

**T.C.
ERCIYES ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
ÜRETİM YÖNETİMİ VE PAZARLAMA BİLİM DALI**

**AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN
ÇÖZÜMÜNDE METASEZGİSEL YAKLAŞIMLARIN
KARŞILAŞTIRMALI OLARAK ANALİZİ**

**Hazırlayan
Murat RUHLUSARAÇ**

**Danışman
Doç. Dr. Mehmet S. İLKAY**

Yüksek Lisans Tezi

**Haziran 2012
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
İŞLETME ANABİLİM DALI
ÜRETİM YÖNETİMİ VE PAZARLAMA BİLİM DALI**

**AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN
ÇÖZÜMÜNDE METASEZGİSEL YAKLAŞIMLARIN
KARŞILAŞTIRMALI OLARAK ANALİZİ
(Yüksek Lisans Tezi)**

**Hazırlayan
Murat RUHLUSARAÇ**

**Danışman
Doç. Dr. Mehmet S. İLKAY**

**Bu çalışma; Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi tarafından
SBY-11-3803 kodlu proje ile desteklenmiştir.**

**Haziran 2012
KAYSERİ**

Bu alıřmadaki tm bilgilerin, akademik ve etik kurallara uygun bir řekilde elde edildiđini beyan ederim. Aynı zamanda bu kural ve davranıřların gerektirdiđi gibi bu alıřmanın znde olmayan tm materyal ve sonuları tam olarak aktardıđımı ve referans gsterdiđimi belirtirim.

Murat RUHLUSARA

Akış Tipi Çizelgeleme Problemlerinin Çözümünde Metasezgisel Yaklaşımların Karşılaştırmalı Olarak Analizi adlı Yüksek Lisans Tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi'ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan
Murat RUHLUSARAÇ

Danışman
Doç. Dr. Mehmet S. İLKAY

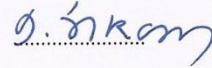
İşletme ABD Başkanı
Prof. Dr. Osman UNUTULMAZ

Doç. Dr. Mehmet S. İLKAY danışmanlığında Murat RUHLUSARAÇ tarafından hazırlanan 'Akış Tipi Çizelgeleme Problemlerinin Çözümünde Metasezgisel Yaklaşımların Karşılaştırmalı Olarak Analizi' adlı bu çalışma jürimiz tarafından Erciyes Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalında **yüksek lisans** tezi olarak kabul edilmiştir.

25/06/2012

JÜRİ:

Danışman : Doç. Dr. Mehmet S. İLKAY



Üye : Prof. Dr. Osman UNUTULMAZ



Üye : Yrd. Doç. Dr. Erdal CANIYILMAZ

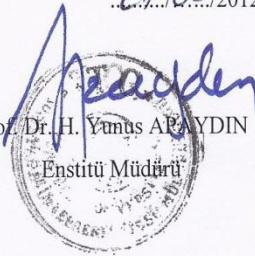
**ONAY:**

Bu tezin kabulü Enstitü Yönetim Kurulunun 27.06.2012 tarih ve 12 sayılı kararı ile onaylanmıştır.

27/06/2012

Prof. Dr. H. Yunus ARAYDIN

Enstitü Müdürü



ÖNSÖZ

Dünyamızın her geçen gün artan rekabet ortamında üretim yapan işletmeler, verimlilik ve etkinlik kavramları çerçevesinde üretim kaynaklarını doğru yerde ve zamanda planlı bir şekilde sarf etmelidirler. Bunun için de; kendi üretim ortamlarına en uygun şekilde, işletme yöneticilerinin amaçları doğrultusunda, sahip oldukları kaynakların sınırları çerçevesinde ve müşteri siparişlerinin özelliklerini içeren üretim çizelgeleri hazırlamaları gerekmektedir. Bu etkin çizelgeler sayesinde kârlılık, devamlılık ve müşteri memnuniyeti sağlayacaklardır.

Bu tez çalışmasında bu amaçlar güdülerek akış tipi bir test problemi ele alınmış çeşitli yöntemlerle optimal çizelgesi (iş akış sıralaması) en kısa sürede elde edilmeye çalışılmıştır. Yöntemlerin sonuçları da karşılaştırılarak performansları ölçülmüştür.

Bu tezin hazırlanma sürecinde, öncelikle tez danışmanlığımı üstlenen, gösterdiği sabır ve güvenle beni teşvik eden, tecrübeleriyle katkı sağlayan ve desteğini esirgemeyen Doç. Dr. Mehmet S. İlkay’a teşekkür ederim. Değerli jüri üyelerime ve kuzenim Ahmet Güneş’e katkı ve eleştirileri için teşekkür ederim. Tezimin uygulama bölümünde algoritmalar konusunda bana yardımcı olan ve içtenlikle destek veren mühendislik fakültemiz öğretim üye ve elemanlarından Yrd. Doç. Dr. Erdal Canıyılmaz, Arş. Gör. Fehim Köylü ve Arş. Gör. Betül Benli’ye ayrı ayrı teşekkürlerimi sunarım. Ayrıca tezimi destekleyerek katkıda bulunan Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimine de teşekkürü borç bilirim. Son olarak bu uzun ve zaman zaman zorlu olan süreçte, olumsuzluğa kapıldığım zamanlar olsa da beni hep destekleyen, duydukları güven ve gösterdikleri fedakârlık ile hep yanımda olan, beni yetiştiren ve bugünlerde olmamı sağlayan aileme en derin duygularla teşekkür ederim.

Murat RUHLUSARAÇ

Kayseri, Haziran 2012

AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNİN ÇÖZÜMÜNDE METASEZGİSEL YAKLAŞIMLARIN KARŞILAŞTIRMALI OLARAK ANALİZİ

Murat RUHLUSARAÇ

Erciyes Üniversitesi, Sosyal Bilimler Enstitüsü

Yüksek Lisans Tezi, Haziran 2012

Danışman: Doç. Dr. Mehmet S. İLKAY

KISA ÖZET

Üretim çizelgeleme problemleri araştırmacılar tarafından yoğun şekilde ilgi gören ama son derece karmaşık yapılabilen optimizasyon problemleridir. Bu problemlerin bir türü olan akış tipi çizelgeleme problemleri (FSP) ise, $m \geq 3$ olduğu durumlarda NP-tam zorluk düzeyinde bulunmaktadır. Bu zorluk düzeyinde olması nedeniyle akış tipi çizelgeleme problemlerinin çözümü için çok sayıda sezgisel yöntem geliştirilmiştir. Son zamanlarda ise bilgisayar sistemlerinin gelişmesi ve doğadaki sürü davranışlarından esinlenilmesi sonucu yapay zekâ teknolojisi hızla ilerlemiş ve bu gelişmeler ve yeni fikirlerle literatüre, kombinatoriyal optimizasyon problemlerine daha hızlı ve daha kaliteli çözümler veren metasezgisel yaklaşımlar sunulmuştur.

Bu çalışmada, bir üretim çizelgeleme problemi olan 5 makine ve 23 işe sahip akış tipi bir çizelgeleme test problemi bazı metasezgiseller ile çözülmüştür. Problem için çeşitli parametreler kullanılarak elde edilen çizelgeler ile işlerin maksimum tamamlanma süreleri ve hesaplanan CPU işlem zamanları tablolar halinde verilmiştir. Sonuç olarak, tüm yöntemlerin çözüm sonuçları karşılaştırmalı olarak analiz edilmiş ve ABC algoritmasının daha küçük parametrede ve daha kısa sürede optimal çizelgeyi sunduğu tespit edilmiştir.

Anahtar Kelimeler: Üretim Çizelgeleme, Akış Tipi Çizelgeleme Problemleri, Metasezgisel Yaklaşımlar

**A COMPARATIVE ANALYSIS of METAHEURISTIC APPROACHES
DEVELOPED for SOLVING FLOWSHOP SCHEDULING PROBLEMS**

Murat RUHLUSARAÇ

Erciyes University, Institute of Social Sciences

Master's Thesis, June 2012

Supervisor: Doç. Dr. Mehmet S. İLKAY

ABSTRACT

Researchers are extensively interested in production scheduling problems but also they may be extremely complex optimization problems. The flow shop scheduling problems (FSP) are considered NP-complete level of difficulty scheduling problems when $m \geq 3$. Because of this difficulty level, a large number of scheduling heuristics have been developed to solve FSPs. Recently, as a result of the development in computer systems and the inspiration by the swarm behaviors in nature, artificial intelligence technology has progressed rapidly. These new ideas and developments encouraged the researchers to develop some metaheuristic approaches which presented faster and better solutions methods for solving combinatorial optimization problems in the literature.

In this study, the test production scheduling problem, $m = 5$ and $n = 23$, which is a flow type is solved employing some metaheuristics. Calculated completion and CPU times and the schedules which are obtained by using various parameters are presented in tabular forms. Finally, results of the methods that are to be compared analyzed and pointed out that the ABC algorithm provided optimal schedule with smaller parameter and shortest time.

Keywords: Production Scheduling, Flowshop Scheduling Problems, Metaheuristic Approaches

İÇİNDEKİLER

	<u>Sayfa</u>
BİLİMSEL ETİĞE UYGUNLUK	i
YÖNERGEYE UYGUNLUK.....	ii
KABUL VE ONAY SAYFASI	iii
ÖNSÖZ	iv
KISA ÖZET	v
ABSTRACT.....	vi
İÇİNDEKİLER	vii
TABLolar LİSTESİ.....	ix
ŞEKİLLER LİSTESİ	x
KISALTMALAR VE SİMGELER.....	xi
GİRİŞ	1

BİRİNCİ BÖLÜM LİTERATÜR DEĞERLENDİRMESİ

1.1. GENETİK ALGORİTMA	4
1.2. BAĞIŞIKLIK SİSTEMLERİ	8
1.3. SİNİR AĞLARI.....	9
1.4. DOYUMSUZ ALGORİTMA.....	9
1.5. KARINCA KOLONİLERİ.....	10
1.6. TAVLAMA BENZETİMİ.....	12
1.7. TABU ARAMASI	13
1.8. PARÇACIK SÜRÜLERİ.....	14
1.9. YAPAY ARI KOLONİLERİ.....	14
1.10. DİFERANSİYEL GELİŞİM.....	15
1.11. NEH SEZGİSELİ	16
1.12. DİĞER MATEMATİKSEL MODELLER VE SEZGİSELLER.....	18

İKİNCİ BÖLÜM ÇİZELGELEME

2.1. ÇİZELGELEMENİN TANIMI VE AMACI.....	24
2.1.1. Çizelgelemenin Tanımı.....	25
2.1.2. Çizelgelemenin Amacı	25
2.2. ÜRETİM ÇİZELGELEME.....	26

2.2.1.	Üretim Çizelgelemenin Tanımı Ve Amacı	26
2.2.2.	Üretim Çizelgeleme Tipleri	29
2.3.	ÜRETİM ÇİZELGELEME PROBLEMLERİ	30
2.3.1.	Üretim Çizelgeleme Problemlerinin Sınıflandırılması	30
2.3.2.	Üretim Çizelgeleme Problemlerinin Parametreleri	32
2.3.3.	Üretim Çizelgeleme Problemlerinde Öncelik Kuralları	35
2.3.4.	Üretim Çizelgeleme Problemlerinin Performans Ölçüm Kriterleri	37

ÜÇÜNCÜ BÖLÜM

AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ

3.1.	PROBLEMİN TANIMI VE ÖZELLİKLERİ	40
3.2.	PROBLEMİN ÇEŞİTLERİ	43
3.2.1.	Permütasyon Akış Tipi Üretim Çizelgeleme	43
3.2.2.	Esnek Akış Tipi Üretim Çizelgeleme	44
3.3.	PROBLEMİN ÇÖZÜM YAKLAŞIMLARI	45
3.3.1.	Yapay Arı Kolonisi Algoritması	46
3.3.2.	Genetik Algoritmalar	51
3.3.3.	Karınca Koloni Optimizasyonu Algoritması	57
3.3.4.	Parçacık Sürü Optimizasyonu Algoritması	62

DÖRDÜNCÜ BÖLÜM

AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNE METASEZGİSEL YAKLAŞIMLARLA BİR UYGULAMA DENEMESİ

4.1.	UYGULAMA PROBLEMİNİN TANITIMI	67
4.2.	PROBLEMİN ABC ALGORİTMASI İLE ÇÖZÜMÜ	68
4.3.	PROBLEMİN GA İLE ÇÖZÜMÜ	74
4.4.	PROBLEMİN LEKİN PROGRAMI İLE ÇÖZÜMÜ	83
4.5.	ÇÖZÜM YAKLAŞIMLARININ GENEL ANALİZİ	86

SONUÇ	89
KAYNAKÇA	92
EKLER	104
ÖZGEÇMİŞ	107

TABLOLAR LİSTESİ

Tablo 1: İterasyon 1000 İken ABC Algoritmasının Çözüm Sonuçları	70
Tablo 2: İterasyon 5000 İken ABC Algoritmasının Çözüm Sonuçları	71
Tablo 3: İterasyon 10000 İken ABC Algoritmasının Çözüm Sonuçları	72
Tablo 4: İterasyon 50000 İken ABC Algoritmasının Çözüm Sonuçları	73
Tablo 5: İterasyon 1000 İken GA Çözüm Sonuçları.....	75
Tablo 6: İterasyon 5000 İken GA Çözüm Sonuçları.....	79
Tablo 7: İterasyon 10000 İken GA Çözüm Sonuçları.....	81
Tablo 8: Problemin Lekin Programında SB İle Çözümü	85
Tablo 9: Problemin Çizelgeleme Kurallarıyla Çözümü.....	85
Tablo 10: ABC ve GA Çözüm Sonuçlarının Karşılaştırılması	86
Tablo 11: ABC, GA ve Lekin Sonuçları CPU Analizi	88
Tablo 12: ABC, GA ve Lekin Sonuçları Makespan Analizi.....	88

ŞEKİLLER LİSTESİ

Şekil 1: Bir İşletmede Çizelgeleme Fonksiyonunun Yeri.....	28
Şekil 2: Çizelge Tipleri Arası İlişki	29
Şekil 3: Üretim Çizelgeleme Problemlerinin Sınıflandırılması	30
Şekil 4: Saf Akış Tipi Yerleşim	41
Şekil 5: Genel Akış Tipi Yerleşim	42
Şekil 6: Özel Akış Tipi Yerleşim Modelleri	42
Şekil 7: Kaşif Arıların Kovanda Yaptıkları Salınım Dansı.....	48
Şekil 8: ABC Algoritmasının Akış Şeması.....	50
Şekil 9: Genetik Algoritmaların İşleyişi	53
Şekil 10: Karıncaların Yuvaları İle Yiyecek Kaynakları Arasında Engel Oluşması Durumu	59
Şekil 11: <i>i</i> Parçacığının Şimdiki En İyi Ve Yerel En İyiye Doğru Hareketi.....	63
Şekil 12: Uygulama Problemindeki 5 Makine Ve Dizilimi	68
Şekil 13: İterasyon 1000 İken, Makespan ve NP İlişkisi	71
Şekil 14: İterasyon 5000 İken, Makespan ve NP İlişkisi	72
Şekil 15: İterasyon 10000 İken, Makespan ve NP İlişkisi	73
Şekil 16: İterasyon 50000 İken, Makespan ve NP İlişkisi	74
Şekil 17: İterasyon 1000 ve Popülasyon 50 İken Makespan ile Mut.-Sel. Oranının İlişkisi.....	77
Şekil 18: İterasyon 1000 ve Popülasyon 100 İken Makespan ile Mut.-Sel. Oranının İlişkisi.....	78
Şekil 19: İterasyon 1000 ve Popülasyon 200 İken Makespan ve Mut.-Sel. Oranının İlişkisi.....	78
Şekil 20: İterasyon 5000 ve Popülasyon 100 İken Makespan ve Mut.-Sel. Oranının İlişkisi.....	80
Şekil 21: İterasyon 5000 ve Popülasyon 200 İken Makespan ve Mut.-Sel. Oranının İlişkisi.....	80
Şekil 22: İterasyon 10000 ve Popülasyon 100 İken Makespan ve Mut.-Sel. Oranının İlişkisi.....	82
Şekil 23: İterasyon 10000 ve Popülasyon 200 İken Makespan ve Mut.-Sel. Oranının İlişkisi.....	82
Şekil 24: Matlab Programında GA Bir Çözüm Bulmuşken Ekran Görüntüsü	83
Şekil 25: Lekin Programından SB İle Cmax Bulunmasına Dair Ekran Görüntüsü	86

KISALTMALAR VE SİMGELER

ABC	: Yapay Arı Kolonisi (Artificial Bee Colony)
ACO	: Karınca Kolonisi Optimizasyonu (Ant Colony Optimization)
BATÇ	: Beklemesiz Akış Tipi Çizelgeleme
Cmax	: Maksimum Tamamlanma Zamanı (makespan)
DE	: Diferansiyel Gelişim (Differential Evolution)
DSA	: Dal Sınır Algoritması
EA	: Gelişimsel Algoritma (Evolutionary Algorithm)
ES	: Gelişimsel Strateji (Evolutionary Strategy)
FFSP	: Esnek Akış Tipi Çizelgeleme Problemi (EATÇP)
FIFO	: İlk Giren İlk Çıkar Metodu (First In First Out)
FSP	: Akış Tipi Çizelgeleme Problemi (Flowshop Scheduling Problem)
GA	: Genetik Algoritmalar (Genetic Algorithms)
GRASP	: Greedy Randomized Adaptive Search Procedure
JSP	: Atölye Tipi Çizelgeleme Problemi (Jobshop Scheduling Problem)
KKE	: Karınca Kolonileri Eniyileme
m	: Makine Sayısı (1, 2, ... , m)
MMAS	: Max Min Karınca Sistemi (Max Min Ant System)
n	: İş Sayısı (1, 2, ... , n)
NEH	: Nawaz Enscore Ham Yöntemi
NP-Tam	: Polinom Zamanlı Olmayan-Tam Problemler (NP-Complete)
NP-Zor	: Polinom Zamanlı Olmayan-Zor Problemler (NP-Hard)
PDA	: Paralel Doyumsuz Algoritma
PFSP	: Permütasyon Akış Tipi Çizelgeleme Problemi
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
SA	: Tavlama Benzetimi (Simulated Annealing)
SB	: Darboğazı Öteleme Metodu (Shifting Bottleneck)
TSP	: Gezgin Satıcı Problemi (Travelling Salesman Problem)
VNS	: Değişken Komşuluk Arama (Variable Neighborhood Search)
YBA	: Yapay Bağışıklık Algoritması
YSA	: Yapay Sinir Ağları

GİRİŞ

Günümüz işletmeleri, hızla küreselleşen dünya ve değişen teknoloji nedeniyle kendilerini sürekli geliştirmek durumundadırlar. Bu pazar şartlarına ayak uyduramayan işletmeler zamanla pazardan çekilmek zorunda kalmaktadırlar. Ancak, işletmelerin bulundukları pazara uyum sağlamaları ve kaynaklarını etkin şekilde kullanmalarıyla devamlılık sağlayıp kalıcı olmalarından bahsetmek mümkün olacaktır.

Günümüzdeki işletmeler yoğun rekabetin olduğu bu ortamda çok düşük kâr marjları ile çalıştıkları için işletmelerde plansızlığa, atıl kapasiteye ve kısaca herhangi bir kaynak israfına yer yoktur. Bu yüzden işletme yöneticileri, üretim planlama çalışmalarını profesyonel şekilde ele alıp özenle yapmalıdırlar. Hazırladıkları etkin üretim planları ile de işletmeler kaynaklarını verimli şekilde kullanıp doğru zamanda doğru yere tahsis ederek pazarda hayatlarını sürdürürler.

Üretim sürecinin planlanması aşamasında gerçekleşen termin planı ve iş ile işlem sıralamaları, işletmelerde üretim çizelgelerine olan ihtiyacı doğurmuştur. Bu noktadan hareketle üretim çizelgeleme bir plan yapma ve karar verme sürecidir. Bu süreçte üretim hattına katılacak işlerin hangi sırada işlem göreceğine ve işlenmesine ne zaman başlanıp ne zaman bitirileceğine karar verilmektedir.

İşlerin sıralanması konusu ise sadece basit bir sıralama olayı değildir. İşletme yönetiminin kararları, işletme kaynaklarının limitleri, müşterilerin istekleri, rakipler nedeniyle standartlaşmış pazar koşulları ve çizelge planını hazırlayan kişilerin hedeflediği amaç bu iş sıralamasını etkilemektedir. Yöneticiler bir yandan işletme kaynaklarını maksimum etkinlikte kullanmaya çalışarak müşterilerini memnun edici teslim tarihleri gerçekleştirmek isterler bir yandan da ürünlerin maliyetine ve diğer pazar koşullarına uygun hareket etmek isterler. Bu nedenlerle çizelgeleme konusu işletmelerde her geçen gün artan bir öneme sahip olmuştur.

Çizelgeleme yapmak için işletmelerin hangi üretim tipinde çalıştıklarını iyi tespit etmeleri gerekmektedir. Üretimi sürekli aynı şekilde aynı ürünlerle büyük

miktarlarda mı yapıyorlar yoksa değişen miktarlarda değişik ürünlerle parti parti mi üretim yapıyorlar, bu konuda durum tespiti gereklidir. Ayrıca, işlemler için birbiriyle aynı işleve sahip paralel dizilmiş makineler mi kullanıyorlar yoksa birbiri ardına dizilmiş farklı görevlere sahip makineler mi kullanıyorlar, belirlemelidirler. Bu tespitlerden sonra işletmenin üretim çizelgeleme problemi belli olacaktır ve işletmede işlerin akış şekline göre; akış tipi veya atölye tipi bir ortamdan söz edilebilecektir.

Çalışmanın konusu olan akış tipi çizelgeleme problemleri, tek ve iki makineli problemlerde geleneksel yöntemlerle çözülürken, ikiden fazla makineli (çok makineli) problemlerde ise problemin karmaşıklığı ve çözüm zorluğu nedeniyle sezgisel ve metasezgisel yöntemler kullanılmaktadır. Çalışmada çok iş ve çok makineli uygulama problemi kullanılacağı için çoğunlukla metasezgisellerden yararlanılacaktır.

Çalışmanın amacı, işletmeler için hayati öneme sahip olan üretim çizelgeleme problemlerini incelemek ve akış tipi bir çizelgeleme test problemine birtakım metasezgiseller uygulayarak optimum sürelerde etkili çizelgeler elde etmektir. Ayrıca, kullanılan çözüm yöntemlerinin karşılaştırmalı analizi yapılarak performanslarının ölçümü yapılacaktır.

Bu çalışma kapsamında, üretim çizelgeleme problemlerinin bir türü olan akış tipi bir çizelgeleme konusu ele alınacaktır. Çalışmada gerçek hayattan derlenen akış tipi bir test problemi zaman sınırı nedeniyle sadece ABC ve GA metasezgiselleriyle çözülerek sonuçları karşılaştırmalı olarak analiz edilecektir.

Bu amaçlar ve sınırlar doğrultusunda çalışmanın birinci bölümünde; çözüm yöntemlerine göre sınıflandırılmış ve kendi içerisinde geçmişten günümüze doğru yıllara göre sıralanmış 61 çalışmadan oluşan genel bir literatür değerlendirmesi verilmiştir.

İkinci bölümde; çizelgelemenin tanımı ve amacından bahsedilmiş, üretim çizelgeleme ve tiplerinden söz edilmiştir. Üretim çizelgeleme problemlerinin sınıflandırılması yapılmış, problemlerin notasyonu ve gösterimi verilmiş, problemlerdeki öncelik kuralları ve performans ölçüm kriterlerinden bahsedilmiştir.

Üçüncü bölümde; akış tipi çizelgeleme problemlerinin tanımı, özellikleri ve çeşitlerinden bahsedilmiş, bu problemlerin çözümünde de kullanılan bazı popüler metasezgiseller açıklanmıştır.

Dördüncü bölümde de; gerçek hayattan bir problemin verileri kullanılarak test problemine dönüştürülen akış tipi uygulama problemimiz anlatılmış, probleme GA ve ABC algoritmaları ile SB sezgiseli ve birtakım çizelgeleme kuralları uygulanmış ve tüm bu yöntemlerin sonuçları karşılaştırılmıştır. Çalışma test problemimizin optimum çözümleri olan en küçük tamamlanma zamanına sahip iş sıralamalarını sunarak ve akış tipi problemler için algoritmalar geliştirerek literatüre katkı sağlamıştır.

BİRİNCİ BÖLÜM

LİTERATÜR DEĞERLENDİRMESİ

Akış tipi çizelgeleme problemlerinin çözümünde metasezgisel yaklaşımların ve bazı klasik yöntemlerin kullanıldığı, literatürün ulaşılabilen güncel çalışmalarının bir değerlendirmesi olarak aşağıda konuyla ilgili 61 çalışmaya yer verilmiştir. Çalışmalardan en güncelinin yayın tarihi 2012 yılı iken en eskisinin tarihi de 1994 yılıdır. Literatür değerlendirmesi, çalışmalarda kullanılan çözüm yöntemlerine göre sınıflandırılarak sunulmuştur. Ayrıca her bir yöntemin çalışmaları da kendi içinde eski tarihliden yeni tarihliye göre sıralanmıştır.

1.1. GENETİK ALGORİTMA

Engin (2001), akış tipi çizelgeleme problemlerinin genetik algoritma ile çözüm performansının artırılması konusunu incelemiştir. Çizelgeleme problemlerinde kullanılan rassal arama metotlarından bahsetmiş ve sonrasında da genetik algoritma modeli ve parametreleriyle uygulama alanlarını açıklamıştır. Parametrelerin optimizasyonu ile ilgili on farklı problem için 10250 adet deneme yapmıştır ve öncelikli olarak GA'nın performansını etkileyen altı parametreyi iki makine problemleri üstünde test etmiştir. Sonuçta GA, tek optimum iş sırası yerine alternatifli iş sırası oluşturduğundan Johnson algoritmasından daha iyi performans göstermiştir. Çok makine problemi olarak da 10 tane problem üzerinde parametrelerle ilgili deneyler yapmıştır. Akış tipi çizelgeleme problemlerinde daha düşük nesil sayılarında optimum yada optimuma yakın çözüme ulaşabilmek için başlangıç popülasyonunun 40; iki makine için 'kısmi yapay seçim' üreme yöntemi, çok makine için 'akış zamanlı rulet çemberi' üreme yöntemi; çaprazlama yöntemi olarak 'sıralı çaprazlama' yöntemi; çaprazlama oranı olarak %60 ila %100 arası; mutasyon oranı olarak da %40 ila %70 arasında seçilmesinin uygun olacağını tespit etmiştir. Yaptığı deney tasarımının bir

sonucu olarak; akış tipi çizelgeleme problemlerinde GA'nın performansını etkileyen en önemli faktörler, üreme ve çaprazlama yöntemleridir.

Chakraborty, Laha ve Chakraborty (2001), akış tipi çizelgelemenin, bazı çizelgeleme kriterlerini sağlamak için bir takım makinelerde işlem görecekt işlerin optimum sırasını belirlemekle ilgilendiğini ve bunun NP-tam türünde olduğunu ifade etmişlerdir. Bu çalışmada ise, GA ve sezgisel aramanın iyi özelliklerini bünyesinde birleştiren yeni bir sezgisel genetik algoritma (NGA) geliştirmişlerdir. NGA metodunu, bir takım problemlerde çalıştırmışlar ve performansını, iyi bilinen NEH sezgiseli ve konvansiyonel genetik algoritmayla karşılaştırmışlardır. Sonuç olarak ise NGA metodunun hemen hemen tüm örneklerde daha iyi performans gösterdiğini görmüşlerdir.

Engin ve Fırlalı (2002), tamamlanma zamanı kriterli akış tipi çizelgeleme problemlerinin genetik algoritma yardımı ile çözümü için uygun bir çaprazlama operatörü belirlemeye çalışmışlardır. Genetik algoritmanın çözüm performansını önemli ölçüde etkileyen parametrelerden birisi olan çaprazlama operatörünün akış tipi çizelgeleme problemleri için en etkinini belirlemek amacıyla, bu tip problemlerin çözümüne uygun olan altı ayrı çaprazlama operatörünü işlem süreleri 1-1000 dk aralığında değişen çok makine-çok iş problemleri üzerinde test etmişlerdir. Sonuç olarak, etkin çaprazlama yönteminin makine sayısına bağlı olarak değiştiğini belirlemişlerdir.

Engin ve Fırlalı (2002)(1) bu çalışmada ise, akış tipi çizelgeleme problemlerinin genetik algoritma ile çözümünde çözüm süresi ve kalitesi üzerinde etkin olan bir üreme operatörü belirlemişlerdir. Literatürde kullanılan akış zamanlı rulet çemberi ve yapay seçim yöntemi ile yeni geliştirilen kısmi yapay seçim, makine verimli rulet çemberi ve ters yapay seçim üreme yöntemlerini farklı büyüklüklerdeki 10 problem üzerinde test etmişlerdir. İşlem süreleri uniform dağılıma uygun olarak [1-25] dakika arasında rassal olarak üretilen problemler üzerinde yaptıkları toplam 1250 adet deney sonucunda, akış tipi çizelgeleme problemlerinin genetik algoritma ile çözümünde, iki makine problemleri için kısmi yapay seçiminin; çok makine problemleri için akış zamanlı rulet çemberinin iyi performans gösterdiğini ifade etmişlerdir.

Körez (2005), sıralı akış tipi çizelgeleme problemlerini ele almış ve problem için bir genetik algoritma geliştirmeye çalışmıştır. Bu algoritmanın performansını

değerlendirmek için; iki kısımda ele alınan sıralı akış tipi probleminin ilk kısmında işlem zamanları matrisinde en büyük işlem zamanlarının ara makinelerden birinde oluşması durumunu ele almıştır. İkinci kısmında ise en büyük işlem zamanlarının ilk ya da son makinede oluşması özel durumunu ele almış ve tüm bu kısımlarda genetik algoritma Smith Panwalkar Dudek algoritmasına çok yakın sonuçlar, tavlama benzetimi yaklaşımına göre ise daha iyi sonuçlar elde etmiştir. Sonuç olarak bu iki durum uygulaması ele alınınca, geliştirdiği genetik algoritma, Smith Panwalkar Dudek algoritmasından çok daha kısa sürede iyi bir çözüm elde edebilmektedir.

Uçar (2005), permütasyon akış tipli atölyelerdeki pozitif gecikmeli iş sayısını minimuma indirmeyi amaçlamıştır. Bu amaç için, geleneksel bir genetik algoritma ve kesikli bir parçacık sürü optimizasyonu algoritması geliştirmiştir. Her bir algoritma için gecikmeli iş sayısına ait değerler ve işlemci süreleri için istatistiki ölçümler yapmıştır. Her iki makinenin de m makinede en uygun iş sıralamasını bulmadaki performanslarını karşılaştırmıştır. Yaptığı bu deneylerin sonucu olarak, parçacık sürü optimizasyonu algoritmasının en küçük konum (EKK) sezgisel yöntemi sayesinde umut verici neticeler verdiğini bulmuştur.

Ruiz, Maroto ve Alcaraz (2006), permütasyon akış tipi çizelgeleme problemlerini çözmek için diğer iyi bilinen algoritmalarla karşılaştırıldığı zaman rekabet edebilecek yeni genetik algoritmalar önermişlerdir. Optimizasyon kriteri olarak toplam tamamlanma zamanı kriterini ele almışlardır. Bize güçlü bir genetik algoritma ve hızlı bir hibrit uygulama göstermişlerdir. Önerdikleri algoritmalar, yeni bir jenerasyon planı oldukça etkili bir popülasyon başlatma ve yerel arama ile melezleştirme gibi gelişmiş teknikleri olan yeni genetik operatörler kullanmaktadır. Algoritmaların etkinliği için, genetik algoritmalar, tabu aramaları, tavlama benzetimi ve diğer gelişmiş ve son zamanlarda kullanılan algoritmalarla oluşan 11 metot ile karşılaştırma yapmışlardır. Değerlendirme için Taillard'ın kıyas problemlerini kullanmışlardır. Sonuçlar, önerilen algoritmaların çok etkili ve aynı zamanda kolay uygulanabilir olduğunu göstermiştir.

Ceran (2006), NP-Zor sınıfındaki esnek akış tipi çizelgeleme problemlerinin çözümü ve çözüm aşamalarının analizi için maksimum tamamlanma zamanını minimize edecek çözümler üreten genetik algoritmalar kullanmıştır. Bu çözümler arasındaki ilişkiyi ortaya çıkarmak için de veri madenciliği kullanmıştır. Geliştirdiği genetik algoritma kısa sürede etkili çözümler sunmuştur. Yaptığı karşılaştırma ve analizler

neticesinde problem türüne ait birtakım kurallar çıkarmıştır. Bu kuralları da, optimum iş sıraları ve C_{max} 'ı göz önüne alarak belirlemiştir.

Kellegöz (2006), toplam geç bitirme zamanının en küçüklenmesi performans ölçütlü permütasyon akış tipi çizelgeleme problemlerini detaylıca incelemiş ve bu problemleri çözmek için bir genetik algoritma geliştirmiştir. İlave olarak da bir çözüm iyileştirme algoritması ile geliştirdiği uygun bir çaprazlama operatörünü bu genetik algoritmaya entegre etmiştir. Karşılaştırma problemlerini kullanarak literatürdeki en iyi olarak bilinen algoritmalarla kıyas yapmıştır. Yaptığı simülasyon çalışması sonucunda da, geliştirdiği algoritmanın diğerlerinden daha iyi performans sergilediğini göstermiştir.

Temiz ve Erol (2007), günümüz üretim sistemlerinde büyük öneme sahip m-makinelili akış tipi çizelgeleme problemlerinde işlem zamanları ve teslim tarihleri gibi zaman parametrelerinin belirsiz olduğu durumu ele alarak üretimin tamamlanma zamanı, maksimum gecikme ve toplam akış zamanı performans ölçüm kriterlerini eş zamanlı olarak optimize eden genetik algoritma temelli çok amaçlı bir yaklaşım geliştirmişlerdir. Geliştirdikleri bulanık iş ve teslim zamanlı çok amaçlı genetik algoritma sonucunda amaç değerlerinin üyelik fonksiyonlarıyla ifade edildiği etkin çözümler elde etmişlerdir. Algoritmaları küçük boyutlu problemlerde etkinliğini kanıtlamıştır. Genetik algoritmanın en iyi parametre değerlerini faktöriyel deney tasarımı ile belirlemişlerdir. Algoritmalarının orta ve büyük boyutlardaki problemler için de makul zamanda etkin çözümler ürettiğini göstermişlerdir.

Ruiz ve Allahverdi (2007), hazırlık zamanlarının işlem zamanlarından ayrı olarak ele alındığı ve sıranın bağımsız olduğu maksimum gecikmenin minimize edilme amacı taşıyan m makinelili beklemesiz akış tipi problemini dikkate almışlardır. Üç makinelili örnek için bir üstünlük ilişkisi geliştirmişlerdir ve birtakım sezgiseller ile dört yeni etkili genetik algoritma önermişlerdir. Önerdikleri sezgisellerin, genetik algoritmaların ve üstünlük kuralının performansını değerlendirmek için geniş deneyler uygulamışlardır. İstatistiki analizler ve yapılan deneyler sonucu, bu genetik algoritmaların aynı koşullarda literatürde var olanlardan daha iyi sonuçlar sağladığı görülmüştür. Ayrıca, önerilen üstünlük kuralı işlem ve hazırlık zamanlarının ayrık olduğu örnekler için büyük bir potansiyel olarak görülmektedir.

Gözen (2007), NP problemleri grubunda yer alan çok prosesli esnek akış tipi çizelgeleme problemlerini bulanık hale getirip genetik algoritma ve tavlama benzetimi yöntemleriyle çözmüştür. Çalışmasında bulanık mantık ve problemlerin bulanıklaştırılması sürecini de sunmuştur. Bu çok prosesli akış tipi çizelgeleme problemleri için genetik algoritma ve tavlama benzetimi sonuçlarını kıyaslamıştır. Sonuç olarak, genetik algoritmanın tüm problemlerde daha başarılı sonuçlar verdiği gözlemlenmiştir.

Zobolas, Tarantilis ve Ioannou (2009), permütasyon akış tipi çizelgeleme problemlerinde tamamlanma zamanının minimizasyonu için hibrit bir metasezgisel önermişlerdir. Önerdikleri bu kolay yapılandırılan, hızlı ve güçlü çözüm yaklaşımı üç bileşene sahiptir, bunlar; açgözlü rassal yapı sezgiseline dayalı bir başlangıç popülasyonu oluşturma metodu, çözüm gelişimi için bir genetik algoritma ve popülasyon iyileştirme için bir değişken komşuluk arama bileşenidir. Kıyas problemleri data setleriyle hesaplanan sonuçlar, önerdikleri hibrit metasezgiselin kısa hesaplama zamanlarında yüksek çözüm kalitelerine ulaşabildiğini göstermiştir. Ayrıca çok az kullanıcı tanımlı parametre gerektirmesi, bu metasezgiseli gerçek hayattaki akış tipi çizelgeleme problemleri için kabul edilebilir kılmıştır.

1.2. BAĞIŞIKLIK SİSTEMLERİ

Engin ve Döyen (2007), akış tipi çizelgeleme problemi için toplam tamamlanma zamanının minimizasyonu ile ilgilenmişlerdir. Çalışmada, yapay bağışıklık sistemindeki bağışıklık tepkisinin iki ayrı mekanizması olan klonel seçim prensibi ve benzerlik mekanizması üzerine kurulmuş bir yöntem uygulamışlardır. Yapay bağışıklık sisteminin parametrelerinin belirlenmesinde çok aşamalı bir deney tasarımı prosedürü uygulamışlardır. Deney sonuçları, yapay bağışıklık sistemlerinin klasik çizelgeleme ve tavlama benzetimi algoritmalarından daha iyi sonuçlar verdiğini göstermiştir.

Akçay (2009), permütasyon akış tipi çizelgeleme problemlerinde en son işin tamamlanma zamanı ve işlerin akış zamanlarının toplamalarının en küçüklenmesi amacıyla, açışal dağıtıklı çok amaçlı bağışıklık algoritması (AD-ÇABA) önermiştir. Önerdiği algoritma ile elde ettiği sonuçlarla bazı var olan algoritmaların pareto optimum sonuçlarını birleştirmiş ve çalışmaların net pareto cepheye yer alan çözüm sayılarını

karşılaştırmıştır. Önerdiği algoritma, literatürde kullanılan 75 problemin 19'unda net pareto cephenin tamamını, 56 problemde ise pareto cephede yer alan çözümlerin yüzde ellisini geliştirerek diğer algoritmalarından daha fazla katkı sağlamıştır.

1.3. SİNİR AĞLARI

Seçme (2006) yüksek lisans tezinde n iş m makineli akış çizelgeleme problemlerinin gezgin satıcı problemi yaklaşımıyla Hopfield sinir ağları ile çözümünü göstermiştir. $n \times n$ boyutlu permütasyon çizelgelerinin Hopfield sinir ağları ile simülasyonlarını yapmıştır. Oluşturduğu simülasyon modelinde ağ, öncelikle değerleri rastgele oluşturulan 5×5 bir problem üzerinde denenmiş ve ağın uygun ve optimum/optimuma yakın çözümler üretebildiğini görmüştür. Daha sonra ağı 20×20 boyutlu bir permütasyon çizelgesine uygulamış ve Hopfield sinir ağları yaklaşımı çözümü ile yöneylem araştırması tekniklerinden olan dal sınır algoritması çözümlerini karşılaştırmıştır. Sonuç olarak, n^2 nöron sayısı gerektirmesi ve optimal çözümü garanti edememesi Hopfield ağlarının dezavantajları olurken bilinen diğer yöntemlere göre daha az karmaşık olması ve paralel yapıları ile hızlı işlem kapasiteleri, etkin zamanlı çözümler sunmasını sağlamıştır.

Günaydın (2008), beklemesiz akış tipi çizelgeleme (BATÇ) problemlerini çözmek için yapay sinir ağları (YSA) algoritması önermiştir. 2006 yılında Agarwal ve arkadaşları tarafından geliştirilen yapay sinir ağları için adaptif öğrenme yaklaşımını, BATÇ problemlerine uygulamıştır. Önerdiği yapay sinir ağları algoritmasının performans testi için, 2003 yılında Aldowasian ve Allahverdi'nin önerdiği genetik algoritma ile karşılaştırma yapmıştır. BATÇ problemleri için literatüre uygun 192 problem seti oluşturmuştur. Bu problem setlerini GA ve YSA için çözmüş ve sonuçları karşılaştırmıştır. Sonuçta önerdiği YSA yaklaşımının önerilen GA yaklaşımından iyi sonuçlar verdiğini gözlemlemiştir.

1.4. DOYUMSUZ ALGORİTMA

Öztürk (2007), esnek akış tipi çizelgeleme (EATÇ) problemlerini ve çok işlemcili esnek akış tipi çizelgeleme (ÇİEATÇ) problemlerini çözmek için paralel doyumsuz algoritma (PDA) önermiştir. İki aşamada uygulanan PDA için bu çalışmada, kontrol parametreleri olarak başlangıç popülasyonu, alt grup sayısı, iterasyon sayısı, doyum oranı ve inşa yöntemleri kullanmıştır. Literatürden elde ettiği EATÇ ve ÇİEATÇ

problemlerini çözerek önerdiği PDA ile bulduğu değerlerle, yine literatürden bazı dal sınır algoritması (DSA), yapay bağışıklık sistemi (YBS) ve genetik algoritmalarından elde ettiği değerleri karşılaştırmıştır. Bu kıyaslamaların sonucunda ise, önerdiği algoritmanın bulduğu değerlerin, literatürdeki YBS ve bir GA'nın performanslarıyla aynı, DSA ve diğer GA'nın performanslarından üstün olduğunu göstermiştir.

İşler (2009), hazır giyimin iş gücü kayıplarının çok ve termin sürelerinin kısa olduğu dinamik bir üretim şekli olarak NP-Zor sınıfta yer alması nedeniyle esnek akış tipi çizelgeleme problemleri için bir paralel doyumsuz algoritma geliştirmiştir. Problemin çözümünde bir hazır giyim işletmesinde örnek olarak uygulama yapmıştır. Bu bulanık ortamın şartlarını göz önüne alarak bir paralel doyumsuz algoritma geliştirmiş ve işletmeden elde ettiği verilerle test etmiştir. Bu testlerde genetik algoritma ile karşılaştırma da yapmıştır. Önerdiği paralel algoritmanın genetik algoritmadan daha iyi sonuçlar verdiğini ortaya çıkarmıştır.

Li ve Zhao (2009), esnek akış tipi çizelgeleme problemlerinin karmaşık atölye tipi çizelgeleme problemlerinin bir türü olduğunu ifade etmişlerdir. Esnek akış tipi çizelgeleme problemleri için sadece bazı özel durumlarda çok terimli optimal algoritmalar var olduğunu, ancak çoğu durumda problemin NP-zor olduğunu söylemişlerdir. Açgözlü algoritmayla esnek akış tipi çizelgeleme problemlerini çözmek için orijinal problemin bir basitini ve algoritmayla verimin bir kombinasyonunu kullanmışlardır. Sonuç olarak bu çalışmada esnek akış tipi çizelgeleme problemleri için açgözlü bir algoritma vermişler ve algoritmanın yeteneğini değerlendirmişlerdir.

1.5. KARINCA KOLONİLERİ

Stützle (1998), akış tipi problemler için bir karınca koloni optimizasyonu (ACO) yaklaşımı sunmuştur. Bu yaklaşımın yeni bir algoritmik yaklaşım olduğunu ve kombinatoriyal optimizasyon problemlerini çözmede uygulanabileceğini söylemiştir. Yapay karıncaların akış tipi problemleri çözmek için kullanıldığını ve sonra da yerel arama prosedürü kullanarak iyileştirme yapıldığını anlatmıştır. Akış tipi problemler için olan diğer sezgisellerle yaptığı karşılaştırmada, sunduğu yaklaşımın çok gelecek vaat eden sonuçlar verdiği gözlemlenmiştir.

Ying ve Liao (2004), karınca koloni sistemi ilk kez bir $n/m/P/C_{max}$ problemine uygulanmıştır. Geliştirdikleri algoritmayı kanıtlamak için Taillard'ın iyi

bilinen kıyas problemleri setindeki örneklerle deney yapmışlardır. Bu karınca kolonisi sistemini literatürden genetik algoritma, tavlama benzetimi ve komşuluk arama gibi metasezgisellerle karşılaştırmışlardır. Deney sonuçları da, $n/m/P/Cmax$ problemi için bu karınca koloni sisteminin daha etkin bir metasezgisel olduğunu göstermiştir.

Rajendran ve Ziegler (2004), permütasyon akış tipi çizelgeleme problemini, işlerin toplam akış zamanının minimizasyonunu dikkate alarak tamamlanma zamanı minimizasyonu ile ele almışlardır. İki karınca koloni optimizasyonu algoritması önermişler ve permütasyon akış tipi için analiz etmişlerdir. İlk algoritma, MMAS (max-min ant system) adında olan ve Stuetzle tarafından geliştirilmiş algoritmayı genişletmiş ikinci algoritma ise yeni geliştirilmiştir. Önerdikleri algoritmaları Taillard'ın 90 tane kıyas problemine uygulamışlardır. Tamamlanma zamanı minimizasyonu için Taillard tarafından verilen sezgisellerin sonuçlarıyla karşılaştırma yapmışlar ve sonuçta önerdikleri algoritmalar MMAS sonuçlarından daha iyi performans göstermiştir. Son olarak, 90 problemten 83'ü dikkate alındığında önerilen bu iki karınca koloni optimizasyonu algoritması tarafından daha iyi çözümler bulunmuşlardır.

Yağmahan ve Yenisey (2006), NP-zor tipi problem olarak bilinen permütasyon akış tipi çizelgeleme probleminin Karınca Kolonileri Eniyileme (KKE) ile çözümünde kaliteli sonuçlar elde etmek üzere parametre eniyileme üzerine çalışmışlardır. Ele aldıkları çizelgeleme problemi, tüm makinelerde aynı sıraya sahip işlerin çizelgelenmesiyle oluşmaktadır. KKE ise, birleşik eniyileme (kombinatoriyal optimizasyon) problemleri için son zamanlarda kullanılan umut verici metasezgisel bir yaklaşım olarak öne çıkmaktadır. Böylelikle, önce örnek test problemleri üzerinde deneyler yapmışlar ve test edilecek uygun parametre değerlerini belirlemişlerdir. Sonra ise en iyi parametreleri belirlemek için deneysel tasarım kullanmışlardır. Sonuç olarak yapılan deneysel çalışmaların algoritmanın performansını dikkate değer şekilde artırdığını gözlemlemişlerdir.

Güçlü (2010), beklemesiz akış tipi çizelgeleme (BATÇ) problemlerini konu edinmiştir. BATÇ problemlerindeki art arda gelen makineler arasındaki bekleme ortadan kaldırma amacı için, karınca kolonileri algoritmasını kullanmıştır. Oluşturduğu m sayıda makine ve n sayıda işli toplam 192 problem setini C++ ile yapılan programda çözmüştür. Elde ettiği sonuçları, Günaydın'ın 2008 yılında yaptığı yapay sinir ağlarıyla ilgili çalışmasının sonuçlarıyla karşılaştırmış ve sonuç olarak sunduğu karınca kolonileri

algoritmasının o yapay sinir ağıları algoritmasından iyi sonuçlar verdiğini gözlemlemiştir.

1.6. TAVLAMA BENZETİMİ

Nearchou (2004), akış tipi çizelgeleme problemlerini çözmek için yeni bir hibrit tavlama benzetimi (SA) algoritması sunmuştur. Bu hibrit SA algoritmasında, tavlama benzetiminin basit yapısı ile birlikte genetik algoritmanın ve yerel arama tekniklerinin de alanından bazı özellikleri bünyesinde birleştirerek kullanmıştır. Algoritması özellikle, bir aday çizelge popülasyonundan çalışmakta ve uygun küçük karışıklık planlarına başvurarak komşu çizelgeler için yeni popülasyonlar üretmektedir. Ayrıca tavlama süreci esnasında, performansını iyileştirmek ümidiyle çizelge popülasyonlarına bir yinelenen tepe tırmanışı prosedürü uygulamıştır. Önerilen yaklaşımı hızlı ve kolayca uygulamıştır ve 500 iş 20 makineye kadar varan birçok kıyas problemine uyguladığı bilgisayar sonuçları, yaklaşımın yüksek kaliteli performansını ve etkinliğini göstermiştir. Daha önceki SA ve GA yöntemleriyle karşılaştırıldığında ise önerilen yaklaşım üstün bulunmuştur.

Low (2005), bağlantısız paralel makinelerle çok aşamalı bir akış tipi çizelgeleme problemi tarif etmiştir. Bağımsız hazırlık ve bağımlı taşıma zamanları gibi bazı süreç sınırlamalarını da dikkate almıştır. Amaç olarak ise sistemin toplam akış zamanını belirlemiştir. Tarif edilen problemi makul sürede çözmek için tavlama benzetimi temelli bir sezgisel önermiştir. Bu sezgisel öncelikle, iyi dizayn edilmiş bir başlangıç çözüm üreticisiyle başlamakta ve daha sonra çözümü daha da iyileştirmek için bir tavlama benzetimi prosedürü ile devam etmektedir. Önerdiği sezgiselin çözüm kalitesinden ve etkinliğinden emin olmak için sezgisele güvenilir belirli mekanizmalar geliştirmiş ve dahil etmiştir. Son olarak, hesaplanan bilgisayarlı sonuçlar, önerilen SA temelli sezgiselin çözüm doğruluğu ve etkinliğiyle iyi performans gösterdiğini ifade etmiştir.

Janiak, Kozan, Lichtenstein ve Oğuz (2007), makine merkezinde paralel makineli akış tipi çizelgeleme problemlerini konu edinmişlerdir. Her iş için her operasyonda işlem zamanları olduğu gibi teslim ve bekleme zamanları da verilmiştir. Çizelgeleme kriterini ise üç kısımdan oluşturmuşlardır; toplam ağırlıklı erken bitirme, toplam ağırlıklı gecikme ve toplam ağırlıklı bekleme zamanları. Bu kriterler, üretim

sürecindeki yarı mamullerin depolanma maliyetini ve müşteriyle anlaşılan ürünün son teslim tarihinde teslim edememe cezalarını hesaba katmaktadır. Problemi çözmek için, üç yapısal algoritma ve tabu araması ile tavlama benzetimine dayanan üç metasezgisel geliştirmişler ve deneysel olarak analiz etmişlerdir. Önerdikleri tüm algoritmalar, her bir makinedeki işlemlerin sırası (operation processing order) kavramıyla çalıştırılmıştır. Verilen işlem süreç sırası temelinde çizelgeleme yapısı probleminin doğrusal programlama görevi için azaltılabileceğini göstermişlerdir. Çizelge yapısı için aynı zamanda bir yaklaşma algoritması önermişler ve en iyi olduğu durumu göstermişlerdir.

Temiz (2008), üretim sistemlerinde çok yaygın olan permütasyon çizelgeleme probleminde tamamlanma zamanı, toplam akış zamanı ve en büyük gecikme zamanı ölçütlerinin eş zamanlı optimizasyonunu amaçlamıştır. Çok ölçütlü çizelgeleme olarak modellediği problemde etkin çözümlerin belirlenmesi için üç aşamalı tavlama benzetimi algoritmasını geliştirerek sunmuştur. Geliştirdiği algoritmayı değişik boyutlardaki problemlere uygulamıştır. Sunduğu algoritmanın etkin çözümleri kabul edilebilir bir hesaplama zamanında bulunduğu görülmüştür.

1.7. TABU ARAMASI

Ben-Daya ve Al-Fawzan (1998), permütasyon akış tipi çizelgeleme problemlerinin çözümü için bir tabu araması algoritması önermişlerdir. Önerdikleri tabu araması yaklaşımı uygulamasıyla, verilen bir dizinin komşularını üretmek için basit teknikler ve daha önce ele alınmamış yoğunlaştırma ve çeşitlendirme için birleştirilmiş bir tasarı teklif etmişlerdir. Bu yeni özellikler de, basit karşılaştırılabilir mekanizmaların kullanıldığı önceki tabu arama uygulamalarının üzerinde iyileştirilen bir uygulamada sonuç vermiştir. Aynı zamanda, literatürden bir tavlama benzetimi algoritması tarafından üretilmiş sonuçlardan daha iyi sonuçlar verdiğini gözlemlemişlerdir.

Grabowski ve Wodecki (2004), tamamlanma zamanı ile klasik akış tipi çizelgeleme problemlerini konu edinmişlerdir. Engelleriyle ilgili problemin bazı yeni özelliklerini sunmuşlar ve tartışmışlardır. Bu özellikler onlara, bir tabu arama yaklaşımına dayanan yeni ve hızlı bir yerel arama prosedürü önermeye imkân tanımıştır. 500 iş 20 makineye kadar uzanan bilgisayarlı deneyler yapmışlar ve sonuçları, literatürde tartışılan en iyi algoritmalarca elde edilmiş sonuçlarla karşılaştırmışlardır. Sonuçlar ise, önerdikleri algoritmanın bu akış tipi örnekleri yüksek bir doğrulukla ve

kısa bir zamanda çözdüğünü göstermiştir. Ayrıca sundukları bu özellik ve fikirlerin her yerel arama prosedürüne uygulanabileceğini ifade etmişlerdir.

1.8. PARÇACIK SÜRÜLERİ

Tasgetiren, Liang, Sevkli ve Gencyilmaz (2007), tamamlanma zamanı ve işlerin toplam akış süresini minimize etme amaçlı permütasyon akış tipi çizelgeleme problemlerini çözmek için bir parçacık sürü optimizasyonu algoritması önermişlerdir. Bu amaçla, Bean'ın rasgele anahtar temsilinden alınan ve SPV adı verilen en küçük pozisyon değeri sezgisel kuralını kullanmışlardır. Buna ilaveten, literatürdeki iyi bilinen kıyas problemlerini çözmek için PSO algoritmasına etkili bir yerel arama prosedürü olan (VNS) değişken komşuluk arama prosedürü eklemişlerdir. Sonuç olarak, toplam akış süresi kriteri için en iyi olarak bilinen 90 sonuçtan 57'si ve tamamlanma zamanı kriteri için en iyi olarak bilinen 800 sonuçtan 195'i geliştirilmiştir.

Rahimi-Vahed ve Mirghorbani (2007), ağırlıklı ortalama tamamlanma zamanının ve ağırlıklı ortalama gecikmenin eş zamanlı olarak minimize edildiği iki kriterli permütasyon akış tipi çizelgeleme problemini dikkate almışlardır. Akış tipi çizelgeleme problemi NP-zor sınıfta yer aldığı için, etkili bir çok amaçlı parçacık sürüsü (MOPS) dizayn etmişler ve problemin optimal sınırı olan bölgesel Pareto'yu bulmak için bu algoritmayı kullanmışlardır. Önerdikleri algoritmanın etkinliğini kanıtlamak için, çeşitli test problemlerini çözmüşler ve önerdikleri algoritmanın güvenilirliği için de, seçkin birçok amaçlı genetik algoritma olan SPEA-II ile bazı karşılaştırma ölçülerine dayalı olarak karşılaştırmışlardır. Deney sonuçları, önerilen MOPS algoritmasının özellikle büyük boyutlu problemler için genetik algoritmadan daha iyi sonuçlar verdiğini göstermiştir.

1.9. YAPAY ARI KOLONİLERİ

Pan, Tasgetiren, Suganthan ve Chua (2011), boşta çalışmanın olduğu ve olmadığı durumların her ikisinde de toplam ağırlıklı erken bitirme ve gecikme cezaları kriterleri ile çok akışlı akış tipi çizelgeleme problemlerini çözmek için kesikli bir yapay arı kolonisi (DABC) algoritması önermişlerdir. ABC algoritmasında olanın aksine, DABC algoritmasında bir yiyecek kaynağı kesikli bir iş permütasyonu olarak temsil edilmiş ve kaşiflere, izleyicilere ve işçi arılara yeni komşu yiyecek kaynağı oluşturmak için kesikli operatörler kullanılmıştır. Dahası, yerel yoğunlaşma yeteneğini artırmak için

önerilen DABC algoritmasına basit ama etkili bir yerel arama yaklaşımı ilave etmişlerdir. Yaptıkları deneysel sonuçların analizine göre, literatürdeki en iyi performans gösteren algoritmalara karşı önerilen DABC algoritmasının yüksek derecede etkili performansını göstermişlerdir.

Liu ve Liu (2011), permütasyon akış tipi çizelgeleme problemleri için tamamlanma zamanını minimize etmek amacıyla hibrit kesikli bir yapay arı kolonisi (HDABC) algoritması sunmuşlardır. Algoritmada, problem için her bir çözüm bir yiyecek kaynağı olarak adlanırken kesikli bir iş permütasyonu tarafından da temsil edilmiştir. İlk olarak başlangıç çözümünü belirli bir kalite ve çeşitlilikle NEH sezgiseline dayalı GRASP ile üretmişlerdir. İkinci olarak, işçi arılar, kaşif arılar ve izleyici arılar için yeni çözümler üretmek için ekleme, atlama, yeniden yol bağlama ve GRASP gibi kesikli operatörler ve algoritmalar uygulamışlardır. Daha sonra da en iyi çözüm için yerel arama uygulamışlardır. Sundukları algoritmayı çizelgeleme kıyas problemleri ile test etmişler ve hesaplanan sonuçlarla algoritmanın etkinliğini göstermişlerdir.

1.10. DİFERANSİYEL GELİŞİM

Onwubolu ve Davendra (2006), kesikli değişkenler ve tamsayı içeren doğrusal olmayan programlama problemlerini çözmek için diferansiyel gelişim (DE) algoritması ve onun uygulamalarına dayalı yeni bir optimizasyon metodu tanımlamışlardır. Özellikle, akış tipi bir üretim sisteminde gecikme, akış zamanı ve tamamlanma zamanının minimizasyonu için diferansiyel gelişim algoritmasının bu uygulamasını, yöntemin pratik kullanımı ve kapasitelerin gösterimi için vermişlerdir. Diferansiyel gelişim algoritması ve genetik algoritmanın hangisinin daha güçlü olma konusunda itibarının yüksek olduğunu göstermek adına sonuçlarının karşılaştırılması için deneyler yapmışlardır. Gözlenen sonuçlarla, genetik algoritmayla karşılaştırılan DE algoritmasının çözüm kalitesindeki başarısını kanıtlamışlardır.

Pan, Tasgetiren ve Liang (2008), permütasyon akış tipi çizelgeleme problemleri için, Pan tarafından son zamanlarda geliştirilen yeni diferansiyel gelişim algoritmasını ve Ruiz & Stützle tarafından geliştirilen yinelenen açgözlü algoritmayı toplam akış zamanı kriterini kullanmadıkları için geliştirerek tamamlanma zamanı kriteri için gösterdikleri performansı toplam akış zamanı kriteri için de sağlamak

istemişlerdir. Dahası, çözüm kalitesini daha da iyileştirmek için bu iki algoritmaya melez olarak yeni bir öncelikli yerel arama prosedürü eklemişlerdir. Bilgisayarda yapılan deney sonuçları, öncelikli yerel aramalı bu iki algoritmanın, toplam akış zamanı ve tamamlanma zamanı kriterlerinin ikisinde birden literatürde bulunan diğer metotlardan daha iyi ve yüksek derecede rekabet gücü olan algoritmalar olduğunu göstermiştir. Son olarak, Taillard'ın örneklerinden toplam akış zamanı kriteri için 90 örnekten 40'ında, tamamlanma zamanı kriteri için de 4 en iyi bilinen çözümde geliştirme sağlamışlardır.

Pan, Wang ve Qian (2009), maksimum gecikme ve tamamlanma zamanı kriterleriyle beklemez akış tipi çizelgeleme problemlerinin çözümü için yeni bir kesikli diferansiyel gelişim (DDE) algoritması sunmuşlardır. Bilgisayarlı simülasyon sonuçlarını iyi bilinen kıyas problemlerine dayandırmışlar ve istatistiki performans karşılaştırmaları sağlamışlardır. Sonuçlara göre, önerdikleri DDE algoritmasını son zamanlarda yayınlanmış hibrit diferansiyel gelişim (HDE) algoritmasından ve iyi bilinen çok amaçlı genetik yerel arama (IMMOGLS2) algoritmasından arama kalitesi, güçlülük, etkinlik ve çeşitlilik leveli açılarından daha üstün bulmuşlardır. Aynı zamanda, yerel aramanın DDE algoritmasına ilavesiyle oluşan etkinliği de incelemişlerdir.

1.11. NEH SEZGİSELİ

Framinan, Leisten ve Rajendran (2003), NEH çok terimli sezgiselinin statik-deterministik permütasyon akış tipi problemlerinde tamamlanma zamanını minimize etmede en iyi sezgisellerden biri olduğundan ve yaklaşımın iki adımdan oluştuğundan bahsetmişlerdir. İlk adım, bir gösterge değerine göre işlerin başlangıç sıralarının üretilmesi, ikinci adım ise, birinci adımın başlangıç sırasına göre kısmi bir dizinin içine işlerin yinelenen şekilde eklenmesidir. Akış zamanını, boş zamanları ve tamamlanma zamanını minimize etmek için yaklaşımı genellemişler, 177 farklı başlangıç dizilimi oluşturmuşlar ve bunların performansını NEH ekleme yaklaşımında değerlendirmişlerdir. Kapsamlı bir nümerik çalışmaya dayanarak, önerdikleri sezgiselin bu üç amaç fonksiyonunun tümü için karşılaştırdıkları literatür sezgisellerinden önemli bir şekilde daha iyi performans gösterdiğini ifade etmişlerdir.

Chakraborty ve Laha (2007), tamamlanma zamanı minimizasyonu için kullanılan NEH sezgiselinin basitliği, zaman karmaşıklığı ve çözüm kalitesi açılarından en popüler algoritma olma yolunda ilerlediğinden bahsetmişlerdir. Bu çalışmada da NEH sezgiselini modifiye etmişler ve aynı algoritmik karmaşıklıkta iken çözüm kalitesinde önemli iyileşmeler elde etmişlerdir. Önerdikleri yaklaşımın gücünü, popülasyon temelli bir teknik kullanmasından sağladığını ifade etmişlerdir. Çeşitli büyüklüklerde rasgele seçilmiş genelleştirilmiş birçok test probleminde deneysel karşılaştırmalar yapmışlardır. Sonunda yaklaşımları, hem NEH sezgiseli hem de onun bilinen en iyi rakibi olan HFC sezgiselinden daha üstün performans göstermiştir. Yaptıkları istatistiki anlamlılık testleri de, iyileşme iddialarını kanıtlar niteliktedir.

Kalczynski ve Kamburowski (2008), 20 yılı aşkın süredir NEH sezgiselinin permütasyon akış tiplerinde tamamlanma zamanını minimize etmedeki NP-zor problemin çözümü için en iyi sezgisel olduğunun kabul edildiğini söylemişlerdir. NEH sezgiselinin gücünün esas olarak, ekleme fazı sırasında çizelgelenmiş olması için seçilen işlere göre öncelik sırasında yattığını ifade etmişlerdir. Framinan'ın sunduğu geniş deneylerin sonuçlarına göre de NEH, öncelik sırası incelenen 136 farklı sıradan üstündür. Son olarak çalışmada, Johnson algoritması konseptine dayanarak, tüm problem boyutlarında NEH algoritmasından daha üstün olan bir sezgisel için basit bir eşitlik bozma metodu ile kombine edilmiş yeni bir öncelik sırası önermişlerdir.

Pan ve Wang (2012), McCormick'in (PF) profil ayarlamasına dayalı wPF ve PW adında iki basit yapıli sezgisel önermişlerdir. Daha sonra, PF-NEH, wPF-NEH ve PW-NEH adında üç tane gelişmiş yapıda sezgiseli, önerdikleri diğer sezgisellerin NEH sezgiseliyle etkili bir yolla kombinesi ile sunmuşlardır. Daha sonra ise, yapısal sezgisellerin ürettiği çözümleri iyileştirmek için yerel arama metoduna dayalı eklemeler yaparak birleşik sezgiseller geliştirmişlerdir. Bilgisayarlı simülasyonları ve karşılaştırmaları Taillard'ın iyi bilinen akış tipi kıyas örneklerine uygulamışlardır. Sonuçlar, sunulan yapısal sezgisellerin var olanlardan önemli bir şekilde daha iyi performans gösterdiğini ve önerilen birleşik sezgisellerin önemli bir marjla yapısal sezgiselleri iyileştirdiğini ifade etmiştir. Bunlara ek olarak, sunulan sezgiseller 17 tane en iyi olarak bilinen büyük ölçekli Taillard örneği sonucunu geliştirmiştir.

1.12. DİĞER MATEMATİKSEL MODELLER VE SEZGİSELLER

Rajendran (1994) çalışmasında, tamamlanma zamanı amaçlı kısıtlı veya beklemesiz akış tipi çizelgeleme problemlerini ele almıştır. İş ekleme ve sezgisel öncelik ilişkileri temelinde basit bir sezgisel algoritma önermiştir. Çok sayıda ve çeşitli boyutlardaki problemler üzerinde değerlendirme yaptığında önerdiği sezgiselin verdiği çözümlerin açık şekilde doğru ve verdiği diğer iki sezgiselin sonuçlarından üstün olduğunu bulmuştur.

Chen, Glass, Potts ve Strusevich (1996), işlerin tamamlanma zamanını minimize etme amacıyla üç makine akış tipi çizelgelemede n sayıda işi sıralama problemini ele almışlardır. Çalışmada Johnson algoritmasına dayanan bir zaman sezgiseli sunmuşlardır. Bu sezgiselde optimal bir çizelgenin en fazla beşte üçü uzunluğunda yeni bir çizelge üretmeyi göstermişlerdir. Aynı zamanda bu çalışmada, genel akış tipleri için bir uygulama vermişlerdir.

Rajendran ve Ziegler (2003), işlerin hazırlık zamanlarının sıra bağımlı olduğu statik akış tipi çizelgeleme problemleri için etkili sezgiseller önermişlerdir. Objektif fonksiyonda, işlerin ağırlıklı gecikmelerinin ve ağırlıklı akış zamanlarının toplamının minimizasyonunu amaç olarak almışlardır. İyi bir sezgisel permütasyon iş sırası oluşturmak için iki sezgisel öncelik ilişkisi kullanmışlardır. Sonrasında önce bir sonra iki kez, sezgisel sırasında çözüm kalitesini geliştirmek için bir geliştirici plan uygulamışlardır. Önerdikleri algoritmaların nisbi değerlendirmeleri için kıyas metotları olarak bir açgözlü yerel arama ve bir rassal arama prosedürü kullanmışlardır. Geniş performans analizleri önerilen sezgisellerin, daha hızlı hesaplama zamanlarıyla kıyas metotlarından daha kaliteli çözümler ürettiğini göstermiştir.

Allahverdi, Aldowaisan ve Stotskov (2003), toplam tamamlanma zamanını minimize etmek için ayrık hazırlık zamanlı iki makine akış tipi çizelgeleme problemini tarif etmişlerdir. Problemde hazırlık zamanları, OR literatüründe yaygın bir şekilde kullanıldığı gibi deterministik olmak yerine rasgele değişkenler olabilme konusunda serbestliğe sahiptirler. Dahası, sadece alt ve üst sınırların verildiği serbest dağılımlı hazırlık zamanları kullanırlar. Çalışmada ilgilendikleri bu akış tipi için küresel ve yerel üstünlük ilişkileri geliştirmişler ve son olarak da açıklayıcı nümerik bir örnek vermişlerdir.

Eren ve Güner (2005), iki ölçütlü akış tipi çizelgeleme problemini beklemesiz olarak ele almışlardır. Ele aldıkları ölçütler, çizelgeleme problemlerinde en çok kullanılan toplam tamamlanma zamanı ve maksimum gecikme ölçütleridir. NP-zor yapıda olan bu problemin çözümü için tamsayılı programlama modeli sunmuşlardır. Ayrıca probleme uyarladıkları NEH yöntemi ile EDD, tabu arama ve rassal arama ile problemin büyük boyutlu çözümlerini gerçekleştirmişlerdir. En iyi çözümleri tabu arama yönteminin verdiğini söylemişler, uyarladıkları NEH ve EDD yöntemlerinin basit yapısıyla da bu tip problemlerde kullanılabileceğini göstermişlerdir. Bu çalışmanın literatüre katkısı; beklemesiz akış tipi çizelgelemede, toplam tamamlanma zamanı ve maksimum gecikme ölçütlerinin aynı anda ilk defa ele alınmış olmasıdır.

Ladhari ve Haouari (2005), tamamlanma zamanını minimize etmek için seri halde yer alan m tane makinede n tane işin çizelgelenmesini gerektiren klasik permütasyon akış tipi çizelgeleme problemlerini dikkate almışlardır. Teslim zamanları, hazırlık zamanları ve gecikme zamanlarıyla sözde iki makine akış tipi problemini temel alan bir alt sınır prosedürü ile bir dal sınır algoritması tanımlamışlardır. 8000 operasyona kadar işlemli rasgele örneklerin ve 2000 operasyona kadar işlemli iyi bilinen kıyas örneklerinin kapsamlı bilgisayar sonuçlarıyla, önerilen algoritmanın büyük ölçekli örnekleri makul CPU zamanlarında çözdüğünü göstermişlerdir. Özellikle bir süre için açık olan kıyas problemleri için kanıtlanmış optimal çözümleri rapor etmişlerdir.

Chandramouli (2005), göreceli önemlerini, taşıma sürelerini ve makinelerin verilen boşta bekleme sürelerini göstermek için ağırlıklarıyla işlerin bağlandığı n sayıda iş ve 3 makinede akış tipi çizelgeleme problemi için basit yeni bir sezgisel algoritma sunmuştur. Problem için toplam ağırlıklı ortalama üretim akış zamanını minimize eden dizilimi ya da buna en yakın dizilimi bulmak için bu sezgisel yaklaşım metodunu önermiştir. Son olarak, 4 iş ve 3 makine için bir nümerik örnek yardımıyla metodu anlatmıştır.

Allahverdi (2006), toplam tamamlanma zamanını minimize etme amacıyla hazırlık zamanlarının işlem zamanlarından ayrı düşünüldüğü iki makine akış tipi çizelgeleme problemini tarif etmiştir. Her iki makinede de tüm hazırlık ve işlem zamanları için, her işin hazırlık ve işlem zamanlarında sadece alt ve üst sınırların bilindiğini yani bu zamanların gerçekleşmesinden önce bilinmeyen değişkenler

olduğunu ifade etmiştir. Böyle bir ortamda, mümkün tüm işlem ve hazırlık zamanlarının gerçekleşmesi için optimal tek bir çizelge var olamayabileceğini ve bu yüzden tüm diğer çizelgelere baskın olan bir baskın çizelgeler kümesi gözlemlenmesi gerektiğini ifade etmiştir. Amacın buradaki baskın çizelgeler kümesinin boyutunu azaltmak olduğundan bahsetmiştir. İlgilendiği problem için iki üstünlük ilişkisi geliştirmiştir. Açıklayıcı nümerik örnekler vermiştir ve rasgele üretilen problemlerde bilgisayarlı deneyler yapmıştır. Yaptığı deneylerin sonucuna göre ise, geliştirdiği üstünlük ilişkilerinin baskın çizelgeler kümesinin boyutunu azaltmada oldukça verimli olduğunu göstermiştir.

Seda (2007), permütasyon akış tipi çizelgeleme ve atölye tipi çizelgeleme problemleri için matematiksel modeller önermiştir. İlk problemi, bir karışık tamsayı programlama modeline dayalıdır. Bu problem NP-tam (NP-complete) olduğu için, model sadece optimum çözümün hesaplanabileceği küçük örneklerde kullanılabilmektedir. Büyük örnekler için, stokastik sezgisel metotlar tarafından problemin çözümü için uygun olan bir başka model önermiştir. Atölye çizelgeleme problemleri için de, matematiksel bir model ve onun ana temsil planını sunmuştur.

Jin, Gupta, Song ve Wu (2008), tamamlanma zamanı dağılımlı permütasyon akış tipi çizelgelerin hemen hemen 50 yıldır tartışma konusu olduğundan, birçok araştırmacının bu yüzden tamamlanma zamanı dağılımlı permütasyon akış tipi çizelgelerin işlerin sayısı yeterince büyükse asimtotik normal olduğunun meşhur iddiasını onaylamakta ya da şüpheye düşmekte kararsız kaldığından bahsetmiştir. Çalışmada teorik ve ampirik olarak tamamlanma zamanı dağılımlı permütasyon akış tipi çizelgelemeyi araştırmışlar ve makine ve iş hakim akış tipleri için bu iddianın doğru olmadığını göstermişlerdir. Tamamlanma zamanı dağılımlı permütasyon akış tipi çizelgelerinin olağanlığı çatısındaki hataları belirtmişlerdir. Bu hataların da problemlerdeki makine sayısına olduğu kadar iş sayısına da bağlı olduğunu göstermiştir.

Fan ve Winley (2008), her makinede aynı olması için sınırlandırılmış iş dizisinin sağlandığı makineler ve işlerin isteğe bağlı bir sayısı ile akış tipi problemler için optimal bir çözümü garanti eden yeni bir zeki arama sezgiselinin (IHSA) gelişimini tanımlamışlardır. Gelişimi, IHSA'nın ilk versiyonuna yaptıkları 3 modifikasyon sayesinde tanımlamışlardır. İlk modifikasyon, geçerli bir sezgisel fonksiyon seçimiyle ilgilidir, ikincisi optimal bir çözüm ilerlemesi için arama olarak sezgisel tahmin hesaplamasıyla ilgilidir ve üçüncüsü, var oldukları zaman çoklu optimal çözümleri

karara bağlar. İlk iki modifikasyon algoritmanın karakteristik performansını iyileştirmektedir ve bu iyileştirmenin deneysel delillerini, IHSA'nın ilk ve son versiyonlarının kullanımını gösteren öğretici örneklerle birlikte sunmuşlardır.

Allahverdi (2008), işlem zamanlarından ayrı olarak ve sıra bağımsız olarak düşünülen hazırlık zamanlarındaki n sayıda iş ve üç makine akış tipi çizelgeleme problemini tarif etmiştir. Hazırlık ve işlem zamanlarının oluş zamanından önce tam olarak bilinemeyen değişkenler olduğundan ancak her hazırlık ve işlem zamanı için bir alt ve üst sınır verildiğinden söz etmiştir. Bu problem için, hazırlık ve işlem sürelerinin mümkün tüm gerçekleştirmeleri için optimal kalan tek bir çizelge var olmayabileceğinden bahsetmiştir. Bu yüzden de mümkünse diğer tüm çizelgelere baskın olan bir çizelgeler kümesi gözlemek istemiştir. Amacı da, bu baskın kümenin boyutunu azaltmak olmuştur. Bu kümenin boyutunu azaltmanın bir yolunun üstünlük ilişkileri ortaya atmak olduğunu söylemiştir. Bu çalışmada problem için yerel ve küresel üstünlük ilişkileri geliştirmiştir. Dahası, kümenin boyutunun azaltılması için geliştirilen üstünlük ilişkilerinin kullanımını bir örnek yardımıyla göstermiş ve rasgele üretilmiş problemlere bilgisayar analizleri uygulamıştır. Sonuçlarla da geliştirdiği üstünlük ilişkilerinin etkinliğini göstermiştir.

Vallada ve Ruiz (2009) çalışmalarında, permütasyon akış tipi çizelgeleme problemi için tamamlanma zamanı ve toplam gecikmeyi ayrı ayrı dikkate alan kooperatif bir metasezgisel yöntem önermişlerdir. Ada modelini kullanmışlar ve tümüyle karşılaştırılabilir senaryolarda çalışan ardışık eşlere karşı bu kooperatif metotların kapsamlı bir karşılaştırmasını oluşturmuşlardır. İstatistiki prosedürler sayesinde sonuçları dikkatlice analiz etmişler ve kooperatif metotların ardışık algoritmalarından daha iyi sonuçlar verdiği sonucuna varmışlardır. Son olarak, bu kooperatif tasarıların diğer algoritmalar ve problemler için de kolayca uygulanabileceğinden bahsetmişlerdir.

Azizi, Liang ve Zolfaghari (2009), son yıllarda kombinatoriyal problemleri çözmek için birçok etkili algoritma sunulduğundan ancak problemlerin değişen doğasından dolayı bu konvansiyonel tekniklerin etkinliklerinin sınırlandığından bahsetmişlerdir. Bu yüzden bu çalışmada kombinatoriyal optimizasyon problemleriyle mücadele için genel bir çerçeve (SAMED) sunmuşlardır. Bu çerçeveye dayanarak, akış tipi çizelgeleme için yeni bir algoritma (SAMED-FSS) tasarlamışlar ve geliştirmişlerdir.

Önerdikleri metodun performansını, konvansiyonel tavlama benzetimi, standart genetik algoritma ve hibrit genetik algoritma gibi tekniklerle karşılaştırmışlardır. Hesaplanan sonuçlar ise, önerilen metodun konvansiyonel sezgisellerden açıkça çok daha etkili olduğunu göstermiştir.

Cihanlı (2010), iki aşamalı karma akış tipi üretim ortamında çizelgeleme problemini ele almıştır. Problemden özdeş olmayan paralel tezgâhlar vardır, işler sadece belirlenen tezgâhta işlenebilmektedir ve işlerin sıralamaya bağlı hazırlık süreleri vardır. Problemden amacı, en büyük tamamlanma zamanının minimizasyonu olarak belirlemiştir. Çözüme yönelik bir matematiksel model geliştirmiş daha sonra modeli tamamlanma zamanını kısaltmak için alt partilere bölerek iyileştirmiştir. Önerilen modeller için en uygun çözümleri ‘matematiksel programlama dili’ olan MPL ile sağlamış, her iki modelin çözüm değerlerini ve performanslarını karşılaştırmıştır.

Pandian ve Rajendran (2010), 3 makinede işlerin ağırlıklarını, bölünen zamanları ve ulaştırma zamanlarını içeren akış tipi çizelgeleme problemleri için optimal bir çizelgeleme sırası gözlemek adına yeni bir metod önermişlerdir. Önerdikleri metod, oldukça basit ve anlaması kolay olup aynı zamanda 3 makine ile kısıtlı akış tipi çizelgeleme problemleri için çizelge dizayn ederken karar vericiler için önemli bir araç sağlamıştır. Önerdikleri algoritmayı nümerik örnekler sayesinde göstermiş ve sunmuşlardır.

Uruk (2011), iki makineli akış tipi çizelgeleme problemini ele almıştır. İşler için ise belli kriterler vardır; her bir iş üç operasyondan oluşmakta, ilk operasyon sadece birinci makinede işlem görebilir, ikinci operasyon sadece ikinci makinede işlem görebilir, üçüncü operasyon ise esnek operasyon olup her iki makinede de işlenebilir ancak işler bölünemez. Problemden imalat maliyetini ve tamamlanma süresini en aza indirmeyi çift kriterli amaç fonksiyonu olarak belirlemiştir. Ortaya çıkan problem doğrusal olmayan karışık tam sayılı matematiksel bir model olmuştur, bu nedenle hesaplama zamanı açısından verimli olmayan durumlar ortaya çıkmıştır. Bu çalışmada böylece, verimli bir yaklaşık algoritma önermiştir.

Demirtaş (2011), m makineli n adet parça üreten bir akış tipi üretim sistemini ele almıştır. Bu sistemde her parça makinelerde sırayla işlenmektedir. Her bir makinenin parça üzerinde yapması gereken belirli işlemler vardır. Bunun yanında önceden belirlenmiş ardışık iki makine, sabit işlemlere ek olarak, gereken bir işlemi

daha yapabilecek kabiliyettedir. Aynı tip parça üretimi yapılan bu sistemde makineler arası stok sınırsız ya da sıfır kapasiteli olabilir. Amaç, yayılma zamanını minimum yapacak şekilde, esnek işlemlerin her bir parça için hangi esnek makinede yapılacağına karar vermektir. Öncelikle problemin optimizasyon modelini yazmış ve optimal çözümün özelliklerini belirlemiştir. Sonra optimizasyon modelinin çözüm üretmesi uzun zaman alınca, optimal çözüm için ispatlanan özelliklerden yararlanarak bir dinamik programlama algoritması geliştirmiştir. Ancak bu algorithmada da verilere bağlı olarak uzun işlem zamanları görülebilmektedir. Bu nedenle, daha kısa sürede kaliteli çözümler üreten sezgisel bir yöntem geliştirmiş, yöntemin etkinliğini deneysel bir çalışma ile göstermiştir. Geliştirdiği bu yöntemler, m makineli ve daha esnek sistemlerin çözümüne yönelik ilk adımı oluşturmuştur.

Laha ve Sapkal (2011), NP-zor olduğu iyi bilinen beklemesiz akış tipi çizelgeleme problemleri için yapısal bir sezgisel önermişlerdir. İşlerin başlangıçtaki sırasını seçmek için darboğaz olan makinelerdeki toplam işlem sürelerini, sıradaki bir işin önceliği varsayımına dayanarak vermişlerdir. Bilgisayarda yaptıkları deney sonuçlarının, özellikle büyük boyutlu problemler için zaman karmaşıklığını etkilemediğini, diğer sezgisellerin üzerinde çözüm kalitesinde önemli bir iyileşme olduğunu gösterdiğini, yaptıkları istatistiki testlerin de önerdikleri metodun sonuçlarının önemini kanıtlamakta olduğunu söylemektedirler.

İKİNCİ BÖLÜM

ÇİZELGELEME

Çizelgeleme günümüz koşullarında değerini her geçen gün katlayan bir işletme planlama şeklidir. İşletmeler üretim çizelgeleri sayesinde kaynaklarını en verimli şekillerde kullanırlar ve ürünlerini en etkin şekilde üretirler. Müşterilerine bu çizelgeler yardımıyla ürün teslim tarihlerini net şekilde verirler. Kısacası üretimin çizelgelenmesi ile planlı bir şekilde hareket edip geleceği kesin bir gözle görürler, yanılmalara, olası fırsat maliyetlerine, müşteri kayıplarına ve diğer beklenmeyen zararlara katlanmak zorunda kalmazlar.

Çizelgeler, açık ve sistemli bir şekilde düzenlenmiş veriler dizisidir ve üretim sürecindeki bir ürünün girdileriyle çıktılarını belirli bir düzen ile görebileceğimiz şekilde hazırlanırlar. Çalışmamız bu belirli düzende oluşturulmuş çizelgelerin üretim çizelgeleme problemlerinde kullanılmasına dair şekillenecektir.

Bu bölümde; çizelgelemeden ve üretim çizelgelemeyle ilgili konulardan bahsedilecek, üretim çizelgeleme problemlerinin sınıflandırılması, problemlerin notasyonu, problemlerdeki öncelik kuralları ve problemlerin performans ölçüm kriterleriyle ilgili gerekli bilgilere ve açıklamalara yer verilecektir.

2.1. ÇİZELGELEMENİN TANIMI VE AMACI

Çizelgeleme bir karar verme süreci olarak, birçok bilgi işleme ortamında olduğu kadar üretim sistemlerinde de kullanımıyla önemli bir rol oynamaktadır. Aynı zamanda ulaştırmada, dağıtım sistemlerinde ve hizmet endüstrilerinin bazı kollarında da çizelgelemeye olan ihtiyaç kendini belli etmektedir. Bu nedenle bu bölümde genel olarak çizelgeleme ve amaçlarından kısaca bahsedilecektir.

2.1.1. Çizelgelemenin Tanımı

Çizelgeleme, bir veya daha fazla performans ölçüsünün optimizasyonu amacıyla aktiviteler için kıt kaynakların bölüştürülmesi ile ilgilidir (Leung, 2004, p.2).

Çizelgeleme, birçok imalat ve hizmet endüstrisinde düzenli bir şekilde kullanılan bir karar verme sürecidir. Verilen zaman periyotları üzerindeki görevler için kaynakların paylaşılması ile ilgilidir. Bir organizasyondaki kaynaklar ve görevler farklı şekillerde olabilmektedir. Bu kaynaklar, işyerindeki makinalar, havaalanındaki pistler, bir yapı sitesindeki ekipler, bir bilgisayar ortamında işlem birimleri ve daha başkaları olabilecektir. Bu görevler ise bir üretim sürecindeki operasyonlar, bir havaalanındaki uçakların iniş ve kalkışları, bir inşaat projesindeki aşamalar, bir bilgisayar programının yapılması ve daha fazlası olabilecektir. Her görev mümkün olan en erken başlama ve bitiş zamanı gibi kesin öncelikli aşamalara sahip olabilir, aynı zamanda amaçlar da farklı şekillerde olabilir. Bir amaç, en son görevin tamamlanma zamanının minimizasyonu olabilirken bir başkası, planlanan kendi bitiş zamanlarından sonra tamamlanan birçok görevin gecikme zamanlarının minimizasyonu da olabilir (Pinedo, 2008, p.1).

Çizelgeler, örneğin bir üretim tesisindeki üretilmeye başlanacak bir ürünün, üretilmesi için gerekli tüm girdilerin tesis şartlarına ve bazı isteklere göre planlanmasında kullanıldığında, o ürün için tüm proses bilgisine sahip olmaktadır. Widmer ve arkadaşları da (2008) bu yüzden çizelgelerin, hangi ürünün ne için, ne zaman, kim tarafından ve hangi ekipmanlarla yapılacağını gösterdiğini ifade etmişlerdir.

2.1.2. Çizelgelemenin Amacı

Çizelgelemenin en temel amacı bir ya da daha fazla hedefi optimize etmektir. Bunu yaparken de mümkün olan en kısa sürede bu işi yapmaktır.

Büyük üretim süreçlerini çizelgelemenin ana amacı, ürünün süreç başlangıcından bitişine kadar olan üretim süresini en az, organizasyonun makine ve işgücü kapasite kullanımını ise en fazla yapmaktır (Todd and Sen, 1997, p.234).

Ekinci (2012) çalışmasında, çizelgeleme problemlerinin üç ana amacının mevcut olduğunu söyler. Bunlardan ilkinin teslim tarihi ile ilgili olduğunu ifade eder; müşteriye ürünü geciktirilmeden teslim edilmelidir. İkinci amaç, akış süreleri ile ilgilidir; işlemlerin akış süreleri minimum yapılmaya çalışılır. Üçüncü amaç ise, iş

merkezlerinin kullanımı ile ilgilidir; makine, teçhizat ve personel açısından her iş merkezinde etkin kullanım amaçlanmaktadır.

Turan (2011) ise çizelgelemenin temel amaçlarını şöyle ifade eder:

- Üretim olanaklarını en verimli şekilde kullanmak
- Müşteri taleplerine olabildiğince çabuk cevap vermek
- İşleri en uygun zamanda bitirerek teslim tarihinde uzamalara ve zaman aşımına yol açmamak
- Yarı mamul envanterini minimuma indirmek
- Fazla mesai çalışmalarını minimuma indirmek

2.2. ÜRETİM ÇİZELGELEME

Üretim çizelgeleme için kısaca bir süreç yönetimi işidir demek doğru olacaktır. Bu sürecin yönetiminde ise faaliyetlerin sıralanması, süreç zamanlarının planlanması, hedeflenen amaçları optimize ederek istenen çözümlerin elde edilmesi gibi işlemler yapılmaktadır. Üretim çizelgelemeyle ilgili daha detaylı açıklamalara ve üretim çizelgeleme problemleri hakkında bilgilere alt başlıklarda yer verilecektir.

2.2.1. Üretim Çizelgelemenin Tanımı Ve Amacı

Çizelgeleme, matematiksel teknikler veya sezgisel yöntemler kullanarak sınırlı kaynakların görevlere dağıtımı işlemidir. Üretim çizelgeleme ise, iş planlarında bulunan faaliyetlerin ve atölyelerin üretim süreçlerine en uygun şekilde dağıtılmasıdır (Turan, 2011:14).

Taşgetiren (1999) üretim çizelgelemeyi, bir ürünü oluşturan iş parçalarının eldeki tek veya çok sayıda makinelerde hangi sırada ve ne zaman işlemek gerektiğini belirlemek olarak tarif etmiştir.

Bir başka tanıma göre üretim çizelgeleme, eldeki üretim imkânlarının istenen ürünlerin üretimi için belirli bir zaman içindeki kullanımının bazı performans ölçütlerini karşılayacak şekilde planlanıp düzenlenmesidir (Roadammer ve White, 1988, p. 842).

Aladağ (2010) ise üretim çizelgelemeyi, bir ürünü oluşturan iş parçalarının eldeki tek veya çok sayıdaki makinelerde hangi sırada ve ne zaman işleneceğinin saptanması olarak tanımlamıştır. Üretim çizelgeleme problemlerinin de üretim tiplerine bağlı olarak çok farklı biçimlerde olabileceğini ifade etmiştir.

Bitran ve Tirupati (1989)'ye göre üretim çizelgeleme, ürün veya hizmet üretimi için kaynakların tahsisi ve işlerin sıralanması ile ilgilendir. Tahsis ve sıralama kararları birbiri ile yakından alakalı olsa da aralarındaki ilişkiyi matematiksel olarak modellemek oldukça zordur.

Üretim çizelgeleme için kısaca aşağıda verilen sorularla ilgilenmektedir demek yeterli olacaktır (Brucker, 2007):

- Hangi iş merkezi hangi işi yapacak?
- Bir işlem/iş ne zaman başlayacak ne zaman bitecek?
- İş hangi donanımla, kim tarafından yapılacaktır?
- İşlemlerin/işlerin sıralaması ne olacaktır?

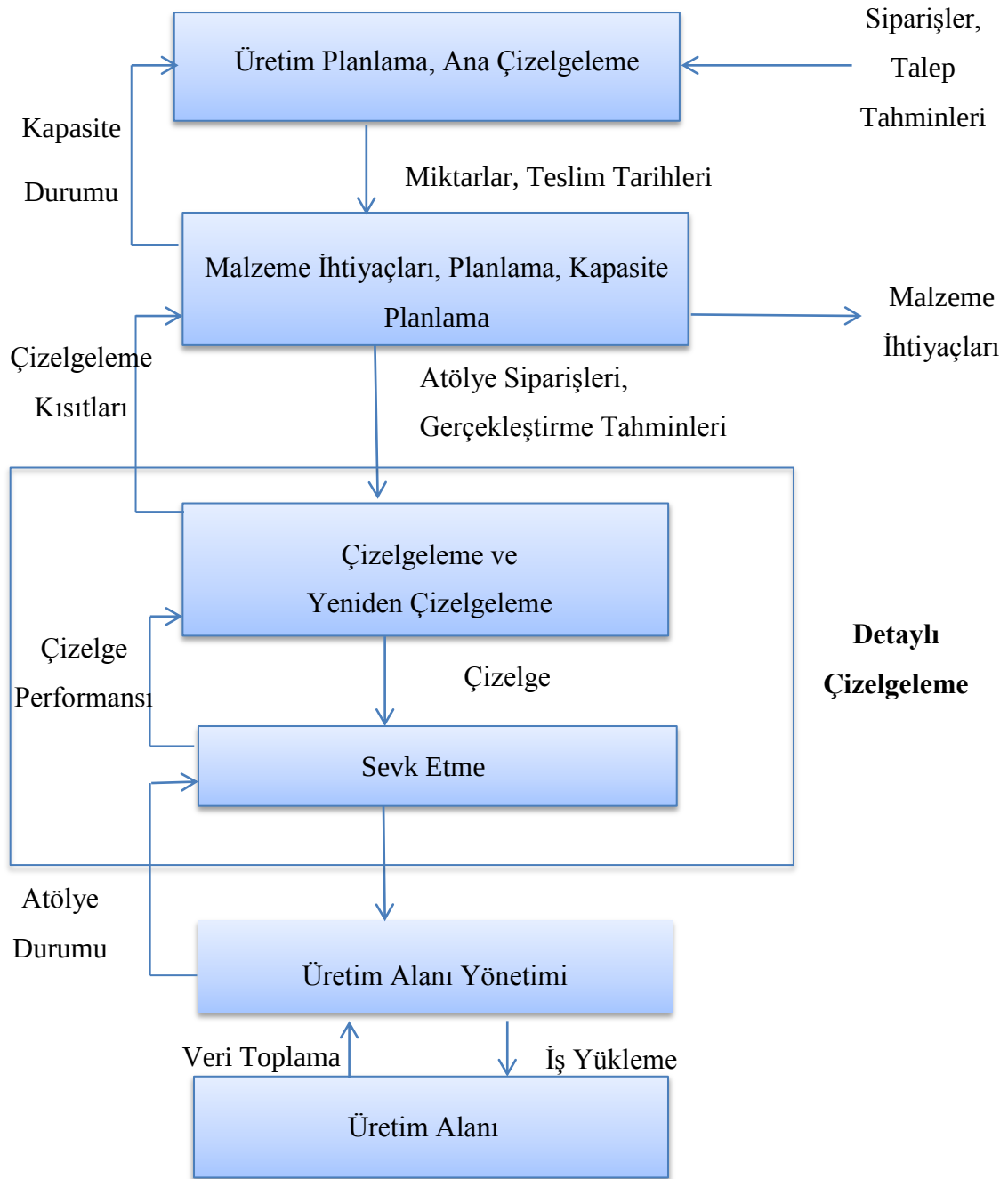
Herrnann (2007) üretim tesislerinin karmaşık, dinamik ve rassal sistemler olduğundan bahsetmiştir. Üretim tesislerinin bu özellikleri nedeniyle, planlı bir üretim faaliyetinin başından itibaren, işçiler, yöneticiler, mühendisler ve üst yöneticiler, üretim faaliyetlerini kontrol altında tutabilmek için pratik ve akıllıca yöntemler geliştirmektedirler. Birçok imalat organizasyonu, kontrol edilebilen belirli aktivitelerin yerine getirilebilmesi için, üretim çizelgeleri hazırlamakta ve bunları sürekli güncellemektedir. Üretim çizelgeleri ise, faaliyetleri koordine ederken, üretkenliği artırmakta ve işletme maliyetlerini düşürmektedir. Bir üretim çizelgesi kaynak çatışmalarını tanımlar, işlerin atölyeye gelişlerini kontrol eder, ihtiyaç duyulan hammaddenin zamanında sipariş edilmesini sağlar, sipariş teminatlarının karşılanıp karşılanamayacağına karar verir ve önleyici bakım için gerekli zaman aralıklarının belirlenmesini sağlar.

Üretim çizelgelemede kurallar genel olarak şu amaçlara ulaşmayı hedeflemektedir (Sarıkaya, 2005:3):

- Gecikme sürelerinin minimizasyonu
- Geciken iş sayısının minimizasyonu
- Ara stok düzeylerinin minimizasyonu
- Akış zamanlarının minimizasyonu
- Makine kullanım oranlarının maksimizasyonu

- Hazırlık sürelerinin minimizasyonu

Üretim çizelgeleme, verimlilik ve etkinlik sağlamak açısından önemli bir işletme fonksiyonudur. Bir işletmede üretim sisteminde bilgi akışı sürecinde çizelgeleme fonksiyonunun yeri Şekil 1’de verilmiştir.



Şekil 1: Bir İşletmede Çizelgeleme Fonksiyonunun Yeri

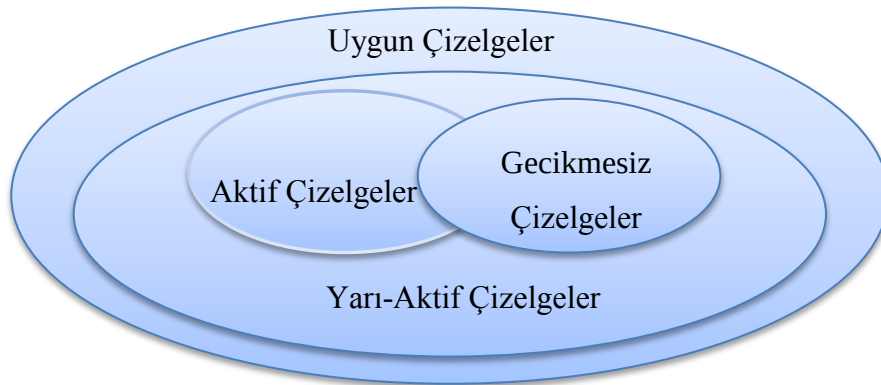
Kaynak: Pinedo, 2008, p. 5.

Üretim çizelgeleme fonksiyonu, üretim sisteminde siparişlerin yani müşteri taleplerinin karşılanması süreci açısından önemlidir. Üretim planlama verileri ve atölye verileri arasında çizelgeleme fonksiyonu köprü vazifesi görür. Üretim planlamadan gelen sipariş bilgilerine ve üretim alanı durumuna göre en iyi ve etkin çizelgenin belirlenmesini ve üretim planlama ile atölye yöntemine geri beslemesini sağlar (Aydemir, 2009:6).

2.2.2. Üretim Çizelgeleme Tipleri

Uygun (feasible) çizelgeler arasında, optimallik konusunda kolayca daha derin çalışmalar yapabilmek için üç belirgin tip çizelge tanımlanmıştır. Bu çizelgeler;

- **Yarı aktif çizelge (semi-active):** Yarı aktif çizelge, işlerin mümkün olabildiğince erken sıralandığı, çizelgelendiği tiptir. Yarı aktif bir çizelgede, hiçbir iş, işlem sıraları değiştirilmeden daha önce başlayamaz. Bir yarı aktif çizelgeler kümesi, bir aktif çizelgeler alt kümesi bir de gecikmesiz çizelgeler alt kümesi içerir.
- **Aktif çizelge (active):** Aktif çizelgelerde, hiçbir faaliyet diğer faaliyeti geciktirmeden veya öncelik kısıtlarını ihlal etmeden daha önce başlayamaz. Bir optimal çizelge ise aktif çizelgeler kümesine aittir.
- **Gecikmesiz çizelge (non-delay):** Gecikmesiz çizelgede, hiçbir makine boş, atıl değildir yani en azından makine üzerinde bir işlem gerçekleştirilmektedir. Optimal çözüm için ise kesin olarak bir gecikmesiz çizelgedir diyemeyiz.



Şekil 2: Çizelge Tipleri Arası İlişki

Kaynak: Lukaszewicz, 2005:15.

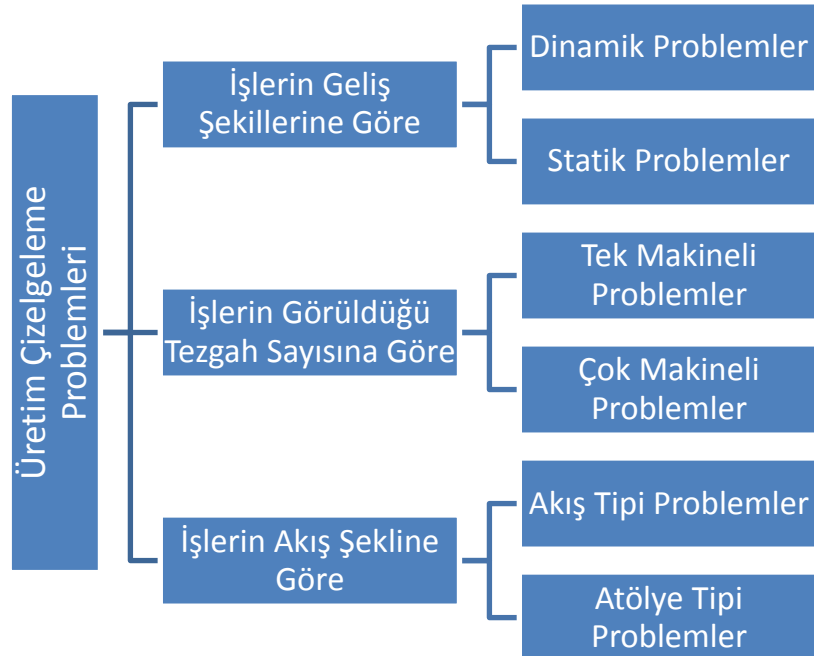
2.3. ÜRETİM ÇİZELGELEME PROBLEMLERİ

Üretim çizelgeleme problemleri, işletmeden işletmeye, tesisten tesise değiştiği gibi, ürüne göre ve zamana göre de değişebilmektedir. Bir ürünün üretilmesi esnasında ürünün işlem sayısının etkisinin olduğu gibi, bu işlemlerin sırasının da etkisi olabilir, veya makinelerin sayısının ve o ürün için makinelerdeki sırasının da çizelgelemede önemi olabilir. Bu nedenle üretim çizelgeleme problemleri farklı sınıflandırmalara tabi tutulmuş ve bu şekilde ayrı ayrı incelenmiştir.

Bu başlık altında ise, üretim çizelgeleme problemleri önce sınıflandırılacak sonra parametrelerinden bahsedilecektir daha sonra ise öncelik (sevk etme) kuralları ve performans ölçüm kriterlerinden söz edilecektir.

2.3.1. Üretim Çizelgeleme Problemlerinin Sınıflandırılması

Üretim çizelgeleme problemlerini; işlerin geliş şekline göre, işlerin görüldüğü tezgâh sayısına göre ve işlerin akışına göre üç grup olarak sınıflandırabiliriz. Bu üç grubu da kendi içlerinde ikiye ayırarak üretim çizelgeleme problemlerini inceleyebiliriz (Şekil 3).



Şekil 3: Üretim Çizelgeleme Problemlerinin Sınıflandırılması

İşlerin geliş şekillerine göre yaptığımız sınıflandırmada problemler ya dinamiktir ya da statiktir. Yani ya düzensiz bir zaman periyodunda, değişen özelliklerde siparişlerin gelmesi ile oluşan dinamik problemler ya da sabit aralıklarla belirli bir

düzenle gelen benzer veya aynı özelliklere sahip siparişlerin gelmesi ile oluşan statik problemler söz konusudur.

Bolat (2004), dinamik ve statik problemlerin genel özelliklerini maddeler halinde sıralamıştır, buna göre dinamik problemlerde;

- İşler düzensiz aralıklarla atölyeye gelir,
- Çok amaç söz konusudur (aylak zaman veya kuyrukta bekleyen işlerin minimizasyonu, erken ya da tam zamanında teslim vb.),
- Herhangi bir tam optimizasyon yöntemi yoktur, simülasyon teknikleri yada öncelik kurallarından en iyi sonucu veren kullanılabilir.

Statik problemlerde ise;

- Belirli bir dönem için iş listesi bilinmektedir,
- İşler atölyeye düzenli olarak gelirler,
- Genel amaç, işlerin sırasını bulmaktır,
- Çözümü için dal sınır algoritması ya da lineer programlar uygun olabilir.

İşlerin görüldüğü tezgâh sayısına göre yaptığımız sınıflandırmaya göre ilk olarak tek makine problemlerine baktığımızda, işlerin sıralanmasıyla bir çizelge belirlenmektedir. Tek makine olduğu için n sayıda işin $n!$ sayıda farklı sıralanabilmesi mümkündür. Ayrıca bu problemlerde işler arası hazırlık sürelerinin olmadığı basit bir üretim düşünürsek, işler hangi sırayla yapılırsa yapılsın tüm işleri bitirmek için gereken tamamlanma süresi (makespan) değişmez.

Tek makine problemleri, en basit sıralama problemi olup elimizde tek kaynak (makine) ve tümü bilinen işlem zamanları vardır. Fakat tek makine problemleri günümüzde de halen önem arz etmektedir. Çünkü karmaşık yapıdaki birçok çizelgeleme problemini çözmek için problemi bazen tek makine gibi düşünmek gerekebilir, bazen de karmaşık problemler iç içe girmiş tek makine problemlerinden oluşmuş olabilmektedir (Baker ve Trietsch, 2009, p. 10).

Tek makine problemlerini $nx1$ olarak çok makineli problemleri de nxm olarak gösterebiliriz. Çok makineli problemleri de aslında kendi içinde, seri makineler ve paralel makineler şeklinde iki gruba ayırabiliriz. Bu ayırmadaki farklılığı da şöyle ifade

edebiliriz; seri makineler farklı işlemleri yapan tezgâhlardan, paralel makineler ise aynı işlemi yapan tezgâhlardan oluşurlar.

İşlerin akış şekillerine göre yaptığımız sınıflandırmaya göre de akış tipi ve atölye tipi problemlerden bahsetmek söz konusudur. Kısaca akış tipi problemler için, tüm işlerin aynı sırada ve aynı makinelerden geçtiği ve sürekli tip üretim yapan, büyük miktarlarda ürün üreten işletmelerde sık karşılaşılan bir çizelgeleme problemi diyebiliriz. Atölye tipi problemler için ise, tüm işlerin farklı makine sıralamalarına sahip olduğu bazı işlerin belki de bazı makinelere uğramadığı, genellikle sipariş tipi, proje tipi üretim yapan, tek seferde nispeten daha düşük miktarlarda ürün üreten işletmelerde sık karşılaşılan bir çizelgeleme problemi diyebilmekteyiz. Ayrıca hem akış tipi hem de atölye tipi problemlerin çözüm karmaşıklığı açısından çözümü en zor olan NP-Zor sınıfta yer aldığını söyleyebiliriz.

Akış tipi çizelgeleme problemleri için ileride daha detaylı açıklamalara yer verileceğinden ve uygulama örneklerine başvurulacağından bu kısımda üzerinde kısa durulmuştur.

2.3.2. Üretim Çizelgeleme Problemlerinin Parametreleri

Üretim çizelgeleme problemlerini ifade etmek için birtakım tanımlamalara ve gösterim şekillerine ihtiyacımız vardır. Bu bölümde de bu tanımlama ve ifadelere yer verilecektir.

2.3.2.1. Problemlerin Notasyonu

Üretim çizelgeleme problemlerinin tümünde yapılması gereken işler kümesi ve bunun için gerekli kaynaklar kümesi (makineler vs.) bulunmaktadır. Bu yüzden problemleri çözerken, bu kümelerin elemanlarının çizelgeleme problemi cinsinden ifadesi gereklidir. Örneğin iş ve makine sayıları, işlerin işlem ve varsa hazırlık ile teslim süreleri ve problemle ilgili özel kısıtlar tanım ve notasyon olarak problemde verilir. Bu tanım ve notasyonlar şu şekildedir (Pinedo, 2008, p. 13-14):

i : m tane makinenin yer aldığı makineler kümesini $i=(1,2,...,m)$

j : n tane işin yer aldığı işler kümesini $j=(1,2,...,n)$ temsilen;

İşlem Süresi (p_{ij}): j . işin i . makinedeki işlem süresini ifade eder. İşlem süresinin makinelere bağlı olmadığı özdeş makineli durumlarda p_j olarak ifade edilebilir.

Hazır Olma Zamanı (r_j): j . işin sisteme geliş zamanını yani o iş için en erken başlama zamanını ifade eder.

Teslim Zamanı (d_j): j . işin müşteriye teslim edileceği tarihi ifade eder. Gecikmelere kesinlikle izin verilmiyorsa işlerin tam zamanında teslim edilmesi isteniyorsa bu son tarih ‘deadline’ olarak adlandırılır.

Müsaade Edilen Zaman ($a_j: d_j - r_j$): j . işin teslim edilmesi gerekli zamanı ile hazır olma zamanı arasındaki işlem görmesi için serbest bırakılmış olduğu zamandır (French, 1982, p. 10).

Ağırlık (w_j): j . işin diğer işlere göre ağırlığının veya bir önceliğinin olduğu durumlarda kullanılan ifadedir. Örnek olarak j işinin sistemde bulunma maliyeti ölçütü için ağırlıklandırma yapılabilir.

Tamamlanma Zamanı (C_j): j . işin tamamlanmış ve teslim hazır olduğu zamanı ifade eder.

Akış Zamanı ($F_j: C_j - r_j$): j . iş için akış zamanı; o işin tamamlanma zamanından o işin hazır olma zamanının çıkarılmasıyla elde edilir.

Gecikme Zamanı ($L_j: C_j - d_j$): j . iş için gecikme zamanı; o işin tamamlanma zamanından o işin teslim zamanının çıkarılmasıyla elde edilir. Bir iş teslim tarihinden önce tamamlanırsa o işin gecikme zamanı negatif değer olacaktır.

Bekleme Zamanı (W_{ij}): j . işin i . makineyi bekleme süresini ifade eder.

Geç Kalma Zamanı [$T_j: \max (C_j - d_j, 0)$]: İşlerin gecikme zamanları arasında en büyüğünü bize verir. Bu değer negatif olamaz (Benli, 2011:1).

Erken Bitirme Süresi [$E_j: \max (d_j - C_j, 0)$]: İşlerin erken bitirme süreleri arasında en büyüğünü bize verir.

2.3.2.2. Problemlerin Gösterimi

Herhangi bir üretim çizelgeleme problemini ifade etmek için $\alpha|\beta|\gamma$ şeklindeki standart bir gösterim kullanılmaktadır. Bu gösterimde α ile gösterilen bölümde

makinelerin bulunduğu üretim ortamı tarif edilir. β ile gösterilen bölümde işlem özellikleri ve kısıtlarla ilgili bilgiler yer alır. Son bölüm olan γ ile gösterilen alanda ise optimize edilmeye çalışılan amaç bilgisi yani performans ölçütü bilgisi yer almaktadır. Bu aşamada problemleri ifade edebilmek için $\alpha|\beta|\gamma$ gösteriminde ilgili alanlara yazılabilecek bilgiler şöyledir (Pinedo, 2008, p. 14-17):

α Bölümü Parametreleri:

Tek Makine (1): En kolay problem sınıfı olan tek makine problemini tanımlar.

Paralel Makine (Pm): Tek kademeli olarak m adet paralel özdeş makinenin bulunduğu durumu tanımlar, makineler özdeş değilse (Qm) olarak gösterilir.

Akış Tipi (Fm): Tüm işlerin aynı rotayı takip ettiği ve m adet makineden aynı sırayla geçtiği üretim durumunu tanımlar.

Atölye Tipi (Jm): m adet makinenin bulunduğu ve her işin kendilerine özgü rotalarıyla farklı makinelerde gezerek üretilebildiği üretim durumunu tanımlar.

β Bölümü Parametreleri:

Hazır Olma Zamanı (rj): İşlerin işe başlama zamanlarının olduğu durumu tanımlar, bu sembol kullanılmadığında ise işler sıfır zamanında başlamaya hazırdır.

Sıra Bağımlı Hazırlık Süresi (sjk): j . ve k . işlerin arka arkaya yapılması durumunda iki iş arasında yapılacak hazırlık süresini tanımlar.

Serbestlik ($prmp$): Bir işin başlangıcından bitişine kadar aynı makinede yapılması şartının olmadığı durumu tanımlar. Bu durumda, işlem devam ederken başka makinelere geçilip kalan işlemlerin o makinelerde tamamlanabileceği serbestliği vardır.

Öncelik ($prec$): Bir işin yapılmasından önce başka bir işin yapılmış olması gerektiği durumlarını tanımlar. Bu durumda, işler arası öncelik sonralık ilişkileri mevcuttur.

Permütasyon ($prmu$): Akış tipi sistemde, her makinedeki sıralamanın FIFO (ilk giren ilk çıkar) disiplinine göre işlenmesi şartının koyulduğu durumdur. Böylece işler de, ilk makineden itibaren sistem boyunca bu sırada akışla geçer.

Beklemesiz (*nwt*): Akış tipi sistemde karşılaşılan bu durum, ardışık iki makine arasında işlerin hiç beklemeden devam etmesini sağlamak gerektiği durumunu ifade eder.

Bloklama (*block*): Akış tipi bir sistemde karşılaşılan bu parametre, bir birini takip eden iki makine arasındaki kuyruğun sınırlı bir kapasiteye sahip ve dolu olduğu zaman önceki makinenin işlemini bitirdiği işi sonraki makineye gönderemeyeceğini ifade eder. Bu durum bloklama olarak bilinir (Kellegöz, 2006:6).

Arıza (*brkdown*): Sistemdeki makinelerin arızalanma durumlarını ifade eder, makineler sürekli olarak çalışmaya devam edebilir nitelikte değildir.

Döngüsellik (*rcrc*): İşlerin yapılma rotalarında bir makineden birden çok kez geçme durumlarını ifade eder.

Eş Süreler ($p_j=p$, $d_j=d$): İşlem sürelerinin veya teslim sürelerinin tamamının tüm işlerde aynı olduğu durumu ifade eder.

γ Bölümü Parametreleri:

Optimize edilmek istenen amaç yani performans ölçütümüz bu bölümde ifade edilir. Birden çok amaç var ise birden çok sembolle gösterilerek problemde dikkate alınacak ölçüt gösterilir. Performans ölçütlerine sonraki bir bölümde değinilecektir.

2.3.3. Üretim Çizelgeleme Problemlerinde Öncelik Kuralları

Üretim çizelgeleme problemlerinde işleri sıralarken bazı kurallar dikkate alınır. Adına sevk etme ve/veya öncelik kuralları dediğimiz bu kurallar, çizelgeleme problemlerinde sıkça kullanılırlar. Öncelik kurallarına göre bir işin makinelerdeki operasyon sırası değişebilir, böylece sıralama kurala göre düzenlenmiş olarak istenen amaç optimize edilmiş olur. Genel olarak kullanılan bu on öncelik kuralı şunlardır:

- **İlk gelen ilk servis alır (FCFS, First Come First Served):** Çizelgelemede sisteme ilk gelen işin ilk önce çizelgelenmesi anlamındaki öncelik kuralıdır. Böyle bir sıralama siparişlerin geliş sıralarına göre servis görmeleri anlamına gelir. Sistem birinci sipariş ile başlar ve son siparişe kadar sırasıyla çizelgelenir.
- **Son gelen ilk servis alır (LCFS, Last Come First Served):** Çizelgelemede sisteme son gelen işin ilk önce çizelgelenmesi anlamındaki

öncelik kuralıdır. Böyle bir sıralama siparişlerin geliş sıralarına dikkate alarak son sipariştten başlayarak ilk siparişe kadar işler sırasıyla çizelgelenir.

- **En kısa işlem süresi (SPT, Shortest Processing Time):** En kısa işlem süresi olan işlemi seçmeye yönelik öncelik kuralı ile toplam akış süresinin minimum olması amaçlanır. n tane iş/sipariş m tane makinede minimum tamamlanma zamanına göre çizelgelenmek istendiğinde her bir işin/siparişin ilk işlemlerinden en kısa olanı SPT olarak seçilerek çizelgelemeye başlanır.
- **En uzun işlem süresi (LPT, Longest Processing Time):** En uzun işlem süreli olan işlerin en erken işlenmesine yönelik öncelik kuralı LPT, genellikle paralel makinelerde yük dengeleme için yapılan çizelgeleme problemlerinde kullanılır. Toplam tamamlanma süresinin minimizasyonu, n tane iş m tane makinede minimum tamamlanma zamanına göre çizelgelenmek istendiğinde her bir işin ilk işlemlerinden en uzun olanı LPT olarak seçilir.
- **En erken teslim süresi (EDD, Earliest Due Date):** En Erken Teslim Zamanı (EDD) Öncelik Kuralı; en erken teslim süresine sahip olan işlerin önce çizelgelendiği kuraldır. Bu sayede teslim kabiliyeti artmış olur ve dolayısıyla müşteri memnuniyetinin yükselmesi beklenir.
- **En az kalan operasyon sayısı (LRNOP, Least Remained Number of Operation):** En Az Kalan Operasyon Sayısı (LRNOP) Öncelik Kuralı; her bir sipariş için kalan operasyon sayısı en az olan işlerin sistemde daha önce çizelgelenmesi kuralına dayanır. Bu sayede operasyon sayısı az olan işler daha önce sisteme alınır ve toplam sipariş tesliminin daha fazla olması beklenir.
- **En çok kalan operasyon sayısı (MRNOP, Much Remained Number of Operation):** En Çok Kalan Operasyon Sayısı (MRNOP) Öncelik Kuralı; her bir sipariş için kalan operasyon sayısı en çok olan işlerin sistemde daha önce çizelgelenmesi kuralına dayanır. Bu sayede operasyon sayısı fazla olan işler daha önce sisteme alınır ve kapasite kullanım oranının daha yüksek seviyelerde olması beklenir.
- **En kısa kalan işlem süresi (SRPT, Shortest Remaining Processing Time):** En Kısa Kalan İşlem Süresi (SRPT) Öncelik Kuralı; her zaman

diliminde her bir işin kalan operasyonlarının toplam işlem süresi en kısa olan işlemin önce çizelgeleneceğini dikkate alır. SPRT ile yapılan çizelgelemede toplam tamamlanma zamanı ve son işin sistemi terk etme zamanının en küçük olması hedeflenir.

- **En uzun kalan işlem süresi (LRPT, Longest Remaining Processing Time):** En Uzun Kalan İşlem Süresi (LRPT) Öncelik Kuralı; her zaman diliminde her bir işin kalan operasyonlarının toplam işlem süresi en uzun olan işlemin önce çizelgeleneceğini dikkate alır. Bu sayede operasyon süresi fazla olan işler daha önce gerçekleştirilerek kapasite kullanım oranının daha yüksek seviyelerde olması beklenir.
- **Rassal Seçim (RS, Random Selection):** Çizelgelemede sisteme gelen tüm siparişlerin sayısına j dersek; $[1, j]$ aralığından rassal olarak üretilen bir sayı ile çizelgelemeye başlanılacak sipariş belirlenir. Belirlenen siparişin ilk işlemi ile çizelgeleme başlar. Daha sonraki adımlarda zaman boyunca çizelgelenebilecek işlemler kümesinden rassal seçimler yapılarak en son siparişin son işi tamamlanana kadar rassal seçim ile çizelgeleme devam eder (Aydemir, 2009:24-35).

2.3.4. Üretim Çizelgeleme Problemlerinin Performans Ölçüm Kriterleri

Üretim çizelgeleme problemlerinin her birinde farklı amaçları optimize etmek gerekebilir. Problemin türü ya da problem çözücünün özel istekleri bu amaçları seçerken etkili olabilmektedir. Aynı zamanda, bazen tek bir amaç için en iyi çözümü arayabilir bazen de birden çok amaç için en iyi çözümü bulmak isteyebiliriz.

Lukaszewicz (2005) tezinde, optimum çizelgenin birçok faktöre göre belirlenebileceğinden bahsetmiştir. Bunlardan birine örnek olarak, tüm işlerin toplam tamamlanma zamanlarının minimize edilmesinin istendiği bir problemi vermiştir. Genel olarak, en çok kullanılan çizelgeleme problemleri amaç fonksiyonu parametreleri diyebileceğimiz bu performans ölçüm kriterleri ise şöyledir:

- **Maksimum Tamamlanma Zamanı (C_{max}):** Sistemden çıkan en son işin tamamlanma zamanını ifade eder. Makespan olarak da bilinen C_{max} 'ın minimizasyonu, makine kullanım oranlarının en yüksek olmasını ve teslim tarihlerinde işlerin bitirilmesini mümkün hale getirecektir.

- **Maksimum Gecikme (L_{max}):** İşlerin bitiş süreleri ile teslim tarihleri arasındaki sapmaların bütün işler içinde en büyük olanını verir. Bu en yüksek gecikmenin minimizasyonu ile müşterinin teslimatın gecikmesinden kaynaklanan şikâyetlerinin ve bu esnada stok taşıma maliyetlerinin en düşük olması beklenir.
- **Maksimum Erken Bitirme (E_{max}):** İşlerin bitiş süreleri ile teslim tarihleri arasındaki ters sapmalarda yani teslim sürelerinden erken sürede işlerin bitirilmesinde, bütün işler içinde en büyük erken bitirme süresine sahip olanın süresini verir. Bu sürenin minimizasyonu da, yine işlerin zamanında teslimi ve makinelerin kullanım oranları ile ilgilidir.
- **Toplam Ağırlıklı Tamamlanma Zamanı ($\sum w_j C_j$):** Bütün işlerin tamamlanma sürelerinin toplamını ifade eder. Bu toplama “akış süresi” de denmektedir. Eğer öncelik söz konusu ise ağırlıklı tamamlanma sürelerinin toplamıdır. Bu kritere göre minimizasyon, çizelgelemenin oluşturduğu toplam stok maliyetinin en düşük olmasını sağlayacaktır.
- **Toplam Ağırlıklı Geç Bitme Süresi ($\sum w_j T_j$):** Tamamlanma zamanı teslim tarihinden sonra gerçekleşen işlerin bu geç bitme sürelerinin toplamını ifade eder. Yine öncelik söz konusu ise ağırlıklı sapma sürelerinin toplamıdır. Bu kriterin minimizasyonu da, müşterinin teslimatın gecikmesinden kaynaklı şikâyetlerini azaltmaya yardımcı olacaktır.
- **Toplam Ağırlıklı Gecikmiş İşlerin Sayısı ($\sum w_j U_j$):** Tamamlanma zamanı teslim tarihinden sonra gerçekleşen işlerin sayısının toplamıdır. Eğer öncelik varsa, bu tür işlerin ağırlıklı sayısının toplamıdır. Bunun minimizasyonu da yine toplam geç bitme süresinde adı geçen maliyet fonksiyonlarının değerini düşürmektedir (Pinedo, 2008, p. 19).

ÜÇÜNCÜ BÖLÜM

AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİ

Üretim çizelgeleme problemlerinde işlerin üretim sistemindeki akışına göre yapılan sınıflandırmada, iki tür çizelgelemeden bahsedebileceğimize değinmiştik. Bu iki tür; akış tipi ve atölye tipi çizelgeleme olarak karşımıza çıkmaktadır. Akış tipi çizelgelemede kısaca, benzer işler nedeniyle aynı akış sıralamasında işlem gören işlerden bahsetmek mümkün iken; atölye tipi çizelgelemede ise kısaca, farklı işler nedeniyle farklı sıralamalarda hatta farklı makinelerde işlem gören işlerden bahsetmek gerekir.

Akış tipi çizelgeleme problemleri (Flowshop Scheduling Problem, FSP), farklı işlemlere sahip makinelerin seri halde diziliminden oluştuğu ortamlarda üretim yapılmasıyla ortaya çıkabilirler. Makineler arasında olası ara stoklar için ara mamul bekleme yerleri ayrılabilir. İki den fazla makineli üretim ortamlarında ise NP-Tam zorluk düzeyine sahiptir.

Atölye tipi çizelgeleme problemleri (Jobshop Scheduling Problem, JSP), benzer işlemlere sahip paralel şekilde dizilmiş makinelerin olduğu ortamlarda yapılan üretimlerde karşımıza çıkabilmektedirler. Farklı sıralamalı ve farklı makinelerde üretim görebilen işler makinelerle yüklendiği için NP-Zor yapıda olup en zor problemler arasında yer almaktadırlar. Bu nedenle akış tipi problemlere göre ise nispeten daha karmaşık yapıdadırlar.

Bu bölümde, ilgilendiğimiz problem türü olan akış tipi çizelgeleme problemlerinden bahsedilecektir. Önce problemin tanımı ve özellikleri verilecek, daha sonra sırasıyla problemin çeşitleri ve popüler metasezgisel çözüm yaklaşımları açıklanacaktır.

3.1. PROBLEMİN TANIMI VE ÖZELLİKLERİ

Akış tipi iş çizelgeleme, birbirinden farklı m makine n işin bulunduğu; her bir işin belirli sayıda operasyondan oluştuğu, her bir operasyonun farklı makinelerde yapıldığı ve bütün işlerin operasyonlarının aynı sırayla yapıldığı problemlere denir. Basit akış tipi iş çizelgeleme problemlerinde her iş seri makinelerde tam olarak aynı sırayla işlem görmektedir (Akçay, 2009:6).

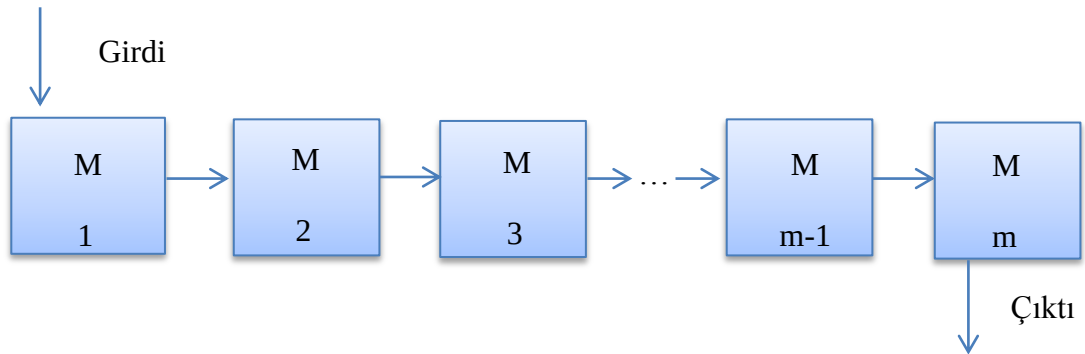
FSP, bütün işlerin birden çok üretim işlem basamağının olduğu ve bütün işlerin bu basamaklardan aynı sırada geçmek zorunda olduğu bir üretim sisteminde karşılaşılan problemlerdir. Diğer bir ifadeyle, n sayıda işin m tane makinede aynı sırayla işlem görmesidir. Bazı işlerin bazı operasyonlarında işlem süreleri sabit ve pozitif veya sıfır olabilir. Yani bir makinede sıfır işlem süreli bir operasyon varsa, o operasyonun o makinede işlem görmeyeceği anlaşılır (Altındaş, 2011:37).

Akış tipi yerleşimde karşılanması gereken temel şartları şöyle sıralayabiliriz (Baker and Trietsch, 2009, p. 226) (Körez, 2005:6):

- Sıfır başlangıç zamanında n tane bağımsız ve çok operasyonlu iş hazırdır. (Her bir iş m tane operasyon ve her bir operasyon farklı bir makine gerektirmektedir.)
- Operasyonların hazırlık zamanları işlem sırasından bağımsız olup işlem sürelerine eklenmiştir.
- İş tanımları ayrıntılı olarak yapılmıştır ve bilinmektedir.
- Tüm makineler sürekli olarak kullanıma hazırdır.
- Bir operasyona başlandığında o operasyon bitene kadar kesinti söz konusu değildir, o işlem bitmeden durulamaz.
- Süreç içi stoğa izin verilir. Herhangi bir iş, çizelgenin bir sonraki adımında ihtiyaç duyduğu makine meşgulse, ilgili makinenin kuyruğuna girerek bekleyebilir.
- Her bir makine tipinden sadece bir tane mevcuttur, süreç içerisinde işin makine seçmesine müsaade edilmez. Ancak bu kabul bazı durumlarda ortadan kaldırılabilir, belirli makineler birden fazla eklenerek darboğaz ortadan kaldırılmaya çalışılabilir.
- Makinaların, boş/aylak süreleri olabilir.

- Olayda rassallık yoktur. Yani; iş sayısı biliniyor ve sabittir, makine sayısı biliniyor ve sabittir, işlem zamanları biliniyor ve sabittir, ihtiyaç duyulacak diğer parametreler baştan beri bilinmektedir.

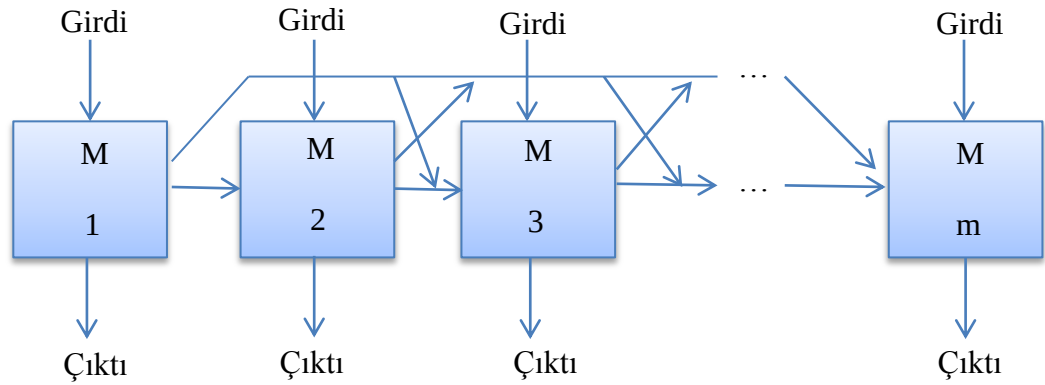
Birçok imalat ve montaj tesisinde her bir işin üzerinde bir dizi operasyonun uygulandığı sıklıkla görülmektedir. Bu operasyonlar tüm işler üzerinde aynı sırayla uygulanırlar ve dolayısıyla işler de aynı rotayı takip eder. Yapılması gereken işler, başlangıç makinesinden başlayarak tamamlanıncaya kadar bir dizi makinede işlem görür ve sonunda son makinede işlemlerini tamamlarlar. Uygulamada ise akış tipi yerleşimin farklı tasarımlarını görmek mümkündür. Örneğin saf akış tipinde, m operasyondan oluşan bir iş ve her bir operasyon için m adet farklı makine bulunmaktadır. Şekil 4’de saf akış tipi yerleşim için küçük bir örnek verilmiştir (Seçme, 2006:22).



Şekil 4: Saf Akış Tipi Yerleşim

Kaynak: Seçme, 2006:22.

Diğer bir tasarım olan genel akış tipi yerleşimde bazı işler m sayıda operasyona ihtiyaç duymayabilirler. Ayrıca işin ihtiyaç duyduğu operasyonlar birbiri ardına gelen makinelerde olmayabilir veya başlangıç ve bitiş operasyonları her bir iş için farklı olabilir. Bütün bu olasılıklara rağmen işlerin akış şekli doğrusaldır ve saf akış tipi yerleşimindeki bazı operasyon zamanları sıfır kabul edilerek genel forma ulaşılabilir. Şekil 5’de genel akış tipi yerleşim verilmiştir (Baker and Trietsch, 2009, p. 226).

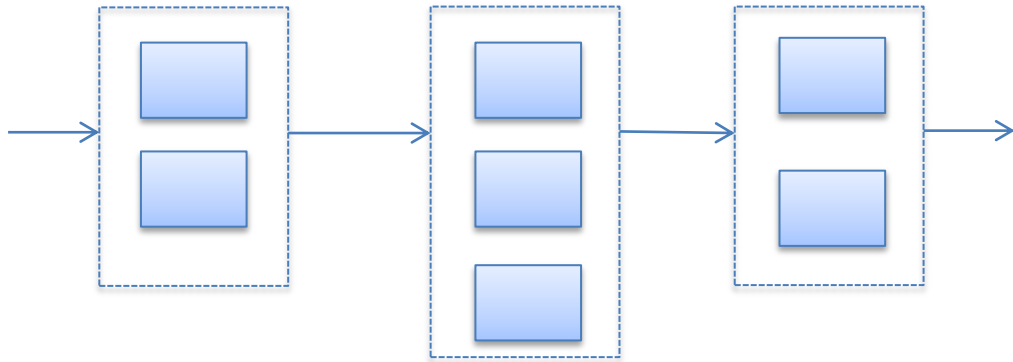


Şekil 5: Genel Akış Tipi Yerleşim

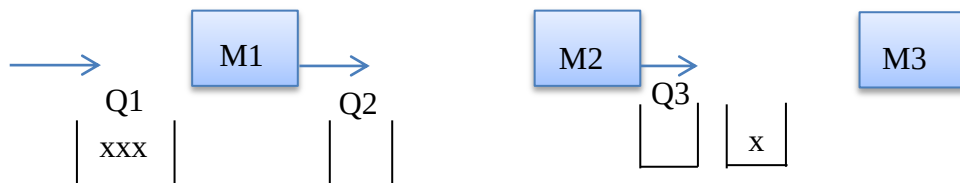
Tekrarlı Akış Yerleşimi



Bileşik/Karışık Akış Yerleşimi



Sonlu Kuyruk Beklemeli Akış Yerleşimi



Şekil 6: Özel Akış Tipi Yerleşim Modelleri

Kaynak: Seçme, 2006:24.

Uygulamada kullanılan bazı özel akış tipi yerleşim modellerine de rastlanılmaktadır. Bunlardan tekrarlı akış yerleşiminde, işler bazı makineleri iki veya daha fazla kez ziyaret edebilirler. Bileşik/karışık akış tipi yerleşiminde ise, birbirine seri halde bağlı makine grupları söz konusudur ve bu makine grupları da kendi aralarında seri ve/veya paralel bağlı olabilirler. Sonlu kuyruk beklemeli akış yerleşiminde de, birbirine seri bağlı makinelerin öncesinde ve sonrasında sınırlı stok alanları bulunmaktadır. Bu durum ise kapasite kısıtlayıcı bir faktör olarak değerlendirilmektedir. Bu özel akış tipi yerleşim modellerinin bir gösterimi Şekil 6'da verilmiştir (Eren, 2001)(Seçme, 2006:24).

3.2. PROBLEMİN ÇEŞİTLERİ

FSP, bir üretim sisteminde n sayıda işin m sayıda makineden geçerek üretimi esnasında işlerin peşpeşe dizili makinelerin tümünde ve aynı sırayla işlem görüp görmemeleri açısından iki gruba ayrılarak incelenmektedir. Bu iki grup; permütasyon akış tipi üretim çizelgeleme problemleri ve esnek akış tipi üretim çizelgeleme problemleridir.

3.2.1. Permütasyon Akış Tipi Üretim Çizelgeleme

Permütasyon akış tipi üretim çizelgeleme problemi (Permutation Flowshop Scheduling Problem, PFSP), akış tipi çizelgeleme problemlerinin özel bir sınıfı olup bu sınıftaki problemlerde bütün makinelerdeki iş sırası aynıdır (yani tüm makinelerdeki kuyruk disiplini ilk gelen ilk işlem görür prensibini ifade eden FIFO'dur). Bu problemlerde uygun bir çözümü gösteren çizelge tek bir permütasyonla tanımlanır. Bu tür çizelgeleme problemleri, işlerin makineden makineye malzeme taşıma sistemleriyle taşındığı üretim sistemlerinde ve montaj hatlarında görülür (Kellegöz, 2006:29).

PFSP'de, iş sıralarının her makinede aynı olduğu permütasyon çizelgeler kullanıldığı için; bu tür çizelgelerde mümkün olan çizelge sayısı $n!$ olmaktadır (Seçme, 2006:25). Böylece $n!$ sıralaması şeklinde elde edeceğimiz dizilimi, problemde istenen performans ölçütlerine göre belirleriz ve her makineden bu sıralamayla geçmesini sağlarız. PFSP için en çok kullanılan performans ölçütü ise, işlerin toplam tamamlanma zamanını (makespan) veren C_{max} 'ın minimize edilmesi olmaktadır. Tamamlanma zamanını minimize eden bir PFSP'yi de $F|prmu|C_{max}$ şeklinde gösterebiliriz.

$Fm/prmu/Cmax$ probleminin Gezin Satıcı Problemine (TSP) olan yapısal benzerliği ilk olarak 1968 yılında Gupta tarafından gösterilmiştir. Problem, n adet şehrin mesafe matrisinin m adet makinedeki işlerin işlem sürelerine benzerliğiyle kurulmaktadır. TSP yapısına dönüştürülerek elde edilen çözüm orijinal akış tipi çizelgeleme probleminin de çözümü olmaktadır. TSP’de amaç en küçük maliyetle şehirlerin hangi sırada dolaşılacağıнын bulunması olduğu için ve problem kesikli optimizasyon problemi olduğundan sonuç bir sayılar dizisi olacaktır. Bu sayılar dizisi ise akış tipi problemimizde işlerin sıralandığı çizelgemiz olacaktır.

İki makineli PFSP için polinom zamanlı optimum çözüm varken, ikiden fazla makine varsa çözüm NP-Zordur. İki makine ve m sayıda makine için geliştirilen yöntemler şöyledir (Altındaş, 2011:37):

- **$F2||Cmax$** : Polinom zamanlı çözümü olan tek akış tipi üretim sistemi problemidir. Johnson tarafından 1954 yılında geliştirilen algoritma ile optimum çözüm elde edilmektedir.
- **$Fm||Cmax$** : Bu çok makineli problemin çözümü NP-Zor olduğu için konuyla ilgili optimum çözüme yakın çözüm veren çok sayıda sezgisel yöntem geliştirilmiştir. Bunlar içinden uzun süredir en iyi olarak kabul gören ve polinom çözüm veren NEH sezgiseli 1983 yılında Nawaz ve arkadaşları tarafından sunulmuştur.

3.2.2. Esnek Akış Tipi Üretim Çizelgeleme

Esnek akış tipi çizelgeleme problemlerinde (Flexible Flowshop Scheduling Problem, FFSP), genel akış tipi yerleşim modelinde de görüldüğü gibi bazı işlerin bazı makinelerde işlem süreleri sıfır olabilir. Bu durumda bazı işler bazı makinelerde işlem görmeyeceği için işlerin makinelerdeki işlem sıraları farklı olabilir. Böylece iş sıralarının makinelerde farklı olduğu durumlar söz konusu olunca, mümkün olan çizelge sayısı da $n!$ sayıda farklı iş sıralaması ve m sayıda makine için $(n!)^m$ sayıda çizelge anlamı taşımaktadır.

İşlerin makinelere göre işlem sıralarının değişmesine izin verilmesi durumu, aynı sırayla işlerin işlem görmesi durumunda oluşturulan permütasyon çizelgelere göre çok daha zor bir durumu ifade etmektedir. NP-Tam olarak nitelendirilen bu iki durum

kendi aralarında karşılaştırıldığında, iş sırasının değişebildiği çizelgelerin permütasyon çizelgelere göre daha zor olduğu söylenebilir (Framinan and Leisten, 2003).

Her ne kadar PFSP için tasarlanmış olsa da Dannenbring algoritması, FFSP için büyük çaplı problemlerde çözüm süresinin kısa olmasından dolayı optimum çözümden ortalama %15 sapmasına rağmen tercih edilmektedir. Dannenbring algoritması, Johnson algoritması üzerine kurulmuştur. Yani m aşamadan oluşan bir akış tipi üretim çizelgesi iki aşamaya bölünmekte ve bu aşamalara Johnson algoritması uygulanmaktadır (Altındaş, 2011:39-40).

3.3. PROBLEMİN ÇÖZÜM YAKLAŞIMLARI

FSP için ilk çözüm yaklaşımı çalışmasını 1954 yılında Johnson yapmıştır. İki makine için optimum çözümü veren basit bir algoritma geliştirmiştir. Daha sonra ise ikiden fazla makine durumları (NP kapsamında olan) için çeşitli sezgiseller geliştirilmiştir. Bu yöntemlerin bazıları şunlardır (Engin, 2001:2):

- Palmer'in Eğim Dizisi Yöntemi (1965)
- Campbell, Dudek ve Smith (CDS) Algoritması (1970)
- Gupta Yöntemi (1971)
- Dannenbring Yöntemi (1972)
- Nawaz, Ensore, Ham (NEH) Yöntemi (1983)
- Hundal ve Rajgopal Yöntemi (1988)
- Widmer ve Hertz Yöntemi (1989)
- Ho ve Chang (HC) Yöntemi (1991)

Bu klasik yaklaşımlar, problemleri çözmedeki uzun işlem süreleri ve problemlere özgü olarak bünyelerinde esneklik barındırmamaları nedenleriyle zamanla cazibesini yitirmeye başlamıştır. Bu nedenlerle, daha az parametre ile daha iyi sonuçlar veren, daha kısa sürede çözüme ulaşan ve problemin tüm özelliklerinin yansıtılabileceği yeni yaklaşımlar (metasezgiseller) geliştirilmeye başlanmıştır. Bu metasezgisellerin temelinde de doğadan esinlenmeler, sürü yaklaşımları ve yapay zeka teknolojileri etkili olmuştur. Son yıllarda ise birçok metasezgisel, hibrit kullanım ile birleştirilerek daha da etkili çözümlere ulaşılmıştır. Üretim çizelgeleme problemleri için son zamanlarda sık olarak başvuru alan metasezgisellere ve algoritmalarından bazılarına çalışmamızın konusunu oluşturdıkları için aşağıda daha detaylı olarak değinilecektir.

3.3.1. Yapay Arı Kolonisi Algoritması

Doğrusal olmayan fonksiyonların optimizasyonunda kullanılan klasik matematiksel metotlar, ispat edilmelerinde yazılım ve donanım açısından bazı zorlukları da beraberinde taşıyan türev ifadeleri içerirler. Optimizasyon hesapları içinde türev ifadelerinden kaçınmak ve nispeten daha basit ve kısa yazılımla, daha kısa sürede sonuca yaklaşmak için ekosistemdeki canlıların doğal seçim davranışlarından esinlenen iterasyona dayalı yeni optimizasyon metotları geliştirilmiştir. Önceleri daha çok mutasyon, çaprazlama ve yeniden üreme gibi evrimsel bir takım operatörler içeren bu metotlar daha sonra yerini, canlıların besin bulma davranışlarından yola çıkan popülasyon tabanlı, sürü zekasına dayalı optimizasyon algoritmalarına bırakmaya başlamıştır (Gözde ve Taplamacıoğlu, 2011, s. 125).

Yapay arı kolonisi algoritması (Artificial Bee Colony, ABC), 2005 yılında Karaboğa tarafından, bal arısı kolonilerinin besin bulma davranışlarından esinlenerek geliştirilmiş sürü zekası temelli bir optimizasyon algoritmasıdır (Karaboğa, 2005, p. 6). 2005 yılından günümüze kadar ise ABC algoritması, bilgisayar, endüstri ve hidrolik mühendisliği, havacılık ve uzay bilimi gibi farklı bilim dallarında ilgili problemlerde başarıyla uygulanmıştır (Ayan; Kılıç 2011, 15).

ABC algoritmasının öne çıkan özellikleri şöyledir (Akay, 2009, s. 65);

- Basit ve esnek bir algoritmadır.
- Gerçek besin arayıcı arı davranışlarını iyi şekilde yansıtır.
- Sürü zekasını temel alan bir algoritmadır.
- Nümerik problemler için geliştirilmişse de farklı problemler için de kullanılabilir.
- Kontrol parametresi azdır.
- Kaşif arılarca yapılan küresel ve gözcü ile işçi arılarca yapılan bölgesel araştırma kabiliyetine sahiptir ve bu ikisi paralel yürütülmektedir.

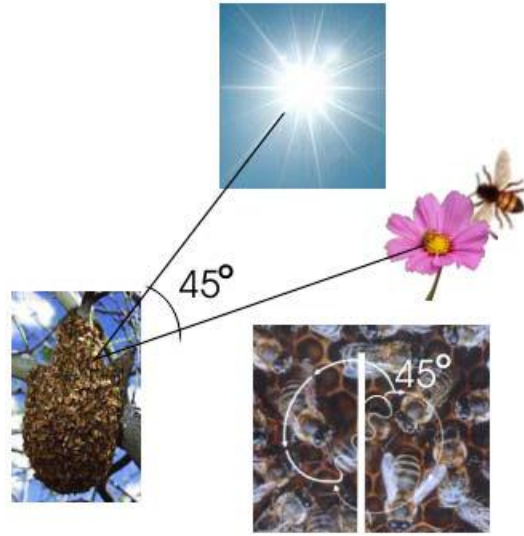
ABC algoritmasında arılar, arama uzayındaki konumlarıyla ifade edilirler. Arama uzayının boyutunu, çözüm içinde optimize edilecek parametrelerin sayısı belirler. Her bir arı, arama uzayında bir besin kaynağına bağlı olarak çalışır. Arıların bulundukları konumlar, yani besin kaynakları aynı zamanda problemin olası

çözümleridir. En iyi besin kaynağı, o besin kaynağının nektar miktarının ölçülmesi ile yani bir maliyet fonksiyonu kullanılarak uygunluk değerinin hesaplanması ile belirlenir. Arılar yeni konumlarını, bir önceki konumlarını ve önceki diğer en iyi besin kaynaklarının konumlarını dikkate alarak değiştirirler. Arama uzayında, arılar tarafından bulunan en uygun besin kaynağının konumu problemin çözümü olarak atanır (Gözde ve Taplamacıoğlu, 2011, s. 125).

ABC algoritmasındaki yapay arı kolonisini üç grup arı oluşturur. Bunlar; besin kaynaklarını tutan işçi arılar, besin kaynaklarını seçmek üzere işçi arıların dansını izleyen gözcü arılar ve rastgele arama yapan kaşif arılardır. Algoritmada görevli arıların sayısı toplam yiyecek kaynağına eşittir. İşçi arıların sayısı gözcü arıların sayısına eşittir. Kaynakta görevli arı kaynaktaki nektar miktarı bitince kaşif arı olmaktadır. Arı kolonilerinin yiyecek kaynaklarının konumları çözülmek istenen problemin muhtemel çözümlerini, nektar miktarı ise çözümün kalitesini ifade etmektedir. ABC algoritması en fazla nektara sahip kaynağın yerini bulmaya çalışarak arama uzayındaki çözümlerden problemin minimumunu veya maksimumunu veren noktayı (çözümü) bulmaya çalışmaktadır (Akay, 2009, s. 57-58).

Yiyecek aranırken başlangıç olarak kaşif arılar rastgele yiyecek aramaya başlarlar. Kaynağı bulan kâşif arı artık görev sahibi olan bir arı haline gelmiştir ve kovana nektar götürmeye başlar. Görevli arı kovana nektarını taşır ve yem getirdiği kaynaklar ile ilgili bilgiyi dans alanında yaptığı dans ile bekleyen gözcü arılara iletir. Kaynaktaki nektar bitmiş ise kendisi de gözcü arı haline gelecektir. Gözcü arılar dansları izledikten sonra yiyeceğin kalitesi doğrultusunda bir kaynağı tercih ederler (Tokmak, 2011, s. 15).

Oldukça geniş bir alanda bulunan kaynaklarla ilgili bilgi alış verişinde arılar kuyruk dansı kullanmaktadır. Bu dans 8 rakamına benzeyen bir dans çeşididir. Dansı izleyen arılar bir titreşim oluşturunca dansı yapan arı dansına son verir. Her 15 saniyede bir dans tekrarlanarak yiyecek kaynağının uzaklığı hakkında bilgi aktarılır. Daha az tekrarlanma sayısı daha uzak bölgelerde yiyecek olduğunu ifade eder. Arılar yön bilgisini, Şekil 7’de olduğu gibi 8 rakamı şeklindeki dansın açılışından elde ederler. Verilen şekildeki örnekte dansı izleyen arılar danstan, güneşle yiyecek arasındaki açının 45 derece olduğunu anlarlar. Ayrıca uzak mesafede bulunan yiyecek kaynağı ile ilgili bilgi verirken arılar arasında daha hızlı dans edilir (Akay, 2009:55-56).



Şekil 7: Kaşif Arıların Kovanda Yaptıkları Salınım Dansı

Kaynak: Akay, 2009:56.

Bir bal arısı kolonisi, kendi kendine çok uzun mesafelere yayılabilir, bazen 10 km mesafeden daha uzağa gidebilirler ve çok sayıda yiyecek kaynağından yararlanmak için eş zamanlı olarak birden çok yöne hareket edebilirler (Yang and Koziel, 2011, p. 189).

ABC algoritması besin kaynağının pozisyonunu güncellemek için, gözcü arılarca yürütülen stokastik seçim planını ve gözcü ve işçi arılarca kullanılan doyumsuz seçim planını birlikte kombine eder. Aynı zamanda ABC algoritmasındaki komşu kaynak üretim mekanizması, özüyle uyumlu mutasyon süreciyle benzerdir. Kaşif arılarca yürütülen rastgele seçim süreci çözümde çeşitlilikten bahseder. ABC algoritması, böylece, esnek, basit ve çok değişkenli ve çok kipli problemlerin optimizasyonunda etkili bir şekilde kullanılabilen güçlü bir optimizasyon algoritmasıdır (Rao, 2011, p. 42).

ABC algoritması, problem çözümlerinde en yaygın kullanılan arı algoritmalarından biridir. ABC algoritmasının literatürde başarılı bir şekilde kullanıldığı bazı uygulama örnekleri şöyledir: genelleştirilmiş atama problemi, enerji dağıtım ağı yapılandırması, yapay sinir ağları eğitimi, çok amaçlı problemler ve dijital görüntülerde şablon eşleştirme. Son zamanlarda Karaboğa ve Akay tarafından yapılan bir çalışmada, ABC algoritmasının performansı diğer gelişimsel hesaplama algoritmaları (GA, PSO,

DE, ES) ile birtakım kıyas problemleri üzerinde karşılaştırılmıştır. Sonuçlar, ABC algoritmasının performansının diğer algoritmalarla kıyasla daha iyi veya yakın özellikte olduğunu göstermiştir (Karaboga and Akay, 2009, p. 108-132) (Panigrahi; Shi; Lim, 2011, p. 330). Yang ise kitabında; son yıllarda yapılan çalışmaların ABC algoritmasının, doğru koşullar altında çeşitli test fonksiyonlarında PSO, DE ve EA algoritmalarından daha iyi performans ortaya koyduğunu gösterdiğini belirtmiştir (Yang, 2010, p. 201).

Şekil 8’de ABC algoritmasının aşama aşama tüm akışını gösteren şema vardır. ABC algoritmasının kısaca adımları ise şöyledir (Karaboga and Akay, 2007, s. 2):

Initialization

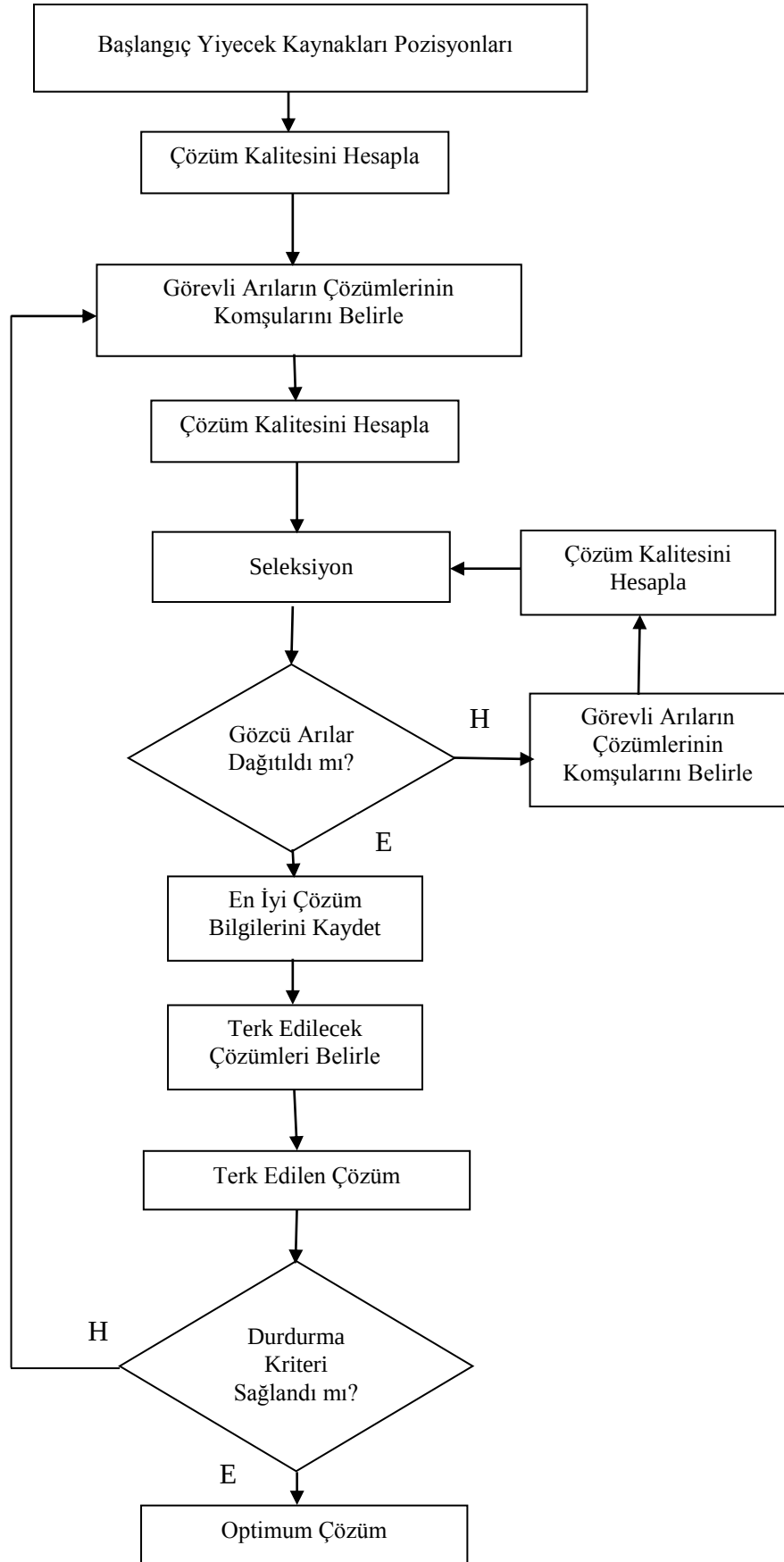
REPEAT

- İşçi arıları yiyecek kaynaklarına gönder ve nektar miktarlarını hesapla
- Gözcü arıları yiyecek kaynaklarına gönder ve nektar miktarlarını hesapla
- Rasgele yeni yiyecek kaynakları bulmaları için kaşif arıları dışarıya

gönder

- O ana kadarki en iyi kaynağı hafızada tut

UNTIL (Durma kriteri sağlanana kadar)



Şekil 8: ABC Algoritmasının Akış Şeması

Kaynak: Akay, 2009:63.

3.3.2. Genetik Algoritmalar

Genetik algoritmalar (Genetic Algorithms, GA), John Holland tarafından 1975 yılında geliştirilen, iyi bilinen ve çokça kullanılan bir evrimsel hesaplama tekniğidir. Bu yöntemde, güçlü olanın hayatta kalması fikrine dayalı Darwin'in doğal seleksiyon teoreminden esinlenilmiştir (Türkeli, 2010:20).

Genetik algoritmalar, yapay zekâ ürünlerinin evrimsel hesaplama, koloni zekâsı başlıkları altında değerlendirilen bir alt kolunda yer almaktadır. GA, gerçek hayatta gözlemlenebilen doğal yaşam sürecine benzer bir şekilde çalışan arama ve en iyileme yöntemleridir. Karmaşık ve çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü ararlar. GA, problemlerin çözümü için gerekli olan evrimsel süreci bilgisayar ortamında simüle ederler. Diğer optimizasyon yöntemlerinde olduğu gibi çözüm için tek bir yapı geliştirmek yerine, bu yapılardan meydana gelen bir küme oluştururlar. Problem için olası birçok çözümü içeren bu kümeye genetik algoritma terminolojisinde popülasyon adı verilir. Bu popülasyonlar, vektör, kromozom veya kromozom adı verilen sayı dizilerinden oluşur. Kromozom içindeki her bir elemana gen adı verilir. Popülasyondaki kromozomlar ise evrimsel süreç içinde genetik algoritma işlemcileri tarafından belirlenmektedir (Düğenci, 2007:62).

Genetik algoritmalar daha çok; matematiksel modeli kurulamayan, çözüm alanı çok geniş olan, problemi etkileyen faktörlerin çok fazla olduğu problemlerin çözümünde etkin olarak kullanılmaktadır. Bu problemlerin başında ise, endüstride karşılaşılan iş sıralama, üretim çizelgeleme problemleri gelmektedir (Erdem, 2008:23).

John Holland tarafından 1975 yılında geliştirildikten sonra diğer metotlar arasında popüler olmaya başlayan GA, başarılı bir şekilde literatürdeki birçok kombinatoriyal optimizasyon problemini çözmek için adapte edilmiştir. Bu popülerliğinin ve başarısının nedeni, diğer metotlara nazaran bazı avantajlarına dayanmaktadır. Bu avantajlarını bir liste halinde verecek olursak, liste şöyledir (Akgündüz, 2008: 22-23):

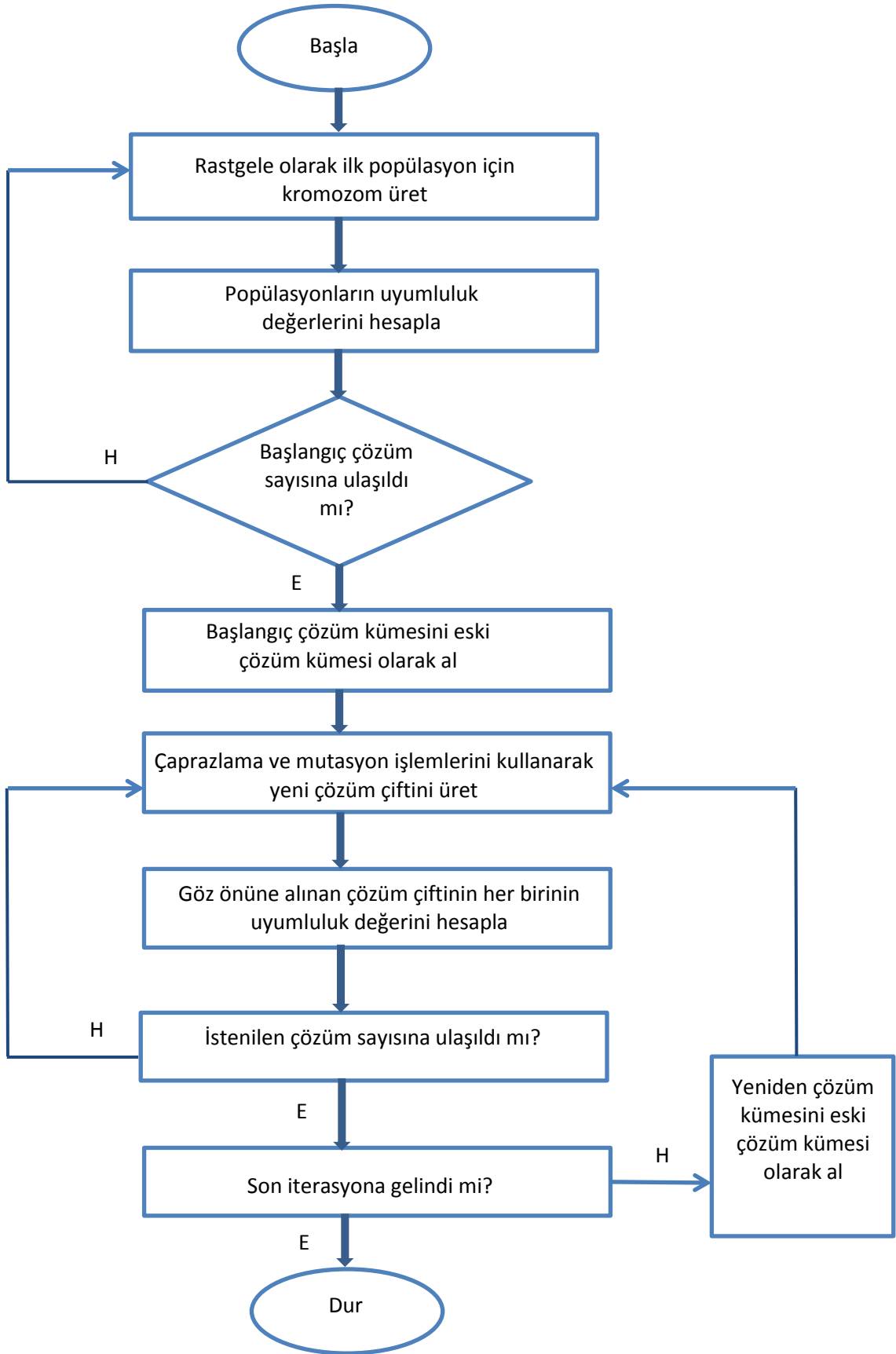
- Sürekli ve kesikli değişkenleri optimize eder,
- Türetilmiş, ikincil bilgi gerektirmez,
- Maliyet yüzeyinin geniş bir örnekleminden eş zamanlı olarak arama yapar,

- Çok sayıda değişkenle ilgilenir,
- Paralel bilgisayarlar için çok uygundur,
- Aşırı derecede karmaşık maliyet yüzeyleriyle değişkenleri optimize eder,
- Sadece tek bir çözüm değil, optimum değişkenlerin bir listesini sağlar,
- Değişkenleri kodlar ve optimizasyon kodlanmış değişkenlerle yapılır,
- Nümerik şekilde oluşturulmuş verilerle, deneysel verilerle veya analitik fonksiyonlarla çalışır.

Genetik algoritmaların diğer geleneksel yöntemlere oranla sahip olduğu avantajlara rağmen bazı sınırlamaları da bulunmaktadır. Bunlar (Rao, 2011, p. 31);

- Son türetilmiş tüm çözümler kabul edilmiş ve onları üreten çiftlerin uygunluk değerleri ne olursa olsun her nesil sonunda bu çiftler terkedilmiştir. Bu ise şöyle bir risk doğurur; belki daha iyi ebeveyn çiftlerin yerini kötüleşmiş genç çiftler alacaktır. Böylece, ebeveyn popülasyonu üzerindeki genç popülasyonun ortalama performansındaki iyileşme her zaman garanti edilemeyecektir.
- Diğerlerince daha iyi yavrular oluşturma olasılığı önemsizdir.
- Yakınsama hızı dikkate alındığında etkin değildir.
- Optimal çözümü garanti etmemektedir.

Genetik algoritmalarındaki evrimsel süreçte, çaprazlama, mutasyon ve seçme fonksiyonları yerine getirilir. GA bir problemi çözerken çaprazlama operatörünü kullanarak iki veya daha fazla adımı birleştirerek daha iyi sonuç veren bir adıma ulaşabilir. Mutasyon, değişik adımların rastgele sıralanması ve birleştirilmesi işlemini uygulayarak sonuçları değerlendirir. Son olarak seçme işlemi ile daha iyi sonuç veren adımlar tutulup diğerlerine göre kötü sonuçlar veren, iyilere yer açmak için algoritmadan atılırlar. Şekil 9'da Genetik Algoritmaların işleyişine dair bir akış şeması verilmiştir (Özdemir, 2006:71-72).



Şekil 9: Genetik Algoritmaların İşleyişi

Kaynak: Özdemir, 2006:72.

- **Genetik Algoritmaların Temel Bileşenleri**

Doğadaki genetik süreçten esinlenerek oluşturulmuş GA yapısı üç temel bileşen üzerine kuruludur. Bunlar; çözmeye çalışılan problemin kısmi bir çözümünü veya bir özelliğini temsil eden ‘Gen’, probleme ilişkin tüm çözümü temsil eden ‘Kromozom’ ve tüm olası çözümleri içeren alternatif çözümlerle dolu, kromozomların bir araya gelmesiyle oluşmuş ‘Popülasyon’ dur. GA’ yı daha iyi anlamak için bu bileşenler kısaca tanıtılacaktır (Düğenci, 2007:64-65).

Gen: Tek başına bir anlamı olmayan ve genetik bilgiyi taşıyan en küçük genetik yapı birimidir. Bir fonksiyon optimizasyonu düşünüldüğünde herhangi bir x,y vb değişkenler genetik yapıda gen olarak adlandırılırlar. Genler genelde 0-1 ikilileri şeklinde temsil edildiği gibi nümerik olarak veya karakterlerce de temsil edilebilirler. Örnek bir gen temsili:

Gen 1 = 1010 şeklinde gösterilebilir.

Kromozom: Bir ya da daha fazla genin bir araya gelmesiyle oluşan ve probleme ait tüm gerekli bilgileri içeren genetik yapılardır. Her bir kromozom probleme ait alternatif tam bir çözümdür. Kromozomları ikili sayılar halinde şu şekilde gösterebiliriz:

Kromozom 1 = 1101100100110110

Kromozom 2 = 1101111000011110

Ancak çizelgeleme problemlerinde 0-1 şeklinde dizilimle kromozom oluşturmak problemimizi temsil etmede yeterli olmayacaktır. Bu yüzden gezgin satıcı problemleri ve çizelgeleme, sıralama problemleri için permütasyon kodlama biçimleri kullanılmaktadır. 9 iş için sıralamanın sağlandığı bir kromozom örneği şöyle verilebilir:

Kromozom A = 1 5 3 2 6 4 7 9 8

Kromozom B = 8 5 6 7 2 3 1 4 9

Popülasyon: Popülasyon kromozomların bir araya gelmesiyle yani ilgili problemin alternatif tüm çözümlerinin bir araya gelmesiyle oluşan bir topluluktur. Aynı anda bir popülasyonda bulunan birey sayısı sabit ve probleme bağlı olarak kullanıcı tarafından belirlenmektedir. Birey sayısının sabit olması olayı, algoritma sürecinde popülasyona yeni kromozomların dahil olması ve doğal seleksiyona benzer bir uygulama ile bazı eski kromozomların popülasyondan çıkarılmasıyla sağlanmaktadır.

- **Genetik Algoritmaların Operatörleri**

GA'nın en önemli kısımlarını oluşturan operatörler bölümünde dört kavramdan bahsetmek yararlı olacaktır. Bunlar; eşleştirme, çaprazlama, mutasyon ve elitizmdir.

Eşleştirme: Üreme işlemi belli bir seçme kriterine göre kromozomların seçilip yeni neslin oluşturulması işlemidir. Seçme kriterleri uyumluluğu esas alır ve sonraki çaprazlama ve mutasyon aşamalarında daha uyumlu yeni kromozomların ortaya çıkma olasılığı artar. Bir teoriye göre, iyi olan kromozomlar yaşamını sürdürmeli ve yeni kromozomlar bu kromozomlardan oluşmalıdır. Bu seçim kriterlerinden bazıları ise şöyledir: Rulet seçimi, Boltzman seçimi, Turnuva seçimi, Sıralı seçim (Düğenci, 2007:67).

Çaprazlama: İki kromozom arasında karşılıklı bilgi değişimini gerçekleştiren ve böylece yeni kromozomların oluşmasını sağlayan operatördür. Çaprazlama işlemi çeşitli şekillerde yapılır. Bunlar; tek noktalı çaprazlama, çift noktalı çaprazlama, pozisyona göre çaprazlama, sıraya göre çaprazlama (Erdem, 2008:29-31).

Mutasyon: Kromozomların genleri veya genleri oluşturan daha ufak birimler üzerinde değişiklik yapmayı sağlayan operatörlerdir. Mutasyon, oluşan yeni çözümlerin önceki çözümü kopyalamasını önlemek ve sonuca daha hızlı ulaşmak amaçlarıyla yapılır. Mutasyon aşamasında da pozisyona göre ve sıraya göre değişimler yapılabilmektedir.

Elitizm: Üreme, çaprazlama ve mutasyon işlemleri sonrasında nesilde bulunan en iyi uyumluluğa sahip kromozom sonraki nesle aktarılamama riski taşımaktadır. Bu risk şöyle ortadan kaldırılabilir; bu işlemlerden sonra oluşan yeni nesilde, bir önceki neslin en iyi (elit) kromozomu yeni nesildeki herhangi bir kromozomla değiştirilir. Bu işleme elitizm denir (Düğenci, 2007:33).

- **Genetik Algoritmaların Performansına Etki Eden Faktörler**

Genetik algoritmaların operatörleri ve temel bileşenleri, algoritma performansına etki eden en temel faktörler olsa da diğer etki edebilecek faktörler aşağıda sunulmuştur (Erdem, 2008:37-38):

1. Kromozom Sayısı: Kromozom sayısını artırmak işlem zamanını artırır ancak azaltmak ise kromozomda çeşitliliği yok etmektedir.
2. Değişim Oranı: Kromozomlar birbirine yakınlığında henüz çözüm noktalarının uzağında kalınıyorsa değişim işlemi genetik algoritmayı sıkıştığı yerden kurtarmak için tek yoldur. Fakat değişim oranına yüksek bir değer vermek genetik algoritmanın kararlı bir noktaya ulaşmasını engeller.
3. Çaprazlama Noktası Konusu: Genelde tek noktalı çaprazlama kullanılmakta ancak araştırmalar bazı problemlerde çok noktalı çaprazlama yapmanın daha yararlı olduğunu göstermektedir.
4. Bireylerin Nasıl Değerlendirileceği Konusu: Çaprazlama sonucu oluşan yeni iki bireyin hemen kullanılıp kullanılmayacağı konusu bazen önem arz etmektedir.
5. Nesillerin Birbirinden Ayrık Olup Olmama Durumu: Genelde normal olarak her nesil tümüyle bir önceki nesle bağlı olarak oluşturulur. Ancak bazı durumlarda yeni nesli oluştururken eski nesille birlikte o ana kadarki eldeki diğer bireylerle birlikte oluşturmak daha yararlı olacaktır.
6. Parametre Kodlamasının Nasıl Yapıldığı Konusu: Kodlamanın nasıl yaptığı önemli noktalardan biridir ki mesela, bazen bir parametrenin logaritmik ya da doğrusal kodlanmış olması GA performansında önemli bir fark oluşturabilmektedir.
7. Kodlama Gösteriminin Nasıl Yapıldığı Konusu: Bu konu da GA'nın performansında fark oluşturabilmektedir. En çok kullanılan gösterim yöntemleri; ikili düzen, kayan nokta aritmetiği ve gray kodu ile gösterimdir.
8. Başarının Nasıl Değerlendirildiği Konusu: Değerlendirme fonksiyonunu akıllıca belirlemek gerekir aksi halde çalışma zamanını uzatabileceği gibi sonuca hiçbir zaman ulaşamamaya da neden olabilir.

3.3.3. Karınca Koloni Optimizasyonu Algoritması

Günümüzde biyolojik yapılardan esinlenerek oluşturulan sistemler, büyük ilgiyle birlikte artan bir öneme sahip hale gelmişlerdir. Doğadaki bazı sosyal sistemler, kısıtlı yetenekli bireylerden oluşmasına rağmen sürü zekâsı davranışı sergilemekte ve bu davranış birçok problemin çözümünde, problemlere uyarlanarak kullanılmaktadır. Karıncaların kolektif davranışlarının bir benzetimi olan Karınca Algoritmaları da son zamanlarda geliştirilen popülasyon tabanlı metasezgisellerden biridir. Karıncalar tek başlarına birey iken basit kabiliyetlere sahip olmalarına rağmen, koloni halinde iken; kendilerinden çok büyük boyutlardaki cisimleri taşıyabilirler, köprüler kurabilirler veya yuvaları ile yiyecek kaynağı arasındaki en kısa yolu bulmak gibi karmaşık işlemlere karşı oldukça zeki çözümler üretebilirler (Yılmaz, 2008:53).

- **Doğada Karıncalar**

Karıncalar, yapıları itibariyle sosyal böceklerdir ve nüfusları 2 ila 25 milyon arasında değişen organize koloniler halinde birlikte yaşayan canlılardır (Yang, 2010, p.189).

Bireysel karıncalar oldukça küçük (sadece 2.2 ila 2.6 mm uzunlukta ölçülen) olmalarına ve ayrıık şekilde iken oldukça amaçsızca dolaşmalarına rağmen, bir grup oluşturduklarında olağanüstü zeki davranışlar sergilemektedirler (Grosan ve Abraham, 2011, p. 416).

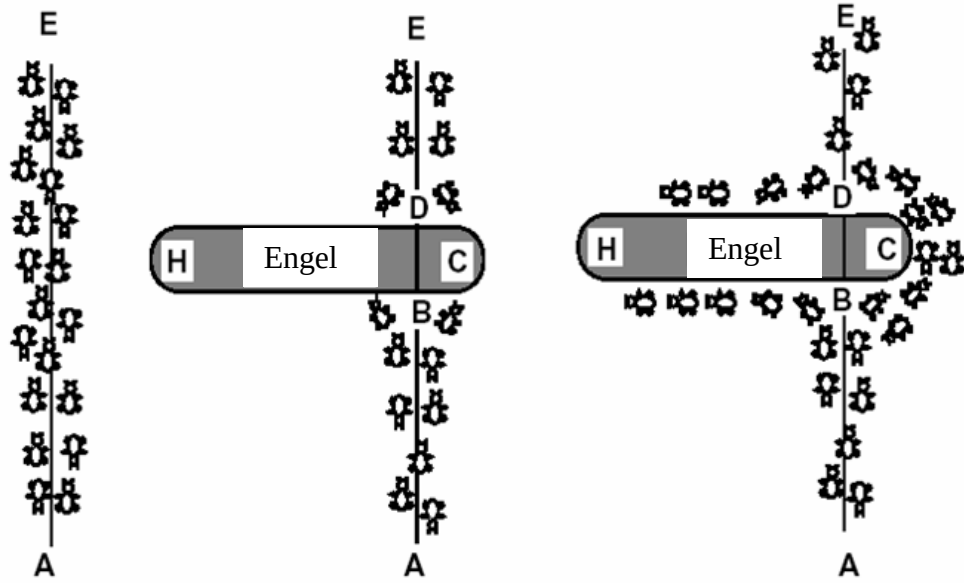
Karıncalar, yuvalarından bir yiyecek kaynağına giden en kısa yolu, herhangi bir görsel ipucuna gerek olmadan bulma yetisine sahiptirler. Gerçek bir karınca kolonisinin yiyecek arayıcı üyesi diğer koloni üyeleriyle iletişimini, stigmergy adı verilen ortamın değişimine dayalı iletişimin dolaylı bir şekli ile sağlamaktadır. Gerçek bir karınca kolonisinde stigmergy adlı bu sistemin temel bileşeni, feromon (pheromone) olarak adlandırılan kimyasal bir maddedir. Karıncalar yiyecek ararken geçtikleri yollara bu kimyasallardan bırakırlar. Aynı yolu takip eden karınca miktarı artışı ile feromon miktarı da artacağı için o yolun en kısa yol olarak tercih edilme olasılığı da artmış olur. Çünkü karıncalar rasgele hareket etmek yerine feromon miktarının yoğun olduğu yoldan gitmeyi tercih ederler. Yoldaki feromon miktarı ise buharlaşmak durumundadır ve zamanla aynı yoldan yine gidilmezse tükenecektir. Böylece karıncaların o yolu takip etme şansları azalacaktır. Yola bırakılan ve buharlaşan feromon miktarı, yuva ve

yiyecek kaynağı arasındaki yolun uzaklığı ile ilişkilidir. Bu yol ne kadar uzaksa o kadar daha fazla buharlaşma olacaktır. Aksine yol ne kadar kısa ise yola o kadar fazla feromon bırakılacaktır. Böylece kısa mesafelerde daha yüksek oranlarda feromon bulunacaktır. Aynı zamanda yiyeceğin kalitesi de, yola bırakılan feromon miktarını etkileyen bir başka faktördür (Yıldırım, 2008:3).

Bilim adamları feromon kimyasalı hakkında çok fazla bilgiye sahip değildirler. Sadece dolaylı deneylerde, değişen zaman aralığında, farklı miktar ve yoğunluktaki feromona karıncaların gösterdiği davranışlar incelenmiş ve feromonların buharlaşma oranı, emilim oranı, difüzyon sabiti gibi kaba ve yaklaşık miktarlara ve sonuçlara ulaşılmıştır. Feromonların ömürleri, karıncaların türlerine, enzim yapılarına, kolonilerin büyüklüğüne ve hava şartlarına bağlı olarak birkaç saatten birkaç aya kadar olabilmektedir. Karıncalardaki ortak bir akıl sistemi ise, feromonun buharlaşmasını kontrol altında tutarak uzun bir yolda feromon izlerinin sürekliliğini sağlamaktadır (Çatal, 2009:39).

Yuva ile yiyecek kaynağı arasında gidip gelen karıncalar başlangıçta rastsal olarak farklı yolları tercih edebilirler. Ancak sonraları, daha kısa yoldan gidip gelen karıncalar daha çok sayıda gidiş geliş yapacağı için o yoldaki feromon miktarı da çok olacaktır. Böylece feromon maddesinin kokusunu alan karıncalar, bir ayırım noktasına geldiklerinde yollarını seçerken, kokunun yoğun olduğu tarafı büyük olasılıkla seçme eğiliminde olacaklardır (Küçük, 2010:57).

Şekil 10'da karıncaların yuvaları ve yiyecek kaynağı arasında gidip geldikleri yolda, bir engel oluşması hali gösterilmiştir. Bu durumda karıncalar eşit olasılıkla sağa veya sola gitmekte serbesttirler. Ancak kısa olan yolu (A-B-C-D-E/E-D-C-B-A) rastgele tercih eden karıncalar daha kısa sürede gidiş geliş yapacakları için yoğunlaşan feromon sayesinde o yolun diğer karıncalarca da tercih edilme olasılığını artırırlar. Böylece kısa olan yolda artan yoğunlukta feromon depolanırken, uzun olan yolda da (A-B-H-D-E/E-D-H-B-A) çok tercih edilmediği için azalan bir feromondan bahsedebiliriz. Sürekli buharlaşma nedeniyle de zamanla feromon tükenecektir ve aynı engel aynı şekilde o yolda kaldıkça, karıncalar feromonun yoğun olduğunu anladıkları kısa yolu tercih edeceklerdir.



Şekil 10: Karıncaların Yuvaları İle Yiyecek Kaynakları Arasında Engel Oluşması Durumu

Kaynak: Küçük, 2010:58.

Bazı karınca türlerinin yiyecek arama modelleri olağanüstü bir düzenlilik gösterir. Ordu karıncaları adı verilen bu türler, yaklaşık 123 derecelik bir açı ile düzenli rotalar halinde yiyecek ararlar. Bu düzeni nasıl takip ettikleri ve yönettikleri bilinmemekle beraber, konuyla ilgili çalışmalar, bir alanda hareket ettiklerini, bir karargâh kurduklarını ve yiyecek aramaya başladıklarını göstermektedir. İlk gün, rasgele bir yöne birkaç yüz metre kadar hareket ederler, daha sonra dağılarak geniş bir alanı kapsarlar. Sonraki gün, önceki gündeki yönden yaklaşık olarak 123 derece saparak farklı bir yön seçecekler ve yine geniş alana yayılacaklardır. Takip eden gün, ikinci günün yönünden de 123 derece saparak yine farklı bir yön seçeceklerdir. Bu yolla, yaklaşık iki haftada tüm alanı kapsayacak şekilde dolaşmış olurlar ve yeni bir alanda yiyecek aramak ve karargâh kurmak için eski bölgeden taşınırlar. Buradaki ilginç şey; $360/3 = 120$ derece açısını kullanmazlar, 120 derece açı kullanılmış olması ile dördüncü gün, zaten birinci günde aranmış boş alanı yeniden aramak gerekecektir. 123 derecelik açıdaki güzellik, birinci gündeki yönden yaklaşık 10 derecelik bir açı bırakmış olmasıdır. Sonuç olarak bunun anlamı, karıncalar 14 gün içinde tüm çemberi önceden

aranmış bir yeri tekrar etmeden taramaktadırlar. Bu da şaşırtıcı bir fenomendir (Yang, 2010, p.190).

- **Yapay Karıncalar**

Gerçek karıncalar, kör olmalarına rağmen sezgisel olarak ve sürü zekâsı ile hareket ederek yuvaları ile yiyecek kaynağı arasındaki en kısa yolu hızlıca bulabilmektedirler. Karıncaların bu yol seçim özelliğine bazı ilaveler yapılarak oluşturulan algoritma, gerçek problemlerin çözümünde kullanılabilir hale getirilmektedir. Gerçek karıncalardan aynen alınan özellikler şöyledir:

- Karıncalar arasında feromon kimyasalı ile gerçekleşen iletişim
- Feromon miktarının fazla olduğu ve yoğun hissedildiği yolların öncelikle seçilmesi
- Daha kısa yollar üzerinde feromon miktarının hızlı artması, yok olmaması

Aynen alınan özelliklerin yanı sıra, gerçek problemlere tam uyum ve karıncalardan daha iyi verim alabilmek için birtakım özellikler de eklenmiştir. Bunlar (Yılmaz, 2008:56):

- Yapay karıncalar zamanın ayırık olarak hesaplandığı bir ortamda yaşarlar.
- Yapay karıncalar tamamen kör olmayıp problem ile ilgili verilere, detaylara ulaşabilirler.
- Yapay karıncalar belirli bir miktar hafızayla problemin çözümü için oluşturdukları verileri tutabilirler.

Yapay karıncalarda geri iz sürebilme özelliği, karıncanın takip ettiği yolu hafızasında tutmasından dolayı feromon bırakacağı yolları da bilmesi anlamındadır. İleriye bakış özelliği ise, karıncanın bulunduğu noktaya doğrudan bağlantılı olan tüm noktaların uzaklıklarını bilmesi ve böylece yol seçiminde mesafeleri de hesaba katmasını ifade eder (Çatal, 2009:44).

• Karınca Kolonisi Optimizasyonu

Karınca davranışlarının bu karakteristiklerine dayalı olarak bilim adamları, son yıllarda önemli atılımlarla birçok karınca koloni algoritmaları geliştirdiler. 1992 yılında Marco Dorigo bu alandaki çalışmalara öncülük etmiş ve Karınca Koloni Optimizasyonu algoritmasını (Ant Colony Optimization, ACO) geliştirmiştir. Daha sonraları ise birçok farklı çeşitleri literatüre katılmıştır (Yang, 2010, p.190).

ACO algoritması birbirini takip eden şu tanımları gerektirir (Sumathi ve Surekha, 2010, p. 663-664):

- Problem uygun şekilde temsil edilmelidir öyle ki; diğer probleme has bilgiler ve yoldaki feromon miktarına dayalı olasılıksal geçiş kurallarının kullanımı boyunca çözümleri artan şekilde güncellemeleri için karıncalara izin verilir. Aynı zamanda, problem tanımı için karşılık gelen sadece geçerli çözümler kurmak için bir strateji uygulamak önemlidir.
- Bir problem bağımlı sezgisel fonksiyon olan η ; şimdiki kısmi çözümler için eklenmiş olabilecek bileşenlerin kalitesini ölçer.
- Feromon güncellemesi için bir kural seti; τ feromon değerini nasıl değiştireceğini belirtir.
- η sezgisel fonksiyonun değerine ve τ feromon değerine bağlı olan bir olasılıksal geçiş kuralı, tekrar eden bir şekilde bir çözüm kurmak için kullanılır.

ACO algoritmaları, yerel arama teknikleri, sevk kuralları ve diğer teknikleri tek bir çerçevede kombine eder. ACO paradigmasında karınca kolonilerinin iz takibi davranışından esinlenilmiştir. Karıncalar, bir yerden bir yere giderken feromon adı verilen bir kimyasal diğer karıncalar da öndekini izlesin diye bir sinyal olarak yolları boyunca bırakırlar. Bir ACO algoritması, yapay bir karınca kolonisinin, sezgisel bilgide olduğu kadar iyi önceki bulunan çözümlerle bağlantılı olan yapay feromon izleri kullanarak eldeki problemler için tekrarlamalı çözümler oluşturduğunu varsayar. Karıncalar, birbirleriyle sadece dolaylı yoldan, algoritmanın uygulanması esnasında yollarda depoladıkları feromon miktarındaki değişimler ile iletişim kurarlar. Çünkü karıncaların oluşturduğu çözümler bölgesel olarak optimal olmayabilirler, çoğu ACO algoritması da bir yerel arama prosedürü boyunca çözümlerini geliştirmeleri için

karıncalara imkan verir. Basit ACO çerçevesinin dört adımdan oluştuğunu söyleyebiliriz. Bu adımlar (Pinedo, 2008, p. 387):

1. Adım: Başlatma

Parametreleri ayarla ve feromon izlerini başlat.

2. Adım: Çözümleri oluştur

Sevk kuralları ve feromon izlerinin bir kombinasyonunu kullanarak ℓ çözümlerini oluştur.

3. Adım: Çözümleri geliştir

Her bir ℓ çözümleri için bir yerel arama prosedürü uygula.

4. Ayarları güncelle

ℓ çözümlerinin en iyisi 3. Adımda üretildiyse şimdiye kadarki en iyi çözümden iyidir, şimdiye kadar gözlenen en iyi çözümün yerine şimdiki çözümü yerleştir ve amaç fonksiyonuna karşılık gelen değeri hafızaya al.

Feromon iz değerlerini güncelle ve sonraki iterasyona başlamak için 2. Adıma geri dön.

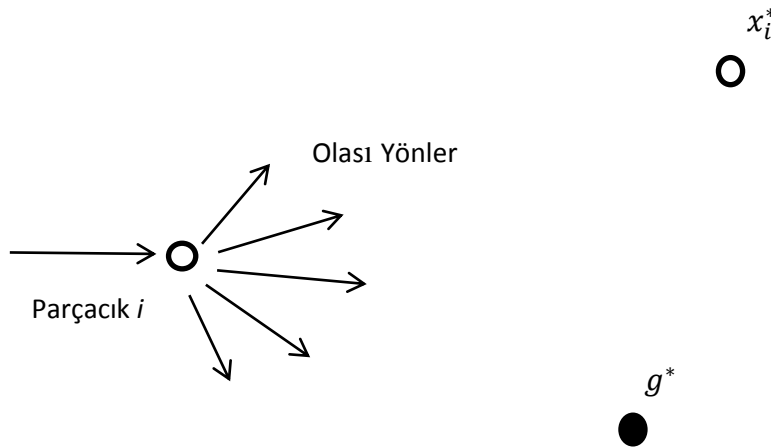
3.3.4. Parçacık Sürü Optimizasyonu Algoritması

Parçacık sürü optimizasyonu (Particle Swarm Optimization, PSO), doğadaki kuş ve balık sürülerinin davranışlarının bir benzetimi olarak Kennedy ve Eberhart tarafından 1995 yılında geliştirilen bir yöntemdir. Karınca koloni algoritmaları ve yapay karınca algoritmaları gibi birçok algoritma da sözde sürü zekâsının bu davranışlarını kullanırlar. Parçacık sürü optimizasyonu, genetik algoritmalar ve karınca algoritmalarıyla benzerlikler gösterebilir ancak aslında onlardan daha basittir çünkü mutasyon/çaprazlama operatörleri veya feromon kullanmazlar. Onun yerine, sürü parçacıkları arasında yerel iletişim ve gerçek sayılı rasgelelik kullanırlar. Bu anlamda, genetik algoritmalarındaki ikili diziler içindeki parametreleri kodlama ve kod çözme olmadığı için PSO uygulamak daha basittir.

Son zamanlarda, PSO birçok alanda başarılı şekilde uygulanmıştır. PSO, optimizasyon problemlerinin birçok türünü evrimsel algoritmalara göre daha ucuz ve hızlı bir şekilde ilk iterasyonlarda çözmektedir ancak, jenerasyon sayıları arttıkça hesaplama etkinliği azalabilecektir. Buna ilave olarak, PSO algoritmasının ayarlanacak

parametre sayısı da azdır. Algoritma doğru bir şekilde dizayn edildiğinde, problemlerin farklı türleri için de çok iyi çözümler üretecektir (Uysal, 2006:24).

Bu algoritma, bir amaç fonksiyonunun uzayını parçacık adı verilen bireysel temsilcilerin yörüngesini ayarlayarak ararken, bu yörüngeler yarı stokastik bir şekilde parçalı yollar oluştururlar. Parçacık sürüsünün bu hareketi iki önemli bileşen içerir: stokastik bir bileşen ve deterministik bir bileşen. Her parçacık, aynı zamanda rasgele hareket etmek için bir eğilim içinde iken, şimdiki yerel en iyi g^* nin pozisyonuna ve geçmişindeki kendi en iyi pozisyonuna x_i^* ye doğru çekilir (Yang, 2010, p. 203-204). Şekil 11’de bu durum gösterilmiştir.



Şekil 11: i Parçacığının Şimdiki En İyi Ve Yerel En İyeye Doğru Hareketi

Kaynak: Yang, 2010, p. 204.

PSO algoritmasında kuşların uzayda, yerini bilmedikleri yiyeceği aramaları, bir problem için optimum çözümü aramaya benzetilmiştir. Doğada kuşlar yiyecek ararken aralarındaki özel iletişimle yiyeceğe en yakın olan kuş hakkında bilgilenip onu takip ederler. Her tekil çözüm, aslında arama uzayındaki bir kuştur. Parçacıklar her birim hareketlerinin sonunda bulundukları koordinatları bir uygunluk fonksiyonuna gönderirler ve böylece o parçacığın uygunluk fonksiyonu değeri belli olur. Aynı zamanda bir parçacık, koordinatlarını, çözüm uzayında her bir boyutta ne kadar hızla ilerlediğini, şimdiye kadar elde ettiği en iyi uygunluk değerini ve bu değeri elde ettiği koordinatları hafızasında tutar. Parçacık, her iterasyonda çözüm uzayında nereye

gideceğini, diğer parçacıkların (komşular) en iyi koordinatlarının ve kendi kişisel en iyi koordinatlarının bileşimi olarak tespit eder.

PSO algoritması, parçacıklar arasındaki bilgi paylaşımını esas almaktadır. Her bir parçacık optimum çözüme ulaşmak isterken yeni pozisyonunu, kendi önceki bilgilerinden ve bir önceki iterasyondaki (yada sürüdeki) en iyi pozisyonu dikkate alarak belirler. Global komşular, her bir iterasyonda sürü içerisindeki en iyi uygunluk fonksiyonu değerini veren parçacığın değerleridir. Her bir iterasyonda bir global komşu değeri bulunur. Yerel komşular ise, ilk iterasyonda her bir parçacığın aynı zamanda pozisyon değerleridir ve her iterasyonda parçacık sayısı kadar yerel en iyi ortaya çıkar. Bu durumda her bir parçacık yerel ve global komşu değerlerini kullanarak bir önceki iterasyonda en iyi değere sahip parçacığın pozisyonuna doğru hareket eder. Algoritmada durdurma kriteri parçacıklardan en az birinin optimum çözüme ulaşması yada istenen iterasyon sayısına ulaşılmasıdır. PSO algoritmasında sürünün başarısı önem arz eder, yani parçacıklardan en az birinin optimuma ulaşması yeterlidir, tüm parçacıkların optimuma ulaşması gerekmez (Bali, 2009:25). PSO algoritmasının genel hâli için sahte (pseudo) kod şöyledir:

```

Parametreleri belirle
Başlangıç popülasyonunu oluştur
Her bir parçacık için{
    Amaç fonksiyonu değerlerini bul
    Yerel en iyiyi bul}
Global en iyiyi bul
Yap{
    Her bir parçacık için{
        Hız vektörünü güncelle
        Pozisyon vektörünü güncelle
        Amaç fonksiyon değerlerini bul
        Yerel en iyiyi güncelle}
    Global en iyiyi güncelle
Durdurma kriteri

```

PSO algoritmasında son aşamada durdurma kriteri olarak sayabileceklerimiz şöyledir (Grosan and Abraham, 2011, p. 412):

- Maksimum iterasyon sayısı
- İyileşme olmadan iterasyon sayısı
- Minimum amaç fonksiyonu hatası

PSO için verilen pseudo kodunda ve algoritmada sıkça kullanılan terimleri şu şekilde açıklayabiliriz (Bali, 2009:26-28):

Pozisyon Vektörü: Pozisyon vektörü x_i^k ile gösterilir ve j kadar parçacıktan oluşan bir problem için, her bir parçacık bir çözümü ifade etmek üzere;

$X_i^k = [x_{i1}^k, x_{i2}^k, \dots, x_{ij}^k]$ pozisyon vektörü olarak tanımlanır.

Hız Vektörü: Hız, parçacığın bir sonraki koordinatlarını belirlemede yani yeni bir çözüme gitmede en etkili faktörlerden olup hız vektörü V_i^k ile gösterilir. Parçacığın hız vektörü ise;

$V_i^k = [v_{i1}^k, v_{i2}^k, \dots, v_{ij}^k]$ şeklinde ifade edilir.

Yerel En İyi Değer: Yerel en iyi değer, bir parçacığın o ana kadar elde edilmiş en iyi amaç fonksiyonu değerine sahip bulunduğu çözümdür. Sürüde bulunan parçacık sayısı kadar her iterasyonda yerel en iyiler bulunur.

Yerel En İyi Komşular: Her iterasyonda parçacık sayısı kadar elemandan oluşan bir vektör olarak ifade edilir. Aynı zamanda, yerel en iyi değerlere sahip pozisyon vektörü olarak da tanımlanır. Bir popülasyondaki i . parçacığın yerel en iyi komşuları;

$P_i = [p_{i1}, p_{i2}, \dots, p_{ij}]$ şeklinde ifade edilir.

Global En İyi Değer: Elde edilmiş en iyi amaç fonksiyonu yada uygunluk fonksiyonu değeri ve bu değere sahip parçacığın pozisyonu yani bulunduğu çözümüdür. Bir çözüm uzayında en fazla iterasyon sayısı kadar global en iyi olabilir.

Global En İyi Komşular: Her iterasyonda bulunan global en iyi değerlere sahip parçacık pozisyon değerleridir. Bir popülasyondaki i . parçacığın global en iyi komşuları;

$G = [g_1, g_2, \dots, g_j]$ şeklinde ifade edilir.

Parçacığın Tecrübesi (c_1): Sosyal parametre olarak da adlandırılan bu terim, parçacığın önceki iterasyonlarda elde ettiği en iyi sonuçların bir sonraki iterasyona etkisidir.

Popülasyonun Tecrübesi (c_2): Kavramsal parametre olarak da adlandırılan bu terim, popülasyonda önceki iterasyonlarda en iyi sonuca sahip parçacığın sonuçlarının bir sonraki iterasyona etkisidir.

Atalet Ağırlığı: Bir önceki hız vektörünün mevcut hız vektörü üzerindeki etkisini kontrol eden parametreye atalet ağırlığı denir ve w ile gösterilir.

DÖRDÜNCÜ BÖLÜM

AKIŞ TİPİ ÇİZELGELEME PROBLEMLERİNE METASEZGİSEL YAKLAŞIMLARLA BİR UYGULAMA DENEMESİ

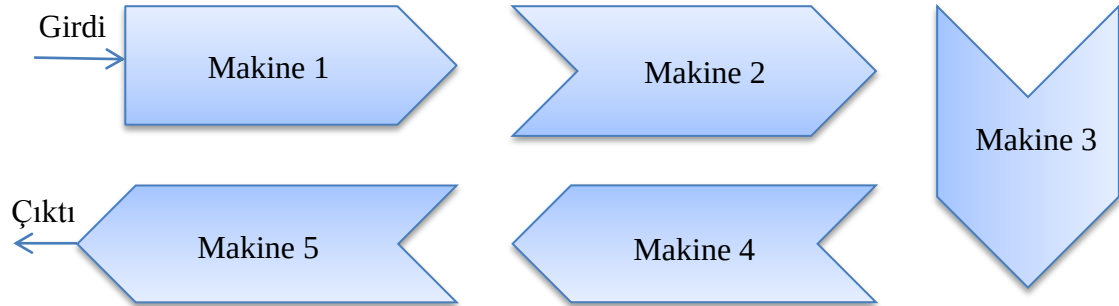
Uygulama bölümleri, tez çalışmalarında anlatılan konunun ve savunulan düşüncenin net şekilde ifade edildiği, konunun uygulama ile somutlaştırılarak daha da anlaşılır hale getirildiği ve verilmek istenen bilginin doğruluğunun okuyucuya ispat edildiği bölümlerdir. Böylece teorik bilginin pratiğe dönüşmesi ile konu açık şekilde sonuçlandırılmış olmaktadır.

Bu bölüme kadar FSP ve çözüm metotlarından bahsedilerek karşılaşılabilecek olası bir akış tipi problem ve bunun çözümü ifade edilmeye çalışılmıştır. Bu bölümle birlikte akış tipi test problemine dönüştürülmüş bir FSP incelenerek problemle ilgili verilere yer verilecektir. Daha sonra problem çeşitli metasezgiseller ile ve birtakım çizelgeleme kurallarıyla Matlab ve Legin programlarında çözülecektir. Farklı parametreler denenerek optimum çözüm elde edilmeye çalışılacak, metotların her birinden elde edilen çözüm sonuçları değerlendirilecektir. Böylece bu FSP için metotların performans ölçümleri de yapılmış olacaktır.

4.1. UYGULAMA PROBLEMİNİN TANITIMI

Uygulamada kullandığımız problem, İlkay ve arkadaşlarının (2009) çalışmasındaki problemden esinlenerek ve verileri kullanılarak oluşturulmuş akış tipi bir üretim çizelgeleme test problemidir. Problem yatak üretimi ile uğraşan bir firmanın, 5 makinesi ve bu makinelerle belirli bir dönemde yüklenen 23 işi içermektedir. İşlerin tamamı tüm makinelerden geçmekte ve 23 iş 5 makinede aynı rotayı izleyerek üretilmektedir. Makineler seri halde dizilidir ve bu 23 işin belirli bir sırada tüm makinelerden geçmesi istenmektedir. 23 iş üretim açısından birbirlerine benzerlik gösteren 5 ayrı gruba ayrılmıştır. Buna göre benzerlik gösteren işler aynı grupta olup sıralanmaları esnasında aralarında kurulum zamanı gerektirmezler. Ancak farklı iş

gruplarındaki işler sıralanırken işler arası 40’ar dakikalık kurulum zamanları gerekmektedir. Üretim ortamındaki söz konusu 5 makinenin basit şekilde dizilimi Şekil 12’de verilmiştir.



Şekil 12: Uygulama Problemindeki 5 Makine Ve Dizilimi

23 işin 5 ana gruba ayrılması ile işlerin birbirine benzerlikleri tespit edilmiştir ve bu sayede işlerin her birinin arasında kurulum zamanına katlanılmamaktadır. Bu gruplar işlere göre şöyle ayrılmıştır:

- Walthe yataklar A grubu olmuştur,
- Emmen yataklar B grubu olmuştur,
- Coolmax ve Roll yataklar C grubu olmuştur,
- Domino yatak D grubu olmuştur,
- Azalee, Prestige, Reina, Venüs ve Pedli yaylı yataklar E grubu olmuştur.

Problemde amaç fonksiyonu olarak, tüm işlerin tüm makinelerde maksimum tamamlanma zamanının (makespan) minimize edilmesi alınmıştır. Böylece 23 işin de 5 makineden en kısa hangi sürede geçeceği bulunmak istenmiştir. Sonuç olarak da 3 farklı metot ve bazı çizelgeleme kuralları kullanarak çözümler elde edilmiştir.

Problemdeki işlerin türleri, hangi gruba ait oldukları, her bir işin her bir makinedeki işlem süreleri, teslim süreleri ve iş grupları arasındaki kurulum süreleri ilişkilerini gösteren matris toplu olarak EK 1’de verilmiştir.

4.2. PROBLEMİN ABC ALGORİTMASI İLE ÇÖZÜMÜ

ABC algoritması, esinlendiği arı sürülerinin doğal davranışı itibarıyla; kâşif arıların yiyecek kaynağı araması ve buldukları kaynaktan aldıkları nektarla kovana

gittiklerinde yaptıkları salınım dansını izleyen gözcü arıların yiyeceğin kalitesi hakkında verdikleri karara göre o kaynağa gidilip kovana yiyecek taşınması mantığına dayalıdır. Bu mantığın üretim çizelgelemesi mantığına uyarlanmış hali ise şöyledir; sanal kâşif arılar problemin verilerine dayanarak çözüm uzayında dolaşırlar ve bir çözüm ararlar, buldukları çözümle ilgili sanal gözcü arılara aktardıkları bilgi neticesinde gözcü arılar bir karar verir, böylece en iyi çözümü yani problemimizdeki en kısa tamamlanma zamanını veren iş sıralamasını elde etmiş oluruz.

Problemin özüne sadık kalmak ve özelliklerini tam olarak yansıtmak için klasik ABC (nümerik ABC) üzerinde birtakım düzenlemeler yaparak algoritma kodu dolayısıyla da çalışma prensibi kesikli hale getirilmiştir. Bunun nedeni, klasik ABC algoritmasında örneğin (-1,1) arasında ondalıklı ve sürekli değerler alabilen çözümler üretilir. Ancak bizim tamsayılar halinde iş sıralamaları elde etmemiz gerekmektedir. Bu nedenle rasgele tamsayı dizileri üreten kod yazılarak istenen çözümler elde edilmiştir. Bu çözümlerde gerçekleşen minimum tamamlanma zamanları da amaç fonksiyonu değerleri olarak incelenmiştir.

Algoritmadaki kontrol edilebilir parametreler ve anlamları şöyledir:

NP: Koloni büyüklüğünü ifade eder ve işçi arılarla gözcü arıların toplamından oluşur. Kısaca popülasyon sayısı olarak da tanımlanabilir.

FoodNumber (NP/2): Yiyecek kaynağı sayısı, algoritmanın koloni büyüklüğünün yarısı şeklinde ifade edilir ve bu şekilde hesaplanır.

Limit=100: Limit sayısı kadar denemelerle iyileştirilememiş bir yiyecek kaynağı kendi işçi arısı tarafından terk edilir.

maxCycle: Yiyecek aramadaki toplam döngü sayısıdır, aynı zamanda bir durdurma kriteri olarak görev yapar.

Ayrıca bu değiştirilebilir parametrelerin yanında probleme özgü veriler olarak;

D=23: Problemin optimize edilecek parametrelerinin sayısıdır yani bizim problemde işlerin sayısıdır.

ub: Upper bound yani üst sınır olarak 1 satır 23 sütunluk birlerden oluşan bir matrisin 23 ile çarpımı $\{ ub = \text{ones}(1,D) * 23 \}$ şeklinde ifade edilir.

lb: Lower bound yani alt sınır olarak da yine 1 satır 23 sütunluk birlerden oluşan bir matrisin 1 ile çarpımı { lb= ones(1,D)*1 } şeklinde ifade edilir.

Runtime=30: Algoritmanın güçlülüğünü görmek ve çözümlerin doğruluğunun istatistiksel olarak anlam kazanması için algoritmanın 30 kez çalıştırıldığını ifade eder.

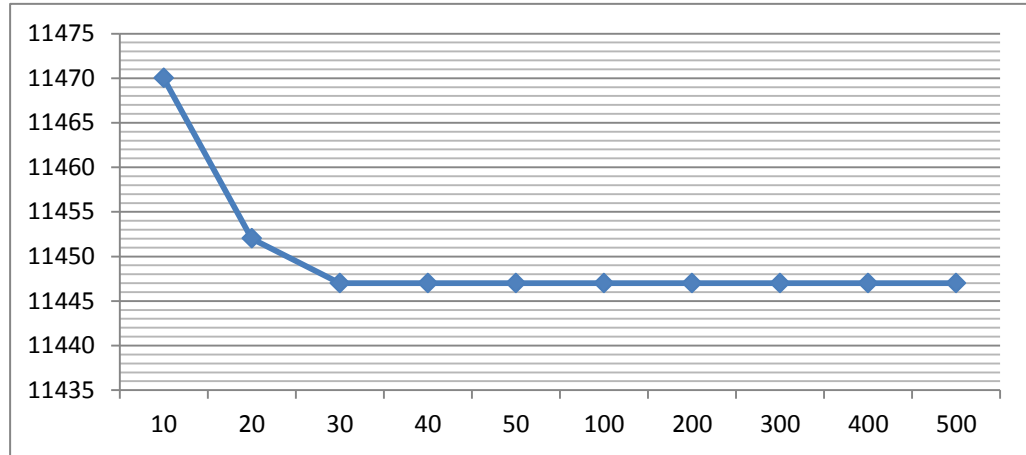
Bu verileri kullanarak ve kontrol edilebilir parametrelerde değişiklik yaparak en iyi çözüm (minimum makespan'a sahip iş sıralaması) elde edilmeye çalışılmıştır. Bu çalışmamızın sonucu olarak; elde edilen çözümler, iş sıralarının dizilimleri ve CPU işlem sürelerine ait bilgiler aşağıdaki tablolarda sunulmuştur (Tablo 1, Tablo 2, Tablo 3, Tablo 4). Verilen grafikler yardımıyla da makespan ve NP arasındaki ilişkiler gösterilmiştir (Şekil 13-14-15-16).

Tablo 1: İterasyon 1000 İken ABC Algoritmasının Çözüm Sonuçları

NP	SIRALAMA	CPU SÜRESİ (DK)	MAKESPAN
10	>>>> 6 21 14 23 15 22 20 10 11 3 13 9 2 1 12 18 17 16 4 8 5 7 19	2,74	11470
20	>>>> 21 19 23 15 22 6 20 1 12 11 3 2 10 16 14 18 17 13 9 8 4 7 5	5,22	11452
30	>>>> 12 10 3 6 21 19 23 17 14 15 16 18 22 20 2 1 11 13 9 8 5 7 4	6,78	11447
40	>>>> 10 6 21 18 23 14 7 4 8 19 15 22 20 9 3 11 12 1 13 17 16 5 2	8,85	11447
50	>>>> 14 23 6 21 3 12 8 19 15 2 11 10 22 20 13 5 4 18 16 17 1 7 9	10,97	11447
100	>>>> 12 9 18 13 11 10 2 23 15 14 6 21 22 20 7 8 3 1 16 17 19 5 4	22,15	11447
200	>>>> 9 6 20 21 19 23 15 14 4 1 11 13 2 3 10 12 22 17 18 16 5 7 8	43,3	11447
300	>>>> 21 12 3 20 18 23 10 11 15 19 16 22 1 13 2 8 6 17 14 9 4 7 5	63,22	11447
400	>>>> 9 1 10 2 18 23 19 14 4 21 3 12 13 11 15 17 5 8 7 6 20 16 22	84,26	11447
500	>>>> 21 23 9 3 13 18 17 14 15 19 20 1 12 10 11 2 8 22 6 16 4 5 7	104,95	11447

İterasyon (maxCycle parametresi) 1000 seçildiğinde NP 10 ve 20 iken sırasıyla 11470 ve 11452 makespan değerleri elde edilmiş daha sonraki NP değerlerinde yani

popülasyon sayısını artırdığımızda 11447 makespan değerine ulaşılmıştır. Sonuç olarak; 30 ve daha büyük boyutlardaki popülasyonlarda 11447 dk olan ve daha da iyileştirilemeyen optimum makespan değerine ulaşılmıştır (Tablo 1). Şekil 13’de bu durumun grafik üzerinde gösterimi vardır.

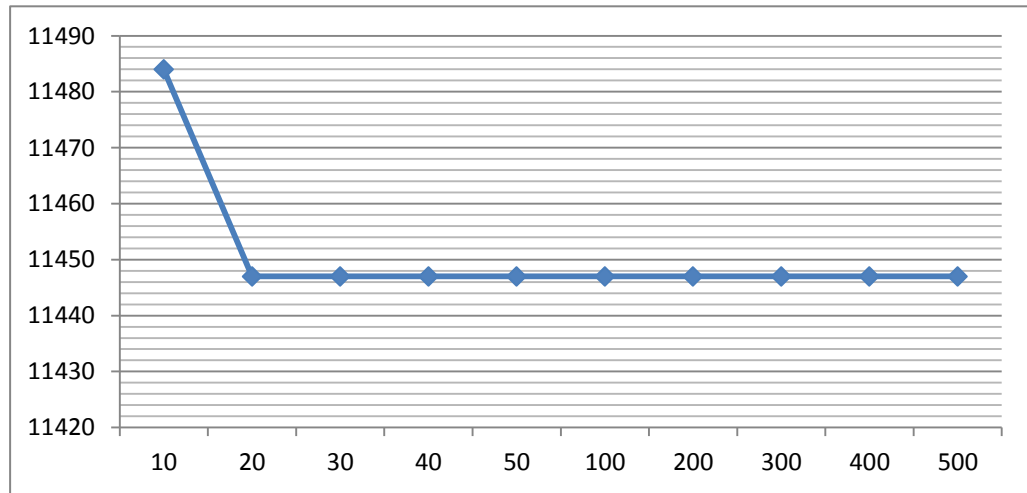


Şekil 13: İterasyon 1000 İken, Makespan ve NP İlişkisi

Tablo 2: İterasyon 5000 İken ABC Algoritmasının Çözüm Sonuçları

NP	SIRALAMA	CPU SÜRESİ (DK)	MAKESPAN
10	>>> 19 23 6 21 11 3 2 10 13 12 1 20 22 9 15 14 18 17 16 7 4 5 8	11,8	11484
20	>>> 19 23 9 6 21 20 22 18 17 14 15 16 2 3 13 1 12 11 10 4 5 7 8	22,37	11447
30	>>> 14 23 6 21 20 22 9 15 19 17 16 18 5 8 4 7 1 3 10 11 2 13 12	32,72	11447
40	>>> 9 14 23 21 22 12 11 6 20 19 18 16 15 17 1 2 13 5 7 8 4 10 3	43,04	11447
50	>>> 21 9 18 15 23 16 14 8 20 22 6 12 2 1 19 17 4 10 13 7 5 11 3	52,51	11447
100	>>> 10 12 14 15 3 20 21 19 23 8 2 1 13 22 17 18 16 11 5 7 4 6 9	108,8	11447
200	>>> 8 3 1 13 10 11 2 6 12 16 19 9 20 21 14 15 23 18 17 22 7 4 5	210,39	11447
300	>>> 10 12 3 2 6 21 14 23 19 16 18 15 17 20 22 9 1 11 13 8 7 4 5	326,96	11447
400	>>> 9 13 12 2 10 3 23 19 21 6 7 8 4 20 22 18 14 16 15 17 11 1 5	407,62	11447
500	>>> 14 19 23 9 12 6 21 20 16 15 11 3 10 2 1 13 22 8 4 5 7 18 17	518,86	11447

İterasyon 5000 seçildiğinde ise koloni büyüklüğümüz 10 iken 11484 dk ve büyüklüğü daha da artırdığımızda 11447 dk optimum makespan değerine ulaşılmıştır. Bu iterasyonda da koloni büyüklüğü 500'e kadar artırılmış ancak 11447 dakikadan daha iyi bir çözüm bulunamamıştır (Tablo 2). Şekil 14'de bu durum grafik yardımıyla gösterilmiştir.

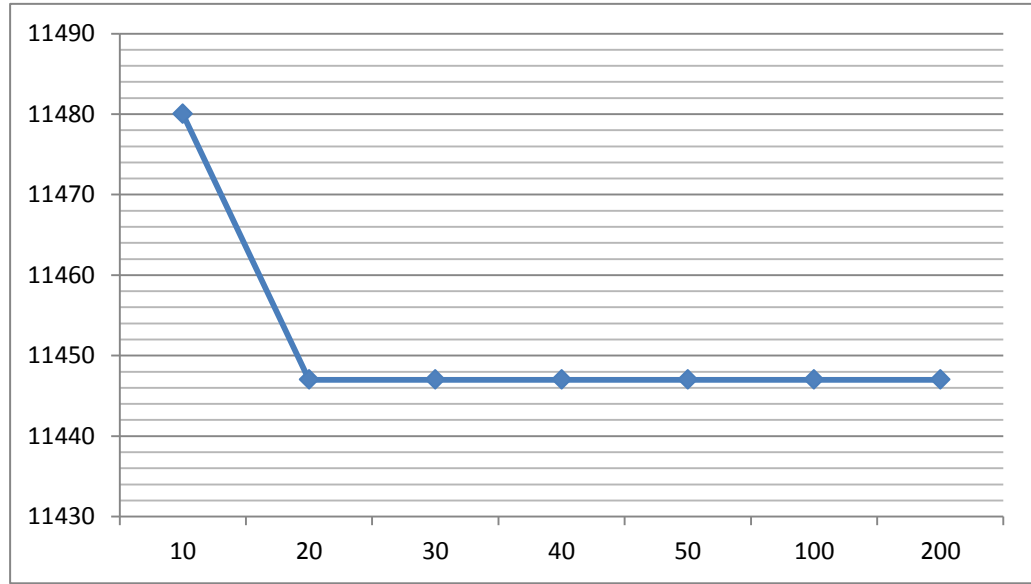


Şekil 14: İterasyon 5000 İken, Makespan ve NP İlişkisi

Tablo 3: İterasyon 10000 İken ABC Algoritmasının Çözüm Sonuçları

NP	SIRALAMA	CPU SÜRESİ (DK)	MAKESPAN
10	>>> 15 2 10 21 17 23 16 19 22 20 6 9 18 14 4 13 11 1 3 12 8 7 5	25,61	11480
20	>>> 3 19 23 20 21 4 8 16 14 18 9 17 15 2 13 12 1 11 10 6 22 5 7	45,81	11447
30	>>> 6 21 19 23 22 20 12 3 1 2 13 11 10 9 18 15 16 14 17 4 8 7 5	67,94	11447
40	>>> 2 10 19 14 23 3 6 21 15 18 9 11 13 20 22 17 16 1 12 8 7 4 5	88,32	11447
50	>>> 21 2 12 3 13 10 11 15 23 9 8 19 17 14 22 6 20 18 16 4 7 5 1	108,48	11447
100	>>> 6 21 9 23 13 19 18 16 15 14 20 8 22 11 3 2 1 10 17 12 4 7 5	210,33	11447
200	>>> 3 2 6 21 10 19 23 18 14 15 9 8 4 7 5 20 22 17 16 1 12 13 11	442,57	11447

İterasyon 10000 seçildiğinde ise koloni büyüklüğümüz 10 iken optimum değer olan 11447 dk elde edilememiş daha yüksek bir değer (11480) bulunmuştur. Büyüklük 20 yapıldığında ve daha da artırıldığında ise optimum çözüme ulaşılmıştır (Tablo 3). Bu durum Şekil 15’de gösterilmiştir.



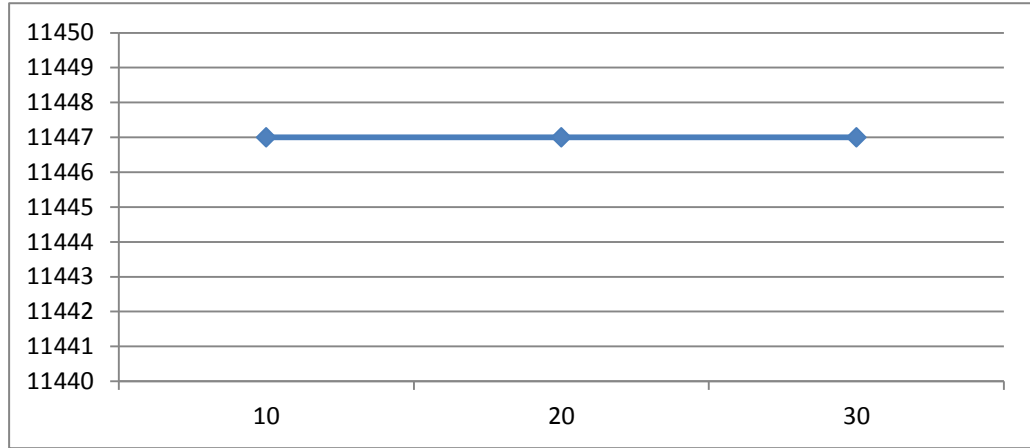
Şekil 15: İterasyon 10000 İken, Makespan ve NP İlişkisi

Tablo 4: İterasyon 50000 İken ABC Algoritmasının Çözüm Sonuçları

NP	SIRALAMA	CPU SÜRESİ (DK)	MAKESPAN
10	>>> 6 21 19 23 15 2 11 3 1 13 12 10 9 20 22 4 7 8 5 14 16 17 18	114,41	11447
20	>>> 9 19 23 6 21 20 22 17 14 18 15 16 7 5 8 4 2 10 1 11 13 12 3	212,71	11447
30	>>> 6 21 9 19 23 14 16 17 15 18 20 22 3 1 13 2 10 11 12 4 8 7 5	338,66	11447

İterasyon 50000 yapıldığında ise denenen en küçük koloni büyüklüğü olan 10 seçildiğinde bile 11447 dk olan makespan değerine ulaşılmıştır. Büyüklüğü artırdığımızda da makespan açısından bir değişme elde edilememiş optimum değer tekrarlanmıştır sadece çizelgedeki işlerin sıraları değişmiş ve CPU işlem süreleri

uzamıştır (Tablo 4). Şekil 16’da değişmeyen makespan durumunu gösteren grafik verilmiştir.



Şekil 16: İterasyon 50000 İken, Makespan ve NP İlişkisi

ABC algoritmasının bu probleme uyarlanması ile birlikte elde edilen çözümlerden görüleceği üzere en küçük tamamlanma zamanı (makespan) olarak bulunan minimum değer 11447 dk olmuştur. Farklı koloni büyüklükleri ve iterasyon sayıları denenerek test edilen geliştirdiğimiz bu ABC algoritması optimum amaç fonksiyonu değeri olarak 11447 değerini bulmuştur. Bu değeri en kısa sürede (6,78 dk) 1000 iterasyon ve 30 NP büyüklüğü ile vermiştir. Bu çözüme göre en iyi iş sıralaması da; “12 10 3 6 21 19 23 17 14 15 16 18 22 20 2 1 11 13 9 8 5 7 4” olarak gerçekleşmiştir.

4.3. PROBLEMİN GA İLE ÇÖZÜMÜ

Genetik algoritmalar, doğadaki genetik süreçten esinlenerek oluşturulduğu için en iyinin hayatta kalması ilkesine uygun olarak çalışır. Bunu yaparken de sanal olarak genler arasında çaprazlama, mutasyon ve seçme operatörleri kullanır; kromozomlar ve genler üzerinde çeşitlilik yaparak değişik nesiller üretmeyi hedefler. Böylece de değişik nesiller arasındaki bireylerden birinin optimum çözüm olabilme ihtimali artmaktadır.

Genetik algoritmalar yapısı itibariyle hem sürekli hem de kesikli optimizasyon yapabilmeye uygundur. Bu nedenle orijinal genetik algoritma kodunda sadece probleme özgü bir kaç değişiklik yaparak kesikli bir GA geliştirilmiştir. Geliştirdiğimiz GA, gezgin satıcı problemi ve çizelgeleme için kullanılabilecek olup kromozomlar permütasyon kodlama şeklinde oluşturulmuştur. Başlangıç popülasyonu, ABC

algoritmasında olduğu gibi random dizilim ile sağlanmıştır. Çaprazlama mantığı olarak da tek noktalı çaprazlama kullanılmıştır. Algoritmanın kontrol edilebilir parametreleri olan popülasyon boyutu, mutasyon oranı, seleksiyon oranı, iterasyon sayısı ve runtime değişkenlerinde çeşitli birçok değer girilerek hem istatistiksel olarak anlam kazanması hem de optimum çözümü yada optimum çözüme yakın çözümü bulması sağlanmıştır. Parametrelerle ilgili kısaca açıklama yapacak olursak;

Runtime=30: İstatistiksel olarak anlam kazanması için her bir farklı parametreyle algoritmanın 30 kez çalıştırıldığını ifade eder.

Popsiz: Popülasyondaki birey sayısını ifade eder.

Mutrate: Kullanılacak mutasyon oranını ifade eder. 0 ile 1 arasında bir değer alır. Oran büyüdükçe genelde çözümün kötüleştiği görülmüştür.

Selection: Kullanılacak seleksiyon oranını ifade eder. 0 ile 1 arasında bir değer alır. Oran büyüdükçe genelde çözümün kötüleştiği görülmüştür.

Maxit: Yapılacak maksimum iterasyon sayısını ifade eder. Durdurma kriteri olarak görev yapar. İterasyon arttıkça genelde çözümün iyileştiği görülmüştür.

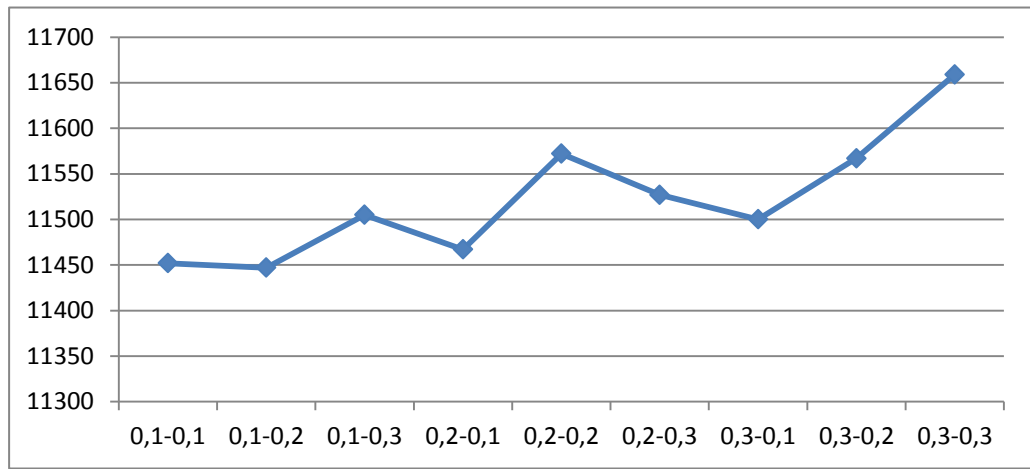
İşte bu parametrelerde değişiklik yapılarak ve problemin özüne uygun şekilde yeniden dizayn edilen algoritma yapısı kullanılarak, problemdeki işlerin toplam tamamlanma zamanının minimize edilmesi amaçlanmıştır. Elde ettiğimiz makespan değerleri, işlerin dizilimlerinin olduğu çizelgeler ve algoritma çözümü için harcanan CPU işlem süreleri aşağıdaki tablolarda verilmiştir (Tablo 5, Tablo 6, Tablo 7). Şekil 17-18-19’da popülasyon 50, 100 ve 200 iken iterasyonun 1000 olduğu durumlardaki ilişkiler gösterilmiştir.

Tablo 5: İterasyon 1000 İken GA Çözüm Sonuçları

Pop.	Mut.- Sel.	Sıralama	CPU (DK)	Makespan
10	0,1- 0,1	>>> 19 23 6 21 20 22 9 18 14 15 17 16 3 2 10 1 11 13 12 7 8 5 4	2,39	11452
20	0,1- 0,1	>>> 19 23 6 21 22 20 16 18 14 17 15 10 3 1 13 2 12 11 9 4 5 8 7	4,30	11452
30	0,1- 0,1	>>> 19 23 6 21 20 22 11 2 10 3 12 13 1 18 17 15 14 16 9 4 7 8 5	7,06	11452
40	0,1- 0,1	>>> 6 21 19 23 15 14 17 18 16 20 22 12 13 11 3 1 2 10 7 5 4 8 9	8,64	11447

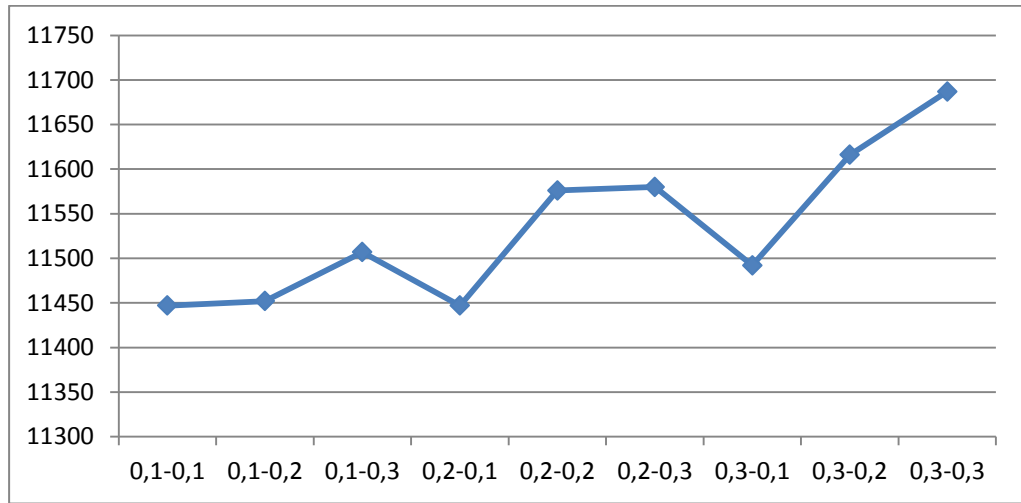
50	0,1-0,1	>>> 19 23 6 21 20 22 9 14 18 15 16 17 7 4 5 8 13 11 3 2 1 12 10	10,7	11452
	0,1-0,2	>>> 6 21 19 23 17 18 16 15 14 22 20 10 12 13 1 3 11 2 7 4 8 5 9	10,85	11447
	0,1-0,3	>>> 14 23 6 21 20 22 5 8 7 4 19 15 2 11 12 1 3 10 13 9 17 18 16	10,62	11505
	0,2-0,1	>>> 6 21 9 19 23 18 16 17 14 15 22 20 7 8 5 4 10 3 1 13 20 11 12	10,78	11467
	0,2-0,2	>>> 19 23 6 21 12 13 20 22 9 1 3 2 11 15 14 17 18 16 7 8 5 4 10	10,56	11572
	0,2-0,3	>>> 6 21 19 23 18 17 14 15 22 20 9 2 13 10 3 4 7 8 5 16 11 12 1	10,12	11527
	0,3-0,1	>>> 19 23 3 6 21 22 20 18 17 15 16 14 1 11 13 2 12 10 7 4 5 8 9	10,98	11500
	0,3-0,2	>>> 6 21 19 23 15 2 11 10 13 22 20 1 12 3 7 17 18 16 14 8 5 4	10,18	11567
	0,3-0,3	>>> 20 21 15 23 12 2 22 5 8 4 7 6 3 1 11 17 14 18 19 16 10 13 9	10,85	11659
100	0,1-0,1	>>> 6 21 19 23 16 17 18 14 15 22 20 7 4 8 5 3 10 13 2 11 12 1 9	19,97	11447
	0,1-0,2	>>> 19 23 6 21 20 22 12 1 2 10 13 11 3 17 18 15 16 14 9 4 8 5 7	19,03	11452
	0,1-0,3	>>> 6 21 9 19 23 15 18 14 16 20 22 12 11 1 13 2 3 10 8 7 5 4 17	18,83	11507
	0,2-0,1	>>> 6 21 19 23 15 14 18 17 16 9 22 20 1 13 2 11 12 10 3 4 8 5 7	20,31	11447
	0,2-0,2	>>> 20 21 19 23 16 15 17 14 18 22 5 8 4 7 13 12 1 10 11 6 2 3 9	21,48	11576
	0,2-0,3	>>> 6 20 21 14 23 22 12 19 17 16 15 18 5 8 7 4 9 10 13 1 11 3 2	21,22	11580
	0,3-0,1	>>> 19 23 6 21 20 22 18 16 15 17 11 13 1 2 10 12 3 14 9 5 8 7 4	21,68	11492
	0,3-0,2	>>> 20 21 19 23 16 15 17 14 5 4 12 11 10 13 1 9 6 22 2 3 7 8 18	21,66	11616
	0,3-0,3	>>> 6 21 19 23 22 20 12 1 10 13 15 17 4 8 9 3 11 2 14 7 16 18 5	21,48	11687
200	0,1-0,1	>>> 6 21 19 23 15 16 18 14 17 20 22 1 13 12 11 10 2 3 7 4 5 8 9	40,76	11447
	0,2-0,2	>>> 6 20 21 11 12 19 23 17 16 15 18 14 3 1 10 2 13 22 9 7 8 5 4	41,26	11540
	0,3-0,3	>>> 6 21 19 23 17 3 20 22 18 15 11 12 2 1 7 9 16 8 4 5 14 13 10	41,99	11687

İterasyon (Maxit parametresi) 1000 seçildiğinde popülasyonun 50, 100 ve 200 olduğu durumlarda farklı mutasyon ve seleksiyon oranları ile algoritma ayrı ayrı çalıştırılmıştır. Sonuçlar incelendiğinde tüm popülasyon boyutlarında en düşük elde edilen makespan değerinin 11447 dk olduğu görülmektedir. Mutasyon ve seleksiyon oranlarının artırılmaya başlamasıyla da çözüm değerinin genelde büyüdüğü ve dolayısıyla optimumdan uzaklaştığı görülmektedir. Şekil 17-18-19'da da bu durumlar grafiklerin yardımıyla gösterilmiştir.

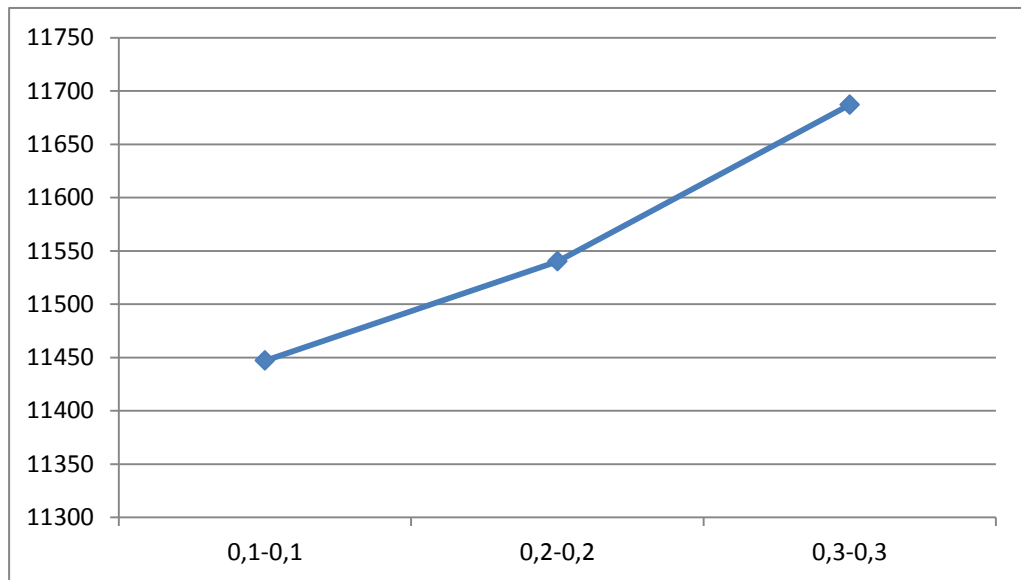


Şekil 17: İterasyon 1000 ve Popülasyon 50 İken Makespan ile Mut.-Sel. Oranının İlişkisi

Mutasyon ve seleksiyon oranlarının 0,1-0,1 ile başladığı ve 0,3-0,3 olduğu duruma kadarki makespan dağılımlarının grafiklerinde gittikçe artan bir eğilimin olduğu açıkça görülmektedir. Bu durumun da tüm popülasyon durumları için aynı olduğu söylenebilir. Seleksiyonun 0,1 seçildiği durumlarda ise genelde optimum çözüme ulaşıldığı ya da çok yaklaşıldığı görülmüştür. 0,3-0,3 mutasyon ve seleksiyon oranı seçildiğinde ise yine istisnasız her popülasyonda en yüksek makespan değeri elde edilmiştir.



Şekil 18: İterasyon 1000 ve Popülasyon 100 İken Makespan ile Mut.-Sel. Oranının İlişkisi



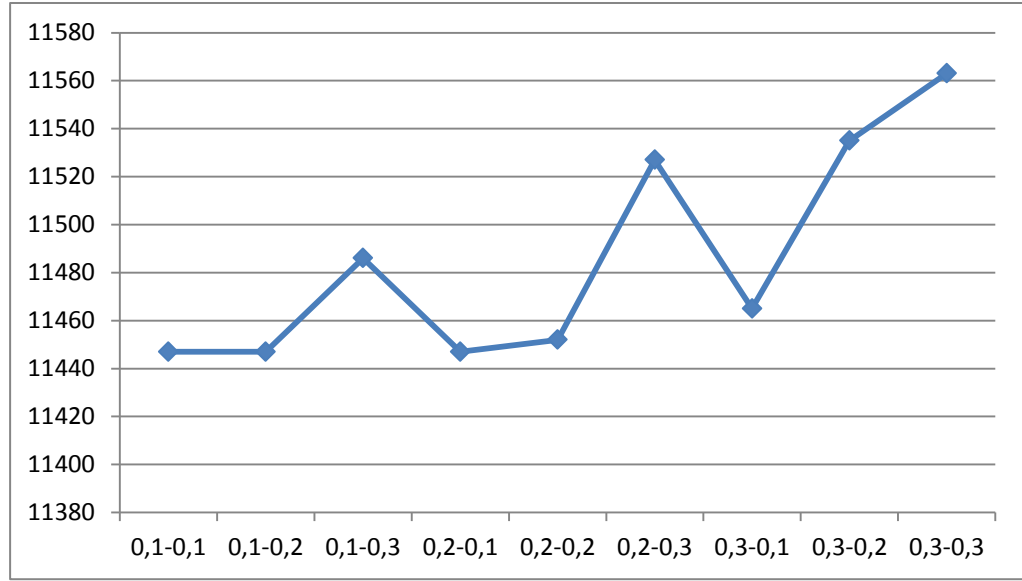
Şekil 19: İterasyon 1000 ve Popülasyon 200 İken Makespan ve Mut.-Sel. Oranının İlişkisi

İterasyon 5000 yapıldığında ise CPU işlem süreleri uzamıştır ancak iterasyon 1000 iken bulunandan daha fazla sayıda deneme çözümlerinde optimum değer olan 11447 değeri elde edilmiştir. İterasyonun 1000 olduğu duruma benzer şekilde mutasyon ve seleksiyon oranları artırıldıkça genellikle çözüm değeri optimumdan uzaklaşmıştır. Örneğin 0,3-0,3 oranlarının seçildiği durumlarda, tüm popülasyonlar için yapılan deneylerin çözümleri arasında en yüksek değerler elde edilmiş, minimum makespan amacından uzaklaşmıştır (Tablo 6).

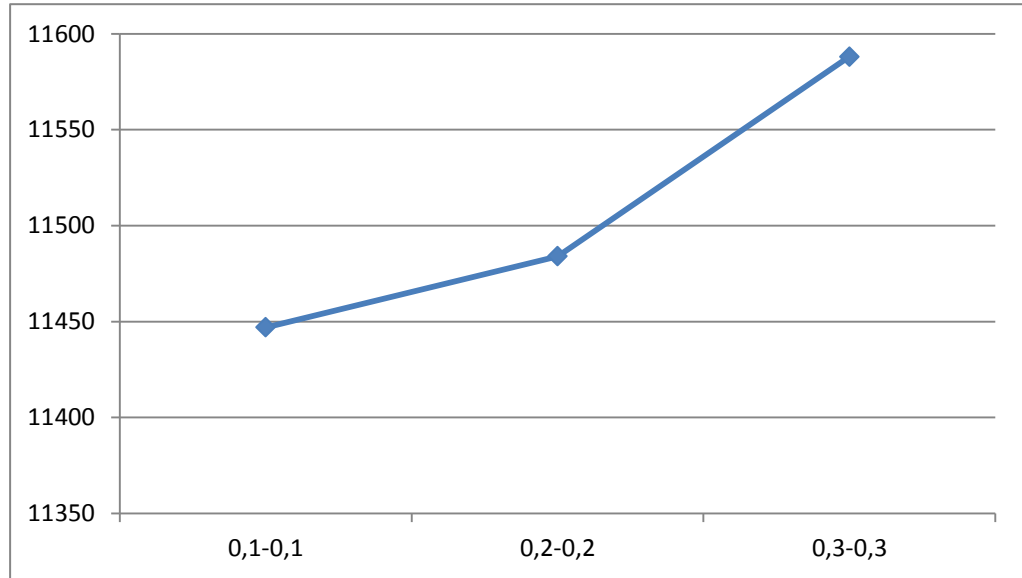
Tablo 6: İterasyon 5000 İken GA Çözüm Sonuçları

Pop.	Mut.-Sel.	Sıralama												CPU (DK)	Makespan
50	0,1-0,1	>>> 6 21 19 23 17 15 18 16 14 22 20 13 10 3 12 1 2 11 9 4 5 8 7	56,30	11447											
100	0,1-0,1	>>> 6 21 19 23 15 16 17 14 18 20 22 9 12 13 1 2 3 11 10 7 8 4 5	95,71	11447											
	0,1-0,2	>>> 6 21 19 23 17 18 14 15 16 20 22 3 10 13 12 1 2 11 9 4 8 5 7	96,75	11447											
	0,1-0,3	>>> 6 21 18 23 19 15 17 14 16 9 22 20 2 3 12 11 10 13 1 4 8 7 5	97	11486											
	0,2-0,1	>>> 6 21 19 23 14 18 16 15 17 9 22 20 1 10 11 3 13 12 2 5 7 4 8	99	11447											
	0,2-0,2	>>> 19 23 6 21 20 22 7 4 8 5 18 15 14 17 16 9 12 3 2 13 10 11 1	102,55	11452											
	0,2-0,3	>>> 6 21 19 23 16 15 18 13 2 11 1 3 10 12 20 22 7 4 14 17 5 8 9	107,97	11527											
	0,3-0,1	>>> 14 23 6 21 20 22 18 16 17 15 19 5 8 7 4 9 10 13 3 12 11 1 2	99,65	11465											
	0,3-0,2	>>> 14 23 15 6 21 22 9 8 4 5 7 20 19 17 18 16 1 13 10 12 3 11 2	103,71	11535											
	0,3-0,3	>>> 18 23 6 21 20 22 11 3 2 10 13 12 7 4 17 14 15 16 19 9 8 5 1	106,75	11563											
200	0,1-0,1	>>> 6 21 19 23 15 14 16 18 17 22 20 3 1 2 12 10 13 11 7 8 4 5 9	215,55	11447											
	0,2-0,2	>>> 19 23 20 21 6 22 18 14 16 15 17 12 3 2 1 10 3 11 4 5 7 8 9	206,38	11484											
	0,3-0,3	>>> 9 12 19 23 6 21 22 20 1 10 13 17 15 18 8 4 7 5 3 11 2 16 14	199,48	11588											

Şekil 20 ve Şekil 21'deki grafikler incelendiğinde, oranların artması ile makespan değerlerinin de doğru orantılı olarak arttığı görülmüştür. Grafikte bazı oranlarda küçük düşmeler olsa da genel trend, oranların artmasıyla makespan'ın artacağı yönündedir. Ayrıca mutasyon veya seleksiyon oranlarından birinin sabit tutulduğu durumlarda diğer oran artırılınca makespan değeri de artmıştır.



Şekil 20: İterasyon 5000 ve Popülasyon 100 İken Makespan ve Mut.-Sel. Oranının İlişkisi



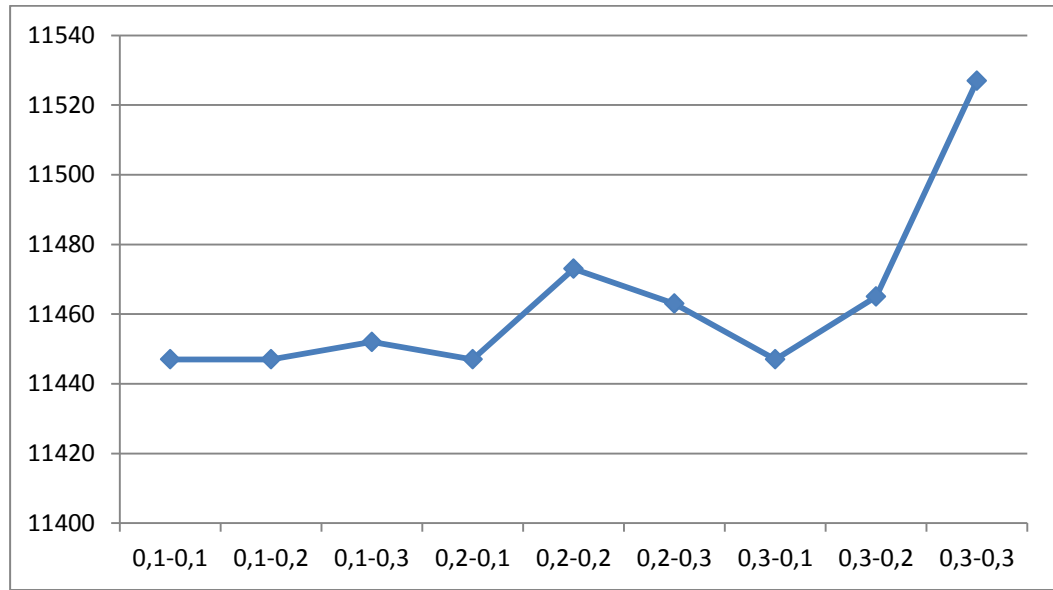
Şekil 21: İterasyon 5000 ve Popülasyon 200 İken Makespan ve Mut.-Sel. Oranının İlişkisi

İterasyon 10000 yapıldığında da CPU işlem süreleri iyice uzamıştır ancak seleksiyon oranının 0,1 seçildiği tüm denemelerde optimum çözüm değeri olan 11447 makespan değerine ulaşılmıştır. Sonuçlar incelendiğinde hem en kısa sürede çözümü bulması hem de optimum çözümü vermesi nedenleriyle, her popülasyonda oranların 0,1-0,1 seçildiği durumlar en etkin deneme çözümleri olarak görülmektedir (Tablo 7).

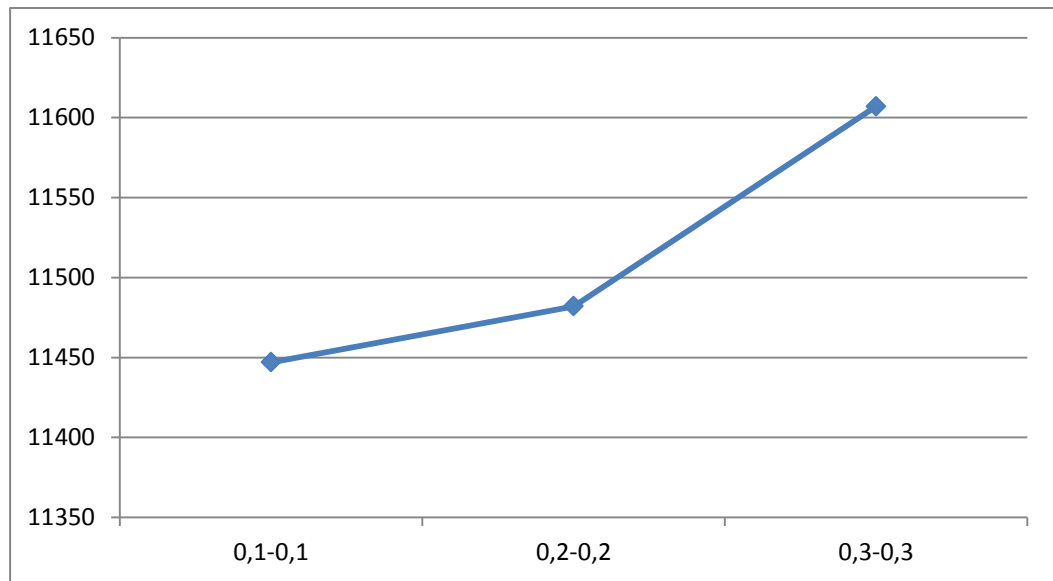
Tablo 7: İterasyon 10000 İken GA Çözüm Sonuçları

Pop.	Mut.-Sel.	Sıralama												CPU (DK)	Makespan
50	0,1-0,1	>>> 20	6 12	21 11	19 13	23 3	14 1	18 10	15 2	16 4	17 7	9 8	22 5	110,11	11447
100	0,1-0,1	>>> 9	6 3	21 11	19 10	23 13	15 2	16 1	18 12	14 7	17 4	22 8	20 5	192,7	11447
	0,1-0,2	>>> 3	6 2	21 13	19 11	23 10	14 1	16 12	17 5	15 4	18 8	22 7	20 9	204,47	11447
	0,1-0,3	>>> 4	19 5	23 7	6 8	21 9	20 10	22 3	16 11	18 12	14 13	15 2	17 1	196,38	11452
	0,2-0,1	>>> 22	6 13	21 12	19 3	23 10	15 11	18 1	16 2	17 8	14 7	9 5	20 4	205,13	11447
	0,2-0,2	>>> 3	18 11	23 15	6 14	21 16	20 17	22 19	2 9	13 7	1 4	12 8	10 5	213,47	11473
	0,2-0,3	>>> 14	9 5	19 7	23 8	6 4	21 3	20 12	22 1	16 2	17 13	18 10	15 11	205,38	11463
	0,3-0,1	>>> 4	6 8	21 5	19 7	23 12	14 13	18 2	16 3	15 1	17 11	20 10	22 9	209,75	11447
	0,3-0,2	>>> 2	14 3	23 12	6 10	21 11	20 13	22 1	18 5	19 8	15 7	16 4	17 9	207,64	11465
	0,3-0,3	>>> 12	6 11	21 2	19 10	23 9	18 22	15 7	16 5	14 8	17 4	20 1	13 3	210,83	11527
200	0,1-0,1	>>> 20	6 2	21 10	19 11	23 1	18 12	15 3	16 13	17 4	14 5	9 8	22 7	413,98	11447
	0,2-0,2	>>> 11	19 3	23 1	15 10	6 13	21 12	22 2	20 4	14 8	16 5	18 7	17 9	409,41	11482
	0,3-0,3	>>> 22	6 15	21 17	19 18	23 7	10 14	12 16	3 4	11 5	1 8	2 9	20 13	394,27	11607

Şekil 22 ve Şekil 23 incelendiğinde ise, diğer iterasyonlardaki çözümlere göre nispeten daha yakın çözümler elde edildiği için grafikteki dalgalanmaların daha küçük olduğu görülmektedir. Çünkü iterasyon sayısının artması çözümleri daha da standart bir değere, optimuma yakınlıştırmaktadır. Grafiklerdeki genel trendin yine doğrusal artan bir eğilimde olması beklenen bir durumdur ve oranları 0,3-0,3 seçtiğimiz örnek denemelerinde en yüksek makespan değerleri elde edilmiştir. Yine şüphesiz en kısa zamanda çözümü sunan ve optimum sonucu veren 0,1-0,1 oranlı çözüm denemeleri en küçük makespan değerleri ile grafiklerin ilk ve en düşük noktalarını temsil etmektedir.

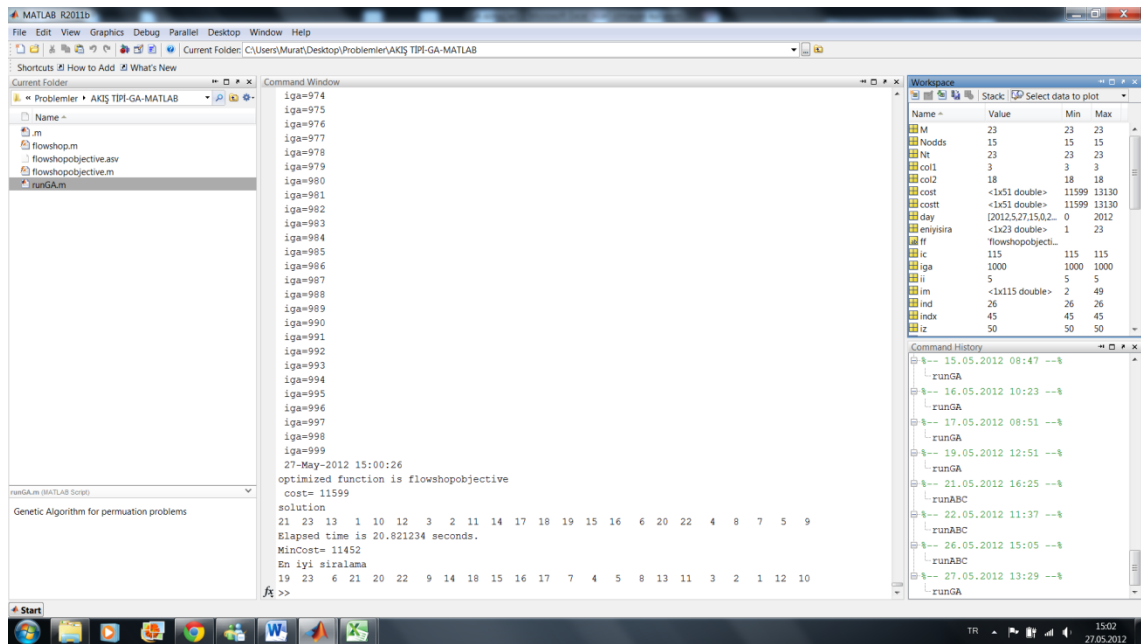


Şekil 22: İterasyon 10000 ve Popülasyon 100 İken Makespan ve Mut.-Sel. Oranının İlişkisi



Şekil 23: İterasyon 10000 ve Popülasyon 200 İken Makespan ve Mut.-Sel. Oranının İlişkisi

Problemin Matlab’de GA ile çözümlerinin her birinin sonunda algoritma bize geçen zamanı, en düşük maliyeti (amaç fonksiyonu değeri) ve en iyi sıralamayı vermiştir. İşte bu çözümlerden birinin gösterildiği ekran görüntüsü Şekil 24’de verilmiştir.



Şekil 24: Matlab Programında GA Bir Çözüm Bulmuşken Ekran Görüntüsü

4.4. PROBLEMİN LEKİN PROGRAMI İLE ÇÖZÜMÜ

Lekin paket programı, New York Üniversitesi'nde akademik amaçla geliştirilmiş olan bir üretim çizelgeleme programıdır. Birtakım çizelgeleme algoritmalarını ve sezgisel yöntemleri içermektedir. Bununla birlikte Lekin, kullanıcıya kendi sezgisel yöntemlerini programa ekleme ve performansını programda mevcut olan algoritmalar ve sezgisel yöntemlerle karşılaştırma imkânı da verecek şekilde tasarlanmıştır (Pinedo, 2008, p. 537).

Lekin paket programında bulunan ve 2'den fazla iş ve makineli problemlerde artan popülerliği olan Shifting Bottleneck (SB) yöntemi, makinelerdeki darboğazı öteleme metodu olarak da anılır. Yöntemin uygulama adımları şöyledir (Altındaş, 2011:42-44):

Adım 1: Başlangıç şartlarının oluşturulması

Adım 2: Seçilecek makinenin analizi

Adım 3: Darboğaz seçimi

Adım 4: Daha önce çizelgelenmiş işlerin yeniden çizelgelenmesi

Adım 5: Durum şartları kontrolü

Lekin'deki birtakım çizelgeleme kuralları ve anlamları da şöyledir (Pinedo, 2008):

- En Küçük Serbest Zaman (MS)
- Hazırlık Süreli Görünen Gecikme Maliyeti (ATCS)
- En Erken Bitiş Süresi (EDD)
- İlk Gelen İlk İşlem Görür (FCFS)
- En Uzun İşlem Süresi (LPT)
- En Kısa İşlem Süresi (SPT)
- Ağırlıklı En Kısa İşlem Süresi (WSPT)
- Kritik Oran (CR)

Bu bölümde uygulama problemimize Lekin kullanarak SB prosedürü ve bu çizelgeleme kuralları uygulanmıştır. Problem çeşitli amaçlar ve kurallar için ayrı ayrı çözülmüş ve sonuçlar tablolar halinde aşağıda sunulmuştur (Tablo 8 ve Tablo 9). SB prosedürü için minimum makespan amacının çözümünü dikkate alırsak; Lekin paket programının bize 60 saniyede 30 kez çalışarak 12363 dk amaç fonksiyonu değerini verdiğini görürüz. Aynı zamanda bu değer tüm amaçlar içinde SB prosedürü ile elde edilmiş en küçük değerdir. En yüksek değer ise 14178 dk ile toplam gecikme, ağırlıklı gecikme ve maksimum gecikme amaçlarında ortaya çıkmıştır. CPU işlem süreleri açısından da, program SB prosedüründe tüm amaçlarda 60 saniyede çözüm sonucunu sunmuştur (Tablo 8).

Problemin çizelgeleme kurallarıyla çözüm sonuçlarını gösteren Tablo 9 incelendiğinde en iyi değeri LPT yani en uzun işlem süresi çizelgeleme kuralının verdiği görülmektedir. Kural, 30 kez çalışarak 30 saniye CPU işlem zamanıyla 11919 dk makespan değerini bulmuştur. En yüksek makespan değerini ise 14347 dk ile CR (Kritik Oran) vermiştir. Aynı zamanda CR çözümünde teslim sürelerine göre geciken 11 adet iş mevcuttur ve toplam gecikme 3207 dk hesaplanmıştır. SPT ve WSPT kuralları birbirleriyle aynı sonuçları vermiştir, çünkü işlerin kendi aralarında bir ağırlığı, önceliği yoktur. Program, çizelgeleme kurallarının her biri için bize 30 saniyede çözüm vermiştir. Ayrıca çizelgeleme kuralları ve SB sezgiselinde geciken iş sayıları ve işlerin gecikme süreleri ile ilgili çözümleri elde etmemiz için bilinmesi gereken işlerin teslim süreleri EK 1'de verilmiştir.

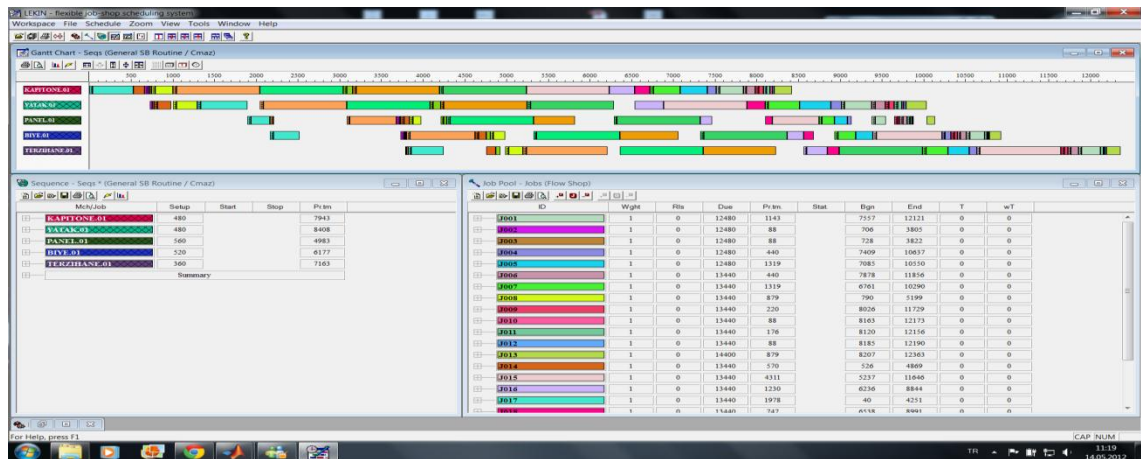
Tablo 8: Problemin Lakin Programında SB İle Çözümü

Amaç (Minimize)	Makespan	Toplam Akış	CPU (sn)
Makespan	12363	204209	60
Toplam Akış Zamanı	12902	133379	60
Toplam Gecikme	14178	245248	60
Top. Ağırlıklı Gecikme	12902	133379	60
Ağırlıklı Gecikme	14178	245248	60
Maksimum Gecikme	14178	245248	60

Tablo 9: Problemin Çizelgeleme Kurallarıyla Çözümü

Kural	Makespan	Toplam Akış	Max.Gecikme	Toplam Gecikme	Geciken İş	CPU (sn)
ATCS	14095	245322	0	0	0	30
EDD	12831	180820	0	0	0	30
MS	13038	232249	0	0	0	30
FCFS	12586	109321	2026	2026	1	30
LPT	11919	236652	0	0	0	30
SPT	12619	80552	900	900	1	30
WSPT	12619	80552	900	900	1	30
CR	14347	280559	694	3207	11	30

Problemde minimum makespan bulunması için SB ile çözümüne dair bir Lakin ekran görüntüsü Şekil 25’de verilmiştir. Şeklin üst bölümünde çözümün Gantt şemasında gösterimi bulunmaktadır. Sağ alt bölüm olan iş havuzunda, 23 tane işin teslim süreleri, varsa işlerin öncelikleri ve ön hazırlık süreleri, işlem süreleri ve makineye girdikleri ile çıktıkları süreler gösterilmektedir. Sol alt bölüm olan makine parkında ise, her bir makinedeki toplam hazırlık zamanları ve toplam tamamlanma zamanları verilmektedir. Aynı zamanda makinelerin her birinde her bir işin hazırlık, makineye giriş ve makineden çıkış sürelerini ayrı ayrı izlemek mümkündür.



Şekil 25: Lekin Programından SB İle Cmax Bulunmasına Dair Ekran Görüntüsü

4.5. ÇÖZÜM YAKLAŞIMLARININ GENEL ANALİZİ

Probleme özgü olarak geliştirdiğimiz ve problemimiz için uygun çözüm sonuçları veren metasezgisel algoritmalar (ABC ve GA) ile Lekin paket programındaki çizelgeleme kurallarıyla SB sezgiselinin karşılaştırmalı analizleri bu bölümde sunulmuştur (Tablo 10, Tablo 11 ve Tablo 12).

Tablo 10: ABC ve GA Çözüm Sonuçlarının Karşılaştırılması

İterasyon	Pop.	ABC		GA		
		Makespan	CPU (dk)	Mut.-Sel.	Makespan	CPU (dk)
1000	10	11470	2,74	0,1-0,1	11452	2,39
	20	11452	5,22	0,1-0,1	11452	4,30
	30	11447	6,78	0,1-0,1	11452	7,06
	40	11447	8,85	0,1-0,1	11447	8,64
	50	11447	10,97	0,1-0,2	11447	10,85
	100	11447	22,15	0,1-0,1	11447	19,97
	200	11447	43,3	0,1-0,1	11447	40,76
5000	50	11447	52,51	0,1-0,1	11447	56,30
	100	11447	108,8	0,1-0,1	11447	95,71
	200	11447	210,39	0,1-0,1	11447	215,55
10000	50	11447	108,48	0,1-0,1	11447	110,11
	100	11447	210,33	0,1-0,1	11447	192,7
	200	11447	442,57	0,1-0,1	11447	413,98
50000	10	11447	114,41	0,1-0,1	11452	124,49
	20	11447	212,71	0,1-0,1	11452	207,97
	30	11447	338,66	0,1-0,1	11447	325,39

Tablo 10’da verilen karşılaştırmalı verilere baktığımızda verilen tüm 40 ve üzeri popülasyon boyutlu denemelerde 11447 dk olan optimum makespan değerine ulaşıldığı görülmektedir. Genetik algoritma için aynı popülasyon düzeyinde farklı oranlar kullanılarak farklı değerler elde edildiği için bu değerler içinden en küçük değeri veren örnekler dikkate alınmıştır. Yine bu popülasyonlar ve iterasyonlarda ABC algoritmasının da verdiği değerler alınarak karşılaştırma yapmak için uygun hale getirilmiştir.

İterasyon 1000 iken küçük popülasyonlar denenerek yöntemlerin en küçük hangi popülasyonda ve en az kaç dakikada çözüm sonucu verdikleri de karşılaştırılmak istenmiştir. Bu bağlamda, ABC algoritması en küçük olarak 2,74 dakikada 11470 dk makespan değerini bulurken, GA daha iyi bir çözüm olan 11452 dk makespan değerini 2,39 dakikada bulmuştur. Ancak ABC algoritması popülasyon boyutu 30 iken optimum değer olan 11447 sonucunu 6,78 dakikada bulurken GA bu popülasyonda 7,06 dk çalışarak 11452 değerini bulmuştur. Bu nedenle ABC algoritmasının daha kısa zamanda ve daha küçük popülasyonda optimum değere ulaşması ile genetik algorithmadan bu karşılaştırmada üstün olduğunu söyleyebiliriz.

Artan popülasyonlar ve iterasyonlar dikkate alındığında optimum 11447 değerini elde etmek her iki algoritma için de mümkün olup sadece elde etmedeki CPU işlem sürelerinde ufak farklılıklar yaşanmıştır. Bu sürelere bakıldığında GA’nın birçok örnekte biraz daha kısa sürede aynı çözümü sunduğu görülmüştür.

CPU işlem süreleri açısından tüm yöntemleri karşılaştırdınca en iyi çözümler sunan yöntemler arasında en kısa sürede çözüm veren yöntem LPT çizelgeleme kuralıdır. Bu kural, en uzun işlem süresi olan işlemin çizelgelemede en önce yer alması gerektiği mantığı üzerinedir. Bu kurala göre de 0,5 dakikada 11919 dk makespan değerine ulaşılmıştır. Diğer yöntemlerde de elde edilen en düşük CPU süreleri belirlenerek karşılaştırma yapılmıştır. Bu karşılaştırmaya göre en düşük popülasyonda en iyi sonucu (11452) veren GA, 2,39 dakikada en yüksek değeri veren SB prosedürü ise 1 dakikada çözüm vermiştir (Tablo 11).

Tablo 11: ABC, GA ve Lakin Sonuçları CPU Analizi

Yöntem	Makespan	CPU (dk)
ABC	11470	2,74
GA	11452	2,39
SB	12363	1
LPT	11919	0,5

Tablo 12’de ise, tüm yöntemlerden elde edilen en iyi sonuçlar ve bu sonuçların elde edildiği en düşük CPU zamanları beraber verilmiştir. Bu tabloya göre optimum değeri (11447) veren iki algoritmadan ABC algoritması, en kısa sürede (6,78 dk) bu değeri vermiştir. Bu nedenle karşılaştırılan dört yöntem arasında ABC algoritmasının en etkin algoritma olduğunu ifade edebiliriz.

Tablo 12: ABC, GA ve Lakin Sonuçları Makespan Analizi

Yöntem	Makespan	CPU (dk)
ABC	11447	6,78
GA	11447	8,64
SB	12363	1
LPT	11919	0,5

Yapılan tüm uygulamalar bilgisayar ortamında 30 tekrar çalışma ile yürütülmüştür. Tüm uygulama örnekleri Intel Core i7 - 2,0 Ghz işlemcili 4 GB Ram bellekli bilgisayarda çalıştırılmıştır. GA ve ABC algoritmaları, Matlab dilinde kodlanmış ve tüm uygulama örnekleri MATLAB R-2011b programı ile çalıştırılmıştır. Lakin paket programında ise SB prosedürü ve çizelgeleme kuralları örnekleri çalıştırılmıştır.

SONUÇ

Bu tez çalışmasında, akış tipi çizelgeleme problemlerinin çözümünde son yıllarda önem kazanan ABC ve GA metasezgisel yaklaşımları, gerçek hayattan derlenen örnek bir test problemi üzerinde uygulanarak elde edilen sonuçlar kendi aralarında ve SB sezgiseli ile bazı çizelgeleme kurallarına göre elde edilen sonuçlarla karşılaştırılmıştır.

Uygulamada kullandığımız test problemi, üretim çizelgeleme problemleri içinde akış tipi çizelgeleme problemlerine uygun özelliktedir. Problemden 23 işin 5 farklı makinenin her birinden aynı sırayla ve rotayla geçerek üretilmesi söz konusudur ve amaç fonksiyonumuz bu 23 işin toplam tamamlanma zamanını (makespan) minimize etmektir.

Çalışmada, en küçük tamamlanma zamanı değerini ve optimum çizelgeleri bulmak için son zamanlarda geliştirilmiş metasezgisellerden ABC algoritması ve genetik algoritmalar Matlab programı ile kodlanarak kullanılmıştır. Ayrıca Legin paket programında, SB sezgiseli ve bazı çizelgeleme kuralları test problemimize uygulanmıştır. Tüm yöntemlerin sonuçları tablolar halinde sunulmuş ve en sonunda karşılaştırmalı analizler yapılmıştır. GA ve ABC algoritmalarının çözümlerinde optimum olarak tespit edilen sonuca ulaşılırken SB sezgiseli ve çizelgeleme kurallarının çözümlerinde bu optimum sonuca ulaşamamıştır.

Uygulama problemi için kullanılan tüm yöntemlerin genel karşılaştırması yapıldığında ABC ve GA yöntemleri, hız ve optimal olma konularında SB sezgiseli ve çizelgeleme kurallarına göre üstün çıkmıştır. ABC ve GA yöntemlerini kendi aralarında karşılaştırdığımızda ise; ABC algoritmasının optimum değer olarak kabul ettiğimiz 11447 dk olan tamamlanma zamanı değerine en kısa sürede (6,78 dk) ulaşması nedeniyle GA'ya göre etkin bir metasezgisel olduğu görülmüştür. Ancak parametrelerde değişiklik yaparak çeşitli popülasyonlar, iterasyonlar ve oranlar kullanılarak yapılan deney tasarımı; GA'nın çoğu kez ABC algoritmasından hızlı çözüm verdiği tespit edilmiştir. Bu nedenle GA'nın da etkin bir metasezgisel olduğunu, ancak en az popülasyonla en kısa sürede optimum sonuca ulaşmak konusunda ABC algoritmasının gerisinde kaldığını söyleyebiliriz.

ABC algoritmasının en kısa sürede bulduğu bu optimum tamamlanma zamanı değerine göre çizelgelenen iş sıralaması şöyle gerçekleşmiştir:

“12 10 3 6 21 19 23 17 14 15 16 18 22 20 2 1 11 13 9 8 5 7 4”

Ayrıca, Matlab programında kodlanarak problemin tüm özelliklerini bünyesinde barındıran geliştirilmiş bu GA ve ABC algoritmalarının 11447 dk tamamlanma zamanı çözüm sonucunu veren tüm diğer sıralamaları da problem için kullanılabilir birer çizelge olmaktadır. Çünkü amaç en küçük tamamlanma zamanına sahip çizelgeyi kullanarak optimum sürede bu 23 işi tamamlamaktır. Çizelgenin hangi CPU süresinde elde edildiği bu aşamada ikinci planda kalmaktadır. Ancak algoritma performanslarının çalışma konumuz olması nedeniyle, optimum tamamlanma zamanı değerine sahip çizelgeyi en az CPU süresinde veren ABC algoritmasının bu problemlerin çözümünde tercih edilmesi önerilir.

Lekin sonuçlarını kendi içinde değerlendirdiğimizde tüm çizelgeleme kuralları ve SB sezgiseli sonuçları arasında, LPT (Longest Processing Time) kuralının 30 saniyede 11919 dk tamamlanma zamanı değeri ile en iyi çözümü sunduğu görülmüştür. Böylelikle en uzun işlem süresi olan işi daha önce çizelgelemenin tamamlanma zamanı amaç fonksiyonunu minimize etmede daha etkili olduğu söylenebilir.

Çalışmada sadece iki metasezgisel (ABC ve GA) kullanılmıştır. Daha fazla metasezgisel kullanılamamasının nedeni, algoritmaların kodlanması ve uygulanması için uzun zamanlar gerekmesidir. Çünkü algoritmaların çizelgeleme problemlerine uyarlanması aşamasında kesikli forma dönüştürme gereklidir ve algoritmalara büyük parametreler girildiğinde uzayan CPU zamanlarıyla karşılaşmaktadır.

Çalışma literatüre etkin bir şekilde çalışan, iterasyon sayılarındaki artışlarla ve parametrelerdeki değişimlerle optimum sonucu veren iki tane geliştirilmiş algoritma (ABC ve GA) sunmuştur. Özellikle ABC algoritmasının henüz yeni geliştirilmiş olması ve kesikli halde literatürde fazlaca uygulamasının olmaması nedeniyle akış tipi çizelgeleme problemlerine uygulanmış bir çalışmanın literatüre sunulması önemlidir.

Son olarak; çalışmanın, yeterli zaman ve imkân sağlandığında daha fazla metasezgiselle çalışılarak çeşitlilik anlamında iyileştirilebileceğinden bahsedebiliriz. Ayrıca, gerçek hayattan alınan farklı problemlerde metasezgisel çözüm metotlarını

karşılaştırmayı ve literatürden alınmış standart test problemleri kullanarak da bu metasezgisellerin etkinliğini ölçmeyi gelecek çalışmalar için önerebiliriz.

KAYNAKÇA

- Akay, Bahriye. (2009). *Nümerik Optimizasyon Problemlerinde Yapay Arı Kolonisi (Artificial Bee Colony) Algoritmasının Performans Analizi*. Erciyes Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı. Yayımlanmamış Doktora Tezi. Kayseri.
- Akçay, Erkan. (2009). *Akış Tipi İş Çizelgeleme Problemlerinin Yapay Bağışıklık Sistemi İle Çok Amaçlı Optimizasyonuna Yönelik Bir Model Önerisi*. Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Doktora Tezi. Kocaeli.
- Akgündüz, Onur Serkan. (2008). *Solving Mixed-Model Assembly Line Sequencing Problem Using Adaptive Genetic Algorithms*. Dokuz Eylül University Graduate School of Natural and Applied Sciences. Thesis for The Degree of Master of Science in Industrial Engineering. İzmir.
- Aladağ, Alper. (2010). *Tekrar İşlemeli Esnek Atölye Tipi Çizelgeleme Problemi İçin Yapay Bağışıklık Sistemi İle Bir Çözüm Yaklaşımı*. Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Eskişehir.
- Allahverdi, Ali. (2006). “Two-Machine Flowshop Scheduling Problem to Minimize Total Completion Time with Bounded Setup and Processing Times”. *Int. J. Production Economics*, 103, 386-400.
- Allahverdi, Ali. (2008). “Three-Machine Flowshop Scheduling Problem to Minimize Makespan with Bounded Setup and Processing Times”. *Journal of the Chinese Institute of Industrial Engineers*, 25 (1), 52-61.
- Allahverdi, Ali, Tariq Aldowaisan and Yuri N. Sotskov. (2003). “Two-Machine Flowshop Scheduling Problem to Minimize Makespan or Total Completion Time with Random and Bounded Setup Times”. *International Journal of Mathematics and Mathematical Sciences*, 39, 2475-2486.

- Altındaş, Murat. (2011). *Üretim Çizelgeleme ve Bir Uygulama*. Bozok Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Yayınlanmamış Yüksek Lisans Tezi. Yozgat.
- Ayan, K. ve U. Kılıç. (2011, May). “Optimal Güç Akışı Probleminin Çözümü İçin GA, MA ve YAK Algoritmalarının Karşılaştırılması”. *6th International Advanced Technologies Symposium (IATS’11)*, Elazığ.
- Aydemir, Erdal. (2009). *Atölye Tipi Çizelgeleme Problemlerinin Öncelik Kuralı Tabanlı Genetik Algoritma Yaklaşımıyla Simülasyon Destekli Optimizasyonu*. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Isparta.
- Azizi, Nader, M. Liang and S. Zolfaghari. (2009). “Hybrid Simulated Annealing in Flow-Shop Scheduling: A Diversification and Intensification Approach”. *Int. J. Industrial and Systems Engineering*, 4, 326-348.
- Baker, Kenneth R. and Dan Trietsch. (2009). *Principles of Sequencing and Scheduling*. New Jersey: John Wiley & Sons Inc.
- Bali, Özkan. (2009). *Stokastik U-Tipi Montaj Hattı Dengeleme Problemleri İçin Parçacık Sürü Optimizasyonu*. Gazi Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği. Doktora Tezi. Ankara.
- Ben-Daya, M. and M. Al-Fawzan. (1998). “A Tabu Search Approach for The Flow Shop Scheduling Problem”. *European Journal of Operational Research*, 109, 88-95.
- Benli, Betül. (2011). *Çizelgeleme*. Erciyes Üniversitesi Mühendislik Fakültesi Endüstri Mühendisliği Bölümü. Yayınlanmamış Ders Notu. Kayseri.
- Bitran, R. and D. Tirupati. (1989). *Hierarchical Production Planning Working Paper*. Cambridge: Mass. MIT.
- Bolat, Bersam (2004). Web sayfası: [http://www.slidefinder.net/m/merkezlerinde atölvelerde islerin çizelgelenmesi doc/izelgesunuy/21906041](http://www.slidefinder.net/m/merkezlerinde/atölvelerde%20islerin%20cizelgelenmesi%20doc/izelgesunuy/21906041) (Erişim tarihi: 01.05.2012).
- Brucker, Peter. (2007). *Scheduling Algorithms* (Fifth Edition). Berlin: Springer.

- Ceran, Gülşad. (2006). *Esnek Akış Tipi Çizelgeleme Problemlerinin Veri Madenciliği Ve Genetik Algoritma Kullanılarak Çözülmesi*. Selçuk Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Konya.
- Chakraborty, U. K. and D. Laha. (2007). “An Improved Heuristic for Permutation Flowshop Scheduling”, *Int. J. Information and Communication Technology*, 1 (1), 89-97.
- Chakraborty, U. K., D. Laha and M. Chakraborty. (2001). “A Heuristic Genetic Algorithm for Flowshop Scheduling”. *23th Int. Conf. Information Technology Interfaces ITI 2001*, Pula, Croatia.
- Chandramouli, A. B. (2005). “Heuristic Approach for N-Job, 3-Machine Flow Shop Scheduling Problem Involving Transportation Time, Break Down Time and Weights of Jobs”. *Mathematical and Computational Applications*, 10 (2), 301-305.
- Chen, Bo, C. A. Glass, C. N. Potts and V. A. Strusevich. (1996). “A New Heuristic for Three-Machine Flow Shop Scheduling”. *Operations Research*, 44 (6), 891-898.
- Cihanlı, Özlem. (2010). *Hibrit Akış Tipi Atölyede Çizelgeleme*. Uludağ Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Bursa.
- Çatal, Elif. (2009). *Karınca Kolonileri Algoritması İle İş Akış Problemlerinin Optimizasyonu*. Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik Ve Bilgisayar Eğitimi Anabilim Dalı. Yüksek Lisans Tezi. Kocaeli.
- Demirtaş, Hatice Hande. (2011). *M-Makineli Esnek Operasyonlu Akış Tipi Sistemlerde Çizelgeleme*. TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Ankara.
- Düğenci, Muharrem. (2007). *Arı Algoritmasının Yapay Sinir Ağı Öğrenmesi İçin Kullanımı Ve Atıksu Arıtma Tesis Kontrolü Uygulaması*. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Doktora Tezi. Sakarya.

- Ekinci, Volkan. (2012). Web sayfası: <http://www.scribd.com/doc/2527866/Uretim-Planlama-ve-Cizelgeleme> (Erişim tarihi: 25.03.2012).
- Engin, Orhan. (2001). *Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma İle Çözüm Performansının Artırılmasında Parametre Optimizasyonu*. İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü. Doktora Tezi. İstanbul.
- Engin, Orhan ve Alpaslan Fırlalı. (2002). “Akış Tipi Çizelgeleme Problemlerinin Genetik Algoritma Yardımı İle Çözümünde Uygun Çaprazlama Operatörünün Belirlenmesi”. *Doğuş Üniversitesi Dergisi*, 6, 27-35.
- Engin, Orhan ve Alpaslan Fırlalı. (2002)(1). “Genetik Algoritmalarla Akış Tipi Çizelgelemede Üreme Yöntemi Optimizasyonu”. *İTÜ Mühendislik Dergisi*, 1 (1).
- Engin, Orhan ve Alper Döyen. (2007). “Akış Tipi Çizelgeleme Problemlerinin Yapay Bağışıklık Sistemleri İle Çözümünde Yeni Bir Yaklaşım”. *Doğuş Üniversitesi Dergisi*, 8 (1), 12-27.
- Erdem, Ahmet. (2008). *Bakım Kısıtı Altında Genetik Algoritmalarla Üretim Çizelgeleme*. Sakarya Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Sakarya.
- Eren, Tamer ve Ertan Güner. (2004). “Öğrenme Etkili Akış Tipi Çizelgelemede Ortalama Akış Zamanının Enküçüklenmesi”. *Gazi Üniv. Müh. Mim. Fak. Der.*, 19 (2), 119-124.
- Eren, Tamer ve Ertan Güner. (2005). “Öğrenme Etkili Akış Tipi Çizelgeleme Probleminde Maksimum Tamamlanma Zamanı Minimizasyonu”. *Teknoloji*, 8 (4), 341-348.
- Eren, Tamer ve Ertan Güner. (2005). “İki Ölçütlü Beklemesiz Akış Tipi Çizelgeleme Problemi: Toplam Tamamlanma Zamanı ve Maksimum Gecikme”. *V. Ulusal Üretim Araştırmaları Sempozyumu, İstanbul Ticaret Üniversitesi*, 231-236.
- Eren, Derya. (2001). *Application of Neural Networks to Heuristic Scheduling Algorithms*. Dokuz Eylül University Engineering and Science Institute. The Thesis of Master of Science. İzmir.

- Fan, J. P-O. and G. K. Winley. (2008). "A Heuristic Search Algorithm for Flow-Shop Scheduling". *Informatica*, 32, 453-464.
- Framinan, J. M., R. Leisten and C. Rajendran. (2003). "Different Initial Sequences for The Heuristic of Nawaz, Ensore and Ham to Minimize Makespan, Idletime or Flowtime in The Static Permutation Flowshop Sequencing Problem". *Int. J. Prod. Res.*, 41 (1), 121-148.
- French, Simon. (1982). *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. New York: Ellis Horwood Series.
- Gözde, H. ve M. C. Taplamacıoğlu. (2011, May). "Enerji Kalitesi İçin Yük-Frekans Kontrolünde Yapay Arı Kolonisi Algoritması Uygulaması". IV. *Enerji Verimliliği ve Kalitesi Sempozyumu*, Kocaeli.
- Gözen, Şerife. (2007). *Bulanık Esnek Akış Tipi Çok Prosesli Çizelgeleme Problemlerinin Genetik Algoritma Ve Tavlama Benzetimi İle Çözümü*. Selçuk Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Konya.
- Grabowski, J. and M. Wodecki. (2004). "A Very Fast Tabu Search Algorithm for The Permutation Flow Shop Problem with Makespan Criterion". *Computers & Operations Research*, 31, 1891-1909.
- Grosan, Crina and Ajith Abraham. (2011). *Intelligent Systems: A Modern Approach*. Berlin: Intelligent Systems Reference Library Springer.
- Güçlü, Abdullah. (2010). *Beklemesiz Akış Tipi İş Çizelgeleme Problemlerinin Karınca Kolonileri Algoritması İle Çözümü*. Selçuk Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Konya.
- Günaydın, Cengiz. (2008). *Beklemesiz Akış Tipi Çizelgeleme Problemlerinin Çözümünde Yapay Sinir Ağları Yaklaşımı*. Selçuk Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Konya.
- Herrnana, J. (2007). "The Legacy of Taylor, Gantt and Johnson: How to Improve Production Scheduling". *The Institute for Systems Research ISR Technical Report*, No.26.

- İlkay, M. S., A.İ. Özdemir ve M. Yüzükırmızı. (2009). “Yüklemeye Dayalı Üretim Çizelgeleme Ve Siparişe Göre Üretim Yapan Bir İşletmede Uygulaması”. 9. *Ulusal Üretim Araştırmaları Sempozyumu*, 15-17 Ekim 2009, Eskişehir, Türkiye.
- İşler, Meral. (2009). *Bulanık Esnek Akış Tipi Çizelgeleme Problemlerinin Paralel Doyumsuz Algoritma İle Çözümü: Bir Hazır Giyim İşletmesine Uygulanması*. Selçuk Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Konya.
- İşler, Mesut Cemil, Veli Çelik ve Bilal Toklu. (2009). “İki Makine Akış Tipi Öğrenme Etkili Çizelgelemede Ortak Teslim Tarihinden Mutlak Sapmaların En Küçüklenmesi”. *Gazi Üniv. Müh. Mim. Fak. Der.*, 24 (2), 351-357.
- Janiak, Adam, Erhan Kozan, Maciej Lichtenstein and Ceyda Oğuz. (2007). “Metaheuristic Approaches to The Hybrid Flow Shop Scheduling Problem with A Cost-Related Criterion”. *Int. J. Production Economics*, 105, 407-424.
- Jin, Feng, Jatinder N. D. Gupta, Shi-ji Song and Cheng Wu. (2008). “Makespan Distribution of Permutation Flowshop Schedules”. *J Sched*, 11, 421-432.
- Kalczynski, P. J. and J. Kamburowski. (2008). “An Improved NEH Heuristic to Minimize Makespan in Permutation Flow Shops”. *Computers & Operations Research*, 35, 3001-3008.
- Karaboğa, D. (2005). “An Idea Based on Honey Bee Swarm for Numerical Optimization”. *Technical Report TR06*, Computer Engineering Department, Erciyes University, Turkey.
- Karaboga, D. and B. Akay. (2007). “Artificial Bee Colony Algorithm on Training Artificial Neural Networks”. *Signal Processing and Communications Applications, SIU 2007, IEEE*, 1-4.
- Karaboga, D. and B. Akay. (2009). “A Comparative Study of Artificial Bee Colony Algorithm”. *Applied Mathematics and Computation*, 214, 108-132.
- Kellegöz, Talip. (2006). *Toplam Geç Bitirme Zamanının En Küçüklenmesi Performans Ölçütlü Permütasyon Akış Tipi Çizelgeleme Problemlerinin Çözümünde*

- Genetik Algoritma Yaklaşımı*. Kırıkkale Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Kırıkkale.
- Körez, Mehmet Tufan. (2005). *Sıralı Akış Tipi Çizelgeleme Problemlerinde Genetik Algoritma Uygulaması*. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Programı. Yüksek Lisans Tezi. İstanbul.
- Küçük, Birgül. (2010). *Özdeş Paralel Makineli Bir Üretim Sisteminin Karınca Koloni Algoritması İle Çizelgelenmesi*. İstanbul Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Üretim Yönetimi Bilim Dalı. Doktora Tezi. İstanbul.
- Ladhari, T. and M. Haouari. (2005). “A Computational Study of The Permutation Flow Shop Problem Based on A Tight Lower Bound”. *Computers & Operations Research*, 32, 1831-1847.
- Laha, D. and S. U. Sapkal. (2011). “An Efficient Heuristic Algorithm for m-Machine No-Wait Flow Shops”. *Proceedings of The International MultiConference of Engineers and Computer Scientists IMECS 2011*, 1, Hong Kong.
- Leung, Joseph Y-T. (2004). *Handbook of Scheduling Algorithms, Models and Performance Analysis*. United States of America: CRC Press Company.
- Li, Xiaofeng and Hai Zhao. (2009). “Greedy Algorithm Solution of Flexible Flow Shop Scheduling Problem”. *International Journal of Computer Science and Network Security*, 9 (11), 177-180.
- Liu, Yan-Feng and San-Yang Liu. (2011). “A Hybrid Discrete Artificial Bee Colony Algorithm for Permutation Flowshop Scheduling Problem”. *Applied Soft Computing*, 1-5.
- Low, Chinyao. (2005). “Simulated Annealing Heuristic for Flow Shop Scheduling Problems with Unrelated Parallel Machines”. *Computers & Operations Research*, 32, 2013-2025.
- Lukaszewicz, P. P. (2005). *Metaheuristics for Job Shop Scheduling Problem, Comparison of Effective Methods*. Aarhus School of Business. Master Thesis.
- Nearchou, A. C. (2004). “A Novel Metaheuristic Approach for The Flow Shop Scheduling Problem”. *Engineering Applications of Artificial Intelligence*, 17, 289-300.

- Onwubolu, G. and D. Davendra. (2006). "Scheduling Flow Shops Using Differential Evolution Algorithm". *European Journal of Operational Research*, 171, 674-692.
- Özdemir, Gökhan. (2006). *Kısıtlı Kaynaklarla Proje Çizelgelemesi Problemlerinde Kullanılan Genetik Algoritma Metodları Ve Bunların Karşılaştırılması*. Ankara Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı. Yüksek Lisans Tezi. Ankara.
- Öztürk, Reşide Elif. (2007). *Esnek Akış Tipi Ve Çok İşlemcili Esnek Akış Tipi Çizelgeleme Problemlerinin Paralel Doyumsuz Algoritma İle Çözümü*. Selçuk Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Konya.
- Pan, Q-K., L. Wang and B. Qian. (2009). "A Novel Differential Evolution Algorithm for Bi-Criteria No-Wait Flowshop Scheduling Problems". *Computers & Operations Research*, 36, 2498-2511.
- Pan, Q-K. and L. Wang. (2012). "Effective Heuristics for The Blocking Flowshop Scheduling Problem With Makespan Minimization". *Omega*, 40, 218-229.
- Pan, Q-K., M. F. Tasgetiren and Y-C. Liang. (2008). "A Discrete Differential Evolution Algorithm for The Permutation Flowshop Scheduling Problem". *Computers & Industrial Engineering*, 55, 795-816.
- Pan, Q-K., M. F. Tasgetiren, P. N. Suganthan and T. J. Chua. (2011). "A Discrete Artificial Bee Colony Algorithm for The Lot-Streaming Flow Shop Scheduling Problem". *Information Sciences*, 181, 2455-2468.
- Pandian, P. and P. Rajendran. (2010). "Solving Constrained Flow-Shop Scheduling Problems with Three Machines". *Int. J. Contemp. Math. Sciences*, 5 (19), 921-929.
- Panigrahi, B.K., Y. Shi and M-H. Lim. (2011). *Handbook of Swarm Intelligence, Concepts, Principles and Applications*. Berlin: Springer.
- Pinedo, Michael L. (2008). *Scheduling: Theory, Algorithms and Systems* (Third Edition). New York: Springer Science+Business Media, LLC.

- Rahimi-Vahed, A. R. and S. M. Mirghorbani. (2007). "A Multi-Objective Particle Swarm for A Flow Shop Scheduling Problem". *J Comb Optim.*, 13, 79-102.
- Rajendran, C. (1994). "A No-Wait Flowshop Scheduling Heuristic to Minimize Makespan". *J. Opl Res. Soc.*, 45 (4), 472-478.
- Rajendran, C. and H. Ziegler. (2003). "Scheduling to Minimize The Sum of Weighted Flowtime and Weighted Tardiness of Jobs in A Flowshop With Sequence-Dependent Setup Times". *European Journal of Operational Research*, 149, 513-522.
- Rajendran, C. and H. Ziegler. (2004). "Ant Colony Algorithms for Permutation Flowshop Scheduling to Minimize Makespan/Total Flowtime of Jobs". *European Journal of Operational Research*, 155, 426-438.
- Rao, R. Venketa. (2011). *Advanced Modeling and Optimization of Manufacturing Processes*. London: Springer.
- Roadammer, F. A. and K. P. White. (1988). "A Recent Survey of Production Scheduling". *IEEE Transactions on Systems, Man, And Cybernetics*, 18 (6), 841-851.
- Ruiz, R., C. Maroto and J. Alcaraz. (2006). "Two New Robust Genetic Algorithms for The Flowshop Scheduling Problem". *Omega*, 34, 461-476.
- Ruiz, R. and A. Allahverdi. (2007). "No-Wait Flowshop with Separate Setup Times to Minimize Maximum Tardiness". *Int J Adv Manuf Technol*, 35, 551-565.
- Sağır, Müjgan ve Tuğba Saraç. (2007). "Hazırlık Zamanı Kaynaklı Üretim Kayıplarını Enazlayan Çok Amaçlı Bir Çizelgeleme Algoritması". *Endüstri Mühendisliği Dergisi*, 18 (4), 14-25.
- Sarikaya, Banu. (2005). *Otomotiv Yan Sanayinde Bir Üretim Çizelgeleme Probleminin İncelenmesi*. Çukurova Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. Adana.
- Seçme, Gökhan. (2006). *Akış Tipi Çizelgeleme Problemlerinin Yapay Sinir Ağları İle Modellenmesi*. Erciyes Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı. Yüksek Lisans Tezi. Kayseri.

- Seda, Milos. (2007). "Mathematical Models of Flow Shop and Job Shop Scheduling Problems". *World Academy of Science, Engineering and Technology*, 31, 122-127.
- Stützle, T. (1998). "An Ant Approach to The Flow Shop Problem". *Proceedings of The Sixth European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, Verlag Mainz, Aachen, Germany, 1560-1564.
- Sumathi, S. and Surekha Paneerselvam. (2010). *Computational Intelligence Paradigms: Theory and Applications Using Matlab*. New York: CRC Press Taylor & Francis Group.
- Tasgetiren, M. F., Y-C. Liang, M. Sevkli and G. Gencyilmaz. (2007). "A Particle Swarm Optimization Algorithm for Makespan and Total Flowtime Minimization in The Permutation Flowshop Sequencing Problem". *European Journal of Operational Research*, 177, 1930-1947.
- Temiz, İzzettin ve Serpil Erol. (2007). "Bulanık Akış Tipi Çizelgeleme Problemi İçin Çok Amaçlı Genetik Algoritma". *Gazi Üniv. Müh. Mim. Fak. Der.*, 22 (4), 855-862.
- Temiz, İzzettin. (2008). "Çok Kriterli Permütasyon Akış Tipi Çizelgeleme Problemi İçin Bir Tavlama Benzetimi Yaklaşımı". *1. Mühendislik ve Teknoloji Sempozyumu*, 24-25 Nisan 2008, Çankaya Üniversitesi, Ankara, 300-308.
- Todd, D. and P. Sen. (1997). "Multiple Criteria Scheduling Using Genetic Algorithms in a Shipyard Environment". *Proc. Ninth Int. Conf. Computer Applications in Shipbuilding*, 234-265.
- Tokmak, M. (2011). *Yapay Arı Kolonisi Algoritması İle Ders Çizelgeleme Probleminin Çözümü*. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Elektronik Bilgisayar Eğitimi Ana Bilim Dalı. Yayımlanmamış Yüksek Lisans Tezi. Isparta.
- Turan, Utkan. (2011). *Üretim Planlama ve Çizelgeleme Çalışması*. Mersin Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı. Üretim Yönetimi Dersi Yayımlanmamış Ödevi. Mersin.

- Türkeli, Bikem. (2010). *Solution of Bicriteria Open Shop Scheduling Problem with Genetic Algorithm and Tabu Search and Detailed Analysis of Results*. Marmara University Institute for Graduate Studies in Pure and Applied Sciences. Thesis for The Degree of Master of Science in Industrial Engineering. Istanbul.
- Uçar, Hatice. (2005). *Comparison of Genetic Algorithm and Particle Swarm Optimization Algorithm for Permutation Flow Shop Sequencing Problem with Criterion of Number of Tardy Jobs*. The Graduate Institute of Sciences and Engineering of Fatih University. Master of Science in Industrial Engineering. Istanbul.
- Uruk, Zeynep. (2011). *Two-Machine Flowshop Scheduling with Flexible Operations and Controllable Processing Times*. The Department of Industrial Engineering and The Graduate School of Engineering and Science of Bilkent University. The Degree of Master of Science. Ankara.
- Uysal, Özgür. (2006). *Comparison of Genetic Algorithm and Particle Swarm Optimization Algorithm for Bicriteria Permutation Flowshop Scheduling Problem*. Marmara University Institute for Graduate Studies in Pure and Applied Sciences. Dissertation for The Degree of Doctor of Philosophy in Industrial Engineering Programme. Istanbul.
- Vallada, Eva and Ruben Ruiz. (2009). “Cooperative Metaheuristics for The Permutation Flow Shop Scheduling Problem”. *European Journal of Operational Research*, 193, 365-376.
- Widmer, M., A. Hertz and D. Costa. (2008). “[Metaheuristics and Scheduling](#)”. Chapter 3 of the book *Production Scheduling* (Eds P. Lopez and F. Roubellat), Wiley, 33-68.
- Yağmahan, Betül ve M. Mutlu Yenisey. (2006). “Akış Tipi Çizelgeleme Problemleri İçin KKE Parametre Eniyileme”. *İTÜ Mühendislik Dergisi*, 5 (2), Kısım. 2, 133-141.
- Yang, Xin-She. (2010). *Engineering Optimization, An Introduction with Metaheuristic Applications*. New Jersey: John Wiley and Sons Inc.

- Yang, Xin-She and S. Koziel. (2011). *Computational Optimization and Applications in Engineering and Industry*. Berlin: Springer.
- Yıldırım, U. Mahir. (2008). *An Ant Colony Algorithm for The Time-Independent and Time-Dependent Vehicle Routing Problem with Time Windows*. Sabanci University the Graduate School of Engineering and Natural Sciences. Thesis of Master Science. Istanbul.
- Yılmaz, Şule. (2008). *Çok Depolu Araç Rotalama Probleminin Karınca Kolonisi Optimizasyonu İle Modellenmesi Ve Bir Çözüm Önerisi*. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı. Yüksek Lisans Tezi. İstanbul.
- Ying, K-C. and C-J. Liao. (2004). “An Ant Colony System for Permutation Flow-Shop Sequencing”. *Computers & Operations Research*, 31, 791-801.
- Zobolas, G. I., C. D. Tarantilis and G. Ioannou. (2009). “Minimizing Makespan in Permutation Flow Shop Scheduling Problems Using A Hybrid Metaheuristic Algorithm”. *Computers & Operations Research*, 36, 1249-1267.

EKLER

EK 1: İŞLERLE İLGİLİ VERİLER

- İşlerin Türleri

İş No	İşin Türü
1	Walthe Yatak 80x200
2	Walthe Yatak Kahve 80x200
3	Walthe Yatak 80x210
4	Emmen Yatak 80x200
5	Emmen Yatak 90x200
6	Coolmax Yatak 90x200
7	Emmen Yatak 90x200
8	Emmen Yatak 80x200
9	Domino Yatak 160x200
10	Walthe Yatak 80x200
11	Walthe Yatak 90x210
12	Walthe Yatak Kahve 90x220
13	Walthe Yatak 80x200
14	Azalee 90x200
15	Prestige 14 cm 90x200
16	Prestige 20 cm 90x200
17	Prestige 20 cm 140x200
18	Reina 160x200
19	Venüs Comfort 160x200
20	C. Comfort 90x190
21	20 cm Roll Yatak 137x190
22	20 cm Roll Yatak 152x198
23	Pedli Yaylı Yatak 150x200

- İş Grupları İçin Hazırlık Zamanları Matrisi

Gruplar	A	B	C	D	E
A		40	40	40	40
B	40		40	40	40
C	40	40		40	40
D	40	40	40		40
E	40	40	40	40	

- **İşlerin Teslim Süreleri**

İş No	Teslim Süresi (dk)
1	12480
2	12480
3	12480
4	12480
5	12480
6	13440
7	13440
8	13440
9	13440
10	13440
11	13440
12	13440
13	14400
14	13440
15	13440
16	13440
17	13440
18	13440
19	13440
20	12480
21	14400
22	14400
23	10560

- **İşlerin Hangi İş Grubunda Oldukları**

Grup	Gruptaki İşlerin Numaraları
A	1-2-3-10-11-12-13
B	4-5-7-8
C	6-20-21-22
D	9
E	14-15-16-17-18-19-23

• İşlerin İşlem Süreleri

İş No	İşin Grubu	Makine 1	Makine 2	Makine 3	Makine 4	Makine 5	Toplam (dk)
1	A	281	325	130	182	225	1143
2	A	22	25	10	14	17	88
3	A	22	25	10	14	17	88
4	B	108	125	50	70	87	440
5	B	324	375	150	210	260	1319
6	C	108	125	50	70	87	440
7	B	324	375	150	210	260	1319
8	B	216	250	100	140	173	879
9	D	54	63	25	35	43	220
10	A	22	25	10	14	17	88
11	A	43	50	20	28	35	176
12	A	22	25	10	14	17	88
13	A	216	250	100	140	173	879
14	E	140	162	65	91	112	570
15	E	999	999	560	784	969	4311
16	E	302	350	140	196	242	1230
17	E	486	563	225	315	389	1978
18	E	183	213	85	119	147	747
19	E	75	87	35	49	60	306
20	C	999	999	560	784	969	4311
21	C	999	999	999	999	999	4995
22	C	999	999	500	700	866	4064
23	E	999	999	999	999	999	4995

ÖZ GEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı : Murat RUHLUSARAÇ
 Uyruğu : Türkiye (TC)
 Doğum Tarihi ve Yeri : 3 Mart 1987, Kayseri
 Medeni Durumu : Bekâr
 Telefon : +90 352 207 66 66 / 30259
 Faks : +90 352 437 52 39
 E-mail Adresi : mrhulusarac@erciyes.edu.tr
 Yazışma Adresi : Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi
 İşletme Bölümü D-318 38039 Talas / KAYSERİ

EĞİTİM

Derece	Eğitim Kurumu	Mezuniyet Tarihi
Yüksek Lisans	EÜ Sosyal Bilimler Enstitüsü	2012
Lisans	EÜ İİBF İşletme	2009
Lise	Özel Hisarcıklıoğlu Fen Lisesi	2005

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2010 - Halen	EÜ İİBF İşletme	Arş. Gör.

YABANCI DİL

İngilizce