



**T.C.
HALIÇ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**ATAMA PROBLEMİ İÇİN SÜRÜ ZEKÂSI TABANLI EN
İYİLEŞTİRME YÖNTEMLERİNİN UYARLANMASI VE
BAŞARIM DEĞERLENDİRMESİ**

YÜKSEK LİSANS TEZİ

**Hazırlayan
Erdi KUDU**

**Danışman
Dr. Öğr. Üyesi Faruk BULUT**

İstanbul - 2020

**T.C.
HALIÇ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**ATAMA PROBLEMİ İÇİN SÜRÜ ZEKÂSI TABANLI EN
İYİLEŞTİRME YÖNTEMLERİNİN UYARLANMASI VE
BAŞARIM DEĞERLENDİRMESİ**

YÜKSEK LİSANS TEZİ

**Hazırlayan
Erdi KUDU**

**Danışman
Dr. Öğr. Üyesi Faruk BULUT**

İstanbul - 2020

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalı Tezli Yüksek Lisans Programı Öğrencisi Erdi KUDU tarafından hazırlanan “*Atama Problemi İçin Sürü Zekâsı Tabanlı En İyileştirme Yöntemlerinin Uyarlanması ve Başarım Değerlendirmesi*” konulu çalışması jürimizce Yüksek Lisans Tezi olarak kabul edilmiştir.

Tez Savunma Tarihi: 11.02.2020

(Jüri Üyesinin Ünvanı, Adı, Soyadı ve Kurumu):

İmzası

Jüri Üyesi : Dr. Öğr. Faruk BULUT
: İstanbul Rumeli Üniversitesi (Danışman)

Jüri Üyesi : Prof. Dr. Mübariz EMİNLİ
: Haliç Üniversitesi

Jüri Üyesi : Dr. Öğr. Üyesi Hakan KOYUNCU
: İstanbul Gelişim Üniversitesi

Bu tez Enstitü Yönetim Kurulunca belirlenen yukarıdaki jüri üyeleri tarafından uygun görülmüş ve Enstitü Yönetim Kurulunun kararıyla kabul edilmiştir.


Prof. Dr. M. Burcu IRMAK YAZICIOĞLU
Vekil Müdür

Erdi_Kudu_Tez (1).docx

ORIGINALITY REPORT

14%

SIMILARITY INDEX

PRIMARY SOURCES

1	id.scribd.com Internet	1354 words — 7%
2	link.springer.com Internet	376 words — 2%
3	www.p005.net Internet	116 words — 1%
4	Anuar, Nurul Izah, and Adi Saptari. "Performance evaluation of different types of particle representation procedures of Particle Swarm Optimization in Job-shop Scheduling Problems", IOP Conference Series Materials Science and Engineering, 2016. Crossref	103 words — 1%
5	www.coursehero.com Internet	82 words — < 1%
6	www.kecl.ntt.co.jp Internet	62 words — < 1%
7	suanpalm3.kmutnb.ac.th Internet	55 words — < 1%
8	docplayer.biz.tr Internet	54 words — < 1%
9	mmf.rumeli.edu.tr Internet	52 words — < 1%
10	www.it.bond.edu.au Internet	



13/03/2020

TEZ ETİK BEYANI

Yüksek Lisans Tezi olarak sunduğum “Atama Problemi İçin Sürü Zekası Tabanlı Eniyileştirme Yöntemlerinin Uyarlanması ve Başarım Değerlendirmesi” başlıklı bu çalışmayı baştan sona kadar danışmanım Dr. Öğr. Üyesi Faruk BULUT’un sorumluluğunda tamamladığımı, verileri kendim topladığımı, analizleri ilgili laboratuvarlarda yaptığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

(İmza)

Erdi KUDU

ÖNSÖZ

Geçmişten günümüze en önemli problemlerden biri olan görev atama problemi için çeşitli çalışmalar yapılmıştır. Bir iş/projenin yapım aşamasındaki en büyük maliyet kalemlerinden biri olan kaynak maliyeti her zaman optimum seviyede kullanılmaya çalışılmıştır.

Kaynakların kullanılmasında yapılan yanlış planlama sebebiyle bazı projelerin başarısız olduğu görülmüş ve hatta bu sorun, bazı sektörlerde bu maliyeti kontrol altına alamayan işletmelerin iflas etmelerine sebep olmuştur. Önceleri kâğıt üzerinde yapılan bu kaynak planlamaları teknolojinin gelişmesi ile dijitalleştirilmiş, bu dijital dönüşüm sürecinde problemin çözümüne çeşitli yönlerden yaklaşımlar olmuştur.

Bu çalışma boyunca dersinde anlattığı konudan yola çıkmamda bana fikir sunan, gösterdiği her türlü destek ve yardımlarından dolayı çok değerli hocam Sayın Dr. Öğr. Üyesi Faruk BULUT 'a en içten dileklerle teşekkür ederim. Bütün tez sürecinde bana gösterdiği anlayış ve destekten dolayı eşim Ece YÖNET KUDU'ya da teşekkür ederim.

Ayrıca Haliç Üniversite'sindeki eğitim hayatım boyunca derslerdeki anlayış ve sabırlarından dolayı bütün hocalarıma da teşekkür ederim.

Şubat, 2020

Erdi KUDU

İÇİNDEKİLER

Sayfa No.

İÇİNDEKİLER	I
KISALTMALAR	III
ŞEKİLLER...	V
ÇİZELGELER	VI
ÖZET	VII
SUMMARY	VIII
1. GİRİŞ	1
1.1. Problemin Tanımlanması	3
2. LİTERATÜR ARAŞTIRMASI	5
2.1. Sürü Zekâsının Sınıflandırması	5
2.1.1. Doğal / Bilimsel: Karıncaların Yiyecek Arama Davranışı.....	6
2.1.2. Yapay / Bilimsel: Bir Robot Sürüsü Tarafından Kümelenme	6
2.1.3. Doğal / Mühendislik: Hayvan Topluluklarının Kolektif Davranışlarının Kullanılması.....	7
2.1.4. Yapay / Mühendislik: Sürü Tabanlı Veri Analizi	7
2.2. Sürü Zekâsı Sisteminin Özellikleri	7
2.3. İş Atama Problemi.....	9
2.4. Bal Arısı Koloni Algoritması ile İş Atama Problemi.....	10
2.4.1. Arıların Doğadaki Davranışlarının Tanımlanması.....	15
2.4.2. Yapay Arı Sistemlerinin Avantajları ve Dezavantajları.....	17
2.4.3. Yapay Arı Sistemlerinde Yapılan Çalışmaların Gözden Geçirilmesi ve Sınıflandırılması	18
2.5. Parçacık Sürü Algoritması ile İş Atama Problemi	38
2.5.1. PSO Varyantlarının Dağılım Metotları	41
2.5.2. Farklı Komşuluk Topolojilerini Kullanan PSO Varyantları	41
2.5.3. Parçacık Temsil Prosedürleri	43
2.5.4. PSO Avantaj ve Dezavantajları.....	44

2.5.5. Parçacık Sürü Optimizasyonunda Yapılan Çalışmaların Gözden Geçirilmesi ve Sınıflandırılması	44
3. GEREÇ VE YÖNTEM	49
3.1. Gantt Diyagramı	49
3.2. İstatistiksel Hipotez Testi	50
3.3. İş Atama Problemi İçin Yapay Arı Kolonisi Optimizasyonu	51
3.4. İş Atama Problemi İçin Parçacık Sürü Optimizasyonu.....	53
4. BULGULAR	56
5. TARTIŞMA	70
6. SONUÇ.....	72
7. KAYNAKLAR.....	73
8. ÖZGEÇMİŞ	83

KISALTMALAR

ABC	: Artificial Bee Colony
ACO	: Ant Colony Optimization
AMPSO	: Adaptive Mutation Particle Swarm Optimization
APSO	: Asynchronous Partical Swarm Optimization
BCO	: Bee Colony Optimization
BSO	: Bee Swarm Optimization
CSP	: Constriction Satisfaction Problem
EPD	: Economic Power Distribution
EPSO	: Ecological Partical Swarm Optimization
FBS	: Fuzzy Bee System
FDRPSO	: Fitness Distance Ratio Particle Swarm Optimization
FKBC	: Fuzzy Knowledge Based Controller
GA	: Genetic Algorithms
GCPSO	: Guaranteed Convergence Particle Swarm Optimization
GRASP	: Greedy Randomized Adaptive Search
GSAT	: Greedy Satisfiability
GSO	: Glowworm Swarm Optimization
GWSAT	: Global Best and Worst Satisfiability
HBMO	: Honey Bees Mating Optimization
HBPI	: Honey Bee Politic Iteration
HSAT	: Heterogeneous Simulated Annealing Team
IWCFA	: Incoming Weight Contraction Factor Approaching
LDWPSO	: Linearly decreasing weight PSO
LS	: Link State
LVQ	: Learning Vector Quantification
MAX-RWA	: Maximum Routing and Wavelength Assignment
MAX-W-SAT	: Maximum Weighted Satisfiability
MBO	: Marriage in Honey- Bees Optimization
MDP	: Markov Decision Process

MLP	: Multi Layer Perceptron
NLP	: Natural Language Processing
NP	: Non Deterministic Polinomial
OHGBPPSO	: One Half Global Best Position PSO
PBPPSO	: Personel Best Position PSO
PSO	: Particle Swarm Optimization
SB	: Shifting Bottleneck
SB	: Simulated Annealing
SDP	: Stochastic Decision Process
SPSO	: Standard PSO
SSAT	: Simple Satisfiability
SVRP	: Stochastic Vehicle Routing Problem
TS	: Tabu Search
TSP	: Travelling Salesman Problem
VBA	: Virtual Bee Algorithm
VRP	: Vehicle Routing Problem

ŞEKİLLER

Sayfa No.

Şekil 1.1. Klasik ve modern optimizasyon stratejilerinin resimsel olarak karşılaştırılması (Baykasoğlu, 2001).....	3
Şekil 2.1. 3×3'lük problem için çözümün gantt şema gösterimi (Shailendra ve dig., 2014)	10
Şekil 2.2. Bal arılarının titreme dansı	11
Şekil 2.3. Titreme dansı örnek gösterim	13
Şekil 2.4. BCO akış şeması.....	14
Şekil 2.5. Bal arısı toplayıcısının tipik davranışı	17
Şekil 2.6. PSO akış şeması.....	40
Şekil 2.7. Komşuluk yapısı türleri	42
Şekil 3.1. Örnek gantt şeması	50
Şekil 4.1. Algoritma averaj performans karşılaştırması-1	59
Şekil 4.2. Algoritma averaj performans karşılaştırması-2	60
Şekil 4.3. Ortalama makespan performansı	61
Şekil 4.4. Çalışma zamanı performansı	62
Şekil 4.5. İşlemci averaj performansı	63
Şekil 4.6. Genel Performans Haritası.....	64
Şekil 4.7. 50×10'luk veri seti örnek gantt şema çıktısı.....	67

ÇİZELGELER

	Sayfa No.
Çizelge 1.1. FT06 atama problemi.....	4
Çizelge 2.1. 3×3'lük bir iş atama sorunu	9
Çizelge 2.2. Literatür ve uygulamaların sınıflandırılması	19
Çizelge 2.3. (devam)Literatür ve uygulamaların uınıflandırılması	20
Çizelge 2.4. (devam)Literatür ve uygulamaların sınıflandırılması	21
Çizelge 3.1. PSO Algoritmasının kaba kodu	55
Çizelge 4.1. BCO ve PSO algoritmalarının parametreler çizelgesi	56
Çizelge 4.2. BCO ve PSO çeşitli veri seti performans değerlendirme sonuçları.....	57
Çizelge 4.3. 50×10 veri seti (swv12) performans testi (I=100).....	58
Çizelge 4.4. İstatistiksel hipotez testi karşılaştırması	68
Çizelge 4.5. İstatistiksel hipotez testi sonuçları	69

ÖZET

ATAMA PROBLEMİ İÇİN SÜRÜ ZEKÂSİ TABANLI EN İYİLEŞTİRME YÖNTEMLERİNİN UYARLANMASI VE BAŞARIM DEĞERLENDİRMESİ

Görev atama, hesaplama sistemlerinin yeteneklerinden etkin bir şekilde yararlanmanın temel adımlarından biridir. Ayrıca etkin bir görev atama, başarılı bir görev yönetimini sağlar. Böylece en büyük maliyet kayıplarından olan kaynak maliyeti azaltılır. Bu çalışmada, bu sorunun çözümünde kullanılan meta sezgisel algoritmalarından sürü zekâsı algoritmaları kullanılmıştır. Sürü zekâsı algoritmalarından arı ve parçacık kolonisi optimizasyon algoritmalarının görev atama sorunu üzerindeki çözümlerinin performansı karşılaştırılmıştır. Çeşitli boyutta ve zorlukta veri setleri seçilmiş ve arı ve parçacık kolonisi ile optimize edilmiştir. Sonuç olarak, hangi optimizasyon yönteminin veri setine göre daha iyi çalıştığı yorumlanmıştır.

Anahtar Kelimeler: Sürü optimizasyon algoritmaları, atama problemi, arı kolonisi, parçacık kolonisi

SUMMARY

IMPLEMENTATION AND PERFORMANCE EVALUATION OF SWARM INTELLIGENCE BASED OPTIMIZATION METHODS FOR ASSIGNMENT PROBLEM

The task assignment is one of the core steps to effectively exploit the capabilities of computing system. In addition, an effective task assignment provides successful task management. So that, this reduces the cost of resources from the biggest cost losses. In this study, swarm intelligence algorithms which are used in the solution of this problem are used. It has been compared the performance of the solution of bee and particle colony optimization algorithms on assignment problem. Data sets of various size and difficulties are selected and optimized with bee and particle colony. As a result, it is interpreted which optimization method works better according to data set.

Keywords: Swarm optimization algorithm, assignment problem, bee colony, particle colony

1. GİRİŞ

Bu bölümde, tezde araştırılan konu ile ilgili genel arařtırmalar yapılmıřtır. Problemin ne olduđu ve bu problemin ne olduđunu aıklamak iin mantıksal bir rnek sunulmuřtur. Bu problemin özümünde göz önünde bulundurulacak varsayımlar bildirilmiřtir.

Gemiřten günümüze kaynak atama problemi, bir diđer adıyla izelgeleme problemi birok sektörde soruna sebep olmuřtur. Kaynakların verimsiz kullanılması, süre ve maliyet bazında kurumlara en ok külfet getiren kalemdir.

Kaynak olarak sadece insan kaynađı düşünülmemelidir. Kaynak olarak adlandırılan maliyet bazen insan kaynađı olurken bazen de bir materyal veya hammadde kaynađı olabilir. Materyal türü kaynakların maliyetleri genellikle insan veya hammadde gibi sürekli ihtiya duyulan ve artan yönde ilerleyen maliyetler deđildir. Ancak bir karřılařtırma yapmak gerekirse insan kaynađı yapılan iř boyunca sürekli bir ihtiya olduđundan elzem bir konu halini almıřtır.

Bu sebeple kaynak maliyetlerini azaltmak üzere birok arařtırma yapılmıř ve eřitli özüm yöntemleri geliřtirilmeye alıřılmıřtır. özüm bulunmaya alıřılsa da bu konuda tam olarak bir bařarı sađlanamamıřtır. Ancak incelemelerde yaklařık özümlerin bulunabildiđi, yani özümeye yaklařılabildiđi gözlemlenmiřtir. Bu da sorunun tam özülemez de optimum özümlere ulařılmasını sađlamıřtır. Bu tür problemlerin yüksek hesaplama karmařıklıđı olması sebebiyle deterministik yöntemlerle ile özümü her zaman sađlanamamıřtır. Mutlak sonuca ulařmaya alıřan bu yöntemlerde sonuca ulařmak karmařıklıđa göre bazen günleri, bazen haftaları ve bazen de yılları aldıđı gözlemlenmiřtir. Bir bařka dezavantajı da özüm bulunabilse bile hesaplamalardaki karmařıklık sebebiyle kaynak yetersizliđi hesaplama sonucunun bulunmasını engellemiřtir.

Bu tür problemler Deterministik Olmayan Polinom (NP)-zor kategorisindedir. Yani polinomsal zamanda bir özümü olduđunu ispatlayamadıđımız karar problemlerinin karmařıklık sınıfıdır. Bu sebeple bu tür problemler kombinatoryel

problemler kategorisine girmektedir ve stokastik yöntemlerle çözülmesi daha kolaydır. Meta sezgisel optimizasyon algoritmaları da stokastik çözüm yöntemlerinden biridir. Meta sezgisel optimizasyon yöntemlerinin optimum çözüme ulaşmada daha verimli olduğu çeşitli çalışmalar sonucunda ispatlanmıştır. Optimum sonuç denmektedir. Çünkü meta sezgisel optimizasyon yöntemler mutlak çözüme ulaşmaya değil, çeşitli kısıtlara göre alınabilecek sonuca ulaşmayı hedeflemiştirlerdir. Bu kısıtlar, bazen belli bir süre çalışma sonucunda elde edilen en iyi sonucun alınması ile olurken bazen de belli bir döngü sonucunda ulaşılan en iyi sonucun alınması ile sonuçlanır.

Klasik optimizasyon algoritmalarının karakteristikleri, çözüm algoritmasını verilen bir probleme uyarlama esnekliğidir. Genellikle belirli bir problem simpleks algoritma gibi klasik bir algoritmanın bu sorunu çözebileceği şekilde modellenir.

Bu sınırlamaların üstesinden gelmek için daha esnek ve uyarlanabilir genel amaçlı algoritmalara ihtiyaç vardır. Bu algoritmaları belirli bir problemi olabildiğince yakın gerçeğe modellemek için uyarlamak kolay olmalıdır. Bu motivasyona dayanarak, literatürde genetik algoritmalar, simüle tavlama ve tabu arama gibi doğadan ilham alan birçok algoritma geliştirilmiştir. Bu algoritmaların klasik yöntemlere kıyasla çok daha iyi çözümler sağlayabileceği de gösterilmiştir.

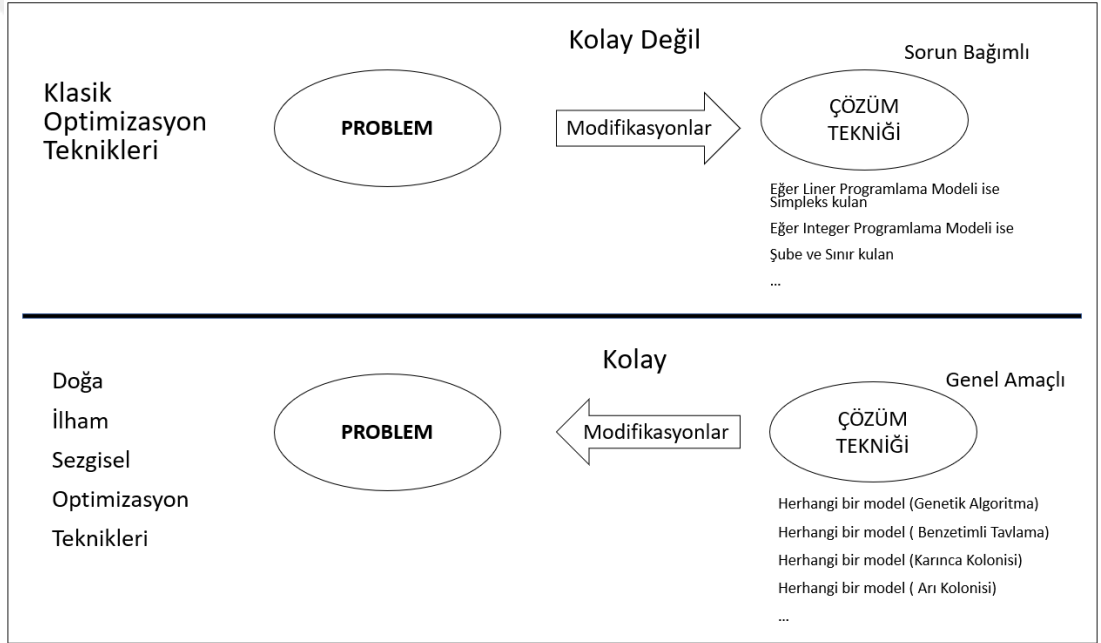
Sürü zekâsı olarak bilinen doğadan ilham alan bir algoritma, böceğin problem çözümü yeteneklerini taklit edebilen bazı meta-sezgisel yöntemler geliştirmek için böcek davranışına odaklanır. Karınca kolonisi optimizasyonu, parçacık sürüsü optimizasyonu, Yapay Arı Kolonisi (ABC) vb. problem modelleme ve çözümünde böcek davranışını taklit eden iyi bilinen algoritmalarından bazılarıdır.

ABC, gerçek bal arılarının gıda aramadaki doğal davranışlarını modellemeye çalışır. Bal arıları, yiyecek kaynaklarını en uygun şekilde bulmak ve yenilerini aramak için sallantı dansı gibi çeşitli mekanizmalar kullanır. Bu onları yeni akıllı arama algoritmaları geliştirme için iyi bir aday yapar.

Klasik optimizasyon tekniklerinin matematiksel programlama ve yöneylem araştırması modellerinin çözümünde çeşitli sınırlamalar getirdiği iyi bilinen bir gerçektir. Klasik optimizasyon algoritmalarının çözüm stratejileri genellikle nesnel ve kısıtlama fonksiyonlarının türüne (doğrusal, doğrusal olmayan vb.) ve problem modellemesinde kullanılan değişkenlerin türüne (tamsayı, gerçek vb.) bağlıdır.

Verimlilikleri ayrıca çözüm alanının büyüklüğüne, problem modellemesinde kullanılan bir dizi değişken, kısıtlama ve yapıya bağlıdır.

Belirli bir çözüm prosedürüne uyan gerçek yaşam problemini formüle etmek genellikle kolay değildir. Bunu başarmak için orijinal problem parametreleri (yuvarlama değişkenleri, yumuşama kısıtlamaları vb.) üzerinde bazı değişiklikler ve / veya varsayımlar yapmak gerekir. Bu kesinlikle çözüm kalitesini etkiler. Klasik optimizasyon prosedürlerinin dezavantajlarının üstesinden gelmek için araştırmacılar tarafından yeni bir dizi problem ve modelden bağımsız doğadan ilham alan sezgisel optimizasyon algoritmaları önerildi. Bu teknikler verimli ve esnektir. Bunlar spesifik problem gereksinimlerine uyacak şekilde değiştirilebilir veya uyarlanabilir. Aşağıdaki şekilde bu tekniklerinin prosedürleri sunulmuştur (Şekil 1.1).



Şekil 1.1. Klasik ve modern optimizasyon stratejilerinin resimsel olarak karşılaştırılması (Baykasoğlu, 2001)

1.1. Problemin Tanımlanması

Sürü algoritmalarının performanslarını test etmek için Brunel Üniversitesi Londra web sitesinden 82 adet çeşitli boyutlardaki veri setleri kullanılmıştır (<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/jobshop1.txt>, Erişim Tarihi: 15 Eylül 2019). Bu veri setlerinden her boyut için ikişer adet seçilecektir. Aşağıda bu veri setlerinden 6 makineden ve her makinede 6 işlemden oluşan 6 işin örnekleme

yapılmıştır. FT06 adı verilen sorunun ayrıntıları Çizelge 1.1’de açıklanmıştır. çizelgede, her satır parantez içinde işlem süresi olan bir iş dizisidir (Çizelge 1.1).

Çizelge 1.1. FT06 atama problemi

İş No	Makinelerin sırası (İşlem süresi)					
1	2 (1)	0 (3)	1 (6)	3 (7)	5 (3)	4 (6)
2	1 (8)	2 (5)	4 (10)	5 (10)	0 (10)	3 (4)
3	2 (5)	3 (4)	5 (8)	0 (9)	1 (1)	4 (7)
4	1 (5)	0 (5)	2 (5)	3 (3)	4 (8)	5 (9)
5	2 (9)	1 (3)	4 (5)	5 (4)	0 (3)	3 (1)
6	1 (3)	3 (3)	5 (9)	0 (10)	4 (4)	2 (1)

Çizelge 1.1.’den görülebileceği gibi, 6 makine kullanılarak planlanması gereken 6 iş vardır. 6 satır (parantez içinde işlem süresi olan 6 işi temsil eder) ve 6 sütun (6 makineye karşılık gelir) vardır. Her satır, bir işin ziyaret etmesi gereken makine sırasını temsil eden sayıların permütasyonudur. Örneğin, İş 1 başlangıçta Makine 2’de 1 birim işlem süresi boyunca işlenmelidir; daha sonra 3 birim süre Makine 0’a gitmelidir ve daha sonra 6 zaman birimi için Makine 1’e gitmelidir. (Makineler 0’dan başlayarak numaralandırılır).

Bu çalışmada ele alınan varsayımlar şunlardır:

- Makinelerin kurulum süreleri ihmal edilebilir düzeydedir.
- Makineler birbirinden bağımsızdır.
- İşler birbirinden bağımsızdır.
- Bir makine aynı anda birden fazla işi işleyemez.
- Farklı işlerin işlemleri arasında herhangi bir öncelik kısıtlaması yoktur.

Tezde, birinci bölümde, giriş yapılmış konuyla ilgili genel bilgiler verilmiştir. İkinci bölümde, literatür taraması yapılmış, daha önceden benzer konularda yapılmış çalışmalar hakkında bilgi verilmiştir. Üçüncü bölümde, kullanılacak olan gereç ve yöntemler hakkında bilgi verilmiştir. Dördüncü bölümde, yapılan çalışma sonucunda tespit edilen bulgular sunulmuştur. Beşinci bölümde, bulgular üzerine tartışılmıştır. Altıncı bölümde, bu çalışma kapsamında elde edilen sonuçlar belirtilmiştir.

2. LİTERATÜR ARAŞTIRMASI

Sürü zekâsı, merkezi olmayan kontrol ve öz-örgütlenme kullanarak koordine eden birçok kişiden oluşan doğal ve yapay sistemlerle uğraşan bir disiplindir. Özellikle disiplin, bireylerin birbirleriyle ve çevreleriyle olan yerel etkileşimlerinden kaynaklanan kolektif davranışlara odaklanır. Sürü zekâsı tarafından incelenen sistemlere örnek olarak karıncalar ve termitler kolonileri, balık okulları, kuş sürüleri, kara hayvan sürüleri sayılabilir. Bazı insan eserleri aynı zamanda sürü zekâsı alanına, özellikle bazı çoklu robot sistemlerine, optimizasyon ve veri analizi problemlerini çözmek için yazılmış bazı bilgisayar programlarına da girer.

Bu bölümde kullanılacak olan optimizasyon yöntemlerinden biri olan sürü algoritmaları ile genel bilgi verilmiş. Sonrasında kullanılacak olan Arı Kolonisi ve Parçacık Sürü Optimizasyon algoritmalarının genel çalışma mantığı anlatılmıştır. Algoritmaların esinlenilen doğal sürüler üzerinde yapılan çalışmaları ve bu çalışmaların literatürde nasıl işlendiği konusunda kronolojik tanıtımı yapılmıştır.

2.1. Sürü Zekâsının Sınıflandırması

Sürü zekâsı çok disiplinli bir karaktere sahiptir. Çünkü yukarıda belirtilen özelliklere sahip sistemler çeşitli alanlarda gözlenebilir. Sürü zekâsı araştırmaları farklı kriterlere göre sınıflandırılabilir.

Doğal ve Yapay: Sürü zekâsı araştırmalarını analiz edilen sistemlerin doğasına göre iki alana bölmek geleneksel bir yöntemdir. Bu nedenle biyolojik sistemlerin çalışıldığı doğal ve insan eserlerinin çalışıldığı yapay sürü zekâsı araştırmasından bahsediyoruz.

Bilimsel ve Mühendislik: Sürü zekâsı araştırmalarının alternatif ve bir şekilde daha bilgilendirici bir şekilde sınıflandırılması, izlenen hedeflere dayanarak verilebilir: bir bilimsel ve bir mühendislik akışını tanımlayabiliriz. Bilimsel akışın amacı, sürü zekâsı sistemlerini modellemek ve bir sistemin bir bütün olarak yerel

bireysel-bireysel ve bireysel-çevre etkileşimlerinin bir sonucu olarak koordineli bir şekilde çalışmasını sağlayan mekanizmaları ortaya çıkarmak ve anlamaktır. Öte yandan, mühendislik akışının amacı, pratik alaka problemlerini çözebilecek sistemleri tasarlamak için bilimsel akım tarafından geliştirilen anlayıştan yararlanmaktır.

Doğal / yapay ve bilimsel / mühendislik ikiye ayrımı dikeydir, tipik bilimsel araştırma doğal sistemleri ilgilendirse ve tipik mühendislik uygulaması yapay bir sistemin geliştirilmesini ilgilendirse de biyolojik sistem modellerinin matematiksel geçerliliğini doğrulamak için robot sürüleri ile bir dizi sürü zekâsı çalışması yapılmıştır. Bu çalışmalar sadece spekülâtif bir yapıya sahiptir ve kesinlikle sürü zekâsının bilimsel akışına aittir. Öte yandan, bireyler biyolojik bir kümedeki davranışlarını etkileyebilir veya değiştirebilir. Böylece bir miktar pratik ilgi görevinin çözümünde işlevsel olan yeni bir kümelenme seviyesindeki davranış ortaya çıkar. Bu durumda, eldeki sistem doğal bir sistem olmasına rağmen, takip edilen hedefler kesinlikle bir mühendislik uygulamasının hedefleridir. Aşağıda, dört olası durumun her biri için bir örnek verilmiştir.

2.1.1. Doğal / Bilimsel: Karıncaların Yiyecek Arama Davranışı

1990'da yapılan klasik bir deneyde, Deneubourg ve grubu, bir yiyecek kaynağına birleştirilen farklı uzunluktaki iki yol arasındaki seçim yapıldığında, bir karınca kolonisinin kolektif olarak daha kısa olanı seçme ihtimalinin yüksek olduğunu gösterdi. Deneubourg, bu davranışın her karınca nereye gideceğine karar verirken zeminde algılanan feromonun yoğunluğuna dayanarak rastgele kararlar alırken feromon yuvadan besin kaynağına geçerken ve geriye dönerken karıncalar tarafından biriktirildiğini basit bir olasılıklı modelle açıklanabileceğini göstermiştir.

2.1.2. Yapay / Bilimsel: Bir Robot Sürüsü Tarafından Kümelenme

Birkaç karınca türü mezarlık oluşturmak için ceset toplar. Deneubourg ve diğ. (1991) bu kümelenme davranışını açıklamak için dağıtılmış bir olasılıksal model öneren ilk kişiler arasındaydı. Modellerinde, karıncalar, karıncalar için yerel olarak mevcut olan ceset yoğunluğuna bağlı olarak maddeleri alır ve bırakırlar. Beckers ve diğ. (1994), yapay ajanların kullanıldığı ilk sürü zekâsı bilimsel yönelimli

çalışmalardan birini bu şekilde gösteren benzer bir kümelenme davranışı uygulamak için bir grup robot programlamıştır.

2.1.3. Doğal / Mühendislik: Hayvan Topluluklarının Kolektif Davranışlarının Kullanılması

Sürü zekâsının muhtemel bir gelişimi, hayvan topluluklarının kolektif davranışlarının kontrollü bir şekilde kullanılmasıdır. Bazı ümit vaat eden araştırmalar halen devam etmesine rağmen, bu tür sürü istihbarat alanında hiçbir örnek bulunmamaktadır. Örneğin, Leurre projesinde, bir grup hamamböceği davranışını etkilemek için küçük böcek benzeri robotlar yem olarak kullanılmaktadır. Bu proje kapsamında geliştirilen teknoloji, tarım ve büyükbaş hayvancılık da dahil olmak üzere çeşitli alanlara uygulanabilir.

2.1.4. Yapay / Mühendislik: Sürü Tabanlı Veri Analizi

Mühendisler, karıncaların kümelenme davranışı modellerini veri madenciliği algoritmaları tasarlamak için bir ilham kaynağı olarak kullandılar. Bu yönde yeni ufuklar açan bir çalışma, 1994 yılında Lumer ve Faieta tarafından üstlenildi. Yapay karıncaların, civarlarında zaten mevcut olan diğer veri maddelerinin benzerlikleri tarafından yönetilen olasılıklara sahip veri öğelerini toplayıp bıraktıkları yapay bir ortam tanımladılar. Aynı algoritma, kümelenme problemleri olarak yeniden düzenlenmiş kombinatoriyel optimizasyon problemlerini çözmek için de kullanılmıştır (Bonabeau ve ark. 1999).

2.2. Sürü Zekâsı Sisteminin Özellikleri

Tipik sürü zekâsı sistemi aşağıdaki özelliklere sahiptir:

- Birçok kişiden oluşur.
- Bireyler nispeten homojendir. Yani hepsi aynıdır veya birkaç tipolojiye aittir.
- Bireyler arasındaki etkileşimler, bireylerin doğrudan ya da çevre yoluyla takas ettikleri yalnızca yerel bilgileri kullanan basit davranış kurallarına dayanmaktadır.

- Sistemin genel davranışı, bireylerin birbirleriyle ve çevreleriyle etkileşimlerinden kaynaklanmaktadır. Yani grup davranışı kendi kendine organize eder.

Bir sürü istihbarat sisteminin karakteristik özelliği, bir koordinatör veya harici bir kontrolör olmadan koordineli bir şekilde hareket edebilmesidir. Grubu kontrol eden herhangi bir kişi olmadan bazı genel davranışlar sergileyen veya genel grup davranışının farkında olmayan sürülerin doğasında birçok örnek gözlemlenebilir. Gruptan sorumlu olan bireylerin olmamasına rağmen, bir sürü akıllı davranış gösterebilir. Bu, basit kurallara dayanarak hareket eden mekânsal olarak komşu bireylerin etkileşiminin sonucudur.

Çoğu zaman, sürünün her birisinin davranışı olasılıksal terimlerle tanımlanmaktadır. Her bireyin, civarını yerel olarak algılamasına bağlı olan stokastik bir davranışı vardır.

Yukarıdaki özelliklerden ötürü, ölçeklenebilir, paralel ve hata toleranslı bir sürü zekâsı sistemi tasarlamak mümkündür.

- Ölçeklendirilebilir bir sistemin, parçalarının etkileşim şeklini yeniden tanımlamak zorunda kalmadan boyutunu büyütürken işlevini koruyabileceği anlamına gelir. Sürü zekâsı sistemi etkileşimlerinde yalnızca komşu bireyleri içerdiği için, etkileşimlerin sayısı sürünün içindeki toplam birey sayısı ile artma eğilimindedir. Her bireyin davranışı yalnızca sürü boyutundan esnek bir şekilde etkilenir. Yapay sistemlerde ölçeklenebilirlik ilginçtir, çünkü ölçeklenebilir bir sistem, herhangi bir yeniden programlamaya gerek kalmadan, sadece boyutunu artırarak performansını artırabilir.
- Sürü zekâsı sistemlerinde paralel eylem mümkündür, çünkü sürüyü oluşturan bireyler aynı anda farklı yerlerde farklı eylemler gerçekleştirebilirler. Yapay sistemlerde, paralel eylem istenir çünkü sistemin daha esnek olmasına, yani karmaşık bir görevin farklı yönleriyle aynı anda ilgilenen ekiplerde kendi kendini organize etme yeteneğine sahip olabilir.
- Hata toleransı, kontrol yapılarının merkezi olmayan, kendi kendine organize doğası nedeniyle, sürü zekâsı sistemlerinin doğal bir özelliğidir. Sistem birbiriyle değiştirilebilir birçok kişiden oluştuğundan ve bunların hiçbiri

genel sistem davranışını kontrol etmekten sorumlu olmadığından, başarısız olan bir kişi kolayca görevden alınabilir ve tamamen çalışan bir başkası tarafından ikame edilebilir.

2.3. İş Atama Problemi

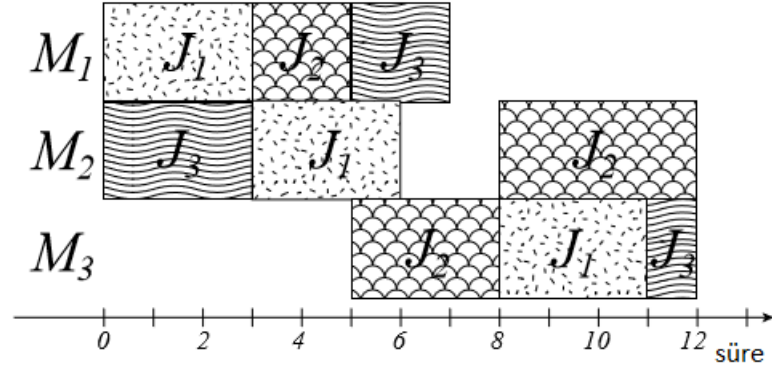
Çizelgeleme, zaman içinde paylaşılan kaynakların rakip faaliyetlere tahsis edilmesidir. Yöneylem araştırması alanında önemli miktarda literatüre konu olmuştur. İşlerin faaliyetleri temsil ettiği ve makinelerin kaynakları temsil ettiği makine çizelgeleme problemlerinin araştırılması üzerinde durulmuştur; her makine aynı anda en fazla bir işi işleyebilir. Aşağıda örnek bir iş atama problemi veri seti görülmektedir (Çizelge 2.1).

Çizelge 2.1. 3×3'lük bir iş atama sorunu

İş	Operasyon Rotası (İşlem Süresi)		
1	1 (3)	2 (3)	3 (3)
2	1 (2)	3 (3)	2 (4)
3	2 (3)	1 (2)	3 (1)

Bundan sonra iş atama problemi $n \times m$ minimum-tamamlanma zamanında, bir dizi m makinesinde $\{M_r\} 1 \leq r \leq m$ işlenecek olan bir dizi $\{J_i\} 1 \leq i \leq n$ işiyle tanımlanabilir. Her işin işlenecek teknolojik bir makine sırası vardır. M_r makinesinde, J_j işinin işlenmesine O_{jr} işlemi denir. Operasyon O_{jr} , M_r 'nin işlem süresi boyunca kesintisiz bir süre için özel olarak kullanılmasını gerektirir. Bir program, bu kısıtlamaları karşılayan her işlem için $\{C_{jr}\} 1 \leq j \leq n, 1 \leq r \leq m$ tamamlanma süreleridir. Tüm işleri tamamlamak için gereken süreye L tamamlanma zamanı denir. Bu genel problemi çözerken veya optimize ederken amaç, L 'yi minimize edecek zamanlamayı belirlemektir. İş Atama Problemi 'nin 3×3'lük bir örneği Çizelge 2.1'de verilmektedir. Veriler, her işin her makine boyunca yönlendirilmesini ve her işlem için işlem süresini (parantez içinde) içerir.

Gantt Şeması, iş atamanın bir çözümünü görsel olarak temsil etmenin uygun bir yoludur. Çizelge 2.1'deki 3×3 problemi için bir çözüm örneği Şekil 2.1'de verilmektedir (Şekil 2.1).



Şekil 2.1. 3×3'lük problem için çözümün gantt şema gösterimi (Shailendra ve dig., 2014)

İş atama sadece NP zor değil, aynı zamanda sınıfın en kötü üyelerinden biridir.

Bunun bir göstergesi, Muth ve Thompson tarafından formüle edilen bir 10×10 sorununun 20 yıldan fazla bir süre çözülmemiş kalmasıyla verilmektedir.

Şube ve sınır yöntemlerine dayanan kapsamlı arama algoritmalarının yanı sıra, birkaç yaklaşım algoritması geliştirilmiştir. Uygulamada en popüler olanları öncelik kurallarına ve aktif program üretmeye dayanmaktadır (S. S. Panwalkar and Wafix Iskander, 1977). Değişen Darboğaz (SB) olarak adlandırılan daha sofistike bir yöntemin çok başarılı olduğu gösterilmiştir (J. Adams, E. Balas, and D. Zawack, 1988). Ek olarak, Benzetilmiş Tavlama (SA), Tabu Arama (TS) (M. Dell'Amico and M. Trubian, 1993) ve Genetik Algoritmalar (GA) gibi stokastik yaklaşımlar son zamanlarda başarılı bir şekilde uygulanmıştır.

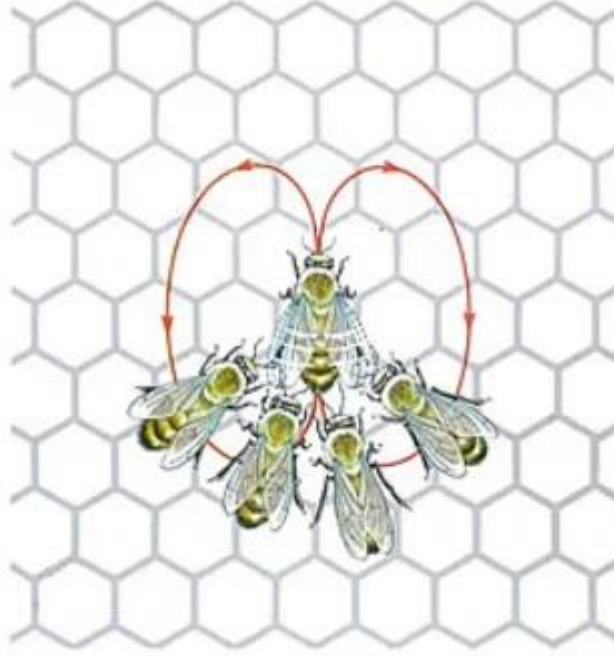
2.4. Bal Arısı Koloni Algoritması ile İş Atama Problemi

Sürü zekâsı olarak bilinen bir doğadan esinlenilmiş algoritma, böceklerin problem çözme yeteneklerini taklit edebilecek bazı meta sezgiler geliştirmek için böcek davranışına odaklanır. Karınca kolonisi optimizasyonu, parçacık sürüsü optimizasyonu, eşekarısı ağları vs., sorun modelleme ve çözümünde böcek davranışını taklit eden iyi bilinen algoritmalarından bazılarıdır. ABC, sürü zekâsının nispeten yeni bir üyesidir. ABC, yiyecek aramada gerçek bal arılarının doğal davranışlarını modellemeye çalışır. Bal arıları, besin kaynaklarını en uygun şekilde konumlandırmak ve yenilerini aramak için sallanma dansı gibi çeşitli mekanizmalar kullanır. Bu, onları yeni akıllı arama algoritmaları geliştirmek için iyi bir aday yapar.

Bu bölümde yapay arı algoritmaları ile ilgili çalışmaların kapsamlı bir derlemesi verilmiştir. Daha sonra, NP-zor problemi olarak bilinen atama problemini çözmek için bir ABC algoritmasının geliştirilmesi, bazı karşılaştırmalarla birlikte ayrıntılı olarak sunulmuştur.

Böcekler arasındaki etkileşim, sosyal böcek kolonilerinin ortak zekâsına katkıda bulunur. Böcekler arasındaki bu iletişim sistemleri, optimizasyon için bilimsel sorunlara uyarlanmıştır. Bu tür etkileşimli davranış örneklerinden biri, gıda tedariki sırasında arıların sallantılı dansıdır. Başarılı dansçılar bu dansı yaparak, çiçek yamalarının yönü ve uzaklığı ile bu çiçek içindeki nektar miktarı hakkındaki bilgiyi kovan arkadaşları ile paylaşırlar. Bu nedenle, yem toplayıcıların kolonilerindeki diğer arıları, çeşitli kaynakları toplamak için verimli yerlere toplayabilecekleri başarılı bir mekanizmadır. Arı kolonisi değişen nektar kaynaklarına göre zaman ve mekandaki arama düzenini hızlı ve hassas bir şekilde ayarlayabilir.

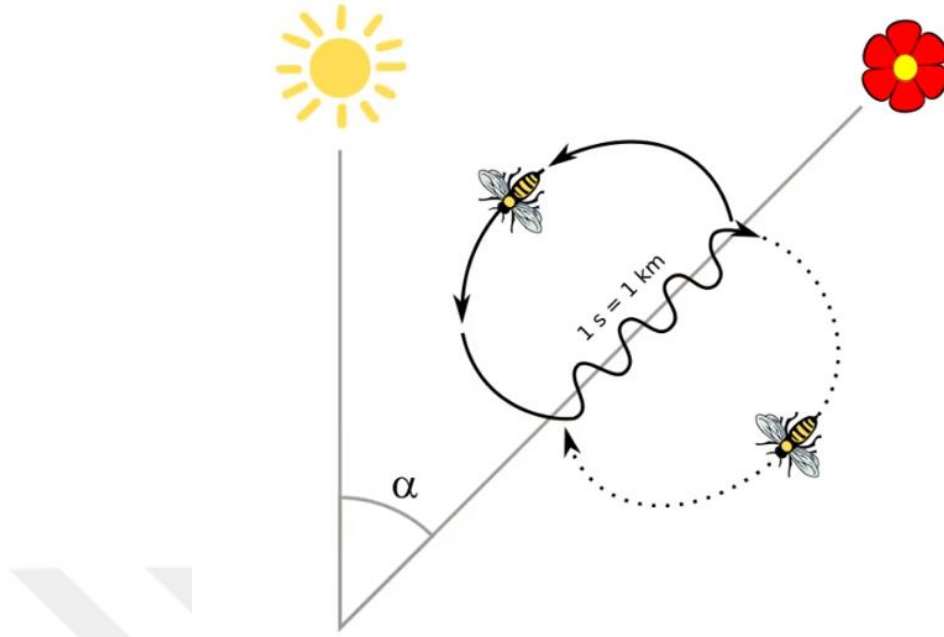
Bireysel böcekler arasındaki bilgi alışverişi, kolektif bilginin en önemli parçasıdır. Dans eden alanda arılar arasında besin kaynaklarının kalitesi ile ilgili iletişim kurma dansı yapılarak sağlanmaktadır (Şekil 2.2).



Şekil 2.2. Bal arılarının titreme dansı

Sürü zekâsı olarak adlandırılan doğadan ilham alan algoritmaların bir dalı, bir böceğin problem çözme yeteneklerini taklit edebilecek bazı meta-sezgisel tarama geliştirmek için böcek davranışına odaklanır. Böcekler arasındaki etkileşim, sosyal böcek kolonilerinin kolektif zekasına katkıda bulunur. Böcekler arasındaki bu iletişim sistemleri, optimizasyon için bilimsel sorunlara uyarlanmıştır. Bu tür etkileşimli davranışların örneklerinden biri, yiyecek alımı sırasında arıların titreme dansıdır. Bu dansı yaparak başarılı yemciler, çiçeğin yön / mesafe ve bu çiçeğin içindeki nektar miktarı hakkındaki bilgileri kovan arkadaşlarıyla paylaşırlar. Dolayısıyla bu, yem arama makinelerinin kolonilerindeki diğer arıları çeşitli kaynakları toplamak için verimli yerlere toplayabildikleri başarılı bir mekanizmadır. Bir arı kolonisi, değişen nektar kaynaklarına göre zaman ve mekandaki arama modelini hızlı ve hassas bir şekilde ayarlayabilir. Bireysel böcekler arasında bilgi alışverişi, kolektif bilginin en önemli parçasıdır. Arılar arasında dans kaynaklarında titreme dansı yapılarak gıda kaynaklarının kalitesi ile ilgili iletişim sağlanmaktadır.

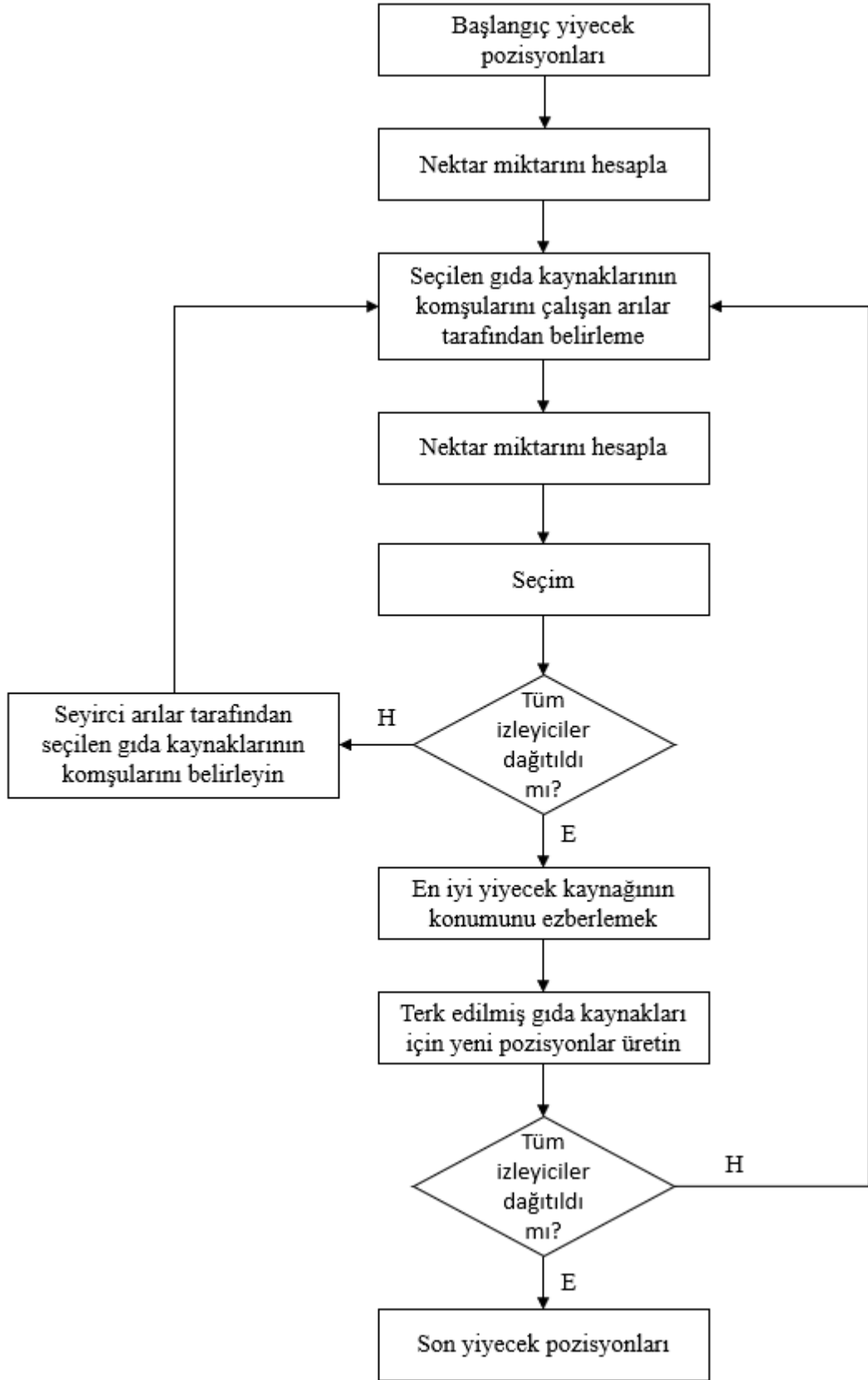
Şekilde 2.3’de görüldüğü üzere, arıların dans etme davranışı üzerine yapılan önceki çalışmalar, titreme dansı yaparken arıların yönünün Güneş ile ilgili olarak besin kaynağının yönünü gösterdiğini, dansın süresinin ilgili gıda kaynağına ne kadar uzaklıkta olduğunu, titreme sıklığı ise ilgili gıda kaynağındaki nektar miktarı hakkında bilgi verdiğini göstermiştir. Her 1sn’lik titremenin 1km gibi bir uzaklığa denk geldiği varsayılmaktadır. Kovanda uzun süredir bulunan dans eden arılar, danslarının açısını güneşin değişen yönüne uyacak şekilde ayarlar. Bu nedenle dansın salınımını takip eden arılar, güneşe göre açılarını değiştirmiş olsa bile, hala doğru bir şekilde gıda kaynağına yönlendirilmektedir. Arıların toplu zekâsı, titreme dansı sırasında karşılıklı etkili olan bilgi alışverişine dayanır (Şekil 2.3).



Şekil 2.3. Titreme dansı örnek gösterim

Bal arısı davranışlarına ilişkin gözlemler ve çalışmalar yeni nesil optimizasyon algoritmaları ile sonuçlandı. Bu bölümde arı kolonisine dayalı algoritmaların detaylı bir incelemesi verilmektedir. Daha sonra "yapay arı kolonisi" olarak adlandırdığımız arı temelli bir algoritma, NP-zor bir problem olarak bilinen "iş atama problemi" bir uygulama ile birlikte ayrıntılı olarak açıklanmaktadır.

Bu problemin çözümünde kullanılacak Arı Kolonisi algoritmasının anlaşılmasını kolaylaştırmak için, Şekil 2.4'deki gibi bir akış şeması verilmiştir (Akay, B., Karaboga D, 2011).



Şekil 2.4. BCO akış şeması

Bu akışa göre, yiyecek kaynakları çözüm uzayına dağıtılır. Bu yiyecek kaynaklarının nektar miktarları (uygunluk değerleri) hesaplanır. Her gıda kaynağının komşuları için de uygunluk değeri hesaplanır. Eğer komşunun uygunluk değeri daha iyi ise, seçilen yiyecek kaynağı değiştirilir. Kovanda bulunan izleyici arılar çözüm uzayına dağıtılır ve bir yiyecek kaynağına gönderilir. Bu seyirci arılar, gıda kaynaklarının komşularının bilgilerini de tutar ve ilk başta seçtikleri yiyecek kaynağının uygunluk değeri ile karşılaştırırlar. Bu karşılaştırma sonucunda bulunan yiyecek kaynağının bilgisi tutulur. Bu esnada, uygunluk değeri azalmış olan yiyecek kaynaklarının yerine yeni gıda kaynakları tespit edilir. Bu algoritmanın lokal arama özelliğini göstermektedir. Bu döngü bütün seyirci arılar dağıtılabildiği kadar tekrarlanır. Bütün seyirci arıların yiyecek kaynağı uygunluk tespiti sonrası son yiyecek kaynaklarının konumları belirlenir.

2.4.1. Arıların Doğadaki Davranışlarının Tanımlanması

Sosyal böcek kolonileri, çevreden bilgi toplayan ve davranışını buna göre ayarlayan dinamik bir sistem olarak düşünülebilir. Bilgi toplama ve ayarlama işlemleri sırasında, bireysel böcekler uzmanlıklarından dolayı tüm görevleri yerine getirmezler. Genel olarak, tüm sosyal böcek kolonileri, morfolojileri ile ilgili bölüm çalışmalarına göre davranır. Arı sistemi iki temel bileşenden oluşur:

- **Gıda Kaynakları**

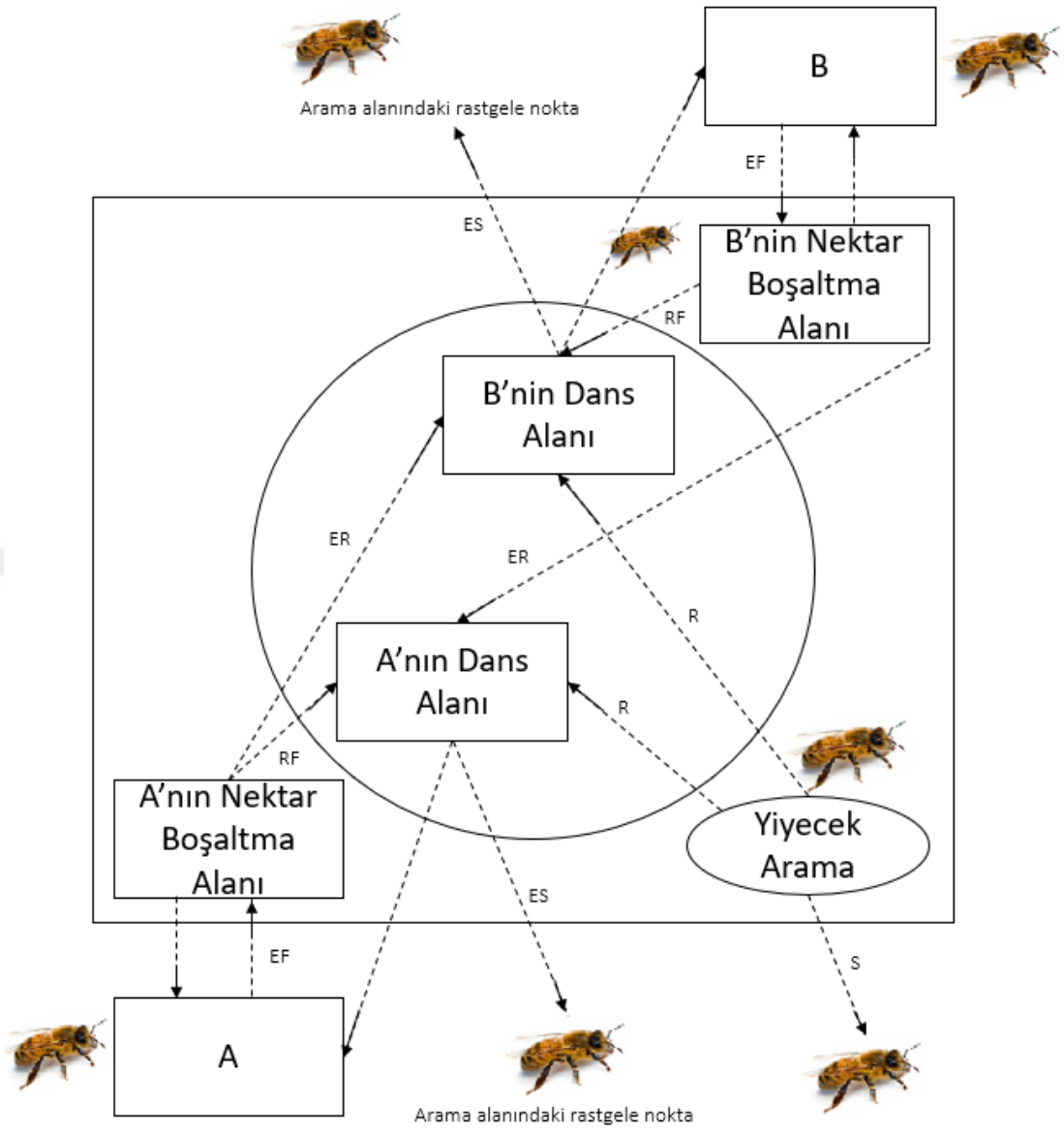
Bir besin kaynağının değeri, yuvaya yakınlığı, enerji zenginliği ve bu enerjiyi çıkarmanın kolaylığı gibi farklı parametrelere bağlıdır.

- **Toplayıcılar**

- **İşsiz Toplayıcılar:** Bir arının, arama alanındaki gıda kaynakları hakkında hiçbir bilgisi olmadığı varsayılırsa, arı, işsiz bir arama aracı olarak araştırmasını başlatır. İşsiz bir toplayıcı için iki olasılık var:

- **İzci Arısı (Şekil 2.5'deki S):** Arı herhangi bir bilgiye sahip olmadan kendiliğinden aramaya başlarsa, izci arısı olacaktır. İzci arılarının yüzdesi, yuvadaki bilgilere göre %5 ile %30 arasında değişmektedir. Koşullar üzerinden ortalama izci sayısı yaklaşık %10'dur (Seeley, 1995).

- Acemi Arı (Şekil 2.5'deki R): Eğer işsiz yiyecek arayan kişi başka bir arı tarafından yapılan bir titreme dansına katılırsa, arı titreme dansındaki bilgiyi kullanarak aramaya başlar.
- İşçi toplayıcılar (Şekil 2.5'deki EF): Acemi arı gıda kaynağını bulup kullandığında, gıda kaynağının yerini ezberleyen çalışan bir işçi olacaktır. Kullanılan yem toplayıcı arı, yiyecek kaynağından nektarın bir kısmını yükledikten sonra, kovana geri döner ve nektarı kovandaki yiyecek alanına boşaltır. Toplayıcı arı için kalan nektar miktarıyla ilgili üç seçenek vardır.
 - Nektar miktarı düşük bir seviyeye düşerse veya tükenirse, yem toplayıcı arı gıda kaynağını terk eder ve işsiz bir arı olur.
 - Gıda kaynağında hala yeterli miktarda nektar varsa, gıda kaynağı bilgilerini yuva arkadaşlarıyla paylaşmadan toplamaya devam edebilir
 - Yuva arkadaşlarına aynı besin kaynağı hakkında bilgi vermek için sallama dansı yapmak dans alanına gidebilir. Bu seçenekler için olasılık değerleri, gıda kaynağının kalitesiyle büyük ölçüde ilgilidir.
- Deneyimli toplayıcılar: Bu toplayıcı türleri, hafızalarını yiyecek kaynaklarının yeri ve kalitesi için kullanırlar.
 - Daha önce keşfedilen gıda kaynağının son durumunu kontrol eden bir müfettiş olabilir.
 - Titreme dansından elde edilen bilgileri kullanarak yeniden etkinleştirilmiş bir toplayıcı olabilir. Aynı gıda kaynağının kalitesini onaylayan başka arılar varsa, kendisi tarafından keşfedilen aynı gıda kaynağını keşfetmeye çalışır (Şekil 2.5'deki RF).
 - Besin kaynağının tamamı tükenirse, yeni kaynakları aramak için izci arısı olabilir (Şekil 2.5'deki ES).
 - Dans alanında ilan edilen yeni bir yiyecek kaynağını başka bir çalışan arı tarafından arayan yeni bir arı olabilir (Şekil 2.5'deki ER).



Şekil 2.5. Bal arısı toplayıcısının tipik davranışı

2.4.2. Yapay Arı Sistemlerinin Avantajları ve Dezavantajları

Avantajları:

1. Uygulaması kolaydır.
2. Karmaşık fonksiyonlarda veya sürekli, ayrık veya karışık değişkenlerde bile geniş uygulanabilirlik.
3. Doğayı gözlemleyerek sorunun spesifik bilgilerinin ayarlanmasına ve tanıtılmasına izin veren yüksek esneklik.

4. Objektif fonksiyonun farklılaştırılabilir, sürekli veya matematiksel olarak temsil edilebilir olmasını gerektirmez.
5. Algoritmanın yapısı paralel işleme için uygundur, böylece zaman kazanır.
6. Yerel çözümleri keşfedebilir.
7. Yüksek karmaşıklıkta bile etkili arama süreci ve düşük erken yakınsama riski ile küresel optimizasyon.

Dezavantajları:

1. Sorun hakkında ikincil bilgilerin kullanılmaması.
2. Performansı artırmak için yeni algoritma parametreleri üzerinde yeni kondisyon testleri gerektirir.
3. Optimize edilecek işlevin davranışı hakkında ilgili bilgileri kaybetme olasılığı.
4. Çok sayıda objektif fonksiyon değerlendirmesi.
5. Sıralı işlemlerde kullanıldığında yavaş çalışır.
6. Çözüm popülasyonu, hesaplama maliyetini artırır.
7. Deterministik yöntemler, yerel minimumda takılmadığında bulgu çözümlerinde daha yüksek doğruluktadır

2.4.3. Yapay Arı Sistemlerinde Yapılan Çalışmaların Gözden Geçirilmesi ve Sınıflandırılması

Arıların yiyecek arama davranışı, öğrenme, ezberleme ve bilgi paylaşma özellikleri, son zamanlarda sürü zekâsındaki en ilginç araştırma alanlarından biri olmuştur. Bal arıları üzerine yapılan araştırmalar literatürde son birkaç yıl içinde artış eğilimindedir. Detaylı bir literatür taramasından sonra, bu çalışmada önceki arı algoritmaları, bal arılarının davranışsal özelliklerine göre sınıflandırılmıştır.

- Yiyecek arama davranışları
- Evlilik davranışları
- Kraliçe Arı Kavramı

Araştırmalar, ana katkıları ve uygulamaları aşağıdaki çizelgelerde gösterildiği gibi özetlenmiştir (Çizelge 2.2, 2.3, 2.4).

Çizelge 2.2. Literatür ve uygulamaların sınıflandırılması

Bal Arısı Literatürü	Algoritma	Uygulama
Yonezawa and Kikuchi (1996) Seeley and Buhrman (1999) Schmickl et al. (2005) Lemmens (2006)		Biyolojik Simülasyon
Sato and Hagiwara (1997)	Arı Sistemi	Genetik Algoritma Gelişimi
Karaboga (2005)	ABC (Yapay Arı Kolonisi Algoritması)	Sürekli Optimizasyon
Yang (2005)	VBA (Sanal Arı Algoritması)	Sürekli Optimizasyon
Basturk and Karaboga (2006)	ABC	Sürekli Optimizasyon
Pham et al. (2006a)	Arı Algoritması	Sürekli Optimizasyon
Lucic and Teodorovic (2001)	Arı Sistemi	TSP (Gezgin Satıcı Algoritması)
Lucic (2002)	Arı Sistemi	Araç Rotalama Problemi ve TSP
Lucic and Teodorovic (2002)	Arı Sistemi	TSP
Lucic and Teodorovic (2003a)	Arı Sistemi	TSP
Lucic and Teodorovic (2003a)	Arı Sistemi	TSP
Lucic and Teodorovic (2003b)	Arı Sistemi + Bulanık Mantık	Stokastik Araç Rotalama

Çizelge 2.3. (devam)Literatür ve uygulamaların sınıflandırılması

Teodorovic and Dell'Orco (2005)	BCO (Arı Kolonisi Optimizasyonu) + FBS (Bulanık Arı Sistemi)	Sürücü Eşleme Problemi
Teodorovic and Dell'Orco (2005)	BCO (Arı Kolonisi Optimizasyonu) + FBS (Bulanık Arı Sistemi)	Sürücü Eşleme Problemi
Nakrani and Tovey (2003)	Arı Kolonisi Algoritması	İnternet Servislerinde Dinamik Tahsis
Wedde et al. (2004)	Arı Yaşamı	Telekomünikasyon Ağı Yönlendirme
Bianco (2004)		Büyük Ölçekli Hassas Navigasyon
Chong et al. (2006)		Atölye Çizelgeleme
Drias et al. (2005)	Arı Sürüsü	Max-W Sat Problemi
Pham et al. (2006b)	Arı Algoritması	LVQ – Sinirsel Ağ
Pham et al. (2006c)	Arı Algoritması	MLP – Sinirsel Ağ
Pham et al. (2006d)	Arı Algoritması	Sinirsel Ağ
Quijano and Passino (2007)		Dinamik Kaynak
Markovic et al. (2007)	BCO Tabanlı	Max – Routing ve Dalga boyu atama
Abbass (2001a, b, c)	MBO (Arı Algoritmalarında Evlilik)	3- Sat Problemi
Teo and Abbass (2001, 2003)	Değiştirilmiş Arı Algoritmalarında Evlilik	3- Sat Problemi
Bozorg Haddad and Afshar (2004)	Arı Algoritmalarında Evlilik	Su Kaynakları Yönetim Problemleri

Çizelge 2.4. (devam)Literatür ve uygulamaların sınıflandırılması

Bozorg Haddad et al. (2006)	HBMO (Bal Arıları Çiftleşme Optimizasyonu)	Kısıtlı ve Kısıtsız Lineer Olmayan Optimizasyon
Afshar et al. (2007)	Geliştirilmiş MBO	Sürekli Optimizasyon
Fathian et al. (2007)	HBMO Tabanlı	Veri Madenciliği-Kümeleme
Teodorovic et al. (2016)	Arı Algoritması	Yakınsama
Akay (2020)	Arı Kolonisi Algoritması	Yazılım Kusur Tahmini
Nayyar et al. (2020)	Geliştirilmiş MBO	Robot Yolu Planlama
Karaboga ve Aslan (2020)	Arı Kolonisi	Büyük Veri Optimizasyonu
Karaboga et al. (2020)	ABC Algoritması	Bulanık Ağ Regresyonu

Bu bölümde, bal arılarına dayalı optimizasyon algoritmalarının arka planının açıklığa kavuşturulması için yukarıdaki çizelgede gösterilen araştırmaların katkıları ayrıntılı olarak açıklanmaktadır.

Yonezawa ve Kikuchi (1996) bal arılarının yiyecek arama davranışlarını incelemekte ve grup istihbarat müdürlerinin önemini gösteren bir algoritma oluşturmaktadır. Algoritma bir ve üç yemleme arısı ile simüle edilebilir ve hesaplama simülasyon sonuçları üç yemleme arısının karar verme sürecinde bir yemleme arısı bulunan sistemden daha hızlı olduğunu göstermiştir. Ayrıca bal arılarının karmaşık bir ortamda uyarlanabilir bir yemleme davranışına sahip olduklarını gösterirler.

Seeley ve Buhrman (1999), bal arısı kolonilerinin yuva yeri seçim davranışını araştırmışlardır. Yuva yeri seçim süreci, potansiyel yuva yerlerini arayan birkaç yüz keşif arısı ile başlar. İzciler daha sonra kümeye geri dönerler, bulgularını sallantı dansıyla rapor ederler ve yeni yuva bölgesine karar verirler. Sallantı dansı türü, reklamı yapılan kaynağın kalitesine bağlıdır. Yazarlar, üç bal arısı kolonisinde modern video kayıt ve arı etiketleme tekniklerinden faydalanarak Lindauer'ın gözlemlerini 1955 yılında tekrarladılar. Sonuçları çoğu önceki çalışma ile

doğrulanmış ve sonuçların bazıları yeni ve önemli görüşler sağlamıştır. Bir koloninin karar verme stratejisinin, en doğru fakat en çok bilgi isteyen strateji olan ağırlıklı bir katkı stratejisi olduğuna dikkat çektiler. Bu strateji, her bir alternatifi göreceli özelliklere göre değerlendirir, her özelliğe önemine göre ağırlık verir, her bir alternatif için ağırlıklı özellikleri toplar ve son olarak toplam değeri en yüksek olan alternatifi seçer. Benzer şekilde, arı kolonisi bir düzine veya daha fazla alternatif yuva bölgesini göz önünde bulundurur, her alternatif yuva bölgesini farklı ağırlıklara sahip en az altı farklı niteliğe göre değerlendirir; boşluk hacmi, giriş yüksekliği, giriş alanı, giriş yönü, vb. Sonuç olarak, arı kolonisi bu stratejiyi hem alternatif alanları değerlendirme görevi hem de bu alanların en iyilerini belirleme görevi olmak üzere birçok arı arasında dağıtarak kullanır.

Schmickl ve diğ. (2005) arıların yemleme davranışlarının sağlamlığını çok değişkenli bir simülasyon platformu kullanarak değerlendirmiştir. Çevresel dalgalanmaların zaman örüntüsünün yemleme stratejisini ve yemleme verimliliğini nasıl etkilediğini araştırırlar. Bir bal arısı kolonisinin ortak yem arama stratejisinin sağlam ve uyarlanabilir olduğu ve ortaya çıkan özelliklerinin koloninin en uygun çözümleri bulmasına izin verdiği sonucuna varmışlardır.

Lemmens (2006), yemleme görevinde feromon bazlı navigasyon algoritmalarının (biyolojik karınca kolonisi davranışından esinlenerek) feromon tabanlı olmayan navigasyon algoritmalarından (biyolojik arı kolonisi davranışından esinlenerek) daha iyi olup olmadığını araştırdı. Deneyle sonuçları:

- (i) Feromon tabanlı navigasyon algoritmalarının, küçük boyutlu dünyalarda yineleme basamağı başına daha az zaman kullandığını,
- (ii) Feromon tabanlı olmayan algoritmaların, yiyecekleri bulup toplarken önemli ölçüde daha hızlı olduğunu,
- (iii) Büyüyen dünya boyutlarıyla, feromon tabanlı olmayan algoritma sonunda zaman başına adım ölçümünde feromon tabanlı algoritmalarından daha iyi performans gösterir.

Tüm bu kârlara rağmen feromon tabanlı olmayan algoritmaların feromon tabanlı algoritmalarından daha az uyarlanabilir olduğu belirtilmektedir.

Sato ve Hagiwara (1997) bal arılarının yemleme davranışına dayanan geliştirilmiş bir genetik algoritma önermişlerdir. Bir bal arısı kolonisinde, her arı ayrı

ayrı yem arar. Bir arı yemi bulduğunda, bilgiyi dansla diğer birçok arıya bildirir ve yemi taşımak için bir işe başlarlar. İş bitirdiklerinde, her arı tekrar yeni bir kaynak bulmaya çalışır. Benzer şekilde, Arı Sistemi adlı önerilen algoritmada, önce küresel arama yapılır ve oldukça yüksek kondisyona sahip bazı kromozomlar (üstün kromozomlar) basit genetik algoritma kullanılarak elde edilir. İkincisi, birçok kromozom konsantre geçit tarafından üstün kromozomların bilgisini elde eder ve çok sayıda popülasyon kullanarak orada yoğun bir şekilde arama yapar. Geleneksel geçişte, her bir çift rastgele yapılırken, konsantre geçişte tüm kromozomlar, üstün kromozomlarla çiftler oluşturur. Son olarak, yalancı simpleks yöntemi Arı Sisteminin yerel arama kabiliyetinin artırılmasına katkıda bulunmaktadır. Bir döngüde bulunan çözüm tatmin edici değilse, genel arama tekrarlanır. Bilindiği gibi genetik algoritmaların küresel aranabilirliği iyidir, ancak yerel aranabilirlikten yoksundurlar. Diğer yandan, Arı Sisteminin yerel ve küresel aramanın bir araya gelmesi nedeniyle yerel bir optimum duruma düşme olasılığı düşüktür, çünkü algoritmanın amacı küresel algoritmayı bozmadan genetik algoritmanın yerel arama yeteneğini geliştirmektir. Deneysel çalışmalarda, Arı Sistemi geleneksel genetik algoritma ile karşılaştırılmış ve Arı Sisteminin özellikle oldukça karmaşık çok değişkenli fonksiyonlar için konvansiyonel genetik algoritmadan daha iyi performans gösterdiği bulunmuştur.

Karaboğa (2005), bal arısı sürüsünün yiyecek arama davranışını analiz eder ve Yapay Arı Kolonisi (ABC) adı verilen çok boyutlu ve çok modlu optimizasyon problemlerini çözmek için bu davranışı simüle eden yeni bir algoritma önerir. Algoritmanın ana adımları şunlardır:

- 1) Kullanılan arıları gıda kaynaklarına göndermek ve nektar miktarlarını belirlemek.
- 2) Seyirci arılar tarafından tercih edildikleri kaynakların olasılık değerini hesaplar.
- 3) Arılar tarafından terk edilen kaynakların sömürü sürecini durdurur,
- 4) Keşif kollarını rastgele yeni besin kaynakları bulmak için arama alanına gönderir.
- 5) Şimdiye kadar bulunan en iyi gıda kaynağını hatırla.

Algoritmada yapay bir arı kolonisi üç grup arıdan oluşur: işçi arılar, seyirci arılar ve izci arılar. İşçi arılar şu anda topladıkları belirli bir besin kaynağı ile ilişkilidir. Bu kaynak hakkında bilgi taşırlar ve bu bilgiyi sallantı dansı ile belirli bir olasılıkla paylaşırlar. İşsiz arılar toplayacakları bir besin kaynağı ararlar. İki tür işsiz arı vardır: İzci ve seyirciler. İzci çevrede yeni yiyecek kaynakları aramaya gerek kalmadan ararlar. İzci zaman zaman zengin, tamamen bilinmeyen gıda kaynaklarını yanlışlıkla keşfedebilirler. Öte yandan, seyirciler sallantı dansı gözlemlemektedir ve bu nedenle olasılık tabanlı bir seçim süreci kullanılarak gıda kaynaklarına yerleştirilmektedir. Bir gıda kaynağının nektar miktarı arttıkça, gıda kaynağının seyirciler tarafından tercih edilme olasılığı da artar. ABC algoritmasında, koloninin ilk yarısı kullanılan arılardan, ikinci yarısı ise seyirci arıları içerir. Her gıda kaynağı için sadece bir tane işçi arı vardır. Algoritmada dikkate alınan bir diğer konu, besin kaynağı arılar tarafından tüketilen arıların izci olmasıdır. Başka bir deyişle, bir gıda kaynağını temsil eden bir çözüm önceden belirlenmiş sayıda denemeyle iyileştirilmezse, o zaman gıda kaynağı, onu toplayan arısı tarafından terk edilir ve işçi arı bir izciye dönüştürülür. Algoritma iyi bilinen üç test fonksiyonu üzerinde test edilmiştir. Simülasyon sonuçlarından, önerilen algoritmanın tek-modlu ve çok modlu sayısal optimizasyon problemlerini çözmek için kullanılabileceği sonucuna varılmıştır.

Yang (2005) fonksiyon optimizasyonu problemlerinde etkili olan bir sanal arı algoritması (VBA) sunmaktadır. Algoritmanın ana adımları şunlardır:

- 1) Her bir arının bir hafıza ile ilişkili olduğu ilk sanal arı popülasyonunun oluşturulması.
- 2) Optimizasyon fonksiyonunu sanal yiyeceğe kodlamak.
- 3) Gıda lokasyonunu başkalarıyla iletme ölçütünü tanımlamak.
- 4) Sanal yiyecek arama için tüm sanal arıları rastgele yeni pozisyonlara yürütmek, yiyecek bulunmasını ve sanal sallantı dansı ile komşu arılarla iletişim kurulmasını sağlamak.
- 5) Arıların kodlanmış yoğunluğunu / yerlerini değerlendirmek.
- 6) Soruna çözüm bulmak için sonuçları çözümlenmek.

Bununla birlikte, önerilen algoritma genetik bir algoritmaya benzer, çoklu bağımsız arıların paralellüğünden dolayı çok daha verimlidir. Bu ifadeyi

gerçekleştirmek için, VBA algoritması, biri iki uçlu diğeri çok uçlu olmak üzere iki parametrelili iki işlev üzerinde test edilir. Sonuçlar, yeni algoritmanın genetik algoritmadan çok daha verimli olduğunu göstermektedir.

Baştürk ve Karaboğa (2006) başka bir ABC algoritması sundular ve Karaboğa'nın (2005) deney sonuçlarını genişlettiler. Algoritmanın performansı, beş çok boyutlu karşılaştırma fonksiyonu üzerinde test edildi ve sonuçlar genetik algoritmalarla karşılaştırıldı. ABC algoritmasının, çok-modlu ve tek-modlu fonksiyonlar için genetik bir algoritmadan daha iyi performans gösterdiğine dikkat çekilmektedir.

Pham ve diğ. (2006a), bal arılarının Arılar Algoritması adı verilen doğal yemleme davranışlarından esinlenen bir optimizasyon algoritması önermiştir. Önerilen algoritma hem kombinatoriyel hem de fonksiyonel optimizasyon problemleri için de geçerlidir. Gerçek hayatta yiyecek arama süreci, umut verici yiyecek kaynakları aramak için keşif arılarının gönderilmesiyle başlar. Kovana döndüklerinde, nektarlarını boşaltır ve koloni iletişimi için gerekli olan sallanma dansı olarak bilinen bir dans yapmak için dans pistine gider. Sallanarak dans ettikten sonra, dansçı kovanın içinde bekleyen izci arılarla yiyecek kaynağına geri döner. Daha fazla izci arı daha gelecek vaat eden nektar kaynaklarına gönderilir. Bu, koloninin yiyecekleri hızlı ve verimli bir şekilde toplamasına olanak tanır. Benzer şekilde, Arılar Algoritması, izci arıların arama alanına rastgele yerleştirilmesiyle başlar. Algoritmanın ana adımları şunlardır:

- 1) Popülasyonu rastgele çözümlerle başlatmak.
- 2) Nüfusun uygunluğunu değerlendirmek.
- 3) Belirli sayıda en uygun arıları belirlemek ve komşulukları aramak için yerlerini seçmek.
- 4) Seçilen alanlar için belirli sayıda arı yetiştirmek, uygunluklarını değerlendirmek.
- 5) Yeni popülasyonu oluşturmak için her bölgeden en uygun arıyı seçmek.
- 6) Rastgele arama yapmak ve uygunluklarını değerlendirmek için kalan arıları atamak.

Arılar Algoritması, sırasıyla iki ve altı boyutlu iki standart fonksiyonel optimizasyon problemine uygulandı. Sonuçlar, Arılar Algoritmasının optimum seviyeye çok yakın

çözümler bulabildiğini gösterdi. Algoritma ayrıca sekiz kıyaslama fonksiyonuna uygulanır ve sonuçlar deterministik simpleks yöntemi, stokastik simüle tavlama optimizasyon prosedürü, genetik algoritma ve karınca kolonisi sistemi ile karşılaştırılmıştır. Arılar Algoritması, optimizasyon hızı ve sonuçların doğruluğu açısından genellikle diğer tekniklerden daha iyi performans gösterdi. Öte yandan Arılar Algoritması, çok fazla ayarlanabilir parametreye sahiptir.

Luck ve Teodorovic (2001) doktora tezinde Arı Sistemi üzerine Gezgin Satıcı Sorunu (TSP) için 6 test problemi yayınladı. Luck (2002), karmaşık trafik ve ulaştırma mühendisliği problemlerinin çözümünde kolektif arı istihbaratının olası uygulamalarını araştırmayı amaçlamıştır. Bu bağlamda TSP ve stokastik araç yönlendirme problemi (SVRP) incelenmiştir. TSP, her düğümden sadece bir kez geçen minimum mesafe devresini bulmayı amaçlayan NP-zor bir sorundur. Algoritma, arıların nektarı topladığı grafikteki düğümlerden birinde kovanın bulunmasıyla başlar, yani seyahat eden satıcı yolunun keşfedilmesi gereken grafikdir. Yapay arılar belirli bir zaman aralığında nektarı toplar ve kovanın konumu rastgele değiştirilir. Arılar nektarı yeni yerden toplamaya başlar ve tekrar kovanın yeri rastgele değiştirilir. Arama işlemindeki yineleme, kovanın konumunun bir değişikliğini temsil eder ve bir veya daha fazla uygulanabilir çözüm oluşturulduğunda yineleme sona erer. Yapay arılar, ayrık zamanla karakterize edilen bir ortamda yaşarlar ve sonuç olarak, her bir yineleme belirli sayıda aşamadan oluşur. Herhangi bir aşamada, arılar rastgele ziyaret edilecek düğümleri seçerler. Bu olasılık fonksiyonu ile, iki düğüm arasındaki mesafe ne kadar büyük olursa, bir arının bu bağlantıyı seçme olasılığı o kadar düşük olur. Mesafenin etkisi, arama işleminin başlangıcında daha düşüktür. Yineleme sayısı arttıkça, mesafenin etkisi de artar. Öte yandan, geçmişte belirli bağlantılar tarafından ziyaret edilen toplam arı sayısı arttıkça, gelecekte bu bağlantıyı seçme olasılığı da artar. Bu, kolonideki bireysel arılar arasındaki etkileşimi temsil eder. Bir aşamada arı belirli sayıda düğümü ziyaret eder, kısmi bir seyahat satıcısı turu oluşturur ve kovana geri döner. Kovanda arı bir karar verme sürecine katılır. Arı, yiyecek kaynağına dönmeden önce dans ederek yuva arkadaşlarını işe almadan yiyecek kaynağına yem vermeye veya yiyecek kaynağını terk edip etmemeye karar verir. Bir arının aynı kısmi turu kullanma olasılığı (veya onu terk etmesi) kısmi turun uzunluğuna bağlıdır. Arının

keşfettiği tur ne kadar uzun olursa, arı aynı tur boyunca tekrar uçuşma olasılığı o kadar küçük olur. Belli bir bağlantı boyunca nektar miktarının bağlantı uzunluğu ile ters orantılı olduğu not edilir. Herhangi bir aşamanın başlangıcında, eğer arı aynı kısmi gezici satıcı turunu kullanmazsa, arı dans alanına gider ve olasılık fonksiyonuna göre başka bir arıyı takip eder. Bu işlev, kısmi yolun toplam uzunluğuna ve bu rotayı tanıtan arı sayısına bağlıdır. Ayrıca, kovanın yerini değiştirmeden önce, mevcut iterasyonda arılar tarafından elde edilen çözümü geliştirmek için 2-opt ve 3-opt sezgisel algoritmalar uygulanır. Öte yandan doğada, tüm arılar eşzamanlı olarak toplamaya başlamamaktadır ve algoritmada, her yinelemenin başlangıcında tüm arıların kovanda olduğu ve sonraki her aşamada toplayıcı arı sayısının arttığı varsayılmaktadır. Algoritmanın performansı, 10 karşılaştırma problemi üzerinde test edildi. Deneysel sonuçlar, 100'den az düğümlü tüm durumlarda, Arı Sisteminin en uygun çözümü ürettiğini ve Arı Sisteminin en iyi çözümleri bulmak için gereken sürenin düşük olduğunu göstermiştir. Bu tezin ikinci bölümünde, Arı Sistemi bulanık mantıkla bütünleştirilmiştir ve bu yaklaşım Araç Yönlendirme Sorunlarına (VRP) uygulanmıştır. Prosedür ve sonuçlar, Lucic ve Teodorovic (2003b) 'de sunulmuştur. Lucic ve Teodorovic (2002, 2003a), Arı Sistemi üzerine ikinci ve üçüncü çalışmalarını Lucic 'ın (2002) 8 ve 10 TSP test problemlerine dayanarak yayınladılar.

Lucic ve Teodorovic (2003b), ilk olarak Lucic ve Teodorovic (2001) tarafından önerilen Arı Sistemi algoritmasını ve stokastik VRP için iyi çözümler elde etmek için bulanık bir mantık yaklaşımını birleştirdi. Önerilen yaklaşım iki adım içerir:

- 1) Arı Sistemi kullanarak VRP'yi TSP olarak çözme ve orijinal probleme sıklıkla mümkün olmayan bir çözüm elde etme.
- 2) Önceki adımda oluşturulan çözümü ve Wang-Mendel'in algoritması tarafından oluşturulan bulanık kural tabanını kullanarak bir aracın rotasını ne zaman bitireceğine ve bir sonraki aracın rotasına ne zaman başlayacağına karar verme.

Stokastik VRP, depo yerlerinin, servis edilecek düğümlerin ve araç kapasitesinin bilindiği ve sadece düğümlerdeki talebin yaklaşık olarak tahmin edildiği nakliye maliyetini en aza indirecek bir dizi yol bulmayı amaçlamaktadır. Düğümlerdeki talep

belirsizliđi nedeniyle, bir araç yetersiz kapasite nedeniyle oraya geldiđinde bir düđüme hizmet veremeyebilir. Bu gibi durumlarda, aracın depoya geri döndüğü, şimdiye kadar aldığı şeyi boşalttığı, arızası olduđu düđüme geri döndüğü ve planlanan rotanın geri kalanında hizmet vermeye devam ettiđi varsayılmaktadır. Sonuç olarak, düđümlerdeki talep rastgele bir deđişken olarak ele alınır ve gerçek talep deđeri sadece düđüme ziyaretten sonra bilinir. Geliştirilen model 10 TSP örneđi üzerinde test edilmiştir. Orijinal TSP problemlerini karşılık gelen VRP'lere dönüştürmek için, ilk düđüm depo olarak kabul edildi. Sonuçlar, Arı Sistemine dayalı sezgisel algoritma ile elde edilen en iyi çözüm ile karşılaştırıldı. Gelecekteki düđüm talep modelinin bilindiđi varsayımıyla, sonuçların en iyi çözüme çok yakın olduđu bulunmuştur.

Teodorovic ve Dell'Orco (2005), Lucic (2002) tarafından sunulan Arı Sisteminin geliştirilmesi olan Arı Kolonisi Optimizasyonu (BCO) meta-sezgisel yöntemini önermiştir. BCO, deterministik kombinatoriyal problemlerin yanı sıra belirsizlik ile karakterize edilen kombinatoriyal problemleri de çözebildi. Makalelerinin temel amacı, belirsizlikle karakterize edilen kombinatoriyal problemlerin çözümünde kolektif arı zekasının olası uygulamalarını araştırmaktır. Bu bağlamda, ajanların iletişim ve eylemlerinde bulanık mantığın yaklaşık akıl yürütme ve kurallarını kullandığı Bulanık Arı Sistemi (FBS) tanıtıldı. FBS algoritmasının performansı, tüm katılımcıların kat ettiđi toplam mesafeyi en aza indirerek, toplam gecikmeyi en aza indirerek veya araç kullanımını dengeleyerek araçların ve yolcuların yönlendirme ve zamanlamasını oluşturmayı amaçlayan bir sürüş eşleştirme sorunu üzerinde test edildi. Önerilen yaklaşımı destekleyebilecek hiçbir teorik sonuç yoktu. Ancak ön sonuçlar çok umut vericiydi.

Nakrani ve Tovey (2003) internet hizmetlerinin dinamik tahsisi için bir bal arısı algoritması önermişlerdir. Önerilen algoritmada, bir İnternet sunucusu kolonisindeki sunucular ve HTTP istek kuyrukları sırasıyla yem, arılar ve yiyecek kaynakları olarak modellenmiştir. Algoritmada, optimum tahsisi hesaplamak için tarihçe kayıtlarını hesaplayan bir algoritma ile olası tüm statik tahsis politikaları arasında en iyi çözümü hesaplayan algoritma karşılaştırıldı. Deneysel sonuçlar, algoritmanın statik veya açgözlü algoritmalarından daha iyi performans gösterdiğini

göstermiştir. Öte yandan, bazı düşük değişkenli erişim kalıpları için açgözlü algoritma ile daha iyi performans gösterdi.

Wedde ve diğ. (2004), BeeHive adlı bir telekomünikasyon ağına yönlendirme için bal arılarının dans dilinden ve yiyecek arama davranışlarından esinlenerek hataya dayanıklı, uyarlanabilir ve sağlam bir yönlendirme protokolü başlattı. Algoritmanın performansını değerlendirmek için Japon İnternet omurgasında test edildi ve AntNet, DGA ve OSPF ile karşılaştırıldı. Sonuçlar, BeeHive'in diğer algoritmalara kıyasla benzer veya daha iyi performans elde ettiğini gösterdi.

Bianco (2004), arıların geniş ölçekli navigasyon davranışlarından ilham alan büyük ölçekli hassas navigasyon için bir haritalama paradigması sundu. Arılar beslenirken çok uzun yolculuklar yaptılar, kilometrelerce seyahat ettiler, ancak aynı zamanda küçük kovanlarına döndüklerinde mükemmel bir hassasiyet elde ettiler. Test sonuçları, bu tür yeteneklerin oldukça iyi bir hassasiyet elde etmek için yeterli olduğunu göstermiştir.

Chong ve diğ. (2006) Nakrani ve Tovey'den (2004) esinlenen bal arıları yemleme modelini kullanan yeni bir yaklaşım sunarak atölye çizelgeleme problemini çözdüler. Atölye planlaması, belirli bir objektif işlevi optimize eden rakip kaynakların sıralı olarak tahsis edilmesiyle ilgilidir. Her makine yalnızca bir işi işleyebilir ve her iş aynı anda yalnızca bir makine tarafından işlenebilir. Algoritmanın performansı 82 atölye problemi örneğinde test edildi ve karınca kolonisi ve tabu arama algoritmaları ile karşılaştırıldı. Deneysel sonuçlar tabu aramasının çözüm kalitesi ve yürütme süresine göre diğer iki buluşsal yöntemden daha iyi performans sergilediğine rehberlik etti. Diğer yandan, arı algoritması karınca algoritmasından biraz daha iyi performans gösterdi ve her iki buluşsal yöntem için yürütme süresi yaklaşık olarak eşitti.

Drias ve diğ. (2005), Arılar Sürüsü Optimizasyonu (BSO) adında yeni bir akıllı yaklaşımı ortaya koydu; bu, özellikle en kolay erişim kaynaklarının nektarını her zaman en zenginleri toplayan gerçek arıların davranışlarından esinlenmiştir. Önerilen algoritma, NP-Complete olan maksimum ağırlıklı memnuniyetlik (MAX-W-SAT) problemine uyarlanmıştır. MAX-W-SAT problemi, bir dizi ağırlıklı cümle göz önüne alındığında, herhangi bir ödev tarafından karşılanabilecek maksimum

ağırlığı ister. Algoritmanın performansı Açgözlü Rastgele Uyarlamalı Arama (GRASP) ve Basit Memnuniyetlik (SSAT) ile karşılaştırıldı ve BSO'nun diğer evrimsel algoritmalarından daha iyi performans gösterdiği sonucuna varıldı.

Pham ve diğ. (2006b) önerilen Arılar Algoritması kontrol grafiği örüntü tanıma ve Öğrenim Vektörü Niceleme (LVQ) sinir ağını eğitmek için Pham ve ark. (2006a) tarafından kullanıma sunuldu. Bir LVQ ağıнын eğitimi, bir hata fonksiyonunun en aza indirilmesi olarak görülebilir. Hata fonksiyonu, bir dizi egzersiz modeli üzerinde gerçek çıkış ve ağıнын istenen çıkışı arasındaki toplam farkı tanımlar. Arılar Algoritması açısından, her arı belirli bir referans vektörleri kümesiyle bir LVQ ağını temsil eder. Algoritmanın amacı arıyı, hata fonksiyonunun en küçük değerini üreten referans vektörleri setiyle bulmaktır. Sorunun yüksek boyutluluğuna rağmen, algoritma hala standart LVQ eğitim algoritması tarafından üretilenden daha doğru sınıflandırıcılar yetiştirmeyi başardı.

Pham ve diğ. (2006c) Arılar Algoritmasını kontrol şeması örüntü tanıma ve Çok Katmanlı Perceptron (MLP) sinir ağını eğitmek için Pham ve ark. (2006a) tarafından sunuldu. MLP ağıнын eğitimi, bir hata işlevinin en aza indirilmesi olarak görülebilir. Hata fonksiyonu, bir dizi egzersiz modeli üzerinde gerçek çıkış ve ağıнын istenen çıkışı arasındaki toplam farkı tanımlar. Arılar Algoritması açısından, her arı belirli bir ağırlık vektörleri kümesine sahip bir MLP ağını temsil eder. Algoritmanın amacı arıyı, hata fonksiyonunun en küçük değerini üreten ağırlık vektörleri setiyle bulmaktır. Sorunun yüksek boyutluluğuna rağmen, algoritma geri yayılım algoritmasından daha doğru sınıflandırıcılar yetiştirmeyi başardı.

Pham ve diğ. (2006d), Arılar Algoritmasını ahşap kaplama levhalarındaki kusurların tanımlanması için sinir ağlarının optimizasyonu için Pham ve ark. (2006a) tarafından sunuldu. Sinir ağıнын ağırlıklarını optimize etmek için bir geri yayılım algoritması yerine Arılar Algoritması kullanılmıştır. Arılar Algoritması kullanılarak yapılan optimizasyon, her bir arının belirli bir ağırlık kümesiyle bir sinir ağını temsil ettiği ağ içindeki nöronlar arasındaki bağlantılara atanan ağırlıkların optimal değerlerini araştıran arıları içerir. Arılar Algoritmasının amacı, hata fonksiyonunun en küçük değerini üreten arıyı bulmaktır. Deneysel sonuçlar, Arılar Algoritmasının, geri yayılım yöntemiyle karşılaştırılabilir bir doğruluk elde edebildiğini

göstermektedir. Bununla birlikte, Arılar Algoritmasının çok daha hızlı olduğunu kanıtladı.

Quijano ve Passino (2007), kaynak tahsisi sorununu çözmek için bal arılarının yiyecek arama davranışına dayanan bir algoritma geliştirdiler. Önerilen modelin bileşenlerini oluşturmak için birincil kaynaklar şunlardır: dans gücü belirleme, dans eşiği, boşaltma alanı, dans pisti ve işe alım oranları, kâşif tahsisi ve işe alımla ilişkisi. Ayrıca, bal arısı sosyal yemleme algoritmasının dinamik çalışmasının temel özelliklerini vurgulamak için çok bölgeli sıcaklık kontrolü için dinamik kaynak tahsisi üzerine bir mühendislik uygulaması önerdiler.

Markovic ve diğ. (2007), Teodorovic ve Dell'orco (2005) tarafından tüm optik ağlarda Maks. Yönlendirme ve Dalga Boyu Ataması (Max-RWA) problemini çözmek için tanıtılan BCO algoritmasını kullanmıştır. Max-RWA problemi, belirli bir trafik talebi matrisi ve verilen dalga boyu sayısı için belirli bir optik ağda yerleşik ışık yollarının sayısını en üst düzeye çıkarmaktır. Önerilen BCO-RWA algoritması Avrupa Optik Ağı üzerinde test edildi ve sonuçlar LP gevşeme yaklaşımı ve tabu meta-sezgisel algoritma ile elde edilen sonuçlarla karşılaştırıldı. BCO-RWA algoritması, karşılaştırılan algoritmaların sonuçlarından her zaman daha iyi performans gösterdi ve makul hesaplama süresinde çok iyi çözümler üretebildi.

Abbass (2001a) bal arılarındaki evlilik sürecinden esinlenen ilk yeni arama algoritmasını sundu. Bir bal arısı kolonisi kraliçe, erkek arı, işçi (ler) ve damızlıklardan oluşur. Bu çalışmada, koloninin tek bir kraliçe ve tek bir işçi olduğu varsayılmıştır. Gerçek hayatta, çiftleşme uçuşu, kraliçelerin gerçekleştirdiği bir dansla başlar ve dronlar, kraliçeleri onlarla eşleşmek için takip eder. Her çiftleşmede sperm spermatekaya ulaşır ve orada koloninin genetik havuzunu oluşturmak için birikir. Her kraliçe döllenmiş yumurta bıraktığında, yumurtayı dölemek için spermatekste biriken spermlerin bir karışımını rastgele alır. Benzer şekilde, MBO algoritmasında, çiftleşme uçuşu, kraliçenin uzaydaki farklı durumlar arasında hareket ettiği ve her bir durumda olasılıkla karşılaşılan drone ile eşleştiği bir durum uzayında bir dizi geçiş olarak görselleştirilebilir. Kraliçe hala çiftleşme uçuşunun başlangıcındayken ve bu nedenle hızı yüksek olduğunda veya drone'nun zindeliği kraliçeler kadar iyi olduğunda çiftleşme olasılığı yüksektir. Algoritma kraliçenin genotipini rastgele başlatmakla başlar. Bundan sonra, kraliçenin işçiler tarafından

gerçekleştirilen genotipini geliřtirmek için bir buluşsal yöntem kullanılır. Daha sonra bir çiftleşme uçuşu yapılır. Her çiftleşme uçuşunda kraliçenin enerjisi ve hızı rastgele başlatılır. Kraliçe daha sonra hızına göre uzaydaki farklı durumlar (çözümler) arasında hareket eder ve drone ile eşleşir. Bir drone kraliçe ile başarılı bir şekilde çiftlenirse (drone olasılıksal karar kuralını geçer), spermleri kraliçenin spermatekine (kısmi çözümleri listesi) eklenir ve kraliçenin hızı ve enerjisi azalır. Kraliçe çiftleşme uçuşunu bitirdikten sonra yuvaya geri döner, rastgele bir sperm seçer, çaprazlama ve mutasyon yapar. Daha sonra işçi, sonuçtaki yavruları iyileřtirmek için kullanılır ve işçi sayısı, programda kodlanan buluşsal yöntemlerin sayısını temsil eder. Daha sonra, ikincisi öncekinden daha iyi ise, kraliçe en uygun kuluçka ile deęiřtirilir. Kalan kuluçkalar öldürölür ve yeni bir çiftleşme uçuşu başlar. MBO algoritmasının kullanıcı tanımlı üç parametresi vardır: tek bir çiftleşme uçuşunda maksimum çiftleşme sayısını temsil eden kraliçenin spermateka boyutu, kraliçe tarafından doęacak kuluçka sayısı ve derinlięi gösteren kuluçka bakımına ayrılan yerel arama süresidir. Genel bir kısıtlama memnuniyeti sorunu (CSP), bu deęişkenlerin alanları üzerinde bir dizi kısıtlamayı karşılayan bir deęişken grubuna atama bulma sorunudur. Memnuniyet problemleri (SAT), her deęişkenin alanının doęru veya yanlış olduęu özel bir CSP vakasıdır. Ayrıca, 3-SAT, her kısıtlamanın üç deęişken içerdiięi özel bir SAT durumudur. MBO algoritması yüz farklı 3-SAT problemine uygulandı ve deney sonuçları algoritmanın çok başarılı olduęunu gösterdi. İşçilerin kullandıęı buluşsal yöntemler Açęözlü SAT (GSAT) ve rastgele yürüyüşü. Deneysel çalışmalarda GSAT, random walk, GSAT ile MBO ve random walk ile MBO karşılaştırıldı ve MBOGSAT dięer üçü arasında en iyi performansı gösterdi.

Abbass (2001b), ilk olarak Abbass (2001a) tarafından önerilen ve koloninin birden fazla işçi içeren tek bir kraliçe içerdiięi MBO algoritmasının bir varyasyonunu sundu. İşçiler için altı farklı sezgisel tarama kullanıldı: GSAT, rastgele yürüme, rastgele çevirme, rastgele yeni, 1 noktalı ve 2 noktalı geçiş. Algoritma yüz sert 3-SAT probleminden oluşan bir grup üzerinde test edilmiştir. En iyi sonuçlar, en küçük koloni boyutu ve ortalama spermateka boyutu ile ortaya çıkmıştır. Öte yandan, en uygun işçi GSAT idi ve bunu rastgele bir yürüyüş izledi. Ayrıca MBO'nun Ateşböceęi Sürüsü Optimizasyonu (GSO)'ndan daha yüksek performans

göstermesine rağmen, MBO'nun tek başına GSAT'tan daha iyi performans gösterdiği gösterilmiştir.

Abbass (2001c), çalışmaların devamı (Abbass, 2001a) olarak bal arılarının evlilik davranışlarını tekrar analiz etmiştir. Bu çalışmalar arasındaki fark, kraliçe ve işçi sayısıdır. Abbass (2001c), bir grup işçiye ek olarak birden fazla kraliçeye sahip bal arısı kolonisi olarak kabul edildi, Abbass (2001a) kolonisinde sadece bir kraliçe ve bir işçi vardı. Bu makalede, MBO algoritması 50 değişken ve 215 kısıtlama içeren elli farklı 3-SAT problemine uygulanmıştır. Deneysel sonuçlar, en büyük spermateka boyutunun, ortalama koloni boyutunun ve en az kraliçenin en iyi performansı verdiği sonucuna vardı. Öte yandan, algoritma SAT için son teknoloji algoritmalarından biri olan WalkSAT ile karşılaştırıldı ve MBO algoritması WalkSAT'den daha iyi performans gösterdi.

Teo ve Abbass (2001), MBO algoritmasının Abbass (2001a) ve Abbass (2001c)'nin bir uzantısı olarak düşünülebilecek bir modifikasyonunu sundu. Bu modifikasyonun amacı, arama sürecini daha optimum bir çözüm alanına doğru yönlendirmek için yörünge kabul kararı sırasında daha geleneksel bir tavlama yaklaşımı kullanmaktır. Yeni yörüngeler, yalnızca yörüngenin kraliçenin genotipinden daha uygun olması durumunda daha uygun bir yörünge ise çiftleşme için potansiyel bir drone olarak kabul edildi. Aksi takdirde, aramayı daha az optimum çözüm alanına götüren bir yörünge ise, o zaman sadece yeni tavlama fonksiyonuna tabi olarak kabul edilir. Başka bir deyişle, bir kraliçenin uçuşu sırasında yaratılan tüm yörüngeleri orijinal MBO algoritmasında olduğu gibi kabul etmek yerine, yeni bir yörünge, ancak daha uygun bir çözüm uzayına geçiş olması durumunda kabul edilir. Aksi takdirde, algoritma, kraliçenin uygunluğunun bir fonksiyonuna göre, olasılıkla daha az optimal bir çözüm uzayına geçişi kabul edecektir. Öte yandan, işçilerin kuluçkalarını iyileştirmek için beş farklı buluşsal yöntem kullanıldı: GSAT, rasgele yürüyüş, olasılık açgözlü, tek noktalı geçit ve Walk SAT. Abbass (2001a)'da olduğu gibi, Teo ve Abbass da sadece bir kraliçe ile bal arısı kolonisini düşündüler. Deneysel çalışmalar üç şekilde gerçekleştirildi: MBO'suz tek başına çalışan beş farklı sezgisel yöntemin her birini test etmek, orijinal MBO ve modifiye edilmiş MBO ile her sezgiselliğin performansını test etmek ve son olarak, önerilen beş farklı sezgisel tarama yöntemini kullanarak orijinal

MBO'ya karşı önerilen algoritmayı test etmek. Test problemleri için, her biri 1075 kısıtlama ve 250 değişken içeren on farklı 3-SAT problemi üretildi. Sezgisel performans, ilk deney grubu için şu sırayla sonuçlandı: WalkSAT, GSAT, rasgele yürüyüş, olasılıklı açgözlü ve tek noktalı geçit. İkinci deney grubunda, çiftleşme uçuşu işlemi sırasında kullanılan hem orijinal hem de önerilen tavlama fonksiyonları, tüm buluşsal yöntemlerle benzer şekilde etkiliydi. Bununla birlikte, MBO'nun WalkSAT ile çözüm bulmadaki etkinliği, MBO'nun önerilen sürümü orijinal versiyondan daha fazla çözüm bulduğu için yeni tavlama fonksiyonu ile biraz iyileştirildi. Son olarak üçüncü deney grubunda her iki tavlama stratejisi de yine benzer şekilde etkiliydi.

Teo ve Abbass (2003), Teo ve Abbass (2001)'e dayanan MBO algoritmasında başka bir değişiklik önerdi. Hem Abbass (2001a) hem de Teo ve Abbass (2001)'de, tavlama işlevi, kraliçenin uygunluğunu, yumurtlama veya çiftleşme aşamasında dronun uzayında bir geçişi kabul etmek / reddetmek için temel olarak kullanmıştır. Geleneksel simüle edilmiş tavlama yaklaşımında, geçiş için temel olarak önceki durum kullanılmıştır. Dahası, biyolojik bir bakış açısıyla, dronun yaratılışı, ilişkili olmalarına rağmen, genellikle başka bir koloniden geldikleri için kraliçeden bağımsızdır. Bu nedenle, dronun kendi uygunluğuna dayalı bir geçişi kabul etmek daha doğaldır. Sonuç olarak, makalelerinin amacı, dron havuzunu belirlemek için temel olarak tamamen geleneksel bir tavlama yaklaşımını test etmektir. Değiştirilmiş algoritmanın performansı on farklı 3-SAT sorunu üzerinde test edildi ve MBO'nun önceki sürümleriyle karşılaştırıldı. Tek başına çalışırken tüm sezgisel tarama tek bir çözüm bile bulamamışken, MBO ile birleştirildiğinde performansları önemli ölçüde iyileştirilmiştir. Öte yandan, MBO'nun önerilen sürümü önceki çalışmalara egemen oldu ve önceki sürümlerin yapamadığı sorunlara çözüm bulabildi.

Bozorg Haddad ve Afshar (2004), Abbass (2001c) çalışmasına dayanan MBO algoritmasından yararlandı ve su kaynakları yönetimi sorunlarına bir uygulama yaptılar. Algoritma, tek bir rezervuarın optimum yönetimi için iyi çözümler bulmak üzere modellenmiştir. Sonuçlar benzer sezgisel yöntemlerle ve küresel optimal sonuçlarla çok iyi karşılaştırıldı.

Bozorg Haddad ve diğ. (2006), oldukça doğrusal olmayan kısıtlanmış ve kısıtlanmamış gerçek değerli matematiksel modelleri çözmek için Abbass (2001a, 2001c) Bal Arısı Çiftleşme Optimizasyonu (HBMO) algoritmasını önerdi. HBMO'nun performansı, birkaç kısıtlı ve kısıtsız matematiksel optimizasyon fonksiyonu üzerinde test edildi ve genetik algoritma ile elde edilen sonuçlarla karşılaştırıldı. Genetik algoritma ve HBMO algoritmasından elde edilen sonuçlar, HBMO çözümünde küçük bir iyileşme ile iyi yakınma sağladı. Ayrıca, model uygulamasını ve performansını göstermek için, HBMO algoritması tek bir rezervuar için optimum çalışma politikası geliştirmek için de kullanılmıştır. HBMO yine önemli ölçüde daha iyi bir çözüm üretti.

Chang (2006), birleştirici optimizasyon problemlerini ve stokastik ardışık karar verme problemlerini çözmek için teorik bir bakış açısıyla MBO yaklaşımının kabiliyetinin ilk gösterimini yaptı. Çalışma ilk olarak stokastik olmayan kombinyonel optimizasyon problemlerini çözmek için MBO algoritması ile ilgiliydi ve MBO'nun küresel optimum değere yakınlaşma yeteneğine sahip olduğunu kanıtladı. MBO daha sonra Markov karar süreçleri (MDP'ler) olarak da bilinen sonsuz ufukta iskonto edilmiş maliyet problemlerini stokastik dinamik programlama (SDP) ile çözmek için "Bal Arıları Politika İterasyonu" (HBPI) adlı bir algoritmaya uyarlandı. Chang (2006), MBO'nun, benzetilmiş tavlama ve genetik algoritmanın melez şeması olarak kabul edilebileceğini belirtiyor. Simüle edilmiş tavlama, kraliçenin spermatekste potansiyel drone spermlerini elde etmek için çiftleşme uçuşuna karşılık gelir ve genetik algoritma, kuluçka üretimine ve bazı farklılıklar ile iyileştirme adımına karşılık gelir. Afshar ve ark. (2007) sürekli optimizasyon problemleri için HBMO algoritmasının geliştirilmiş bir versiyonunu ve bunun doğrusal olmayan kısıtlamalı sürekli tek rezervuar problemine uygulanmasını sundu. LINGO 8.0 NLP çözücünden elde edilen global optimum değerlerle karşılaştırıldığında algoritmanın optimum seviyeye yaklaşmasının çok hızlı olduğu gözlenmiştir.

Fathian ve diğ. (2007), birçok alanda kullanılan çekici veri madenciliği tekniklerinden biri olan kümeleme için HBMO algoritmasının bir uygulamasını sundu. Algoritmanın kümelemedeki performansını değerlendirmek için, birkaç gerçek veri kümesi üzerinde test edildi ve ACO algoritması, simüle edilmiş tavlama

yaklaşımı, genetik algoritmalar ve tabu arama yaklaşımı gibi birkaç tipik stokastik algoritma ile karşılaştırıldı. Sonuçlar, önerilen HBMO yaklaşımının, kümelenme problemlerine optimal veya neredeyse optimal çözümler bulmak için uygun ve verimli bir buluşsal yöntem olarak kabul edilebildiğini gösterdi.

Koudil ve diğ. (2007), ilk olarak Abbass (2001) tarafından sunulan ve kodlamadaki entegre bölümlenme / çizelgeleme sorunlarını çözmek için MBO algoritmasını uyarladı. Önerilen yaklaşım bir kıyaslama problemi üzerinde test edilmiş ve sonuçlar genetik algoritma ile karşılaştırılmıştır. Test sonuçları MBO'nun çözüm kalitesi açısından iyi sonuçlar verdiğini ve yürütme süreleri bakımından genetik algoritmadan daha iyi sonuçlar verdiğini göstermiştir. Benatchba ve diğ. (2005), Abbass (2001a, 2001b, 2001c) tarafından ilk kez Max-SAT problemi olarak ifade edilen bir veri madenciliği problemini çözmek için sunulan MBO algoritmasını kullanmıştır. MBO için, işçilerin kuluçkalarını iyileştirmek için dört farklı buluşsal yöntem kullanıldı: Bağlantı Durumu (LS), GSAT, Heterojen Simüle Tavlama Ekibi (HSAT) ve Genel En iyi ve En kötü Memnuniyeti (GWSAT). Bir karşılaştırma ölçütü olarak kullanılan eğitim seti, ana safra kanalının laparotomisinin varlığına veya olmamasına ilişkin en açık belirtileri analiz etmeyi amaçlayan tıbbi bir setten çıkarılmıştır. MBO ile elde edilen en iyi sonuç, GSAT'ı işçi olarak kullanarak %96 memnuniyeti olan çözümdü.

Sung (2003), genetik algoritmaların kapasitesini arttırmak için kraliçe arı gelişimi önerdi. Ana arı evrimi algoritmasında kraliçe arı, rulet tekerleği seçimi gibi bilinen seçim algoritmaları yerine farklı bir seçim algoritması ile ebeveyn olarak seçilen diğer arılarla melezleşir. Bu prosedür genetik algoritmaların sömürülmesini arttırırken, diğer taraftan, erken yakınsamaya düşme olasılığını arttırır. Bu olasılığı azaltmak için bazı bireyler, normal evrimde olduğu gibi küçük mutasyon olasılığı olan tüm bireyleri mutasyona uğratmak yerine güçlü bir şekilde mutasyona uğramıştır. Önerilen algoritma bir kombinatoriyel ve iki tipik fonksiyon optimizasyon problemi ile test edilmiştir. Deney sonuçları, önerilen algoritmanın genetik algoritmaların küresel optimum seviyeye hızla yaklaşmasını sağladığını göstermiştir.

Qin ve diğ. (2004) Sung (2003) tarafından önerilen kraliçe arı evrimini ekonomik güç dağıtımını sorununa (EPD) uygulamıştır. EPD problemi, genel maliyet oranını en aza indirmek ve bir güç sisteminin yük talebini aynı anda karşılamak ve

doğrusal olarak kısıtlanmamış karmaşık optimizasyon problemi olarak formüle edilmiştir. Sayısal sonuçlar, önerilen algoritmanın geleneksel genetik algoritmadan daha hızlı ve daha sağlam olduğunu göstermiştir. Kara (2004), genetik algoritmanın performansını artırmak için Bee Crossover adı verilen yeni bir geçit türü önerdi. Arı kraliçesi, diğer erkek arılarla cinsel ilişkilere sahiptir ve benzer şekilde, belirli bir kromozom, çapraz geçişin ilk ebeveyni için arı kraliçesi olarak kabul edilebilir ve diğer ebeveyni kolonideki kalan kromozomlardan biridir. Yazar üç farklı geçit türü önerdi. Birinci tipte, en iyi kondisyon değerine sahip kromozom sabit ebeveyndir ve kalan tüm kromozomlar bu sabit ebeveyn ile her nesilde en az bir kez çaprazlanır. İkinci tipte, en kötü kondisyon değerine sahip kromozom sabit bir ebeveyndir ve kalan prosedür birinci tipte aynıdır. Üçüncü tipte, popülasyon kondisyon değerlerine göre sıralanır ve ilk nesildeki sabit ebeveyn bu listedeki ilk kromozom tarafından belirlenir. İkinci kuşakta, sabit ebeveyn listedeki ikinci kromozomdur ve bu böyle devam eder. Bu geçit tiplerinin performansı, tek tip geçit ile karşılaştırıldı. Sonuçlar, çoğu zaman, bal arısı geçitlerinin daha az yinelemeyle sonuçlandığını ve en kötü sonuçların tek tip geçit ile elde edildiğini göstermiştir. Öte yandan, tek tip geçit, küçük bir süre içinde nüfusun çeşitliliğini kaybetti, bal arısı geçitleri ise daha geniş zaman aralıklarında nüfus çeşitliliğini kaybetti.

Azeem ve Saad (2004), ilk olarak Sung (2003) tarafından sunulan değiştirilmiş bir kraliçe arı evrimi önermişlerdir. Önerilen algoritmada, herhangi bir çözümün ana arının uygunluğuna çok yakın veya daha yüksek olması durumunda, bu çözüm yeni bir havuza, orijinal algoritmanın tek bir havuzla sınırlı olduğu bir ana arı olarak tanımlanır. Orijinal ve önerilen algoritmalar arasındaki bir diğer fark, geçiş operatörüdür. Orijinal algoritma, her genin bir olasılıkla çaprazlandığı tek tip bir çaprazlama kullanır. Öte yandan, önerilen algoritma, popülasyondaki test modellerinin benzerliğine göre her bir gene ağırlıkların verildiği ağırlıklı tek tip geçit kullanır. Bu tür bir geçişle, genetik algoritma daha fazla yeni durum alanı arayacaktır. Algoritma, iki karmaşık doğrusal olmayan örnek üzerinde Bulanık Bilgi Tabanı Denetleyicisi (FKBC) için ölçeklendirme faktörünün ayarlanması için test edilmiştir. Deneyler, sistem modelinin veya parametrelerinin elde edilmesinin zor olduğu karmaşık durumlarda FKBC'nin geleneksel kontrol algoritmalarından daha

üstün sonuçlar verdiğini gösterdi. Ayrıca, sonuçlar rulet çarkının ebeveyn seçimi ile karşılaştırılmış ve elde edilen sonuçlar cesaret vericidir.

Bu çalışmanın ilerleyen bölümlerinde, genelleştirilmiş görev problemine doğadan ilham alan arı temelli algoritmanın (Yapay Arı Kolonisi, ABC olarak adlandırdığımız) ilk uygulaması sunulmaktadır.

2.5. Parçacık Sürü Algoritması ile İş Atama Problemi

Görünüşte karmaşık görünen davranışlar üretmek için basit yollarla birlikte hareket eden birçok özerk parçacık kullanma fikrinin başlangıçta bilgisayar animasyonlarında görüntü oluşturma sorununu çözdüğü düşünülmüştür (Reeves WT., 1983). Bir parçacık sistemi stokastik olarak bir dizi hareket noktası oluşturur. Her parçacığa bir başlangıç hızı vektörü atanır. Ayrıca renk, doku ve sınırlı ömür gibi ek özelliklere de sahip olabilir. İteratif olarak, hız vektörleri bazı rastgele faktörlerle ayarlanır. Bilgisayar grafikleri ve bilgisayar oyunlarında, parçacık sistemleri her yerde bulunur ve yangın, duman, bulutlar, silah sesleri, su, kumaş, patlamalar, büyü, aydınlatma, elektrik, akın ve diğerleri gibi animasyonlu efektler üretmek için fiili yöntemdir. Bunlar, uzayda davranışlarını ve görünümelerini yönlendiren bir dizi kuralla tanımlanır; örneğin, hız, renk, boyut, şekil, şeffaflık ve dönüş. Bu, matematik ve programlamadan yeni karmaşık efektlerin oluşturulmasını birbirinden ayırır. Günümüzde parçacık sistemleri küresel optimizasyonda daha da popülerdir (Clerc M, Kennedy J, 2002).

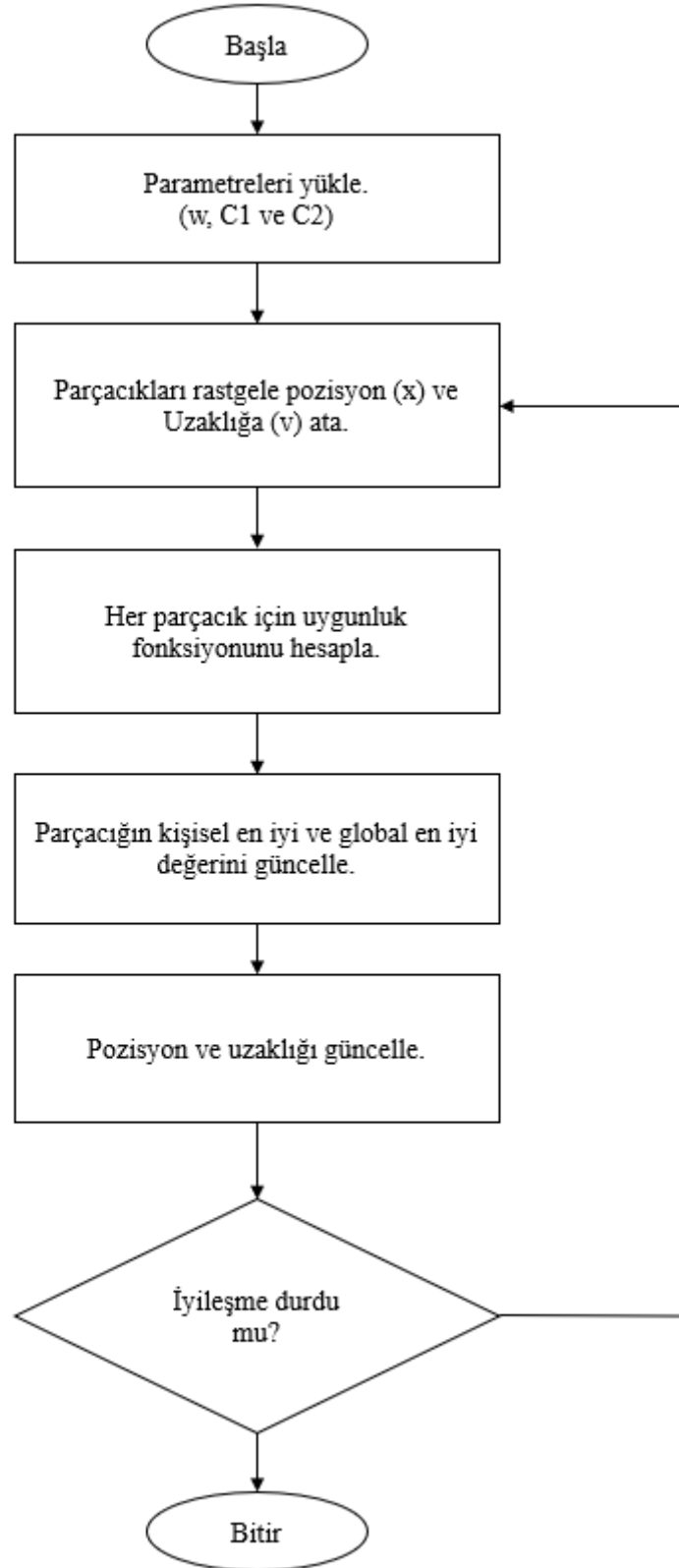
PSO, senkron kuş sürüsü, balık eğitimi ve arıların uğultu çalışmalarından kaynaklanmaktadır (Kennedy J, Eberhart RC, 2001, Kennedy J, Eberhart R, 1995). Partikül adı verilen bireylerin popülasyonlarını veya sürülerini geliştirir. Parçacıklar sürülerde sosyal davranışlar altında çalışır (Loengarov A, Tereshko V, 2006). PSO, her bir parçacığın hareket eden vektörünü kişisel en iyi (bilişsel yön) ve her bir iterasyondaki tüm kümedeki parçacıkların küresel en iyi (sosyal yönü) konumlarına göre ayarlayarak küresel en iyi çözümü bulur (Shi Y, Eberhart RC, 1998).

Karınca kolonisi algoritmaları ve EA'ları ile karşılaştırıldığında, PSO sadece ilkel matematik operatörleri, daha az hesaplamalı defter tutma ve genellikle daha az kod satırı gerektirir ve bu nedenle hem bellek gereksinimleri hem de hız açısından

hesaplama açısından ucuzdur. PSO, uygulama basitliđi ve makul bir řekilde kabul edilebilir bir çözüme hızlı bir řekilde yaklaşma yeteneđi nedeniyle popülerdir.

Bu problemin çözümünde kullanılacak Parçacık Sürü optimizasyon algoritmasının anlaşılmasını kolaylařtırmak için, Şekil 2.6 'deki gibi bir akış şeması sağlanmıştır. Bu akışa göre, w , c_1 ve c_2 rastgele deđerleri atanır. Belirlenen her bir parçacık çözüm uzayına dağıtılır. Bu parçacıklar için uygunluk deđeri hesaplanır. Uygunluk deđeri lokal en iyi deđerinden iyiyse lokal en iyi deđer güncellenir. Her parçacığın lokal en iyi deđer global en iyi uygunluk deđer ile karşılaştırılır. Eđer daha iyiyse güncellenir. Parçacığın pozisyon ve uzaklık deđerleri hesaplanır. Bu döngü uygunluk deđerinde iyileşme gerçekleşmeyene kadar tekrarlanır (Şekil 2.6).





Şekil 2.6. PSO akış şeması

2.5.1. PSO Varyantlarının Dağılım Metotları

Temel PSO'da r_1 ve r_2 katsayıları için rasgele sayılar üretmek üzere Gauss veya Cauchy olasılık dağılımları kullanılır. Gauss veya Cauchy olasılık dağılımlarının kullanılması, ince ayar yapma ve hatta yerel optimadan kaçma yeteneğini artırabilir. Coelho LS, Krohling RA (2003)'te, hız güncelleme denklemi için rasgele sayılar üretmek üzere kesilmiş Gauss ve Cauchy olasılık dağılımları kullanılmıştır. Secrest BR, Lamont GB (2003)'de, sürünün parçacıklarını g_{best} ve l_{best} 'ten bir Gauss mesafesine taşımak için bir kural kullanılır. Gaussian mutasyon operatörü Higashi N., Iba H. (2003) veya Cauchy mutasyon operatörü Esquivel SC, Coello CAC (2003) olarak hız güncelleme denklemine ek bir pertürbasyon terimi eklenebilir. Basitleştirilmiş bir PSO algoritmasında Gauss dağılımı da kullanılır (Kennedy J, 2003). Hız denklemi, c_1 ve c_2 sabitlerinin sıfır ortalama ve birim standart sapma ile Gauss dağılımının mutlak değeri kullanılarak üretildiği Gauss dağılımına göre güncellenebilir (Krohling RA, 2004).

Hakli H, Uguz H. (2014)'de PSO, yerel minimalden kurtulmak ve küresel arama yeteneğini geliştirmek için Levy uçuşları ile birleştirilmiştir. Levy uçuşu, Levy dağılımını kullanarak basamak boyutunu belirleyen rastgele bir yürüyüştür. Parçacıklar tarafından yapılacak uzun sıçramalar sayesinde arama alanında daha verimli bir arama yapılır. Her bir parçacık için bir sınır değer tanımlanır ve parçacıklar mevcut yinelemenin sonunda kendi kendine çözümleri geliştiremezse, bu sınır artar. Belirlenen sınır değer bir parçacık tarafından aşılsa, parçacık Levy uçuş yöntemi ile arama alanına yeniden dağıtılır.

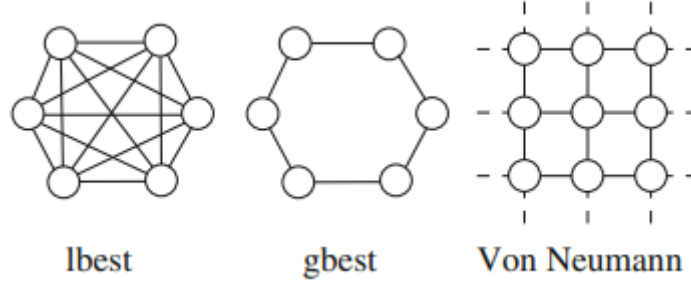
2.5.2. Farklı Komşuluk Topolojilerini Kullanan PSO Varyantları

PSO'nun temel özelliği komşulukları arasında sosyal bilgi paylaşımıdır. Tipik komşuluk topolojileri, Şekil 2.7'de gösterildiği gibi, Von-Neumann komşuluğu, g_{best} ve l_{best} 'tir (Şekil 2.7). En basit komşu yapı halka yapısı olabilir.

Temel PSO, komşuluğun tüm sürüden oluştuğu g_{best} topolojisini kullanır, yani tüm parçacıklar küresel olarak bulunan en iyi çözüm bilgisine sahiptir. Her parçacık diğer her parçacığın komşusudur.

En iyi komşulukta halka kafes topolojisi vardır: her parçacık kendisinden ve iki veya daha fazla yakın komşusundan oluşan bir komşuluk üretir. Komşular, nesnel

işlev değerleri veya konumlar açısından üretici parçacığa yakın olmayabilir, bunun yerine bitişik endeksleri tarafından seçilirler.



Şekil 2.7. Komşuluk yapısı türleri

Von-Neumann komşuluğunda, her bir parçacığın dört tarafına (torus) sarılmış iki boyutlu bir kafes üzerinde dört komşusu vardır ve bir parçacık dört komşusunun ortasındadır. Olası parçacık sayısı dördür.

Çeşitli sosyal ağ yapıları üzerinde yapılan testlere dayanarak, küçük bir komşuluğa sahip PSO, karmaşık problemlerde daha iyi performans gösterirken büyük bir komşuluğa sahip PSO, basit problemlerde daha iyi performans gösterecektir (Kennedy J., 1999). Von-Neumann komşuluk topolojisi g_{best} ve l_{best} 'ten daha iyi performans gösterir (Kennedy J, Mendes R., 2002).

Erken yakınsamayı önlemek için, tamamen bilgilendirilmiş PSO'da bir parçacık hızı güncellemek için tüm topolojik komşularından gelen bilgileri kullanır. (Mendes R, Kennedy J, 2004). Her parçacığın komşuları üzerindeki etkisi, fitness değerine ve komşuluk boyutuna göre ağırlıklandırılır. Bu şema temel PSO'dan daha iyi performans gösterir. Daralma faktörü, tam olarak bilgilendirilmiş PSO'da kabul edilir, değer a bir parçacığın tüm komşuları arasında eşit olarak dağıtılır.

Birleşik PSO, yerel varyantın sömürücü davranışını ve yerel bir komşuluk varyantının sömürücü doğasını kullanacak şekilde daraltılmış algoritmayı değiştirerek elde edilir (Parsopoulos KE, 2004). İki hız güncellemesi başlangıçta hesaplanır ve daha sonra geçerli konuma uygulanan birleşik hız güncellemesi oluşturmak için doğrusal olarak birleştirilir.

L_{best} topoloji, arama alanını keşfetmek için daha iyidir, g_{best} daha hızlı birleşir. Değişken komşuluk operatörü aramaya en iyi halka kafesi ile başlar ve nüfus tamamen bağlanana kadar komşuluk boyutunu yavaşça artırır (Suganthan PN, 1999).

Hiyerarşik PSO'da parçacıklar, komşuluk yapısını tanımlamak için dinamik bir hiyerarşide düzenlenir (Janson S, Middendorf M., 2005). Zararlılarının kalitesine bağlı olarak parçacıklar hiyerarşide yukarı veya aşağı hareket eder. Yüksek hiyerarşideki iyi bir parçacığın sürü üzerinde daha büyük bir etkisi vardır. Hiyerarşinin şekli dinamik olarak uyarlanabilir. Tek tek parçacıklara hiyerarşideki seviyelerine göre farklı davranışlar da atanabilir.

2.5.3. Parçacık Temsil Prosedürleri

PSO'daki partikül pozisyonu ile iş atama problemindeki çizelgeleme çözümü arasındaki eşlemeyi oluşturmak için bir partikül temsil prosedürü kullanılır. Kilit nokta, PSO'daki her parçacığın belirli bir sorunun iş atama problemindeki bir programı gösterebilmesi için uygun bir yöntem bulmaktır. Aşağıda örnek 3 prosedür anlatılmaktadır:

Çalışma ve Partikül Pozisyon Dizisi (OPPS), parçacığı bir çizelge olarak temsil etmek için Liu Z (2007) tarafından önerilmiştir. Bu teknik, parçacığın rastgele sıralı sayılardaki konumunun, konum sekansına (artan sırada) bağlanmasıyla başlar. Parçacığın konumunu artan veya azalan bir düzende sıralarsak, konum sırası da buna göre değişir. Bu yeni pozisyon dizisindeki her pozisyon daha sonra her makineye iş atama probleminin öncelik kısıtlamalarına göre atanacaktır.

Rastgele Tuşların Temsili, Pongchairerks P. ve Kachitvichyanukul V. (2009) tarafından tanıtıldı. Bu teknik, parçacığın konumunu artan düzende sıralamakla başlar. Parçacığın bu yeni sırası iş dizisine bağlıdır. Parçacığın konumu, iş sırasının da buna göre değişeceği orijinal sıraya göre sıralanır. Bu yeni iş dizisindeki her iş daha sonra her makineye iş atama probleminin öncelik kısıtlamalarına göre atanacaktır.

Rastgele Anahtar Kodlama Şeması, 2010 yılında Lin, Horng, Kao, Chen, Run, Chen, Lai ve Kuo (2010) tarafından tanıtıldı. Bu teknik, parçacığın konumunu artan düzende sıralamakla başlar. Parçacığın konumunun bu yeni sırası, konum sırası ile bağlanır. Parçacığın konumu, konum sırasının da buna göre değişeceği orijinal sıraya göre sıralanır. Bu yeni pozisyon dizisindeki her pozisyon daha sonra her makineye iş atama probleminin öncelik kısıtlamalarına göre atanacaktır.

2.5.4. PSO Avantaj ve Dezavantajları

Avantajları:

1. PSO algoritması, türev içermeyen bir algoritmadır.
2. Uygulaması kolaydır, bu yüzden hem bilimsel araştırmada ve mühendislik problemleri için uygulanabilir.
3. Sınırlı sayıda parametreye ve parametrelerin çözümleri diğer optimizasyon tekniklerine kıyasla küçüktür.
4. PSO algoritmasında hesaplama çok basittir.
5. Yakınsama ve optimumluğu sağlayan bazı teknikler vardır ve sorunun değeri kısa sürede kolayca hesaplanır.
6. PSO, bir dizi başlangıç noktasına diğer optimizasyon tekniklerinden daha az bağımlıdır.
7. Kavramsal olarak çok basittir.

Dezavantajları:

1. PSO algoritması, hızının ve yönünün düzenlenmesini bozan kısmi iyimserlikten mustarıptir.
2. Koordinat dışı sistemle ilgili sorunlar (örneğin, enerji alanında) çıkar.

2.5.5. Parçacık Sürü Optimizasyonunda Yapılan Çalışmaların Gözden Geçirilmesi ve Sınıflandırılması

Standart PSO'da k komşularına sahip bir parçacık, bir etki kaynağı olmayı seçer ve diğerlerini yok sayar. Mendes, partiküllerin komşularıyla etkileşme biçimini revize etti (Kennedy ve Mendes, 2002). Geleneksel algoritmada her bir parçacık kendi önceki performansından ve komşuluğunda bulunan en iyi başarıdan etkilenirken, Mendes'in tamamen bilgilendirilmiş parçacık sürüsünde (FIPS), parçacık tüm komşularından etkilenir, bazen hiçbir etkisinden etkilenmez.

Silva ve diğ. (2002), bir yırtıcı parçacığın sürünün en iyi parçacığına çekildiği, ancak sürünün geri kalanını püskürttüğü ve arama alanının diğer alanlarını dağıtmaya ve keşfetmeye zorlayan bir Predator-Prey PSO'yu tanıttı. Bu yaklaşım, yüksek boyutlu test problemleri kullanılarak test edilir ve PSO'nun performansını çeşitli parametre ayarlarıyla iyileştirdiği gösterilmiştir.

Peram ve diğ. (2003) ve Veeramachaneni ve diğ. (2003), FIPS gibi komşuların etkilerini birleştiren Uygunluk Mesafe Oranı Parçacık Sürü Optimizasyonu (FDRPSO) adında yeni bir PSO başlattı (Mendes ve ark. 2002). Önerilen yeni algoritma, her bir parçacığı şimdiye kadar herhangi bir parçacık tarafından keşfedilen en iyi konuma doğru çekmek yerine parçacıkları daha yüksek uygunluk parçacıklarına doğru hareket ettirir. Bu, parçacık konumunun her bir bileşeninin değiştirilmesi gereken yönü belirlemek için nispi uygunluk oranı ve diğer parçacıkların mesafesi kullanılarak gerçekleştirilir. FDRPSO'nun orijinal PSO algoritmasından ve bazı varyantlarından önemli ölçüde daha iyi performans gösterdiği gösterilmiştir. Parçacıkların evriminin ampirik bir incelemesi, algoritmanın yakınsamasının PSO'dan farklı olarak parçacık evriminin erken bir aşamasında gerçekleşmediğini göstermektedir. Erken yakınsamadan kaçınmak, FDRPSO'nun zor çok modlu optimizasyon problemlerinde global optimum aramasına devam etmesini sağlar.

PSO arama sürecinin durgunluğunu önlemek için Van den Bergh, (2006), küresel en iyi parçacık için farklı bir hız güncelleme denklemi kullanan Garantili Yakınsama PSO'sunu (GCPSO) tanıttı. Bu denklem, parçacığın, küresel en iyi şekilde durmamada parçacığın başarısına uyum sağlayan bir ölçeklendirme faktörü tarafından tanımlanan bir yarıçap içinde küresel en iyi etrafında rastgele bir arama yapmasına neden olur. Tek modlu problemlerde GCPSO, modifiye edilmiş PSO'dan daha iyi performans gösterdi. Ancak çok modlu ortamlarda kullanıldığında benzer sonuçlar verdi.

Qi Kang, Lei Wang ve Qi-di Wu ve diğ. (2006) FAOPSO (Parçacık Sürüsü Optimizasyonunun Bulanık Uyarlamalı Optimizasyon Stratejisi) algoritmasını sunmuştur. Önerilen yaklaşım, standart literatürdeki fonksiyon optimizasyonu problemi ile doğrulanmaktadır. Simülasyon sonuçları, yaklaşımın daha iyi genel yakınsama performansı için oldukça rekabetçi olduğunu göstermiştir.

Yu ve diğ. (2007), bir merkez parçacığın doğrusal olarak azalan ağırlık parçacık sürüsü optimizasyonuna (LDWPSO) dahil edildiği bir CenterPSO önermiştir. Orta parçacığın konumu, her iterasyonda sıradan parçacıklardan oluşan sürünün merkezi ile güncellenir. Sürü aktivitesinin özelliği nedeniyle, tüm parçacıklar sürünün merkezi etrafında salınır ve yavaş yavaş ona doğru yakınsar.

Merkez parçacık genellikle iyi bir pozisyon alır ve genellikle koşu sırasında sürünün en iyisi olur. Bu nedenle, sadece bir merkez parçacığa rağmen, tüm sürünün aranmasına rehberlik etmek için daha fazla fırsata sahiptir ve performansı büyük ölçüde etkiler. Deneysel sonuçlar, CenterPSO'nun LDWPSO'dan daha iyi performans elde ettiğini göstermiştir. Zhang ve Zhu (2008), geliştirilen partikülleri seçmek için bir turnuva seçim operatörü tanıtarak CentrePSO'yu değiştirdiler. Üç kıyaslama fonksiyonundaki testler algoritmanın CentrePSO'dan daha istikrarlı ve hızlı olduğunu göstermiştir.

Wenxin Liu, Li Liu, David A. Cartes ve G.K. Venayagamoorthy ve diğ. (2007) BPSO algoritması olarak bilinen değiştirilmiş bir versiyon üretmiştir. Önerilen algoritma bir güç sistemini istenen sayıda adaya bölebilir. Ek olarak, kısıtlamalar uygun şekilde tanımlanabilirse başka gereksinimler de gerçekleştirilebilir. Örneğin, dinamik yanıtın göz önünde bulundurulması gerekiyorsa, jeneratör grupları önce yavaş tutarlılığa dayanarak bulunabilir ve daha sonra istenen işleve en fazla benzerliği sağlayabilecek bir çözüme en yüksek uygunluğu sağlamak için kondisyon fonksiyonunun değiştirilmesi gerekir. Makul, verimli ve zamanında çözümler sağlamak için sezgisel bir yanıt testi gereklidir.

Millie Pant, Radha Thangaraj ve Ajith Abraham ve ark. (2008), potansiyel bir çözümün tüm popülasyonunu mutasyona uğratmak yerine, popülasyonun sadece kişisel ve küresel en iyi parçacığını mutasyona uğratan modifiye edilmiş bir Uyarlamalı Mutasyon Parçacık Sürü Optimizasyonu (AMPSO1) ve AMPSO2 versiyonunu üretmiştir.

Shang ve Zhu (2008), geliştirilen partikülleri seçmek için bir turnuva seçim operatörü sunarak CentrePSO'yu değiştirdiler. Üç kıyaslama fonksiyonundaki testler algoritmanın CentrePSO'dan daha istikrarlı ve hızlı olduğunu göstermiştir.

Chuang ve diğ. (2008), bir yayın balığı parçacığını LDWPSO'ya dahil ederek bir Yayın Balığı PSO'yu tanıttı. Bir grubun aktif birkaç üyesinin bir grubun bir bütün olarak performansını uyarabilmesinden ilham alınmıştır. Yayın balığı PSO'da parçacıklar, en iyi değer birkaç yinelemeden sonra ilerleme kaydetmediğinde, arama alanının uç noktalarından yeni bir arama başlatır. Yayın Balığı PSO, LDWPSO ve CentrePSO'dan daha iyi performans elde etti.

Kang ve diğ. (2008) ekolojik nüfus yoğunluğu kavramını kullanarak bir Ekolojik PSO (EPSO) önerdi. EPSO algoritmasında, nüfus yoğunluğuna dayalı nüfus rekabeti ve iş birliği nüfus organizasyonuna ve evrimine eklenir; buradaki nüfus yoğunluğu nüfus rekabet modeli (Latka-Volterra modeli) tarafından kontrol ediliyor. Nüfus büyüklüğü Latka-Volterra modeline göre dinamik olarak hesaplanır ve ayarlanır, ancak popülasyondaki parçacık bireylerin evrimi, PSO'nun standart versiyonlarında aynı evrim stratejilerine göre her bir popülasyondaki bireyler için, nüfus iş birliği yoluyla evrimlerini etkilemek için genel bilgiler elde edilebilirler. Sonuçlar EPSO'nun küresel yakınsama performansının umut verici bir yöntem olduğunu göstermektedir.

Travis Besell, Malik Magdon-Ismail, Boleslaw Szymanski, Carlos Varels, Heidi Newberg ve Naithan Cole ve arkadaşları (2009) türevli APSO (Asenkron Parçacık Sürüsü Optimizasyonu) algoritmasının, arama parametrelerine daha az duyarlı olmakla birlikte, optimum sonuçlara daha hızlı yakınsama sağladığı gösterilmiştir. Sunulan doğrulama stratejisinin hem AGS hem de APSO için etkili olduğu gösterilmiştir ve amaçlanan algoritma bilimsel hesaplama problemlerine uygulanmaktadır.

Behnamian, Fatemi Ghomi, Tang ve ark. (2010) yeni bir hibrit PSO algoritması türetmiştir. Önerilen algoritma, yaklaşan atalet ağırlıklarının ve yaklaşan daralma faktörünün (IWCFA) özelliklerini birleştiren, IPSO adı verilen ve PSO'nun yakın zamanda geliştirilmiş bir varyantından esinlenmiştir. Güçlü yerel arama gücüne sahip değiştirilmiş simüle edilmiş tavlama yaklaşımı, daha fazla iyileştirme için dahil edilmiştir. Önerilen IPSO-SA yaklaşımı, IPSO ve SA algoritmalarındaki keşif yeteneklerinden tam olarak yararlanır ve her algoritmanın zayıflıklarını dengeler.

Narinder Singh ve S.B. Singh ve diğ. (2011) bir Yarım Küresel En İyi Pozisyon Parçacık Sürüsü Optimizasyonu (OHGBPPSO) algoritması geliştirmiştir. Bu algoritmanın performansı sayısal ve grafiksel sonuçlarla test edilmiştir. Elde edilen sonuçlar, ölçeklenebilir ve ölçeklenebilir olmayan problemler için standart PSO (SPSO) ile karşılaştırılır.

K. Thanushkidi, K. Deeba ve ark. (2011) yeni bir geliştirilmiş parçacık sürüsü optimizasyonu önermiş ve çok işlemcili atölye planlamasına uygulanmıştır. Önerilen

yeni algoritma, en uzun işlem süresi, en kısa işlem süresi ve partikül sürüsü optimizasyonu açısından diğer algoritmalar ile karşılaştırıldığında minimum bekleme süresi ve bitiş zamanı elde ederek işlemcilerdeki işleri bölümlere ayırmaya yardımcı olur. En iyi bileşenle birlikte dahil edilen en kötü bileşen, bilişsel davranışı ile bekleme süresini ve bitiş süresini en aza indirme eğilimindedir. Böylece aynı sayıda nesil için önerilen algoritma daha iyi sonuçlar elde etmiştir.

Narinder Singh ve S.B. Singh ve diğ. (2012) bir Kişisel En İyi Pozisyon Parçacık Sürü Optimizasyonu (PBPPSO) geliştirmiştir. Bu algoritmanın modifikasyonu, SPSO'nun hız güncelleme denkleminde küresel en iyi terimin ortadan kaldırılmasıyla yapılmıştır. Bu algoritmanın performansı çeşitli kıyaslama sorunları üzerinde test edilmiştir. Sonuçlar, önerilen algoritmanın SPSO'dan daha iyi performans sergilediğini gösterdi.

Bahman Bahmanifirouzi, Mehdi Nafar ve Mosoud Jabbari (2012), Ekonomik Sevk için Modifiye Parçacık Sürüsü Optimizasyonu tabanlı bir algoritma sundular. Önerilen yöntemin performansı, üç üretici birim içeren bir test durumunda gösterilmiştir.

3. GEREÇ VE YÖNTEM

Bu bölümde, BCO ve PSO algoritmalarının görev atama problemi özelinde incelemesi yapılırken kullanılacak olan algoritmalar hakkında bilgi verilmiş ve yöntemlerin algoritma ve matematiksel detayları birlikte açıklanmıştır. Ayrıca bu araştırmanın sonuçlarının gösterilmesinde ve sonuçların değerlendirilmesinde kullanılacak olan görsel ve hesaplama materyalleri hakkında bilgi verilmiştir.

Bu testler esnasında, Java programlama dili ile algoritmalar görev atama problemi için kodlanmış ve çıktıları Gantt şemasında gösterilmiştir. Bu değerlendirmelerin testlerinin yapıldığı bilgisayarın spesifikasyonları; Intel Core i7, 4 çekirdekli, 1.8GHz hızlı bir işlemci ile 8GB RAM 'dir.

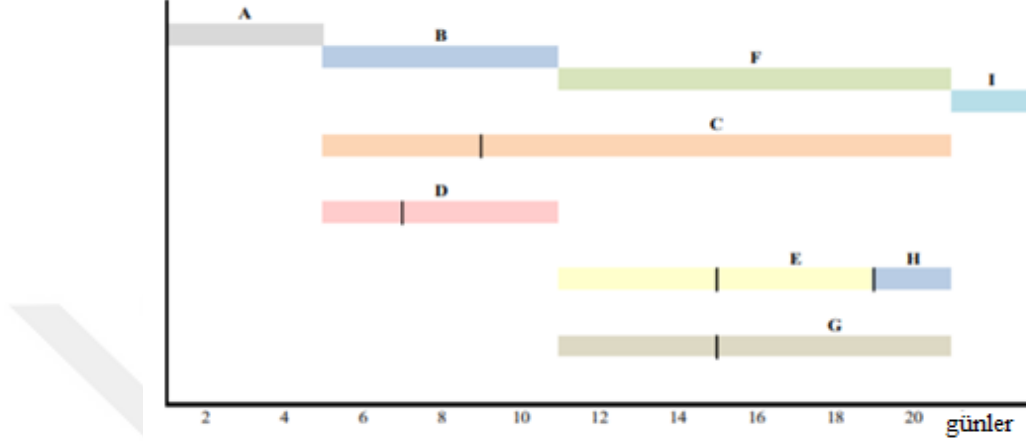
BCO ve PSO algoritmaları için geliştirilen uygulamanın kodları, (<https://github.com/erdikudu/thesis/>, Erişim Tarihi: 10 Şubat 2020) Github bulut depolama sistemine genel erişime açık olarak atılmıştır. Kullanılan veri setleri ile tezde yapılan çalışmaların Gantt şema sonuçları verilmiştir. İstenirse indirilip çalıştırılabilir. İleriki çalışmalar için geliştirmeye müsaittir.

3.1. Gantt Diyagramı

Gantt grafiği, bu işlerin tamamlanma süresini dahil etmek için bir hiyerarşi ve analog gövdede tamamlanması gereken işlerin her birinin grafiksel bir temsidir. Şekil 3.1'de görüldüğü üzere bir görev dizisi Gantt şeması üzerine dağıtılmıştır. Gantt şemaları gösterim türüne göre işleri ardı sıra (senkron) veya paralel (asenkron) olarak gösterebilmektedir. Yani, ilgili işin bir parçası bitmeden diğer parçası başlatılmamaktadır. İşler tek parça olarak gösterilmek zorunda değildir. Şekil 3.1'de görüldüğü üzere A işi bittikten sonra B, C işinin ilk parçası ve D işinin ilk parçası paralel olarak başlamıştır. Daha sonra C ve D işi çalışılmaya devam ederken F, E ve G işleri çalışılmaya başlamıştır. Sırasıyla diğer işler her bir makineye atanmış ve bütün işler çalışıldıktan sonra işler tamamlanmıştır (Şekil 3.1).

Gantt şemasının avantajları:

1. Her çalışma grafiksel olarak tasvir edilmiştir, bu yüzden anlaşılması kolaydır.
2. Her bir işin ilerlemesi, daha önce açıklandığı gibi kolayca gözlemlenir ve kontrol edilir.
3. Daha geniş yönler için diğer yöntemlerle birleştirilebilir.



Şekil 3.1. Örnek gantt şeması

3.2. İstatistiksel Hipotez Testi

İstatistiksel anlamlılık iki veya daha fazla değişken arasındaki ilişkinin şanstan başka bir şeyden kaynaklanma olasılığıdır. Bir veri setinin sonucunun istatistiksel olarak anlamlı olup olmadığını belirlemek için istatistiksel hipotez testi kullanılır.

Bunun olabileceği yerlere örnekler:

- Aynı konularda yapılan öncesi ve sonrası gözlemler (örneğin, öğrencilerin belirli bir modül veya dersten önce ve sonra tanısal test sonuçları).
- Ölçümlerin / tedavilerin aynı kişilere uygulandığı iki farklı ölçüm yönteminin veya iki farklı tedavinin karşılaştırılması (örneğin, bir stetoskop ve bir dinamap kullanılarak kan basıncı ölçümleri).

P değeri, bir karşılaştırmada “istatistiksel anlamlı fark vardır” kararın verileceği zaman yapılacak olası hata miktarını gösterir. Ünlü bir istatistikçi olan Fisher tarafından bu hatanın maksimum kabul edilebilir düzeyi 0,05 olarak önerilmiş ve kabul görmüştür. Bir test sonucunda bulunan P değeri 0,05’in altında bir değer ise

karşılaştırma sonucunda anlamlı farklılık bulunduğu anlamına gelir (Bulut Faruk, 2016).

P değeri küçüldükçe istatistiksel olarak anlamlı farklılığın kanıtı artar.

- P değeri 0,01 ile 0,05 aralığında; İstatistiksel olarak anlamlı fark vardır.
- P değeri 0,001 ile 0,01 aralığında; Yüksek düzeyde olarak anlamlı fark vardır.
- P değeri 0,001'den daha küçük ise; Çok yüksek düzeyde istatistiksel olarak anlamlı fark vardır.
- P değeri 0,10 ile 0,05 aralığında; Sınırdan anlamlılık anlamına gelmektedir.

Veri setlerinin algoritmadan çıkan sonuçları Paired T-Test Calculator (<http://www.statskingdom.com/160MeanT2pair.html>, Erişim Tarihi: 04 Ocak 2020) aracılığıyla değerlendirilmiştir. Bu değerlendirme sonucuna göre gözle yapılan performans yorumlarını istatistiksel olarak doğrulanmıştır (Bulut Faruk, 2020).

3.3. İş Atama Problemi İçin Yapay Arı Kolonisi Optimizasyonu

ABC algoritmasında yapay arılar üç gruba ayrılır: istihdam edilen arılar, seyirciler ve izciler. Bir gıda kaynağından yararlanan bir arı istihdam edilmiş olarak sınıflandırılır. Çalışan arılar kovanda bekleyen ve çalışan arıların danslarını izleyen seyirci arılarla bilgi paylaşırlar. Ondan sonra arılar, o gıda kaynağının kalitesiyle orantılı bir gıda kaynağı seçecektir. Bu nedenle, iyi gıda kaynakları kötü olanlardan daha fazla arı çeker. İzci arılar, kovanın yakınında rastgele yeni yiyecek kaynakları arar. Bir keşif kolu veya seyirci bir arı bir besin kaynağı bulduğunda işe başlar. Bir gıda kaynağından tam olarak yararlandığında, onunla bağlantılı olarak kullanılan tüm arılar pozisyonu terk eder ve tekrar izci olabilir. Dolayısıyla izci arılar “keşif” işini yaparken, çalışan ve seyirci arılar “sömürü” işini yaparlar. Algoritmada bir gıda kaynağı, optimizasyon problemine olası bir çözüme karşılık gelir ve bir gıda kaynağının nektar miktarı, ilgili çözümün uygunluğuna karşılık gelir.

ABC'de, koloninin ilk yarısı istihdam edilmiş arılardan, diğer yarısı da izleyicilerden oluşur. Kullanılan arıların sayısı gıda kaynaklarının sayısına (SN) eşittir, çünkü her gıda kaynağı için yalnızca bir tane arı olduğu varsayılmaktadır. Dolayısıyla, seyirci arıların sayısı da göz önüne alınan çözümlerin sayısına eşittir. ABC algoritması, rastgele oluşturulmuş bir grup besin kaynağı ile başlar. ABC'nin ana prosedürü aşağıdaki gibi tarif edilebilir.

Adım 1: Makine kaynaklarının konumunu belirtilir.

SN başlangıç çözümleri rastgele üretilen D-boyutlu gerçek vektörlerdir. $X_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\}$, tarafından elde edilen i. makine kaynağını temsil eder.

$$x_{i,d} = x_d^{min} + r \times (x_d^{max} - x_d^{min}), d = 1, \dots, D \quad (3.1)$$

Burada $r \in [0, 1]$ aralığında tekdüze bir rasgele sayıdır ve x_d^{min} ve x_d^{max} , sırasıyla boyut d için alt ve üst sınırlardır.

Adım 2: Çalışan her arı bir makine kaynağı üzerinde çalışmaya başlar.

Bu aşamada, kullanılan her arı bir çözüm ile ilişkilidir. Yeni bir çözüm (yeni makine kaynağı) bulmak için çözümde (orijinal makine kaynağı) rastgele bir değişiklik yapar. Bu, komşuluk aramanın işlevini uygular. Yeni v_i , bir diferansiyel ifade kullanılarak x_i 'den üretilir:

$$V_{i,d} = x_{i,d} + r' \times (x_{i,d} - x_{k,d}) \quad (3.2)$$

Burada d, $\{1, \dots, D\}$ arasında rastgele seçilir, k rastgele $\{1, \dots, SN\}$ arasından seçilir, $k \neq i$ ve $r' \in [-1, 1]$ aralığında bir rasgele sayıdır.

V_i elde edildiğinde, değerlendirilecek ve x_i ile karşılaştırılacaktır. V_i 'nin uygunluğu x_i 'den daha iyi ise (yani, yeni makine kaynağının iş miktarı eskisinden daha yüksekse), arı eski çözümü unutacak ve yenisini ezberleyecektir. Aksi halde, x_i üzerinde çalışmaya devam edecektir.

Adım 3: Her seyirci arı, çalışanlar tarafından paylaşılan iş bilgisine göre bir makine kaynağı seçer.

Çalışan tüm arılar yerel aramalarını bitirdiğinde, her biri bir makine kaynağını olasılıklı bir şekilde seçecek olan seyircilerle makine kaynaklarının iş bilgilerini paylaşırlar. Bir seyirci arının makine kaynağı x_i 'yi seçme olasılığı p_i , aşağıdaki gibi hesaplanır:

$$p_i = \frac{f_i}{\sum_{i=1}^{SN} f_i}, i = 1, \dots, SN \quad (3.3)$$

Adım 4: Makine kaynaklarını rastgele arayacak olan izci arıları belirleyin.

ABC’de, bir çözümün kalitesi önceden belirlenmiş sayıda denemeden (limit) sonra iyileştirilemezse, makine kaynağının terk edildiği varsayılır ve kullanılan arı bir izci haline gelir. İzci daha sonra Denklem (3.1) kullanarak rastgele bir besin kaynağı üretecektir.

Adım 5: Sonlandırma koşulunun karşılanıp karşılanmadığı test edilir. Değilse, 2. Adım'a geri dönülür.

3.4. İş Atama Problemi İçin Parçacık Sürü Optimizasyonu

Temel PSO’nun sosyo-bilişsel öğrenme süreci, bir parçacığın kendi deneyimine ve en başarılı parçacığın deneyimine dayanır. N değişkenli bir optimizasyon problemi için, NP parçacıklarının bir sürüsü tanımlanır, burada her parçacığa aday çözüm olarak n-boyutlu uzayda rastgele bir konum atanır. Her parçacık kendi yörüngesine, yani x_i ve v hızına sahiptir ve yörüngesini art arda güncelleyerek arama alanında hareket eder. Partikül popülasyonları, yörüngelerini daha önce kendileri ve diğer partiküller tarafından ziyaret edilen en iyi pozisyonlara göre değiştirir. Tüm parçacıkların optimize edilecek uygunluk fonksiyonu tarafından değerlendirilen uygunluk değerleri vardır. Parçacıklar, mevcut optimum parçacıklar izlenerek çözüm boşluğundan geçirilir. Algoritma, rastgele konumlara sahip bir grup parçacığı başlatır ve sonra nesiller güncellenerek optimumu arar. Her yinelemede, her bir parçacık, şu ana kadar elde ettiği en iyi çözüm olan x^*_i , $i = 1, \dots, NP$ olarak belirtilen en iyi iki değer, yani en iyi iki parçayı takip ederek güncellenir. Popülasyondaki herhangi bir parçacık tarafından şimdiye kadar elde edilen en iyi değer olan x^g ile gösterilen küresel en iyi g_{best} ’dir. Bir nesildeki nüfus için en iyi değer yerel en iyi l_{best} ’dir.

T + 1 yinelemesinde, sürü şu şekilde güncellenebilir:

$$v_i(t + 1) = v_i(t) + cr_1[x_i^*(t) - x_i(t)] + cr_2[x^g(t) - x_i(t)] \quad (3.4)$$

$$v_i(t + 1) = \begin{cases} -v_{max} & \text{eğer } v_i(t + 1) \leq -v_{max} \\ v_{max} & \text{eğer } v_i(t + 1) \geq v_{max} \end{cases} \quad (3.5)$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1), \quad i = 1, \dots, NP \quad (3.6)$$

Burada hızlanma sabiti $c > 0$ ve r_1 ve r_2 , $[0, 1]$ içinde eşit rasgele sayılardır. Bu temel PSO, hızların büyüklüğünün kontrolünün olmaması nedeniyle sürü patlamasına ve

ayrışmasına neden olabilir. Bu, v_i hızının mutlak değeri v_{max} eşiği ayarlanarak çözülebilir.

PSO, fiziksel olarak stokastik sönümlü bir kütle yay sisteminin özel bir ayrıklaştırılması olarak yorumlanabilir: PSO sürekli modeli, (3.4) 'den itibaren, parçacıkların hızları önceki hızları, bilişsel öğrenme (ikinci terim) ve sosyal öğrenme (üçüncü terim) ile belirlenir. Sosyal öğrenme nedeniyle, tüm parçacıklar g_{best} tarafından çekilir ve ona doğru hareket eder. Diğer iki kısım, parçacıkların kendi bilgilerini tutmasını sağlayan özerklik özelliğine karşılık gelir. Bu nedenle, arama sırasında, tüm parçacıklar g_{best} 'in bulunduğu bölgeye doğru hareket eder.

Sürüdeki tüm parçacıklar, g_{best} küresel optimumdan uzak olsa bile g_{best} 'ten öğrendiğinden, parçacıklar g_{best} bölgesine kolayca çekilebilir ve çok modlu problemler için yerel bir optimumda sıkışabilir. En iyi konumların lokal minimumda bulunması durumunda, sürüdeki diğer parçacıklar da sıkışabilir. Erken bir çözüm yetersizse, sürünün araştırmaya devam etmek için herhangi bir baskı olmadan kolayca durgunlaşabilir. Bu, (3.4) 'den görülebilir.

$$X_i(t) = x_i^*(t) = x^s(t) \text{ ise,}$$

hız güncellemesi yalnızca $\alpha_{v_i}(t)$ değerine bağlı olacaktır. Önceki hızları $v_i(t)$ sıfıra çok yakınsa, tüm parçacıklar g_{best} parçacığına yakalandıktan sonra hareket etmeyi durduracaktır. Daha da kötüsü, en iyi nokta yerel bir minimum olmayabilir. Bu fenomen durgunluk olarak adlandırılır. Durgunluğu önlemek için, yeniden arama veya kısmi yeniden başlatma, arama alanının farklı yerlerinde yeni parçacıklar üretilerek tanıtılır. PSO'nun hemen hemen tüm varyantları yerel optimum veya durgunluk problemini çözmeye çalışır.

PSO, optimum bölgeyi EA'lardan daha hızlı bulabilir. Bununla birlikte, bu bölgede bir kez sabit hız kademeli boyutu nedeniyle yavaş ilerler. Doğrusal olarak azalan ağırlık PSO (LDWPSO), parçacığın önceki hızına doğrusal olarak azalan atalet ağırlığı ekleyerek sürünün küresel ve yerel arama yeteneklerini etkili bir şekilde dengeler (3.4) (Shi Y, Eberhart RC, 1998):

$$v_i(t+1) = \alpha v_i(t) + C_1 r_1 [x_i^*(t) - x_i(t)] + C_2 r_2 [x^s(t) - x_i(t)] \quad (3.7)$$

Burada α atalet ağırlığı olarak adlandırılır ve c_1 ve c_2 pozitif sabitleri sırasıyla bilişsel ve sosyal parametrelerdir. Tipik olarak, $c_1 = 2.0$, $c_2 = 2.0$ ve α yavaş yavaş α_{max} 'tan α_{min} 'e düşer:

$$\alpha(t) = \alpha_{max} - (\alpha_{max} - \alpha_{min}) \frac{t}{T} \quad (3.8)$$

T maksimum yineleme sayısıdır. $\alpha_{max} = 1$ ve $\alpha_{min} = 0.1$ seçilebilir. PSO'nun algoritma adımları Çizelge 3.1'de verilmiştir.

Merkez PSO LDWPSO'ya bir merkez parçacığı ekler ve her yinelemede sürü merkezi olarak güncellenir (Liu Y ve diğ., 2007). Merkez partikülün hızı yoktur, ancak hız hesaplaması hariç, fitness değerlendirmesi, en iyi partikül için rekabet gibi sıradan partiküllerle aynı şekilde tüm operasyonlarda yer alır. Tüm parçacıklar sürü merkezi etrafında salınır ve yavaş yavaş ona doğru yakınsar. Merkez partikül genellikle koşu sırasında sürünün en iyisi haline gelir. Bu nedenle, tüm sürünün araştırılmasına rehberlik etmek için daha fazla fırsata sahiptir ve performansı büyük ölçüde etkiler. CenterPSO, LDWPSO'dan daha iyi çözümler sunmakla kalmaz, aynı zamanda daha hızlı yakınsama sağlar.

Çizelge 3.1. PSO Algoritmasının kaba kodu

-
1. $t=1$ yap.
 x_i konumu ve v_i hızı, $i = 1, \dots, NP$ için değerleri rastgele seçerek popülasyondaki her bir parçacığı başlatılır.
 2. Tekrarla
 - a. Her bir i parçacığın uygunluk değerini hesaplanır.
Her parçacık i için uygunluk değeri, şimdiye kadar bulunan en iyi uygunluk değerinden büyükse, $x_i^*(t)$ değerini düzeltilmelidir.
 - b. En yüksek uygunluğa sahip parçacığın yerini belirlenir ve gerekirse $x^g(t)$ revize edilir.
 - c. Her parçacık i için, hızını (3.4) veya (3.7) 'e göre hesaplanır.
 - d. Her bir parçacığın i konumunu (3.6) 'ye göre güncellenir.
 - e. $t = t + 1$ yap.
Durdurma ölçütleri karşılanıncaya kadar.
-

4. BULGULAR

Bu bölümde Arı Kolonisi Optimizasyon algoritması ve Parçacık Sürü Optimizasyon algoritmalarının çeşitli veri setleri üzerinde, çeşitli parametreler üzerinden değerlendirmeleri yapılmıştır. Çıkan sonuçlar çizelgeler üzerinde incelenmiş ve tespit edilen bulgular hakkında her Çizelge altında bilgi verilmiştir. Tezi daha okunabilir hale getirmek için sonuçlar çeşitli perspektiflerden çizelgelere dönüştürülerek anlatımların görsel olarak da anlaşılması amaçlanmıştır. Algoritmalarda kullanılan parametre değerleri aşağıdaki çizelgedeki gibidir (Çizelge 4.1).

Çizelge 4.1. BCO ve PSO algoritmalarının parametreler çizelgesi

BCO	
Makine (Arı) Sayısı	100
İterasyon	100
Komşuluk Sayısı	0,5

PSO	
Makine (Kuş) Sayısı	25
İterasyon	100
Komşuluk Sayısı	3
C1 sabiti	2,0
C2 sabiti	2,0

Çizelge 4.2. ve Çizelge 4.3.'de elde edilen sonuçlardan bazıları koyu renkli gösterilmektedir. Koyu renkle gösterilen değerler ilgili ölçüt ve klasmanda diğer algoritma ile karşılaştırıldığında en başarılı sonuç olduğunu simgelemektedir (Çizelge 4.1, 4.2).

Çizelge 4.2. BCO ve PSO çeşitli veri seti performans değerlendirme sonuçları

Veri Seti İsmi	Makine × İş	Arı Kolonisi				Parçacık Sürü			
		İşlemci Performans	Makespan	Averaj	Standart Sapma	İşlemci Performans	Makespan	Averaj	Standart Sapma
Ft06	6×6	13,60	55	55,02	0,14	13,60	58	57,51	0,98
La01	10×5	14,84	686	679,91	5,55	14,03	713	690,33	11,89
La02	10×5	13,41	678	680,03	5,32	17,12	700	693,79	12,53
Abz5	10×10	13,85	1273	1267,34	4,52	12,60	1295	1275,44	7,87
Abz6	10×10	13,68	956	959,94	6,64	13,25	977	978,68	4,67
La06	15×5	13,97	926	926,0	0,0	14,46	926	926,0	0,0
La07	15×5	12,93	890	890,2	0,81	12,75	894	895,91	10,59
La21	15×10	13,16	1153	1153,09	10,22	12,68	1181	1153,56	16,11
La22	15×10	13,06	1050	1029,21	13,86	14,34	1040	1029,74	17,41
La36	15×15	13,09	1402	1406,81	14,56	13,91	1406	1405,68	27,69
La37	15×15	12,91	1525	1524,52	11,46	13,09	1512	1515,81	17,73
La11	20×5	13,06	1222	1222,0	0,0	15,84	1222	1222,08	0,63
La12	20×5	12,91	1039	1039,0	0,0	13,56	1039	1039,0	0,0
La26	20×10	13,08	1328	1336,07	13,72	14,31	1322	1325,76	19,30
La27	20×10	12,92	1366	1384,79	12,82	14,25	1334	1376,27	14,86
Swv07	20×15	13,22	1947	1928,99	21,16	14,22	1896	1919,81	22,35
Swv08	20×15	12,87	2141	2119,41	19,69	13,26	2169	2106,14	28,54
Yn1	20×20	16,22	993	1001,03	7,37	24,81	998	993,04	9,97
Yn2	20×20	17,62	1044	1039,93	8,75	26,49	1042	1029,69	14,04
La31	30×10	14,38	1832	1830,04	14,01	22,67	1847	1811,74	22,68
La32	30×10	13,22	1945	1940,55	17,68	19,41	1952	1918,99	22,36
Swv12	50×10	13,22	3640	3662,98	27,22	17,88	3551	3632,48	55,21
Swv13	50×10	13,05	3741	3727,76	28,78	14,09	3702	3686,52	51,74

Çizelge 4.2’de Parçacık Sürü optimizasyon algoritması ve Arı Kolonisi optimizasyon algoritmalarının Londra’da bulunan Brunel Üniversite’sinin ORLibrary’sinden alınan 23 adet veri setine uygulanmış sonuçları görülmektedir. 23 örnekte, Lawrence tarafından tasarlanan LA01-LA40 örneklerinin bazıları (Lawrence, 1984), SWV01-SWV20 örneklerinin bazıları (Storer, Wu and Vaccari, 1992) kullanılmıştır. Bu çizelgede kullanılan makespan değeri iş atama problemlerine kullanılan bir veri setinin tamamlanma süresini belirten birim değerdir. Bir veri setinin toplamda tamamlandığı birime makespan denmektedir.

Çizelge 4.2’de veri setlerinin küçük olması sebebiyle çalışma zamanı analizi yapılmamıştır. Çünkü sonuçlarının birçoğunun 1sn’nin altında kaldığı gözlemlenmiştir. Ancak çeşitli boyuttaki veri setlerinde yapılan bu incelemeler bize göstermektedir ki: BCO algoritması 15×15 boyutuna kadar çoğunlukla PSO algoritmasına göre performans ve kararlı çalışma yönünden daha iyi çalışsa da 15×15’den sonraki veri seti büyüklüklerinde PSO algoritmasının BCO algoritmasına göre daha iyi sonuçlar verdiği gözlemlenmiştir.

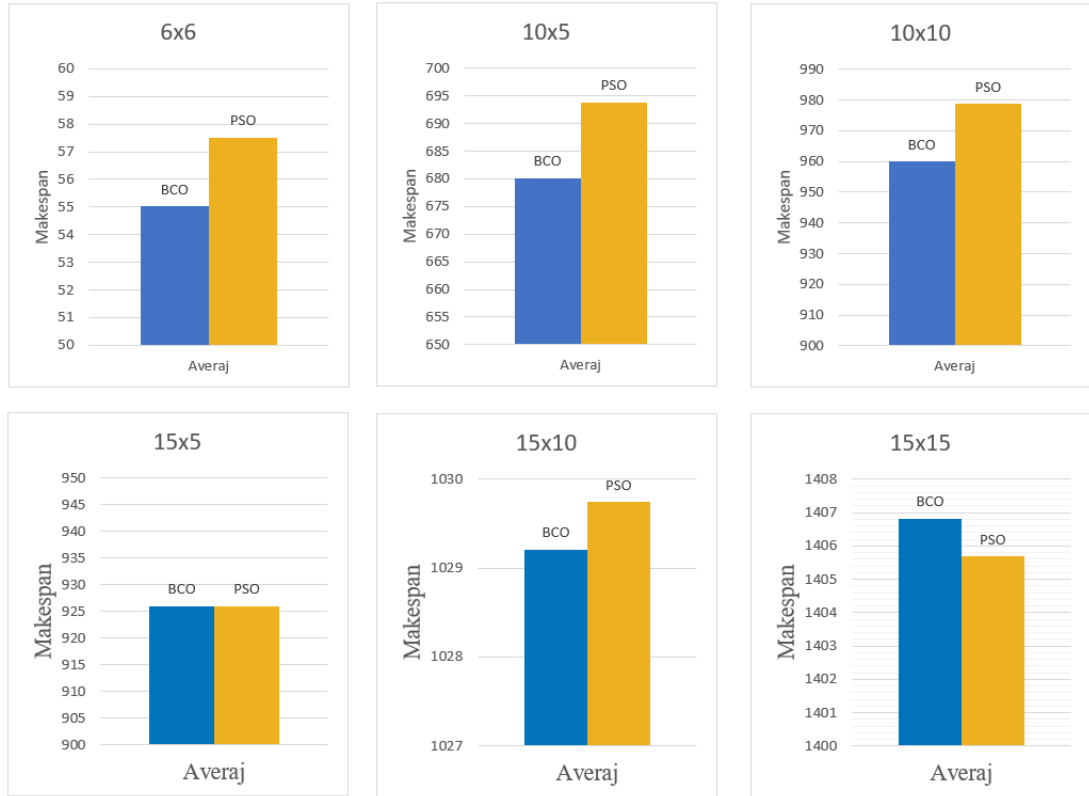
Çizelge 4.3. 50×10 veri seti (swv12) performans testi (I=100)

Test No	BCO					PSO				
	En İyi	En Kötü	Ortalama	Std. Sapma	Hesap. Zamanı (ms)	En İyi	En Kötü	Ortalama	Std. Sapma	Hesap. Zamanı (ms)
1	3583	3724	3667,46	26,58	577710	3506	3744	3623,54	50,75	108966
2	3551	3725	3662,24	35,78	570021	3510	3769	3618,17	50,06	97870
3	3570	3732	3670,66	30,06	566885	3498	3725	3625,3	50,04	99138
4	3589	3729	3661,74	30,20	637396	3481	3759	3627,61	52,20	98750
5	3579	3735	3664,75	30,19	571492	3537	3732	3638,62	46,57	98932
6	3573	3718	3662,98	28,56	576172	3490	3726	3621,37	48,08	100402
7	3585	3714	3663,4	31,10	583672	3493	3774	3626,85	49,18	99942
8	3595	3717	3667,0	30,21	576518	3508	3800	3624,03	42,70	99951
9	3562	3722	3663,38	29,33	566924	3516	3739	3626,06	50,21	93470
10	3574	3736	3662,08	30,32	528508	3499	3762	3630,93	51,17	93208

Çizelge 4.3’de Parçacık Sürü optimizasyon algoritması ve Arı Kolonisi optimizasyon algoritmalarının Londra’da bulunan Brunel Üniversite’sinin ORLibrary’sinden alınan “swv12” adlı 50×10 (İş × Makine)’luk veri seti üzerinde yapılan incelemelerdir.

Çizelge 4.3’de gözüktüğü üzere PSO aynı set üzerinde benzer parametrelerle averaj açısından daha iyi değerler vermesine rağmen standart sapması BCO ile karşılaştırıldığında daha kötüdür. BCO’nun standart sapma sonuçlarından görüldüğü üzere, BCO’nun değerleri PSO’ya göre birbirine daha yakındır. Yani PSO performans açısından daha iyi değerler verse de BCO daha kararlıdır.

Şekil 4.1 ve Şekilde 4.2’de Çizelge 4.2’den her bir veri seti boyutu için seçilen bir örneğin averaj değerinin sonuçları BCO ve PSO algoritmaları için karşılaştırılmıştır (Şekil 4.1, 4.2)



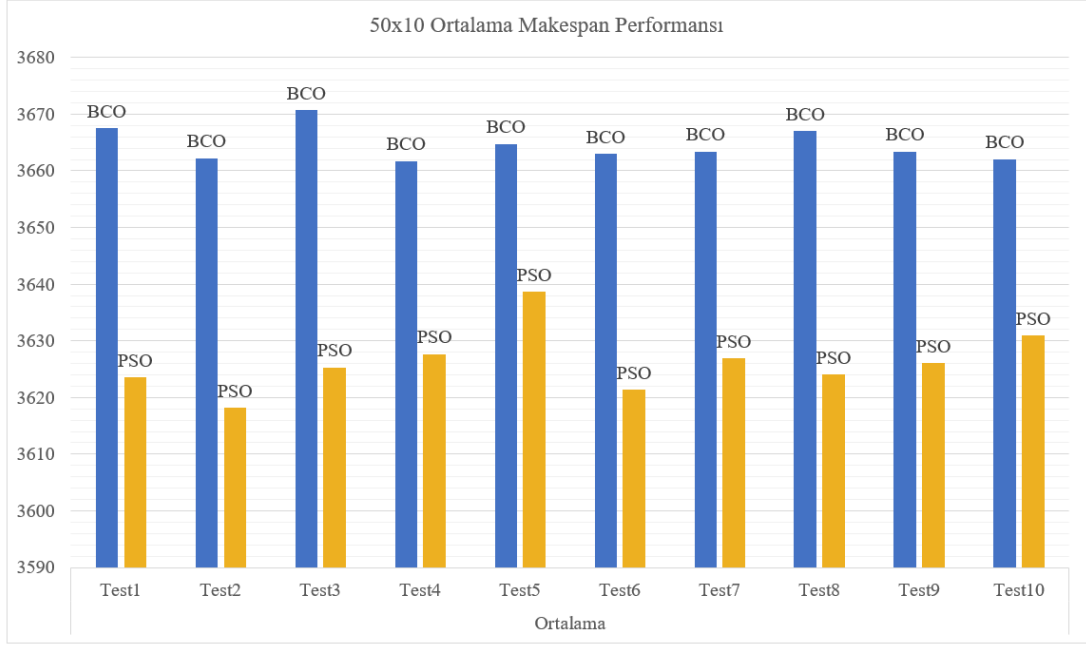
Şekil 4.1. Algoritma averaj performans karşılaştırması-1



Şekil 4.2. Algoritma averaj performans karşılaştırması-2

Şekil 4.1 ve Şekil 4.2'nin değerleri incelendiğinde açık bir şekilde küçük veri setleri için BCO'nun büyük veri setlerinde ise PSO'nun performanslı çalıştığı gözlemlenmiştir. Bu değerler algoritmaların üst üste 100 kere çalıştırılması ve çıkan sonucun aritmetik ortalamasının alınması sonucunda ortaya çıkmıştır. Küçük veri setlerinde BCO algoritması PSO algoritmasına göre daha performanslı çalışsa da 15×15 'lik veri setinden sonra PSO algoritması BCO algoritmasına ciddi oranda daha iyi performans göstermiştir.

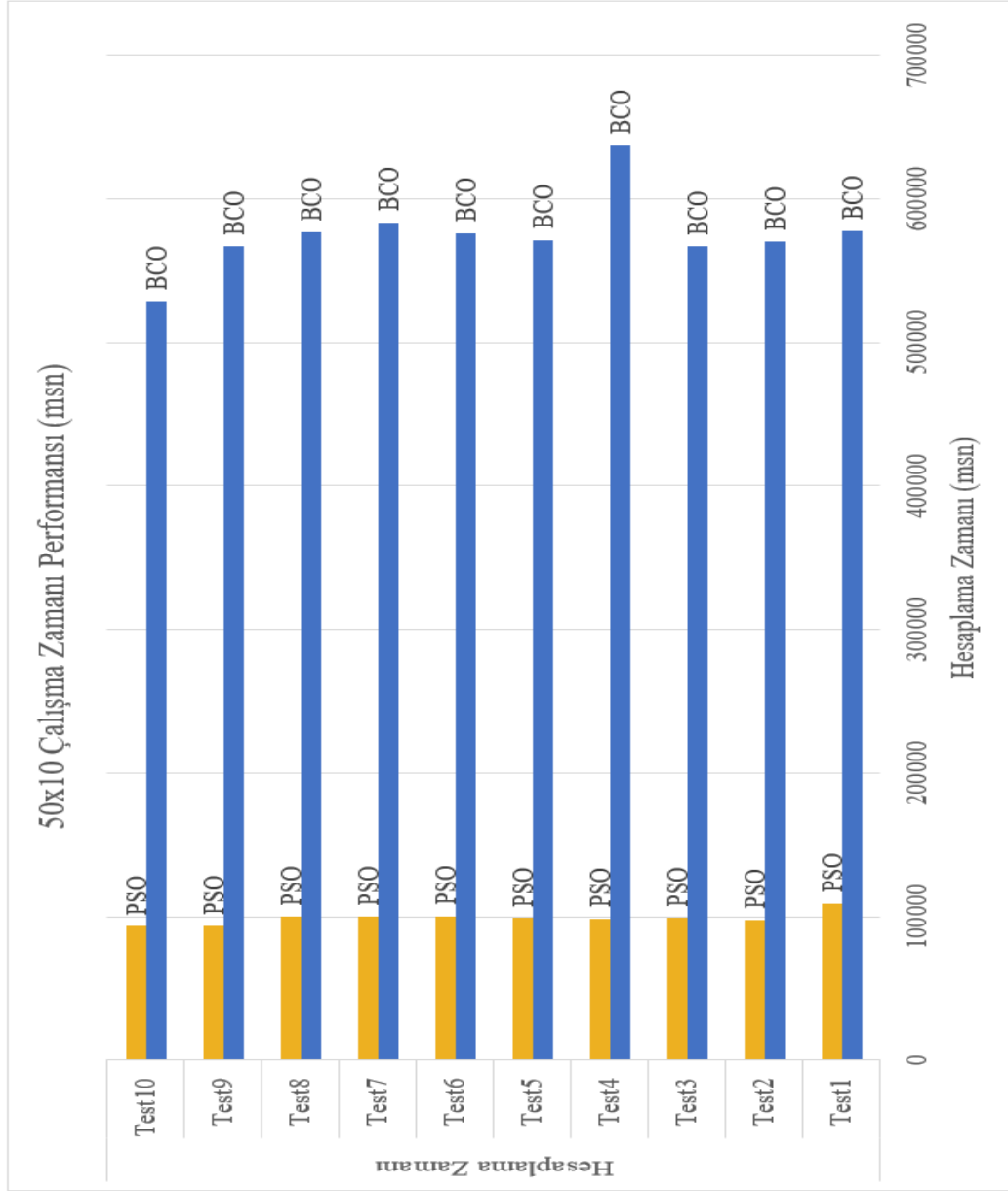
Şekil 4.3'de, Çizelge 4.3'de yapılan 10 adet test sonucunun "Ortalama" sütunu verilerine göre BCO ve PSO algoritmaları için analiz yapılmıştır (Şekil 4.3).



Şekil 4.3. Ortalama makespan performansı

Şekil 4.3’de 50×10 ’luk “swv12” isimli veri setinin yapılan uygulama aracılığıyla, 100 kere çalışma sonucunda alınan sonuçların ortalama makespan değerleri üzerinden analiz edilmiştir. Bu bir minimizasyon problemi olduğu için Çizelge bize PSO algoritmasının bu büyüklükteki bir veri setinde daha performanslı çalıştığını göstermektedir.

Şekil 4.4’de, Çizelge 4.3’de yapılan 10 adet test sonucunun “Hesaplama Zamanı” sütunu verilerine göre BCO ve PSO algoritmaları için analiz yapılmıştır (Şekil 4.4).

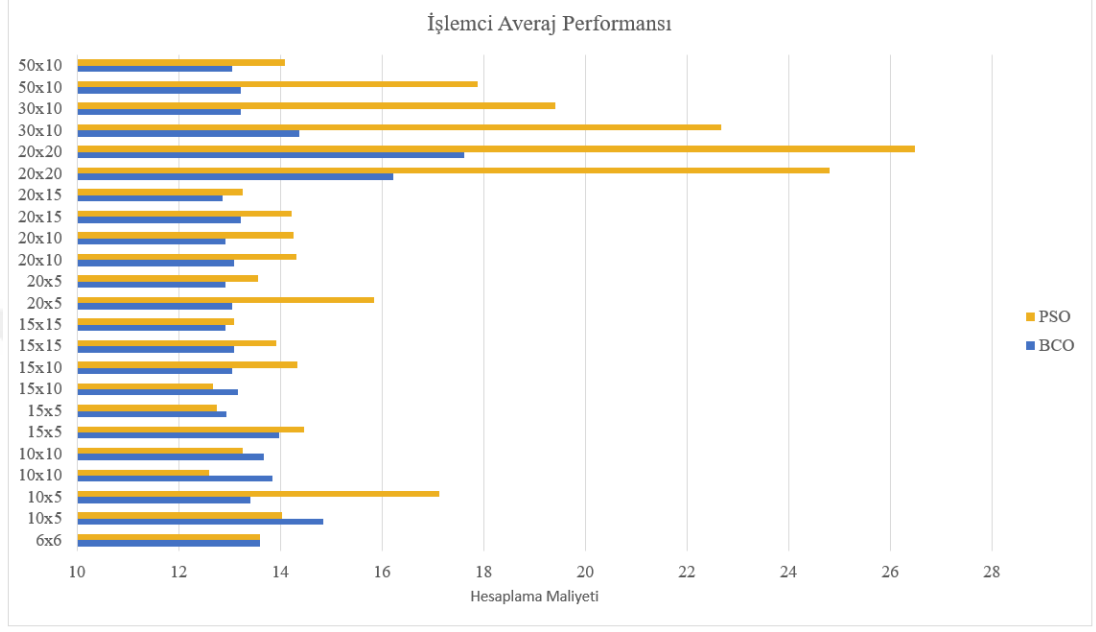


Şekil 4.4. Çalışma zamanı performansı

Şekil 4.4’de 50×10’luk “swv12” isimli veri setinin yapılan uygulama aracılığıyla, 100 kere çalışma sonucunda alınan sonuçların çalışma zamanı değerleri üzerinden analiz yapılmıştır. Bu testler üst üste 10 kere çalıştırılarak alınan sonuçların aritmetik ortalaması üzerinden yapılmıştır. Böylelikle sonuçların kararlılığı çıkan değerlere göre doğrulanmıştır. Bu çizelgeye göre PSO algoritmasının BCO algoritmasına göre daha kısa sürede sonuç verdiği açıkça görülmektedir. Bu sonuçlar ile 50×10’luk veya benzeri büyüklükteki bir veri setinde

var olan en iyi sonuçlardan birini almak isteyen birinin PSO algoritmasını tercih etmesi daha doğru olacaktır.

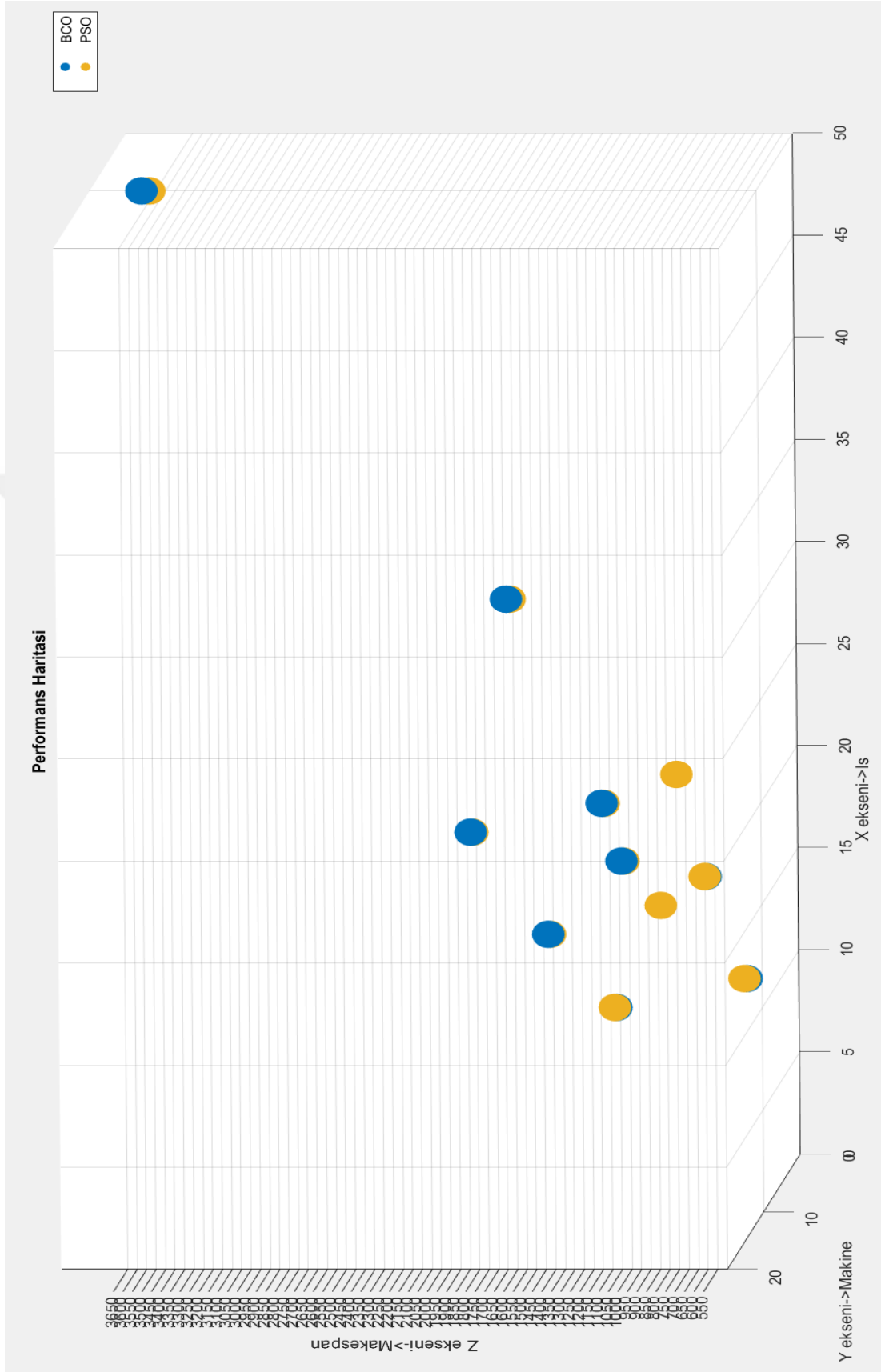
Şekil 4.5’de, Çizelge 4.2’de yapılan 23 adet test sonucunun “İşlemci Performans” sütunu verilerine göre BCO ve PSO algoritmaları için analiz yapılmıştır (Şekil 4.5).



Şekil 4.5. İşlemci averaj performansı

Şekil 4.5 de görüldüğü üzere BCO algoritması işlemcinin hesaplama maliyeti yönünden değerlendirildiğinde PSO’ya göre hesaplama maliyeti daha yüksektir. Veri seti büyüdükçe PSO’nun işlemciye daha fazla yüklendiği görülmektedir. Bu testler esnasında kullanılan kaynakla birlikte çalışma zamanı (ms) üzerinden de gözlem yapılmış, buna göre BCO’nun daha uzun sürede sonuç vermesine rağmen işlemci kaynağını daha az yordduğu gözlemlenmiştir. Buna karşın PSO algoritmasının daha kısa sürede sonuçları verdiği, ancak işlemciyi BCO’ya göre daha fazla ve yoğun kullandığı gözlemlenmiştir. Bu yoğun kullanım büyük veri setlerinde daha belirginleşmiştir. Veri seti daha da büyüdükçe bu yoğun kullanımın daha da artacağını işaret etmektedir.

Şekil 4.6’da Çizelge 4.2’de yapılan 23 adet test sonucundan her bir veri seti boyutu için bir örnek seçilmiş ve bu örneklerin “Averaj” sütununun verilerine göre BCO ve PSO algoritmaları için analiz yapılmıştır (Şekil 4.6).



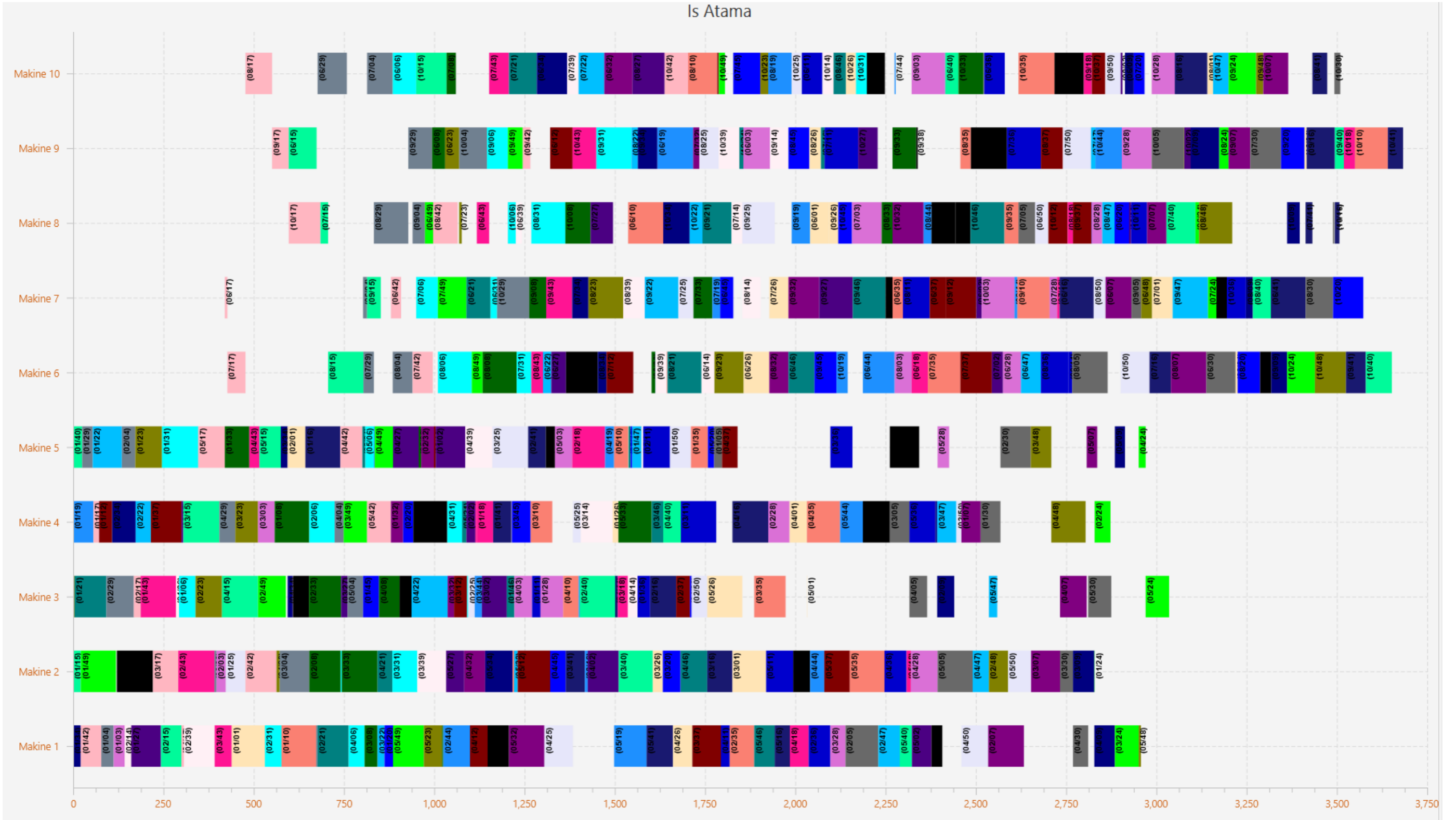
Şekil 4.6. Genel Performans Haritasi

Şekil 4.6’da başarımların sınırları verilmiştir. Makine ve iş sayısının özel olduğu olası bir senaryoda bu performans haritasına göre alınacak sonuç görülebilir ve uygun algoritma seçimi yapılabilir. Haritada X eksenini iş sayısını, Y eksenini makine sayısını, Z eksenini ise ilgili algoritmanın aldığı makespan performans değerini göstermektedir. Çizelgede, Z eksenine göre ve incelenen İş × Makine boyutuna göre bakıldığında daha aşağıda olan nokta performansın daha iyi olduğunu göstermektedir. Çizelgeden de görüleceği üzere mavi renkli noktalar Arı Kolonisi algoritma performans sonuçlarını, kırmızı noktalar Parçacık Sürü algoritma performans sonuçlarını göstermektedir. Harita, MATLAB uygulamasında çizdirilmiştir. 3 boyutlu bir Çizelge olduğu için sonuçların MATLAB dosyası uygulama kodlarının da bulunduğu Github kaynak kod deposuna atılmıştır. Bu haritanın her daim doğru sonuçları vereceği düşünülemez. Bu Çizelge öneride bulunur. Bu çalışma bir prototip çalışması olduğu için ara değerler için dâzinelerce farklı testler yapılarak daha güvenilir ve daha net sonuçlar elde edilebilir. Elbette ki çalışmadaki amaç temel anlamda algoritmaların karşılaştırılması ve ortaya taslak bir çalışmanın çıkarılmasıdır.

Çizelge sonuçlarında dikkat edilmesi gereken nokta, üstte kalan rengin ilgili İş × makine boyutunda daha kötü performans sergilediğidir. Şekil 4.6 incelendiğinde makine ve iş sayısının düşük olduğu testlerde PSO algoritması BCO algoritmasına göre daha kötü performans sergilemektedir. İş sayısı ve makine sayısının küçük ve birbirine yakın olduğu veri setlerinde PSO ve BCO algoritması birbirine yakın performans göstermiştir. BCO algoritması 15 × 5 boyutuna kadar PSO algoritmasına göre daha iyi performans sergilerken bu boyutlardan sonrasında performanslar birbirine eşitlenmiş ve 20 × 15 boyutundan sonra PSO algoritması BCO algoritmasına göre daha iyi performans sergilemeye başlamıştır. İş sayısı veya makine sayısından birinin yükselmeye başladığı veri setlerinde ise PSO algoritması BCO algoritmasına göre daha iyi performans göstermektedir. İş ve makine sayısının yüksek olduğu senaryolarda ise PSO algoritması BCO algoritmasına göre daha iyi performans göstermiştir. Bu bilgiler ışığında, çizelgede incelemesi yapılmamasına rağmen 100×5, 100×10 veya 100×20 gibi bir veri setine sahip olduğunda, PSO algoritmasının BCO algoritmasına göre daha iyi performans göstereceği tahmin edilebilmektedir.

Şekil 4.7’de Brunel Üniversitesi veri setlerinden 50×10^3 ’lük bir veri seti için kod çalıştırılmış ve örnek Gantt şeması aşağıdaki gibi sunulmuştur (Şekil 4.7).





Şekil 4.7. 50×10'luk veri seti örnek gantt şema çıktısı

Şekil 4.7.'de Java dilinde yazılan uygulamadan alınan görev atama çözümünün örnek Gantt şeması görülmektedir. Şemada x eksenini birim zamanı gösterir, y eksenini ise problemde girdi olarak sunulan makineleri göstermektedir. Her bir makine için belirtilen veri setindeki makine işleme sırasına göre BCO algoritmasının uygulanmış sonuçları gösterilmektedir.

Çizelge 4.4'de, Çizelge 4.3'de sunulan 10 adet 50×10^3 luk veri seti sonucunun "Ortalama" sütunu verileri hipotez testi için seçilmiştir (Çizelge 4.4).

Çizelge 4.4. İstatistiksel hipotez testi karşılaştırması

Ortalama Makespan Değerleri	
BCO	PSO
3667,46	3623,54
3662,24	3618,17
3670,66	3625,3
3661,74	3627,61
3664,75	3638,62
3662,98	3621,37
3663,4	3626,85
3667	3624,03
3663,38	3626,06
3662,08	3630,93

Çizelge 4.4.'ün verileri, BCO ve PSO algoritmalarına 50×10^3 luk veri setine 100 kere uygulamasından elde edilen makespan değerlerinin ortalamasıdır. Testte PSO (H_0) algoritma sonuçlarının BCO (H_1) algoritmasına göre karşılaştırması yapılmaktadır. Bu testte kullanılan parametreler; Önem Seviyesi (0-1), H_0 doğruyken H_0 'ın reddedilmesine izin verilen maksimum şans 0.05, standartlaştırılmış etki büyüklüğü 0.15, beklenen fark 0'dır.

Çizelge 4.5'de, Çizelge 4.4'de sunulan BCO ve PSO Ortalama değerlerinin istatistiksel değerlendirme sonuçları görülmektedir (Çizelge 4.5).

Çizelge 4.5. İstatistiksel hipotez testi sonuçları

X_d	38,321000
n	10
S_d	6,395525
Çarpıklık	-0,748828
P değeri normallik	0,3172

Çizelge 4.5.'de istatistiksel sonuçların çıktısı alınmıştır. Aşağıda çıktı sonuçlarının değişkenleri açıklanmıştır. $\mu_0 = 0$ 'dır.

X_d : Farkın ortalaması

n : Veri boyutu

S_d : Farkın standart sapması

μ_0 : Beklenen fark

BCO eksi PSO popülasyonunun ortalamasının μ_0 'a eşit olmadığı düşünülmektedir. Başka bir deyişle, BCO eksi PSO ve μ_0 ortalaması arasındaki fark, istatistiksel olarak anlamlı olacak kadar büyüktür.

P değeri $1,46160e-8$ 'e eşittir ($p(x \leq t) = 1,00000$). Bu, tip1 hatası olasılığının (doğru bir H_0 reddetme) küçük olduğu anlamına gelir: $1,462e-8$ (%0,0000015). P değeri ne kadar küçük olursa H_1 o kadar çok desteklenir.

Test istatistiği t 18.947880'e eşittir, kabul edilen %95 kritik değer aralığında değildir: [-2,2622: 2,2622]. $x = 38.32$, %95 kabul edilen aralıkta değil: [-4,5800: 4,5800].

Gözlenen standart etki büyüklüğü büyüktür (5,99). Bu, ortalama ve μ_0 arasındaki farkın büyüklüğünün büyük olduğunu gösterir. Aykırı değerlerinin tespit yöntemi olarak Tukey Çit ($k=1,5$) kullanılmıştır. Normalite varsayımı, Shapiro-Wilk Testine göre kontrol edildi. ($A = 0,05$) BCO eksi PSO'nun normal olarak dağıtıldığı (p değeri 0,317'dir) veya daha doğru bir şekilde normalite varsayımı reddedilemez. Öncelik gücü düşük olmasına rağmen (0,07103), H_0 reddedilir.

5. TARTIŞMA

Elbette ki literatürde tanımlı, doğruluğu kanıtlanmış birçok optimizasyon algoritması vardır. Bu algoritmalar aynı veri seti üzerinde denenip, farklı açılardan en iyi sonuç tespit elde edilebilir. Ancak bu tezdeki amaç bir prototip çalışma ile literatüre karşılaştırmalı çalışma sunmaktır. Elbette ki iş ve makine sayısının farklı olduğu veri setleri denenerek kıyaslamalı bir çalışma yapılabilir.

Bu çalışmada iş atama problemi ile ilgili n tane iş m tane makine ile yapılmak istenmiştir. Fakat bu problem tipinin basitliği ve endüstride çok fazla kullanılmayışı daha gerçekçi ve kabul edilebilir olan veri setlerine itmiştir. Bu veri setlerinin sonuçlarının karmaşık yapısı nedeniyle Gantt diyagramı kullanma zorunluğu ortaya çıkmıştır. Yapılan araştırmalar neticesinde tezde kullanılan veri seti türünde bulunan iş atama problem modelinin en yaygın model olduğu görülmüştür. Makine olarak belirtilen kısımlar farklı uygulamalarda insan veya kişi olarak da ele alınabilir. Örneğin; bir üretim tesisinde fireze, taşlama, boyama makinelerinin yerine boyama ustası, fireze ustası vb. ünvanlar ile de problem tasarlanabilir.

Bilindiği üzere herhangi bir algoritmanın performansı algoritmaya girilen parametre ile doğru orantılıdır. Özellikle tezde kullanılan sürü algoritmalarının birçoğunda ortak mantık olarak kullanılan lokal en iyi ve global en iyi parametrelerinin doğru seçilmediğinde bu özellikleri kaybedebilirler. Bu sebeple doğru parametreler girildiğinde yüksek başarımlar değerleri, yanlış parametre girildiğinde düşük başarımlar değerleri elde edilebilir. Bu çalışmada karşılaştırma yapılmadan önce bazı testler ve deneyler yapılarak en uygun parametreler algoritmalar için seçilmiştir. En uygun parametrelerin seçildiği şu açıdan bakıldığında görülmektedir. Her iki algoritmanın verdiği sonuçlar birbirine oldukça yakındır. Bu parametrelerin doğru seçildiğini göstermektedir. Elbette ki bir algoritmanın herhangi bir parametresi değiştirildiğinde farklı sonuçlar verdiği bilindiği için diğer algoritma karşısında düşük başarımlar değeri gösterecektir. Bu sonuçların yanlış değerlendirmesine sebep olacaktır. Bu nedenle deneysel

çalışmalara başlamadan önce en uygun parametre seçimi uzun uğraşlar gerektirmiştir. Çünkü verilen küçük bir yanlış değer çalışmanın yönünü eşit şartlar olmadan değerlendirme yapmaya götürecektir.

Bu çalışmanın hipotez aşamasında karşılaştırması yapılan algoritmalarından birinin diğerine göre açık ara farkla daha iyi performans göstermesi beklenmekteydi. Ancak yapılan deneysel çalışmalar gösterdi ki; makespan, standart sapma, makespan'ın ortalama averaj değeri gibi kıstasların sonuçları incelendiğinde değerlerin birbirine çok yakın olduğu görülmüştür. Bu sebeple istatistiksel analiz yöntemlerine ihtiyaç duyulmuştur.

Bu çalışma baz alınarak uygulanan veri setlerinin boyutları artırılarak birbirine yakın çıkan sonuçların daha büyük veri setlerinde birbirinden ayrılıp ayrılmayacağı test edilebilir. Bu bağlamda aslında algoritmaların hangisinin belirgin bir şekilde daha iyi çıkacağı doğrulanabilir. Bu çalışmada kullanılan veri setleri üniversite test veri setlerinden alınarak yapılmıştır. Bunun yerine gerçek yaşam senaryoları baz alınarak oluşacak sonuçlar incelenebilir ve algoritmaların endüstriyel alandaki performansı test edilebilir. Böylece algoritmaların iş dünyasına uyarlanıp uyarlanamayacağı örnekleme sunulabilecektir. Çalışma süresince kullanılan optimizasyon yöntemlerinin standart algoritmaları üzerine çalışılmıştır. Başka bir çalışmada, özellikle arama algoritmaları değiştirilerek algoritmanın performansını etkileyecek olan global ve lokal en iyilerin daha hızlı bulunması sağlanabilir. Böylece algoritmaların performanslarında artış gözlemlenebilir.

6. SONUÇ

Optimizasyon algoritmaları, yani en iyileme yöntemleri, rastsal süreçlerden geçerek en iyi sonucu bulmaktadır. Rastsal süreçlere stokastik işlemler de denmektedir. Bu çalışmada da literatürde sıkça rastlanılan stokastik iş çizelgeleme probleminin, sürü algoritmalarından Parçacık Sürü Optimizasyonu ve Arı Kolonisi Optimizasyon algoritmalarıyla çözümünün sonuçları değerlendirilmiştir.

Aslında bu yöntemin uygulamasındaki amaç, sürü algoritmalarının temelinde yatan rastgele dağıtım sebebiyle sonuçların bazen iyi bazen kötü çıkmasındaki şansa bağlı çıkan sonuçların doğrulanmasıdır. Bu sebeple sonuçların doğruluğunun ispatı için çeşitli veri setleri üzerinde inceleme yapılmıştır. Bu çalışmada sonuçlar performans açısından değerlendirmesi için her veri setinin kodlama esnasında yüz kez arda çalıştırılarak aritmetik ortalamasının alınması sağlanmıştır ve bu sonuçların çalışma zamanı ve standart sapma sonuçları değerlendirilmiştir. Bu sonuçlar, rastgele dağıtım sebebi ile bazen iyi bazen kötü çıkan tekil sonuçlara göre çoklu çalıştırmalarda sonucun farklı çıktığını göstermiştir. 20×20 'lik bir veri setinde tekil çalıştırmada makespan değerine göre sonuç BCO algoritmasının daha iyi olduğunu gösterirken, çoklu çalıştırmada averaj değerine göre PSO algoritmasının daha iyi olduğunu doğrulamıştır. Bu veri seti boyutundan sonraki birçok örnekte de bu sonuç görülerek bulgu doğrulanmıştır.

Bu bilgilerden yol çıkararak, yapılan kısıtlı denemelerden çıkan sonuçların istatistiki olarak da doğrulanması için, istatistiki doğrulama yöntemlerinden olan İstatistiksel Hipotez Testi yöntemi ile bu sonuçlar doğrulanmıştır. Çalışma bazlı doğrulanan sonuçların, İstatistiksel Hipotez Testi ile karşılaştırması yapıldığında, Çizelgeler üzerinden yorumlayarak yapılan değerlendirmelerin istatistiki olarak da doğru olduğu görülmektedir.

7. KAYNAKLAR

Abbass H.A., A Single Queen Single Worker Honey-Bees Approach to 3-SAT, GECCO2001 The Genetic and Evolutionary Computation Conference, San Francisco, USA, 2001a.

Abbass H.A., A Monogenous MBO Approach to Satisfiability, CIMCA'2001 International Conference on Computational Intelligence for Modeling, Control and Automation, Las Vegas, NV, USA, 2001b.

Abbass H.A., MBO: Marriage in Honey Bees Optimization A Haplometrosis Polygynous Swarming Approach, CEC2001 Proceedings of the Congress on Evolutionary Computation, Seoul, Korea, 207-214,2001c.

Afshar A., Bozorg Haddad O., Marino M.A., Adams B.J., Honey-Bee Mating Optimization (HBMO) Algorithm for Optimal Reservoir Operation, Journal of the Franklin Institute, 2007.

Akay, B., Karaboga D. Artificial bee colony algorithm for large-scale problems and engineering design optimization. J. Intell. Manuf. Doi:10.1007/s10845-010-0393-4, 2011.

Akay R., Akay B., Artificial Bee Colony Algorithm and an Application to Software Defect Prediction, Doi: 10.1007/978-3-030-26458-1_5, 2020.

Azeem M.F., Saad A.M., Modified Queen Bee Evolution Based Genetic Algorithm for Tuning of Scaling Factors of Fuzzy Knowledge Base Controller, IEEE INDICON 2004 Proceedings of the India Annual Conference, 299-303, 2004.

Basturk B., Karaboga D., An Artificial Bee Colony (ABC) Algorithm for Numeric Function Optimization, IEEE Swarm Intelligence Symposium 2006, Indianapolis, Indiana, USA, 2006. Baykasoğlu A., Goal Programming using the Multiple

Objective Tabu Search, Journal of Operational Research Society, 52 (12), 1359-1369,2001.

Benatchba K., Admane L., Koudil M., Using Bees to Solve a Data Mining Problem Expressed as a Max-Sat One, In Proceedings of IWINAC'2005, International Work Conference on the Interplay between Natural and Artificial Computation, Canary Islands, Spain, 212-220,2005.

Bianco G.M., Getting Inspired from Bees to Perform Large Scale Visual Precise Navigation, Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 619-624,2004.

Bozorg Haddad O., Afshar A., MBO Algorithm, A New Heuristic Approach in Hydrosystems Design and Operation, 1st International Conference on Managing Rivers in the 21st Century, 499-504,2004.

Bozorg Haddad O., Afshar A., Mariano M.A., Honey-Bees Mating Optimization (HBMO) Algorithm: A New Heuristic Approach for Water Resources Optimization, Water Resources Management, 20, 661-680,2006.

Brunel University, Londra. ORLibrary, JSP test instances. (<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/jobshop1.txt>, Erişim Tarihi: 15 Eylül 2019).

Bulut, Faruk, and Mehmet Fatih AMASYALI. "Katı kümeleme ve yeni bir geçiş fonksiyonuyla uzman karışımlarında sınıflandırma." Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi 31, no. 4, 2016.

Bulut, Faruk. "Performance evaluations of supervised learners on imbalanced datasets." In 2016 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT), pp. 1-4. IEEE, 2016.

Bulut, Faruk. "The minimum ratio of preserving the dataset similarity in resampling:(1- 1/e)." International Journal of Information Technology 12, no. 1 (2020): 231-244.

Cattrysse D., Set Partitioning Approaches to Combinatorial Optimization Problems, Ph.D. Thesis, Katholieke Universiteit Leuven, Centrum Industrieel Beleid, Belgium, 1990.

Cattrysse D., Salomon M., Van Wassenhove L.N., A Set Partitioning Heuristic for the Generalized Assignment Problem, *European Journal of Operational Research*, 72,167-174,1994.

Chang H.S., Converging Marriage in Honey-Bees Optimization and Application to Stochastic Dynamic Programming, *Journal of Global Optimization*, 35,423-441,2006.

Chong C.S., Low M.Y.H., Sivakumar A.I., Gay K.L., A Bee Colony Optimization Algorithm to Job Shop Scheduling, *Proceedings of the 37th Winter Simulation*, Monterey, California, 1954-1961,2006.

Chu P.C., Beasley J.E., A Genetic Algorithm for the Generalized Assignment Problem, *Computers Operations Research*, 24,17-23,1997.

Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput.*,6 (1):58–73, 2002.

Coelho LS, Krohling RA. Predictive controller tuning using modified particle swarm optimisation based on Cauchy and Gaussian distributions. In: *Proceedings of the 8th online world conference soft computing and industrial applications*, Dortmund, Germany. p. 7–12, 2003.

Drias H., Sadeg S., Yahi S., Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem, *IWAAN International Work Conference on Artificial and Natural Neural Networks*, Barcelona, Spain, 318-325,2005.

D. Costa and A. Hertz. Ants can colour graphs. *Journal of the Operational Research Society*, 48:295-305, 1997.

E. Lumer and B. Faieta. Diversity and adaptation in populations of clustering ants. *Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats3*, MIT Press, Cambridge, CA, pp. 501-508, 1994.

E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial System*. Oxford University Press, New York, 1999.

Esquivel SC, Coello CAC. On the use of particle swarm optimization with multimodal functions. In: Proceedings of IEEE congress on evolutionary computation (CEC), Canberra, Australia. p. 1130–1136, 2003.

Fisher M.L., Jaikumar R., Van Wassenhove L.N., A Multiplier Adjustment Method for the Generalized Assignment Problem, *Management Science*, 32,1095-1103,1986.

Fathian M., Amiri B., Maroosi A., Application of Honey-Bee Mating Optimization Algorithm on Clustering, 2007.

G. Leguizamón and Z. Michalewicz. A new version of ant system for subset problems. In *Proceedings of the Congress on Evolutionary Computation*, pages 1459—1464, 1999.

Hakli H, Uguz H. A novel particle swarm optimization algorithm with Levy flight. *Appl Soft Comput*, 23:333–45, 2014.

H. Lourenco and D. Serra. Adaptive search heuristics for the generalized assignment problem. Working Paper 2002-01, European Chapter on Metaheuristics, 2002.

Higashi N, Iba H. Particle swarm optimization with Gaussian mutation. In: *Proceedings of IEEE swarm intelligence symposium*, Indianapolis, IN, USA. p. 72–79, 2003.

Janson S, Middendorf M. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Trans Syst Man Cybern Part B.*, 35 (6):1272–82, 2005.

<https://github.com/erdikudu/thesis/> (10 Şubat 2020).

<http://www.statskingdom.com/160MeanT2pair.html> (04 Ocak 2020).

J. Adams, E. Balas, and D. Zawack. The shifting bottleneck procedure for job shop scheduling. *Mgmt. Sci.*, 34 (3):391–401, 1988.

J. Beasley. OR-Library. Online at <http://www.ms.ic.ac.uk/info.html>, 2004. Alp S (2011).

J.F. Muth and G.L. Thompson. *Industrial Scheduling*. Prentice-Hall, Englewood Cliffs, N.J., 1963.

J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3:159-168, 1990.

Kara A., Imitation of Bee Reproduction as a Crossover Operator in Genetic Algorithms, *PRICAI 2004*, LNAI 3157, C. Zhang, H.W. Guesgen, W.K. Yeap (Eds.), SpringerVerlag, Berlin Heidelberg, 1015-1016,2004.

Karaboga D., An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

Karaboga D., Aslan S., A genetic Artificial Bee Colony algorithm for signal reconstruction based big data optimization, Doi: 10.1016/j.asoc.2019.106053 , 2020.

Karaboga D., Caliskan A., Çil Z., Badem H., Regression Based Neuro-Fuzzy Network Trained by ABC Algorithm for High-Density Impulse Noise Elimination, Doi: 10.1109/TFUZZ.2020.2973123, 2020.

Kennedy J, Eberhart RC. *Swarm intelligence*. San Francisco, CA: Morgan Kaufmann; 2001.

Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of IEEE international conference on neural networks*, Perth, WA, USA, vol. 4. p. 1942–1948, 1995.

Kennedy J. Bare bones particle swarms. In: *Proceedings of IEEE swarm intelligence symposium*, Indianapolis, IN, USA. p. 80–87, 2003.

Kennedy J, Mendes R. Population structure and particle swarm performance. In: *Proceedings of congress on evolutionary computation*, Honolulu, HI, USA. p. 1671–1676, 2002.

Kennedy J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: *Proceedings of congress on evolutionary computation (CEC)*, Washington, DC, USA. p. 1931–1938, 1999.

Koudil M., Benatchba K., Tarabet A., Sahraoui E.B., Using Artificial Bees to Solve Partitioning and Scheduling Problems in Codesign, *Applied Mathematics and Computation*, 186 (2), 1710-1722, 2007.

Krohling RA. Gaussian swarm: a novel particle swarm optimization algorithm. In: Proceedings of IEEE conference cybernetics and intelligent systems, Singapore. p. 372– 376, 2004.

Lemmens N.P.P.M., To Bee or Not to Bee: A Comparative Study in Swarm Intelligence, Master Thesis, Maastricht University, Maastricht ICT Competence Centre, Institute for Knowledge and Agent Technology, Maastricht, Netherlands, 2006.

Lin T-L, Horng S-J, Kao T-W, Chen Y-H, Run R-S, Chen R-J, Lai J-L and Kuo I-H. An efficient job-shop scheduling algorithm based on particle swarm optimization☆ Expert Syst. Appl. 37 2629–36, 2010.

Liu Y, Qin Z, Shi Z, Lu J. Center particle swarm optimization. Neurocomputing.70:672– 9, 2007.

Liu Z. Investigation of Particle Swarm Optimization for Job Shop Scheduling Problem 3rd Int. Conf. Nat. Comput. (ICNC 2007) vol 3 (Haikou: IEEE) pp 799– 803, 2007.

Loengarov A, Tereshko V. A minimal model of honey bee foraging. In: Proceedings of IEEE swarm intelligence symposium, Indianapolis, IN, USA, p. 175–182, 2006.

Lucic P., Teodorovic D., Bee system: Modeling Combinatorial Optimization Transportation Engineering Problems by Swarm Intelligence, Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis, Sao Miguel, Azores Islands, 441-445,2001.

Lucic P., Teodorovic D., Transportation Modeling: An Artificial Life Approach, ICTAI'02 14th IEEE International Conference on Tools with Artificial Intelligence, 216-223,2002.

Lucic P., Teodorovic D., Computing with Bees: Attacking Complex Transportation Engineering Problems, International Journal on Artificial Intelligence Tools, 12 (3), 375-394,2003a.

Lucic P., Teodorovic D., Vehicle Routing Problem with Uncertain Demand at Nodes: The Bee System and Fuzzy Logic Approach, in Fuzzy Sets in Optimization, Editor J.L. Verdegay, Springer-Verlag, Berlin Heidelberg, 67-82,2003b.

Luckic P., Teodorovic D., Transportation Modeling: An Artificial Life Approach, ICTAI'02 14th IEEE International Conference on Tools with Artificial Intelligence, 216-223,2002.

Markovic G.Z., Teodorovic D., Acimovic-Raspopovic V.S., Routing and Wavelength Assignment in All-Optical Networks Based on the Bee Colony Optimization, AI Communications- The European Journal on Artificial Intelligence, 2007.

Martello S., Toth P., An Algorithm for the Generalized Assignment Problems, Operational Research, ed. J.P. Brans. North-Holland, 589-603,1981.

Martello S., Toth P., Knapsack Problems: Algorithms and Computer Implementations, Wiley, New York, 1990.

Mendes R, Kennedy J, Neves J. The fully informed particle swarm: simpler, maybe better. IEEE Trans Evol Comput., 8 (3):204–10, 2004.

M. Dell'Amico and M. Trubian. Applying tabu search to the job-shop scheduling problem. Annals of Operations Research, 41:231–252, 1993.

M. Dorigo and L. Gambardella. Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1997.

M. Randall and D. Abramson. A general meta-heuristic solver for combinatorial optimisation problems. Journal of Computational Optimization and Applications, 20: 185—210, 2000.

Nakrani S., Tovey C, On Honey Bees and Dynamic Allocation in an Internet Server Colony, Proceedings of 2nd International Workshop on the Mathematics and Algorithms of Social Insects, Atlanta, Georgia, USA, 2003.

Nauss R.M., Solving the Generalized Assignment Problem: An Optimizing and Heuristic Approach, INFORMS Journal of Computing, 15,249-266,2003.

Nayyar A., Nhu N.,Kumari R.,Kumar S., Robot Path Planning Using Modified Artificial Bee Colony Algorithm, Doi: 10.1007/978-981-13-9920-6_3, 2020.

Osman I.H., Heuristics for the Generalized Assignment Problem: Simulated Annealing and Tabu Search Approaches, *OR Spektrum*, 17,211-225,1995.

Parsopoulos KE, Vrahatis MN. UPSO: a unified particle swarm optimization scheme. In: *Proceedings of the international conference of computational methods in sciences and engineering*, 2004. The Netherlands: VSP International Science Publishers. pp. 868–873, 2004.

Pham D.T., Kog E., Ghanbarzadeh A., Otri S., Rahim S., Zaidi M., The Bees Algorithm – A Novel Tool for Complex Optimisation Problems, *IPROMS 2006 Proceeding 2nd International Virtual Conference on Intelligent Production Machines and Systems*, Oxford, Elsevier, 2006a.

Pham D.T., Otri S., Ghanbarzadeh A., Kog E., Application of the Bees Algorithm to the Training of Learning Vector Quantisation Networks for Control Chart Pattern Recognition, *ICTTA'06 Information and Communication Technologies*, 1624-1629, 2006b.

Pham D.T., Koc E., Ghanbarzadeh A., Otri S., Optimisation of the Weights of Multi-Layered Perceptions Using the Bees Algorithm, *Proceedings of 5th International Symposium on Intelligent Manufacturing Systems*, Sakarya, Turkey, 38-46,2006c.

Pham D.T., Soroka A.J., Ghanbarzadeh A., Kog E., Otri S., Packianather M., Optimising Neural Networks for Identification of Wood Defects Using the Bees Algorithm, *IEEE International Conference on Industrial Informatics*, 8, 1346-1351, 2006d.

Pongchairerks P and Kachitvichyanukul V. A Particle Swarm Optimization algorithm on Job-Shop Scheduling Problems with multi-purpose machines *Asia-Pacific J. Oper. Res.* 26 161–84, 2009.

Qin L.D., Jiang Q.Y., Zou Z.Y., Cao Y.J., A Queen-Bee Evolution Based on Genetic Algorithm for Economic Power Dispatch, *UPEC 2004 39th International Universities Power Engineering Conference*, Bristol, UK, 453-456,2004.

Quijano N., Passino K.M., Honey Bee Social Foraging Algorithms for Resource Allocation Theory and Application, *American Control Conference*, New York City, USA, 2007.

- Racer M., Amini M.M., A Robust Heuristic for the Generalized Assignment Problem, *Annals of Oper. Res.*, 50,487-503, 1994.
- Reeves WT. Particle systems—a technique for modeling a class of fuzzy objects. *ACM Trans Graph.*, 2 (2):91–108, 1983.
- Ross G.T., Soland P.M., A Branch and Bound Based Algorithm for the Generalized Assignment Problem, *Mathematical Programming*, 8,91-103,1975.
- Sato T., Hagiwara M., Bee System: Finding Solution by a Concentrated Search, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 4 (C), 39543959,1997.
- Secret BR, Lamont GB. Visualizing particle swarm optimization Gaussian particle swarm optimization. In: *Proceedings of the IEEE swarm intelligence symposium*, Indianapolis, IN, USA. p. 198–204, 2003.
- Seeley T.D., *The Wisdom of the Hive*, Harvard University Press, Cambridge, MA, 1995.
- Seeley T.D., Buhrman S.C., Group Decision Making in Swarms of Honey Bees, *Behav. Ecol. Sociobiol.*, 45,19-31,1999.
- S. S. Panwalkar and Wafix Iskander. A survey of scheduling rules. *Oper. Res.*, 25 (1):45–61, 1977.
- Shailendra Patidar, Narendra Singh Kushwah, Abhishek Yadav: A Novel Approach for Job Shop Scheduling using Particle Swarm Optimization, 2319-4863,2014.
- Shi Y, Eberhart RC. A modified particle swarm optimizer. In: *Proceedings of IEEE congress on evolutionary computation*, Anchorage, AK, USA, p. 69–73, 1998.
- Schmickl T., Thenius R., Crailsheim K., Simulating Swarm Intelligence in Honey Bees: Foraging in Differently Fluctuating Environments, *GECCO'05*, Washington, DC, USA, 273-274,2005.
- Sung H.J., Queen-Bee Evolution for Genetic Algorithms, *Electronic Letters*, 39 (6), 575-576, 2003.

Suganthan PN. Particle swarm optimizer with neighborhood operator. In: Proceedings of IEEE congress on evolutionary computation (CEC), Washington, DC, USA. p. 1958–1962, 1999.

Teo J., Abbass H.A., An Annealing Approach to the Mating-Flight Trajectories in the Marriage in Honey Bees Optimization Algorithm, Technical Report CS04/01, School of Computer Science, University of New South Wales at ADFA, 2001.

Teo J., Abbass H.A., A True Annealing Approach to the Marriage in Honey-Bees Optimization Algorithm, International Journal of Computational Intelligence and Applications, 3 (2), 199-211,2003.

Teodorovic D., Dell'Orco M., Bee Colony Optimization- A Cooperative Learning Approach to Complex Transportation Problems, Advanced OR and AI Methods in Transportation, 51-60,2005.

Teodorovic D., Selmic M., Davidovic T., Krüger T., The Bee Colony Optimization Algorithm and its Convergence, Vol:8, No:5, 2016.

Uslu, Mehmet Fatih, Süleyman Uslu, and Faruk Bulut. "An adaptive hybrid approach: Combining genetic algorithm and ant colony optimization for integrated process planning and scheduling." Applied Computing and Informatics, 2018.

Wedde H.F., Farooq M., Zhang Y., BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior, Ant Colony, Optimization and Swarm Intelligence, Eds. M. Dorigo, Lecture Notes in Computer Science 3172, Springer Berlin, 83-94,2004.

Yang X.S., Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms, IWINAC 2005, LNCS 3562, Yang, J. M. and J.R. Alvarez (Eds.), Springer-Verlag, Berlin Heidelberg, 317-323,2005.

Yonezawa Y., Kikuchi T., Ecological Algorithm for Optimal Ordering Used by Collective Honey Bee Behavior, 7th International Symposium on Micro Machine and Human Science, 249-256,1996.

8. ÖZGEÇMİŞ

Erdi KUDU

1988 yılında Zonguldak'ta doğan Erdi Kudu aslen Trabzonludur. İlköğretim ve Lise eğitimini Zonguldak'ta tamamladıktan sonra 2006 yılında Karadeniz Teknik Üniversitesi Bilgisayar Teknolojileri ve Programlama bölümüne girdi. Burayı bitirdikten sonra ilk iş yaşamına kurumsal bir şirketin Bilgi Teknolojileri bölümünde yardım masası biriminde başladı. 2010 yılında özel bir eğitim kuruluşundan aldığı yazılım eğitimi ile birlikte hastane yazılımları yapan bir yazılım firmasında Yazılım Uzman Yardımcısı olarak göreve başladı. Bu görevi esnasında Anadolu Üniversitesi İşletme Fakültesi İşletme eğitimini bitirdikten sonra askerlik görevini yaptı. Askerlik sonrası yine bir yazılım firmasında Yazılım Uzmanı olarak göreve başladı. Bu esnada Haliç Üniversitesi Bilgisayar Mühendisliği bölümündeki eğitimini tamamladı. Şu anda bir havayolu firmasının Bilgi Teknolojileri departmanında Yazılım Uzmanı olarak görev yapmaktadır.