

**T.C.
PAMUKKALE ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI**

DOKTORA TEZİ

**SIRA BAĞIMLI İLİŞKİSİZ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ
İÇİN YENİ BİR SEZGİSEL ALGORİTMA ÖNERİSİ**

**Hazırlayan
Günay KILIÇ**

**Danışman
Prof. Dr. Arzu ORGAN**

**DENİZLİ
Ocak- 2023**

**SIRA BAĞIMLI İLİŞKİSİZ PARALEL MAKİNE ÇİZELGELEME
PROBLEMİ İÇİN YENİ BİR SEZGİSEL ALGORİTMA ÖNERİSİ**

**Pamukkale Üniversitesi
Sosyal Bilimler Enstitüsü
Doktora Tezi
İşletme Ana Bilim Dalı
Genel İşletme Doktora Programı**

Günay KILIÇ

Danışman: Prof. Dr. Arzu ORGAN

**Ocak 2023
DENİZLİ**

Bu tezin tasarımı, hazırlanması, yürütülmesi, arařtırmalarının yapılması ve bulgularının analizlerinde bilimsel etięe ve akademik kurallara özenle riayet edildiđini; bu arařtırmanın doğrudan birincil ürünü olmayan bulguların, verilerin ve materyallerin bilimsel etięe uygun olarak kaynak gösterildiđini ve alıntı yapılan alıřmalara atıfta bulunulduđunu beyan ederim.

Günay KILIÇ

ÖN SÖZ

Bu tezin hazırlanma sürecinde kıymetli vakitlerini ayırarak beni yönlendiren danışman hocam Prof. Dr. Arzu ORGAN başta olmak üzere tez savunma jüri üyeleri Prof. Dr. Sezai TOKAT, Doç. Dr. Nilfen KUNDAKCI, Doç. Dr. Hüseyin ŞENKAYAS ve Dr. Öğr. Üyesi Gürhan GÜNDÜZ hocalarıma en içten teşekkürlerimi sunarım. Tez yazım sürecinde bana fikirleri ile destek olan Dr. Öğr. Üyesi Mehmet Ulaş KOYNCUOĞLU'na ve büyük dikkat ile tezimi okuyan Doç. Dr. Ayşegül TUŞ ve Dr. Ayşe GENÇ hocalarıma da destekleri sebebiyle teşekkür ederim.

Sadece doktora sürecinde değil, hayatımın her anında bana destek olan sevgili eşim Bedia Sündüz KILIÇ'a bana gösterdiği anlayış ve sabır için ayrıca teşekkür ederim. Bugünlere gelmemde emekleri olan bana her türlü maddi manevi desteklerini sunan, her zaman yanımda olan aile büyüklerime şükranlarımı sunarım. Bir teşekkür de kıymetli vakitlerinden çaldığım kızlarım Zehra, Zeynep ve Elif'e. Güzel kızlarım, umarım sizler ile daha fazla vaktimiz, daha güzel günlerimiz olacak.

ÖZET

SIRA BAĞIMLI İLİŞKİSİZ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ İÇİN YENİ BİR SEZGİSEL ALGORİTMA ÖNERİSİ

KILIÇ, Günay

Doktora Tezi

İşletme ABD

Genel İşletme Doktora Programı

Tez yöneticisi: Prof. Dr. Arzu ORGAN

Ocak 2023, VIII +94 Sayfa

Üretim aşamasında makinelerin işleyişinin belirlendiği çizelgeleme, işletmelerde etkin bir üretim için son derece önemlidir. İşletmeler büyüdükçe planlama ve üretim aşamasında işlerin hangi çizelgede yapılması gerektiği daha karmaşık hal almaktadır. Küçük makine-iş konfigürasyonunda en uygun çizelgeleme, uzman görüşü veya tam sayılı programlama teknikleri ile elde edilebilir. İşlerin ve makinelerin sayısı arttıkça kesin çözümün bulunması imkânsız hale gelmektedir. Kesin çözümün bulunamadığı durumlarda yaklaşık çözümler metasezgisel algoritmalar ile elde edilmektedir. Çalışma kapsamında ele alınan problem, kesin çözümü bulunamayan ve metasezgisel algoritmalar ile çözülebilen sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemidir. Bu çalışma, üretim süresini minimize etmeyi amaçlamaktadır. Bu çalışmada, birçok araştırmacı tarafından daha önce farklı metasezgisel algoritmalar ile yaklaşık en iyi çözümü bulunmaya çalışılmış bir kıyas veri seti üzerine çalışılmıştır. Bu tez kapsamında kıyas veri setini çözmek için uyarlanmış yeni bir değişken komşuluk arama algoritması önerilmektedir. Yeni önerilen algoritma, komşuluk yapılarına 2 farklı parametre seçimi eklemektedir. Bu eklenen yeni parametrelerin sonuç üzerine etkisi tartışılmış ve bulgular bu yeni parametreleri kullanmanın faydalı olduğunu göstermektedir. Değişken komşuluk arama algoritmasındaki yerel arama modülü, birden fazla yerel arama ile çeşitlendirilmiş ve komşuluk değerlerine göre yerel arama seçim fazı eklenmiştir. Eklenen seçim fazı, algoritmayı gereksiz yerel aramalardan kurtarmaktadır. Algoritma kıyas veri setinde 50'den fazla test edilmiş ve en iyi sonuçlar raporlanmıştır. Denenen 540 adet veri setinin 141 tanesinde bilinen en iyi sonuçtan daha iyi değerler elde edilmiştir. 140 örnekte ise bilinen en iyi çözümlere ulaşılmıştır. Önerilen algoritma, tüm örneklerde ortalama %0,095 hata ile çalışmaktadır.

Anahtar Kelimeler: Değişken Komşuluk Arama, Metasezgisel, Sıra Bağımlı İlişkisiz Hazırlık Süreli Paralel Makine Çizelgeleme, Yerel Arama

ABSTRACT

A NEW METAHEURISTIC PROPOSAL FOR UNRELATED PARALLEL MACHINE SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT SETUP TIMES

KILIÇ, Günay

Doctoral Thesis

Business Administration Department

PhD. In Business Administration

Adviser of Thesis: Prof. Dr. Arzu ORGAN

January 2023, VIII +94 Pages

Scheduling, determines which job will be performed by which machine at the production stage, is extremely important for an effective production in businesses. As a business grows, scheduling will be more complex in the planning and production stages. In a business, optimal scheduling in a small machine job environment can be done by an expert opinion or integer programming techniques. As the number of jobs and machines increases, it becomes impossible to find an exact solution. When the exact solution cannot be found, approximate solutions are obtained with metaheuristic algorithms. The problem studied in this study is the one that can be solved by metaheuristic algorithms that cannot find exact solutions. This is an unrelated parallel machine scheduling problem with setup times. The study aims to minimize makespan. In this problem, machines are unrelated. Each job has different process time on different machines and different setup times on different machines. In this study, a benchmark dataset has been studied. Many researchers have tried to find the approximate best solution with different metaheuristic algorithms. In this thesis, a new Adapted Variable Neighborhood Search Algorithm is proposed to solve the benchmark dataset. The new proposed algorithm adds 2 different parameters to the neighborhood structures. The effect of added parameters on the result was discussed and it was seen that the findings were beneficial in using these new parameters. The local search module in the proposed variable neighborhood search algorithm has 4 different local search algorithms. A new local search selection phase was added according to the neighborhood makespan values. The local search selection phase saves the algorithm from unnecessary local searches. The benchmark dataset was tested more than 50 times and best results were stored. When the results obtained with the proposed algorithm were compared with the best known results, it was seen that the proposed algorithm is successful in some machine job configurations. Better results than the best known results were obtained in 141 of 540 samples in the data set. In 140 samples, the best known solutions were reached. The proposed algorithm works with an average error of 0.095% in all samples.

Keywords: Variable Neighborhood Search, Metaheuristic, Unrelated Parallel Machine Scheduling, Local Search

İÇİNDEKİLER

ÖN SÖZ	i
ÖZET.....	ii
ABSTRACT	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ.....	vi
TABLolar DİZİNİ	vii
SİMGE VE KISALTMALAR DİZİNİ	viii
GİRİŞ	1

BİRİNCİ BÖLÜM ÇİZELGELEME PROBLEMİ

1.1. Sıralama Öncelik Kuralları	5
1.2. İş Çizelgeleme Notasyonları	7
1.3. Çizelgeleme Probleminin Sınıflandırılması	11
1.3.1. İşlerin Geliş Şekline Göre Sınıflandırılması	11
1.3.1.1. Deterministik Problemler	12
1.3.1.2. Dinamik Problemler	13
1.3.2. İşlerin Görüleceği Makine Sayısına Göre Sınıflandırma	14
1.3.2.1. Tek Makineli Model.....	14
1.3.2.2. Birden Çok Makineli Model	14
1.4. Çizelgeleme Problemi için Önerilen Yöntemler	17
1.4.1. Kesin Çözüm Yöntemleri.....	17
1.4.2. Sezgisel Çözüm Yöntemleri.....	18
1.4.3. Metasezgisel Çözüm Yöntemleri	19
1.4.4. Hibrit Algoritmalar	20
1.5. Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi	21
1.5.1. Matematiksel Model	21
1.6. Çizelgeleme ile İlgili Çalışmalar	23

İKİNCİ BÖLÜM OPTİMİZASYON VE METASEZGİSELLER

2.1. Kombinatoriyal Optimizasyon	42
2.2. Metasezgisel Algoritmalar	45
2.2.1. Tek Çözüm Tabanlı Metasezgiseller.....	46
2.2.1.1. Yerel Arama	47
2.2.1.2. Benzetimli Tavlama	48
2.2.1.3. Tabu Arama.....	49
2.2.1.4. Değişken Komşuluk Arama	50

2.2.2. Popülasyon Tabanlı Metasezgiseller.....	51
2.2.2.1. Genetik Algoritmalar	52
2.2.2.2. Karınca Kolonisi Optimizasyonu.....	53
2.2.2.3. Parçacık Sürü Optimizasyonu	53
2.2.3. Hibrit Metasezgiseller	54

ÜÇÜNCÜ BÖLÜM

SIRA BAĞIMLI HAZIRLIK SÜRELİ İLİŞKİSİZ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ İÇİN ÖNERİLEN DEĞİŞKEN KOMŞULUK ARAMA ALGORİTMASI

3.1. Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi Örnek Veri Seti	56
3.2. Başlangıç Çözümü Oluşturma.....	57
3.2.1. Rasgele Başlangıç Çözümü Oluşturma.....	57
3.2.2. Açgözlü Başlangıç Çözümü Oluşturma.....	58
3.3. Önerilen Değişken Komşuluk Arama Algoritması.....	60
3.3.1. Çalışmada Kullanılan Komşuluk Yapıları	61
3.3.1.1. Literatürde Bulunan Komşuluk Yapıları.....	61
3.3.1.2. Önerilen DKİ Makine Seçim Parametresi	64
3.3.1.3. Önerilen DKİ Serbestlik Parametresi.....	64
3.3.1.4. Önerilen Değişken Komşuluk İniş Test Algoritması.....	65
3.3.2. Yerel Aramalar.....	69
3.3.3. Önerilen Yerel Arama Seçim Stratejisi.....	73
3.3.4. Önerilen Yeni Değişken Komşuluk Arama Algoritmasının Parametreleri ...	73
3.4. Deneysel Sonuçlar.....	78
SONUÇ	83
KAYNAKLAR	86
ÖZ GEÇMİŞ	944

ŞEKİLLER DİZİNİ

Şekil 1. 6 iş 2 Makine Rasgele Başlangıç Gantt Şeması.....	58
Şekil 2. 6 iş 2 Makine Açgözlü Başlangıç Gantt Şeması.....	60
Şekil 3. K1 Komşuluk Yapısının Uygulanması	62
Şekil 4. K2 Komşuluk Yapısının Uygulanması	62
Şekil 5. K3 Komşuluk Yapısının Uygulanması	63
Şekil 6. K4 Komşuluk Yapısının Uygulanması	63
Şekil 7. K5 Komşuluk Yapısının Uygulanması	64
Şekil 8. $sp=2$ Olduğu Durumda K3 Komşuluk Yapısının Uygulanması	65
Şekil 9. Makine Sayısına Göre Çalışma Süreleri	76
Şekil 10. İş Sayısına Göre Algoritma Süreleri	76
Şekil 11. Makine Sayısına Göre Algoritmaların Ortalama Yüzdesel Hataları	81
Şekil 12. İş Sayısına Göre Algoritmaların Ortalama Yüzdesel Hataları.....	82

TABLOLAR DİZİNİ

Tablo 1. SBHSİPMÇP'ne Ait Gösterimler	22
Tablo 2. SBHSİPMÇP'nin Matematiksel Modeli.....	22
Tablo 3. 6 İş ve 2 Makine İçin Süreler	56
Tablo 4. 6 İş ve 2 Makine Çizelgelemede 1. Makine için Hazırlık Süreleri.....	56
Tablo 5. 6 İş ve 2 Makine Çizelgelemede 2. Makine için Hazırlık Süreleri.....	56
Tablo 6. Rasgele Başlangıç Algoritması C_{max} Değerleri	58
Tablo 7. 6 İş 2 Makine İşlerin Makinelere Göre Ortalama Hazırlık Süreleri	59
Tablo 8. Önerilen ve Temel DKA algoritması kıyaslaması	61
Tablo 9. Önerilen Değişken Komşuluk İniş Test Algoritması Senaryoları	66
Tablo 10. Rasgele Başlangıç Çözümü ile DKİ_KM_SP Makine Sayısına Göre İlk Çözüm Gelişim Yüzdeleri.....	67
Tablo 11. Rasgele Başlangıç Çözümü ile DKİ_KM_SP İş Sayısına Göre İlk Çözüm Gelişim Yüzdeleri	67
Tablo 12. Açgözlü Başlangıç Çözümü ile DKİ_KM_SP Makine Sayısına Göre İlk Çözüm Gelişim Yüzdeleri.....	68
Tablo 13. Açgözlü Başlangıç Çözümü ile DKİ_KM_SP İş Sayısına Göre İlk Çözüm Gelişim Yüzdeleri	68
Tablo 14. Rasgele Başlangıç Çözümü ile DKİ_KM_SP En iyi Çözüme Hata	68
Tablo 15. Açgözlü Başlangıç Çözümü ile DKİ_KM_SP En iyi Çözüme Hata.....	69
Tablo 16. Yerel Arama İşlem Karmaşıklıkları.....	73
Tablo 17. Bir İterasyon ile Algoritma Ortalama Süreleri	75
Tablo 18. İterasyon Sayılarına Göre Süre ve Hatalar	77
Tablo 19. Revize İterasyon Sayısına Göre Süre ve Hata	78
Tablo 20. Veri Seti ile İlgili Çalışmalar	79
Tablo 21. Küçük Veri Setinde Deneysel Sonuçlar.....	79
Tablo 22. Büyük Veri Setinde Elde Edilen C_{max} Değerleri.....	80

SİMGE VE KISALTMALAR DİZİNİ

AKİSÖ	Ağırlıklı Kısa İşlem Süresi Önce
ARUA	Açgözlü Rasgele Uyarlanabilir Arama Algoritması
BT	Benzetimli Tavlama
C_i	i Makinesine Atanan Bütün İşlerin Toplam Tamamlanma Süresi
C_{max}	Çizelgeleme Sonucu Son İşin Sistemden Ayrılma Süresi
$C_{maxEniyi}$	Bulunan En İyi C_{max} Değeri
DBT	Değiştirilen Bitiş Tarihi
DKA	Değişken Komşuluk Arama
DKİ	Değişken Komşuluk İniş
EBİÖ	En Erken Bitecek İlk Önce
EKGS	En Kısa Gevşeklik Süresi İlk
ETZ	Erken Teslim Zamanı Önce
EYTÖ	Erken Yayın Tarihi Önce
GA	Genetik Algoritma
GMH	Göreceli Maliyetler Hesabı
GSP	Gezgin Satıcı Problemi
İBG	İşlem Başına Gevşeme
İBTE	İşlem Bitiş Tarihine Göre En Erken
İGİG	İlk Gelen İlk İşlem Görür
SBHSİPMÇP	Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi
KİSE	Kalan İşlem Sayısı En Az Olan İlk Yapılır
KİSÖ	Kısa İşlem Süresi Önce
KO	Kombinatoriyal Optimizasyon
KKO	Karınca Kolonisi Optimizasyonu
kmp	Komşuluk Seçiminde Kritik Makine Seçim Parametresi
M	Çizelgeleme Problemi İçin Toplam Makine Sayısı
N	Çizelgeleme Problemi İçin Toplam İş Sayısı
p_{ij}	i Makinesinde j İşinin Tamamlanma Süresi
PMÇP	Paralel Makine Çizelgeleme Problemi
PSO	Parçacık Sürü Optimizasyonu
$S_{i,j,k}$	j İşinin k Makinesinde i İşinden Sonra İşlenmesi İçin Gereken Hazırlık Süresi
SKBÇ	Sonraki Kuyrukta Beklenen Çalışma
SO	Sürü Optimizasyonu
sp	Komşuluk Değişiminde Serbestlik Parametresi
TA	Tabu Arama
TZÜ	Tam Zamanında Üretim
UİSÖ	Uzun İşlem Süresi Önce
YA	Yerel Arama
YAK	Yapay Arı Kolonisi Algoritması

GİRİŞ

İşletmelerin amacı hizmet veya mal üreterek insanların ihtiyaçlarını karşılamak, kâr elde etmek ve bu sayede ekonomik sürdürülebilirliği sağlamaktır. Sürdürülebilirlik, rakip işletmeler ile rekabet edecek etkin kaynak yönetimi ile mümkündür. Bu amaç doğrultusunda işletmeler, diğer işletmeler ile rekabet edebilmek için daha kısa zamanda rakiplerinden daha fazla üretim yaparak daha fazla kar elde etmek isterler. Üretim miktarı, kısıtlı kaynak altında mevcut bulunan kaynakların en etkin bir şekilde planlanması sayesinde arttırılabilir. Üretimin en önemli aşamalarından biri de planlamadır. Etkili bir planlama ile daha fazla ve etkili bir üretim yapabilmek için işlerin etkin bir şekilde planlanması ve işlerin mümkün olduğunca erken tamamlanması gerekmektedir. Etkin planlama sürecinin en önemli kısmı ise kaynakların etkin bir şekilde kullanılmasıdır. Makineler ile üretim yapılan işletmelerde kaynaklar işlere tahsis edilirken, en uygun çizelgelemenin yapılması ve sıradaki işin en uygun makinede işlenmesi gerekmektedir.

Çizelgeleme, belirli amaçlarla görevlere sınırlı kaynakların tahsis edilmesi ile ilgilidir. Çizelgeleme problemleri genellikle karmaşık, büyük ölçekli, zorlayıcı konulardır ve çeşitli kısıtlamalar içerir. Gerçek bir çözüm bulmak için, gerçek dünyadaki sorunların çoğu optimizasyon problemleri olarak formüle edilmelidir. Ayrıca, sistemin sık kullanımı için verimli ve düşük maliyetli prosedürler bulmak da çok önemlidir. Sorunları çözmek için çeşitli çözümler önerilmesine rağmen, yine de daha uygun sonuçlar elde eden yöntemlere ihtiyaç vardır (Gandomi vd, 2020: 1-2).

Çizelgeleme problemlerinde, etkili çözümler geliştirmek adına literatürde birçok yöntem önerilmiştir. Küçük boyutlu problemlerin çözümünde kesin algoritmaları kapsayan yöntemler önerilirken, büyük boyutlu problemlerde ise metasezgisel yöntemler önerilmektedir. Metasezgisel yöntemler, tek çözümlü ve popülasyon tabanlı algoritmalar olmak üzere iki başlık halinde toplanabilir. Son yıllarda metasezgisel ve kesin çözüm yöntemlerinin birlikte kullanıldığı hibrit uygulamalar çokça kullanılmaktadır.

Çizelgeleme probleminin çözümünde hangi yöntemin seçileceğine karar verilirken, iş yerinin üretim sisteminin yapısına bakmak gerekir. Bir üretim sistemi, işlerin işlendiği tek bir makineden veya bir işi aynı veya farklı sürelerde yapabilen, birbirine paralel veya seri bağlanmış birden çok makineden oluşabilir. Birden çok makinenin bulunduğu üretim sistemlerinde sıradaki işi işlemek için, işin ve makinenin özelliğine göre, farklı bir hazırlık süresi ve işleme süresine ihtiyaç duyulabilir. Her işin her bir makinede farklı hazırlık süresi ve işleme süresine sahip olduğu ve makinelerin paralel biçimde sıralandığı ilişkisiz paralel makineli üretim sistemleri birçok imalat endüstrisinde karşımıza çıkmaktadır. Bu imalat endüstrilerinin başlıca sorunlarından biri makinelerin işleri etkin şekilde işleyeceği çizelgelerin oluşturulmasıdır.

Paralel makine çizelgeleme problemi (PMÇP) birçok üretim sisteminde karşılaşılan tipik bir kombinatoriyal optimizasyon problemidir. PMÇP bütün makinelerin özdeş, makinelerin farklı hızda olduğu tekdüze ve farklı makinelerde işlerin farklı tamamlanma sürelerinin olduğu ilişkisiz makinelerde olmak üzere üç farklı şekilde sınıflandırılabilir. Özdeş ve tekdüze makinelerde çizelgeleme problemi, ilişkisiz makine çizelgeleme probleminin özel bir halidir. Karmaşık bir çizelgeleme problemi olarak PMÇP 2 makineli durumda bile NP-Zordur. Bu nedenle, PMÇP’yi incelemek ve özellikle en genel olan ilişkisiz PMÇP için etkili çözüm algoritmaları geliştirmek önemlidir (Wang vd. 2016:235). Özellikle sıra bağımlı hazırlık süreli ilişkisiz paralel makinelere sahip firmalarda çizelgeleme problemlerini çözmek için geliştirilen algoritmalara ihtiyaç vardır. Bu çalışmada, “Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi” nin çözümü için algoritma geliştirmek amaçlanmıştır.

Çalışmada ele alınan “Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi” (SBHSİPMÇP) akademik yazında ve farklı imalat endüstrilerinde karşımıza çıkmaktadır. Bazı akademik çalışmalar, fabrikalardaki gerçek verileri kullanırken bazı çalışmalarda ise kıyas veri setinden faydalanılmıştır. Bu çalışmada kullanılan kıyas veri seti 2004 yılında oluşturulmuş birçok çalışmada kullanılmıştır.

Tezde çalışan problemin çözümü için literatürde yeni bir uyarlanmış metasezgisel algoritma önerilmektedir. Önerilen algoritma tek çözümlü metasezgisel algoritmalarından değişken komşuluk arama algoritmasıdır. Çalışma kapsamında değişken komşuluk arama metasezgiseline üç farklı yenilik katılmaktadır. Katılan ilk yenilik ile başlangıç çözümü oluşturmak için yeni bir açgözlü başlangıç algoritması önerilmektedir. Bu sayede algoritma daha kaliteli çözümler ile başlatılıp nihai çözüme daha hızlı yaklaşılabilecektir. İkinci yenilik

ile komşuluk değişimine yeni iki parametre önerilmektedir. Yeni komşuluk parametreleri test edilmiş ve etkinliği ortaya konulmuştur. Üçüncü yenilik ise keşif yengeçlerinin kabuk değişiminden esinlenen yeni bir yerel arama tekniğidir. Uyarlanmış algoritma çalışılan problemde test edilmiş ve umut verici sonuçlar ortaya çıkmıştır. SBHSİPMÇP çözümü için geliştirilen algoritma benzer probleme sahip işletmeler tarafından kullanılıp işletmelere fayda sağlayabilir. Değişken komşuluk arama algoritması için önerilen yenilikler aynı yapıyı kullanan farklı problemlerde de uyarlanıp kullanılabilir.

Tez üç bölümden oluşmaktadır. Birinci bölümde çizelgeleme problemlerinden bahsedilmiştir. Bu bölümde çizelgelerin sınıflandırılması, çizelgeleme problemlerinin çözüm yöntemleri anlatılmıştır. Literatürde çizelgeleme problemi ile alaklı çalışmalara yer verilmiştir. Kıyas veri setini kullanan çalışmalar irdelenmiş çözüm yöntemleri ve sonuçlarına yer verilmiştir. İkinci bölümde optimizasyon ve optimizasyon yöntemlerinden bahsedilmiştir. Kombinatoriyel optimizasyon kavramı açıklanmıştır. Kıyas veri setinin çözümünde kullanılan metasezgisel yöntemler açıklanmıştır. İkinci bölümde ayrıca bu çalışma kapsamında kullanılan yöntem olan değişken komşuluk arama algoritmasından bahsedilmiştir. Üçüncü bölüm ise uygulama bölümüdür. Bu bölümde çözümde önerilen yeni algoritmaya eklenen komşuluk seçim parametreleri açıklanmıştır. Yeni komşuluk parametreleri için önerilen değişken komşuluk iniş algoritması dört farklı senaryo ve iki farklı başlangıç algoritması ile test edilmiştir. Ayrıca üçüncü bölümde yeni yerel arama yönteminin de dâhil olduğu yerel arama algortimaları açıklanmıştır. Önerilen uyarlanmış değişken komşuluk arama algoritması ile SBHSİPMÇP çözülmüştür. Önerilen algoritma, çalışmada ele alınan kıyas veri setini kullanan diğer algoritmalar ile karşılaştırılmıştır. Karşılaştırma sonuçları, makine ve iş sayısına göre tablo halinde ve şekilsel grafikler ile ortaya konulmuştur.

BİRİNCİ BÖLÜM

ÇİZELGELEME PROBLEMİ

Günlük hayatımızda yaygın bir şekilde hastane, banka veya alışveriş merkezi gibi yerlerde insanlar sıraya girmektedir. Bu sıralar literatürde kuyruk olarak isimlendirilmektedir. Bu kuyruklarda genelde ilk gelen ilk işlem görür kuralı uygulanmaktadır. Bu kural işletmelerde her zaman geçerli değildir. İşletmelerde işleri belirli bir etkinlik ölçüsüne göre sıralamak kaçınılmazdır. Ancak bu şekilde işlerde verimlilik ve rasyonellik sağlanabilir. Sıralama kavramı ile çizelgeleme kavramı da ortaya çıkar. Çizelgeleme hangi işin hangi makinede işlenmesinin planlanmasıdır (Tanyaş ve Baskak, 2003: 405-406).

Çizelgeleme, birçok imalat ve hizmet endüstrisinde düzenli olarak kullanılan bir karar verme sürecidir. Verilen sürelerde kaynakların görevlere tahsisi ile ilgilidir. Çizelgelemenin amacı bir kaynağı bir göreve atarken bir veya daha fazla amacı optimize etmektir. Kaynaklar bir atölyedeki makineler ise, görevler bir üretim sürecindeki işler olarak söylenebilir. Her görevin belirli bir öncelik düzeyi, mümkün olan en erken başlangıç zamanı ve bitiş tarihi olabilir. Amaçlar ayrıca birçok farklı şekilde olabilir. Bir amaç, son görevin tamamlanma süresinin en aza indirilmesi, diğeri ise ilgili bitiş tarihlerinden sonra tamamlanan görev sayısının en aza indirilmesi olabilir. Bu yüzden bir karar verme süreci olarak çizelgeleme, çoğu üretim sisteminde önemli bir rol oynamaktadır (Pinedo, 2018: 1).

Çizelgeleme alanındaki gelişmelerin çoğu, imalatta ortaya çıkan sorunlardan kaynaklanmıştır. Günümüzde çizelgeleme çoğu alanda önemli yer edinse de kaynaklara genellikle makine/iş istasyonu/sunucu, görevlere ise iş/parça denir. Bazen işler, operasyon adı verilen birkaç temel görevden oluşabilir (Baker ve Trietsch, 2019: 4). Başka bir deyişle çizelgeleme, her iş için bir veya daha fazla makinenin, bir veya daha fazla zaman aralığının tahsis edilmesidir. Çizelgelemede amaç, belirli kısıtlamaları karşılayan bir çizelge bulmaktır (Brucker, 2007: 1).

Çizelgeleme, belirli amaçlarla görevlere sınırlı kaynakların tahsis edilmesi ile ilgilidir. Toplam Kalite Yönetimi popülerliği nedeniyle, zamanında iş teslimi müşteri memnuniyeti için önemli faktörlerden biri haline gelmiştir. Çizelgeleme, bu hedefe ulaşmada önemli bir rol oynar (Lee, 1997: 1).

Çizelgeleme problemleri genellikle karmaşık, büyük ölçekli, zorlayıcı konulardır ve çeşitli kısıtlamalar içerir. Gerçek bir çözüm bulmak için, gerçek dünyadaki sorunların çoğu optimizasyon problemleri olarak formüle edilmelidir. Ayrıca, sistemin sık kullanımı için verimli ve düşük maliyetli prosedürler bulmak da çok önemlidir. Bu sorunlara yönelik çeşitli çözümler önerilmesine rağmen, yine de daha uygun sonuçlar elde eden yöntemlere ihtiyaç vardır (Gandomi vd., 2020: 1-2).

Günümüzde hem hizmet hem de üretim alanındaki çizelgeleme problemlerinde önemli bir konu da kurulum süresi ve maliyetleridir. Kurulum süresi ve maliyetlerini işlem süre ve maliyetlerinden ayrı düşünmemek işlemlerin aynı anda yapılmasını sağlar ve işlem sürelerini kısaltır. Bu sayede özellikle modern üretim sistemlerinden tam zamanında üretim (TZÜ), rekabete dayalı üretim gibi sistemlerde fayda sağlanır. Bu faydalara daha az masraf, artan üretim hızı, artan çıktı, daha kısa teslim süreleri, daha hızlı geçişler, artan rekabet gücü, artan karlılık ve memnuniyet, yalın üretimi mümkün kılma, daha akıcı akışlar ve daha az stok örnek olarak verilebilir. Bu sayede müşteri memnuniyeti sağlanmış olur (Allahverdi ve Soroush, 2008: 979).

Çizelgeler oluşturulurken ve makinelere iş atanırken, işler farklı sıralama kuralları ile atanmaktadır. Problemin çeşidine göre farklı sıralama öncelik kuralları bulunmaktadır. Bu kurallardan aşağıda bahsedilmiştir.

1.1. Sıralama Öncelik Kuralları

Dağıtım veya öncelik kuralları, kolay uygulama ve hesaplama gücünde düşük gereksinimler nedeniyle çizelgeleme problemleri için en yaygın sezgisel yöntemlerdir. Her ne kadar bazı durumlarda çok iyi performans gösterse de tüm çizelgeleme problemlerine uygulanabilecek ve tatmin edici bir performans sergileyen hiçbir kural yoktur. Daha da kötüsü, belirli bir örnek için bir gönderme kuralının performansını önceden tahmin etmenin bir yolu yoktur (Xhafa ve Abraham, 2008: 12). İşyerlerinde veya iş araçlarında siparişlerin yerine getirilme sırasının sezgisel planlaması için öncelik kuralları kullanılır. Öncelik kuralları, bir işyerinde kuyruğa alınan siparişlerin hangi sırayla yerine getirileceğini belirler (Zupan vd., 2016: 31). Özellikle dinamik atölye çizelgelemesinde, düzenli ve düzenli olmayan performans ölçümlerinin çoğunu en aza indiren tek bir dağıtım kuralı bulunamamıştır (Dominic 2004: 71). İş öncelik sıralama kurallarından bazıları şunlardır (Pinedo, 2018: 373-378; Baker ve Trietsch, 2019: 430-432):

İlk Gelen İlk İşlem Görür (İGİG) (*First in First Out - FIFO*): İşler arasında ilk gelenin ilk işlendiği durumdur.

Kısa İşlem Süresi Önce (KİSÖ) (*Shortest Process Time First - SPT*): Bir işletmede bekleyen işler arasında en kısa süreye sahip işin ilk yapılması durumudur. Tek makine çizelgeleme ve paralel özdeş makinelerin kısıtlarının olmadığı durumda bu kural ile toplam iş süresi minimize edilebilir (Pinedo, 2018: 130).

Ağırlıklı Kısa İşlem Süresi Önce (AKİSÖ) (*Weighted Shortest Process Time First - WSPT*): Bir işletmede bekleyen işler arasında ağırlığı ve en kısa süreye sahip işin ilk yapılması durumudur.

Uzun İşlem Süresi Önce (UİSÖ) (*Longest Process Time First - LPT*): Bir işletmede işlem süresi en uzun olan işin önce yapılması durumudur.

Erken Teslim Zamanı Önce (ETZ) (*Early Due Date - EDD*): İşler arasında teslim tarihi en yakın olan önce yapılır. Tek makineli çizelgelemede veya tek iş ailesine sahip paralel makine çizelgelemede bu kural ile toplam gecikme minimize edilebilir. (Baker ve Trietsch, 2019: 374).

Kalan İşlem Sayısı En Az Olan İlk Yapılır (KİSE) (*Least Work Remaining - LWKR*): Kendinden sonraki işlem sayısı en az olan önce yapılır.

Kalan İşlem Sayısı En Fazla Olan İlk Yapılır (KİSF) (*More Operation Remaining - MOR*): Kendinden sonraki işlem sayısı en fazla olan önce yapılır.

Sonraki Kuyrukta Beklenen Çalışma (SKBÇ) (*Anticipated Work in Next Queue - AWINQ*): Kendinden sonra en az iş olması beklenen iş önce yapılır.

En Erken Bitecek İlk Önce (EBİÖ) (*First of First On - FOFO*): En erken bitirilecek iş seçilir en erken bitirilecek iş kuyrukta yok ise makine çalışmaz.

Erken Yayın Tarihi Önce (EYTÖ) (*Early Release Date - ERD*): Yayınlanma tarihi önce olan iş ilk yapılır.

En Kısa Gevşeklik Süresi İlk (EKGS) (*Minimum Slack Time- MST*): Bitiş ve bekleme süresi arasında en az süre olan işin önce yapılır. Gevşeklik süresi, bitiş tarihi ile işin mümkün olan en erken bitiş zamanı arasındaki farka eşittir.

İşlem Bitiş Tarihine Göre En Erken (İBTE) (*Operation Due Date - ODD*): En erken işlem bitiş tarihine sahip işin seçilmesidir. Bir iş bitiş tarihi, iş bitiş tarihi ile

yayınlanma tarihi arasındaki aralığı, işlemler kadar alt aralığa bölerek belirlenir. Her alt aralığın sonu, ilgili işlem için bir bitiş tarihini temsil eder.

İşlem Başına Gevşeme (İBG) (*Slack / Operation - S/OPN*): İşle ilişkili, gevşeme süresinin (bitmesi gereken –bitiş zamanı) geri kalan işlemlere minimum oranı olan işlemin seçilmesidir.

Kesilmiş Kısa İşlem Süresi Önce (KKİSÖ) (*Truncated SPT- TSPT*): Öncelikle işler belirli bir süreden fazla bekleyen işler KİSÖ kuralına göre işlenir.

Değiştirilen Bitiş Tarihi (DBT): (*Modified Due Date - MDD*): En erken değiştirilen bitiş tarihine sahip işle ilişkili işlerin seçilmesidir. Değiştirilen bitiş tarihi, hangisi daha büyükse, orijinal bitiş tarihi veya mümkün olan en erken bitiş saatidir.

İşlerin sıralama kuralları iki farklı şekilde sınıflandırılabilir. Birincisi kuralların yerel ve bütünsel kurallar olarak sınıflandırılmasıdır. Yerel kurallar sadece kuyrukta bekleyen işlerin özelliklerine göre karar verirken, bütünsel kurallar makine dışındaki bilgileri de kullanır. Yerel kurallara KİSÖ, KİSE örnek olarak verilebilir. Bütünsel kurallara ise SKBÇ ve EBİÖ örnek olarak verilebilir. Sezgisel olarak bütünsel kuralların daha etkili olması beklenirken bu kuralların yerel kurallardan iyi olduğuna dair bir kanıt yoktur. Bütünsel kurallar için gerekli bilgi tabanı çok büyüktür ve birçok simülasyon çalışmalarında yerel kurallar kullanılmıştır. İkinci sınıflandırma ise kuralın zamanla değişip değişmediğine göre belirlenir. Eğer kural zamanla değişmez ise statik zamanla, değişir ise dinamik kural olur. KİSÖ, KİSE statik kurallara EKGS, İBTE, İBG, KKİSO, DBT ise dinamik kurallara örnek olarak verilebilir (Baker ve Trietsch, 2019: 431-432).

1.2. İş Çizelgeleme Notasyonları

Graham, R vd. (1979) yılında teorik çizelgeleme problemleri için uygun bir notasyon oluşturmuş ve birçok çalışmada bu notasyon kullanılmıştır. Aşağıdaki çizelgeleme notasyonları aynı şablonu kullanan Pinedo (2018:13-19)'nun çalışmasından alınmıştır.

Çizelgeleme problemlerinde iş ve makine sayısının sınırlı olduğu öngörülür. İş sayısı N , makine sayısı M ile gösterilir. Genellikle, alt simge j bir işi belirtirken, alt simge i bir makineyi ifade eder. (i, j) ikilisi i . makinede j işinin işlenmesi adımı veya çalışmasını ifade eder. j işi ile ilgili tanımlamalar şu şekildedir:

İşlem Süresi (p_{ij}): i işinin j makinesinde tamamlanma süresidir.

Yayın Zamanı (r_j): j işinin başlayacağı zamanı gösterir.

Bitiş Zamanı (d_j): j işinin tamamlanma tarihini gösterir. Bir işin bitiş tarihinden sonra tamamlanmasına izin verilir, ancak daha sonra bir ceza verilir. Eğer bitiş zamanında tamamlanması istenirse ortalama bitiş zamanı belirtilmelidir.

Ağırlık (w_j): Bir işin sistemdeki diğer işlere göre önem derecesini gösterir.

Bir çizelgeleme problemi $\alpha | \beta | \gamma$ üçlüsü ile tanımlanmaktadır. α alanı makine ortamını tanımlar ve yalnızca bir giriş içerir. β alanı, işleme karakteristiklerini ve kısıtlamalarını içerir. Bu alan ya hiç girdinin olmadığı, tek bir girdi veya birden fazla girdi içerebilir. γ alanı en aza indirilecek amaç fonksiyonunu tanımlar ve genellikle tek bir giriş içerir.

Makine ortamı (α) olası girişler şu şekildedir:

Tek Makineli Model (1): Tek bir makine durumu, tüm olası makine ortamlarının en basitidir ve diğer daha karmaşık makine ortamlarının özel bir durumudur.

Paralel Özdeş Makineler (Pm): M adet aynı işi yapan paralel makineden oluşan ortamdır. j işi tek bir işlem gerektirir ve M makinelerinden herhangi birinde veya belirli bir alt kümeye ait herhangi bir makinede işlenebilir. Eğer j işi özel bir M_j alt makine grubunda işlenemez ise bu bir kısıttır ve M_j alt makine grubu β alanına kısıt olarak yazılmalıdır.

Farklı hızlarda paralel makineler (Qm): M adet farklı hızda paralel makinenin olduğu durumdur. Makine hızları v_i ile ifade edilir ve bir j işinin i makinesinde yapılma süresi $p_{ij}=p_j/v_i$ olarak ifade edilir.

Paralelde Özdeş Olmayan Makineler (Rm): M adet ilişkisiz makinenin bulunduğu durumdur. i makinesi j işini v_{ij} sürede yapar. Bütün süreler aynı olduğu durumda Pm durumu ile aynıdır.

Seri Makineler (Fm): M adet seri makinenin bulunduğu durumdur. Her iş m adet makinede aynı sırada işlenmelidir. Bir makineden öbür makineye geçince İĞİG kuralı işlenir. Eğer yer değiştirme durumu olacaksa bu $prmu$ girişi ile kısıt β alanına eklenmelidir.

Esnek Akış Atölyesi (FFc): Esnek akış atölyesi, akış atölyesinin ve paralel makine ortamlarının genelleştirilmesidir. Serideki M makineleri yerine, seri c adet aşama ve her

aşamada paralel olarak bir dizi özdeş makine vardır. Her iş sırası ile tüm seri aşamaları geçer, bu aşamalar içinde j işi yalnızca bir makine tarafından yapılır.

Atölye (Jm): M makineli bir atölyede, her işin önceden belirlenmiş kendine ait bir rotası vardır. Kısıtsız durumda her bir işin her makineyi en fazla bir kez ziyaret ettiği atölye durumudur. Bir işin her makineyi bir kereden fazla ziyaret edebileceği atölye durumunda β alanı devridaim için $rcrc$ girişini içerir.

Esnek Atölye (FJc): Atölyenin ve paralel makine ortamlarının geliştirilmesidir. Serideki M makineleri yerine, her iş merkezinde paralel olarak bir dizi özdeş makine bulunan c iş merkezleri vardır. Her işin atölyede takip etmesi gereken yolu vardır; j işi, her bir iş merkezinde yalnızca bir makinede işlem görür ve herhangi bir makine bunu yapabilir. Bir işin her makineyi bir kereden fazla ziyaret edebileceği durumda ise β alanı devridaim için $rcrc$ girişini içerir.

Açık Atölye (Om): M makinenin bulunduğu bu atölye tipinde her iş M makinelerinin her birinde işlenmelidir. İşler makinelerden geçerken belli bir sıra takip edilmez. Ancak, bu işlem sürelerinden bazıları sıfır olabilir. Çizelgelemede her bir iş için bir rota belirlemesine izin verilir ve farklı işlerin farklı yolları olabilir.

İş işleme karakteristikleri ve kısıtlamalar (β) şu şekildedir.

Yayın Zamanı (r_j): Eğer β alanında bu sembol var ise j işi r_j yayınlanma tarihinden önce başlayamaz. Aksi durumda işi başlatmak için zaman kısıtı yoktur. Bitiş zamanı β alanında verilmez, bitiş zamanı amaç fonksiyonu olup γ alanında yer alır.

Önleyici ($prmp$): Önlemler, bir kez başlatıldıktan sonra, tamamlanıncaya kadar bir makinede iş tutmanın gerekli olmadığı anlamına gelir. Zamanlayıcı, bir işin işlenmesini (önemini) herhangi bir zamanda kesebilir ve bunun yerine makineye farklı bir iş koyabilir. β alanında $prmp$ kısıtının eklenmesi, çizelgelemede bu esnekliği sağlar. İşin kalan kısmı başka makinede devam ettirilir. β alanında $prmp$ yok ise makinede işlerin kesilmesine izin verilmez.

Önleyici Kısıtlamalar ($prec$): Tek makineli veya paralel makineli modellerde görülür. Bir işin başlamasından önce başka bir işin yapılması veya bittikten sonra bir işin yapılması gerektiği durumları belirtir.

Sıra Bağlı Kurulum Süreleri (s_{jk}): Sıra bağımlı j ve k işleri arasındaki hazırlık sürelerini ifade eder. β alanında hiç s_{jk} yok ise bütün işlemler bağımsızdır denilebilir.

İş Aileleri (*fmls*): Bu durumda N adet iş, F farklı iş ailesine aittir. Aynı aileden gelen işlerin farklı işlem süreleri olabilir, ancak aralarında herhangi bir kurulum gerektirmeden bir makinede birbiri ardına işlenebilirler. Bununla birlikte, makine bir aileden diğerine geçiyorsa, mesela g ailesinden h ailesine geçiyorsa, bir kurulum gereklidir. Bu kurulum süresi g ve h ailelerine bağlıysa ve diziye bağlıysa, s_{gh} ile gösterilir. Kurulum iş ailesine bağlı değilse s ile gösterilir.

Toplu İşleme (*batch(b)*): Bir makine b gibi bir dizi işi aynı anda işleyebilir; yani, aynı anda en fazla b iş grubunu işleyebilir. Bir gruptaki işlerin işlem süreleri tamamen aynı olmayabilir ve tüm parti yalnızca grubun son işi tamamlandığında bitirilir, bu da tüm partinin tamamlanma süresinin en uzun olan iş tarafından belirlendiğini ima eder.

Arızalar (*brkdown*): Makine arızaları, bir makinenin sürekli olarak bulunamayabileceği anlamına gelir. Paralel özdeş makinelerde belli bir zamanda kullanılacak makine sayısı olarak da ifade edilebilir.

Permütasyon (*prmu*): Akış atölyesi ortamında ortaya çıkabilecek bir kısıtlama, her makinenin İGİG disiplinine göre çalışmasıdır. Bu, işlerin ilk makineden geçtiği sıranın (veya permütasyonun) sistem genelinde korunduğunu gösterir.

Engelleme (*blok*): Engelleme, akış atölyelerinde meydana gelebilecek bir olgudur. Bir akış atölyesi birbirini izleyen iki makine arasında sınırlı bir arabellek içeriyorsa, arabellek dolduğunda, yukarı akış makinesinin tamamlanmış bir işi serbest bırakmasına izin verilmeyebilir.

Beklemesiz (*nwt*): İşlerin birbirini izleyen iki makine arasında beklemesine izin verilmez. Bu, işin herhangi bir makine için beklemek zorunda kalmadan akış atölyesinden geçebilmesini sağlamak için ilk makinedeki bir işin başlama süresinin ertelenmesi gerektiği anlamına gelir.

Devridaim (*rcrc*): Bir işin bir makineyi veya iş merkezini birden fazla kez ziyaret edebildiği durumlarda ortaya çıkar.

Çizelgeleme problemlerinde, minimize edilecek amaç fonksiyonları şu şekildedir:

Makespan (C_{max}): C_i i . Makinedeki işlerin tamamlanma süresini ifade eder. C_{max} ise enbüyük (C_1, \dots, C_n) olarak tanımlanır. Bu tanımlama sistemden ayrılan son işin tamamlanma süresine eşdeğerdir. C_{max} 'ın küçük bulunması genelde makinelerden iyi yararlandığını gösterir.

Maksimum Gecikme (L_{max}): L_i i . makinedeki gecikmeyi ifade eder. L_{max} ise enbüyük (L_1, \dots, L_n) olarak tanımlanır. Teslim tarihlerinin en kötü olduğu durumu ölçer.

Toplam Ağırlıklı Tamamlanma Süresi: N adet işin ağırlıklı tamamlanma sürelerinin toplamıdır ve genellikle akış süresi olarak adlandırılır. Bunun yanı sıra birim başına maliyetlerin daha düşük olarak hesaplandığı indirimli toplam ağırlıklı tamamlanma süresi de vardır.

Toplam Ağırlıklı Gecikme: Bu aynı zamanda toplam ağırlıklı tamamlanma süresinden daha genel bir maliyet işlevidir.

Göreceli Maliyet Hesabı (GMH) (*Apparent Tardiness Cost*) (ATC): Bu aynı zamanda toplam ağırlıklı tamamlanma süresinden daha genel bir maliyet işlevidir. GMH kuralında işler birer birer sıralanır yani makine her serbest kaldığında kalan her iş için bir sıralama tekrar yapılır (Pinedo, 2018:199). GMH, paralel makine çizelgeleme problemini çözmek için bağımsız olarak bir gönderme kuralı olarak kullanılır (Mönch vd., 2005: 2736).

Ağırlıklı Geçici İş Sayısı: Ağırlıklı geciken iş sayısını ifade eder. Bu amaç fonksiyonu pratikte de kolay hesaplanır.

1.3. Çizelgeleme Probleminin Sınıflandırılması

Bir çizelgeleme problemi birçok şekilde sınıflandırılabilir. Bu çalışmada çizelgeleme probleminin sınıflandırılması işlerin sisteme geliş şekline ve makine sayısına göre sınıflandırma olmak üzere iki başlık altında incelenmiştir.

1.3.1. İşlerin Geliş Şekline Göre Sınıflandırılması

Bir problem deterministik ve stokastik olarak iki şekilde sınıflandırılabilir. Tüm veriler biliniyor ise problem deterministik, veriler doğada olasılıklı olduğunda, yani rastgele değerler aldığında stokastik olarak adlandırılır. Çizelgeleme problemlerinin çoğu ya deterministiktir ya da deterministik modellere yaklaştırılabilir. Stokastik bir problemi deterministik bir soruna dönüştürmenin en yaygın yolu ortalama değerlerle çalışmaktır (Sule, 2007: 13).

Çizelgeleme genellikle işlerin sıralanması, hangi işin hangi makineye atanacağını bir çizelge oluşturma ve bir çizelgeleme politikasından oluşur. Çizelgeleme politikası olası durumlarda alınan kararları kapsar ve genellikle stokastik durumlar için geçerlidir.

Deterministik durumlarda ise genellikle sadece işlerin sıralanması ve çizelgelenmesi vardır (Pinedo, 2018: 21-22).

1.3.1.1. Deterministik Problemler

Deterministik çizelgelemeye hiçbir makinenin durmadığı gecikmesiz durumlar örnek olarak verilebilir. Çizelgelenmelerin en baştan oluşturulup değişikliğe izin verilmediği çizelgeleme durumlarıdır (Pinedo, 2018: 22-24). Deterministik çizelgeleme modellerinde, bir dizi iş bir dizi makine tarafından işlenmeli ve belirli performans ölçümleri optimize edilmelidir (Lee, 1997: 2).

Çizelgeleme araştırmasının büyük çoğunluğu, makinelerin çizelgeleme süresi boyunca sürekli çalışır durumda olacağını varsayar. Bununla birlikte, bu varsayım gerçek endüstriyel ortamlarda doğru olmayabilir, çünkü bir makine belirli zaman dilimlerinde makine arızası veya önleyici bakım nedeniyle kullanılamayabilir. Kullanılamazlık kısıtlamaları önleyici bakıma bağlı olduğunda, iki durum mümkündür. Birincisi, bakım sürelerinin önceden sabitlendiği deterministik durumdur. Diğeri, bakım periyotlarının aynı zamanda karar değişkenleri olduğu durumdur. Yani işlerin çizelgelenmesi ve bir makinenin bakımının aynı anda değerlendirildiği durumdur. Kullanılmayan makinelerin, makine arızalarından kaynaklanması stokastik bir durumdur, yani arızalar ve onarım süreçleri rastgeledir (Ying vd. 2010: 199).

Çevrimdışı (statik) zamanlama, tüm görev seti parametreleriyle birlikte eksiksiz bir planlama sırası oluşturur. Ancak, bu statik yaklaşım çok katıdır; tüm parametrelerin sabit olduğunu ve çevresel değişikliklere uyum sağlayamadığını varsayar (Cottet vd., 2002:15).

Temel modelin varsayımları spesifikdir. Bu varsayımların gerçek problemleri çözmek için genişletilmesi gerekmektedir. Temel çizelgeleme modelleri 7 varsayımdan oluşur (Baker ve Trietsch, 2019: 215):

1. N adet iş başlangıçta hazır durumdadır ve işler tek seferde işlenir.
2. Her makinede tek bir iş işlenebilir.
3. İşler için kurulum süreleri iş sırasından bağımsızdır ve işlem sürelerine dâhildir.
4. İş tanımlayıcıları deterministiktir ve önceden bilinir.
5. Makineler sürekli çalışır arıza vermez.
6. Bekleyen iş var ise makineler boşta tutulmaz.
7. Bir işlem başladıktan sonra, bu kesinti olmadan ilerler.

1.3.1.2. Dinamik Problemler

Gerçek dünyadaki üretim ortamında bir sürü belirsizlik ve rastgelelik vardır. Üretim ortamına önemli ölçüde etki edecek belirsizlik kaynaklarına makine arızaları ve yüksek öncelikli işlerin gelmesi örnek verilebilir. Ayrıca işlerin sürelerinin önceden tam belli olmaması da önemli bir belirsizlik kaynağıdır. Bir çizelgeleme problemi için ele alınan model bu belirsizlik biçimlerini de ele almalıdır (Pinedo, 2018: 243).

Belirsizliğin olmadığı durumlar çevrimdışı programlama problemi belirsizliklerin olduğu durumlar ise çevrimiçi programlama problemi olarak ifade edilir. Çevrimdışı bir programlama probleminde tüm veriler (örn. İşlem süreleri, çıkış tarihleri, teslim tarihleri) önceden bilinir ve optimizasyon sürecinde dikkate alınabilir. Buna karşılık, çevrimiçi bir çizelgeleme probleminde problem verisi önceden bilinmemektedir. Bir işin işleme süresi yalnızca tamamlandığı anda bilinir ve yayın tarihi yalnızca bir işin serbest bırakıldığı anda bilinir. Açıkçası, çevrimiçi zamanlama problemleri için algoritmalar, çevrimdışı zamanlama problemleri için algoritmalarından oldukça farklı olma eğilimindedir (Pinedo, 2018: 112). Statik problemlerde bulunan varsayımların belli durumlar altında değişmesi ile dinamik problemler elde edilir. Örneğin bütün işler tek seferde işlenmek yerine, işler aralıklı olarak kullanılabilir hale gelebilir. Bu nedenle bu aralıklı işlerin farklı yayım tarihleri olabilir. Bu durumlarda çizelgeleme yapmak için bu dinamik çizelgeleme gerekir. Yine aynı şekilde makinelerin boşta tutulmaz ve kesinti olmaz varsayımları esnetilerek dinamik problemler elde edilebilir. Dinamik problemlerin varsayımlarından işlerin kurulum süreleri ve tanımlamaların önceden bilinmesi varsayımları da esnetilerek dinamiklik elde edilebilir. Bu durumda işlem süreleri rasgele olduğunda, ortaya çıkan soruna stokastik çizelgeleme problemi denir (Baker ve Trietsch, 2019: 129-216).

Sürekli yeni işlerin geldiği ve mevcut çizelgelemede sapmaların bulunduğu atölye sistemleri dinamik atölye olarak adlandırılır (Dominic, 2004: 70). Farklı bitiş tarihine sahip işlerin olduğu ve bitiş tarihlerinden sapmalar için cezaların olduğu dinamik çizelgeleme, tek makine durumu için bile çok karmaşıktır (Dewan ve Joshi 2000: 3762). Çevrimiçi (dinamik) zamanlama, herhangi bir zamanda bir sonraki görevin seçilmesine olanak sağlar ve o anda tetiklenen görevlerin parametreleri hakkında bilgi sahibidir. Yeni bir olay meydana geldiğinde, seçilen görev, bu olayın gerçekleşme zamanını önceden bilmeden değiştirilebilir. Bu dinamik yaklaşım, daha az bilgi kullandığı ve daha yüksek uygulama yüküne sahip olduğu için statik olandan daha az kesin ifadeler sağlar. Ancak,

öngörülemeyen görevlerin gelişini yönetir ve planlama dizisinin aşamalı olarak oluşturulmasına izin verir. Bu nedenle, çevrimiçi programlama, periyodik olmayan görevler ve anormal aşırı yükleme ile başa çıkmak için kullanılır (Cottet vd., 2002:15).

1.3.2. İşlerin Görüleceği Makine Sayısına Göre Sınıflandırma

Üretim çizelgelemesi, tek bir makine veya birden fazla makine gibi farklı üretim kaynaklarının yapılandırması altında gerçekleştirilmesi gereken bir dizi görevi ifade eder. Önerilen üretim planlama modellerinin çoğu, makinelerin her zaman kullanılabilir olduğunu ve önerilen planların bakım eylemleri nedeniyle üretimdeki kesintileri göz ardı etme eğiliminde olduğunu varsayar (Azimpoor ve Taghipour, 2019: 1).

1.3.2.1. Tek Makineli Model

Bu modelde yalnızca bir makine bulunduğu için, gelen işlerin bu tek makinede işlenmesi gerekir. İşler makine tarafından birer birer işlenir. Her işin işlem süresi ve bitiş tarihi vardır. Ayrıca işlerin öncelik değerleri gibi başka özellikleri de olabilir. Teslim tarihinden sapan işler için bir ceza fonksiyonu da olabilir. En yaygın amaç, genellikle Gecikme Cezası adı verilen geç kalma cezasını en aza indirmek için makinelerdeki işleri sıralamaktır (Sule, 2007: 10). Herhangi bir zamanda, her makine en fazla bir işi işleyebilir ve her iş en fazla bir makinede işlenebilir. Makine ortamı için, en basit model tek makinedir. Tek makine çizelgeleme en basit model olsa da izlenebilir bir modelde çeşitli çizelgeleme konularını gösterir. Birçok farklı performans ölçüsünün ve çeşitli çözüm tekniklerinin araştırıldığı bir bağlam sağladığı için önemlidir (Baker ve Trietsch, 2019: 11).

1.3.2.2. Birden Çok Makineli Model

Çizelgeleme, verilen bir grup işin, bazı kısıtlar ve ölçümler altında bir grup iş istasyonuna atanma işi olarak tanımlanabilir (Tsai vd, 2014: 237). Bu iş istasyonlarının bir arada bulunmalarına atölye denilebilir. Çizelgeleme, atölye planlamasında önemli bir rol oynar. Bir çizelge, işin gerektirdiği her makinede belirli bir işin işlenmesinin planlandığı zamanı gösterir. Ayrıca, işin her makinede ne zaman tamamlanacağını gösterir. Bu nedenle çizelgelemede amaç hem iş hem de makineler için bir zaman çizelgesi oluşturmaktır (Sule, 2007: 8). Çizelgeleme teorisinde paralel sistemler, seri (akış atölyesi) sistemleri ve atölye sistemleri olmak üzere üç temel model vardır. Paralel sistemlerde, işler tek makine modelinde olduğu gibi bir işlemde oluşur; ancak seri ve hibrit sistemlerde işlerin yapısı daha karmaşıktır (Baker ve Trietsch, 2019: 255).

1.3.2.2.1. Paralel Makineli Sistemler

Birçok imalat sektöründe özdeş makinelerin bulunduğu üretim ortamları vardır (Park vd., 2000: 190). Paralel makineli durum, işlerin birbirine özdeş makinelerde ve bu makinelerin herhangi birinde işlenebileceği durumdur. İşlerin bağımlılığı olabilir, yani dizideki önceki iş tamamen işlenmedikçe, dizideki aşağıdaki iş başlatılamayabilir. Amaç, yapım süresini en aza indirmek olabilir (Sule, 2007: 10). Bir başka tanımda ise paralel sistemlerde işler bağımlı veya bağımsız olabilir. Makine tipleri olarak da tek tip, özdeş, ilişkisiz makineler bulunabilir (Brucker, 2007: 107). Bu kategoriler arasında ilişkisiz makineler; aynı işlevi yerine getiren fakat farklı yetenek ve kapasiteye sahip makineleri ifade eder. Uygun sayıda farklı yetenek ve kapasiteli makine işletmelerde esnekliği arttıracığından tekstil, kimya, elektronik imalat vb. birçok sektörde yaygın olarak kullanılmaktadır (Lin vd., 2011a: 353). İşleri makineye atama problemi, öncelik, kullanılabilirlik, üretim hızı, makineye uygunluk (süreç planlama) ve iş yükü dengeleme gibi çeşitli faktörlere dayanmaktadır.

Paralel makine çizelgelemede de birçok değişik yaklaşım ile çizelgeleme problemi ele alınmıştır. Üretilen çizelgeleme algoritması, matematiksel iterasyonlar yoluyla toplam süreyi en aza indirmek gibi bir amacı olan verimli bir çizelge üretir (Rajakumar vd., 2004: 367). Genelde iş planlayıcılar paralel makinelerde işleri makinelere eşit oranda dağıtmayı amaçlar bu toplam akış süresini minimize ederek sağlayabilir. Paralel makineli durum iki önemli adımdan oluşur. İlk olarak hangi işin hangi makinede yapılacağı belirlenir sonra da belirlenen makinede işlerin sıralanması yapılır (Pinedo, 2018: 111). Paralel makinelerde kurulum süresi olmadığı durumlarda işleri teslim sırasına göre sıralayarak teslim tarihine göre gecikme en aza indirilebilir (Baker, 1999: 979).

1.3.2.2.2. Akış Atölyesi

Akış tipi atölyenin özelliği N adet işin M adet makinede aynı sıra ile işlenmesi olarak ifade edilebilir. İşlerin makineler üzerinde işlenme süreleri farklı olabilir. İşlerin başlangıçta kullanılabilir olduğu ve makineler arası bekleme durumunda yeterli tampon alanının olduğu varsayılır. Bu tip çizelgelemede genelde amaç toplam süreyi minimize etmektir. Toplam süre minimize edilirken diğer önemli amaç fonksiyonlarından toplam iş bekleme süresinin minimize edilmesi ve makine boşa bekleme süresinin minimize edilmesini de sağlar (Sule, 2007: 213).

Makinelerin seri olarak sıralandığı durumdur. Bu tip çizelgelemede işler belirli bir sıra ile makinelerden geçerler. Akış tipi çizelgelemede başlangıçta işler bellidir. Makineler sürekli çalışır. Temel modelden farklı olarak makinelerin boşta kalması optimallik için gerekli olabilir (Baker ve Trietsch, 2019: 233). Akış tipi çizelgelemede bütün işlerin her makinede aynı sırada gerçekleştiği duruma Permütasyon çizelgeleme denir. Bu çizelgelemenin performanslı olması için ilk iki makineye göre sıralama yapmak veya toplam akış süresini minimum yapmak için son iki makineye göre sıralama yapmak yeterlidir. Eğer akış tipinde N adet iş ve 2 makine var ise ve amaç fonksiyonu toplam süreyi minimize etmek ise problem “Jhonson kuralı” ile çözülebilir (Baker ve Trietsch, 2019: 288-289). Akış tipi çizelgelemenin atölye tipi çizelgelemeye göre bazı avantajları vardır. Bunlar minimum malzeme kullanımı, makineler arasında daha kolay taşıma ve makine aktivitelerinin daha fazla kontrol edilebilmesi, sade ve daha izlenebilir bir akış, kolay kontrol edilebilirlik ve modern üretim tekniklerin akış tipine daha kolay uygulanabilmesi olarak söylenebilir (Framinan, 2007: 1).

1.3.2.2.3. Atölye Sistemleri

Atölye Tipi: İş atölye sistemleri birçok endüstri dalında kullanıldığı için son derece önemlidir. Atölye ortamında herhangi bir günde, her biri farklı bir işlem sırasına sahip ve farklı makinelerde farklı işlem süreleri olan işler bulunabilir (Sule, 2007: 325). Bir akış tipi çizelgeleme modelinde, tüm işler aynı yolu takip eder. Takip edilecek makineler sabitlendiğinde, ancak her bir iş için gidilecek rota mutlaka aynı olmadığında, modele bir iş atölyesi denir. Bu çizelgeleme türünde bazı önemli makinelerden bazı işlerin birden fazla geçmesi gerektiği durum olabilir (Pinedo, 2018: 179). Akış sırasının bulunmadığı atölye sistemlerinde akış tipi çizelgelemedeki gibi işlerin başladığı ve son bulunduğu makineler belli değildir (Baker ve Trietsch, 2019: 399). Genel atölye tipleri içinde yer alan iş atölye tipi üretim ortamı akış tipi üretimin atölyeye genelleştirilmesidir. İş atölye tipinde aksi belirtilmedikçe her bir işin makinelerde işlenme süreleri farklı kabul edilir (Brucker, 2007: 178).

Açık Atölye: Çizelgelenecek işler için tanımlı bir işlem sırasının bulunmadığı durumdur. Bir işin bütün işlemleri tamamlandığında iş bitmiş sayılır. Açık atölye ortamı çizelgelemede esneklik sağlar ve bu esneklik optimum çizelgeyi bulmayı zorlaştırır (Sule, 2007: 367). Atölye tipi sistemlerde her işin önceden belirlenmiş sabit bir yolu vardır. Açık atölye ise, genellikle işin rotasının önemsiz olduğu ve işlerin sırasının

çizelgeleyen tarafından verildiği ve işlerin yollarının açık olduğu, modeldir (Pinedo, 2018: 217).

1.4. Çizelgeleme Problemi için Önerilen Yöntemler

Son yıllarda teknolojiadaki gelişim farklı tipteki makinelerin işletmelerin üretim aşamasında yer almasını sağlamıştır. Bununla beraber klasik çizelgeleme problemleri bu farklılıkları karşılayacak seviyede değildir. Bu yüzden akademisyenler ve işletmeler bu farklılıkları kapsayacak yeni çizelgeleme çözümleri geliştirmektedir (Kaya, 2014: 7). Kombinatoriyal optimizasyon problemleri, iki farklı yaklaşım kullanılarak çözülebilir veya optimize edilebilir. İlk yaklaşım tam veya kesin algoritmalar kullanmaktır. İkinci yaklaşım daha karmaşık veya daha büyük problemlerde yaklaşık çözümlerin uygulanmasıdır. İkinci yaklaşım sezgisel ve metasezgisel yöntemler olarak ayrılabilir (Xhafa ve Abraham, 2008: 11). Küçük ölçekli optimizasyon problemlerinin çözümünde kesin çözüm yöntemleri kullanılabilir. Bu kesin çözüm yöntemlerine dal ve sınır yöntemi ve bu yöntemin türevleri (dal ve kesme, dal ve fiyatlandırma), lineer programlama, dinamik programlama örnek verilebilir. Problemler NP-Zor olarak sınıflandırılınca problem çözüm yöntemleri ile yeterli zaman içinde çözülemez bu gibi durumlarda metasezgisel algoritmalar kullanılır. Bu metasezgisel algoritmalar tek çözümlü ve popülasyon tabanlı olmak üzere iki kategoride değerlendirilebilir. Tek çözümlü metasezgisel algoritmalara Yerel Arama, Açlık Sezgiseli, BT, TA, Yinelemeli Yerel Arama örnek olarak verilebilir. Popülasyon tabanlı algoritmalara GA, Karınca Kolonisi Optimizasyonu ve Dağılım Arama gibi yöntemler örnek olarak verilebilir. Son yıllarda özellikle gerçek yaşam problemlerinin çözümünde sezgisel ve kesin çözüm yöntemlerinin iyi yönlerinin birleşimi olan hibrit uygulamalar sıkça kullanılmaktadır (Jourdan vd., 2009: 620).

1.4.1. Kesin Çözüm Yöntemleri

Kesin çözüm yöntemleri sadece belirli problemlerin çözümünde etkili olan ve geçerliliği az olan yöntemlerdir. Bu yöntemlere karışık tam sayı programlama, dal ve sınır algoritması, Gezgin Satıcı Problemi (GSP)'nin çözümünde kullanılan yöntemler ve öncelik kurallarına göre sezgisel yöntemler örnek verilebilir (Zupan vd., 2006: 32).

Bu araştırma alanında, bazı temel çizelgeleme modelleri için dal ve kesme gibi kesin çözüm yöntemleri ve ayrıca iyi performans garantili yaklaşım algoritmaları üretilmiştir. Dal-sınır algoritmasının yeni bir formu olan dal ve fiyat adı verilen yeni

algoritma da paralel makine çizelgelemesinde başarılı bir şekilde uygulanmıştır. Deterministik çizelgelemede bir başka araştırma yönü NP-Zor problemler için Polinomal Zaman Yaklaşma Şemaları alanıdır (Pinfjoedo, 2018: 548). Bu alanda kullanılan yöntemlere dal-sınır, dinamik programlama, iş çiftlerinin yer değişimi, tam sayılı programlama ve ödünleşim eğrileri örnek olarak verilebilir. Bu yöntemlerden dal-sınır ve dinamik programa teknikleri ile belirli kısıtlar altında çok sayıdaki aday çözüm elimine edilir. Ancak her iki yöntem de büyük boyutlu problemlerde etkili değildir. Geliştirilen tam sayı programlama modelleri de küçük ölçekli problemlerde etkin olmasına karşın büyük problemlerin çözümünde başarılı değildir (Eren ve Güner, 2002: 39-40).

Bazı çizelgeleme problemleri, doğrusal programlar, maksimum akış problemleri veya taşıma problemleri gibi iyi bilinen kombinatorial optimizasyon problemlerine indirgenerek etkili bir şekilde çözülebilir. Diğerleri dinamik programlama ve dal ve sınır yöntemleri gibi standart teknikler kullanılarak çözülebilir (Brucker, 2007: 11).

Dal ve sınır yöntemi, tüm uygulanabilir çözümleri akıllıca numaralandırma fikrine dayanır (Brucker, P. 2007: 51). Çizelgeleme problemi NP-zor problem olması nedeni ile tamsayı programlama çözüm için yeterli değildir. Yine de erken ve geç tamamlama problemleri için geliştirilen sezgiselleri test etmek amacı ile tam sayı programlama kodları kullanılabilir (Zhu ve Heady, 2000: 304).

1.4.2. Sezgisel Çözüm Yöntemleri

Karmaşık kombinatorial optimizasyon problemlerini çözmek için tam ve kesin yöntemlerin kullanılması genellikle pratik olmayan hesaplama sürelerine yol açar. Bu yüzden araştırmacıların büyük çoğunluğu, bu tür problemlerde, yaklaşık çözümlerin bulunduğu yöntemlere yönelmiştir. Yaklaşık çözüm yöntemlerinde, makul ve pratik hesaplama zamanlarında kabul edilebilir çözümleri elde etmek için en uygun çözümleri bulma garantisi feda edilir. Temel yaklaşım algoritmalarına sezgisel algoritma denir. Sezgisel algoritmalar yapıcı algoritmalar ve yerel aramalar olarak ikiye ayrılabilir. Yapıcı algoritmalar bazı durumlar dışında en iyi sonucun garanti edilmediği öncelik kurallarının kullanılmasıdır. Yerel aramalar ise çözüm uzayında komşu çözümleri arama esasına dayanır. Yerel aramada en büyük problem algoritmanın yerel minimuma takılmasıdır. Yerel aramadaki yerel minimuma takılma problemini çözmek için meta – stratejiler geliştirilmiştir (Xhafa ve Abraham 2008: 12-13). Araştırmacıların uzun yıllara dayanan araştırmaları sonucunda, zor kombinatorial problemler için çözüm algoritması ortaya

konulamamış ve böyle bir algoritmanın mevcut olmadığı belirtilmiştir. Optimizasyonda da bu tip çözümsüz problemlere NP-Zor problemler denir. Çizelgeleme problemlerinde sıkça karşılaştığımız bu tip bir problemde daha mütevazı bir hesaplama gereksinimi olan, ancak en iyi çözümü garanti etmeyen sezgisel çözüm prosedürünü kullanmak daha iyi olabilir (Baker ve Trietsch, 2019: 6). Yaklaşık çözüm olarak adlandırılan bu algoritmalar, gerçek optimumun sabit bir yüzdesi içinde olduğu garanti edilen çözümler üretir. NP-Zor optimizasyon problemlerinin en başarılı yöntemlerinden biri, olan yerel (veya komşu) arama olarak bilinen yöntemdir. Yerel arama yöntemleri genellikle sadece en uygun olduğu garanti edilemeyen uygulanabilir çözümler sunar. Kesin performans garantisi olmayan herhangi bir yaklaşım “sezgisel” olarak değerlendirilebilir. Daha iyi yöntemler mevcut değilse, bu tür yaklaşımlar pratik durumlarda yararlıdır (Brucker, 2007: 51). Toplam akış süresi gibi bazı amaç fonksiyonları için, işleri sıralama ile optimal bir çözüm elde edilebilir. Toplam ağırlıklı gecikme gibi diğer amaç fonksiyonları için basit bir çözüm prosedürü mevcut değildir ve kombinatorial optimizasyonun daha genel tekniklerine başvurulmalıdır (Baker ve Trietsch, 2019: 71).

Toplam ağırlıklı gecikme problemi için popüler bir sezgisel yaklaşım, her işe öncelik değeri atayan sevk kurallarını kullanmaktır. Bir işin önceliği, işlem süresi, son tarih, ağırlık ve makine ortamına ilişkin parametreler gibi iş parametrelerine göre hesaplanabilir. ETZ kuralı ve AKİSÖ en basit kurallardır. Bu basit kurallar belirli sorun koşullarında iyi çözümler üretmez. Bu yüzden örneğin toplam ağırlıklı gecikmeyi düşürmek için birçok kural önerilmiştir (Park vd., 2000: 190). Çoğu sezgisel algoritmada liste çizelgeleme yöntemi kullanılır. Yöntemde, bir makine bir işi işlemek için kullanılabilir hale geldiğinde, o sırada makinede işlenebilen işlerden biri belirli bir öncelik kuralına göre seçilir ve makinede çizelgelenir. Benzer şekilde, bir iş işlenmeye hazır hale geldiğinde, işi işleyebilenler arasından bir öncelik kuralına göre bir makine seçilir ve iş makinede çizelgelenir (Shim ve Kim, 2007: 136).

1.4.3. Metasezgisel Çözüm Yöntemleri

Bir metasezgisel algoritma, yaklaşık- en uygun çözümü bulmak için öğrenme stratejileri, bilgileri ve alt sezgisel algoritmaları kullanan yinelemeli bir algoritma olarak tanımlanabilir (Osman ve Laporte, 1996: 513-514) Metasezgisel yöntemlerin çözüm alanını keşfetmenin akıllı bir yolu olduğu söylenebilir. Metasezgisel yöntemler kesin çözümler gibi yavaş ve basit sezgisel algoritmalar gibi çözüm uzayında yerel optimuma takılmazlar. Metasezgiseller, çözüm uzayında olası tüm çözümlerin aranması amaçlanır.

(Xhafa ve Abraham, 2008: 14). Daha karmaşık problemlerin çözümü için geliştirilen çözüm algoritmalarına Genetik Algoritma (GA), Tabu arama (TA) örnek olarak verilebilir (Eren ve Güner, 2002: 40; Zupan vd., 2006: 32). Bu türdeki problemlerin çözümünde Sürü tabanlı algoritmalarından Ateş Böceği Algoritması, Parçacık Sürü Optimizasyonu, Guguk Kuşu Algoritması, Yapay Arı Kolonisi Algoritması ve Karınca Kolonisi Optimizasyonu da kullanılmaktadır. Yapay Sinir Ağları ve Kesikli Olay Simülasyonu da çözümde kullanılan yöntemlerdir (Zupan vd., 2006: 32-33). Önceki çalışmalardan elde edilen sonuçlar BT'nin verimli bir şekilde iyi sonuçlar elde edebileceğini öne sürdü, TA'nın BT'den daha iyi olup olmadığı tartışılmakla birlikte kesin bir cevap bulunamamıştır. TA performansının üstünlüğü, akıllı komşuluk aramasından ve toplam süreyi hızlı hesaplamasından kaynaklanıyor olabilir. Ancak aynı komşuluk yapısı ve güncelleme hesaplamaları diğer arama yaklaşımlarına da uygulanabilir. Ayrıca, daha önce test edilen bir arama tekniği, değişen bir darboğaz yapısına dayanıyordu ve bu teknik de tekrar gözden geçirilebilir. Ancak şu anda, komşuluk arama algoritmaları, çizelgeleme sorununun büyük sürümlerini çözmede en etkili yaklaşımı temsil etmektedir (Baker ve Trietsch, 2019: 421).

1.4.4. Hibrit Algoritmalar

Hibrit algoritmalar kesin ve sezgisel algoritmaların iyi yönlerini birleşiminden oluşur. Bu birleşimler 5 farklı şekilde gerçekleştirilebilir (Jourdan vd., 2009: 620):

- Yerel arama algoritmasını daha geniş komşuluklarda yapabilmek için kesin çözüm yöntemlerinin kullanılması.
- Yerel arama algoritmasını birden fazla çalıştırarak bilgi tabanı oluşturulması ve problemin daha küçük problemler olarak ifade edilmesi. Oluşturulan küçük problemlerin kesin çözüm yöntemleri ile çözülmesi.
- Yapıcı sezgiseller ile sınırların genişletilmesi.
- Tam sayı programlarından elde edilen bilgilerin yerel arama ya da yapıcı sezgisellerde kullanılması.
- Metasezgisel algoritmaların bazı prosedürlerinin kesin çözüm yöntemleri ile çözülmesi

Tasarım olarak iki farklı tasarım yönteminin kombinasyonları ile metasezgisel algoritma oluşturulabilir. Bunlar (Düşük –Yüksek) ve (Ardışık – İşbirlikçi) yöntemlerin birleşimidir.

Düşük-Ardışık: Bu sınıf, belirli bir yöntemin başka bir yöntemle gömüldüğü algoritmalara karşılık gelir; gömülü yöntemin ardışık olarak yürütülmesi gerekir, yani global yöntemin yürütülmesi gömülü yöntemin elde ettiği sonuçlara bağlıdır. Kesin bir yaklaşımı iyileştirmek için sezgisel bir yaklaşım kullanıldığında bu tür iş birliği yaygındır. Meta-sezgiler arasındaki iş birliği bağlamında, en çok önerilen yaklaşım evrimsel bir algoritma çalıştırmak ve daha sonra en iyi çözümlere yönelik aramayı yoğunlaştırmak için yerel arama başlatmaktır. Kesin ve sezgisel yöntemler arasındaki iş birliğini düşünürsek, en doğal yaklaşım, kesin yöntemin arama stratejisini geliştirmek için sezgisel bir tasarım tasarlamaktır.

Düşük- İşbirlikçi: Bu yöntemde bir yöntemin ögesi başka bir öge ile değiştirilir ve bu sayede metasezgisel yöntemin etkinliği artırılır. Bu sınıfa GA ve yerel aramanın beraber çalıştığı Memetik algoritmalar örnek verilebilir.

Yüksek–Ardışık: Bu sınıfta, farklı yöntemler bağımsızdır ve sırayla uygulanır. Bu iş birliği şeması genel hibridizasyon için en çok temsil edilenidir. Genel yaklaşım önce sezgisel algoritmanın çalışması daha sonra kesin algoritmanın çalışması şeklindedir.

Yüksek- İşbirlikçi: Bu sınıfta aynı problemi çözmek için algoritmalar aynı anda uygulanır. Algoritmalar arasında bilgi akışı olur ve durdurma kriteri sağlanınca işlem durur. Aynı problemde BT ve Dal ve Sınır algoritmasının birlikte kullanılması bu sınıfa örnek olarak verilebilir.

1.5. Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi

Sıra Bağımlı Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi (SBHSİPMÇP)'inde her bir işin her bir makinede tamamlanma süreleri farklıdır. Bu problemde makinelerin işleri işlerken, işlere göre farklı kapasiteler ile çalışırlar. Aynı zamanda her bir işin işlenmek için öncesinde o makinede bir hazırlık süresine ihtiyaç vardır. İşlerin makinede işlenen son işe göre veya o makinede ilk işlenme durumlarına göre farklı hazırlık süreleri vardır. Bu çalışmada ele alınan SBHSİPMÇP'nde amaç toplam işleri en az sürede bitirecek şekilde uygun bir çizelgeleme elde etmektir.

1.5.1. Matematiksel Model

Problemin daha iyi anlaşılması adına çalışmanın bu bölümünde SBHSİPMÇP'nin modellenmesine ait gösterimler ve formüller verilmiştir. Bu formüller (Rabadi 2006: s

87) tarafından geliştirilmiş karma tam sayılı program için yazılmıştır. SBHSİPMÇP'ne ait gösterimler Tablo 1'de verilmiştir.

Tablo 1. SBHSİPMÇP'ne Ait Gösterimler

Gösterim	Gösterimin Açıklaması
C_j	j işinin tamamlanma süresi
$p_{j,k}$	j işinin k makinesinde işlenme süresi
$S_{i,j,k}$	j işinin k makinesinde i işinden sonra işlenmesi için gereken hazırlık süresi
$S_{0,j,k}$	j işinin k makinesinde ilk işlenen iş olması durumunda gereken hazırlık süresi
$x_{i,j,k}$	Eğer j işi i işinden hemen sonra işlendi ise 1 diğer durumlarda 0.
$x_{0,j,k}$	Eğer j işi k makinesinde ilk işlenen iş ise 1 diğer durumlarda 0.
$x_{j,0,k}$	Eğer j işi k makinesinde son işlenen iş ise 1 diğer durumlarda 0.
PM	Pozitif büyük bir sayı

SBHSİPMÇP'nin Matematiksel Modeli Tablo 2'de belirtilmiştir.

Tablo 2. SBHSİPMÇP'nin Matematiksel Modeli

Minimize C_{max}	(1)
$\sum_{\substack{i=0 \\ i \neq j}}^n \sum_{k=1}^m x_{i,j,k} = 1 \quad \forall j = 1, \dots, n$	(2)
$\sum_{\substack{i=0 \\ i \neq j}}^n x_{i,h,k} - \sum_{\substack{j=0 \\ j \neq h}}^n x_{h,j,k} = 0 \quad \forall h = 1, \dots, n, \forall k = 1, \dots, m$	(3)
$C_j > C_i + \sum_{k=1}^m x_{i,j,k}(S_{i,j,k} + p_{j,k}) + PM \left(\sum_{k=1}^m x_{i,j,k} - 1 \right) \quad \forall i = 0, \dots, n, \forall j = 1, \dots, n$	(4)
$\sum_{j=0}^n x_{0,j,k} = 1 \quad \forall k = 1, \dots, m$	(5)
$x_{i,j,k} \in \{0,1\} \quad \forall i = 0, \dots, n, \forall j = 0, \dots, n, \forall k = 1, \dots, m$	(6)
$C_0 = 0$	(7)
$C_j \geq 0 \quad \forall j = 1, \dots, n$	(8)

Tablo 2'deki eşitliklerden 1. eşitlikte amaç fonksiyonu belirtilmiştir. 2. eşitlikteki kısıt ile her işin sadece bir makinede bir defa işlenmesi sağlanmıştır. 3. eşitlikteki kısıt ile işlenen işlerin her birinin sadece bir işin öncesinde bulunması kısıtıdır. 4. eşitlikteki kısıt ile tamamlanma süresini hesaplamak ve işlenen bir işin sadece bir işten sonra işlenme kısıtıdır. 5. kısıtta makinelerde ilk başta birer işten fazla iş işlenemeyeceğinin kısıtıdır. 6. kısıt x karar değişkeninin ikili kodlandığını belirtir. 7. kısıt aylak değişkenin tamamlanma

süresinin 0 olduğunu belirtir. 8. değişken ise tamamlanma sürelerinin negatif olmayacağını garanti eder.

1.6. Çizelgeleme ile İlgili Çalışmalar

Bu bölüm altında çizelgeleme ile ilgili genel çalışmalar ve özelde İlişkisiz paralel makine çizelgeleme problemi SBHSİPMÇP ilgili çalışmalara kronolojik sıralamayla yer verilmiştir. Çalışmada aynı kıyas veri setini kullanan akademik çalışmalar (*) ile belirtilmiştir.

Park vd. (2000), çalışmalarında özdeş paralel makinelerdeki çizelgeleme problemleri ile ilgilenmişlerdir. Her işin bir işlem süresi, bir bitiş tarihi ve gecikmeyi cezalandırmak için bir ağırlığı vardır. İşlerin, makineden bağımsız kurulum süreleri olduğu varsayılır. Amaç, yapılan çizelgeleme ile ağırlıklı gecikmenin toplamını en aza indirmektir. Araştırmacılar kullanılan parametreleri, ileri beslemeli geri yayımlı yapay sinir ağları kullanarak optimize etmeye çalışmışlardır. 4032 adet rastgele üretilmiş veri setinden yararlanmışlar ve sonuç olarak %4 lük bir iyileştirme elde etmişlerdir.

Kim vd. (2002), çalışmasında hazırlık zamanı bulunan işlerin paralel makinede çizelgelenmesi için Benzetimli Tavlama (BT) yöntemini kullanmışlardır. BT yöntemine yeni 6 farklı komşuluk yöntemi önermişlerdir. Farklı makine ve iş sayılarında komşuluk yöntemleri ile geliştirdikleri BT yöntemini, geleneksel BT ve komşuluk arama sezgiselleri ile kıyaslamışlardır. Küçük problemlerde komşuluk arama sezgiselinin hızlı olduğu, problem büyüdükçe BT içeren sezgisellerin daha hızlı çözüm ürettiği belirtilmiştir. Geliştirilen komşuluk tabanlı BT algoritmasının diğer iki algoritmadan daha başarılı olduğu ifade edilmiştir.

Bilge vd. (2004), çalışmalarında paralel makine çizelgelemede toplam gecikmeyi minimize etmek için TA yöntemini kullanmışlardır. Çalışmasında önceki çalışmaların aksine makinelerin farklı oldukları durumu değerlendirmiş her bir işin işlem süresi, tamamlanma süresi gibi tanımların makine tiplerine göre değişebileceğini belirterek, problemi gerçek yaşam problemlerine benzetmişlerdir. Çalışmada çözüm ararken komşuluk oluşturmada aday listeler oluşturulmuş ve bu aday listeler için 3 farklı strateji belirlenmiştir. Bu listeler sayesinde hem algoritmanın arama hızını arttırmışlar hem de daha uygun değerler elde etmişlerdir. Tabu liste tanımlamalarında da iki farklı liste yapısı kullanmışlardır. Çalışmada WinMeta adını verdikleri bir yazılım geliştirilmiştir. Daha önce GA ile çözülmüş olan 20,40 ve 60 işin 2 ve 4 makinede çizelgelenme problemine

odaklanmışlardır. 40 ve 60 adet işte TA'nın önceki çalışmadan daha iyi olduğunu ortaya koymuşlardır.

Rajakumar vd. (2004), araştırmalarında makineler arasında iş akışı dengeleme ölçüsü ile paralel makine çizelgeleme üzerinde çizelgeleme stratejilerinin performansını değerlendirmeye odaklanmıştır. Bir işi ilk kullanılabilir makineye atamak yerine, bir makine üzerindeki toplam iş akışını KİSÖ, UİSÖ ve rastgele sıralama olmak üzere 3 farklı stratejide hesaplamışlardır. Yaptıkları hesap sonucunda, yeni atanan işin toplam işleri en az olan makineye atanması gerektiğini belirtmişlerdir. Farklı makine sayıları ile simülasyon yapılmış ve KİSÖ stratejisinin paralel makine çizelgeleme problemleri üzerinde dengeli iş yükü çizelgeleri ürettiği bulunmuştur.

Gupta ve Sivakumar (2005), çalışmalarında tek makinede 2 aileye ait kurulum sürelerinin olduğu N tane bağımsız işin çizelgeleme problemini ele almışlardır. Çalışma ortalama çevrim süresini en aza indirmek, ortalama gecikmeyi en aza indirmek ve makine kullanımını en üst düzeye çıkarmak olmak üzere 3 amaç içermektedir. Birleştirici sümüle çizelgeleme kullanılmış ve amaç fonksiyonlarının hepsini en uygun hale getirmek için karşılaştırmalı programlama kullanılmıştır. Karşılaştırmalı programlama ortalama çevrim süresi, ortalama gecikme ve makine kullanımı açısından ETZ ve KİSÖ sezgiselleri ile karşılaştırılmıştır. Geliştirilen algoritma ile ortalama çevrim süresinde ortalama %16,7 azalma, ortalama gecikmede ortalama %25,6 azalma ve makine kullanımında ortalama %21,6 iyileşme olduğunu belirtilmiştir.

Perez vd. (2005), yarı iletken baskı devresi üreten bir fabrikada farklı önemlere sahip işler için, iş ailelerine sahip tek bir toplu işlem makinesinde çizelgeleme problemi ile ilgilenmişlerdir. Çizelgelemede amaç toplam ağırlıklı gecikmeyi en aza indirmektir. Problem iki aşamada ele alınmış önce parti içinde işler sıralanmış daha sonra ise toplam gecikmeyi minimize etmek için partiler sıralanmıştır. Partileri sıralarken, sıralamalar için öncelik kuralları olarak ETZ ve GMH kuralları kullanılmıştır. Oluşturulan partileri sıralamak için bu kurallar ile dinamik programlama ve ayrışma sezgiseli de kullanılarak iki aşamalı 6 farklı varyasyon oluşturulmuştur. Araştırmacılar 30,40,50 ve 60 işi 3,4 ve 5 iş ailesinde 4 ve 8 parti büyüklüğünde oluşturdukları varyasyonlar ile denemişlerdir. Çizelge oluşturmada GMH kuralının en iyi performans gösterdiği belirtilmiştir. Ayrışma sezgiseli de iş yığınlarını sıralamada başarılı olmuştur. 3 iş ailesinden daha büyük işlerde önerilen hibrit algoritmanın kullanılması gerektiği belirtilmiştir.

Mönch vd. (2005), çalışmasında üstteki çalışmaya benzer şekilde yarı iletken baskı devresi üreten bir fabrikada farklı önemlere sahip işler için toplam ağırlıklı gecikmeyi en aza indirmeyi amaçlamışlardır. Çizelgeleme oluştururken ilk yaklaşım önce sabit grupların oluşturulması ve sonra GA kullanarak bu grupların makinelere atanması ve sıralanması şeklindedir. İkinci yaklaşımda ise önce GA ile işler makinelere atanır sonra makinelerde gruplar oluşturulur ve sıralanır. İş grupları oluştururken sıralama kuralı olarak GMH kullanılmıştır. Makinelerin özdeş olduğu, aynı iş ailesinde işlerin aynı işlem süresine sahip olduğu ve işlerde kesintiye izin verilmediği kabul edilmiştir. Farklı varyasyonlar ile algoritmalar denenmiş ilk yaklaşımın daha hızlı olduğu belirtilmiştir.

(*) Helal vd. (2006), hazırlık süreli ilişkisiz paralel makine çizelgelemede TA yöntemini kullanmıştır. Başlangıç çözümü oluşturmak için KİSÖ kuralı kullanılmıştır. Önerilen TA yöntemi aynı problemde kullanılan Al-Salem (2004)' in Bölümleme Sezgiseli yöntemi ve karışık tamsayı programlama yöntemi ile kıyaslanmıştır. TA'da farklı makine ve iş kombinasyonlarına göre farklı tabu liste boyutları ve arama aşamaları denenmiştir. Küçük boyutlu problemlerde TA ve Bölümleme Sezgiseli yöntemi kesin çözümler ile karşılaştırılmıştır. 4 makineden daha büyük problemlerde TA'nın daha önce önerilen Bölümleme Sezgiselinden daha iyi olduğu vurgulanmıştır.

(*) Rabadi vd. (2006), çalışmalarında işler arasında hazırlık sürelerinin bulunduğu paralel makine çizelgeleme problemine odaklanmıştır. Çalışmanın amacı toplam işlem süresini minimize etmektir. Yöntem olarak rasgele öncelikli bir metasezgisel algoritma ve bölümleme sezgiseli kullanılmıştır. Yeni önerdikleri metasezgisel algoritma çözüm uzayında çözümü ararken yerel minimumlara takılmamak bazen en iyi önceliği olmayan işleri de sıralamaya alır. Bölümleme sezgiselinde ise çizelgeleme 3 aşama olarak ele alınmıştır. İlk aşamada işler makinelere atanmıştır. 2. aşamada sezgisel algoritma geliştirilmiştir. 3. aşamada ise her bir makineye atanan işler GSP olarak ele alınmış ve çözülmüştür. Küçük çaplı problemlerin çözümünde sezgisel algoritmalar tamsayı programlama ile karşılaştırılmıştır. Çalışma sonucu olarak farklı denemeler sonucunda 3 örnek dışında 2250 örnekte kendi önerdikleri algoritmanın daha başarılı olduğu vurgulanmıştır.

De Paula vd. (2007) tarafından yapılan çalışmada, kurulum zamanı gerektiren paralel makine çizelgelemede DKA algoritması kullanılmıştır. Rastgele oluşturdukları veri setinde 3 farklı yerel arama algoritması ile destekledikleri DKA algoritması kullanmışlardır. DKA algoritmasını literatürde bulunan algoritmalar ile kıyaslamışlardır.

Özellikle 60 işten fazla olduğu durumlarda DKA algoritmasının başarılı olduğu ifade edilmiştir.

Shim ve Kim (2007), özdeş paralel makinelerde toplam gecikmeyi minimize etmek için dal ve sınır yöntemini kullanmıştır. Dal-sınır yönteminde ilk üst sınırı belirlemek için BT yöntemi kullanmışlardır. Farklı problem büyüklüklerinde geliştirdikleri yöntemi sınımışlar ve geliştirilen algoritmanın 30 iş ve 5 makineye kadar olan problemler için en uygun çözümleri bulabileceğini göstermiştir. Daha büyük problemler için sezgisel algoritmaların geliştirilmesi gerektiğini vurgulamıştır.

Chen (2009a) tarafından yapılan çalışmada, işler arasında hazırlık süresi bulunan bir tekstil firması ele alınmıştır ve sıra bağımlı hazırlık süresi olan işlerin çizelgelenmesi için bir sezgisel algoritma geliştirilmiştir. Bütün hazırlık sürelerinin bilindiği, kesintilerin olmadığı, işlerin yarıda kalmadığı varsayımları altında geliştirilen algoritma dal sınır algoritması ile kıyaslanmış ve sezgisel algoritmanın daha hızlı olduğu vurgulanmıştır.

Chen (2009b), bitiş tarihi kısıtları altında hazırlık süresi gerektiren paralel makine çizelgeleme problemi ile ilgilenmiştir ve iki farklı sezgisel algoritma geliştirmiştir. Bunlardan ilki GMH kuralı ile BT yönteminin birleşimi olan bir hibrit metasezgiseldir. Diğer sezgisel rasgele iniş sezgiselidir. Rasgele iniş sezgiseli, diğer hibrit sezgiselin aksine en kötü durumları hesaplamaya katmayan bir yöntemdir. Çalışma için veri seti oluşturulmuş ve bu veri seti ileri çalışmalarda da kullanılmıştır. Oluşturulan veri setine algoritmalar uyarlanmış, GMH kuralı ile kıyaslanmıştır. 960 kombinasyonun 498 tanesinde yeni önerilen her iki algoritma da en iyi performansı göstermiştir. Kalan kombinasyonların 434 tanesinde ilk önerilen hibrit metasezgisel ve 28 tanesinde rasgele iniş sezgiseli en iyi sonucu göstermiştir. Genel anlamda ilk önerilen algoritmanın daha başarılı olduğu vurgulanmıştır.

(*) Arnaout vd. (2010), kurulum zamanı olan paralel makine çizelgeleme probleminde iki aşamalı bir Karınca Klonisi Optimizasyon (KKO) metasezgisel algoritma önermişlerdir. İlk aşamada önce işlerin makinelere atanması ikinci aşamada ise sıralama yapılması amaçlanmaktadır. Önerdikleri algoritma KKO'nun performansını iyileştirmek için bir yerel arama prosedürü kullanmaktadır. Önerilen algoritma aynı problemde daha önce kullanılan Tabu Arama, Parçalama Sezgiseli, (Rabadi vd. 2006)'nın önerdiği algoritmaları ile kıyaslanmıştır. Kombinatoriyal optimizasyon problemlerinde

KKO algoritmasının iki aşamalı olarak kullanabileceğini göstermişlerdir. Önerilen algoritmanın kıyaslananlardan daha başarılı olduğu vurgulanmıştır.

Chyu ve Chang (2010), iş ve makine bağımlı hazırlık süresi bulunan ilişkisiz makine çizelgeleme problemini ele almışlardır. Aynı anda toplam ağırlıklı akış süresini ve toplam ağırlıklı gecikmeyi en aza indirmek olmak üzere iki amaç vardır. Bu amaçları gerçekleştirmek için dört farklı yöntem önerilmiştir. Bu yöntemlerden ikisi pareto yakınsak GA diğer ikisi de çok amaçlı BT temellidir. Çaprazlama işlemi yeni bireyin rasgele oluşturulan ondalık sayılar ile belirlendiği yönteminin diğer 3 yöntemden daha başarılı olduğu gösterilmiştir.

Niu vd. (2010), özdeş paralel makinelerde toplam gecikmeyi minimize etmek için hibrit parçacıklı sürü optimizasyon (PSO) algoritması önermişlerdir. Önerilen PSO geliştirmek için Klon Seçim Algoritması ile birleştirilir. Klon seçim algoritmasının eklenmesi ile PSO içerisindeki sürü çeşitliliği artırılmış ve erken yakınsama problemini ortadan kaldırarak algoritmanın yerel minimumlara takılmasını önlemiştir. 250 farklı örnekte, önerilen algoritma geleneksel GA ve standart PSO algoritma ile kıyaslanmıştır. Sonuçlar önerilen algoritmanın daha başarılı olduğunu göstermiştir. Önerilen algoritma 250 problemin 237'sinde optimal çözüme ulaşmıştır.

Ying ve Cheng (2010), dinamik özdeş paralel makinelerde, sıra bağımlı hazırlık süreli çizelgeleme problemi için yinelemeli açgözlü sezgisel algoritma önermişlerdir. Önerilen algoritma işlerin dinamik geldiği problem yapısına ve gecikmeyi minimize etme amaç fonksiyonuna uygun bir algoritmadır. Bu sayede gerçek problemlere uygundur. Önerilen algoritma literatürde bulunan veri setinin denendiği 5 farklı algoritmanın en iyi sonuçları ile kıyaslanmıştır. Sonuçlar önerilen algoritmanın başarısını göstermektedir.

Sabouni ve Mansouri (2010), paralel makine çizelgeleme probleminde toplam tamamlanma süresi ve maksimum geç bitirmeyi minimize etmeyi amaçlamışlardır. Problemi çözmek için baskın olmayan paralel toplu programlama algoritması adında sezgisel algoritma önermişlerdir. Ayrıca başlangıç çözümü için GA kullandıkları GA tabanlı üç farklı algoritma önerilmiştir. Önerilen algoritmalarının 2 farklı amacı minimize ederken dahi toplam geç bitirme minimize etme sezgisel kuralından daha başarılı sonuçlar verdiği raporlanmıştır. Algoritmalar kıyaslandığında GA tabanlı paralel toplu programlama algoritmasının önerilen diğer algoritmalarından başarılı olduğu belirtilmiştir.

Behnamian vd. (2011), hazırlık süreli paralel makine çizelgeleme probleminde iki amaç ile ilgilenmişlerdir. Bu amaçlardan biri işlerin maksimum tamamlanma süresini minimize etmektir. Diğer amaç ise TZÜ felsefesine göre önce ve geç bitirmeleri minimize ederek işleri tam zamanında bitirmektir. Bu amaçlar doğrultusunda hibrit bir metasezgisel algoritma geliştirmişlerdir. Bu hibrit algoritma; başlangıç çözümü oluşturmak için KKO algoritmasını, yerel minimumlara takılmamak için BT ve bulunan çözümü daha da iyileştirmek için yerel aramada DKA algoritmasını kullanır. Geliştirilen algoritmanın KKO algoritması kısmında karınca feromonları belli bir değer arasında sınırlandırılır. Bu sayede feromonların değeri belli bir olasılık ile tanımlanır. Bu tanımlama ile önerilen üç aşamalı hibrit algoritma, BT ve DKA algoritmasının birleşimi, KKO ve DKA algoritmasının birleşimi ve DKA algoritmaları ile kıyaslanmıştır. Farklı kombinasyonlarda 300 farklı problem belirtilen algoritmalar ile çözülmüştür. Sonuçlar önerilen üç aşamalı metasezgisel algoritmanın diğerlerinden daha başarılı olduğunu göstermektedir.

(*) Chang ve Chen (2011), ilişkisiz paralel makine çizelgeleme probleminde hazırlık sürelerinin bulunduğu problemi ele almışlardır. Bir çizelgeleme sırasının baskınlık özelliklerini metasezgisel algoritmalar ile belirlemişlerdir. Geliştirilen baskınlık özellikler, BT ve GA'ya eklenerek sonuçlar Al-Salem (2004) tarafından geliştirilen algoritma ile kıyaslanmıştır. Geliştirilen baskın özellikli GA ile küçük boyutlu problemlerde optimal sonuçların elde edildiği vurgulanmıştır. Önerilen algoritmanın büyük problemlerde de karşılaştırılan algoritmadan daha başarılı olduğu belirtilmiştir.

Chyu ve Chang (2011), çalışmasında iş ve makine bağımlı hazırlık süresi bulunan ilişkisiz makine çizelgeleme problemi ile ilgilenmişlerdir. İşlerin tamamlanma süresini ve ortalama gecikmeyi minimize etme olmak üzere iki bulanık optimizasyon amacı vardır. Bu amaç için 3 farklı metasezgisel algoritma geliştirilmiştir. Bu metasezgisellerden ikisi çok amaçlı BT temellidir. Sayısal sonuçlar amaç fonksiyonunun uygunluğuna bakan algoritmanın diğer iki algoritmadan daha başarılı olduğunu göstermektedir.

Driessel ve Mönch (2011), çalışmalarında sıraya bağlı kurulum süreleri, öncelik kısıtlamaları ve hazır zamanları bulunan özdeş paralel makine çizelgeleme problemi ile ilgilenmişlerdir. Probleme amaç, toplam ağırlıklı gecikmeyi minimize etmektir. Bu amaç doğrultusunda 7 farklı komşuluk algoritması denenilen DKA metasezgisel algoritma önerilmiştir. Önerilen DKA algoritmaları Kurulum ve Hazırlık Süreli GMH sıralama

kuralı ve GA ile kıyaslanmıştır. Sonuçlar DKA algoritmasının üstünlüğünü ortaya koymaktadır.

Lin vd. (2011a), özdeş paralel makine çizelgeleme probleminde, hazırlık süresi ve işlerin yayımlanma tarihleri kısıtları altında toplam geç kalmayı minimize etmeyi amaçlamışlardır. Bu amaç için BT'dekine benzer şekilde batma sıcaklığı kabul değeri olan yinelemeli açlık yaklaşımı algoritma önermişlerdir. Önerilen algoritma hem Yerel Arama sezgisel algoritmanın basitliğini hem de BT yönteminin soğuma mekanizmasını birleştirmektedir. Önerilen algoritma daha önce bu problem seti için önerilen sezgisel algoritmaların en iyi sonuçları, Sınırlandırılmış BT ve sabit sıcaklı yinelemeli açlık yaklaşım algoritması ile kıyaslanmıştır. Sonuçlar önerilen algoritmanın üstünlüğünü ortaya koymaktadır.

Lin vd. (2011b), sıra ve makine bağlı kurulum süresi olan, paralel makineli çizelgeleme problemi için Yinelemeli Yerel Arama Algoritması kullanmışlardır. Chen (2009) tarafından oluşturulan veri seti kullanılmış ve yine geliştirilen algoritma Chen (2009) ile kıyaslanmıştır. Algoritmanın etkinliğini ölçmek için sonuçlar üzerinde t – testi yapılmış önerilen algoritmanın daha üstün olduğu belirtilmiştir.

Lin vd. (2011c), birden fazla amacı (tamamlanma süresini minimize etmek, ağırlıklı tamamlanma süresini minimize etmek, ağırlıklı gecikmeyi minimize etmek) olan ilişkisiz paralel makine çizelgeleme problemi ile ilgilenmişlerdir. Toplam tamamlanma süresini minimize etmek için 2 aşamalı sezgisel bir algoritma önermişlerdir. Önerdikleri algoritmayı daha önce geliştirilen sezgisel algoritmalar ve GA metasezgiseli ile kıyaslamışlardır. Ağırlıklı toplam süreyi minimize etmek için yeni bir sezgisel algoritma geliştirmişlerdir. Bu sezgisel algoritma AKİSÖ ve DBT sıralama kuralları, geliştirilen sezgiseller ve GA metasezgiseli ile kıyaslanmıştır. Toplam ağırlıklı gecikmeyi minimize etmek için 2 sezgisel önerilmiştir. Bu iki yeni önerilen sezgisel algoritma GA metasezgiseli ile kıyaslanmıştır. Kendi önerdikleri sezgisel algoritmaların mevcuttan daha başarılı olduğunu belirttikleri çalışmada en başarılı yöntemin GA metasezgiseli olduğu söylenmektedir.

Vallada ve Ruiz (2011), ilişkisiz paralel makinelerde toplam gecikmeyi minimize etmek için GA kullanmışlardır. GA'nın performansını farklı iki yerel arama tekniği (yerel arama ve hızlı yerel arama) ile 4 farklı şekilde kullanmışlardır. Algoritmalar farklı iş ve makine sayılarında denenmiştir. Bu varyasyonlardan sonuçları iyi olan algoritmaları

literatürde daha önce önerilen 2 farklı algoritma ile kıyaslamışlardır. Önerdikleri yerel arama algoritmasının başarılı olduğu belirtilmiştir.

Cesaret vd. (2012), sipariş kabullü çizelgeleme problemi için bir TA algoritması geliştirmişlerdir. Ancak, kabul ve sıralama kararlarının aynı anda alınması gerektiğinde TA algoritmasının çözüm alanını etkili bir şekilde ayarlayamadığını belirtmişlerdir. TA algoritmasını geliştirmek için TA algoritmasını her iterasyonda olasılıklı yerel arama prosedürü ile desteklemişlerdir. Geliştirdikleri algoritmayı küçük veri seti için Karma Tamsayı Doğrusal Programlama ile veri seti büyüdükçe ise iki sezgisel algoritma ile kıyaslanmıştır. Geliştirilen algoritmanın iki sezgisel algoritmadan daha uygun olduğu belirtilmiştir.

Chen (2012), çalışmasında ilişkisiz paralel makine çizelgemedeki geciken iş ve hazırlık süreli ve hazırlık süresinin eşit olmadığı durumu ilk defa kendisinin ele aldığını belirtmiştir. Bu problemi çözme amacı ile yinelemeli hibrit bir metasezgisel algoritma önermiştir. Başlangıç çözümünü dört farklı sezgisel kural ile oluşturmuştur. Bu kurallardan en iyi çalışmaya en uygun genişletilmiş makine tabanlı ilk çözüm oluşturucu kuralı kullanılmıştır. Geliştirilen metasezgisel algoritma başlangıç çözümünü iyileştirmek için DKİ ve TA algoritmalarını birleştirir. 4 farklı komşuluk arama stratejisi denenilen yeni önerilen algoritma literatürde bulunan yinelemeli Tabu Arama algoritması ile kıyaslanmıştır. Kendi önerdikleri algoritmanın değişik problem setlerinde daha başarılı olduğu belirtilmiştir.

(*) Fleszar vd. (2012), tarafından ilişkisiz paralel makine çizelgeleme probleminin çözümüne çoklu başlangıç DKİ algoritması önerilmiştir. Başlangıç esnasında 10 farklı başlangıç ile çözüm uzayında daha fazla alanın taranması amaçlanmıştır. Bu başlangıç çözümlerinden en iyisi saklanır ve en iyi çözüm geliştirilmeye çalışılır. Küçük komşuluk arama ve büyük komşuluk arama adını verdikleri iki sezgisel yöntem ile başlangıç çözümlerinden en iyisi geliştirilmeye çalışılır. Önerilen algoritma daha önce aynı problem setinde önerilen algoritmalar ile kıyaslanmış ve daha başarılı olduğu vurgulanmıştır.

Kaplan ve Rabadi (2012), çalışmasında askeri yakıt ikmal problemini özdeş paralel makine çizelgeleme problemi olarak ele almışlardır. Bu problemde amaç fonksiyonu toplam ağırlıklı gecikmeyi minimize etmektir. Bu amaç fonksiyonunda sıkça kullanılan GMH kuralını askeri kurallara göre değiştirerek uyarlanmış GMH kuralını ortaya koymuşlardır. Bu yeni kuralın performans ölçümünü çizelgeleme problemlerinde

sıkça kullanılan sezgisel yöntemlerden BT yöntemi ile kıyaslamışlardır. Hesaplamalarında, önerilen kuralın küçük boyutlu problemler için optimal çözümden BT'den daha kötü performansına sahip olmasına rağmen, problem boyutu arttığında BT'den daha iyi performans göstermesinin daha olası olduğunu belirtilmiştir.

Lamothe vd. (2012), hazırlık zamanı ve takvim kısıtları altında paralel makineleri çizelgelerken toplam gecikmeyi minimize etmeyi amaçlamışlardır. Mevcut ETZ ve GMH sezgiselleri araştırılırken GMH temelli iki sezgisel algoritma önerilmiştir. Önerilen algoritmalar diğer sezgisel algoritmalar ile kıyaslanmış ve daha başarılı bulunmuştur. Mevcut çözümü daha da iyileştirmek için SA temelli 3 farklı komşuluk yapısı önerilmiştir. Bunlar Temel BT ve çalışmada önerilen işlem ekleme adımlarını kullanan algoritma, Farklı komşuluk yapıları kullanan (çeşitli ekleme, değiştirme, taşıma ve işlem grupları) BT algoritmaları ile kıyaslanmıştır. Çalışma benzetimli tavlama komşuluk yapılarının önemini ve önerilen metasezgisel algoritmaların önerilen sezgisel algoritmalara göre çok daha hızlı sonuçlar verdiğini ortaya koymaktadır.

Li vd. (2012), birden fazla amaç fonksiyonu (toplam süre ve toplam gecikmeyi minimize etmek) bulunan paralel makine çizelgeleme problemi için sezgisel ve kesin çözüm içeren çözüm yöntemleri önermişlerdir. Çözüm için baskın olmayan sıralama genetik algoritma ve algoritmasına bulanık mantık kontrolörü ekleyerek iki sezgisel algoritma önermişlerdir. Bulanık algoritmada genetik algoritmanın kullandığı sabitler olan çaprazlama olasılığı ve mutasyon olasılığını amaç fonksiyonuna göre çıktı değerlerinden elde etmek için bulanık kontrol kullanılmıştır. Ayrıca çalışmada genelde tek amaç fonksiyonunda kesin çözüm yöntemi olan iki fazlı çözüm yöntemi, birden fazla amaç fonksiyonu olan problemin çözümüne göre uyarlanmıştır. 960 adet test probleminde sezgisel algoritmalar denenmiş ve bulanık tabanlı algoritmanın optimal çözümler verdiği belirtilmiştir. Kesin çözüm yöntemini en fazla 10 işe kadar çözümlerde kullanıldığı belirtilmiştir.

Tran ve Beck (2012), mantık tabanlı ayrıştırıcı algoritma ile hazırlık süreli ilişkisiz paralel makine problemini çözmeyi amaçlamışlardır. Bender ayrıştırması işlerin makinelere atanma ana probleminde karma tam sayılı programlama modeli, işlerin sıralanması alt problem kısmında ise GSP algoritmasını kullanır. Benders yaklaşımı en uygun çözüm bulunana kadar ana problem ile alt problem arasında yinelemeler yapar. İlk koşul yineleme yaparken bulunan sonucun ana problemdeki sonuçtan iyi olması ikinci koşul ise bulunan sonucun en iyi sonuç olmasıdır. Önerilen çözüm küçük problemlerde

Karışık Tamsayı Programlama ile büyük problemlerde ise KKO ve BT algoritmaları ile kıyaslanmıştır. Sonuçlar algoritmanın başarılı olduğunu göstermektedir.

Ying ve Lin (2012), hazırlık süreli paralel makine çizelgeleme probleminde sürü tabanlı bir metasezgisel yöntem olan Yapay Arı Kolonisi (YAK) yöntemini kullanmışlardır. Çizelgeleme probleminde amaç toplam gecikmeyi minimize etmektir. Öncelikli müşterinin işi önce yapılır kuralı uygulanmıştır. Bu kural ile bulunan çözümlere yerel arama eklenerek sonuçlar geliştirilmiştir. Önerilen YAK tabanlı algoritmanın performansı, iyi bilinen bir kıyaslama problem setinden toplam 960 örnek ile test edilmiştir. Algoritma sonuçları aynı veri setini kullanan 3 farklı yöntem ile kıyaslanmıştır. Kıyaslanan yöntemler ile işlem sürelerinin yaklaşık olmasına rağmen işlerin ortalama gecikme sürelerinde, önerilen YAK tabanlı algoritmanın daha başarılı olduğu belirtilmiştir.

(*) Ying vd. (2012), yılında hazırlık süreli paralel makine çizelgelemede toplam gecikmeyi minimize etmek için kısıtlanmış BT algoritması önermişlerdir. Yerel arama kısmında geliştirme sağlamak için daha önce çözümde gereksiz olan çözümlerin hesaplanıp kısıtlanması amaçlanmıştır. Geliştirilen algoritma aynı problemin çözümünde kullanılan Bölümlenme Sezgiseli, TS, Meta-RaPS, BT ve KKO algoritmaları ile kıyaslanmıştır. Teorik çalışmalar ile endüstriyel uygulamalar arasındaki boşluğu doldurmaya yardım ettiklerini belirttikleri yöntemin kıyaslandığı algoritmalarından daha başarılı olduğu belirtilmiştir.

Figielska (2013), çalışmasında kaynak kısıtlamalı, sıraya bağımlı kurulum maliyeti olan paralel makine çizelgeleme problemi için iki aşamalı bir algoritma önermiştir. İlk aşamada hazırlık sürelerine bakılmaksızın işlerin toplam tamamlanma süresi minimize edilir. İlk aşamada olası çözümlerden sadece iyilerinin alındığı sütün oluşturma algoritması kullanılmıştır. İkinci aşamada da KKO algoritması ile çizelgelenenler toplam hazırlık süreleri dikkate alınarak sıralanır. 25 ve 50 iş 2,4 ve 6 farklı makine kombinasyonlarında geliştirilen algoritma ile çizelgelenmiştir.

Lin vd. (2013), ilişkisiz paralel makine çizelgeleme probleminde ağırlıklı gecikmeyi minimize etmeyi amaçlamışlardır. Bu amaç için yeni bir sezgisel kural içeren KKO algoritması önermişlerdir. Önerilen algoritma KKO algoritmasına yeni bir sezgisel algoritma, makine yeniden seçme ve yerel arama olmak üzere 3 farklı yenilik katmaktadır. Önerdikleri yeni sezgisel kural, daha önce benzer problemlerde önerilen iki

farklı sezgisel kurallar ile karşılaştırılmış ve kendi önerdikleri sezgisel kuralın daha başarılı olduğunu tespit etmişlerdir. Önerilen yerel arama farklı yerel aramalar ve KKO algoritmasının varsayılan yerel arama prosedürleri ile kıyaslanmıştır. Çalışmanın diğer bir amacı önerilen yeniliklerin algoritmadan çıkarılarak gerekliliğini araştırmaktır. Başlangıç çözümünün sezgisel kural ile oluşturmanın pek de etkili olmadığı tespit edilmiştir. Makine yeniden seçme ve yerel arama adımlarının eklendiği KKO algoritmasının daha başarılı olduğu belirtilmiştir.

(*) Arnaout vd. (2014), daha önceki çalışmasında (Arnaout vd. 2010) geliştirdikleri ACO I algoritmasını daha da geliştirmiş ve ACO II adını verdikleri yeni bir algoritma önermişlerdir. Aynı problem ve veri setini dikkate aldıkları çalışmada önceki çalışmadan farklı olarak arama yaparken çözümde tıkanıklık yaşandığı sırada “feromon yeniden başlatma” adımını algoritmaya eklemişlerdir. Bu çalışmada, önerilen ACO II algoritmasını daha önceki çalışmadaki algoritmalar ve Ying vd. (2012) tarafından önerilen kısıtlayıcı BT algoritması ile kıyaslamışlardır. ACO II'nin ACO I'den %8.12 daha başarılı olduğu belirtilmiştir. Önerilen ACO II'nin diğer algoritmalarından daha başarılı olduğu vurgulanmıştır.

Nogueira vd. (2014), TZÜ felsefesine uygun olarak, erken ve geç bitirmelerin cezalandırıldığı ilişkisiz paralel makine çizelgeleme problemi ile ilgilenmişlerdir. Problemin yaklaşık çözümü için ARUA temelli 3 farklı sezgisel algoritma önermişlerdir. Önce iki aşamalı ARUA algoritması önerilmiştir. ARUA algoritması yeniden bağlama tekniği ile geliştirilmiş ve ARUA+PR algoritması önerilmiştir. Son olarak önerilen algoritmaya yinelemeli yerel arama algoritması da eklenerek ARUA+PR+ILS hibrit metasezgisel algoritması önerilmiştir. Küçük boyutlu problemler için karışık tam sayılı program ile çözüm önerilmiş, büyük boyutlu problemlerde ise önerilen 3 algoritma kıyaslanmıştır. Önerilen ARUA+PR ve ARUA+PR+ILS algoritmalarının ARUA algoritmasından daha iyi sonuç verdiği belirtilmiştir. Bu sayede çalışmada ARUA algoritmasının iyileştirildiği ifade edilmiştir.

Eroğlu vd. (2014a), çalışmalarında hem işlerin bölünebilir olduğu hem de işler arasında hazırlık sürelerinin olduğu bir tekstil fabrikasında çizelgeleme problemini ele almışlardır. Siparişlerin bölünerek kaç alt işe dönüşecekleri fabrikadaki uzman görüşü ile belirlenmiştir. Yöntem olarak GA kullanılmıştır. Çizelgeleme problemindeki amaç fonksiyonu, işlerin tamamlanma sürelerinin minimize edilmesidir. Önerilen algoritma ile,

75 makine ve 111 işle ilgili problemlere, makul bir işlemci zamanında çözüm bulunabileceği gösterilmiştir.

(*) Erođlu vd. (2014b), hazırlık süreli, ilişkisiz paralel makine çizelgeleme probleminde GA kullanmışlardır. GA'ya yerel arama ekleyerek GALA adını verdikleri yeni bir algoritma önermişlerdir. Önerilen algoritmayı yaygın kullanılan kıyaslama veri setinde denemişler ve sonuçları Arnaout vd. (2010) tarafından geliştirilen ACO ve Chang ve Chen (2011) tarafından geliştirilen algoritma ile kıyaslanmıştır. Sonuçta önerilen algoritmanın daha başarılı olduğu vurgulanmıştır.

Haddad vd. (2014) çalışmasında ilişkisiz makinelerde hazırlık süreli çizelgeleme probleminde toplam tamamlanma süresini minimize etme problemi ile ilgilenmiştir. Bu amaç doğrultusunda DKİ ve yinelemeli Yerel Arama ile bir metasezgisel algoritma geliştirmişlerdir. Başlangıç çözümü için uyarlanmış en kısa süreli kurala göre bir açgözlü yaklaşım benimsemişlerdir. İstatiksel analizler çalışmanın literatürdeki sonuçları geliştirdiğini göstermektedir.

(*) Lin ve Ying (2014), hazırlık süreli ilişkisiz paralel makine çizelgelemede, toplam işlem süresini minimize etmek için bir Hibrit Yapay Arı Kolonisi algoritması önermişlerdir. Çalışmada önerilen hibrit algortmada YAK algoritması BT ile desteklenerek çözüm uzayında esneklik sağlanmıştır. Önerilen algoritma daha önce bu problemin çözümünde önerilen algortmalar ile kıyaslanmış ve sonuçta önerilen algortmanın performansının en iyisi olduğu vurgulanmıştır.

Kerkhove ve Vanhoucke (2014), Belçika'da dokuma fabrikasında ilişkisiz paralel makineleri çizelgelerken ağırlıklı geç kalma/gecikme problemi ile ilgilenmişlerdir. Bu problemin modeli oluşturulurken daha önce değinilmeyen coğrafi olarak dağınık üretim yerleri ve teknisyenlerin bakabilecekleri makine sayısına göre sıraya bağlı geçiş süreleri de modele katılmıştır. Modelin büyük problemlerde çözümü iki fazda yapılmıştır. İlk fazda önce öncelik kısıtları da değerlendirilerek açlık yaklaşımı ile bir başlangıç çözümü oluşturulmuş, daha sonra GA ve yerel aramada BT dan oluşan hibrit bir metasezgisel algoritma ile başlangıç çözüm iyileştirilmiştir. İkinci fazda sınırlı teknisyenden dolayı yaşanacak gecikmelerden dolayı mevcut çizelgelemenin iyileştirilmesi için bazı sıralama kuralları önerilmiş ve denenmiştir. Küçük ve büyük olmak üzere iki farklı veri seti kullanılmıştır. Sonuçta geliştirilen metasezgisel algortmanın başlangıç çözümünü iyileştirdiği ve teknisyenlere bağlı gecikmelerin de %23 iyileştirildiği belirtilmiştir.

Avalos-Rosales vd. (2015), makine sıra bağımlı hazırlık süreli paralel makine çizelgeme problemlerinde küçük boyutlu problemler için yeni bir matematiksel model ve büyük boyutlu problemler için birden fazla başlangıç noktalı DKİ algoritması önermişlerdir. Önerilen metasezgisel algoritmanın geliştirme fazında iki aşamalı arama sezgiseli kullanır. Bu iki aşamalı arama ayrı ayrı ve birlikte kullanılarak denenmiştir. Önerilen algoritma aynı veri setinin çözümünde en başarılı olan Vallada ve Ruiz (2011) tarafından geliştirilen GA ile kıyaslanmış ve sonuçta kendi önerdikleri algoritmanın başarılı olduğu belirtilmiştir.

Canıyılmaz vd. (2015), işler arası hazırlık süresi bulunan ilişkisiz paralel makine çizelgeme problemi ile ilgilenmişlerdir. Probleme amaç hem toplam maksimum tamamlanma süresini en küçükleme hem de toplam gecikmeyi en küçükleme. Çözüm için metasezgisel algoritmalarından KKO ve GA'yı kullanmışlardır. Yeni aday çözümü seçerken önerdikleri yeni komşuluk arama algoritmasını her iki sezgisel algoritmada da denemişlerdir. 131 gerçek fabrika işi işler arasında hazırlık sürelerine göre 31 gruba ayrılmıştır. 1 hafta içinde sıralanması gereken problemde her iki algoritmada başarılı olduğu halde KKO algoritmasının daha başarılı olduğu belirtilmiştir.

Lin ve Ying (2015), çalışmasında birden fazla amacı olan ilişkisiz makine çizelgeme problemini ele almışlardır. Bu amaçlar toplam süreyi minimize etmek, ağırlıklı gecikmeyi ve ağırlıklı toplam süreyi minimize etmektir. 3 farklı sezgisel kural ile başlangıç çözümü oluşturulmuştur. Başlangıç çözümünü geliştirmek için birden fazla noktalı benzetimli tavlama yöntemi geliştirmişlerdir. Geliştirdikleri algoritmada Lin vd. (2011) tarafından geliştirilen problem setini kullanmışlardır. Geliştirilen algoritmayı BT, GA ve GA'nın farklı versiyonları olarak sunulan algoritmalar ile kıyaslamışlardır. Aynı problemlerde önerilen algoritmanın daha başarılı olduğu vurgulanmıştır.

Afzalirad ve Rezaeian (2016), makine uygunluğu ve öncelik kısıtlamaları ile ilişkisiz paralel makine çizelgeme problemi ile ilgilenmişlerdir. Amaç fonksiyonu toplam geciken iş sayılarını minimize etmek olan çalışmada bir karma tam sayılı model önerilmiştir. Karma tam sayılı model yanı sıra GA ve KKO algoritmalarının benzetilmiş tavlama algoritmasının (Metropolis kabul kuralı) kabul stratejisi ile kullanıldığı iki farklı metasezgisel algoritma önerilmiştir. Sonuçlar, önerilen hibrit karınca kolonisi optimizasyonunun, büyük boyutlu test problemlerini çözmede önerilen hibrit genetik algoritmadan istatistiksel olarak daha iyi performansa sahip olduğunu göstermektedir.

Liao vd. (2016), makinelerin birden fazla özelliğe göre kurulum sürelerinin bulunduğu ilişkisiz paralel makine çizelgeleme problemi ile ilgilenmişlerdir. Tekstil makineleri ile ilgili bu uygulamada amaç toplam gecikmeyi minimize etmektir. Bu amaç doğrultusunda, üç aşamalı sezgisel bir yöntem ve yine üç farklı yerel arama yöntemi kullanan GA önerilmiştir. Önerilen GA kullanılarak, kendi önerdikleri sezgisel yöntem ile buldukları başlangıç çözümünü daha da geliştirmişlerdir. Her iki yöntem ile mevcut çözümden daha iyi sonuçlar elde ettiklerini ve büyük problemleri makul sürede çözebildiklerini belirtmişlerdir.

Wang vd. (2016) çalışmasında Dağıtım algoritmaları tahmini ve yinelemeli yerel arama algoritmalarının hibrit hali ile hazırlık süreli ilişkisiz makine çizelgeleme probleminin çözümü ile ilgilenmişlerdir. Geliştirilecek başlangıç çözümünü oluşturmak için önce bütün işlerin bütün makinelerde ortalama işlem süreleri hesaplanmıştır. İlk çözüm bu işlerin en kısa süre ile sıralanmasından oluşur. Daha sonra her adımda 2 farklı yerel arama stratejisinde olasılık matrisi güncellenir. Vallada ve Ruiz (2011) ile kıyasladığı algoritmada, küçük örneklerde ve 530 büyük örnekte kıyaslanan algoritmalarından daha başarılı olduğu belirtilmiştir.

(*) Cota vd. (2017), çalışmasında hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi ile ilgilenmişlerdir. Problemi çözmek için Otomata öğrenmeli uyarlamalı geniş komşuluk arama metasezgisel algoritma önerilmiştir. Çözümde 6 farklı ekleme ve 6 farklı çıkarma sezgisel yöntemin kullanım olasılıkları otomata öğrenmesi sayesinde hesaplanır. Ayrıca çözüme yeni Macar algoritması kullanan ekleme modeli de önerilmiştir. Hem öğrenme otomati hem de Macar algoritması bu çalışmada bu tür çizelgeleme probleminin çözümü için ilk defa sunulmuştur. Önerilen yöntem daha önceki çalışmalarda kullanılan ACOI ve ACOII yöntemleri ile kıyaslanmıştır. Çözümlerin %88'inde yeni önerilen yönteminin başarılı olduğu belirtilmiştir.

Manupati vd. (2017), ilişkisiz paralel makine çizelgenirken, hazırlık süreleri yanı sıra ikincil kaynak kısıtlarının değerlendirmesini de ele almışlardır. Birden fazla amacın bulunduğu problem için bulanık karışık tamsayı doğrusal olmayan programlama ile bir model önerilmiştir. Büyük problemler için yeni çok amaçlı evrimsel yapay bağışıklık, baskın olmayan sıralama genetik algoritma önermişlerdir. Öncelikle çok amaçlı fonksiyonda pareto-optimal alanları bulmak için geleneksel baskın olmayan genetik algoritma kullanılmıştır. Önerdikleri algoritmanın eksik kalan erken yakınsama problemini çözmek için çok amaçlı partikül sürüsü optimizasyonu algoritması

denenmiştir. Bu 3 metasezgisel algoritma 24 farklı iş makine senaryosunda farklı amaç fonksiyonlarına göre denenmiştir. Çalışmada belirtilen sonuçlar yeni algoritmanın başarısını ortaya koymaktadır.

Diana vd. (2018), ilişkisiz paralel makinelerde işler arası hazırlık süresinin bulunduğu ve işlerin başlangıç sürelerinin farklı olduğu çizelgeleme problemine yeni bir metasezgisel yöntem önermişlerdir. Önerilen algoritma DKİ ve YYA algoritmalarını birlikte kullanır. Geliştirdikleri algoritmayı Lin ve Hsieh (2014), Lin vd. (2013) ve Arnout vd. (2010) algoritmaları ile kıyaslamışlardır. Sadece bitiş tarihlerinin geniş dağılım gösterdiği durumlarda Lin ve Hsieh (2014) daha başarılı olurken diğer çoğu durumda önerilen algoritmanın başarılı ve hızlı olduğu belirtilmiştir.

Gedik vd. (2018), hazırlık süresi gerektiren paralel makine çizelgelemede toplam işlem süresini minimize etmek için bir kısıt programlama modeli önermişlerdir. Bu problem tipi için kısıt program modelinin ilk defa önerildiğini belirtmişlerdir. Çalışmada önerdikleri algoritmayı küçük boyutlu problemlerde mevcut bulunan sezgisel algoritmalar ile kıyaslamışlar ve kendi geliştirdikleri modelin daha başarılı olduğunu belirtmişlerdir. Büyük boyutlu problemlerde çözümünde, sezgisel algoritmaların en iyi çözümden %3 daha kötü sonuç ile yaklaşık çözümler elde ettiği belirtilmiştir.

Lee (2018), özdeş paralel makinelerde toplam gecikmeyi en aza indirmek için yeni bir sıralama kuralı ve rasgele yenilenen açgözlü metasezgisel bir algoritma önermiştir. Önerilen sıralama kuralı bitiş ve işlem zamanı indeksi ve GMH kuralının birleşiminden oluşur. Çalışmada bilinen sezgisel sıralama kuralları ve mevcut yöntem kıyaslanmıştır. Yeni önerilen sıralama kuralının, mevcut yöntemlere göre yaklaşık %30 daha iyi performans gösterdiği belirtilmiştir. Önerilen metasezgisel algoritma yeni önerilen sıralama kuralı ile başlangıç çözümü oluşturur. Yeni önerilen metasezgisel yöntemin gösterdiği performans, diğer sezgisel yöntemlerle başladığı için ve bir başka metasezgisel algoritma olan TA yönteminden daha başarılıdır.

Azimpoor ve Taghipour (2019), çalışmalarında, stokastik makine arızası riskini göz önünde bulundurarak denetim planlaması ile ilgili tek makine çizelgeleme problemini ele almıştır. Çalışmada Genetik Algoritma (GA) kullanarak ve yine GA'ya bir sezgisel algoritma ekleyerek 2 farklı model geliştirmişlerdir. Geliştirilen modeller öncelikle küçük çaplı problemlerde optimal çözüm ile karşılaştırılmış ve geliştirilen algoritmanın optimal

çözümü verdiği doğrulanmıştır. Sonuçta çalışma orta ve büyük boyutlu problemler için GA'nın bakım arızalarını dikkate alarak çizelgelemede etkili olduğunu göstermiştir.

Báez vd. (2019), hazırlık süreli paralel makine çizelgelemede toplam tamamlanma süresini minimize etmek için yeni bir metasezgisel algoritma önermişlerdir. ARUA ve DKA algoritmalarını birleştirdikleri hibrit algoritma, yapıcı ve geliştirici olmak üzere iki aşamadan oluşmaktadır. Geliştirici aşamada rasgele DKA algoritması kullanılmaktadır. Hibrit algoritma önce küçük problemlerde matematiksel model oluşturularak kesin çözüm ile kıyaslanmıştır. Hibrit algoritmanın sonuçları DKA ve daha önceki çalışmalarda en iyi sonuç veren algoritma ile kıyaslanmıştır. Sadece 2 makineye kadar problemlerde daha önce önerilen algoritmanın daha hızlı olduğu belirtilmiştir. Çalışmada verilen sonuçlar önerilen algoritmanın başarısını göstermektedir.

He vd. (2019), sipariş kabullü çizelgeleme problemleri için Sparrow ismini verdikleri bir algoritma geliştirmişlerdir. Bu algoritma evrimsel algoritmaların ve yerel arama algoritmalarının hibrit biçimde bir Memetic algoritmadır. Bu hibrit algoritmada evrimsel algoritma için (Biased random key genetic algorithm) BRKGA algoritması kullanılmıştır. BRKGA Evrimsel Arama'nın bir varyantıdır. Standart Genetik Algoritma yöntemleriyle karşılaştırıldığında, BRKGA kodlama çözümlerinde daha fazla esneklik sunar ve daha iyi çözümler üretir. Yerel arama için ise çizelgeleme kabullerinde başarılı bir şekilde kullanılan geniş komşuluk arama sezgisel algoritması (ALNS) kullanılmıştır. ALNS genel aramadan daha az duyarlı olduğu için bir sezgisel arama ile üretilen başlangıç çözümüne ihtiyaç duyar. Sparrow algoritması BRKGA'nın geniş çözüm yayılımı ve ALNS'nin hızlı çözüm bulma özelliklerini birleştirmiştir. Literatürde bu iki algoritma ilk defa hibrit hale getirilmiştir. Sparrow literatürde iyi bilinen algoritmalar ile standart veri setlerini çözerken karşılaştırılmış ve bu veri setlerinde zaman olarak daha avantajlı olduğu belirtilmiştir.

(*) Arnout (2020), çalışmasında hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi ile ilgilenmiş ve problemi Solucan Optimizasyon Algoritması (WO) ile çözmüşlerdir. Çözümü daha önce aynı kıyas probleminin çözümünde kullanılan TA - Helal vd. (2006), ACO - Arnout vd. (2014), RSA- Ying vd. (2012), YAK ve HABC Lin ve Ying (2014) ve GALA-Yılmaz Eroglu vd. (2014) ile kıyaslamıştır. Sonuçlar karşılaştırılan yedi algoritma içinde WO ve HABC algoritmalarının diğer algoritmalarından daha başarılı olduğunu göstermektedir.

(*) de Abreu ve de Athayde (2020), çalışmalarında hazırlık süreli ilişkisiz paralel makine çizelgeleme problemini çözmek için GA, DKİ ve BT algoritmalarını kullanan hibrit bir metasezgisel algoritma önermişlerdir. Önerilen algoritmayı hem Arnout (2010) hem de Vallada ve Ruiz (2011) de kullanılan veri seti ile test edilmiştir. Önerilen algoritmanın belirlenen test kümesinde başarılı sonuçlar verdiği vurgulanmıştır.

Diana ve de Souza (2020), hazırlık süreli ilişkisiz paralel makine çizelgeleme probleminde DKİ algoritmasını sezgisel algoritmalarda yerel arama aracı olarak kullanmayı amaçlamışlardır. Çalışmada üç farklı DKİ algoritması önerilmiş ve önerilen algoritmalar Yinelemeli Açgözlü Arama, GA, YAK algoritmalarında yerel arama aracı olarak kullanılmıştır. Aynı problemin çözümünde geliştirilen farklı 2 sezgiselin klasik yerel arama kullanan YAK ve GA'dan performans olarak daha üstün olduğu Yinelemeli Açgözlü Arama ile aynı performansı gösterdiği belirtilmiştir. YAK, GA ve IGS algoritmalarında yerel arama olarak DKİ kullanıldığında daha önce önerilen iki metasezgisel algoritmadan daha iyi sonuçlar elde edildiği gösterilmiştir.

Kim ve Kim (2020), çalışmasında işler arasında hazırlık zamanının bulunduğu 3D yazıcıların çizelgeleme problemini ele almışlardır. Problem önce tek makine çizelgeleme problemine dönüştürülmüş ve GSP olarak ele alınmıştır. Geliştirilen GSP tabanlı hibrit GA ile çizelgeleme probleminin etkin bir şekilde çözüldüğü gösterilmiştir.

Kim vd. (2020), çalışmada işlerin bölünebildiği, sıra bağımlı hazırlık sürelerinin olduğu özdeş paralel makine çizelgeleme problemi ile ilgilenmişlerdir. Toplam gecikmenin minimize edilmesinin amaçlandığı çalışma için 2 farklı kodlama (tek sıralı ve iki sıralı) ve 3 farklı kod çözme yöntemi önermişlerdir. Bu yöntemler ile 2 farklı kodlama ve 3 farklı çözme kombinasyonu ile 6 farklı SA ve 6 farklı GA algoritması önermişlerdir. Farklı problem boyutlarında yapılan çözümler, kod çözme yöntemleri 1 ve 2'nin iki sıralı kodlama ile iyi çalıştığını, kod çözme yöntemi 3'ün ise tek sıralı kodlama ile iyi çalıştığını ve en başarılı algoritmalar olduğunu göstermektedir. BT temelli algoritmaların GA temelli algoritmalarından daha başarılı olduğu belirtilen çalışmada en başarılı algoritma olarak SO3 (tek sıralı kodlama ve 3. kod çözme yöntemi olan NT algoritması) bulunmuştur. Toplam gecikmenin 10, 20, 30 ve 100 boyutlu iş için sırasıyla %51,0, %63,5, %69,6 ve %68,4 oranında azalttığı belirtilmiştir.

Soleimani vd. (2020), çalışmalarında ilişkisiz paralel makine çizelgeleme probleminde ortalama gecikmeyi ve güç tüketimini minimize etmeyi amaçlamışlardır.

Çalışmada yeni bir elitizm yaklaşımı önerilen Kedi Sürüsü Algoritması (CSA-Elit), GA ve İnteraktif Yapay Arı Kolonisi Algoritmaları ile kıyaslanmıştır. Problemin küçük boyutlarında Lindo paket programı ile matematiksel model çözülmüştür. Problemler büyüdükçe sezgisel algoritmalar ile çözüme ulaşılmıştır. Küçük boyutlu problemlerde GA ve İnteraktif Yapay Arı Kolonisi Algoritmalarının kesin çözüme yakın ve hızlı sonuçlar vermiştir. Problem büyüdükçe metasezgisel algoritmalarından CSA-Elit'in diğer algoritmalarından daha başarılı olduğu vurgulanmıştır. Çalışmada diğer bir sonuç hem işlem süresini hem de güç tüketimini en aza indirme göz önünde bulundurulduğunda deneysel sonuçlar, tüm makineler kullanımdayken optimum çözümün elde edilmediğini ortaya koymaktadır.

İKİNCİ BÖLÜM

OPTİMİZASYON VE METASEZGİSELLER

Dünya gittikçe daha karmaşık ve rekabetçi hale gelmektedir. Bu rekabetçi dünyada karar alma durumunda kalan mühendisler, yöneticiler ve bilim insanları mantıklı ve optimal kararlar almalıdırlar. Karar vermede klasik süreç formüle etme, modelleme, çözme ve uygulama adımlarından oluşmaktadır. Uygulamada, bu süreç, kabul edilebilir bir çözüm bulunana kadar optimizasyon modelini veya algoritmasını geliştirmek için yinelenebilir (Talbi, 2009: 2-3). Optimizasyon, çoğunlukla bir algoritmayla anılan, bir dizi kurallar bütünüdür bir tasarım, sistem ve karar verme olgusu sürecinde mantıksal olarak kullanılmasıdır (Akpınar, 2016:28). Başka bir tanımlamayla, ele alınan problemin yapısına uygun araçlar kullanarak, arzu edilen yönde en iyiyi elde etmektir. Optimizasyon genellikle problem içeriğinde saptanan kantitatif bir veya birkaç değerin veya bu değerlere bağlı kalitatif bir veya birkaç değerin maksimize edilmesi ya da minimize edilmesi şeklinde kaşımıza çıkabilir. Optimizasyon problemleri minimizasyon veya maksimizasyon değerlendirmesine tabi tutulan amaç fonksiyonu ve kısıtların rastgelelik içerip içermemesi durumuna göre stokastik veya deterministik optimizasyon problemleri olarak tanımlanabilmektedir.

Optimizasyonda kullanılan araçlar son birkaç on yılda önemli ölçüde gelişim sağlamıştır. Matematiksel programlama ve algoritma tasarımının ilerlemesi, bilgisayar performanslarındaki artış ve multi disiplinler fikirleri bir araya getirecek iletişim ortamının yaratılması gibi gelişmeler optimizasyon araçlarının gelişmesinin ve çeşitlenmesinin önünü açmıştır (Hansen ve Mladenovic 2016: 2). Bu gelişmeleri izler şekilde, optimizasyon araçları mühendislik, işletme, bilgisayar bilimi ve sanayi gibi alanların temel araçları haline gelmiştir.

Optimizasyon problemi aşağıdaki gibi formüle edilebilir:

$\min\{f(d) d \in D, D \subseteq U\},$	(9)
veya	
$\max\{f(d) d \in D, D \subseteq U\},$	

Eşitlik 9'daki alternatif formülasyonlarda U çözüm uzayını, D uygun değişken kümesini, d bir uygun çözümü ve $f(d)$ ise amaç fonksiyonunu göstermektedir. U sınırlı fakat çok büyük bir küme ise bir kombinatoriyal optimizasyon problemi tanımlanır (Hansen ve diğ. 2010: 320).

Yöneylem araştırması ve diğer alanlarda ele alınan problemlerin pek çoğu makul bir sürede kesin çözümü bulanamayacak kadar büyük ve karmaşıktır. Garey ve Johnson'ın (1978) hesaplama karmaşıklığı teorisinde bahsettiği üzere, bu tür problemler NP-Zor problemler sınıfındadır. Bu nedenle, yüksek kaliteli yaklaşık bir çözümü veya optimumluk kanıtı olmaksızın optimum bir çözümü hızlı bir şekilde üretebilen sezgisel yöntemleri kullanmak kaçınılmazdır. Genellikle, yerel bir optimuma ulaşmak zor değildir, ancak global bir optimum elde etmek zordur. Diğer taraftan yerel optimumda takılmak ve global optimuma elde edememek sezgisel yöntemlerin zayıf yönlerindedir. Bu zayıflığın üstesinden gelmek için ise yine benzer sezgisel algoritmaların kullanımı gereklidir (Hansen vd. 2017: 321).

Metasezgisel algoritmalar kombinatoriyal ve global optimizasyon problemlerine sezgisel bir çerçeve oluşturan genel kullanımlı algoritmalarıdır (Hansen vd. 2008: 321).

Tez çalışmasının bu bölümünde problemi çözmek için kullanılan optimizasyon algoritmalarının temel versiyonlarına yer verilmiştir.

Bu başlık altında optimizasyon genel kavramları ve yöntemleri açıklanacaktır. Metasezgisel algoritmalar kısmında kıyaslama veri setinin çözümünde daha önce kullanılmış metasezgisel algoritmalar ve bu tez kapsamında kullanılan ve önerilen algoritmalar bahsedilecektir. Algoritmaların sözde kodları uygulama başlığı altında ayrıca verilecektir.

2.1. Kombinatoriyal Optimizasyon

Bir optimizasyon problemi, problemin karmaşıklığına göre problem kesin yöntemler veya yaklaşık yöntemler ile çözülür. Kesin çözüm yöntemleri makul sürede çözülebilecek görece küçük boyutlu problemlerde kesin çözümü garanti eder. Yaklaşık (veya sezgisel) yöntemler ise pratik kullanım için makul bir sürede yüksek kaliteli çözümler üretir, ancak global bir optimal çözüm bulma garantisi yoktur. (Talbi, 2009: 18).

Bir problemin çözümünde geçen süre çözüm için seçilecek algoritmanın seçiminde etkilidir. Eğer bir problem kesin çözüm yöntemleri ile makul bir sürede çözülüyor ise kesin çözüm yöntemleri kullanılmalıdır. Bu yüzden önce problemin karmaşıklığı ölçülmelidir. Problem literatürde klasik veya daha önceden çözülmüş bir probleme indirgenebiliyorsa, problemi çözen bilinen algoritmalara bakılmalıdır (Talbi, 2009: 29).

Metasezgisel ve kesin çözüm yöntemleri işbirlikçi ve bütünleştirici olmak üzere iki şekilde kombinasyonları vardır. İşbirlikçi kombinasyonda yöntemler ardışık olarak sıralı yürütebilir. Bir diğer işbirlikçi kombinasyon yöntemlerin paralel veya içi içe edilir veya kesin yöntemlere metasezgiseller dahil edilir. İşbirlikçi kombinasyonda yöntemler birleşmez sadece bilgi paylaşımı vardır. Diğer kombinasyonlarda ise yöntemlerden biri diğerinin içine gömülür (Puchinger ve Raidl 2005: 42).

Optimizasyon modelleri matematiksel modeller, kombinatoriyal optimizasyon, kısıtların karşılandığı modeller ve analitik olmayan modeller olarak 4 sınıfa ayrılabilir (Talbi, 2009: 4).

Bir optimizasyon problemi, (S, f) çifti ile tanımlanabilir, burada S , uygulanabilir çözüm kümesini temsil eder ve $f: S \rightarrow \mathbb{R}$, optimize edilecek amaç fonksiyonu temsil eder. Amaç fonksiyonu ile arama alanının her çözümünün değeri hesaplanır ve bu hesaplama ile çözümün uygunluğu belirlenir. (Talbi, 2009: 3).

Eğer $s^* \in S$ çözümünden amaç fonksiyonuna göre daha iyi bir çözüm bulunamıyor ise bu s^* çözümü çözüm alanındaki global optimum çözümdür (Talbi, 2009: 3).

Pratik ve teorik öneme sahip pek çok optimizasyon problemi, bazı hedeflere ulaşmak için bir dizi değişkenin en iyi konfigürasyonunu araştırmaktan oluşur. Optimizasyon problemleri çözümlerin gerçek değerli değişkenlerle kodlandığı ve çözümlerin kesikli değişkenlerle kodlanması olarak 2 farklı sınıfa ayrılabilirler. Ayrık değişkenlerin bulunduğu optimizasyon problemlerine Kombinatoriyal Optimizasyon (KO) problemleri adı verilir. Bir KO problemi S çözüm uzayı olarak adlandırılan uzayda amaç fonksiyonunu en iyileyecek şekilde s^* çözümü bulunarak çözülür (Blum ve Roli, 2003: 269).

KO, bir problemin çözüm kümesi olan X kümesinde, her bir çözüm değerinin hesaplandığı bir f fonksiyonunda amaca göre en uygun çözümün bulunmasını amaçlayan

ayrık matematiksel alandır. Örnek olarak, bir makinedeki bir iş planlama problemi için, X , öncelik ve öncelik kısıtlamalarını karşılayan tüm iş dizilerinden oluşabilirken, f , son işin tamamlandığı tarihe (tamamlanma süresi) karşılık gelebilir (Widmer vd., 2008: 36).

Sonlu veya sayılabilir sonsuz sayıda alternatif çözüme sahip çoğu pratik problem, KO problemleri olarak formüle edilebilir. KO, optimal bir düzenleme, gruplandırma, sıralama veya genellikle sayı olarak sonlu olan ayrı nesnelerin seçilmesine ilişkin matematiksel çalışma olarak tanımlanır. Matematiksel programlamada bu KO problemlerine kesin çözümler bulma şansını büyük ölçüde artıran önemli gelişmeler olsa da makul hesaplama sürelerinde büyük problem örneklerini çözmek zordur. Bu zorluk, birçok yaklaşık algoritmanın geliştirilmesine yol açmış ve sezgisel algoritmaları hızla büyüyen bir araştırma alanı haline getirmiştir (Osman ve Laporte, 1996: 514).

Diğer problem türleri gibi KO problemlerinde de işlem karmaşıklığı kavramları vardır. İşlem karmaşıklığı, problemin çözümünde kullanılan algoritmanın tamamlanma süresi, problemin büyüklüğü ve problemin daha alt problemler ile ifade edilebilmesi ile belirlenir. Bir problem P ile ifade ediliyor ise bu problemi polinomal zamanda çözecek bir algoritma vardır. Problem NP (Non-deterministic Polynomial Time) ile ifade ediliyorsa çözümlerden biri polinom zamanında doğrulanabilir. Bir problem NP sınıfındaysa ve NP sınıfındaki tüm problemler bu probleme polinom zamanda indirgenebiliyorsa bu problem $NP-Tam$ 'dır. $NP-Zor$ sınıfına giren bir problem çözülebilir ise diğer bütün $NP-Tam$ problemleri de çözebileceği anlamına gelir. Çoğu bilgisayar bilimcisine göre polinom zamanlı çözümü olmayan $NP-Tam$ ve $NP-Zor$ problemlerde kabul edilebilir çözümler üretmek için metasezgisel algoritmalar kullanılmalıdır. Bu problemler için kanıtlanabilir şekilde en iyi olan bir çözüm bulma fikrinden kesinlikle vazgeçilmelidir (Eiben ve Smith, 2015: 9-12).

Zor kombinatoriyal optimizasyon problemleri araç rotalama, çizelgeleme, ağ tasarımı vb. endüstriyel ve bilimsel öneme sahip birçok alanda ortaya çıkar. Bu problemlerin çözümü için önerilen çözüm yöntemleri kesin ve sezgisel çözümler olarak iki sınıfa ayrılabilir. Kesin çözüm yöntemlerinde en uygun çözümün bulunması garanti edilmekle birlikte, problem boyutu arttıkça çözüm süreleri uzayabilmektedir. Hesaplanamayacak kadar uzun sürelerde olan büyük problemlerin çözümü için, optimal çözümden vaz geçilip, makul bir zamanda yaklaşık çözümün bulunması yeterli görülür. Kesin yöntemler arasında dal ve sınır, dinamik programlama, Lagrange gevşeme tabanlı yöntemler ve dal ve kes, dal ve fiyat ve dal ve kes gibi doğrusal ve tam sayı programlama

tabanlı yöntemler bulunur. Sezgisel yöntemlere; Genetik Algoritma (GA), Benzetimli Tavlama (BT), Tabu Arama (TA), Yinelemeli Yerel Arama (YYA), Değişken Komşuluk Arama (DKA), Evrimsel Algoritmalar, Dağılım Arama ve Memetik Algoritma örnek verilebilir (Puchinger ve Raidl, 2005: 41-42).

Bu yöntemler, çok sayıda kombinatoriyal optimizasyon problemine uygulanabilecek kadar genel olduğunda, bunlara metasezgisel yöntem denilir. (Widmer vd., 2008: 36).

2.2. Metasezgisel Algoritmalar

Bu başlık altında tezin kapsamında çalışılan probleme ait kıyas veri setinin çözümünde daha önce diğer araştırmacılar tarafından kullanılan ve bu tez kapsamında kullanılan algoritmalarından bahsedilecektir. Kullanılan ve yeni önerilen algoritmaların sözde kodlarına Bölüm 3'te ayrıca değinilecektir.

Karmaşık yapıdaki problemlerin çözümünde metasezgiseller karşımıza çıkmaktadır. Problemlerin etkin çözülebilmesi için bir metasezgisel algoritma tasarlanırken iki aşama dikkate alınmalıdır. Bu iki aşamadan ilki arama alanının keşfi çeşitlendirme aşamasıdır. Çeşitlendirmede daha iyi çözümlerin bulunduğu alanlar keşfedilmeye çalışılır. İkinci aşama ise ve bulunan en iyi çözümlerin kullanılması yoğunlaştırma aşamasıdır. Yoğunlaştırma aşamasında bulunan çözümün daha da iyisi araştırılır (Talbi, 2009: 24).

Metasezgisel algoritmalar aday bir çözümü yineleyerek verilen kriterlere göre, aday çözümü geliştirmeyi amaçlar. Kesin çözümü garanti etmeseler de metasezgisel algoritmalar diğer optimizasyon tekniklerinin etkili bir şekilde çözemediği optimizasyon problemlerini çözebilirler. Metasezgisel algoritmalar planlama, çizelgeleme ve rotalama gibi üretim problemlerinde yaygın bir şekilde kullanılmaktadır. Metasezgisel algoritmalar bir arama sürecinin problemlere uygulanmasıdır. Metasezgisel algoritmalarda arama işlemi çeşitlendirme ve yoğunlaştırma olmak üzere iki fazda yapılır. Çeşitlendirme fazında bütün çözüm uzayı araştırılırken, yoğunlaştırma fazında daha küçük bir alanda optimal çözüm aranır. Problem çözümlerinde optimal çözümü garanti etmeseler de metasezgisel algoritmaların üstün yönleri şu şekilde sıralanabilir (Widmer vd., 2008: 37).

Optimal çözüm; problemin büyük olması, verilerin kesin olmaması, problemin matematiksel olarak ifade edilememesi, birden fazla amacın çakışması gibi nedenlerden dolayı bulunamayabilir. Kesin çözümler metasezgisel yöntemlerden daha yavaştır.

Sezgisel yöntemler daha az deneyimli kullanıcılar tarafından da uygulanabilir. Bir metasezgisel algoritma problemlere ve bir başka algoritmaya kolay uyarlanabilir. Bu sayede farklı problemlerin çözümünde etkilidir.

Birçok metasezgisel yöntem olmakla birlikte bu yöntemler iki ana kategoride değerlendirilir. Eğer her yinelemede sadece tek sonuç oluşturuluyor ise yineleme tabanlı, her yinelemede birçok çözüm oluşturuluyor ise popülasyon tabanlı metasezgisel yöntemler olarak adlandırılır (Onar vd., 2016: 2). Bununla birlikte metasezgisel algoritmalar Doğadan esinlenen – Doğadan esinlenilmeyen, Hafıza kullanan – Hafıza kullanmayan, Yinelemeli- Açgözlü (yapıcı) olarak da sınıflandırılabilir (Talbi, 2009: 25).

Metasezgisel algoritmaların çoğu yinelemeli olmakla birlikte Açgözlü (Yapıcı) metasezgisellerden bahsetmek gerekir. Açgözlü (Yapıcı) metasezgisellerin her adımında, mevcut çözüm, nihai çözüm maliyeti ile ilgili olarak yaratacağı tüm sonuçlar dikkate alınmadan mümkün olan en iyi şekilde tamamlanır. Örneğin çizelgeleme için başlangıçta boş olan iş sıralamalarında işler sırayla eklenirken eklenen iş için en uygun yer aranır. Daha önce eklenmiş işlerin sıraları değiştirilmez (Widmer vd., 2008: 38).

2.2.1. Tek Çözüm Tabanlı Metasezgiseller

Tek çözümlü metasezgisel algoritmalar (S- Metasezgisel) sadece tek çözümden oluşturulan aday çözümleri araştırır ve uygun olanı seçer. Popüler olan tek çözümlü metasezgiseller Yerel Arama, Değişken Komşuluk Arama, Benzetimli Tavlama ve Tabu Arama'dır (Talbi, 2009: 88).

S-Metasezgisellerde ortak kavramlar komşuluk, yerel optimum ve ilk çözüm olarak tanımlanabilir. Bütün S-metasezgisel algoritmalarda komşuluk tanımlanması gerekli ve ortak bir adımdır. Tanımlanan komşuluk algoritmanın performansında önemli rol oynar. Komşuluk yapısı yeterli olmayan algoritma, problemin çözümünde yetersiz kalacaktır (Talbi, 2009: 88). Örneğin, çizelgeleme problemi için konuma bağlı komşuluk (bir işin başka bir konuma taşınması) veya sıraya bağlı komşuluk (2 işin karşılıklı yer değiştirmesi, işlerin sıralamada tersine çevrilmesi) gibi komşuluk yapıları seçilebilir (Talbi, 2009: 93). Bir S-metasezgisel tasarlanırken komşuluk yapısının büyük seçilmesi çözüm kalitesini arttırmakla birlikte hesaplama süresini de arttırmaktadır (Talbi, 2009: 94).

Bir komşuluk yapısı içinde bütün komşu çözümlerden en iyisi yerel optimum çözümdür (Talbi, 2009: 91). Sezgisel algoritma ile en iyi çözümü araştırmak için

başlangıç çözümü oluşturmak gereklidir İlk çözüm üretmek için rasgele ve açgözlü olmak üzere iki ana yaklaşım vardır. Hesaplama süresi ve çözüm kalitesi arasında ters ilişki vardır. Rasgele başlangıç çözümü oluşturmak hızlı iken açgözlü yaklaşım ile çözüm oluşturmak uzun zaman alabilir. Geniş komşuluk yapısında rasgele çözüm ile başlangıç çözümü üretildi ise arama yapmak çok uzun zaman alabilir. Açgözlü yaklaşımlar daha kaliteli yerel optimumlara neden olur. Bu S-metasezgisellerinin daha az iterasyon ile arama yapacağı anlamına gelir. Ancak iyi bir ilk çözüm ile başlamak iyi bir yerel minimum yakalanacağını garanti etmez (Talbi, 2009: 94).

2.2.1.1. Yerel Arama

Yerel arama yöntemleri yapıcı bir yöntem ile veya keyfi seçilen bir başlangıç çözümü ile her yinelemeli adımda X çözüm uzayının araştırılmasıdır. Yerel arama yöntemlerinde komşuluk seçimlerinde en bilineni İniş Yöntemi'dir. İniş Yönteminde örneğin bir çözüm uzayı araştırılırken amaç minimizasyon ise komşu çözümlerden çözüm değeri artana kadar çözüm aramaya devam edilir. Bu yerel arama tekniği birçok optimizasyon problemlerinde kullanılmasına rağmen aranan komşulukların büyük oluşu ve yerel minimuma takıldığından aramaya devam edilemediğinden daha etkili yerel arama teknikleri geliştirilmiştir (Widmer vd., 2008: 38-40). İlk çözümün ve komşuluğun tanımına ek olarak, bir yerel arama algoritmasının tasarlanması, bir sonraki mevcut çözümü belirleyecek olan komşunun seçim stratejisini ele almalıdır. Komşuluk seçiminde en iyi komşuluk (iniş yöntemi), bulunan ilk daha iyi çözüm ve rasgele çözüm olmak üzere üç farklı komşuluk yapısı kullanılır (Talbi, 2009: 123).

Yerel arama hızlı ve kolay uyarlanabilir bir yöntemdir. Bu avantajlarından dolayı yaygınca kullanılmaktadır. Yerel arama yöntemlerinin en büyük dezavantajı yerel optimuma takıldığı için başlangıç çözüme duyarlı olmasıdır. Yerel optimumdan kaçınmak için dört farklı yaklaşım mevcuttur (Talbi, 2009: 125-126):

Farklı Başlangıç Çözümlerinden Yenileme: Bu strateji, çok başlangıçlı yerel arama, yinelenen yerel arama, açgözlü rastgele uyarlamalı arama (ARUA) (*Greedy Randomized Adaptive Search Procedure*) (*GRASP*) ve benzerlerinde uygulanır.

Amaç Fonksiyonun Gelişmediği Komşulukların Kabul Edilmesi: Bu yaklaşım ile yerel arama alanın dışına çıkabilir. Bu yöntemde en popüler yöntemler Benzetimli Tavlama ve Tabu Arama olarak belirtilebilir

Komşuluğun Değiştirilmesi: Bu yapı Değişken Komşuluk Arama yöntemlerinde kullanılır.

Amaç Fonksiyonun veya Giriş Verilerinin Değiştirilmesi: Bu yaklaşım, rehberli yerel aramada, yumuşatma stratejilerinde ve gürültü yöntemlerinde uygulanmıştır.

2.2.1.2. Benzetimli Tavlama

Sonlu bir sıcaklıkta sıcaklığın değişimi ile sistemin davranışının, çok değişkenli veya kombinatorial optimizasyon problemlerinin çözümü arasında faydalı bir bağlantı vardır. Katılarda tavlama işlemi ile büyük optimizasyon problemlerinin çözümünde bir çerçeve sağlanır. Bu çerçevede yeni bilgileri ortaya çıkarılır ve geleneksel optimizasyon problemleri ve yöntemleri hakkında alışılmadık bir bakış açısı sağlar (Kirkpatrick vd. 1983 :671).

Benzetimli Tavlama (BT), fizikteki katı maddenin ısıtılarak sıvı halden geçip tekrar soğutulma aşamasında maddenin davranışından ilham alınarak modellenmiştir. BT'de önce çeşitlendirme fazında mevcut çözümlerden rasgele çözümler üretilir. Yerel minimuma takılmamak için başlangıçta daha kötü sonuçlar kabul edilebilir. Yoğunlaşma aşamasında ise kötü sonuçları kabul etme olasılığı düşürülür. Yoğunlaşma kısmında optimal sonuçlara ulaşılması beklenir. Başlangıç sıcaklığı, soğutma parametresi, komşuluk büyüklüğü, iterasyon ve döngü sayıları modelin parametreleri olarak söylenebilir. Bu parametreler etkili bir çözüm için iyi ayarlanmalıdır. Başlangıç sıcaklığı kabul edilme olasılıklarını ayarladığı için yerel ve genel arama ihtimallerini ayarlar. Soğuma parametresi iterasyon sayılarını etkiler. Komşuluk büyüklüğü ise optimal sonucu ve yöntemin hızını tanımlar (Onar vd., 2016: 4).

BT, arama mekanizmasının Metropolis vd. algoritmasına ve termodinamik tavlama ilkelerine göre modellendiği yerel bir arama yöntemidir. Fikir, Metropolis algoritmasını azalan sıcaklık değerleri t ile kullanmaktan ibarettir. Bir partikül sisteminin aşamalı soğutulması, bir yandan sistem enerjisi ile bir kombinatorial optimizasyon probleminin nesnel işlevi arasında ve diğer yandan sistem durumları ile dikkate alınan problemin çözümleri arasında bir analogi yapılarak simüle edilir. Mümkün olduğunca az enerjiye sahip durumlara ulaşmak için, sistem başlangıçta çok yüksek bir sıcaklığa getirilir ve ardından yavaşça soğutulur. Sıcaklık düştüğünde, atom hareketleri daha az rastgele hale gelir ve sistem düşük enerji durumlarında olma eğilimindedir. Her t sıcaklığında denge durumuna ulaşmak için sistem soğutulması çok yavaş yapılmalıdır.

Belirli bir t sıcaklığında yeni bir durum kabul edilmediğinde, sistem donmuş kabul edilir ve minimum enerji seviyesine ulaştığı varsayılır (Widmer vd., 2008: 42).

Algoritma, rastgele bir çözüm üretmekle başlatılır. Bundan sonra, komşu çözüm üretilir ve bu iki çözüm bir amaç fonksiyonu ile değerlendirilir. Komşunun maliyeti ilk çözümün maliyetinden daha düşükse ve sistemin enerjisini düşürüyorsa, komşu iyileştirilmiş bir çözüm olarak kabul edilecektir. İyileşmeyen bir çözüme gelince, bir olasılık fonksiyonu tarafından verilen bir olasılık değeri ile kademeli olarak kabul edilecektir. BT uygulaması kolay ve yakınsama özelliği güçlü bir yöntemdir. BT yerel optimalden kaçmak için tepe tırmanma hareketlerini kullanması sayesinde son yirmi yılda popüler bir teknik haline gelmiştir (Zarandi vd., 2020: 539).

BT hafıza özelliği olmayan bir metasezgiseldir. Arama sırasında elde edilen bilgiler kaydedilmez. BT'ye özgü olan özellikler; daha kötü komşuların da kabul edilebilmesi ve algoritmanın her adımında sıcaklığının belirlendiği soğuma takvimidir (Talbi, 2009: 129).

Yerel arama tekniklerinde BT'nin daha düşük çözümleri kabul etme düzeylerinin deterministik bir şekilde belirlendiği eşik kabul etme yöntemleri de vardır (Widmer vd., 2008: 44). BT yöntemine benzer algoritmalara Eşik Kabul Etme, Büyük Tufan Algoritması, Kayıttan Kayda Seyahat ve Şeytan(demon) Algoritmaları örnek verilebilir. Bu algoritmalarda amaç sonuçların kalitesinden ödün vermeden algoritma hızını arttırmaktır (Talbi, 2009: 133).

2.2.1.3. Tabu Arama

Tabu arama (TA), özellikle çizelgeleme olmak üzere önemli sayıda kombinatoriyal optimizasyon probleminin çözümünde kullanılan çok güçlü bir yöntemdir. TA yerel minimuma gelene kadar herhangi bir yerel arama fonksiyonu kullanılabilir. Yerel minimuma geldiğinde yerel minimumdan kurtulmak için daha kötü bir çözümün olduğu komşu bir çözüme gidilir ve aramaya devam edilir. Yerel minimum çevresinde döngüyü engellemek için yapılan hareketler bir listede tutulur. Başka bir deyişle tabu araması her adımda bir daha gidilmemesi gereken T tabu listesi tutar (Widmer vd., 2008: 46-47).

TA arama sürecini daha etkili hale getirmek için yinelemeler sırasında toplanan bilgileri kullanır. BT yönteminde olduğu gibi, yerel optimalden çıkmak için kötü çözümleri de kabul eder. Bununla birlikte, TA algoritması, rastgele arama kullanan BT'den farklı olarak tüm komşu çözümleri belirleyici olarak arar. TA'nın ayırt edici

özelliđi, daha önce ziyaret edilen çözümlerin tekrar kabul edilmesini engelleyen kısa süreli bir belleđin kullanılmasıdır. Bu özellik optimum çözüme ulaşmayı hızlandırır (Zarandi vd., 2020: 541).

Diđer S-metasezgisellerinden ayırt edici özelliđi, yerel optimalde döngü oluşturmamak için kısa dönem hafızalı gidilmemesi gereken çözümlerin tabu listesinde tutulması ve yine belli bir olasılık ile bu listedeki çözümlerin kullanılabilir olmasıdır. TA'da yoğunlaşmada orta vadeli bellek ile arama sırasında bulunan seçkin çözümlerin depolandığı ve çeşitlendirme fazında uzun süreli bellek ile arama sırasında ziyaret edilen çözümlerle ilgili bilgileri depolandığı gelişmiş çözümler de vardır. Bu sayede yoğunlaştırma aşamasında iyi sonuçlar tutulurken çeşitlendirme aşamasında ise gidilmemiş çözümlerde keşfe çıkılabilir (Talbi, 2009: 141).

Tabu kelime olarak riskli veya yasaklı anlamı içerir. Arama prosedüründe ise çözüm araştırılırken komşuluk içinde yapılmasına izin verilmeyen hareketi ifade eder. Tabu hareketleri yerel minimumdan kaçınmak için tanımlanır. Daha önce tanımlanmış sonuçlar tabu listesine kaydedilir. Tabu listesi büyüdükçe farklı stratejiler geliştirilmiştir. TA algoritmasında parametreler komşuluk yapıları, hareketlerin tabu listesine eklenme ve çıkarılma koşulları, tabu listesinin büyüklüğü ve bitirilme kriteri olarak sıralanabilir (Onar vd., 2016: 3-4).

Çeşitli çizelgeleme problemleri için kullanılan TA, çizelgeleme ve yönlendirme problemleri için oldukça iyi performans gösterdiği bilinmektedir. TA ilk uygun bir çözümle başlar ve bunu geliştirmeye çalışır. Yeni bir çözüm ararken döngüden kaçınmak için en son hamleler Tabu Listesi'ne yazılır. (Cesaret vd., 2012: 1199).

2.2.1.4. Deđişken Komşuluk Arama

Bu tez çalışması kapsamında çizelgeleme probleminin çözümü için Deđişken Komşuluk Arama (DKA) algoritması kullanılmıştır. DKA komşuluk yapılarının sistematik bir şekilde deđiştirilmesi fikrine dayanır. Deđişken komşuluk arama ilk önce komşuluk yapıları belirlenir. Sonra ana adıma geçilir. Ana adımda uygulanan komşuluk yapısına göre yeni bir çözüm bulunur. Bulunan çözüme yerel arama uygulanır. Yerel arama ile bulunan çözümde ilk çözümden iyi ise bu çözüm geçerli çözüm olarak kabul edilir ve komşuluk yapıları ilk komşuluktan başlayacak şekilde en başa alınır. Çözüm iyi deđil ise yeni çözüm bir sonraki komşuluk yapısı ile belirlenir. Bu ana adım bilgisayarda

kısıtlı bir süre içinde veya belli bir iterasyon sayısına bağlı olabilir (Mladenović, N., ve Hansen, 1997: 1098).

Değişken komşuluk aramadaki (DKA) ana amaç önceden tanımlanan komşuluk yapıları ile yerel optimalden kaçınmaktır. Farklı komşuluklar farklı yerel optimaller oluşturur. Global optimumun bu yerel optimallerden biri olması beklenir. Belirlenen komşuluk yapıları büyük olduğunda arama süresi artar. Komşuluk yapıları genelde karmaşıklığına göre azdan çoğa doğru sıralanır (Talbi, 2009: 150-151).

DKA'nın farklı varyantları vardır. Bu varyantlardan Değişken Komşuluk İniş (DKİ) DKA algoritmasının deterministik versiyonudur. DKİ, DKA algoritmasının temel bileşenidir ve DKA algoritmasının aksine yerel arama yoktur. Komşuluk yapıları rasgele seçilir ise İndirgenmiş Değişken Komşuluk Arama Algoritması adını alır (Reduced VNS). Temel DKA algoritmasında komşuluk yapıları sıra ile uygulanır ve her komşuluk yapısı uygulandıktan sonra yerel arama uygulanır. Genel DKA algoritmasında ise DKİ algoritması yerel arama olarak kullanılır. Çarpık DKA algoritmasında daha geniş komşuluk alanları araştırabilmek adına yeni bulunan çözüm mevcut çözümden daha kötü olsa bile belli bir olasılıkla kabul edilir. Değişken Komşuluk Ayırışma Arama algoritmasının farklı yerel arama yapılırken bütün çözüm uzayını araştırmayıp, problemlerin alt problemler olarak ele alıp alt problemlerde yerel arama uygulanmasıdır (Hansen ve Mladenovic, 2003:1-20).

Değişken komşuluk arama algoritmasının performansı başlangıç çözümünün kalitesine, seçilen komşuluklara, kullanılan yerel arama tekniklerine, komşulukların uygulanma sırasına göre değişir (Driessel ve Mönch, 2011: 338).

Çalışmada kullanılan DKA varyantı Temel DKA algoritmasıdır. Geliştirilen algorithmada hedef yeni komşuluk yapıları ve yeni yerel arama yöntemleri önermektir. Temel DKA sözde kodu ve bu tez kapsamında önerilen yeni DKA sözde kodu bölüm 3'te kıyaslamalı olarak verilecektir.

2.2.2. Popülasyon Tabanlı Metasezgiseller

Popülasyon tabanlı metasezgiselerde çözüm popülasyonları yinelemeler ile geliştirilerek yeni popülasyonlar oluşturulur ve çözümün araştırılır. Popüler P-metasezgisellerine evrimsel algoritmalar, karınca kolonisi optimizasyonu, dağılım araması, parçacık sürüsü optimizasyonu, arı kolonisi ve yapay bağışıklık sistemi örnek verilebilir. P- Metasezgiselerde; farklılıklar, çözüm araştırırken bilginin tutulduğu hafıza

yapısı, yeni nesil oluşturmakta ve yeni çözümlerin seçiminde ortaya çıkar. Çoğu p-metasezgiselinde hafıza yapısı; popülasyonları tutarken, karınca kolonisi optimizasyonunda feromonları, dağıtım algoritmalarında ise olasılık öğrenme modelini tutar. Yeni nesil oluşturmada ise evrimsel tabanlı yaklaşım ve yapay zekâ destekli kara kutu yaklaşımı vardır. Yeni çözümün seçiminde yeni oluşturulan çözümü seçmek, mevcut çözüm ile elit çözümleri seçmek veya kara kutu yaklaşımında yeni çözümlerin seçiminde kesinlik olmadan seçimler vardır. S-metasezgisellerin aksine p – metasezgisellerinde çözüm uzayı daha geniştir. Çözüm uzay eğer bir yerel arama ile veya açgözlü yaklaşım ile yapıldıysa erken yakınsama ortaya çıkabilir. Başlangıç popülasyonu rasgele, sıralı, paralel ve sezgisel olarak oluşturulabilir (Talbi, 2009: 190-198).

2.2.2.1. Genetik Algoritmalar

Genetik Algoritmalar (GA) biyolojik evrimin hesaplamalı bir modelidir ve problem çözmeye kullanışlıdır. Doğal seleksiyonun biyolojik süreci ve en uygun olanın hayatta kalması ilkeleri optimizasyon problemlerinin çözümünün araştırılmasında uygulanmaktadır. Genetik algoritmalar, genel olarak popülasyon genetiğinden gelen fikirlere dayanmaktadır. İlk olarak, rastgele bir birey popülasyonu oluşturulur. Bu popülasyondaki bazı bireyler diğer bireylerden daha başarılıdır. Çözümler araştırılırken bu başarılı bireylerin varyasyonları oluşturulur (Forrest, 1996: 77). Popülasyon tabanlı bu yaklaşım Genetik Algoritmalar (GA) olarak adlandırılır. GA evrimsel bir algoritmadır. İlk önce çözüm çiftleri oluşturulur. Bu çiftlerden yeni çözümler oluşturulur. İlk önce çözüm havuzu (kromozomlar) oluşturulur. Bu süreç metasezgisel algoritmanın çeşitlendirme fazıdır. Çözümler ikili, sayısal ya da alfa numerik diziler olarak kodlanabilirler. Yeni çözümler üretmek için ebeveyn kromozomlar seçilir. Farklı stratejiler ile ebeveynler çaprazlanarak yeni çözümler oluşturulur. Yeni çözümler popülasyonda yeni bireyler olur. Bütün bireyler belli bir amaç fonksiyonuna göre değerlendirilir ve kalitesiz çözümler çözüm havuzundan çıkarılır (Onar vd., 2016: 3). Üretim ve uygulamada uzmanlar geçmiş yıllara TA ve BT yöntemlerini kullanmışlardır. GA'da son zamanlarda uygulamalarda belli bir hızlanma vardır. Bu hızlanma ile GA'ya olan ilgi artmıştır. GA doğal evrim sürecine benzer şekilde en iyi çözümlerin hayatta kaldığı bir strateji güder (Aytuğ vd., 2003: 3955). Problem boyutu büyüdükçe genetik algoritmanın yüksek hesaplama süresi gibi sınırları vardır. Bu gibi durumları aşmak için başka bir sezgisel algoritma daha eklemek gereklidir (Azimpoor ve Taghipour, 2019: 19).

2.2.2.2. Karınca Kolonisi Optimizasyonu

Karınca Kolonisi Optimizasyonu (KKO) karıncaların davranışlarından esinlenmiştir. KKO'da bir dizi yapay karınca, bir optimizasyon sorununa çözümler oluşturur ve gerçek karıncalar tarafından benimsenmiş olanı anımsatan bir iletişim şeması aracılığıyla kaliteleri hakkında bilgi alışverişinde bulunur. Karınca hareketlerinden çeşitli algoritmalar önerilse de KKO algoritması aralarında en başarılısıdır (Dorigo vd. 2006: 30,37).

KKO karınca kolonisinin birlikte çalışma davranışından uyarlanmıştır. Karıncalar bir alana rasgele yayılarak yiyecek ararlar. Yiyeceği bulup yuvaya dönerken bıraktıkları feromon sayesinde karıncaların yiyeceğe doğru en kısa yolu belirlenmiş olur. Algoritmada kullanılan parametreler buharlaşma katsayısı, feromon fonksiyonu ve başlangıç feromon miktarıdır. Her yinelemede feromon seviyesi ve ihtimaller güncellenir (Onar vd., 2016: 5). KKO, KO problemlerine uygulanmıştır ve farklı problem türlerinin (çizelgeleme, yönlendirme, atama vb.) çözümünde yaygın başarı elde etmişlerdir (Talbi 2009:240).

2.2.2.3. Parçacık Sürü Optimizasyonu

Doğada sürü ve iş birliği halinde hareket eden birçok böcek türü vardır. Bu sürülerin birlikte hareket etmeleri ile etkin çözüm yöntemleri modellenmiştir. Kuşların sürü hareketlerinden esinlenen popüler metasezgisel yöntemlerden biri de parçacık sürü optimizasyonudur (PSO). Bir problemin aday çözümleri, bir popülasyondan gelen parçacıklar olarak temsil edilir. Her bir parçacığın konumu, parçacığın konumu ve hızı ve mevcut optimum parçacıklarla tanımlanır. Prosedürdeki arama yönü, popülasyondaki parçacıkların iletişimiyle tanımlanır. Bu ilkeler kullanılarak gelişme sağlanır ve sürü optimal çözüme yönlendirilir. Algoritmada kullanılan parçacık sayısı hesap süresini ve karmaşıklığını arttırdığı için önemlidir. Algoritmada kullanılan eylemsizlik katsayısı yerel ve genel arama büyüklüklerini belirler (Onar vd., 2016: 5-6). PSO, GA gibi popülasyon temelli bir algoritmadır. Durma kriterine kadar yinelemeli algoritma kullanılır. Sürü adını verdikleri parçacıklardan oluşur. Her bir parçacık çözüm uzayında bir çözümü temsil eder (Zarandi vd., 2020: 547). Bir parçacığın kendi konumu ve hızı vardır, bu da parçacığın uçuş yönü ve adımı anlamına gelir. Optimizasyonun başarısı parçacıkların iletişimi ile alakalıdır. Parçalar kendi ziyaret ettikleri en iyi konum ve tüm sürünün en iyi konumuna göre konumlarını ayarlarlar (Talbi, 2009: 248).

2.2.3. Hibrit Metasezgiseller

Saf metasezgisel algoritmalar karmaşık problemlerin çözümünde çok etkili değildir. Bu karmaşık problemleri etkin şekilde çözülmesi için metasezgisel algoritmalar, matematiksel programlama, kısıt programlama, makine öğrenimi, grafik teorisi, paralel ve dağıtılmış hesaplama gibi yöntemlerin birleşimi kullanılmaktadır. Günümüzde yaygın olan hibrit metasezgiseller bu karmaşık problemlerin çözümünde en etkili yöntemi temsil etmektedir. Hibrit metasezgiseller, P- metasezgiseli ile S-metasezgisellerin birleşiminden oluşabilir. Bu sayede çok güçlü bir arama algoritması ortaya çıkarır. P-metasezgiselleri çözüm uzayında yoğunlaşmada çok etkili değil ama çeşitlendirme fazında çok etkilidir. Bu metasezgiseller yoğunlaşmada etkili S-metasezgiselleri (yerel arama, TA) birleştirilmelidir. Bir başka hibrit algoritma yaklaşımı ise kesin çözüm yöntemleri ile metasezgisel yöntemlerin verimlilik ve etkinlik açısından değerlendirilerek birleşimini ifade eder. Kısıt programını çözen programlar sayesinde kısıt program ile metasezgisellerin birleşimi üzerine çalışmalar artmaktadır. Metasezgisel algoritmalar veri madenciliği problemlerinin çözümünde yaygın olarak kullanılmaktadır. Ancak amaç veri madenciliği teknikleri ile geçmiş verilerden faydalı bilgilerin çıkarılması ve algoritmanın etkinliği ve verimliliğin artırılması amaçlanır (Talbi, 2009: 452-453).

ÜÇÜNCÜ BÖLÜM

SIRA BAĞIMLI HAZIRLIK SÜRELİ İLİŞKİSİZ PARALEL MAKİNE ÇİZELGELEME PROBLEMİ İÇİN ÖNERİLEN DEĞİŞKEN KOMŞULUK ARAMA ALGORİTMASI

Sıra bağımlı hazırlık süreli ilişkisiz paralel makine çizelgeleme problemi (SBHSİPMÇP) birçok üretim alanında sıkça karşılaşılan bir problem türüdür. Bu problem akademik çalışmalarda da araştırmacıların dikkatini çekmiş ve birçok araştırmaya konu olmuştur. Problem boyutu arttıkça kesin çözüm yöntemleri problemi çözmeye yeterli gelmemiş ve araştırmacılar tarafından farklı metasezgisel algoritmalar ile çözümler geliştirilmiştir.

Metasezgisel algoritmalar ile çözümde, tek çözüm, popülasyon tabanlı çözüm ve hibrit metasezgisel algoritmalar olmak üzere farklı çözüm yöntemleri mevcuttur. Bu tez kapsamında problemi çözmek için uygulanan algoritma çizelgeleme problemlerinde başarı ile uygulanan tek çözümlü metasezgisel algoritma olan Değişken Komşuluk Arama (DKA) algoritmasıdır. Önerilen ve kullanılan tüm algoritmalarda, aynı iş yüküne sahip birden fazla makine bulunduğu ve bu makinelerden birinin seçilmesi gerektiği durumlarda veya makinelere iş atanırken aynı işleme süresine sahip birden fazla iş bulunduğu ve bu işlerden sadece birinin seçilmesi gerektiği durumlarında makine ve iş seçimleri rasgele yapılmıştır.

Bölüm 3.1’de, daha önce Bölüm 1.5’te açıklanan ve matematiksel modeli verilen SBHSİPMÇP problemin tanımı, notasyonu ve örnek bir problem tanıtılmıştır. Bölüm 3.2’de başlangıç çözümü oluşturmak için iki farklı başlangıç algoritması açıklanmış ve bu başlangıç algoritmalarının sözde kodları verilmiştir. Başlangıç algoritmalarına göre örnek problemlerin ilk çözümleri örnek problem üzerinde Gantt şemasında gösterilmiştir. Bölüm 3.3’te Önerilen DKA algoritmasının ve temel DKA algoritmalarının sözde kodları verilmiştir. Bölüm 3.3.1’de DKA algoritmasının yapı taşlarından komşuluk yapıları anlatılarak ve şekilsel gösterimi verilmiştir. Bu çalışmaya özgün olarak komşuluk yapılarında önerilen yeni parametrelerden bu bölümde bahsedilmiştir. Bölüm 3.3.2’de yine DKA algoritmasının önemli aşamalarından biri olan yerel aramalardan bahsedilmiş ve önerilen 4 farklı yerel arama sözde kodları ve yeni önerilen yerel arama seçim stratejisi tartışılmıştır. Bölüm 3.4’te Deneysel Sonuçlar başlığında daha önceki araştırmacıların

önerdikleri yöntemlerden bahsedilecek ve bu çalışmadaki önerilen algoritma ile sonuçları kıyaslanmıştır.

3.1. Hazırlık Süreli İlişkisiz Paralel Makine Çizelgeleme Problemi Örnek Veri Seti

Ele alınan problem; N adet işin M adet makinede iş ve makineye bağımlı olmak üzere hazırlık süreleri göz önüne alınarak en kısa sürede tamamlanacak şekilde çizelgelenmesidir. Makineler özdeş olmadığından işlerin makinelerde işlenme süreleri farklıdır. P_{ij} (i . makinede j . işin işlenme süresi) 6 adet işin 2 makinede gösterimi Tablo 3'te görülebilir.

Tablo 3. 6 İş ve 2 Makine İçin Süreler

İş	1. Makine	2. Makine
1	59	68
2	76	60
3	68	73
4	52	73
5	84	77
6	85	59

Tablo 3'ten anlaşılacağı üzere örneğin 4 numaralı iş 1. makinede işlenmesi (P_{14}) 52 birim süre iken 2. makinede (P_{24}) ise 73 birim süredir.

Daha önce de bahsedildiği üzere işlerin makinelere ve işlenme sıralarına göre farklı hazırlık süreleri vardır. S_{ijk} (i . makinede j işinden sonra k işinin işlenmesi için ihtiyaç duyulan hazırlık süresi) olmak üzere 1. ve 2. makinedeki hazırlık süreleri Tablo 4 ve Tablo 5'te verilmiştir.

Tablo 4. 6 İş ve 2 Makine Çizelgelemede 1. Makine için Hazırlık Süreleri

j .iş/ k .iş	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
$j=1$	57	91	98	70	95	56
$j=2$	68	66	52	93	86	95
$j=3$	84	89	62	98	59	60
$j=4$	65	89	89	57	89	52
$j=5$	59	86	90	56	73	52
$j=6$	73	94	68	99	97	67

Tablo 5. 6 İş ve 2 Makine Çizelgelemede 2. Makine için Hazırlık Süreleri

j .iş/ k .iş	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
$j=1$	95	75	98	53	62	97
$j=2$	76	75	55	85	88	51
$j=3$	88	94	76	72	50	77
$j=4$	79	92	86	63	93	81
$j=5$	76	82	95	98	50	68
$j=6$	98	97	99	99	96	72

Tablo 4 ve Tablo 5'te $j=k$ olduğu durumlar, o işin makinede ilk iş olarak işlenmesi durumunda gereken hazırlık süresini ifade etmektedir. Örneğin 2. makinede 4. işten sonra 5. işin işlenmesi için 93 birim süreye ihtiyaç vardır. Aynı zamanda eğer 5. iş 2. Makinede ilk işlenecek iş ise hazırlık için 50 birim süreye ihtiyaç vardır. Bu örnek veri seti için optimal çözüm Arnaout vd. (2010) tarafından hesaplanmış ve C_{max} değeri 390 olarak bulunmuştur.

3.2. Başlangıç Çözümü Oluşturma

Bir metasezgisel algoritmada başlangıç çözümü oluşturmak için rasgele veya açgözlü algoritmalar ile başlangıç çözümü oluşturmak gibi iki farklı yaklaşım mevcuttur. Bir sezgisel algoritmanın başarısı başlangıç çözümünün kalitesine bağlıdır. Açgözlü algoritmalar daha kaliteli çözümler oluştururken, çözümün daha dar alanda araştırılıp yerel optimuma takılma olasılığını artırır. Rasgele başlangıç algoritmalar ise sezgisel algoritmanın çözüm uzayında daha farklı alanları araştırmasına olanak sağlar. Bu durumda da nihai çözümün bulunması daha uzun zaman alacaktır. DKA başlangıç çözümüne ihtiyaç duyar. Öncelikle önerilen algoritmalarda başlangıç çözümü oluşturmak için bir rasgele ve bir açgözlü algoritma önerilmiştir.

3.2.1. Rasgele Başlangıç Çözümü Oluşturma

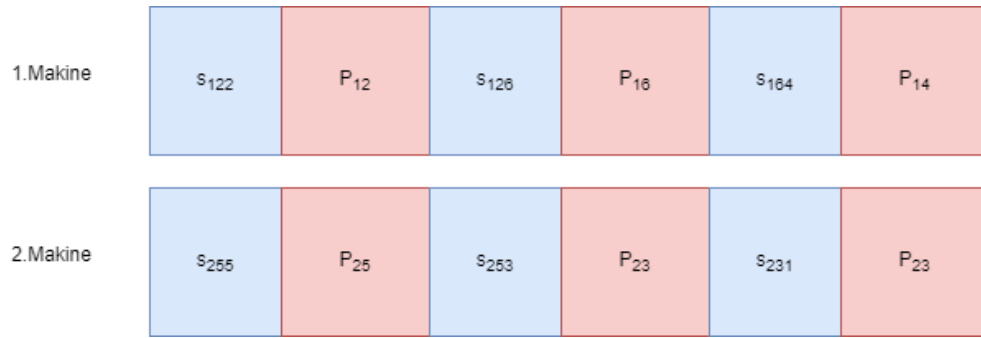
Bu başlangıç çözümünde rasgele işler rasgele makinelere atanırken makinelerin eşit sayıda işler ile yüklenmesi ve bu sayede makinelerde işlerin tamamlanma sürelerinin dengelenmesi amaçlanmıştır. Algoritmanın sözde kodu şu şekildedir.

```

Girdi: N: Toplam işler, M: Bütün Makineler UM: Uygun
Makineler
N, M, UM=M tanımla, m=j=L=∅ ata
N boş olana kadar tekrarla:
  Eğer UM Boş ise:
    UM =M
    j=Rasgele (N)
    m=Rasgele(UM)
    L'ye (m, j) makine iş çiftini ekle ()
    N. Çıkar(j), UM. Çıkar(m)
Bitir
Çıktı: L: Atanan işler

```

Yukarıda sözde kodu verilen rasgele algoritma ile oluşturulan bir çizelgeleme sonucunda 1.makineye (2,6,4) numaralı işler 2.makineye (5,3,1) numaralı işler belirtildiği sırada atanmış olsun. Oluşturulan bu başlangıç çözümünün Gantt şeması gösterimi Şekil 1'de verilmiştir.



Şekil 1. 6 iş 2 Makine Rasgele Başlangıç Gantt Şeması

Şekil 1'deki makineye atanan işlere ve hazırlık sürelerine göre Tablo 3, Tablo 4 ve Tablo 5'ten yararlanılarak makine yükleri (C) şu şekilde hesaplanır.

$$C_1 = S_{122} + P_{12} + S_{126} + P_{16} + S_{164} + P_{14} = 66 + 76 + 95 + 85 + 99 + 52 = 473$$

$$C_2 = S_{255} + P_{25} + S_{253} + P_{23} + S_{231} + P_{23} = 50 + 77 + 95 + 73 + 88 + 68 = 451$$

Burada toplam tamamlanma süresi $C_{max} = \max(C_1, C_2) = 473$ olarak bulunmuştur.

Rasgelelik metasezgisel algoritmalarda son derece önemlidir. Rasgele çözüm ile çok iyi bir çözüm yakalanabileceği gibi istenen çözümden çok uzak bir çözüm de elde edilebilir. Rasgele başlangıç algoritması ile verilen test örneği için 100 defa başlangıç çözümü oluşturulmuş ve sonuçları Tablo 6'da verilmiştir. 100 deneme arasında 1 defa optimal çözüme ulaşılmıştır.

Tablo 6. Rasgele Başlangıç Algoritması C_{max} Değerleri

En düşük C_{max}	Ortalama C_{max}	En yüksek C_{max}
390	446,72	503

Tablo 6 incelenirse küçük boyutlu problemlerde yeteri kadar deneme yapıldığı takdirde rasgele çözümler ile de optimal çözümün (390) elde edilebileceği görülebilir. Çok sayıda deneme ile optimal çözüm elde edilse de ortalama değerleri optimal çözümden çok uzaktır. Problemin optimal çözümü 1. makede sırası ile 4,1 ve 6 numaralı işlerin 2. makede ise 2,3 ve 5 numaralı işlerin çizelgelenmesidir.

3.2.2. Açgözlü Başlangıç Çözümü Oluşturma

Hazırlık sürelerinin başlangıç çözümünü oluşturmada kullanan algoritma (Helal vd., 2006:184)'de önerilen başlangıç çözümünden esinlenmiştir. Helal vd. (2006) önerdikleri algortirmada hazırlık sürelerinin ortalamalarını ve işlem sürelerini değerlendirirerek işler makineye atanırken en az süreli işi ve bir sonraki en az süre ile arasındaki orana bakarak işleri atar. Hesaplanan oran belirtilen orandan büyük ise işleri

atanmayan işler listesine ekler. İkinci aşamada belli oranın üstündeki atanmayan işleri, makineler arasındaki en az tamamlanma süresine sahip makineye atar.

Önerilen açgözlü başlangıç algoritmasında da işlem sürelerinin yanı sıra işler makinelere eşit sayıda atanırken işlerin atandığı makinedeki hazırlık sürelerinin ortalamaları ve o makinedeki işlem süreleri değerlendirmeye alınır. Optimal sıralamada bir makinede bir işin hangi sırada hangi işten önce işlenmesi gerektiği kesin olarak bilinmemektedir. Zaten geliştirilmesi hedeflenen algoritmanın amacı da bu sıralamayı elde etmektir. Tablo 4 ve Tablo 5'ten faydalanılarak oluşturulan hazırlık süreleri ortalamaları Tablo 7'de verilmiştir.

Tablo 7. 6 İş 2 Makine İşlerin Makinelere Göre Ortalama Hazırlık Süreleri

Makine / İşler	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
Makine 1	77,83	76,60	75,30	73,50	69,30	83,00
Makine 2	80,00	71,60	76,16	82,30	78,16	93,50

Bu algoritma Helal vd. (2006)'nın aksine en az yüklü makineyi ve hazırlık süreleri ile işlem sürelerini birlikte değerlendirir. Sürekli en az yüklü makineye iş atayarak makinelerin eşit yüklenmesi hedeflenir. Ayrıca yeni eklenen işi makine içi değişik pozisyonlara ekleyerek makine bazlı tamamlanma süresini minimize etmeyi hedefler. Bu başlangıç çözümü algoritması sözde kodu şu şekildedir.

Girdi: N: Toplam işler, M: Bütün Makineler, UM: Uygun Makineler

N, M, $m=\emptyset$, $j=\emptyset$, L = \emptyset , UM= \emptyset tanımla.

Her bir makinede ortalama hazırlık sürelerini hesapla().

N boş olana kadar tekrarla:

m =En az yüklü makineyi bul().

j =işlerinden m makinesinde en kısa süre + ortalama hazırlık süresi en küçük j işini bul().

j işi için m makinesinde en uygun pozisyonu bul().

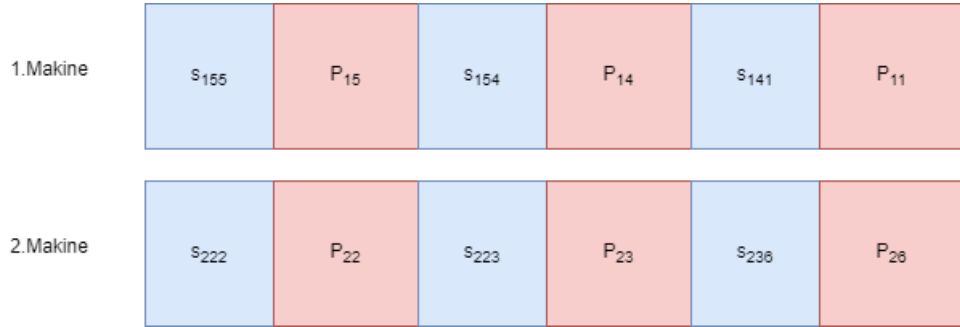
L 'ye (m, j) makine iş çiftini ekle()

N. Çıkar(j)

Bitir

Çıktı: L: Atanan işler

Yukarıdaki açgözlü başlangıç algoritmasına göre oluşan çizelgeleme sonucunda 1.makineye (5,4,1) numaralı işler ve 2. makineye (2,3,6) numaralı işler atanmıştır. Çizelgelemenin Gantt şeması gösterimi Şekil 2’de verilmiştir.



Şekil 2. 6 iş 2 Makine Açgözlü Başlangıç Gantt Şeması

Şekil 2’deki makineye atanan işlere ve hazırlık sürelerine göre Tablo1, Tablo2 ve Tablo3 kullanılarak makine yükleri (C) şu şekilde hesaplanır.

$$C_1 = S_{155} + P_{15} + S_{154} + P_{14} + S_{141} + P_{11} = 73 + 84 + 56 + 52 + 65 + 59 = 389$$

$$C_2 = S_{222} + P_{22} + S_{223} + P_{23} + S_{236} + P_{26} = 75 + 60 + 55 + 73 + 77 + 59 = 399$$

Burada toplam tamamlanma süresi $C_{max} = \max(C_1, C_2)$ 399 olarak bulunmuştur. Bölüm 3.1’ de bahsedildiği üzere bu problemin optimal çözüm değeri 390’dır. Makineleri işleyeceği uygun işlerle çizelgeleme oluşturmanın C_{max} değerini düşürdüğü açıkça görülmektedir.

3.3. Önerilen Değişken Komşuluk Arama Algoritması

Değişken komşuluk arama algoritması önceden tanımlı komşuluk yapılarının sistematik şekilde değiştirilmesi ve bulunan çözümün uygunluğunun her aşamada kontrol edilmesi fikrine dayanır. Komşuluklardaki çözümler de yerel arama teknikleri ile iyileştirilmeye çalışılır. DKA uygulanırken komşuluk yapıları çözüm uzayında farklı alanları etkin taramak adına son derece önemlidir.

Daha önce de bahsedildiği üzere bu çalışma kapsamında komşuluk seçimi ve yerel arama seçimlerinin uyarlandığı yeni bir DKA algoritması önerilmiştir. Önerilen algoritma ve Temel DKA algoritmasının kıyaslanması Tablo 8’de verilmiştir.

Tablo 8. Önerilen ve Temel DKA algoritması kıyaslaması

Temel Değişken Komşuluk Arama	Önerilen Değişken Komşuluk Arama
<p>Başlangıç Çözümünü Oluştur. Komşuluk yapılarını belirle (N_k, $k=1,2, \dots, k_{max}$). Durma kriteri sağlanana kadar Tekrarla: $k=1$ $k=k_{max}$ olana kadar tekrarla: k. Komşuluk ile x çözümden x' çözümünü oluştur. ($x' \in N_k(x)$). x' Çözümüne yerel arama prosedürünü uygula ve x'' çözümünü bul Eğer $f(x'') < f(x)$ ise: $k \leftarrow 1$, $x \leftarrow x''$ Değilse: $k \leftarrow k+1$ Çıktı x</p>	<p>Başlangıç Çözümünü Oluştur. (Açgözlü Başlangıç Çözüm Algoritmaları ile) Komşuluk yapılarını belirle (N_k, $k=1,2, \dots, k_{max}$). Durma kriteri sağlanana kadar tekrarla: $k=1$ $k=k_{max}$ olana kadar tekrarla: DKİ_KM_SP komşuluk algoritması ile x çözümden x' çözümünü oluştur. x' Çözümüne YSS yerel arama seçim stratejisini uygula ve x'' çözümünü bul Eğer $f(x'') < f(x)$ ise: $k \leftarrow 1$, $x \leftarrow x''$ Değilse: $k \leftarrow k+1$ Çıktı x</p>

Önerilen DKA algoritması, Temel DKA algoritmasından başlangıç çözümü oluşturmada, komşuluk yapıların seçiminde ve yerel arama seçimlerinde farklılıklar barındırmaktadır. Bölüm 3.3.1'de literatürde bulunan komşuluk yapıları ve yeni önerilen komşuluk değişim algoritmasından bahsedilmiştir. Bölüm 3.3.2 de kullanılan yerel arama tekniklerinden bahsedilmiş ve bölüm 3.3.3'te önerilen yerel arama seçim stratejisi anlatılmıştır. Bölüm 3.3.4'te ise bu çalışma kapsamında kıyaslanan algoritmalarındaki çalışma zamanı ve algoritma durdurma kriterlerinden bahsedilmiş ve makine iş sayısına göre yeni bir algoritma maksimum iterasyon sayısı belirleyen fonksiyon tanımlanmıştır.

3.3.1. Çalışmada Kullanılan Komşuluk Yapıları

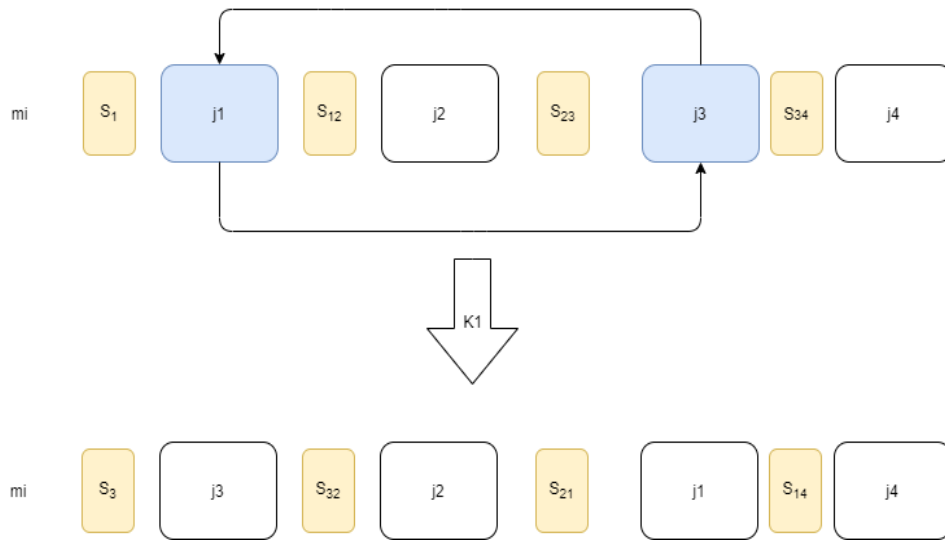
Çalışmada temelde 5 farklı komşuluk yapısı kullanılmıştır. Bu komşuluk yapılarına yeni parametreler eklenerek daha etkin şekilde kullanılması hedeflenmiştir.

3.3.1.1. Literatürde Bulunan Komşuluk Yapıları

Aşağıda tanımlanan komşuluk yapıları literatürde birçok araştırmacı tarafından başarı ile kullanılmıştır. Sıra bağımlı ilişkisiz paralel makine çizelgeleme problemlerinde daha iyi bir çözüm araştırmak için, tipik 5 farklı komşuluk yapıları kullanılır. Bunlardan üçü makine içi değişimi ikisi ise makineler arası iş değişimi ile yapılır. Makine içi değişimler aynı makinede iki işin yer değiştirmesi (K1), bir işin daha sonraki bir işin

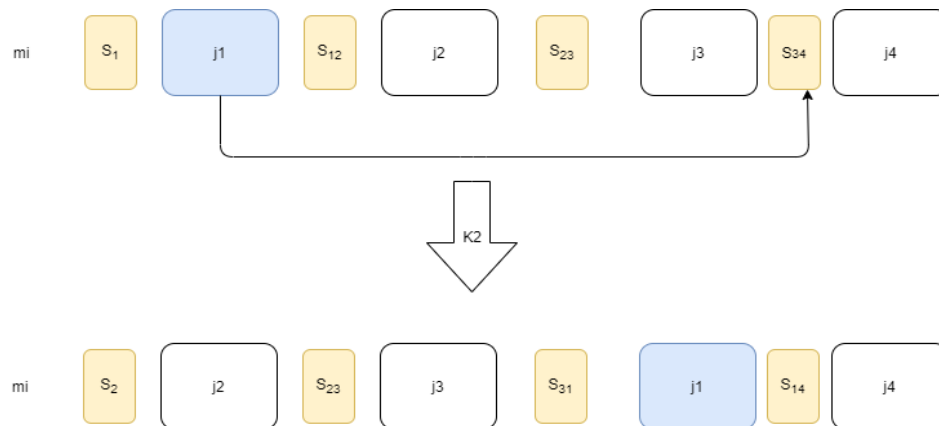
arkasına eklenmesi (K2) ve aynı makine üzerinde iki iş arasındaki sıralamanın tersine çevrilmesidir (K5). Makineler arası iş değişimi de iki farklı makinedeki işlerin değiştirilmesi (K3) ve bir makinedeki işin başka bir makinedeki işin arkasında eklenmesi (K4) ile yapılır (Wang vd. 2016:235).

K1: Aynı makinede işleri yer değiştir: Kullanılan ilk komşuluk yapısı aynı makinede işlerin yer değiştirilmesidir. Rasgele seçilen bir makineden, rastgele seçilen iki iş yer değiştirilir. Komşuluk yapısının uygulanması Şekil 3’ te gösterilmiştir.



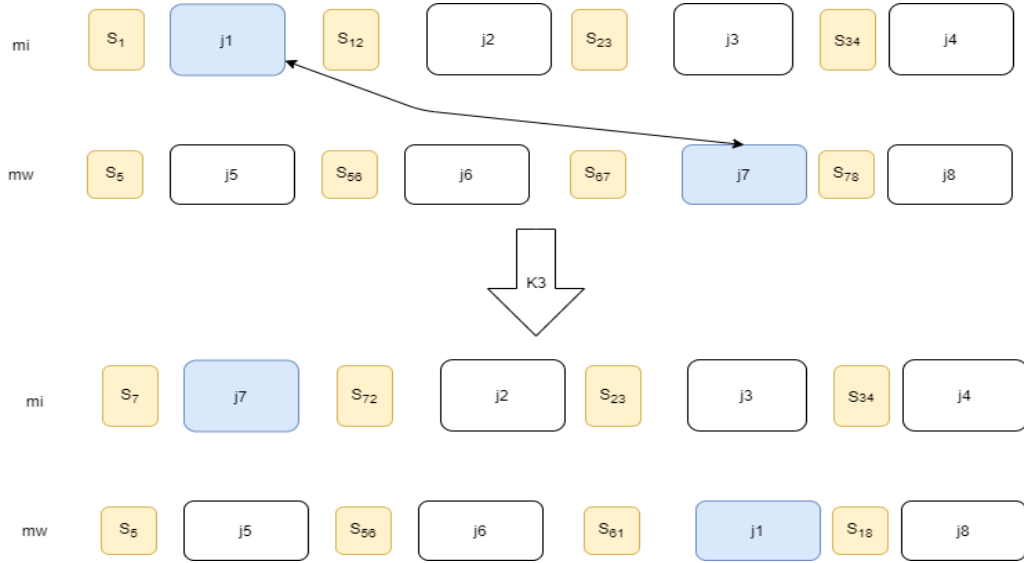
Şekil 3. K1 Komşuluk Yapısının Uygulanması

K2: Aynı makineden bir işi başka bir yere ekle: Bu komşuluk yapısında aynı makinedeki bir işin başka bir yere eklenmesi söz konusudur. Rastgele seçilen bir makineden rasgele seçilen bir iş aynı makine üzerinde rasgele bir noktaya eklenir. Komşuluk yapısının uygulanması Şekil 4’te gösterilmiştir.



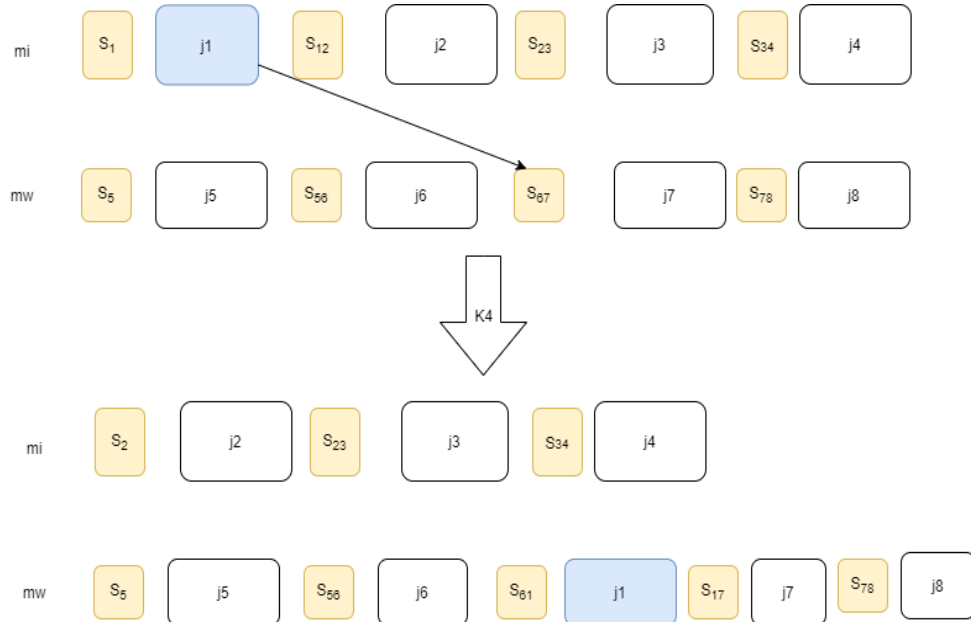
Şekil 4. K2 Komşuluk Yapısının Uygulanması

K3: Farklı iki makineden işleri yer değiştir: Bu komşuluk yapısında ise rasgele seçilen 2 makineden, rasgele 2 iş yer değiştirilir. K3 komşuluk yapısı gösterimi Şekil 5'te verilmiştir.



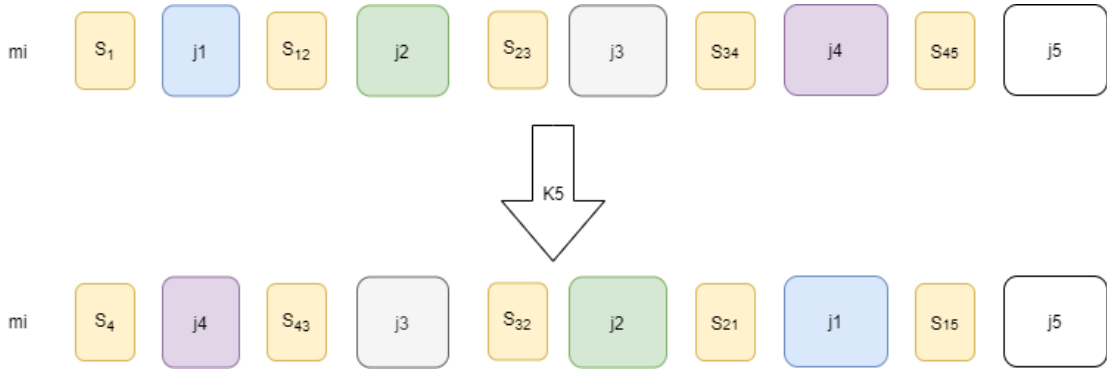
Şekil 5. K3 Komşuluk Yapısının Uygulanması

K4: İşi farklı makinede bir yere ekle: Bu komşuluk yapısında Rasgele seçilen bir makinedeki iş bir başka rasgele makinedeki bir noktaya eklenir. K4 komşuluk yapısı gösterimi Şekil 6'da verilmiştir.



Şekil 6. K4 Komşuluk Yapısının Uygulanması

K5: Blok Tersine çevirme: Bu komşuluk yapısında rasgele seçili bir makinede 2 nokta arasındaki işler tersine çevrilir. K5'in gösterimi Şekil 7'de verilmiştir.



Şekil 7. K5 Komşuluk Yapısının Uygulanması

3.3.1.2. Önerilen DKİ Makine Seçim Parametresi

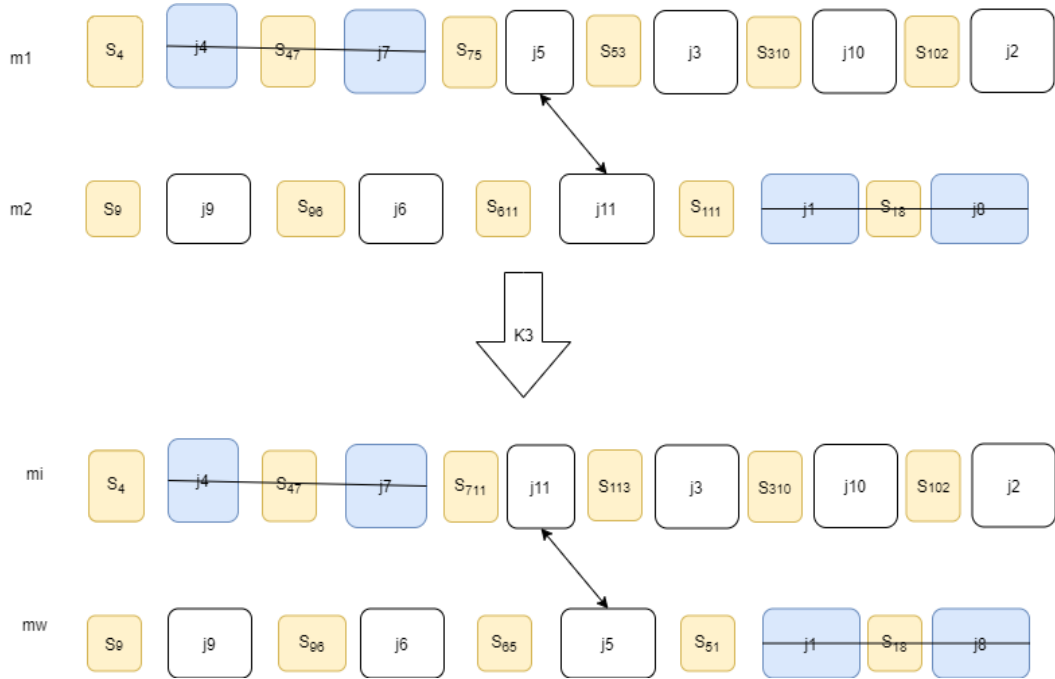
Amaç fonksiyonu C_{max} olan çizelgeleme problemlerinde C_{max} değerini en fazla yüklü makine belirlemektedir. Bu makine kritik makine olarak adlandırılabilir. Komşuluk değişimlerinde rastgele iki makine seçilirken bu makinelerden birinin kritik makine olmaması durumunda, komşuluk yapısı uygulandığında seçilen makinelerdeki iş yükleri değişse bile C_{max} değeri düşmemektedir. Bu nedenle literatürde bulunan komşuluk yapılarına kritik makine parametresi (kmp) eklenmiştir. kmp parametresi 1 olduğu durumda uygulanacak komşuluk yapısında etkilenen makine sayısı 1 ise bu makinenin kritik makine olması sağlanır. Eğer etkilenen makine sayısı 2 ise bu makinelerden birinin kritik makine olması sağlanır. Bu parametre sayesinde komşuluk yapısında C_{max} değerinin bir an önce küçültülmesi hedeflenir. Önerilen kmp parametresi sayesinde çözüm uzayının daha etkin araştırılması hedeflenmiştir.

3.3.1.3. Önerilen DKİ Serbestlik Parametresi

Çalışmada, komşuluk yapılarında değişebilecek işleri kısıtlamak için, serbestlik parametresi (sp) önerilmektedir. Önerilen serbestlik parametresi N toplam iş sayısı olmak üzere $sp = [0- N/2)$ arasında değerler alabilmektedir. Komşuluk değişiminde sp kritik makine seçimi ile, farklı bir senaryoda sadece kritik makine işlerinin değişimini kısıtlamaya zorlanabilir. Bu durumda sp değişken kritik makinedeki iş sayısı n_{km} olmak üzere $sp = [0- n_{km} /2)$ arasında değerler alabilir. $sp=0$ olduğu durumda iş değişimi sırasında bütün işlerin seçilebilmesi mümkündür. sp parametresinin 1 artışına karşılık makinelere atanan işler arasında en düşük işlem ve hazırlık süresine ait ardışık 2 iş belirlenir. Belirlenen en düşük hazırlık ve işlem süreli işler, atandığı makinede tutularak değişime engellenmektedir.

Örnek üzerinde açıklamak gerekirse $sp = 2$ olarak alındığı 11 iş ve 2 makinenin bulunduğu bir durumda (test veri kümesi 11on2Rp50Rs50_1Ameer adlı veri seti) 1. makine (m1), 2. makine (m2) olmak üzere iş atamaları $m1 = \{4,7,5,3,10,2\}$, $m2 = \{9,6,11,1,8\}$ şeklinde olsun. Öncelikle makine iş atamalarında sırası ile makineye atanan bütün makine iş ikilileri arasındaki $p_i + s_{ij} + p_j$ süreleri hesaplanır. En az süreye sahip makine iş atamaları belirlenir. Yukarıdaki örnek için en düşük 2 atama değerleri m1 makinesinde 4 ve 7 ($p_4 + s_{47} + p_7 = 168$) ve m2 makinesinde 1 ve 8 ($p_1 + s_{18} + p_8 = 177$) olarak bulunmuştur. Komşuluk yapılarında iş değişiminde bu işler en düşük değere sahip olduğundan değişime uğramayacaktır.

Örnek olarak $sp = 2$ ve rasgele seçilen işlerin 5 ve 11 olduğu K3 komşuluk yapısının gösterimi Şekil 8’de verilmiştir.



Şekil 8. $sp=2$ Olduğu Durumda K3 Komşuluk Yapısının Uygulanması

Komşuluk yapısına yeni önerilen parametrelerin etkisini ölçmek için yeni bir DKİ algoritması önerilmiştir. Bölüm 3.2’de tanıtilen rasgele ve ağgözlü başlangıç çözümleri ile oluşturulan ilk çözümlere tanıtilen parametreler uygulanarak DKİ algoritması test edilmiştir. Bölüm 3.3.1.4’ te algoritma detaylı şekilde açıklanmıştır.

3.3.1.4. Önerilen Değişken Komşuluk İniş Test Algoritması

DKİ iniş algoritmasına önerilen yeni parametreler eklenerek DKİ_KM_SP algoritması önerilmiştir. Bu parametreler 4 farklı senaryo ve 2 farklı başlangıç çözümü

ile test edilmiştir. Bölüm 3.1.3.2' teki kritik makine seçimi (kmp) ve Bölüm 3.1.3.3' teki serbestlik parametresi (sp) olmak üzere senaryolar ve parametrelerin değerleri Tablo 9'da verilmiştir.

Tablo 9. Önerilen Değişken Komşuluk İniş Test Algoritması Senaryoları

Senaryo	kmp	sp	Açıklama
senaryo-1	0	0	Makine ve iş seçimi serbest komşuluk değişiminde bütün makineler ve işler seçilebilir.
senaryo-2	1	0	Makine seçiminde seçilen makinenin birisi mutlaka kritik makine olmalıdır. İş seçiminde bütün işler seçilebilir.
senaryo-3	1	deg_sp	Makine seçiminde seçilen makinenin birisi mutlaka kritik makine olmalıdır. İş değişiminde ise sp parametresine göre işlerin bir kısmının değişimine izin verilmez.
senaryo-4	1	km_sp	Makine seçiminde seçilen makinenin birisi mutlaka kritik makine olmalıdır. İş değişiminde ise sp parametresine göre sadece kritik makinedeki işlerin bir kısmının değişimine izin verilmez.

Tablo 9'deki parametreler, Bölüm 3.2.1' deki rasgele ve Bölüm 3.2.2' deki açgözlü başlangıç algoritmaları ile oluşturulan başlangıç algoritmaları ile test edilmiştir. Test veri kümesindeki her bir örnek 50 defa çalıştırılmıştır. Algoritmanın durdurma kriteri iş sayısına eşittir. İş sayısına göre durdurma kriteri tanımlanarak daha büyük problem setleri daha fazla sayıda çalıştırılmıştır.

DKİ_KM_SP algoritmasının parametreleri şu şekildedir. Komşuluk yapıları = $(N_k, k=1,2, \dots, k_{max})$, durdurma kriteri ($maxiterasyonSayısı=iş\ sayısı$), x = başlangıç çözümü, senaryo = (senaryo-1, senaryo-2, senaryo-3, senaryo-4), $N=iş\ sayısı$

Bu parametreler ile oluşturulan DKİ_KM_SP algoritmasının sözde kodu aşağıda verilmiştir.

```

DKİ_KM_SP ( $N_k, x, senaryo, maxiterasyonSayısı, N$ ):
( $i = iterasyon sayısı$ ) <=  $maxiterasyonSayısı$ :
  Eğer ( $senaryo = senaryo-1$ ) ise  $sp=0, kmp=0$ 
  Eğer ( $senaryo = senaryo-2$ ) ise  $sp=0, kmp =1$ 
  Eğer ( $senaryo = senaryo-3$ ) ise  $maxsp = n/2, kmp =1$ 
  Eğer ( $senaryo = senaryo-4$ ) ise  $maxsp = kritik\ makine\ iş\ sayısı/2, kmp =1$ 
   $k=1$ 

```

$k=k_{\max}$ olana kadar tekrarla:
 Eğer (senaryo=senaryo-3 veya senaryo=senaryo-4) ise
 $sp=\text{mod}(i, \text{maxsp})$
 k . Komşuluk ile x çözümden x' çözümünü oluştur. ($x' \in N_k(x, sp, kmp)$)
 Eğer x' çözüm x çözümünden iyi ise $x \leftarrow x'$, $k=1$
 $i = i+1$
 Çıktı x
 Bitir.

Serbestlik parametresi iterasyonlar boyunca maksimum alacağı değere göre modu alınarak başlangıçta 0 değeri ile başlar. Bu durumda bütün işlerin seçimi rastgeledir ve daha geniş bir arama yapılmasına olanak verir. sp parametresi alabileceği en büyük değere yaklaşırken kritik makine üzerinde en az maliyetli işlerin değişimi sınırlandırılmış daha dar alanda daha ince aramalara olanak sağlanmış olur.

DKİ_KM_SP büyük veri setinde, her iki başlangıç çözümü ile her veri seti için 50 kez çalıştırılmıştır. İlk çözümlerin geliştirilme yüzdelerinin ortalamaları iş ve makine sayısına göre gruplanmıştır. Ayrıca bilinen en iyi çözüme (Arnaout (2020)) göre hata yüzdeleri hesaplanmıştır. Bulunan değerler her iki başlangıç çözümünde, iş sayısı ve makine sayısına göre gruplanmıştır. Tablo 10 - Tablo 15'teki sonuçlar elde edilmiştir. Tablolardaki en iyi çözümler koyu renk ile belirtilmiştir.

Tablo 10. Rasgele Başlangıç Çözümü ile DKİ_KM_SP Makine Sayısına Göre İlk Çözüm Gelişim Yüzdeleri

Makine sayısı	senaryo-1	senaryo-2	senaryo-3	senaryo-4
2	8,594	9,521	11,039	10,359
4	7,62	10,106	11,78	10,869
6	8,591	11,707	12,941	12,295
8	8,186	11,465	12,507	11,929
10	6,884	10,115	10,678	10,469
12	8,397	12,302	12,69	12,537

Tablo 10 incelendiğinde, makine sayısına göre rasgele başlangıç çözümünü geliştirmede en başarılı senaryonun, makinelerin birinin kritik makine seçimine zorlandığı ve serbestlik parametresinin değişken olduğu senaryo-3 olduğu görülmektedir.

Tablo 11. Rasgele Başlangıç Çözümü ile DKİ_KM_SP İş Sayısına Göre İlk Çözüm Gelişim Yüzdeleri

İş sayısı	senaryo-1	senaryo-2	senaryo-3	senaryo-4
20	9,533	13,87	13,501	13,939
40	8,809	11,978	12,915	12,486
60	7,834	10,429	11,739	11,03
80	7,663	10,062	11,496	10,697
100	7,794	10,114	11,675	10,853
120	6,638	8,763	10,309	9,452

Tablo 11'e bakıldığında, algoritmaların geliştirme yüzdeleri iş sayısına göre gruplandığında rasgele başlangıç çözümünü geliştirmede en başarılı senaryonun, makinelerin birinin kritik makine seçimine zorlandığı ve serbestlik parametresinin değişken olduğu senaryo-3 olduğu görülmektedir.

Tablo 12. Açgözlü Başlangıç Çözümü ile DKİ_KM_SP Makine Sayısına Göre İlk Çözüm Gelişim Yüzdeleri

Makine sayısı	senaryo-1	senaryo-2	senaryo-3	senaryo-4
2	0,151	0,177	0,202	0,212
4	0,4	0,679	0,721	0,777
6	0,909	1,74	1,821	1,848
8	0,856	2,071	2,066	2,182
10	1,238	2,785	2,641	2,967
12	1,648	3,776	3,673	3,871

Tablo 12'ye bakıldığında, algoritmaların geliştirme yüzdeleri makine sayısına göre gruplandığında, açgözlü başlangıç çözümünü geliştirmede en başarılı senaryonun makinelerin birinin kritik makine seçimine zorlandığı ve kritik makinedeki işlere serbestlik parametresinin uygulandığı senaryo-4 olduğu görülmektedir.

Tablo 13. Açgözlü Başlangıç Çözümü ile DKİ_KM_SP İş Sayısına Göre İlk Çözüm Gelişim Yüzdeleri

İş sayısı	senaryo-1	senaryo-2	senaryo-3	senaryo-4
20	2,053	4,3	3,851	4,325
40	1,172	2,375	2,404	2,51
60	0,77	1,7	1,784	1,846
80	0,499	1,143	1,194	1,236
100	0,458	1,065	1,159	1,183
120	0,25	0,644	0,733	0,758

Tablo 13'e bakıldığında, algoritmaların geliştirme yüzdeleri iş sayısına göre gruplandığında, açgözlü başlangıç çözümünü geliştirmede en başarılı senaryonun makinelerin birinin kritik makine seçimine zorlandığı ve kritik makinedeki işlere serbestlik parametresinin uygulandığı olduğu senaryo-4 olduğu görülmektedir.

Tablo 14. Rasgele Başlangıç Çözümü ile DKİ_KM_SP En iyi Çözüme Hata

senaryo-1	senaryo-2	senaryo-3	senaryo-4
21,805	17,387	15,738	16,606

Tablo 14'te görüleceği üzere rasgele çözümü geliştirirken senaryo-3 başlangıç çözümünü bilinen en iyi çözüme %15,738 yaklaştırarak en başarılı algoritma olmuştur.

Tablo 15. Açgözlü Başlangıç Çözümü ile DKİ_KM_SP En iyi Çözüme Hata

senaryo-1	senaryo-2	senaryo-3	senaryo-4
5,01	4,109	4,115	4,029

Tablo 15 açgözlü çözümü geliştirirken, senaryo-4'ün kullanılmasının başlangıç çözümünü bilinen en iyi çözüme %4,029 yaklaştırarak en başarılı algoritma olduğunu göstermektedir.

Tablo 14 ve Tablo 15'e bakıldığında bilinen en iyi çözüme yaklaşık hata arasındaki farkın başlangıç algoritmalarındaki kaliteden kaynaklandığı görülmektedir. Bu DKA algoritmasının açgözlü başlangıç algoritması ile başlanması konusunda önemli bir göstergedir.

3.3.2. Yerel Aramalar

Yerel arama mevcut bulunan bir aday çözümü daha da iyileştirme çabasıdır. Yerel arama ile bulunan çözümün global en iyi çözüm olma garantisi yoktur. DKA'da yerel arama ile komşuluk yapısı uygulanarak bulunan çözümünün daha da iyileştirilmesi amaçlanır. Bulunan çözüm aday ilk çözüm ile kıyaslanır ve belirlenen amaç fonksiyonuna göre daha iyi bir çözüm elde edildiyse bulunan iyi çözüm aday çözüm ile yer değiştirilir. Yerel aramalar DKA algoritması için son derece önemlidir. Bu çalışmada 4 farklı yerel arama önerilmiştir.

Bu çalışma kapsamında kullanılan yerel aramalardan ikisi geniş çaplı ve maliyetli yerel aramalardır. Bu yerel aramalardan ilki (**YA1**) kritik makinedeki bütün işleri sırası ile listedeki bütün işler ile değiştirmeyi ve amaç fonksiyonuna göre daha uygun çözüm değerini elde etmeyi hedefler. İkinci yerel arama (**YA2**) ise yine kritik makinedeki bütün işleri sırası ile diğer makinelerdeki işlerin öncesine ekleme daha uygun çözüm değerini elde etmeyi hedefler. Eğer öncesine eklediği iş makinedeki son işlenecek iş ise sonrasına da eklemeyi dener. YA1 ve YA2 yerel aramalar amaç fonksiyonunda bulunan en iyi çözümden daha maliyetli sadece 1 makine var ise etkili çalışır. Bir önceki iterasyona göre kritik makineden maliyetli birden fazla makine olması durumunda kritik makinedeki işleri değiştirmek amaç fonksiyonunda gelişmeye olanak sağlamayacaktır.

Diğer iki yerel arama kritik makinedeki işler ile ilgilenmez ve daha geniş alanda uç noktalarda araştırma olanağı arar. Bu yerel aramalardan (**YA3**) bir önceki iterasyonda elde edilen C_{max} değerinden daha büyük 2 makinedeki bütün işleri kendi arasında değiştirerek iki makinenin de aynı anda daha iyi çözümlenmesini hedefler. Bu çalışmada

ayrıca keşif yengeçlerinin hareketlerini taklit eden doğadan esinlenilmiş yeni bir yerel arama da önerilmiştir (YA4). Yukarıdaki belirtilen 4 farklı yerel arama aşağıda sırası ile verilmiştir.

YA1: Kritik Makinedeki Bütün İşleri Sırasıyla Farklı Bütün İşlerle Değiştirmeyi Dene

Girdi: L: Makine-iş ikililerinin tutulduğu liste M: bütün makineler, j: seçilen iş, m_k : kritik makine; jm: m makinesindeki iş

L listesine göre m_k =en yüksek maliyetli makineyi bul ()

Her $j \in m_k$ işi için tekrarla

Her $m \in M$ makinesi için

Her $jm \in m$ için

j işini jm işi ile yer değiştir ().

Yeni liste maliyetini hesapla (). Eğer yeni maliyet düşük ise L listesini güncelle.

Çıktı: L

YA2: Kritik Makineden Bütün İş Sırasıyla Farklı Bütün Yerlere Eklemeyi Dene

Girdi: L: Makine-iş ikililerinin tutulduğu liste M: bütün makineler, j: seçilen iş, m_k : kritik makine; jm: m makinesindeki iş

L listesine göre m_k =en yüksek maliyetli makineleri bul ()

Her $j \in m_k$ işi için tekrarla

Her $m \in M$ makinesi için

Her $jm \in m$ için

j işini jm işinin öncesine ekle()

Yeni liste maliyetini hesapla (). Eğer yeni maliyet düşük ise L listesini güncelle.

Eğer jm m makinesindeki son iş ise:

j işini jm işinin sonrasına ekle(). Yeni liste maliyetini hesapla (). Eğer yeni maliyet düşük ise L listesini güncelle.

Çıktı: L

YA3: Aşırı Yüklü İki Makineyi Yeniden Sırala

Girdi: L: Makine-iş ikililerinin tutulduğu liste M: bütün makineler, j: seçilen iş, m_{k1} : bir önceki iterasyondan daha maliyetli ilk kritik makine, m_{k2} = bir önceki iterasyondaki C_{max} değerinden daha maliyetli ikinci kritik makine.

L listesine göre M maliyetini hesapla ve m_{k1} =en yüksek maliyetli 1.makineyi bul (), m_{k2} = en yüksek maliyetli 2.makineyi bul (),

Her $jm1 \in m_{k1}$ işi için tekrarla:

Her $jm2 \in m_{k2}$ işi için tekrarla:

jm1 ve jm2 işini yer değiştir()

L listesinin maliyetini hesapla maliyet düştü ise
listeyi güncelle ()

Çıktı: L

YA4: Önerilen Her Makineden Rasgele Birer İş Çıkar Yeni İşleri Yeniden Ata

Keşif Yengeçleri

Kaynak planlamasındaki boşluk zinciri, *Pagurus Longicarpus* türü yengeçleri ile de tanımlanmıştır. Boş pozisyon zincirleri aracılığıyla kaynak dağıtımında, bir kişi başlangıçtaki boş bir kaynak birimini aldığı anda, ikinci bir kişi tarafından alınabilecek eski birimini geride bırakır ve bu böyle devam eder. Bu süreç, birbiri ardına boşaltılan kaynak birimlerinin birbiriyle bağlantılı bir zinciri olarak özetlenebilir; bu zincir, bir dizi bireyin, kaynakların başkaları tarafından önceki edinimine bağlı olarak yeni birimler edinmesine izin verir (Chase, I. D. vd. 1988:1265). Yengeçlerin kabuk değiştirme eş zamanlı ve eş zamansız olmak üzere iki farklı şekilde modellenmiştir. Senkronize boşluk zincirleri, mevcut bir boş kabuğa bitişik birkaç yengeç küçülen boyut sırasına göre sıraya girdikten sonra oluşur. En büyük yengeç boş kabuğa geçer geçmez, hızlı bir dizi sıralı kabuk değiştirme gerçekleşir. Eş zamansız boşluk zincirlerinde, uygun bir boş kabukla karşılaşan bir yengeç kabuğunu değiştirecek ve daha sonra atılan kabuğu bir başka yengecin keşfedip kullanması beklenecektir. Bu nedenle, eş zamansız boşluk zincirleri sosyal etkileşimleri veya kuyruk oluşumunu içermez ve sıralı kabuk geçişleri önemli ölçüde daha uzun zaman dilimlerinde gerçekleşir. Senkronize kabuk değiştirmede en uygun kabuğun bulunması ihtimali daha yüksektir (Rotjan vd. 2010: 639-640).

Yengeçlerin kabuk değiştirme algoritması ilk defa 2019 yılında önerilmiştir. Önerilen algoritma her bir yengecin edindiği kabuktan daha iyi bir kabuk araması üzerine kurulmuştur. Algoritma 3 adımdan oluşur. Her bir yengeç her seferinde rasgele bir yeni bir kabuk bulur. Bulunan en iyi kabuk bir sonraki nesle taşınır. Kabuk sayısı sabittir ve yengeçlerin kabukları bulma ihtimalleri belli olasılıklar ile tanımlıdır (Murali vd. 2019). Bu yöntem yengeçlerin eş zamansız kabuk değiştirmesine benzemektedir. Önerilen algoritmada makinelerden seçili işler kabuklarından çıkarılan ve yeni kabuklar arayan yengeçlere benzetilmektedir. Uygun kabuk boşlukları her bir makineden birer iş çıkarıldıktan sonra kalan makinelerdeki yüklere göre hesaplanır.

Girdi: L: Makine-iş ikililerinin tutulduğu liste M: bütün makineler, j: seçilen iş, m = seçili makine, AĪ = atanmayan işler, jm: seçili makineden rasgele bir iş

Her m makinesi $\in M$ olmak üzere her bir m makinesinden rastgele bir iş al (jm) ve işi $A\bar{I}$ listesine ekle (). (Seçilen işler kabuklarından çıkmış yengeçlere benzemektedir.)

L listesinden $A\bar{I}$ deki işlerin bulunduğu makine iş ikililerini çıkar().

L listesindeki kalan işlere göre makine maliyetleri hesapla(). (En yüksek maliyetli makine en düşük kabuğa sahiptir.)

$A\bar{I}$ listesi boş olana kadar tekrarla:

En yüksek maliyetli makineden en az yüklü makineye doğru her makine m için tekrarla:

$j = A\bar{I}$ listesinden makinede işlenecek en kısa süreli işi bul(). (İşleri yengeçler gibi sırala)

L listesine (m, j) ikilisini ekle.

j işini m makinesinde en uygun pozisyona yerleştir.

j işini $A\bar{I}$ listesinden çıkar

L listesini güncelle

Çıktı: L

Yukarıda bahsedilen 4 yerel arama iki farklı stratejide gruplanmış ve bölüm 3.3.3'te açıklanmıştır. Yerel arama algoritmalarının işlem karmaşıklıkları şu şekilde hesaplanır. N toplam iş sayısı M toplam makine sayısı olmak üzere işlerin makinelere dengeli dağıldığı varsayılır ise her bir makinede N/M kadar iş olması beklenir. YA1' de kritik makinedeki her bir iş diğer bütün işler ile yer değiştirildiğinden işlem karmaşıklığı $N * (N/M)$ kadar olur. YA2'de işlerin makinelerin önüne ve arkasına eklenme durumu vardır ve kritik makine üzerine atanan işler $N + M$ adet yere eklenebilir. Bu durumda işlem karmaşıklığı $(N + M) * (N/M)$ kadar olur. YA3 yerel aramada sadece 2 makinedeki işler kendi arasında değiştiğinden işlem karmaşıklığı $(N/M) * (N/M)$ kadar olur. YA4' te ise her bir makineden 1 iş alınır ve toplam yeni sıralanacak iş sayısı M adet olur. M adet iş her bir makinedeki N/M kadar işin arasında uygun pozisyona yerleştirilmeye çalışıldığından işlem karmaşıklığı $M * (N/M)$ kadar olur. Algoritmaların karmaşıklıklarını hesaplamayla başa çıkmak için Bachman tarafından 1892'de tanıtılan ve büyük O adı verilen notasyon kullanılmıştır. Bu notasyon, üzerinde çalıştıkları girdi veri kümesinin boyutuna bağlı olarak algoritmaları nasıl performans gösterdiklerine göre sınıflandırmak için kullanılır. Bu genellikle hem çözüm uzayı büyüklüğüne hem de zaman davranışlarına bakmak anlamına gelir (Chivers ve Sleightholme 2015:359). Yerel arama işlem karmaşıklıkları Tablo 16'da verilmiştir.

Tablo 16. Yerel Arama İşlem Karmaşıklıkları

Yerel Arama	İşlem Karmaşıklığı	Büyük O Notasyonu
YA1	N^2/M	$O(N^2/M)$
YA2	$(N^2 + M * N) / M$	$O(N^2/M)$
YA3	N^2/M^2	$O(N^2/M^2)$
YA4	N	$O(N)$

Tablo 16 incelendiğinde yerel arama işlem karmaşıklıkları genelde iş sayısına göre karesel artarken, makine sayısına bağımlı olarak düşmektedir. Algoritma çalışma süresi belirlenirken bu karmaşıklıklar göz önüne alınacaktır.

3.3.3. Önerilen Yerel Arama Seçim Stratejisi

Önerilen DKA algoritmasında iki farklı yerel arama aşaması önerilmiştir. Bu önerilen aşamalar komşuluk değişiminden elde edilen sonuca göre hangi yerel aramaların kullanılacağına karar veren bir yapıdır. Bunlardan ilki hassas ve detaylı arama aşaması olan ince arama aşaması iken, diğeri ise daha geniş alanlarda daha az hassas arama olan geniş arama aşamasıdır. İnce arama aşaması komşuluk değişimi sonucunda bilinen en iyi çözümden daha kötü veya eşit 1 adet kritik makine bulunması durumunda komşuluk değişiminden elde edilen listeye YA1 ve YA2 yerel aramaları uygular. Geniş arama ise birden fazla kritik makinenin bulunduğu durumda YA1 ve YA2 yerel aramaların mevcut çözümü iyileştirme olanağı bulunmadığından çözüme YA3 ve YA4 yerel aramaları uygular. Yerel arama seçim stratejisi sözde kodu aşağıdaki gibidir.

Girdi: $C_{maxEniyi}$: Bulunan en iyi C_{max} değeri, KL: komşuluk yapısı uygulanmış liste, Yerel Aramalar: YA1, YA2, YA3, YA4, YL: yerel arama sonucu olan liste.

KL listesindeki kalan işlere göre makine maliyetleri hesapla (), YL= \emptyset

Maliyetleri $C_{maxEniyi}$ değerine göre kıyasla ()

Eğer kritik makine sayısı =1 ise:

YL=İnce Arama Aşaması (YA1, YA2)

Eğer YL listesinin maliyeti L listesinden düşük ise L listesini YL ile güncelle ()

Eğer kritik makine sayısı >1 ise:

YL=Geniş Arama Aşaması (YA3, YA4)

Eğer YL listesinin maliyeti L listesinden düşük ise L listesini YL ile güncelle ()

Çıktı: L

3.3.4. Önerilen Yeni Değişken Komşuluk Arama Algoritmasının Parametreleri

DKA arama algoritmasının durdurma kriteri izin verilen maksimum CPU süresi, maksimum yinleme sayısı veya iki iyileştirme arasındaki maksimum yinleme sayısı

olabilir (Hansen, P., Mladenović, N., 2002: 416). İncelenen problemin çözümünde araştırmacılar tarafından farklı durdurma kriterleri ve algoritma çalışma süreleri belirlenmiştir. Chang ve Chen (2011) maksimum iterasyon sayısını iş sayısının 500 katı olarak belirlemiştir. Ying vd. (2012) 214 iterasyon veya 28 defa iyileşme olmadığı durumlarda algoritmayı sonlandırmıştır. Helal vd. (2006) makine içi ve makineler arası değişimini makine sayısına göre değişen bir iterasyon sayısına bağlamışlar ve makineler arası değişimi en az 25 kez çalıştırmışlardır. Arnaout vd. (2010) görece küçük problemlerde az iterasyon sayısının yeterli olduğunu belirtmiş, bununla birlikte büyük veri setinde bilinen en iyi sonuçları elde etmek için algoritmayı 15000 iterasyon çalıştırmışlardır. Rabadi vd. (2006) maksimum çalıştırılacak iterasyon sayısını 5000'e sabitlemişlerdir. Arnaout (2020) max iterasyon sayısını 5000-15000 arasında denemiş en iyi sonuçların 14000 iterasyon ile elde edilebileceğini belirtmişlerdir. Arnaout vd. (2010) maksimum iterasyon sayısının 15000 olduğunu ve çoğu durumlarda iterasyon sayısının 5000-7000 arasında yeterli olduğunu kalan denemelerin boşa gittiğini belirtmiştir. Arnaout vd. (2014) çalışmasında yine maksimum iterasyon sayısını 15000 olarak belirtmiştir. De Abreu, L. R., ve De Athayde (2020) algoritmanın toplam çalışma süresi daha önce Vallada ve Ruiz (2011) tarafından önerilen iş sayısı (N) ve makine sayısına (M) bağımlı bir fonksiyon olarak ele almış ve Eşitlik 10'dan yararlanmıştır. t değeri küçük boyutlu problemlerde 250 iken büyük problemlerde 50 olarak alınmıştır.

$\text{çalışma zamanı} = N \times \frac{M}{2} \times t \text{ ms}$	(10)
--	------

Bu çalışmada Tablo 16'da ki yerel aramaların işlem karmaşıklıkları değerlendirildiğinde makine başına düşen iş sayısı arttıkça işlem karmaşıklığının arttığı görülmektedir. Makine sayısı $m = \{2, 4, 6, 8, 10, 12\}$ ve toplam iş sayısına (N) bağlı bir toplam iterasyon sayısı önermekteyiz. Öncelikle toplam veri kümesindeki ortalama makine sayısı eşitlik 11 ile hesaplanır. Eşitlik 11 ile ortalama makine sayısı 7 olarak bulunmuştur. İş ve makine karmaşıklığı bu ortalama makine sayısına bölünerek, makine başına iş sayısı arttıkça eşitlik 12 ile algoritmanın göreceli olarak daha az çalışması sağlanır.

$m_{sayiOrt} = \frac{\sum_{m=2}^{12} m}{6} = 7$	(11)
---	------

Eşitlik 11 ile ortalama makine sayısı adında bir değişken oluşturulmuş ve bu sayı eşitlik 12’de *iterasyon sayısı* hesaplanırken kullanılmıştır.

$$\text{iterasyon sayısı} = N \times \frac{M}{m_{\text{sayiOrt}}} \quad (12)$$

Eşitlik 12 sayesinde iş sayısına göre problemin iterasyon sayısı artarken makine sayısı fazla olan problemlerde de daha fazla artış sağlanır.

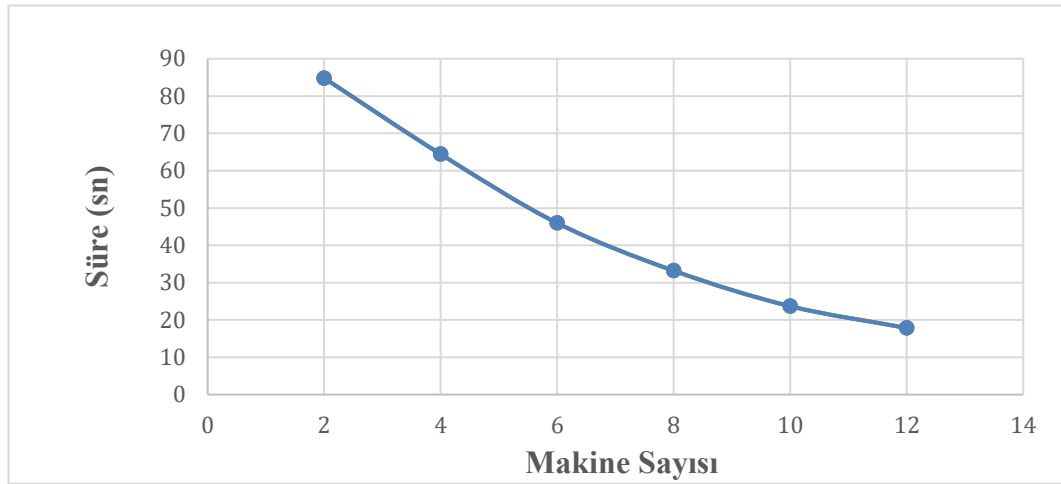
Önerilen algoritma kıyas veri setindeki 540 örnek için bir iterasyon çalıştırılmıştır. Yazılım platformu Pycharm 2022.1.2 seçilmiş belirlenen algoritmalar Python (3.7) dilinde kodlanmıştır. Çalışmanın çalıştırıldığı bilgisayarın konfigürasyonu (Intel(R) Core (TM) i7-6700 CPU @ 3.40GHz, 16GB Ram) şeklindedir. Bütün makine iş varyasyonlarına göre çalışma süreleri raporlanmıştır. Her bir makine ve iş sayısına göre ortalama süreler Tablo 17’ de verilmiştir.

Tablo 17. Bir İterasyon ile Algoritma Ortalama Süreleri

Makine	İş	Ortalama Süre(sn)	Makine	İş	Ortalama Süre(sn)
2	20	1,065	8	20	0,289
2	40	9,932	8	40	1,894
2	60	35,648	8	60	11,215
2	80	92,260	8	80	24,851
2	100	134,340	8	100	49,227
2	120	235,757	8	120	111,676
4	20	0,364	10	20	0,083
4	40	5,603	10	40	1,144
4	60	21,956	10	60	6,336
4	80	63,842	10	80	19,354
4	100	118,003	10	100	36,384
4	120	176,802	10	120	78,832
6	20	0,381	12	20	0,093
6	40	2,735	12	40	1,406
6	60	15,595	12	60	4,417
6	80	44,827	12	80	10,376
6	100	82,566	12	100	39,843
6	120	129,759	12	120	50,968

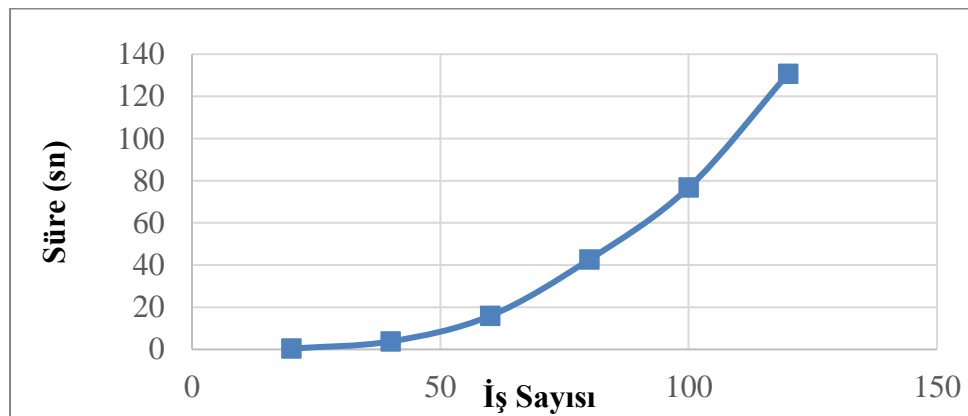
Tablo 17’de makine iş sayısına göre ortalama süreler algoritmanın beklediği gibi iş sayısı arttıkça sürenin de arttığını göstermektedir. 540 örnek bir bütün olarak ele alındığında kıyas veri setindeki tüm örnekler için toplam süre 24297,374 sn’dir. Bu süre 404,956 dk ve 6,75 saate eşittir. Değişken komşuluk algoritmasının komşuluk

değişiminde veya yerel arama aşamasında gelişme sağlanması durumunda komşuluk değişiminin başa alınması ilk iterasyonun daha uzun olmasına sebep olmaktadır. 24297,374 saniyelik toplam süre örnek sayısına (540) bölüldüğünde örnek başına çalışma süresi 44,995 sn olarak bulunur. Çalışma sürelerinin makine sayısına göre değişimi Şekil 9’da iş sayısına göre değişimi ise Şekil 10’da verilmiştir.



Şekil 9. Makine Sayısına Göre Çalışma Süreleri

Şekil 9 incelendiğinde, Tablo 16’da verilen algoritma karmaşıklığının süreler üzerinde etkisi olduğu görülmektedir. Makine sayısı arttıkça makine başına düşen iş sayısı arttıkça algoritma süresinin azaldığı görülmektedir.



Şekil 10. İş Sayısına Göre Algoritma Süreleri

Şekil 10 incelendiğinde, iş sayısının artışı problem büyüklüğünü arttırdığından algoritma sürelerinin iş sayısının artışına göre arttığı görülmektedir.

Algoritmanın iterasyon sayısına göre etkinliğinin ölçümü için, iterasyon sayısına göre algoritma süresi ve bilinen en iyi sonuca göre yüzdesel hata değeri hesaplanmıştır. Yüzdesel hata değeri eşitlik 13 ile ölçülmüştür.

$$\text{yüzdesel hata} = \frac{\text{algoritma sonucu} - \text{bilinen en iyi sonuç}}{\text{bilinen en iyi sonuç}} \times 100 \quad (13)$$

Önerilen algoritmanın daha etkin sonuçlar elde edebilmesi ve daha fazla test edilebilmesi için sürelerinin hızlandırılması gerekmektedir. Bu amaç doğrultusunda bir sunucuda (Intel(R) Xeon(R) CPU E5-4620 v2 @ 2.60GHz 2.60 GHz (2 işlemci)) paralel 10 işlemci ile asenkron bir şekilde algoritma çalıştırılmıştır. Önerilen algoritma paralel işlemler ile birden fazla işlemci ile aynı anda çalıştırılmış ve süreler kısaltılmıştır. Algoritma sonuçları iş ve makine sayısına göre değil, bir bütün olarak 540 test veri kümesinin toplam sürelerini değerlendirir. Örnek olarak 1 iterasyon için gerekli olan süre 404,956 dakikadan 85,111 dakikaya düşmüştür. Paralel işlemler ile 1,10,50 ve Eşitlik 12’de hesaplanan iterasyon sayısına göre önerilen algoritma çalıştırılmış ve sonuçları Tablo 18’de verilmiştir.

Tablo 18. İterasyon Sayılarına Göre Süre ve Hatalar

İterasyon sayısı	Süre (dk)	İterasyon başına süre (dk)	Yüzdesel hata
1 iterasyon	85,111	85,111	2,99
10 iterasyon	360	36	1,88
50 iterasyon	1157	23,14	1,4
Önerilen İterasyon Sayısı	1393,222	Değişken	1,83

Tablo 18 incelendiğinde iterasyon sayısının artışı ile yüzdesel hatanın düştüğü görülmektedir. Paralel işlemler ile birden fazla işin aynı anda test edilmesinin süreleri kısalttığı görülmektedir. Ayrıca iterasyon sayısı arttıkça iterasyon başına düşen sürenin de azaldığı görülmektedir. Bu DKA algoritmasının ilk iterasyonlarda çözümü daha fazla geliştirdiğinin ve komşuluk yapılarının baştan tekrar uygulandığının sonucudur. Tabloda göze çarpan bir detay da Eşitlik 12’de önerdiğimiz iterasyon sayısının süre ve yüzdesel hata açısından 50 iterasyonun denendiği durumdan daha kötü sonuçlar elde ettiğidir. Bu kötü sonuçlar hesapladığımız iterasyon sayısının bazı makine iş konfigürasyonlarında çok düşük kalmasındandır. Örnek olarak 20 işin 2 makinede çizelgeleme problemi için hesaplanan iterasyon sayısı eşitlik 12 ile 5 olarak bulunur. Hesaplanan iterasyon sayısı 50’den az olduğu durumlarda en az 50 olacak şekilde yenilenmiş ve Eşitlik 14’te belirtilmiştir.

$revize\ iterasyon\ sayisi = \max(50, N \times \frac{M}{m_{sayiOrt}})$	(14)
--	------

Eşitlik 14 sayesinde küçük boyutlu problemler de yeteri kadar çalıştırılmıştır. Eşitlik 14 ile revize edilen iterasyon sayısının sonuçları Tablo 19’da verilmiştir.

Tablo 19. Revize İterasyon Sayısına Göre Süre ve Hata

İterasyon sayısı	Süre (dk)	İterasyon başına süre (dk)	Yüzdesel hata
Revize İterasyon Sayısı	1584	Değişken	1,2145

3.4. Deneysel Sonuçlar

Bölüm 3.2.2’de açıklandığı üzere başlangıç çözümü açgözlü başlangıç çözüm algoritması ile oluşturulursa, çok daha kaliteli başlangıç çözümleri elde edilmektedir. Testlerde bu başlangıç çözümü kullanılmıştır. Ayrıca komşuluk yapılarının testleri ile Bölüm 3.3.1.3’teki sonuçlara göre komşuluk yapılarında önerilen parametrelerin kullanılması sonucu daha da iyileştirmiştir. DKA uygulanırken yerel arama seçim stratejisi ile yerel arama algoritmalarının etkin kullanılması hedeflenmiştir. Önerilen DKA arama algoritmaları büyük ve küçük veri setinde denenmiştir.

Çalışmada kullanılan küçük veri seti farklı sayıda işler {6, 7, 8, 9, 10 ve 11} farklı sayıda {2, 4, 6 ve 8} makinede oluştuğu 270 farklı veri dosyasından oluşmaktadır. Büyük veri seti ise {20, 40, 60, 80, 100 ve 120} için {2, 4, 6, 8, 10 ve 12} makinede işlenmesi üzerine hazırlanmış hazırlık ve işlem süresinin dengeli olduğu 540 adet veri dosyasından oluşmaktadır. Büyük veri setinde hazırlık süresinin baskın olduğu ve işlem süresinin baskın olduğu toplam 1080 veri dosyası daha mevcuttur. Bu çalışma hazırlık ve işlem süreci dengeli 540 büyük veri ile sınırlandırılmıştır.

Tezde değinilen problem daha önce birçok araştırmacının dikkatini çekmiş ve farklı çözümler geliştirilmiştir. Bu çalışmada önerilen çözümlerin etkinliğini ölçmek adına kıyaslama veri seti kullanılmış ve sonuçlar aynı kıyaslama verisini kullanan çalışmaların sonuçları ile kıyaslanmıştır.

Kullanılan kıyaslama veri seti Al-Salem (2004) tarafından geliştirilmiş ve daha sonra birçok araştırmacı tarafından kullanılmıştır. Kıyaslama veri setinin kullanıldığı çalışmalar Tablo 20’de verilmiştir. Veri seti, çalışma ve sonuçlarına www.schedulingresearch.com (Erişim Tarihi: 14.03.2022) üzerinden erişilebilir.

Tablo 20. Veri Seti ile İlgili Çalışmalar

Çalışma No	Çalışma	Kullanılan Algoritma	Kıyaslanan Çalışma
1	Al-Salem (2004)	PH (Bölümlendirme Sezgiseli)	-
2	Rabadi vd. (2006)	Meta-RaPS	1
3	Helal vd. (2006)	TA	1
4	Arnaout vd. (2010)	Karıncı Kolonisi Optimizasyon	1,2,3
5	Chang ve Chen (2011)	BT ve GA	1
6	Ying vd. (2012)	Kısıtlanmış BT	1,2,3,4
7	Fleszar vd. (2012)	Çok Başlangıçlı DKİ	2,4
8	Lin ve Ying (2014)	Yapay Arı Kolonisi- Hibrit Yapay Arı Kolonisi	1,2,3,4,5
9	Arnaout vd. (2014)	Karıncı Kolonisi Optimizasyon	1,2,3,4
10	Eroğlu vd. (2014b)	GA+ Yerel Arama	4,5
11	Cota vd. (2017)	Otomata- Uyarlamalı Geniş Komşuluk Arama	4,9
12	De Abreu, L. R., ve De Athayde (2020)	GA+DKİ+BT (GIVP)	4,5,10
13	Arnaout (2020)	Solucan Optimizasyon Algoritması	3,6,8,9,10

Tablo 20 görüleceği üzere tez kapsamında ilgilenilen problemin birçok araştırmacının dikkatini çeken bir problemidir. Bu problem araştırmacılar tarafından birçok metasezgisel algoritma ile çözülmüştür.

Öncelikle önerilen algoritmanın başarısı kıyas veri setindeki küçük boyutlu problemler ile test edilmiştir. Algoritma 50 kez çalıştırılmış ve en iyi sonuçlar raporlanmıştır. Bilinen en iyi sonuç koyu ve italik olarak belirtilmiştir. Test sonuçları Tablo 21’de görülmektedir.

Tablo 21. Küçük Veri Setinde Deneysel Sonuçlar

Makine	İş	MetaRaPs	TS	ACO	ACOr	GIVP	Önerilen DKA
2	6	394,73	396,40	394,73	394,73	394,73	394,73
2	7	491,00	495,07	491,00	491,00	491,00	491,00
2	8	517,40	522,60	517,40	517,40	518,13	517,40
2	9	598,87	603,80	598,47	598,47	598,47	598,47
2	10	638,93	645,33	638,93	638,93	639,20	638,93
2	11	710,73	721,27	710,73	710,73	710,73	710,40
4	6	245,00	251,73	245,00	245,00	245,00	245,00
4	7	252,27	265,07	252,27	252,27	252,93	252,27
4	8	264,73	271,27	264,73	264,73	264,73	264,73
4	9	346,87	346,40	346,07	346,07	346,07	346,07
4	10	359,47	361,60	359,53	359,47	359,47	359,47
4	11	366,33	374,33	366,47	366,33	367,53	366,33
6	8	234,47	242,07	234,47	234,47	233,87	234,47
6	9	238,53	249,53	238,53	238,53	238,00	238,53
6	10	246,00	259,47	246,47	246,00	226,13	245,93
6	11	251,27	274,53	251,60	251,27	232,67	250,80
8	10	226,13	232,00	226,13	226,13		226,13
8	11	232,47	235,60	232,47	232,47		232,47

Kaynak: De Abreu, L. R., ve De Athayde (2020:820)

Tablo 21 incelendiğinde küçük veri setinde DKA algoritmasının başarılı sonuçlar elde ettiği görülmektedir. Daha önce aynı veri setinde uygulanan algoritmalar ile kıyaslandığında bilinen en iyi sonuçların birçok örnekte yakalandığı 2 makine 11 iş

örneklerinde en iyi sonucu verdiği görülmektedir. Önerilen DKA algoritması ACOII ve daha önce önerilen bütün algoritmalarından daha başarılıdır. Sonuçlar 2020 yılında önerilen GIVP algoritması ile kıyaslandığında bazı örneklerde GIVP bazı örneklerde ise önerilen algoritmanın başarılı olduğu görülmektedir.

Metasezgisel algoritmalarda küçük boyutlu problemlerde hesaplanan $gap=(f-f_0)$ %of değerine göre, $gap<0,1$ 'den küçük ise kabul edilir (Jiang, Z. vd. 2021: 11). Önerilen DKA algoritması küçük boyutlu veri setinde optimal çözümlerin hepsini elde ettiği için başarılı olduğu söylenebilir.

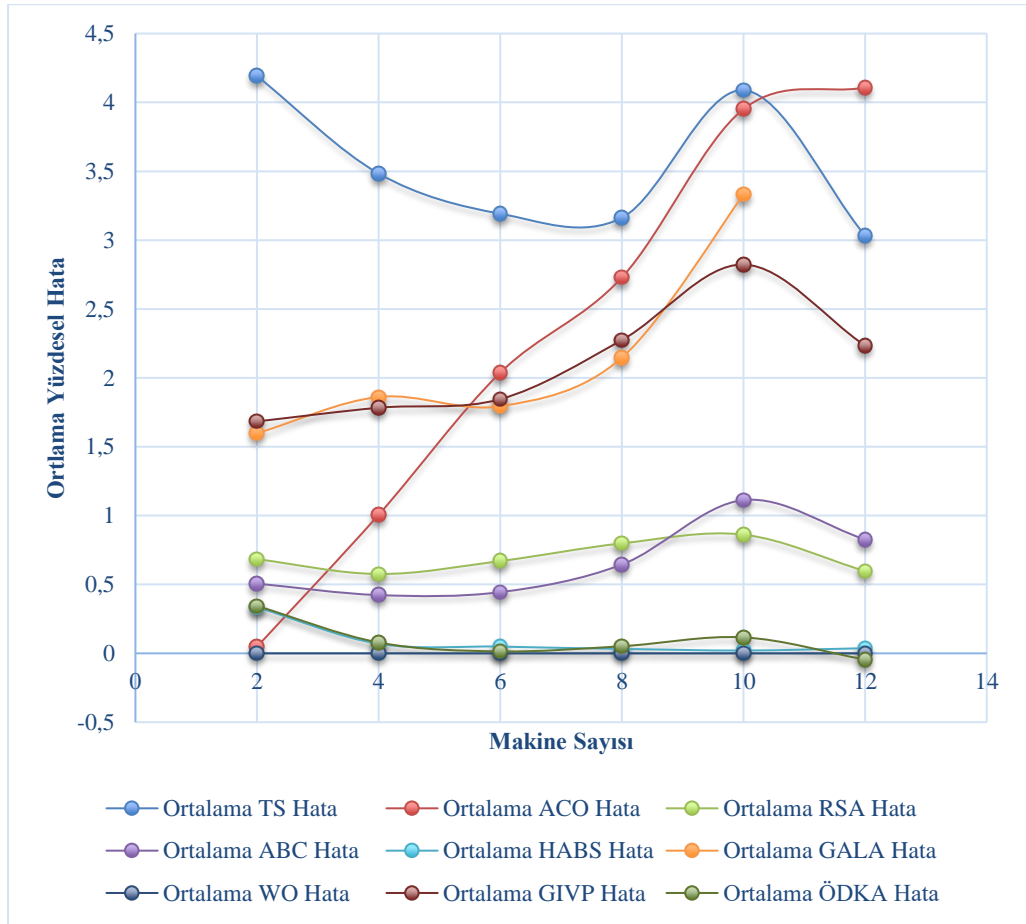
Küçük veri setindeki başarı ile algoritma daha önce belirlenen durdurma kriteri ile 50'kere çalıştırılmış ve en iyi sonuçlar raporlanmıştır. Tabloada bilinen en iyi sonuç koyu ve italik olarak belirtilmiştir. Test sonuçlarına Tablo 22'den ulaşılabilir.

Tablo 22. Büyük Veri Setinde Elde Edilen C_{max} Değerleri

M	İ	TS	ACO	RSA	ABC	HABS	GALA	WO	GIVP	Önerilen DKA
2	20	1264,87	1237,80	1238,53	1235,80	1234,87	1236,20	1234,87	1240,27	1234,87
2	40	2486,53	2397,80	2411,27	2405,33	2400,27	2425,87	2397,00	2436,80	2400,67
2	60	3736,47	3574,60	3598,87	3591,87	3583,80	3641,60	3573,93	3653,40	3584,47
2	80	4942,27	4730,40	4776,07	4764,13	4755,33	4834,00	4730,40	4844,60	4755,20
2	100	6180,87	5897,60	5937,20	5936,07	5924,20	6019,80	5897,60		5927,00
2	120	7447,60	7082,60	7144,40	7135,53	7126,27	7232,80	7082,60		7124,60
4	20	622,93	617,13	609,80	609,40	608,33	609,73	608,27	610,07	608,27
4	40	1200,67	1179,87	1165,13	1161,67	1158,40	1182,73	1157,73	1181,27	1159,20
4	60	1785,53	1737,93	1729,07	1725,20	1718,13	1757,00	1717,53	1759,80	1719,73
4	80	2370,13	2298,53	2302,53	2294,40	2286,87	2337,87	2285,73	2339,27	2287,33
4	100	2934,13	2849,93	2852,47	2850,60	2835,13	2892,53	2833,80		2836,80
4	120	3515,13	3405,13	3410,20	3413,40	3400,67	3465,13	3393,27		3394,80
6	20	449,40	452,73	446,13	446,00	445,87	445,93	445,87	446,20	445,87
6	40	803,73	805,40	783,67	782,93	779,40	790,87	778,20	794,73	778,67
6	60	1179,27	1163,47	1141,20	1137,67	1133,47	1162,40	1133,00	1162,13	1135,00
6	80	1568,60	1545,33	1527,93	1523,07	1514,13	1543,93	1513,33	1552,87	1513,07
6	100	1940,60	1897,47	1883,47	1881,33	1869,33	1910,13	1869,67		1869,87
6	120	2313,07	2253,93	2252,40	2241,60	2235,20	2286,13	2234,20		2230,80
8	20	342,80	347,60	339,73	339,87	339,47	340,13	339,47	342,07	339,47
8	40	588,67	599,27	577,60	578,53	572,67	589,27	572,47	585,47	574,07
8	60	893,13	893,80	874,80	873,93	867,93	882,67	867,73	888,33	867,20
8	80	1164,60	1142,40	1125,73	1122,80	1116,27	1150,20	1116,27	1150,07	1118,20
8	100	1449,27	1439,07	1414,40	1411,87	1403,73	1430,87	1403,07	1438,73	1401,93
8	120	1739,73	1686,07	1685,40	1675,73	1664,47	1713,07	1663,00	1707,33	1663,00
10	20	260,20	252,53	244,13	245,93	242,73	245,73	242,73	248,20	242,73
10	40	474,60	485,53	464,20	465,93	459,13	474,53	459,07	470,60	459,33
10	60	692,73	708,27	682,27	682,13	673,20	699,47	673,07	695,27	676,07
10	80	920,80	925,87	901,53	902,33	893,20	926,60	893,20	920,53	895,27
10	100	1153,27	1141,53	1115,20	1117,00	1107,67	1149,00	1107,67	1141,53	1109,13
10	120	1376,33	1351,67	1332,27	1334,33	1326,87	1378,60	1325,67	1362,20	1323,67
12	20	245,00	241,87	231,20	232,00	231,00		231,00	233,67	231,00
12	40	436,87	448,13	433,47	433,67	431,13		431,07	437,87	431,33
12	60	576,87	597,33	570,33	572,20	561,87		561,80	581,07	564,87
12	80	778,47	790,07	766,27	768,73	762,27		762,20	780,80	762,73
12	100	981,73	988,67	964,27	963,07	961,40		960,07	978,93	953,73
12	120	1146,40	1138,73	1110,27	1115,33	1105,87		1105,33	1136,67	1102,33

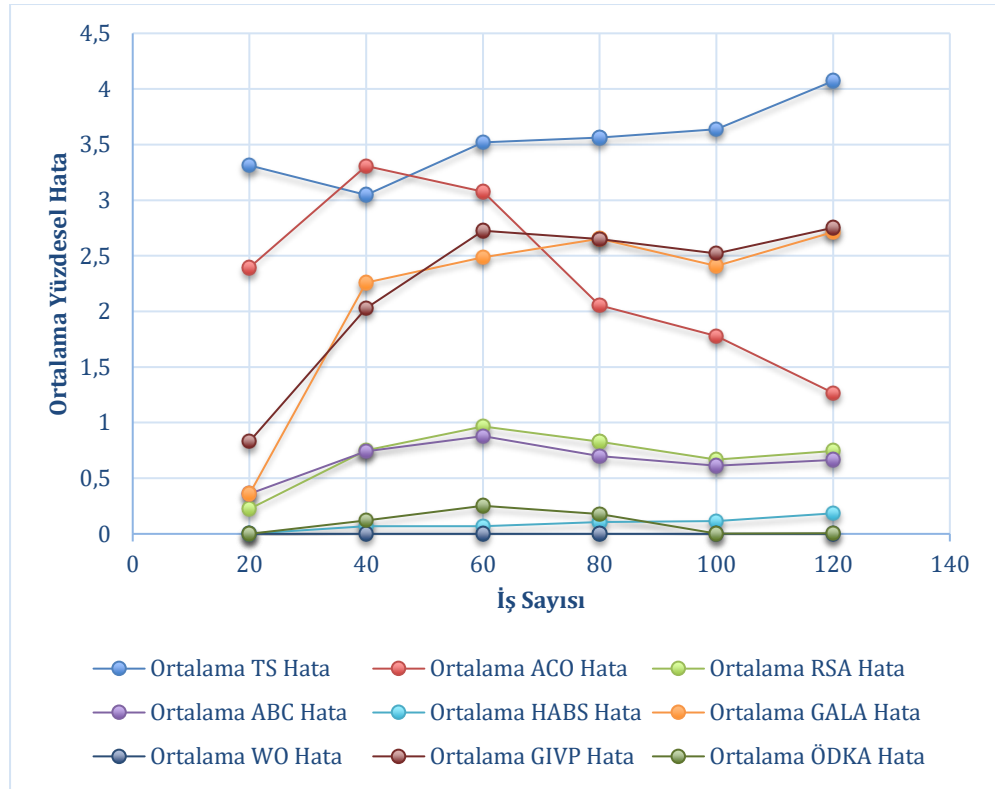
Kaynak: De Abreu, L. R., ve De Athayde (2020: 820) ve schedulingResearch.com

Arnout 2020'den alınan en iyi sonuca Önerilen DKA algoritması ile %0,95 hata oranı ile yaklaşılmıştır. Önerilen DKA algoritması 141 örnekte bilinen en iyi sonucu geçmiştir. 140 örnekte ise bilinen en iyi sonuç ile aynı C_{max} değerleri elde edilmiştir. Önerilen algoritmanın literatürde kıyaslandığı birçok algoritmadan daha başarılı olduğu söylenebilir. Tabloda 36 farklı makine iş konfigürasyonuna ait problemlerin (her konfigürasyon 15 farklı veri dosyası içerir) 13'ünde bilinen en iyi örneklerin önerilen DKA algoritmasına ait olduğu görülmektedir. Önerilen algoritma, diğer araştırmacılar tarafından daha karmaşık bir problem olarak ele alınan, makine sayısının arttığı durumlarda başarılı olmuştur. En karmaşık kabul edilen ve diğer araştırmacılar açısından en fazla çözüm süresi önerilen, 120 işin 12 makinede çizelgelenmesi ve 120 işin 10 makinede çizelgelenmesi gibi büyük boyutlu iki problemde diğer bütün algoritmalarından daha başarılıdır. Kıyaslanan algoritmaların büyük veri setindeki ortalama yüzdesel hataları hesaplanmıştır. Hesaplamaların makine sayısına göre grafiği Şekil 11'de, iş sayısına göre grafiği ise Şekil 12'de verilmiştir.



Şekil 11. Makine Sayısına Göre Algoritmaların Ortalama Yüzdesel Hataları

Şekil 11 incelendiğinde önerilen DKA algoritmasının en yüksek makine sayısında, kıyaslandığı algoritmalarından daha başarılı olduğu görülmektedir. Diğer makine sayılarında da bilinen en iyi algoritmalar ile benzer sonuçlar verdiği görülmektedir.



Şekil 12. İş Sayısına Göre Algoritmaların Ortalama Yüzdese Hataları

Şekil 12 incelendiğinde önerilen DKA algoritmasının iş sayısı arttıkça hata oranının düştüğü görülmektedir. Daha büyük boyutlu problemlerin çözümünde algoritmanın kullanılmasının uygun olduğu söylenebilir.

SONUÇ

Çizelgeleme hayatın her aşamasında karşılaşılan son derece önemli bir süreçtir. İşletmeler açısından bu süreci doğru bir şekilde planlamak sürdürülebilirlik açısından hayati öneme sahiptir. İşletmeler planlama aşamasında karşılaştığı problemleri, literatürde bulunan daha önce sonuçları kanıtlanmış kesin çözüm yöntemleri ile çözmeyi hedeflerler. Fakat bazı durumlarda, problemin büyüklüğüne göre bu problemi etkin şekilde çözmek mümkün değildir. Bu durumlarda makul bir süre içinde problemin yaklaşık çözümünün bulunması yeterlidir. Yaklaşık çözümleri elde etmek için birçok metasezgisel yöntem önerilmiştir. Metasezgisel yöntemler, kesin çözümler gibi yavaş değildir. Ayrıca metsezgisel yöntemler basit sezgisel yöntemler gibi yerel optimuma takılmazlar. Metasezgisellerde olası tüm çözümlerin aranması amaçlanır.

Bu çalışmada kıyas veri setindeki çizelgeleme problemini çözmek için tek çözümlü metasezgisel algoritmalarından DKA algoritması probleme göre uyarlanmış ve yeni başlangıç çözümleri, komşuluk yapıları ve yerel aramalar önerilmiştir.

Çalışmada ilgilenilen problem N adet işin, M adet makineye toplam en düşük maliyet ile atanması problemi. M adet makine ilişkisiz paralel makinedir ve her bir işin makinelerde işleme ve hazırlık süreleri farklıdır. Bu problemde, amaç fonksiyonu işlerinin tamamlanma sürelerinin minimize edilmesidir.

Çalışmada, yeni bir uyarlanmış değişken komşuluk arama algoritması önerilmektedir. Çalışmanın veri seti www.schedulingresearch.com internet sitesinden alınmıştır. Veri seti {2, 4, 6, 8, 10 ve 12} makinede {20, 40, 60, 80, 100 ve 120} işin çizelgelenmesi problemi. Her bir makine iş kombinasyonundan 15 adet toplam 540 adet veri dosyasından oluşmuştur. Çalışma kapsamı makinelerin işi işleme süreleri ve hazırlık süreleri makinelere göre dengeli olan veri seti ile sınırlandırılmıştır.

Çalışmada çizelgeleme problemini çözmek ve ilk başlangıç çözümünü elde etmek için iki farklı başlangıç algoritması önerilmiştir. Önerilen başlangıç algoritmalarından biri işler makineye atanırken rasgelelik içermektedir. Diğer önerilen algoritma makinelerin işi işleme süresi ve o işin makinelerdeki ortalama hazırlık sürelerini göz önüne almaktadır. Makinelere en az işlem ve hazırlık süresi ortalamasına sahip işleri atayan açgözlü algoritmanın testler sonucunda rasgele başlangıç algoritmasından çok daha iyi ilk çözümler elde ettiği bulunmuştur. Algoritma test edilirken açgözlü algoritma ile oluşturulan başlangıç çözümleri kullanmıştır.

Önerilen yeni algoritma değişken komşuluk arama algoritmasının 2 önemli aşaması olan komşuluk değişimi ve yerel arama aşamalarına yeni yaklaşımlar önermektedir. Bu bağlamda komşuluk değişimine iki yeni parametre önerilmiştir. Bu parametrelerden ilki, iş değişiminde seçilen makinelerden birinin, en maliyetli makine olmasını zorunlu tutmayı hedefleyen kritik makine seçim parametresidir. Diğer parametre ise makinelerdeki toplam iş yükünü oluşturan işlerden az iş yükü oluşturan işlerin makinede kalmasını zorlayan serbestlik parametresidir. Komşuluk yapılarına önerilen kritik makine seçimi ve serbestlik parametresi, aynı veri setinde 50'şer kez test ederek uygun parametrelerin belirlenmesi amaçlanmıştır. Bu amaç doğrultusunda veri seti için serbestlik parametresi ve kritik makine seçiminin farklı olduğu 4 farklı senaryo önerilmiş bu senaryoların performansları ölçülmüş ve raporlanmıştır.

Önerilen yeni algoritma, değişken komşuluk arama algoritmasının önemli aşaması olan yerel arama aşamasına 2 katmanlı bir yapı önermektedir. Her bir yapı kendi içinde yinelemeli birden fazla yerel arama algoritması barındırır. Komşuluk değişiminde elde edilen sonuçlara göre bu yerel arama aşamasındaki katmanlardan hangisine karar verileceği seçilir ve o aşama altında bulunan yerel aramalar sırası ile uygulanır. Yerel arama aşamasındaki katmanlı yapı komşuluk seçiminden doğan sonuçlara göre gereksiz yerel arama aşamalarını atlayarak daha etkin bir şekilde araştırma olanağı sağlar.

Kıyas veri setini çalışan araştırmacıların çalışma süreleri için farklı yaklaşımlar önermesi ile bu çalışma değişken komşuluk arama algoritmasının iterasyon sayısını esas alır. Bu çalışmada iterasyon sayısını hesaplamak için yeni bir formül oluşturulmuştur.

Çalışmada, kesin çözümü olmayan problemlerde yaklaşık çözümün elde edilmesi için yeni bir metasezgisel algoritma önerilmiştir. Önerilen algoritma ile yapılan deneyler sonuçlar sonucunda 540 adet veri setinin 141 tanesinde bilinen en iyi sonuçtan daha iyi değerler elde edilmiştir. 140 örnekte ise bilinen en iyi çözümlere ulaşılmıştır. Önerilen algoritma tüm örneklerde ortalama %0,095 hata ile bilinen en iyi sonuca yaklaşmıştır. En kötü örnekte dahi hata oranı bir metasezgisel algoritmanın başarılı olması için kabul edilen %1 hatadan daha düşüktür. Bundan dolayı önerilen algoritmanın başarılı olduğu söylenebilir.

Tez kapsamında kullanılan DKA ve DKİ algoritmaları hızlı ve birçok probleme kolay uygulanabilir, tek çözümlü metasezgisel algoritmalarıdır. Bu algoritmaların yapısında bulunan komşuluk değişimleri ve yerel arama aşamaları birçok hibrit

metasezgisel algoritmada başarılı bir şekilde kullanılmaktadır. Önerilen komşuluk değişimlerinin ve yeni yerel arama stratejilerinin SBHSİPMÇP'ni çözmekte başarılı olduğu ortadadır.

Bu çalışmadan referans ile diğer çizelgeleme problemlerine de bu tez kapsamında önerilen yöntemler uyarlanıp daha iyi sonuçlar elde edilebilir. Bu çalışmada geliştirilen algoritmaya yeni yerel aramalar eklenerek veya başka metasezgisel algoritmalar ile birleştirilip hibrit metasezgisel algoritmalar elde edilerek geliştirilebilir. İşletmeler önerilen algoritma ile çizelgeleme problemlerini çözerek mevcut çözümlerinden daha iyi sonuçlar elde edebilir ve rekabet ettikleri diğer işletmelerin önüne geçebilir.

KAYNAKLAR

- Afzalirad, M., ve Rezaeian, J. (2016). "Design of high-performing hybrid meta-heuristics for unrelated parallel machine scheduling with machine eligibility and precedence constraints", *Engineering Optimization*, 48(4), 706-726.
- Akpınar, Ş., (2016). "Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem", *Expert Syst. Appl.*, 61, 28-38.
- Allahverdi, A., ve Soroush, H. M. (2008). "The significance of reducing setup times/setup costs", *European Journal of Operational Research*, 187(3), 978-984.
- Al-Salem, A. (2004). "Scheduling to minimize makespan on unrelated parallel machines with sequence dependent setup times", *Engineering Journal of the University of Qatar*, 17.1, 177-187.
- Arnaut, J. P., Rabadi, G., ve Musa, R. (2010). "A two-stage ant colony optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times.", *Journal of Intelligent Manufacturing*, 21(6), 693-701.
- Arnaut, J. P. (2020). "A worm optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times.", *Annals of Operations Research*, 285(1), 273-293.
- Arnaut, J. P., Musa, R., ve Rabadi, G. (2014). "A two-stage Ant Colony optimization algorithm to minimize the makespan on unrelated parallel machines—part II: enhancements and experimentations.", *Journal of Intelligent Manufacturing*, 25(1), 43-53.
- Avalos-Rosales, O., Angel-Bello, F., ve Alvarez, A. (2015). "Efficient metaheuristic algorithm and re-formulations for the unrelated parallel machine scheduling problem with sequence and machine-dependent setup times.", *The International Journal of Advanced Manufacturing Technology*, 76(9-12), 1705-1718.
- Aytuğ, H., Khouja, M., ve Vergara, F. E. (2003). "Use of genetic algorithms to solve production and operations management problems: a review.", *International Journal of Production Research*, 41(17), 3955-4009.
- Azimpoor, S., ve Taghipour, S. (2019). "Optimal job scheduling and inspection of a machine with delayed failure.", *International Journal of Production Research*, 1-21.
- Báez, S., Angel-Bello, F., Alvarez, A., ve Melián-Batista, B. (2019). "A hybrid metaheuristic algorithm for a parallel machine scheduling problem with dependent setup times.", *Computers & Industrial Engineering*, 131, 295-305.
- Baker, K. R. (1999). "Heuristic procedures for scheduling job families with setups and due dates." *Naval Research Logistics (NRL)*, 46(8), 978-991.
- Baker, K. R., ve Magazine, M. J. (2000). "Minimizing maximum lateness with job families.", *European Journal of Operational Research*, 127(1), 126-139.
- Baker, K. R., ve Trietsch, D. (2019). *Principles of sequencing and scheduling*. (second edition) John Wiley & Sons.
- Behnamian, J., Zandieh, M., ve Ghomi, S. F. (2011). "Bi-objective parallel machines scheduling with sequence-dependent setup times using hybrid metaheuristics and weighted min-max technique.", *Soft Computing*, 15(7), 1313-1331.

- Bilge, Ü., Kıracı, F., Kurtulan, M., ve Pekkün, P. (2004). "A tabu search algorithm for parallel machine total tardiness problem.", *Computers & Operations Research*, 31(3), 397-414.
- Blum, C., ve Roli, A. (2003). "Metaheuristics in combinatorial optimization: Overview and conceptual comparison." *ACM computing surveys (CSUR)*, 35(3), 268-308.
- Brucker, P. (2007). *Scheduling algorithms* (Vol. 3, p. 1). Berlin: Springer.
- Canıyılmaz, E., Benli, B., ve Ilkay, M. S. (2015). "An artificial bee colony algorithm approach for unrelated parallel machine scheduling with processing set restrictions, job sequence-dependent setup times, and due date.", *The International Journal of Advanced Manufacturing Technology*, 77(9-12), 2105-2115.
- Cesaret, B., Oğuz, C., ve Salman, F. S. (2012). "A tabu search algorithm for order acceptance and scheduling.", *Computers & Operations Research*, 39(6), 1197-1205.
- Chase, I. D., Weissburg, M., ve Dewitt, T. H. (1988). "The vacancy chain process: a new mechanism of resource distribution in animals with application to hermit crabs.", *Animal behaviour*, 36(5), 1265-1274.
- Chang, P. C., ve Chen, S. H. (2011). "Integrating dominance properties with genetic algorithms for parallel machine scheduling problems with setup times.", *Applied Soft Computing*, 11(1), 1263-1274.
- Chen, C. L. (2012). "Iterated hybrid metaheuristic algorithms for unrelated parallel machines problem with unequal ready times and sequence-dependent setup times.", *The International Journal of Advanced Manufacturing Technology*, 60(5-8), 693-705.
- Chen, J. F. (2009b). "Scheduling on unrelated parallel machines with sequence-and machine-dependent setup times and due-date constraints.", *The International Journal of Advanced Manufacturing Technology*, 44(11-12), 1204-1212.
- Chen, W. J. (2009a). "Scheduling with dependent setups and maintenance in a textile company.", *Computers & Industrial Engineering*, 57(3), 867-873.
- Chivers, I., ve Sleightholme, J. (2015). "An introduction to Algorithms and the Big O Notation. In Introduction to programming with Fortran", *Springer*, Cham, 359-364
- Chyu, C. C., ve Chang, W. S. (2010). "A Pareto evolutionary algorithm approach to bi-objective unrelated parallel machine scheduling problems.", *The International Journal of Advanced Manufacturing Technology*, 49(5-8), 697-708.
- Chyu, C. C., ve Chang, W. S. (2011). "Optimizing fuzzy makespan and tardiness for unrelated parallel machine scheduling with archived metaheuristics.", *The International Journal of Advanced Manufacturing Technology*, 57(5-8), 763.
- Cota, L. P., Guimarães, F. G., de Oliveira, F. B., ve Souza, M. J. F. (2017, June). "An adaptive large neighborhood search with learning automata for the unrelated parallel machine scheduling problem.", In 2017 *IEEE Congress on Evolutionary Computation (CEC)* (185-192). IEEE.
- Cottet, F., Delacroix, J., Mammeri, Z., ve Kaiser, C. (2002). *Scheduling in real-time systems*. J. Wiley.

- De Abreu, L. R., ve de Athayde Prata, B. (2020). "A genetic algorithm with neighborhood search procedures for unrelated parallel machine scheduling problem with sequence-dependent setup times.", *Journal of Modelling in Management*, 15(3), 809-828.
- De CM Nogueira, J. P., Arroyo, J. E. C., Villadiego, H. M. M., ve Gonçalves, L. B. (2014). "Hybrid GRASP heuristics to solve an unrelated parallel machine scheduling problem with earliness and tardiness penalties.", *Electronic Notes in Theoretical Computer Sc*
- De Paula, M. R., Ravetti, M. G., Mateus, G. R., ve Pardalos, P. M. (2007). "Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search.", *IMA Journal of Management Mathematics*, 18(2), 101-115.
- Dewan, P., ve Joshi, S. (2000). "Dynamic single-machine scheduling under distributed decision-making.", *International Journal of Production Research*, 38(16), 3759-3777.
- Diana, R. O. M., ve de Souza, S. R. (2020). "Analysis of variable neighborhood descent as a local search operator for total weighted tardiness problem on unrelated parallel machines.", *Computers & Operations Research*, 117, 104886.
- Diana, R. O., de Souza, S. R., ve Moacir Filho, F. F. (2018). "A Variable Neighborhood Descent as ILS local search to the minimization of the total weighted tardiness on unrelated parallel machines and sequence dependent setup times.", *Electronic Notes in Discrete Mathematics*, 66, 191-198.
- Dominic, P. D., Kaliyamoorthy, S., ve Kumar, M. S. (2004). "Efficient dispatching rules for dynamic job shop scheduling.", *The International Journal of Advanced Manufacturing Technology*, 24(1-2), 70-75.
- Dorigo, M., Birattari, M., ve Stutzle, T. (2006). "Ant colony optimization.", *IEEE computational intelligence magazine*, 1(4), 28-39.
- Driessel, R., ve Mönch, L. (2011). "Variable neighborhood search approaches for scheduling jobs on parallel machines with sequence-dependent setup times, precedence constraints, and ready times.", *Computers & industrial engineering*, 61(2), 336-345.
- Eiben, A. E., ve Smith, J. E. (2015). *Introduction to evolutionary computing*. Springer.
- Eren, T., ve Güner, E. (2002). "Tek ve Paralel Makinalı Problemlerde Çok Ölçütlü Çizelgeleme Problemleri İçin Bir Literatür Taraması." *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 17(4).
- Eroğlu, D. Y., Özmutlu, H. C., ve Köksal, S. A. (2014a). "A Genetic Algorithm For The Unrelated Parallel Machine Scheduling Problem With Job Splitting And Sequence-Dependent Setup Times-Loom Scheduling." *Tekstil ve Konfeksiyon*, 24(1), 66-73.
- Eroğlu, D. Y., Ozmutlu, H. C., ve Ozmutlu, S. (2014b). "Genetic algorithm with local search for the unrelated parallel machine scheduling problem with sequence-dependent set-up times. *International Journal of Production Research*, 52(19), 5841-5856.

- Figielska, E. (2013). "An Ant Colony Optimization Algorithm for Scheduling Parallel Machines with Sequence-Dependent Setup Costs.", *Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki*, (9), 15-26.
- Fleszar, K., Charalambous, C., ve Hindi, K. S. (2012). "A variable neighborhood descent heuristic for the problem of makespan minimisation on unrelated parallel machines with setup times.", *Journal of Intelligent Manufacturing*, 23(5), 1949-1958.
- Forrest, S. (1996). "Genetic algorithms". *ACM Computing Surveys (CSUR)*, 28(1), 77-80.
- Framinan, J. M. (2007). "An adaptive branch and bound approach for transforming job shops into flow shops.", *Computers & Industrial Engineering*, 52(1), 1-10.,
- Gandomi, A. H., Emrouznejad, A., ve Rahimi, I. (2020). "Evolutionary Computation in Scheduling: A Scientometric Analysis.", *Evolutionary Computation in Scheduling*, 1-10.
- Garey, M. R. and Johnson, D. S., (1978). "Computers and intractability: a guide to the theory of NP-completeness". New York, Freeman.
- Gedik, R., Kalathia, D., Egilmez, G., ve Kirac, E. (2018). "A constraint programming approach for solving unrelated parallel machine scheduling problem.", *Computers & Industrial Engineering*, 121, 139-149.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., ve Kan, A. R. (1979). "Optimization and approximation in deterministic sequencing and scheduling: a survey.", *Annals of discrete mathematics*, 5, 287-326. Elsevier.
- Gupta, A. K., ve Sivakumar, A. I. (2005). "Multi-objective scheduling of two-job families on a single machine.", *Omega*, 33(5), 399-405.
- Haddad, M. N., Cota, L. P., Souza, M. J. F., ve Maculan, N. (2014, April). "AIV: A Heuristic Algorithm based on Iterated Local Search and Variable Neighborhood Descent for Solving the Unrelated Parallel Machine Scheduling Problem with Setup Times.", *In ICEIS*, 1, 376-383.
- Hansen, P., Mladenovic, N. (2002). "Developments of Variable Neighborhood Search. In: Essays and Surveys in Metaheuristics." *Operations Research/Computer Science Interfaces Series*, vol 15. Springer, Boston, MA
- Hansen, P., Mladenovic, N., Todosijeovic, R. and Hanafi, S., (2017). "Variable neighborhood search: basics and variants", *EURO J. Comput. Optim.*, 5, 423-454.
- Hansen, P., Mladenovic, N. and Moreno Pérez, J. A., (2008). "Variable neighborhood search", *Eur J Oper Res.*, 191 (3), 593-595.
- Hansen, P., Mladenovic, N. and Perez, J. A. M., (2010). "Variable neighbourhood search: methods and applications", *4OR*, 6, 319-360.
- Hansen, P., ve Mladenovic, N. (2003). "A tutorial on variable neighborhood search.", *Les Cahiers du GERAD* ISSN, 711, 2440.
- Hansen, P. and Mladenovic, N., (2016). "Variable Neighborhood Search", *Handbook of Heuristics*, Springer, Boston
- He, L., Guijt, A., de Weerd, M., Xing, L., ve Yorke-Smith, N. (2019). "Order acceptance and scheduling with sequence-dependent setup times: A new memetic algorithm

- and benchmark of the state of the art.”, *Computers & Industrial Engineering*, 138, 106102.
- Helal, M., Rabadi, G., ve Al-Salem, A. (2006). “A tabu search algorithm to minimize the makespan for the unrelated parallel machines scheduling problem with setup times.”, *International Journal of Operations Research*, 3(3), 182-192.
- Jourdan, L., Basseur, M., ve Talbi, E. G. (2009). “Hybridizing exact methods and metaheuristics: A taxonomy.”, *European Journal of Operational Research*, 199(3), 620-629.
- Jiang, Z., Chen, Y., Li, X., ve Li, B. (2021). “A heuristic optimization approach for multi-vehicle and one-cargo green transportation scheduling in shipbuilding.”, *Advanced Engineering Informatics*, 49, 101306.
- Kaplan, S., ve Rabadi, G. (2012). “Exact and heuristic algorithms for the aerial refueling parallel machine scheduling problem with due date-to-deadline window and ready times.”, *Computers & Industrial Engineering*, 62(1), 276-285.
- Kaya, S. (2014). “Çok amaçlı esnek atölye tipi çizelgeleme problemlerinin geliştirilmiş parçacık sürü optimizasyonu ile çözümüne yönelik model önerileri”, Doktora Tezi, Kocaeli Üniversitesi.
- Kerkhove, L. P., ve Vanhoucke, M. (2014). “Scheduling of unrelated parallel machines with limited server availability on multiple production locations: a case study in knitted fabrics.”, *International Journal of Production Research*, 52(9), 2630-2653.
- Kim, D. W., Kim, K. H., Jang, W., ve Chen, F. F. (2002). “Unrelated parallel machine scheduling with setup times using simulated annealing.”, *Robotics and Computer-Integrated Manufacturing*, 18(3-4), 223-231.
- Kim, J., ve Kim, H. J. (2020). “Parallel machine scheduling with multiple processing alternatives and sequence-dependent setup times.”, *International Journal of Production Research*, 1-16.
- Kim, J. G., Song, S., ve Jeong, B. (2020). “Minimising total tardiness for the identical parallel machine scheduling problem with splitting jobs and sequence-dependent setup times.”, *International Journal of Production Research*, 58(6), 1628-1643.
- Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., (1983). “Optimization by simulated annealing”, *Science*, 220 (4598), 671-680
- Lamothe, J., Marmier, F., Dupuy, M., Gaborit, P., ve Dupont, L. (2012). “Scheduling rules to minimize total tardiness in a parallel machine problem with setup and calendar constraints.”, *Computers & Operations Research*, 39(6), 1236-1244.
- Lee, C. H. (2018). “A dispatching rule and a random iterated greedy metaheuristic for identical parallel machine scheduling to minimize total tardiness.”, *International Journal of Production Research*, 56(6), 2292-2308.
- Lee, C. Y., Lei, L., ve Pinedo, M. (1997). “Current trends in deterministic scheduling.”, *Annals of Operations Research*, 70, 1-41.
- Lee, J. H., ve Jang, H. (2019). “Uniform parallel machine scheduling with dedicated machines, job splitting and setup resources.”, *Sustainability*, 11(24), 7137.

- Li, X., Yalaoui, F., Amodeo, L., ve Chehade, H. (2012). "Metaheuristics and exact methods to solve a multiobjective parallel machines scheduling problem.", *Journal of Intelligent Manufacturing*, 23(4), 1179-1194.
- Liao, C. J., Lee, C. H., ve Tsai, H. T. (2016). "Scheduling with multi-attribute set-up times on unrelated parallel machines.", *International Journal of Production Research*, 54(16), 4839-4853.
- Lin, C. W., Lin, Y. K., ve Hsieh, H. T. (2013). "Ant colony optimization for unrelated parallel machine scheduling.", *The International Journal of Advanced Manufacturing Technology*, 67(1-4), 35-45.
- Lin, S. W., Lu, C. C., ve Ying, K. C. (2011)(a). "Minimization of total tardiness on unrelated parallel machines with sequence-and machine-dependent setup times under due date constraints.", *The International Journal of Advanced Manufacturing Technology*, 53(1-4), 353-361.
- Lin, S. W., Lee, Z. J., Ying, K. C., ve Lu, C. C. (2011)(b). "Minimization of maximum lateness on parallel machines with sequence-dependent setup times and job release dates.", *Computers & Operations Research*, 38(5), 809-815.
- Lin, S. W., ve Ying, K. C. (2014). "ABC-based manufacturing scheduling for unrelated parallel machines with machine-dependent and job sequence-dependent setup times." *Computers & Operations Research*, 51, 172-181.
- Lin, S. W., ve Ying, K. C. (2015). "A multi-point simulated annealing heuristic for solving multiple objective unrelated parallel machine scheduling problems.", *International Journal of Production Research*, 53(4), 1065-1076.
- Lin, Y. K., ve Hsieh, F. Y. (2014). "Unrelated parallel machine scheduling with setup times and ready times.", *International Journal of Production Research*, 52(4), 1200-1214.
- Lin, Y. K., Pfund, M. E., ve Fowler, J. W. (2011) (c). "Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems.", *Computers & Operations Research*, 38(6), 901-916.
- Ma, Y., Chu, C., ve Zuo, C. (2010). "A survey of scheduling with deterministic machine availability constraints.", *Computers & Industrial Engineering*, 58(2), 199-211.
- Manupati, V. K., Rajyalakshmi, G., Chan, F. T., ve Thakkar, J. J. (2017). "A hybrid multi-objective evolutionary algorithm approach for handling sequence-and machine-dependent set-up times in unrelated parallel machine scheduling problem.", *Sādhanā*, 42(3), 391-403.
- Mladenović, N., ve Hansen, P. (1997). "Variable neighborhood search.", *Computers & operations research*, 24(11), 1097-1100.
- Mönch, L., Balasubramanian, H., Fowler, J. W., ve Pfund, M. E. (2005). "Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times.", *Computers & Operations Research*, 32(11), 2731-2750.
- Murali, G. B., Biswal, B. B., Deepak, B. B. V. L., Rout, A., ve Mohanta, G. B. (2019, March). "A New Crab Shell Search Algorithm for Optimal Assembly Sequence Generation." In 2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON) (108-114). IEEE.

- Niu, Q., Zhou, T., ve Wang, L. (2010). "A hybrid particle swarm optimization for parallel machine total tardiness scheduling.", *The International Journal of Advanced Manufacturing Technology*, 49(5-8), 723-739.
- Onar, S. Ç., Öztayşi, B., Kahraman, C., Yanık, S., ve Şenvar, Ö. (2016). "A literature survey on metaheuristics in production systems. In *Metaheuristics for Production Systems*" (1-24). Springer, Cham.
- Osman, I. H., ve Laporte, G. (1996). *Metaheuristics: A bibliography*.
- Park, Y., Kim, S., ve Lee, Y. H. (2000). "Scheduling jobs on parallel machines applying neural network and heuristic rules.", *Computers & Industrial Engineering*, 38(1), 189-202.
- Perez, I. C., Fowler, J. W., ve Carlyle, W. M. (2005). "Minimizing total weighted tardiness on a single batch process machine with incompatible job families.", *Computers & Operations Research*, 32(2), 327-341.
- Pinedo, M. L (2018). *Scheduling: theory, algorithms, and systems*. Springer.
- Puchinger, J., ve Raidl, G. R. (2005, June). "Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification.", In *International work-conference on the interplay between natural and artificial computation* (41-53). Springer, Berlin, Heidelberg.
- Rabadi, G., Moraga, R. J., ve Al-Salem, A. (2006). "Heuristics for the unrelated parallel machine scheduling problem with setup times.", *Journal of Intelligent Manufacturing*, 17(1), 85-97.
- Rajakumar, S., Arunachalam, V. P., ve Selladurai, V. (2004). "Workflow balancing strategies in parallel machine scheduling.", *The International Journal of Advanced Manufacturing Technology*, 23(5-6), 366-374.
- Rotjan, R. D., Chabot, J. R., ve Lewis, S. M. (2010). "Social context of shell acquisition in *Coenobita clypeatus* hermit crabs.", *Behavioral Ecology*, 21(3), 639-646.
- Sabouni, M. Y., Jolai, F., ve Mansouri, A. (2010). "Heuristics for minimizing total completion time and maximum lateness on identical parallel machines with setup times.", *Journal of Intelligent Manufacturing*, 21(4), 439-449.
- Shim, S. O., ve Kim, Y. D. (2007). "Scheduling on parallel identical machines to minimize total tardiness.", *European Journal of Operational Research*, 177(1), 135-146.
- Soleimani, H., Ghaderi, H., Tsai, P. W., Zarbakhshnia, N., ve Maleki, M. (2020). "Scheduling of unrelated parallel machines considering sequence-related setup time, start time-dependent deterioration, position-dependent learning and power consumption minimization.", *Journal of Cleaner Production*, 249, 119428.
- Sule, D. R. (2007). *Production planning and industrial scheduling: examples, case studies and applications*. CRC press.
- Talbi, E. G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.
- Tanyaş, M., ve Baskak, M. (2003). *Üretim planlama ve kontrol*.
- Tran, T. T., ve Beck, J. C. (2012). "Logic-based Benders Decomposition for Alternative Resource Scheduling with Sequence Dependent Setups.", In *ECAI* (774-779).

- Tsai, C. W., Huang, W. C., Chiang, M. H., Chiang, M. C., ve Yang, C. S. (2014). "A hyper-heuristic scheduling algorithm for cloud.", *IEEE Transactions on Cloud Computing*, 2(2), 236-250.
- Vallada, E., ve Ruiz, R. (2011). "A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times.", *European Journal of Operational Research*, 211(3), 612-622.
- Wang, L., Wang, S., ve Zheng, X. (2016). "A hybrid estimation of distribution algorithm for unrelated parallel machine scheduling with sequence-dependent setup times.", *IEEE/CAA Journal of Automatica Sinica*, 3(3), 235-246.
- Widmer, M., Hertz, A., ve Costa, D. (2008). "Metaheuristics and scheduling.", *Production Scheduling*, 33-68
- Khafa, F., ve Abraham, A. (Eds.). (2008). *Metaheuristics for scheduling in industrial and manufacturing applications* (Vol. 128). Springer.
- Ying, K. C., ve Cheng, H. M. (2010). "Dynamic parallel machine scheduling with sequence-dependent setup times using an iterated greedy heuristic.", *Expert Systems with Applications*, 37(4), 2848-2852.
- Ying, K. C., ve Lin, S. W. (2012). "Unrelated parallel machine scheduling with sequence- and machine-dependent setup times and due date constraints.", *International Journal of Innovative Computing, Information and Control*, 8(5), 3279-3297.
- Ying, K. C., Lee, Z. J., ve Lin, S. W. (2012). "Makespan minimization for scheduling unrelated parallel machines with setup times.", *Journal of Intelligent Manufacturing*, 23(5), 1795-1803.
- Zarandi, M. H. F., Asl, A. A. S., Sotudian, S., ve Castillo, O. (2020). "A state of the art review of intelligent scheduling.", *Artificial Intelligence Review*, 53(1), 501-593.
- Zhu, Z., ve Heady, R. B. (2000). "Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach.", *Computers & Industrial Engineering*, 38(2), 297-305.
- Zupan, H., Herakovic, N., Starbek, M., ve Kusar, J. (2016). "Hybrid algorithm based on priority rules for simulation of workshop production.", *International Journal of Simulation Modelling*, 15(1), 29-41.