



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ



**PERİYODİK MAKİNE KULLANILABİLİRLİK KISITI OLAN TEK
MAKİNE BAKIM PROBLEMİ İÇİN YAPAY ARI KOLONİ
ALGORİTMASI**

Cahit YALÇIN

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

MAYIS-2022
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Cahit YALÇIN tarafından hazırlanan “PERİYODİK MAKİNE KULLANILABİLİRLİK KISITI OLAN TEK MAKİNE BAKIM PROBLEMİ İÇİN YAPAY ARI KOLONİ ALGORİTMASI” adlı tez çalışması 20/05/2022 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

İmza

Başkan

Prof. Dr. Orhan ENGİN

.....

Danışman

Doç. Dr. Ahmet SARUCAN

.....

Üye

Dr. Öğr. Üyesi Kadir BÜYÜKÖZKAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
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.

Cahit YALÇIN

20/05/2022

ÖZET

YÜKSEK LİSANS TEZİ

PERİYODİK MAKİNE KULLANILABİLİRLİK KISITI OLAN TEK MAKİNE BAKIM PROBLEMİ İÇİN YAPAY ARI KOLONİ ALGORİTMASI

Cahit YALÇIN

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

Danışman: Doç. Dr. Ahmet SARUCAN

2022, 66 Sayfa

Jüri

**Prof. Dr. Orhan ENGİN
Doç. Dr. Ahmet SARUCAN
Dr. Öğr. Üyesi Kadir BÜYÜKÖZKAN**

Çizelgeleme problemlerinde, genellikle makinelerin her zaman müsait olduğu varsayılır ve bakım ile kurulum zamanları çoğunlukla göz ardı edilir. Ancak gerçek hayat problemlerinde bu varsayım geçerli değildir. Aslında bir üretim sisteminde, makineler genellikle periyodik olarak kullanılabilir veya kullanılabilir olmayan durumdadır. Makine kullanılabilirliği literatürde makine bakım olarak belirtilmektedir, çünkü genellikle bu kullanılabilirlik sürelerinin bakım faaliyetlerini gerçekleştirmek için kullanıldığı varsayılmaktadır. Bu çalışmada periyodik makine kullanılabilirlik kısıtı olan tek makine periyodik bakım problemi incelenmiştir. Bu çalışmanın ele alınmasının başka bir sebebi de, vardiya/mesai süresi içinde imalat operasyonlarını tamamlanması gerekli görülmüştür. Bundan dolayı tek makine çizelgeleme problemini, periyodik makinenin kullanılabilirliğine uygun bir tamamlama zamanı (makespan) bulmayı amaç edinilmiştir. Verilen problemin, polinomiyele olmayan zor problemler (NP-Zor) olması nedeniyle yaklaşık sonuçlar veren farklı birkaç sezgisel verilmiştir. Ayrıca, daha önceki çalışmalarda bu problem klasik kutulama problemine (bin packing) benzetilerek sonuçlar alınmıştır. Bu çalışmada periyodik makine kullanılabilirlik kısıtı olan tek makine periyodik bakım problemi, Yapay Arı Kolonisi (YAK) algoritması kullanılarak çözülmüştür. Sonuçlar literatürdeki çalışmalarla kıyaslanmıştır. YAK algoritmasının küçük problemlerde iyi fakat büyük problemlerde ise etkili sonuçlar vermediği görülmüştür. Bunun için literatürde yer alan yeni en iyi sığdır (NEW_BF) sezgiseli, YAK algoritmasına dahil edilerek bir iyileştirilmiş YAK (İYAK) algoritması önerilmiştir. Önerilen bu algoritma bütün problem boyutları için iyi sonuçlar vermiştir. İYAK algoritmasının sonuçları NEW_BF ile karşılaştırılarak etkinliği kanıtlanmıştır.

Anahtar Kelimeler: Çizelgeleme, Tek Makina, Periyodik Makine Kullanılabilirliği, Yapay Arı Kolonisi, Makine Bakımı.

ABSTRACT

MS THESIS

**ARTIFICIAL BEE COLONY ALGORITHM FOR A SINGLE MACHINE
MAINTENANCE PROBLEM WITH PERIODIC MACHINE AVAILABILITY
CONSTRAINTS**

Cahit YALÇIN

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

Advisor: Assoc. Prof. Dr. Ahmet SARUCAN

2022, 66 Pages

Jury

Prof. Dr. Orhan ENGİN

Assoc. Prof. Dr. Ahmet SARUCAN

Asst. Prof. Dr. Kadir BÜYÜKÖZKAN

In scheduling problems, machines are often assumed to be available at all times, and maintenance and setup times are often ignored. However, this assumption is invalid for real-life problems. In fact, in a production system, machines are often either periodically available or unavailable. Machine availability is referred to in the literature as machine maintenance because it is generally assumed that these availability periods are used to perform maintenance activities. In this study, the single machine periodic maintenance problem, which has periodic machine availability constraints, is investigated. Another reason for considering this research is the requirement to complete the manufacturing operations within the shift/working hours. Therefore, in the single machine scheduling problem, it is aimed to find a completion time suitable for the availability of the periodic machine. Since this problem is NP-hard, approximate results are obtained using different heuristic algorithms. Additionally, in previous studies, this problem was compared to the classical packaging problem (bin packing), and results were obtained. In this study, the single machine periodic maintenance problem with periodic machine availability constraint is solved using Artificial Bee Colony (ABC) algorithm and the results are compared with previous studies. It has been seen that the ABC algorithm does not give good results in large problems, but is effective in small problems. For this, a New Best Fit (NEW_BF) heuristic in the literature was included in the ABC algorithm and an improved ABC algorithm was proposed. The proposed algorithm gave good results for all problem sizes. The results of the improved ABC algorithm were compared with the NEW-BF and its effectiveness was proven.

Keywords: Scheduling, Single Machine, Periodic Machine Availability, Artificial Bee Colony, Machine Maintenance

ÖNSÖZ

Yüksek lisans tez çalışmam sırasında bana yol gösteren, bilgi ve tecrübelerinden yararlandığım, desteğini benden esirgemeyen değerli danışman hocam, Doç. Dr. Ahmet SARUCAN'a, önemli katkılarından dolayı Dr. Öğr. Üyesi Kadir BÜYÜKÖZKAN ve Doç. Dr. Mehmet Emin BAYSAL hocalarıma, çeşitli aşamalarda bana destek olan değerli abim Mehmet ATAŞ'a ve her zaman maddi ve manevi destekleri ile yanımda olan çok değerli aileme sonsuz teşekkürlerimi sunarım.

Cahit YALÇIN
KONYA-2022



İÇİNDEKİLER

TEZ BİLDİRİMİ	vi
ÖZET	iv
ABSTRACT	v
ÖNSÖZ	vi
İÇİNDEKİLER	vii
ŞEKİLLER LİSTESİ	ix
ÇİZELGE LİSTESİ	x
SİMGELER VE KISALTMALAR	xi
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	4
2.1. TMPMB Kaynak Araştırması.....	4
2.2. YAK Algoritması Kaynak Araştırması	15
3. MATERYAL VE YÖNTEM	20
3.1. Materyal.....	20
3.1.1. TMPMB problemlerinin notasyonu	20
3.1.2. TMPMB problemlerinin gösterimi	20
3.1.3. TMPMB problemlerinin matematiksel modeli	21
3.2. Yöntem	25
3.2.1. Doğada arılar.....	26
3.2.2. Arıların yem bulma davranışları	28
3.2.3. YAK algoritması ve temel adımları	31
3.2.3.1. Başlangıç Parametreleri	31
3.2.3.2. Besin kaynaklarının oluşturulması	31
3.2.3.3. İşçi arı aşaması	31
3.2.3.4. Gözlemci arı aşaması	31
3.2.3.5. Kâşif arı aşaması	32
3.2.3.6. Temel YAK algoritmasının adımları	33
4. TMPMB PROBLEMİNİN ÖNERİLEN YAK ALGORİTMASIYLA ÇÖZÜMÜ	34
4.1. Örnek Setleri.....	37

4.2. Alt Sınırın Hesaplanması.....	38
4.3. Algoritma Parametreleri ve Paremetre Optimizasyonu.....	39
4.4. TMPMB Problemine İYAK Algoritmasıyla Çözümü.....	42
4.5. İYAK Algoritması için Örnek Çözüm.....	43
4.6. Örnek Setleri.....	45
4.7. Algoritma Parametreleri ve Paremetre Optimizasyonu.....	45
5. SONUÇ VE TARTIŞMA.....	49
5.1. Sonuç.....	49
5.2. Tartışma.....	50
KAYNAKLAR.....	51

ŞEKİLLER LİSTESİ

Şekil 1. İncelenen problemin gösterimi	20
Şekil 2. Farklı operatörlere bir örnek	25
Şekil 3. Arılarda görev dağılımı ve yaşan evreleri.....	28
Şekil 4. Arıların yiyecek arama davranışları.....	30
Şekil 5. Öncü arı ve gözcü besin kaynaklarını oluşturmak için bir besin kaynağındaki takas mutasyonları.....	35
Şekil 6. YAK algoritması akış diyagramı	38
Şekil 7. Programın şeması.....	40
Şekil 8. Öncü arı ve Komşu besin kaynaklarını oluşturmak için takas mutasyonları ve EUİSGS uygulama	44
Şekil 9. NEW_BF ve İYAK algoritmasının OBYF değerleri.....	47
Şekil 10. NEW_BF ve İYAK algoritmasının CPU değerleri.....	48



ÇİZELGE LİSTESİ

Çizelge 1. TMPMB problemlerinin çözümünde özet olarak kullanılan yöntemler.....	13
Çizelge 2. Tam faktöriyel parametre optimizasyonu sonuçları	40
Çizelge 3. 10'lu iş grubu için alınan sonuçlar.....	41
Çizelge 4. TMPMB probleminde kullanılan yöntemlerin OBYF değerleri.....	42
Çizelge 5. Tam faktöriyel parametre optimizasyonu sonuçları	46
Çizelge 6. 10'lu iş grubu için alınan sonuçlar.....	47
Çizelge 7. NEW_BF ve İYAK algoritmasının OBYF ve CPU değerleri	47



SİMGELER VE KISALTMALAR

Simgeler

C_j	: Tamamlanma Zamanı
C_{max}	: En Büyük Tamamlanma Zamanı
d_j	: j İşinin Teslim Tarihi (Due Date)
j	: İş İndisi
k	: Makine İndisi
L_j	: Geç Kalma Süresi (Lateness)
L_{max}	: Maksimum gecikme
m	: Makine Sayısı
mg	: Minimum Gecikme
n	: İş sayısı
P	: Farklılaştırma Kümesi
p_{ij}	: İşlem Süresi (Processing Time)
R	: Referans Kümesi
r_j	: Serbest Bırakma Zamanı (Release Date)
T_j	: İki bakım arasındaki süre
t	: Bakım süresi
w_j	: Ağırlık (Weight)
α	: Makine Çevresi
β	: Kısıtlar
γ	: Amaç Fonksiyonu
π	: İş Dizilimi
MIN	: Karşılaştırılan yöntemler için elde edilen optimal veya en iyi bağlı tamamlama zamanı değeri
METH	: Her bir yöntem için elde edilen tamamlama zamanı değeridir

Kısaltmalar

ATÇ	: Akış Tipi Çizelgeleme
BYF	: Bağlı Yüzde Fark
EİSA	: En İyi Sığdır Azalan
EKİSGS	: En Kısa İşlem Sırasına Göre Sıralama
EUİSGS	: En Uzun İşlem Sırasına Göre Sıralama
İUSA	: İlk Uygun Sığdır Azalan
KTDP	: Karma Tamsayı Doğrusal Programlama
KTP	: Karma Tamsayı Programlama
NEH	: Nawaz, Enscore ve Ham
NEW_BF	: Yeni en iyi sığdır sezgiseli
NP	: Polinomiyel olmayan problemler
NP-Zor	: Polinomiyel olmayan zor problemler
OBYF	: Ortalama Bağlı Yüzde Fark
P	: Polinomiyel problemler
PSO	: Parçacık Sürüsü Optimizasyonu
TMPMB	: Tek Makineli Periyodik Makine Bakım
YA	: Yerel Arama
YAK	: Yapay Arı Kolonisi
İYAK	: İyileştirilmiş Yapay Arı Kolonisi

1. GİRİŞ

Günümüzün rekabete dayalı ortamında, müşterilerin istek ve beklentilerine hızlı bir şekilde cevap verilmesi, şirketlerin varlıklarını sürdürmeleri için çok önemlidir. Üretim prosesinde işletme yönetimi, verimliliğin ve hizmet değerinin iyileştirilmesinde önemli bir etkiye sahiptir. Özellikle, birden fazla fabrika arasındaki çizelgeleme problemleri, operasyonların otomasyonu için öncelikli bir faktör olarak kabul edilir. Şirketler, giderek artan rekabet ortamında ayakta kalabilmek için etkili yöntemler seçerek optimizasyon tekniklerinden yararlanılmalı ve üretim süreçlerinde daha iyi beklenen neticelerin alınabilmesi için gerçek hayatta karşılaşılabilecek problemlerin kısıtlarıyla daha gerçekçi bir çizelgeleme problemini analiz etmelidirler.

Çizelgeleme işlevi, matematiksel tekniklerle veya sezgisel yöntemlerle ilgili görevlere sınırlı kaynaklar atamaktır. Şirketin amaç ve hedeflerine ulaşması için uygun pozisyonlara kaynakların atanması sağlanmaktadır. Çizelgeleme literatüründe parametrelerin belirli veya belirsizlik olduğu, tekli veya çoklu makine olan ve statik olarak geliş sürecinin değiştiği farklı birçok problem yapıları vardır. Birden fazla kriter içeren çizelgeleme problemi çalışması son yıllarda önemli ölçüde artmıştır. Ancak bu tür problemlerin çözümü, tek kriterli problemler kadar basit değildir. Çakışan hedefler aynı anda optimize edildiğinden tek bir çizelgeyi oluşturmak kolay değildir. Çizelgeleme problemleri tümleşik optimizasyon problemleri, yani çok amaçlı karar verme problemleri olduğu için en iyi çözümü bulmak zordur.

Başka bir kaynakta; genel olarak, kesikli ve parça şeklinde üretimin var olduğu ortamında çizelgeleme, kıt kaynakların belli zaman alanında gerçekleştirilecek görevler için bir veya daha fazla hedefin optimizasyonuna tahsis edilmesi olarak tanımlanır. Bu nedenle, imalat endüstrilerinde çok önemli bir karar verme sürecidir. Kaynakları doğru bir şekilde tahsis ederek, işletmenin, amaç ve hedeflerine mümkün olan en iyi şekilde ulaşması sağlanır. Üretim çizelgeleme probleminde iki tür kullanılabilirlik kısıtlaması vardır: makine kapasite kısıtlamaları ve teknoloji kısıtlamaları. Çizelgeleme probleminin çözümü, bu iki tür kısıtlamanın karşılıklı olarak bağımlı ve tutarlı bir çözümdür. Elde edilen sonucun bize gerçekleştirilecek her görev için hangi kaynakların tahsis edilmesi gerektiğini ve her bir görevin ne zaman gerçekleştirileceğini söylemesi faydalı olacaktır. Bu nedenle, genellikle çoğu çizelgeleme problemi, kısıtları olan optimizasyon problemleri olarak düşünülmüştür (Cowling ve ark., 2002).

Çizelgeleme problemlerine ilk çözüm geliştirme Johnson (1954) tarafından olmuştur. Bu problemlere olan ilgi giderek arttı ve bunları çözmek için birçok yöntem önerilmiştir.

Johnson'ın algoritması, n iş-2 ve n iş-3 makine çizelgeleme problemlerine bazı özel şartlarda optimum çözümler sunmaktadır. Ancak makine sayısı arttıkça sorun daha da karmaşık hale gelmekte ve çözüm bulmak giderek zorlaşmaktadır. Bu problemler literatürde NP-Zor olarak geçmektedir. Problem çözme zorluğu, kısa sürede optimale yakın çözümler bulmayı mümkün kılan sezgisel ve meta sezgisel algoritmaların geliştirilmesinde rol oynamıştır.

Sezgisel ve meta sezgisel algoritmalar, çizelgeleme problemlerini çözmek için sık olarak kullanılmaktadır. Problemlerin çok boyutluluğu ve hesaplama karmaşıklığı, bu algoritmaların öncelikli olmasında önemli faktörlerdir. Birçok meta sezgisel algoritma doğadan ilham alır. Bunlara örnek olarak; Genetik Algoritma, Karınca Kolonisi Optimizasyonu, Parçacık Sürü Optimizasyonu (PSO), YAK, Benzetim Tavlama, Yasak Arama vb. verilebilir.

Akış tipi, çizelgeleme problemlerinin önemli bir alt dalıdır. Akış tipi çizelgeleme, bütün işlerin makineler arasında aynı sırada hareket ettiği üretim süreçleri için geliştirilmiş bir çizelgeleme yöntemidir. Bu özelliği ile akış tipi çizelgeleme; birbirinden farklı, m makine ile n işin olduğu; her bir işin m işleminden oluştuğu, her bir işlemin farklı makinelerde yapıldığı ve bütün işlerin işlemleri aynı sıra ile yapıldığı problemler olarak izah edilebilir. Literatürdeki çalışmalarda, akış tipi çizelgeleme problemleri için en çok kullanılan amaç fonksiyonu toplam tamamlanma zamanını minimize etmektir (Engin ve Fığlalı, 2002).

Çizelgeleme konusundaki literatürün çoğu, makinelerin her zaman mevcut olduğunu varsayar. Bununla birlikte, bakıma tabi makineler, proses endüstrilerinde ve üretim sistemlerinde yaygın olarak bulunur. Gerçek dünyada, makineler genellikle sürekli olarak mevcut değildir. Örneğin, havacılık endüstrisinde, ürünlerin yüksek kalitede kalmasını sağlamak için önleyici bir bakım veya alet değişikliği nedeniyle makine zamanlama döneminde kullanılamaz.

Üretim çizelgeleme ve önleyici bakım planlaması, imalat endüstrisinin karşılaştığı en yaygın ve önemli problemlerdir. Bazı araştırmacılar, çizelgeleme literatüründe makinenin kullanılamaması problemini incelemişler. Fakat yapılan araştırmalar neticesinde periyodik makine kullanılabilirlik kısıtı olan Tek Makineli Periyodik Makine Bakım (TMPMB) probleminde YAK algoritması kullanılmadığı görüldü. Bundan dolayı YAK algoritmasıyla bahsettiğimiz problem çözümlenince küçük sayıdaki iş problemlerinde güzel sonuçlar verdi. Fakat çok sayıda iş bulduran problemler ise kötü sonuçlar çıktı. Buna çözüm olarak aynı problem için en iyi sonuç veren sezgisel olan NEW_BF sezgiseli ile YAK algoritması birleştirilerek çözüme gidildi ve güzel sonuçlar bulundu. Bu çalışmada daha önceden yapılan sezgisel algoritmalar ve bir KTP model ile yapılan çözümlerin sonuçlarıyla YAK ile yapılan çözümlerin sonuçları kıyaslanmıştır.

Çalışmanın ikinci bölümünde, TMPMB ve YAK ile ilgili literatür araştırması yapılmıştır ve geçmişten günümüze sıralama yapılarak listelenmiştir. Ayrıca literatürde çalışılmış olan TMPMB problemlerinin kullanılan çözüm yöntemlerine göre sıralayıp listelenmiştir. Üçüncü bölümünde, materyal kısmında, TMPMB problemleri hakkında bilgi verilirken, yöntem kısmında ise, YAK algoritması hakkında bilgiler verilmiştir. Dördüncü bölümde, YAK algoritmasıyla yapılan çalışma paylaşılmış ve literatürde bilinen diğer sezgisel algoritma ile kıyas yapılarak sonuçlar tartışılmıştır. Beşinci bölümde ise İYAK algoritmasıyla yapılan çalışma paylaşılmış ve literatürde verilen NEW_BF sezgiseliyle kıyası yapılarak sonuçlar tartışılmıştır. Son bölüm olan altıncı bölümde, nihai sonuçlar hakkında bilgi verilmiştir ve gelecek çalışmalar için öneride bulunulmuştur.



2. KAYNAK ARAŞTIRMASI

Bu bölümde, TMPMB ve YAK ile ilgili literatür araştırması yapılmıştır ve geçmişten bugüne doğru sıralayıp listelenmiştir. Ayrıca literatürde çalışılmış olan TMPMB problemlerinin kullanılan çözüm yöntemlerine göre sıralayıp listelenmiştir.

2.1. TMPMB Kaynak Araştırması

Adiri ve ark. (1991), gecikmeli işlerin sayısını stokastik olarak en aza indirmek için tek bir arıza ile tek bir makinede çizelgeleme problemini çözmüşlerdir.

Lee ve Liman (1992), planlı bakıma için toplam iş akış süresini en aza indirmeye yönelik, tek makineli çizelgeleme problemini araştırmışlardır. En kısa işlem süresi dizisinin daha sonra 2/7'lik en kötü durum hata bandına sahip olduğunu göstermişlerdir.

Lee (1996) çizelgeleme ile ilgili yaptığı literatür çalışmasında, makinelerin her zaman aynı anda kullanılabilir olduğunu varsayıldığını ancak gerçek üretim ortamında bunun doğru olmadığını, makinelerin her zaman kullanılamayacağını söylemiştir. Bunun nedenin çizelgeleme periyodu boyunca bir makine arızası (stokastik) veya önleyici bakım (deterministik) çalışmalarından kaynaklanabileceğini belirtmiştir. Bu genel ve deterministik durum için zamanlama problemini, çeşitli performans ölçümleri ve makine ortamları/çevresinde incelemiştir. Her durumda önce sorunu çözmek için problemin bir polinom optimal algoritması veya NP-Zor olduğunu göstermiştir. Daha sonra ise, problemi en iyi şekilde çözmek ve/veya sezgisel olarak hataya bağlı bir analiz (provide heuristics with an error bound analysis) sağlamak için sahte (pseudo)-polinom dinamik programlama modelleri geliştirmiştir.

Sanlaville ve Schmidt (1998) makinelerin sürekli olarak işleme için uygun olmadığı deterministik çizelgeleme problemleriyle ilgili sonuçları gözden geçirmişlerdir. Zaman makinelerinin kullanılabilirliğinin değiştiği noktalarda eksik bilgi olabileceğini söylemişlerdir. Tek ve çoklu makine problemlerinin karmaşıklığı, tamamlanma süreleri ve teslim tarihlerine ilişkin kriterleri dikkate alarak analiz etmişlerdir. Bazı durumlar için, sayısal algoritmalarından ve sezgisel yöntemlerden elde edilen sonuçları da incelemişlerdir.

Yang ve ark. (2002) tek bir esnek bakım faaliyeti ile tek bir makine çizelgeleme problemlerini incelemişlerdir. Üretim süresini en aza indirmeyi amaçlamışlardır. Problemin NP-zor olduğunu kanıtlamışlar ve önerilen problemin $O(n \log n)$ karmaşıklığına sahip verimli bir sezgisel algoritma olduğunu göstermişlerdir.

Liao ve Chen (2003) periyodik bakım ile tek bir makine çizelgeleme problemini çözmüşlerdir. Büyük boyutlu problemler için optimale yakın çözümü bulmak için bir sezgisel algoritma önermişlerdir. Periyodik bakım ve yarıda kestirilemez kısıtlamaya sahip problem için maksimum gecikmeyi en aza indirmeyi amaçlamışlardır.

Wu ve Lee (2003), tek bir makinede kullanılabilirlik kısıtlaması ile üretim süresini en aza indirmek için doğrusal bozulan işleri çizelgelemeyi araştırmışlardır. Basitlik için, çalışmalarında devam ettirilebilir kullanılabilirlik kısıtlaması dikkate almışlar ve problemi 0-1 tamsayı programlama tekniğini kullanarak çözmüşlerdir.

Sadfi ve ark. (2005) bir tek makinenin çizelgeleme problemini bakım periyodu ile toplam tamamlama zamanını incelemişlerdir. Problemi 3/17'lik en kötü durum hatası sınıрыyla çözmek için bir yaklaşım algoritması önermişlerdir. Bir bakım periyoduna tabi olan tek makine toplam tamamlama çizelgeleme problemini incelemişlerdir.

Wang ve Xia (2005) kullanılabilirlik kısıtlamaları olan tek bir makinede n iş zamanlama problemini incelediler. İki zorlu özel durumu, yani orantılı ağırlık durumu ve tek kullanılabilirlik kısıtlaması durumunu ele aldılar. Amaçları, toplam ağırlıklı iş tamamlama süresini en aza indirmektir. Problemin güçlü anlamda NP-Zor olduğunu gösterdiler ve bu durumlar için iki sezgisel yöntem önerdiler ve en kötü durum hata sınırlarını analiz ettiler.

Yong ve ark. (2005) bir oran değiştirme faaliyetinin gerçekleştirilebileceği tek bir makinede bir dizi işin çizelgelemesi problemini ele aldılar. Oran değiştirme faaliyetinin yalnızca önceden belirlenmiş belirli zaman noktalarında gerçekleşebileceğini varsaydılar. Hedefleri, üretim süresini ve toplam tamamlanma süresini en aza indirmektir. Yaptıkları analiz, bazı özel durumlar için bile problemlerin NP-Zor olduğunu göstermektedir. Problemler için bir sahte-polinom zaman optimal algoritması önerdiler.

Chen (2006a), sınırlı makine kullanılabilirliği ile tek makineli bir çizelgeleme problemini incelemiştir. Periyodik bakım ve devam ettirilemeyen işlere tabi olan toplam akış süresini en aza indiren bir çizelge bulmayı amaçlamıştır.

Chen (2006b), esnek ve periyodik bakım ile tek makineli ve paralel makine çizelgeleme üzerinde çalışmıştır. Bu sorunları en iyi şekilde çözmek için birkaç karışık ikili tamsayı programlama modeli geliştirmiştir. Büyük boyutlu problemleri çözmek için bir sezgisel yöntem önermiştir.

Ji ve ark. (2006), Wu ve Lee'nin çalışmalarına benzer bir konuyu araştırdılar. Yarıda kesilemeyen iş durumu ele aldılar ve her iki problemin de yapım süresini ve toplam tamamlanma süresini en aza indiren amaç fonksiyonları altında NP-Zor olduğunu kanıtladılar.

Ji ve ark. (2007) Periyodik bakım faaliyetleri olan tek makineli zamanlama problemini ele aldılar. Bakımla ilgili çizelgeleme problemi arařtırmacıların dikkatini çekse de, gemiř alıřmaların oęu yalnızca bir bakım periyodu üzerinde yapıldı. Bu arařtırmada, her bir bakım faaliyetinin periyodik bir zaman aralıęından sonra çizelgelendięi birkaç bakım dnemi gz nnde bulundular. Amaları, periyodik bakım ve yarıda kesilemeyen iřlere baęlı olarak, tamamlama zamanını en aza indiren bir çizelgeleme bulmaktır. İlk nce klasik En Uzun İřlem Sırasına Gre Sıralama (EUİSGS) algoritmasının en kt durum oranının 2 olduęunu kanıtladılar. Ardından, Polinomiyele problemler (P) polinomiyele olmayan problemlere (NP) eřit (P = NP) olmadıka, en kt durum oranı 2'den kk olan bir polinom zaman yaklařımı algoritması olmadıęını gsterdiler. Bylece EUİSGS algoritmasının mmkn olan en iyi algoritma olduęu anlamına geldięini gsterdiler.

Sourd (2008) periyodik kullanılabilirlik kısıtlamaları ve dzensiz maliyet fonksiyonları ile çizelgeleme problemi üzerinde durmuřtur. alıřmasında, makine etkin kullanımı sabit olmadıęı, yani makinenin devamlı alıřamayıp kullanılabilir olmadı durumları olan tek makineli bir çizelgeleme problemini ele almıřtır. Etkinlik kısıtlamalarının zamansal-periyodik olduęu varsayımı altında, iř dizilerinin sabitlendięi zel durumun polinomal srede zleebileceęini gstermiřtir. Genel olarak bu durum NP-Tam'dır. Bu yzden iyi zmler bulmak iin iki ařamalı bir sezgisel nermiřtir.

Chen (2008) çizelgeleme periyodu boyunca sabit bir w sresi boyunca makinenin bakım iin periyodik olarak durdurulduęu varsayılarak, periyodik bakım ile tek bir makine çizelgeleme problemini incelemiřtir. Optimal zm elde etmek iin iki KTP modeli geliřtirmiřtir. Ek olarak, byk boyutlu problemler iin optimale yakın zm bulmak iin verimli bir sezgisel yntem nermiřtir.

Xu ve ark. (2008) tamamlanma zamanı minimizasyonu ile paralel bir makine çizelgeleme probleminde periyodik bakım faaliyetlerini arařtırdı. Problemin NP-Zor olduęunu gsterdiler. Daha sonra problemi zmek iin EİSA-EUİSGS adlı bir yaklařım algoritması nermiřlerdir.

Xu ve ark.(2009a) tamamlanma zamanını en aza indirmek amacıyla bir makinenin periyodik olarak kullanılmadıęı iki paralel makine çizelgeleme problemi gz nnde bulundurdular. Klasik EUİSGS algoritmasının en kt durum oranının ve Yerel Arama (YA) algoritmasının rekabet oranının evrimdiři durumu ve problemin evrimii durumu iin sırasıyla 3/2 ve 2 olduęu gstermiřlerdir.

Xu ve ark. (2009b), problem iin P = NP olmadıka, en kt durum performansı bandı 2'den kk olan hibir polinom zaman yaklařımı algoritması olmadıęını gstermiřlerdir.

Sonuç olarak, Chen'in sezgisel algoritmasının, dikkate alınan çizelgeleme problemi için mümkün olan en iyi polinom zaman yaklaşımı algoritması olduğunu göstermişlerdir.

Ma ve ark. (2009) deterministik makine kullanılabilirliği kısıtlamaları olan bir çizelgeleme problemini incelemişlerdir. Önleyici bakımın kullanılabilirlik kısıtlamaları ile çeşitli deterministik çizelgeleme problemleri daha çok literatürde çalışılmıştır. Bu kısıtlamayı içeren birçok çalışma son yıllarda yayınlanmıştır. Çalışmada, tek makine, paralel makine, akış tipi, açık atölye, atölye tipi çizelgeleme problemleri incelenmiştir.

Hsu ve ark.(2010) iki bakım stratejisi altında periyodik bakım faaliyeti olan tek makinalı çizelgeleme problemini incelemişlerdir. Tek veya periyodik bakım ve yarıda kesilemeyen işler için çizelgeleme problemi iyi çalışılmış olsa da, geçmiş çalışmaların çoğu sadece bir bakım stratejisi olarak düşünülmüştür. Araştırmalarında, sabit bir periyodik aralık veya sabit sayıda iş işlendikten sonra makinenin bakım için durdurulması gereken tek makineli bir programlama problemi ile ilgilenmişlerdir. Ele alınan problem için tamamlama zamanını en aza indirmeyi amaçlamışlardır.

Low ve ark. (2010a) kullanılabilirlik kısıtı ile tek makineli çizelgeleme problemini araştırmışlardır. Makinenin kullanılmamasının periyodik bakım faaliyetlerinden kaynaklandığını ifade ettiler. Çalışmalarında, birden fazla bakım döneminden oluşan periyodik bakım problemini incelediler. Makinede işlenen bir iş sonrasında kalıpların değiştirilmesi veya periyodik bir zaman aralığında bakım için makinenin durdurulmasından sonra her bakımın belirli zaman aralığında çizelgelendiğini ifade ettiler. Deterministik çevre ve esnek bakım konuları altındaki problemleri incelediler. İşlerin yarıda kesilmeyeceğini varsaydılar. Ek olarak, gerçek üretim ayarları için daha makul bir esnek model önerdiler. Tamamlama zamanını en aza indirmeyi amaçladılar. Önerilen problem NP-Zor olduğundan, bazı sezgisel algoritmalar ile çözümünü sağladılar. Yaptıkları hesaplamalar sonucunda, önerilen İlk Uygun Sığdır Azalan (İUSA) algoritmasının iyi performans ortaya koyduğunu gösterdiler.

Low ve ark. (2010b) periyodik bakım faaliyetleriyle tek makine programlama problemini çözmek için bir PSO algoritması sunmuşlardır. Ele alınan çizelgeleme problemine PSO algoritmasını başarıyla uygulamak için en önemli hususun, etkili bir “problem haritalaması” ve “parçacık dizisi yapısı” mekanizmasının nasıl geliştirileceğinin belirlenmesi olduğunu açıklamışlardır. Ayrıca, önerilen PSO algoritmasının etkinliğini arttırmak için bir yeniden başlatma stratejisi ve üç durdurma kriteri yerleştirmişlerdir. Amaçları tamamlama zamanının en aza indiren bir çizelgeleme bulmaktır. Ele alınan problemin NP-Zor olduğunu 3 bölmeli probleme dönüştürülerek göstermişlerdir. Hesaplama sonuçları, önerilen değiştirilmiş

PSO algoritmasının hem çözüm doğruluğu hem de ele alınan problemi çözme verimliliği açısından oldukça tatmin edici olduğunu göstermişlerdir.

Xu ve ark. (2010) bir makinedeki işlerin kesintisiz yapıldığı ve periyodik kullanılmadığı iki paralel makine çizelgeleme problemini çalışmışlardır. Tamamlama zamanı en aza indirme amacı olan bir yarıda kesilebilir (preemptive) çevrimdışı zamanlama problemini araştırmışlardır.

Liu ve ark. (2010) bir makinenin periyodik olarak kullanılmadığı 2 tek tip makinede çevrimiçi zamanlama problemini çalışmışlardır. Problemin online olmasının anlamı, bir iş başlanıldığında, bir sonraki iş görülmeden önce 2 tek tip makineden birine atama yapılması gerektiğini ifade etmişlerdir. Yarıda kesilmeye izin verilmemiştir. Amaç tamamlama zamanını en aza indirmek olmuştur. Periyodik olarak kullanılmayan makinenin hızının 1'e normalize edildiğini, diğerinin hızının ise s olduğunu varsaymışlardır.

Ángel-Bello ve ark. (2011) programlanmış/periyodik önleyici bakım ve sıraya bağlı kurulum süreleri olan bir makinedeki işlerin sıralaması problemini ele almışlardır. Yazarlara göre, bu problem yöneylem araştırması literatüründe yeterince ele alınmamıştır. Yaptıkları hesaplamalı deneyler problemi kesin yöntemlerle çözmenin çok zor olduğunu göstermişlerdir. Önerdikleri yöntem çok kısa hesaplama zamanlarında yüksek kaliteli çözümler bulmuştur.

Liu ve ark. (2011) Periyodik kullanılabilirlik kısıtı ile paralel makinelerde çevrimiçi çizelgelemenin tamamlama zamanını en aza indirmek için en uygun algoritmaları araştırmışlardır. Bunun için iki çevrimiçi zamanlama problemini incelemiştir. Bunlardan ilki, periyodik olarak kullanılmayan bir makineyle m paralel makinelerde çevrimiçi çizelgelemedir. İkinci problem, herhangi bir makinenin periyodik olarak kullanılmadığı durum için iki aynı paralel makinede çevrimiçi çizelgelemedir. Çevrimiçi paradigma, bir iş sunulduğunda, bir sonraki iş görmeden önce geri alınamaz bir şekilde atanmasıdır yani bir işin araya girilmesine izin verilmediğini göstermişlerdir. Var olan her sürenin uzunluğunun 1 olarak normalleştirildiğini ve kullanılmayan her sürenin uzunluğunun $\alpha > 0$ olduğunu varsaymışlardır. Ayrıca özel bir durumu incelemişler ve Xu ve ark. (2009) tarafından önerilen EUIGS algoritmasının rekabetçi oran $3/2$ ile optimal olduğunu kanıtlamışlardır.

Pacheco ve ark. (2012) programlanmış önleyici bakım ve sıraya bağlı kurulum zamanları olan bir makinedeki işleri sıralama problemini ele almışlardır. İki NP-Zor problemi birleştirmişlerdir. Problemi çözmek için Tabu Arama ile çoklu başlangıç stratejilerini melezleyen bir sezgisel yöntem önermişlerdir. Yöntemlerini, problem için 420 örnek üzerinde yayınlanan yalnızca metasezgisel algoritma ile karşılaştırmışlardır. Sonuçların geliştirilen

metodu desteklediği ve çok kısa hesaplama sürelerinde yüksek kalitede çözümler bulabildiğini göstermişlerdir.

Xu ve Yang (2013) bir makinenin periyodik olarak kullanılmadığı, işlerin yarıda kesilmediği ve amacın tamamlama zamanı en aza indirdiği iki paralel makine programlama problemi için bir matematiksel programlama modeli önermişlerdir. İki paralel makine ayarının tek bir makine ayarına dönüştürülmesiyle bir model oluşturmuşlardır. Klasik EUİSGS ve YA algoritmasının ortalama durum analizleri sunmuşlardır. Hesaplamalı deneyler, EUİSGS algoritmasının YA algoritmasını iki ana parametrenin 96 kombinasyonunun hepsinde ortalama bir hata bakış açısıyla (the average-case error point of view) geçtiğini ve iş sayısı yirmiden büyük olduğundan, EUİSGS algoritmasının ortalama durum hatasının % 2'den az olduğunu ve YA algoritmasının ortalama durum hata bakış açısından EUİSGS algoritmasını geçebileceğini gösteren örnekler olduğunu göstermişlerdir.

Yu ve ark. (2013) periyodik bakım ve yarıda kesilmeyen işler ile tek makineli bir çizelgeleme problemini ele almışlardır. Amaçları, tamamlama zamanını en aza indirmektir. Klasik YA algoritmasının, EUİSGS algoritmasının ve değiştirilmiş EUİSGS algoritmasının, ele alınan problem için aynı en kötü durum oranına (worst-case ratio) ve aynı hesaplama karmaşıklığına sahip olduğunu göstermişlerdir. Göz önünde bulundurulan üç algoritmanın performanslarını ayrıntılı olarak karşılaştırmak için, algoritmaların en kötü durum oranları, optimum çizelgelemede ihtiyaç duyulan toplam bakım faaliyeti sayısının tek değişkenli fonksiyonları olarak oluşturmuşlardır. Analiz sonuçları, YA algoritmasına her zaman EUİSGS algoritması ile ilişkili sınırın üstün olduğunu ve EUİSGS ise daima değiştirilmiş EUİSGS algoritması ile ilişkili sınırın egemen olduğunu göstermişlerdir.

Maşkani ve Moslehi (2016) iki modlu esnek periyodik kullanılabilirlik kısıtlamaları altında tek bir makine çizelgeleme problemi üzerine çalışma yapmışlardır. Birçok çizelgeleme probleminde, makineler kullanılabilirlik kısıtlamaları ile karşılaşabilir. Bu çalışmada, esnek periyodik kullanılabilirlik kısıtlamaları olan tek makine çizelgeleme problemi için yeni bir tanım sunmuşlardır. Bu tanıma göre, her periyotta makinenin sürekli çalışma süresine karşılık gelen kullanılamazlık süresi ayrı bir şekilde değişmekte ve iki farklı değerde dikkate alınmıştır. Bu tür kullanılabilirlik kısıtlamaları iki modlu kullanılabilirlik kısıtlamaları olarak adlandırmışlardır. Amaçları, toplam tamamlanma zamanını en aza indirmektir. Karmaşıklık problemleri matematiksel bir modelle ele alındığında, problemlerin en iyi şekilde çözülmesi için $O(n \log n)$ zaman karmaşıklığına sahip sezgisel bir algoritma ve birkaç lemma eşliğinde dal-sınır algoritması ve etkin baskınlık kuralları önermişlerdir. Dal-sınır algoritmasının 22 işe

kadar problem çözebildiğini ve sezgisel algoritma için ortalama hatanın %1,05 olduğunu göstermek için 1680 örnek problemi için hesaplama sonuçları kullanmışlardır.

Li ve Lu (2016) bir makinenin periyodik kullanılabilirlik sınırlamalarına sahip olduğu iki makineli bir programlama problemi incelemişlerdir. Amaçları tamamlama zamanı en aza indirmek olmuştur. Bunun için 4/3'lük performans oranla bir çözüm algoritması geliştirmişlerdir.

Cui ve Lu (2017) işlerin çıkış tarihlerinin de dikkate alındığı esnek periyodik önleyici bakımlar ile tek bir makinede üretim çizelgeleme sorununu ele almışlardır. İşlerin hem yarıda kesilebilir hem de yarıda kesilemez olan durumlarını incelemişlerdir. İşlerin devam ettiği durumda, problemin polinom zamanında çözülebileceği erken çıkış tarih kuralı ile kanıtlamışlardır. İşlerin kesintiye uğradığı durum için, güçlü anlamda NP-Zor olduğunu kanıtlamışlardır. Ayrıca bir KTP matematiksel model önermişlerdir. Ardından, optimal çözümün özelliklerine dayalı etkin bir sezgisel erken çıkış tarih kurallı EUİSGS önermişler.

Xu ve ark. (2018) iki makineden birinde kullanılabilirlik kısıtlaması olan iki makineli bir akış atölyesi çizelgeleme senaryosunu araştırmışlardır. Kullanılabilirlik kısıtlamasının ilk makineye uygulandığı problem için yedi KTP diğeri içinse yedi analog önermişlerdir. Sayısal sonuçları, iki problemde herhangi biri için, karşılık gelen ilk üç KTP'nin her birinin, 100'e kadar iş boyutunu makul zamanlarda çözebileceğini göstermişlerdir.

Moslehi ve Mashkani (2018) esnek ve periyodik kullanılabilirlik kısıtlamaları olan iki modlu tek makine çizelgeleme problemini ele almışlardır. Bu problemde, her makine için maksimum sürekli/aralıksız çalışma süresi, izin verilen iki farklı değerle kademeli artmıştır. Ayrıca, her bir periyot için kullanılmama süresi, aynı periyotta makinenin maksimum aralıksız çalışma süresine ve yine kademeli artırılan iki farklı değere bağlıdır. Amaçları, gecikmiş iş sayısını en aza indirmek olmuştur. Birinci aşamada, sorunun karmaşıklığı araştırıp ve ikinci aşamada bir ikili tamsayılı programlama modeli, bir sezgisel algoritma ve bir dal-sınır algoritması önermişlerdir. 1680 örnek problem çözümlerinin hesaplama sonuçları, dal-sınır algoritmasının sadece 20 işe kadar olan problemleri çözmekle kalmayıp, aynı zamanda toplam problem sayısının %94.76'sını optimal olarak çözebildiğini göstermişlerdir. Elde edilen sayısal sonuçlara dayanarak, sezgisel algoritma için ortalama %2'lik bir ortalama hata elde etmişlerdir.

Shen ve Zhu (2018) işlem süresi ve onarım süresinin deterministik olmadığı, periyodik bakımlı tek bir makine çizelgeleme problemini incelemişlerdir. Deterministik olmayan problemlerle başa çıkmak için, belirsiz bir ortam altında tamamlama zamanı en aza indirmek için belirsizlik teorisini tanıtmışlardır. Belirsizliğin ters dağılımına dayalı olarak deterministik biçimlere dönüştürülebilen üç belirsiz programlama modeli önermişlerdir. YA ve EUİSGS

algoritmaları problemi çözmek için kullanmışlardır. İki algoritmanın farklı güven seviyeleri altında aynı en kötü durum oranına sahip olduğu ve EUİSGS algoritmasının daha iyi bir performans sınırına sahip olduğu kanıtlanmışlardır. Problem için hibrit bir akıllı algoritma tasarlayıp ve bazı sayısal deneyler önerilen modellerin ve algoritmanın etkinliğini göstermişlerdir.

Gonzalez ve Framinan (2018) tek bir makinede işlerin çizelgeleme problemini periyodik makine kullanılabilirlik dönemleriyle ele almışlardır. Bu problemde, programlama ufku, makinenin mevcut/kullanılabilir olduğu dönemlerden ve hiçbir işlemin gerçekleştirilemeyeceği diğer dönemlerden oluşmuştur. Literatürde, genellikle bu kullanılmayan dönemlerin bakım faaliyetlerini gerçekleştirmek için kullanıldığı varsayıldığından, problem periyodik bakım ile çizelgeleme olarak belirtilmiştir. Vardiya/mesai süresi içinde imalat operasyonlarının tamamlanması gerekliliği çalışmanın başka bir amacını oluşturmuştur. Bundan dolayı tek makine çizelgeleme problemine, periyodik makinenin kullanılabilirliğine uygun bir tamamlama zamanının bulunması amaçlanmıştır. Gösterilen problemin NP-Zor olması nedeniyle yaklaşık sonuçlar veren farklı birkaç yöntem verilip ve bu yöntemlerin hangisinin iyi olduğunu anlamak için karşılaştırmalar yapılmıştır. Ayrıca, problem klasik kutulama problemine benzediği için bu her iki problem arasındaki ilişki araştırılmıştır. Çalışmalarında, iki problemi ele alıp yeni sezgisel çözüm prosedürleri önermişlerdir.

Wang ve ark. (2018) bozulan işler ve esnek periyodik bakımın dikkate alındığı tek bir makine çizelgeleme problemi üzerinde çalışmışlardır. Tek makinede, değişen işlem periyotları ve bakım periyotları ile bir dizi işi işlemek için çalıştırmışlardır. Bir işleme döneminde, işlerin bir alt kümesi sırayla işlenip ve son işin tamamlanma süresi izin verilen maksimum süreyi aşmadığını ve her işin fiili işlem süresi, doğrusal bir işe özgü bozulma oranında büyüyüp ve dönem içindeki başlangıç zamanına bağlı olduğunu ifade etmişlerdir. İki işlem periyodu arasında sabit süreli bir bakım periyodu bulup ve bakım faaliyetleri, makinenin işleme hızının normal orana döndürülmesi için yapmışlardır. Amaçları, tüm işleri bir dizi işlem periyoduna programlamak ve zamanlamanın süresini en aza indirmek olmuştur. Problemi bir küme-bölümleme modeli kullanarak formüle edip ve bir çözüm yöntemi için bir dal-fiyat algoritmasını kullanmışlardır. Baskınlık kuralına sahip bir etiket belirleme algoritması, sütun oluşturmadaki fiyatlandırma sorununu çözmek için tasarlamışlardır. Önerilen yöntemin performansını değerlendirmek için rastgele oluşturulmuş bir dizi test örneği üzerinde hesaplama deneyleri yapmışlardır.

Nesello ve ark. (2018) periyodik bakım ve sıraya bağlı kurulum süreleri ile tek makine çizelgeleme problemiyle, periyodik bakım ve kurulumların gerekli olduğu tek bir makine

üzerinde işlerin çizelgelenmesini çalışmışlardır. Amaçları, tamamlama zamanını en aza indirilmesi olmuştur. Üç alternatif ark-zaman indeksli modelin yinelemeli çözümüne dayanan kesin bir algoritma önermişlerdir.

Krim ve ark. (2019) ağırlıklı tamamlama sürelerinin toplamını en aza indirmek için periyodik önleyici bakım ile tek makine çizelgeleme problemini ele almışlardır. Bu kriter kesinlikle tamamlama zamanından daha az çalıştılar. Ancak yine de teorik ve pratik seviyelerde ilgi çekici olmaya devam ediyor. Ağırlıklar, dönüştürülecek ürünlerin birim zaman başına elde tutma maliyetini ölçebilir olup bu kriterin küresel elde tutma maliyetini temsil edebilir olduğunu ifade etmişlerdir. Bu problemin NP-Zor olduğu kanıtlayıp problemin küçük boyutlu örneklerini çözmek için bir karışık tamsayılı doğrusal programlama formülasyonu önermişlerdir. Büyük örnekleri çözmek için, bu problem için hâlihazırda var olan çalışmaları genelleştiren üç özellik önermişlerdir. Bu özellikler, 1000'e kadar iş içeren örnekleri çözebilen verimli sezgisel yöntemlerin tasarlanmasında çok kullanışlı olmuştur. Önerilen sezgisel yöntemlerin performanslarını değerlendirmek için, problemin özel durumlarına dayalı alt sınırlar sağlamışlardır. Hesaplamalı deneyler, en iyi sezgisel yöntemin ortalama yüzde hatasının %10'dan az olduğunu göstermişlerdir.

Zhou ve ark. (2020), makinenin her zaman kullanılamaması ve işlerin birbirini izleyen iki bakım periyodu arasında işlenmesi gereken işleri, önceden belirlenmiş periyodik önleyici bakım faaliyetleriyle ilgili tek makineli bir çizelgeleme problemiyle ele almışlardır. Tamamlanma zamanını en aza indirmek için bir KTP modeli ve iki sezgisel yöntem önermişlerdir. Modellerinde daha fazla kısıtlama olduğu için model, Perez-Gonzalez ve Framinan'ın son modelinden daha verimli ve modellerinin elli işe kadar olan sorunları çözebildiğini ifade etmişlerdir. Ek olarak, ele alınan tüm sezgisel algoritmalar, iyileştirme prosedürüyle birleştiğinde, hesaplama giderlerinde çok küçük bir artışla benzer ve yüksek kaliteli çözümler elde edilebileceğini anlatmışlardır.

Qamhan ve ark. (2020) tek bir makine problemi için farklı süre pencerelerine sahip bir zaman aralığı periyodik bakım stratejisi ve iş çizelgeleme etkinliklerini ele almışlardır. Amaçları, üretim çizelgeleme ve periyodik bakım aralıklarının entegrasyonu yoluyla geciken işlerin sayısını en aza indirmek olmuştur. Küçük boyutlu test örneklerini optimize etmek için bir Karma Tamsayılı Doğrusal Programlama (KTDP) modeli önermişlerdir. Ayrıca, daha büyük boyutlu test örneklerini çözmek için bir karınca kolonisi optimizasyonu algoritması geliştirilmişlerdir.

Prata ve ark. (2021) periyodik kaynak kısıtlamaları olan tek makineli bir çizelgeleme probleminin çözümünde sezgisel yöntemler sunmayı amaçlamışlardır. Tek makineli

çizelgeleme probleminin bu yeni varyantında, her üretim periyodunda kaynak tüketimi kısıtlamaları vardır. Bu önerilen varyant literatüre bakıldığında daha önce ele alınmamıştır. İki paketleme kısıtlamasını dikkate alan bir kutulama formülasyonuna dayalı bir tamsayı doğrusal programlama modeli sunmuşlardır. Benimsenen amaç fonksiyonları tamamlama zamanı minimizasyonu olmuştur. Performans kriteri olarak nispi sapma kullanmışlardır. İncelenen problem NP-Zor olduğundan, kabul edilebilir hesaplama sürelerinde yüksek kaliteli çözümler elde etmek için sezgisel algoritmalar önermişlerdir. On sekiz yapıcı sezgisel yöntem, iki yerel arama sezgisel yöntemi ve boyut küçültme ve simüle edilmiş tavlama algoritmalarına dayanan bir hibrit metasezgisel yöntemi sunmuşlardır. Gerçekleştirilen kapsamlı hesaplama deneyimi, önerilen yerel arama algoritmalarının yanı sıra önerilen meta sezgisellerin büyük boyutlu örnekleri çözmek için umut verici araçlar olduğunu göstermişlerdir.

Çizelge 1. TMPMB problemlerinin çözümünde özet olarak kullanılan yöntemler

Yıl	Yazar	Kullanılan Çözüm Yöntemi
1991	Adiri ve ark.	Yeni çözüm politikası
1992	Lee ve Liman	En Kısa İşlem Sırasına Göre Sıralama (EKİSGS) sezgiseli
1996	Lee	Sahte (pseudo)-polinom dinamik programlama modeli ve Sezgisel yöntem
1998	Sanlaville ve Schmidt	Sezgisel Yöntem
2002	Yang ve ark.	Sezgisel yöntem
2003	Liao ve Chen	Sezgisel yöntem
2003	Wu ve Lee	İkili tamsayılı programlama model
2005	Sadfi ve ark.	Yaklaşım algoritması
2005	Wang ve Xia.	Sezgisel yöntem
2005	Yong ve ark.	Sahte-polinom zaman optimal algoritması
2005	Sadfi ve ark.	Yaklaşım algoritması
2006a	Chen	Sezgisel yöntem
2006b	Chen	YA ve EUİSGS sezgiselleri
2006	Ji ve ark.	Sezgisel Yöntem
2007	Ji ve ark.	EUİSGS algoritması
2008	Sourd	iki aşamalı bir sezgisel
2008	Xu ve ark.	YA ve EUİSGS sezgiselleri
2008	Chen	Karışık ikili tamsayı programlama modeli ve Sezgisel
2009a	Xu ve ark.	EİSA-EUİSGS sezgiseli
2009b	Xu ve ark.	Polinom zaman yaklaşımı algoritması
2010	Ma ve ark.	Yöntemleri inceleme

Yıl	Yazar	Kullanılan Çözüm Yöntemi
2010	Hsu ve ark.	İkili tamsayı programlama modeli ve En Uygun Azaltma (EUİSGS)
2010a	Low ve ark.	İUSA sezgiseli
2010b	Low ve ark.	PSO algoritması
2010	Liu ve ark.	YA algoritması
2010	Xu ve ark.	Wrap-Around kuralına dayanan polinom algoritması
2011	Ángel-Bello ve ark.	Metasezgisel prosedürlere dayanan bir çözüm yaklaşımı
2011	Liu ve ark.	Optimal bir algoritma ve EUİSGS sezgiseli
2012	Pacheco ve ark.	Tabu araması ve Sezgisel
2013	Yu ve ark.	EUİSGS, YA ve MLPT sezgiselleri
2013	Xu ve Yang	EUİSGS, YA sezgiselleri
2016	Mashkani ve Moslehi	Matematiksel model, birkaç lemma eşliğinde dal-sınır algoritması ve etkin baskınlık kuralları
2016	Li ve Lu	Yaklaşım algoritması
2017	Cui ve Lu	KTP, sezgisel ERD-EUİSGS ve dal ve sınır algoritması
2018	Xu ve ark.	Yedi KTP formülasyonu
2018	Nesello ve ark.	Üç alternatif ark-zaman indeksli model
2018	Shen ve Zhu	YA, EUİSGS ve Hibrit bir akıllı algoritma
2018	Moslehi ve Mashkani.	Sezgisel algoritma ve Dal-sınır algoritması
2018	Wang ve ark.	Küme-bölümleme modeli ve Dal-fiyat algoritması
2018	Gonzalez ve Framinan	KTDP, kutulama problemi ve sezgisel yöntem
2018	Shen ve Zhu	EUİSGS ve YA sezgiselleri
2019	Krim ve ark.	KTDP ve sezgisel yöntem
2020	Qamhan ve ark.	KTDP ve büyük boyutlu test örneklerini çözmek için bir karınca kolonisi optimizasyonu algoritması
2020	Zhou ve ark.	Bir KTP modeli ve iki sezgisel yöntem
2021	Prata ve ark.	Yerel arama, sezgisel yöntemi ve bir hibrit metasezgisel yöntemi

Çizelge 1’de TMPMB probleminin çözümünde kullanılan yöntemler incelendiğinde son yıllarda bu konuya rağbet olduğu görülmektedir. Fakat meta sezgisel yöntemlerin az kullanıldığı genellikle sezgisel yöntemlere ağırlık verildiği gözlemlenmiştir. Yapılan kaynak araştırmasındaki çalışmaların dağılımı göz önüne alındığında, YAK algoritması periyodik

bakım çizelgeleme probleminde hiç kullanılmadığı sadece bir defa esnek bakım çizelgeleme probleminde kullanıldığı görülmüştür.

Bu çalışmada TMPMB probleminin çözümünde kullanılan KTDP ve sezgisel yöntemlere ek olarak YAK algoritması kullanılıp sonuçları önceki sonuçlarla kıyaslanmıştır. Ardından daha iyi sonuçlar için YAK algoritmasına EISA sezgiseli entegre edilerek hibrit bir yöntem olan iyileştirilmiş YAK algoritması elde edilmiştir. Sonra TMPMB problemi iyileştirilmiş YAK algoritmasıyla çözümlenerek elde edilen sonuçlar NEW_BF ile karşılaştırılmıştır.

2.2 YAK Algoritması Kaynak Araştırması

Bu bölümde, literatürde çizelgeleme problemleri için YAK kullanan çalışmalar kısaca sunulmuştur.

Alataş (2010) yakınsama özelliklerini iyileştirmek ve YAK'ın yerel çözümlere takılıp kalmasını önlemek için parametre uyarlaması için kaotik haritalar kullanan yeni YAK algoritmaları önermiştir. Bu, klasik YAK algoritması tarafından her rasgele sayıya ihtiyaç duyulduğunda kaotik sayı üreticileri kullanılarak yapılmıştır. Yedi yeni kaotik YAK algoritması önermiş ve kıyaslama fonksiyonlarında farklı kaotik haritaları analiz etmiştir. YAK ve karmaşık dinamikler gibi farklı alanlarda ortaya çıkan sonuçları birleştirmenin, bazı optimizasyon problemlerinde sonuçların kalitesini iyileştirebileceğini tespit etmiştir.

Lei (2012) yarıda kesilemeyen işler ve esnek bakım ile aralıklı atölye çizelgeleme problemi ele almış ve çizelgeyi oluşturmak ve önleyici bakım operasyonunu idare etmek için etkili bir kod çözme prosedürünün kullanıldığı birçok amaçlı YAK algoritması önermiştir. Amacı, aralık oluşturma süresini ve toplam aralık gecikmesi adı verilen yeni tanımlanmış bir hedefi en aza indirmek olmuştur. Her döngüde, baskınlığa dayalı açgözlülük ilkesi benimsenmiştir. Gözcü arı için çözüm seçmek için baskınlığa dayalı bir turnuva ve baskın olmayan seti güncellemek için baskın olmayan sıralama uygulanmıştır. Çok amaçlı YAK algoritması ile elde sonuçlar, problem üzerinde iyi performans göstermiştir.

Han ve ark (2012a) toplam akış süresini minimize etmeyi amaçlayan engelli akış tipi çizelgeleme problemini çözmek için üç adet etkili melez ayırık YAK algoritmasını sunmuşlardır. Üç melez ayırık YAK algoritması sırasıyla işçi arılar, gözlemciler ve keşifçiler için yeni besin kaynakları oluşturmak için ayırık operatörleri uygulamış ve besin kaynakları sunmak için ayırık iş permütasyonunu kullanmıştır. Öncelikle iki sezgisel kural, çeşitlilik ve kesin kalite seviyesiyle bir başlangıç popülasyonu inşa etmek için sunmuşlardır. İkinci olarak kendi kendine adapte olabilme stratejisi işçi arılara uygulamışlardır. Üçüncü olarak dağıtım

algoritmasıyla, seçilmiş bireylerden kesin olarak öğrenildikten sonra gözlemci arılar için iyi çözümler oluşturmuşlardır. Son olarak algoritmanın yerel kullanma yeteneği, çok etkili yerel arama tabanlı komşu ekleme algoritmasını üç aşamada göstermişlerdir. Bunlar; işçi arı aşamasında belirlenen çözüm için yerel arama, gözlemci arı aşamasında yerel arama ve kaşif arı aşamasında yerel aramadır.

Han ve ark (2012b) çalışmasında, tamamlanma zamanını en aza indirmek amacıyla engelleyici akış tipi çizelgeleme problemini çözmek için gelişmiş YAK algoritması önermişlerdir. Önerilen gelişmiş YAK, çözümleri temsil etmek için ayrı iş permütasyonları kullanmışlardır. İşçi ve gözlemci arılar için yeni çözümler üretmede ekleme ve takas operatörleri uygulamışlardır. İzci arılar için çözümler üretmede diferansiyel evrim algoritması kullanmışlardır. Belirli bir kaliteye ve çeşitlilik düzeyine sahip bir başlangıç popülasyonu oluşturmak için probleme özel sezgisele dayanan bir başlatma düzeni oluşturmuşlardır. Algoritmanın yerel kullanım yeteneğini artırmak için ekleme komşuluğuna dayalı bir yerel arama algoritması önermişlerdir.

Han ve ark (2012c) toplam akış süresini en aza indirmek için engelli akış tipi çizelgeleme probleminin çözümünde ayrık YAK algoritması ele almışlardır. Ayrık YAK algoritması, besin kaynaklarını temsil eden ayrı iş permütasyonları ile çalışan, gözlemci ve izci arılar için yeni gıda kaynakları üreten ayrı operatörleri kullanmışlardır. İlk olarak, belirli kalite ve çeşitlilik seviyesine sahip bir başlangıç popülasyonu oluşmak için MinMax ve NEH'nin kombinasyonuna dayanan bir başlatma planı sunmuşlardır. İkinci olarak, ek komşu tabanlı yerel arama, algoritmanın yerel yerleşim kabiliyetini artırmak için gözlemci aşamasına uygulamışlardır. Son olarak, yıkım-yapım operatörünü izci arılar için çözümler elde etmede kullanmışlardır.

Cui ve Gu (2013) öncelikle vektör gösterimini kullanarak hibrit akış tipi problem modelini oluşturmuşlardır. İyileştirilmiş ayrık arı koloni algoritması bu problemde tamamlanma zamanını en aza indirmek için kullanmışlardır. İyileştirilmiş ayrık arı koloni algoritması, yeni evrim ve değişken komşuluk arama, işçi ve gözlemci arılar için yeni çözümler üretmişlerdir. Yıkım ve onarım prosedürleri, izci arılar için çözümler elde etmek için kullanılmışlardır.

Brajevic ve Tuba (2013) kısıtlı optimizasyon problemleri için geliştirilmiş bir YAK algoritmasını tanıtmışlardır. Geliştirilmiş YAK algoritmasını, modifikasyon oranı parametresinin ince ayar özelliklerinin geliştirilmesinde ve YAK algoritmasının değiştirilmiş izci arı aşamasında kullanmışlardır.

Vijaychakaravarthy ve ark. (2014), deęişken büyüklükteki alt alanlara sahip bir parti akışlı n -iş, m -makinelik akış tipi çizelgeleme probleminin tamamlanma süresini, toplam akış süresini en aza indirmek ve en uygun alt alan boyutunu belirlemek için problemin çözümünde geliştirilmiş koyun sürüsü kalıtım algoritması ve YAK algoritması kullanmışlardır. Önerilen algoritmalarla elde edilen sonuçlar, dięer geleneksel sezgisel çalışmaların performansı ile da karşılaştırmışlardır.

Zhang ve Gu (2015) etkili ayrık YAK algoritmasını, ara tamponlu akış tipi çizelgeleme probleminin maksimum tamamlanma süresini en aza indirmek için önermişlerdir. Ekleme etkili kombinasyonu ve deęişim operatörü, işçi arı aşamasında komşu bireyin üretilmesinde kullanmışlardır. Turnuva seçiminde iyileştirilmiş ekleme operatörü, komşu çözüm algoritmasının yerel aranabilirliğini geliştirmek için gözlemci arı aşamasına geçerek, yerel optimuma düşmesini önlemişlerdir. Seçilmiş iyi çözüm tekrarlı açgözlü algoritmanın kurulması, bozulması ve sonucun kötüleşmesiyle 2 boyutlu turnuva seçimi tekrar uygulamışlardır. Simülasyon sonuçları, bu algoritmanın son zamanlarda ortaya konan etkili iki algoritma ile kıyaslandığında daha iyi performans gösterdiğini ortaya koymuşlardır.

Yao ve ark. (2015) birleştirilmiş otomobil parçalarının bakım teslimatı problemini ele almışlardır. Birkaç otomobil yedek parça deposu arasında ortak teslimat yoluyla maliyetler düşürülmüştür. Teslimat problemi çok karmaşık olduğundan, müşterileri birkaç kümeye ayırarak ve YAK algoritmasını kullanarak çözmeye çalışmışlardır. YAK algoritmasının performansını iyileştirmek için izci ve kraliçenin oranını kontrol etmek için uyarlanabilir bir strateji kullanmışlardır. Son olarak, 23 kıyaslama problemi için hesaplama sonuçları, önerilen algoritmanın çok depolu araç rotalama problemlerini çözmek için etkili bir yöntem olduğunu göstermişlerdir.

Panda ve Swamy (2018) bakım üstyapı yenileme optimizasyon problemini çözmek için YAK algoritması önermişlerdir. Bu algorithmada, nektar ararken arıların toplu davranışları taklit edilmektedir. Bu yaklaşımda çeşitli senaryolar oluşturulmuş, her bir durumun optimallięi değerlendirilmiş ve bu şekilde üretilen bilgi, global optimallięe ulaşılan kadar sonraki değerlendirmede kullanmışlardır. Önerilen yöntemin etkinlięi sayısal bir örnekle gösterilmiştir. Elde edilen çözüm, literatürde bildirilen kesin çözüme benzemiştir. Sonuçları, önerilen yaklaşımla çok az hesaplama çabasıyla optimal yüzey yenileme değerlerinin elde edilebileceğini göstermişlerdir. Önerilen algoritmanın ana avantajı, bakım kararı için tetik değerlerinin spesifikasyonunu kaldırmış olmalarıdır.

Choong ve ark.(2019) bir hiper-sezgisel yöntem, yani modifiye seçim fonksiyonu, görevli ve gözcü arılar tarafından benimsenen komşuluk arama sezgisellerinin seçimini

otomatik olarak düzenleyebilecek şekilde uygulamışlardır. Lin-Kernighan yerel arama stratejisi, önerilen modelin performansını iyileştirmek için entegre etmişlerdir. Önerilen modelin etkinliğini göstermek için, 64 gezgin satıcı problemi örneğini değerlendirmişlerdir. Ortalama olarak, önerilen model 64 örneği yaklaşık 2,7 dakika içinde bilinen optimumdan %0.055'e çözmüştür. Diğer son teknoloji algoritmalarla yapılmış performans karşılaştırmaları, önerilen modelin etkinliğini ayrıca göstermişlerdir.

Yue ve ark. (2019) esnek paralel üretim hatlarında partileme ve karma model çizelgeleme problemini, karışık model ürünler arasındaki sıraya bağlı kurulum sürelerini göz önünde bulundurarak, üretim hatlarının yapım süresini en aza indirmeyi, hatlar arasındaki iş yükünü dengelemeyi ve aynı anda net karı en üst düzeye çıkarmayı amaçlamışlardır. Ek olarak, probleme yeni bir malzeme mevcudiyeti kısıtlaması getirilmişlerdir. Mevcut problem için optimale yakın Pareto çözümleri elde etmek için üç farklı güdümlü mekanizmaya sahip yeni bir Pareto tabanlı güdümlü YAK algoritması tasarlanmışlardır. Önerilen Pareto tabanlı güdümlü YAK algoritmasının etkin parametrelerini ayarlamak için Taguchi yöntemini benimsemişlerdir. Ayrıca, Pareto tabanlı güdümlü YAK kullanılarak 90 problem içeren dokuz farklı örnek seti oluşturup ve test etmişlerdir. Pareto tabanlı güdümlü YAK'ın performansını, çok amaçlı YAK algoritması, baskın olmayan sıralama genetik algoritması III ve geliştirilmiş güçlü Pareto evrimsel algoritması dahil olmak üzere literatürde çok amaçlı optimizasyon için kullanılan diğer üç ünlü yöntemle karşılaştırmışlardır. Hesaplama sonuçları, önerilen Pareto tabanlı güdümlü YAK 'ın, dikkate alınan test problemi örneklerine dayalı olarak hem çözüm çeşitliliği hem de kalite açısından diğer dikkate alınan algoritmalarından daha iyi performans gösterdiğini göstermişlerdir.

Lei ve Liu (2020) önleyici bakım ile dağıtık ilişkisiz paralel makine çizelgeleme problemlerini araştırmışlar ve üretim süresini en aza indirmek için bir bölmeli YAK algoritması önermişlerdir. Kaliteli çözümler üretmek için bütün taze bir işçi arı kolonisi ve üç gözcü arı kolonisine bölünen dört koloni, arama stratejilerinde farklılık göstermiştir. İzciyi ele almak için yeni bir yöntem kullanmışlar ve sürü optimizasyon verileri kullanılarak güncellemişlerdir. Bölmeli YAK algoritmasının performansını test etmek için kapsamlı deneyler yapmışlardır ve hesaplama sonuçları, Bölmeli YAK algoritmasının önleyici bakım ile dağıtık ilişkisiz paralel makine çizelgeleme problemleri üzerinde umut verici avantajlara sahip olduğunu göstermişlerdir.

Mao ve ark.(2020) önleyici bakım operatörü ile dağıtılmış permütasyon akış atölyesi çizelgeleme problemini çözmek için geliştirilmiş bir ayrık YAK algoritmasını sunmuşlardır. Optimize edilecek kriter tamamlama zamanıdır. Popülasyonu etkin bir şekilde

başlatmak için geliştirilmiş bir NEH sezgisel yöntemi önerilmişlerdir. İşçi arı evresi ve gözcü arı evresinde komşu çözümler üretmek için ekleme ve takas operatörüyle yerel bir arama yöntemini uyarlamışlardır. Keşif arısı aşamasında yerel optimumdan kaçınmak için imha ve yeniden yapılandırma aşamasına sahip küresel bir arama yöntemi tanıtmışlardır. Önerilen ayrık YAK algoritması için parametreler, bir deney tasarımı ve varyans analizi yoluyla kalibre edip ayrık YAK algoritmasının performansını test etmek için kapsamlı deneyler yapmışlardır.

Li ve Han (2020) bir bulut bilişim sistemindeki esnek görev çizelgeleme problemini hibrit bir akış tipi çizelgeleme problemi olarak modelleyip hibrit ayrık YAK algoritması ile çözmüşlerdir. Hem tek bir amaç hem de birden çok amaç göz önünde bulundurmışlardır. Çok amaçlı hibrit bir akış tipi çizelgeleme problemlerinde, üç hedef, yani maksimum tamamlanma süresinin minimizasyonu, maksimum cihaz iş yükü ve tüm cihazların toplam iş yükleri aynı anda ele almışlardır. İki farklı hibrit bir akış tipi çizelgeleme türü göz önünde bulundurmışlardır. Bunlar aynı paralel makinelere sahip hibrit bir akış tipi çizelgeleme ve bağlantısız makinelere sahip hibrit bir akış tipi çizelgelemedir. Önerilen algoritmada klasik YAK algoritmasında olduğu gibi işçi arı, gözcü arı ve kâşif arı olmak üzere üç tür yapay arı yer almıştır. Her çözüm bir tamsayı dizisi olarak temsil edilmiştir. Problem özelliklerini dikkate almak için, arama yeteneklerini geliştirmek için birkaç farklı türde permütasyon yapısı araştırmışlardır.

Özcan ve Simsir (2020) bir ray üretim hattı için bir yedek çizelgeleme modeli önermişlerdir. Değiştirme çizelgeleme problemi yıllık 1349 parça için tartışmışlardır. Problem hem boyutu hem de ele alınış şekliyle literatürde ilk kez incelenmiştir. Değiştirilenleri bir kalem olarak kabul ederek, değiştirme çizelgeleme problemine kutulama çözüm yöntemini entegre eden yeni bir model önermişlerdir. Alt sol kutulama algoritması, değiştirmeleri iki çözüm yönteminde atamak için kullanmışlardır. İlk olarak klasik zamana dayalı değiştirme yöntemi analiz etmişlerdir ve duruş süresini hesaplamışlardır. İkinci olarak, YAK algoritmasını entegre ederek esnek zamana dayalı değiştirme yöntemini geliştirmişlerdir. Sonuçlar her iki çözüm için de incelendiğinde esnek zamana dayalı değiştirme yöntemi, zamana dayalı değiştirme yöntemine göre duruş süresinde yaklaşık %12'lik bir azalma sağlamıştır.

3. MATERYAL VE YÖNTEM

3.1. Materyal

Bu çalışmada materyal, periyodik makine kullanılabilirlik kısıtı olan TMPMB çizelgeleme problemidir. TMPMB probleminin notasyon gösterimi ve matematiksel modeli aşağıda sunulmuştur.

3.1.1. TMPMB probleminin notasyonu

TMPMB probleminde makine, çizelgeleme ufku boyunca her zaman kullanılır durumda değildir (Şekil.1' bakınız). T zaman birimi, makine kullanılabilirlik periyodunda oluşan ve t zaman birimi, makine kullanılmadığında oluşan zaman birimleridir. TMPMB problemi için gereken notasyon ve tanımlar aşağıda sunulmuştur.

$i=(1,2,\dots,n)$: n tane blok sayısı;

$j=(1,2,\dots,n)$: n tane makinelerde işlenmesi gereken işin oluşturduğu kümeyi temsilen;

P_j : j . işinin işlem süresi;

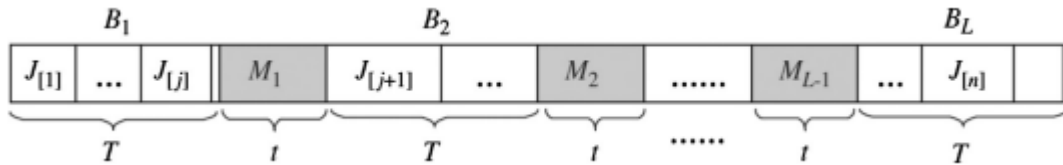
T : iki bakım arası süre;

t : bakım süresi;

C_{max} : En büyük tamamlanma zamanı;

C_j : j . işin tamamlanma zamanını yani teslim hazır olduğu zamanı ifade eder.

M : büyük bir tam sayıdır ($M \geq T$).



Şekil 1. İncelenen problemin gösterimi Ji ve ark.(2007)

3.1.2. TMPMB problemlerinin gösterimi

TMPMB problemini ifade etmek için $\alpha|\beta|\gamma$ şeklindeki bir gösterim kullanılmaktadır. Bu gösterimde α makinelerin bulunduğu akış tipi ortamını ifade eder. β işlem özellikleri ve beklemez (no-wait) olma durumunu ifade eder. γ ile gösterilen alanda ise performans ölçütü

yani amaç fonksiyon bilgisini ifade eder. Bu problem standart notasyona göre J_i ve ark.(2007) tarafından $1 | nr - pm | C_{max}$ olarak gösterilir.

α : Tek Makine (1): Tek bir makine düzeni durumu tanımlar.

β : Kesintisiz işler ($nr - pm$): Periyodik bakım (pm) kullanılamama dönemlerinin kısıtlama olarak kabul edildiğini ve işlerin devam ettirilemez (nr) olduğunu, yani işlerin kesintiye uğramaması gerektiğini gösterir (eğer kesinti olursa tekrar baştan işleneceğini).

γ : (C_{max}): Amacın makinedeki son işin tamamlanma süresinin minimizasyonu olduğunu gösterir.

3.1.3. TMPMB probleminin matematiksel modeli

Bu çalışmada, materyalimiz, periyodik kullanılabilirlik kısıtı olan tek makine bakım çizelgeleme problemidir. TMPMB literatürde, kullanılabilirlik dönemleri gruplar, kutular veya bloklar olarak belirtilmiştir. Yarı kesilemeyen işlerin operasyonları vardır. Dolayısıyla problemin fizibilitesini garanti etmek için $T \geq P_j \forall j = 1, \dots, n$ olduğu varsayılmaktadır. Mesela demir döküm işlemi yarıda kesilemeyen işlerdendir.

Bu çizelgeleme için, bir bloğun gevşekliliğini her bloğun boş zamanı olarak, yani T ile bu bloğa atanan tüm işlerin işlem sürelerinin toplamı arasındaki fark olarak gösterilmiştir. Ayrıca, bir bloğun yükü, bu bloğa atanan tüm işlerin toplam işlem süresini ifade eder.

$1 | nr - pm | C_{max}$ problemi güçlü bir şekilde NP-Zordur. Bu problem aynı zamanda klasik kutulama problemine benzer. Bu nedenle, bu tür periyodik bakım problemini çözmek için bazı iki aşamalı sezgisel yöntemler olan İlk Sığdır (FF) ve En İyi Sığdır (BF) gibi kutulama politikalarıyla karşılaştırılır. Bu sezgisel algoritmalar önce işleri sıralar ve ardından sipariş edilen işleri bloklara atamak için kutu paketleme politikaları izler. Perez-Gonzalez ve Framinan, literatürdeki mevcut iki aşamalı sezgisel algoritmaları bir listede gösterdiler. Ayrıca, problemler için optimal veya optimele yakın çözümler bulmak için bir Karma Tamsayı Programlama (KTP) formülasyonu ve yerel aramalı iki aşamalı sezgisel algoritmalar geliştirdiler.

Göz önünde bulundurulmuş problemimiz ile kutu paketleme problemi arasındaki benzerlikten esinlenerek, minimum blok (veya kutu) amaç edinen, Perez-Gonzalez ve Framinan'ın periyodik bakım KTDP modeli ve işlerin kutulara atama için iki farklı kutu paketleme politikasının kullanıldığı iki sezgisel algoritma çalışması gösterildi. Algoritma daha sonra probleme uygulanmış ve elde edilen sonuçlar, YAK algoritmasının sonuçlarıyla kıyaslanmıştır. Kutu paketleme politikalarından biri, tam parti özelliği tarafından motive edilen

minimum boş zamanıdır. Yani mümkün olduğunca dolu bir parti/blok elde etmek ve böylece üretim süresini en aza indirme amacına yarar. Diğeri ise minimum alt sınırdır.

Perez-Gonzalez ve Framinan (2018)'in formülasyonu:

Perez-Gonzalez ve Framinan (2018) tarafından ele alınan KTP modelinin notasyonları ve değişkenleri aşağıdadır:

Notasyonlar:

n , iş sayısı;

P_j , j işinin işlem süresi, $j = 1, \dots, n$;

T , kullanılabilirlik süresi (partilerin boyutu);

M , büyük bir tam sayıdır ($M \geq T$).

Değişkenler:

b_i , i partisi doluysa $b_i = 1$ aksi takdirde $b_i = 0$ ve $i = 1, \dots, n$;

x_{ij} , j işi i bloğuna atanmışsa $x_{ij} = 1$ aksi takdirde $x_{ij} = 0$, $i = 1, \dots, n$ ve $j = 1, \dots, n$.

$maxslack$, partilerin maksimum boşluğunu hesaplayan sürekli bir değişkendir.

δ_i i grubu tarafından maksimum boşluk verilirse $\delta_i = 1$.

Yukarıdaki değişkenleri kullanarak, Perez-Gonzalez ve Framinan (2018)'in KTP modeli şu şekilde açıklanmaktadır:

Bu modelde, t bakım süresini göstermektedir. Bu süre sabit olduğundan optimum sonucu etkilemez ve ihmal edilir. Denklem (1), minimuma indirilecek amaç fonksiyonunu gösterir: tamamlama zamanı, atanan blokların sayısı ile T çarpılır ve verilen blokların maksimum gevşekliğini çıkarmakla elde edilmiştir.

Denklem (2), bloklara atanan işlerin işlem sürelerinin toplamının T 'ye eşit veya daha düşük olması gerektiğini ve tüm bloklar için böyle olduğunu ifade eder. Denklem (3), her işin bir ve tek bir bloğa atanması gerektiğini ve tüm işler için böyle olduğunu ifade eder. Denklem (4) kullanılan bloklar arasında maksimum gevşekliği belirler. Bir blok açılmışsa en az bir işin atanması gerektiği Denklem (5) ile sağlanır.

Denklem (6) yalnızca bir bloğun maksimum gevşekliği sağladığını gösterir. Denklem (7) bir blok atanmazsa, bu bloğun maksimum gevşekliği sağlamadığını belirtir. Son olarak, Denklem (8) ve (9) sırasıyla, modelde kullanılan ikili ve sürekli değişkenleri tanımlar. Perez-

Gonzalez ve Framinan (2018)'in KTP modeli ile elde edilen tüm optimal çözümler daha sonra YAK algoritmasıyla kıyaslanmıştır.

$$\text{Min}(T \times \sum_{i=1}^n b_i - \text{maxslack}), \quad (1)$$

Kısıtlar;

$$\sum_{j=1}^n P_i x_{ij} \leq T b_i, \quad \forall i = 1, \dots, n, \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j = 1, \dots, n, \quad (3)$$

$$\text{maxslack} \leq M(1 - \delta_i) + T b_i - \sum_{j=1}^n P_i x_{ij} \leq T b_i, \quad \forall i = 1, \dots, n, \quad (4)$$

$$\sum_{j=1}^n x_{ij} \geq b_i, \quad \forall i = 1, \dots, n, \quad (5)$$

$$\sum_{i=1}^n \delta_i = 1, \quad (6)$$

$$\delta_i \leq b_i, \quad \forall i = 1, \dots, n, \quad (7)$$

$$x_{ij}, b_i, \delta_i \in \{0,1\}, \quad \forall i = 1, \dots, n; j = 1, \dots, n \quad (8)$$

$$\text{maxslack} \geq 0, \quad (9)$$

Perez-Gonzalez ve Framinan (2018)'in kullandığı en iyi en sonuç veren sezgiseller:

Literatürde kullanılan yapıcı buluşlar iki aşamadan oluşmaktadır. İlk aşamada işler, işlerin işleme sürelerine bağlı olarak belirli sıralama kriterine göre sıralanır. İkinci aşamada, belirli bir kutulama problemi politikası kabul edilir. Bu nedenle, yapıcı sezgiseller için genel bir şablon, probleme gerekli mevcut sezgiselleri sınıflandırmak için önerilebilir. İlk aşama ile ilgili olarak, genel bir kayıp olmaksızın, planlanan işlerin işlem sürelerinin azalan düzende verildiğini, yani $p_i \leq p_{i+1}$, $i = 1, \dots, n-1$ olduğu varsayılır. Bu nedenle, böyle endeksli işler aşağıdaki sıralama kriterlerine göre sıralanabilir:

- Rastgele (Random-R), yani işleri rasgele sıralamak.
- Azalan (Decreasing-D), yani işleri EUİSGS kuralına göre sıralama: p_n, \dots, p_1 .
- Artan (Increasing-I), yani işleri EKİSGS kuralına göre sıralamak: p_1, \dots, p_n .

- V-Şekli (V), yani işleri en düşük işlem süreleriyle sıralamanın ortasına doğru sıralamak: $p_n, p_{n-2}, \dots, p_1, \dots, p_{n-3}, p_{n-1}$.
- A-Şekli (A), yani işleri en yüksek işlem süreleriyle sıralamanın ortasına doğru sıralamak: $p_2, p_4, \dots, p_{n-1}, p_n, p_{n-2}, \dots, p_3, p_1$.
- Yüksek sonra düşük işlem süreleri ekleme (High and Low-HILO): $p_n, p_1, p_{n-1}, p_2, p_{n-2}, p_3, \dots$
- Düşük sonra yüksek işlem süreleri ekleme (Low and High-LOHI): $p_1, p_n, p_2, p_{n-1}, p_3, p_{n-2}, \dots$

İkinci aşamada, işler ilk aşamadan elde edilen diziye bir kutulama ilkesi uygulayan bloklara atanır. İlk aşamadan bir dizi $\pi = (\pi_1, \dots, \pi_n)$ elde edildiği varsayılarak, aşağıdaki kutulama politikalarından biri uygulanabilir:

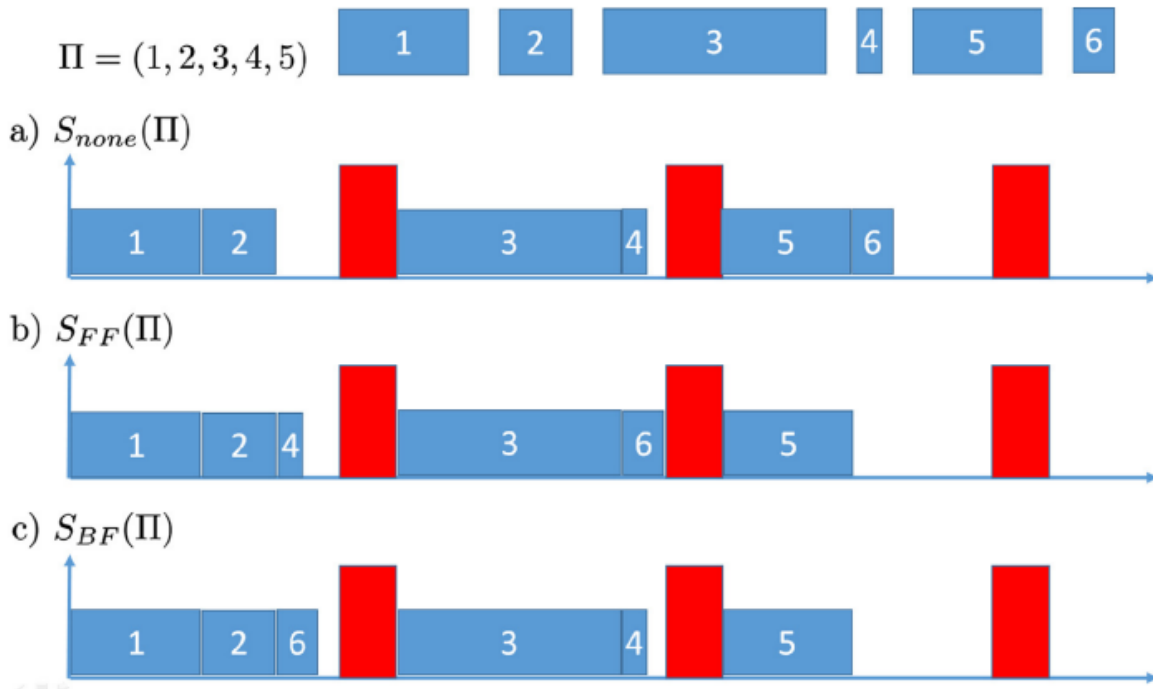
- * İlk Sığdır (FF) kutulama politikası, yani tek tek işleri alın ve yeterli gevşekliğin olduğu ilk mevcut blokta atamayı deneyin.
- * En İyi Sığdır (BF) kutulama, yani tek tek işleri alıp ve uygun minimum gevşeklik ile ilk mevcut blokta bunları atamak için deneyin.

Perez-Gonzalez ve Framinan (2018)'in önerdikleri sezgisel yöntem iki aşamadan oluşmaktadır: İlk önce EUISGS kuralını kullanarak bir çözüm oluşturulur sonra tamamlama zamanı hesaplamadan önce yukarıda bahsedilen operatörlerden biri uygulanır. Ardından çözümlerin boşluklarını keşfetmek için basit bir yerel arama uygulanır. Komşuluk yapısı, rastgele seçilen bir işin tüm pozisyonlarda bir ilk geliştirme stratejisi göz önünde bulundurularak yerleştirilmesine dayanmaktadır, yani eğer belirli bir pozisyondaki yerleştirmeden sonra elde edilen yeni çözüm, şimdiye kadarki en iyi çözümü geliştiriyorsa, mevcut çözüm değiştirilir ve geri kalan pozisyonlar kontrol edilmez. Tüm pozisyonlar kontrol edilmeden iyileştirilme yapılırsa, yöntem durur.

Şekil 2'de, altı ve bir permütasyonu $\pi = (1, 2, 3, 4, 5, 6)$ örnek olarak verilmektedir. Uygulanmış operatöre göre aşağıdaki seçeneklere sahiptir:

(a) Herhangi bir operatöre başvurmazsa, verilen permütasyona göre işler çizelgelenir ve Şekil 2. (a) 'da gösterilen $S_{\text{none}}(\pi)$ programını almış olur.

(b) FF operatörünü π 'ye uygularsak, iş 1, ilk bloğa atanır, iş 2 de ilk bloğa, daha sonra, ilk bloğa uymadığından iş 3, ikinci bloğa atanır. İş 4, uyduğu ilk blok olduğu için ilk bloğa atanır. Aynı şekilde, üçüncü bloğa iş 5 atanır ve iş 6 ikinci bloğa atanır. Bu nedenle, $S_{\text{FF}}(\pi)$, FF operatörü tarafından elde edilen ve Şekil 2 (b) 'de temsil edilen, öncekinden farklı bir çizelgeleme çözümüdür.



Şekil 2. Farklı operatörlere bir örnek (Gonzalez ve Framian 2018)

(c) BF operatörünü π 'ye uygularsak, iş 1 ilk bloğa, iş 2 de ilk bloğa atanır, daha sonra iş 3 ikinci bloğa atanır, çünkü ilk bloğa uymaz. Daha sonra, iş 4, ikinci bloğa atanır, çünkü bu, uyduğu minimum gevşek bloktur. Aynı şekilde, iş 5 üçüncü bloğa atanır ve iş 6 ilk bloğa atanır. Bu nedenle, $S_{BF}(\pi)$, Şekil 2 (c) 'de gösterilen önceki durumlardan farklı bir çizelgeleme çözümüdür.

3.2. Yöntem

Bal arılarının sürü halinde kendilerine besin bulmak için yaptıkları zekice davranış ve özel yöntemlerinden etkilenecek geliştirilmiş olan YAK bir optimizasyon algoritmasıdır. Sürü şeklinde hareket eden arıların besin bulmaya çalışırken doğada uyguladıkları davranışları esas alarak sürü zekasına istinat edilen algoritmayı optimizasyon problemleri çözümünde uygulanmaktadır.

Karıncalar, kuşlar, termitler, arılar, balıklar gibi sosyal bir toplulukta yaşayan bireylerin davranışlarını örnek alarak, problemlere çözüm getirmeyi hedefleyen bir yapay zeka tekniğine sürü zekası denir. Sürü zekâsının ortaya çıkışına dayalı akıllı(zeki) davranışlara örnek olarak: Karıncaların gittikleri yollarda kimyasallar bırakarak diğer karıncalara bilgi aktarması, birlikte hareket ederken kuşlar ile balıkların konum ve hızlarını ayarlayarak ve birbirleriyle iletişim

kurarak ilerlemeleri ve arıların kovan etrafındaki yaptıkları dans hareketi ile bilgi aktarımı verilebilir (Aybars ve Dođan, 2006; Karaboga ve Akay, 2007).

Sürüdeki bireylerin iki önemli özelliđi vardır: kendi kendine örgütlenme ve iş bölümü oluşturma. Kendi kendini örgütleme, sistemdeki diđer bireylerden aldığı bilgileri kullanan ve bu görevle sistemi kendi başına işleten ve etkileyen bireyler olarak tanımlanabilir. Bonabeau vd. kendi kendine örgütlenmeyi dört özellik ile tanımlarlar: olumlu geri bildirim, olumsuz geri bildirim, salınımlar ve çoklu etkileşimlerdir (Akay, 2009). İş bölümü, aynı anda farklı işlevleri yerine getiren bir topluluktaki bireylerin pratiđidir.

3.2.1. Doğada arılar

Sosyal bir intizamda yaşayan bal arıları içgüdüsel olarak bu intizamı bilir ve bu yasalara göre yaşamlarını sürdürürler. Kovandaki her arının görevi bellidir. Arılar bu görevlerin dışına çıkmaz. Yiyecek stoklamak, bal getirmek, iletişim kurmak, yiyecek aramak gibi işler toplumsal düzende kendilerine verilen görevlerdir. Bu şaşırtıcı denge ve aktivite, araştırmacılar tarafından fark edildi ve onları bu davranışları modellemeye teşvik etti.

Sürü halinde yaşayan arılar üç tipe ayrılır. Bunlar: kraliçe arı (queen), erkek arı (drone) ve dişi olan işçi arıdır (worker).

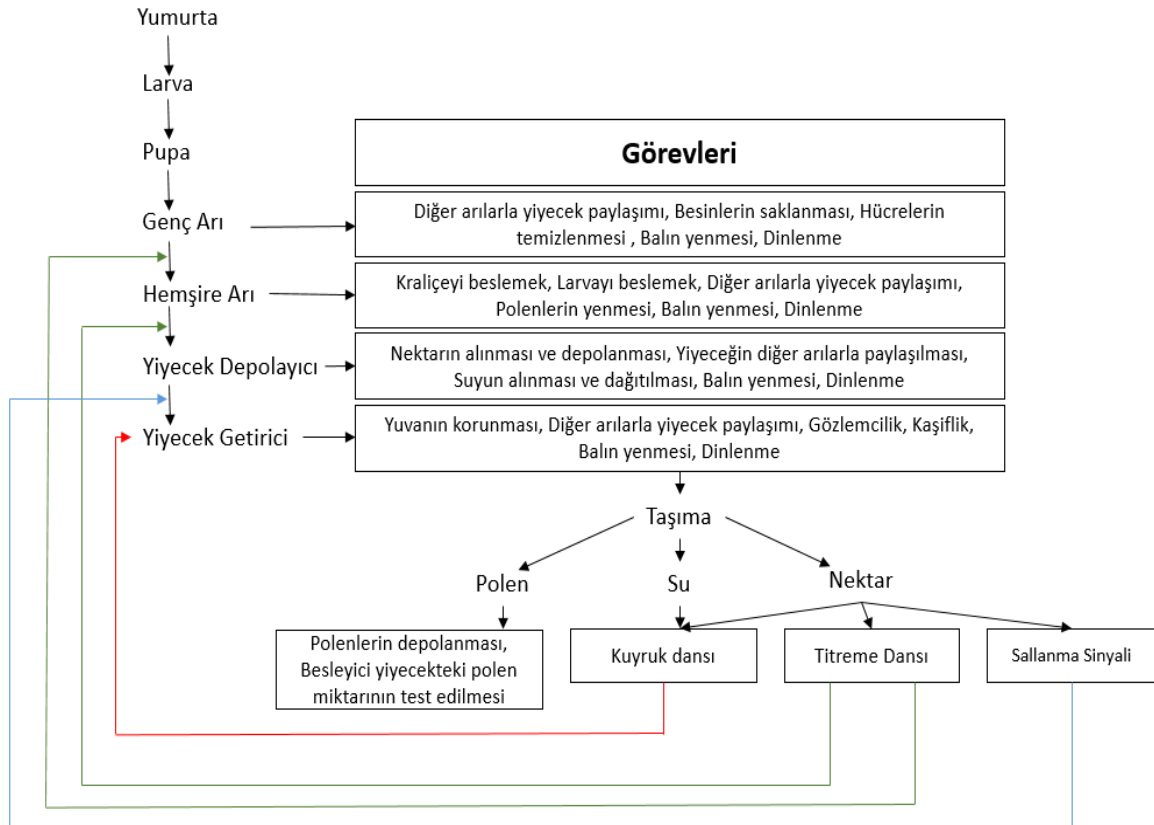
Kraliçe arı (queen bee): Her kolonide sadece bir kraliçe vardır ve bu kraliçe diđerlerinden daha büyüktür. Ana görevi yumurta yapmaktır. Üreme sadece ana arılar tarafından gerçekleşir, diđer dişiler erkeklerle çiftleşemez. Kraliçe arı, çiftleşme sırasında depoladığı spermle iki yıl yumurta bırakabilir. Yumurtlamanın yanı sıra koloninin bütünlüğünü ve yuva sisteminin işleyişini sağlayan önemli maddeler de salgırlar. Kraliçe arının salgıladığı bu madde sayesinde diđer arılar onu tanır ve bu koku etrafındaki tüm arı kolonisine yayılır. Farklı bir kraliçe arı ve ailesi bu şekilde anlaşılır. Bu şekilde kraliçe koloninin bütünlüğünü sağlar (Akay, 2009; Tokmak, 2011).

Erkek arılar (drones): Kolonide erkek arıların görevi kraliçe arıyı dölemektir. Dişi arılardan görünüş olarak daha büyüktürler ancak iğneleri veya yiyecek toplamak için organları yoktur. Erkek arılar döllenmemiş yumurtalardan oluşurken kraliçe ve işçi arılar döllenmiş yumurtalardan meydana gelir. Erkek arıların kolonideki görevi kraliçe arının döllenmesidir. Kraliçe arı ile çiftleşen erkek arı ölür (Akay, 2007; Tokmak, 2011).

İşçi arılar (workers): Arı kolonilerinin çoğunluğunu oluşturan işçi arılar, döllenmiş yumurtalardan çıkan üremeyen dişilerdir. Koloninin hayatta kalmasını sağlamak için her türlü içgüdüsel ve yapısal yeteneğe sahiptir. İşleri kendi aralarında bölerek çeşitli görevleri düzenli bir şekilde yerine getirirler. Besin toplamak, bu besinleri depolamak, kovayı temizlemek, ölü arı ve kalıntıları uzaklaştırmak, larva ve diğer arıları beslemek gibi birçok görevi üstlenirler. (Akay, 2009; Tokmak, 2011).

Çiftleşme uçuşu (mating flight): Kraliçe arılar ile genellikle etrafını saran, bakımını yapan, temizleyen ve besleyen bir erkek arı sürüsü arasında görülür. Gelişimini tamamladıktan sonra, yaklaşık bir hafta boyunca çiftleşmek için yuvadan uçar. Bu uçuşlar sırasında çiftleşme gerçekleşir. Sperm, Kraliçe arının spermatika olarak bilinen kesesinde birikir ve koloni devamı için korunur.

Görev paylaşımı (task selection): Bal arısı kolonisindeki çok sayıda farklı görev için bu görevlere uygun sayıda birey atanmalıdır (Dornhaus vd, 1998). Ana arının yumurtlaması ile arının gelişimi başlar. Yumurtalar zamanla larva ve pupaya dönüşür. Arıların yumurta ve larva döneminde beslenmeleri, beslenme alanından çıktıktan sonraki görevleri için de önemlidir. Hemşire arının larvaları beslerken verdiği arı sütünün süresi ve zamanına bağlı olarak, arı ya kraliçe ya da dişi arı olacaktır. Bu işleme göre yumurtadan çıkan arı, kendisine verilen görevleri tamamlamak için hayatına başlayacaktır. Erişkin arılar kovanın çıkışında dururken, besleme alanında genç arılar ve hemşire arılar çalışır. Yumurtadan yetişkinliğe geçiş döneminde her arının yiyecek arama, depolama, bal ve polen elde etme ve dağıtma, iletişim kurma ve besleme gibi görevleri vardır. Bir arının herhangi bir zamanda yaptığı iş, onun mevcut davranışsal rolüne, çevresi hakkında edindiği algılara ve bu algılara tepki eşliğine bağlıdır. Şekil 3'de arılara ait görev paylaşımı verilmiştir (Akay, 2009).



Şekil 3. Arılarda görev dağılımı ve yaşan evreleri (Akay, 2009).

3.2.2. Arıların yem bulma davranışları

Arı kolonisinin devamı için en önemli görevlerden biri yiyecek aramaktır. Kovanda biriken kaynaklar ve çevrede bulunabilecek besin kaynakları ve arı etkileşimleri bu süreç için çok önemli faktörlerdir.

Besin arama süreci, arının kovandan ayrılmasıyla başlar. Başta rastgele arama denemeleri ile devam eder. Bulunan kaynaktaki besin miktarı azaldıkça arılar, diğer arılardan aldıkları bilgilere göre yeni besin aramaya veya başka kaynaklara yönelmeye başlarlar. Polen, su vb. buldukları kaynakları kovana getirmeleri ve bulunan kaynak bilgisinin aralarında aktarımı bu süreçte gerçekleştirilen faaliyetlerdir.

Tereshko'ye göre besin arama modelinde üç ana unsur vardır. Bunlarda: besin kaynakları (food sources), görevi belirli arayıcılar (employed foragers:), görevi belli olmayan arayıcılar (unemployed foragers) dır (Tereshko ve Loengarov, 2005). Akay çalışmasında bu unsurları aşağıdaki gibi açıklamıştır (Akay, 2009).

Besin kaynakları: Arıların besin buldukları kaynaklardır. Bir besin kaynağının değeri, kaynağın türü, yuvadan uzaklığı, nektar miktarı veya nektarın ne kadar kolay elde edilebileceği gibi birçok faktöre bağlı olsa da, basitlik adına, kaynağın zenginliği adı altında tek kriter olarak alınabilir.

Belirli görevleri olan işçi arılar: Bu işçi arılar, önceden belirlenmiş kaynaklardan topladıkları besinleri kovana getirmekle sorumludur. Bir diğer görevi de, kovandaki diğer arılar ile gittikleri kaynağın konumu ve kalitesi hakkında bilgi paylaşmaktır.

Belirli görevi olmayan işçi arılar: Bu arılar, yiyecek toplayabilecek kaynakları arama eğilimindedir. Belirli görevi olmayan iki tür işçi arı vardır. Rastgele kaynak arayan kâşif arılardır, diğeri kovanda bekleyip belirli görevi olan arıları takip ederek bu arıların aldığı bilgilere göre yeni kaynaklara geçen gözcü arılardır.

Kolektif yapı ve ortak bilgi oluşturmada arılar arasındaki bilgi aktarımı en önemli konudur. Arıların yaşadığı kovan birkaç bölüme ayrılabilir. Bu alanlardan biri de bilgi paylaşımının yapıldığı dans alanı (dancing area) denilen bölgedir. Arılar arasındaki bilgi paylaşımı, arıların yaptığı bir dans (waggle dance) ile gerçekleşir. Bilgilerin paylaşımı sayesinde kaliteli yeni besin kaynakları bulunur(Grüter ve Farina, 2009).

Kaynaklardan besin taşıyan arılar, diğer arıları bu kaynaklara yönlendirmek için kaynağın konumu hakkındaki bilgileri diğer arılara iletmelidir. Konum bilgisi alan arı, bu amaca ulaşmak için güneş ışığını kullanır. Yörüngeleri ile güneş arasındaki açığı hesaplayabilirler. Enerji tüketimlerine göre mesafe belirleyen arılar, yem yüklerine göre farklı irtifalarda uçarak enerjilerini ayarlarlar (Akay, 2009).

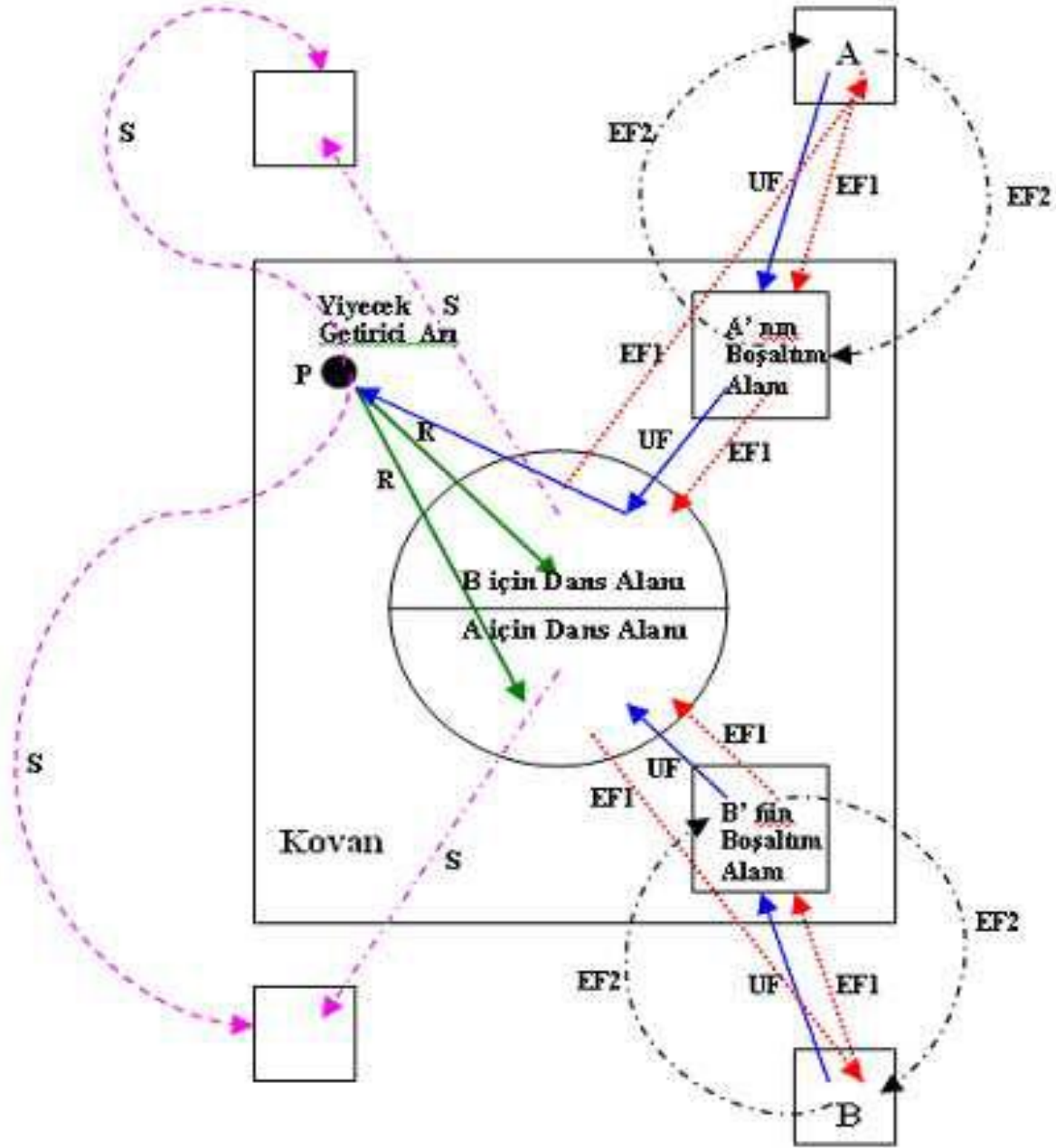
Arıların yem arama ve kovana getirmeleri hareketleri Şekil 4'de gösterilmiştir. Akay bu hareketleri şekilde ele almış ve açıklamıştır: A ve B'nin keşfedilmiş kaynak olduğu varsayılmaktadır. İlk başta görevi belli olmayan ve kaynakları bilinmeyen arı aramaya başlar. Bu arı için iki senaryo var: birincisi; kâşif arı olabilir. S ile gösterildiği gibi besin aramaya başlar. İkicisi; dans alanında dans eden arıları seyrederek tasvir edilen kaynağa gitmesi mümkündür. Bu arı da R olarak tanımlanmıştır.

Yiyecek için yola çıkan arılar buldukları kaynaklardan nektar getirirler ve artık bu arılar belirli görevi olan arı olmuştur. Arıların kovana nektarı getirmesinden sonra üç alternatifleri vardır:

i) Bilgiyi paylaşmadan kaynağından nektar getirmeye devam edebilir. EF2 bu arıyı temsil eder.

ii) Kaynağa dönmeden önce diğer arılar için dans ederek ve diğer arılarla nektar kaynağının yeri ve miktarı hakkında bilgi paylaşarak o kaynağa yönlendirebilir. EF1 bu arıyı temsil eder.

iii) Kaynaktan ayrılarak dans alanına gidip gözcü arıya dönüşür, dans alanında pusuda bekleyebilir. UF bu arıyı temsil eder.



Şekil 4. Arıların yiyecek arama davranışları(Akay, 2009)

3.2.3. YAK algoritması ve temel adımları

Bal arılarının doğada yiyecek arama davranışı üzerine modellenen bu algoritma Karaboğa ve Baştürk (2007) tarafından geliştirilmiştir. Bu algoritmada yem toplamak için aramaya çıkan arılar 3 gruba ayrılır. Bunlar; işçi arılar, gözlemci arılar ve kâşif arılardır. Arı koloninin bütünü işçi arılar ve gözlemci arılar oluşturur. İşçi arılar yiyecek kaynağı bulmak için aramaya çıkarlar, buldukları yiyeceklerin bilgisini hafızalarında bulundurlar. İşçi arılar, Gözlemci arılardan daha düşük toplayıcı olma özelliğindedir. Yeni yiyecekler bulmak için bazı işçi arılar, kâşif arı olurlar. Döngü, durdurma kriteri sağlanana kadar devam eder. YAK algoritması, diğer popülasyon temelli algoritmalar gibi yinelemeli bir procestir.

3.2.3.1. Başlangıç parametreleri

Temel YAK algoritmasının üç tane parametresi vardır (Tosun 2012). Bunlar:

- Besin kaynağı sayısı (BS), işçi arı ve gözlemci arıların toplamına eşittir.
- Besin kaynağının ayrılmak için sınır değeri
- Tekrar sayısıdır.

3.2.3.2. Besin kaynaklarının oluşturulması

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

Besin kaynaklarının oluşturulmasında aşağıdaki denklemden yararlanılır (Pan ve ark. 2011).

$$X_{i,j} = X_j^{min} + r(X_j^{max} - X_j^{min}) \quad (10)$$

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.

3.2.3.3. İşçi arı aşaması

İşçi arılar rastgele besin kaynaklarına gönderilir. Besin kaynağı başına sadece bir işçi arı bulunabilir. İşçi arılar, çözüm değerini hesaplayarak besin kaynağındaki nektar miktarını kaydederler. Böylece bu aşamadayken, her işçi arı, şu anki $X_{i,j}$ pozisyonunun yakınında, aşağıdaki gibi çözüm arama denklemi kullanarak yeni bir besin kaynağı üretir (Li ve Yin 2012):

$$V_{i,j} = X_{i,j} + \varphi_{i,j}(X_{i,j} - X_{k,j}) \quad (11)$$

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.

3.2.3.4. Gözlemci arı aşaması

Gözlemci arılar, işçi arıların bulduğu besin kaynağının kalitesini değerlendirerek bir $X_{i,j}$ seçerler. Bu seçim aşağıdaki denkleme göre hesaplanan olasılık değerine göre yapılır.

$$p_i = \frac{u_i}{\sum_{i=1}^{BS} u_i} \quad (12)$$

Buradaki u_i , i . besin kaynağının uygunluk değerini belirtir. p_i ise, i besin kaynağının seçme olasılığını gösterir. p_i oranı büyük olan besin kaynaklarının seçilmesi olasılığı daha yüksektir (Pan ve ark. 2011).

Bulunan kaynağın kalitesine bağlı olarak kaynağın uygunluk değeri Denklem 13'a göre hesaplanır.

$$fitness(uygunluk)_i = \begin{cases} \frac{1}{1 + f_i}, & f \geq 0 \\ 1 + abs(f_i), & f < 0 \end{cases} \quad (13)$$

f_i değerinin komşu çözümünün, yani kaynak v_i 'nin maliyet değeridir. Açgözlü(greedy) seçim işlemi, mevcut kaynak ile bulunan kaynak arasında uygulanır. Bulunan v_i çözümünün kalitesi mevcut x_i çözümünün kalitesinden yüksek ise eski kaynak bilgisini silerek yeni v_i çözümünü hafızasında saklar ve geliştirmeme sayacını (*failurei*) sıfırlar. Çözümün öncekinden daha niteliksiz olduğu tespit edilirse, eski kaynağa geri döner ve geliştirmeme sayacını bir tane artırır.

3.2.3.5. Kâşif arı aşaması

Besin kaynağı $X_{i,j}$ sınır değeri boyunca daha fazla iyileştirilemezse, işçi arılar yeni besin kaynaklarını bulmak için kâşif arıya dönüşürler. Kaşif arılar aşağıdaki denkleme göre rastgele yeni bir besin kaynağı bulurlar (Zhang ve Gu 2015).

$$X_{i,j} = X_j^{min} + r(X_j^{max} - X_j^{min}) \quad (14)$$

Yeni bir besin kaynağı bulunduğunda YAK algoritması tekrarlanır. Bu yinelemeler, iterasyon sayısına ulaşılan kadar gerçekleştirilir.

3.2.3.6. Temel YAK algoritmasının adımları

YAK algoritmasının temel adımları şu şekilde özetlenebilir (Pan ve ark. 2011).

Adım 1: BS parametresi (besin kaynağı sayısı) ve popülasyon başlatma,

Adım 2: İşçi arıları besin kaynağına gitmesi,

Adım 3: Gözlemci arıları besin kaynağındaki nektar miktarına göre besin kaynağına gitmesi,

Adım 4: Yeni besin kaynakları bulmak için kaşif arılarının aramaya çıkması,

Adım 5: Bulunan en iyi besin kaynağının hafızaya alınması,

Adım 6: Yineleme kriterleri karşılanana kadar algoritmaya devam edilir.



4. TMPMB PROBLEMİNİN ÖNERİLEN YAK ALGORİTMASIYLA ÇÖZÜMÜ

TMPMB problemimizin YAK algoritmasıyla çözümünü yapmak için sıralı işlem tabanlı tamsayı dizisi benimsendi. Çünkü Denklem (11) ve (14) doğrudan problem için yeni çözümler üretmek için kullanılırsa, elde edilen dizi çoğu durumda gerçek olacaktır ve elde edilen çözüm uygulanabilir değildir. Bu probleme, uygulanabilir çözümler üretmek için takas (swap) yöntemi kullanılır. Baskınlık tabanlı açgözlülük ilkesi ve baskınlık tabanlı turnuva benimsenir. Diziyi güncellemek için her belirli döngüde en yüksek sıra değerine sahip çözüm baskın olmayan bir çözümle değiştirilir.

YAK algoritmasının uygulama adımları:

Adım 1: Rassal olarak Öncü(İşçi) arı sayısı kadar öncü besin kaynakları oluştur.

Adım 2: Her bir öncüye takas yöntemiyle Gözcü(Komşu) arı sayısı kadar komşu besin kaynakları oluştur.

Adım 3: Her bir gözcünün C_{max} değerini hesaplayarak öncünün C_{max} değeriyle kıyasla. Eğer küçük ise bu gözcüyü yeni öncü olarak seç, değilse diğer gözcüleri kıyasla.

Adım 4: Yeni seçilen öncüleri birbiriyle kıyasla ve en küçük C_{max} değerine sahip öncüyü hafızaya al.

Adım 5: İterasyon sayısı kadar bu işlemi tekrarla.

Bu problemin adımlarındaki öncü arı, rassal olarak oluşturulmuş iş sırasındır. Her bir arı bulunduğu besin kaynağıyla aynı değeri ifade eder. Yani her bir arı, bir besin kaynağıdır ve dolayısıyla bir iş sırasındır. Şekil.5'deki (a) sırası öncü arıyı ifade eder. Gözcü arı ise öncü arılar kullanılarak takas işlemi sonucunda oluşan iş sırasına denir. Şekil.5'de (d) ve (e) iş sıraları gözcü arıyı ifade eder. Her bir gözcü arı alternatif bir çözüm demektir.

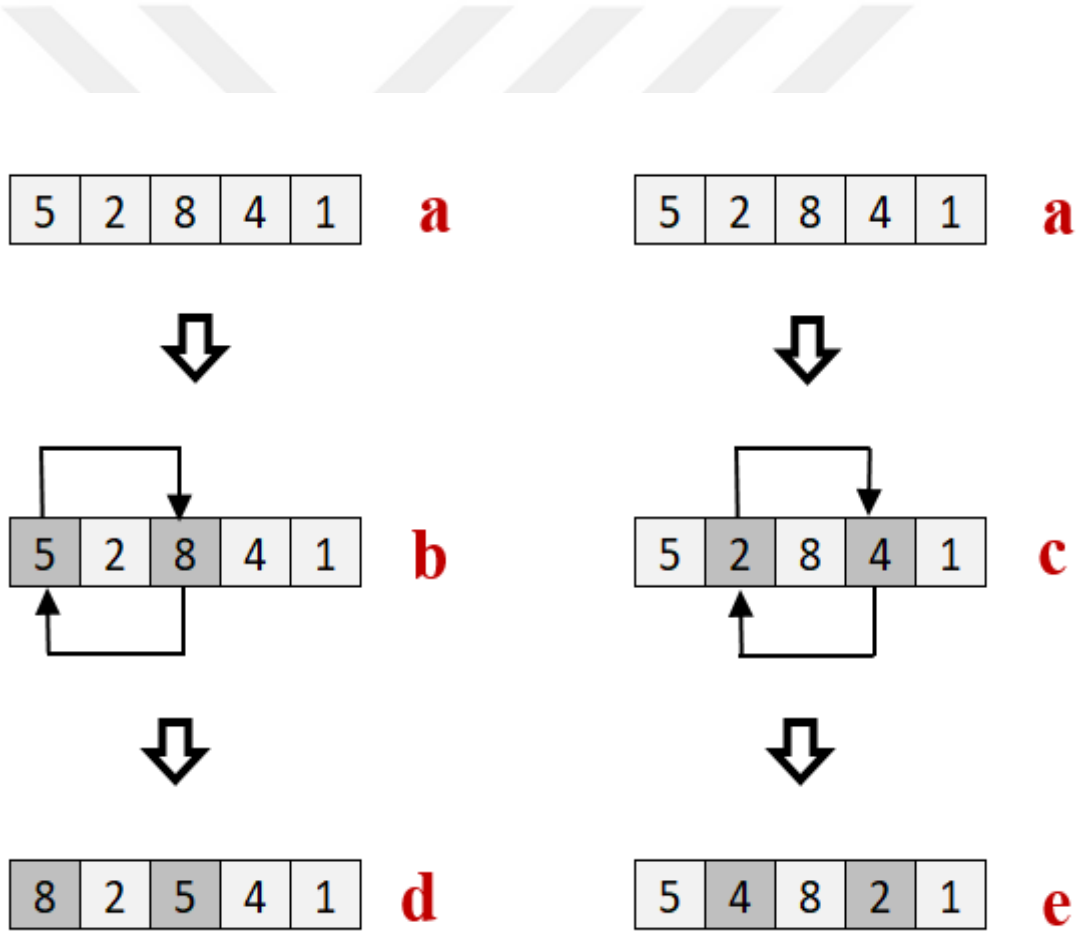
Problemin adımlarındaki C_{max} değeri uygunluk değeridir. C_{max} değeri Denklem(15)'deki verildiği şekilde hesaplanır.

$$C_{max} = T \times \sum_{i=1}^n b_i - maxslack \quad (15)$$

C_{max} değeri ne kadar düşük ise uygunluk değerimiz o kadar yüksek olduğu anlamına gelir. Çünkü problemimizin amaç fonksiyonuna uygun olan C_{max} değerinin düşürülmesidir. Onun için problemimizin amaç fonksiyonu Denklem(16)'da verildiği gibidir.

$$MinC_{max} \quad (16)$$

Takas işlemi: Her bir öncü arıya verilen parametre değeri kadar gözcü arı oluşturur. Gözcü arı, öncü arının sıra diziliminden rassal olarak seçilen iki elemanın takas mutasyonu ile oluşturulur. Şekil.5'de (b) ve (c) bir öncü arıya atanan farklı komşu besin kaynağı oluşturmak için besin kaynağındaki takas mutasyon işlemini göstermektedir. Bir besin kaynağının farklı komşuları, ele alınan problemin yerel olarak aranması açısından önemlidir.



Şekil 5. Öncü arı ve gözcü besin kaynaklarını oluşturmak için bir besin kaynağındaki takas mutasyonları

Örnek Çözüm:

Şekil.5' de verilen arıları (iş sırasını) kullanarak blok (vardiya) süresi (T) 10 saat olan ve gelen işlerin yarıda kesilemeyen işler olan bir işletmenin bu işler için kaç blok gerekli olduğunu ve C_{max} değerinin bulunması:

Verilen öncü arıya göre iş sırası süreleri: 5, 2, 8, 4, 1 yani 1. iş 5 saat 2. iş 2'dir.

Bu işleri atanırsa;

(5+2) 1. blok, boşluk 3 saat

(8) 2. blok, boşluk 2 saat

(4+1) 3. blok, boşluk 5 saat, bu işlerin tamamlanması için toplam 3 blok gerekiyor.

Maxslack=3

$$C_{max} = 10 * 3 - 5 = 25$$

Elde edilen 1. gözcü arıya göre iş sırası süreleri: 8, 2, 5, 4, 1 yani 1. iş 8 saat 2. iş 2'dir.

Bu işleri atanırsa;

(8+2) 1. blok, boşluk yok

(5+4+1) 2. blok, boşluk yok, bu işlerin tamamlanması için toplam 2 blok gerekiyor.

Maxslack=0

$$C_{max} = 10 * 2 - 0 = 20$$

Elde edilen 2 gözcü arıya göre iş sırası süreleri: 5, 4, 8, 2, 1 yani 1. iş 5 saat 2. iş 4'dür.

Bu işleri atanırsa;

(5+4) 1. blok, boşluk 1 saat

(8+2) 2. blok, boşluk 2 saat

(1) 3. blok, boşluk 7 saat, bu işlerin tamamlanması için toplam 3 blok gerekiyor.

Maxslack=9

$$C_{max} = 10 * 3 - 9 = 21$$

Elde edilen gözcü arılardan en iyisi 1. gözcü arıdır. Gözcü arını C_{max} değeri öncü arıdan iyi olduğu için yeni öncü arı 1. gözcü arı seçilir. İterasyon sayısı kadar bu işleme devam edilir.

Yukarıda gösterildiği gibi, YAK algoritması, komşuluk aramasının kullanımı için birkaç parametreye, basit yapıya ve iyi yerel optimizasyon yeteneğine sahiptir; aynı zamanda, baskınlığa dayalı açgözlü seleksiyon ve turnuva seçimi gibi birden çok hedefi aynı anda ele almak için bazı basit ve etkili yöntemler uygular.

Problemimizde bazı baskın olmayan çözümler açgözlü seleksiyonda silinebilir, bu nedenle baskın olmayan baskın olmayan kümesi, problemimiz tarafından üretilen baskın olmayan çözümleri depolamak için kullanılır.

Genellikle, kaşif arılar Denklem (14)'ü kullanarak rastgele arama yaparlar; ancak, ele alınan problem için Denklem (14)'ü kullanılarak elde edilen çözümler çoğu durumda uygulanabilir değildir ve rastgele aramanın çözümleri genellikle iyi değildir. Gözcü arı deyiminde seçilme şansları çok azdır ve bir baskın olmayanlar üyesi ile değiştirilmeleri kolaydır, bu nedenle hiçbir kaşif arı dikkate alınmaz.

Şekil 6' da verilen akış diyagramında da görüldüğü gibi YAK algoritmasında kaşif arı evresi kullanılmamıştır.

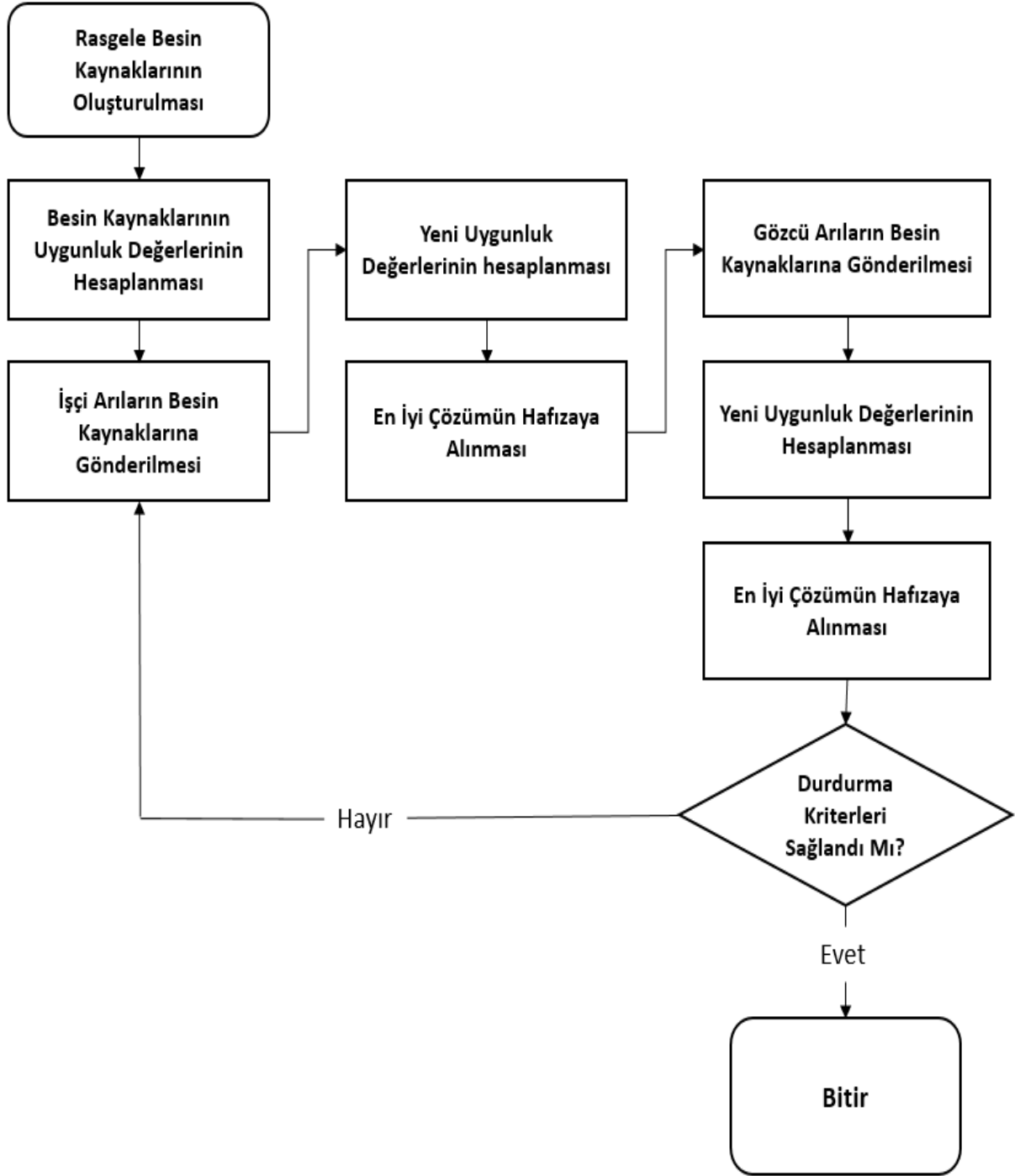
4.1. Örnek Setleri

Bu bölümde TMPMB problemine uyarlanmış olan YAK'nin etkinliği gösterilmesi ve kıyaslanması için Low, Hsu ve ark. (2010) tarafından kullanılan örnek setleri kullanılmıştır. Daha spesifik olarak, $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300\}$ ve $P_j \sim U[1, 50]$ her boyut için 50 örnek (700 örnek) çözüldü. Her bir örnek için, $T \sim U[150, 200]$ ile bir T değeri üretilmiştir. Yani çalışmada, iş sürelerini düzgün dağılıma uygun olarak rassal olarak 1 ile 50 dakika arasında ve bloklar arası süre ise düzgün dağılıma uygun olarak rassal olarak 150 ile 200 dakika arasında üretilmiştir (problem yalnızca $P_j \leq T$, $\forall j = 1, \dots, n$ olduğunda kullanılmalıdır). 14 tane örnek seti ve her örnek seti için 50 tane örnek çözülmüştür. Toplamda 700 örnek çözülmüştür. Daha önce tartışıldığı gibi, sonuçlar üzerinde etkisi olmadığı için t 'nin parametresini dahil edilmediği daha önce belirtmişti.

Kıyaslama için, karşılaştırılan her bir örneğin Bağlı Yüzde Fark (BYF) değerini dikkate alındı.

$$BYF = \frac{C_{max}(METH) - MIN}{MIN} 100 \quad (17)$$

Denklem (17)'de verilen MIN, karşılaştırılan yöntemler için elde edilen optimal veya en iyi bağlı tamamlama zamanı değeridir, METH her bir yöntem için elde edilen tamamlama zamanı değeridir. OBYF ise Ortalama BYF'ye karşılık gelir.



Şekil 6. YAK algoritması akış diyagramı

4.2. Alt Sınırın Hesaplanması

NP-Zor bir çizelgeleme problemi için, optimal çözümlerin elde edilmesi zor olduğundan, literatürde sezgisel algoritmaların performansını ölçmek için kullanılan alt sınırlarla yaygın olarak karşılaşmaktadır. Ek olarak, alt sınır, dal-sınır algoritmaları gibi örtük numaralandırma yöntemlerinin geliştirilmesinde kilit bir rol oynar veya sezgisel algoritmaların

geliştirilmesinde bir indeks görevi görür. Bu nedenle, alt sınırların kesin tahmini önemli bir konudur. Bir problem için Low ve ark.(2010a) sezgisel algoritmaların performansını değerlendirmek için bir alt sınır kullandı. Alt sınır buradaki gibi gösterilir ve şu şekilde tahmin edilir:

$$LB_{Low} = \sum_{j=1}^n P_j + \left(\frac{\sum_{j=1}^n P_j}{T} - 1 \right) \cdot t \quad (18)$$

Denklem(18)' de olan ilk terim toplam işlem süresi, ikinci terim ise minimum toplam bakım süresidir. Bakım süresi olan t 'yi dikkate almadığından alt sınır, toplam işlem süresi olur.

$1 | nr - pm | C_{max}$ problemi için YAK algoritmasının sonuçları daha önceden en iyi çözümler veren NEW, NEW_BF ve NEW_FF (Perez-Gonzalez ve Framian, 2018) üç sezgiseli ile literatürde olan onbeş ve KTDP (Perez-Gonzalez ve Framian, 2018)'in sonuçları verilmiş ve kıyaslanmıştır.

4.3. Algoritma Parametreleri ve Paremetre Optimizasyonu

Parametreler belirlenirken tam faktöriyel deney tasarımıyla parametre optimizasyonu yapılmıştır. Parametre optimizasyonu için kullanılan başlangıç değerleri aşağıda verilmiştir.

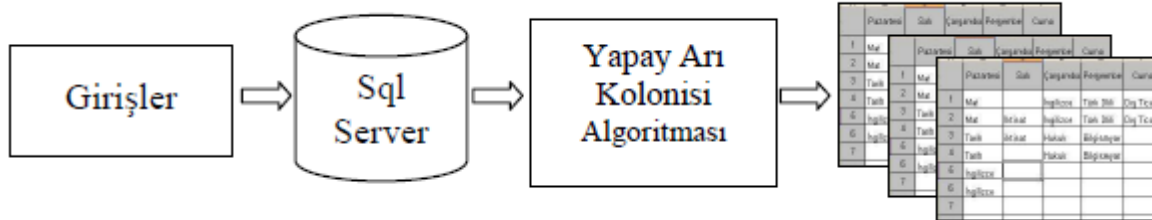
Öncü arı sayısı	: 10, 20, 30, 40, 50, 100
Takipçi arı sayısı	: 10, 20, 30, 50, 100, 300
İterasyon sayısı	: 100, 200, 300, 500, 1000

Tam faktöriyel olarak yapılan parametre optimizasyonu sonucunda TMPMB problemi için en iyi sonuç veren parametre değerleri Çizelge 2'de verilmiştir.

Algoritma C# dilinde yazılmıştır ve Intel Core i5 2.40 GHz 6 GB RAM'e sahip bir bilgisayarda çözülmüştür. Bilgilerin saklandığı ortam için ise Microsoft SQL Server 2008 veritabanı kullanılmıştır. Programın akış şeması Şekil 7'de verilmiştir.

Çizelge 2. Tam faktöriyel parametre optimizasyonu sonuçları

Parametre Optimizasyonu Sonuçları			
İş Seti	Öncü Arı Sayısı	Takipçi Arı Sayısı	İterasyon Sayısı
10	10	10	100
20	30	50	500
30	100	100	1000
40	100	100	1000
50	100	100	1000
60	100	100	1000
70	100	100	1000
80	100	100	1000
90	100	100	1000
100	100	100	1000
150	100	100	1000
200	100	100	1000
250	100	100	1000
300	100	100	1000



Şekil 7. Programın şeması

Programın çalışması neticesinde 10'lu iş grubu için alınan sonuçlar Çizelge 3'de verilmiştir. Çizelgede verilenler içinde; ilk sütunda verin T değerleridir. P değerleri ise işlerin işlem süreleridir. İşlem süreleri verilen işlerin sırası algoritmanın verdiği sıralamadır. Bu sıralama neticesinde her bir 10'lu iş grubunun verilmiş iş sırasına göre C_{max} , boş zaman (waste time), en iyi zaman (best time)/Alt sınır ve BYF değeri hesaplanıp 50 örnek seti için de verilmiştir. BYF değerlerinin ortalaması alınarak OBYF değeri hesaplanıp Çizelge 4.'e yazıldı.

Çizelge 3. 10'lu iş grubu için alınan sonuçlar

T	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	Tamamlama Zamanı	Alt Sınır	Boş Zaman	İşlemci Zamanı	BYF
154	13	45	42	27	27	18	44	34	34	18	302	0	0	13	0
200	49	7	6	39	43	33	23	26	17	18	261	0	0	26	0
162	40	28	47	47	50	42	14	8	48	10	334	0	0	9	0
161	37	35	8	45	36	9	45	14	33	41	303	0	0	6	0
172	43	24	13	43	32	17	50	26	40	22	310	0	0	2	0
188	29	6	44	22	10	41	36	13	13	21	235	0	0	5	0
177	27	20	29	30	14	8	1	48	43	44	264	0	0	2	0
172	19	10	39	30	12	17	2	43	23	10	205	0	0	3	0
155	49	43	42	21	24	12	11	8	7	5	222	0	0	2	0
161	35	10	8	39	44	25	48	21	4	36	270	0	0	2	0
152	23	42	43	14	30	49	14	23	4	35	277	0	0	9	0
182	34	3	43	42	35	25	48	39	38	6	313	0	0	6	0
151	44	41	41	25	35	30	23	12	10	9	270	0	0	2	0
162	9	42	16	47	48	4	22	16	49	13	266	0	0	2	0
187	33	28	50	34	6	36	48	31	14	17	297	0	0	6	0
151	10	13	16	45	48	19	21	40	23	32	267	0	0	2	0
153	7	20	49	5	34	16	22	29	16	15	213	0	0	9	0
195	11	48	3	29	29	29	19	27	11	25	231	0	0	6	0
168	40	39	26	20	13	13	8	2	1	1	163	0	0	2	0
173	3	39	28	27	36	40	45	26	31	3	278	0	0	2	0
165	50	42	5	33	6	29	30	19	37	18	269	0	0	2	0
159	24	48	48	34	5	30	48	7	45	29	318	0	0	2	0
195	45	39	35	33	20	11	9	3	3	1	199	0	0	2	0
160	27	50	26	25	32	46	45	16	18	29	314	0	0	2	0
174	44	50	11	31	38	16	47	17	27	26	307	0	0	2	0
170	16	23	18	35	11	49	18	39	39	28	276	0	0	2	0
182	49	41	35	31	26	25	24	24	19	15	289	0	0	2	0
161	26	12	4	39	19	46	15	28	8	6	203	0	0	2	0
190	42	23	23	20	17	16	13	11	5	2	172	0	0	2	0
155	14	26	30	2	36	22	25	41	43	10	249	0	0	2	0
153	33	32	31	26	21	10	16	11	9	1	190	0	0	2	0
199	26	25	23	21	18	18	17	14	7	2	171	0	0	2	0
199	46	47	29	36	36	5	49	38	42	22	350	0	0	2	0
151	15	30	40	47	19	49	6	4	47	26	283	0	0	6	0
190	46	42	29	27	25	21	24	4	2	1	221	0	0	2	0
186	49	38	34	22	43	45	44	35	14	3	327	0	0	2	0
177	48	38	36	30	25	22	16	15	7	5	242	0	0	2	0
161	18	22	41	42	10	17	11	27	24	6	218	0	0	5	0
177	4	31	34	19	8	42	16	23	26	6	209	0	0	5	0
154	10	21	22	37	16	48	23	17	44	49	287	0	0	6	0
151	44	40	36	31	36	18	15	8	3	2	233	0	0	2	0
190	2	31	22	19	1	3	40	36	36	49	239	0	0	2	0
167	3	44	50	49	21	30	11	11	27	31	277	0	0	2	0
152	11	24	44	3	31	11	28	37	21	18	228	0	0	2	0
171	37	47	39	35	11	2	28	14	1	23	237	0	0	2	0
182	49	46	44	41	2	39	33	29	23	19	325	0	0	2	0
152	47	45	44	16	36	35	25	15	14	5	282	0	0	2	0
152	21	1	6	15	6	22	33	35	13	10	162	0	0	5	0
181	31	23	29	39	34	25	37	35	49	36	338	0	0	5	0
157	45	32	29	41	10	22	45	32	22	36	314	0	0	3	0

Çizelge 4. TMPMB probleminde kullanılan yöntemlerin OBYF değerleri

YÖNTEM	10	20	30	40	50	60	70	80	90	100	150	200	250	300
NEW	0.695	1.258	1.124	1.472	1.219	0.712	0.791	1.254	1.422	1.294	1,308	1,59	1,264	1,967
NEW_BF	0.000	0.035	0.087	0.049	0.085	0.062	0.029	0.037	0.038	0.020	0,008	0,004	0,002	0,001
NEW_FF	0.000	0.035	0.088	0.056	0.096	0.083	0.048	0.061	0.053	0.035	0,009	0,004	0,004	0,002
BFR	4.240	3.232	2.871	1.981	2.255	1.472	1.709	1.538	1.409	1.299	0,921	0,775	0,727	0,696
EİSA	0.651	0.738	0.415	0.265	0.024	0.161	0.107	0.093	0.072	0.057	0,01	0,004	0,004	0,002
BFI	8.996	9.805	9.577	10.655	10.316	10.067	10.273	10.700	10.363	11.278	10,792	11,125	10,859	10,899
BFHILO	3.238	3.451	4.367	3.975	4.043	3.528	3.703	3.628	3.818	3.618	3,534	3,894	3,862	3,574
BFLOHI	3.858	3.539	4.066	4.130	4.558	3.955	3.815	4.187	3.971	3.694	3,576	3,879	3,865	3,529
BFV	5.929	5.168	5.094	5.752	5.578	4.885	5.109	5.233	5.282	5.562	5,332	5,544	5,391	5,348
BFA	1.639	1.450	1.035	0.781	0.621	0.458	0.281	0.305	0.336	0.217	0,109	0,087	0,053	0,034
FFR	4.240	3.274	3.006	2.041	2.289	1.589	1.775	1.620	1.465	1.409	1,044	0,985	0,809	0,796
FFD/MFFD	0.565	0.698	0.415	0.278	0.239	0.176	0.113	0.098	0.075	0.060	0,011	0,006	0,005	0,002
FFI	8.996	9.805	9.577	10.655	10.316	10.067	10.273	10.700	10.363	11.278	10,792	11,125	10,859	10,899
FFHILO	3.238	3.451	4.297	4.064	4.043	3.528	3.754	3.651	3.843	3.626	3,58	3,922	3,902	3,587
FFLOHI	3.858	3.689	4.066	4.130	4.637	4.014	3.815	4.282	4.035	3.680	3,609	3,867	3,888	3,542
FFV	5.929	5.168	5.094	5.884	5.578	4.885	5.109	5.233	5.282	5.562	5,332	5,544	5,391	5,348
FFA	1.639	1.446	1.038	0.803	0.618	0.472	0.308	0.314	0.343	0.231	0,118	0,096	0,054	0,037
KTDP	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,002	0,006	0,037	0,043	0,085
YAK	0,000	0,000	0,000	0,116	0,404	0,702	1,106	1,336	1,657	1,814	2,621	3,247	3,615	3,942

Bu sonuçların neticesinde YAK algoritmasının performansı hemen sezgisellerden daha iyi ve öne çıkan sezgisellerden olan EİSA, FFD, MFFD ve NEW yöntemlerinden küçük ve orta boyuttaki örnek setleri için daha iyi sonuçlar vermiştir. Küçük boyuttaki örnek setleri NEW_BF, NEW_FF ve KTDP kadar iyi sonuçlara ulaşıldı. Fakat büyük boyutlu örnek setleri için optimum sonuçlara ulaşılmadı. Bu problemi ortadan kaldırmak için aşağıda verilen hibrit bir YAK algoritması önerilmiştir.

4.4. TMPMB Problemine İYAK Algoritmasıyla Çözümü

TMPMB problemimizin YAK algoritmasıyla çözümünde küçük boyutlu işlerde iyi sonuçlar alınmasına rağmen büyük boyutlu işler için kötü sonuçlar verdi. Bunun için YAK algoritmasına sonuçlarıyla kıyaslanan sezgiseller içinde en iyi sonuç veren NEW_BF'nin EİSA algoritması eklenerek hibrit bir İyileştirilmiş Yapay Arı Kolonisi (İYAK) algoritması elde edildi. İYAK algoritması da aynı YAK algoritmasındaki adımlar korunup ek olarak EİSA algoritmasının adımları eklendi. Yani bir öncüyü büyükten küçüğe sıralıyoruz. Sonra sırayla gelen işleri mevcut bloklardan sığabilecek ilk en büyük yüke sahip bloğa atıyoruz Yoksa yeni

blok açılır. Önceki hesaplama deneylerindeki örnek setlerindeki değerlerin rassal seçiminden gelen kıyaslamadaki problemlerden dolayı İYAK algoritmasını ve NEW_BF'yi aynı beş iş veri seti için çalıştırılıp sonuçlar kıyaslanmıştır.

İYAK algoritmasının uygulama adımları:

Adım 1: Rassal olarak öncü arı sayısı kadar öncü besin kaynakları oluştur.

Adım 2: Öncü besin kaynaklarından sadece bir tanesinin iş sırasına EUISGS uygula.

Adım 3: Her bir öncüye takas yöntemiyle gözcü arı sayısı kadar komşu besin kaynakları oluştur.

Adım 4: Herbir öncüye BF uygula ve C_{max} değerini hesapla

Adım 5: Her bir gözcüye BF uygula ve C_{max} değerini hesaplayarak öncünün C_{max} değeriyle kıyasla. Eğer küçük ise bu gözcüyü yeni öncü olarak seç, değilse diğer gözcüleri kıyasla.

Adım 6: Yeni seçilen öncüleri birbiriyle kıyasla ve en küçük C_{max} değerine sahip öncüyü hafızaya al.

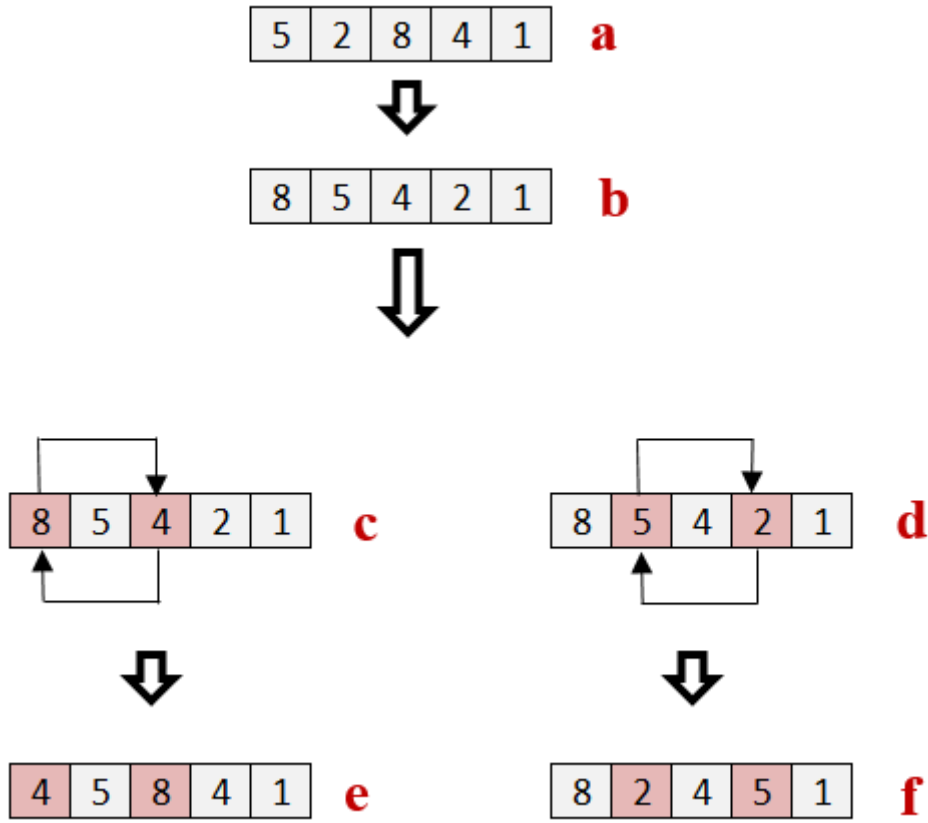
Adım 7: İterasyon sayısı kadar bu işlemi tekrarla.

Bu algoritmanın adımlarındaki BF, en iyi sığıdır sezgiselidir. BF'nin çalışması şöyledir: Sırasına göre gelen işleri bloklara atanırken mevcut olan bloklardan yükü en çok olup sığabildiği ilk bloka atanmasıdır. Eğer uygun blok yoksa yeni blok açılıp, oraya atanır. Diğer ifadeler önceki bölümde tanımlanmıştır.

4.5. İYAK Algoritması için Örnek Çözüm

Şekil.8'de verilen örnekteki arıları(iş sırasını) kullanarak İYAK algoritmasıyla çözüp blok süresi(T) 10 saat olan ve gelen işlerin yarıda kesilemeyen işler olan bir işletmenin bu işler için kaç blok gerekli olduğunu ve C_{max} değerinin bulunması:

Şekil.8'deki (a) sırası öncü arıyı ifade eder. Gözcü arı ise öncü arılar kullanılarak takas işlemi(Şekil.8c ve Şekil.8d) sonucunda oluşan iş sırasına denir. Şekil.8'de (e) ve (f) iş sıraları gözcü arıyı ifade eder. Her bir gözcü arı alternatif bir çözüm demektir. EUISGS, işlerin büyükten küçüğe doğru sıralanmasıdır (Şekil.8b).



Şekil 8. Öncü arı ve Komşu besin kaynaklarını oluşturmak için takas mutasyonları ve EUİSGS uygulama

Verilen öncü arıya göre iş sırası süreleri: 5, 2, 8, 4, 1 yani 1. iş 5 saat 2. iş 2'dir.

EUİSGS uygulanınca öncü arının iş sırası: 8, 5, 4, 2, 1 olur.

Bu işler BF ile atanırsa;

(8+2) 1. blok, boşluk 3 saat

(5+4+1) 2. blok, boşluk yok. Bu işlerin tamamlanması için toplam 2 blok gerekiyor.

Maxslack=0

$C_{\max}=10*2-0=20$

Elde edilen 1. gözcü arıya göre iş sırası süreleri: 4, 5, 8, 2, 1 yani 1. iş 4 saat 2. iş 5'dir.

Bu işler BF ile atanırsa;

(4+5+1) 1. blok, boşluk yok

(8+2) 2. blok, boşluk yok, bu işlerin tamamlanması için toplam 2 blok gerekiyor.

Maxslack=0

$C_{\max}=10*2-0=20$

Elde edilen 2. gözcü arıya göre iş sırası süreleri: 8, 2, 4, 5, 1 yani 1. iş 8 saat 2. iş 2'dir.

Bu işleri atanırsa;

(8+2), 1. blok, boşluk 1 saat,

(4+5+1), 2. blok, boşluk yok, bu işlerin tamamlanması için toplam 2 blok gerekiyor.

Maxslack=0

$C_{\max}=10*2-0=20$

Elde edilen sonuçlara göre hem öncü arı hem de gözcü arılar optimal C_{\max} değerini buldular. Onun için öncü arıyı değiştirmeye ve başka iterasyon çalıştırmaya gerek kalmadı. Minimum $C_{\max}=20$ olur.

4.6. Örnek Setleri

Bu bölümde TMPMB problemine uyarlanmış olan İYAK'nin etkinliği gösterilmesi ve kıyaslanması için Low, Hsu ve ark. (2010) tarafından kullanılan örnek setleri kullanılmıştır. Daha spesifik olarak, $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300\}$ ve $P_j \sim U[1, 50]$ her boyut için 5 örnek (70 örnek) çözüldü. Her bir örnek için, $T \sim U[150, 200]$ ile bir T değeri üretilmiştir. Yani çalışmada, iş sürelerini düzgün dağılıma uygun olarak rassal olarak 1 ile 50 dakika arasında ve bloklar arası süre ise düzgün dağılıma uygun olarak rassal olarak 150 ile 200 dakika arasında üretilmiştir (problem yalnızca $P_j \leq T, \forall j = 1, \dots, n$ olduğunda kullanılmalıdır). 14 tane örnek seti ve her örnek seti için 5 tane örnek problem çözülmüştür. Toplamda 70 örnek çözülmüştür. Daha önce tartışıldığı gibi, sonuçlar üzerinde etkisi olmadığı için t 'nin parametresini dahil edilmediği daha önce belirtildi.

Karşılaştırma için, kıyaslanan her bir örneğin yine Denklem(17)'deki BYF değerleri dikkate alındı.

Alt Sınırın hesaplanması için yine Denklem(18) kullanıldı.

$1 | nr - pm | C_{\max}$ problemi için İYAK algoritmasının sonuçlarını daha önceden en iyi olan NEW_BF(Gonzalez ve Framian,2018) sezgiseli sonuçlarıyla verilir kıyaslandı.

4.7. Algoritma Parametreleri ve Paremetre Optimizasyonu

Parametreler belirlenirken tam faktöriyel parametre optimizasyonu yapılmıştır. Parametre optimizasyonu için kullanılan başlangıç değerleri aşağıda verilmiştir.

Öncü(İşçi) arı sayısı : 10, 20, 30, 40, 50, 60, 70, 80, 90, 100

Takipçi(Gözlemci) arı sayısı : 10, 50, 100, 200, 300, 400, 500

İterasyon sayısı : 100

Kullanılan başlangıç değerleri sonucunda problemin iyi sonuç vermesi için her bir örnek seti için farklı parametreler seçilip denenmiş olup sonuçları Çizelge 5’de verilmiştir.

Çizelge 5. Tam faktöriyel parametre optimizasyonu sonuçları

Parametre Optimizasyonu Sonuçları			
İş Seti	Öncü Arı Sayısı	Takipçi Arı Sayısı	İterasyon Sayısı
10	10	10	100
20	10	10	100
30	20	10	100
40	30	200	100
50	50	500	100
60	20	400	100
70	40	400	100
80	40	10	100
90	40	100	100
100	100	200	100
150	100	200	100
200	100	200	100
250	100	200	100
300	100	200	100

Parametre optimizasyonu sonucunda 60’lı iş setine kadar optimal sonucu bulan parametre değerleri verildi. 70, 80 ve 90’lı örnek setleri için en küçük BYF değerleri veren parametreler kullanıldı. Bu parametreler 90’dan daha büyük örnek setleri için optimumu bulmuyor ama yüksek işlemci süresinden dolayı optimuma yakın sonuçlar için bu parametreler seçilmiştir.

Programın çalışması neticesinde 10’lu iş grubu için alınan sonuçlar Çizelge 6.’da verilmiştir. Çizelgede verilenler içinde; ilk sütunda verin T değerleridir. P değerleri ise işlerin işlem süreleridir. İşlem süreleri verilen işlerin sırası algoritmanın verdiği sıralamadır. Bu sıralama neticesinde her bir 10’lu iş grubunun verilmiş iş sırasına göre C_{max} , boş aman, alt sınır ve BYF değeri hesaplanıp 5 örnek seti için de verilmiştir. BYF değerlerinin ortalaması alınarak OBYF değeri hesaplanıp Çizelge 7’de gösterilmiştir.

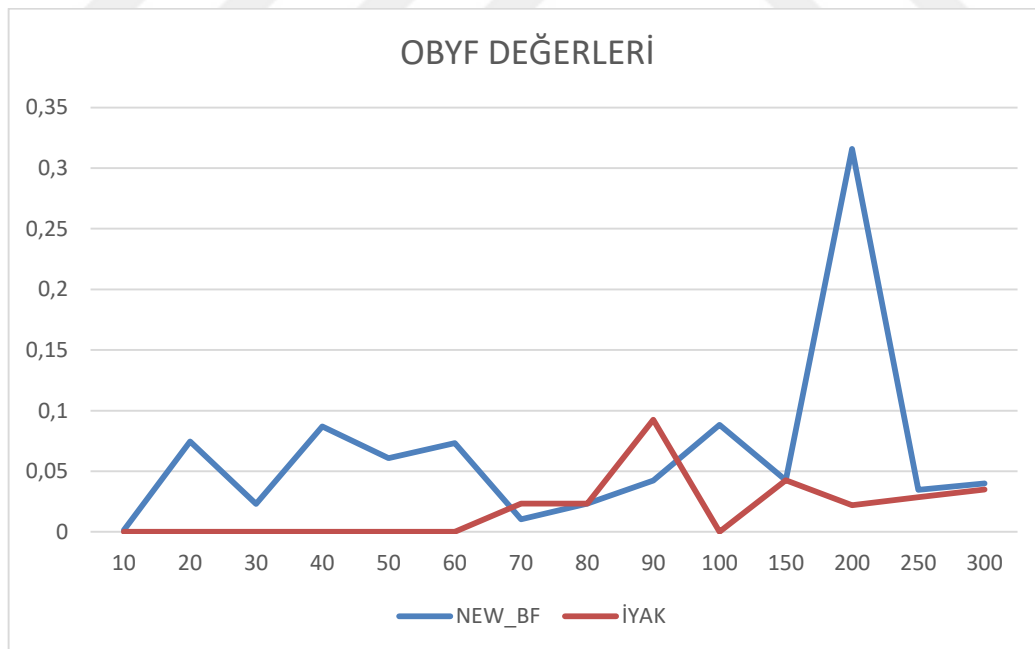
TMPMB probleminde en iyi sonuç veren NEW_BF ve İYAK’ın OBYF ve işlemci değerleri Çizelge 6.’daki beş setin çözümü için verildi. Şekil 9.’da NEW_BF ve İYAK’ın karşılaştırmalı OBYF değerlerinin grafiği, Şekil 10.’da ise NEW_BF ve İYAK’ın karşılaştırmalı işlemci değerlerinin grafiği verilmiştir.

Çizelge 6. 10'lu iş grubu için alınan sonuçlar

T	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	C_{max}	İşlemci Zamanı	Boş Zaman	Alt Sınır	BYF
154	18	45	44	34	13	34	18	27	27	42	302	215	0	302	0
200	7	49	23	17	39	26	33	6	18	43	261	213	0	261	0
162	28	47	47	40	48	50	14	42	8	10	334	436	0	334	0
161	8	37	35	45	36	45	9	41	14	33	303	199	0	303	0
172	50	26	13	43	40	43	22	32	17	24	310	165	0	310	0

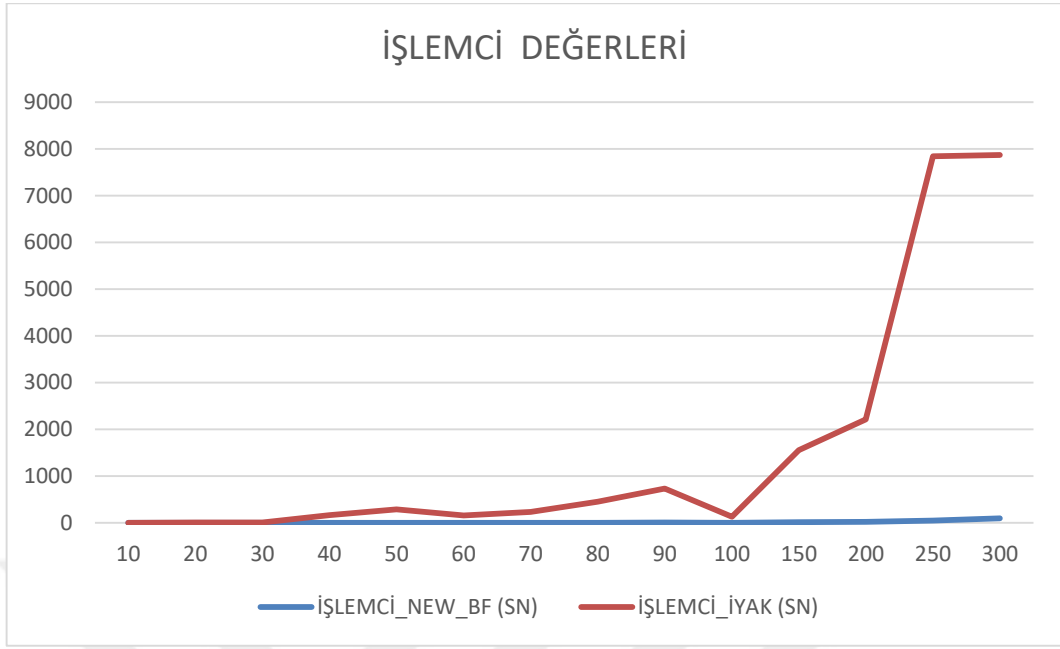
Çizelge 7. NEW_BF ve İYAK algoritmasının OBYF ve işlemci değerleri

YÖNTEM	10	20	30	40	50	60	70	80	90	100	150	200	250	300
NEW_BF	0,001	0,075	0,023	0,087	0,061	0,073	0,011	0,023	0,042	0,088	0,043	0,316	0,035	0,04
İŞLEMÇİ NEW_BF (SN)	0,004	0,01	0,027	0,082	0,288	0,344	0,318	0,614	1,555	0,572	8,237	14,62	42,31	92,22
İYAK	0	0	0	0	0	0	0,023	0,023	0,092	0	0,043	0,022	0,029	0,035
İŞLEMÇİ İYAK (SN)	0,206	0,864	4,507	159	282,2	152,5	231,4	452,8	731,6	127,1	1556	2213	7845	7869



Şekil 9. NEW_BF ve İYAK algoritmasının OBYF değerleri

Grafik incelendiğinde TMPMB problemi için önerilen İYAK algoritması 70'li ve 90'lı iş grubu dışındaki kalan iş grupları aynı ya da daha iyi sonuçlar vermiştir.



Şekil 10. NEW_BF ve İYAK algoritmasının işlemci değerleri

Grafiğe göre İYAK algoritması işlemci (CPU) değeri olarak 100'lü iş grubuna kadar NEW_BF'ye yakın değerlere sahip fakat daha büyük iş grupları için çok daha büyük değerlere ulaşıldı. Bunun sebebi, algoritmanın yapısından kaynaklı bir durumdadır. Çünkü İYAK algoritması bir iterasyonda verilen parametreye göre bazen bir anda binlerce iş sırası(gözcü arı ve öncü arı) üretip bunları deneyip karşılaştırarak çözümü bulmaya çalışır.

5. SONUÇ VE TARTIŞMA

5.1. Sonuç

Bu tez çalışmasında tek makinalı periyodik makine kullanılabilirlik kısıtıyla makine bakım probleminin çözümünde 14 tane örnek grubu ve her örnek grubu için 50 tane örnek set el alındı. Toplamda 700 örnek YAK algoritmasıyla çözüldü ve literatürdeki son yıllarda geliştirilen yöntemler ile kıyaslandı. Bu yöntemlerden öne çıkan NEW, NEW_BF, NEW_FF ve literatürde çözümleri olan sezgisellerle karşılaştırıldı. Bu çalışmada verilen yöntemlere alternatif olacak güçlü bir algoritmaya sahip YAK algoritması daha önceden kullanılan örnek setleriyle çözüme gidildi. Önerilerin verimliliğini değerlendirmek için hesaplama değerlendirmesi yapıldı.

Test edilen örnek veri setlerinden küçük boyutlu olan problemlerde optimum, fakat büyük boyutlu problemlerde optimumdan daha uzak sonuçlara ulaşıldı. Bu olumsuzluğu ortadan kaldırmak için ilk olarak büyük boyutlu problemler için öncü ve gözlemci arılarının sayısını artırmak düşünülebilirdi. Ancak bu da yüksek işlemci değerlerine neden olur. Bunun için YAK algoritmasına sonuçlarıyla kıyaslanan sezgiseller içinde en iyi sonuç veren NEW_BF'nin EİSA algoritması eklenerek hibrit bir İYAK algoritması elde edildi. İYAK algoritması da aynı YAK algoritmasındaki adımlar korunup ek olarak EİSA algoritmasının adımları eklendi. Önceki hesaplama deneylerindeki örnek setlerindeki değerlerin rassal seçiminden gelen kıyaslamadaki problemlerden dolayı İYAK algoritmasını ve NEW_BF'yi aynı beş iş veri seti için çalıştırılıp sonuçlar karşılaştırıldı.

Problemin sonucunda OBYF değerleri küçük boyutlu problemlerden 60'lı iş grubuna kadar olan işlerde %0,0 ve büyük boyutlu problemlerde ise %0.10'dan daha küçük değerler bulundu. Bunun yanında, problemde verilen parametreler büyüdükçe küçük boyutlu problemlerde diğer yöntemlere göre optimuma daha fazla yaklaştığı görülmektedir. Literatürde yer alan çalışmalardan en iyi sonuç veren NEW_BF ile aynı veri seti kullanılarak karşılaştırıldığında küçük boyutlu problemlerde çok daha iyi sonuçlar elde edildi. Büyük boyutlu problemlerde ise muadil bazen daha iyi sonuçlara ulaşıldı.

Ayrıca 10, 20, 30'dan 300'e kadar olan tüm iş gruplarının ortalama OBYF değerlerine bakıldığında NEW_BF ve İYAK algoritması için sırasıyla; 0,066 ve 0,019 vermiştir. Bu sonuçları göre yine İYAK algoritmasının ne kadar üstün bir yöntem olduğunu bize göstermektedir.

5.2. Tartışma

Gelecekteki arařtırmalar için önerilen yöntemde kullanılan İYAK algoritmasıyla beraber farklı metasezgisel yöntemler birleřtirilerek farklı hibrit yöntemler de geliřtirilebilir. Periyodik bakım ile zamanlama problemlerinin pratik ilgi ve kullanıřlılıđından dolayı, bu sorunu akıř planı veya paralel makineler gibi diđer yerleřme düzenlerinde incelemek yararlı olabilir. Ayrıca, literatürde, tamamlama zamanı farklı amaçlar için dikkate alınarak problem yeniden çözümlü sonuçlar karşılařtırıla bilir.



KAYNAKLAR

- Alatas B., 2010, Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, 37(8), 5682-5687.
- Angel-Bello, F., Alvarez, A., Pacheco, J., and Martínez, I., 2011, A heuristic approach for a scheduling problem with periodic maintenance and sequence-dependent setup times. *Computers & Mathematics with Applications*, 61(4), 797–808.
- Adiri, I., Frostig, E., and Rinnooy Kan, A. G. H. 1991. Single machine flow-time scheduling with a single breakdown to minimize stochastically the number of tardy jobs. *Naval Research Logistics*, 38, 261–271.
- Blanchard, B. S., Verm, D. and Peterson, E. L., 1995, Maintainability: A key to effective and maintenance management. NY: *John Wiley & Sons*, New York.
- Brajevic I. and Tuba M., 2013, An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems. *Journal of Intelligent Manufacturing*, 24, 729-740.
- Chen, J.S., 2006a, Minimizing total flow time in the single-machine scheduling problem with periodic maintenance. *Journal of the Operational Research Society*, 57(4), 410-415.
- Chen, J.S., 2006b, Optimization models for the machine scheduling problem with a single flexible maintenance activity. *Engineering Optimization*, 38(1), 55-71.
- Chen, J.S., 2008, Scheduling of nonresumable jobs and flexible maintenance activities on a single machine to minimize makespan. *European Journal of Operational Research*, 190(1), 90-102.
- Choong S.S. and Wong L.P., Lim C.P., 2019, An artificial bee colony algorithm with a Modified Choice Function for the traveling salesman problem. *Swarm and Evolutionary Computation* 44, 622-635
- Cowling, P. and Johansson, M., 2002, Using real time information for effective dynamic scheduling. *European Journal of Operational Research*, 139, 230-244.
- Cui Z, Gu X., 2013, An Improved Discrete Artificial Bee Colony Algorithm for Hybrid Flow Shop Problems. In: *Intelligent Computing for Sustainable Energy and Environment: Second International Conference*, 355, 294–302
- Cui, W.W. and Lu, Z., 2017, Minimizing the makespan on a single machine with flexible maintenances and jobs' release dates. *Computers & Operations Research*, 80, 11-22.
- Deng, G., Xu, Z. and 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.
- Duffua, S. O., Ben-Daya, M., Al-Sultan, K. S. and Andijani, A. A., 2001, A generic conceptual simulation model for maintenance systems. *Journal of Quality in Maintenance Engineering*, 7(3), 207-219.
- Dündar, D. R., Sarıçiçek, İ., Yazıcı, A., (2021), Machine Scheduling with Maintenance Activities: Literature Review. *Uludağ University Journal of The Faculty of Engineering* 26(2), 737-756.
- Engin, O., Fırlalı, A., 2002, Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma Yardımı İle Çözümünde Uygun Çaprazlama Operatörlerinin Belirlenmesi, *Doğuş Üniversitesi Dergisi*, 6, 27-35.
- Gonzalez P. P. and Framian J. M., 2018, Single machine scheduling with periodic machine availability. *Computers & Industrial Engineering*, 123, 180-188.
- Gerstbakh, I. B., 1977, Models of preventive maintenance. New York-Oxford: *North-Holland Publishing Company*.

- Ganggang Li, Xiwen Lu., 2015, Two-machine scheduling with periodic availability constraints to minimize makespan. *Journal of Industrial & Management Optimization*, 11 (2) : 685-700.
- Grüter, C. and Farina W. M., 2009, The honeybee waggle dance: can we follow the steps? *Trends in Ecology & Evolution*, 24(5),242-247.
- Han Y-Y, Duan J-H, Yang Y-J, Zhang M, Yun B., 2012, Minimizing the Total Flowtime Flowshop with Blocking Using a Discrete Artificial Bee Colony. In: *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence: 7th International Conference, ICIC 2011, Zhengzhou, China, August 11-14, 2011, Revised Selected Papers*. Eds: Huang D-S, Gan Y, Gupta P, Gromiha MM. Berlin, Heidelberg: Springer Berlin Heidelberg, p. 91-7.
- Han Y-Y, Liang JJ, Pan Q-K, Li J-Q, Sang H-Y, Cao NN., 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.
- Han Y-Y, Pan Q-K, Li J-Q, Sang H-y., 2012, An improved artificial bee colony algorithm for the blocking flowshop scheduling problem. *The International Journal of Advanced Manufacturing Technology*, 60, 9, 1149-59.
- Hendel Y., Sourd F., 2007, An improved earliness–tardiness timing algorithm, *Computers & Operations Research*, 34, 2931-2938.
- Hsu, C.-J., Low, C., & Su, C.-T., 2010, A single-machine scheduling problem with maintenance activities to minimize makespan. *Applied Mathematics and Computation*, 215(11), 3929–3935.
- Huang S., Zhou H., Tsai Y.-C., Chen Y., Chou F.-D., 2019, Minimizing the Total Weighted Completion Time of a Single Machine With Flexible Maintenance, *Journals & Magazines* , 122164 – 122182
- Ji M., He Y., Cheng T.C.E., 2006, Scheduling linear deteriorating jobs with an availability constraint on a single machine, *Theoretical Computer Science*, 362, 1-3, 115-126.
- Ji, M., He, Y., Cheng, T.C.E., 2007, Single-machine scheduling with periodic maintenance to minimize makespan. *Computers & Operations Research* 34(6),1764-1770.
- Köksal, M., 2007, *Maintenance Planning*. Seçkin, Ankara.
- Karaboga, D., 2005, An idea based on honey bee swarm for numerical optimization. *Technical Report, Computer Engineering Department*, Erciyes University, Turkey.
- 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.
- Krim, H., Benmansour, R., Duvivier, D., Artiba, A., (2019), A variable neighborhood search algorithm for solving the single machine scheduling problem with periodic maintenance. *RAIRO Operations Research*, 53, 289-302.
- Lee C.-Y. ve Liman S. D., 1992, Single machine flow-time scheduling with scheduled maintenance, *Acta Informatica*, 29, 375–382
- Lee, C.-Y., 1996, Machine scheduling with an availability constraint. *Journal of Global Optimization*, 9(3–4), 395–416.
- Lei, D., 2013, Multi-objective artificial bee colony for interval job shop scheduling with flexible maintenance. *Int J Adv Manuf Technol* 66, 1835–1843.
- Lei D. and Liu M., 2020, An artificial bee colony with division for distributed unrelated parallel machine scheduling with preventive maintenance, 141, 106320.
- Li G. and Lu X., (2016), Approximation algorithms for the single-machine scheduling with a period of maintenance, *Optimization Letters*, 10, 543–562

- Li W., Ding Y., Yang Y., Sherratt R. S., Park J. H. and Wang J., 2020, Parameterized algorithms of fundamental NP-hard problems: a survey, *Human-centric Computing and Information Sciences*, 10, Article number: 29
- Liao, C.J. and Chen, W.J., 2003, Single-machine scheduling with periodic maintenance and nonre-sumable jobs. *Computers & Operations Research* 30(9),1335-1347.
- Liu M., Chu C., Xu Y., Wang L., 2010, Online Scheduling on Two Uniform Machines to Minimize the Makespan with a Periodic Availability Constraint. *Algorithmic Aspects in Information and Management*, 6124, 191-198.
- Li, X. and 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.
- Liu, Y.-F. and Liu, S.-Y., 2013, A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem, *Applied Soft Computing*, 13 (3), 1459-1463.
- Low, C., Ji, M., Hsu, C.-j. and Su, C.-t., 2010a, Minimizing the makespan in a single machine scheduling problems with flexible and periodic maintenance. *Applied Mathematical Modelling*, 34(2), 334–342.
- Low, C., Hsu, C.-J., and Su, C.-T., 2010b) A modified particle swarm optimization algorithm for a single-machine scheduling problem with periodic maintenance. *Expert Systems with Applications*, 37(9), 6429–6434.
- Ma. Y., Chu, C., and Zuo, C., 2009, A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering*, 58(2), 199–211.
- Mashkani, O. and Moslehi G., 2016, Minimising the total completion time in a single machine scheduling problem under bimodal flexible periodic availability constraints. *International Journal of Computer Integrated Manufacturing* 29(3): 323-341
- Moslehi G., Mashkani O., 2018, Minimizing the Number of Tardy Jobs in the Single Machine Scheduling Problem Under Bimodal Flexible and Periodic Availability Constraints. *IJIEPR*. 29 (1) :15-34.
- Nesello, V., Subramanian, A., Battarra, M., Laporte, G., 2018, Exact solution of the single-machine scheduling problem with periodic maintenances and sequence-dependent setup times. *European Journal of Operational Research*, 266(2), 498-507.
- Ozcan S.,Simsir F., 2019, A new model based on Artificial Bee Colony algorithm for preventive maintenance with replacement scheduling in continuous production lines, *Engineering Science and Technology, an International Journal*, 22, 1175-1186
- Pacheco, J., Angel-Bello, F. and Alvarez, A., 2012, A multi-start tabu search method for a single-machine scheduling problem with periodic maintenance and sequence-dependent set-up times. *Journal of Scheduling*, 16(6), 661–673.
- Pan, J.-Q., Zou, W.-Q. and Duan, J.-H., 2018, A Discrete Artificial Bee Colony for Distributed Permutation Flowshop Scheduling Problem with Total Flow Time Minimization, *IEEE*, 1934-1768
- Pan, Q.-K., Fatih Tasgetiren, M., Suganthan, P. N. and Chua, T. J., 2011, A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem, *Information Sciences*, 181 (12), 2455-2468.
- Prata, B.A., de Abreu, L.R. and Lima, J.Y.F., 2021, Heuristic methods for the single-machine scheduling problem with periodical resource constraints. *TOP*, 29, 524–546.
- Qamhan, A.A., Ahmed, A., Al-Harkan, I.M., Badwelan, A., Al-Samhan, A.M. and Hidri, L., 2020, An exact method and ant colony optimization for single machine scheduling problem with time window periodic maintenance. *IEEE*, (8), 44836-44845.

- Ribas, I., Companys, R. and 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.
- Sadfi, C. , Penz, B. , Rapine, C. , Blazewicz, J. , Formanowicz, P. 2005, An improved approximation algorithm for the single machine total completion time scheduling problem with availability constraints. *European Journal of Operational Research*, 161(1), 3-10.
- Sanlaville, E. and Schmidt G., 1998, Machine scheduling with availability constraints. *Acta Informatica volume 35*,795–811
- Schmidt G., 2000, Scheduling with limited machine availability. *European Journal of Operational Research*, 121(1), 1-15.
- Shen J., Zhu K., (2018), An uncertain single machine scheduling problem with periodic maintenance, *Knowledge-Based Systems*, 144, 32-41.
- Sourd, F., 2008, New Exact Algorithms for One-Machine Earliness-Tardiness Scheduling. *Infoms Journal on Computing*, 21,1.
- Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N. and 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. and 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.
- Vieira G. E., Herrmann J. W. and Lin E., 2010, Analytical models to predict the performance of a single-machine system under periodic and event-driven rescheduling strategies. *Journal International Journal of Production Research*, 38(8),1899-1915.
- Vijaychakaravarthy G, Marimuthu S, 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, 4285-300.
- Wang J.-B. and Xia Z.-Q., 2005, Scheduling jobs under decreasing linear deterioration, *Information Processing Letters*, 94(2), 63-69.
- Wang T., Baldacci R., Lima A., Hua Q., 2018, A branch-and-price algorithm for scheduling of deteriorating jobs and flexible periodic maintenance on a single machine, *European Journal of Operational Research*, 271(3), 826-838
- Wu, C. C. and Lee, W. C., 2003, Scheduling linear deteriorating jobs to minimize makespan with an availability constraint on a single machine. *Information Processing Letters*,87(2), 89-93.
- Xu D., Sun K. and Li H., 2008, Parallel machine scheduling with almost periodic maintenance and non-preemptive jobs to minimize makespan. *Computers and Operations Research*, 35, 344–1349.
- Xu D., Cheng Z., Yin Y., Li H., 2009a, Makespan minimization for two parallel machines scheduling with a periodic availability constraint, *Computers & Operations Research*, 36(6), 1809-1812.
- Xu, D., Sun, K. and H. Li, 2009b, A note on scheduling of nonresumable jobs and flexible maintenance activities on a single-machine to minimize makespan, *European Journal of Operations Research* 197, 825–827
- Xu D. And Qu M., 2010, Makespan Minimization for Two Parallel Machines Scheduling with a Periodic Availability Constraint: The Preemptive Offline Version. *Third International Joint Conference on Computational Science and Optimization 2*: 177-181

- Xu D., Yang D. L., 2013, Makespan minimization for two parallel machines scheduling with a periodic availability constraint: Mathematical programming model, average-case analysis, and anomalies. *Applied Mathematical Modelling*, 37(14-15):7561-7567.
- Xu Z., Xu D., He J., Wang Q., Liu A. VE Xiao J., (2018), Mixed Integer Programming Formulations for Two-Machine Flow Shop Scheduling with an Availability Constraint, *Arabian Journal for Science and Engineering*, 43, 777–788
- Yang, D. L., Hung, C. L., Hsu, C. J., and Chern, M. S., 2002, Minimizing the makespan in a single machine scheduling problem with a flexible maintenance. *Journal of the Chinese Institute of Industrial Engineers*, 19, 63–66.
- Yong, H., Ji, M., & Cheng, T. C. E. 2005. Single machine scheduling with a restricted rate-modifying activity. *Naval Research Logistics*, 52, 361–369.
- Yu, X., Zhang, Y., & Steiner, G., 2014, Single-machine scheduling with periodic maintenance to minimize makespan revisited. *Journal of Scheduling*, 17(3), 263–270.
- Yu, X., Zhang, Y., Xu, D. and Yin, Y., 2013, Single machine scheduling problem with two synergetic agents and piece-rate maintenance. *Applied Mathematical Modelling*, 37(3), 1390-1399.
- Zhang S-J. and 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, 3471-84.
- Zhou, H., Tsai, Y.-C., Wu, F.-C. and Huang S., 2020, Improved Approaches to Minimize the Makespan on Single-Machine Scheduling with Periodic Preventive Maintenance Activities. *Mathematical Problems in Engineering*, 1-13.