Önerilen algorithmada yerel arama kullanmışlardır.

Lin ve Zhang (2016) DMPATÇP'yi tamamlanma süresini minimize etmek amacıyla çeşitli yeni sezgiselleri birleştiren Hibrit Biyo-Coğrafyaya Dayalı Optimizasyon algoritmasını önermişlerdir. İlk olarak, yeniden yol bağlama sezgiseli, montaj sırasını optimize etmek için göç aşamasında ürün yerel arama stratejisi olarak kullanmışlardır. İkinci, her ürün için iş permütasyonunu belirlemek üzere mutasyon aşamasında ekleme esaslı bir sezgisel tarama kullanmışlardır. Ardından, problem özelliklerine dayalı yeni bir yerel arama yöntemi tasarlamışlardır.

Wang ve ark. (2016) DPATÇP'yi çözmek için Hibrit Ayırık Guguk Kuşu Arama (HAGKA) algoritmasını önermişlerdir. HAGKA için, DPATÇP'nin çözümünde parametre ayarının etkisini araştırmışlardır. Ayrıca, 420 küçük ve 720 büyük boyutlu örneğe dayalı sayısal simülasyonlar gerçekleştirmişlerdir.

Rifai ve ark. (2016) aday özelliğini, dağıtık aday permütasyon akış tipi çizelgeleme modeline eklenerek geliştirilen dağıtık sürelerin yeni bir modelini önermişlerdir. Bu problemde işler, aynı özelliklere sahip bir dizi makineyi riske sokan fabrikalarda işlenecek çok sayıda geçişli katmanı olan belirli bir iş kümesi olarak tanımlanmaktadır. Çalışmanın amacı, kullanılabilir fabrika sayısını, belirli fabrikaya işlerin atanmasını, fabrikaya atanan iş sırasını, aynı zamanda üretim süresini, toplam maliyeti ve ortalama gecikmeyi en aza indirmek için üç hedefi yerine getirmek şeklinde tanımlamışlardır.

DPATÇP yakın zamanda birden fazla fabrikanın işleri işlemek için uygun olduğu düzenli akış tesis ayarının bir genellemesi olarak önerilmiştir. Problemin iki boyutu vardır: fabrikalara iş atama ve her fabrikaya atanan işlerin zamanlanması. Naderi ve Ruiz (2014) tamamlanma süresini optimize etmede bu problem için bir dağılım arama yöntemi önermişlerdir.

Hatami ve ark. (2013) karmaşık tedarik zincirlerini modellemek ve incelemek için iki aşamalı bir DMPATÇP'yi önermişlerdir. Bu problem DPATÇP'nin geliştirilmesidir. DMPATÇP'nin ilk aşaması aynı üretim fabrikalarından oluşmaktadır. Her biri, ikinci montaj aşamasında nihai ürünlere monte edilecek işleri üreten bir akış atölyesidir. Amaç tamamlanma süresini en aza indirmektir.

DPATÇP çok tesisli ortamlarda önemli uygulamalara sahip olan akış tipi çizelgeleme alanında yeni önerilen bir konudur. Lin ve ark. (2013) tüm fabrikalar arasında maksimum tamamlanma süresini en aza indirmek için değiştirilmiş bir TAA önermişlerdir. Önceki yaklaşımlarla karşılaştırıldığında, önerilen algoritma DPATÇP'nin çözümünde daha basit ancak daha etkili ve verimli olduğunu gözlemlemişlerdir. Önerilen TAA'nın performansını doğrulamak için, Taillard'ın test problemlerini kullanmışlardır.

Wang ve ark. (2013), DPATÇP'yi çözmek için Etkin Dağıtım Algoritması tahmini (EDA) önermişlerdir. İlk olarak, en erken tamamlama süreli fabrikası kuralı, uygulanabilir programlar oluşturmak ve program objektif değerini hesaplamak için permütasyona dayalı kodlama için kullanmışlardır. Ardından, çözüm alanının olasılık dağılımını tanımlamak için bir olasılık modeli oluşturmuş ve olasılık modelini üstün bireylerle güncellemek için bir mekanizma sağlamışlardır. Olasılık modelini örnekleyerek, gelecek vaat eden arama bölgesi arasında yeni bireyler oluşturmuşlardır. Ayrıca, yerel sömürüyü geliştirmek için, bazı yerel arama operatörleri problem özelliklerine dayanarak tasarlanmış ve gelecek vaat eden bireyler için kullanmışlardır.

Ayrıca, EDA'nın parametre ayarlarının etkisi, deneylerin Taguchi tasarım yöntemine dayanarak incelemiş ve uygun bir parametre ayarı önermişlerdir. Son olarak, 420 küçük boyutlu örneğe ve 720 büyük boyutlu örneğe dayalı sayısal simülasyonlar gerçekleştirmişlerdir.

Gao ve ark. (2013) DPATÇP'nin çözümü için tabu stratejisinden yararlanan yeni bir TA önermişlerdir. Mahalle oluşturmak için işlerin alt sıralarını değiştiren bir yöntem sunulmuştur. Gelişmiş bir yerel arama yöntemi önerilmiş ve aynı zamanda TA ile birleştirilmiştir. Algoritmayı test etmek için, Taillard'ın test problemleri kullanmışlardır. Yapılan deneyler, önerilen tabu algoritmasının, diğer sezgisel algoritmalarını geride bıraktığı gözlemlenmiştir.

Gao ve Chen (2011b) NEH tabanlı bir sezgisel algoritma önermişlerdir. DPATÇP'nin gönderme kuralında, fabrikalara bir seferde bir işi yerleştirmek yerine, bir seferde bir grup iş yerleştirmektedir. Sonuçlar, önerilen algoritmanın çözüm kalitesinin mevcut sezgiselden daha iyi olduğunu ortaya koymuştur.

Gao ve Chen (2011a) maksimum tamamlanma süresini en aza indirmek amacıyla GA tabanlı bir algoritma önermişlerdir. Önerilen algoritmada, geçit ve mutasyon operatörleri, kısmi iş dizileri kümesinin kullanıldığı DPATÇP çözümleri uygun hale getirmek üzere tasarlamışlardır. Ayrıca, komşu çözümleri keşfetmek için etkili bir yerel arama yöntemi kullanılmıştır.

Liu ve Gao (2010) DPATÇP için sezgisel bir çözüm yöntemi olarak elektromanyetizma benzeyen bir mekanizma (EM) algoritması uygulanmasını önermişlerdir. Bir DPATÇP 'nin ayrık etki alanına uyması için; mesafe, hareket, yük ve kuvvet hesaplamasının yeniden tanımlanması dahil olmak üzere orijinal EM algoritmasında değişiklikler yapmışlardır. Önerilen algoritmanın performansını arttırmak için, kritik fabrika konseptini kullanan bir değişken komşu arama mekanizması geliştirilmiştir. Naderi'nin Taillard örneklerinden uyarlanmış test problemleri kullanılarak yeni en iyi bilinen çözümler bulmuşlardır.

Permütasyon akış tipi çizelgeleme problemi temel alınarak birden fazla fabrikanın dahil olduğu yeni bir problem ortaya çıkarılmıştır (Naderi ve Ruiz, 2010). Bu genelleme kapsamında, her biri seri halinde yerleştirilmiş m makineli toplam F özdeş fabrika olduğu varsayılmıştır. F fabrikaları arasında bir n kullanılabilir iş kümesi dağıtılmış ve her bir fabrikaya atanan işler için bir işlem dizisi oluşturulmuştur. Optimizasyon kriteri, fabrikalar arasında tamamlanma süresinin veya üretim süresinin

minimumuna indirilmesi olarak belirtilmiştir. DPATÇP'yi altı farklı alternatif karışık tamsayı doğrusal programlama modeli ile tanımlamışlardır. İki adet fabrika atama kuralı kullanılarak 14 adet sezgiselin ve mahalle arama prensibine dayanan algoritmanın performansını değerlendirmişlerdir.

2.2. Yapay Arı Koloni Algoritması

Blokeli parti akışlı akış tipi çizelgeleme problemi (BPAATÇP), ardışık makineler arasında ara tamponlar bulunmazken, her bir işin birkaç alt noktaya bölündüğü birden fazla makine üzerinde bir dizi iş planlamaktır. BPAATÇP, geleneksel atölye çizelgeleme problemlerinden ek sınırlamalar ile ayrılmaktadır. Gong ve ark. (2018), BPAATÇP için iki çelişen kritere (tamamlanma süresi ve erkencilik) sahip Hibrit Yapay Arı Kolonisi (HYAK) algoritması önermişlerdir.

Ribas ve ark. (2015) akış süresi kriterli blokeli akış tipi çizelgeleme problemi için yüksek performanslı bir ayrık YAKA'yı önermişlerdir. Önerilen algoritmayı geliştirmek için, gıda kaynağı aşaması için dört strateji ve kalan üç fazın her biri için iki strateji (işçi, gözlemci ve kâşif arılar) olarak değerlendirmişlerdir. Algoritmanın en iyi konfigürasyonunu bulmak için deney tasarımı kullanmışlardır.

Zhang ve Gu (2015) maksimum tamamlanma süresini en aza indirmek amacıyla ara tamponlu akış tipi çizelgeleme problemi için ayrık YAKA önermişlerdir. Ekleme ve yer değiştirme operatörlerinin kombinasyonu, işçi arı safhasında komşu bireylerin üretilmesi için uygulanmıştır. Turnuva seçimi, yerel optimuma düşmekten kaçınmak için kabul edilmişken, optimize edilmiş uç operatör, yerel arama algoritmasını geliştirmek için komşu çözümü araştırmak üzere gözlemci arı aşamasına geçiş yapılmıştır. Boyutu 2 olan turnuva seçimi tekrar uygulanmış ve daha iyi bir çözüm, TAA kullanılarak imhası ve inşası yapılmış ve ardından sonuç en kötüsünün yerini almıştır.

Parti akışı, çok aşamalı bir üretim sistemindeki işlemlerin üst üste binmesini sağlamak ve böylece üretim süresini kısaltmak için çok aşamalı işlemi birkaç alt noktaya (transfer partileri) bölmek için kullanılan bir tekniktir. Burada amaç, değişken boyuttaki alt noktalara sahip bir akış tipinde n işin, m makine parti akış probleminin üretim süresini, toplam akış süresini en aza indirmek ve ayrıca optimum alt parti boyutunu belirlemektir.

Vijaychakaravarthy ve ark. (2014) çalışmalarında Geliştirilmiş Koyun Sürüsü Kalıtım Kalemi Algoritmasını ve YAKA'yı problemi çözmek için kullanmışlardır. Önerilen algoritmalarla elde edilen sonuçları, diğer geleneksel sezgisellerin performansları ile de karşılaştırmışlardır.

Tasgetiren ve ark. (2013) boşluksuz permütasyon akış tipi çizelgeleme problemini toplam gecikme kriteri ile çözmek için ayrık YAKA önermişlerdir. Bu problem, boşta çalışma süresinin bulunmadığı permütasyon akış tipi çizelgeleme probleminin bir çeşididir. Başka bir deyişle, bir makinede ilk işin başlama süresinin boşta kalma kısıtlamasını karşılamak için ertelenmesidir.

Cui ve Gu (2013) çalışmalarında vektör temsilini kullanarak, problemin tamamlanma süresini en aza indirmek üzere Geliştirilmiş Ayrık Yapay Arı Kolonisi (GAYAK) algoritmasını önermişlerdir. GAYAK algoritmasında, işçi ve gözlemci arılara yeni çözümler üretmek için yeni bir diferansiyel evrim ve değiştirilmiş değişken komşuluk araması incelemiştir. Kâşif arılar, çözümler elde etmek için yıkım ve yapım prosedürleri kullanılmıştır. Simülasyon sonuçları, önerilen GAYAK algoritmasının, aynı kıyaslama durumundaki son teknoloji algoritmalara kıyasla oldukça etkili ve verimli olduğunu göstermişlerdir.

Liu ve Liu (2013) permütasyon akış tipi çizelgeleme problemlerinde tamamlanma süresinin minimizasyonu için, HYAK algoritmasını önermişlerdir. HYAK'de, problemin her çözümü bir gıda kaynağı olarak adlandırılmış ve ayrı bir iş permütasyonu ile temsil edilmiştir. Belirli kalite ve çeşitliliğe sahip ilk popülasyon, NEH sezgiseline dayalı Açgözlü Rastgele Uyarlanmış Araştırma Prosedüründen (ARUAP) elde edilmiştir. Ekleme, yer değiştirme, yeniden birleştirme ve ARUAP gibi ayrık operatörler ve algoritma işçi, gözlemci ve kâşif arılar için yeni bir çözüm üretmek üzere uygulamışlardır.

Deng ve ark. (2012) toplam akış süresi kriterli blokajlı akış tipi çizelgeleme probleminin çözülmesi için ayrık YAKA önermişlerdir. İlk olarak, algoritmadaki çözüm, iş permütasyonu olarak temsil edilmiştir. İkincisi, NEH sezgiselini temel alan bir başlangıç şeması ve yerel arama, ilk popülasyonun hem kalitesini hem de çeşitliliğini sağlamak için tasarlanmıştır. Üçüncüsü, TAA fikrine dayanarak, işçi, gözlemci ve kâşif arılar için yeni tasarlanmış bazı programlar sunmuşlardır. Önerilen algoritmanın performansını, iyi bilinen Taillard test problemleri üzerinde test etmişlerdir.

Li ve Yin (2012) karma ayrık arı kolonisi algoritması olarak adlandırılan permütasyon akış tipi çizelgeleme problemi (PATÇP) için yerel olarak en iyisine ulaşmak, kolay olan tek mutasyon planının kusurlarını telafi etmek için kompozit mutasyon stratejileri içeren ayrık yapay arı kolonisi önermişlerdir. Öncelikle, YAKA'nın PATÇP'ye uygun hale getirilmesi için, her bir iş permütasyon bir gıda kaynağı olarak kabul edilmiş ve farklı arılarla yeni bir komşu gıda kaynağı üretmek için farklı işlemler uygulanmıştır. NEH, popülasyonu belirli bir kalite ve çeşitlilikle başlatmak için rastgele başlatma ile birleştirilmiştir. Kompozit mutasyon stratejileri, PATÇP'yi çözmek için önerilmiştir.

Han ve ark. (2012) toplam akış süresini en aza indirmek amacıyla blokeli akış tipi çizelgeleme problemini çözmek için üç etkili hibrit ayrık yapay arı koloni algoritmaları önermişlerdir. Bu üç algoritma, gıda kaynaklarını temsil etmek için ayrık iş permütasyonları kullanır ve sırasıyla kullanılan işçi, gözlemci ve kâşif arılar için yeni gıda kaynakları üretmek üzere ayrık operatörler uygulamışlardır.

Pan ve ark. (2011) parti akışlı akış tipi çizelgeleme problemini, hem uygun hem de uygun olmayan durumlar altında toplam ağırlıklı erken ve geciktirme ceza ölçütüyle çözmek için ayrık YAKA önermişlerdir. Orijinal YAKA'dan farklı olarak, önerilen ayrık YAKA, ayrı bir iş olarak bir gıda kaynağını temsil etmektedir ve işçi, gözlemci ve kâşif arılar yeni komşu gıda kaynakları üretmek için ayrık operatörlerini uygulamıştır.

Tasgetiren ve ark. (2011) en küçük toplam akış zamanını veren permütasyonu bulmak için çeşitli yinelenmiş açgözlü algoritmalarla hibritleştirilmiş ayrı bir yapay arı kolonisi algoritması önermişlerdir. TAA, yerleştirme ve yer değiştirme komşu yapılarına dayanan yerel arama prosedürlerle oluşturulmuştur. Önerilen algoritmaların performansı, Taillard'ın test problemleriyle test etmişlerdir.

Han ve ark. (2011a) toplam akış süresini en aza indirmek için blokajlı akış tipi çizelgeleme problemini ayrık YAKA ile ele almışlardır. Ayrık YAKA'da, gıda kaynaklarını temsil etmek için ayrık iş permütasyonları kullanılmış ve işçi, gözlemci ve kâşif arılar için yeni gıda kaynakları üretmek için ayrık operatörler uygulanmıştır.

Han ve ark. (2011b) tamamlanma süresini minimuma indirmek amacıyla blokajlı akış tipi çizelgeleme problemini çözmek için geliştirilmiş YAKA önermişlerdir. Önerilen geliştirilmiş YAKA, çözümleri temsil etmek için ayrık iş permütasyonlarını kullanmıştır ve işçi, gözlemci arılar için yeni çözümler üretmek üzere ekleme ve yer değiştirme operatörlerini uygulamıştır. Diferansiyel evrim algoritması kâşif arılar için çözümler elde etmekte kullanılmıştır. Probleme özgü sezgilere dayanan bir başlangıç

şeması, belirli bir kalite ve çeşitlilik seviyesine sahip bir başlangıç popülasyonu oluşturmak için sunulmuştur. Geliştirilmiş YAKA, Taillard'ın iyi bilinen akış tipi test problemlerine dayanan mevcut hibrit ayrık diferansiyel evrim ve ayrık YAKA ile karşılaştırılmıştır.

Karaboga ve Basturk (2007), çok değişkenli fonksiyonları optimize etmek için YAKA kullanmış ve GA, PSA ve Partikül Sürü Evrim Algoritması tarafından üretilen sonuçlar ile karşılaştırmışlardır. Sonuçlar, YAKA'nın diğer algoritmaları geride bıraktığını göstermiştir.

DPATÇP, bu zamana kadar birçok farklı algoritma ile çözümleri yapılmıştır. Arı kolonisi algoritmaları da akış tipi çizelgeleme problemlerinde yaygın olarak kullanılmış hatta daha iyi çözümler bulmak için bu algoritmalarda farklı yöntemler uygulanmıştır. Bu nedenle akış tipi çizelgeleme probleminin bir türü olan DPATÇP'nin çözümünde YAKA kullanılmıştır. Taillard test problemleri kullanılarak YAKA'nın performansı gösterilmiş ve kıyaslanan algoritmaları geride bırakmıştır.

3. MATERYAL VE METOT

Bu bölümde problem ve problemin çözümünde kullanılacak metasezgisel hakkında bilgi verilecektir.

3.1. Akış Tipi Çizelgeleme Problemleri

Akış tipi çizelgeleme, tüm işlerin tüm operasyonlarını gerçekleştirmek için sıkı bir sıraya sahip olduğu özel bir çizelgeleme problemidir. Akış tipi çizelgeleme problemleri, her bir işin bir dizi makine üzerinde veya belirli işlem sürecine uygun diğer kaynaklar üzerinde işlenmesi için uygun bir sıralamayı bulmayı amaçlayan problemlerdir (Kumar S., 2014).

Akış tipi çizelgeleme problemlerindeki (Wang ve ark., 2013) varsayımlar aşağıdaki gibidir. Bunlar:

- Tüm işler birbirinden bağımsızdır ve “0” zamanında işlenebilir.
- Her makine aynı anda bir işi işleyebilir ve her iş aynı anda bir makinada işlenebilir.
- Yarıda kesmelere izin verilmez, diğer bir deyişle, her işlem başlatıldıktan sonra kesintisiz tamamlanmalıdır.
- Makinelerin hazırlık süreleri ve işlemler arasındaki taşıma süreleri önemsizdir.
- Tüm işlerin belirli bir işlem sırası vardır (öncelik kısıtı).

Bu çalışmada kriter olarak maksimum tamamlama zamanı düşünülmektedir.

3.1.1. Dağıtık Permütasyon Akış Tipi Çizelgeleme Problemi

DPATÇP, akış tipi çizelgeleme problemlerinin yeni bir türüdür. Bu problem üzerinde yapılan ilk çalışmalar Naderi ve Ruiz (2010) tarafından yapılmıştır. DPATÇP'nin varsayımları şu şekilde tanımlanabilir:

DPATÇP'de N kümesindeki n adet iş, M kümesindeki m adet makinede işlem görmektedir. j işinin ($j \in N$) i makinesindeki ($i \in M$) işlem süresi $P(i, j)$ kadardır. n adet iş $F = \{f_1, f_2, \dots, f_k\}$ adet fabrikada işlem görmektedir ($k \in F$). Her k fabrika aynı

makine kümesini içermektedir $M = \{M_1, M_2, \dots, M_n\}$. Belli bir fabrikaya bir iş atandığında, başka bir fabrikaya aktarılamaz ve tüm operasyonlar yalnızca o fabrikada yapılabilir.

$\pi^k = [\pi^k(1), \pi^k(2), \dots, \pi^k(n_k)]$ denklemi, k fabrikasındaki iş kümesini ifade eder ve denklemdeki n_k , k fabrikasına atanan toplam iş sayısını gösterir. $C_{i,j}$ j işinin i makinesindeki tamamlanma süresini gösterir. Maksimum tamamlanma süresinin hesaplanması Denklem 1-5 ile sağlanır (Naderi ve Ruiz, 2010).

$$C_{\pi^k(1),1} = P_{\pi^k(1),1}, k = 1, 2, \dots, F \quad (1)$$

$$C_{\pi^k(j),1} = C_{\pi^k(j-1),1} + P_{\pi^k(j),1}, k = 1, 2, \dots, F; j = 2, 3, \dots, n_k \quad (2)$$

$$C_{\pi^k(1),i} = C_{\pi^k(1),i-1} + P_{\pi^k(1),i}, k = 1, 2, \dots, F; i = 2, 3, \dots, m \quad (3)$$

$$C_{\pi^k(j),i} = \max\{C_{\pi^k(j-1),i}, C_{\pi^k(j),i-1}\} + P_{\pi^k(j),i} \quad (4)$$

$$k = 1, 2, \dots, F; j = 2, 3, \dots, n_k; i = 2, 3, \dots, m$$

$$C_{max} = \max C_{n_k, m}, k = 1, 2, \dots, F \quad (5)$$

DPATÇP, belirli bir zamanlama amaç fonksiyonunu en aza indirmek için hem fabrikalara, hem de tüm fabrikalardaki işlerin atamalarını belirlemektedir. Bu problemde klasik akış tipi çizelgeleme problemlerindeki varsayımlar benimsenmiştir.

3.2. Yapay Arı Koloni Algoritması

Doğadaki bal arılarının besin arama davranışları model alınarak yapılan bu algoritma Karaboga ve Basturk (2007) tarafından geliştirilmiştir. Bu algorithmada, besin arayan arılar 3 gruba ayrılırlar. Bunlar: işçi arılar, gözlemci arılar ve kâşif arılardır. Koloninin yarısını işçi arılar diğer yarısını da gözlemci arılar oluşturur. İşçi arılar besin kaynağı bulmaya çalışırlar, buldukları besin kaynağını hafızada tutarlar. Gözlemci arılar işçi arılara göre daha iyi besin kaynağı bulma eğilimindedirler. İşçi arıların birkaçı kâşif arılara dönüşerek yeni besin kaynakları bulmaya çalışırlar. Durdurma kriteri sağlanıncaya kadar döngü devam eder. Diğer popülasyon tabanlı algoritmalara benzer şekilde, YAKA tekrar eden bir süreçtir.

3.2.1. Başlangıç Parametreleri

Temel YAKA'nın üç adet parametresi bulunmaktadır (Tosun, 2012). Bunlar:

- İşçi ve gözlemci arıların toplamına eşit olan besin kaynakların sayısı (BS),
- Besin kaynağının terk edilmesi için limit değeri ve
- İterasyon sayısıdır.

3.2.2. Besin Kaynaklarının Oluşturulması

$X_{i,j} = \{X_{i,1}, X_{i,2}, \dots, X_{i,n}\}$ Popülasyondaki i . besin kaynaklarını göstermektedir.

Besin kaynaklarının oluşturulmasında Denklem 6'dan yararlanılır (Pan ve ark., 2011).

$$X_{i,j} = X_j^{min} + r(X_j^{max} - X_j^{min}) \quad (6)$$

Burada i besin kaynağı sayısı ($i = 1, 2, \dots, BS$), j parametre sayısını ($j = 1, 2, \dots, n$) ifade etmektedir. r ise $[0,1]$ aralığında düzgün bir rassal sayıdır. X_j^{max} ve X_j^{min} j parametresinin alt ve üst değerlerini belirtir.

3.2.3. İşçi Arı Aşaması

İşçi arılar besin kaynaklarına rastgele olarak gönderilirler. Her bir besin kaynağında yalnız bir işçi arı bulunabilmektedir. İşçi arılar, besin kaynağındaki nektar miktarını yani çözüm değerini hesaplayarak hafızaya alırlar. Bu aşamada, her işçi arı, şu anki $X_{i,j}$ pozisyonunun yakınında, Denklem 7 kullanarak yeni bir besin kaynağı üretir (Li ve Yin, 2012).

$$V_{i,j} = X_{i,j} + \varphi_{i,j}(X_{i,j} - X_{k,j}) \quad (7)$$

Buradaki k , rasgele seçilen indekstir $k \neq i$ ve $j \in \{1, 2, \dots, n\}$ ve $\varphi_{i,j} [-1,1]$ aralığında düzgün dağılımlı rastgele bir sayıdır. Bulunan yeni besin kaynağı $V_{i,j}$ $X_{i,j}$ ile

kıyaslanır. Eğer $V_{i,j} X_{i,j}$ 'den daha iyi veya eşit ise, $V_{i,j} X_{i,j}$ 'nin yerini alacaktır. Aksi takdirde $X_{i,j}$ korunacaktır.

3.2.4. Gözlemci Arı Aşaması

Gözlemci arılar, işçi arıların buldukları besin kaynaklarının kalitesini değerlendirerek bir $X_{i,j}$ seçerler. Bu seçim, aşağıdaki Denklem 8 ile hesaplanan olasılık değerine bağlı olarak yapılır.

$$p_i = \frac{u_i}{\sum_{i=1}^{BS} u_i} \quad (8)$$

Buradaki u_i , i . besin kaynağının uygunluk değerini ifade etmektedir. p_i ise, i besin kaynağının seçim olasılığını belirtir. p_i oranı yüksek olan besin kaynaklarının seçilme olasılığı yüksektir (Pan ve ark., 2011).

3.2.5. Kâşif Arı Aşaması

Bir besin kaynağı $X_{i,j}$ limit değeri boyunca daha da iyileştirilemezse, yeni besin kaynakları bulmak için işçi arı kâşif arıya dönüşür. Kâşif arılar rastgele yeni besin kaynaklarını Denklem 9 ile bulurlar (Zhang ve Gu, 2015).

$$X_{i,j} = X_j^{min} + r(X_j^{max} - X_j^{min}) \quad (9)$$

Yeni besin kaynakları bulunduktan sonra, YAKA tekrarlanır. Bu tekrarlar, iterasyon sayısı tamamlanuncaya kadar yapılır.

3.2.6. Temel YAKA Adımları

Temel YAKA adımları şöyle özetlenebilir (Pan ve ark., 2011).

Adım 1: BS parametresinin (besin kaynakları sayısı) ve popülasyonun başlatılması

Adım 2: Besin kaynaklarına işçi arıların gönderilmesi,

Adım 3: Besin kaynaklarındaki nektar miktarına göre gözlemci arıların besin kaynaklarına gönderilmesi,

Adım 4: Yeni besin kaynaklarının bulunması için kâşif arıların gönderilmesi,

Adım 5: Bulunan en iyi besin kaynağının hafızada tutulması ve

Adım 6: İterasyon kriteri sağlanıncaya kadar algoritmanın devam ettirilmesi.

3.3. NEH Sezgiseli

Akış tipi çizelgeleme problemlerinin çözümünde veya başlangıç çözümlerinin iyileştirilmesinde NEH sezgiseli oldukça sık kullanılmaktadır. NEH sezgiseli Nawaz, Ensore ve Ham tarafından ortaya çıkarılmıştır. Bu tip problemlerin çözümünde bu sezgisel başarılı sonuçlar ortaya koymaktadır. NEH'in adımları (Kalczyński ve Kamburowski, 2007);

Adım 1: Her bir işin, toplam iş süresi hesaplanır.

Adım 2: İşler toplam iş süresi baz alınarak azalan sırada sıralanır.

Adım 3: Sıradaki ilk iki iş seçilir ve tüm alternatif sıraların toplam tamamlanma süreleri hesaplanır. Tamamlanma süresi küçük olan sıra seçilir.

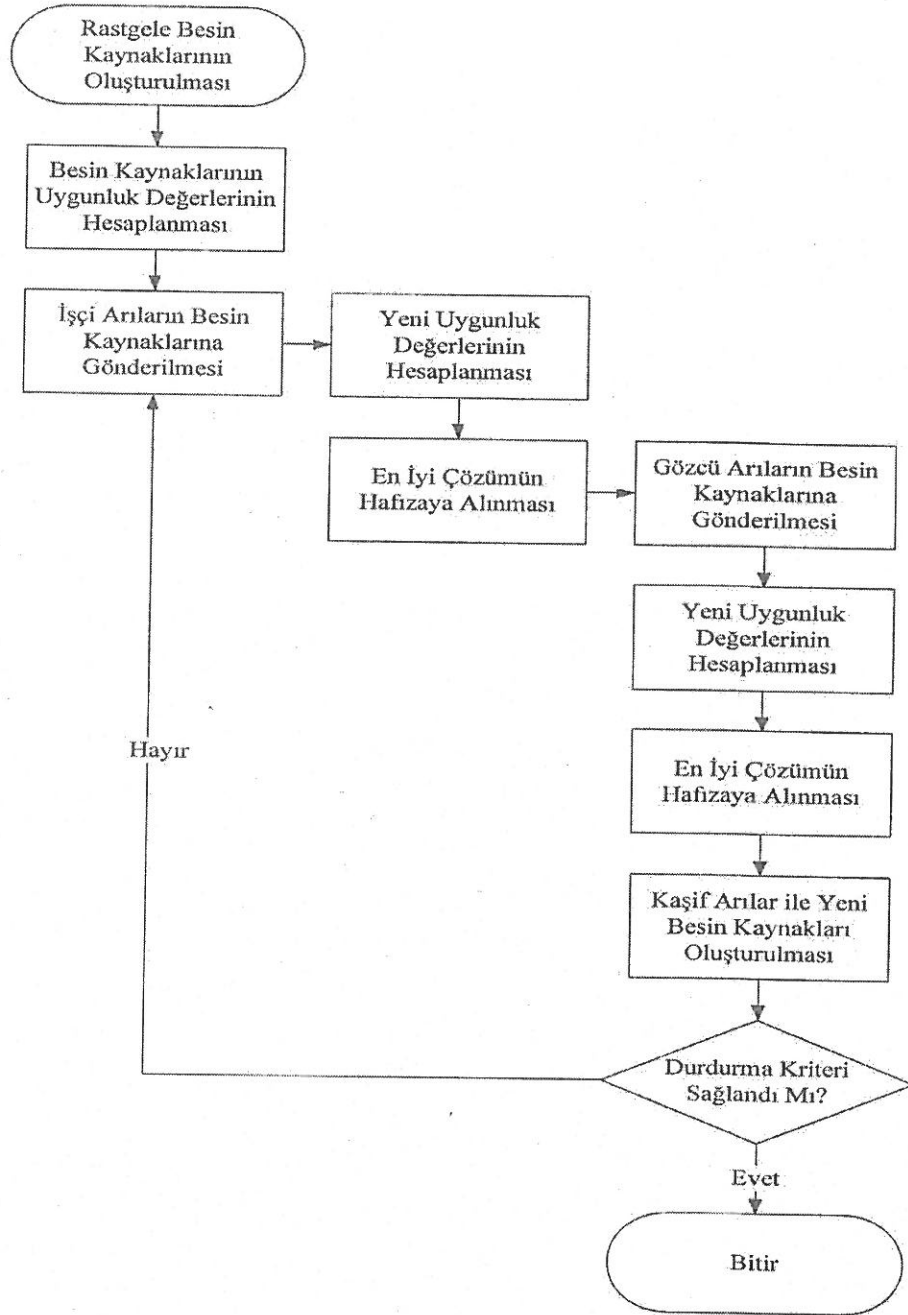
Adım 4: Adım 2'de belirlenen sıradaki diğer iş seçilir ve bu iş mevcut sıradaki tüm pozisyonlara koyularak toplam tamamlanma süreleri hesaplanır. Tamamlanma süresi küçük olan sıra seçilir.

Adım 5: Adım 2'de belirlenen sıradaki tüm işler bitinceye kadar Adım 4 tekrarlanır.

3.4. Algoritmanın Probleme Uyarlanması

Şekil 1'de algoritmanın akış diyagramı verilmektedir. Algoritmanın başlangıcında rastgele besin kaynakları oluşturulur. BS, başlangıç parametrelerinde belirlediğimiz işçi ve gözlemci arı sayılarının toplamı kadar olacaktır. Algoritma belirlediğimiz iterasyon sayısı kadar tekrar çalışacaktır. Küçük boyutlu test problemlerinde besin kaynakları rastgele oluşturulurken, büyük boyutlu test problemlerinde çözüm kalitesini iyileştirmek için, besin kaynaklarının biri NEH

sezgiseli ile oluşturulmakta $BS - 1$ adedi ise, rastgele oluşturulmaktadır. Oluşturulan besin kaynaklarının nektar miktarları hesaplanır.



Şekil 1: YAKA Akış Diyagramı

İşçi arı evresine geçilir ve işçi arılar rastgele besin kaynaklarına gönderilir. İşçi arılar komşu besin kaynakları oluştururlar ve komşu besin kaynaklarının nektar miktarları hesaplanır.

Komşu besin kaynakları, besin kaynaklarından seçilen rastgele 2 işin birbirleri ile yer değiştirmesiyle elde edilir. Komşu besin kaynakları, limit değerine ulaşıncaya kadar bulunmaya devam ettirilir. Komşu besin kaynaklarının nektar miktarları hesaplanır. Yeni besin kaynağı, mevcut olan besin kaynağından daha iyi ise hafızaya alınır, kötü ise mevcut olan korunur.

İşçi arı evresinden sonra gözlemci arı evresine geçer ve gözlemci arılar turnuva metodu ile en iyi olan besin kaynaklarına gönderilirler. Gözlemci arılar komşu besin kaynakları üretirler ve komşu besin kaynaklarının nektar miktarları hesaplanır. Yeni besin kaynağı, mevcut olan besin kaynağından daha iyi ise hafızaya alınır, kötü ise mevcut olan korunur.

En son evre olan kâşif arı evresinde işçi arılar kâşif arıya dönüşür. Bir önceki evreden kalan besin kaynaklarını kullanarak yeni besin kaynakları oluştururlar. İterasyon sayısı kadar bu döngü tekrarlanır.

3.4.1. Kullanılan Parametreler

Bu çalışmada tüm parametreler deneysel olarak belirlenmiştir. Parametrelerdeki sayıların artırılması çözüm kalitesinin iyileşmesinde katkıda bulunmamış aksine CPU sürelerinin artmasına neden olmuştur. Bu nedenle parametreler küçük tutularak kısa sürelerde daha çok çözüm elde edilmiştir. Taillard küçük ve büyük test problemlerinin çözümünde işçi ve gözlemci sayıları eşit olarak alınmıştır.

- İşçi arı sayısı: 5
- Gözlemci arı sayısı: 5
- Limit sayısı: 2
- İterasyon sayısı: 1000

3.5. YAKA Örneği

Algoritmanın çalışma adımları aşağıdaki bir örnekle açıklanmıştır.

Fabrika sayısı (F): 2

İş sayısı (n): 4

Makine sayısı (m): 2

İşçi arı sayısı: 2

Gözlemci arı sayısı: 2

Limit: 1

İterasyon: 1

Çizelge 1: İşlerin makinelerdeki işlem süreleri

İş / Makine	1. Makine	2. Makine
1. İş	2	4
2. İş	5	6
3. İş	10	12
4. İş	8	9

İşlerin makinelerdeki işlem süreleri Çizelge 1'de verilmiştir. İşçi ve gözlemci arı sayısı toplamı kadar rastgele iş sıraları oluşturulmuştur ve Çizelge 2'deki gibidir. Çizelge 1'deki işlem süreleri kullanılarak Çizelge 2'de gösterilen iş sıralarının C_{max} değerleri hesaplanmıştır.

Çizelge 2: Rastgele İş Sıraları ve C_{max} Değerleri

Fabrika	İş Sırası	C_{max}
1	1-2	13
2	3-4	31
<i>Genel C_{max}</i>		31

Fabrika	İş Sırası	C_{max}
1	2-3	27
2	4-1	21
<i>Genel C_{max}</i>		27

Fabrika	İş Sırası	C_{max}
1	1-4	19
2	2-3	27
<i>Genel C_{max}</i>		27

Fabrika	İş Sırası	C_{max}
1	1-3	24
2	2-4	22
<i>Genel C_{max}</i>		24

İşçi arı evresinde, işçi arı sayısı kadar rastgele iş sıraları seçilir (Çizelge 3 ve 4). Seçilen iş sıralarında seçilen herhangi bir işin yeri değiştirilerek yeni iş sırası oluşturulur.

Çizelge 3: İşçi Arı Evresi ve C_{max} Değerleri (1.sıra)

Fabrika	İş Sırası	C_{max}
1	1-2	13
2	3-4	31
<i>Genel C_{max}</i>		31

Fabrika	İş Sırası	C_{max}
1	2-1	15
2	4-3	30
<i>Genel C_{max}</i>		30

Fabrika	İş Sırası	C_{max}
1	2-1	15
2	3-4	31
<i>Genel C_{max}</i>		31

Çizelge 3'te C_{max} değeri en iyi olan iş sırası seçilir. 2. Çözüm ilk çözümden daha iyi olduğu limit değeri sabit kalır ve limit değerine ulaşıncaya kadar çözüm iyileştirilir.

Çizelge 4: İşçi Arı Evresi ve C_{max} Değerleri (2.sıra)

Fabrika	İş Sırası	C_{max}
1	2-3	27
2	4-1	21
<i>Genel C_{max}</i>		27

Fabrika	İş Sırası	C_{max}
1	3-2	28
2	1-4	19
<i>Genel C_{max}</i>		28

Limit değerine ulaşıldığında en iyi çözüm hafızaya alınır diğer çözümler terkedilir. Çizelge 4'te C_{max} değeri en iyi olan seçilir. 2. Çözüm ilk çözümden daha kötü olduğu için limit değeri bir artırılır ve 2. Çözüm değeri terkedilir. Çizelge 5'teki gibi güncel iş sıraları oluşturulur. Bir sonraki evreye geçilir.

Çizelge 5: Güncellenmiş iş sıraları

Fabrika	İş Sırası	C_{max}
1	2-1	15
2	4-3	30
<i>Genel C_{max}</i>		30

Fabrika	İş Sırası	C_{max}
1	2-3	27
2	4-1	21
<i>Genel C_{max}</i>		27


Fabrika	İş Sırası	C_{max}
1	1-4	19
2	2-3	27
<i>Genel C_{max}</i>		27

Fabrika	İş Sırası	C_{max}
1	1-3	24
2	2-4	22
<i>Genel C_{max}</i>		24

Gözlemci arı evresinde, işçi arı evresinden farklı olarak gözlemci arılar iş sıralarını Turnuva metodu ile en iyi C_{max} değerine sahip iş sıralarını seçerler (Çizelge 6-7).

Çizelge 6: Gözlemci Arı Evresi ve C_{max} Değerleri (3. sıra)


Fabrika	İş Sırası	C_{max}
1	1-4	19
2	2-3	27
<i>Genel C_{max}</i>		27



Fabrika	İş Sırası	C_{max}
1	4-1	21
2	3-2	28
<i>Genel C_{max}</i>		28

Çizelge 7: Gözlemci Arı Evresi ve C_{max} Değerleri (4. sıra)

Fabrika	İş Sırası	C_{max}
1	1-3	24
2	2-4	22
<i>Genel C_{max}</i>		24




Fabrika	İş Sırası	C_{max}
1	3-1	26
2	4-2	23
<i>Genel C_{max}</i>		26

Çizelge 6-7'de, seçilen iş sıralarından komşu iş sıraları üretilir ve C_{max} değeri en iyi olan seçilir. 2. Çözüm ilk çözümden daha kötü olduğu için limit değeri bir artırılır ve 2. Çözüm değeri terkedilir. Yeni iş sıraları eski iş sıralarından daha iyi sonuçlar veremediği için iş sıraları değişmez ve Çizelge 5'teki gibidir. Bir sonraki evreye geçilir.

Kâşif arı evresinde, işçi arılar yerini kâşif arılara bırakır. Kâşif arı sayısı kadar iş sırası rastgele olarak seçilir. Çizelge 8-9'da C_{max} değeri en iyi olan seçilir. 2. Çözüm ilk çözümden daha kötü olduğu için limit değeri bir artırılır ve 2. Çözüm değeri terkedilir.

Çizelge 8: Kâşif Arı Evresi ve C_{max} Değerleri (2. sıra)

Fabrika	İş Sırası	C_{max}
1	2-3	27
2	4-1	21
<i>Genel C_{max}</i>		27



Fabrika	İş Sırası	C_{max}
1	3-2	28
2	4-1	21
<i>Genel C_{max}</i>		28

Çizelge 9: Kâşif Arı Evresi ve C_{max} Değerleri (4. sıra)

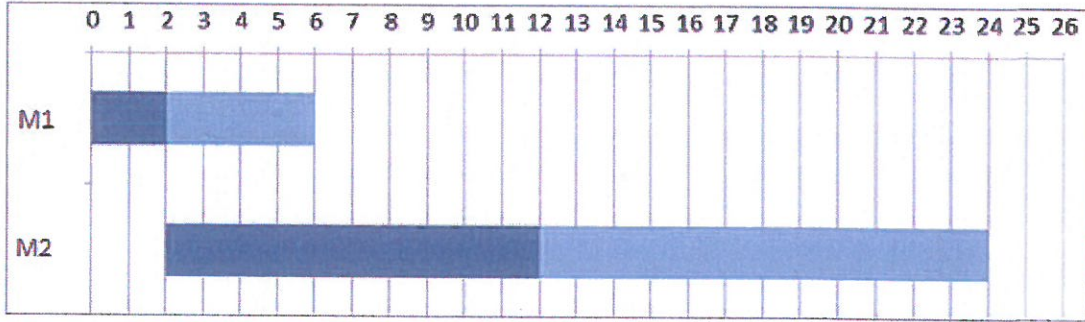
Fabrika	İş Sırası	C_{max}
1	1-3	24
2	4-2	23
Genel C_{max}		24

Fabrika	İş Sırası	C_{max}
1	3-1	26
2	4-2	23
Genel C_{max}		26

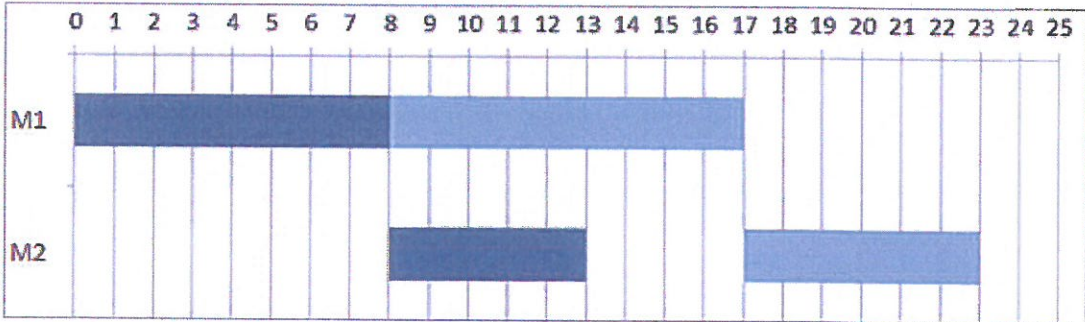
Çizelge 10: En iyi iş sırası

Fabrika	İş Sırası	C_{max}
1	1-3	24
2	4-2	23
Genel C_{max}		24

Yeni iş sıraları eski iş sıralarından daha iyi sonuçlar veremediği için iş sıraları değişmez ve Çizelge 5'teki gibidir. İterasyon sayısı 1 olduğu için algoritma adımları tamamlanmıştır. Kâşif arı evresinde bulunan iş sırası içerisinde en iyi sonuca sahip olan çözüm seçilir (Çizelge 10). Fabrikaların Gantt şemaları Şekil 2 ve Şekil 3'te gösterilmiştir.



Şekil 2: 1. Fabrikanın Gantt Şeması



Şekil 3: 2. Fabrikanın Gantt Şeması

Şekil 2'deki C_{max} değeri 24 ve Şekil 3'teki C_{max} değeri 23'tür. Bu nedenle C_{max} değeri en büyük olana göre Genel C_{max} değeri 24 olarak bulunur.

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu bölümde YAKA'nın performansı, Taillard'ın test problemleri kullanılarak diğer algoritmaların performansları ile kıyaslanacaktır. Problem ve problemin çözümünde kullanılan algoritma, C # ile kodlanmış ve Intel(R) Core(TM) i7 - 2.00 GHz - 6 GB RAM – Windows 10 Pro 64 bit özelliklerine sahip bilgisayarda çalıştırılmıştır.

4.1. Bulgular

Algoritmaların, Taillard test problemlerindeki çözüm performanslarının kıyaslanmasında BSY (RPD) değerleri kullanılmıştır. Önerilen algoritmanın performansı için aşağıdaki eşitlik kullanılacaktır (Naderi ve Ruiz, 2010).

$$BSY (RPD) = \frac{Alg_{\text{çözüm}} - Bei_{\text{çözüm}}}{Bei_{\text{çözüm}}} * 100 \quad (10)$$

Eşitlikte yer alan $Alg_{\text{çözüm}}$ YAKA ile bulduğumuz optimal çözüm değerlerini ifade eder. $Bei_{\text{çözüm}}$ ise, Karışık Tamsayılı Doğrusal Programlama (KTDP) modelleri ile bulunan, bilinen en iyi sonuçları ifade eder. Bu eşitlik kullanılarak “Bağıl Sapma Yüzdesi (BSY)” hesaplanacaktır.

4.1.1. Küçük Boyutlu Test Problemleri

Küçük boyutlu test problemleri 420 adet örnekten oluşmaktadır. Test problemleri: fabrika sayısı $F = \{2,3,4\}$, iş sayısı $n = \{4,6,8,10,12,14,16\}$ ve makine sayısı $m = \{2,3,4,5\}$ olmak üzere 84 kombinasyondan oluşmaktadır. Her kombinasyonun 5 örneği, toplamda 420 örnek oluşmaktadır.

YAKA'nın performansı, 14 adet sezgisel algoritma ile karşılaştırılmıştır. YAKA'nın BSY değerleri Çizelge 11'de verilmiştir. BSY değerleri $F \times n$ olarak gruplandırılmıştır ve her satırda 20 adet veri bulunmaktadır (Naderi ve Ruiz, 2010).

Çizelge 11: Sezgisel Algoritmaların BSY Değerleri (Küçük Boyutlu)

Örnek $F \times n$	SPT1	SPT2	LPT1	LPT2	Johnson1	Johnson2	CDS1	CDS2	Palmer1	Palmer2	NEH1	NEH2	VND(a)	VND(b)	YAKA
2x4	12,95	10,02	19,31	17,80	6,13	3,76	2,69	0,72	7,75	5,22	2,61	0,15	0,00	0,15	0,83
2x6	13,30	10,38	29,53	28,37	10,34	8,08	7,56	4,60	7,57	4,83	4,42	1,44	1,26	1,44	0,26
2x8	17,31	16,16	35,43	33,09	9,75	7,98	12,33	10,33	9,97	8,64	4,43	2,53	2,10	2,52	0,48
2x10	21,03	17,49	36,93	37,12	8,38	7,52	9,72	8,38	10,18	9,25	4,28	3,27	3,22	2,89	0,96
2x12	20,40	17,23	39,36	37,42	10,14	8,29	11,81	10,49	10,77	10,32	6,95	4,13	3,38	3,90	1,97
2x14	17,73	17,34	40,00	38,36	10,27	9,21	6,59	5,24	10,57	9,24	6,05	3,16	2,70	2,82	2,60
2x16	17,11	16,95	42,42	41,63	12,33	10,99	10,53	8,26	10,97	9,00	6,66	3,84	2,92	3,30	3,67
3x4	8,87	5,36	6,42	5,41	5,31	1,99	5,01	2,55	4,82	1,50	0,43	0,43	0,00	0,00	0,00
3x6	21,24	11,93	27,63	25,26	6,91	5,28	8,50	6,99	9,40	3,40	4,04	1,39	0,68	1,39	1,10
3x8	17,71	17,93	26,87	25,14	10,14	8,30	9,17	8,67	11,84	6,98	5,08	2,43	1,86	2,43	1,46
3x10	24,48	16,23	37,98	35,43	12,67	10,07	13,56	10,81	13,97	11,67	8,42	3,79	2,59	3,63	1,90
3x12	25,92	19,55	42,12	40,14	14,90	10,12	14,66	11,00	14,39	11,36	7,66	5,08	3,66	5,08	2,90
3x14	23,44	19,95	41,00	38,56	14,62	14,04	16,45	13,47	17,97	14,05	10,54	4,90	4,48	4,46	4,26
3x16	25,31	22,83	41,71	39,39	14,39	10,41	15,55	11,60	15,68	11,06	8,18	3,98	3,50	3,81	4,31
4x4	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
4x6	13,34	11,31	14,52	12,52	7,65	5,62	4,99	3,39	6,02	2,98	3,10	0,47	0,00	0,00	0,16
4x8	15,21	13,89	16,86	16,44	9,20	5,56	9,23	8,61	10,17	7,04	4,19	0,77	0,77	0,77	2,23
4x10	22,65	16,84	34,47	30,64	13,19	8,93	13,90	9,30	16,21	7,23	7,07	2,30	1,57	2,22	2,36
4x12	25,51	20,54	38,72	34,06	13,61	8,68	18,13	14,49	15,49	12,39	8,58	4,97	4,23	4,68	3,95
4x14	26,61	22,31	42,89	39,79	13,50	8,51	15,47	11,23	16,53	12,25	10,94	4,54	4,25	4,46	5,15
4x16	28,70	24,09	44,01	40,67	19,12	13,96	17,62	13,69	19,34	15,84	9,79	5,59	5,08	5,55	5,76
Ortalama	18,99	15,63	31,34	29,39	10,60	7,97	10,64	8,28	11,41	8,30	5,88	2,82	2,30	2,64	2,20

YAKA diğer sezgisel yöntemlere göre oldukça başarılı sonuçlar vermektedir. Çizelge 10'daki verilere bakıldığında, 2 fabrikalı 6,8,10,12,14 işlerde, 3 fabrikalı 4,8,10,12,14 işlerde ve 4 fabrikalı 4,12 işlerde YAKA, VND(a) algoritmasına göre çok düşük sonuçlar elde etmiştir. 4 fabrika 4 işte diğer algoritmalar ile aynı sonuçları bulmuştur. Geri kalan $F \times n$ örneklerde VND(a) algoritmasının sonuçlara yakın değerler bulmuştur. Ortalamaya bakıldığında YAKA'nın BSY değeri %2,20'dir ve BSY değerlerini büyükten küçüğe sıraladığımızda 1. sırada yer almaktadır.

4.1.2. Büyük Boyutlu Test Problemleri

Büyük boyutlu test problemleri 720 adet örnekten oluşmaktadır. Test problemleri: fabrika sayısı $F = \{2,3,4,5,6,7\}$, iş sayısı $n = \{20,50,100,200,500\}$ ve makine sayısından $m = \{5,10,20\}$ oluşmaktadır.

YAKA'nın performansı, 14 adet sezgisel algoritma ile karşılaştırılmıştır. YAKA'nın BSY değerleri Çizelge 12'de verilmiştir. BSY değerleri F olarak gruplandırılmıştır ve her satırda 120 adet veri bulunmaktadır (Naderi ve Ruiz, 2010).

Çizelge 12: Sezgisel Algoritmaların BSY Değerleri (Büyük Boyutlu)

Örnek F	SPT1	SPT2	LPT1	LPT2	Johnson1	Johnson2	CDS1	CDS2	Palmer1	Palmer2	NEH1	NEH2	VND(a)	VND(b)	YAKA
2	18,71	17,71	33,70	32,36	12,68	11,58	9,29	8,52	10,44	9,34	2,92	1,21	0,10	0,32	0,08
3	19,95	17,58	33,05	31,57	13,40	11,18	10,83	8,92	11,72	9,43	3,42	1,15	0,10	0,35	-0,04
4	20,07	17,23	33,30	30,42	13,48	10,67	11,19	8,54	11,95	8,90	4,21	1,11	0,06	0,46	0,05
5	20,04	16,73	32,89	29,57	13,18	10,24	11,29	8,07	12,24	8,76	4,28	0,92	0,11	0,46	0,10
6	20,32	16,07	32,58	28,55	13,57	9,94	11,50	7,83	12,70	8,45	4,73	0,95	0,11	0,51	0,10
7	21,04	15,42	32,02	27,14	13,61	9,71	11,49	7,31	12,53	8,21	4,86	0,81	0,10	0,45	0,09
Ortalama	20,02	16,79	32,92	29,93	13,32	10,55	10,93	8,20	11,93	8,85	4,07	1,03	0,10	0,43	0,06

Büyük boyutlu test problemlerinde fabrika, makine ve iş sayıları küçük boyutlu test problemlerine göre fazla olduğundan dolayı problemin çözümünün iyileştirilmesinde NEH sezgiseli uygulanmıştır. NEH sezgiseli rastgele besin kaynakları oluşturma evresinde kullanılmıştır.

Çizelge 12’de gösterildiği gibi YAKA diğer sezgisel yöntemlere göre oldukça başarılı sonuçlar vermektedir. Çizelge 12’deki ortalama değerler temel alınarak, BSY değerlerini büyükten küçüğe sıraladığımızda YAKA 1. sırada yer almaktadır ve BSY değeri %0,06’dır.

Tüm fabrikalarda YAKA, diğer algoritmalarından daha iyi sonuçlar elde etmiştir. 3. fabrikada YAKA “-0,04” değerini bulmuştur. Diğer algoritmalar, bu fabrikada eksi değere ulaşamamıştır. Bu durum, KTDP modelleri ile bulunan sonuçlardan daha iyi sonuçlar bulduğunu göstermektedir.

Çizelge 13: Ta001 Örneğinin Sonuçları

Örnek	Fabrika	İş	Makine	Bilinen En İyi Çözüm	YAKA
	F	n	m		
Ta001_3.txt	3	20	5	598	580

3 fabrika 20 iş x 5 makineli bir örnekte en iyi çözüm değeri “598” bulunmuştur (Çizelge 13). YAKA ile bu örneği çözdüğümüzde “580” değeri bulunmuştur. Her bir fabrikada bulunan C_{max} değerleri ve Genel C_{max} değeri Şekil 4’te gösterilmiştir.

İş Sayısı	<input type="text" value="20"/>	İşçi An Sayısı	<input type="text" value="5"/>
Makine Sayısı	<input type="text" value="5"/>	Gözetmeni An Sayısı	<input type="text" value="5"/>
Fabrika Sayısı	<input type="text" value="3"/>		
Hedef Değer:	<input type="text" value="582"/>		
İterasyon Sayısı:	<input type="text" value="1000"/>		
Limit Sayısı:	<input type="text" value="2"/>		

SONUÇ

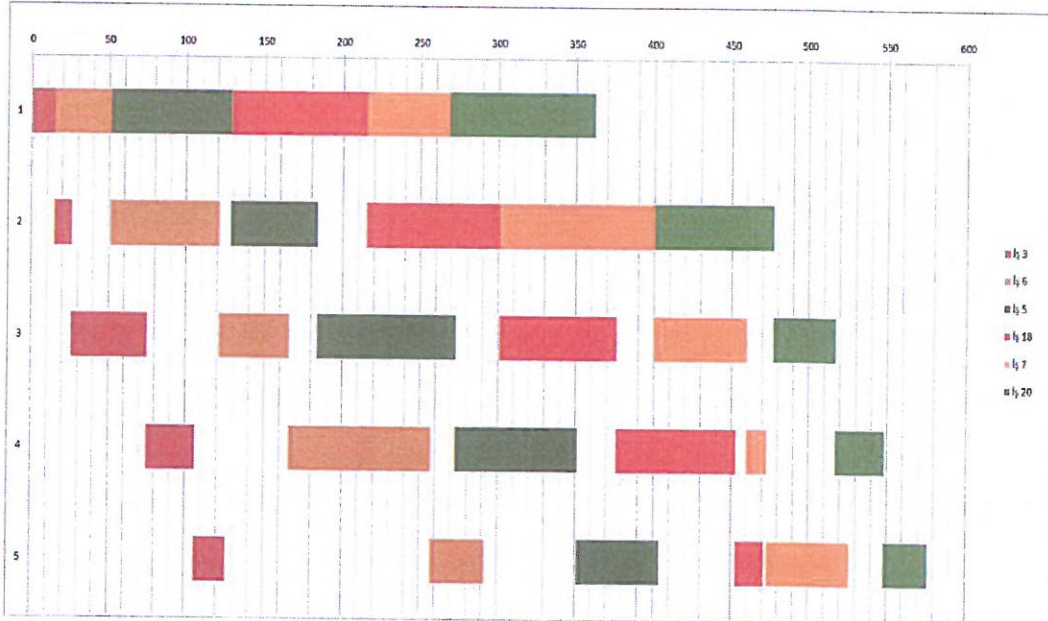
Genel C_{max} = 580
 1. Fabrika Sırası / C_{max} = 3-6-5-18-7-20 / 576
 2. Fabrika Sırası / C_{max} = 9-14-16-11-2-13-12 / 578
 3. Fabrika Sırası / C_{max} = 17-15-1-4-19-8-10 / 580

Şekil 4: Ta001 Örneğinin YAKA ile Çözümünden Elde Edilen Sonuçlar

Şekil 4' e göre Ta001 örneğinin YAKA ile çözümünde fabrikalara atanan iş sıraları C_{max} değeri aşağıdaki gibidir. Şekil 5-6-7'de fabrikaların Gantt şemaları verilmiştir.

- 1. Fabrikadaki iş sırası 3-6-5-18-7-20

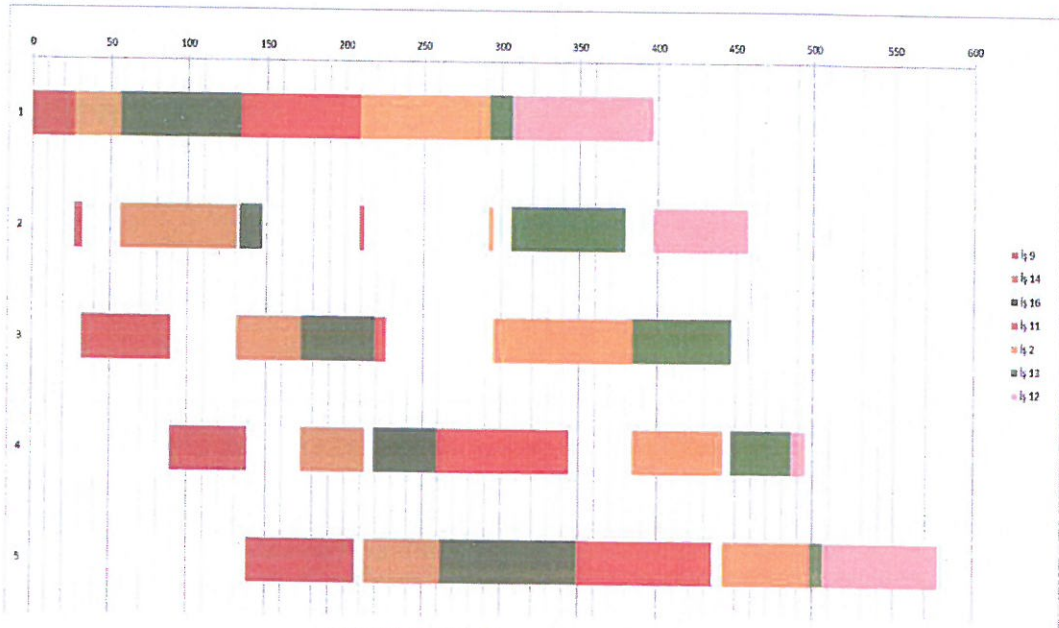
$$C_{max} = 576$$



Şekil 5: 1. Fabrikanın Gantt Şeması

- 2. Fabrikadaki iş sırası 9-14-16-11-2-13-12

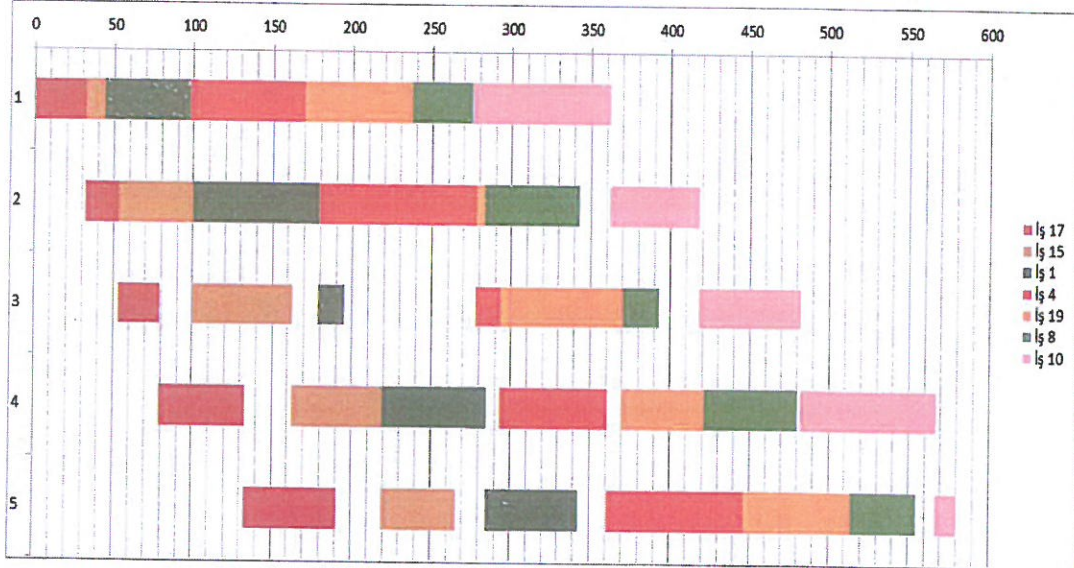
$$C_{max} = 578$$



Şekil 6: 2. Fabrikanın Gantt Şeması

- 3. Fabrikadaki iş sırası 17-15-1-4-19-8-10

$$C_{max} = 580$$

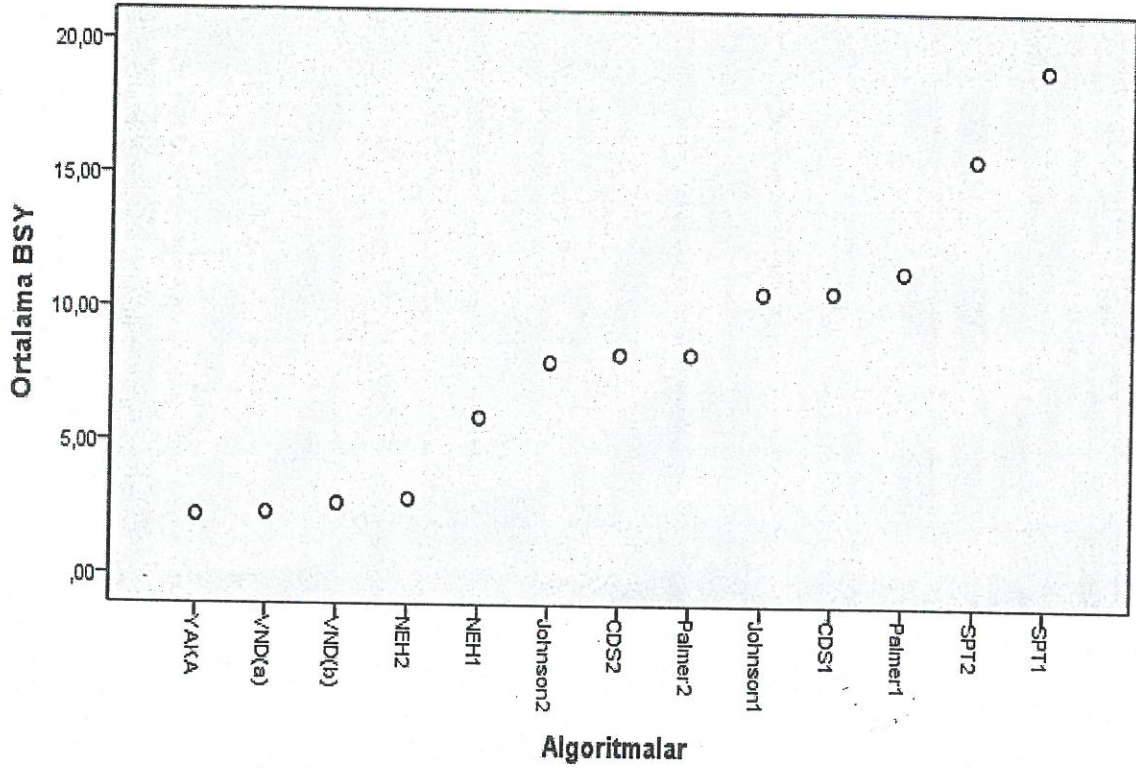


Şekil 7: 3. Fabrikanın Gantt Şeması

Genel C_{max} değeri ise en büyük C_{max} değeri seçilerek "580" olarak belirlenir.

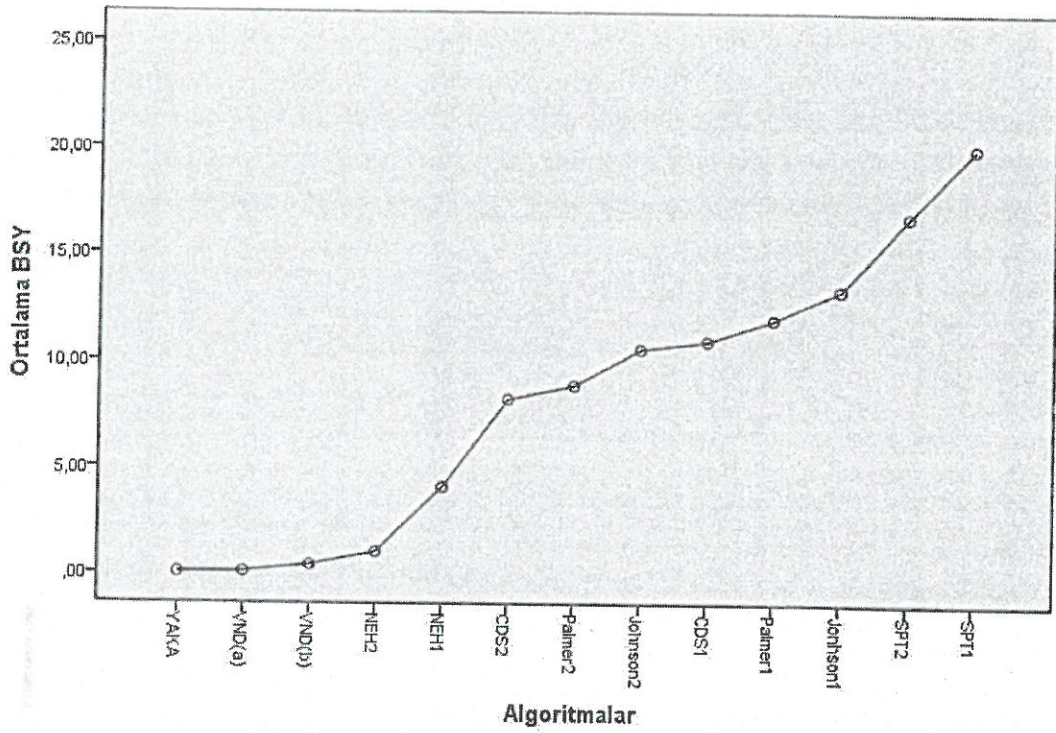
Eşitlik (10) kullanılarak Ta001 örneği YAKA ile çözüldüğünde BSY değeri "-3,01" elde edilmiştir.

Çizelge 11 ve 12'deki BSY değerlerinin arasındaki farkın istatistiksel olarak anlamlı olup olmadığının kontrolü için ANOVA One Way testi kullanılmıştır. $13 \times 420 = 5460$ adet veri kullanılarak Şekil 8 ve $13 \times 720 = 9360$ adet veri kullanılarak Şekil 9 oluşturulmuştur.



Şekil 8: Sezgisel Algoritmaların %99 Güven Aralığındaki BSY Değerleri (Küçük Boyutlu)

LPT1 ve LPT2 algoritmalarının BSY değerleri diğer algoritmalara göre çok yüksek olduğu için grafiklere dâhil edilmemiştir. Burada parametrik ANOVA'nın her üç hipotezi (normallik, eş varyanslılığı ve bağımsızlığı) kontrol edilmektedir. Yanıt değişkeni BSY'dir ve kontrol edilen faktörler n , m ve F 'dir. BSY yanıt değişkeninde istatistiksel olarak anlamlı farklılıklar oluşmaktadır. %99 güven aralığında anlamlı fark aralıklarını göstermektedir.



Şekil 9: Sezgisel Algoritmaların %99 Güven Aralığındaki BSY Değerleri (Büyük Boyutlu)

Burada YAKA'da bulduğumuz sonuçların istatistiksel olarak diğer algoritmaların sonuçları ile kıyasladığımızda Çizelge 11 ve 12'deki sonuçları destekler nitelikte olduğu görülmektedir.

5. SONUÇ

DPATÇP akış tipi çizelgeleme problemlerinin yeni bir çeşididir. Literatürde akış tipi çizelgeleme problemleriyle ilgili birçok çalışma yapılmasına rağmen DPATÇP ilk olarak tarafından ortaya atılmış ve gitgide popüler bir konu haline gelmiştir.

Klasik akış tipi çizelgeleme problemlerinden farklı olarak, işler birden fazla fabrikada her fabrikada aynı makine kümesinde işlem görmektedirler. Diğer varsayımları akış tipi çizelgeleme varsayımları ile aynıdır. Bu durum çizelgelemeyi akış tipine göre oldukça zorlaştırmaktadır. Bu nedenle problemin çözümünde birçok araştırmacı farklı çözüm yöntemlerinden ve algoritmalarından yararlanmışlardır.

Doğadaki bal arılarının nektar arama yeteneğinden ilham alınarak yapılan bu algoritma akış tipi çizelgeleme problemlerinde yaygın olarak kullanılan güçlü bir algoritmadır. Algoritma optimal sonuca yakın sonuçlar bulmada oldukça başarılıdır. Bu nedenle çalışmamızda DPATÇP çözümü için YAKA seçilmiştir. Amaç fonksiyonumuz tamamlanma süresinin minimizasyonudur ve DPATÇP çözümü için YAKA kullanılmıştır. YAKA'nın problem çözümündeki etkinliği Taillard'ın test problemleri ile kıyaslanmıştır. Karşılaştırılan diğer sezgisel yöntemlere göre en iyi çözüm üreten algoritma YAKA olmuştur. İleride yapılacak olan çalışmalarda işlem süreleri bulanık sayılardan seçilerek gerçeğe daha yakın sonuçlar elde edilebilir.

KAYNAKLAR

- Bargaoui, H., Belkahla Driss, O. ve Ghédira, K., 2017, A novel chemical reaction optimization for the distributed permutation flowshop scheduling problem with makespan criterion, *Computers & Industrial Engineering*, 111, 239-250.
- Cui, Z. ve Gu, X., 2013, An Improved Discrete Artificial Bee Colony Algorithm for Hybrid Flow Shop Problems, *Intelligent Computing for Sustainable Energy and Environment*, Berlin, Heidelberg, 294-302.
- Deng, G., Xu, Z. ve Gu, X., 2012, A Discrete Artificial Bee Colony Algorithm for Minimizing the Total Flow Time in the Blocking Flow Shop Scheduling, *Chinese Journal of Chemical Engineering*, 20 (6), 1067-1073.
- Deng, J., Wang, L., Shen, J. ve Zheng, X., 2016a, An Improved Harmony Search Algorithm for the Distributed Two Machine Flow-Shop Scheduling Problem, Berlin, Heidelberg, 97-108.
- Deng, J., Wang, L., Wu, C., Wang, J. ve Zheng, X., 2016b, A Competitive Memetic Algorithm for Carbon-Efficient Scheduling of Distributed Flow-Shop, In: *Intelligent Computing Theories and Application*, Eds, p. 476-488.
- Deng, J. ve Wang, L., 2017, A competitive memetic algorithm for multi-objective distributed permutation flow shop scheduling problem, *Swarm and Evolutionary Computation*, 32, 121-131.
- Duan, W., Li, Z., Yang, Y., Liu, B. ve Wang, K., 2017, EDA based probabilistic Memetic Algorithm for distributed blocking permutation flowshop scheduling with sequence dependent setup time. 2017 IEEE Congress on Evolutionary Computation (CEC), IEEE: 992-999.
- Engin, O. ve Fiğlalı, A., 2002, Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma Yardımı İle Çözümünde Uygun Çaprazlama Operatörünün Belirlenmesi, *Doğuş Üniversitesi Dergisi*, 6, 27 – 35.
- Fernandez-Viagas, V., Perez-Gonzalez, P. ve Framinan, J. M., 2018, The distributed permutation flow shop to minimise the total flowtime, *Computers & Industrial Engineering*, 118, 464-477.
- Gao, J. ve Chen, R., 2011a, A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem, *International Journal of Computational Intelligence Systems*, 4 (4), 497-508.
- Gao, J. ve Chen, R., 2011b, An NEH-based heuristic algorithm for distributed permutation flowshop scheduling problems, *Scientific Research and Essays*, 6, 3094-3100.
- Gao, J., Chen, R. ve Deng, W., 2013, An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem, *International Journal of Production Research*, 51 (3), 641-651.
- Gong, D., Han, Y. ve Sun, J., 2018, A novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems, *Knowledge-Based Systems*, 148, 115-130.
- Han, Y.-Y., Duan, J.-H., Yang, Y.-J., Zhang, M. ve Yun, B., 2011a, Minimizing the Total Flowtime Flowshop with Blocking Using a Discrete Artificial Bee Colony, *7th international conference, ICIC 2011*, 0302-9743, p. 91-97.
- Han, Y.-Y., Pan, Q.-K., Li, J.-Q. ve Sang, H.-y., 2011b, An improved artificial bee colony algorithm for the blocking flowshop scheduling problem, *The International Journal of Advanced Manufacturing Technology*, 60 (9-12), 1149-1159.

- Han, Y.-Y., Liang, J. J., Pan, Q.-K., Li, J.-Q., Sang, H.-Y. ve Cao, N. N., 2012, Effective hybrid discrete artificial bee colony algorithms for the total flowtime minimization in the blocking flowshop problem, *The International Journal of Advanced Manufacturing Technology*, 67 (1-4), 397-414.
- Hatami, S., Ruiz, R. ve Andrés-Romano, C., 2013, The Distributed Assembly Permutation Flowshop Scheduling Problem, *International Journal of Production Research*, 51 (17), 5292-5308.
- Kalczynski, P. ve Kamburowski, J., 2007, On the NEH heuristic for minimizing the makespan in permutation flow shops☆, *Omega*, 35 (1), 53-60.
- Karaboga, D. ve Basturk, B., 2007, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39 (3), 459-471.
- Kumar S., J. P., 2014, A Novel Hybrid Algorithm for Permutation Flow Shop Scheduling, *International Journal of Computer Science and Information Technologies*, 5 (4), 5057-5061.
- Li, X. ve Yin, M., 2012, A discrete artificial bee colony algorithm with composite mutation strategies for permutation flow shop scheduling problem, *Scientia Iranica*, 19 (6), 1921-1935.
- Li, Z., Duan, W., Ji, M., Yang, Y., Wang, S. ve Liu, B., 2016, The distributed permutation flowshop scheduling problem with different transport timetables and loading capacities. 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE: 2433-2437.
- Lin, J. ve Zhang, S., 2016, An effective hybrid biogeography-based optimization algorithm for the distributed assembly permutation flow-shop scheduling problem, *Computers & Industrial Engineering*, 97, 128-136.
- Lin, S.-W., Ying, K.-C. ve Huang, C.-Y., 2013, Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm, *International Journal of Production Research*, 51 (16), 5029-5038.
- Lin, S.-W. ve Ying, K.-C., 2016, Minimizing makespan for solving the distributed no-wait flowshop scheduling problem, *Computers & Industrial Engineering*, 99, 202-209.
- Ling-Fang, C., Ling, W. ve Jing-jing, W., 2018, A Two-Stage Memetic Algorithm for Distributed No-Idle Permutation Flowshop Scheduling Problem. 2018 37th Chinese Control Conference (CCC): 2278-2283.
- Liu, H. ve Gao, L., 2010, A Discrete Electromagnetism-Like Mechanism Algorithm for Solving Distributed Permutation Flowshop Scheduling Problem. 2010 International Conference on Manufacturing Automation: 156-163.
- Liu, Y.-F. ve Liu, S.-Y., 2013, A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem, *Applied Soft Computing*, 13 (3), 1459-1463.
- Naderi, B. ve Ruiz, R., 2010, The distributed permutation flowshop scheduling problem, *Computers & Operations Research*, 37 (4), 754-768.
- Naderi, B. ve Ruiz, R., 2014, A scatter search algorithm for the distributed permutation flowshop scheduling problem, *European Journal of Operational Research*, 239 (2), 323-334.
- Pan, J.-Q., Zou, W.-Q. ve Duan, J.-H., 2018, A Discrete Artificial Bee Colony for Distributed Permutation Flowshop Scheduling Problem with Total Flow Time Minimization. 2018 37th Chinese Control Conference (CCC). China: 8379-8383.

- Pan, Q.-K., Fatih Tasgetiren, M., Suganthan, P. N. ve Chua, T. J., 2011, A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem, *Information Sciences*, 181 (12), 2455-2468.
- Ribas, I., Companys, R. ve Tort-Martorell, X., 2015, An efficient Discrete Artificial Bee Colony algorithm for the blocking flow shop problem with total flowtime minimization, *Expert Systems with Applications*, 42 (15-16), 6155-6167.
- Rifai, A. P., Nguyen, H.-T. ve Dawal, S. Z. M., 2016, Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling, *Applied Soft Computing*, 40, 42-57.
- Ruiz, R., Pan, Q.-K. ve Naderi, B., 2019, Iterated Greedy methods for the distributed permutation flowshop scheduling problem, *Omega*, 83, 213-222.
- Shao, W., Pi, D. ve Shao, Z., 2017, Optimization of makespan for the distributed no-wait flow shop scheduling problem with iterated greedy algorithms, *Knowledge-Based Systems*, 137, 163-181.
- Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N. ve Chen, A. H. L., 2011, A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops, *Information Sciences*, 181 (16), 3459-3475.
- Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N. ve Oner, A., 2013, A discrete artificial bee colony algorithm for the no-idle permutation flowshop scheduling problem with the total tardiness criterion, *Applied Mathematical Modelling*, 37 (10-11), 6758-6779.
- Tosun, Ö., 2012, Yapay arı kolonisi algoritması ve permütasyon akış tipi çizelgeleme problemine uygulanması, *Akdeniz Üniversitesi Sosyal Bilimler Enstitüsü*, 1-111.
- Vijaychakaravarthy, G., Marimuthu, S. ve Naveen Sait, A., 2014, Comparison of Improved Sheep Flock Heredity Algorithm and Artificial Bee Colony Algorithm for lot Streaming in m-Machine Flow Shop Scheduling, *Arabian Journal for Science and Engineering*, 39 (5), 4285-4300.
- Wang, J., Wang, L. ve Shen, J., 2016, A hybrid discrete cuckoo search for distributed permutation flowshop scheduling problem. 2016 IEEE Congress on Evolutionary Computation (CEC): 2240-2246.
- Wang, K., Huang, Y. ve Qin, H., 2017a, A fuzzy logic-based hybrid estimation of distribution algorithm for distributed permutation flowshop scheduling problems under machine breakdown, *Journal of the Operational Research Society*, 67 (1), 68-82.
- Wang, K., Li, Z., Duan, W., Feng, X. ve Liu, B., 2017b, Variable neighborhood based memetic algorithm for Just-in-Time distributed assembly permutation flowshop scheduling. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC): 3700-3704.
- Wang, S.-y., Wang, L., Liu, M. ve Xu, Y., 2013, An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem, *International Journal of Production Economics*, 145 (1), 387-396.
- Ying, K.-C., Lin, S.-W., Cheng, C.-Y. ve He, C.-D., 2017, Iterated reference greedy algorithm for solving distributed no-idle permutation flowshop scheduling problems, *Computers & Industrial Engineering*, 110, 413-423.
- Zhang, S.-j. ve Gu, X.-s., 2015, An effective discrete artificial bee colony algorithm for flow shop scheduling problem with intermediate buffers, *Journal of Central South University*, 22 (9), 3471-3484.

ÖZGEÇMİŞ**KİŞİSEL BİLGİLER**

Adı Soyadı : İpek ARSEVEN
Uyruğu : TC
Doğum Yeri ve Tarihi : Antalya 1992
Telefon :
Faks :
e-mail : ipekarseven@hotmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Gazi Anadolu Lisesi, Muratpaşa, Antalya	2010
Üniversite	: Selçuk Üniversitesi, Selçuklu, Konya	2014
Yüksek Lisans	: Selçuk Üniversitesi, Selçuklu, Konya	2019
Doktora	: -	

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2015	Çelikel Tarım Makinaları	Planlama Müh.
2016	Ziya Usta Hipomak	Ürün Ağacı Sorumlusu

YABANCI DİLLER

İngilizce-Orta Seviye