



T.C.
KONYA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

SOSYAL ÖRÜMCEK ALGORİTMASININ
SÜREKLİ VE AYRIK OPTİMİZASYON
PROBLEMLERİNDE PERFORMANS
İYİLEŞTİRMELERİ

Emine BAŞ

DOKTORA TEZİ

Kasım-2020
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Emine BAŞ tarafından hazırlanan “SOSYAL ÖRÜMCEK ALGORİTMASININ SÜREKLİ VE AYRIK OPTİMİZASYON PROBLEMLERİNDE PERFORMANS İYİLEŞTİRMELERİ” adlı tez çalışması 18/11/2020 tarihinde aşağıdaki jüri üyeleri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Prof.Dr. Erkan ÜLKER

.....

Danışman

Prof.Dr. Erkan ÜLKER

.....

Üye

Dr.Öğr.Üyesi Onur İNAN

.....

Üye

Dr.Öğr.Üyesi Ayşe ELDEM

.....

Üye

Dr.Öğr.Üyesi Sait Ali UYMAZ

.....

Üye

Dr.Öğr.Üyesi Ömer Kaan BAYKAN

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN
Enstitü Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.


Emine BAŞ
18.11.2020

ÖZET

DOKTORA TEZİ

SOSYAL ÖRÜMCEK ALGORİTMASININ SÜREKLİ VE AYRIK OPTİMİZASYON PROBLEMLERİNDE PERFORMANS İYİLEŞTİRMELERİ

Emine BAŞ

Konya Teknik Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof.Dr. Erkan ÜLKER

2020, 181 Sayfa

Jüri

Prof.Dr. Erkan ÜLKER
Dr.Öğr.Üyesi Ömer Kaan BAYKAN
Dr.Öğr.Üyesi Onur İNAN
Dr.Öğr.Üyesi Ayşe ELDEM
Dr.Öğr.Üyesi Sait Ali UYMAZ

Doğa olguları temelli evrimsel hesaplama iki önemli gruba ayrılabilir. Bunlar evrimsel algoritmalar ve akıllı sürü temelli algoritmalar. Evrimsel algoritmalar doğadan ilham alınarak üretilmiş algoritmalar. Evrimsel algoritmalar optimizasyon problemlerinde ve gerçek dünya problemlerinde başarılı sonuçlar elde edebilmektedir. Bu tip algoritmalar genetik algoritma (GA), genetik programlama (GP), evrimsel stratejiler (ES) ve diferansiyel evrim (DE) örnek verilebilir. Akıllı sürü temelli algoritmalar ise son yıllarda oldukça ilgi çekmektedir. Bu tip algoritmalar Parçacık Sürü Optimizasyon algoritması (PSO), Yapay Arı Kolonisi algoritması (ABC), Karınca Kolonisi Optimizasyon algoritması (ACO) ve Yarasa Algoritması (BA) örnek verilebilir. Sürü terimi birbirleri ile iletişim halinde olan bireylerden oluşan topluluğu ifade etmektedir. Akıllı sürü temelli algoritmalar doğada var olan kuş, karınca, arı, bakteri, kelebek, örümcek v.b. gibi canlıların davranışlarını taklit ederler. Akıllı sürü temelli algoritmalar sürekli, ayrık ve ikili optimizasyon problemleri gibi pek çok farklı tip problemleri başarılı bir şekilde çözebilmektedirler. Çözülmesi çok zor olan NP-hard türü problem tiplerini çözmekte yüksek başarı göstermektedirler.

Sosyal Örümcek Algoritması (SSA), son yıllarda yeni geliştirilmiş sürü temelli bir algoritmadır. Doğada birlikte yaşayan örümcek türlerinin taklit edilmesi ile oluşturulmuştur. Bu tez kapsamında SSA'nın sürekli, ayrık ve ikili optimizasyon problemleri üzerindeki başarısı incelenmiş ve başarısını arttırmak için yeni yöntemler eklenerek SSA geliştirilmiştir.

SSA'nın ilk önerildiği orijinal algoritması küçük boyutlu (10, 20 ve 30) sürekli optimizasyon problemlerinde test edilmiştir. Bu tezde SSA'nın orijinal haline *Örümcek patlaması* ve *kaşif örümcek belleği* özellikleri eklenerek geliştirilmiştir ve Sosyal Örümcek Algoritması (ISSA) önerilmiştir. SSA'nın eklenen tekniklerle yerel ve global arama yeteneği artmıştır. ISSA'nın düşük, orta ve yüksek boyutlu (10, 20, 30, 100, 500 ve 1000) sürekli optimizasyon problemlerinde başarısı test edilmiştir. Sürekli optimizasyon görevleri için gerçekleştirilen farklı ikinci bir çalışmada orijinal SSA'ya *çaprazlama*, *mutasyon*, *G_{best} yakınsaması* ve *sessiz örümcek* şeklinde dört özellik eklenerek MPEF üzerinde ISSA'nın başarısı incelenmiştir. MPEF, ölçeklenebilir, basitleştirilmiş Moleküler Potansiyel Enerji Fonksiyonudur. Üçüncü çalışmada sürekli optimizasyon problemleri odaklı önerilen SSA, birbirinden bağımsız ayrık değerler alan problem türleri olarak bilinen ayrık optimizasyon problemlerini çözebilmesi için ayrık bir şekle getirilmiş ve Ayrık Sosyal Örümcek Algoritması (DSSA) önerilmiştir. DSSA ile literatürde sıklıkla

tercih edilen bir ayrık optimizasyon problemi olan *Gezgin Satıcı Problemi* (TSP) çözülmüştür. DSSA da, *yetenekli örümcek* ve *acemi örümcek* özellikleri eklenerek keşif ve sömürü yeteneği geliştirilmiştir.

Tez kapsamında ayrık optimizasyonun bir alt grubu olan ikili optimizasyon problemleri de çözülmüştür. Bu çalışmada önerilen İkili Sosyal Örümcek Algoritması (BinSSA) ile dört farklı ikili optimizasyon problemi (özellik seçme problemi, kapasitesiz tesis yerleşim problemi, rüzgar türbinleri yerleşim problemi ve sürekli optimizasyon problemleri) çözülmüştür. Bu çalışmada, sürekli arama uzayı ikili arama uzayına çevrilirken S-şekli, V-şekli ve mod tabanlı transfer fonksiyonları kullanılmıştır. İkili optimizasyonda transfer fonksiyonlarının başarısını desteklemek için son yıllarda sıklıkla kullanılan *lojik kapılar* (*xor lojik kapısı*), *benzerlik ölçme teknikleri* (*Jaccard* ve *Dice*) ve *çaprazlama operatörü* yeni aday çözümler elde etmek için tercih edilmiştir. Bu yöntemler sayesinde BinSSA ikili arama uzayında yeni noktalar keşfedebilmekte veya yerel noktalar etrafında yeni noktalar bulabilmektedir. Böylece BinSSA'nın başarısı geliştirilmiştir.

Anahtar Kelimeler: Ayrık optimizasyon, İkili optimizasyon, Özellik seçme, Kapasitesiz tesis yerleşim problemi, Rüzgar türbinleri yerleşim problemi, Sosyal Örümcek Algoritması, Sürekli optimizasyon.



ABSTRACT

PhD THESIS

PERFORMANCE IMPROVEMENTS OF SOCIAL SPIDER ALGORITHM IN CONTINUOUS AND DISCRETE OPTIMIZATION PROBLEMS

Emine BAŞ

Konya Technical University
Institute of Graduate Studies
Department of Computer Engineering

Advisor: Prof.Dr. Erkan ÜLKER

2020, 181 Pages

Jury

Prof.Dr. Erkan ÜLKER
Dr.Öğr.Üyesi Ömer Kaan BAYKAN
Dr.Öğr.Üyesi Onur İNAN
Dr.Öğr.Üyesi Ayşe ELDEM
Dr.Öğr.Üyesi Sait Ali UYMAZ

Evolutionary calculation based on natural phenomena can be divided into two important groups. These are evolutionary algorithms and intelligent swarm-based algorithms. Evolutionary algorithms are algorithms inspired by nature. It can achieve successful results in evolutionary optimization problems and real world problems. Such algorithms include genetic algorithm (GA), genetic programming (GP), evolutionary strategies (ES), and differential evolution (DE). Intelligent swarm based algorithms have attracted a lot of attention in recent years. Such algorithms include the Particle Swarm Optimization algorithm (PSO), the Artificial Bee Colony algorithm (ABC), the Ant Colony Optimization algorithm (ACO), and the Bat Algorithm (BA). The term swarm refers to the community of individuals who are in contact with each other. Intelligent swarm-based algorithms imitate the behavior of bird, ant, bee, bacteria, butterfly, spider etc. Intelligent swarm-based algorithms can successfully solve many different types of problems such as continuous, discrete and binary optimization problems. They are very successful in solving the problem types of NP-hard which are very difficult to solve.

The newly developed Social Spider Algorithm (SSA) in recent years is a swarm-based algorithm. It is a swarm based algorithm created by imitating spider species living together in nature. Within the scope of this thesis, the success of SSA on continuous, discrete and binary optimization problems is examined and developed by adding new methods to increase its success.

Social Spider Algorithm (SSA) is a newly developed herd-based algorithm in recent years. It was created by imitating spider species that live together in nature. Within the scope of this thesis, the success of SSA on continuous, discrete and binary optimization problems has been examined and SSA has been developed by adding new methods to increase its success.

The SSA was tested in small size (10, 20, and 30) continuous optimization problems in the original algorithm for which SSA was proposed first. In this thesis, the original version of SSA was developed by adding *spider explosion* and *explorer spider memory* features and Social Spider Algorithm (ISSA) was proposed. In addition, the ability to search locally and globally has increased. The success of ISSA in low, middle, and high dimensional (10, 20, 30, 100, 500, and 1000) continuous optimization problems has also been tested. In a second study performed for continuous optimization tasks, the success of ISSA on MPEF was examined by adding four features to the original SSA in the form of *crossover*, *mutation*, *G_{best} convergence* and *silent spider*. MPEF is a scalable, simplified Molecular Potential Energy Function. In the third study, the proposed SSA focusing on continuous optimization problems was

transformed into a discrete form in order to solve the discrete optimization problems known as the problem types that take independent discrete values, and the Discrete Social Spider Algorithm (DSSA) was proposed. *The Traveling Salesman Problem* (TSP), which is a discrete optimization problem frequently preferred in the literature, has been solved with DSSA. In DSSA, exploration and exploitation capabilities are enhanced by adding *skilled spider* and *novice spider* features.

In this thesis, binary optimization problems, a subset of discrete optimization, are also solved. With the Binary Social Spider Algorithm (BinSSA) proposed in this study, four different binary optimization problems (feature selection problem, uncapacitated facility location problem, wind turbines placement problem and continuous tasks) are solved. In this study, S-shaped, V-shaped and mode-based transfer functions are used while converting the continuous search space to a binary search space. In order to support the success of transfer functions in binary optimization, *logic gates (xor logic gate)*, *similarity measurement techniques (Jaccard and Dice)*, and *the crossover operator*, which are frequently used in recent years, have been preferred to obtain new candidate solutions. Thanks to these methods, BinSSA can discover new points in the binary search space or find new points around local points. Thus, the success of BinSSA has been improved.

Keywords: Binary optimization, Continuous optimization, Discrete optimization, Feature selection, Social Spider Algorithm, Uncapacitated facility location problem, Wind turbines placement problem.



ÖNSÖZ

Doktora tez çalışmam boyunca değerli katkılarını, yönlendirici desteğini ve anlayışını hiçbir zaman esirgemeyen danışmanım Sayın Prof. Dr. Erkan ÜLKER'e;

Tezin gelişmesine yönlendirici görüş ve önerileri ile yardımcı olan ve manevi desteklerini esirgemeyen tez izleme komitesi üyelerim Sayın Dr. Öğr. Üyesi Ömer Kaan BAYKAN'a ve Sayın Dr. Öğr. Üyesi Onur İNAN'a;

Bu süre boyunca göstermiş oldukları manevi desteklerinden, ilgilerinden ve yardımlarından dolayı Dr. Öğr. Gör. Vahit TONGUR'a, Dr. Öğr. Üyesi Sedat KORKMAZ'a, Selçuk Üniversitesi Kulu Meslek Yüksekokulu'nun tüm öğretim elemanlarına ve diğer mesai arkadaşlarıma;

Çalışmalarım boyunca ve hayatımın her anında anlayışları ve destekleri ile yanımda ve yardımcım olan sevgili eşim Şaban BAŞ'a;

İçtenlikle teşekkür eder, şükran ve minnetlerimi sunarım.

Emine BAŞ
KONYA-2020

İÇİNDEKİLER

TEZ BİLDİRİMİ	iii
ÖZET	iv
ABSTRACT.....	vi
ÖNSÖZ	viii
İÇİNDEKİLER	ix
SİMGELER VE KISALTMALAR	xi
1. GİRİŞ	1
1.1. Tezin Konusu, Amacı ve Literatüre Katkısı.....	5
1.2. Tezin Organizasyonu.....	7
2. KAYNAK ARAŞTIRMASI	9
2.1. Sosyal Örümcek Algoritması (SSA)	9
2.2. Sürekli Optimizasyon Problemleri için Sürü Zekası Yöntemleri	10
2.2.1. Büyük Ölçekli Küresel Optimizasyon (LSGO) için sürü yöntemleri.....	10
2.2.2. Moleküler Potansiyel Enerji Fonksiyonu problemi (MPEF) için sürü yöntemleri	11
2.3. Ayırık Optimizasyon Problemleri için Sürü Zekası Yöntemleri.....	11
2.4. İkili Optimizasyon Problemleri için Sürü Zekası Yöntemleri.....	12
2.4.1. Sürekli optimizasyon görevleri için sürü yöntemleri.....	13
2.4.2. Özellik Seçme (FS) problemi için sürü yöntemleri	13
2.4.3. Kapasitesiz Tesis Yerleşim Problemi (UFLP) için sürü yöntemleri.....	15
2.4.4. Rüzgâr türbini yerleşim problemi için sürü yöntemleri.....	16
3. MATERYAL VE YÖNTEM.....	18
3.1. Orijinal Sosyal Örümcek Algoritması (SSA)	18
3.1.1. Başlangıç ve uygunluk değerlendirme	19
3.1.2. Titreşim üretme.....	19
3.1.3. İkili maske değişimi ve random yürüyüş	20
3.1.4. Sınır ihlallerini kontrol etme	21
3.1.5. r_a , p_c and p_m parametre değerleri	22
3.2. Sürekli Optimizasyon Görevleri için ISSA.....	22
3.2.1. Büyük Ölçekli Küresel Optimizasyon (LSGO) için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA)	22
3.2.2. Moleküler Potansiyel Enerji Fonksiyonu (MPEF) problemi için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA)	27

3.3. Ayırık Optimizasyon için Ayırık Sosyal Örümcek Algoritması (DSSA)	34
3.3.1. DSSA için TSP tasarımı	34
3.3.2. DSSA için yetenekli örümcekler tekniği	34
3.3.3. DSSA için acemi örümcek tekniği.....	36
3.3.4. DSSA için 2-OPT arama algoritması.....	36
3.4. İkili Optimizasyon için İkili Sosyal Örümcek Algoritması (BinSSA)	38
3.4.1. Sürekli optimizasyon görevleri için İkili Sosyal Örümcek Algoritması (BinSSA).....	39
3.4.2. Özellik seçme (FS) problemi için İkili Sosyal Örümcek Algoritması (BinSSA).....	47
3.4.3. Kapasitesiz Tesis Yerleşim Problemi (UFLP) için İkili Sosyal Örümcek Algoritması (BinSSA).....	51
3.4.4. Rüzgâr türbini yerleşim problemi için İkili Sosyal Örümcek Algoritması (BinSSA).....	55
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	62
4.1. Sürekli Optimizasyon Görevleri için ISSA Test Sonuçları.....	64
4.1.1. Büyük Ölçekli Küresel Optimizasyon (LSGO) için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA) sonuçları	64
4.1.2. Moleküler Potansiyel Enerji Fonksiyonu (MPEF) problemi için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA) sonuçları.....	77
4.2. Ayırık Optimizasyon Görevleri için DSSA Test Sonuçları.....	86
4.3. İkili Optimizasyon Görevleri için BinSSA Test Sonuçları.....	91
4.3.1. Sürekli optimizasyon görevleri için BinSSA.....	92
4.3.2. Kapasitesiz Tesis Yerleşim Problemi (UFLP) için BinSSA.....	106
4.3.3. Rüzgar türbini yerleşim problemi için BinSSA	122
4.3.4. Özellik Seçme (FS) problemi için BinSSA	131
5. SONUÇLAR VE ÖNERİLER	162
5.1. Sonuçlar	162
5.2. Öneriler.....	168
6. KAYNAKLAR	169
ÖZGEÇMİŞ.....	181

SİMGELER VE KISALTMALAR

ABC	:Yapay Arı Kolonisi algoritması (Artificial Bee Colony algorithm)
ACO	:Karıncı Kolonisi Optimizasyonu (Ant Colony Optimization)
ACS	:Karıncı Kolonisi Sistemi (Ant Colony System)
ALO	:Karıncı Aslanı Optimizasyonu (Ant Lion Optimization)
BA	:Yarasa Algoritması (Bat Algorithm)
BALO	:İkili Karıncı Aslanı Optimizeri (Binary Ant Lion Optimizer)
BAAA	:İkili Yapay Alg Algoritması (Binary Artificial Algae Algorithm)
BAAASB	:Stigmergic davranışa sahip bir İkili Yapay Alg Algoritması (A Binary Artificial Algae Algorithm with stigmergic behavior)
BBA	:İkili Yarasa Algoritması (Binary Bat Algorithm)
BBHA	:İkili Kara Delik Algoritması (Binary Black Hole Algorithm)
BBSO	:İkili Beyin Fırtınası Optimizasyonu (Binary Brainstorming Optimization)
BCROSAT	:Simüle edilmiş tavlama ile ikili mercan resifleri optimizasyon algoritması (Binary Coral Reefs Optimization algorithm with Simulated Annealing)
BAAA	:İkili Yapay Alg Algoritması (Binary Artificial Algae Algorithm)
BinABC	:İkili Arı Kolonisi Optimizasyonu (Binary Bee Colony Optimization)
BGSA	:İkili Yerçekimi Arama Algoritması (Binary Gravity Search Algorithm)
BHPSOWM	:İkili Hibrit Parçacık Sürüsü Optimizasyonu (Binary Hybrid Particle Swarm Optimization)
BHS	:İkili Uyum Arama (Binary Harmony Search)
BIWO	:İkili Yabani ot optimizasyonu (Binary Weed optimization)
BinSSA	:İkili Sosyal Örümcek Algoritması (Binary Social Spider Algorithm)
BJA	:İkili Jaya algoritması (Binary Jaya Algorithm)
BMOA	:İkili Manyetik Optimizasyon Algoritması (Binary Magnetic Optimization Algorithm)
BOP	:İkili Optimizasyon Problemi (Binary Optimization Problem)
BPSO	:İkili Parçacık Sürü Optimizasyonu (Binary Particle Swarm Optimization)
BSalpSA	:İkili Salp Sürüsü Algoritması (The Binary Salp Swarm Algorithm)
BSCA	:İkili Sinüs Kosinüs Algoritması (Binary Sine Cosine Algorithm)
BSSA	:İkili Sosyal örümcek Algoritması (Binary Social Spider Algorithm)
BSSO	:İkili Sosyal Örümcek Optimizasyon Algoritması (Binary Social Spider Optimization Algorithm)
CLPSO	:Kapsamlı Öğrenme Parçacık Sürüsü Optimize Edici (Comprehensive Learning Particle Swarm Optimizer)
CMA-ES	:Kovaryans Matris Adaptasyon Evrim Stratejisi (Covariance Matrix Adaptation Evolution Strategy)
CSA	:Guguk kuşu Arama Algoritması (Cuckoo Search Algorithm)
CSSA	:Kaotik Sosyal Örümcek Algoritması (Chaotic Social Spider Algorithm)
DABC	:Ayrık Yapay Arı Kolonisi (Discrete Artificial Bee Colony)
DCS	:Ayrık Guguk kuşu Arama Algoritması (Discrete Cuckoo Search Algorithm)
DE	:Diferansiyel Evrim (Differential Evolution)
DBA	:Ayrık Yarasa Algoritması (Discrete Bat Algorithm)

DEVIIC	:DE ile geliştirilen VIIC tekniği
DFA	:Ayrık Ateş böceği Algoritması (Discrete Firefly Algorithm)
DICA	:Ayrık Emperyalist Rekabet Algoritması (Discrete Imperialist Competition Algorithm)
DIWO	:Ayrık İstilacı Ot Optimizasyonu (Discrete Invasive Weed Optimization)
DMRSA	:Ayrık Dinamik çok ölçekli bir bölge Arama Algoritması (Discrete Dynamic multiscale region Search Algorithm)
DPSO	:Ayrık Parçacık Sürüsü Optimizasyonu (Discrete Particle Swarm Optimization)
DSOS	:Ayrık Simbiyotik Organizmalar Arama algoritması (Discrete Symbiotic Organisms Search algorithm)
DSSA	:Ayrık Sosyal Örümcek Algoritması (Discrete Social Spider Algorithm)
DVNS	:Ayrık Değişken Komşuluk Arama algoritması (Discrete Variable Neighborhood Search algorithm)
DWCA	:Ayrık Su Döngüsü Algoritması (Discrete Water Loop Algorithm)
EDFA	:Evrimsel Ayrık Ateşböceği Algoritması (Evolutionary Discrete Firefly Algorithm)
ELD	:Ekonomik Yük Dağıtımı (Economic Load Distribution)
ESA	:Ayrık Arama Algoritması (Discrete Search Algorithm)
ESM	:Kaşif örümcek hafızası (Explorer spider memory)
FA	:Ateşböceği Algoritması (Firefly Algorithm)
FS	:Özellik seçme (Feature Selection)
GA	:Genetik Algoritma (Genetic Algorithm)
GSA	:Yerçekimi Arama Algoritması (Gravity Search Algorithm)
GSOA	:Büyüyen Kendi Kendini Düzenleyen Dizi (The Growing Self-Regulating Series)
GWO	:Gri Kurt Optimizeri (Gray Wolf Optimizer)
HACO	:Hibrit Karınca Kolonisi Optimizasyonu (Hybrid Ant Colony Optimization)
HBEPESO	:Yarasa ve Geliştirilmiş Parçacık Sürü Optimizasyonu algoritmalarının ikili bir versiyonu (A binary version of the Bat and Enhanced Particle Swarm Optimization algorithms)
HFBGA	:Çiçek tozlaşması ve Genetik algorithmadan oluşan bir hibrit algoritma (A hybrid algorithm consisting of flower pollination and Genetic algorithm)
HGWOGA	:Gri Kurt Optimizeri ve Genetik Algorithmadan oluşan bir hibrit algoritması (A hybrid algorithm consisting of Gray Wolf Optimizer and Genetic Algorithm)
HPSOGE	:Parçacık Sürü Optimizasyonu ve Genetik Algorithmadan oluşan bir hibrit algoritması (A hybrid algorithm of Particle Swarm Optimization and Genetic Algorithm)
HS	:Uyumlu Arama algoritması (Harmony Search algorithm)
HSSOGA	:Sosyal Örümcek Optimizasyonu ve Genetik Algorithmadan oluşan bir hibrit algoritması (A hybrid algorithm consisting of Social Spider Optimization and Genetic Algorithm)
IDGA	:Ada modelli Ayrık Genetik Algoritma (Island Model Discrete Genetic Algorithm)
IGWO	:Yaratıcı Gri Kurt Optimizasyonu (Creative Gray Wolf Optimization)

ISCA	:Geliştirilmiş Sinüs Kosinüs Algoritması (Improved Sine Cosine Algorithm)
ISS	:Geliştirilmiş Dağılım Arama (Improved Scatter Search)
ISSA	:Geliştirilmiş Sosyal Örümcek Algoritması (Improved Social Spider Algorithm)
Jaya	:Jaya algoritması (Jaya algorithm)
LS	:Yerel Arama (Local Search)
LSGO	:Büyük Ölçekli Küresel Optimizasyon (Large Scale Global Optimization)
MBACO	:İkili kodlanmış Karınca Kolonisi Optimizasyon algoritması (Binary coded Ant Colony Optimization algorithm)
MCSO	:Modifiye Rekabet Sürüsü Optimizeri (Modified Competition Swarm Optimizers)
MGWO	:Modifiye edilmiş Gri Kurt Optimizeri (Modified Gray Wolf Optimizer)
MOSSO	:Çok Amaçlı Sosyal Örümcek Optimizasyonu (Multipurpose Social Spider Optimization)
MPEF	:Moleküler Potansiyel Enerji Fonksiyonu (Molecular Potential Energy Function)
MPSO	:Modifiye edilmiş Parçacık Sürü Optimizasyonu (Modified Particle Swarm Optimization)
MSSA	:Modifiye edilmiş Sosyal Örümcek Algoritması (Modified Social Spider Algorithm)
MSSO	:Memetik Sosyal Örümcek Optimizasyon algoritmasını (Memetic Social Spider Optimization algorithm)
MWOA	:Balina Optimizasyon Algoritması (Whale Optimization Algorithm)
NGWO	:Doğrusal Olmayan Kontrol Parametrelili GWO
NNS	:En Yakın Komşuluk Arama (Nearest Neighborhood Search)
NBSG	:Yeni İkili Çözüm Üreticisi (New Binary Solution Generator)
PSO	:Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
Rc-BBFA	:İade maliyetine dayalı ikili FFA
rHYB	:Gerçek kodlu bir genetik-simpleks hibrit algoritması (A real coded genetic-simplex hybrid algorithm)
SA	:Benzetimli Tavlama (Simulated Annealing)
SFLA	:Karışık Kurbağa Sıçrayan Algoritması (Tangled Frog Leaping Algorithm)
SOMAQI	:Karesel İnterpolasyonlu Kendi Kendini Düzenleyen Bir Göç Algoritması (A Self-Regulating Migration Algorithm with Quadratic Interpolation)
SSA	:Sosyal Örümcek Algoritması (Social Spider Algorithm)
SSO	:Sosyal Örümcek Optimizasyonu (Social Spider Optimization)
Std.Dev.	:Standart sapma (Standard deviation)
TS	:Tabu Arama (Tabu Search)
TSA	:İkili Ağaç Tohumu Algoritması (Binary Tree Seed Algorithm)
TSP	:Gezgin Satıcı Problemi (Traveling Salesman Problem)
UFLP	:Kapasitesiz Tesis Yerleşim Problemi (Uncapacitated Facility Settlement Problem)

1. GİRİŞ

Son 20 yıldır, sürü temelli algoritmalar dikkat çekmektedir. Sürü terimi, birbirleriyle iletişim halinde olan bireylerden oluşan bir topluluğu ifade eder. Sürü zekasına dayalı algoritmalar, problemleri çözmek için sosyal hayvan ve böcek davranışlarını dikkate alırlar (Parpinelli ve Lopes, 2011). Hesaplamalı bir optimizasyon metodolojisi, sonlu bir çözüm kümesinden uygun çözümler bulmayı ve yalnızca en uygun çözüm veya çözümleri belirlemeyi içerir. Akıllı sürü algoritmaları, hesaplama optimizasyonu metodolojisinin bir alt sınıfını oluşturur (Blum ve Li, 2008). Akıllı sürü algoritmaları, etkileşimde bulunan bireylerin yerel ve / veya küresel ortamlara adaptasyonunda topluluğa ait kolektif davranışların ortaya çıkmasına dayanılarak geliştirilmiştir. Akıllı sürü algoritmalarının literatürde birçok örneği mevcuttur. Örneğin: Parçacık Sürü Optimizasyonu (PSO) (Kennedy ve Eberhart, 1995), Karınca Kolonisi Optimizasyonu (ACO) (Dorigo, 1992), Yarasa Algoritması (BA) (Yang, 2010b), Ateşböceği Algoritması (FA) (Yang, 2010a; Yang ve Deb, 2009), Guguk kuşu Arama Algoritması (CSA) (Gandomi ve ark., 2013), Yapay Arı Kolonisi (ABC) (Karaboga, 2005; Karaboga ve Basturk, 2007; Akay, 2013), vb.

Bu tez kapsamında son yıllarda yeni geliştirilen ve literatürde yeterinde çalışılmamış sürü zekâsına dayanan Sosyal Örümcek Algoritması (SSA) çalışılmıştır. SSA, James ve Li (2015a) tarafından sosyal örümcek davranışlarından esinlenilerek geliştirilmiş bir algoritmadır. Örümcekler genellikle doğada yalnız yaşarlar. Fakat bazı örümcek türlerinin koloni olarak da yaşadığı yapılan incelemelerle ortaya çıkmıştır. Kolonilerde yaşayan örümcekler örümcek ağı yardımıyla birbirleriyle iletişim kurabilirler. Örümcek ağı üzerinde ürettikleri titreşim ile bu iletişimi sağlarlar. Paylaştıkları titreşim bilgisiyle birbirlerine rastgele yürüyüşler yapabilirler (James ve Li, 2015a). SSA, düşük ölçekli optimizasyon problemlerinde çözülmüş ve bazen optimum seviyeye ulaşmakta zayıf görünen sürekli bir optimizasyon algoritmasıdır. SSA, son yıllarda geliştirilmiş bir algoritma olduğu için literatürde çok fazla çalışması mevcut değildir. El-Bages ve Elsayed (2017), SSA temelli bir algoritma önermişlerdir ve statik iletim genişleme planlama problemini önerdikleri algoritma ile çözmüşlerdir. Önerilen yöntemde; DC güç akışı alt sorunu, orijinal bir SSA tabanlı ağ geliştirilerek ve sonuca potansiyel çözümler eklenerek çözülmüştür. James ve Li (2016)'da, orijinal SSA'yı geliştirmiş ve Ekonomik Yük Dağıtımı (ELD) problemini çözmüşlerdir. Mousa ve

Bentahar (2016)'da QoS-bilinçli ağ hizmeti seçim sürecini SSA ile çözmüşlerdir. Çalışmalarında SSA yaklaşımını kullanmışlardır. Elsayed ve ark. (2016)'da dışbükey olmayan ekonomik yük gönderme problemini SSA temelli bir yaklaşımla çözmüşlerdir. Baş ve Ülker (2020a), SSA'nın ayrık versiyonunu geliştirmiş ve gezgin satıcı problemi üzerinde performansını test etmiştir. Baş ve Ülker (2020b), SSA'nın ikili versiyonunu "xor" lojik kapısı ve Jaccard'ın benzerlik ölçme yöntemlerini ekleyerek geliştirmiş ve sürekli optimizasyon görevlerinde performansını test etmiştir. Baş ve Ülker (2020c), SSA'nın sekiz farklı transfer fonksiyonuna göre ikili versiyonunu çarpazlama operatörü ekleyerek geliştirmiş ve özellik seçme problemini çözmüşlerdir. Baş ve Ülker (2020d), SSA'nın ikili versiyonunu geliştirmiş ve kapasitesiz tesis yerleşim problemini çözmüşlerdir.

Mühendislik gibi çeşitli disiplinlerdeki birçok problem, bir dizi ayrık değerle sınırlı olabileceği gibi sürekli değerlere sahip olan bir dizi sistem (karar) parametresi (veya değişkeni) de içerebilmektedir. Ayrık değişkenler, malzemenin kalınlığında olduğu gibi gerçek değerleri veya aşındırıcıların ağ boyutunda olduğu gibi tamsayı değerlerini alabilir (Liao ve ark., 2012). Literatürde birçok ayrık optimizasyon problemi vardır (Gezgin Satıcı Problemi (TSP), sırt çantası problemi vb.) ve bunlar genelde NP-hard problemlerdir (Kuzu ve ark., 2014). Bu tür problemleri klasik yöntemlerle çözmek zordur. Teknik olarak, bazı sistem parametreleri de sonsuz değerleri alabilir. Bu tür değişkenli optimizasyonlar sürekli optimizasyonlardır. Sürekli optimizasyonların düşük veya çok sayıda karar değişkeni olabilir. Çoğu gerçek dünya optimizasyon probleminin çok sayıda karar değişkeni vardır. Bu kadar çok sayıda karar değişkeniyle ilgili sorunlar Büyük Ölçekli Küresel Optimizasyon (LSGO) olarak bilinir. Bu problemlerin çoğu, büyük ölçekli boyutları nedeniyle karmaşık ve çözülmesi zordur. Bu tür sorunların boyutu arttıkça nihai çözümün kalitesi önemli ölçüde azalır (boyut>10) ve hesaplama zamanı artar. Birçok gerçek yaşam problemi büyük ölçekli optimizasyon problemleridir. Örneğin: su yönetimi probleminin optimizasyonu (Goh ve ark., 2011), Norveç doğal gaz üretiminin optimizasyonu ve ulaşımı (Ray ve Yao, 2009), ABD ordusu istasyonunun optimizasyonu (Dell ve ark., 2008), vb. Bu tür problemler için yüksek kaliteli karar verme gereksinimi, büyük ölçekli optimizasyon problemlerini çözme konusunu değerli ve zorlu bir araştırma alanı haline getirmiştir (Sayed ve ark., 2015). Birçok araştırmacı literatürde büyük boyutlu optimizasyon üzerinde çalışmaya başlamıştır ve birçok meta-sezgisel algoritma önerilmiştir (DEVIIC (Sayed ve ark.,

2015), MWOA (Sun ve ark., 2018), CC (Trunfio ve ark., 2016), DE (Maućec ve Brest, 2019), FDA (Nakib ve ark., 2017), vb.).

Bu tez kapsamında orijinal SSA geliştirilerek üç farklı türde optimizasyon problemi çözülmüştür. Bunlar: i) küçük, orta ve geniş ölçekli sürekli optimizasyon problemleri, ii) ayrık optimizasyon problemleri, ve iii) ikili optimizasyon problemleridir.

Bu tezde gerçek dünya problemlerinden biri olan Moleküler Potansiyel Enerji Minimizasyonu problemi (MPEF) çözülmüştür. MPEF, sürekli optimizasyon problemlerinden birisidir. Proteinlerin ve peptitlerin küresel minimum potansiyel enerji konformasyonunun saptanması, MPEF problemi olarak bilinir (Tawhid ve Ali, 2017c). Moleküler potansiyel enerji fonksiyon probleminin (MPEF) en aza indirilmesi, küresel bir optimizasyon problemi olarak formüle edilebilir. Protein içindeki moleküllerin kararlı durumunu bulmak, proteinin 3D yapısını tahmin etmeye yardımcı olabilir. Sorunun zorluğu, molekülün büyüklüğü ile katlanarak büyüyen enerji fonksiyonunun yerel minimizeörlerinin sayısından kaynaklanmaktadır (Wales ve Scheraga, 1999). Bu sorunu çözmek için literatürde birçok yöntem uygulanmıştır (örnek: HSSOGA (Tawhid ve Ali, 2017b), HGWOGA (Tawhid ve Ali, 2017c), HFPGA(Tawhid ve Ali, 2017a), vb.).

Gezgin Satıcı Problemi (TSP) ilk 1930'da matematiksel bir problem olarak ifade edilmiştir ve 1950'den sonra popülerliği artmıştır. Son yıllarda TSP, ayrık optimizasyonda yoğun olarak incelenen problemlerden biri olmuştur (Bello ve ark., 2007). Kombinasyonel optimizasyon alanında, TSP en çok çalışılan ve en çok uygulamaya sahip problemlerden biridir (Pandiri ve Singh, 2019). TSP, belirli bir şehir listesindeki her şehri ziyaret etmeye ve nihayet başlangıç şehrine dönüş yolunu olabildiğince kısa bulmaya dayalı bir problemdir (Liao ve ark., 2012). Asıl sorun, ağırlıklı bir grafta minimum maliyetle bir Hamilton yolu bulmaktır (Cinar ve ark., 2019). Algoritmanın ve anlayışın matematiksel modeli çok kolay olmasına rağmen, bu problemi etkili bir şekilde çözmek zordur. Kuzu ve ark. (2014) NP-hard problemlerinden biri olan TSP'yi en genel Stokastik optimizasyon teknikleri sınıfıyla ele almıştır. TSP, birçok optimizasyon yönteminde bir benchmark olarak kullanılmıştır (En Yakın Komşuluk Arama (NNS), Benzetimli Tavlama (SA), Tabu Arama (TS), Sinir Ağları (NN), Karınca Kolonisi Sistemi (ACS) ve Genetik Algoritma (GA)) (Wang ve ark., 2003). Çünkü TSP çok önemli problemleri modelleyebilir (Zamanlama, bilgisayar kabloları, baskılı devre kartlarının ve devrelerin otomatik delinmesi, insanların hareketi,

X-ışını kristalografisi, vb.). Günümüzde TSP'yi tartışan ve burma14, berlin52 gibi standart TSPLIB formatında kıyaslama veri kümelerine sahip birçok web sitesi vardır (Reinelt, 1995). Literatürde klasik TSP'nin amaç işlevini değiştirerek ve / veya kısıtlamaları ayırt ederek elde edilen farklı çeşitleri vardır; simetrik TSP, asimetrik TSP, kârlı TSP (Aras ve ark., 2007), vb. Karınca Kolonisi Optimizasyonu (ACO) TSP'ye uygulanan ilk meta-sezgisel algoritmalarından biridir (Akhand ve ark., 2012). Literatürde ACO dışında birçok sezgisel algoritma da TSP'deki başarılarını test etmiştir. Örnek: DPSO (Yan ve ark., 2012), DIWO (Zhou ve ark., 2015), IVNS (Hore ve ark., 2018), DWCA (Osaba ve ark., 2018), DBA (Osaba ve ark., 2016), ACO with ABC (Gündüz ve ark., 2015), DMRSA (Zhang ve Zhou, 2016), vb.

İkili Optimizasyon Problemi (BOP), kombinasyonel optimizasyon problemlerinin önemli bir sınıfını temsil eden ikili tabanlı bir problem alanı olarak gösterilmektedir (Rizk-Allah ve ark., 2019). Sürekli optimizasyonda, arama ajanları arama alanında sürekli değerler alırken, ikili optimizasyonda arama alanındaki arama ajanları $\{0, 1\}$ değerlerini alır. “0” değeri yokluğu temsil eder ve “1” değeri varlığı temsil eder. Birçok problem, ikili bir arama uzayında bu iki değer kullanılarak çözülebilmektedir. Bazı sezgisel algoritmalar sürekli arama alanlarıyla ilgili problemleri çözebilirken, bazı problemler ayırık arama alanlarına sahiptir (Kennedy ve Everhart, 1997; Mirjalili ve Lewis, 2013). BOP'nin literatürde birçok problem uygulaması mevcuttur (tesis yeri yerleşim problemleri (acil durum araçları, sağlık merkezleri ve ticari banka şubeleri, vb.) ve planlama görevleri (bütçeleme, esnek üretim sistemleri, telekomünikasyon, toplu taşıma hizmetleri ve rüzgâr türbini yerleşimi vb.)) (Beşkirli ve ark., 2018; Korkmaz ve Kiran, 2018). Ayrıca, BOP iyi bilinen NP-hard problemlerin çözümünde de kullanılır (sırt çantası problemi, kaynak tahsisi problemi, boyutsal küçülme, özellik seçimi, ağ optimizasyonu, birim taahhüdü ve hücre oluşumu, vb.) (Babaoglu ve ark., 2010; Fan ve ark., 2013; Rizk-Allah, 2014; Emary ve ark., 2016; Rizk-Allah ve Hassanien, 2018). Literatürde BOP'u çözmek için birçok geleneksel yöntem (gevşeme yöntemleri, lagrange teknikleri, dal ve sınır yöntemleri, redüksiyon şemaları ve tamsayı programlama vb.) (Rizk-Allah ve ark., 2019) önerilmiştir. Bu yöntemler düşük ölçekli problemlerde iyi performans gösterse de, büyük ölçekli problem çözümlerinde iyi performans gösterememiştir. BOP'lerinin çözümü için birçok yöntem olmasına rağmen, meta-sezgisel algoritmalara dayalı teknikler bize daha çok avantaj sağlamıştır ($\{0,1\}$ değerlerinin seçim olasılıkları, arama uzayında keşif yapma yeteneği vb.). Meta-sezgisel algoritmalara dayalı teknikler özellikle yüksek ölçekli

problemlerin çözüm süresini kısaltmıştır. Böylece gittikçe daha fazla araştırmacı son yıllarda meta-sezgisel algoritmalarla ilgilenmeye başlamıştır (Kennedy ve Everhart, 1997; Babaoglu ve ark., 2010; Wang ve ark., 2010; Wang ve ark., 2011; Mirjalili ve Lewis, 2013; Emary ve ark., 2016; Beşkirli ve ark., 2018; Korkmaz ve Kiran, 2018; Rizk-Allah ve Hassanien, 2018; Shukla ve Nanda, 2018; Rizk-Allah ve ark., 2019). Arama süreci kavramlarını korurken, sürekli bir sezgisel algoritmanın ikili versiyonunu geliştirmek için farklı yöntemler vardır (Mirjalili ve Lewis, 2013). Wang ve ark. (2010) tarafından DE nin ikili problemleri çözmek için bir olasılık tahmin operatörü geliştirmiştir. Transfer fonksiyonları ve pozisyon güncelleme kuralları kullanılarak İkili Manyetik Optimizasyon Algoritması (BMOA) ve İkili Yerçekimi Arama Algoritması (BGSA) önerilmiştir (Rashedi ve ark., 2010). İkili Uyum Arama (BHS) algoritması bir dizi uyum arama düşüncesi ve perde ayarlama kuralları kullanmıştır (Wang ve ark., 2011). Kennedy ve Everhart (1997)'de Parçacık Sürü Optimizasyonunun (PSO) ikili versiyonlarını önermişlerdir. Onlar BPSO için iki farklı bileşen kullanmışlardır (yeni bir transfer fonksiyonu ve farklı bir pozisyon güncelleme prosedürü). Rizk-Allah ve Hassanien (2018)'de Yarasa Algoritmasının (BBA) ikili versiyonunu önermişlerdir. Shukla ve Nanda (2018)'de yüksek boyutlu veri kümeleri için İkili Sosyal Örümcek Optimizasyon algoritması (BSSO) geliştirmişlerdir. Rizk-Allah ve ark. (2019)'da İkili Salp Sürüsü Algoritması'nı (BSalpSA) ve Rizk-Allah (2018)'de İkili Sinüs Kosinüs Algoritmasını (BSCA) önermişlerdir. Bu tezde BOP'lerden olan sürekli optimizasyon görevleri, özellik seçimi problemi, kapasitesiz tesis yerleşim problemi ve rüzgâr türbinleri yerleşim problemi için SSA'nın ikili versiyonu (BinSSA) üretilmiş ve test edilmiştir.

1.1. Tezin Konusu, Amacı ve Literatüre Katkısı

Bu tez kapsamında James ve Li (2015a) tarafından geliştirilmiş Sosyal Örümcek Algoritması (SSA) incelenmiştir. SSA, doğada birlikte yaşayan örümcek davranışlarının taklit edilmesi ile oluşturulmuş sezgisel, akıllı sürü tabanlı bir algoritmadır. SSA, orijinalde sürekli optimizasyon problemlerini çözmek için geliştirilmiştir. Düşük ölçekli sürekli optimizasyon problemleri üzerinde performansı test edilmiştir. Literatür incelendiğinde SSA üzerine yeterince odaklanılmadığı görülmüştür. Bu tezin amacı SSA'yı geliştirmek, sürekli, ayrık ve ikili optimizasyon problemlerini çözebilecek

şekilde tekrar yapılandırarak performansını farklı optimizasyon problemleri üzerinde de test etmektedir.

Bu kapsamda, öncelikle düşük ölçekli (10, 20 ve 30) sürekli optimizasyon problemlerindeki SSA'nın başarısı *örümcek patlaması* ve *kaşif örümcek belleği* özellikleri eklenerek geliştirilmiş ve Geliştirilmiş Sosyal Örümcek Algoritması (ISSA) önerilmiştir. Bu iki teknik sayesinde ISSA'nın keşif ve sömürü yeteneği artmıştır. *Örümcek patlaması* tekniğinde sistemin her iterasyonda en kötü uygunluk değerine sahip örümceği tespit edilmiş ve yerine rastgele yeni bir örümcek oluşturulup yer değiştirmiştir. Böylece ISSA sürekli arama uzayında yeni noktalar keşfedebilmiştir. *Kaşif örümcek belleği* ile mevcut örümcek popülasyonunun dört katı büyüklüğünde bir hafıza alanı oluşturularak buraya her iterasyonun en iyi örümcekleri yerleştirilmiştir. Her iterasyonda sistemin en iyi örümceklerini saklayan bu bellek hafızasından belli sayıdaki örümcek sisteme tekrar eklenerek ISSA'nın sömürü yeteneği geliştirilmiştir. Yapılan literatür incelemesinde SSA'nın yüksek ölçekli problemlerde başarısının test edilmediği fark edilmiştir. Gerçek dünya problemleri her zaman düşük karar değişkenlerine sahip değildir. ISSA'nın performansı yüksek ölçekli (100, 500 ve 1000) sürekli optimizasyon problemlerinde de test edilmiştir.

Daha sonra ISSA ile gerçek dünya problemlerinden biri olan Moleküler Potansiyel Enerji Minimizasyonu problemi (MPEF) çözülmüştür. MPEF, sürekli optimizasyon problemlerinden birisidir. ISSA'ya *G_{best} yakınsaması*, *sessiz örümcek*, *çaprazlama operatörü* ve *mutasyon operatörü* özellikleri eklenerek SSA geliştirilmiştir. *Çaprazlama* ve *G_{best} yakınsama* teknikleri, sürekli arama alanında ISSA'nın yerel arama kapasitesini artırırken, *mutasyon* ve *sessiz örümcek* teknikleri sürekli arama alanında ISSA'nın küresel arama kapasitesini arttırmıştır. Bu tekniklerin birleşimi sürekli arama uzayında aramayı hızlandırır ve algoritmanın makul bir sürede optimal çözüme ulaşmasına yardımcı olur.

SSA, orijinalinde sürekli optimizasyon problemlerini çözmek için geliştirilmiştir. Bu tez kapsamında SSA, ayrık optimizasyon problemlerini çözebilecek şekilde geliştirilmiş ve Ayrık Sosyal Örümcek Algoritması (DSSA) önerilmiştir. DSSA'nın performansı *yetenekli örümcek* ve *acemi örümcek* özellikleri eklenerek geliştirilmiştir. *Yetenekli örümcek* tekniği ile yerel noktalar keşfetme yeteneği, *acemi örümcek* tekniği ile yeni noktalar keşfetme yeteneği geliştirilmiştir. Ayrık optimizasyonda birçok ayrık optimizasyon problem çeşidi vardır (Gezgin satıcı problemi, sırt çantası problemi, v.b.). DSSA'nın performansı, düşük, orta ve yüksek

ölçekli TSP örnekleri üzerinde test edilmiştir. Literatürde genellikle düşük ve orta ölçekli TSP örnekleri kullanılırken bu tez kapsamında yüksek ölçekli TSP örnekleride kullanılmıştır.

İkili Optimizasyon Problemi (BOP), kombinasyonel optimizasyon problemlerinin önemli bir sınıfını temsil eden ikili tabanlı bir problem alanı olarak gösterilmektedir (Rizk-Allah ve ark., 2019). Sürekli optimizasyonda, arama ajanları arama alanında sürekli değerler alırken, ikili optimizasyonda arama alanındaki arama ajanları $\{0, 1\}$ değerlerini alır. “0” değeri yokluğu, “1” değeri ise varlığı temsil eder. Birçok problem, ikili bir arama uzayında bu iki değer kullanılarak çözülebilmektedir. Yapılan literatür incelemesinde SSA ile BOP üzerinde yeterince çalışılmadığı fark edilmiştir. Bu tez kapsamında SSA, ikili Sosyal Örümcek Algoritması (BinSSA) şeklinde ikili optimizasyon problemlerini çözebilecek şekilde tekrardan yapılandırılmıştır. Sürekli sezgisel optimizasyon algoritmaları transfer fonksiyonları aracılığıyla ikili optimizasyon algoritmalarına çevrilebilmektedir. Transfer fonksiyonları sürekli arama uzayını ikili arama uzayına dönüştürür. Literatürde çeşitli transfer fonksiyonları vardır. Bunlardan en çok kullanılanları S-şekilli, V-şekilli ve mod tabanlı olanlardır. İkili optimizasyon algoritmalarının başarılarını geliştirmek için literatürde farklı yeni aday çözüm üretim yöntemleri mevcuttur. Bunlardan en çok kullanılanları; çaprazlama, mutasyon, benzerlik ölçme ve lojik kapılardır. Literatürde birçok BOP çeşidi vardır. Örneğin özellik seçme problemi, boyut azaltma problemi, rüzgâr türbini yerleşim problemi, tesis yeri seçme problemi gibi. Yapılan literatür incelemesinde SSA ile özellik seçme problemi, rüzgâr türbini yerleşim problemi, tesis yeri seçme problemi ve sürekli görevlerin çözülmediği ve performansının test edilmediği görülmüştür. Bu tez kapsamında dört farklı BOP, BinSSA ile çözülmüştür. Her bir BOP’de farklı aday çözüm üretim yöntemi kullanılmıştır. Literatürde en çok kullanılan transfer fonksiyonlarının başarıları BinSSA ile test edilmiştir.

1.2. Tezin Organizasyonu

Bu tez çalışması beş bölümden oluşmaktadır.

İlk bölümde Sosyal Örümcek Algoritması (SSA) ile ilgili genel bilgiler verilerek tez içeriği hakkında kısa bir giriş yapılmıştır.

İkinci bölümde SSA ile tez çalışmasında kullanılan problemler için kaynak araştırması yapılarak literatürde bulunan çalışmalar açıklanmıştır.

Üçüncü bölüm materyal ve metotları içermektedir. SSA algoritması açıklanmıştır. SSA ile çözülen problemler tek tek açıklanmış ve yapılan yenilikler anlatılmıştır. SSA'nın probleme uygulanması anlatılmıştır.

Dördüncü bölümde geliştirilen SSA, farklı problem türlerine uygulandıktan sonra elde edilen sonuçlar karşılaştırılmış ve analiz edilmiştir.

Son bölümde ise tez çalışması ve sonuçlar genel olarak değerlendirilip gelecek çalışmalar için öneriler paylaşılmıştır.



2. KAYNAK ARAŞTIRMASI

Kaynak araştırması dört ana başlık altında incelenecektir. Bunlar: Sosyal Örümcek Algoritması (SSA), sürekli optimizasyonlar için sezgisel algoritmalar, ayrık optimizasyonlar için sezgisel algoritmalar ve ikili optimizasyonlar için sezgisel algoritmalar.

2.1. Sosyal Örümcek Algoritması (SSA)

Çizelge 2.1. SSA için araştırma çalışmalarını göstermektedir. Bunlardan dört tanesi yazarın tez kapsamında yayınladığı eserlerdir.

Çizelge 2.1. Literatürde SSA için araştırma çalışmaları

Algoritmalar	Açıklamalar
SSA	James ve Li (2014), Sosyal Örümcek Algoritması ile yeşil hücrese ağlar için baz istasyonu anahtarlama sorununu çözmüşlerdir.
SSA	James ve Li (2015a), Sosyal Örümcek Algoritması (SSA) literatürde ilk olarak sürekli optimizasyon görevleri için önerilmiştir.
SSA	James ve Li (2015b), Sosyal Örümcek Algoritmasının parametre duyarlılık analizini yapmıştır.
BSSA	Nguyen ve ark. (2017), 0-1 sırt çantası problemi için yeni bir ikili Sosyal örümcek (BSSA) Algoritması önermişlerdir.
SSA	James ve Li (2016), SSA tabanlı yeni bir yaklaşım geliştirerek Ekonomik Yük Dağıtımı (ELD) problemini çözmüşlerdir. ELD, güç sistemi kontrol ve işletimindeki temel bileşenlerden biridir.
SSA	Mousa ve Bentahar (2016), QoS bilinçli web hizmeti seçimini SSA yaklaşımına uyarlamışlardır.
MSSA	Elsayed ve ark. (2016), dışbükey olmayan ekonomik yük sevkiyat problemini modifiye etmişler ve SSA tabanlı bir yaklaşımla çözmüşlerdir.
SSA	El-Bages ve Elsayed (2017), SSA kullanarak statik iletim genişleme planlama problemini çözmüşlerdir.
SSA	AL-Saati ve Alabajee (2020), en iyi yazılım güvenilirliği büyüme modellerini seçmede Sosyal Örümcek Algoritması tabanlı bir yaklaşım önermişlerdir.
SSA	Elsayed ve ark. (2017), faydalı mikro şebekelerin enerji yönetimi problemini çözmek için Sosyal Örümcek Algoritmasını önermişlerdir.
CSSA	Xavier ve Annadurai (2019), bulut bilişimde yük dengesi farkında görev zamanlaması için Kaotik Sosyal Örümcek Algoritması önermişlerdir.
SSA	Sung ve ark. (2019), tiren enerjisini optimize etmek için bir Sosyal örümcek Algoritması uygulaması gerçekleştirmişlerdir.
BinSSA	Baş ve Ülker (2020b), SSA'nın ikili versiyonunu "xor" lojik kapısı ve Jaccard'ın benzerlik ölçme yöntemlerini ekleyerek geliştirmiş ve sürekli optimizasyon görevlerinde performansını test etmiştir.
BinSSA	Baş ve Ülker (2020c), SSA'nın sekiz farklı transfer fonksiyonuna göre ikili versiyonunu çaprazlama operatörü ekleyerek geliştirmiş ve bir ayrık optimizasyon problemi olan özellik seçme problemini çözmüşlerdir.
DSSA	Baş ve Ülker (2020a), SSA'nın ayrık bir şeklini önererek, TSP problemleri üzerinde performansını test etmiştir.
BinSSA	Baş ve Ülker (2020d), SSA'nın ikili versiyonunu "xor" lojik kapısı ve Dice'ın benzerlik ölçme yöntemlerini ekleyerek geliştirmiş ve UFL problemleri üzerinde performansını test etmiştir.

2.2. Sürekli Optimizasyon Problemleri için Sürü Zekası Yöntemleri

Sosyal Örümcek Algoritması (SSA) nın geliştirilmesi ve iyileştirilmesi çalışmaları ile ISSA üretildikten sonra iki farklı sürekli optimizasyon için performansı test edilmiştir. Bunlar Büyük Ölçekli Küresel Optimizasyonlar (LSGO) ve Moleküler Potansiyel Enerji Fonksiyonu problemi (MPEF)'dir.

2.2.1. Büyük Ölçekli Küresel Optimizasyon (LSGO) için sürü yöntemleri

Çizelge 2.2.'de LSGO için önceki araştırma algoritmalarından bazıları gösterilmiştir.

Çizelge 2.2. Literatürde LSGO görevleri için önceki araştırma algoritmaları

Algoritmalar	Açıklama
DEVIIC	Sayed ve ark. (2015), DE ile geliştirilen VIIC tekniği ile LSGO problemini çözmüşlerdir.
MWOA	Sun ve ark. (2018), LSGO problemlerini çözmek için değiştirilmiş bir Balina Optimizasyon Algoritması (MWOA) önermişlerdir.
CC	Trunfio ve ark. (2016), LSGO problemlerini çözmek için Kooperatif Coevolutionary (CC) yaklaşımını geliştirmişlerdir.
DE	Mauçec ve Brest (2019), LSGO'da kullanım için Diferansiyel Evrim (DE) algoritmasının bir incelemesini yapmışlardır ve CEC 2013 LSGO benchmark paketinde LSGO için DE tabanlı bir analizini sunmuşlardır.
FDA	Nakib ve ark. (2017), temel geometrik form olarak hiper küreye dayanan geometrik fraktal ayrışma yöntemini önermişlerdir.
SOMAI	Singh ve Agrawal (2016), Karesel İnterpolasyonlu Kendi Kendini Düzenleyen Bir Göç Algoritmasını (SOMAI) geliştirmişlerdir. Yalnızca 10'luk sabit nüfus büyüklüğüne sahip 100 ile 3000 arasındaki boyutlar için LSGO problemlerini çözmüşlerdir.
MCSO	Mohapatra ve ark. (2017), LSGO için Modifiye Rekabet Sürüsü Optimize ediciyi (MCSO) geliştirmişlerdir. Önerilen MCSO ile standart CEC2008 ve CEC2013 LSGO problemlerini çözmüşlerdir.
IGWO	Long ve ark. (2018), kontrol parametresinin doğrusal olmayan bir ayarlama stratejisi ve kişisel tarihsel en iyi konuma ve küresel en iyiye dayanan değiştirilmiş bir konum güncelleme denklemi ekleyerek orijinal Gri Kurt Optimize ediciyi genişleten Geliştirilmiş Gri Kurt Optimizerini (IGWO) sunmuşlardır.
ISCA	Long ve ark. (2019), yüksek boyutlu problemleri çözmek için Sinüs Kosinüs Algoritmasının (ISCA) geliştirilmiş bir versiyonunu sunmuşlardır. Yakınsamayı hızlandırmak ve yerel optimal sonuca düşmekten kaçınmak için atalet ağırlığı ekleyerek modifiye edilmiş bir pozisyon güncelleme denklemi önermişlerdir.
MGWO	Mittal ve ark. (2016), küresel mühendislik optimizasyonu için Modifiye edilmiş Gri Kurt Optimizerini (MGWO) önermişlerdir.
μ DSDE	Yildiz ve Topal (2019), LSGO problemlerini çözmek için küçük bir popülasyon büyüklüğü kullanarak Yönlü Yerel Arama (μ DSDE) algoritması ile mikro Diferansiyel Evrim algoritmasını önermişlerdir.
JOA	Sun ve ark. (2016), Ortak Operasyon Algoritmasını (JOA) geliştirmişlerdir. JOA'yı IEEE CEC 2010 özel oturumunun yirmi LSGO benchmark fonksiyonu ve dört gerçek yaşam sorunu üzerinde test etmişlerdir.

2.2.2. Moleküler Potansiyel Enerji Fonksiyonu problemi (MPEF) için sürü yöntemleri

SSA, molekülün enerji fonksiyonunun basitleştirilmiş bir amaç fonksiyonunu en aza indirmek için kullanılmıştır. Moleküler potansiyel enerji fonksiyonu problemi, bir proteinin 3D yapısını tahmin etmeye çalışır (Tawhid ve Ali, 2017a). Çizelge 2.3.'de MPEF için önceki araştırma algoritmalarından bazılarını gösterilmiştir.

Çizelge 2.3. Literatürde MPEF için önceki araştırma çalışmaları

Algoritmalar	Açıklamalar
HSSOGA	Tawhid ve Ali (2017b), Sosyal Örümcek Optimizasyonu ve Genetik Algoritmadan oluşan bir hibrit algoritma (HSSOGA) önermiş ve MPEF problemini çözmüştür.
HGWOGA	Tawhid ve Ali (2017c), MPEF için Gri Kurt Optimizeri ve Genetik Algoritmadan oluşan bir hibrit algoritma (HGWOGA) önermişlerdir. Önerilen HGWOGA algoritması üç presedüre dayanmaktadır. İlk presedürde, keşif ve sömürü süreçleri ile güçlü performansı ile gri kurt optimize edici algoritmasını uygulayarak ve ikinci presedür boyutsallık azaltımına dayanır. Son presedür, tüm popülasyonda genetik algoritma mutasyon operatörünü uygulayarak erken yakınsamayı önlemişlerdir.
HFPGA	Tawhid ve Ali (2017a), MPEF için Çiçek tozlaşması ve Genetik Algoritmadan oluşan bir hibrit (HFPGA) önermiştir.
HPSOGA	Ali ve Tawhid (2017), MPEF için geniş ölçekli optimizasyon problemleri için popülasyon bölümlenmeli, Parçacık Sürü Optimizasyonu ve Genetik Algoritmadan oluşan bir hibrit önermişlerdir.
rHYB	Barbosa ve ark. (2005), MPEF için kararlı durum gerçek kodlu bir genetik-simpleks hibrit algoritması önermişlerdir.
-	Pardalos ve ark. (1994), en yaygın kullanılan potansiyel enerji fonksiyonlarından bazılarını tanıttı ve MPEF'te kullanılan farklı optimizasyon yöntemlerini tartışmışlardır.

2.3. Ayırık Optimizasyon Problemleri için Sürü Zekası Yöntemleri

Ayrık niceliklerin optimal olarak düzenlenmesi, gruplanması, sıraya konulması veya seçilmesi ayırık optimizasyon olarak adlandırılmaktadır. Gezgin Satıcı Problemi (TSP), ayırık optimizasyon algoritmalarının performans analizinde kullanılan standart test problemlerinden biridir. TSP, satıcının şehirler arasında en kısa turu bulmaya çalıştığı iyi bilinen bir kombinasyonel ayırık optimizasyon problemidir. TSP'nin dağıtım, yönlendirme, konum belirleme, planlama ve lojistik gibi geniş bir uygulama alanı vardır. TSP, araştırmacılar tarafından uzun yıllardır incelenen NP-hard sınıfta bir problemdir (Kuzu ve ark., 2014). TSP için literatürde önerilen birçok farklı çözüm sunan algoritma vardır. Son zamanlarda, TSP için literatüre sürü temelli birçok sezgisel algoritma önerilmiştir. Tez sürecinde üretilmiş olan Ayırık Sosyal Örümcek Algoritması

(DSSA) sürü bazlı bir algoritma olduğu için literatür incelemesinde sürü yöntemlerine daha fazla odaklanılmıştır. Çizelge 2.4.'de TSP için önceki araştırma algoritmalarından bazıları gösterilmiştir.

Çizelge 2.4. Literatürde TSP problemi için önceki araştırma algoritmaları

Algoritmalar	Açıklama
DPSO	Yan ve ark. (2012), TSP'yi Ayrık Parçacık Sürüsü Optimizasyon (DPSO) algoritması ile çözmüşlerdir.
DIWO	Zhou ve ark. (2015), TSP'yi Ayrık İstilacı Ot Optimizasyon (DIWO) algoritması ile çözmüşlerdir. Bu algortmada yabancı ot bireyleri, IWO'nun normal dağılımının değişmemesi temelinde pozitif tamsayıyı kodlar ve daha sonra yabancı ot bireylerinin canlılık değerini hesaplamışlardır.
IVNS	Hore ve ark. (2018), TSP'yi çözmek için değişken komşuluk aramasını geliştirmişlerdir.
DWCA	Osaba ve ark. (2018), simetrik ve asimetrik TSP'yi çözmek için bir Ayrık Su Döngüsü Algoritması (DWCA) geliştirmişlerdir.
ACO with ABC	Gündüz ve ark. (2015), TSP'yi çözmek için sürü zekâsına dayanan hiyerarşik bir yaklaşım geliştirmişlerdir. ACO algoritması ile ABC algoritmasını hibrit şekline sunmuşlardır.
PSO-ACO-3Opt	Mahi ve ark. (2015), TSP için Parçacık Sürüsü Optimizasyonu, Karınca Kolonisi Optimizasyonu ve 3-Opt algoritmalarına dayanan yeni bir hibrit yöntem geliştirmişlerdir.
DBA	Osaba ve ark. (2016), simetrik ve asimetrik TSP için bir Ayrık Yarasa Algoritması (DBA) geliştirmişlerdir.
DMRSA	Zhang ve Zhou (2016), Gezgin satıcı problemi (TSP) için canlılık seçimini kullanan dinamik çok ölçekli bir bölge arama algoritması sunmuşlardır. Canlılık seçimi (VS), TSP için en eski silme seçimine dayanan yeni bir değişiklik şeması olarak önermişlerdir.
GSOA	Faigl (2018), TSP için kendi kendini organize eden haritaların ilkelerinden esinlenerek, Büyüyen Kendi Kendini Düzenleyen Dizi (GSOA) adı verilen yeni bir denetimsiz öğrenme presedürü sunmuştur.
CLPSO	Zhong ve ark. (2018), TSP için önerilen Benzetilmiş Tavlama Algoritmasının kabul kriterlerini kullanan ayrık kapsamlı öğrenme PSO algoritmasını (CLPSO) geliştirmişlerdir.
DSOS	TSP için en uygun çözümü bulmak için bir Ayrık Simbiyotik Organizmalar Arama (DSOS) algoritması önerilmiştir. Ezugwu ve Adewumi (2017) tarafından SOS geliştirilmiştir ve popülasyonunu yeniden yapılandırmak, keşif ve sömürü yeteneğini geliştirmek ve yakınsama hızını hızlandırmak için üç mutasyon tabanlı yerel arama operatörü kullanarak genişletmişlerdir.
IPGA	Zhou ve ark. (2018), çoklu TSP'nin çözümünde GA ve PSO'nun iyileştirilmesini sunmuşlardır.

2.4. İkili Optimizasyon Problemleri için Sürü Zekası Yöntemleri

İkili Optimizasyon Problemi (BOP), kombinasyonel optimizasyon problemlerinin önemli bir sınıfını temsil eden ikili tabanlı bir problem alanı olarak gösterilmektedir (Rizk-Allah ve ark., 2019). Sürekli optimizasyonda, arama ajanları arama alanında sürekli değerler alırken, ikili optimizasyonda arama alanındaki arama ajanları $\{0, 1\}$ değerlerini alır. BOP'nin literatürde boyut azaltma, özellik seçme, rüzgâr

türbini yerleşimi ve tesis yeri yerleşimi gibi birçok problem uygulaması mevcuttur. Bu tez kapsamında İkili Sosyal Örümcek Algoritması (BinSSA) önerilmiş ve dört farklı BOP türü incelenmiştir. Bunlar: Sürekli optimizasyon görevleri için BinSSA, özellik seçme (FS) problemi için BinSSA, kapasitesiz tesis yeri seçme (UFL) problemi için BinSSA, ve rüzgâr türbini yerleşim problemi için BinSSA'dır. BOP için literatür incelemesi de problem bazlı sürü algoritmalarına dayanarak yapılmıştır.

2.4.1. Sürekli optimizasyon görevleri için sürü yöntemleri

Bu çalışmada SSA, dört farklı transfer fonksiyonu ile ikili şekle (BinSSA) çevrilmiştir. İkili arama uzayında aday çözüm üretim şemasında “xor” lojik kapısı ve *Jaccard benzerlik yöntemi* kullanılarak performansı geliştirilmiştir ve ikili arama uzayında keşif ve sömürü dengesi sağlanmıştır. Sürekli optimizasyon problemlerinde kullanılan standart on sekiz benchmark fonksiyonu BinSSA'yı test etmek için seçilmiştir. Bu nedenle literatür incelemesinde sürekli görevler için kullanılan sürü temelli ikili algoritmalar incelenmiştir. Çizelge 2.5.'de sürekli optimizasyon görevleri için önceki araştırma algoritmalarından bazıları gösterilmiştir.

Çizelge 2.5. Literatürde sürekli optimizasyon görevi için önceki araştırma algoritmaları

Algoritmalar	Açıklamalar
BSalpSA	Rizk-Allah ve ark. (2019), İkili Salp Sürüsü Algoritmasını (BSalpSA) kullanarak sürekli optimizasyon problemlerini çözmüşlerdir.
BSCA	Rizk-Allah (2018), Sinüs Kosinüs Algoritmasının ikili versiyonunu (BSCA) önermiştir ve sürekli optimizasyon görevleri üzerinde performansını test etmiştir.
BSSO	Shukla ve Nanda (2018), Sosyal Örümcek Algoritmasının (SSO) ikili versiyonunu sıkıştırılmış hiperspektral görüntülerde denetimsiz bant seçimi için önermişlerdir.
BBA	Rizk-Allah ve Hassanien (2018), 0-1 sırt çantası problemi için ikili yarasa Algoritmasını (BBA) önermişlerdir.
BPSO	Rizk-Allah ve Hassanien (2018), 0-1 sırt çantası problemi için İkili Parçacık Sürü Optimizasyonunu (BPSO) önermişlerdir.
BHPSOWM	Jiang ve ark. (2017), dalgacık mutasyonu ile İkili Hibrit Parçacık Sürüsü Optimizasyonu (BHPSOWM) önermişlerdir.

2.4.2. Özellik seçme (FS) problemi için sürü yöntemleri

Sınıflandırma, veri madenciliğinde en yaygın görevlerden biridir (Yin ve Gai, 2015). Dünyadaki verilerdeki büyüme nedeniyle, özellik seçimi (FS) gibi bir ön işleme prosedürü veri madenciliği uygulamalarında zorlu ve temel bir görev haline gelmiştir

(Jensen, 2005). FS, veri setinden tekrarlanan özellikler ve alakasız özellikler kaldırılarak öğrenme algoritmasının performansının iyileştirmesi anlamına gelmektedir (Liu ve Motoda, 1998). Birçok FS algoritması, hesaplama süresini azaltmak için en uygun veya en uygun özellik alt kümesini bulmak için sezgisel veya rastgele arama stratejilerini içermektedir. Literatürde özellik seçimi için önerilen birçok farklı algoritma vardır. Son zamanlarda, özellik seçimi için literatürde sürü bazlı birçok algoritma da önerilmiştir. Literatür taraması BinSSA sürüsü tabanlı bir algoritma olduğundan daha çok sürü yöntemlerine odaklanmıştır. Çizelge 2.6.'da nitelik seçimi için önceki araştırma algoritmalarından bazıları gösterilmiştir.

Çizelge 2.6. Literatürde FS problemi için önceki araştırma algoritmaları

Algoritmalar	Açıklamalar
GWO	Emary ve ark. (2016), FS için Gri Kurt Optimizasyonunun bir ikili versiyonunu önermişlerdir.
BALO	Zawbaa ve ark. (2015), İkili Karınca Aslanı Optimize edicisi (BALO) kullanılarak bir FS yöntemi sunmuşlardır.
BBHA	Pashaei ve Aydın (2017), FS için İkili Kara Delik Algoritmasını (BBHA) tanıtmışlardır. BBHA, mevcut BHA'nın uygun binarizasyon yoluyla bir uzantısıdır.
MBACO	Wan ve ark. (2016), Genetik Algoritma (GA) ile birleştirilmiş, Modifiye edilmiş İkili kodlanmış Karınca Kolonisi Optimizasyon algoritmasına (MBACO) dayanan bir özellik seçim yaklaşımı tanıtmışlardır.
HBEPESO	Tawhid ve Dsouza (2018), Yarasa ve Geliştirilmiş Parçacık Sürü Optimizasyonu algoritmalarının ikili bir versiyonunu (HBEPESO) tanıtmışlardır. Arama algoritmasındaki en iyi küresel çözüme yakınsama özelliği ile yarasa algoritmasını çevreyle ilgili yerleştirme kapasitesi ve özellik alanı ile parçacık sürü optimizasyonunun geliştirilmiş versiyonuyla birleştirilmiştir.
BCROSAT	Yan ve ark. (2019), yüksek boyutlu biyomedikal veri kümelerinde özellik seçimi için simüle edilmiş tavlama ile ikili mercan resifleri optimizasyon algoritmasını birleştirerek tanıtmışlardır.
BBSO	Papa ve ark. (2018), FS için İkili Beyin Fırtınası Optimizasyonunu (BBSO) tanıtmışlardır. Gerçek değerli çözümlerin farklı aktarım işlevleri kullanılarak bir ikili hiperküp üzerine eşlemişlerdir.
ISSA	Hegazy ve ark. (2020), FS için çözüm doğruluğunu, güvenilirliğini ve yakınsama hızını artırmak için geliştirilmiş Salp Swarm Algoritması (ISSA) geliştirmişlerdir. Mevcut en iyi çözümü ayarlamak için atalet ağırlığı olan yeni bir kontrol parametresi eklemişlerdir.
BSSA	Faris ve ark. (2018), FS için bir İkili Salp Sürüsü Algoritması (BSSA) tanıtmışlardır. Arama stratejisi olarak SSA kullanan iki yeni sarıcı özellik seçim yaklaşımı önermişlerdir.
Rc-BBFA	Zhang ve ark. (2017), FS için iade maliyetine dayalı ikili FFA (Rc-BBFA) önermişlerdir. Önerilen yöntem, erken yakınsamayı önleme yeteneğine sahiptir.

2.4.3. Kapasitesiz tesis yerleşim problemi (UFLP) için sürü yöntemleri

Tesis Yerleşim Problemi (FLP), doğal ikili optimizasyon problemlerinden biridir. FLP, kamu hizmetlerinin yerleri (yurtlar, okullar, kütüphaneler, polis karakolları, hastaneler, vb.) ve özel şirket tesisleri (pazarlar, mağazalar, depolar, vb.) hakkında yerleşim sorunlarını içermektedir (Bastı, 2012). Bu problemler, genellikle kapasitesiz olarak yani tesislerin genellikle en büyük talepleri bile karşılayabilecek kapasiteye sahip olduğu düşünülmektedir (UFLP). Kapasitesiz Tesis Yerleşim (UFL) problemi, ikili optimizasyon çalışmalarında bir test problemi olarak yaygın olarak kullanılmaktadır. UFL sorunu, müşteri taleplerini minimum maliyetle karşılayabilecek tesis konumlarını belirleme zorluğunu içeren bir sorundur. Bu çalışmanın literatür incelemesinde UFL problemi için kullanılan sürü temelli ikili algoritmalar incelenmiştir. Çizelge 2.7.'de UFL problemleri için önceki araştırma algoritmalarından bazıları gösterilmiştir.

Çizelge 2.7. Literatürde UFLP problemi için önceki araştırma algoritmaları

Algoritmalar	Açıklamalar
BPSO	Kennedy ve Everhart (1997), İkili Parçacık Sürüsü Optimizasyonu (BPSO) iyileştirmişlerdir ve çeşitli ikili problemleri çözmüşlerdir.
ISS	Haklı ve Ortacay (2019), UFL sorunu için Geliştirilmiş Dağılım Arama (ISS) algoritmasını önermişlerdir. Farklı çaprazlama teknikleriyle SS'nin küresel arama yeteneğini geliştirmişlerdir.
TSA	Cinar ve Kiran (2018), BOP için bir ikili Ağaç Tohumu Algoritması (TSA) önermişlerdir. Algoritmayı geliştirmek için lojik kapı ve benzerlik ölçme tekniklerini kullanmışlardır. Onlar TSA'nın başarısını UFL problemi üzerinde test etmişlerdir.
BAAASB	Korkmaz ve Kiran (2018), BOP için stigmergic davranışa sahip bir İkili Yapay Alg Algoritmasını (BAAASB) geliştirmişlerdir. BAAASB ile UFL problemini çözmüşlerdir.
binABC	Kiran ve Gündüz (2013), ikili optimizasyon için bir İkili Arı Kolonisi Optimizasyonu önermişlerdir ve UFLP'leri çözmüşlerdir.
ABC _{bin}	Kiran (2015), Yapay Arı Kolonisi (ABC) algoritmasını ikili versiyonunu önermiş ve on beş UFL problemi üzerinde test etmiştir.
binAAA	Korkmaz ve ark. (2018), AAA'nın temel versiyonundaki aday çözüm üretme süreci, popülasyondan rastgele seçilen bir komşu çözümü ve bu çözümün üç karar değişkenini kullanan bir mekanizma ile yer değiştirmişlerdir. AAA'nın ikili versiyonu olan binAAA'yı önermişlerdir. Önerilen algoritmanın, kısaca binAAA'nın performansı, saf ikili optimizasyon problemi olan UFL problemi üzerinde incelemiştir
DUS	Ardjmand ve ark. (2014), UFLP'yi çözmek için farklı bir Bilinçsiz Arama varyantı önermişlerdir.
DPSO	Guner ve Sevkli (2008), UFL problemi için ayrık bir Parçacık Sürüsü Optimizasyonu algoritması önermişlerdir
JayaX ve JayaX-LSM	Aslan ve ark. (2019), Jaya algoritmasını xor operatörü (JayaX) ile ve yerel arama stratejisi ekleyerek (JayaX-LSM) geliştirmiştir. JayaX ve JayaX-LSM performansını UFL problemi üzerinde test etmişlerdir.

2.4.4. Rüzgâr türbini yerleşim problemi için sürü yöntemleri

Son yıllarda, fosil yakıtların azalması nedeniyle rüzgâr enerjisinin önemi artmıştır. Teknolojinin gelişmesiyle birlikte, rüzgâr üretimi için rüzgâr türbinlerinin rüzgâr çiftliği sistemine yerleştirilmesi gerekmektedir. Rüzgâr çiftliği tasarımının en önemli özelliklerinden biri, rüzgâr türbinlerinin belirli bir alandaki konumunu belirlemektir. Rüzgâr türbinleri arasındaki uyanma etkisi en aza indirilmelidir (Gao ve ark., 2015). Bu nedenle literatürde rüzgâr türbinlerinin bir rüzgâr çiftliğine yerleştirilmesi için çeşitli çalışmalar yapılmıştır. En sık kullanılan yöntemler literatürdeki optimizasyon yöntemleridir. Bu çalışmanın literatür incelemesinde rüzgâr türbinleri yerleşim problemi için kullanılan algoritmalar incelenmiştir. Çizelge 2.8.'de rüzgâr türbinleri yerleşim problemi için önceki araştırma algoritmalarından bazıları gösterilmiştir.

Çizelge 2.8. Literatürdeki rüzgâr türbinleri yerleşim problemi için önceki araştırma algoritmaları

Algoritmalar	Açıklamalar
GA	Mosetti ve ark. (1994), rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemini çözen literatürde ilk çalışmayı yapmışlardır. Rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemi için en yaygın kullanılan optimizasyon tekniklerinden biri olan genetik algoritmayı (GA) tanıtmışlardır.
GA	Grady ve ark. (2005), rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemini GA ile çözmüşlerdir.
GA	Parada ve ark. (2017), rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemini GA ile çözmüşlerdir.
GA	Emami ve Noghreh (2010), rüzgâr türbinlerinin rüzgâr çiftliği içerisine genetik algoritmalarla yerleştirilmesinde optimizasyon için yeni bir yaklaşım geliştirmişlerdir.
BAAA	Beşkirli ve ark. (2018), rüzgâr türbini yerleştirme problemini çözmek için yeni bir optimizasyon algoritması önermişlerdir. Bu algoritma, ikili bir Yapay Alg Algoritmasıdır (BAAA).
BIWO	Beşkirli ve ark. (2019), İkili Yabani ot optimizasyonu kullanarak rüzgâr türbini yerleştirme problemini çözmüşlerdir.
GA	González ve ark. (2010), rüzgâr türbinlerinin yerleşimini optimize etmek için evrimsel bir algoritma önermişlerdir.
Lazy Greedy	Changshui ve ark. (2011), Lazy Greedy isimli yeni bir yaklaşım önermişlerdir ve rüzgâr türbini yerleştirme problemini çözmüşlerdir.
MILP	Shakoor ve ark. (2016), alan boyutları ve belirli nokta seçim teknikleri kullanılarak rüzgâr çiftliği yerleşim optimizasyonunu çözmüşlerdir.
SA	Yang ve Cho (2019), Simüle edilmiş Tavlama Algoritması (SA) kullanarak rüzgâr türbinleri yerleşim problemini çözmüşlerdir.
BPSO-TVAC	Pookpunt ve Ongsakul (2013), zamanla değişen ivme katsayıları ile ikili parçacık sürüsü optimizasyonu kullanılarak rüzgâr türbinlerinin rüzgâr çiftliği içine en uygun şekilde yerleştirilmesini sağlamışlardır.

Yapılan literatür incelemelerine göre SSA'nın farklı problem türleri üzerinde yeterince performansı incelenmemiştir. SSA, sürekli optimizasyon problemlerini

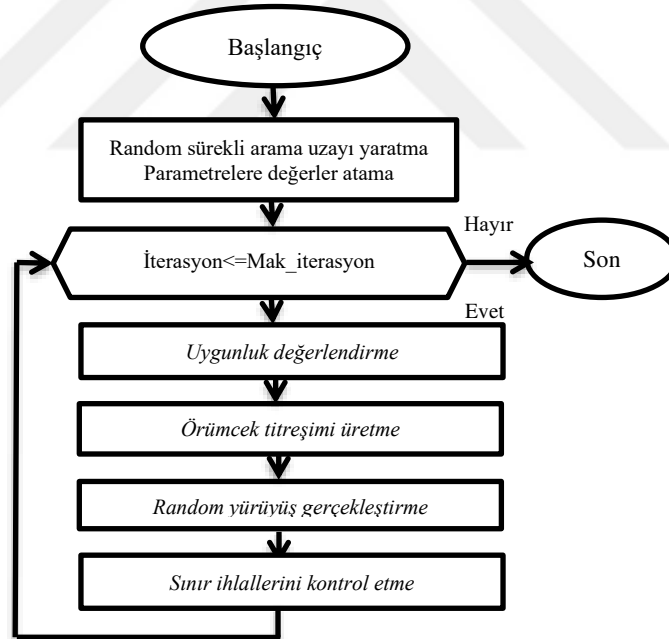
özmek için geliştirilmiştir. SSA'nın sürekli optimizasyon problemleri üzerindeki başarısı düşük ölçekli sürekli optimizasyon problemlerinde test edilmesine rağmen geniş ölçekli optimizasyon problemlerinde test edilmemiştir. Sürekli optimizasyon görevleri için geliştirilmiş olan SSA'nın ayrık optimizasyon problemleri üzerindeki başarısı TSP gibi ayrık optimizasyon problemleri üzerinde yeterince incelenmemiştir. Ayrıca ayrık optimizasyonun özel bir şekli olan ikili optimizasyon problemleri üzerinde de SSA'nın performansına yeterince odaklanılmamıştır.



3. MATERYAL VE YÖNTEM

3.1. Orijinal Sosyal Örümcek Algoritması (SSA)

James ve Li (2015a) tarafından Sosyal Örümcek Algoritması (SSA) önerilmiştir. SSA, doğada sosyal yaşayan örümceklerin yiyecek bulma davranışlarından esinlenilerek oluşturulmuş meta-sezgisel bir algoritmadır. SSA, arama uzayını bir örümcek ağı gibi tasarlamıştır. SSA'da, örümcekler optimizasyon gerçekleştiren SSA araçlarıdır ve onlar mevcut konumlarında titreşim üretirler. Titreşim, SSA'yı diğer sezgisel algoritmalarından ayırt eden en önemli özelliktir. Bu titreşim sayesinde SSA'da örümcekler arasında bilgi paylaşımı gerçekleşir. Her örümcek, diğer örümcekler tarafından üretilen sönmüş titreşim şiddetini alır ve rastgele yürüyüşünü hangi yöne doğru gerçekleştireceğine karar verir. Orijinal SSA için akış şeması Şekil 3.1.'de gösterilmiştir. Orijinal SSA'nın tüm çalışma adımları Şekil 3.2.'de gösterilmiştir.



Şekil 3.1. Orijinal SSA için akış şeması

Adım 1. Başlangıç: Parametre değerleri ayarlanır. Sürüdeki her örümcek için random başlangıç pozisyonları tespit edilir. Her örümcek için başlangıç hedef titreşim değeri atanır.

Adım 2. Uygunluk değerlendirme: Sürüdeki her örümcek için uygunluk değeri hesaplanır. Sürüdeki en iyi değer belirlenir ve kaydedilir.

Adım 3. Titreşim üretme: Sürüdeki her örümcek için titreşim değeri denklemler (3.1, 3.2 ve 3.3) kullanılarak hesaplanır. Her birey için en iyi değer belirlenir ve hedef titreşim değeri ile karşılaştırılır. Hedef titreşim değerinden daha iyi bir titreşim değeri bulunmuş ise hedef titreşim değeri güncellenir aksi halde bir değişiklik yapılmaz.

Adım 4. Random yürüyüş: Boyut maskesi güncellenir. P^{follow} değeri denklem 3.5 ile hesaplanır. Her örümcek yeni pozisyonuna denklem 3.4 ile hareket eder.

Adım 5. Sınır ihlallerini kontrol etme: İhlaller denklem 3.6 ile kontrol edilir.

Adım 6. Tekrar: Durdurma kriteri sağlanana kadar adım 2'den itibaren bütün adımlar tekrar edilir.

Adım 7. Çıktı: Optimum çıktılar elde edilir.

Şekil 3.2. Orijinal SSA'nın çalışma adımları

3.1.1. Başlangıç ve uygunluk değerlendirmesi

Gerekli sabit parametre setleri ayarlanır. Örneğin, değişkenlerin alt ve üst sınırları, maksimum yinleme sayısı, vb. SSA'da, örümceklerin konumları arama alanında rastgele oluşturulur. Örümcek popülasyonunda ki her örümcek için uygunluk değeri hesaplanır.

3.1.2. Titreşim üretme

SSA'da arama uzayındaki tüm örümcekler mevcut konumlarını kullanarak titreşim yoğunluklarını hesaplarlar. Titreşim yoğunluğu denklem 3.1 (James ve Li, 2015a) ile hesaplanmaktadır. SSA'da titreşimler yayılım sürecini aktif etmek için kullanılmaktadır. Mesafe üzerindeki titreşim zayıflama miktarı denklem 3.2 ve mesafe denklem 3.3 ile hesaplanmaktadır (James ve Li, 2015a). “V” örümcek titreşimlerini ifade etmektedir. SSA'da her örümcek diğer örümcekler tarafından üretilen titreşimleri almaktadır. Bir örümcek bu aldığı titreşimlerin en güçlüsünü seçer ve bu değeri V_x^{best} olarak gösterir. x bir örümceği ifade etmektedir. Örümcek x hafızasındaki hedef titreşimini V_x^{target} şeklinde depolar. Her örümcek x , V_x^{best} ile V_x^{target} değerini karşılaştırır. Eğer V_x^{best} daha küçükse V_x^{target} değerinden, V_x^{target} değeri V_x^{best} olarak yer değiştirir. Aksi halde herhangi bir değişiklik yapılmaz.

$$I(P_x, P_x, t) = \log\left(\frac{1}{f(P_x) - c}\right) + 1 \quad (3.1)$$

Burada $I(P_x, P_x, t)$, t zamanda x örümceğinin kaynak pozisyonunda üretmiş olduğu titreşim değeridir. x bir örümceği temsil etmektedir. $P_x(t)$, t zamanda x örümceğinin pozisyonunu tanımlar ve t zamanda basitçe P_x olarak gösterilmiştir. $f(P_x)$, bir örümceğin mevcut konumundaki uygunluk değerini ifade etmektedir. C , küçük bir güvenlik sabitidir ve minimizasyon problemleri için olası tüm uygunluk değerlerinden daha küçük bir değerdir. Logaritmlar, çok büyük veya çok küçük sayılar ile olan işlemler için kullanılır.

$$I(P_x, P_y, t) = I(P_x, P_x, t) \times \exp\left(-\frac{\text{Distance}(P_x, P_y)}{\bar{\sigma} \times r_a}\right) \quad (3.2)$$

$$\text{Distance}(P_x, P_y) = \|P_x - P_y\| \quad (3.3)$$

Burada $I(P_x, P_y, t)$, “ y ” noktasındaki bir örümceğin titreşim değerinin “ x ” noktasındaki bir örümcek tarafından hissedilme değerini ifade etmektedir. $P_x(t)$ veya basitçe P_x , t zamanda x örümceğinin pozisyonunu temsil etmektedir. $P_y(t)$ veya basitçe P_y , t zamanda y örümceğinin pozisyonunu temsil etmektedir. $I(P_x, P_x, t)$, t zamanda x örümceğinin üretmiş olduğu titreşim değeridir. $\bar{\sigma}$, her boyuttaki tüm örümceklerin konumlarının standart sapmasının ortalamasını temsil eder. r_a kullanıcı kontrollü bir parametre değeridir ($r_a \in (0, \infty)$). $\text{Distance}(P_x, P_y)$, x ve y örümcekleri arasındaki mesafeyi göstermektedir.

3.1.3. İkili maske değişimi ve random yürüyüş

SSA’da her örümcek için random yürüyüş, denklem 3.4 (James ve Li, 2015a) ile hesaplanmaktadır. x örümceği öncelikle bir önceki yinelemedeki hareket yönü olan önceki yönü boyunca hareket eder. Bu yön boyunca hareket edeceği mesafe, önceki hareketin rastgele bir kısmını oluşturur. Daha sonra x örümceği $(0, 1)$ aralığında üretilmiş rasgele bir faktör ile her boyut boyunca P_x^{follow} ’a yaklaşır. P_x^{follow} , takip edilecek pozisyonu temsil etmektedir. SSA’da her örümcek takip edilecek pozisyonuna (P_x^{follow}) boyut maskesine bakarak karar verir.

$$P_x(t+1) = P_x + (P_x - P_x(t-1)) \times r + (P_x^{\text{follow}} - P_x) \odot R \quad (3.4)$$

Burada R , $(0, 1)$ aralığında üretilen rasgele noktalı sayıların bir vektörüdür. Θ , bire bir öge çarpımı anlamına gelmektedir.

Boyut maskesi, random yürüyüş esnasında V_x^{target} 'a doğru random bir yürüyüş gerçekleşmesi için kullanılmaktadır. Boyut maskesi, n uzunluklu ikili $\{0, 1\}$ bir vektördür. n optimizasyon problem boyutunu temsil etmektedir. Her iterasyonda örümcekler boyut maskelerini değiştirmek için $1 - p_c^{cs}$ 'nin bir olasılığına sahiptir. p_c , kullanıcı kontrollü bir parametredir ($p_c \in (0, 1)$). Eğer boyut maskesinin değişimine karar verildiyse boyut maskesinin (ikili vektör) her bir biti p_m 'nin bir olasılığına sahiptir ($p_m > 0.5$ ise "1", değilse "0"). cs , örümceğin hedef titreşimini en son değiştirdiği yineleme sayısıdır. Boyut maskesi tespit edildikten sonra, yeni takip edilecek pozisyon (P_x^{follow}) maske temelli üretilir. P_x^{follow} değeri denklem 3.5 ile hesaplanmaktadır. Eğer boyut maskesinin bit değeri 0 ise, x örümceği hafızasındaki hedef örümceğin pozisyonuna doğru hareket eder, 1 ise random bir örümceğin pozisyonuna doğru hareket eder.

$$P_{x,i}^{follow} = \begin{cases} P_{x,i}^{target}, & dimension_mask_{x,i} = 0 \\ P_{x,i}^r, & dimension_mask_{x,i} = 1 \end{cases} \quad (3.5)$$

Burada r integer bir değerdir ki bu değer $[1, \text{popülasyon boyutu}]$ arasında üretilir. $dimension_mask_{x,i}$, x örümceğinin boyut maskesinin i 'inci boyutunu temsil eder.

3.1.4. Sınır ihlallerini kontrol etme

SSA'da her örümcek random yürüyüşünü gerçekleştirdikten sonra arama uzayını oluşturan ağın dışına hareket edebilir. Bu örümcekler optimizasyon problem sınırlarını çiğnerler. Denklem 3.6 (James ve Li, 2015a), sınır ihlallerini önlemek için kullanılmıştır.

$$P_{x,i}(t+1) = \begin{cases} (\overline{U}_i - P_{x,i}) \times r & \text{if } P_{x,i}(t+1) > \overline{U}_i \\ (P_{x,i} - \underline{L}_i) \times r & \text{if } P_{x,i}(t+1) < \underline{L}_i, \end{cases} \quad (3.6)$$

Burada x örümceğinin i 'inci boyutu için $\overline{U_i}$; üst sınırı ve $\underline{L_i}$; alt sınırı göstermektedir. $P_{x,i}(t+1)$, $t+1$ zamanda x örümceğinin i 'inci boyutu için pozisyon tanımlar. r ; $[1, \text{popülasyon boyutu}]$ aralığında üretilen, random bir tamsayı değeridir.

3.1.5. r_a, p_c and p_m parametre değerleri

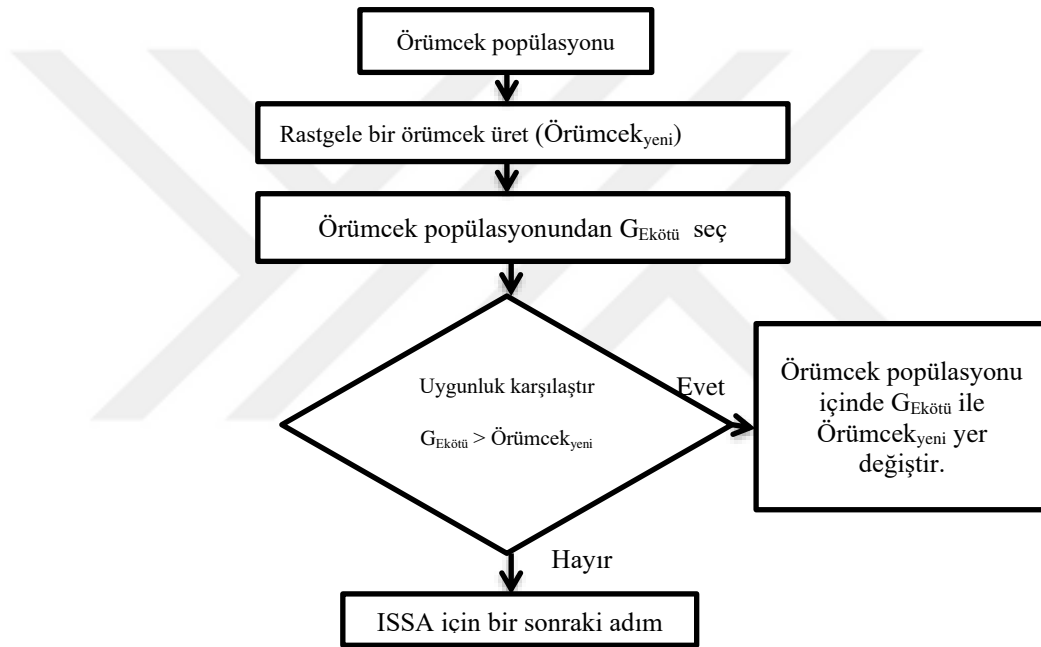
r_a , titreşim yoğunluğunun mesafe üzerindeki zayıflama oranını kontrol etmektedir. p_c , kullanıcı tanımlı bir parametre değeridir ve boyut maskesinin değişim olasılığını gösterir ($p_c \in (0, 1)$). p_m , kullanıcı tanımlı bir parametre değeridir ve boyut maskesinin bit değerlerinin belirlenmesinde kullanılır. r_a değeri $\{1/10, 1/5, 1/4, 1/3, 1/2, 1, 2, 3, 4, 5, 10\}$ değerlerinden seçilmiştir. p_c ve p_m değerleri $\{0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$ değerlerinden seçilmiştir. James ve Li (2015a) beş tane 10-boyutlu benchmark fonksiyonunu kullanarak bu parametrelerin SSA'nın performansı üzerindeki etkisini incelemiştir. Onların sonuçlarına göre r_a, p_c , ve p_m değerleri sırasıyla 1, 0.7, ve 0.1 olarak belirlenmiştir.

3.2. Sürekli Optimizasyon Görevleri için ISSA

3.2.1. Büyük Ölçekli Küresel Optimizasyon (LSGO) için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA)

Orijinal Sosyal Örümcek Algoritması (SSA), sürekli optimizasyon problemlerinin çözümü için geliştirilmiş bir sezgisel optimizasyon algoritmasıdır. Bu tez kapsamında SSA'nın çalışma adımları incelenmiş ve performansını artıran yeni iki teknik eklenerek geliştirilmiştir (ISSA). Bu teknikler örümcek patlatması ve kâşif örümcek belleği yöntemleridir. Örümcek patlatma tekniğinde, her iterasyonun en kötü uygunluk değerine sahip örümceği belirlenir ve rastgele üretilmiş yeni bir örümcek ile yer değiştirir. Böylece ISSA sürekli arama uzayında yeni noktaları keşfedebilir. Kâşif örümcek belleği tekniği ile önceki iterasyonlarda ki iyi uygunluk değerine sahip örümceklerin unutulması önlenir. Bu örümcekler mevcut iterasyonlarda değerlendirilerek, örümcek popülasyon nesillerinin daha iyi gelişmesi sağlanır. Bu teknik ISSA'nın sürekli arama uzayındaki yerel arama yeteneğinin gelişmesini sağlar.

Örümcek patlaması tekniği: Orijinal SSA’da her iterasyon boyunca sistemin en kötü uygunluk değerine sahip örümceği ($G_{Ekötü}$) saklanır. Bu örümceğin küçük de olsa optimum sonuç üzerinde etkisi vardır. Bu teknik ile ISSA her iterasyonda sistemin en kötü uygunluk değerine sahip örümceğini tespit eder ve random bir şekilde üretilmiş yeni bir örümcek ($Örümcek_{yeni}$) ile yer değiştirir. Yeni üretilen örümcek optimum sonuç üzerinde daha iyi bir etkiye sahip olabilir. Eğer üretilen yeni örümceğin uygunluk değeri sistemin en kötü örümceğinin uygunluk değerinden daha kötü ise bu yer değişim işlemi gerçekleşmez. Bu işlem her iterasyon boyunca gerçekleştirilir. Sadece bir iterasyon için örümcek patlaması tekniği şekil 3.3.’de gösterilmiştir.

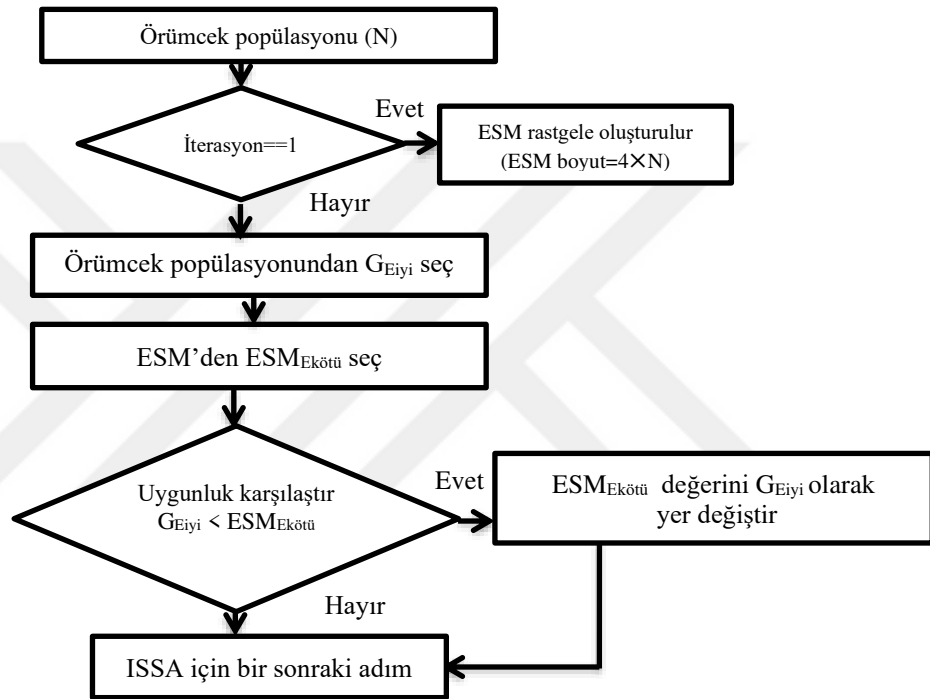


Şekil 3.3. ISSA için örümcek patlaması süreci

Kâşif örümcek belleği tekniği: ISSA’nın performansını geliştirmek için bir kâşif örümcek hafıza alanı geliştirilmiştir. Bu teknik iki adımdan oluşmaktadır. Birinci adımda kâşif örümcek hafıza alanı oluşturulur. İkinci adımda kâşif örümcek alanından bir miktar örümcek seçilir ve mevcut örümcek popülasyonuna eklenir.

Birinci adım: Sistemin başlangıcında mevcut örümcek popülasyon büyüklüğünün dört katı büyüklüğünde bir Kaşif Örümcek Hafızası (ESM) oluşturulur (N : örümcek popülasyon büyüklüğü; ESM: $4 \times N$). Başlangıçta, ESM’deki örümcekler rastgele pozisyonlarda oluşturulur. Her iterasyonda, ESM’deki en kötü uygunluk

değerine sahip örümcek ($ESM_{Ekötü}$) mevcut örümcek popülasyonunun en iyi uygunluk değerine sahip örümcek (G_{Eiyi}) ile uygunluk değerleri karşılaştırılır. Eğer G_{Eiyi} , $ESM_{Ekötü}$ 'den daha iyi bir uygunluk değerine sahipse ESM'ye eklenir ve $ESM_{Ekötü}$ ESM'den kaldırılır. Böylece ESM içindeki birey sayısı saklanmış olunur. Aksi durumda herhangi bir işlem yapılmaz. Böylece ESM, ilk iterasyondan mevcut iterasyona kadar sistemin en iyi uygunluk değerine sahip örümcek bilgilerini saklar. Sadece bir iterasyon için ESM oluşturma süreci, Şekil 3.4.' de gösterilmiştir.



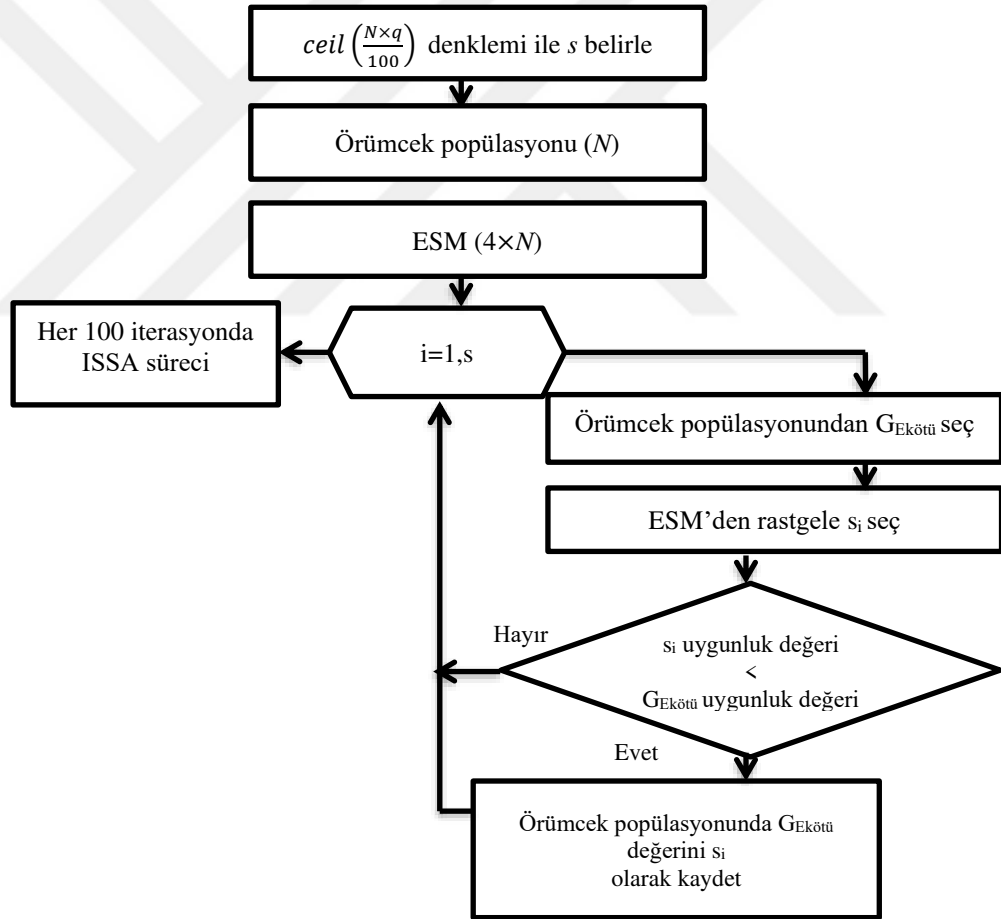
Şekil 3.4. ISSA için kaşif örümcek hafızası (birinci adım)

İkinci adım: Kaşif örümcek hafızası (ESM) birinci adımda oluşturulmuştur. İkinci adımda ise ESM'den belli bir miktar kaşif örümcek seçilir ve mevcut örümcek popülasyonuna eklenir. Başlangıçta ESM'den seçilecek örümcek sayısını gösteren oran (q) belirlenir. Bu oran yüzde olarak belirlenir ve popülasyonda çeşitliliği sağlar. Bu oranın (q) hesaplanması denklem 3.7. de gösterilmiştir.

$$s = \text{ceil}\left(\frac{N \times q}{100}\right) \quad s_i = \{s_1, s_2, \dots, s_N\} \quad (3.7)$$

Burada s ESM'den seçilecek örümcek sayısını göstermektedir. N örümcek popülasyon büyüklüğü ve q ; ESM'den seçilecek örümcek popülasyon büyüklük oranını göstermektedir. $ceil()$ ise sonuçları yukarı yuvarlayan bir matematik fonksiyonudur.

s_i , ESM'den rastgele tespit edilen kaşif örümceği temsil etmektedir. Her 100 iterasyonda s_i 'nin uygunluk değeri mevcut popülasyonun en kötü değerine sahip örümceğinin ($G_{Ekötu}$) uygunluk değeri ile karşılaştırılır. Eğer s_i 'nin uygunluk değeri $G_{Ekötu}$ 'den daha iyi ise $G_{Ekötu}$ değeri s_i olarak yer değiştirir. Aksi durumda herhangi bir değişiklik yapılmaz. Böylece ESM'den seçilen s miktar kaşif örümcek, mevcut popülasyon bireylerinin en kötü bireyi ile karşılaştırılıp, mevcut popülasyon bireylerinin daha iyi hale gelmesi sağlanmıştır. Her 100 iterasyonda s_i kaşif örümcek ekleme süreci, Şekil 3.5.' de gösterilmiştir.



Şekil 3.5. ISSA için kaşif örümcek hafızası (ikinci adım)

Şekil 3.6.'da ISSA için sözde-kod gösterilmiştir. Şekil 3.7.'de ISSA için akış şeması gösterilmiştir.

```

1:  ISSA için parametre değerleri ata
2:  Örümcek popülasyonunu yarat (  $N$  )
3:  Kaşif örümcek hafızası yarat (ESM) ( $4 \times N$ ) ve her örümcek için random bir pozisyon üret
4:  Başlat her örümcek için  $V_x^{target}$ 
5:  while durdurma kriteri sağlanana kadar do
6:    for each popülasyon içindeki örümcek  $x$  do
7:      örümcek  $x$ 'in uygunluk değerini hesapla
8:      örümcek  $x$ 'in pozisyonunda bir titreşim değeri üret
9:    end for
10:   for each popülasyon içindeki örümcek  $x$  do
11:     Bütün örümcekler tarafından üretilen  $V_{örümcek}$  titreşim yoğunluklarını hesapla
12:      $V_{örümcek}$  'den en güçlü titreşimi  $V_x^{Eiyi}$  seç
13:     if  $V_x^{Eiyi}$ 'nin yoğunluğu  $V_x^{target}$  daha büyükse then
14:        $V_x^{target} \leftarrow V_x^{Eiyi}$  olarak kaydet
15:     end if
16:   Güncelle  $cs$ 
17:    $[0,1)$ 'den rastgele bir  $r$  sayısı üret
18:   if  $r > p_c^{cs}$  then
19:     Boyut maskesi güncelle
20:   end if
21:   Üret  $P_x^{follow}$ 
22:   Her örümcek için random bir yürüyüş gerçekleştir

23:   Popülasyondan  $örümcek_{Ekötu}$  seç
24:   Yeni bir örümcek üret ( $örümcek_{yeni}$ )
25:   if  $örümcek_{yeni}$ 'nin uygunluk değeri <  $örümcek_{Ekötu}$ 'nün uygunluk değeri then
26:      $örümcek_{Ekötu}$ 'yü  $örümcek_{yeni}$  olarak güncelle
27:   end if

28:   ESM'den  $ESM_{Ekötu}$  ve popülasyondan  $örümcek_{Eiyi}$  seç
29:   if  $örümcek_{Eiyi}$  uygunluk değeri <  $ESM_{Ekötu}$  uygunluk değeri then
30:      $ESM_{Ekötu}$  'yü  $örümcek_{Eiyi}$  olarak güncelle
31:   end if

32:    $s$  sayısını belirle
33:   ESM'den  $s$  örümcekleri ve popülasyonun  $örümcek_{Ekötu}$  seç
34:   for each ESM'de  $s$  örümcekleri do
35:     if  $s_i < örümcek_{Ekötu}$  then
36:        $örümcek_{Ekötu}$  'yü  $s_i$  olarak değiştir
37:     end if
38:   end for

39:   Sınır ihlallerinin kontrolü
40:   end for
41: end while
42:  En iyi sonuçların çıktısı

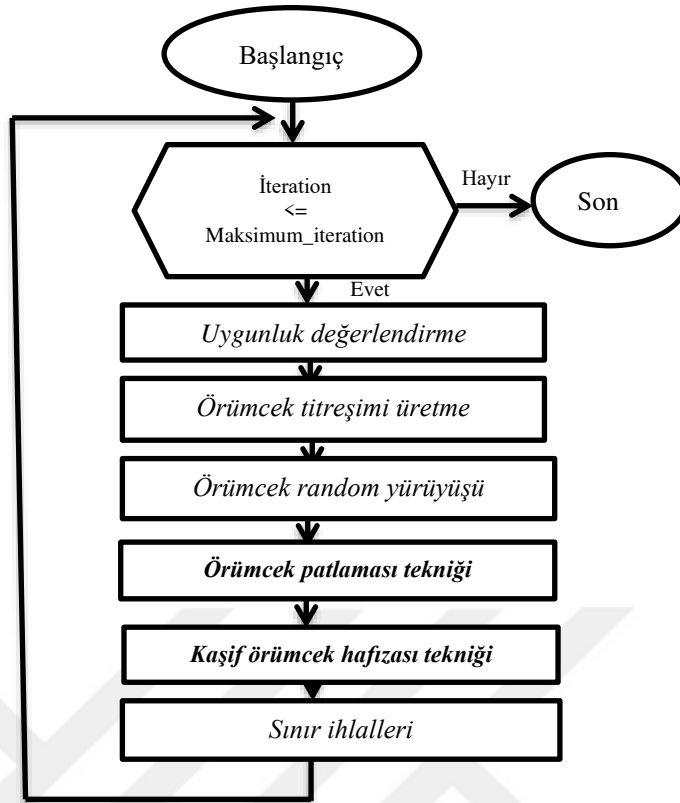
```

Örümcek Patlama Tekniği

Kaşif Örümcek Hafızası Tekniği 1.adım

Kaşif Örümcek Hafızası Tekniği 2.adım

Şekil 3.6. ISSA için sözde-kod



Şekil 3.7. ISSA için akış şeması

3.2.2. Moleküler Potansiyel Enerji Fonksiyonu (MPEF) problemi için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA)

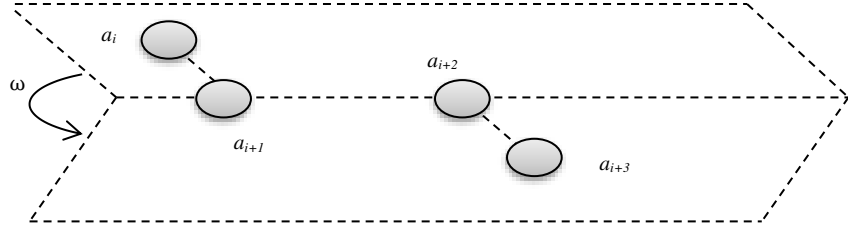
Moleküler Potansiyel Enerji Fonksiyonu (MPEF):

ISSA'nın sürekli optimizasyon problemlerinde performansını incelemek için ikinci bir çalışmada Moleküler Potansiyel Enerji Fonksiyonu (MPEF) tercih edilmiştir. MPEF ilk olarak (Lavor ve Maculan, 2004)'te önerilmektedir. Bir molekülün potansiyel enerjisi, Newton fiziğinin prensiplerine dayanan moleküler etkileşimleri tanımlayan moleküler mekanikten türetilmiştir (Tawhid ve Ali, 2017b).

Moleküler model, üç boyutlu bir uzayda bir k atom zincirinden ($a_1, a_2, \dots, a_k$) oluşur. Literatürde de benzer modeller bulunmaktadır (Hedar ve ark., 2011). Bu tezde potansiyel enerjiyi burulma açılarının bir fonksiyonu, denklem 3.8. 'de olduğu gibi tanımlanmıştır.

$$E = \sum_i \left(1 + \cos(3\omega_{i,i+3}) + \frac{(-1)^i}{\sqrt{10.60099896 - 4.141720682(\cos(\omega_{i,i+3}))}} \right) \quad (3.8)$$

Burada $i = 1, \dots, k-3$ ve k verilen atomların sayısıdır. $\omega_i, i+3$; a_i, a_{i+1}, a_{i+2} ve $a_{i+1}, a_{i+2}, a_{i+3}$ atomlarının burulma (torsion) açısıdır. Şekil 3.8.'de burulma açısı (torsion) gösterilmiştir.



Şekil 3.8. Normal düzlemler arasındaki a_i, a_{i+1}, a_{i+2} , ve $a_{i+1}, a_{i+2}, a_{i+3}$ atomlarının burulma açısı

Problemın amacı $\omega_{1,4}, \omega_{2,5}, \dots, \omega_{k-3,k}$ ($\omega_{i,j} \in [0, 5]$) bulmaktır. Problem basit görünmesine rağmen çözülmesi oldukça zor bir problemdir. Bir molekül en az 20 atomdan oluşur ki bu da $2^{17} = 131.072$ yerel minimizer bulunması demektir.

Bu tezde E, büyük ölçekli kısıtsız global optimizasyon problemi için $f(x)$ olarak yeniden yazılmıştır. Denklemler 3.9 - 3.11'a kadar uygunluk fonksiyonunu göstermektedir.

$$f(x) = \sum_{i=1}^n \left(1 + \cos(3x_i) + \frac{(-1)^i}{\sqrt{10.60099896 - 4.141720682(\cos(x_i))}} \right) \quad (3.9)$$

$$\text{ve} \quad 0 \leq x_i \leq 5, \quad i = 1, \dots, n \quad (3.10)$$

$$\min f(x), \quad x \in \mathbb{R}^n \quad l_b \leq x_i \leq u_b \quad i = 1, 2, \dots, n \quad (3.11)$$

Burada $f: \mathbb{R}^n \rightarrow \mathbb{R}$ bir amaç fonksiyonu ve x bir karar vektör değişkenini temsil eder. l_b ve u_b karar değişkenlerinin (x_i) alt ve üst sınırlarını sırasıyla göstermektedir.

MPEF problemi için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA):

Orijinal SSA, sürekli optimizasyon problemlerinin çözümü için geliştirilmiş bir optimizasyon algoritmasıdır. Bu tezde SSA incelenmiş, performansını artıran yeni dört

özellik eklenerek geliştirilmiştir ve ISSA önerilmiştir. Bu özellikler *çaprazlama*, *mutasyon*, G_{best} *yakınsaması* ve *sessiz örümcek* özellikleridir. Her dört özellikte de, ISSA sürekli arama alanında yerel ve küresel arama yapabilme yeteneğini geliştirilmiştir. Çaprazlama ve G_{best} yakınsama özellikleri, sürekli arama alanında ISSA'nın yerel arama kapasitesini artırırken, mutasyon ve sessiz örümcek teknikleri sürekli arama alanında ISSA'nın küresel arama kapasitesini arttırmıştır.

İdeal çaprazlama oranı (cr), mutasyon oranı (mr) ve G_{best} yakınsama oranını (gr) belirlemek için, popülasyon boyutu = 25, boyut = 30 ve maksimum yinleme = 1000 parametre değerleri sabit seçilerek, çaprazlama oranı (cr), mutasyon oranı (mr) ve G_{best} yakınsama oranını (gr) rastgele seçilen beş farklı tek modlu karşılaştırma fonksiyonunda ve rastgele seçilen beş farklı çok modlu karşılaştırma fonksiyonunda öngörülen bazı kombinasyonlar şeklinde test edilmiştir. Elde edilen sonuçlar tezin deneysel bölümünde gösterilmiştir.

Çaprazlama-mutasyon teknikleri:

ISSA, çaprazlama ve mutasyon operatörleri ile yerel nokta etrafında sıkışıp kalmadan kaçınır. Çaprazlama oranı (cr) ve mutasyon oranı (mr), { %10, %20, %30, %40, %50, %60, %70, %80, %90 } değerlerinden seçilmiştir. En uygun cr değeri %90 ve en uygun mr değeri %10 olarak belirlenmiştir. N popülasyon boyutunu, n ise problem boyutunu göstermek üzere şekil 3.9. çaprazlama sürecini; şekil 3.10. ise mutasyon sürecini göstermektedir.

Çaprazlama sürecinde her popülasyon bireyi ile rastgele üretilmiş bir örümceğin boyut değerleri çaprazlanır. Bu çaprazlama işleminde cr oranına göre belirlenen sayıdaki boyut değeri popülasyon bireyinden kalan boyutlar ise random üretilmiş örümcekten kopyalanarak yeni bir aday örümcek elde edilir. Bu üretilen yeni aday örümceğin mr oranına göre belirlenmiş boyut sayısı random olarak değiştirilir. Böylece yeni üretilen birey mutasyon sürecinden geçer ve çeşitlilik sağlanır. Üretilmiş yeni aday bireyin gr oranına göre tespit edilmiş miktardaki boyut bitleri (random seçilir) sistemin en iyi örümceğinin boyut değerleri ile yer değiştirir. Böylece üretilen yeni aday örümceğin lokal arama yeteneği de geliştirilmiştir. Üretilen her aday örümcek mevcut popülasyonun en kötü amaç fonksiyon değerine sahip örümcek ile karşılaştırılır ve üretilen yeni aday örümcek daha iyi ise yer değiştirir değilse mevcut popülasyon bireyleri korunur.

$n=10$, $cr=90$, $\text{Round}(n*90)/100=9$ (Rastgele 1 boyut için 9 bit seçilir.)

$N=1$.örümcek

0,040	0,025	0,041	0,751	0,325	0,472	0,254	0,324	0,010	0,475
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Random şekilde üretilen örümcek

0,750	0,460	0,120	0,745	0,135	0,745	0,472	0,042	0,530	0,635
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------



Çaprazlama sürecindeki N_{yeni} örümcek

0,040	0,025	0,041	0,751	0,135	0,472	0,254	0,324	0,010	0,475
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Şekil 3.9. Çaprazlama süreci

$n=10$, $mr=10$, $\text{Round}(n*10)/100=1$ (Rastgele 1 boyut için 1 bit seçilir.)

Çaprazlama sonrası üretilen N_{yeni} örümcek

0,040	0,025	0,041	0,751	0,135	0,472	0,254	0,324	0,010	0,475
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------



Mutasyon sürecindeki N_{yeni} örümcek

0,040	0,025	0,041	0,751	0,135	0,472	0,003	0,324	0,010	0,475
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Şekil 3.10. Mutasyon süreci

G_{best} yakınsaması tekniği:

G_{best} yakınsaması (gr) tekniği { %10, %20, %30, % 40, %50, %60, %70, %80, %90} değerlerinden seçilmiştir. En uygun gr değeri %40 şeklinde belirlenmiştir. G_{best} yakınsaması, ISSA'nın sürekli arama uzayında yerel olarak arama yapma yeteneğini geliştirmiştir. N popülasyon boyutunu, n ise problem boyutunu göstermek üzere şekil 3.11., G_{best} yakınsaması (gr) sürecini göstermektedir.

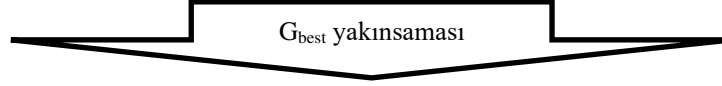
$n=10$, $gr=40$, $\text{Round}(n*40)/100=4$ (Rastgele 1 boyut için 4 bit seçilir.)

Popülasyon içindeki en iyi değeri tutan örümcek

0,003	0,035	0,012	0,032	0,140	0,001	0,042	0,230	0,023	0,035
-------	--------------	-------	-------	--------------	-------	--------------	-------	--------------	-------

Mutasyona uğramış N_{yeni} örümcek

0,040	0,025	0,041	0,751	0,135	0,472	0,003	0,324	0,010	0,475
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------



G_{best} yakınsaması sürecindeki N_{yeni} örümcek

0,040	0,035	0,041	0,751	0,140	0,472	0,042	0,324	0,023	0,475
-------	--------------	-------	-------	--------------	-------	--------------	-------	--------------	-------

Şekil 3.11. G_{best} yakınsaması süreci

Sessiz örümcek tekniği:

Orjinal SSA'da sistemin en kötü örümceği (sessiz örümcek) her iterasyon boyunca saklanır. Bu örümcek küçüktü olsa optimum sonuç üzerinde bir etkisi vardır. ISSA'da sessiz örümcek G_{best} yakınsaması sonrası üretilen yeni aday örümcek ile karşılaştırılır. Yeni üretilen örümceğin optimum sonuç üzerinde daha etkili olma olasılığı mevcuttur. Yeni üretilen aday örümceğin uygunluk değeri sessiz örümceğin uygunluk değeriyle karşılaştırılarak eklenir. Eğer sessiz örümcek daha kötü bir uygunluk değerine sahipse yok edilir ve yeni aday örümcek eklenir, değilse her hangi bir değişiklik olmaz. Böylece sistem var olan popülasyon sayısını korur. ISSA'da her iterasyon boyunca bu teknik uygulanır. Şekil 3.12.'de sessiz örümcek süreci gösterilmiştir.

Şekil 3.13.'de ISSA için akış şeması ve şekil 3.14.'de ISSA için sözde kod gösterilmiştir.

$N=5, n=10$

Mevcut örümcek popülasyonu ($N \times n$)(5 x 10)

0,210	0,225	0,121	0,750	0,150	0,470	0,033	0,324	0,085	0,765
0,130	0,423	0,751	0,750	0,075	0,488	0,003	0,924	0,910	0,495
0,045	0,756	0,360	0,790	0,130	0,493	0,883	0,384	0,011	0,405
0,047	0,120	0,860	0,420	0,233	0,052	0,759	0,378	0,750	0,485
0,052	0,960	0,320	0,360	0,186	0,402	0,753	0,234	0,086	0,755

$N=3$ (Popülasyonun en kötü örümceği) (sessiz örümcek)

0,045	0,756	0,360	0,790	0,130	0,493	0,883	0,384	0,011	0,405
--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------	--------------

G_{best} yakınsaması sonrası üretilen N_{yeni} örümcek

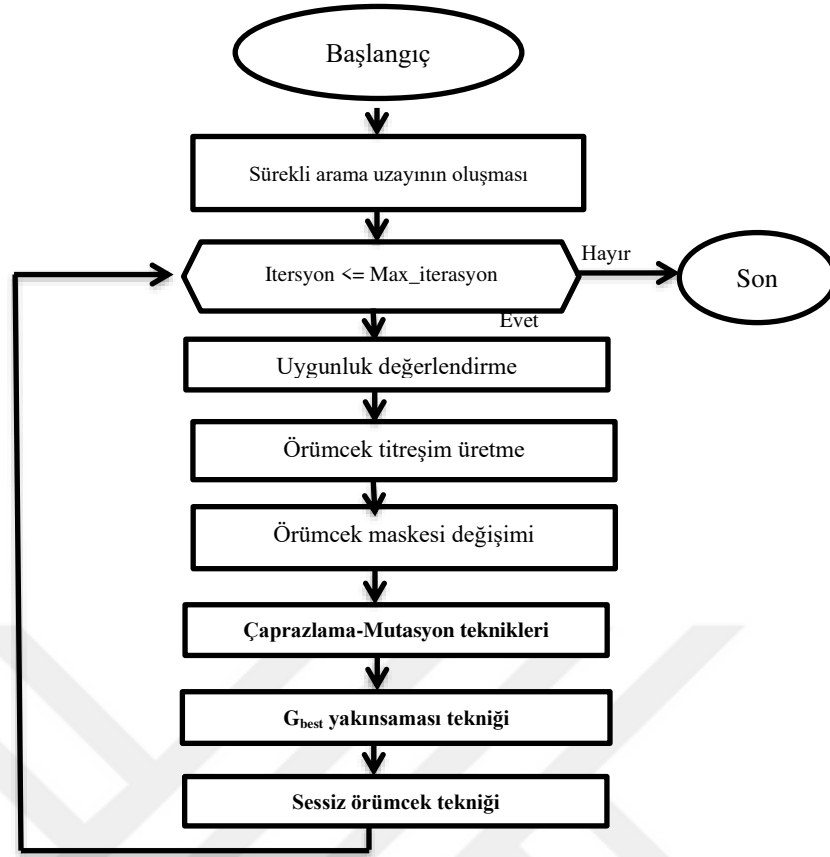
0,040	0,035	0,041	0,751	0,140	0,472	0,042	0,324	0,023	0,475
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Sessiz örümcek (karşılaştırma yapılır)

Yeni örümcek popülasyonu ($N \times n$)(5 x 10)

0,210	0,225	0,121	0,750	0,150	0,470	0,033	0,324	0,085	0,765
0,130	0,423	0,751	0,750	0,075	0,488	0,003	0,924	0,910	0,495
0,040	0,035	0,041	0,751	0,140	0,472	0,042	0,324	0,023	0,475
0,047	0,120	0,860	0,420	0,233	0,052	0,759	0,378	0,750	0,485
0,052	0,960	0,320	0,360	0,186	0,402	0,753	0,234	0,086	0,755

Şekil 3.12. Sessiz örümcek süreci



Şekil 3.13. ISSA için akış şeması

```

1:  ISSA'nın parametre değerlerinin belirlenmesi
2:  Örümcek popülasyonu yaratma
3:  Her örümcek için  $V_x^{target}$  başlatma
4:  while durdurma kriterine ulaşamadığı sürece do
5:    Bütün popülasyon üzerinde SSA adımlarını uygula
6:    for each popülasyonda her örümcek  $x$  için do
7:      random  $r$  örümceğini yarat
8:      Çaprazlama sürecini  $x$  ve  $r$  örümceklere uygula ve yeni örümcek elde et (örümcekyeni1)
9:      Mutasyon sürecini örümcekyeni1 üzerinde uygula ve yeni örümcek elde et (örümcekyeni2)
10:     Gbest yakınsama sürecini GEiyi ile örümcekyeni2 örümceğine uygula ve yeni örümcek elde
11:     et(örümcekyeni3)
12:     Popülasyondan sessiz örümceği seç
13:     if örümcekyeni3 uygunluk değeri < sessiz örümceğin uygunluk değeri then
14:       Sessiz örümceği örümcekyeni3 şeklinde güncelle
15:     end if
16:   end for
17:   Sınır ihlallerini kontrol et
18: end while
19:  En iyi çıktı çözümlerini
  
```

Şekil 3.14. ISSA için sözde-kod

3.3. Ayırık Optimizasyon için Ayırık Sosyal Örümcek Algoritması (DSSA)

Bu alt bölümde, sürekli optimizasyon problemlerini çözmek için oluşturulmuş SSA, ayırık problemleri çözebilmesi için yeniden düzenlenmiştir. Yeni geliştirilen SSA'ya Ayırık Sosyal Örümcek Algoritması (DSSA) ismi verilmiştir. DSSA'nın, literatürde sıklıkla test için kullanılan Gezgin Satıcı Problemi (TSP) üzerinde performansı ölçülmüştür.

3.3.1. DSSA için TSP tasarımı

DSSA'da, örümcekler, pozitif tamsayı olarak popülasyonunu yeniden yapılandırarak kodlarlar. Bu tamsayılar şehirleri temsil ederler. DSSA'nın başında, popülasyon boyutu (N) ve şehir turu rotasının uzunluğu ($n=problem\ boyutu$) tanımlanır. Popülasyon boyutu ve şehir turlarının rota boyutu TSP türüne göre tespit edilir. Şehir koordinat bilgileri her boyutta tutulur. Başlangıçta, her örümcek için rastgele bir şehir turlarının rotası oluşturulur.

Örneğin burma 14 veri seti için; $n=14$, $N=30$ olsun. Sonuç olarak 30×14 ($N \times n$) bir matris elde edilir. TSP'de gezgin satıcı bir şehirde sadece bir kez ziyaret edebilir ve ilk başladığı noktaya geri dönmelidir. Şehir turu güzergâhları bu bilgiler dikkate alınarak oluşturulmuştur. DSSA'da, uygunluk (fitness) fonksiyonu şehir turu rotasının toplam mesafesi şeklinde belirlenmiştir. Şekil 3.15.'de burma14 veri seti için örümcek popülasyon tasarımı gösterilmiştir.

	1	2	.	.											n
1	8	1	11	10	6	12	3	7	13	9	4	2	14	5	
2	8	2	13	10	4	11	3	7	12	9	6	5	14	1	
.	7	1	14	3	6	13	10	8	12	9	4	2	11	5	
.	5	7	12	10	3	14	6	1	13	4	9	2	11	8	
N	10	1	14	8	6	11	3	7	12	5	4	2	13	9	

Şekil 3.15. Burma 14 TSP veri seti için örümcek popülasyon tasarımı (5×14)

3.3.2. DSSA için yetenekli örümcekler tekniği

Her şehir turu için; titreşim sönme değerleri hesaplanır ($N \times n$ matrisi kullanılır). Titreşim değerlerine göre, şehirlerin mesafeleri tahmin edilmeye çalışılmaktadır. Orijinal SSA'da, her örümcek diğer örümceklere titreşim göndererek en yakın örümcekleri tespit etmeye çalışır. Örümcek, en yakın örümcek için rastgele bir eylem gerçekleştirir. Bu yaklaşımı kullanarak şehir turları için bir titreşim matrisi oluşturulur. Bu titreşim matrisine göre, şehirler arasındaki mesafelere göre en büyüğünden en küçüğe sıralanır ve minimum uzunlukta şehir turları bulunmaya çalışılır. Bu şehir turları küresel optimal şehir turları değildir, ancak bu şehir turları mevcut iterasyonun en iyi sonuçlarıdır. Bu şehir turlarını tutan örümcekler “yetenekli örümcekler” olarak adlandırılmıştır. Rastgele seçilen mevcut örümcek popülasyonuna rastgele seçilen yetenekli örümceklerin eklenmesi DSSA'nın yerel arama yeteneğini geliştirmiştir. Şekil 3.16., burma14 veri seti için yetenekli örümceklerin mevcut örümcek popülasyonuna eklenmesini göstermektedir.

Burma 14 TSP veri seti için yetenekli örümcekler ($N \times n$)

	1	2	.	.										n
1	6	10	11	8	1	12	3	7	13	9	4	2	14	5
2	10	14	13	2	4	11	3	7	12	9	6	5	8	1
.	13	1	14	7	6	3	10	8	12	9	4	2	11	5
.	1	7	12	10	3	14	6	4	13	5	9	2	11	8
	5	1	14	8	6	11	3	7	12	10	4	2	13	9
	2	10	11	8	1	12	3	7	13	9	4	6	14	5
	7	14	13	10	4	11	3	1	12	9	6	5	8	2
	9	3	14	7	6	13	10	1	12	8	4	2	11	5
	11	7	12	10	3	14	6	4	13	5	1	2	9	8
	14	1	10	8	6	11	3	7	12	5	4	2	13	9
	3	14	13	10	4	12	11	7	12	9	6	5	8	1
	4	1	14	7	6	13	10	8	3	9	12	2	11	5
	8	7	12	10	3	14	6	1	4	5	9	2	11	13
N	12	1	5	8	6	11	3	7	14	10	4	2	13	9

En iyi yetenekli örümceği seç



Burma 14 TSP veri seti için örümcek popülasyonu (5×14)

	1	2	.	.										14
1	8	1	11	10	6	12	3	7	13	9	4	2	14	5
2	8	2	13	10	4	11	3	7	12	9	6	5	14	1
.	7	14	13	10	4	11	3	1	12	9	6	5	8	2
.	5	7	12	10	3	14	6	1	13	4	9	2	11	8
N	10	1	14	8	6	11	3	7	12	5	4	2	13	9

Şekil 3.16. Bir yetenekli örümceğin mevcut örümcek popülasyonuna eklenmesi

3.3.3. DSSA için acemi örümcek tekniği

Rasgele şehir turları şeklinde üretilen örümceklere “*acemi örümcekler*” denir. Her iterasyonda üretilen acemi örümcek sayısı, mevcut popülasyon büyüklüğünün yarısı kadardır. Global en iyi değer her bir iterasyonda değişmezse, popülasyondaki en kötü örümcek belirlenir ve en iyi acemi örümcek ile yer değiştirir. DSSA’da acemi örümceklerin üretilmesinin amacı, yerel çözüm tuzaklarından kaçınmak ve sistemde farklı iyi veya kötü sonuçlarında denenmesini sağlamaktır. DSSA’nın keşif yeteneği acemi örümceklerle geliştirilmiştir. Şekil 3.17., burma14 veri seti için acemi örümceklerin mevcut örümcek popülasyonuna eklenmesini göstermektedir.



Şekil 3.17. Bir acemi örümceklerin mevcut örümcek popülasyonuna eklenmesi

3.3.4. DSSA için 2-OPT arama algoritması

2-Opt arama algoritması, TSP'yi çözmek için basit bir yerel arama algoritmasıdır. 2-Opt algoritması k-opt algoritmasının özel bir halidir (Mahi ve ark., 2015). Bu algortmada, 2-Opt analizi bir ağ (veya tur) içindeki iki bağlantıyı (veya kenarları) silmeyi, ağı diğer tüm olası yollarla yeniden bağlamayı ve ardından en uygun olanı bulmak için her yeniden bağlanma yöntemini değerlendirmeyi içerir. Bu tezde, yerel minimum durumları ortadan kaldırmak için önerilen yöntem (DSSA) 2-Opt arama yöntemi eklenmiştir. DSSA’nın son aşamasında, 2-Opt algoritması çözümü

geliştirmek için global en iyi çözüme uygulanmıştır. Şekil 3.18., DSSA'nın sözde kodunu ve şekil 3.19., DSSA'nın akış şemasını göstermektedir.

DSSA için yalancı kod

```

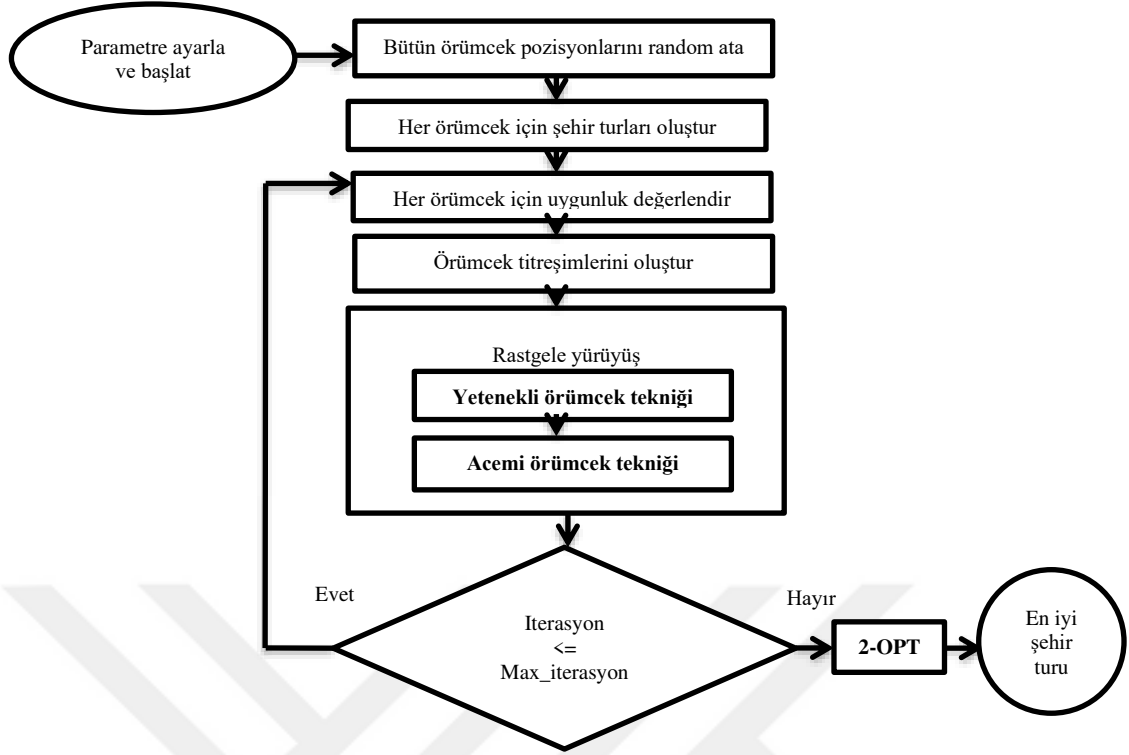
1: DSSA için parametre değerlerini ata
2: Sabit değişken ve parametre değerlerini belirle
3: Her şehir için şehir koordinat bilgilerini ata
4: Her örümcek  $x$  için rastgele şehir turları oluştur
5: while durdurma kriteri sağlanana kadar do
6:     for each popülasyon içindeki örümcek  $x$  do
7:         Örümcek  $x$ 'in uygunluk değerini değerlendir
8:         Her örümcek  $x$  için  $V_x^{şehir}$  başlat
9:     end for
10:    for each popülasyon içinde yetenekli örümcek do
11:        Bütün örümcekler tarafından üretilen  $V$  titreşim yoğunluklarını hesapla
12:        Şehirleri titreşim yoğunluğa göre sırala (yetenekli örümcekler)
13:    end for
14:    if iterasyon  $\geq 1$  and iterasyon  $\leq 5000$  then
15:        Popülasyondaki örümcekler  $\times 0,05$ 'ini random (yetenekli örümcek)  $\times 0,05$  değiştir
16:    else if iterasyon  $\geq 20000$  and iterasyon  $\leq 25000$  then
17:        Popülasyondaki örümcekler  $\times 0,05$ 'ini random (yetenekli örümcek)  $\times 0,05$  değiştir
18:    else if iterasyon  $\geq 40000$  and iterasyon  $\leq 50000$  then
19:        Popülasyondaki örümcekler  $\times 0,05$ 'ini random (yetenekli örümcek)  $\times 0,05$  değiştir
20:    else if iterasyon  $\geq 65000$  and iterasyon  $\leq 75000$  then
21:        Popülasyondaki örümcekler  $\times 0,05$ 'ini random (yetenekli örümcek)  $\times 0,05$  değiştir
22:    else if iterasyon  $\geq 95000$  and iterasyon  $\leq 100000$  then
23:        Popülasyondaki örümcekler  $\times 0,05$ 'ini random (yetenekli örümcek)  $\times 0,05$  değiştir
24:    end if
25:    if the örümcekEiyi  $\geq$  örümcekEiyiglobal then
26:        Random acemi örümcekleri yarat
27:        En iyi acemi örümceği tespit et (acemiEiyi)
28:        if acemiEiyi  $\leq$  örümcekEkötü then
29:            örümcekEkötü 'yü acemiEiyi olarak ata
30:        end if
31:    end if
32: end while
33: Global en iyi sonuca 2-Opt algoritmasını uygula
34: En iyi şehir turlarının çıktısı.

```

Yetenekli örümcek tekniği

Acemi örümcek tekniği

Şekil 3.18. DSSA için detaylı sözde-kod



Şekil 3.19. DSSA için akış şeması

3.4. İkili Optimizasyon için İkili Sosyal Örümcek Algoritması (BinSSA)

İkili optimizasyon, ayrık optimizasyon tekniklerinin özel bir şeklidir. Sürekli gerçek arama alanlarına sahip algoritmalar, değişkenleri ikili değişkenlere dönüştürerek İkili Optimizasyon Problemlerini (BOP) çözebilir. BOP türüne bakılmaksızın, ikili arama alanının bazı sınırlamalarla kendine özgü bir yapısı vardır. İkili bir arama alanı hiper küp olarak kabul edilebilir. İkili bir optimizasyon algoritmasının ajanları (örümcekleri), çeşitli sayıdaki bitleri çevirerek hiper küpün daha yakın ve daha uzak köşelerine kayabilir (Kennedy ve Everhart, 1997). Bu nedenle, SSA'nın ikili versiyonunu tasarlarken, pozisyon güncelleme işlemi değiştirilmelidir. SSA'da sosyal örümcekler, alt ve üst sınır aralığına göre arama alanlarındaki konumlarını sürekli olarak rastgele bir konum olarak değiştirirler. İkili optimizasyonda, çözüm ikili $\{0, 1\}$ değerleriyle sınırlıdır. Sürekli optimizasyon problemleri için geliştirilmiş SSA, ikili optimizasyon problemlerini de çözebilecek şekilde yeniden yapılandırılmıştır. Böylece BinSSA önerilmiştir. Genel olarak, özellik seçimi, azalan boyutsallık, tesis yerleşim problemi ve rüzgâr türbini yerleşim problemi gibi ikili arama alanlarıyla ilgilenen

birçok problem vardır. BinSSA ile bu alt bölümde farklı ikili problem türleri çözülerek performansı test edilmiştir.

İkili dönüşümün ana parçası aktarım (transfer) fonksiyonlarıdır. BinSSA'da, çözüm havuzu $\{0, 1\}$ ikili dize değerleri ile temsil edilir. Örümceğin konumu 0–1 çevirme işlemi ile güncellenir. Bu çevirme işlemi, transfer fonksiyonuyla ilgili belirli bir eşik değeri ile gerçekleştirilir. Transfer fonksiyonları, pozisyon vektörünün elemanlarını 0'dan 1'e değiştirme ve bunun tersi olasılığını tanımlar (Kennedy ve Everhart, 1997). Literatürde S şeklinde, V şeklinde ve mod tabanlı birçok transfer fonksiyonu bulunmaktadır. S şekilli, V şekilli ve mod tabanlı transfer fonksiyonları çizelge 3.1.'de gösterilmiştir. İkili dönüşüm için seçilecek transfer fonksiyonu BOP'larda kullanılan algoritmanın performansını etkilemektedir. Bir transfer fonksiyonu hem yüksek bir olasılık hem de küçük bir pozisyon değiştirme olasılığı sunmalıdır.

Çizelge 3.1. S-şekilli, V-şekilli ve mod tabanlı transfer fonksiyonları (Mirjalili ve Lewis, 2013; Beşikçili ve ark., 2018)

S-şekilli		V-şekilli	
İsim	Transfer Fonksiyonu	İsim	Transfer Fonksiyonu
S1	$T(x) = \frac{1}{1 + e^{-2x}}$	V1	$T(x) = \left \operatorname{erf}\left(\frac{\sqrt{\pi}}{2}x\right) \right $
S2	$T(x) = \frac{1}{1 + e^{-x}}$	V2	$T(x) = \tanh(x) $
S3	$T(x) = \frac{1}{1 + e^{\frac{-x}{2}}}$	V3	$T(x) = \left \frac{x}{\sqrt{1 + x^2}} \right $
S4	$T(x) = \frac{1}{1 + e^{\frac{-x}{3}}}$	V4	$T(x) = \left \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x\right) \right $
M1	$T(x) = \operatorname{mod}(\operatorname{floor}(x), 2);$		
M2	$T(x) = \operatorname{mod}(\operatorname{round}(\operatorname{mod}(x), 2), 2);$		

3.4.1. Sürekli optimizasyon görevleri için İkili Sosyal Örümcek Algoritması (BinSSA)

Bu alt bölümde BinSSA sürekli optimizasyon görevleri için ikili şekle çevrilmiştir. Sürekli arama uzayı ikili uzaya çevrilirken, iki S-şekilli (Sigmoidal transfer fonksiyonu ve Modifiye sigmoidal transfer fonksiyonu) ve iki V-şekilli (Tanjant hiperbolik transfer fonksiyonu ve Arktanjan transfer fonksiyonu) transfer fonksiyonları kullanılmıştır. Böylece, BinSSA-Tanh, BinSSA-Sigm, BinSSA-MSigm ve BinSSA-Arctan olarak dört farklı BinSSA varyasyonu oluşturulmuştur. BinSSA'da ikili arama uzayında keşif ve sömürü arasındaki dengeyi sağlamak için aday çözüm üretim

aşamasında iki farklı teknik (SimSSA ve LogicSSA) geliştirilmiştir. Bunlar benzerlik ölçümü (SimSSA) ve mantıksal kapı (LogicSSA) teknikleridir. Her iki tekniğin farklı tiplerde sunulan avantajları vardır. Örneğin, LogicSSA yerel aramada çok güçlü iken SimSSA çözüm alanında yeni çözümler keşfetmede çok iyidir.

Benzerlik tabanlı İkili Sosyal Örümcek Algoritması (SimSSA):

Benzerlik ölçüm tekniği, iki farklı ikili yapı arasındaki benzerliği ölçmek için geliştirilmiştir. İkili veriler için 76 benzerlik ölçüm tekniği vardır (Qiao ve ark., 2006). En sık kullanılan genel amaçlı benzerlik ölçüm tekniklerinden biri denklem 3.12'de (Jaccard'ın benzerlik ölçüm tekniği) gösterilmiştir (Cinar ve Kiran, 2018).

$$Jaccard'in_benzerli\ddot{g}i(X_i, Y_i) = \frac{m_{11}}{m_{01} + m_{10} + m_{11}} \quad (3.12)$$

Burada X_i ve Y_i iki ikili yapıdaki bireyleri ifade etmektedir. ' n ', X_i ve Y_i için problem boyutunu göstermektedir. $X_{in} \in \{0,1\}$, $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]$, $Y_{in} \in \{0,1\}$, $Y_i = [y_{i1}, y_{i2}, \dots, y_{in}]$. X_i ve Y_i arasındaki benzerliğin ölçülmesi için bit değerleriyle karşılaştırılır. Dört olası durum vardır ve bunlar çizelge 3.2.'de gösterilmiştir.

Çizelge 3.2. Benzerlik için olası durumlar

durumlar	x_{in}	y_{in}	m_{11}	m_{10}	m_{01}	m_{00}
1	1	1	1	0	0	0
2	1	0	0	1	0	0
3	0	1	0	0	1	0
4	0	0	0	0	0	1

m_{11} : n problem boyutu olmak üzere ve $i = 1, 2, \dots, n$ ise X_i içindeki $x_{in} = 1$ ve Y_i içindeki $y_{in} = 1$ olduğu zaman aşağıdaki gibi elde edilir (Emary ve ark., 2016).

$$m_{11} = \sum_{n=1}^n I_n, \quad \text{where } I_n = \begin{cases} 1, & \text{if } (x_{in} = 1 \text{ ve } y_{in} = 1) \\ 0, & \text{aksi takdirde} \end{cases} \quad (3.13)$$

m_{10} : n problem boyutu olmak üzere ve $i = 1, 2, \dots, n$ ise X_i içindeki $x_{in} = 1$ ve Y_i içindeki $y_{in} = 0$ olduğu zaman aşağıdaki gibi elde edilir (Emary ve ark., 2016).

$$m_{10} = \sum_{n=1}^n I_n, \quad \text{where } I_n = \begin{cases} 1, & \text{if } (x_{in} = 1 \text{ ve } y_{in} = 0) \\ 0, & \text{aksi takdirde} \end{cases} \quad (3.14)$$

m_{01} : n problem boyutu olmak üzere ve $i = 1, 2, \dots, n$ ise X_i içindeki $x_{in} = 0$ ve Y_i içindeki $y_{in} = 1$ olduğu zaman aşağıdaki gibi elde edilir (Emary ve ark., 2016).

$$m_{01} = \sum_{n=1}^n I_n, \quad \text{where } I_n = \begin{cases} 1, & \text{if } (x_{in} = 0 \text{ ve } y_{in} = 1) \\ 0, & \text{aksi takdirde} \end{cases} \quad (3.15)$$

m_{00} : n problem boyutu olmak üzere ve $i = 1, 2, \dots, n$ ise X_i içindeki $x_{in} = 0$ ve Y_i içindeki $y_{in} = 0$ olduğu zaman aşağıdaki gibi elde edilir (Emary ve ark., 2016).

$$m_{00} = \sum_{n=1}^n I_n, \quad \text{where } I_n = \begin{cases} 1, & \text{if } (x_{in} = 0 \text{ ve } y_{in} = 0) \\ 0, & \text{aksi takdirde} \end{cases} \quad (3.16)$$

m_{00} , Jaccard'ın benzerlik ölçüm tekniğinde kullanılmaz. Çünkü iki ikili yapı arasındaki 0 değeri Jaccard'ın benzerlik ölçüm tekniğinin tanımında yer almamaktadır.

Jaccard'ın benzerlik katsayısını dikkate alarak, X_i ve Y_i arasında birbirinden ne kadar uzakta olduklarını tanımlayan sezgisel bir farklılık ölçüsü şu şekilde hesaplanabilir (Cinar ve Kiran, 2018):

$$\text{Benzeşmezlik}(X_i, Y_i) = 1 - \text{Jaccard'ın_benzerliği}(X_i, Y_i) = 1 - \frac{m_{11}}{m_{01} + m_{10} + m_{11}} \quad (3.17)$$

Burada $\text{Benzeşmezlik}(X_i, Y_i) [0,1]$ arasında değerler alır.

SimSSA için yeni bir aday çözüm üretme:

SimSSA'da yeni bir çözüm üretmek için mevcut örümcek ve optimum en iyi örümcek ($\text{örümcek}_{\text{Eiyi}}^{\text{Global}}$) veya mevcut örümcek ve rastgele seçilmiş komşu örümcek kullanılmaktadır. BinSSA için ikili arama alanında yeni bir çözüm üretmek için denklem 3.18 kullanılmaktadır.

$$\text{Benzeşmezlik}(X_{\text{yeni}}, X_i) \approx \varphi \times \text{Benzerlik}(X_i, Y_i) \quad (3.18)$$

Burada X_{yeni} yeni üretilmiş aday çözümü ifade etmektedir. X_i mevcut örümceği ve Y_i rastgele bir örümceği ifade etmektedir. φ pozitif random ölçekleme faktörüdür. $\approx$ hemen hemen eşit operatörüdür.

Yeni ikili çözümü X_{yeni} üretmek için, aşağıdaki üç değişkenin (M_{11} , M_{10} , ve M_{01}) değeri belirlenmelidir.

$$\min \left| \left(1 - \frac{M_{11}}{M_{11} + M_{10} + M_{01}} \right) - \varphi \times \left(1 - \frac{m_{11}}{m_{11} + m_{10} + m_{01}} \right) \right| \quad (3.19)$$

$$M_{11} + M_{01} = n1 \quad (3.20)$$

$$M_{10} \leq n0 \quad (3.21)$$

$$M_{10}, M_{01}, M_{11} \geq 0 \text{ ve tamsayı} \quad (3.22)$$

Burada $n1$, X_i 'de 1 değerine sahip toplam bit sayısı ve $n0$, X_i 'de 0 değerine sahip toplam bit sayısıdır. M_{11} , M_{10} , ve M_{01} değerlerini belirlemek için, denklemler tarafından verilen tamsayı matematiksel model. (3.19) - (3.22) çözülmelidir. Matematiksel modeli çözdükten ve M_{11} , M_{10} , ve M_{01} 'in optimal değerlerini bulduktan sonra, X_{yeni} aday çözümü Yeni İkili Çözüm Üreticisi (NBSG) kullanılarak elde edilebilir. NBSG algoritmasının çalışması, aşağıdaki 16 bitlik bireyler için örnekle açıklanmıştır:

$$X_i = [0011010110101110],$$

$$Y_i = [1000101111000110] \text{ ve } \varphi = 0.7$$

$$m_{11}=4, m_{01}=4, m_{10}=5, m_{00}=3$$

(m_{00} Jaccard'ın benzerlik ölçüm tekniğinde kullanılmaz (Wang ve ark., 2010)).

$$\begin{aligned} Benzeşmezlik(X_i, Y_i) &= 1 - Benzerlik(X_i, Y_i) = 1 - \frac{m_{11}}{m_{01} + m_{10} + m_{11}} \\ &= 1 - \frac{4}{4 + 4 + 5} = 1 - \frac{4}{13} = 0.69 \end{aligned}$$

$$A = \varphi \times Benzeşmezlik(X_i, Y_i) = 0.7 \times 0.69 = 0.48$$

Denklemler tarafından verilen tamsayı matematiksel modele göre (3.19) - (3.22):

$$\min f \left| \left(1 - \frac{M_{11}}{M_{11} + M_{10} + M_{01}} \right) - A \right| = \min f \left| \left(1 - \frac{M_{11}}{M_{11} + M_{10} + M_{01}} \right) - 0.48 \right|$$

$$M_{11} + M_{01} = 8(n1)$$

$$M_{10} \leq 8 (n0)$$

$$M_{10}, M_{01}, M_{11} \geq 0 \text{ ve tamsayı}$$

Model bir numaralandırma şeması kullanılarak çözüldükten sonra, optimal çıktı $M_{01}=0$, $M_{10}=5$, ve $M_{11}=8$ ve $f=0.1$ olarak elde edilir. X_{yeni} , 1x16 sıfır-bit dizisi olarak üretilir. Numaralandırma şeması şu şekilde hesaplanmıştır. $m_{11}=4$, $m_{01}=4$ değerleri kullanılarak $M_{11}+M_{01}=8$ bilgisi ve $16-8=8$ ($M_{10}<8$) bilgisi kullanılarak $M_{10}=5$ değeri belirlenmiştir. $M_{11}+M_{01}=8$ ise $M_{01}=0$ ve $M_{11}=8$ şeklinde belirlenmiştir.

Kalıtım Aşaması: X_i 'de 1'e eşit olan bitlerin pozisyonları bulunur ($Ones_{X_i} = [3,4,6,8,9, 11,13,14,15]$) ve $M_{11} = 8$ bitlik konumlar $Ones_{X_i}$ 'den seçilir (rastgele seçim yapılır $[3,4,6,8,9,11,13,15]$). Bu konumlardaki X_{yeni} bitleri 1 olarak değiştirilir. Yeni durumda $X_{yeni}=[0011010110101010]$ şeklinde elde edilir.

Ayrılma Aşaması: X_i 'de 0'a eşit olan bitlerin pozisyonları bulunur ($Zeros_{X_i} = [1,2,5,7,10,12,16]$) ve $M_{10} = 5$ bitlik konumlar $Zeros_{X_i}$ 'den seçilir (rastgele seçimi yapılır $[1,2,5,12,16]$). Bu konumlardaki X_{yeni} bitleri 1 olarak değiştirilir. Son durumda $X_{yeni}=[1111110110111011]$ şeklinde elde edilir.

Farklı NBSG algoritma örnekleri (Jaccard, 1901; Choi ve ark., 2010; Cinar ve Kiran, 2018) kaynaklarından incelenebilir.

Mantık (Logic) kapı tabanlı İkili Sosyal Örümcek Algoritması (LogicSSA):

Mantık kapıları, ikili çözüm uzayındaki ikili yapılar üzerinde çalışmak için çok uygundur. Bu kapıların giriş ve çıkış değerleri ikili değerlerdir $\{0, 1\}$. Burada temelde üç mantık kapısı vardır. Bunlar "AND", "OR" ve "Özel veya (XOR)" mantık kapılarıdır. Özel veya (XOR) mantık kapısı "AND" ve "OR" mantık kapılarının bir

kombinasyonudur ve mantıksal durumlarda yaygın olarak kullanılmaktadır (Kiran ve Gündüz, 2013; Cinar ve Kiran, 2018; Aslan ve ark., 2019). LogicSSA’da “XOR” mantık kapısı yeni aday çözüm üretim aşamasında kullanılmıştır. Çünkü bu kapıda her çıkış bitinin değişim olasılığı (bit değeri = 0 ve bit değeri = 1) % 50’dir. “XOR” mantık kapısında her çıkış bitinin değişim olasılığı birbirine eşittir. “OR” kapısında her çıkış bitinin değişim olasılığı (bit değeri = 0 ve bit değeri = 1) % 75’dir. “AND” kapısında her çıkış bitinin değişim olasılığı (bit değeri = 0 ve bit değeri = 1) % 25’dir. Doğruluk tabloları ve bu değişim durumları çizelge 3.3.’te verilmiştir. ‘+’ sembolü “OR” mantık kapısını, ‘&’ sembolü “AND” mantık kapısını ve ‘ $\oplus$ ’ sembolü “XOR” mantık kapısını temsil etmektedir. BinSSA’nın arama yeteneğini geliştirmek için, XOR lojik kapısı aday çözüm üretiminde kullanılmıştır. Bu tezde ST değeri 0.3 olarak alınmıştır (Cinar ve Kiran, 2018). LogicSSA’da, denklem 3.23 ile yeni bir aday çözüm (örümcek) oluşturulmuştur.

$$X_{yenij} = \begin{cases} X_{ij} \oplus (B_j \oplus N_{rj}), & \text{if } (rand_{ij} < ST) \\ X_{ij}, & \text{aksi takdirde} \end{cases} \quad (3.23)$$

Burada X_{ij} ; i inci örümceğin j inci boyutunu temsil eder. B_j ; optimal en iyi örümceğin j inci boyutunu temsil eder. N_{rj} ; rastgele seçilmiş bir komşu örümceğin j inci boyutunu gösterir. X_{yenij} ; yeni üretilen örümceğin j inci boyutunu gösterir.

Çizelge 3.3. OR, AND, ve XOR kapıları için doğruluk tabloları

Pozisyonlar		OR Kapısı	AND Kapısı	XOR Kapısı
X_i	X_k	$X_i + X_k$	$X_i \& X_k$	$X_i \oplus X_k$
0	0	0	0	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	0
		75%=1, 25%=0	25%=1, 75%=0	50%=1, 50%=0

Mantık kapısı ve benzerlik ölçütü tabanlı BinSSA:

BinSSA, LogicSSA ve SimSSA tekniklerinin kombinasyonundan oluşan bir algoritmadır. BinSSA, aday çözümlerin üretiminde her iki tekniği de kullanmaktadır. SimSSA veya LogicSSA tekniklerinden biri [0,1] aralığında rastgele üretilen bir değışkene göre seçilir. Bu seçim işlemi denklem 3.24’te gösterilmiştir. BinSSA’nın ayrıntılı sözde kodu şekil 3.20.’de gösterilmektedir.

$$\text{Aday Çözüm Üretim Methodu} = \begin{cases} \text{SimSSA}, & \text{if } (rand < 0.5) \\ \text{LogicSSA}, & \text{aksi takdirde} \end{cases} \quad (3.24)$$

BinSSA'nın sözde-kodu

- 1: BinSSA'nın parametre değerlerini ata
- 2: Sabit parametre değerlerini ve değişkenlerini belirle

```

3: for i=1 to popülasyon boyutu do
4:   for j=1 to problem boyutu do
5:     if r~U(0,1)<0.5 then
6:        $\Delta_i^j=1$ 
7:     else
8:        $\Delta_i^j=0$ 
9:     end if
10:   end for
11: end for

```

Başlangıçta örümcek
popülasyonu için rastgele
ikili arama uzayının
yaratılması

```

12: Her x örümceği için  $V_x^{target}$  başlat
13: while durdurma kriteri sağlanana kadar do
14:   for each popülasyon içindeki örümcek x için do
15:     Örümcek x'in uygunluk değerini hesapla ve
16:     örümcek x'in pozisyonunda titreşim değeri üret
17:   end for
18:   for each popülasyon içindeki örümcek x için do
19:     Bütün örümcekler tarafından üretilen V titreşim yoğunluklarını hesapla
20:     V'den en güçlü titreşim  $V_x^{Eiyi}$  seç
21:     if  $V_x^{Eiyi}$ 'nin yoğunluğu  $V_x^{target}$ 'dan daha büyükse then
22:        $V_x^{Eiyi}$ 'i  $V_x^{target}$  olarak depola
23:     end if
24:     cs değerini güncelle
25:     [0,1) aralığında r isimli bir random değer üret
26:     if  $r > p_c^{cs}$  then
27:       Boyut maskesini güncelle
28:     end if
29:      $P_x^{follow}$  üret
30:     Her örümcek için bir random yürüyüş gerçekleştir
31:     Sınır ihlallerini düzelt
32:   end for

```

```

33: for k=1 to varyasyon sayısı do
34:   for i=1 to popülasyon boyutu do
35:     for j=1 to problem boyutu do
36:        $\lambda \sim U(0,1)$ 
37:       if k=1 then
38:          $\theta(x_i^j) = \text{Tanh}(x_i^j)$ 
39:       else if k=2 then
40:          $\theta(x_i^j) = \text{Sig}(x_i^j)$ 
41:       else if k=3 then
42:          $\theta(x_i^j) = \text{Msig}(x_i^j)$ 
43:       else if k=4 then
44:          $\theta(x_i^j) = A(x_i^j)$ 
45:       end if
46:        $\Delta_i^j = \begin{cases} 1 & \text{if } \theta(x_i^j) > \lambda \\ 0 & \text{Aksi takdirde} \end{cases}$ 
47:     end for
48:   end for
49: end for

```

Sürekli arama
uzayının ikili arama
uzayına dönüşüm
süreci

```

50:   if r U(0,1)<0.5 then
51:     Aday çözümler (X) için SimSSA çalıştır
52:   else
53:     Aday çözümler (X) için LogicSSA çalıştır
54:   end if
55:   for i=1 to popülasyon boyutu do
56:     for j=1 to problem boyutu do
57:       if uygunluk fonksiyonu( $X_i^j$ ) > uygunluk fonksiyonu ( $x_i^j$ ) value then
58:          $x_i^j = X_i^j$ 
59:       end if
60:     end for
61:   end for
62: end while
63: En iyi çözüm çıktısı bulma

```

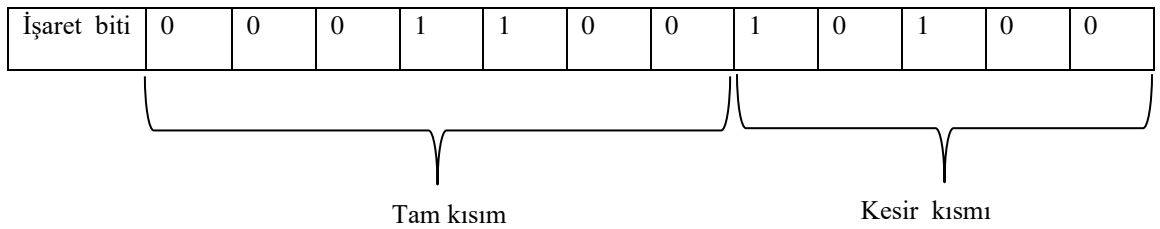
Aday
çözüm
üretim
metodu

Şekil 3.20. BinSSA'nın detaylı sözde-kodu

Bu tezde, sürekli test fonksiyonlarının uygunluk fonksiyonları için bir popülasyon bireyinin her boyutu Q-bit ikili dize şeklinde düşünülmüştür. Bu ikili dizenin ilk biti işaret biti, tamsayı kısmı için sonraki P bitleri ve son Q-P-1 bitleri ise noktalı sayı kısmı olarak kodlandığı bir BOP olarak ele alınmaktadır. Q ve P değerleri her test fonksiyonu için arama uzayının aralık değerlerine göre seçilmiştir. f1 (sphere) $([-100,100])$ için karşılaştırma fonksiyonu için $Q = 13$, $P = 7$, Q değerinin ilk biti işaret biti ve kesirli kısım = $Q - (P + 1)$ (işaret biti) = $13 - (7 + 1) = 5$ şeklinde belirlenir. Uygunluk fonksiyonu; bu Q-bit dizelerinin her birini gerçek bir sayıya dönüştürür ve uygunluk değerini hesaplar.

f1 (sphere) $([-100,100])$ için;

$(1100100)_2 = 100$ ise P değeri 7 bit ile temsil edilir. Kesirli kısım 5 bit ile ifade edildiğini kabul edersek $Q = 1$ (işaret biti) + P + Kesir kısım = 13 bit eder. Popülasyon bireylerinin her boyutu 13 bitlik ikili ifadelerden oluşur. Her bir boyutun tam sayıya çevrilmesi ile boyutların sürekli değerleri elde edilmiş olur. Bu sürekli gerçek değerler kullanılarak uygunluk fonksiyonları hesaplanmaktadır. Şekil 3.21'de sadece bir boyut için 13 bitlik ikilik bir değerin gerçek bir sayıya çevrilmesi gösterilmiştir.



Şekil 3.21. Bir boyut için ikilik-onluk hesaplaması

$$(00001100,10100)_2=12,625$$

$$(10001100,10100)_2=-12,625$$

3.4.2. Özellik Seçme (FS) problemi için İkili Sosyal Örümcek Algoritması (BinSSA)

Bu bölümde bir BOP olarak BinSSA ile Özellik Seçme (FS) problemi çözülmüştür. Sürekli arama uzayının ikili arama uzayına dönüşümünde çizelge 3.1’de yer alan transfer fonksiyonları kullanılmıştır. Dört adet S-şekilli ve dört adet V-şekilli transfer fonksiyonuna göre sekiz adet BinSSA varyasyonu oluşturulmuştur (BinSSA1, BinSSA2, BinSSA3, BinSSA4, BinSSA5, BinSSA6, BinSSA7 ve BinSSA8). Aday çözüm üretimi için bir çaprazlama operatörü eklenmiş ve böylece BinSSA’da kâşif ve sömürü arasındaki denge kurulmuştur (BinSSA1-CR, BinSSA2-CR, BinSSA3-CR, BinSSA4-CR, BinSSA5-CR, BinSSA6-CR, BinSSA7-CR ve BinSSA8-CR).

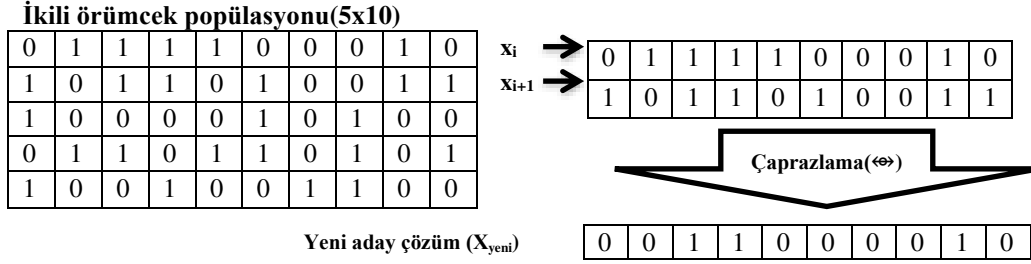
BinSSA için çaprazlama operatörü (BinSSA-CR):

BinSSA tarafından elde edilen çözümlerin kalitesini artırmak için, çaprazlama operatörü ile yeni bir aday çözüm seti elde edilir. Çaprazlama operatörü denklem 3.25 ve 3.26’ da gösterilmiştir.

$$X_i = x_i \Leftrightarrow x_{i+1} \quad (3.25)$$

$$X_i^n = \begin{cases} x_i^n & , rand \geq 0.5 \\ x_{i+1}^n & , aksi takdirde \end{cases} \quad (3.26)$$

Burada $\Leftrightarrow$ çaprazlama operatörünü göstermektedir ve bu çaprazlama operatörü iki ikili çözüm üzerine uygulanır. X_i, x_i ve x_{i+1} üzerine çaprazlama işlemi uygulanmış ve elde edilen yeni aday çözümü temsil etmektedir ($n=1, 2, \dots$, problem boyutu) ($i=1, 2, \dots$, popülasyon boyutu (N)). Şekil 3.22.’de çaprazlama süreci gösterilmektedir.



Şekil 3.22. Çaprazlama süreci

BinSSA ile özellik seçme problemi:

Özellik seçim problemlerinde, ikili çözüm kümesi $\{0, 1\}$ değerlerinden oluşur. BinSSA'daki ikili çözüm alanı popülasyon boyutu $(N) \times$ özellik numarası (n) olarak tanımlanır. Özellik seçimi için, “1” değeri karşılık gelen özelliğin seçildiğini ve “0” değeri karşılık gelen özelliğin seçilmediğini gösterir. Şekil 3.23.’de BinSSA için ikili arama uzayı gösterilmiştir.

		Özellik sayısı (n)			
Popülasyon boyutu (N)	0	1	...	1	0
	1	0	...	0	1
	0	1	...	0	0
	.	.	...	.	.
	.	.		.	.
	.	.		.	.
	0	0	...	0	1
	1	1	...	1	1

Şekil 3.23. BinSSA için ikili arama uzayı

Liu ve Motoda'ya göre, gereksiz ve alakasız özellikler sınıflandırıcının performansını birçok yönden olumsuz etkileyebilir; örneğin veri kümesindeki daha fazla özellik, sınıflandırıcının öğrenmesi için daha uzun zamana mal olan daha fazla örneğin eklenmesi ihtiyacını artırır (Liu ve Motoda, 1998). Özellik seçimi, alakasız ve gereksiz özellikleri kaldırarak sınıflandırıcının doğruluk performansını ve sınıflandırıcının öğrenme süresi performansını geliştirir. Bu amaçla, BinSSA ile en az sayıdaki özellik ile maksimum sınıflandırma doğruluğunu elde etme hedeflenir. Denklem 3.27’de özellik seçme problemi için BinSSA’da kullanılacak uygunluk fonksiyonu gösterilmiştir.

$$\downarrow \text{Uygunluk Fonksiyonu} = \rho Hata(D) + \phi \frac{|M|}{|n|} \quad (3.27)$$

Hata (D); sınıflandırma hata oranını göstermektedir. ρ ve φ ; sınıflandırma doğruluğunu ve özellik azaltmayı kontrol etmek için kullanılan sabitlerdir. $|M|$; özellik alt kümesi boyutunu temsil eder. $|n|$; toplam özellik sayısını (problem boyutunu) gösterir. ρ ; $[0,1]$ arasından seçilir ve $\varphi = (1 - \rho)$ dir.

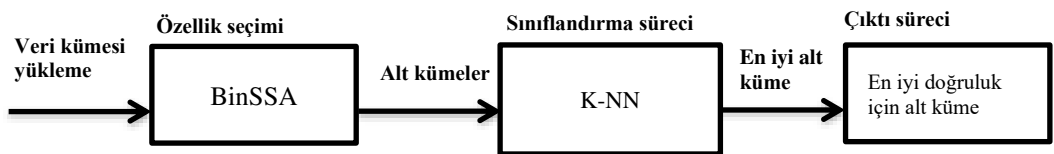
K-en yakın komşu sınıflandırıcısı (K-NN) ve destek vektör makinesi sınıflandırıcısı (SVM):

K-en yakın komşu sınıflandırıcısı (K-NN) algoritması, belirli bir etiketlenmemiş örnek ile en yakın K örnekleri (K komşuları) arasındaki mesafeyi ölçerek etiketlenmemiş örnekleri sınıflandırmaya dayanan basit, parametrik olmayan ve örnek tabanlı bir sınıflandırıcıdır (Pernkopf, 2005). Literatürde K-NN için farklı mesafe ölçümleri olmasına rağmen, en yaygın kullanılan ölçüm türü öklid mesafesidir. Denklem 3.28’de öklid mesafesi hesaplaması gösterilmiştir. K-NN sınıflandırıcısı (burada $K = 5$) optimal alt kümeyi üretmek için tercih edilir.

$$\text{öklid_mesafe}(x_a - x_b) = \sqrt{\sum_{i=1}^n (x_{ai} - x_{bi})^2} \quad (3.28)$$

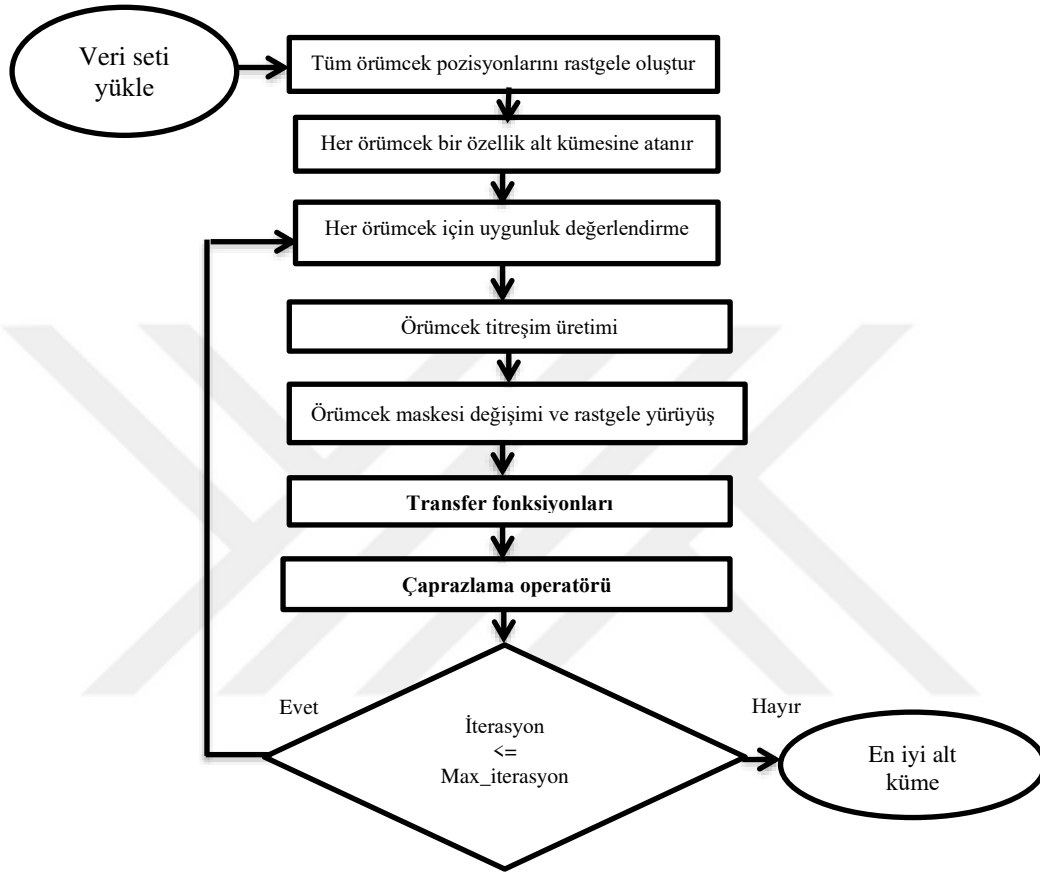
Destek vektör makinesi (SVM), Vapnik (1995) tarafından geliştirilmiş ve sınıflandırma ve regresyon görevleri için önerilmiştir. SVM, Vapnik-Chervonenkis boyutu ve yapısal risk minimizasyonu gibi güçlü bir istatistiksel öğrenme teorisi üzerine inşa edilmiştir (Babaoglu ve ark., 2010). Destek vektör makineleri sınıflandırmada önemli başarılarla sahiptir.

K-NN veya SVM, BinSSA’da sınıflandırıcı olarak kullanılmıştır. Bu çalışmada seçilen özellikleri kullanarak sınıflandırma işlemini gerçekleştirmesi için K-NN sınıflandırıcısı tercih edilmiştir. BinSSA için özellik seçme süreci şekil 3.24.’de gösterilmiştir.



Şekil 3.24. BinSSA için özellik seçme süreci

SVM ve K-NN'in her ikisi içinde, k-fold (burada $k = 10$) çapraz geçerlilik tekniği eğitim ve test seti üretmek için veri setine uyarlanmıştır. Veri setlerinin % 80'i eğitim seti için ve % 20'si test seti için kullanılmıştır. BinSSA için akış şeması şekil 3.25.'de ve BinSSA'nın çalışma adımları şekil 3.26.'da gösterilmiştir.



Şekil 3.25. BinSSA için akış şeması

Adım 1. Parametre ayarları: Gerekli sabit parametre setleri ayarlanır. Örneğin, değişkenlerin alt ve üst sınırları, maksimum yineleme sayısı, vb.

Adım 2. Başlatma: Sürüdeki her örümcek için rastgele başlangıç pozisyonu yerleri belirlenir. Onlar için hafıza atanır. Her örümcek için hedef titreşim başlatılır.

Adım 3. Her örümcek bir özellik alt kümesine atanır.

Adım 4. Değerlendirme: Uygunluk değeri her örümcek için hesaplanır. Sürünün en iyisi tespit edilir ve kaydedilir.

Adım 5. Titreşim değeri, her bir örümcek için titreşim Denklemleri (3.1, 3.2 ve 3.3) kullanılarak hesaplanır. Sürünün en iyisi tespit edilir ve kaydedilir.

Adım 6. Titreşim karşılaştırması: Alınan en iyi titreşim, hedef titreşim değerleriyle karşılaştırılır.

Adım 7. Örümcek maskesi güncellenir.

Adım 8. P_x^{follow} değeri denklem 3.5 ile hesaplanır.

Adım 9. Hareket: Her örümcek denklem 3.4 ile yeni bir pozisyona hareket eder ve yeni bir popülasyon elde edilir.

Adım 10. Transfer Fonksiyonu: Sürekli arama alanı sekiz farklı transfer fonksiyonu ile ikili arama uzayına dönüştürülür.

Adım 11. Çaprazlama operatörü: Yeni aday çözümler üretmek için iki çözüm arasında çaprazlama operatörü uygulanır ve yeni aday çözümler değerlendirilir.
Adım 12. Tekrar: Adım 4'ten başlayarak tüm kriterler durma kriterlerine ulaşana kadar tekrarlanır.
Adım 13. Çıktı: En iyi alt küme elde edilir.

Şekil 3.26. BinSSA'nın adımları

3.4.3. Kapasitesiz Tesis Yerleşim Problemi (UFLP) için İkili Sosyal Örümcek Algoritması (BinSSA)

Bu bölümde bir BOP olarak BinSSA ile Kapasitesiz Tesis Yerleşim Problemi (UFLP) çözülmüştür. UFLP problemleri saf ikili optimizasyon problemlerinden biridir ve literatürde optimum değerleri bilinmektedir. BinSSA, iki farklı UFLP veri seti üzerinde (15 farklı ölçekli Cap veri seti ve 20 farklı ölçekli M* veri seti) test edilmiştir. Ayrıca BinSSA'nın performansı CEC2015 benchmark fonksiyonları üzerinde de test edilmiştir. Sürekli arama uzayının ikili arama uzayına dönüşümünde çizelge 3.1'de yer alan transfer fonksiyonlarından V4 tercih edilmiştir. SSA'da yer alan rastgele yürüyüş aşaması kaldırılarak aday çözüm üretim şeması eklenmiştir. Aday çözüm üretimi için iki farklı yöntem geliştirilmiştir. Bunlar Dice'nin benzerlik ölçüm tekniği ve lojik kapılardır. Dice'nin benzerlik ölçüm metodu ikili arama uzayında yeni noktalar keşfedebilmek için kullanılmıştır. Son yıllarda ikili optimizasyon problemlerinde sıklıkla kullanılan lojik kapılar SSA'da yer alan ikili maske ile birleştirilmiştir ve xor-maske tekniği elde edilmiştir. Bu maske ile BinSSA'nın yerel arama yeteneği geliştirilmiştir. Böylece bu tezde BinSSA(Sim&Logic) önerilmiştir. BinSSA'da keşfetme ve sömürü arasındaki denge kurulmuştur.

Kapasitesiz Tesis Yerleşim Problemi (UFLP):

BinSSA'nın performansı UFLP Problemleri üzerinde incelenmiştir. UFLP'ler ikili optimizasyonlar için çok uygun olduğu için ve ikili yapılar için kolayca uygulanabilir olduğu için bir ikili test problemi olarak UFLP'ler bu tezde tercih edilmiştir. Ayrıca problemlerin optimum sonuçları literatürde bilinmektedir. UFLP, NP-hard sınıfında bir problem türüdür. Temel UFLP formülasyonunda, UFLP bir dizi (I) potansiyel tesis içerir. Her tesis açık veya kapalı olabilir. BinSSA'da, tesis '1' ise açık veya '0' ise kapalıdır. Bu tesisler bir dizi (J) müşteriye hizmet vermektedir. Bu problemin amacı, I ve J arasındaki sevkiyat masraflarının toplamını ve tesislerin açılış masraflarını en aza

indirmektir (Cinar ve Kiran, 2018). UFLP'nin matematiksel yapısı aşağıda gösterilmiştir.

$$f(UFLP) = \min\{\sum_{i \in I} \sum_{j \in J} c_{ij}x_{ij} + \sum_{i \in I} f_i y_i\} \quad (3.29)$$

Burada $I = \{1, 2, \dots, k\}$: olası tesis konumları kümesidir. $J = \{1, 2, \dots, g\}$: müşteri talep noktaları kümesidir. f_i : $i \in I$ 'de bir tesis açmanın sabit maliyetidir. c_{ij} : i inci tesis yeri ile j inci müşteri noktası arasındaki gönderi maliyetidir. x_{ij} : bir karar değişkenidir ve i inci tesise karşılık gelen j inci müşterinin talebidir. y_i : ikili değişkendir ve $y_i = 1$ ise $i \in I$ 'de tesis mevcuttur yada $y_i = 0$ tesis yoktur.

Kurallar:

$$\sum_{i \in I} x_{ij} = 1 \quad j \in J \quad (3.30)$$

$$x_{ij} \leq y_i, \quad i \in I, j \in J \quad (3.31)$$

$$x_{ij} \in \{0, 1\}, i \in I, j \in J \quad (3.32)$$

$$y_i \in \{0, 1\}, i \in I \quad (3.33)$$

Denklem 3.30 müşterilerin tüm taleplerini karşılamayı sağlarken, denklem 3.31 bir müşterinin bir tesisten ancak bir tesis açık olduğunda hizmet alabilmesini sağlar. Denklem 3.32 ve 3.33 ikili yapıdaki karar değişkenlerini tanımlar (Cinar ve Kiran, 2018).

Dice'nin benzerlik ölçüm tekniği:

Benzerlik ölçüm tekniği iki farklı ikili yapı arasındaki benzerliği ölçmek için geliştirilmiştir. İkili veriler için 76 benzerlik ölçüm tekniği vardır (Choi ve ark., 2010). Jaccard'ın benzerlik ölçüm tekniği (Jaccard, 1901) literatürde sıklıkla kullanılmıştır. Bu bölümde yeni aday çözümler üretmek için Dice'in benzerlik ölçüm tekniği tercih edilmiştir. Her iki teknik birbirine benzese de farklılıklar vardır. Dice'in benzerlik ölçüm tekniğinde, m_{11} değeri 2 ile çarpılır. Bu çalışmada Dice'in benzerlik ölçüm tekniğinin tercih edilmesindeki sebep, UFLP'de açık tesislerde benzerlik katsayısının

(bit değeri = 1) sayısını artırmak istenmesidir. Denklem 3.34’de Dice’s benzerlik ölçüm tekniği gösterilmiştir.

$$Dice'in_benzerliğı(X_i, Y_i) = \frac{2 \times m_{11}}{2 \times m_{11} + m_{01} + m_{10}} \quad (3.34)$$

Burada X_i ve Y_i iki ikili yapıdaki bireyleri ifade etmektedir. ‘ n ’, X_i ve Y_i için problem boyutunu göstermektedir. $X_{in} \in \{0,1\}$, $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]$, $Y_{in} \in \{0,1\}$, $Y_i = [y_{i1}, y_{i2}, \dots, y_{in}]$. X_i ve Y_i arasındaki benzerlik ölçümü için bit değerleri karşılaştırılır. Dört olası durum vardır ve bunlar çizelge 3.4.’de gösterilmiştir.

Çizelge 3.4. Dice’in benzerlik ölçümü için olası durumlar

Durumlar	X_i	Y_i	m_{00}	m_{01}	m_{10}	m_{11}
1	0	0	1	0	0	0
2	0	1	0	1	0	0
3	1	0	0	0	1	0
4	1	1	0	0	0	1

Dice’in benzerlik katsayısını dikkate alarak, X_i ve Y_i arasında birbirinden ne kadar uzakta olduklarını tanımlayan sezgisel bir farklılık ölçüsü X_i ve Y_i şu şekilde hesaplanabilir (Cinar ve Kiran, 2018):

$$Benzeşmezlik(X_i, Y_i) = 1 - Dice'in_benzerliğı(X_i, Y_i) = 1 - \frac{2 \times m_{11}}{2 \times m_{11} + m_{01} + m_{10}} \quad (3.35)$$

Burada benzeşmezlik(X_i, Y_i): [0, 1] aralığındadır. Yeni aday çözüm üretim aşamaları Jaccard’da olduğu gibidir.

xor-maske tekniği:

“özel veya (*xor*)” mantık kapısı, "ve" ve "veya" mantık kapılarının bir kombinasyonudur ve mantıksal durumlarda yaygın olarak kullanılır. BinSSA’da “*xor*” mantık kapısı yeni bir aday çözüm (yeni örümcek) üretim aşamasında kullanılmaktadır. Çünkü arama uzayında bitin bit değeri (bit değeri = 0 ve bit değeri = 1) değişim olasılığı % 50’dir. SSA’da random yürüyüş için kullanılan ikili maske, BinSSA’da *xor* mantık kapısı ile birleştirilmiştir. BinSSA’da yeni aday çözümler üretmek için seçilecek random bir örümcek ve onun maskesi *xor* işleminden geçirilmekte ve elde edilen çözüm

tekrardan mevcut örümcekle *xor* işleminden geçirilir. Ve böylece yeni aday çözüm elde edilir. Denlem 3.36'da xor-maske tekniği gösterilmiştir.

$$X_{yenij} = \begin{cases} X_{ij} \oplus (X_j^r \oplus mask_{rj}), & \text{if } (rand_{ij} < ST) \\ X_{ij}, & \text{aksi takdirde} \end{cases} \quad (3.36)$$

Burada X_{yenij} : i inci örümcek için yeni üretilen örümceği göstermektedir. X_{ij} : mevcut i inci örümceğin j inci boyutunu temsil eder. X_j^r : random komşu örümceğin j inci boyutunu temsil eder. $Mask_{rj}$: random komşu örümceğin ikili örümcek maskesinin j inci boyutunu gösterir. ST bir kontrol parametresidir ve değeri 0.3 olarak belirlenmiştir. SSA için ikili maske üretimi denklemler 3.5'de gösterilmiştir.

Her iki teknik de BinSSA'da aday çözümlerin üretiminde kullanılmaktadır. Dice'nin benzerlik ölçümü tekniği veya xor-maske tekniklerinden biri, $[0,1]$ aralığında rastgele üretilen bir değişkene göre seçilmiştir. Şekil 3.27.'de UFLP için BinSSA'nın çalışma adımları ve şekil 3.28.'de UFLP için BinSSA'nın akış şeması gösterilmiştir.

Adım 1. Parametre ayarları: Gerekli sabit parametre setleri ayarlanır. Örneğin, değişkenlerin alt ve üst sınırları, maksimum yineleme sayısı, vb.

Adım 2. Başlat: Sürüdeki her örümcek için rastgele başlangıç pozisyonu yerleri belirlenir. Her örümcek için hedef titreşim başlatılır.

Adım 3. Değerlendir: Uygunluk değeri her örümcek için hesaplanır. Sürünün en iyisi tespit edilir ve kaydedilir.

Adım 4. Titreşim değeri, her bir örümcek için titreşim denklemleri (3.1, 3.2 ve 3.3) kullanılarak hesaplanır. Sürünün en iyisi tespit edilir ve kaydedilir.

Adım 5. Titreşim karşılaştırma: Alınan en iyi titreşim, hedef titreşim değerleriyle karşılaştırılır.

Adım 6. Örümcek ikili maskesi güncellenir.

Adım 7. p_{follow} değeri denklemler 3.5 ile hesaplanır.

Adım 8. hareket: Her örümcek denklemler 3.4 ile yeni bir pozisyona hareket eder ve yeni bir popülasyon elde edilir.

Adım 9. Transfer fonksiyonu: Sürekli arama alanı, V4 transfer fonksiyonu ile ikili arama alanına dönüştürülür.

Adım 10. $[0,1]$ 'den rastgele sayı r üretilir. r 0,5'ten küçükse, adım 11 çalıştırılır, aksi takdirde adım 12 çalıştırılır.

Adım 11. Dice'in benzerlik ölçüm tekniği: Yeni aday çözümler üretmek için iki çözüm arasında benzerlik ölçüm tekniği uygulanmaktadır.

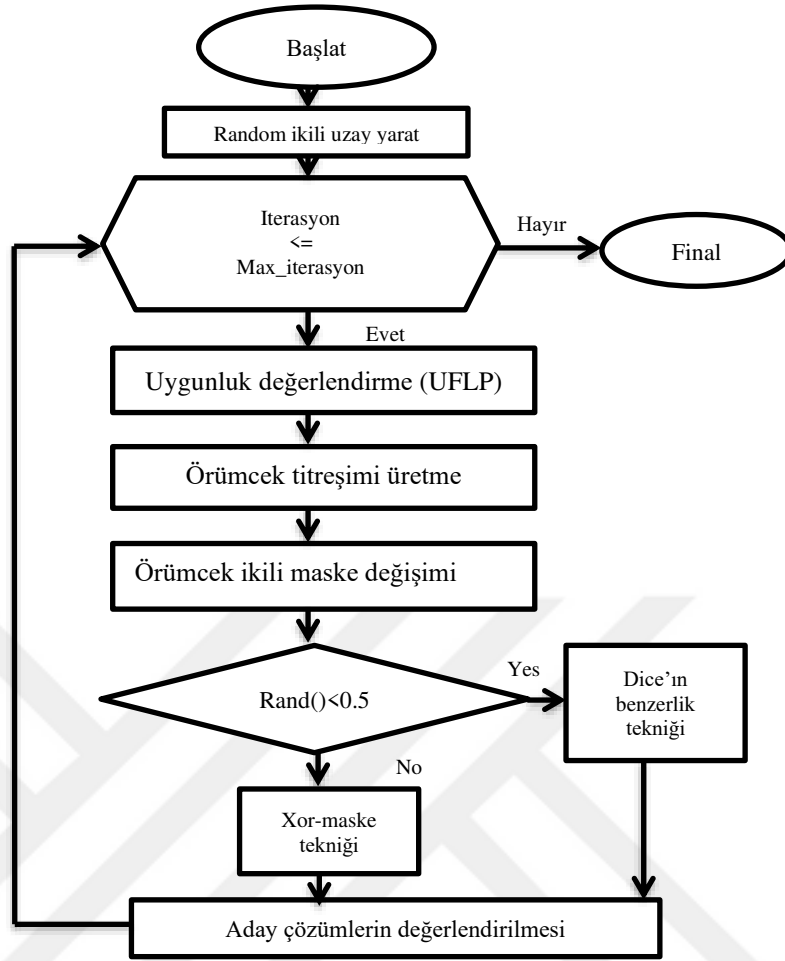
Adım 12. Xor-maske tekniği: Yeni aday çözümler üretmek için iki çözüm arasında xor-maske tekniği uygulanır.

Adım 13. Aday çözümler değerlendirilir.

Adım 14. Tekrar: Adım 3'ten başlayarak tüm kriterler durma kriterlerine ulaşana kadar tekrarlanır.

Adım 15. Çıktı: Optimum en iyi çözüm elde edilir.

Şekil 3.27. UFLP için BinSSA'nın çalışma adımları



Şekil 3.28. UFLP için BinSSA'nın akış şeması

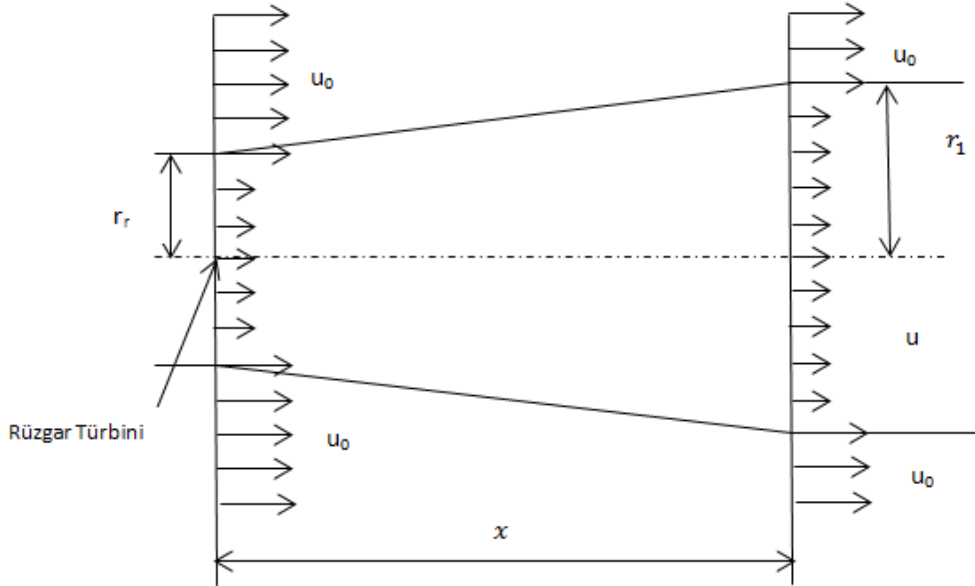
3.4.4. Rüzgâr türbini yerleşim problemi için İkili Sosyal Örümcek Algoritması (BinSSA)

Bu bölümde bir BOP olarak BinSSA ile rüzgâr türbini yerleşim problemi çözülmüştür. Rüzgâr yenilenebilir enerji kaynaklarından biridir. Rüzgâr enerjisi kullanımı tüm dünyada yaygınlaştıkça, rüzgâr türbinlerinin yerleştirilmesinin önemi artmaktadır. Bu tezde, mümkün olan en yüksek rüzgâr enerjisini elde etmek için rüzgâr türbinlerinin bir rüzgâr çiftliğine optimum şekilde yerleştirilmesi incelemiştir. Sürekli arama uzayının ikili arama uzayına dönüşümünde çizelge 3.1.'de yer alan transfer fonksiyonları kullanılmıştır. Dört adet S-şekilli, dört adet V-şekilli ve iki adet mod tabanlı transfer fonksiyonuna göre on adet BinSSA varyasyonu oluşturulmuştur (BinSSA1, BinSSA2, BinSSA3, BinSSA4, BinSSA5, BinSSA6, BinSSA7, BinSSA8, BinSSA9 ve BinSSA10). Aday çözüm üretimi için bir lojik kapı operatörü eklenmiş ve böylece BinSSA'da kâşif ve sömürü arasındaki denge kurulmuştur (BinSSA1-LG,

BinSSA2- LG, BinSSA3-LG, BinSSA4-LG, BinSSA5-LG, BinSSA6-LG, BinSSA7-LG, BinSSA8-LG, BinSSA9-LG ve BinSSA10-LG). Bu çalışmada, rüzgâr türbinleri iki farklı ızgara yapısına (10x10 ve 20x20) göre yerleştirilmiştir.

Rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemi:

Rüzgâr türbini yerleşiminin temel amaçlarından biri, türbinler arasındaki uyarılmaların neden olduğu kayıpları en aza indirmektir. Bu nedenle, rüzgâr türbinlerinin yerleştirilmesi için bir uyanma modeli gereklidir. Literatürdeki birçok çalışmada Jensen'in uyanma modeli kullanılmıştır (Jensen, 1983). Bu model, 1983 yılında Jensen tarafından önerilmiştir. Bu çalışmada, rüzgâr türbinlerinin rüzgâr çiftliği problemine yerleştirilmesi için bir uyanma modeli olarak Jensen uyanma modeli tercih edilmiştir. Rüzgâr santrallerinden maksimum verim elde etmek için rüzgâr türbinleri arasındaki mesafe dikkate alınmalıdır. Mosetti ve ark. Jensen'in uyanma ve maliyet modelini önermişlerdir (Mosetti ve ark., 1994). Bu model, bu konuyla ilgili ilk çalışmadır. Bu çalışmada kullanılan model, 1983 yılında Jensen tarafından geliştirilen uyanma modeline çok benzemektedir. Şekil 3.29.'da Jensen'in uyanma modeli gösterilmiştir.



Şekil 3.29. Jensen'in uyanma modeli (Wan ve ark., 2009)

Denklem 3.37, bir rüzgâr türbinin altındaki rüzgâr hızını (u) tanımlamak için kullanılır.

$$u = u_0 \left[1 - \frac{2a}{1 + \alpha \left(\frac{x}{r_1} \right)} \right] \quad (3.37)$$

Burada u_0 : rüzgâr hızı ortalamasıdır, a : aksenal indüksiyon faktörüdür, x : türbinin akış yönündeki mesafesidir, r_1 : akış yönü rotor yarıçapı ve α : sürüklenme sabitidir. Denklem 3.38, aşağı akış rotor yarıçapını (r_1) ve rotor yarıçapını (r_r) göstermektedir.

$$r_1 = r_r \sqrt{\frac{1-a}{1-2a}} \quad (3.38)$$

Denklem 3.39, sürüklenme sabitini göstermektedir.

$$\alpha = \frac{0.5}{\ln(z/z_0)} \quad (3.39)$$

Burada z : rüzgâr türbininin göbek yüksekliği ve z_0 : yüzey pürüzlülüğünü göstermektedir. Denklem 3.40, rüzgâr türbininin (C_T) itme katsayısını göstermektedir.

$$C_T = 4a(1 - a) \quad (3.40)$$

Birden fazla uyanma ile karşılaşan bir rüzgâr türbini, karışık uyanmanın kinetik enerjisinin kinetik enerji açıklarının toplamına eşit olduğu varsayılmıştır (Grady ve ark., 2005). Denklem 3.41, n sayıdaki türbinlerin aşağı yönlü akışta ortaya çıkan hızı olarak hesaplanmıştır.

$$u_i = u_0 \left[1 - \sqrt{\sum_{i=1}^n \left(1 - \frac{u}{u_0} \right)^2} \right] \quad (3.41)$$

Denklem 3.42, rüzgâr çiftliğindeki tüm rüzgâr türbinlerinin toplam güç üretimini göstermektedir.

$$P_T = \sum_{i=1}^n 0.3 \times u^3 \quad (3.42)$$

Denklem 3.43, Mosetti ve ark. (1994) tarafından yaratılan maliyet modelini göstermektedir. Maliyet modeli, rüzgâr çiftliğindeki rüzgâr türbinlerinin sayısına dayanmaktadır. Bu modelde, bir türbin kurma maliyetinin 1 olduğu varsayılmaktadır. Birden fazla rüzgâr türbini monte edilirse, maksimum maliyet azalmasının 1/3 olduğu varsayılır. Maliyetin geri kalanı 2/3 sabit bir oranda kabul edilir (Mosetti ve ark., 1994). Bu çalışmada, rüzgâr çiftliğinin maliyetini belirlemek için bir maliyet modeli seçilmiştir. Bu model literatürdeki diğer çalışmalarda da kullanılmıştır (Mosetti ve ark., 1994; Pookpant ve Ongsakul, 2013; Beşkirli ve ark., 2018).

$$Maliyet = n \left[\frac{2}{3} + \frac{1}{3} e^{-0.00174n^2} \right] \quad (3.43)$$

Denklem 3.44 amaç fonksiyonunu göstermektedir. Bu çalışmada amaç fonksiyonu, bir rüzgâr çiftliğinde minimum yatırım maliyeti (minimum rüzgâr türbini kurulumu) ile maksimum rüzgâr santrali gücü elde etmeyi amaçlamaktadır (Mittal, 2010).

$$Uygunluk \text{ fonksiyonu} = \frac{Maliyet}{P_T} \quad (3.44)$$

Rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemi için BinSSA:

Rüzgâr türbinlerinin rüzgâr çiftliği yerleştirilmesi probleminde arama uzayı ikili çözüm kümesi $\{0, 1\}$ değerlerinden oluşmaktadır. BinSSA'daki ikili çözüm uzayı, popülasyon büyüklüğü (N) x rüzgâr türbinlerinin sayısı (n : problem boyutu) olarak tanımlanır. Rüzgâr türbinlerinin rüzgâr çiftliği yerleştirilmesi probleminde; “1” değeri, karşılık gelen rüzgâr türbininin seçildiğini ve “0” değeri, karşılık gelen rüzgâr türbininin seçilmediğini gösterir. Şekil 3.30., BinSSA'daki sadece bir birey için ikili arama uzayı göstermektedir. Çizelge 3.5., rüzgâr türbini özelliklerini göstermektedir.

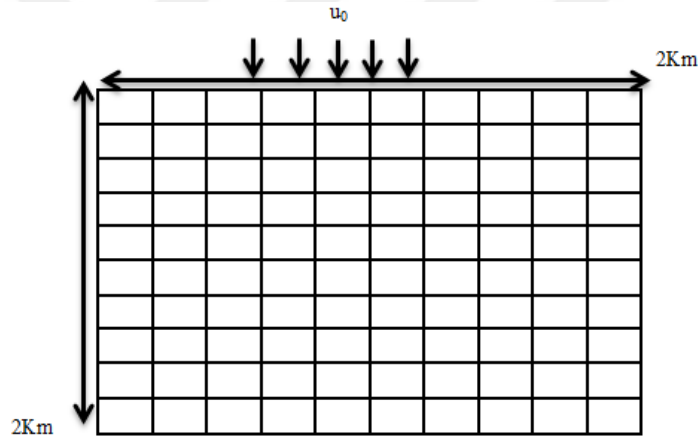
Problem boyutu (n)										
Popülasyon bireyleri (N)	0	1	0	0	0	...	1	1	0	1
	0	0	0	1	0	...	0	1	0	0
	1	1	1	0	1	...	1	1	1	1
	0	0	0	1	0	...	0	1	0	0
	1	1	0	0	1	...	1	0	0	1

Şekil 3.30. BinSSA için ikili arama uzayı

Çizelge 3.5. Rüzgâr türbini özellikleri (Pookpunt ve Ongsakul, 2013)

Özellikler	Değerler
Hub yüksekliği (z)	60m
Rotor yarıçapı (r_r)	40m
İtme katsayısı (C_T)	0,88
Arazi pürüzlülüğü (z_0)	0,3
Minimum mesafe	200m veya 5D

Bu çalışmada, 100 olası türbin yerine bölünmüş bir kare ızgara kullanılmıştır. Rüzgâr çiftliği $2000 \times 2000 \text{m}^2$ ($50D \times 50D$) bir alandan oluşur. Bu alan 10×10 ızgaraya veya 20×20 ızgaraya bölünebilir. Merkezine bir rüzgâr türbini yerleştirebileceğimiz her hücre, beş rotor çapına eşit bir genişliğe sahiptir ($5D$ veya 200m). (Marmidis ve ark., 2008; Emami ve Noghreh, 2010). Bu çalışmada 12 m/s (u_0) rüzgâr hızına sahip tek yönlü olgusu kullanılmıştır. Şekil 3.31., bölünmüş grubu ve tek rüzgâr yönü durumunu göstermektedir.

**Şekil 3.31.** Alt bölümlere ayrılmış rüzgâr çiftliği

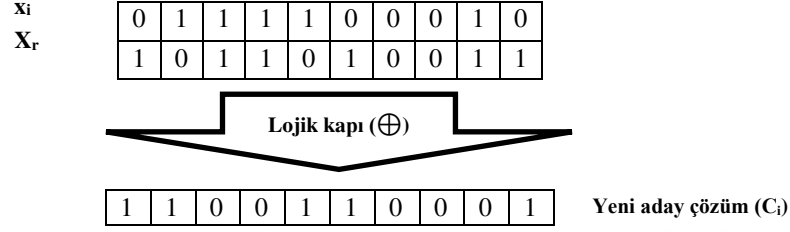
Mantık kapısı operatörlü BinSSA (BinSSA-LG):

BinSSA tarafından elde edilen çözümlerin kalitesini artırmak için, mantık kapısı operatörü ile yeni bir aday çözüm seti elde edilir. Lojik kapısı operatörü denklemler 3.45 ve 3.46'da gösterilmektedir.

$$C_i = x_i \oplus x_r \quad (3.45)$$

$$C_i^n = \begin{cases} x_i^n & , rand \geq 0.5 \\ x_i \oplus x_r & , aksi takdirde \end{cases} \quad (3.46)$$

Burada $\oplus$: iki ikili çözüm üzerinde mantık kapısı düzenini gerçekleştiren bir mantık kapısı operatörüdür. C_i : x_i ve x_r mantık kapısı operatörü uyguladıktan sonra elde edilen yeni aday çözümdür ($n=1, 2, \dots$, problem boyutu) ($N=1, 2, \dots$, popülasyon boyutu). x_r , örümcek popülasyonunda rastgele bir örümcektir. Şekil 3.32., mantık kapısı operatörünü göstermektedir.

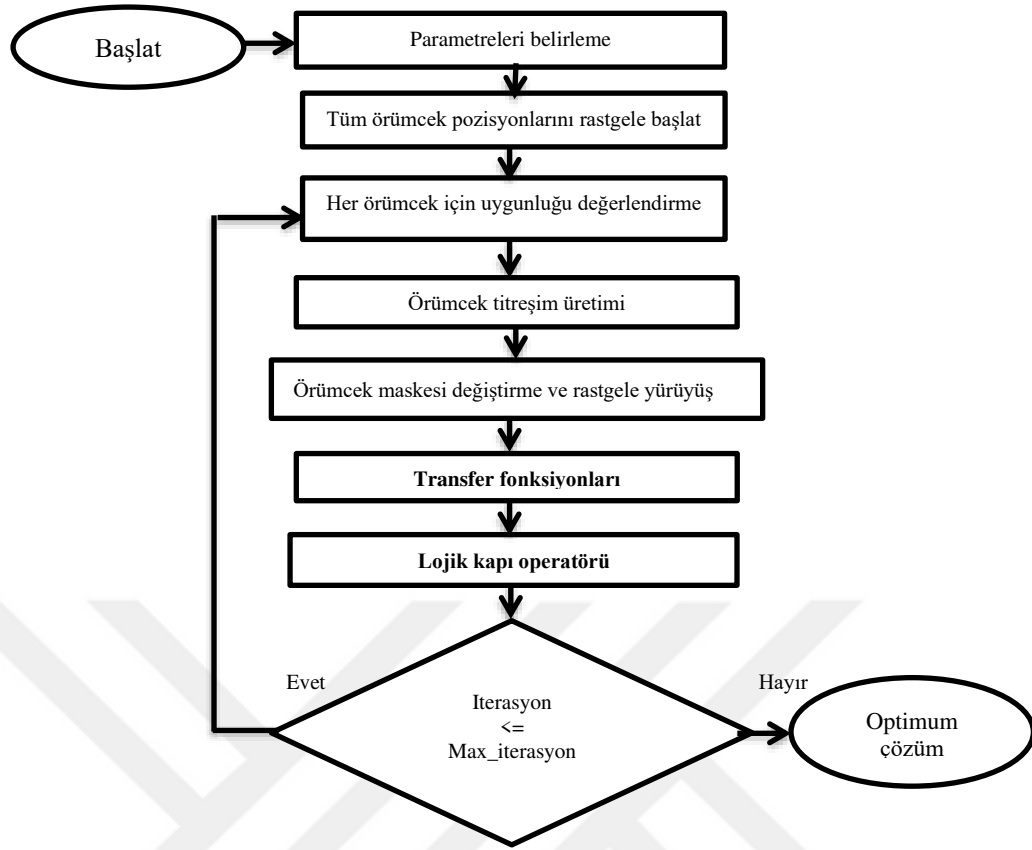


Şekil 3.32. Lojik kapı operatörü

Rüzgâr türbini yerleşim problemi için BinSSA'nın çalışma adımları şekil 3.33.'de ve BinSSA için akış şeması şekil 3.34.'de gösterilmiştir.

-
- Adım 1. Parametre ayarı:** Gerekli sabit parametre setleri ayarlanır. Örneğin, değişkenlerin alt ve üst sınırları, maksimum yineleme sayısı, vb.
- Adım 2. Başlat:** Sürüdeki her örümcek için rastgele başlangıç pozisyonu yerleri belirlenir. Her örümcek için hedef titreşim başlatılır.
- Adım 3.** Her örümceği bir rüzgâr türbini alt kümesine atanır.
- Adım 4. Değerlendirme:** Uygunluk değeri her örümcek için hesaplanır. Sürünün en iyisi tespit edilir ve kaydedilir.
- Adım 5.** Titreşim değeri, titreşim denklemleri (3.1, 3.2 ve 3.3) kullanılarak hesaplanır. Sürünün en iyisi tespit edilir ve kaydedilir.
- Adım 6.** Titreşim karşılaştırma: Alınan en iyi titreşim, hedef titreşim değerleriyle karşılaştırılır.
- Adım 7.** Örümcek maskesi güncellenir.
- Adım 8.** p^{follow} değeri denklem 3.5 ile hesaplanır.
- Adım 9. Hareket:** Her örümcek denklem 3.4 ile yeni bir pozisyona hareket eder ve yeni bir popülasyon elde edilir.
- Adım 10. Transfer fonksiyonu:** Sürekli arama uzayı on farklı transfer fonksiyonu ile ikili arama uzayına dönüştürülür.
- Adım 11. Lojik kapı operatörü:** Yeni aday çözümler üretmek için iki çözüm arasında mantık kapısı operatörü uygulanır ve yeni aday çözümler değerlendirilir.
- Adım 12. Tekrar:** Adım 4'ten başlayarak tüm kriterler durma kriterlerine ulaşana kadar tekrarlanır.
- Adım 13. Çıktı:** En iyi alt küme elde edilir.
-

Şekil 3.33. Rüzgâr türbini yerleşim problemi için BinSSA'nın adımları



Şekil 3.34. Rüzgâr türbini yerleşim problemi için BinSSA'nın akış şeması

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Tüm deneysel sonuçlar 4GB RAM'li 2.30 GHz işlemci sistemi olan Windows 7, 64 bit üzerine kurulan Matlab R2014a üzerinde test edilmiştir. Bu bölüm üç alt bölümden oluşmaktadır.

Birinci bölümde; Sosyal Örümcek Algoritması (SSA) iki farklı sürekli optimizasyon problemi için geliştirilmiştir. Bunlar: Büyük Ölçekli Küresel Optimizasyon (LSGO) problemleri ve Moleküler Potansiyel Enerji Fonksiyonu (MPEF)'dir.

LSGO için; SSA geliştirilerek Sosyal Örümcek Algoritması (ISSA) önerilmiştir. ISSA'ya iki farklı teknik eklenmiştir. Bu teknikler *örümcek patlatması* ve *kâşif örümcek belleği* yöntemleridir. Örümcek patlatma tekniğinde, her iterasyonun en kötü uygunluk değerine sahip örümceği belirlenir ve rastgele üretilmiş yeni bir örümcek ile yer değiştirir. Böylece ISSA sürekli arama uzayında yeni noktaları keşfedebilir. Kâşif örümcek belleği tekniği ile önceki iterasyonlarda ki iyi uygunluk değerine sahip örümceklerin unutulması önlenir. ISSA'nın başarısı, küçük, orta ve yüksek ölçekli klasik benchmark fonksiyonlarında test edilmiştir. Orijinal SSA'nın başarısı, (10, 20 ve 30) boyutlarda CEC2015 üzerinde test edilmiştir. Orijinal SSA'nın başarısı, geniş ölçekli boyutlarda test edilmemiştir. Tez kapsamında ISSA'nın performansı {100, 500 ve 1000} şeklinde geniş ölçekli boyutlarda da test edilmiştir.

MPEF için; SSA performansını artıran yeni dört teknik eklenerek geliştirilmiştir ve ISSA önerilmiştir. Bu teknikler *çaprazlama*, *mutasyon*, *G_{best} yakınsaması* ve *sessiz örümcek* teknikleridir. Her dört teknikte de, ISSA sürekli arama alanında yerel ve küresel arama yapabilme yeteneğini geliştirmiştir. Çaprazlama ve G_{best} yakınsama teknikleri, sürekli arama alanında ISSA'nın yerel arama kapasitesini artırırken, mutasyon ve sessiz örümcek teknikleri sürekli arama alanında ISSA'nın küresel arama kapasitesini arttırmıştır. MPEF, ölçeklenebilir, basitleştirilmiş en önemli gerçek hayat problemlerinden biridir ve bir proteinin 3D yapısını tahmin etmeye çalışır. ISSA, molekülün enerji fonksiyonunun basitleştirilmiş bir modelini en aza indirmek için kullanılmıştır.

İkinci bölümde; SSA ayrık optimizasyon problemlerini çözebilmesi için ayrık bir hale getirilmiş ve Ayrık Sosyal Örümcek Algoritması (DSSA) önerilmiştir. Ayrık niceliklerin optimal olarak düzenlenmesi, gruplanması, sıraya konulması veya seçilmesi ayrık optimizasyon olarak adlandırılmaktadır. DSSA, bir ayrık optimizasyon problemi

olan Gezgin Satıcı Problemi (TSP) üzerinde test edilmiştir. TSP, satıcının şehirler arasında en kısa turu bulmaya çalıştığı iyi bilinen bir kombinasyonlu ayrık optimizasyon problemidir. DSSA ya, *yetenekli örümcek* ve *acemi örümcek* teknikleri eklenerek ayrık arama uzayındaki keşif ve sömürü yetenekleri geliştirilmiştir. Literatürde genellikle küçük ve orta ölçekli TSP örnekleri test için kullanılırken, bu tezde DSSA ile geniş ölçekli TSP örnekleri de test için kullanılmıştır.

Üçüncü bölümde; SSA ikili optimizasyon problemleri için ayrık hale getirilmiştir. İkili optimizasyon, ayrık optimizasyon tekniklerinin özel bir şeklidir. Sürekli gerçek arama alanlarına sahip algoritmalar, değişkenleri ikili değişkenlere dönüştürerek İkili Optimizasyon Problemlerini (BOP) çözebilir. İkili optimizasyonda, çözüm ikili $\{0, 1\}$ değerleriyle sınırlıdır. Sürekli optimizasyon problemleri için geliştirilmiş SSA, ikili optimizasyon problemlerini de çözebilecek şekilde yeniden yapılandırılmıştır. Böylece BinSSA önerilmiştir. Bu tezde BinSSA ile dört farklı BOP çözülmüştür. Bu BOP'ler: Özellik Seçme (FS) problemi, Kapasitesiz Tesis Yerleşim Problemi (UFLP), rüzgâr türbinleri yerleşim problemi ve sürekli optimizasyon görevleridir.

Sürekli optimizasyon görevleri için; BinSSA'da sürekli arama uzayı ikili uzaya çevrilirken, iki S-şekilli (Sigmoidal transfer fonksiyonu ve Modifiye sigmoidal transfer fonksiyonu) ve iki V-şekilli (Tanjant hiperbolik transfer fonksiyonu ve Arktanjan transfer fonksiyonu) transfer fonksiyonları kullanılmıştır. Böylece, BinSSA-Tanh, BinSSA-Sigm, BinSSA-MSigm ve BinSSA-Arctan olarak dört farklı BinSSA varyasyonu oluşturulmuştur. BinSSA'da ikili arama uzayında keşif ve sömürü arasındaki dengeyi sağlamak için aday çözüm üretim aşamasında iki farklı teknik (SimSSA ve LogicSSA) geliştirilmiştir. Bunlar *Jaccard'ın benzerlik ölçümü (SimSSA)* ve "*xor*" mantıksal kapısı (*LogicSSA*) teknikleridir. BinSSA'nın performansı on sekiz farklı tek modlu ve çok modlu karşılaştırma fonksiyonu üzerinde test edilmiştir.

UFLP için; BinSSA ile BOP'lerden Kapasitesiz Tesis Yerleşim Problemi (UFLP) çözülmüştür. UFLP problemleri saf ikili optimizasyon problemlerinden biridir ve literatürde optimum değerleri bilinmektedir. BinSSA, iki farklı UFLP veri seti üzerinde (15 farklı ölçekli Cap veri seti ve 20 farklı ölçekli M* veri seti) üzerinde test edilmiştir. Ayrıca BinSSA'nın performansı CEC2015 benchmark fonksiyonları üzerinde de test edilmiştir. Aday çözüm üretimi için iki farklı yöntem geliştirilmiştir. Bunlar *Dice'in benzerlik ölçümü* ve *xor-maskesi* teknikleridir.

Rüzgâr türbinleri yerleşim problemi için; bir BOP olarak BinSSA ile rüzgâr türbini yerleşim problemi çözülmüştür. Bu tezde, mümkün olan en yüksek rüzgâr enerjisini elde etmek için rüzgâr türbinlerinin bir rüzgâr çiftliğine optimum şekilde yerleştirilmesi incelenmiştir. Dört adet S-şekilli, dört adet V-şekilli ve iki adet mod tabanlı transfer fonksiyonuna göre on adet BinSSA varyasyonu oluşturulmuştur. Aday çözüm üretimi için bir *lojik kapı operatörü* eklenmiş ve böylece BinSSA'da kâşif ve sömürü arasındaki denge kurulmuştur. Bu çalışmada, rüzgâr türbinleri iki farklı ızgara yapısına (10x10 ve 20x20) göre yerleştirilmiş ve sonuçlar incelenmiştir.

Özellik seçme (FS) problemi için; bir BOP olarak BinSSA ile Özellik Seçme (FS) problemi çözülmüştür. Sürekli arama uzayının ikili arama uzayına dönüşümünde dört adet S-şekilli ve dört adet V-şekilli transfer fonksiyonu kullanılmıştır. Kullanılan transfer fonksiyonlarına göre sekiz adet BinSSA varyasyonu oluşturulmuştur (BinSSA1, BinSSA2, BinSSA3, BinSSA4, BinSSA5, BinSSA6, BinSSA7 ve BinSSA8). Aday çözüm üretimi için bir *çaprazlama operatörü* eklenmiş ve böylece BinSSA'da kâşif ve sömürü arasındaki denge kurulmuştur. OR-Lib'den (UCI Machine Learning Repository) alınan düşük ölçekli, orta ölçekli ve büyük ölçekli yirmi bir UCI karşılaştırma veri setleri, özellik seçme probleminde BinSSA performansını ölçmek için kullanılmıştır. Çalışmada bir sınıflandırıcı olarak K-NN ve SVM kullanılmıştır. BinSSA'nın performansı sanatsal algoritmalarla (BBA, BPSO ve BGWO) karşılaştırılarak performansı analiz edilmiştir.

4.1. Sürekli Optimizasyon Görevleri için ISSA Test Sonuçları

4.1.1. Büyük Ölçekli Küresel Optimizasyon (LSGO) için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA) sonuçları

SSA ve ISSA, klasik on dokuz farklı tek modlu ve çok modlu kıyaslama fonksiyonunda test edilmiştir. SSA ve ISSA için testlerde kullanılan ortak parametreler çizelge 4.1'de gösterilmiştir. r_a , p_c ve p_m sabit değerleri orijinal SSA makalesinde seçilmiştir. Testlerde kullanılan tek modlu ve çok modlu kıyaslama fonksiyonları çizelge 4.2'de gösterilmiştir.

Çizelge 4.1. SSA ve ISSA için ortak parametre ayarları

Algoritmalar	Popülasyon miktarı	Boyut	Maksimum yinleme	r _a	p _c	p _m	q	ESM boyutu
SSA	25	{10, 20, 30, 100, 500, 1000}	1000	1	0.7	0.1	-	-
ISSA	25	{10, 20, 30, 100, 500, 1000}	1000	1	0.7	0.1	10	4×N

Çizelge 4.2. Çalışmada kullanılan klasik benchmark fonksiyonları (n:boyut)

İsim	Fonksiyon	Aralık	f _{min}
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Schwefel 1.2	$f_2(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Quartic	$f_3(x) = \sum_{i=1}^n i x_i^4$	$[-1.28, 1.28]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Powell	$f_4(x) = \sum_{i=1}^{n/4} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	$[-4, 5]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^n$	$X^* = (1, \dots, 1);$ $f(X^*) = 0$
Dixon&Price	$f_6(x) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$	$[-10, 10]^n$	$x_i = 2^{-(2^{i-1} - 2)/2^{i-1}};$ $f(X^*) = 0$
Sum of Squares	$f_7(x) = \sum_{i=1}^n i x_i^2$	$[-10, 10]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Zakharov	$f_8(x) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5 i x_i)^2 + (\sum_{i=1}^n 0.5 i x_i)^4$	$[-5, 10]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
sum_of_different powers	$f_9(x) = \sum_{i=1}^n x_i ^{i+1}$	$[-1, 1]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Rotated hyper-Ellipsoid	$f_{10}(x) = \sum_{i=1}^n \sum_{j=1}^i x_j^2$	$[-65536, 65536]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Michalewicz	$f_{11}(x) = - \sum_{i=1}^n \sin(x_i) \sin^{2m} \left(\frac{i x_i^2}{\pi} \right)$	$[0, \pi]^n$	$n=10: f(X^*) = -9,66015;$ $n=20: f(X^*) = -19,6370;$ $n=30: f(X^*) = -29,6309$
Trid	$f_{12}(x) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$	$[-n^2, n^2]^n$	$i=1, 2, \dots, n;$ $x_i = i(n+1-i)$ $f(X^*) = -n(n+4)(n-1)/6$
Levy	$f_{13}(x) = 0.1 \{ \sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4); u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-10, 10]^n$	$X^* = (1, \dots, 1);$ $f(X^*) = 0$
Schwefel	$f_{14}(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500, 500]^n$	$X^* = (420.9687, \dots, 420.9687)$ $f(X^*) = -418,9829$
Rastrigin	$f_{15}(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Ackley	$f_{16}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + \exp(1)$	$[-32, 32]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Griewank	$f_{17}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	$[-600, 600]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Salmon	$f_{18}(x) = 1 - \cos \left(2\pi \sqrt{\sum_{i=1}^n x_i^2} \right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$	$[-100, 100]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$
Noise	$f_{19}(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0, 1)$	$[-1.28, 1.28]^n$	$X^* = (0, \dots, 0);$ $f(X^*) = 0$

Karşılaştırma (benchmark) fonksiyonları türlerine göre iki gruba ayrılır.

- Tek modlu karşılaştırma fonksiyonları (f1-f10)
- Çok modlu karşılaştırma fonksiyonları (f11-f19)

Tek modlu benchmark fonksiyonları sadece bir global optimumu olan ve araştırılan algoritmaların sömürü kapasitesini değerlendirebilen fonksiyonlardır. Sayıları problem büyüklüğü ile katlanarak artan birçok yerel optima içeren çok modlu karşılaştırma fonksiyonları, araştırılan algoritmaların keşif kabiliyetini değerlendirmek için kullanılırlar. Bu fonksiyonlar çeşitli kaynaklardan alınmıştır (Cuevas ve ark., 2013; James ve Li, 2015a; Surjanovic ve Bingham, 2017).

Her algoritmanın performansını değerlendirmek için dört ölçme kriteri kullanılmıştır. Bu ölçüm kriterleri: (1) Ortalama (Mean), (2) Standart sapma (Std.Dev.), (3) Minimum değer (En İyi) ve (4) Maksimum değer (En Kötü)'dür. Çizelge 4.3'de bu ölçme kriterleri gösterilmiştir. Burada N_{run} toplam çalışma sayısını temsil eder.

Çizelge 4.3. Karşılaştırma kriterleri

Ortalama:	$Ortalama = \frac{1}{n_{run}} \times \sum_{i=1}^{n_{run}} Fitness_func_i$
Standart Sapma:	$Std.Dev. = \sqrt{\frac{1}{n_{run}} \sum_{i=1}^{n_{run}} (gbest_i - Ortalama)^2}$
En iyi:	$Fitness_func_Best = \min_{i=1 \dots n_{run}} Fitness_func_Best_i$
En kötü:	$Fitness_func_Worst = \max_{i=1 \dots n_{run}} Fitness_func_Worst_i$

Düşük ve orta ölçekli sürekli optimizasyon problemleri için SSA ve ISSA:

Orijinal SSA ve ISSA, on dokuz farklı karşılaştırma fonksiyonu için üç farklı düşük ölçekli boyutta (10, 20, 30) test edilmiştir. Her kıyaslama fonksiyonu on kez çalıştırılmıştır. Elde edilen kıyaslama fonksiyonu sonuçları için ortalama, standart sapma, en iyi ve en kötü hesaplamaları yapılmıştır. Sonuçlar, her bir boyut için sırasıyla çizelge 4.4, 4.5 ve 4.6'da gösterilmiştir. Üstün performans gösteren sonuçlar kalın font ile işaretlenmiştir. Ortalama ve standart sapma sonuçlarına göre, tek modlu kıyaslama fonksiyonlarında ISSA, f8 hariç her fonksiyonda SSA'dan daha iyi performans elde etmiştir. Çok modlu kıyaslama fonksiyonlarında ise ISSA birçok karşılaştırma fonksiyonunda SSA'dan daha iyi performans elde etmiştir.

Çizelge 4.4. SSA ve ISSA için karşılaştırma sonuçları (Boyut=10)

F	Boyut=10							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
$f1(x)$	6,35E-05	3,10E-05	7.64E-06	1.23E-04	1,28E-05	1,07E-05	3,02E-06	4.12E-05
$f2(x)$	3,03E+01	4,82E+01	2.47E+00	1.68E+02	2,45E+01	2,12E+01	2,36E+00	7.37E+01
$f3(x)$	0,00E+00	0,00E+00	0,00E+00	0.00E+00	0,00E+00	0,00E+00	0,00E+00	0.00E+00
$f4(x)$	3,68E-02	6,33E-02	3.84E-03	2.24E-01	6,15E-03	8,00E-03	1,01E-04	2.89E-02
$f5(x)$	6,46E+01	2,43E+01	2.32E+01	1.04E+02	2,00E+01	1,28E+01	5,47E+00	4.96E+01
$f6(x)$	5,43E+00	1,50E+01	1.24E-01	5.03E+01	1,13E+00	1,54E+00	1,06E-01	5.36E+00
$f7(x)$	1,04E-05	9,12E-06	8.18E-07	2.99E-05	5,34E-06	8,06E-06	2,15E-07	2.55E-05
$f8(x)$	9,04E+00	2,15E+00	5,81E+00	1.27E+01	3,51E+01	6,13E+00	2.24E+01	4.36E+01
$f9(x)$	6,10E-18	1,45E-17	5.47E-22	4.19E-19	6,35E-19	1,68E-18	6,58E-23	5.66E-18
$f10(x)$	3,74E+00	3,64E+00	4.73E-01	1.33E+01	1,98E+00	2,40E+00	6,22E-02	7.80E+00
$f11(x)$	-4,41E+00	4,64E-01	-5.27E+00	-3.81E+00	-5,05E+00	2,28E-01	-5,40E+00	-4.71E+00
$f12(x)$	-6,70E+04	4,06E+04	-1,68E+05	-2.46E+04	-7,16E+01	1,68E+01	-2.84E+03	-1.72E+02
$f13(x)$	1,81E-06	1,34E-06	2.25E-07	4.39E-06	3,46E-07	4,25E-07	4,25E-03	1.11E-06
$f14(x)$	-1,66E+03	5,22E+02	-2.80E+03	-8.57E+02	-1,78E+03	2,79E+02	-2,89E+03	-1.32E+03
$f15(x)$	5,26E-01	4,75E-01	6.37E-02	1.37E+00	1,99E-01	3,98E-01	4,53E-06	9.95E-01
$f16(x)$	1,81E-02	1,74E-02	5.45E-03	6.69E-02	2,14E-03	1,27E-03	8,51E-04	5.20E-03
$f17(x)$	5,71E-02	2,67E-02	2.29E-02	1.01E-01	2,87E-02	1,53E-02	1,50E-03	5.80E-02
$f18(x)$	2,50E+01	2,15E+00	1,80E+00	4.34E+01	4,81E+01	4,01E+00	2.95E+00	3.74E+01
$f19(x)$	0,00E+00	0,00E+00	0,00E+00	0.00E+00	0,00E+00	0,00E+00	0,00E+00	0.00E+00

Çizelge 4.5. SSA ve ISSA için karşılaştırma sonuçları (Boyut=20)

F	Boyut=20							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
$f1(x)$	9,82E-02	4,39E-02	2,49E-02	1.72E-01	1,94E-02	1,08E-02	4.39E-03	3.49E-02
$f2(x)$	1,06E+02	1,83E+02	1.74E+00	6.32E+02	2,22E+01	1,95E+01	1,60E+00	4.85E+01
$f3(x)$	0,00E+00	0,00E+00	0,00E+00	0.00E+00	0,00E+00	0,00E+00	0,00E+00	0.00E+00
$f4(x)$	2,26E+00	1,56E+00	5.03E-01	5.29E+00	2,90E-01	1,90E-01	5,42E-02	7.71E-01
$f5(x)$	6,81E+02	3,35E+02	2.49E+02	1.45E+03	1,78E+02	6,24E+01	9,44E+01	2.81E+02
$f6(x)$	8,52E+00	1,25E+01	1,16E+00	4.50E+01	6,75E+00	4,33E+00	2.74E+00	1.70E+01
$f7(x)$	1,35E-02	7,17E-03	4.52E-03	2.92E-02	1,23E-03	5,33E-04	6,38E-04	2.42E-03
$f8(x)$	8,60E+01	3,19E+01	3.27E+01	1.42E+02	1,76E+02	2,38E+01	1,42E+01	2.15E+02
$f9(x)$	9,54E-16	1,89E-15	1.57E-17	6.35E-15	5,95E-18	1,12E-17	8,26E-21	3.63E-17
$f10(x)$	3,42E+04	2,32E+04	1.17E+04	9.30E+04	7,12E+03	4,42E+03	3,23E+03	1.54E+04
$f11(x)$	-6,60E+00	6,19E-01	-7.71E+00	-5.84E+00	-7,43E+00	1,50E-01	-7,82E+00	-7.28E+00
$f12(x)$	-3,94E+03	1,89E+03	-7,00E+03	-1.78E+03	-2,83E+03	1,43E+03	-5.44E+03	-9.80E+02
$f13(x)$	2,05E-03	1,13E-03	5.80E-04	4.57E-03	1,65E-03	7,74E-04	5,71E-04	3.00E-03
$f14(x)$	-2,36E+03	7,24E+02	-3.79E+03	-1.75E+03	-3,23E+03	1,03E+03	-4,91E+03	-1.85E+03
$f15(x)$	1,56E+01	3,07E+00	1.20E+01	2.23E+01	4,90E+00	1,91E+00	2,09E+00	8.22E+00
$f16(x)$	2,28E-01	8,54E-02	7.54E-02	3.77E-01	1,36E-01	5,18E-02	7,50E-02	2.61E-01
$f17(x)$	3,45E-01	1,08E-01	1.56E-01	4.98E-01	1,15E-01	6,99E-02	3,61E-02	3.00E-01
$f18(x)$	5,62E+01	4,06E+00	5,17E+00	9.64E+01	7,95E+01	5,00E+00	6.16E+00	9.45E+01
$f19(x)$	0,00E+00	0,00E+00	0,00E+00	0.00E+00	0,00E+00	0,00E+00	0,00E+00	0.00E+00

Çizelge 4.6. SSA ve ISSA için karşılaştırma sonuçları (Boyut=30)

F	Boyut=30							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
$f1(x)$	1,60E+00	5,70E-01	9,11E-01	4,01E+00	2,90E-01	9,55E-02	1,37E-01	4,25E-01
$f2(x)$	2,99E+02	3,38E+02	7,77E+01	1,23E+03	2,52E+02	2,50E+02	4,31E+01	8,33E+02
$f3(x)$	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f4(x)$	4,95E+00	3,12E+00	1,84E+00	1,25E+01	1,42E+00	1,09E+00	3,56E-01	3,11E+00
$f5(x)$	2,83E+03	1,40E+03	1,15E+03	5,99E+03	8,80E+02	2,14E+02	6,06E+02	1,22E+03
$f6(x)$	1,64E+01	1,25E+01	7,79E+00	5,21E+01	6,07E+00	2,06E+00	3,04E+00	1,11E+01
$f7(x)$	2,22E-01	9,05E-02	1,41E-01	4,08E-01	5,53E-02	3,48E-02	1,77E-02	1,24E-01
$f8(x)$	2,37E+02	3,76E+01	1,63E+02	2,99E+02	3,87E+02	5,29E+01	2,51E+02	4,60E+02
$f9(x)$	7,13E-15	1,21E-14	8,74E-17	4,20E-14	1,27E-15	2,48E-15	2,51E-17	8,65E-15
$f10(x)$	1,32E+06	4,29E+05	4,44E+05	1,91E+06	4,45E+05	1,34E+05	1,82E+05	5,79E+05
$f11(x)$	-8,59E+00	5,16E-01	-9,39E+00	-7,94E+00	-9,50E+00	2,48E-01	-9,85E+00	-8,94E+00
$f12(x)$	-7,11E+06	3,69E+06	-1,71E+07	-2,62E+06	2,37E+03	3,47E+02	1,69E+03	2,83E+03
$f13(x)$	2,49E-02	1,41E-02	1,02E-02	6,15E-02	5,21E-03	1,97E-03	2,41E-03	8,24E-03
$f14(x)$	-2,66E+03	3,38E+02	-3,15E+03	-1,97E+03	-3,69E+03	3,73E+02	-3,33E+03	-2,15E+03
$f15(x)$	4,48E+01	4,97E+00	3,43E+01	5,16E+01	1,19E+01	2,54E+00	7,12E+00	1,51E+01
$f16(x)$	8,42E-01	2,49E-01	3,47E-01	1,11E+00	2,29E-01	6,02E-02	1,40E-01	3,34E-01
$f17(x)$	9,77E-01	2,99E-02	9,31E-01	1,02E+00	5,83E-01	1,51E-01	3,05E-01	8,75E-01
$f18(x)$	8,84E+01	8,57E+00	8,84E+01	3,35E+02	1,10E+02	1,03E+01	9,68E+01	1,10E+02
$f19(x)$	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00

Wilcoxon signed-rank testi, iki örneğin ortalaması arasındaki anlamlı farklılıkları belirlemeyi amaçlamaktadır (Acılar, 2013). İki algoritmanın çıktılarını karşılaştırmak için kullanılacaksa, test pratik olarak iki algoritmanın karşılıklı davranışlarını değerlendirir. Bu tezde SSA ve ISSA'nın düşük ve orta ölçekli karşılaştırma fonksiyonlarındaki davranışlarını ölçmek için Wilcoxon signed-rank testi kullanılmıştır. SSA ve ISSA her karşılaştırma fonksiyonunda on bağımsız çalışma gerçekleştirmiş ve on örnekle veri setleri elde edilmiştir. Bu veri setlerinde önemli bir fark olup olmadığı Wilcoxon signed-rank testi ile bakılmıştır. Sonuçlar çizelge 4.7'de gösterilmiştir. Wilcoxon signed-rank testinde veri seti sayısı=10 için anlam seviyesi $\alpha=0.05$ 'dir (Acılar, 2013). Sonuçlar iki hipoteze göre değerlendirilir. H_0 hipotezinde $M_{data1}=M_{data2}$ ve H_1 hipotezinde $M_{data1} \neq M_{data2}$ 'dir. H_0 hipotezi $h = 1$ değerlerinde reddedilir ve algoritmalar arasında anlamsal bir fark olduğu söylenir. H_0 hipotezi $h = 0$ değerlerinde kabul edilir ve algoritmalar arasında anlamsal bir fark olmadığı söylenir. Hipotezin p değerleri Matlab R2014a yazılımı kullanılarak hesaplanmıştır.

Sonuçlara göre $f3$ ve $f19$ 'da elde edilen sonuçlar eşittir ve Wilcoxon signed-rank testi uygulanmaz. Elde edilen sonuçlara göre, ISSA ve SSA sezgisel algoritmalarından elde edilen sonuçlar arasında anlamsal bir fark olduğu gösterilmiştir.

Çizelge 4.7. ISSA ve SSA için Wilcoxon signed-rank testi sonuçları (Boyut=(10, 20, 30))

F	n=10		n=20		n=30	
	p	h	p	h	p	h
f1(x)	4,40E-04	1	7,69E-04	1	1,83E-04	1
f2(x)	0,6776	0	0,1620	0	0,5205	0
f3(x)	NaN	0	NaN	0	NaN	0
f4(x)	0,0013	1	3,30E-04	1	7,69E-04	1
f5(x)	7,69E-04	1	0,0013	1	4,40E-04	1
f6(x)	0,0376	1	0,0890	0	1,83E-04	1
f7(x)	1,83E-04	1	5,83E-04	1	1,83E-04	1
f8(x)	1,83E-04	1	1,83E-04	1	2,46E-04	1
f9(x)	0,0073	1	2,46E-04	1	0,9097	0
f10(x)	1,83E-04	1	1,83E-04	1	5,83E-04	1
f11(x)	0,0073	1	1,83E-04	1	3,30E-04	1
f12(x)	0,0452	1	0,5205	0	0,0010	1
f13(x)	1,83E-04	1	1,83E-04	1	1,83E-04	1
f14(x)	0,7337	0	0,2413	0	0,5205	0
f15(x)	2,46E-04	1	0,0073	1	2,46E-04	1
f16(x)	1,83E-04	1	1,83E-04	1	1,83E-04	1
f17(x)	0,0013	1	1,83E-04	1	1,83E-04	1
f18(x)	1,83E-04	1	1,83E-04	1	1,83E-04	1
f19(x)	NaN	0	NaN	0	NaN	0

Geniş ölçekli sürekli optimizasyon problemleri için SSA ve ISSA:

Orijinal SSA ve ISSA, on dokuz farklı karşılaştırma fonksiyonu için üç farklı geniş ölçekli boyutlarda (100, 500, 1000) test edilmiştir. Her kıyaslama fonksiyonu on kez çalıştırılmıştır. Elde edilen kıyaslama fonksiyonu sonuçları için ortalama, standart sapma, en iyi ve en kötü hesaplamaları yapılmıştır. Sonuçlar, her bir boyut için sırasıyla çizelge 4.8, 4.9 ve 4.10'da gösterilmiştir. Üstün performans gösteren sonuçlar kalın font ile işaretlenmiştir.

Çizelge 4.8. SSA ve ISSA için karşılaştırma sonuçları (Boyut=100)

F	Boyut=100							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
f1(x)	5,98E+02	1,24E+02	3,77E+02	8,13E+02	3,05E+02	4,75E+01	2,34E+02	3,85E+02
f2(x)	4,03E+04	1,27E+04	1,95E+04	6,09E+04	5,28E+04	2,81E+04	1,52E+04	1,20E+05
f3(x)	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
f4(x)	1,98E+02	4,97E+01	9,39E+01	2,58E+02	5,52E+01	1,47E+01	3,16E+01	7,57E+01
f5(x)	5,95E+04	1,33E+04	4,39E+04	8,18E+04	1,91E+04	4,41E+03	1,41E+04	2,92E+04
f6(x)	1,55E+03	4,56E+02	9,60E+02	2,70E+03	7,99E+02	2,81E+02	5,22E+02	1,58E+03
f7(x)	2,52E+02	4,44E+01	1,98E+02	3,28E+02	1,13E+02	1,71E+01	8,24E+01	1,42E+02
f8(x)	2,51E+03	2,67E+02	2,16E+03	2,98E+03	2,19E+03	4,01E+02	1,45E+03	2,66E+03
f9(x)	3,91E-13	6,49E-13	4,52E-15	2,20E-12	1,18E-07	9,23E-08	1,32E-08	2,99E-07
f10(x)	6,68E+09	1,13E+09	5,11E+09	8,47E+09	4,29E+09	7,51E+08	3,20E+09	5,71E+09
f11(x)	-2,12E+01	1,64E+00	-2,42E+01	-1,83E+01	-2,26E+01	1,06E+00	-2,55E+01	-2,13E+01
f12(x)	-1,53E+09	7,11E+08	-2,75E+09	-7,39E+08	-8,96E+08	5,81E+08	-2,34E+09	-2,87E+08
f13(x)	4,90E+00	6,06E-01	3,99E+00	5,92E+00	3,03E+00	5,42E-01	2,27E+00	4,22E+00
f14(x)	-4,63E+03	3,65E+02	-5,13E+03	-3,94E+03	-5,21E+03	9,93E+02	-7,56E+03	-5,49E+03
f15(x)	5,30E+02	2,09E+01	4,93E+02	5,68E+02	4,09E+02	2,05E+01	4,02E+02	4,70E+02
f16(x)	4,67E+00	2,87E-01	4,06E+00	5,12E+00	3,69E+00	1,27E-01	3,50E+00	3,97E+00
f17(x)	6,20E+00	1,01E+00	5,07E+00	8,19E+00	3,69E+00	3,89E-01	3,22E+00	4,37E+00
f18(x)	2,23E+02	1,10E+01	2,09E+02	2,45E+02	2,26E+02	3,63E+00	2,25E+02	2,37E+02
f19(x)	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00

Çizelge 4.9. SSA ve ISSA için karşılaştırma sonuçları (Boyut=500)

F	Boyut=500							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
<i>f1(x)</i>	8,22E+04	5,20E+03	7,31E+04	9,09E+04	5,73E+04	3,79E+03	5,26E+04	6,57E+04
<i>f2(x)</i>	4,03E+07	6,04E+06	2,82E+07	4,76E+07	4,23E+07	5,60E+06	3,68E+07	4,79E+07
<i>f3(x)</i>	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f4(x)</i>	1,18E+04	1,41E+03	1,02E+04	1,50E+02	5,92E+03	4,74E+02	5,26E+03	6,80E+03
<i>f5(x)</i>	5,05E+07	3,17E+06	4,68E+07	5,76E+07	1,82E+07	2,22E+06	1,44E+07	2,15E+07
<i>f6(x)</i>	5,34E+06	7,33E+05	4,20E+06	6,68E+06	2,07E+06	3,45E+05	1,44E+06	2,89E+06
<i>f7(x)</i>	1,79E+05	9,40E+03	1,59E+05	1,90E+05	1,23E+05	8,32E+03	1,11E+05	1,39E+05
<i>f8(x)</i>	1,53E+20	2,39E+19	1,20E+20	2,00E+20	1,17E+20	2,50E+19	7,16E+19	1,44E+20
<i>f9(x)</i>	1,88E-04	4,94E-04	4,35E-06	1,67E-03	1,10E-05	9,98E-06	1,18E-06	3,16E-05
<i>f10(x)</i>	6,67E+12	3,41E+11	6,33E+12	7,02E+12	4,29E+12	2,06E+11	4,00E+12	4,74E+12
<i>f11(x)</i>	-7,60E+01	1,35E+00	-7,81E+01	-7,38E+01	-8,44E+01	1,82E+00	-8,75E+01	-8,24E+01
<i>f12(x)</i>	3,39E+11	2,76E+11	-4,88E+11	4,64E+11	-3,22E+11	1,83E+12	-5,81E+12	3,10E+11
<i>f13(x)</i>	4,41E+02	3,98E+01	3,94E+02	4,94E+02	2,56E+02	3,15E+01	2,19E+02	2,29E+02
<i>f14(x)</i>	-1,07E+04	2,20E+03	-1,54E+04	-7,39E+03	-1,11E+04	1,19E+03	-1,19E+04	-1,04E+04
<i>f15(x)</i>	4,64E+03	7,80E+01	4,47E+03	4,74E+03	4,14E+03	2,19E+01	4,13E+03	4,21E+03
<i>f16(x)</i>	1,30E+01	2,41E-01	1,27E+01	1,36E+01	1,11E+01	1,93E-01	1,11E+01	1,17E+01
<i>f17(x)</i>	7,67E+02	4,14E+01	6,91E+02	8,64E+02	5,34E+02	2,73E+01	4,91E+02	5,80E+02
<i>f18(x)</i>	6,02E+02	1,29E+01	5,89E+02	6,32E+02	5,85E+02	1,09E+01	5,80E+02	6,09E+02
<i>f19(x)</i>	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00

Çizelge 4.10. SSA ve ISSA için karşılaştırma sonuçları (Boyut=1000)

F	Boyut=1000							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
<i>f1(x)</i>	3,19E+05	1,50E+04	2,99E+05	3,40E+05	2,28E+05	1,04E+04	2,14E+04	2,52E+05
<i>f2(x)</i>	3,53E+08	2,54E+07	3,60E+08	3,96E+08	3,80E+08	1,99E+07	3,15E+08	4,00E+08
<i>f3(x)</i>	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f4(x)</i>	4,44E+04	2,27E+03	4,10E+04	4,91E+04	2,41E+04	1,34E+03	1,92E+04	2,37E+04
<i>f5(x)</i>	2,72E+08	8,13E+06	2,59E+08	2,83E+08	1,04E+08	3,61E+06	8,62E+07	9,88E+07
<i>f6(x)</i>	5,69E+07	4,16E+06	4,78E+07	6,41E+07	2,20E+07	1,51E+06	1,94E+07	2,47E+07
<i>f7(x)</i>	1,39E+06	4,89E+04	1,32E+06	1,47E+06	9,85E+05	7,51E+04	8,77E+05	1,12E+06
<i>f8(x)</i>	6,17E+22	1,33E+22	3,76E+22	8,27E+22	5,22E+22	7,21E+21	3,56E+22	6,49E+22
<i>f9(x)</i>	1,10E+00	3,71E-01	6,66E-01	1,85E+00	3,38E-04	2,91E-04	2,40E-05	1,02E-03
<i>f10(x)</i>	6,07E+14	5,46E+12	6,01E+14	6,12E+14	2,01E+14	3,42E+13	1,66E+14	2,35E+14
<i>f11(x)</i>	-1,41E+02	2,36E+00	-1,43E+02	-1,36E+02	-1,57E+02	3,02E+00	-1,63E+02	-1,53E+02
<i>f12(x)</i>	6,44E+13	1,33E+13	4,99E+13	9,75E+13	1,69E+13	1,30E+12	1,49E+13	1,92E+13
<i>f13(x)</i>	1,52E+03	7,06E+01	1,43E+03	1,65E+03	1,03E+03	8,22E+01	9,26E+03	1,17E+03
<i>f14(x)</i>	-1,39E+04	3,13E+03	-2,07E+04	-9,88E+03	-1,46E+04	2,14E+03	-2,20E+04	-1,18E+04
<i>f15(x)</i>	1,03E+04	1,11E+02	1,01E+04	1,04E+04	9,29E+03	7,24E+01	9,26E+03	9,45E+03
<i>f16(x)</i>	1,54E+01	1,34E-01	1,51E+01	1,56E+01	1,37E+01	2,58E-02	1,37E+01	1,37E+01
<i>f17(x)</i>	2,84E+03	9,79E+01	2,67E+03	3,02E+03	1,93E+03	7,57E+01	1,88E+03	2,12E+03
<i>f18(x)</i>	8,77E+02	1,22E+01	8,54E+02	8,89E+02	8,57E+02	2,06E+01	8,11E+02	8,54E+02
<i>f19(x)</i>	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00

Ortalama ve standart sapma sonuçlarına göre göre, tek modlu kıyaslama fonksiyonlarında ISSA, *f2* ve *f9* hariç her fonksiyonda SSA'dan daha iyi performans göstermiştir. Çok modlu kıyaslama fonksiyonlarında ISSA birçok karşılaştırma fonksiyonunda SSA'dan daha iyi performans göstermiştir.

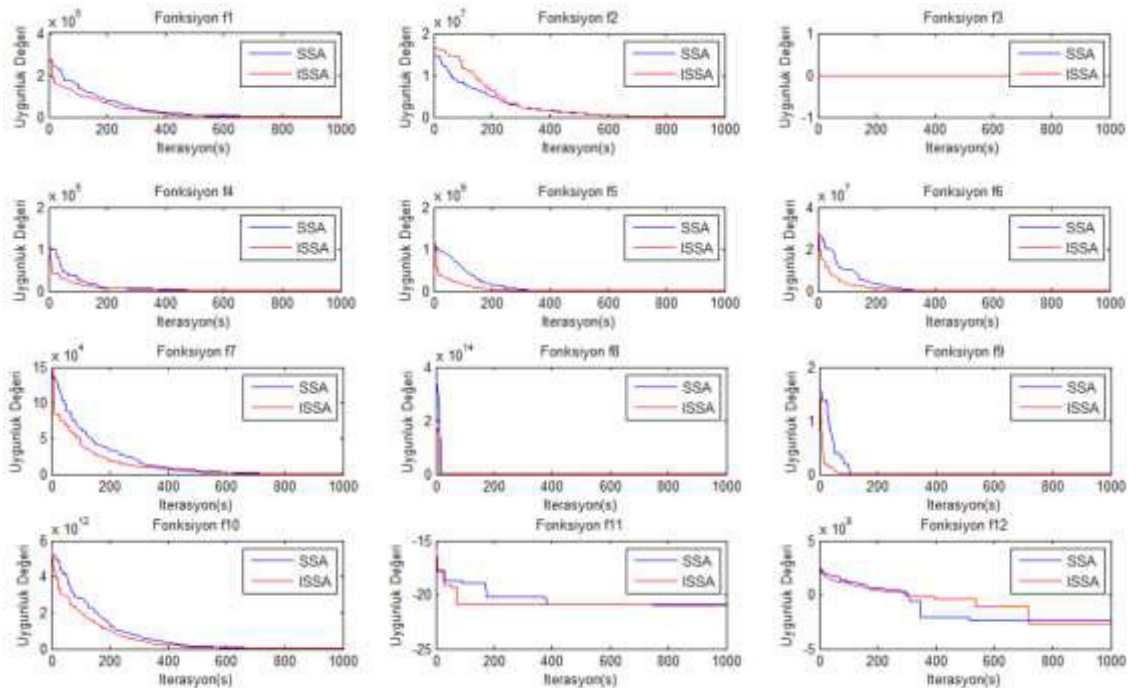
SSA ve ISSA'nın geniş ölçekli karşılaştırma fonksiyonlarındaki davranışlarını ölçmek için Wilcoxon signed-rank testi kullanılmıştır. Sonuçlar çizelge 4.11'de gösterilmiştir. *f3* ve *f19*'da elde edilen sonuçlar eşittir ve Wilcoxon signed-rank testi

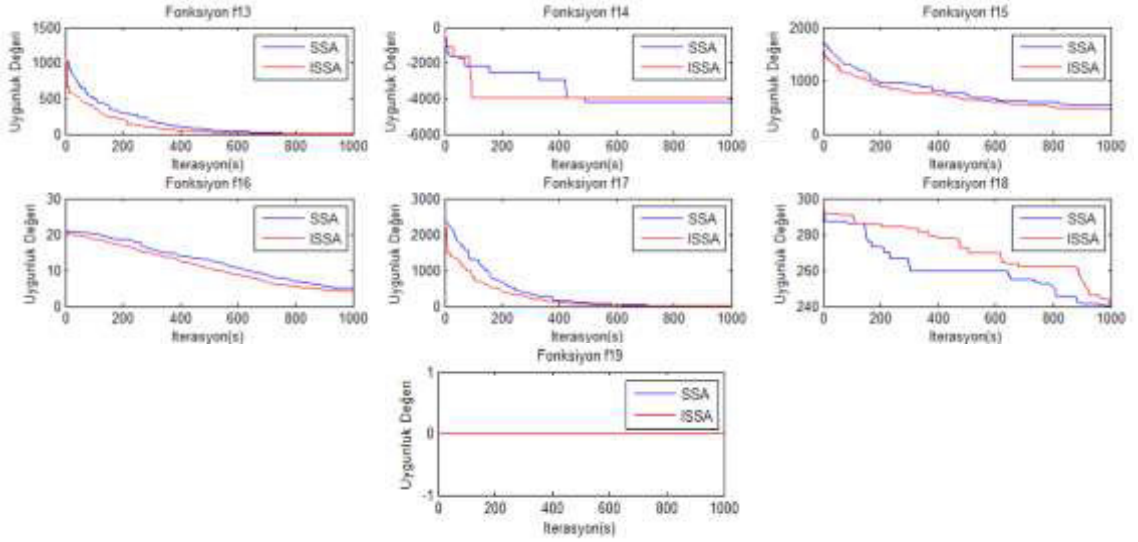
uygulanmaz. Elde edilen sonuçlara göre, ISSA ve SSA sezgisel algoritmalarından elde edilen sonuçlar arasında anlamsal bir fark olduğu gösterilmiştir.

Çizelge 4.11. SSA ve ISSA için Wilcoxon signed-rank testi sonuçları (Boyut=(100, 500, 1000))

F	n=100		n=500		n=1000	
	p	h	p	h	p	h
$f1(x)$	3,30E-04	1	1,83E-04	1	1,83E-04	1
$f2(x)$	0,3847	0	0,4659	0	0,2201	0
$f3(x)$	NaN	0	NaN	0	NaN	0
$f4(x)$	1,83E-04	1	1,83E-04	1	8,75E-05	1
$f5(x)$	1,83E-04	1	1,83E-04	1	1,73E-04	1
$f6(x)$	0,0010	1	1,83E-04	1	1,83E-04	1
$f7(x)$	1,83E-04	1	1,50E-04	1	1,83E-04	1
$f8(x)$	0,1041	0	0,0173	1	0,1212	0
$f9(x)$	1,83E-04	1	0,1859	0	1,83E-04	1
$f10(x)$	3,30E-04	1	1,45E-04	1	8,45E-05	1
$f11(x)$	0,0539	0	1,83E-04	1	1,83E-04	1
$f12(x)$	0,0312	1	0,0022	1	1,83E-04	1
$f13(x)$	3,30E-04	1	1,83E-04	1	1,83E-04	1
$f14(x)$	1,11E-04	1	0,2088	0	0,4274	0
$f15(x)$	8,75E-05	1	8,75E-05	1	1,11E-04	1
$f16(x)$	1,83E-04	1	8,75E-05	1	1,10E-04	1
$f17(x)$	1,83E-04	1	1,83E-04	1	1,50E-04	1
$f18(x)$	0,1042	0	0,0056	1	1,21E-04	1
$f19(x)$	NaN	0	NaN	0	NaN	0

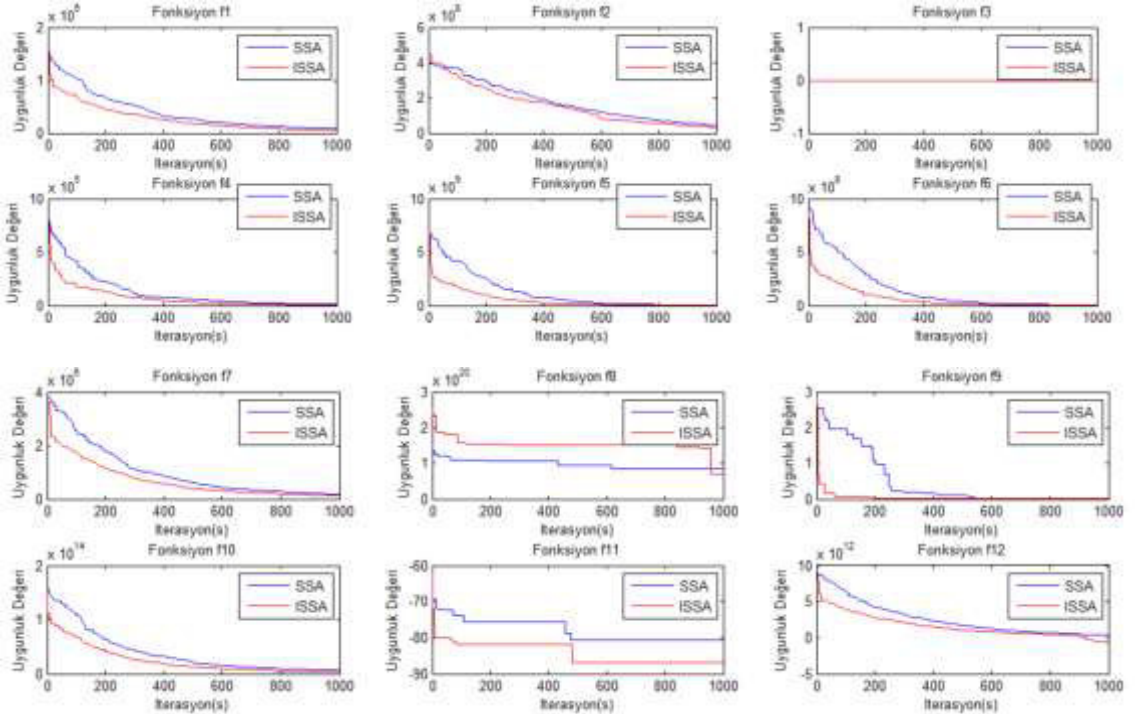
Orijinal SSA ve ISSA için büyük ölçekli boyut = {100}, popülasyon boyutu = 25 ve maksimum yineleme = 1000 parametre değerleri seçilerek on dokuz tek modlu ve çok modlu karşılaştırma fonksiyonu için elde edilen uygunluk değeri karşılaştırma sonuçları şekil 4.1'de gösterilmiştir.

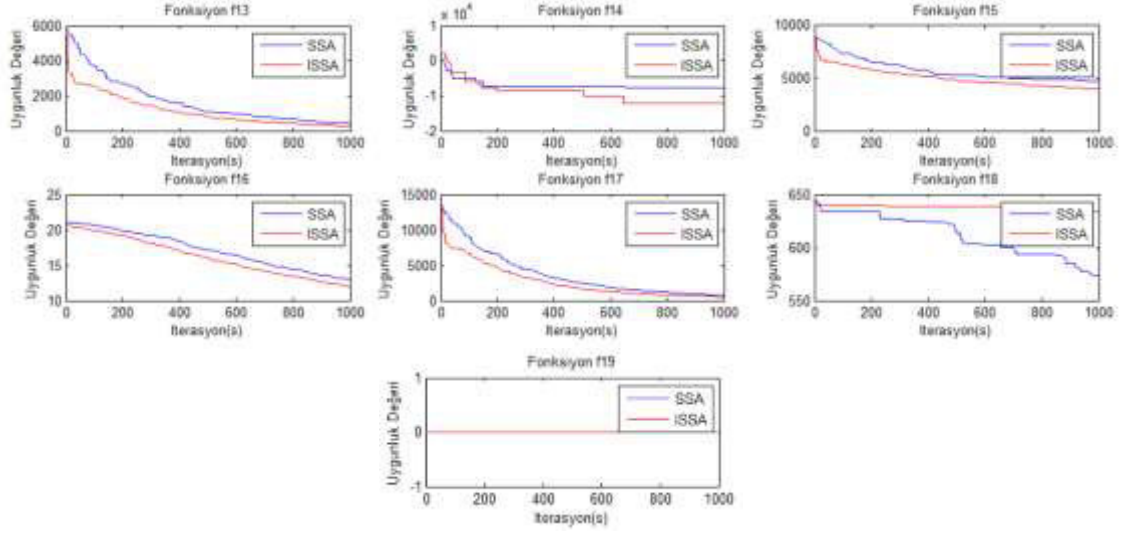




Şekil 4.1. SSA ve ISSA için yakınsama grafikleri (f1-f19) (boyut=100)

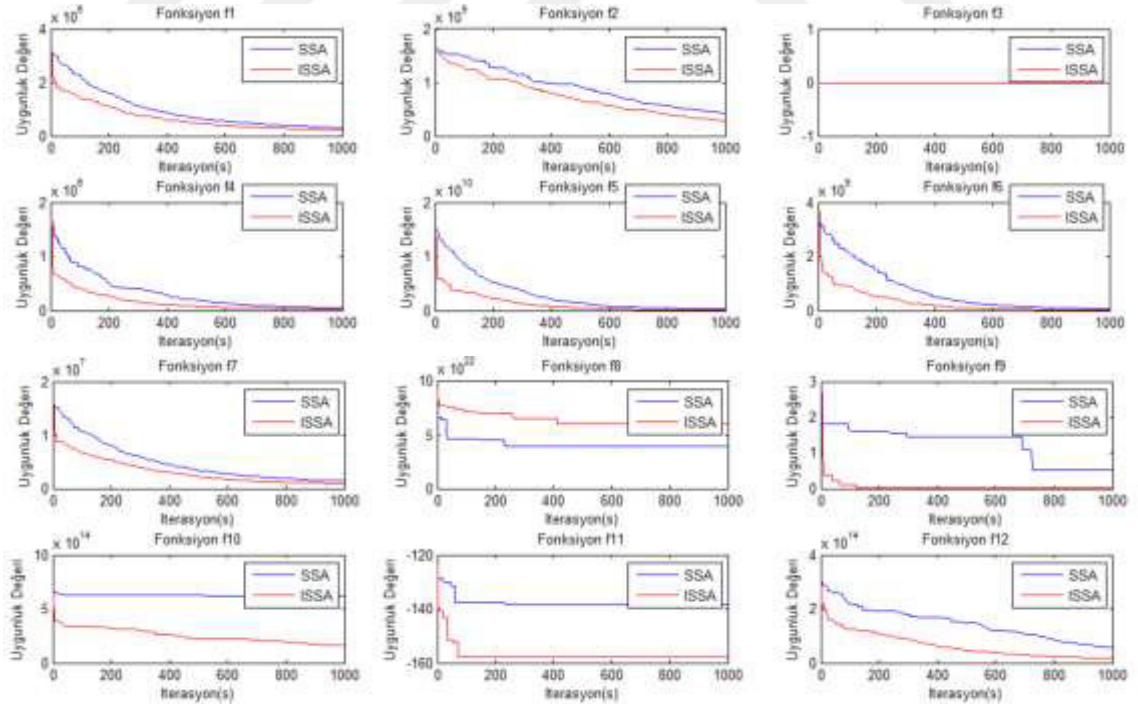
Orijinal SSA ve ISSA için büyük ölçekli boyut = {500}, popülasyon boyutu = 25 ve maksimum yineleme = 1000 parametre değerleri seçilerek on dokuz tek modlu ve çok modlu karşılaştırma fonksiyonu için elde edilen uygunluk değeri karşılaştırma sonuçları şekil 4.2'de gösterilmiştir.

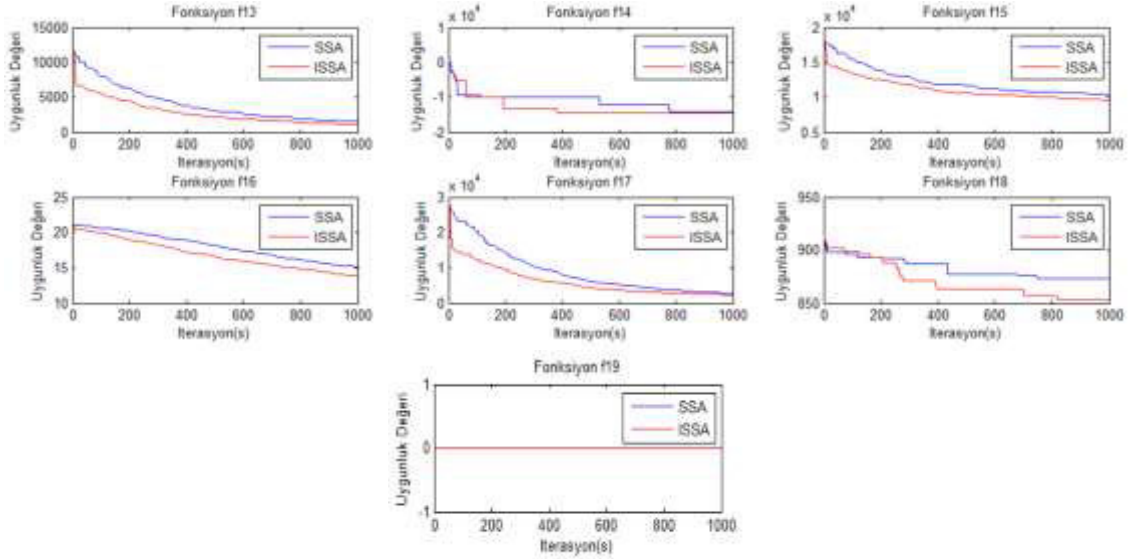




Şekil 4.2. SSA ve ISSA için yakınsama grafikleri (f1-f19) (boyut=500)

Orijinal SSA ve ISSA için büyük ölçekli boyut = {1000}, popülasyon boyutu = 25 ve maksimum yineleme = 1000 parametre değerleri seçilerek on dokuz tek modlu ve çok modlu karşılaştırma fonksiyonu için elde edilen uygunluk değeri karşılaştırma sonuçları şekil 4.3'de gösterilmiştir.





Şekil 4.3. SSA ve ISSA için yakınsama grafikleri (f1-f19) (boyut=1000)

ISSA'nın literatürdeki algoritmalar ile karşılaştırması:

ISSA algoritması iyi bilinen sezgisel algoritmalar ve literatürden seçilen yeni algoritmalar ile karşılaştırılmıştır. Karşılaştırma amacıyla seçilmiş, literatürde iyi bilinen sezgisel algoritmalar ve yakın zamanda geliştirilen algoritmalar: ABC, PSO, DE, SalpSA, CMA-ES, GSA, FA ve SFLA'dır. Çizelge 4.12'de karşılaştırma algoritmaları için ortak kısaltmalar gösterilmiştir.

Çizelge 4.12. Karşılaştırma algoritmaları birinci grup için ortak kısaltmalar

Anahtar	Metot ismi	Referans
ABC	Yapay Arı Kolonisi	(Liang ve ark., 2006)
PSO	Parçacık Sürüşü Optimizasyonu	(Liang ve ark., 2006)
DE	Diferansiyel Evrim	(Liang ve ark., 2006)
SalpSA	Salp Sürüşü Algoritması	(Liang ve ark., 2006)
CMA-ES	Kovaryans Matris Adaptasyon Evrim Stratejisi	(Liang ve ark., 2006)
GSA	Yerçekimi Arama Algoritması	(Liu ve ark., 2019)
FA	Ateşböceği Algoritması	(Liu ve ark., 2019)
SFLA	Karışık Kurbağa Sıçrayan Algoritması	(Liu ve ark., 2019)
ISSA	Geliştirilmiş Sosyal Örümcek Algoritması	

Literatürle kıyaslama çalışması 1:

ABC, PSO, DE, SalpSA, CMA-ES ve ISSA algoritmaları, büyük ölçekli boyutlarda (boyut = 1000) iki farklı kritere göre karşılaştırılmıştır. Bunlar ortalama değeri ve Std.Dev.'i temsil eden standart sapma değeridir. Algoritmaları karşılaştırmak için kullanılan parametre setleri çizelge 4.13'de gösterilmiştir. Karşılaştırma sonuçları

çizelge 4.14’de gösterilmiştir. En iyi sonuçlar kalın olarak işaretlenmiştir. Karşılaştırma sonuçlarına göre, ISSA, f9 ve f15 hariç, seçilen tüm kıyaslama fonksiyonlarında son derece iyi bir performans göstermiştir.

Çizelge 4.13. Karşılaştırma algoritmaları birinci grup için parametre ayarları

Algoritmalar	Popülasyon büyüklüğü	Maksimum yineleme	Boyut	Max_FEs
ABC	100	1000	1000	100000
PSO	100	1000	1000	100000
DE	100	1000	1000	100000
SalpSA	30	1000	1000	30000
CMA-ES	240	500	1000	120000
ISSA	30	1000	1000	30000

Çizelge 4.14. ISSA, ABC, PSO, DE, SalpSA ve CMA-ES karşılaştırma sonuçları (Boyut={1000})

F	Özellikler	ABC	PSO	DE	SalpSA	CMA-ES	ISSA
f1	Ortalama	3,08E+06	3,52E+04	7,34E+05	1,92E+05	9,75E+03	3,11E+03
	Std.Dev.	4,06E+04	4,83E+03	2,49E+04	1,02E+04	6,64E+02	1,25E+02
f2	Ortalama	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-
f3	Ortalama	2,38E+05	7,54E+01	3,74E+04	1,08E+03	5,33E+01	0,00E+00
	Std.Dev.	3,80E+03	1,79E+01	1,45E+03	8,34E+01	1,02E+01	0,00E+00
f4	Ortalama	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-
f5	Ortalama	1,46E+10	7,00E+06	3,18E+09	7,85E+07	8,03E+05	9,21E+04
	Std.Dev.	2,64E+08	9,97E+05	1,78E+08	6,39E+07	1,31E+05	9,81E+03
f6	Ortalama	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-
f7	Ortalama	1,52E+07	1,43E+05	2,51E+06	8,56E+05	1,40E+05	8,07E+03
	Std.Dev.	2,08E+05	1,51E+04	6,44E+04	6,48E+05	7,42E+03	4,43E+03
f8	Ortalama	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-
f9	Ortalama	2,29E+00	6,73E-36	2,43E+00	5,69E-07	5,27E+00	5,35E-06
	Std.Dev.	3,55E-01	1,21E-35	4,39E-01	2,06E-06	1,74E+00	5,35E-06
f10	Ortalama	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-
f11	Ortalama	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-
f12	Ortalama	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-
f13	Ortalama	1,87E+05	3,54E+03	9,38E+04	9,34E+03	2,14E+04	5,03E+02
	Std.Dev.	4,61E+03	4,63E+02	1,12E+03	1,56E+03	1,44E+03	9,25E+01
f14	Ortalama	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-
f15	Ortalama	1,77E+04	3,57E+03	1,35E+04	6,32E+03	1,01E+04	8,50E+03
	Std.Dev.	1,13E+02	1,25E+02	8,59E+01	1,41E+02	1,30E+02	3,87E+02
f16	Ortalama	2,11E+01	9,74E+00	1,87E+01	1,43E+01	5,65E+00	2,38E+00
	Std.Dev.	6,61E-03	1,54E-01	5,55E-02	1,24E-01	8,38E-02	4,44E-16
f17	Ortalama	2,77E+04	2,89E+02	6,44E+03	1,67E+03	8,68E+01	2,01E+01
	Std.Dev.	6,32E+02	2,95E+01	1,33E+02	4,52E+01	5,46E+00	5,25E-01
f18	Ortalama	1,77E+02	2,64E+01	9,91E+01	5,29E+01	1,42E+01	7,30E+02
	Std.Dev.	1,31E+00	5,81E-01	9,80E-01	7,44E-01	8,23E-01	1,09E-01
f19	Ortalama	2,38E+05	1,28E+02	3,65E+04	1,23E+03	5,73E+01	0,00E+00
	Std.Dev.	2,60E+03	3,60E+01	6,21E+02	9,34E+01	4,09E+00	0,00E+00

Literatürle kıyaslama çalışması 2:

GSA, FA, SFLA ve ISSA algoritmaları büyük ölçekli boyutlarda (boyut = 500) iki farklı kritere (ortalama ve standart sapma) göre karşılaştırılmıştır. Karşılaştırılan

algoritmalar için popülasyon büyüklüğü değeri 50, maksimum yineleme değeri 1000 ve boyut değeri 500 eşit olarak belirlenmiştir. Algoritmaları karşılaştırmak için kullanılan parametre setleri çizelge 4.15’de karşılaştırma sonuçları ise çizelge 4.16’da gösterilmiştir. En iyi sonuçlar kalın olarak işaretlenmiştir. Karşılaştırma sonuçlarına göre, ISSA karşılaştırma için seçilen tüm kıyaslama fonksiyonlarında son derece iyi bir performans göstermiştir.

Çizelge 4.15. Karşılaştırma algoritmaları ikinci grup için parametre ayarları

Algoritmalar	Popülasyon büyüklüğü	Maksimum yineleme	Boyut	Max_FEs
GSA	50	1000	500	50000
FA	50	1000	500	50000
SFLA	50	1000	500	50000
ISSA	50	1000	500	50000

Çizelge 4.16. ISSA, GSA, FA ve SFLA karşılaştırma sonuçları (Boyut={500})

F	Özellikler	GSA	FA	SFLA	ISSA
f1	Ortalama	2,28E+04	1,96E+04	3,39E+05	3,03E+04
	Standart Sapma	1,16E+03	8,11E+03	3,13E+03	9,04E+01
f2	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f3	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f4	Ortalama	2,37E+03	4,79E+04	5,32E+05	3,02E+03
	Standart Sapma	3,50E+02	2,95E+04	1,98E+04	1,01E+02
f5	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f6	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f7	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f8	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f9	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f10	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f11	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f12	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f13	Ortalama	9,68E+02	3,43E+03	1,48E+04	1,62E+02
	Standart Sapma	1,66E+02	1,80E+03	2,43E+02	9,54E+00
f14	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f15	Ortalama	1,53E+03	4,15E+03	8,17E+03	4,21E+03
	Standart Sapma	7,83E+01	5,10E+02	8,34E+01	3,08E+01
f16	Ortalama	6,95E+00	1,95E+01	2,10E+01	6,72E+00
	Standart Sapma	6,72E-02	3,93E-01	1,22E-02	2,49E-02
f17	Ortalama	4,40E+03	1,75E+03	3,43E+02	3,36E+01
	Standart Sapma	7,77E+01	7,94E+02	4,37E+00	3,22E+00
f18	Ortalama	-	-	-	-
	Standart Sapma	-	-	-	-
f19	Ortalama	1,43E+02	4,38E+03	4,67E+04	0,00E+00
	Standart Sapma	1,58E+01	2,03E+03	9,73E+02	0,00E+00

4.1.2. Moleküler Potansiyel Enerji Fonksiyonu (MPEF) problemi için Geliştirilmiş Sosyal Örümcek Algoritması (ISSA) sonuçları

Bu tezde SSA, performansını artıran yeni dört teknik eklenerek geliştirilmiştir ve ISSA önerilmiştir. Bu teknikler *çaprazlama (cr)*, *mutasyon (mr)*, *G_{best} yakınsaması (gr)* ve *sessiz örümcek* teknikleridir. Her dört teknikte de, ISSA sürekli arama alanında yerel ve küresel arama yapabilme yeteneğini geliştirilmiştir. Çaprazlama ve G_{best} yakınsama teknikleri, sürekli arama alanında ISSA'nın yerel arama kapasitesini artırırken, mutasyon ve sessiz örümcek teknikleri sürekli arama alanında ISSA'nın küresel arama kapasitesini arttırmıştır. ISSA önce klasik on üç tek modlu ve çok modlu kıyaslama fonksiyonlarında çalıştırılmış daha sonra MPEF üzerinde test edilmiştir. Klasik on üç tek modlu ve çok modlu kıyaslama fonksiyonları çizelge 4.17'de gösterilmiştir.

Çizelge 4.17. Deneysel çalışmalar için klasik kıyaslama fonksiyonları

İsim	Fonksiyon	Aralık	f_{min}
$f1$	$\sum_{i=1}^n x_i^2$	$[-100,100]^n$	$f(X^*)=0$
$f2$	$\sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10,10]^n$	$f(X^*)=0$
$f3$	$\sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100,100]^n$	$f(X^*)=0$
$f4$	$\max_i x_i , 1 \leq i \leq n$	$[-100,100]^n$	$f(X^*)=0$
$f5$	$\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$	$f(X^*)=0$
$f6$	$\sum_{i=1}^n ([x_i + 0.5])^2$	$[-100,100]^n$	$f(X^*)=0$
$f7$	$\sum_{i=1}^n ix_i^4 + \text{random}[0,1)$	$[-1.28,1.28]^n$	$f(X^*)=0$
$f8$	$\sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500,500]^n$	$f(X^*)=-418.9829 \times n$
$f9$	$\sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]^n$	$f(X^*)=0$
$f10$	$-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + \exp(1)$	$[-32,32]^n$	$f(X^*)=0$
$f11$	$\frac{1}{4000}\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^n$	$f(X^*)=0$
$f12$	$1 - \cos\left(2\pi\sqrt{\sum_{i=1}^n x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^n x_i^2}$	$[-100,100]^n$	$f(X^*)=0$
$f13$	$-\sum_{i=1}^n \sin(x_i) \sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$	$[0,\pi]^n$	n=30: $f(X^*)=-29,6309$; n=100: $f(X^*)=-99.6201940166$; n=400: $f(X^*)=-399,616795138572$; n=1000: $f(X^*)=-999,6161052093$

Her algoritmanın performansını değerlendirmek için dört ölçü kriteri kullanılmıştır. Bu ölçüm kriterleri: (1) Ortalama (Mean), (2) Standart sapma (Std.Dev.), (3) Minimum değer (En İyi) ve (4) Maksimum değer (En Kötü)'dür. Bunlar çizelge 4.18'de gösterilmiştir. Burada N_{run} toplam bağımsız çalışma sayısını temsil eder. SSA ve ISSA için kullanılan ortak parametre ayarları çizelge 4.19'da gösterilmiştir.

Kıyaslama fonksiyonları tiplerine göre iki gruba ayrılır. Bu fonksiyonlar çeşitli kaynaklardan elde edilmiştir (Cuevas ve ark., 2013; James ve Li, 2015a; Surjanovic ve Bingham, 2017; Tawhid ve Ali, 2017b).

- *Grup 1:* Tek modlu fonksiyonlar ($f1-f7$)
- *Grup 2:* Çok modlu fonksiyonlar ($f8-f13$)

Çizelge 4.18. Performans ölçütleri

Ortalama:	$Ortalama = \frac{1}{n_{run}} \times \sum_{i=1}^{n_{run}} Fitness_func_i$
Standart Sapma:	$Std.Dev. = \sqrt{\frac{1}{n_{run}} \sum_{i=1}^{n_{run}} (gbest_i - Ortalama)^2}$
En iyi:	$Fitness_func_Best = \min_{i=1 \dots n_{run}} Fitness_func_Best_i$
En kötü:	$Fitness_func_Worst = \max_{i=1 \dots n_{run}} Fitness_func_Worst_i$

Çizelge 4.19. ISSA ve SSA için parametre ayarları

Parametreler	Değerler
Popülasyon miktarı (N)	25
Boyut (n)	{30, 100, 400, 1000}
Maksimum yineleme	1000
Çaprazlama oranı (cr)	%90
Mutasyon oranı (mr)	%10
G _{best} yakınsama oranı (gr)	%40
Bağımsız çalışma sayısı (N _{run})	20
Diğer parametreler	r _a =1; p _c =0,7; p _m =0,1 (James ve Li, 2015a)

Çaprazlama (cr), mutasyon (mr) ve G_{best} yakınsaması (gr) oranlarının belirlenmesi:

İdeal çaprazlama oranı (cr), mutasyon oranı (mr) ve G_{best} yakınsama oranını (gr) belirlemek için, popülasyon boyutu = 25, boyut = 30 ve maksimum yineleme = 1000 parametre değerleri sabit seçilerek, çaprazlama oranı (cr), mutasyon oranı (mr) ve G_{best} yakınsama oranını (gr) rastgele seçilen beş farklı tek modlu karşılaştırma

fonksiyonlarında ($f1$, $f2$, $f4$, $f5$ ve $f7$) ve rastgele seçilen beş farklı çok modlu karşılaştırma fonksiyonlarında ($f8$, $f9$, $f10$, $f11$ ve $f13$) öngörülen bazı kombinasyonlar şeklinde test edilmiştir. Her kıyaslama fonksiyonu 20 kez çalıştırılmıştır. Çizelgeler 4.20 – 4.22’de sonuçlar gösterilmiştir. Sonuçlara göre, uygun çaprazlama oranı (cr) % 90, mutasyon oranı (mr) % 10 ve G_{best} yakınsaması oranı (gr) % 40 olarak belirlenmiştir.

Çizelge 4.20. Çaprazlama (cr) oranının belirlenmesi

F	Çaprazlama oranı (cr)								
	%10	%20	%30	%40	%50	%60	%70	%80	%90
$f1$	27004,5	26156,190	17520,110	18324,940	10140,220	8790,218	7135,412	3347,967	1175,560
$f2$	66,2013	69,99781	55,3307	53,31256	35,07983	37,10695	20,20037	22,91025	11,5507
$f4$	49,8576	45,78515	35,00857	28,66598	22,04498	21,00766	19,00288	20,91694	19,2189
$f5$	4753611	30431267	20364778	9716929	5907726	2776247	970753,5	1060099	38362,99
$f7$	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
$f8$	-3112,39	-2690,11	-2947,04	-2547,16	-3073,51	-2572,21	-3227,95	-2949,12	-2693,33
$f9$	244,970	257,9103	188,5917	152,5475	112,2209	92,60459	64,57314	42,30282	36,73263
$f10$	18,9244	17,49768	17,00651	17,20581	15,78174	14,79526	13,53442	11,89024	10,78819
$f11$	283,20	232,64	188,11	146,74	84,33	78,038	54,262	23,681	9,3741
$f13$	-8,788	-9,000	-8,6787	-9,16	-8,730	-9,338	-8,904	-9,106	-9,189

Çizelge 4.21. Mutasyon (mr) oranının belirlenmesi

F	Mutasyon oranı (mr)								
	%10	%20	%30	%40	%50	%60	%70	%80	%90
$f1$	0,394634	0,702821	0,758151	0,617514	0,702059	0,671051	0,751396	0,905282	0,814592
$f2$	0,141074	0,134244	0,168751	0,210592	0,1827	0,199393	0,208086	0,208631	0,187357
$f4$	18,04599	21,89031	23,99734	24,45129	26,38615	25,51669	25,80048	24,79854	25,72667
$f5$	282,6736	313,285	327,3802	383,4963	331,8956	429,427	360,3212	383,9664	424,8467
$f7$	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
$f8$	-3325,990	-4201,030	-4000,200	-4730,400	-4522,850	-5123,600	-4961,060	-4514,820	-4711,540
$f9$	56,78534	53,34603	49,45195	46,10922	45,66494	47,71347	50,04745	45,39075	46,79847
$f10$	0,267519	0,316517	0,354352	0,340763	0,364685	0,317098	0,30045	0,390763	0,412149
$f11$	0,679518	0,745935	0,752064	0,819712	0,825792	0,84922	0,777499	0,923743	0,826685
$f13$	-8,9389	-9,15235	-9,13946	-8,75084	-9,03193	-9,09553	-9,04802	-9,04717	-8,79987

Çizelge 4.22. G_{best} yakınsaması oranının belirlenmesi

F	G_{best} yakınsaması (gr)								
	%10	%20	%30	%40	%50	%60	%70	%80	%90
<i>f1</i>	0,2992	0,2075	0,1135	0,1014	0,3617	2,5722	1,4404	4,1672	6,6827
<i>f2</i>	0,0869	0,0541	0,0385	0,0363	0,0250	0,0482	0,1329	0,2004	0,3742
<i>f4</i>	15,3631	12,6156	10,9124	9,0924	8,6100	7,9092	8,8164	9,8505	8,7613
<i>f5</i>	252,0041	287,4710	230,5190	141,4368	317,3710	360,8188	708,5943	481,9163	643,2990
<i>f7</i>	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
<i>f8</i>	-6,33e+03	-5,08e+03	-7,6e+03	-8,2e+03	-1,0e+04	-1,2e+04	-1,0e+04	-1,3e+04	-1,4e+04
<i>f9</i>	50,7249	49,6798	38,4998	32,6286	25,8040	16,9285	9,2235	7,4238	5,7207
<i>f10</i>	0,1855	0,1092	0,1908	0,3017	0,6169	0,8428	1,3251	1,3012	1,7077
<i>f11</i>	0,5491	0,4383	0,3003	0,1847	0,3639	0,5313	0,8443	1,0359	1,0551
<i>f13</i>	-9,4599	-8,8794	-9,1509	-8,6585	-9,2841	-8,9549	-8,5483	-8,9557	-8,5900

SSA ve ISSA'nın düşük ve yüksek ölçekli boyutlarda performansı:

Orijinal SSA ve önerilen ISSA, düşük ölçekli ve büyük ölçekli üç farklı boyutta (boyut = 30, 100, 1000), on üç farklı tek modlu ve çok modlu karşılaştırma fonksiyonlarında test edilmiştir. Her kıyaslama fonksiyonu 20 kez çalıştırılmıştır. Elde edilen kıyaslama fonksiyonu sonuçları için ortalama, standart sapma, en iyi ve en kötü hesaplamaları yapılmıştır. Sonuçlar çizelgeler 4.23 ve 4.25'de gösterilmiştir. Sonuçlara göre, ISSA tek modlu ve çok modlu kıyaslama fonksiyonlarında SSA'dan daha iyi performans göstermiştir.

Çizelge 4.23. Karşılaştırma fonksiyonlarında SSA ve ISSA'nın performans karşılaştırması (Boyut=30)

F	Boyut=30							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
<i>f1(x)</i>	1,49E+00	7,48E-01	1,25E+00	2,55E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f2(x)</i>	5,23E-01	3,53E-01	4,11E-01	5,55E-01	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f3(x)</i>	2,51E+02	7,02E+01	2,23E+02	3,51E+03	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f4(x)</i>	3,42E+01	3,18E+00	3,12E+01	3,99E+01	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f5(x)</i>	4,79E+02	1,04E+02	4,55E+02	4,88E+02	1,78E+02	6,16E+01	1,28E+02	1,78E+03
<i>f6(x)</i>	4,27E+00	9,43E-01	4,00E+00	4,85E+00	1,23E+00	4,58E-02	1,10E+00	1,55E+00
<i>f7(x)</i>	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f8(x)</i>	-2,80E+03	9,54E+02	-2,90E+03	-2,24E+03	-8,73E+03	2,67E+03	-8,95E+03	-8,73E+02
<i>f9(x)</i>	4,45E+01	7,80E+00	4,35E+01	4,82E+01	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f10(x)</i>	6,75E-01	2,14E-01	5,75E-01	6,75E-00	8,88E-16	0,00E+00	8,38E-19	8,45E-15
<i>f11(x)</i>	9,96E-01	3,36E-02	6,96E-01	10,86E-01	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f12(x)</i>	8,52E+01	5,76E+00	8,52E+00	7,32E+02	0,00E+00	0,00E+00	0,00E+00	0,00E+00
<i>f13(x)</i>	-8,87E+00	7,17E-01	-8,90E+00	-7,27E+00	-9,03E+00	7,27E-01	-9,23E+00	-8,83E+00

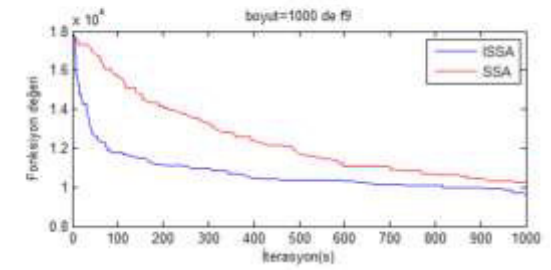
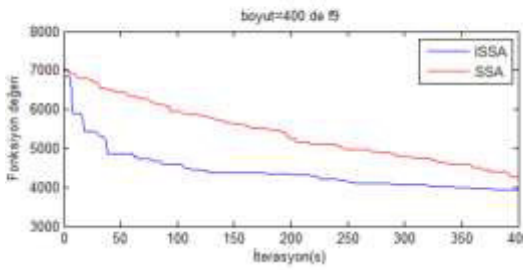
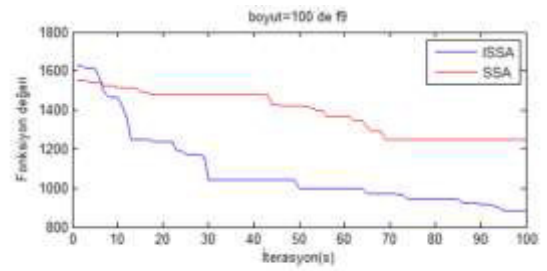
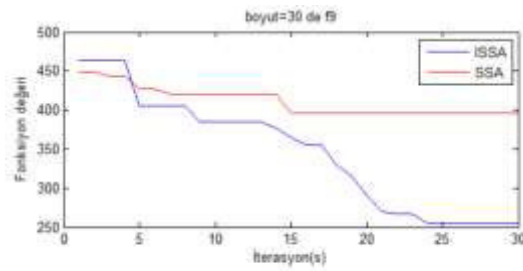
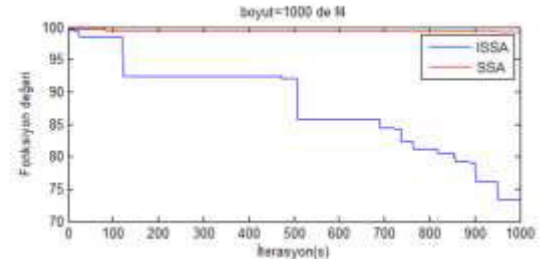
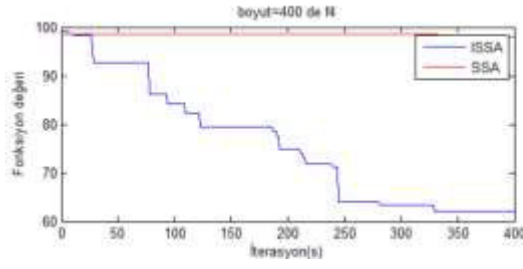
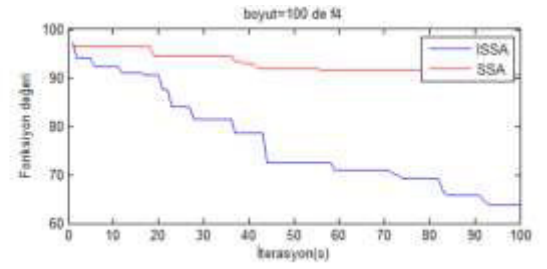
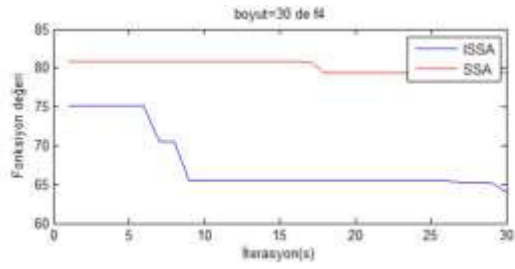
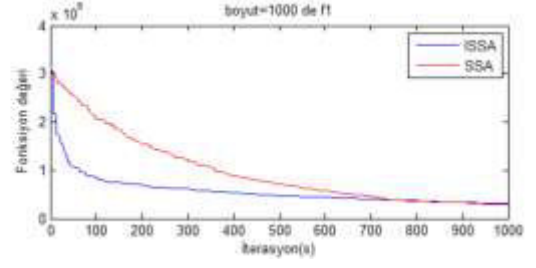
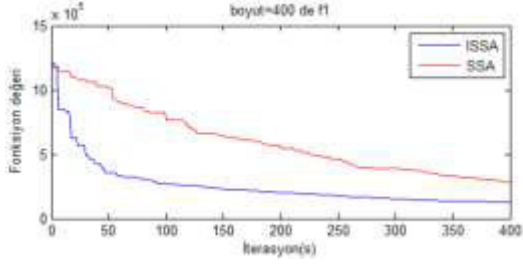
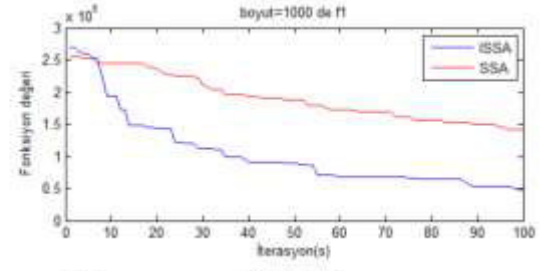
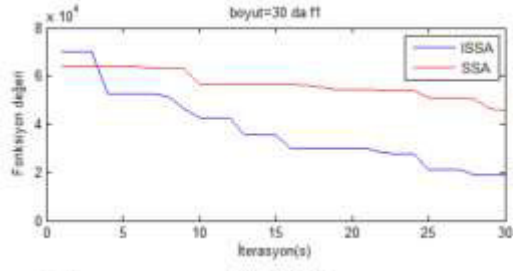
Çizelge 4.24. Karşılaştırma fonksiyonlarında SSA ve ISSA'nın performans karşılaştırması (Boyut= 100)

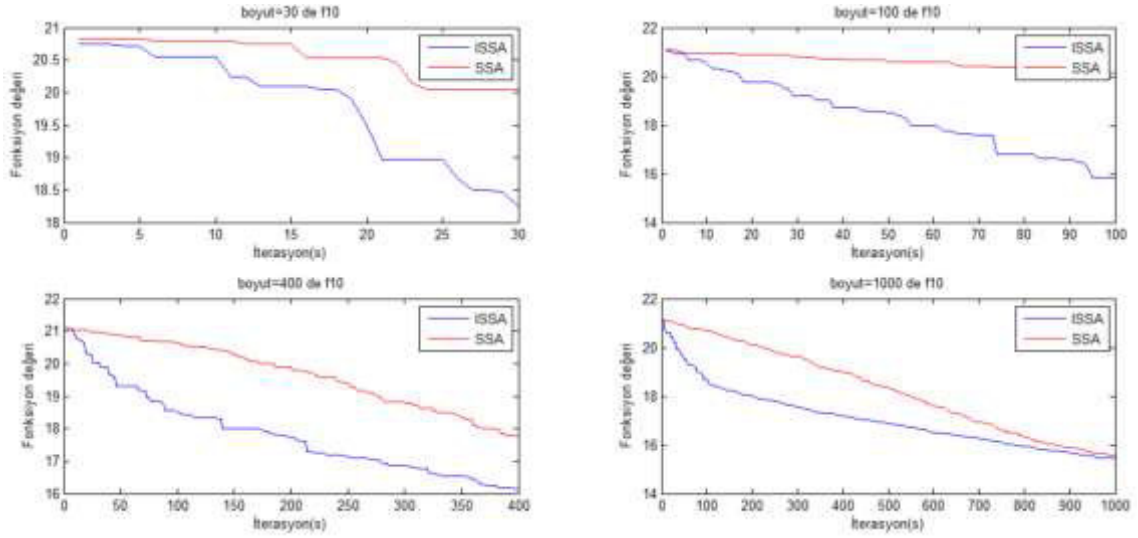
F	Boyut=100							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
$f1(x)$	5,67E+02	1,18E+02	5,27E+02	5,78E+02	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f2(x)$	8,51E+40	1,76E+41	8,00E+40	8,99E+40	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f3(x)$	2,38E+03	6,38E+02	2,18E+02	2,88E+04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f4(x)$	6,35E+01	3,35E+00	5,35E+01	8,75E+01	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f5(x)$	5,99E+04	1,85E+04	5,85E+04	5,99E+05	3,15E+04	1,13E+04	3,52E+03	1,15E+05
$f6(x)$	6,79E+01	5,66E+00	6,89E+00	2,25E+02	6,56E+01	2,70E+00	6,23E+00	6,86E+02
$f7(x)$	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f8(x)$	-4,82E+03	9,31E+02	-5,22E+03	-2,22E+03	-2,02E+04	4,82E+03	-3,02E+05	-2,12E+03
$f9(x)$	5,36E+02	3,00E+01	5,12E+02	5,96E+02	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f10(x)$	4,77E+00	2,45E-01	4,34E+00	4,99E+01	8,88E-16	0,00E+00	7,28E-19	6,88E-13
$f11(x)$	6,42E+00	8,80E-01	5,42E+00	7,22E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f12(x)$	2,32E+02	1,51E+01	2,22E+01	2,72E+03	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f13(x)$	-2,07E+01	9,87E-01	-2,85E+01	-2,85E+00	-2,22E+01	1,48E+00	-3,27E+02	-2,32E+00

Çizelge 4.25. Karşılaştırma fonksiyonlarında SSA ve ISSA'nın performans karşılaştırması (Boyut=1000)

F	Boyut=1000							
	SSA				ISSA			
	Ortalama	Std.Dev.	En iyi	En kötü	Ortalama	Std.Dev.	En iyi	En kötü
$f1(x)$	3,18E+05	1,20E+04	1,18E+03	3,88E+06	3,04E+05	2,60E+04	2,04E+03	3,44E+06
$f2(x)$	9,25E+60	2,25E+55	5,35E+40	7,25E+65	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f3(x)$	1,41E+03	1,93E+04	1,21E+02	1,41E+04	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f4(x)$	9,94E+01	1,33E-01	8,94E+00	6,94E+02	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f5(x)$	2,61E+08	2,09E+07	1,61E+06	2,91E+08	9,99E+02	1,45E-02	8,99E+02	2,19E+03
$f6(x)$	2,52E+03	6,26E+01	2,32E+02	2,82E+03	6,97E+02	1,39E+01	5,37E+00	6,37E+03
$f7(x)$	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f8(x)$	-1,48E+04	3,79E+03	-6,38E+04	-1,88E+02	-9,41E+04	3,42E+04	-6,31E+05	-8,11E+02
$f9(x)$	1,02E+04	1,46E+02	1,32E+02	1,22E+05	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f10(x)$	1,54E+01	1,67E-01	1,34E-01	1,04E+02	8,88E-16	0,00E+00	8,78E-19	8,35E-14
$f11(x)$	2,83E+03	1,35E+02	3,83E+02	2,03E+03	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f12(x)$	8,73E+02	1,65E+01	8,33E+01	9,73E+02	0,00E+00	0,00E+00	0,00E+00	0,00E+00
$f13(x)$	-1,41E+02	2,36E+00	-1,49E+03	-1,31E+00	-1,57E+02	2,18E+00	-1,77E+03	-9,56E+00

Şekil 4.4, tek modlu $f1$ ve $f4$ ve çok modlu $f9$ ve $f10$ karşılaştırma fonksiyonları için ISSA ve SSA yakınsama grafiklerini göstermektedir. Yakınsama grafikleri sonuçlarına göre, ISSA ile en uygun çözümler elde edilmiştir.





Şekil 4.4. SSA ve ISSA için yakınsama grafikleri (f1, f4, f9 ve f10)

Farklı boyutlarda çalıştırılan SSA ve ISSA sonuçları arasında anlamlı bir fark olup olmadığına bakmak için yine Wilcoxon signed-rank testi kullanılmıştır. Test sonuçları çizelge 4.26’da gösterilmiştir. Boyut 30’da ISSA 13 fonksiyonun 8’inde kayda değer bir farkla sonuçlara ulaşmıştır. Boyut 100’de ISSA 13 fonksiyonun 6’sında kayda değer bir farkla sonuçlara ulaşmıştır. Boyut 1000’de ISSA 13 fonksiyonun 5’inde kayda değer bir farkla sonuçlara ulaşmıştır.

Çizelge 4.26. SSA ve ISSA sonuçları için Wilcoxon signed-rank testi sonuçları

<i>F</i>	Boyut=30		Boyut=100		Boyut=1000	
	ISSA(p)	h	ISSA(p)	h	ISSA(p)	h
<i>f1(x)</i>	0,0419	1	0,9698	0	0,9097	0
<i>f2(x)</i>	0,0211	1	0,0472	1	0,0025	1
<i>f3(x)</i>	1	0	0,0444	1	0,0147	1
<i>f4(x)</i>	0,5708	0	0,0757	0	0,2730	0
<i>f5(x)</i>	0,0123	1	0,6756	0	0,9097	0
<i>f6(x)</i>	0,9475	0	1	0	0,8172	0
<i>f7(x)</i>	NaN	0	NaN	0	NaN	0
<i>f8(x)</i>	0,0247	1	0,0474	1	0,0408	1
<i>f9(x)</i>	0,9698	0	0,7337	0	0,0103	1
<i>f10(x)</i>	0,0452	1	0,7337	0	0,0474	1
<i>f11(x)</i>	0,0123	1	0,0274	1	0,7337	0
<i>f12(x)</i>	0,0447	1	0,0257	1	0,0001	1
<i>f13(x)</i>	0,0347	1	0,0457	1	1	0

Moleküler Potansiyel Enerji Fonksiyonunu (MPEF) en aza indirmek için ISSA'nın genel performansı

Bu tezde SSA ve ISSA'nın performansı MPEF üzerinde çeşitli boyutlarda { 20, 40, 60, 80, 100, 120, 140, 160, 180, 200} test edilmiştir. Sonuçlar çizelge 4.27'de gösterilmiştir. Sonuçlara göre MPEF'de ISSA, SSA'dan daha iyi bir performans göstermiştir.

Çizelge 4.27. ISSA ve SSA'nın MPEF için karşılaştırma sonuçları (Boyut = {20, 40, 60, 80, 100, 120, 140, 160, 180, 200})

Boyut	SSA	ISSA
20	15,9100	13,5926
40	33,1715	32,5559
60	57,0631	48,4384
80	76,7605	66,3655
100	98,2967	78,4906
120	110,91	105,76
140	130,45	124,16
160	160,79	148,94
180	170,53	156,09
200	187,36	186,60

ISSA iki farklı yöntem serisinde karşılaştırılmıştır. İlk seride, GA'nın varyantları olan WX-PM, WX-LLM, LX-LLM ve LX-PM moleküler potansiyel enerji fonksiyonunda ISSA ile eşit karşılaştırma parametrelerinde karşılaştırılmıştır (Deep ve Thakur, 2007b; 2007a; Deep ve ark., 2011) İkinci seride, değişken komşuluk arama tabanlı bir yöntem olan VNS-123 ile ISSA, moleküler potansiyel enerji fonksiyonunda eşit karşılaştırma parametrelerinde karşılaştırılmıştır (Dražić ve ark., 2008). rHYB yöntemi hibrit bir genetik algoritmayı ifade eder (Barbosa ve ark., 2005). Tüm karşılaştırma algoritmalarında kullanılan parametre ayarları çizelge 4.28'de gösterilmiştir.

Çizelge 4.28. Parametre ayarları

Anahtar	Metot ismi
VNS-123	VNS-1, VNS-2 ve VNS-3'ün kombinasyonu
WX-PM	Çaprazlama operatörü=WX ve mutasyon operatörü=PM
WX-LLM	Çaprazlama operatörü =WX ve mutasyon operatörü =LLM
LX-LLM	Çaprazlama operatörü =LX ve mutasyon operatörü=LLM
LX-PM	Çaprazlama operatörü =LX ve mutasyon operatörü =PM
rHYB	
Ortak ayarlar	Popülasyon miktarı (N): = 25; çalışma sayısı = 20; maksimum iterasyon = [400-4000]; boyut=[20-200]

Denklem 3.8’de gösterilen E fonksiyonu belirtilen arama uzayında $[0, 5]^n$ en aza indirilir (n problem boyutunu göstermektedir.). E fonksiyonu çizelge 4.29’da gösterildiği gibi $E^*(n) = -0.0411183n$ (Lavor ve Maculan, 2004) olarak boyut ile doğrusal olarak büyür.

Çizelge 4.29. Farklı boyutlarda E^* ‘ın küresel minimum değerleri

n	E^*
20	-0,822366
40	-1,644732
60	-2,467098
80	-3,289464
100	-4,111830
120	-4,934196
140	-5,756562
160	-6,578928
180	-7,401294
200	-8,22366

ISSA’nın performansı MPEF probleminde çeşitli boyutlarda {20, 40, 60, 80, 100, 120, 140, 160, 180, 200}; SSA, VNS-123, WX-PM, WX-LLM, LX-LLM, LX-PM ve rHYB ile karşılaştırılmıştır. Karşılaştırma algoritmalarının sonuçları doğrudan çeşitli kaynaklardan alınmıştır (Barbosa ve ark., 2005; Deep ve Thakur, 2007b; 2007a; Dražić ve ark., 2008; Deep ve ark., 2011) Sonuçlar çizelgeler 4.30 ve 4.31’de gösterilmektedir. Başarılı sonuçlar kalın font ile işaretlenmiştir.

Çizelge 4.30. ISSA’nın MPEF için karşılaştırma sonuçları (boyut = {20, 40, 60, 80, 100})

Boyut	WX-PM	LX-PM	WX-LLM	LX-LLM	ISSA
20	15,574	23,257	28,969	14,586	13,5926
40	59,999	71,336	89,478	39,366	32,5559
60	175,865	280,131	225,008	105,892	48,4384
80	302,011	326,287	372,836	237,621	66,3655
100	369,376	379,998	443,786	320,146	78,4906

Çizelge 4.31. ISSA’nın MPEF için karşılaştırma sonuçları (boyut={20, 40, 60, 80, 100, 120, 140, 160, 180, 200})

Boyut	VNS-123	GA	rHYB	ISSA
20	23,381	36,626	35,836	13,5926
40	57,681	133,581	129,611	32,5559
60	142,882	263,266	249,963	48,4384
80	180,999	413,948	387,787	66,3655
100	254,899	-	-	78,4906
120	375,970	-	-	105,76
140	460,519	-	-	124,16
160	652,916	-	-	148,94
180	663,722	-	-	156,09
200	792,537	-	-	186,60

4.2. Ayırık Optimizasyon Görevleri için DSSA Test Sonuçları

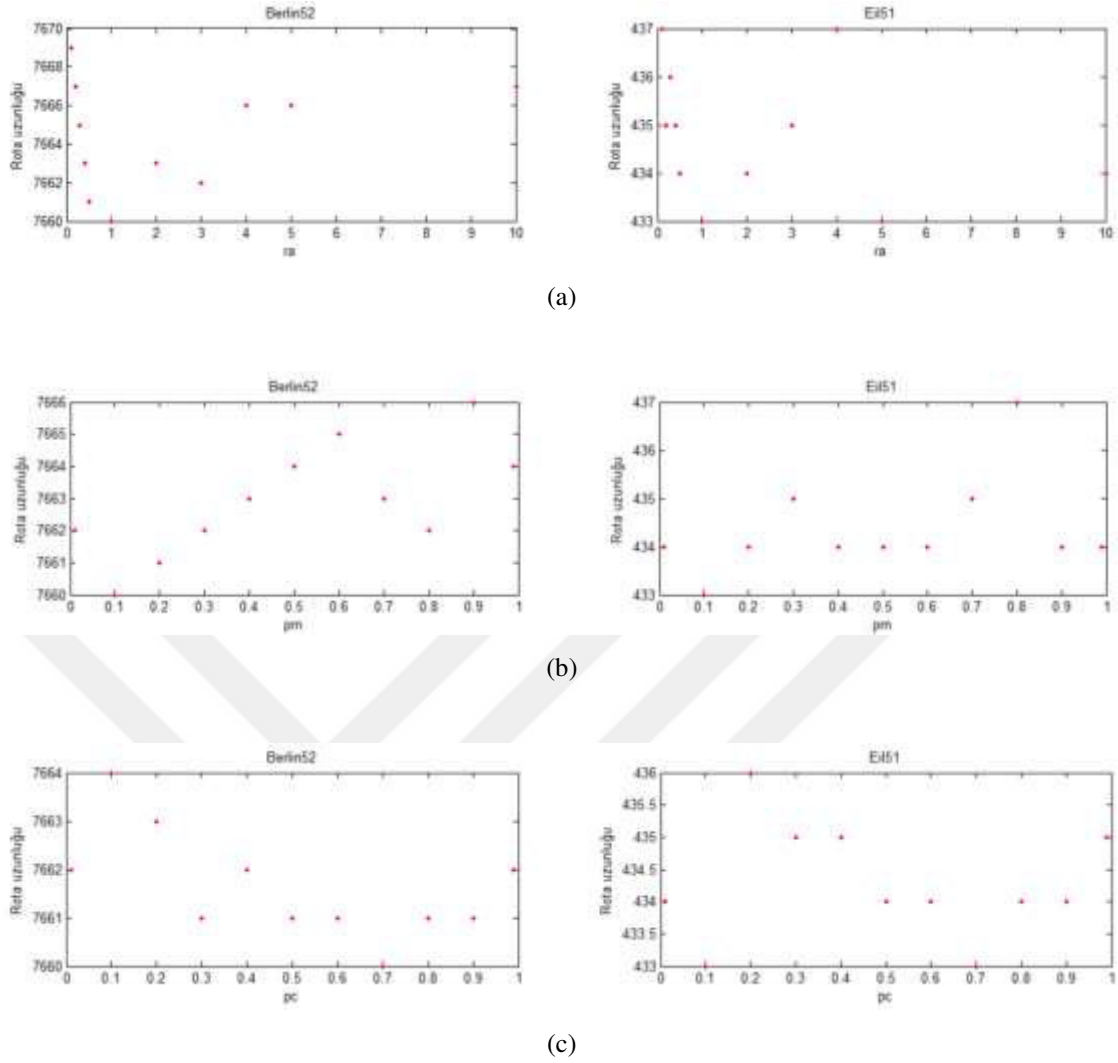
SSA, ayırık optimizasyon problemlerini çözebilmesi için ayırık bir şekle dönüştürülmüştür. Geliştirilen ayırık SSA'ya Ayırık SSA (DSSA) ismi verilmiştir. DSSA, literatürde sıklıkla test için kullanılan Gezgin Satıcı Problemi (TSP) üzerinde performansı test edilmiştir. DSSA'nın ayırık arama uzayında keşif ve sömürü yetenekleri *yetenekli örümcek* ve *acemi örümcek* teknikleri ile geliştirilmiştir.

TSPLIB web sitesinden alınan düşük ölçekli, orta ölçekli ve büyük ölçekli otuz sekiz TSP karşılaştırma veri seti, DSSA performansını ölçmek için kullanılmıştır. Bu TSP kümesi çeşitli sayıda özelliğe (14 ile 1002) sahiptir. Bu çalışmada, algoritmanın performansını göstermek için büyük boyutlu veri setleri tercih edilmiştir. TSP kıyaslama veri kümeleri özellik sayısına göre üç gruba ayrılır.

- *Grup-I:* Düşük ölçekli boyut (<50)
- *Grup-II:* Orta ölçekli boyut ([50,100])
- *Grup-III:* Geniş ölçekli boyut (>100)

DSSA, literatürdeki yeni ve bilinen sezgisel algoritmalar ile karşılaştırılmıştır. Adil karşılaştırmalar yapmak için, tüm algoritmalar aynı programlama dilinde, benzer platformda ve benzer parametre değerlerinde çalıştırılmıştır. DSSA'nın performansı analiz edilmiştir ve tartışılmıştır.

DSSA'da kullanılacak ideal r_a , p_c ve p_m parametre değerlerini bulmak için rasgele seçilmiş iki farklı TSP veri seti üzerinde (Berlin52 ve Eil51) DSSA çalıştırılmıştır. r_a değeri {1/10, 1/5, 1/4, 1/3, 1/2, 1, 2, 3, 4, 5, 10} kümesinden seçilmiştir. p_c ve p_m değerlerinin her ikisi de {0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99} kümesinden seçilmiştir. Her parametre analizi için on bir veri noktası vardır. Durdurma yineleme değeri 100000 şeklinde ayarlanmış ve her parametre on kez bağımsız test edilmiştir. Sonuçlar bu üç parametre değerleri kombinasyonel değiştirilerek gözlenmiştir. Berlin52 ve Eil51 TSP veri kümesi örneklerinin sonuçları şekil 4.5'de gösterilmiştir. Elde edilen sonuçlara göre, r_a , p_c ve p_m için ortalama değerler diğer TSP veri kümesi örneklerine de uygulanmıştır.



Şekil 4.5. Parametre test sonuçları (a) r_a , (b) p_m ve (c) p_c

Farklı popülasyon büyüklüğü (N) kombinasyonları Berlin52, Gr17, Eil76 ve Eil51 veri kümelerinde test edilmiştir. Çizelge 4.32’de testlerin sonuçları gösterilmiştir. Her örnek on kez çalıştırılmış ve sonuçların ortalaması alınmıştır. DSSA’daki şehir sayısı otuzdan azsa, popülasyon büyüklüğü otuz, şehir sayısı otuz ile bin arasında ise, popülasyon büyüklüğü şehir sayısı ile aynı sayıda alınır. Şehirlerin sayısı binden fazla ise popülasyon büyüklüğü bin olarak alınmıştır. SSA ve DSSA parametreleri çizelge 4.33’de gösterilmiştir.

Çizelge 4.32. Performans üzerinde örümcek sayısının etkisi (ortalama(Mean): ortalama rota uzunluğu)

Veri seti	N =10 Ortalama	N=20 Ortalama	N=30 Ortalama	N=şehir sayısı Ortalama
Berlin52	7844	7736	7718	7659
Gr17	7348	2085	2085	2085
Eil76	571,2	571	559,52	559,31
Eil51	444,5	437,1	435,4	431,8
Bays29	9077	6074	2120	2120

Çizelge 4.33. SSA ve DSSA için belirlenen parametre ayarları

Parametreler	Değerler
r_a	1
p_c	0,7
p_m	0,1
Örümcek sayısı	Otuz, şehir sayısı veya bin
Boyut	Şehir sayısı
Maksimum yenileme	100000
Uygunluk (Fitness) fonksiyonu	Şehir turu rotasının uzunluğu

DSSA'nın seçilmiş tüm TSP örnekleri üzerindeki test sonuçları çizelge 4.34'de gösterilmiştir. Her uygulama on bağımsız çalışmaya sahiptir. Bu çalışmada kullanılan ölçüm kriterleri: (1) Ortalama (Mean), (2) Standart sapma (Std.Dev.), (3) Minimum değer (En İyi) ve (4) Maksimum değer (En Kötü)'dür. TSP örneklerinin optimum sonuçları TSPLIB kütüphanesinden alınmıştır.

Sonuçlara göre, DSSA, düşük ölçekli TSP veri kümesi örneklerinde (boyut <50) üstün performans göstermiştir. DSSA, 6 TSP örneğinden (Burma14, Ulysses16, Gr17, Ulysses22, Fri26 ve Bays29) 6 TSP karşılaştırması için % 0 hata oranı ile yüksek başarı göstermiştir. Çizelge 4.34'e göre, DSSA orta ölçekli TSP veri seti örneklerinde iyi bir başarıya sahiptir (boyut = [50-100]). DSSA, 11 TSP örneğinden (Eil76 ve Rd100 hariç Eil51, Berlin52, KroA100, KroB100, KroC100, KroD100, KroE100, Pr76 ve Rat99) 9 TSP karşılaştırması için % 2 hata oranının altında başarı göstermiştir. DSSA'nın büyük ölçekli TSP veri kümesi örneklerinde (boyut > 100) dikkate değer bir başarısı vardır. DSSA, 21 TSP örneğinden (KroA150, KroA200, Pr107, Pr124, Pr144, Pr152, Rat195, Lin105, Lin318, Ts225 ve Tsp225) 11 TSP karşılaştırması için % 2 hata oranının altında başarı göstermiştir. DSSA, 21 TSP örneğinden (Pr226, Pr264, Pr299, pr439, Pr1002, Rd400 ve Linhp318) 7 TSP karşılaştırması için % 4 hata oranının altında başarı göstermiştir. DSSA, 21 TSP örneğinden (Rat575 ve Rat783) 2 TSP karşılaştırması için % 6 hata oranının altında başarı göstermiştir.

Sonuçlar, yetenekli örümcek ve acemi örümcek tekniklerinin DSSA'nın hem küresel hem de yerel olarak arama yeteneğini geliştirdiğini göstermiştir. Yetenekli örümcek ve acemi örümcek süreçleri tarafından üretilen yeni aday çözüm seti istikrarlı bir çözüm sağlamıştır.

Çizelge 4.34. DSSA'nın TSP örnekleri üzerinde test sonuçları (Ortalama: ortalama rota uzunluğu; Std.Dev.: Standart sapma; Hata (%); ilişkisel hata yüzdesi)

No	TSP örnek	TSPLIB'de optimum	Durum	Ortalama	En iyi	En kötü	Hata (%)	Std.Dev.
1	Burma14	3323	=	3323	3323	7702	0	0
2	Ulysses16	6859	=	6859	6859	14230	0	0
3	Gr17	2085	=	2085	2085	5252	0	0
4	Ulysses22	7013	=	7013	7013	18325	0	0
5	Fri26	937	=	937	937	2892	0	0
6	Bays29	2020	=	2020	2020	6709	0	0
7	Eil51	426	<	431,87	431,87	183530	0,65	2,8
8	Berlin52	7542	<	7659	7659	31432	1,55	117
9	Eil76	538	<	559,31	559,31	2720,4	3,96	0,4
10	Pr76	108159	<	108880	108880	621210	0,66	721
11	Rat99	1211	<	1221	1221	9237	0,82	10
12	Rd100	7910	<	8120	8120	59865	2,65	210
13	KroA100	21282	<	21363	21363	189380	0,38	81
14	KroB100	22141	<	22347	22347	188300	0,93	206
15	KroC100	20749	<	20997	20997	191830	1,19	248
16	KroD100	21294	<	21552	21552	147160	1,21	258
17	KroE100	22068	<	22407	22407	198040	1,53	339
18	Lin105	14379	<	14502	14502	135670	0,85	123
19	Pr107	44303	<	44346	44346	68720	0,09	43
20	Pr124	59030	<	59087	59087	76722	0,09	57
21	Pr136	96772	<	103460	103460	914060	6,91	6688
22	Pr144	58537	<	58669	58669	87320	0,22	132
23	KroA150	26524	<	27027	27027	359350	1,59	503
24	Pr152	73682	<	74462	74462	112480	1,05	780
25	Rat195	2323	<	2353	2353	24826	1,29	30
26	KroA200	29368	<	29666	29666	373590	1,01	298
27	Ts225	126643	<	127230	127230	1578500	0,46	587
28	Tsp225	3919	<	3933	3933	43032	0,35	14
29	Pr226	80369	<	82186	82186	180840	2,26	1817
30	Pr264	49135	<	50739	50739	61355	3,26	1604
31	Pr299	48191	<	49559	49559	814590	2,83	1368
32	Lin318	42029	<	42686	42686	58863	1,56	657
33	Linhp318	41345	<	42686	42686	62734	3,24	1341
34	Rd400	15281	<	15678	15678	22459	2,59	397
35	Pr439	107217	<	111450	111450	2021400	3,94	4233
36	Rat575	6773	<	7054	7054	86903	4,14	281
37	Rat783	8806	<	9193	9193	12096	4,39	387
38	Pr1002	259045	<	266440	266440	312240	2,85	7395

Önerilen yöntemin performansını değerlendirmek için, literatürden seçilmiş bazı yeni algoritmalar da DSSA ile karşılaştırılmıştır. Bu algoritmalar: Ayırık Değişken Komşuluk Arama algoritması (DVNS) (Hore ve ark., 2018), Ayırık Su Döngüsü Algoritması (DWCA) (Osaba ve ark., 2018), Genetik Algoritma (GA) ve Ada modeli Ayırık Genetik Algoritma (IDGA) (Osaba ve ark., 2018), Ayırık Yarasa Algoritması1 ve Ayırık Yarasa Algoritması2 (BA1 ve BA2) (Osaba ve ark., 2018), Ayırık Ateş Böceği Algoritması (DFA) (Osaba ve ark., 2018), Ayırık Emperyalist Rekabet Algoritması (DICA) (Atashpaz-Gargari ve Lucas, 2007), Ayırık Arama Algoritması (ESA) (Yip ve

Pao, 1995), ABC ile Ayırık ACO (Gündüz ve ark., 2015), Ayırık PSO – ACO – 3Opt hibrit yöntemi (Mahi ve ark., 2015), Ayırık IBA (Osaba ve ark., 2016), Ayırık Dinamik Çok Ölçekli Bölge Arama Algoritması (DMRSA) (Zhang ve Zhou, 2016), Ayırık Yapay Arı Kolonisi (DABC) (Li ve ark., 2011), Ayırık Parçacık Sürüsü Optimizasyonu (DPSO) (Li ve ark., 2011), Evrimsel Ayırık Ateşböceği Algoritması (EDFA) (Jati, 2011), Ayırık Guguk kuşu Arama Algoritması (Temel DCS) (Ouaarab ve ark., 2014) ve Geliştirilmiş Ayırık Guguk kuşu Arama Algoritması (Geliştirilmiş DCS) (Ouaarab ve ark., 2014)’dır. Adil karşılaştırmalar yapmak için, tüm algoritmalar aynı programlama dilinde, benzer platformda ve benzer parametre değerlerinde çalıştırılmıştır. Karşılaştırma sonuçları sırasıyla çizelge 4.35 ve 4.36’de gösterilmiştir.

Çizelge 4.35. DSSA ile literatürdeki algoritmaların TSP örnekleri üzerinde karşılaştırılması

Metot	Kriter	Bays29	Fri26	Gr17	Eil51	Berlin52	Eil76	KroA100	KroB100	KroC100
DVNS	Ortalama	-	-	-	-	-	-	-	-	-
	Std.Dev.	-	-	-	-	-	-	-	-	-
DWCA	Ortalama	-	-	-	-	-	-	-	-	-
	Std.Dev.	-	-	-	2,00	-	3,30	47,90	164,40	124,60
GA	Ortalama	-	-	-	440,80	7542	565,40	21812,40	22687,40	21510,40
	Std.Dev.	-	-	-	7,30	0,00	9,80	420,80	407,70	390,20
IDGA	Ortalama	-	-	-	434,4	7542	557,70	21731,80	22712,60	21298,70
	Std.Dev.	-	-	-	4,50	0,00	6,80	340,70	312,80	290,70
Ayırık ESA	Ortalama	-	-	-	431,60	7542	553,70	21481,70	22602,20	21170,40
	Std.Dev.	-	-	-	2,90	0,00	4,20	150,10	210,20	188,70
Ayırık BA1	Ortalama	-	-	-	438,30	7676	558,80	21884,20	22842,90	21476,60
	Std.Dev.	-	-	-	2,50	104,40	9,00	213,60	231,20	235,10
Ayırık BA2	Ortalama	-	-	-	436,80	7681,90	560,50	21989,40	22946,70	21631,10
	Std.Dev.	-	-	-	5,30	112,30	11,60	305,20	291,30	325,00
Ayırık IBA	Ortalama	-	-	-	428,10	7542,00	548,10	21445,30	22506,4	21050,0
	Std.Dev.	-	-	-	1,60	0,00	3,80	116,50	221,3	164,70
DFA	Ortalama	-	-	-	430,80	7542	556,80	21483,60	22604,80	21096,30
	Std.Dev.	-	-	-	2,30	0,00	4,90	163,70	243,90	148,30
DICA	Ortalama	-	-	-	432,30	7542	557,60	21500,30	22599,70	21103,90
	Std.Dev.	-	-	-	3,10	0,00	5,80	183,40	244,90	161,10
Ayırık ACO with ABC	Ortalama	-	-	-	443,39	7544,37	557,98	22435,31	-	-
	Std.Dev.	-	-	-	5,25	0,00	4,10	231,34	-	-
Ayırık PSO-ACO- 3Opt	Ortalama	-	-	-	426,45	7543,20	538,30	21445,10	-	-
	Std.Dev.	-	-	-	0,61	2,37	0,47	78,24	-	-
DABC	Ortalama	16751	-	-	-	21338	-	129960	-	-
	Std.Dev.	-	-	-	-	-	-	-	-	-
DPSO	Ortalama	17953	-	-	-	21827	-	132400	-	-
	Std.Dev.	-	-	-	-	-	-	-	-	-
Ayırık DMRSA	Ortalama	-	-	-	426	7542	540,36	21282	-	-
	Std.Dev.	-	-	-	0,19	0,00	1,16	0,00	-	-
Temel DCS	Ortalama	-	-	-	439	7836	565,70	22419,96	23417,06	-
	Std.Dev.	-	-	-	-	-	-	-	-	-
Geliştirilmiş DCS	Ortalama	-	-	-	426	7542	538,03	21282	22141,53	-
	Std.Dev.	-	-	-	0,00	0,00	0,00	0,00	0,00	-
DSSA	Ortalama	2020	937	2085	431,87	7659	559,31	21363	22347	20997
	Std.Dev.	0,00	0,00	0,00	2,80	117	0,40	81	206	248

Çizelge 4.36. DSSA ile literatürdeki algoritmaların TSP örnekleri üzerinde karşılaştırılması (devam)

Metot	Kriter	Ulysses16	Ulysses22	Tsp225	Pr76	KroA150	KroA200	Lin318
EDFA	Ortalama	100,119	100,207	88,332	-	-	-	-
	Std.Dev.	-	-	-	-	-	-	-
Temel DCS	Ortalama	-	-	-	111225,6	28928,83	32896,03	49623,96
	Std.Dev.	-	-	-	-	-	-	-
Geliştirilmiş DCS	Ortalama	-	-	-	108159	26569,26	29446,66	42434,73
	Std.Dev.	-	-	-	0,00	-	-	-
DSSA	Ortalama	6859	7013	3933	108880	27027	29666	42686
	Std.Dev.	0,00	0,00	14	721	503	298	1341

Çizelgeler 4.35 ve 4.36 sonuçlarına göre, DSSA düşük ve orta ölçekli TSP veri setlerinde oldukça iyi bir performansa sahiptir. Ancak, büyük ölçekli TSP veri kümesi için iyi performans göstermez. Düşük ve orta ölçekli TSP kriterleri için kabul edilebilir sonuçlar elde etmiştir. Bu başarının nedeni, yetenekli örümcek ve acemi örümcek teknikleri ile aday çözümler arasında daha dengeli bir çeşitlilikten kaynaklanmaktadır. Sonuç olarak DSSA ayrık görevler için kullanılabilmektedir.

4.3. İkili Optimizasyon Görevleri için BinSSA Test Sonuçları

İkili optimizasyon, ayrık optimizasyon tekniklerinin özel bir şeklidir. Sürekli gerçek arama alanlarına sahip algoritmalar, değişkenleri ikili değişkenlere dönüştürerek İkili Optimizasyon Problemlerini (BOP) çözebilir. Sürekli optimizasyon problemleri için geliştirilmiş SSA, ikili optimizasyon problemlerini de çözebilecek şekilde tekrardan yapılandırılmıştır. Böylece bu tezde BinSSA önerilmiştir. İkili dönüşümün ana parçası aktarım (transfer) fonksiyonlarıdır. BinSSA'da, çözüm havuzu $\{0,1\}$ ikili dize değerleri ile temsil edilir. Bu çevirme işlemi, transfer fonksiyonuyla ilgili belirli bir eşik değeri ile gerçekleştirilir. Transfer fonksiyonları, pozisyon vektörünün elemanlarını 0'dan 1'e değiştirme veya bunun tam tersi olasılığını tanımlar. Literatürde S şeklinde, V şeklinde ve mod tabanlı birçok transfer fonksiyonu bulunmaktadır.

Bu bölümde BinSSA ile literatürde BOP olarak yer alan dört farklı problem türü çözülmüş ve BinSSA'nın performansı test edilmiştir. Bunlar:

- Sürekli optimizasyon görevleri için BinSSA
- Kapasitesiz Tesis Yerleşim Problemi (UFLP) için BinSSA
- Rüzgâr türbini yerleşim problemi için BinSSA
- Özellik Seçme (FS) problemi için BinSSA

4.3.1. Sürekli optimizasyon görevleri için BinSSA

Bu bölümde BinSSA’da sürekli arama uzayı ikili uzaya çevrilirken, iki S-şekilli (Sigmoidal transfer fonksiyonu ve Modifiye sigmoidal transfer fonksiyonu) ve iki V-şekilli (Tanjant hiperbolik transfer fonksiyonu ve Arktanjan transfer fonksiyonu) transfer fonksiyonları kullanılmıştır. Böylece, BinSSA-Tanh, BinSSA-Sigm, BinSSA-MSigm ve BinSSA-Arctan olarak dört farklı BinSSA varyasyonu oluşturulmuştur. BinSSA’da ikili arama uzayında keşif ve sömürü arasındaki dengeyi sağlamak için aday çözüm üretim aşamasında iki farklı teknik (SimSSA ve LogicSSA) geliştirilmiştir. Bunlar Jaccard’ın benzerlik ölçümü (SimSSA) ve “xor” mantıksal kapı (LogicSSA) teknikleridir. LogicSSA yerel aramada çok güçlü iken SimSSA çözüm alanında yeni çözümler keşfetmede çok iyidir. Deneysel çalışmalarda, çizelge 4.37’de gösterilen on sekiz farklı tek modlu ve çok modlu karşılaştırma fonksiyonu ayrı ayrı çözülmüştür.

Çizelge 4.37. Deneysel çalışmada kullanılan kıyaslama (benchmark) fonksiyonları

Fonksiyon	Aralık	f_{min}
$f_1(X)=\sum_{i=1}^n x_i^2$	$[-100,100]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_2(X)=\sum_{i=1}^n ix_i^4$	$[-2.56,2.56]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_3(X)=\sum_{i=1}^n ([x_i + 0.5])^2$	$[-10,10]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_4(x) = \sum_{i=1}^n ix_i^4 + random[0,1)$	$[-2.56,2.56]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_5(x) = \max_i \{ x_i , \quad 1 \leq i \leq n\}$	$[-100,100]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_6(X)=\sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-100,100]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_7(X)=\sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_8(X)=\sum_{i=1}^n ix_i^2$	$[-10,10]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_9(X)=(x_1 - 1)^2 + \sum_{i=1}^n i(2x_i^2 - x_{i-1})^2$	$[-10,10]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_{10}(X)=\sum_{i=1}^n x_i ^{i+1}$	$[-1,1]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_{11}(X)=\sum_{i=1}^n \sum_{j=1}^i x_j^2$	$[-65,65]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_{12}(x) = 1 - \cos\left(2\pi \sqrt{\sum_{i=1}^n x_i^2}\right) + 0.1 \sqrt{\sum_{i=1}^n x_i^2}$	$[-100,100]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_{13}(X)=\sum_{i=1}^{10} (x_i - 1)^2 - \sum_{i=2}^{10} x_i x_{i-1} - 1$	$[-n^2, n^2]^n$	$i=1,2,\dots,n); \quad x_i=(n+1-i)$ $f(X^*)=-n(n+4)(n-1)/6$
$f_{14}(X)=0.1\{\sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^n u(x_i, 5, 100, 4); u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-10,10]^n$	$X^*=(1,\dots,1); f(X^*)=0$
$f_{15}(X)=\sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-50,50]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_{16}(X)=\frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_{17}(X)=-20\exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + \exp$	$[-32,32]^n$	$X^*=(0,\dots,0); f(X^*)=0$
$f_{18}(X)=\sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500,500]^n$	$X^*=(420,\dots,420);$ $f(X^*)=-418,9829 \times n$

Her BinSSA varyasyonu, her çalışma için aynı maksimum yinleme sayısı ile yirmi bağımsız çalışma olarak test edilmiştir. Adil karşılaştırmalar yapabilmek için, tüm BinSSA varyasyonları aynı programlama dilinde, benzer platformda ve benzer parametre değerlerinde dikkatle çalıştırılmıştır. BinSSA varyasyonlarından elde edilen uygunluk değerleri için Wilcoxon signed-rank testi uygulanmıştır.

Benchmark fonksiyonları tiplerine göre iki gruba ayrılmaktadır. Bu kıyaslama fonksiyonları çeşitli kaynaklardan elde edilmiştir (Jamil ve Yang, 2013; James ve Li, 2015a; Surjanovic ve Bingham, 2017; Rizk-Allah ve ark., 2019)

- *Grup I:* tek modlu kıyaslama fonksiyonları (*f1-f11*)
- *Grup II:* çok modlu kıyaslama fonksiyonları (*f12-f18*)

BinSSA'nın performansını değerlendirmek için dört ölçüm kriteri kullanılmıştır. Bu kriterler çizelge 4.38'de gösterilmiştir. Bu ölçüm kriterleri: (1) Ortalama (Mean), (2) Standart sapma (Std.Dev.), (3) Minimum değer (En İyi) ve (4) Maksimum değer (En Kötü)'dür. n_{run} toplam bağımsız çalışma sayısını temsil etmektedir.

Çizelge 4.38. Performans ölçüm kriterleri

Ortalama:	$Ortalama = \frac{1}{n_{run}} \times \sum_{i=1}^{n_{run}} Fitness_func_i$
Standart Sapma:	$Std.Dev. = \sqrt{\frac{1}{n_{run}} \sum_{i=1}^{n_{run}} (gbest_i - Ortalama)^2}$
En iyi:	$Fitness_func_Best = \min_{i=1 \dots n_{run}} Fitness_func_Best_i$
En kötü:	$Fitness_func_Worst = \max_{i=1 \dots n_{run}} Fitness_func_Worst_i$

Uygunluk fonksiyonuna göre popülasyon miktarını tespit etme:

Popülasyon büyüklüğünün farklı kombinasyonları, rasgele seçilmiş tek modlu ve çok modlu karşılaştırma fonksiyonları üzerinde test edilmiştir (*f1, f5, f15 ve f17*). Sonuçlar çizelge 4.39'da gösterilmiştir. Sonuçlara göre, popülasyon büyüklüğü için en uygun değer 30 olarak belirlenmiştir.

Çizelge 4.39. Uygunluk değerlerine göre popülasyon büyüklüğünün farklı kombinasyonları

BinSSA varyasyonları	Popülasyon miktarı	<i>f1</i>	<i>f5</i>	<i>f15</i>	<i>f17</i>
BinSSA- Tanh	10	0,0000	2,0000	34,0000	25,0000
	20	0,0000	0,0000	32,0000	22,0000
	30	0,0000	0,0000	0,0000	20,0000
BinSSA- Sigm	10	12,5000	11,500	20,0000	21,0000
	20	-53,0000	11,4000	6,0000	21,0000
	30	-23,0000	9,0000	14,0000	20,0000
BinSSA- Msigm	10	80,0000	9,0000	50,0000	20,0000
	20	1,7000	7,5000	32,0000	20,0000
	30	0,0000	2,6000	21,0000	20,0000
BinSSA-Arctan	10	0,0000	10,5000	45,0000	1,2E-04
	20	0,0000	0,0000	0,0000	1,2E-05
	30	0,0000	0,0000	0,0000	1,2E-06

BinSSA varyasyonlarının karşılaştırmalarında kullanılan popülasyon boyutu (N), maksimum yineleme değeri ve diğer parametreler çizelge 4.40’da gösterilmektedir.

Çizelge 4.40. BinSSA için ortak parametre ayarları

Parametreler	Değerler
Popülasyon miktarı (N)	30
Maksimum yineleme	50
Boyut (n)	10
Bağımsız çalışma sayısı	20
Diğer parametreler	$r_a=1$, $p_c=0,7$; $p_m=0,1$; $ST=0,3$ (James ve Li, 2015a; Emary ve ark., 2016)

BinSSA varyasyonlarının karşılaştırılması:

BinSSA'nın SimSSA ve LogicSSA temelli varyasyonları on sekiz farklı tek modlu ve çok modlu karşılaştırma fonksiyonunda test edilmiştir. Her kıyaslama fonksiyonu 20 kez çalıştırılmıştır. Her kıyaslama fonksiyonu için en iyi, en kötü, ortalama ve standart sapma (Std.Dev.) sonuçları elde edilmiştir. BinSSA için popülasyon miktarı 30 olarak ve maksimum yenileme miktarı 50 olarak seçilmiştir. Tek modlu karşılaştırma fonksiyonları (*f1-f11*) için elde edilen en iyi, en kötü, ortalama ve standart sapma (Std.Dev.) karşılaştırma sonuçları çizelge 4.41’de gösterilmiştir. Çok modlu karşılaştırma fonksiyonları (*f12-f18*) için elde edilen en iyi, en kötü, ortalama ve standart sapma (Std.Dev.) karşılaştırma sonuçları çizelge 4.42’de gösterilmiştir. En iyi sonuçlar kalın font ile işaretlenmiştir.

Çizelge 4.41'deki standart sapma sonuçlarına göre, BinSSA-Arctan tüm unimodal kıyaslama fonksiyonlarında BinSSA-Tanh, BinSSA-Sigm ve BinSSA-MSigm'den daha iyi performans göstermiştir. BinSSA-Arctan varyasyonu, 11 tek modlu kıyaslama fonksiyonundan 9'u ($f1$, $f2$, $f3$, $f4$, $f5$, $f6$, $f7$, $f9$ ve $f10$) için fonksiyonların % 81,81'inde başarı göstermiştir. Ortalama sonuçlara göre, BinSSA-Arctan tüm tek modlu kıyaslama fonksiyonlarında BinSSA-Tanh, BinSSA-Sigm ve BinSSA-MSigm'den daha iyi performans göstermiştir. BinSSA-Arctan varyasyonu, 11 tek modlu kıyaslama fonksiyonundan 7'si için ($f1$, $f3$, $f5$, $f6$, $f7$, $f9$ ve $f10$) fonksiyonların % 63,63'ünde başarı göstermiştir. En iyi sonuçlara göre, BinSSA-Arctan tüm tek modlu kıyaslama fonksiyonlarında yüksek performans göstermiştir. BinSSA-Arctan varyasyonu, 11 tek modlu kıyaslama fonksiyonundan 11'inde % 100'ünde başarı göstermiştir. Bu başarının nedeni, Jaccard'ın benzerlik ölçümü ve mantık kapısı teknikleri ve transfer fonksiyonunun tipi ile aday çözümler arasında daha dengeli bir çeşitliliktir.

Çizelge 4.42 için, standart sapma sonuçlarına göre, BinSSA-Arctan, tüm çok modlu karşılaştırma fonksiyonlarında BinSSA-MSigm, BinSSA-Sigm ve BinSSA-Tanh'dan daha iyi performans göstermiştir. BinSSA-Arctan varyasyonu, 7 çok modlu kıyaslama fonksiyonundan 5'i ($f12$, $f13$, $f14$, $f15$ ve $f18$) için multimodal fonksiyonların % 71,43'ünde başarı göstermiştir. Ortalama sonuçlara göre, BinSSA-Arctan tüm çok modlu karşılaştırma fonksiyonlarında BinSSA-MSigm, BinSSA-Sigm ve BinSSA-Tanh'dan daha iyi performans göstermiştir. BinSSA-Arctan varyasyonu, 7 çok modlu kıyaslama fonksiyonundan 4'ü ($f13$, $f14$, $f15$ ve $f18$) için fonksiyonların % 57,14'ünde başarı göstermiştir. En iyi sonuçlara göre, BinSSA-Arctan tüm çok modlu karşılaştırma fonksiyonlarında BinSSA-Tanh, BinSSA-Sigm ve BinSSA-MSigm'den daha iyi performans göstermiştir. Sonuçlara göre, BinSSA-Tanh ve BinSSA-Arctan, birçok ölçüm kriterine göre (en iyi, en kötü, ortalama ve Std.Dev.) BinSSA'nın diğer varyasyonlarına kıyasla üstün performans göstermiştir. Gözlemlenen sonuçlara dayanarak, BinSSA-Tanh ve BinSSA-Arctan'ın diğerlerinden daha iyi kalite ve kararlılığa sahip olduğu sonucuna varılabilir.

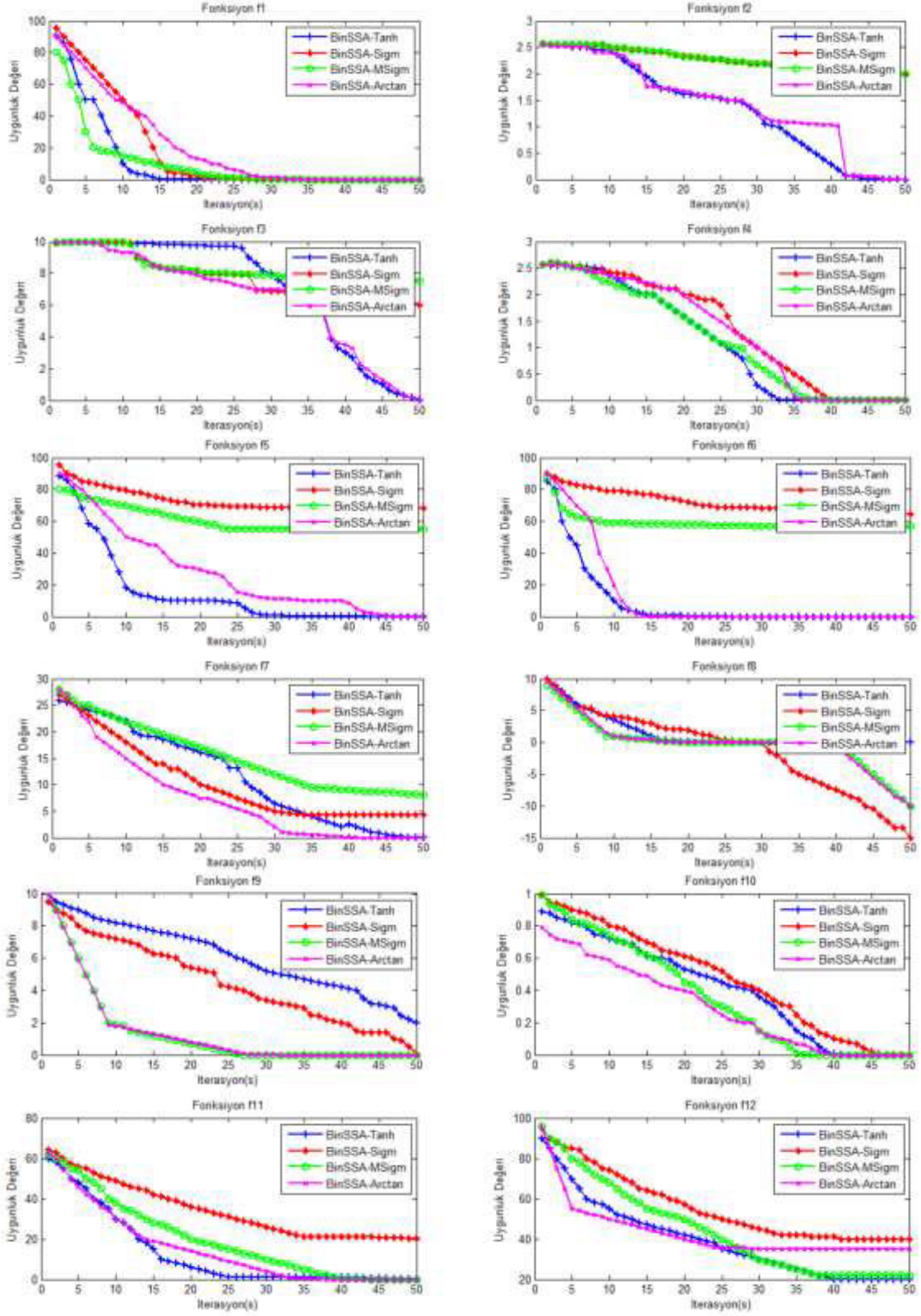
Çizelge 4.41. BinSSA varyasyonlarının tek modlu kıyaslama fonksiyonları üzerinde karşılaştırma sonuçları

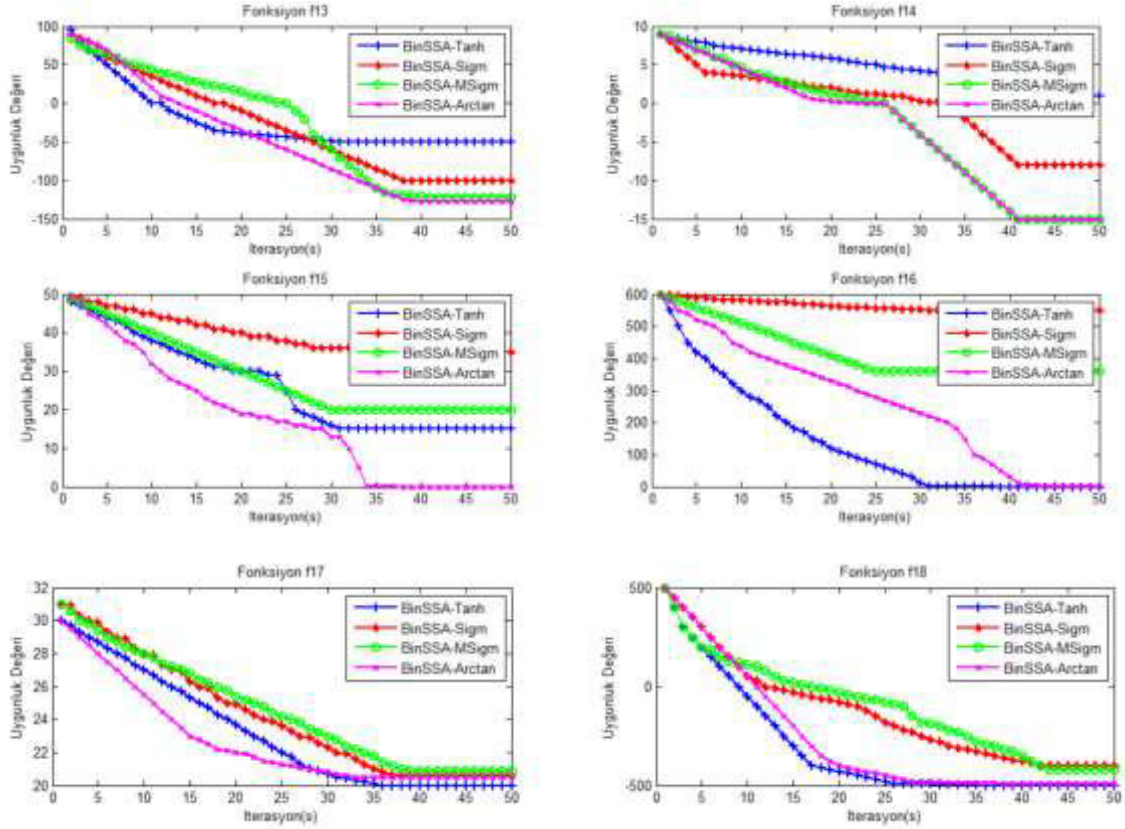
F		BinSSA-Tanh	BinSSA-Sigm	BinSSA-MSigm	BinSSA-Arctan
<i>f1</i>	En iyi	0,0000	0,0000	0,0000	0,0000
	Ortalama	0,0000	0,0000	0,0000	0,0000
	En kötü	0,0000	0,0000	0,0000	0,0000
	Std.Dev.	0,0000	0,0000	0,0000	0,0000
<i>f2</i>	En iyi	0,0000	2,0000	2,0000	0,0000
	Ortalama	1,0124	2,0410	2,1045	1,0214
	En kötü	1,5600	2,5600	2,3600	1,1500
	Std.Dev.	0,0280	0,2800	0,4500	0,0250
<i>f3</i>	En iyi	0,0000	6,0001	7,5000	0,0000
	Ortalama	0,0000	6,0001	7,5000	0,0000
	En kötü	0,0000	6,0001	7,5000	0,0000
	Std.Dev.	0,0000	0,0000	0,0000	0,0000
<i>f4</i>	En iyi	1,26E-05	4,26E-04	4,25E-04	1,22E-05
	Ortalama	0,000135	0,0535	0,0452	1,85E-04
	En kötü	0,00157	7,56E-03	6,85E-03	2,77E-03
	Std.Dev.	0,0006	0,0007	0,0007	0,0006
<i>f5</i>	En iyi	0,0000	68,0050	55,0400	0,0000
	Ortalama	1,0100	75,0900	60,7580	0,0000
	En kötü	6,0000	85,0600	65,7700	0,0000
	Std.Dev.	2,1200	10,0080	3,0020	0,0000
<i>f6</i>	En iyi	0,0000	62,4250	56,7450	0,0000
	Ortalama	0,0000	63,2500	66,9800	0,0000
	En kötü	0,0000	68,1100	87,5500	0,0000
	Std.Dev.	0,0000	5,4780	11,2540	0,0000
<i>f7</i>	En iyi	0,0000	4,0001	8,012700	0,0000
	Ortalama	13,0017	17,0127	15,1027	8,4150
	En kötü	18,0020	35,0002	29,0001	14,0000
	Std.Dev.	0,0129	0,1441	0,02100	0,0128
<i>f8</i>	En iyi	0,0000	-15,0001	-15,0001	-15,0001
	Ortalama	4,0174	-12,0014	-10,0042	-6,4710
	En kötü	10,0000	15,0001	15,0001	12,0002
	Std.Dev.	0,3340	0,3210	0,3220	0,3240
<i>f9</i>	En iyi	0,0000	1,00E-05	0,0000	0,0000
	Ortalama	2,1700	7,1700	0,0000	0,0000
	En kötü	10,0000	10,0000	0,0000	0,0000
	Std.Dev.	3,6704	4,8448	0,0000	0,0000
<i>f10</i>	En iyi	0,0000	0,0000	0,0000	0,0000
	Ortalama	0,0000	5,00E-01	0,0000	0,0000
	En kötü	0,0000	1,00E+00	0,0000	0,0000
	Std.Dev.	0,0000	0,5000	0,0000	0,0000
<i>f11</i>	En iyi	0,0000	20,0000	0,0000	0,0000
	Ortalama	1,3300	35,2000	1,00E-05	2,50E-05
	En kötü	8,0001	65,0000	4,00E-05	9,00E-05
	Std.Dev.	2,9800	13,0000	1,53E-05	3,25E-05

Çizelge 4.42. BinSSA varyasyonlarının çok modlu kıyaslama fonksiyonları üzerinde karşılaştırma sonuçları

F		BinSSA-Tanh	BinSSA-Sigm	BinSSA-MSigm	BinSSA-Arctan
<i>f12</i>	En iyi	4,0000	3,0000	1,000	17,0001
	Ortalama	35,7000	64,800	40,3000	54,0000
	En kötü	80,0010	100,0000	91,0001	86,0001
	Std.Dev.	24,0040	30,0745	29,7247	22,2411
<i>f13</i>	En iyi	-127,0000	-127,0000	-127,0000	-127,0000
	Ortalama	-19,500	-57,5000	-124,0000	-127,0000
	En kötü	127,0000	-125,0000	-111,0000	-127,0000
	Std.Dev.	67,8939	85,9569	5,9628	0,0000
<i>f14</i>	En iyi	0,0000	-15,0001	-15,0002	-15,0002
	Ortalama	2,8000	-0,333	-15,0001	-15,0002
	En kötü	6,0000	14,0000	-15,0001	-15,0002
	Std.Dev.	2,3152	10,8884	3,73E-05	1,78E-15
<i>f15</i>	En iyi	0,0000	31,0540	0,0000	0,0000
	Ortalama	25,0050	41,0010	29,0410	0,0000
	En kötü	47,0140	49,0140	38,0740	0,0000
	Std.Dev.	75,1000	55,1000	95,1000	0,0000
<i>f16</i>	En iyi	0,0000	529,4650	365,7485	0,0000
	Ortalama	0,0000	580,4650	368,7480	0,25470
	En kötü	0,0000	600,0000	524,7840	0,43257
	Std.Dev.	0,0000	233,4580	324,5740	0,09684
<i>f17</i>	En iyi	20,0015	20,5471	20,4175	20,5472
	Ortalama	20,0478	20,9870	21,1004	20,5147
	En kötü	20,5410	21,0141	21,9840	20,5470
	Std.Dev.	0,0147	0,0502	1,1470	0,0287
<i>f18</i>	En iyi	-500,0000	-421,1470	-435,8540	-498,0178
	Ortalama	-468,1247	-395,1240	-400,2140	-489,6045
	En kötü	-250,7480	-120,0124	-80,1450	-330,5480
	Std.Dev.	250,4012	240,1204	410,2100	140,2140

BinSSA varyasyonları için popülasyon (N) boyutu 30 ve maksimum yineleme 50 olarak seçilerek on sekiz tek modlu ve çok modlu karşılaştırma fonksiyonu için elde edilen uygunluk değerinin yakınsama grafikleri şekil 4.6'da gösterilmiştir.





Şekil 4.6. BinSSA varyasyonlarının benchmark fonksiyonları üzerinde yakınsama grafikleri (f1-f18)

Bu grafikten üç önemli sonuç çıkarılabilir;

i-) BinSSA-Tanh ve BinSSA-Arctan, seçilen fonksiyonlarda genellikle en iyi yakınsama hızını sağlar ve BinSSA-Tanh ve BinSSA-Arctan, İkili Sosyal Örümcek Algoritmasının diğer karşılaştırmalı varyasyonlarından daha iyi performans gösterir.

ii-) BinSSA-Tanh, BinSSA-MSigm ve BinSSA-Arctan'ın çoğu sonucu, optimizasyon işlemi sırasında sürekli bir gelişme gösterir.

Özellikle, BinSSA-Arctan varyasyonu, genellikle tüm kıyaslama test fonksiyonlarında hızlı bir şekilde optimum bir çözüme ulaşmıştır. Bu algoritmanın yakınsama hızı, aday çözüm üretim aşamasında eklenen mantık kapısı ve Jaccard'ın benzerlik ölçüm teknikleri ile artırılmıştır. Benzerlik ölçüm tekniği (SimSSA) ile bu algoritma, ikili arama alanında yeni noktalar bulmayı ve yerel optimumlar yerine global optimum bulmayı başarmıştır. Yeni ve başarılı bir teknik olan mantık kapıları son yıllarda çok popüler hale gelmiştir. Bu tezde, mantık kapılarına dayanan LogicSSA tekniği önerilmiştir. LogicSSA tekniği ile BinSSA, yerel arama alanındaki yerel tuzaklardan kurtarılmıştır ve yerel optimumları keşfetme yeteneği artırılmıştır.

BinSSA varyasyonları için Wilcoxon signed-rank testi:

Bu alt bölümde BinSSA varyasyonlarının sonuçları arasında kayda değer bir fark olup olmadığı Wilcoxon signed-rank testi uygulanarak bakılmıştır (Derrac ve ark., 2011). Her BinSSA varyasyonu on sekiz tek modlu ve çok modlu kıyaslama fonksiyonlarında 20 bağımsız çalışma yapılarak 20 örnekli veri setleri elde edilmiştir. Wilcoxon signed-rank test uygulamaları Matlab R2014a yazılımında gerçekleştirilmiştir. Sonuçlar çizelge 4.43’de gösterilmiştir.

BinSSA varyasyonları arasındaki ikili karşılaştırmalar için Wilcoxon testi uygulanarak elde edilen sonuçlara göre, BinSSA-Arctan ile BinSSA-MSigm, BinSSA-Sigm ve BinSSA-Tanh varyasyonları arasında en az %42 oranında anlamsal bir fark vardır.

Çizelge 4.43. BinSSA varyasyonları için Wilcoxon signed-rank test

F	(BinSSA-Arctan) - (BinSSA-Sigm)	(BinSSA-Arctan) - (BinSSA-MSigm)	(BinSSA-Arctan) - (BinSSA-Tanh)	(BinSSA-Sigm) - (BinSSA-MSigm)	(BinSSA-Sigm) - (BinSSA-Tanh)	(BinSSA-MSigm) - (BinSSA-Tanh)
	(P) h	(P) h	(P) h	(P) h	(P) h	(P) h
$f1(x)$	2,016E-05 1	NaN 0	0,0198 1	2,02E-05 1	0,0042 1	0,0198 1
$f2(x)$	0,3039 0	0,0050 1	0,1403 0	0,8816 0	0,4884 0	0,3363 0
$f3(x)$	0,0021 1	NaN 0	0,0400 1	0,0021 1	0,1607 0	0,0400 1
$f4(x)$	0,8710 0	0,0005 1	0,7149 0	0,0759 0	1 0	0,0616 0
$f5(x)$	0,0367 1	0,0079 1	0,2420 0	0,2769 0	0,1758 0	0,4508 0
$f6(x)$	3,72E-04 1	NaN 0	0,005E-04 1	3,71E-04 1	0,8217 0	4,05E-04 1
$f7(x)$	0,1624 0	0,0224 1	0,1624 0	4,68E-10 1	NaN 0	6,69E-10 1
$f8(x)$	0,3421 0	NaN 0	1 0	0,3421 0	0,5940 0	1 0
$f9(x)$	0,002 1	NaN 0	0,3421 0	0,0020 1	0,0084 1	0,3421 0
$f10(x)$	NaN 0	NaN 0	NaN 0	NaN 0	NaN 0	NaN 0
$f11(x)$	1,096E-06 1	NaN 0	1,64E-04 1	1,09E-06 1	0,05 1	1,64E-04 1
$f12(x)$	0,6072 0	0,0167 1	0,9667 0	0,7972 0	0,8709 0	0,7763 0
$f13(x)$	6,59E-05 1	6,59E-05 1	6,59E-05 1	4,68E-10 1	NaN 0	4,68E-10 1
$f14(x)$	0,3421 0	0,0421 1	0,3421 0	4,68E-10 1	NaN 0	4,68E-10 1
$f15(x)$	2,51E-05 1	NaN 0	0,0197 1	2,52E-05 1	0,0350 1	0,0197 1
$f16(x)$	0,3421 0	NaN 0	0,3421 0	0,3421 0	1 0	0,3421 0
$f17(x)$	1,09E-06 1	NaN 0	4,01E-04 1	1,09E-06 1	0,0192 1	4,0005E-04 1
$f18(x)$	3,92E-04 1	0,0031 1	5,33E-06 1	1,54E-04 1	0,0451 1	9,83E-07 1

BinSSA’nın literatürdeki diğer algoritmalar ile karşılaştırılması:

Literatürle kıyaslama çalışması 1:

Literatürdeki son zamanlarda yeni geliştirilen ve iyi bilinen algoritmalar çizelge 4.44’de gösterilmiştir. Eşit parametrelerde çeşitli tek modlu ve çok modlu fonksiyonlarda BinSSA, Parçacık Sürüsü Optimizasyonu (PSO), Genetik Algoritma

(GA), Dalgacık Mutasyonlu İkili Hibrit Parçacık Sürüsü Optimizasyonu (BHPSOWM), İkili Salp Sürüsü Algoritması (BSalpSA), İkili Parçacık Sürüsü Optimizasyonu (BPSO), İkili Yarasa Sürüsü Algoritması (BBA), İkili Sinüs Kosinüs Algoritması (BSCO) ve İkili Sosyal Örümcek Optimizasyonu (BSSO) ile karşılaştırılmıştır. Ortak kullanılan parametre ayarları çizelge 4.45’de gösterilmiştir.

Çizelge 4.44. Karşılaştırma algoritmaları için kullanılan ortak kısaltmalar

Algoritmalar	Açıklama	Referans
PSO	Parçacık Sürüsü Optimizasyonu	(Jiang ve ark., 2017)
GA	Genetik Algoritma	(Jiang ve ark., 2017)
BHPSOWM	Dalgacık Mutasyonlu İkili Hibrit Parçacık Sürüsü Optimizasyonu	(Jiang ve ark., 2017)
BSalpSA	İkili Salp Sürüsü Algoritması	(Rizk-Allah ve ark., 2019)
BPSO	İkili Parçacık Sürüsü Optimizasyonu	(Rizk-Allah ve Hassanien, 2018)
BBA	İkili Yarasa Sürüsü Algoritması	(Rizk-Allah ve Hassanien, 2018)
BSCA	İkili Sinüs Kosinüs Algoritması	(Rizk-Allah, 2018)
BSSO	İkili Sosyal Örümcek Optimizasyonu	(Cuevas ve ark., 2013; Shukla ve Nanda, 2018)
BinSSA	İkili Sosyal Örümcek Algoritması	

Çizelge 4.45. Parametre ayarları

Algoritma	Parametreler
BPSO	Atalet ağırlığı: $w_{min} = 0,4$; $w_{max} = 0,9$; ivme katsayıları: $c_1 = c_2 = 2$
BBA	Minimum ve maksimum frekanslar: $F_{min}=0$ and $F_{max} = 2$; darbe emisyon oranı: $r = 0,5$; gürültü: $A = 0,25$; sabitler: α ; $\gamma = 0,9$
BSCA	$r_1 = 1 - 0$ (doğrusal azalan parametre); $r_2 \in [0,2\pi]$, $r_3 \in [0,2]$
BSalpSA	$c_1 = 1 - 0$ (doğrusal olmayan azalan parametre); $c_2 \in [0,1]$, $c_3 \in [0,1]$
BinSSA-Tanh	$r_a=1$; $p_c=0,7$; $p_m=0,1$; $ST=0,3$
BinSSA-Sigm	$r_a=1$; $p_c=0,7$; $p_m=0,1$; $ST=0,3$
BinSSA-MSigm	$r_a=1$; $p_c=0,7$; $p_m=0,1$; $ST=0,3$
BinSSA-Arctan	$r_a=1$; $p_c=0,7$; $p_m=0,1$; $ST=0,3$
Ortak ayarlar	Popülasyon miktarı (N)= 30; bağımsız çalışma = 20; maksimum yenileme = 50

BPSO, BBA, BSCA, BSalpSA ve BinSSA varyasyonları dört farklı kriterde karşılaştırılmıştır. Bu karşılaştırma kriterleri: ortalama, standart sapma (Std.Dev.), en iyi ve en kötü’dür. Karşılaştırılan algoritmalarda, popülasyon miktarı (N) 30 olarak belirlenmiştir ve maksimum yinleme sayısı 50 olarak belirlenmiştir. Karşılaştırma sonuçları çizelge 4.46’da gösterilmiştir. Üstün performans gösteren sonuçlar kalın font ile işaretlenmiştir.

Çizelge 4.46. BinSSA varyasyonları, BPSO, BBA, BSCA ve BSalpSA algoritmalarının karşılaştırma sonuçları

F	BPSO	BBA	BSCA	BSalpSA	BinSSA-Arctan
f1					
En iyi	1,99E+09	1,76E+08	5,58E+09	0,0000	0,0000
Ortalama	3,42E+09	3,01E+08	7,82E+09	0,0000	0,0000
En kötü	4,45E+09	5E+08	8,69E+09	0,0000	0,0000
Std.Dev.	5,41E+08	97588,6050	7,56E+08	0,0000	0,0000
f3					
En iyi	39863,33	2278,4530	85869,500	7,5000	0,0000
Ortalama	51400,34	5407,4590	112325,20	7,5000	0,0000
En kötü	60413,45	7471,4220	138006,70	7,5000	0,0000
Std.Dev.	5783,194	1488,3310	15623,490	0,0000	0,0000
f4					
En iyi	2,55E+09	21957400	1,01E+10	3,48E-05	1,22E-05
Ortalama	3,33E+09	70533281	1,28E+10	0,000925	1,85E-04
En kötü	4,17E+09	1,39E+08	1,5E+10	0,001824	2,77E-03
Std.Dev.	7,21E+08	49613853	2,33E+09	0,0007	0,0006
f5					
En iyi	79,375	41,5000	102,5000	0,0000	0,0000
Ortalama	90,9125	54,4188	112,1125	0,0000	0,0000
En kötü	100,25	67,6250	118,5000	0,0000	0,0000
Std.Dev.	6,2326	7,5482	4,2774	0,0000	0,0000
f6					
En iyi	949,8750	254,8750	1565,125	0,0000	0,0000
Ortalama	1174,231	330,4313	1,37E+46	0,0000	0,0000
En kötü	1354,500	452,5000	1,36E+47	0,0000	0,0000
Std.Dev.	108,9784	53,8048	3,34E+46	0,0000	0,0000
f15					
En iyi	3,44E+09	1,4E+09	7,3E+09	0,0000	0,0000
Ortalama	4,29E+09	3E+09	8,71E+09	0,0000	0,0000
En kötü	5,23E+09	4,4E+09	1,02E+10	0,0000	0,0000
Std.Dev.	5,67E+08	7,8E+08	7,51E+08	0,0000	0,0000
f16					
En iyi	876800,5	304285,6	1678091	0,0000	0,0000
Ortalama	1078648	488418,4	2249554	0,0000	0,2547
En kötü	1355679	775537,3	2494999	0,0000	0,4326
Std.Dev.	130769,2	130750,7	202932,8	0,0000	0,0968
f17					
En iyi	21,0415	20,4844	21,1001	20,0517	20,5472
Ortalama	21,2117	20,6155	21,1911	20,0675	20,5147
En kötü	21,3183	20,7620	21,3143	20,1489	20,5470
Std.Dev.	0,0836	0,0824	0,0537	0,0240	0,0287
f18					
En iyi	- 1182,27	- 1492,81	- 1146,07	- 1499,51	-498,0178
Ortalama	- 1046,53	- 1114,58	- 1013,75	- 1066,95	-489,6045
En kötü	- 948,489	- 996,391	- 859,003	- 814,186	-330,5480
Std.Dev.	72,5534	141,134	103,7176	203,8991	140,2140

BinSSA varyasyonları, karşılaştırma sonuçlarına göre çeşitli karşılaştırma kriterlerinde seçilen kıyaslama fonksiyonlarında son derece iyi bir performans göstermiştir. Standart sapma sonuçlarına göre, BinSSA tüm tek modlu ve çok modlu kıyaslama fonksiyonlarında BPSO, BBA, BSCA ve BSalpSA'dan daha iyi performans göstermiştir. Ortalama ve standart sapma sonuçlarına göre, BinSSA-Arctan varyantı, 9 kıyaslama fonksiyonlarından 5'i (*f1*, *f3*, *f5*, *f6* ve *f15*) için üstün başarı göstermiştir. BSalpSA, BinSSA'dan sonra en iyi ikinci performans gösteren algoritmadır. Sonuçlar, BinSSA'nın tatmin edici standart sapma değerleri ile üstün çözümler bulduğunu göstermektedir. BinSSA bu başarıyı aday çözüm üretim aşamasında geliştirdiği Jaccard'ın benzerlik ölçümü ve mantık kapısı teknikleri sayesinde başarmıştır. Böylece,

ikili arama uzayında BinSSA'nın keşif ve sömürü kabiliyeti geliştirilmiştir. Bu nedenle, diğer karşılaştırma algoritmalarından daha avantajlı konuma gelmiştir.

Literatürle Kıyaslama Çalışması 2:

Social Spider Optimization (SSO), SSA algoritmasına çok benzeyen ancak ondan tamamen farklı bir başka sezgisel algoritmadır. SSO'da, arama uzayında iki farklı arama ajanı (örümcek) vardır. Bunlar erkek ve dişi örümceklerdir. Cinsiyete bağlı olarak, her birey kolonilerde tipik olarak bulunan çeşitli işbirliği davranışlarını taklit eden bir dizi farklı evrimsel operatörü çalıştırır (Cuevas ve ark., 2013; Shukla ve Nanda, 2018). SSA'da örümcekler arasında cinsiyet ayrımcılığı yoktur. Arama alanında tek bir tür arama ajanı (örümcek) vardır. Bu tezde, İkili Sosyal Örümcek Optimizasyonu (BSSO) ve BinSSA, çeşitli tek modlu ve çok modlu karşılaştırma fonksiyonlarında eşit karşılaştırma parametrelerinde karşılaştırılmıştır. Çizelge 4.47'de karşılaştırmada kullanılan parametre ayarları gösterilmiştir. Karşılaştırma sonuçları çizelgeler 4.48 ve 4.49'da gösterilmiştir. Dört farklı karşılaştırma kriteri mevcuttur. Bunlar: en iyi, ortalama, en kötü ve standart sapma (Std.Dev.)'dir.

Uygunluk fonksiyonu sonuçlarının ortalamasına göre BinSSA, BSSO'dan daha yüksek bir başarı göstermiştir. Uygunluk fonksiyonu sonuçlarının standart sapmasına göre, BinSSA, 18 kıyaslama fonksiyonundan ($f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16$ ve $f18$) 17 kıyaslama fonksiyonunda % 94,44'ünde ve BSSO, 18 kıyaslama fonksiyonundan ($f1, f3, f5, f7, f10, f13$ ve $f17$) 7 kıyaslama fonksiyonunda % 38,89'unda yüksek başarı göstermiştir. Uygunluk sonuçlarının ortalamasına göre, BSSO, 18 kıyaslama fonksiyonundan ($f1, f5, f10$ ve $f17$) 4 kıyaslama fonksiyonu için % 22,22'sinde başarı göstermektedir. Mantık kapısı ve Jaccard'ın benzerlik ölçüm teknikleri sayesinde BinSSA, BSSO'yu geçmiştir.

Çizelge 4.47. BinSSA ve BSSO için parametre ayarları

Algoritma	Parametreler
BSSO	$f_{pl}=0,65$ (Dişi yüzdesinin alt sınırı); $f_{pu}=0,9$ (Dişi yüzdesinin üst sınırı)
BinSSA	$r_a=1$; $p_c=0,7$; $p_m=0,1$; $ST=0,3$
Ortak ayarlar	Popülasyon miktarı (N)= 30; Bağımsız çalışma = 20; maksimum yenileme = 50

Çizelge 4.48. BinSSA varyasyonları ve BSSO'nun tek modlu kıyaslama fonksiyonlarında karşılaştırma sonuçları

F		BinSSA-Tanh	BinSSA-Sigm	BinSSA-MSigm	BinSSA-Arctan	BSSO
<i>f1</i>	En iyi	0,0000	0,0000	0,0000	0,0000	0,0000
	Ortalama	0,0000	0,0000	0,0000	0,0000	0,0000
	En kötü	0,0000	0,0000	0,0000	0,0000	0,0000
	Std.Dev.	0,0000	0,0000	0,0000	0,0000	0,0000
<i>f2</i>	En iyi	0,0000	2,0000	2,0000	0,0000	0,0000
	Ortalama	1,0124	2,0410	2,1045	1,0214	3,1020
	En kötü	1,5600	2,5600	2,3600	1,1500	5,2400
	Std.Dev.	0,0280	0,2800	0,4500	0,0250	0,3250
<i>f3</i>	En iyi	0,0000	6,0001	7,5000	0,0000	1,0579
	Ortalama	0,0000	6,0001	7,5000	0,0000	1,0579
	En kötü	0,0000	6,0001	7,5000	0,0000	1,0579
	Std.Dev.	0,0000	0,0000	0,0000	0,0000	0,0000
<i>f4</i>	En iyi	1,26E-05	4,26E-04	4,25E-04	1,22E-05	4,50E-03
	Ortalama	1,35E-04	5,35E-02	4,52E-02	1,85E-04	1,74E-02
	En kötü	1,57E-03	7,56E-03	6,85E-03	2,77E-03	7,14E-04
	Std.Dev.	0,0006	0,0007	0,0007	0,0006	0,0084
<i>f5</i>	En iyi	0,0000	68,0050	55,0400	0,0000	0,0000
	Ortalama	1,0100	75,0900	60,7580	0,0000	0,0000
	En kötü	6,0000	85,0600	65,7700	0,0000	0,0000
	Std.Dev.	2,1200	10,0080	3,0020	0,0000	0,0000
<i>f6</i>	En iyi	0,0000	62,4250	56,7450	0,0000	1,4428
	Ortalama	0,0000	63,2500	66,9800	0,0000	1,8501
	En kötü	0,0000	68,1100	87,5500	0,0000	2,5410
	Std.Dev.	0,0000	5,4780	11,2540	0,0000	0,0475
<i>f7</i>	En iyi	0,0000	4,0001	8,0127	0,0000	9,1710
	Ortalama	13,0017	17,0127	15,1027	8,4150	9,1710
	En kötü	18,0020	35,0002	29,0001	14,0000	9,1710
	Std.Dev.	0,0129	0,1441	0,0210	0,0128	0,0000
<i>f8</i>	En iyi	0,0000	-15,0001	-15,0001	-15,0001	0,8830
	Ortalama	4,0174	-12,0014	-10,0042	-6,4710	9,1240
	En kötü	10,0000	15,0001	15,0001	12,0002	15,0002
	Std.Dev.	0,3340	0,3210	0,3220	0,3240	15,4100
<i>f9</i>	En iyi	0,0000	1,00E-05	0,0000	0,0000	1,2386
	Ortalama	2,1700	7,1700	0,0000	0,0000	8,1900
	En kötü	10,0000	10,0000	0,0000	0,0000	9,1900
	Std.Dev.	3,6704	4,8448	0,0000	0,0000	3,4750
<i>f10</i>	En iyi	0,0000	0,0000	0,0000	0,0000	0,0000
	Ortalama	0,0000	0,5000	0,0000	0,0000	0,0000
	En kötü	0,0000	1,0000	0,0000	0,0000	0,0000
	Std.Dev.	0,0000	0,5000	0,0000	0,0000	0,0000
<i>f11</i>	En iyi	0,0000	20,0000	0,0000	0,0000	0,0562
	Ortalama	1,3300	35,2000	1,00E-05	2,50E-05	27,2000
	En kötü	8,0001	65,0000	4,00E-05	9,00E-05	40,2000
	Std.Dev.	2,9800	13,0000	1,53E-05	3,25E-05	21,0172

Çizelge 4.49. BinSSA varyasyonları ve BSSO'nun çok modlu kıyaslama fonksiyonlarında karşılaştırma sonuçları

F		BinSSA-Tanh	BinSSA-Sigm	BinSSA-MSigm	BinSSA-Arctan	BSSO
<i>f12</i>	En iyi	4,0000	3,0000	1,0000	17,0001	0,1999
	Ortalama	35,7000	64,8000	40,3000	54,0000	66,9000
	En kötü	80,0010	100,0000	91,0001	86,0001	92,9000
	Std.Dev.	24,0040	30,0745	29,7247	22,2411	30,1240
<i>f13</i>	En iyi	-1,27E+02	-1,27E+02	-1,27E+02	-1,27E+02	-17,4184
	Ortalama	-1,95E+01	-5,75E+01	-1,24E+02	-1,27E+02	-17,4184
	En kötü	1,27E+02	-1,25E+02	-1,11E+02	-1,27E+02	-17,4184
	Std.Dev.	67,8939	85,9569	5,9629	0,0000	0,0000
<i>f14</i>	En iyi	0,0000	-15,0001	-15,0002	-15,0002	7,94E-04
	Ortalama	2,8000	-3,33E-01	-15,000117	-15,000200	7,4800
	En kötü	6,0000	14,0000	-15,0001	-15,0002	16,9000
	Std.Dev.	2,3152	10,8884	3,73E-05	1,78E-15	4,4780
<i>f15</i>	En iyi	0,0000	31,0540	0,0000	0,0000	1,8700
	Ortalama	25,0050	41,0010	29,0410	0,0000	2,6700
	En kötü	47,0140	49,0140	38,0740	0,0000	3,0001
	Std.Dev.	75,1000	55,1000	95,1000	0,0000	15,2140
<i>f16</i>	En iyi	0,0000	529,4650	365,7485	0,0000	8,59E-04
	Ortalama	0,0000	580,4650	368,7480	0,2547	3,68E+02
	En kötü	0,0000	600,0000	0524,7840	0,4326	5,69E+02
	Std.Dev.	0,0000	233,4580	324,5740	0,0968	286,1200
<i>f17</i>	En iyi	20,0015	20,5471	20,4175	20,5472	1,2013
	Ortalama	20,0478	20,9870	21,1004	20,5147	1,2013
	En kötü	20,5410	21,0141	21,9840	20,5470	1,2013
	Std.Dev.	0,0147	0,0502	1,1470	0,0287	0,0000
<i>f18</i>	En iyi	-500,0000	-421,1470	-435,8540	-498,0178	-300,1407
	Ortalama	-468,1247	-395,1240	-400,2140	-489,6045	-259,1470
	En kötü	-250,7480	-120,0124	-80,1450	-330,5480	-17,5120
	Std.Dev.	250,4012	240,1204	410,2100	140,2140	300,1420

Literatürle kıyaslama çalışması 3:

BHPSOWM, PSO, GA ve BinSSA varyasyonları dört farklı kritere göre karşılaştırılmıştır. Bunlar ortalama, standart sapma (Std.Dev.), en iyi ve en kötü kriterleridir. Karşılaştırılan algoritmalarda, popülasyon büyüklüğü (N) 30 olarak ve maksimum yinleme sayısı 50 olarak belirlenmiştir. Sonuçlar çizelge 4.50'de gösterilmiştir. Üstün performans gösteren sonuçlar kalın font ile işaretlenmiştir.

Standart sapma ve ortalama sonuçlarına göre, BinSSA, tüm tek modlu ve çok modlu karşılaştırma fonksiyonlarında BHPSOWM, PSO ve GA'dan daha iyi bir performans göstermiştir. SimSSA ve LogicSSA tekniklerinin BinSSA'nın performansı üzerinde önemli bir etkisi vardır. Elde edilen sonuçlara dayanarak, önerilen BinSSA'nın BOP'lar için umut verici sonuçlar verdiği sonucuna varılmıştır.

Çizelge 4.50. BinSSA varyasyonları, BHPSOWM, PSO ve GA'nın karşılaştırma sonuçları

F	BHPSOWM	PSO	GA	BinSSA-Tanh	BinSSA-Sigm	BinSSA-MSigm	BinSSA-Arctan
f1							
En iyi	0,0000	312,1800	216,3200	0,0000	0,0000	0,0000	0,0000
Ortalama	42,3960	612,2510	487,2180	0,0000	0,0000	0,0000	0,0000
En kötü	-	-	-	0,0000	0,0000	0,0000	0,0000
Std.Dev.	52,1120	204,0760	175,8240	0,0000	0,0000	0,0000	0,0000
f3							
En iyi	7,5000	7,8430	7,7590	0,0000	6,0001	7,5000	0,0000
Ortalama	7,6140	8,6110	8,3570	0,0000	6,0001	7,5000	0,0000
En kötü	-	-	-	0,0000	6,0001	7,5000	0,0000
Std.Dev.	0,0670	0,3530	0,3380	0,0000	0,0000	0,0000	0,0000
f4							
En iyi	1,740 0	1,8000	1,7270	1,26E-05	4,26E-04	4,250E-04	1,22E-05
Ortalama	2,1150	2,0300	1,9810	1,35E-04	5,35E-02	4,52E-02	1,85E-04
En kötü	-	-	-	1,57E-03	7,56E-03	6,85E-03	2,77E-03
Std.Dev.	0,1110	0,0900	0,1120	0,0006	0,0007	0,0007	0,0006
f5							
En iyi	20,8000	16,0000	11,3000	0,0000	68,0050	55,0400	0,0000
Ortalama	25,2760	26,5240	14,1710	1,0100	75,0900	60,7580	0,0000
En kötü	-	-	-	6,0000	85,0600	65,7700	0,0000
Std.Dev.	1,3890	2,1860	1,2810	2,1200	10,0080	3,0020	0,0000
f6							
En iyi	0,0000	5,4200	4,3200	0,0000	62,4250	56,7450	0,0000
Ortalama	2,1460	10,8530	8,5200	0,0000	63,2500	66,9800	0,0000
En kötü	-	-	-	0,0000	68,1100	87,5500	0,0000
Std.Dev.	0,8130	2,5970	2,2780	0,0000	5,4780	11,2540	0,0000
f15							
En iyi	25,2790	115,5050	55,8390	0,0000	31,0540	0,0000	0,0000
Ortalama	76,182 0	275,2180	226,6330	25,0050	41,0010	29,0410	0,0000
En kötü	-	-	-	47,0140	49,0140	38,0740	0,0000
Std.Dev.	26,7490	78,3830	84,4340	75,1000	55,100	95,1000	0,0000
f16							
En iyi	10 (- 9)	10 (- 9)	10 (- 9)	0,0000	529,4650	365,7485	0,0000
Ortalama	10 (- 9)	10 (- 9)	10 (- 9)	0,0000	580,4650	368,7480	0,2547
En kötü	-	-	-	0,0000	600,0000	524,7840	0,4326
Std.Dev.	0,0000	0,0000	0,0000	0,0000	233,4580	324,5740	0,0968
f17							
En iyi	1,7180	1,7180	1,7180	20,0015	20,5471	20,4175	20,5472
Ortalama	1,7180	1,7180	1,7180	20,0478	20,9870	21,1004	20,5147
En kötü	-	-	-	20,5410	21,0141	21,9840	20,5470
Std.Dev.	0,0000	0,0000	0,0000	0,0147	0,0502	1,1470	0,0287
f18							
En iyi	946,4870	1153,4540	1210,0810	-500,0000	-421,1470	-435,8540	-498,0178
Ortalama	1984,8750	1960,3150	2072,1080	-468,1247	-395,1240	-400,2140	-489,6045
En kötü	-	-	-	-250,7480	-120,0124	-80,1450	-330,5480
Std.Dev.	267,4130	401,9830	472,0120	250,4012	240,1204	410,2100	140,2140

4.3.2. Kapasitesiz Tesis Yerleşim Problemi (UFLP) için BinSSA

Bu bölümde bir BOP olarak BinSSA ile Kapasitesiz Tesis Yerleşim Problemi (UFLP) çözülmüştür. UFL problemleri saf ikili optimizasyon problemlerinden biridir ve literatürde optimum değerleri bilinmektedir. BinSSA, iki farklı UFLP veri seti üzerinde (15 farklı ölçekli Cap veri seti ve 20 farklı ölçekli M* veri seti) test edilmiştir. Ayrıca BinSSA'nın performansı CEC2015 benchmark fonksiyonları üzerinde de test edilmiştir. Sürekli arama uzayının ikili arama uzayına dönüşümünde çizelge 3.1'de yer alan transfer fonksiyonlarından V4 tercih edilmiştir. SSA'da yer alan rastgele yürüyüş aşaması kaldırılarak aday çözüm üretim şeması eklenmiştir. Aday çözüm üretimi için

iki farklı yöntem geliştirilmiştir. Bunlar *Dice'in benzerlik ölçümü* ve *xor-maskesi* teknikleridir.

BinSSA, OR-Lib'den alınan düşük ölçekli, orta ölçekli ve büyük ölçekli on beş UFL problemi üzerinde test edilmiştir (Beasley, 1990). On beş UFL problemi çizelge 4.51'de gösterilmiştir.

Çizelge 4.51. Çalışmada kullanılan Cap veri setleri

Problem İsmi	Problem Boyutu	Optimum sonuç maliyeti
Cap71	16 x 50	932615,7500
Cap72	16 x 50	977799,4000
Cap73	16 x 50	1010641,4500
Cap74	16 x 50	1034976,9800
Cap101	25 x 50	796648,4400
Cap102	25 x 50	854704,2000
Cap103	25 x 50	893782,1100
Cap104	25 x 50	928941,7500
Cap131	50 x 50	793439,5600
Cap132	50 x 50	851495,3300
Cap133	50 x 50	893076,7100
Cap134	50 x 50	928941,7500
CapA	100 x 1000	17156454,4800
CapB	100 x 1000	12979071,5800
CapC	100 x 1000	11505594,3300

UFLP veri setleri boyutuna göre üç gruba ayrılır.

- *Grup-I:* Düşük ölçekli boyutlar (Cap71-Cap74)
- *Grup-II:* Orta ölçekli boyutlar (Cap101-Cap104 & Cap131-Cap134)
- *Grup-III:* Geniş ölçekli boyutlar (CapA-CapC)

BinSSA'da kullanılan parametre ayarları çizelge 4.52'de gösterilmiştir. BinSSA, her çalışma için aynı maksimum yineleme sayısı ile yirmi bağımsız çalışma şeklinde test edilmiştir.

Çizelge 4.52. BinSSA için parametre ayarları

Parametreler	Değerler
Popülasyon miktarı (N)	40
Maksimum yineleme	2000
Çalışma sayısı (n_{run})	20
Diğer parametreler	$r_a=1$; $p_c=0,7$; $p_m=0,1$; $ST=0,3$ (James ve Li, 2015a; Cinar ve Kiran, 2018)

UFL problemleri çözümünde BinSSA'nın performansını değerlendirmek için altı ölçüm kriteri kullanılmıştır. Bu kriterler: (1) ortalama (Mean), (2) standart sapma (Std.Dev.), (3) minimum değer (En İyi), (4) maksimum değer (En Kötü), (5) elde edilen en iyi çözüm için optimum çözümden sapma yüzdesi (Gap) ve (6) CPU zamanı (milisaniye cinsinden)'dir. Çizelge 4.53'de performans ölçümü kriterleri gösterilmiştir.

Çizelge 4.53. Performans ölçütleri

Ortalama:	$Mean = \frac{1}{n_{run}} \times \sum_{i=1}^{n_{run}} Fitness_func_i$
Standart Sapma:	$Std.Dev. = \sqrt{\frac{1}{n_{run}} \sum_{i=1}^{n_{run}} (gbest_i - Ortalama)^2}$
En iyi:	$Fitness_func_Best = \min_{i=1, n_{run}} Fitness_func_Best_i$
En kötü:	$Fitness_func_Worst = \max_{i=1, n_{run}} Fitness_func_Worst_i$
Gap:	$Gap = \frac{fitness_func(solution) - fitness_func(optimum)}{fitness_func(optimum)} \times 100$

BinSSA, on beş düşük, orta ve geniş ölçekli UFL problemleri (Cap veri setleri) üzerinde test edilmiştir. Her UFL problemi 20 kez bağımsız bir şekilde çalıştırılmıştır. Her UFL problemi için en iyi, en kötü, ortalama, standart sapma (Std.Dev.), gap ve CPU süresi hesaplanmıştır. BinSSA için popülasyon miktarı (N) = 40 ve maksimum yineleme = 2000 değerleri seçilerek on beş düşük, orta ve geniş ölçekli UFL problemlerinde elde edilen karşılaştırma sonuçları çizelge 4.54'de gösterilmiştir.

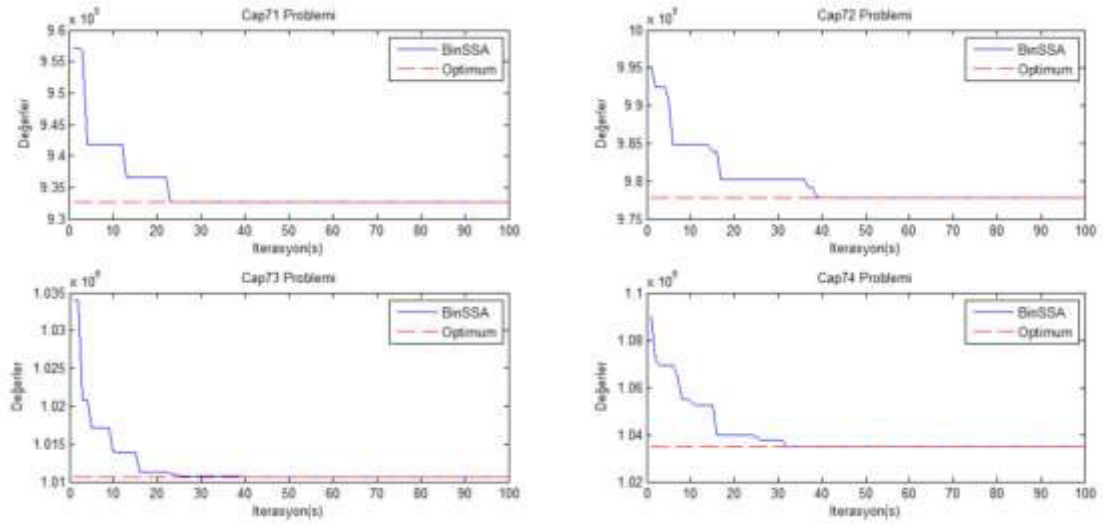
Çizelge 4.54. On beş düşük, orta ve geniş ölçekli UFL problemleri üzerinde BinSSA'nın test sonuçları

Problem	BinSSA					
	En iyi	Ortalama	En kötü	Std.Dev.	Gap	CPU Zamanı
Cap71	932615,7500	932615,7500	932615,7500	0,0000	0,0000	25,5150
Cap72	977799,4000	977799,4000	977799,4000	0,0000	0,0000	26,7450
Cap73	1010641,4500	1010641,4500	1010641,4500	0,0000	0,0000	27,5350
Cap74	1034976,9800	1034976,9800	1034976,9800	0,0000	0,0000	27,2350
Cap101	796648,4400	796648,4400	796648,4400	0,0000	0,0000	26,7450
Cap102	854704,2000	854704,2000	854704,2000	0,0000	0,0000	27,2140
Cap103	893782,1100	893782,1100	893782,1100	0,0000	0,0000	27,6600
Cap104	928941,7500	928941,7500	928941,7500	0,0000	0,0000	28,4570
Cap131	793439,5600	793439,5600	793439,5600	0,0000	0,0000	30,5780
Cap132	851495,3300	851495,3300	851495,3300	0,0000	0,0000	30,2780
Cap133	893076,7100	893076,7100	893076,7100	0,0000	0,0000	31,9850
Cap134	928941,7500	928941,7500	928941,7500	0,0000	0,0000	31,5460
CapA	17156454,4800	17156454,4800	17156454,4800	0,0000	0,0000	476,2570
CapB	12995024,8950	13012123,5230	13092124,5420	36156,1400	0,254656	497,4850
CapC	11525542,3300	11555489,0820	11950239,5040	27598,8000	0,433656	488,4750

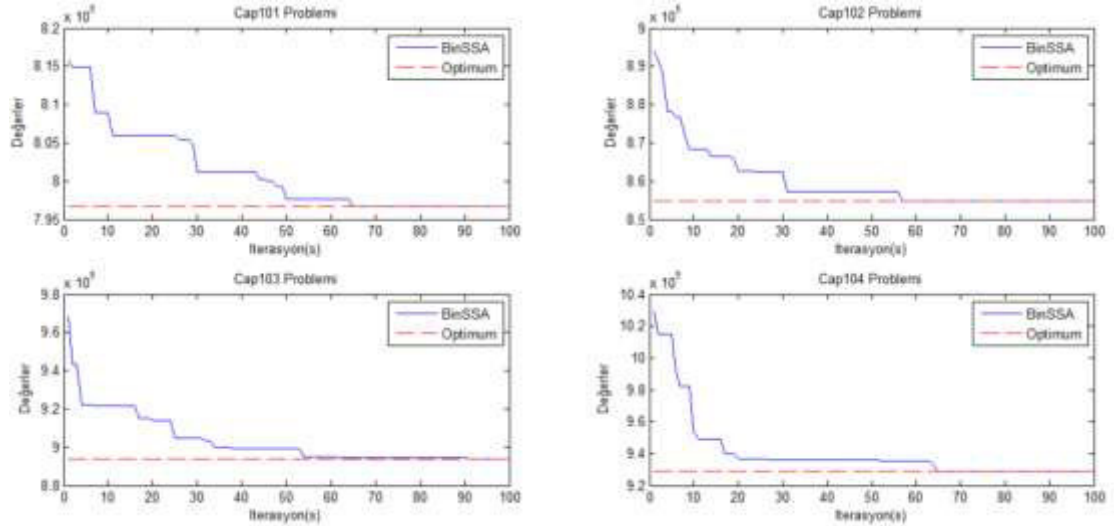
Şekil 4.7'de Cap71, Cap72, Cap73 ve Cap74 problemleri için yakınsama grafikleri gösterilmiştir. Şekil 4.8'de Cap101, Cap102, Cap103 ve Cap104 problemleri için yakınsama grafikleri gösterilmiştir. Şekil 4.9'da Cap131, Cap132, Cap133 ve

Cap134 problemleri için yakınsama grafikleri gösterilmiştir ve Şekil 4.10'da CapA, CapB ve CapC problemleri için yakınsama grafikleri gösterilmiştir.

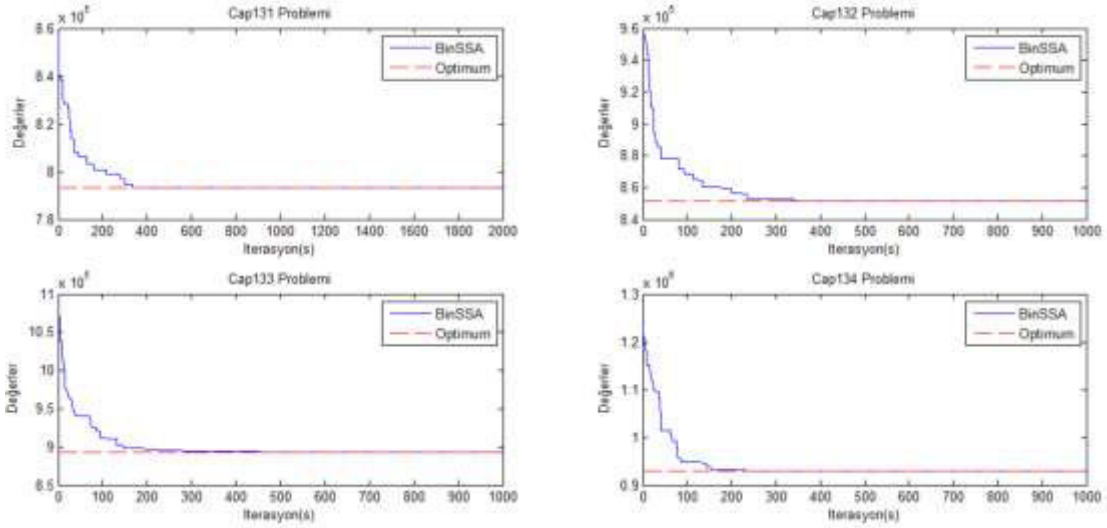
Elde edilen sonuçlara göre, BinSSA % 0 hata ile CapB ve CapC hariç hemen hemen tüm UFL problemlerinde optimum sonuçlara ulaşmıştır. Aday arama üretim şemasında kullanılan teknikler, BinSSA'nın ikili arama uzayında küresel ve yerel arama yeteneklerini geliştirmiştir.



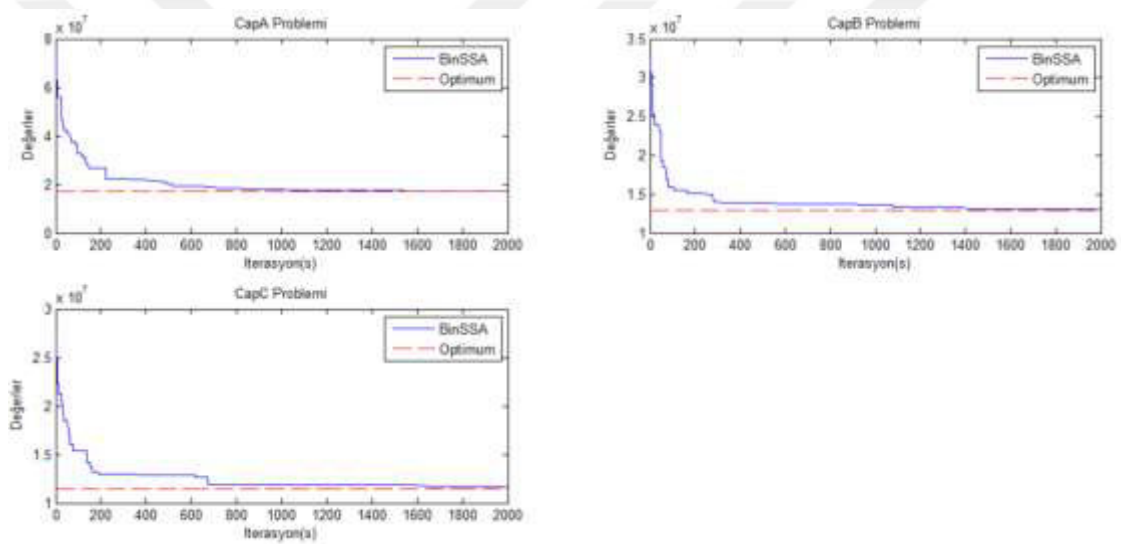
Şekil 4.7. Cap71, Cap72, Cap73 ve Cap74 UFL problemleri için yakınsama grafikleri



Şekil 4.8. Cap101, Cap102, Cap103 ve Cap104 UFL problemleri için yakınsama grafikleri



Şekil 4.9. Cap131, Cap132, Cap133 ve Cap134 UFL problemleri için yakınsama grafikleri



Şekil 4.10. CapA, CapB ve CapC UFL problemleri için yakınsama grafikleri

BinSSA'nın literatürdeki diğer algoritmalar ile karşılaştırılması:

BinSSA'nın literatürde karşılaştırıldığı algoritmalar: farklı çapraz operatörlü İkili Genetik Algoritmalar (Korkmaz ve ark., 2018), İkili Yapay Alg Algoritmasının üç versiyonu (Korkmaz ve Kiran, 2018), İkili Yapay Arı Kolonisinin üç versiyonu (Kiran ve Gündüz, 2013; Kiran, 2015), İkili Parçacık Sürüşü Optimizasyonunun iki versiyonu (Korkmaz ve Kiran, 2018), İkili Diferansiyel Evrim algoritmasının iki versiyonu (Kashan ve ark., 2013), İkili Ağaç Tohum Algoritması (Cinar ve Kiran, 2018), İkili Geliştirilmiş Dağılım Arama Algoritması (Hakli ve Ortacay, 2019), İkili Jaya Algoritmasının üç versiyonu (BJA, JayaX ve JayaX-LSM) (Aslan ve ark., 2019) ve İkili Sosyal Örümcek Algoritması (BinSSA)'dır.

Literatürle kıyaslama çalışması 1:

Farklı çaprazlama operatörü olan İkili Genetik Algoritmalar (GA), İkili Yapay Alg Algoritması'nın (BAAA) üç versiyonu ve BinSSA (V4 transfer fonksiyonu kullanılmıştır), düşük ölçekli, orta ölçekli ve büyük ölçekli UFL problemlerinde eşit karşılaştırma parametrelerinde karşılaştırılmıştır. Adil bir karşılaştırma için; popülasyon büyüklüğü, bağımız çalışma sayısı ve maksimum iterasyon karşılaştırma için seçilen yöntemlerde eşit olarak belirlenmiştir. Tüm karşılaştırma algoritmalarında kullanılan parametre ayarları çizelge 4.55'de gösterilmiştir.

Çizelge 4.55. Parametre ayarları

Algoritma	Parametre açıklaması
GA	Çaprazlama ve mutasyon oranları, genetik algoritmanın tek noktalı çaprazlama, iki noktalı çaprazlama ve düzgün çaprazlama versiyonları için sırasıyla 0,8 ve 0,01 olarak seçilir.
BAAA	BAAA algoritmasının her iki sürümü için, kesme kuvveti kontrol parametresi 2'ye, enerji kaybı kontrol parametresi 0,3'e ve adaptasyon hızı kontrol parametresi 0,5'e ayarlanır. Eğri (T) kontrol parametrelerinin değişim eğilimi Tanh (x) ve Sig (x) lojistik fonksiyonları için sırasıyla 1,5 ve 2 olarak seçilir.
BinSSA	$r_a=1$; $p_c=0,7$; $p_m=0,1$; $ST=0,3$
Ortak ayarlar	Popülasyon miktarı (N)= 40; bağımsız çalışma sayısı = 30; maksimum yenileme = 2000

GA-SP, GA-TP, GA-UP, BAAA-Tanh, BAAA-Sig, BAAASB ve BinSSA, UFL problemleri için dört farklı kritere göre karşılaştırılmıştır. Bunlar ortalama, Std.Dev., gap ve CPU zamanıdır. Karşılaştırılan algoritmaların deneysel sonuçları doğrudan Korkmaz ve Kiran'dan alınmıştır ve algoritmaların adil bir karşılaştırmasını yapmak için önerilen algoritmanın ortak kontrol parametreleri Korkmaz ve Kiran'a göre seçilmiştir (Korkmaz ve Kiran, 2018). Karşılaştırma sonuçları UFL problemleri için çizelgeler 4.56 – 4.59'da gösterilmiştir. Üstün performans gösteren sonuçlar kalın font olarak işaretlenmiştir.

Ortalama, standart sapma (Std.Dev.) ve gap'e göre, BinSSA tüm düşük, orta ve büyük ölçekli UFL problemlerinde istikrarlı performans göstermiştir. BinSSA, UFL problemlerinin % 86,67'sinde başarı göstermiştir (15 UFLP'den 13'ü için). BinSSA, düşük ve orta boyutlu UFLP örneklerinde kısa sürede optimum sonuçlar elde etmiştir. Fakat yüksek ölçekli UFL problemlerinde BAAASB'yi geçememiştir. BinSSA'nın bu başarısının nedeni çeşitlendirme ve yoğunlaşma arasında daha istikrarlı bir denge kurabilmesidir. BinSSA'nın keşif ve sömürü yetenekleri, ikili arama uzayında *Dice'in benzerlik ölçümü* ve ikili *xor maskesi* teknikleri ile geliştirilmiştir.

Çizelge 4.56. BinSSA, GA-SP, GA-TP, GA-UP, BAAASB, BAAA-Tanh ve BAAA-Sig algoritmalarının düşük ölçekli UFLP örneklerinde karşılaştırma sonuçları (Cap71-Cap74)

	GA-SP	GA-TP	GA-UP	BAAA-Tanh	BAAA-Sig	BAAASB	BinSSA
<i>Cap71</i>							
Ortalama	932615,750	932615,750	932615,7500	932615,7500	932615,7500	932615,7500	932615,7500
Std.Dev.	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Gap	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Zaman	26,9570	27,5680	27,7440	25,7480	25,8680	27,9010	25,5150
<i>Cap72</i>							
Ortalama	977799,400	977799,400	977799,4000	977799,4000	977799,4000	977799,4000	977799,4000
Std.Dev.	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Gap	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Zaman	27,8930	28,0560	28,1030	26,2790	26,9270	28,0780	26,7450
<i>Cap73</i>							
Ortalama	1011314,476	1011130,9230	1011069,7390	1010641,4500	1010641,4500	1010641,4500	1010641,450
Std.Dev.	899,6500	825,5760	789,6120	0,0000	0,0000	0,0000	0,0000
Gap	0,06659	0,04843	0,04238	0,0000	0,0000	0,0000	0,0000
Zaman	27,9940	28,0500	28,1010	28,2210	28,2190	27,5280	27,5350
<i>Cap74</i>							
Ortalama	1034976,975	1034976,975	1034976,975	1034976,975	1034976,975	1034976,975	1034976,975
Std.Dev.	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Gap	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Zaman	27,9980	28,430	28,0950	27,5840	27,4610	27,2390	27,2350

Çizelge 4.57. BinSSA, GA-SP, GA-TP, GA-UP, BAAASB, BAAA-Tanh ve BAAA-Sig algoritmalarının orta ölçekli UFLP örneklerinde karşılaştırma sonuçları (Cap101-Cap104)

	GA-SP	GA-TP	GA-UP	BAAA-Tanh	BAAA-Sig	BAAASB	BinSSA
<i>Cap101</i>							
Ortalama	797193,2860	797164,6100	797107,2580	796677,1140	796648,4380	796648,4380	796648,4380
Std.Dev.	421,6550	428,6580	436,5240	157,0660	0,0000	0,0000	0,0000
Gap	0,06839	0,06479	0,05759	0,00360	0,0000	0,0000	0,0000
Zaman	29,3720	29,2060	29,1690	27,473	26,8360	28,1770	26,7450
<i>Cap102</i>							
Ortalama	854704,2000	854704,2000	854704,2000	854704,2000	854704,2000	854704,2000	854704,2000
Std.Dev.	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Gap	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Zaman	28,7300	28,9310	28,8700	26,3340	26,2150	27,9230	27,2140
<i>Cap103</i>							
Ortalama	894351,7820	894329,1790	894427,3820	893782,1130	928941,7500	928941,7500	928941,7500
Std.Dev.	505,0360	540,1600	522,7840	0,0000	0,0000	0,0000	0,0000
Gap	0,06374	0,06121	0,07220	0,0000	0,0000	0,0000	0,0000
Zaman	28,6890	28,9070	28,9150	25,8510	25,9260	27,8380	27,6600
<i>Cap104</i>							
Ortalama	928941,7500	928941,7500	928941,750	928941,750	928941,750	928941,750	928941,750
Std.Dev.	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Gap	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Zaman	32,7060	32,9920	33,0460	25,4350	26,9630	27,5920	28,4570

Çizelge 4.58. BinSSA, GA-SP, GA-TP, GA-UP, BAAASB, BAAA-Tanh ve BAAA-Sig algoritmalarının orta ölçekli UFLP örneklerinde karşılaştırma sonuçları (Cap131-Cap134)

	GA-SP	GA-TP	GA-UP	BAAA-Tanh	BAAA-Sig	BAAASB	BinSSA
Cap131							
Ortalama	793980,1040	794012,9050	793865,0230	793525,5910	793439,5630	793439,5630	793439,5630
Std.Dev.	720,8770	690,5600	433,4670	262,4980	0,0000	0,0000	0,0000
Gap	0,06813	0,07226	0,05362	0,01084	0,0000	0,0000	0,0000
Zaman	34,0170	33,8850	33,5630	45,4980	63,6560	28,0800	30,5780
Cap132							
Ortalama	851495,3250	851495,3250	851517,2000	851495,3250	851495,3250	851495,3250	851495,3250
Std.Dev.	0,0000	0,0000	119,8170	0,0000	0,0000	0,0000	0,0000
Gap	0,0000	0,0000	0,00257	0,0000	0,0000	0,0000	0,0000
Zaman	35,1070	35,9300	36,3290	57,0390	67,1320	28,5390	30,2780
Cap133							
Ortalama	893891,9110	893740,9540	893808,8910	893333,5150	893076,7130	893076,7130	893076,7130
Std.Dev.	685,0760	655,9200	628,6540	324,4510	0,0000	0,0000	0,0000
Gap	0,09128	0,07438	0,08198	0,02875	0,0000	0,0000	0,0000
Zaman	38,1430	37,9060	38,0380	56,2390	66,9430	28,3360	31,9850
Cap134							
Ortalama	928941,7500	928941,7500	928941,7500	928941,7500	928941,7500	928941,7500	928941,7500
Std.Dev.	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Gap	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
Zaman	36,7480	36,6090	36,5880	56,6220	67,6120	28,0110	31,5460

Çizelge 4.59. BinSSA, GA-SP, GA-TP, GA-UP, BAAASB, BAAA-Tanh ve BAAA-Sig algoritmalarının orta ölçekli UFLP örneklerinde karşılaştırma sonuçları (CapA-CapC)

	GA-SP	GA-TP	GA-UP	BAAA-Tanh	BAAA-Sig	BAAASB	BinSSA
CapA							
Ortalama	17164354,456	17205089,145	17166811,915	17471223,794	17210900,533	17156454,480	17156454,480
Std.Dev.	22451,2060	139690,2160	35181,9740	225123,9210	90743,4560	0,0000	0,0000
Gap	0,04605	0,28348	0,06037	1,83470	0,31735	0,0000	0,0000
Zaman	741,5350	735,9390	743,8480	864,7550	880,4520	461,9060	476,2570
CapB							
Ortalama	13054858,045	13063527,186	13107633,077	13153617,764	13093705,559	13011234,616	13012123,523
Std.Dev.	66658,6490	89122,4850	79714,0210	73978,5430	62168,8030	39224,7440	36156,1400
Gap	0,58391	0,65071	0,99053	1,34483	0,88322	0,24781	0,254656
Zaman	743,3700	738,3870	748,1440	473,6600	475,5920	470,0940	497,4850
CapC							
Ortalama	11586692,969	11577797,524	11578600,532	11676427,752	11583462,068	11539496,443	11555489,082
Std.Dev.	51848,2480	46346,0520	57031,2190	101438,6070	45788,6780	29766,3110	27598,8000
Gap	0,70486	0,62755	0,63453	1,48479	0,67678	0,2466	0,433656
Zaman	744,4550	741,8230	745,1280	479,7620	470,9110	455,4700	488,4750

Literatürle kıyaslama çalışması 2:

ABC algoritmasının ikili versiyonları ile BinSSA, düşük, orta ve yüksek ölçekli UFL problemlerinde karşılaştırılmıştır. Karşılaştırılan algoritmaların deneysel sonuçları doğrudan Kiran ve Gunduz'den alınmıştır ve algoritmaların adil bir karşılaştırmasını yapmak için önerilen algoritmanın ortak kontrol parametreleri Kiran ve Gunduz'e göre seçilmiştir (Kiran ve Gündüz, 2013; Kiran, 2015). Karşılaştırılan algoritmalarında, popülasyon büyüklük değeri 40 ve maksimum yineleme sayısı 2000 eşit olarak belirlenmiştir. Karşılaştırma sonuçları UFL problemleri için çizelge 4.60'da gösterilmiştir. Üstün performans gösteren sonuçlar kalın font olarak işaretlenmiştir.

Gap sonuçlarına göre, BinSSA, UFL problemlerinde yüksek performans göstermiştir. BinSSA, UFL problemlerinin % 100'ünde başarı göstermiştir (15 UFLP'den 15'i için). ABC_{Bin}, 15 UFLP'den 8'inde (Cap71, Cap72, Cap73, Cap74, Cap101, Cap102, Cap104 ve Cap134) yaklaşık % 53,33 UFL problemlerinde başarı göstermiştir. DisABC, 15 UFLP'den 9'unda (Cap71, Cap72, Cap73, Cap74, Cap101, Cap102, Cap103, Cap104 ve Cap134) UFL problemlerinin % 60'ında başarı göstermiştir. BinABC, 15 UFLP'den 11'inde (Cap71, Cap72, Cap73, Cap74, Cap101, Cap102, Cap103, Cap104, Cap131, Cap132 ve Cap134) UFL problemlerinin % 73,33'ünde başarı gösterir. Sonuçlar göstermiştir ki Dice'nin benzerlik ölçümü ve xor maskesi tekniklerinin küresel ve yerel olarak BinSSA'nın arama yeteneğini geliştirdiğini göstermektedir. Benzerlik ölçüsü ve xor maskesi teknikleri ile üretilen yeni aday çözüm seti istikrarlı bir çözüm sağlamıştır.

Çizelge 4.60. BinSSA'nın ABC'nin ikili versiyonları ile karşılaştırılması sonuçları

Problem	ABC _{Bin}		DisABC		BinABC		BinSSA	
	Gap	Rank	Gap	Rank	Gap	Rank	Gap	Rank
Cap71	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap72	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap73	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap74	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap101	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap102	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap103	5,1E-03	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap104	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap131	2,0E-01	2	6,2E-01	3	0,0E+00	1	0,0E+00	1
Cap132	2,0E-02	2	9,5E-02	3	0,0E+00	1	0,0E+00	1
Cap133	7,5E-02	3	3,1E-02	2	1,2E-01	4	0,0E+00	1
Cap134	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
CapA	3,2E+00	4	1,5E-01	2	3,0E+00	3	0,0E+00	1
CapB	2,8E+00	3	3,3E+00	4	2,5E+00	2	2,55E-01	1
CapC	2,0E+00	2	4,7E+00	4	2,6E+00	3	4,34E-01	1

Literatürle kıyaslama çalışması 3:

DE algoritmasının ikili versiyonları ile BinSSA, düşük, orta ve yüksek ölçekli UFL problemlerinde karşılaştırılmıştır. Karşılaştırılan algoritmaların deneysel sonuçları doğrudan Kashan ve arkadaşlarından alınmıştır (Kashan ve ark., 2013). Karşılaştırılan algoritmalarında, popülasyon büyüklük değeri 40 ve maksimum yineleme sayısı 2000 eşit olarak belirlenmiştir. Karşılaştırma sonuçları UFL problemleri için çizelge 4.61'de gösterilmiştir. Üstün performans gösteren sonuçlar kalın font olarak işaretlenmiştir.

Gap sonuçlarına göre, BinSSA UFL problemlerinin % 86,67'sinde (15 UFLP'den 13'ü için) başarı göstermektedir. BinSSA, CapB ve CapC dışındaki tüm UFL problemlerinde üstün başarı göstermiştir. DisDE, UFL problemlerinin % 73,33'ünde başarı göstermiştir (15 UFLP'den 11'i için). BinDE, UFL problemlerinin % 60'ında

başarı göstermiştir (15 UFLP'den 9'u için). BinSSA'nın bu başarısının nedeni, Dice'in benzerlik ölçümü ve xor maske teknikleri arasındaki daha dengeli bir çeşitlenmeden kaynaklanmaktadır.

Çizelge 4.61. BinSSA'nın DE'nin ikili versiyonları ile karşılaştırılması sonuçları

Problem	DisDE		BinDE		BinSSA	
	Gap	Rank	Gap	Rank	Gap	Rank
Cap71	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap72	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap73	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap74	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap101	7,2E-03	2	0,0E+00	1	0,0E+00	1
Cap102	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap103	8,4E-04	2	0,0E+00	1	0,0E+00	1
Cap104	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap131	0,0E+00	1	3,6E-03	2	0,0E+00	1
Cap132	0,0E+00	1	5,0E-03	2	0,0E+00	1
Cap133	1,5E-02	3	1,4E-02	2	0,0E+00	1
Cap134	0,0E+00	1	0,0E+00	1	0,0E+00	1
CapA	3,7E-02	2	1,3E+00	3	0,0E+00	1
CapB	6,7E-02	1	1,5E+00	3	2,55E-01	2
CapC	5,8E-02	1	1,6E+00	3	4,34E-01	2

Literatürle kıyaslama çalışması 4:

PSO algoritmasının ikili versiyonları ile BinSSA, düşük, orta ve yüksek ölçekli UFL problemlerinde karşılaştırılmıştır. Karşılaştırılan algoritmaların deneysel sonuçları doğrudan Korkmaz ve Kiran'dan alınmıştır (Korkmaz ve Kiran, 2018). Karşılaştırılan algoritmalarında, popülasyon büyüklük değeri 40 ve maksimum yineleme sayısı 2000 eşit olarak belirlenmiştir. Karşılaştırma sonuçları UFL problemleri için çizelge 4.62'de gösterilmiştir. Üstün performans gösteren sonuçlar kalın font olarak işaretlenmiştir.

Çizelge 4.62. BinSSA'nın PSO'nun ikili versiyonları ile karşılaştırılması sonuçları

Problem	BPSO		CPSO		BinSSA	
	Gap	Rank	Gap	Rank	Gap	Rank
Cap71	0,0E+00	1	5,0E-02	2	0,0E+00	1
Cap72	0,0E+00	1	7,0E-02	2	0,0E+00	1
Cap73	2,42E-02	2	6,0E-02	3	0,0E+00	1
Cap74	8,82E-03	2	7,0E-02	3	0,0E+00	1
Cap101	4,32E-02	2	1,4E-01	3	0,0E+00	1
Cap102	9,89E-03	2	1,5E-01	3	0,0E+00	1
Cap103	4,94E-02	2	1,6E-01	3	0,0E+00	1
Cap104	4,05E-02	2	1,8E-01	3	0,0E+00	1
Cap131	1,71E-01	2	7,5E-01	3	0,0E+00	1
Cap132	5,83E-02	2	7,8E-01	3	0,0E+00	1
Cap133	8,29E-02	2	7,3E-01	3	0,0E+00	1
Cap134	1,95E-01	2	8,9E-01	3	0,0E+00	1
CapA	1,7E+00	2	2,2E+01	3	0,0E+00	1
CapB	1,4E+00	2	1,1E+01	3	2,55E-01	1
CapC	1,6E+00	2	9,7E+00	3	4,34E-01	1

BinSSA, UFL problemlerinin hepsinde (% 100'ünde) başarı göstermiştir (15 UFLP'den 15'i için). BPSO, UFL problemlerinin % 13,33'ünde (15 UFLP'den 2'si için)

başarı göstermiştir. CPSO, UFL problemlerinin hiçbirinde başarı gösterememiştir. BinSSA'nın yüksek performansındaki en önemli faktör, Dice'nın benzerlik ölçümü ve xor maskesi teknikleridir. Böylece BinSSA diğer karşılaştırma algoritmalarına göre üstünlük göstermiştir.

Literatürle kıyaslama çalışması 5:

TSA ve ISS algoritmalarının ikili versiyonları ile BinSSA, düşük, orta ve yüksek ölçekli UFL problemlerinde karşılaştırılmıştır. Karşılaştırılan algoritmaların deneysel sonuçları doğrudan Çınar ve Kiran'dan alınmıştır (Cinar ve ark., 2019). Karşılaştırılan algoritmalarında, popülasyon büyüklük değeri 40 ve maksimum yineleme sayısı 2000 eşit olarak belirlenmiştir. Karşılaştırma sonuçları UFL problemleri için çizelge 4.63'de gösterilmiştir. Üstün performans gösteren sonuçlar kalın font olarak işaretlenmiştir.

BinSSA, UFL problemlerinin % 93,33'ünde başarı göstermiştir (15 UFLP'den 14'ü için). TSA, UFL problemlerinin % 86,67'sinde başarı göstermiştir (15 UFLP'den 13'ü için). ISS, UFL problemlerinin % 93,33'ünde başarı göstermiştir (15 UFLP'den 14'ü için).

Çizelge 4.63. BinSSA'nın TSA ve ISS'in ikili versiyonları ile karşılaştırılması sonuçları

Problem	TSA		ISS		BinSSA	
	Gap	Rank	Gap	Rank	Gap	Rank
Cap71	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap72	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap73	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap74	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap101	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap102	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap103	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap104	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap131	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap132	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap133	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap134	0,0E+00	1	0,0E+00	1	0,0E+00	1
CapA	0,0E+00	1	0,0E+00	1	0,0E+00	1
CapB	0,3176	3	0,2555	2	0,2546	1
CapC	0,4120	2	0,1989	1	0,4336	3

Literatürle kıyaslama çalışması 6:

Jaya algoritmasının ikili versiyonları ile BinSSA, düşük, orta ve yüksek ölçekli UFL problemlerinde karşılaştırılmıştır. Karşılaştırılan algoritmaların deneysel sonuçları doğrudan Aslan ve arkadaşlarından alınmıştır (Aslan ve ark., 2019). Karşılaştırılan algoritmalarında, popülasyon büyüklük değeri 40 ve maksimum yineleme sayısı 8E+5

eşit olarak belirlenmiştir. Karşılaştırma sonuçları UFL problemleri için çizelge 4.64’de gösterilmiştir. Üstün performans gösteren sonuçlar kalın font olarak işaretlenmiştir.

Sonuçlara göre, BinSSA, JayaX-LSM'den sonra en başarılı algoritmadır. BinSSA, UFL problemlerinin % 86,67'sinde yüksek başarı göstermiştir (15 UFLP'den 13'ü için). JayaX-LSM, UFL problemlerinin hepsinde başarı göstermiştir (15 UFLP'den 15'i için). JayaX, UFL problemlerinin % 80'inde başarı göstermiştir (15 UFLP'den 12'si için). BJA, UFL problemlerinin % 20'sinde başarı göstermiştir (15 UFLP'den 3'ü için).

Çizelge 4.64. BinSSA’nın Jaya’nın ikili versiyonları ile karşılaştırılması sonuçları

Problem	BJA		JayaX		JayaX-LSM		BinSSA	
	Gap	Rank	Gap	Rank	Gap	Rank	Gap	Rank
Cap71	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap72	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap73	0,01211	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap74	0,04412	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap101	0,01800	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap102	0,01509	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap103	0,02153	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap104	0,0E+00	1	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap131	0,14290	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap132	0,11215	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap133	0,13623	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
Cap134	0,02459	2	0,0E+00	1	0,0E+00	1	0,0E+00	1
CapA	3,25042	3	0,52596	2	0,0E+00	1	0,0E+00	1
CapB	1,93893	4	0,48063	3	0,07942	1	0,13280	2
CapC	1,22235	4	0,11914	3	0,02169	1	0,03210	2

Farklı UFLP veri setleri için BinSSA:

BinSSA, OR-Lib'den alınan düşük ölçekli, orta ölçekli ve büyük ölçekli yirmi M* problemleri üzerinde test edilmiştir (Beasley, 1990). M* problemleri UFLP’nin farklı bir veri kümesidir. M* problemlerinin optimal çözümün maliyeti çizelge 4.65’de gösterilmiştir. Çizelge 4.66’da BinSSA’nın M* veri kümesinin çözümü için 100 çalışmadan elde edilen en iyi, ortalama, en kötü ve gap değerleri gösterilmiştir.

M* problemleri boyutuna göre üç gruba ayrılmaktadır.

- *Grup-I:* Düşük ölçekli boyutlar (MO1-MO5)
- *Grup-II:* Orta ölçekli boyutlar (MP1-MP5 & MQ1-MQ5)
- *Grup-III:* Geniş ölçekli boyutlar (MR1-MR5)

Çizelge 4.65. M* veri seti için optimum sonuçlar

Problem ismi	Problem boyutu	Optimum çözüm maliyeti
MO1	100 x 100	1305,9500
MO2	100 x 100	1432,3600
MO3	100 x 100	1516,7700
MO4	100 x 100	1442,2400
MO5	100 x 100	1408,7700
MP1	200 x 200	2686,4800
MP2	200 x 200	2904,8600
MP3	200 x 200	2623,7100
MP4	200 x 200	2938,7500
MP5	200 x 200	2932,3300
MQ1	300 x 300	4091,0100
MQ2	300 x 300	4028,3300
MQ3	300 x 300	4275,4300
MQ4	300 x 300	4235,1500
MQ5	300 x 300	4080,7400
MR1	500 x 500	2608,1500
MR2	500 x 500	2654,7300
MR3	500 x 500	2788,2500
MR4	500 x 500	2756,0400
MR5	500 x 500	2505,0500

Çizelge 4.66. M* problemleri üzerinde BinSSA'nın test sonuçları

Problem	En iyi	En kötü	Ortalama	Gap
MO1	1305,9500	1305,9500	1305,9500	0,0000
MO2	1432,3600	1432,3600	1432,3600	0,0000
MO3	1516,7700	1516,7700	1516,7700	0,0000
MO4	1442,2400	1442,2400	1442,2400	0,0000
MO5	1408,7700	1408,7700	1408,7700	0,0000
MP1	2686,4800	2688,2800	2687,6600	0,0440
MP2	2904,8600	2904,8600	2904,8600	0,0000
MP3	2623,7100	2625,1100	2624,0000	0,0111
MP4	2939,5000	2942,1000	2940,5000	0,0600
MP5	2932,4500	2934,2500	2932,5000	0,0058
MQ1	4091,0100	4091,0100	4091,0100	0,0000
MQ2	4028,3300	4028,3300	4028,3300	0,0000
MQ3	4275,4300	4275,4300	4275,4300	0,0000
MQ4	4235,3500	4236,5500	4236,4600	0,0310
MQ5	4083,1400	4086,5400	4086,4500	0,1400
MR1	2609,2500	2610,7500	2610,2400	0,0800
MR2	2654,9900	2655,9800	2655,7300	0,0380
MR3	2789,1500	2791,1500	2790,1400	0,0680
MR4	2756,0400	2756,0400	2756,0400	0,0000
MR5	2505,1500	2505,5500	2505,4000	0,0140

BinSSA'nın performansı, literatürün iyi bilinen sezgisel algoritmaları ile karşılaştırılmıştır. Bu algoritmalar: Yerel Arama (LS) (Cura, 2010), Tabu Arama'nın (TS1 ve TS3) iki farklı sürümü (Cura, 2010) ve Geliştirilmiş Dağılım Arama (ISS) (Hakli ve Ortacay, 2019)'dır. Tüm deneyler, adil bir karşılaştırma için aynı parametrelerde ve koşullarda yapılmıştır. Bütün yöntemlerde çalışma zamanı değeri 100 olarak alınmıştır. LS sonuçları direk Cura'dan ve ISS sonuçları Hakli ve Ortacay'dan alınmıştır (Cura, 2010; Hakli ve Ortacay, 2019). Çizelge 4.67'de M* problemlerinde

BinSSA, ISS ve LS'nin karşılaştırma sonuçları gösterilmiştir. Çizelge 4.68'de M* problemlerinde BinSSA, TS1 ve TS3 karşılaştırma sonuçları gösterilmiştir.

Çizelge 4.67. BinSSA ve LS algoritmalarının M* problemleri için karşılaştırma sonuçları

Problem	BinSSA		LS		ISS	
	Ortalama	Gap	Ortalama	Gap	Ortalama	Gap
MO1	1305,9500	0,0000	1305,9500	0,0000	1305,9500	0,0000
MO2	1432,3600	0,0000	1432,7000	0,0090	1432,3600	0,0000
MO3	1516,7700	0,0000	1520,2700	0,2300	1516,7700	0,0000
MO4	1442,2400	0,0000	1442,2400	0,0000	1442,2400	0,0000
MO5	1408,7700	0,0000	1409,1700	0,0290	1408,7700	0,0000
MP1	2687,6600	0,0440	2688,5000	0,0750	2686,6600	0,0060
MP2	2904,8600	0,0000	2904,8600	0,0000	2904,8500	0,0000
MP3	2624,0000	0,0111	2624,7700	0,0400	2624,3400	0,0240
MP4	2940,5000	0,0600	2939,5300	0,0260	2940,8000	0,0690
MP5	2932,5000	0,0058	2933,4600	0,0380	2932,6000	0,0090
MQ1	4091,0100	0,0000	4091,0100	0,0000	4091,0100	0,0000
MQ2	4028,3300	0,0000	4028,3300	0,0000	4030,0800	0,0430
MQ3	4275,4300	0,0000	4275,4300	0,0000	4275,4300	0,0000
MQ4	4236,4600	0,0310	4235,4700	0,0070	4236,4600	0,0310
MQ5	4086,4500	0,1400	4086,5300	0,1410	4095,4600	0,3600
MR1	2610,2400	0,0800	2608,2400	0,0030	2647,0300	1,4900
MR2	2655,7300	0,0380	2654,7300	0,0030	2691,5400	1,3860
MR3	2790,1400	0,0680	2789,0400	0,0280	2832,3300	1,5810
MR4	2756,0400	0,0000	2756,0400	0,0000	2807,9000	1,8810
MR5	2505,4000	0,0140	2505,4800	0,0170	2549,9700	1,7930

Çizelge 4.68. BinSSA, TS1 ve TS3 algoritmalarının M* problemleri için karşılaştırma sonuçları

Problem	BinSSA	TS1	TS3
	Gap	Gap	Gap
MO1	0,0000	0,0030	0,0020
MO2	0,0000	0,0000	0,0000
MO3	0,0000	0,0018	0,0000
MO4	0,0000	0,0000	0,0040
MO5	0,0000	0,0000	0,0000
MP1	0,0440	0,0000	0,0000
MP2	0,0000	0,0000	0,0000
MP3	0,0111	0,0000	0,0000
MP4	0,0600	0,0000	0,0000
MP5	0,0058	0,0000	0,0000
MQ1	0,0000	0,0000	0,0000
MQ2	0,0000	0,0000	0,0020
MQ3	0,0000	0,0000	0,0000
MQ4	0,0310	0,0000	0,0000
MQ5	0,1400	0,0000	0,0000
MR1	0,0800	0,0000	0,0000
MR2	0,0380	0,0000	0,0000
MR3	0,0680	0,0000	0,0000
MR4	0,0000	0,0000	0,0000
MR5	0,0140	0,0000	0,0040

Çizelge 4.67'de BinSSA nın, M* problemlerinin % 70'inde başarı gösterdiği görülmektedir (20 M* veri kümesinden 14'ü için). LS, M* problemlerinin % 55'inde başarı göstermiştir (20 M* veri kümesinden 11'i için). ISS, M* problemlerinin %

45'inde başarı göstermiştir (20 M* veri kümesinden 9'u için). BinSSA, MO, MP ve MQ problemlerinde başarısı daha baskındır, LS yöntemi MQ ve MR problemlerinde başarısı daha fazladır ve ISS yöntemi MO problemlerinde başarısı daha baskındır. Dice'ın benzerlik ölçümü ve xor maskesi teknikleri sayesinde BinSSA, LS ve ISS'yi geçmiştir. Bu teknikler BinSSA'nın ikili arama uzayında yerel arama ve küresel arama yeteneğini geliştirmiştir.

Çizelge 4.68'de BinSSA'nın, M* problemlerinin % 50'sinde (20 M* veri kümesinin 10'u için) başarı gösterdiği görülmektedir. TS1, M* problemlerinin % 90'ında başarı gösterir (20 M* veri kümesinden 18'i için) ve TS3, M* problemlerinin yaklaşık % 80'inde (20 M* veri kümesinden 16'sı için) başarı göstermiştir. TS1 ve TS3, BinSSA'dan daha yüksek başarı göstermiştir.

CEC2015 için BinSSA:

BinSSA'nın, CEC2015 kıyaslama fonksiyonlarında performansı test edilmiştir. CEC2015'de çalışılmış algoritmalar: SabDE, BQIGSA, GBABC, BHTPSO-QI, BLDE ve SBHS'dir. Bu algoritmaların deneysel sonuçları doğrudan Banitalebi ve arkadaşlarından alınmıştır (Banitalebi ve ark., 2016). Adil bir karşılaştırma için karşılaştırma algoritmalarının kontrol parametreleri Banitalebi ve arkadaşlarından alınmıştır (Banitalebi ve ark., 2016). CEC2015'te kullanılan parametreler çizelge 4.69'da gösterilmiştir. BinSSA'nın diğer kontrol parametreleri $r_a = 1$, $p_c = 0.7$ ve $p_m = 0.1$ olarak seçilmiştir.

Çizelge 4.69. CEC2015 benchmark problemleri için parametreler (Banitalebi ve ark., 2016)

Parametreler	Tanım	Değerler
N	Popülasyon miktarı	50
n	Karar değişkenleri	10
m	Her boyut için bit miktarı	50
n×m	Aday çözümlerin boyutu	500
MaxFEs	Uygunluk fonksiyonu değerlendirme sayısı	n×10000
UB	Üst sınır	100
LB	Alt sınır	-100

BinSSA ve Banitalebi ve arkadaşlarının verdiği diğer algoritmalar ikili optimizasyon algoritmalarıdır. Ancak, bu çalışmayla ilgili kıyaslama problemlerinin karar değişkenleri sürekli değerlerdir. Bu nedenle, aday çözümün her boyutu 50 bit ile temsil edilir ve her aday çözüm için toplam 500 bit kullanılır. Her aday çözüm ikili

değerlerden oluştuğu için, aday çözümlerin maliyeti hesaplanmadan önce sürekli değerlere dönüştürülmelidir. Bu dönüştürme işlemi aşağıdaki gibidir:

$$C_i = LB_i + \frac{(UB_i - LB_i) \times DecVal_i}{2^m - 1} \quad (4.1)$$

Burada C_i , (C) sayısal vektörünün i inci boyutu için sürekli bir değeridir ve $DecVal_i$ sayısal vektörünün i inci boyutu için m-boyutlu ikili vektörün bir ondalık değeri ($DecVal$)'dir. UB_i i inci boyutun üst sınırı ve LB_i i inci boyutun alt sınırıdır. Karşılaştırma sonuçlar çizelge 4.70'de gösterilmiştir.

Çizelge 4.70. BinSSA ve sanatsal algoritmaların CEC2015 üzerinde karşılaştırma sonuçları

F	Ölçü	SBHS	BLDE	BHTPSO-QI	GBABC	BQIGSA	SabDE	BinSSA
F1	Ortalama	3,700E+08	1,087E+11	7,510E+08	2,729E+07	8,419E+07	3,093E+08	1,238E+08
	Std.Dev.	2,271E+08	2,351E+11	2,203E+09	2,267E+07	7,354E+07	1,168E+08	2,588E+07
	Rank	5	7	6	1	2	4	3
F2	Ortalama	4,493E+09	4,747E+09	5,132E+09	2,864E+09	7,834E+09	2,541E+09	1,968E+06
	Std.Dev.	1,790E+06	3,209E+08	4,685E+08	2,374E+09	6,527E+09	5,008E+09	2,024E+07
	Rank	4	5	6	3	7	2	1
F3	Ortalama	3,203E+02	3,201E+02	3,203E+02	3,202E+02	3,202E+02	3,200E+02	3,200E+02
	Std.Dev.	7,013E-02	1,093E-02	7,426E-02	2,641E+02	2,641E+02	2,044E-02	5,244E+00
	Rank	4	2	4	3	3	1	1
F4	Ortalama	4,478E+02	4,378E+02	4,478E+02	4,358E+02	4,389E+02	4,116E+02	5,212E+02
	Std.Dev.	5,171E+00	7,754E+00	1,122E+01	3,599E+02	3,621E+02	7,606E+00	5,472E+01
	Rank	5	3	5	2	4	1	6
F5	Ortalama	2,592E+03	1,934E+03	1,804E+03	1,108E+03	1,542E+03	9,330E+02	6,224E+02
	Std.Dev.	1,959E+02	3,088E+02	3,926E+02	9,736E+02	1,275E+03	9,464E+01	7,542E+01
	Rank	7	6	5	3	4	2	1
F6	Ortalama	5,312E+04	2,484E+06	6,799E+06	7,442E+06	5,582E+05	4,625E+04	3,245E+05
	Std.Dev.	1,512E+04	7,369E+06	1,186E+07	1,321E+07	6,055E+05	4,076E+04	5,214E+05
	Rank	2	5	6	7	4	1	3
F7	Ortalama	8,368E+02	8,472E+02	8,510E+02	7,589E+02	7,392E+02	7,752E+02	7,550E+02
	Std.Dev.	7,981E+00	1,289E+01	1,694E+01	4,668E+02	4,324E+02	4,155E+03	4,254E+02
	Rank	5	6	7	3	1	4	2
F8	Ortalama	3,344E+08	7,104E+09	5,407E+07	3,949E+07	2,948E+06	2,395E+07	5,475E+04
	Std.Dev.	1,810E+09	5,651E+09	8,312E+07	2,442E+08	2,469E+06	5,432E+07	3,124E+03
	Rank	6	7	5	4	2	3	1
F9	Ortalama	1,189E+03	1,190E+03	1,209E+03	1,017E+03	1,048E+03	1,177E+03	1,012E+03
	Std.Dev.	5,210E+00	1,736E+01	2,050E+01	8,397E+02	8,643E+02	4,102E+01	9,24E+02
	Rank	5	6	7	2	3	4	1
F10	Ortalama	9,038E+08	1,326E+10	1,320E+09	9,909E+05	4,426E+04	2,416E+07	5,410E+06
	Std.Dev.	6,596E+08	1,724E+10	2,073E+09	3,659E+06	4,477E+04	8,862E+07	7,243E+06
	Rank	5	7	6	2	1	4	3
F11	Ortalama	1,737E+03	1,599E+03	1,632E+03	1,159E+03	1,172E+03	1,114E+03	1,124E+03
	Std.Dev.	1,661E+01	3,313E+01	3,371E+01	9,557E+02	9,669E+02	1,131E+01	2,145E+02
	Rank	7	5	6	3	4	1	2
F12	Ortalama	1,445E+03	1,440E+03	1,437E+03	1,264E+03	1,255E+03	1,224E+03	1,234E+03
	Std.Dev.	5,683E-01	3,247E+00	7,552E+00	1,044E+03	1,035E+03	1,302E+00	1,042E+02
	Rank	7	6	5	4	3	1	2
F13	Ortalama	1,464E+10	2,882E+10	5,365E+08	1,446E+03	1,452E+03	2,815E+09	1,551E+03
	Std.Dev.	9,749E+09	2,073E+10	2,294E+09	1,193E+03	1,197E+03	4,158E+09	1,214E+03
	Rank	6	7	4	1	2	5	3
F14	Ortalama	2,736E+03	4,267E+03	5,005E+03	2,162E+03	3,356E+03	1,727E+03	1,690E+03
	Std.Dev.	2,220E+02	2,569E+03	2,703E+03	2,091E+03	2,869E+03	4,411E+02	5,412E+03
	Rank	4	6	7	3	5	2	1
F15	Ortalama	1,700E+03	1,700E+03	1,734E+03	2,012E+03	1,530E+03	1,700E+03	1,700E+03
	Std.Dev.	1,552E-04	2,622E-05	1,851E+02	1,659E+03	1,262E+03	2,177E-05	2,142E-03
	Rank	2	2	3	4	1	2	2

Çizelge 4.70'e göre BinSSA, en gelişmiş ikili optimizasyon algoritmalarından daha iyi sonuçlar vermektedir. Sonuçlara göre, BinSSA ikili optimizasyonda etkili bir optimize edici olduğunu göstermiştir.

4.3.3. Rüzgâr türbini yerleşim problemi için BinSSA

Bu bölümde bir BOP olarak BinSSA ile rüzgâr türbini yerleşim problemi çözülmüştür. Bu tezde, mümkün olan en yüksek rüzgâr enerjisini elde etmek için rüzgâr türbinlerinin bir rüzgâr çiftliğine optimum şekilde yerleştirilmesi incelemiştir. Sürekli arama uzayının ikili arama uzayına dönüşümünde çizelge 3.1'de yer alan transfer fonksiyonları kullanılmıştır. Dört adet S-şekilli, dört adet V-şekilli ve iki adet mod tabanlı transfer fonksiyonuna göre on adet BinSSA varyasyonu oluşturulmuştur (BinSSA1, BinSSA2, BinSSA3, BinSSA4, BinSSA5, BinSSA6, BinSSA7, BinSSA8, BinSSA9 ve BinSSA10). Aday çözüm üretimi için bir *lojik kapı operatörü* eklenmiş ve böylece BinSSA'da kâşif ve sömürü arasındaki denge kurulmuştur (BinSSA1-LG, BinSSA2-LG, BinSSA3-LG, BinSSA4-LG, BinSSA5-LG, BinSSA6-LG, BinSSA7-LG, BinSSA8-LG, BinSSA9-LG ve BinSSA10-LG). Bu çalışmada, rüzgâr türbinleri iki farklı ızgara yapısına (10x10 ve 20x20) göre yerleştirilmiştir. En iyi BinSSA varyasyonu, 10x10 ve 20x20 ızgara için uygunluk değerlerine (Maliyet / P_T) göre belirlenmiştir. En iyi BinSSA varyasyonu literatürdeki algoritmalar ile karşılaştırılmıştır. BinSSA'nın performansı analiz edilip tartışılmıştır. Her uygulama 20 bağımsız çalışma şeklinde çalıştırılmıştır. Adil karşılaştırmalar yapmak için, tüm algoritmalar aynı programlama dilinde, benzer bir platformda ve benzer parametre değerlerinde dikkatlice çalıştırılmıştır. Parametre ayarları çizelge 4.71'de gösterilmiştir. BinSSA karşılaştırmalarında kullanılan ölçme kriterleri çizelge 4.72'de gösterilmiştir.

Çizelge 4.71. Rüzgâr türbini yerleşim probleminde BinSSA için parametre ayarları

Parametreler	Değerler
Popülasyon miktarı	50
MaxFEs	6,00e+05
Bağımsız çalışma sayısı	20
Problem boyutu	Rüzgâr türbini sayısı (100)
Hub yüksekliği(z)	60m
Rotor yarıçapı (r_r)	40m
İtme katsayısı (C_T)	0,88
Arazi pürüzlülüğü (z_0)	0,3
Diğer parametreler	$r_a=1$; $p_c=0,7$; $p_m=0,1$ (James ve Li, 2015a)

Çizelge 4.72. BinSSA karşılaştırmaları için ölçme kriterleri

Ölçme kriterleri	Açıklama
Uygunluk fonksiyonu (Maliyet/ P_T)	Bir rüzgâr çiftliğinde minimum yatırım maliyeti (minimum rüzgâr türbini kurulumu) ile maksimum rüzgâr santrali gücü oranlamasından elde edilir.
Toplam güç (P_T)	Denklem 3.44'den elde edilmiştir.
Rüzgâr türbin sayısı	Yerleştirilen rüzgâr türbinlerinin ortalama sayısıdır.

BinSSA'nın farklı varyasyonları arasında karşılaştırma:

Bu alt bölümde, BinSSA'nın varyasyonları (S-şekilli, V-şekilli ve mod temelli) rüzgâr türbinlerinin rüzgâr çiftliği yerleştirilmesi probleminde farklı kriterlerde (uygunluk fonksiyonu (Maliyet / P_T), toplam güç (P_T) ve yerleşen rüzgâr türbinlerinin sayısı) analizleri yapılmıştır. Rüzgâr türbinleri iki farklı ızgara yapısına göre yerleştirilmiştir (10x10 ve 20x20). Çizelge 4.73'de 10x10 ızgara yapısı için BinSSA sonuçları çizelge 4.74'de 20x20 ızgara yapısı için BinSSA sonuçları gösterilmiştir. Üstün performans gösteren sonuçlar kalın font ile işaretlenmiştir.

Uygunluk fonksiyonu sonuçlarının ortalamasına göre, mod temelli ve S-şekilli BinSSA varyasyonları V-şekilli BinSSA varyasyonlarından daha yüksek bir başarı göstermiştir ve mod temelli BinSSA varyasyonları S-şekilli BinSSA varyasyonlarından daha yüksek bir başarı göstermiştir. Uygunluk fonksiyonu sonuçlarının ortalamasına göre, BinSSA10, 10x10 ızgara için 32 rüzgâr türbini ile yüksek başarı gösterirken, BinSSA9 20x20 ızgara için 58 rüzgâr türbini ile yüksek başarı göstermiştir. Rüzgâr türbinlerinin sayısına göre, BinSSA10 ve BinSSA2 varyasyonları 10x10 ızgara için yüksek başarı gösterirken BinSSA1 ve BinSSA9 varyasyonları 20x20 ızgara için yüksek başarı göstermiştir.

Çizelge 4.73. Rüzgâr türbinlerinin yerleştirilmesi problemi için BinSSA varyasyonları (10x10 ızgara)

Varyasyonlar	Uygunluk değeri (Maliyet/ P_T)	Toplam güç (P_T)	Rüzgâr türbini sayısı
BinSSA1	0,0015782	15086	33
BinSSA2	0,0015772	14809	32
BinSSA3	0,0016230	14739	33
BinSSA4	0,0016410	15257	35
BinSSA5	0,0016281	15094	34
BinSSA6	0,0016363	15252	35
BinSSA7	0,0019768	14923	34
BinSSA8	0,0016527	14949	35
BinSSA9	0,0015586	15250	33
BinSSA10	0,0015547	15048	32

Çizelge 4.74. Rüzgâr türbinlerinin yerleştirilmesi problemi için BinSSA varyasyonları (20x20 ızgara)

Varyasyonlar	Uygunluk değeri (Maliyet/ P_T)	Toplam güç (P_T)	Rüzgâr türbini sayısı
BinSSA1	0,0015133	25588	58
BinSSA2	0,0015350	27727	63
BinSSA3	0,0015817	27617	65
BinSSA4	0,0015885	27081	64
BinSSA5	0,0015924	27742	66
BinSSA6	0,0015756	27400	65
BinSSA7	0,0015879	28688	68
BinSSA8	0,0015943	27360	65
BinSSA9	0,0014863	26050	58
BinSSA10	0,0015123	26036	59

BinSSA-LG'nin farklı varyasyonları arasında karşılaştırma:

BinSSA'nın keşfetme ve sömürü yeteneğini geliştirmek için bir lojik kapı operatörü BinSSA'ya eklenmiştir ve BinSSA-LG adı verilmiştir. Böylece, BinSSA'nın arama uzayındaki yerel ve küresel arama kapasitesi arttırılmıştır. BinSSA-LG'nin varyasyonları (S-şekilli, V-şekilli ve mod temelli) rüzgâr türbinlerinin rüzgâr çiftliği yerleştirilmesi probleminde farklı kriterlerde (uygunluk fonksiyonu (Maliyet / P_T), toplam güç (P_T) ve yerleşen rüzgâr türbinlerinin sayısı) analizleri yapılmıştır. Rüzgâr türbinleri iki farklı ızgara yapısına göre yerleştirilmiştir (10x10 ve 20x20). Çizelge 4.75'de 10x10 ızgara yapısı için BinSSA-LG sonuçları ve çizelge 4.76'da 20x20 ızgara yapısı için BinSSA-LG sonuçları gösterilmiştir. Üstün performans gösteren sonuçlar kalın font ile işaretlenmiştir.

Çizelge 4.75. Rüzgâr türbinlerinin yerleştirilmesi problemi için BinSSA-LG varyasyonları (10x10 ızgara)

Varyasyonlar	Uygunluk değeri (Maliyet/ P_T)	Toplam güç (P_T)	Rüzgâr türbini sayısı
BinSSA1-LG	0,0015628	15474	34
BinSSA2-LG	0,0015632	14796	32
BinSSA3-LG	0,0015951	14500	32
BinSSA4-LG	0,0016228	15564	36
BinSSA5-LG	0,0015972	15140	34
BinSSA6-LG	0,0016227	14903	34
BinSSA7-LG	0,0016173	15617	36
BinSSA8-LG	0,0016198	15259	35
BinSSA9-LG	0,0015404	14671	32
BinSSA10-LG	0,0015519	15242	33

Çizelge 4.76. Rüzgâr türbinlerinin yerleştirilmesi problemi için BinSSA-LG varyasyonları (20x20 ızgara)

Varyasyonlar	Uygunluk değeri (Maliyet/Pr)	Toplam güç (Pr)	Rüzgâr türbini sayısı
BinSSA1-LG	0,0015024	24902	56
BinSSA2-LG	0,0015194	26350	60
BinSSA3-LG	0,0015523	27321	64
BinSSA4-LG	0,0015452	27270	64
BinSSA5-LG	0,0015398	27969	65
BinSSA6-LG	0,0015612	27824	64
BinSSA7-LG	0,0015602	28749	69
BinSSA8-LG	0,0015587	26120	63
BinSSA9-LG	0,0014045	23423	50
BinSSA10-LG	0,0014702	25448	56

Uygunluk fonksiyonu sonuçlarına göre, mod temelli ve S-şekilli BinSSA-LG varyasyonları V-şekilli BinSSA-LG varyasyonlarından daha yüksek bir başarı göstermişken mod temelli BinSSA-LG varyasyonları S-şekilli BinSSA-LG varyasyonlarından daha yüksek bir başarı göstermiştir. Uygunluk fonksiyonu sonuçlarının ortalamasına göre, BinSSA9-LG, 10x10 ızgara için 32 rüzgâr türbini ile yüksek başarı gösterirken, BinSSA9-LG, 20x20 ızgara için 50 rüzgâr türbini ile yüksek başarı göstermiştir. Rüzgâr türbinlerinin sayısına göre, BinSSA9-LG ve BinSSA2-LG varyasyonları 10x10 ızgara için, BinSSA9-LG varyasyonları 20x20 ızgara için yüksek başarı göstermiştir.

BinSSA ve BinSSA-LG'nin farklı varyasyonları arasında karşılaştırma:

Bu alt bölümde BinSSA ve BinSSA-LG sonuçları karşılaştırılmış ve lojik kapısı operatörünün BinSSA'nın performansı üzerindeki etkisi incelenmiştir. Çizelge 4.77'de rüzgâr türbinlerinin yerleştirilmesi problemi için BinSSA ve BinSSA-LG varyasyonlarının 10x10 ızgara yapısı için karşılaştırma sonuçları gösterilmiştir. Çizelge 4.78'de rüzgâr türbinlerinin yerleştirilmesi problemi için BinSSA ve BinSSA-LG varyasyonlarının 20x20 ızgara yapısı için karşılaştırma sonuçları gösterilmiştir. Çizelgeler 4.77 ve 4.78'e göre en başarılı BinSSA varyasyonu BinSSA9-LG olarak belirlenmiştir. Sonuçlar lojik kapı operatörünün BinSSA performansını artırdığını göstermiştir. BinSSA'nın arama uzayında arama başarısı artmıştır. Bu operatör, arama uzayında yerel optimum noktalar yerine global optimum noktaların bulunmasını sağlamıştır.

Çizelge 4.77. Rüzgâr türbinlerinin yerleştirilmesi problemi için BinSSA ve BinSSA-LG varyasyonları karşılaştırma sonuçları (10x10 ızgara)

Varyasyonlar	Uygunluk değeri (Maliyet/Pr)	Toplam güç (Pr)	Rüzgâr türbini sayısı
BinSSA1	0,0015782	15086	33
BinSSA2	0,0015772	14809	32
BinSSA3	0,0016230	14739	33
BinSSA4	0,0016410	15257	35
BinSSA5	0,0016281	15094	34
BinSSA6	0,0016363	15252	35
BinSSA7	0,0019768	14923	34
BinSSA8	0,0016527	14949	35
BinSSA9	0,0015586	15250	33
BinSSA10	0,0015547	15048	32
BinSSA1-LG	0,0015628	15474	34
BinSSA2-LG	0,0015632	14796	32
BinSSA3-LG	0,0015951	14500	32
BinSSA4-LG	0,0016228	15564	36
BinSSA5-LG	0,0015972	15140	34
BinSSA6-LG	0,0016227	14903	34
BinSSA7-LG	0,0016173	15617	36
BinSSA8-LG	0,0016198	15259	35
BinSSA9-LG	0,0015404	14671	32
BinSSA10-LG	0,0015519	15242	33

Çizelge 4.78. Rüzgâr türbinlerinin yerleştirilmesi problemi için BinSSA ve BinSSA-LG varyasyonları karşılaştırma sonuçları (20x20 ızgara)

Varyasyonlar	Uygunluk değeri (Maliyet/Pr)	Toplam güç (Pr)	Rüzgâr türbini sayısı
BinSSA1	0,0015133	25588	58
BinSSA2	0,0015350	27727	63
BinSSA3	0,0015817	27617	65
BinSSA4	0,0015885	27081	64
BinSSA5	0,0015924	27742	66
BinSSA6	0,0015756	27400	65
BinSSA7	0,0015879	28688	68
BinSSA8	0,0015943	27360	65
BinSSA9	0,0014863	26050	58
BinSSA10	0,0015123	26036	59
BinSSA1-LG	0,0015024	24902	56
BinSSA2-LG	0,0015194	26350	60
BinSSA3-LG	0,0015523	27321	64
BinSSA4-LG	0,0015452	27270	64
BinSSA5-LG	0,0015398	27969	65
BinSSA6-LG	0,0015612	27824	64
BinSSA7-LG	0,0015602	28749	69
BinSSA8-LG	0,0015587	26120	63
BinSSA9-LG	0,0014045	23423	50
BinSSA10-LG	0,0014702	25448	56

BinSSA ve BinSAA-LG varyasyonları için Wilcoxon signed-rank testi:

BinSSA ve BinSSA-LG varyasyonları 10x10 ızgara ve 20x20 ızgara yapıları için popülasyon miktarı 50 ve MaxFes miktarı 6,00e+5 değerlerinde her varyasyon 20 kere bağımsız çalıştırılmış ve 20 örnekli veri setleri oluşturulmuştur. Wilcoxon signed-rank testi sonuçları çizelge 4.79'da gösterilmiştir.

En yüksek performanslı BinSSA varyasyonu uygunluk (fitness) fonksiyonu sonuçlarının ortalamasına göre seçilmiştir. Sonuçlar, BinSSA-LG varyasyonlarının arama ve sömürü arasında BinSSA varyasyonlarından daha başarılı bir denge kurduğunu göstermektedir. Mantık kapısı operatörü BinSSA'nın arama ve sömürü yeteneklerini geliştirmiştir. BinSSA-LG varyantları arasında BinSSA9-LG, 10x10 ızgara yapısı için en performanslı olanıdır ve BinSSA9-LG, 20x20 ızgara yapısı için en performanslı olanıdır. Çizelge 4.79'a göre, BinSSA9-LG varyasyonu uygunluk fonksiyonu değerlerinin ortalaması için üstün performans göstermektedir.

Wilcoxon signed-rank testi, hem 10x10 ızgara yapısı hem de 20x20 ızgara yapısı için BinSSA9-LG varyasyonu ile BinSSA ve BinSSA-LG'nin diğer varyasyonlarına uygulanmıştır. Elde edilen sonuçlara göre, p -değerleri çoğu vaka için 0,05'in altındadır. Bu sonuç BinSSA9-LG ile diğer varyantlar arasında anlamsal bir fark olduğunu göstermektedir. Bu nedenle, BinSSA9-LG'nin sonuçlarındaki iyileşmeler istatistiksel olarak diğer varyasyonlardan daha üstündür.

Çizelge 4.79. BinSSA9-LG sonuçları ile diğer BinSSA ve BinSSA-LG varyasyonları arasında uygulanmış Wilcoxon signed-rank testinin p değerleri ($p \leq 0.05$ sonuçlarının altı çizilmiştir.)

Varyasyonlar	10x10 ızgara BinSSA9-LG	Varyasyonlar	20x20 ızgara BinSSA9-LG
BinSSA1	<u>4,5168e-06</u>	BinSSA1	0,3366
BinSSA2	<u>2,5937e-05</u>	BinSSA2	8,9967e-08
BinSSA3	<u>5,0445e-04</u>	BinSSA3	<u>1,2130e-07</u>
BinSSA4	<u>1,5968e-05</u>	BinSSA4	<u>1,6308e-07</u>
BinSSA5	<u>1,5966e-05</u>	BinSSA5	8,9842e-08
BinSSA6	<u>1,4123e-05</u>	BinSSA6	<u>1,2147e-07</u>
BinSSA7	<u>1,1981e-06</u>	BinSSA7	<u>1,2114e-07</u>
BinSSA8	<u>6,7860e-08</u>	BinSSA8	<u>6,6626e-08</u>
BinSSA9	<u>1,2330e-07</u>	BinSSA9	<u>2,8961e-07</u>
BinSSA10	<u>1,9152e-06</u>	BinSSA10	<u>8,9671e-07</u>
BinSSA1-LG	<u>2,2091e-07</u>	BinSSA2-LG	<u>2,0166e-06</u>
BinSSA2-LG	<u>9,1601e-08</u>	BinSSA3-LG	<u>6,6063e-08</u>
BinSSA3-LG	<u>6,7860e-08</u>	BinSSA4-LG	<u>6,6065e-07</u>
BinSSA4-LG	<u>6,7860e-08</u>	BinSSA5-LG	<u>8,9967e-08</u>
BinSSA5-LG	<u>6,7850e-08</u>	BinSSA6-LG	<u>1,2114e-07</u>
BinSSA6-LG	<u>6,7760e-08</u>	BinSSA7-LG	8,9842e-08
BinSSA7-LG	<u>6,7650e-08</u>	BinSSA8-LG	8,9967e-08
BinSSA8-LG	<u>6,7560e-08</u>	BinSSA9-LG	<u>9,5806e-06</u>
BinSSA9-LG	<u>0,0071</u>	BinSSA10-LG	<u>0,0012</u>

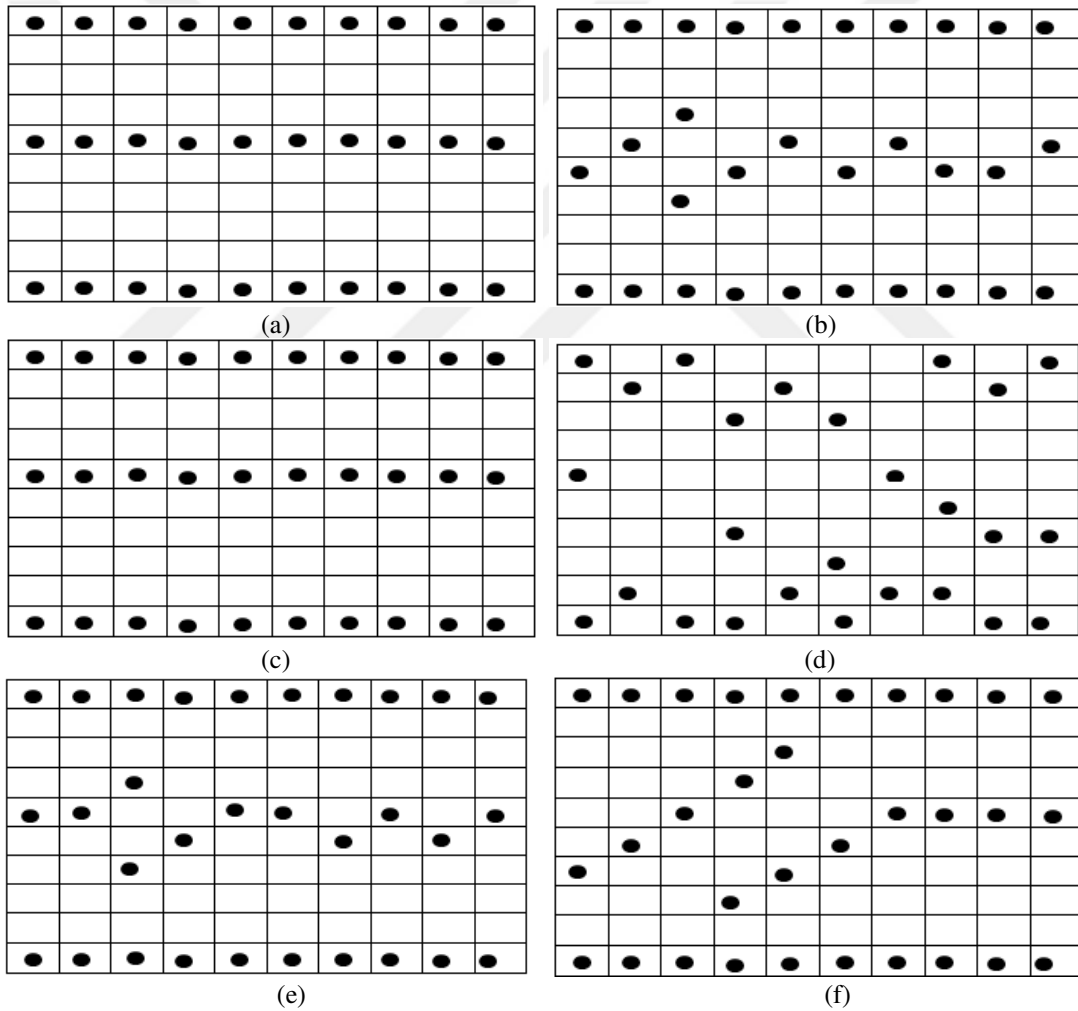
BinSSA ve BinSSA-LG varyasyonlarının literatürdeki diğer algoritmalar ile karşılaştırılması:

BinSSA-LG'nin performansı literatürün iyi bilinen sezgisel algoritmaları ile karşılaştırılmıştır. Bu algoritmalar: Genetik Algoritma (GA) (Mosetti ve ark., 1994; Grady ve ark., 2005; Parada ve ark., 2017), Lazy Greedy (Changshui ve ark., 2011), MILP (Shakoor ve ark., 2016), BPSO-TVAC (Pookpunt ve Ongsakul, 2013), İkili Yapay Alg Algoritması (BinAAA) (Beşkirli ve ark., 2018), İkili Yabani Ot Optimizasyonu (BIWO) (Beşkirli ve ark., 2019), Simüle Tavlama Algoritması (SA) (Yang ve Cho, 2019) ve BinSSA'dır. Tüm deneyler, adil bir karşılaştırma için aynı parametrelerde ve koşullarda yapılmıştır. Çizelge 4.80'de rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemi için BinSSA-LG sonuçları gösterilmiştir (10x10 ızgara). Şekil 4.11'de 10x10 ızgara için Parada ve Grady, BinAAA, (Yang ve Cho), Mosetti ve ark., BIWO ve BinSSA-LG'nin rüzgâr çiftliği yapılandırması gösterilmiştir.

Uygunluk fonksiyonu sonuçlarına göre, en başarılı algoritma 32 rüzgâr türbinine sahip BinSSA-LG'dir. 10x10 ızgara için BinSSA-LG'nin uygunluk değeri, Grady ve Parada'nın uygunluk değerinden % 0,21 daha başarılıdır ve BinSSA-LG'nin toplam gücü % 2,52 artırmıştır. Parada ve Grady, 10x10 ızgara için 30 türbin yerleştirmişlerdir ve her ikisi de aynı yerleştirme işlemini yapmışlardır. BinSSA-LG'deki rüzgâr türbinlerinin sayısı sadece iki rüzgâr türbinini artırmıştır. BinSSA-LG, 10x10 ızgara için farklı bir yerleşim yapılandırması kullanmıştır. Bu yapılandırmayla türbinler arasındaki uyanma etkisi en aza indirme amaçlanmıştır. BinSSA-LG'nin 10x10 ızgara için uygunluk değeri Beşkirli ve arkadaşlarından % 0,06 (BinAAA) ve % 0,058 (BIWO) daha başarılıdır. BinSSA-LG, BinAAA ve BIWO'a göre toplam gücü % 0,03 artırmıştır. BinAAA, BIWO ve BinSSA-LG'nin yerleşim yapılandırması 10x10 ızgara için benzerdir. BinSSA-LG'nin 10x10 ızgara için uygunluk değeri Yang ve Cho'nun uygunluk değerlerinden % 0,50 daha başarılıdır ve BinSSA-LG toplam gücü % 2,82 artırmıştır. BinSSA-LG'nin 10x10 ızgara için uygunluk değeri, Mosetti ve Poloni'nin uygunluk değerlerinden % 4,90 daha başarılıdır ve BinSSA-LG toplam gücü % 18,77 artırmıştır. BinSSA-LG'nin 10x10 ızgara için uygunluk değeri BinSSA'nın uygunluk değerlerinden % 0,92 daha başarılıdır ve BinSSA-LG toplam gücü % 2,51 azaltmıştır. Hem BinSSA hem de BinSSA-LG'de 10x10 ızgaraya yerleşen rüzgâr türbinlerinin sayısı 32'ye eşittir. BinSSA-LG için elde edilen sonuçlar literatürle karşılaştırıldığında, BinSSA-LG üstün başarı gösterdiği gözlemlenmiştir.

Çizelge 4.80. BinSSA-LG ve karşılaştırma algoritmalarının rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemi için karşılaştırma sonuçları (10x10 ızgara)

Algoritmalar	Uygunluk değerleri (Maliyet/ P_T)	Toplam güç (P_T)	Rüzgâr türbini sayısı
GA (Mosetti ve ark., 1994)	0,0016197	12352	26
GA (Grady ve ark., 2005)	0,0015436	14310	30
MILP (Shakoor ve ark., 2016)	0,0015436	14310	30
BPSO-TVAC (Pookpunt ve Ongsakul, 2013)	0,0015436	14310	30
Lazy Greedy (Changshui ve ark., 2011)	0,0015436	14310	30
GA (Parada ve ark., 2017)	0,0015436	14310	30
BinAAA (Beşkirli ve ark., 2018)	0,0015414	14667	31
BIWO (Beşkirli ve ark., 2019)	0,0015413	14667	31
SA (Yang ve Cho, 2019)	0,0015479	14269	30
BinSSA	0,0015547	15048	32
BinSSA-LG	0,0015404	14671	32



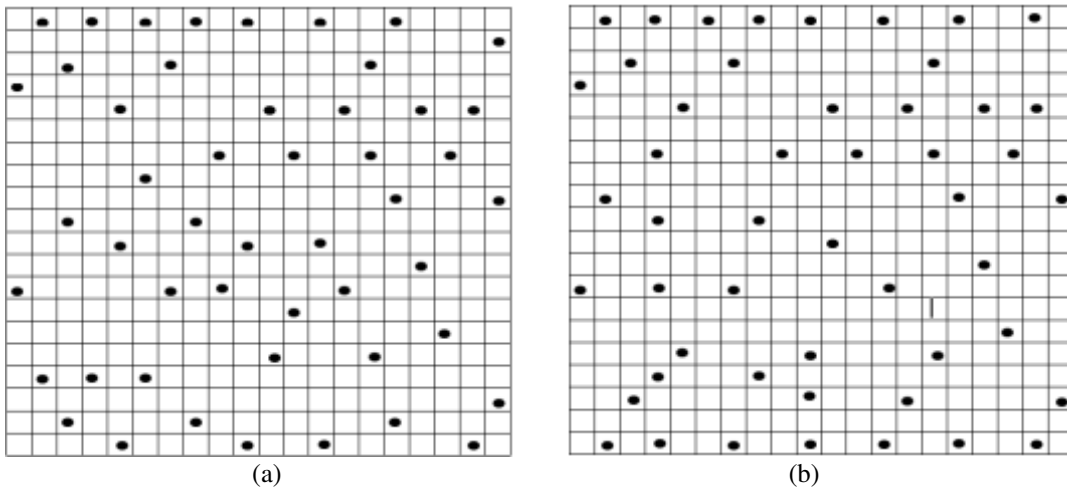
Şekil 4.11. 10x10 ızgara için rüzgâr çiftliğinin yapılandırması: (a) Parada ve Grady, (b) BinAAA, (c) Yang ve Cho, (d) Mosetti ve ark., (e) BIWO ve (f) BinSSA-LG

BinSSA-LG'nin performansı 20x20 ızgara için literatürün iyi bilinen sezgisel algoritmaları ile karşılaştırılmıştır. Bu algoritmalar İkili Yapay Alg Algoritması (BinAAA) (Beşkirli ve ark., 2018) ve BinSSA'dır. Tüm deneyler, adil bir karşılaştırma için aynı parametrelere ve koşullara göre yapılmıştır. Çizelge 4.81, rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemi için BinSSA-LG sonuçlarının karşılaştırmasını göstermektedir (20x20 ızgara). Şekil 4.12, 20x20 ızgara için BinAAA ve BinSSA-LG'nin rüzgâr çiftliği yapılandırmasını göstermektedir.

BinSSA-LG'nin 20x20 ızgara için uygunluk değeri Beşkirli ve arkadaşlarına göre % 0,07 artmıştır ve toplam güç % 0,04 artmıştır. Beşkirli ve ark. 20x20 ızgara için 49 türbin yerleştirmişlerdir ve önerilen algoritma 20x20 ızgara için 50 türbin yerleştirmiştir. BinAAA ve BinSSA-LG'nin yerleşim yapılandırması 20x20 ızgarası için benzerdir. BinSSA-LG'deki türbin sayısı yüksek olmasına rağmen, türbinler arasındaki mesafe dikkate alınarak türbinler daha verimli bir şekilde yerleştirilmiştir. BinSSA-LG'nin başarısı, mantık kapısı operatörü ile yerel ve küresel arama yeteneklerinin geliştirilmesinden kaynaklanmaktadır.

Çizelge 4.81. BinSSA-LG ve karşılaştırma algoritmalarının rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemi için karşılaştırma sonuçları (20x20 ızgara)

Algoritmalar	Uygunluk değerleri (Maliyet/ P_T)	Toplam güç (P_T)	Rüzgâr türbini sayısı
BinAAA (Beşkirli ve ark., 2018)	0,0014054	23422	49
BinSSA	0,0014863	26050	58
BinSSA-LG	0,0014045	23423	50



Şekil 4.12. 20x20 ızgara için rüzgâr çiftliğinin yapılandırması: (a) BinAAA ve (b) BinSSA-LG

4.3.4. Özellik Seçme (FS) problemi için BinSSA

Bu alt bölümde, çaprazlama operatörlü İkili Sosyal Örümcek Algoritmasının (BinSSA-CR) varyasyonları ve İkili Sosyal Örümcek Algoritmasının (BinSSA) varyasyonları özellik seçme problemi üzerinde dikkatle karşılaştırılmış ve farklı kriterlere göre performansları incelenmiştir. En iyi BinSSA varyasyonları farklı kriterlere göre tespit edilmiştir. BinSSA'nın en iyi varyasyonları ve BinSSA-CR'nin en iyi varyasyonları, literatürdeki yeni ve bilinen sezgisel algoritmalar ile karşılaştırılmıştır. BinSSA'nın performansı analiz edilmiş ve tartışılmıştır.

OR-Lib'den (UCI Machine Learning Repository) alınan düşük ölçekli, orta ölçekli ve büyük ölçekli yirmi bir UCI karşılaştırma veri seti, özellik seçme probleminde BinSSA'nın performansını ölçmek için kullanılmıştır. Özellik seçme problemi için çeşitli sayıda öznitelik (9 ila 856), çeşitli sınıf sayısı (2 ila 16) ve çeşitli örnek sayısı (32 ila 5000) içeren veri kümeleri kullanılmıştır. Bu çalışmada, algoritmanın performansını göstermek için büyük boyutlu veri setleride tercih edilmiştir. Yirmi bir UCI karşılaştırma veri seti çizelge 4.82'de gösterilmiştir.

Çizelge 4.82. BinSSA için UCI kıyaslama veri seti

Tür	ID	Veri seti	#Özellik	#Örnek	#Sınıf
Düşük ölçekli ≤19	N1	Wine	13	178	3
	N2	Hepatitis	19	155	2
	N3	Vehicle	18	846	4
	N4	Zoo	16	101	7
	N5	Heart	13	270	2
	N6	Breast Cancer	9	699	2
Orta ölçekli [22 90]	N7	Ionosphere	34	351	2
	N8	Lung Cancer	56	32	3
	N9	Dermatology	34	366	6
	N10	Sonar	60	208	2
	N11	BreastEW	30	569	2
	N12	Soybean Small	35	47	4
	N13	Movementlibras	90	360	15
	N14	Parkinsons	22	195	2
	N15	Spambase	57	4601	2
	N16	Waveform	40	5000	3
Geniş ölçekli ≥101	N17	Arrhythmia	279	452	16
	N18	Semeion	256	1593	10
	N19	Clean	166	476	2
	N20	CNAE	856	1080	9
	N21	Hillvalley	101	606	2

UCI karşılaştırma veri setleri boyuta göre üç gruba ayrılır.

- *Group-I*: Düşük ölçekli boyutlar (N1-N6)
- *Group-II*: Orta ölçekli boyutlar (N7-N16)
- *Group-III*: Geniş ölçekli boyutlar (N17-N21)

Sınıflandırıcı olarak K-NN veya SVM tercih edilmiştir. Veri kümesi iki sınıf içeriyorsa, SVM sınıflandırıcısı, veri kümesi ikiden fazla sınıf içeriyorsa, K-NN sınıflandırıcısı tercih edilmiştir. Veri setlerinin % 80'i eğitim için ve % 20'si test için kullanılmıştır. Optimal alt kümeyi üretmek için 5-NN sınıflandırıcısı (burada $K = 5$) tercih edilmiştir. Hem SVM hem de K-NN için, k-fold (burada $k = 10$) çapraz doğrulama tekniği veri kümesine uyarlanmıştır. Her uygulamanın 20 bağımsız çalışması vardır.

Adil karşılaştırmalar yapmak için, tüm algoritmalar aynı programlama dilinde, benzer bir platformda ve benzer parametre değerlerinde dikkatlice çalıştırılmıştır.

Popülasyon büyüklüğü, maksimum yineleme sayısı, uygunluk fonksiyonu, ϕ ve ρ parametre değerlerinin belirlenmesi:

Farklı popülasyon büyüklüğü ve maksimum iterasyon sayısı kombinasyonları, *the national classification of economic activities (CNAE)* (UCI Machine Learning Repository) veri kümesi üzerinde test edilmiştir. *CNAE* veri kümesinin seçilme nedeni, bu veri kümesinin değerlendirme için daha duyarlı olması ve daha fazla özelliğe sahip olmasıdır. Doğruluk oranı (accuracy), farklı popülasyon boyutu (N) kombinasyonları ve farklı transfer fonksiyonlarına sahip temel BinSSA'nın performansı üzerindeki maksimum yineleme sayısı için ölçülmüştür. Çizelge 4.83'de maksimum yineleme = {50, 75, 100}, popülasyon boyutu = {10, 25, 50}, $\rho = 0.99$ ve $\phi = 0.01$ parametre değerleri seçilmiştir. Sonuçlar çizelge 4.83'de gösterilmektedir. Sonuçlara göre, en uygun değerler popülasyon büyüklüğü için 50 ve maksimum yineleme sayısı için 100 olarak belirlenmiştir.

Çizelge 4.83. Farklı popülasyon büyüklüğü (N) ve maksimum yinleme sayısı kombinasyonlarına göre CNAE veri kümesi üzerinde ortalama doğruluk sonuçları

Popülasyon boyutu (N) / Maksimum yinleme	10			25			50		
	50	75	100	50	75	100	50	75	100
BinSSA1	86	88,52	89	88,75	89	90,45	89	90,5	91,75
BinSSA2	80	81	82	83,15	84,56	85	84	85	86,5
BinSSA3	77,45	77,56	78	79	80	81,56	82,5	84,5	85,75
BinSSA4	80	81,5	82	83	83,45	84,15	84,5	85	86,25
BinSSA5	77	77,25	78	77,41	78,25	79	78,5	79	79,25
BinSSA6	77	77,54	78,5	77,55	78,65	79,65	79	79,75	80,25
BinSSA7	76,98	77,25	78	77,27	78,25	79,05	79	79,5	79,75
BinSSA8	78	78,14	78,58	78,35	79,15	79,75	79,5	80	80,25

CNAE veri kümesi üzerinde ρ ve ϕ değerlerinin etkisi BinSSA ile test edilmiştir. ρ ve ϕ değerleri, uygunluk fonksiyonunun sonucunu hafif değiştirelebilen sabit parametrelerdir. ρ sınıflandırma doğruluğunun ağırlığını belirler, ϕ ise özellikler azaltma oranının ağırlığını belirler. Literatürde bu değerler özellik seçme problemlerinde genellikle rastgele belirlenir. ρ genellikle yüksek bir değer seçilir (yani $\rho \geq 0.9$) ve ϕ genellikle küçük bir değer olarak seçilir (yani $\phi \leq 0.5$). Doğruluk oranı, farklı transfer fonksiyonlarına sahip temel BinSSA'nın performansı üzerinde farklı ρ ve ϕ değerlerinin kombinasyonları için test edilmiştir. Çizelge 4.84'de popülasyon büyüklüğü = 50, maksimum yinleme = 100, $\rho = \{0.5, 0.7, 0.9, 0.99\}$ ve $\phi = \{0.5, 0.3, 0.1, 0.01\}$ parametre değerleri seçilmiştir. Sonuçlar çizelge 4.84'de gösterilmiştir. Genel olarak, ϕ değerini azalarak ve ρ değeri artarak doğruluk değerinde bir artış gözlemlenmiştir. Sonuçlara göre, en uygun değerler ρ için 0.99 ve ϕ için 0.01 olarak belirlenmiştir.

Çizelge 4.84. Ortalama doğruluk sonuçlarına göre, CNAE veri seti üzerinde ρ ve ϕ değerlerinin etkisi

ρ / ϕ	0.5/0.5 Ortalama	0.7/0.3 Ortalama	0.9/0.1 Ortalama	0.99/0.01 Ortalama
BinSSA1	86	86,15	89,25	91,75
BinSSA2	82	84,5	85,5	86,5
BinSSA3	79	82,25	83,5	85,75
BinSSA4	81	83,5	85	86,25
BinSSA5	77	77,5	78,5	79,25
BinSSA6	78	80	80,15	80,25
BinSSA7	78	79,25	79,5	79,75
BinSSA8	79	80	80,15	80,25

BinSSA ve BinSSA-CR'de kullanılan parametreler çizelge 4.85'de gösterilmiştir. BinSSA ve BinSSA-CR'nin her bir varyasyonu 20 bağımsız çalışma şeklinde test edilmiştir. BinSSA'nın karşılaştırılmasında kullanılan ölçütler; uygunluk

değerleri, sınıflandırma doğruluğu, seçilen özellik sayısı ve CPU süresidir. Bu ölçütler çizelge 4.86’da gösterilmiştir.

Çizelge 4.85. BinSSA ve BinSSA-CR varyasyonları için parametre ayarları

Parametreler	Değerler
Popülasyon büyüklüğü (N)	50
Maksimum yineleme	100
Bağımsız çalışma sayısı	20
Problem boyutu (n)	Veri seti içindeki öznitelik sayısı
ρ	0,99
ϕ	0,01
K	5 (K-NN içinde K değeri) (Zawbaa ve ark., 2015)
k	10 (k-fold içinde k değeri)
Diğer parametreler	$r_a=1$; $p_c=0,7$; $p_m=0,1$ (James ve Li, 2015a)

Çizelge 4.86. BinSSA kıyaslamalarında kullanılan karşılaştırma ölçütleri

Ölçme ölçütleri	Açıklama
Uygunluk değeri	Karşılaştırma veri kümelerinde seçilen özellikler kullanılarak uygunluk fonksiyonlarından elde edilir. (Uygunluk fonksiyonunun ortalaması, standart sapması, en iyi ve en kötü değerleridir.)
Sınıflandırma doğruluğu	Karşılaştırma veri kümelerinde seçilen özellikler kullanılarak sınıflandırıcıdan elde edilir.
Seçilen özellik sayısı	Seçilen özelliklerin ortalama sayısıdır.
CPU zamanı	Saniyeler cinsinden bir aday üretme zamanıdır.

BinSSA varyasyonlarının karşılaştırılması:

BinSSA'nın varyasyonları (S-şekilli ve V-şekilli), özellik seçme problemi üzerinde farklı ölçütlerde (Uygunluk fonksiyonunun ortalaması, uygunluk fonksiyonunun standart sapması, uygunluk fonksiyonunun en iyisi, uygunluk fonksiyonunun en kötüsü, doğruluk değerlerinin ortalaması, seçilen özellik sayısının ortalama sayısı ve CPU süresinin ortalaması) analiz edilmiştir. Özellik seçme problemi için yirmi bir karşılaştırma veri kümesi belirlenmiştir.

Çizelge 4.87’de BinSSA varyasyonlarının uygunluk değerlerinin ortalaması gösterilmiştir. En iyi değerler kalın olarak işaretlenmiştir. Uygunluk fonksiyonu sonuçlarının ortalamasına göre, V şeklindeki BinSSA varyasyonları S şeklindeki BinSSA varyasyonlarından daha yüksek bir başarı göstermiştir. BinSSA5, 21 veri kümesinden (N2, N7, N10, N12, N14, N17 ve N19) 7 veri kümesi için kıyaslama veri kümelerinin yaklaşık % 33’ünde yüksek başarı göstermiştir.

Çizelge 4.88’de BinSSA varyasyonlarının uygunluk değerlerinin standart sapması gösterilmiştir. En iyi değerler kalın olarak işaretlenmiştir. Uygunluk

değerlerinin standart sapmasına göre, BinSSA6, BinSSA7 ve BinSSA8 varyasyonları özellik seçme problemi için yüksek başarı göstermiştir. BinSSA6, BinSSA7 ve BinSSA8 varyasyonları, 21 veri kümesinden 5'i için kıyaslama veri kümelerinin % 23,81'inde başarı göstermektedir.

Çizelgeler 4.89 ve 4.90'da BinSSA varyasyonlarının uygunluk değerlerinin en iyi ve en kötü değerlerini göstermektedir. En iyi uygunluk sonuçlarına göre, BinSSA5, BinSSA6 ve BinSSA8 varyasyonları özellik seçme problemi için yüksek başarı göstermiştir. BinSSA5 ve BinSSA6 varyasyonları, kıyaslama veri kümelerinin % 38'inde başarı göstermiştir (21 veri kümesinden 8'i için). BinSSA8 varyasyonu, 21 veri kümesinden 9'u (N1, N4, N7, N9, N10, N12, N13, N14 ve N21) için kıyaslama veri kümelerinin % 42,86'sında üstün performans göstermektedir. En kötü uygunluk sonuçlarına göre, BinSSA5 varyasyonu, özellik seçme problemi için 21 veri kümesinden 7'si için minimum değere sahiptir.

Çizelge 4.91'de BinSSA varyasyonlarının ortalama doğruluk değerlerini göstermektedir. Doğruluk değerlerinin ortalamasına göre, BinSSA1, BinSSA4 ve BinSSA7 varyasyonları yüksek performans göstermektedir. BinSSA1, BinSSA4 ve BinSSA7 varyasyonları, 21 veri kümesinden 7'si için kıyaslama veri kümelerinin % 33,33'ünde başarı göstermektedir.

Çizelge 4.92'de BinSSA varyasyonlarının seçilen özelliklerinin ortalama sayısını göstermektedir. Seçilen özelliklerin ortalama sayısına göre, BinSSA (BinSSA4, BinSSA6 ve BinSSA8) varyasyonları, BinSSA'nın diğer varyasyonlarından daha az özellik seçmiştir. BinSSA4 varyasyonu, 21 veri kümesinden 16'sı (N1, N3, N4, N5, N6, N7, N8, N10, N11, N12, N13, N14, N16, N18, N20 ve N21) için minimum özellikler seçmiştir. BinSSA8 varyasyonu, 21 veri kümesinden 9'u için yüksek bir başarı göstermektedir.

Çizelge 4.93'de BinSSA varyasyonlarının CPU süresinin (saniye cinsinden) ortalamasını göstermektedir. CPU zamanı bir aday üretme zamanıdır (saniye cinsinden). Her bir yinelemede üretilen aday sayısı sabittir. Bu yüzden CPU zamanı BinSSA'da bir karşılaştırma ölçütü olarak hesaplanmıştır. CPU zaman sonuçlarının ortalamasına göre, BinSSA3 varyasyonu, 21 veri kümesinden (N7, N8, N9, N11, N14 ve N15) 6 tanesi için mümkün olan en kısa sürede sonuç elde etmiştir.

Şekil 4.13'de farklı BinSSA varyasyonları arasındaki karşılaştırmalar için yakınsama grafikleri gösterilmiştir. Elde edilen sonuçlara göre BinSSA5, BinSSA6 ve BinSSA7 varyasyonlarının daha başarılı olduğu göze çarpmaktadır.

Çizelge 4.87. Uygunluk değerlerinin ortalamasına göre BinSSA varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA5	BinSSA6	BinSSA7	BinSSA8
		Ortalama	Ortalama	Ortalama	Ortalama	Ortalama	Ortalama	Ortalama	Ortalama
N1	Wine	0,0138	0,0158	0,0114	0,0145	0,0256	0,0203	0,0177	0,0160
N2	Hepatitis	0,0952	0,1043	0,1056	0,0924	0,0821	0,0887	0,0986	0,0869
N3	Vehicle	0,2336	0,2458	0,2331	0,2322	0,2398	0,2392	0,2361	0,2498
N4	Zoo	0,0263	0,0085	0,0186	0,0210	0,0150	0,0248	0,0276	0,0200
N5	Heart	0,1255	0,1164	0,1170	0,1180	0,1304	0,1262	0,1328	0,1036
N6	Breast Cancer	0,0240	0,0260	0,0227	0,0205	0,0211	0,0186	0,0238	0,0248
N7	Ionosphere	0,0342	0,0317	0,0269	0,0352	0,0171	0,0207	0,0206	0,0268
N8	Lung Cancer	0,0714	0,0544	0,0375	0,0372	0,0252	0,0087	0,0005	0,0087
N9	Dermatology	0,0057	0,0063	0,0081	0,0090	0,0101	0,0100	0,0138	0,0148
N10	Sonar	0,0682	0,0500	0,0544	0,0407	0,0308	0,0359	0,0309	0,0458
N11	BreastEW	0,0423	0,0354	0,0427	0,0364	0,0345	0,0386	0,0318	0,0296
N12	Soybean Small	0,0035	0,0031	0,0026	0,0025	0,0005	0,0005	0,0005	0,0005
N13	Movementlibras	0,2227	0,2071	0,2148	0,1877	0,1689	0,1544	0,1689	0,1742
N14	Parkinsons	0,0034	0,0028	0,0023	0,0022	0,0005	0,0057	0,0057	0,0006
N15	Spambase	0,0666	0,0674	0,0708	0,0731	0,0764	0,0794	0,0872	0,0844
N16	Waveform	0,1606	0,1550	0,1644	0,1690	0,1722	0,1693	0,1770	0,1821
N17	Arrhythmia	0,3138	0,3417	0,3124	0,3198	0,2779	0,2931	0,2975	0,3039
N18	Semeion	0,7966	0,7978	0,7992	0,7970	0,7851	0,7827	0,7825	0,7872
N19	Clean	0,0390	0,0337	0,0371	0,0344	0,0268	0,0288	0,0301	0,0369
N20	CNAE	0,1088	0,1385	0,1419	0,1570	0,2084	0,2181	0,2182	0,2097
N21	Hillvalley	0,4428	0,4473	0,4370	0,4310	0,3914	0,3822	0,4019	0,3801

Çizelge 4.88. Uygunluk değerlerinin standart sapmasına (Std.Dev.) göre BinSSA varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA5	BinSSA6	BinSSA7	BinSSA8
		Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.
N1	Wine	0,0133	0,0141	0,0156	0,0186	0,0225	0,0204	0,0139	0,0142
N2	Hepatitis	0,0317	0,0434	0,0337	0,0302	0,0472	0,0401	0,0376	0,0435
N3	Vehicle	0,0215	0,0236	0,0223	0,0232	0,0227	0,0193	0,0201	0,0153
N4	Zoo	0,0464	0,0151	0,0224	0,0369	0,0272	0,0377	0,0299	0,0333
N5	Heart	0,0307	0,0312	0,0365	0,0380	0,0271	0,0237	0,0259	0,0372
N6	Breast Cancer	0,0095	0,0115	0,0135	0,0109	0,0102	0,0070	0,0084	0,0114
N7	Ionosphere	0,0169	0,0291	0,0190	0,0157	0,0120	0,0123	0,0173	0,0180
N8	Lung Cancer	0,0830	0,0942	0,0677	0,0862	0,0604	0,0369	0,0004	0,0369
N9	Dermatology	0,0005	0,0043	0,0060	0,0080	0,0081	0,0092	0,0105	0,0108
N10	Sonar	0,0306	0,0404	0,0299	0,0283	0,0260	0,0281	0,0284	0,0206
N11	BreastEW	0,0132	0,0210	0,0131	0,0121	0,0106	0,0122	0,0097	0,0126
N12	Soybean Small	0,0004	0,0003	0,0003	0,0003	0,0001	0,0001	0,0001	0,0001
N13	Movementlibras	0,0336	0,0408	0,0448	0,0528	0,0340	0,0333	0,0510	0,0392
N14	Parkinsons	0,0006	0,0004	0,0005	0,0003	0,0002	0,0152	0,0221	0,0002
N15	Spambase	0,0041	0,0071	0,0081	0,0063	0,0094	0,0093	0,0107	0,0110
N16	Waveform	0,0093	0,0106	0,0072	0,0038	0,0080	0,0129	0,0131	0,0083
N17	Arrhythmia	0,0526	0,0429	0,0467	0,0543	0,0353	0,0482	0,0325	0,0331
N18	Semeion	0,0112	0,0167	0,0175	0,0115	0,0124	0,0127	0,0127	0,0073
N19	Clean	0,0164	0,0200	0,0197	0,0137	0,0109	0,0161	0,0134	0,0144
N20	CNAE	0,0175	0,0176	0,0185	0,0221	0,0237	0,0138	0,0231	0,0180
N21	Hillvalley	0,0330	0,0306	0,0490	0,0445	0,0332	0,0417	0,0262	0,0315

Çizelge 4.89. Uygunluk değerlerinin en iyi değerine göre BinSSA varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA5	BinSSA6	BinSSA7	BinSSA8
		En iyi	En iyi	En iyi	En iyi	En iyi	En iyi	En iyi	En iyi
N1	Wine	0,0023	0,0023	0,0015	0,0023	0,0015	0,0015	0,0015	0,0015
N2	Hepatitis	0,0393	0,0367	0,0372	0,0367	0,0011	0,0011	0,0346	0,0335
N3	Vehicle	0,1894	0,2042	0,1919	0,1777	0,2042	0,2078	0,1978	0,2248
N4	Zoo	0,0031	0,0025	0,0031	0,0025	0,0019	0,0013	0,0019	0,0013
N5	Heart	0,0640	0,0815	0,0581	0,0405	0,0756	0,0924	0,0756	0,0573
N6	Breast Cancer	0,0105	0,0044	0,0033	0,0022	0,0044	0,0056	0,0105	0,0056
N7	Ionosphere	0,0053	0,0038	0,0038	0,0174	0,0009	0,0012	0,0009	0,0009
N8	Lung Cancer	0,0039	0,0038	0,0038	0,0034	0,0002	0,0002	0,0004	0,0004
N9	Dermatology	0,0047	0,0041	0,0041	0,0038	0,0015	0,0015	0,0021	0,0015
N10	Sonar	0,0053	0,0048	0,0053	0,0047	0,0010	0,0010	0,0008	0,0008
N11	BreastEW	0,0248	0,0050	0,0228	0,0228	0,0205	0,0228	0,0228	0,0109
N12	Soybean Small	0,0023	0,0026	0,0017	0,0017	0,0003	0,0003	0,0003	0,0003
N13	Movementlibras	0,1309	0,1298	0,1298	0,1147	0,0974	0,0973	0,1108	0,0971
N14	Parkinsons	0,0023	0,0018	0,0014	0,0018	0,0005	0,0005	0,0005	0,0005
N15	Spambase	0,0581	0,0581	0,0569	0,0654	0,0622	0,0601	0,0719	0,0659
N16	Waveform	0,1501	0,1471	0,1562	0,1639	0,1607	0,1527	0,1627	0,1743
N17	Arrhythmia	0,2169	0,2813	0,2143	0,2249	0,1990	0,1983	0,2313	0,2426
N18	Semeion	0,7770	0,7486	0,7580	0,7711	0,7640	0,7539	0,7545	0,7737
N19	Clean	0,0073	0,0057	0,0060	0,0155	0,0014	0,0020	0,0120	0,0219
N20	CNAE	0,0822	0,1104	0,1097	0,0995	0,1676	0,1916	0,1915	0,1881
N21	Hillvalley	0,3788	0,3768	0,3429	0,3506	0,3307	0,3220	0,3632	0,3054

Çizelge 4.90. Uygunluk değerlerinin en kötü değerine göre BinSSA varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA5	BinSSA6	BinSSA7	BinSSA8
		En kötü	En kötü	En kötü	En kötü	En kötü	En kötü	En kötü	En kötü
N1	Wine	0,0329	0,0321	0,0581	0,0588	0,0848	0,0573	0,0306	0,0306
N2	Hepatitis	0,1687	0,2017	0,1692	0,1357	0,1666	0,1655	0,1661	0,1991
N3	Vehicle	0,2809	0,2803	0,2681	0,2792	0,2951	0,2775	0,2893	0,2893
N4	Zoo	0,2011	0,0533	0,0533	0,1523	0,1015	0,1510	0,1021	0,1021
N5	Heart	0,1828	0,1813	0,1688	0,1872	0,1681	0,1681	0,1688	0,2040
N6	Breast Cancer	0,0403	0,0536	0,0536	0,0475	0,0464	0,0320	0,0392	0,0453
N7	Ionosphere	0,0760	0,1167	0,0751	0,0616	0,0436	0,0436	0,0577	0,0580
N8	Lung Cancer	0,1711	0,3339	0,1702	0,3336	0,1654	0,1655	0,0021	0,1654
N9	Dermatology	0,0068	0,0192	0,0186	0,0324	0,0292	0,0301	0,0445	0,0307
N10	Sonar	0,1291	0,1291	0,1038	0,1042	0,0753	0,0997	0,1000	0,0758
N11	BreastEW	0,0651	0,0644	0,0641	0,0545	0,0505	0,0601	0,0505	0,0502
N12	Soybean Small	0,0043	0,0034	0,0031	0,0029	0,0006	0,0006	0,0006	0,0006
N13	Movementlibras	0,2824	0,2941	0,3079	0,3204	0,2209	0,1935	0,2621	0,2759
N14	Parkinsons	0,0041	0,0036	0,0032	0,0027	0,0009	0,0500	0,0995	0,0009
N15	Spambase	0,0719	0,0822	0,0820	0,0841	0,0910	0,0935	0,1013	0,1061
N16	Waveform	0,1721	0,1726	0,1758	0,1726	0,1810	0,1822	0,1894	0,1958
N17	Arrhythmia	0,4144	0,4242	0,4125	0,4339	0,3319	0,3961	0,3745	0,3631
N18	Semeion	0,8193	0,8249	0,8237	0,8143	0,8069	0,8064	0,8059	0,8062
N19	Clean	0,0694	0,0893	0,0882	0,0573	0,0533	0,0534	0,0645	0,0737
N20	CNAE	0,1267	0,1651	0,1739	0,1788	0,2552	0,2315	0,2647	0,2506
N21	Hillvalley	0,5110	0,5088	0,4996	0,5242	0,4456	0,4543	0,4456	0,4292

Çizelge 4.91. Ortalama sınıflandırma doğruluğuna göre BinSSA varyasyonlarının karşılaştırılması

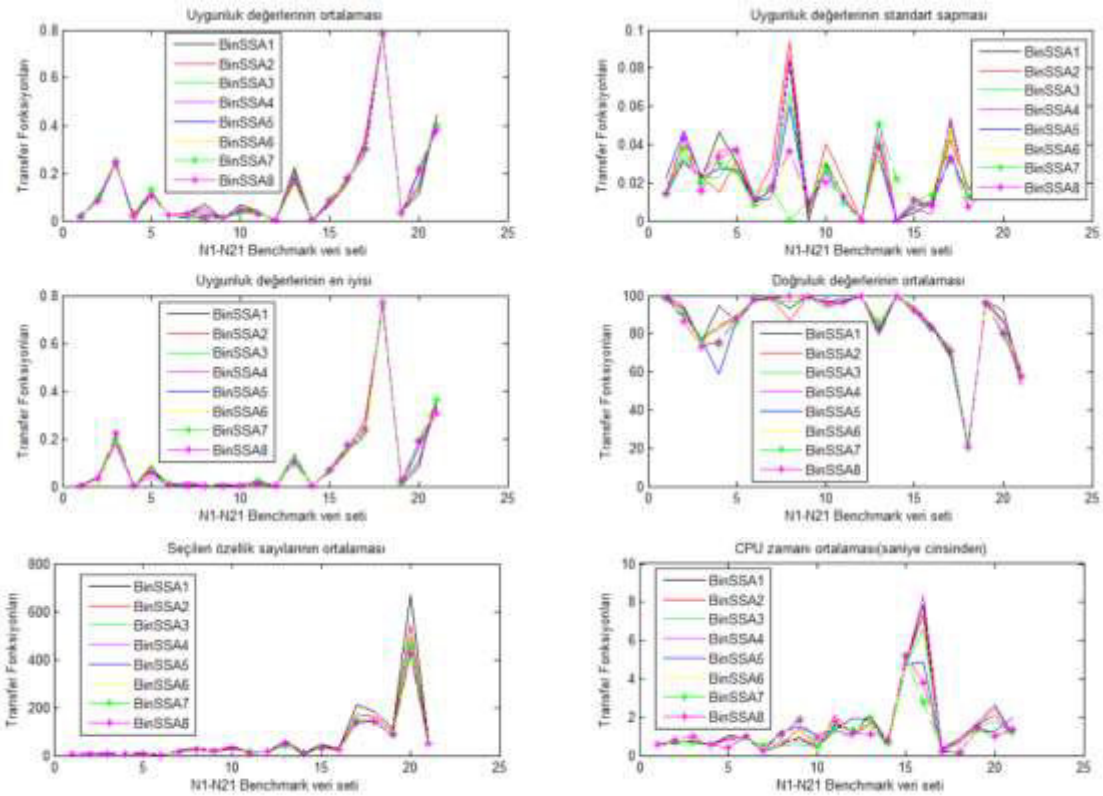
ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA5	BinSSA6	BinSSA7	BinSSA8
N1	<i>Wine</i>	99,44	98,33	99,44	100	99,44	98,33	98,33	99,44
N2	<i>Hepatitis</i>	92,67	94	89,33	94,63	88,67	90	90,67	86,67
N3	<i>Vehicle</i>	75,26	76,09	77,75	75,6	76,9	74,56	75,86	73,01
N4	<i>Zoo</i>	95	85	82	84	59	85	74	76
N5	<i>Heart</i>	85,93	88,52	90	87,41	87,04	87,41	86,30	88,15
N6	<i>Breast Cancer</i>	97,38	98,55	97,12	97,39	98,84	98,41	99,13	98,12
N7	<i>Ionosphere</i>	99,14	98,57	98,28	98	98,86	99,14	99,14	98,57
N8	<i>Lung Cancer</i>	93,33	86,67	96,67	100	100	100	100	100
N9	<i>Dermatology</i>	99,73	99,73	99,45	100	98,63	98,90	99,18	99,18
N10	<i>Sonar</i>	95,5	95,5	93	95	97	95,5	96,5	96,5
N11	<i>BreastEW</i>	97,2	96,6	97,6	95,8	97,4	96,8	97	96,6
N12	<i>Soybean Small</i>	100	100	100	100	100	100	100	100
N13	<i>Movementlibras</i>	80	78,61	78,61	80,83	82,78	84,72	86,11	82,5
N14	<i>Parkinsons</i>	100	100	100	100	100	100	100	100
N15	<i>Spambase</i>	94,38	94,08	93,71	92,58	91,83	93,17	92,38	92,13
N16	<i>Waveform</i>	84,03	84,43	83,23	84,67	81,77	81,83	82,4	82,93
N17	<i>Arrhythmia</i>	69,56	68,22	67,33	70,11	73,33	72,22	71,33	70,44
N18	<i>Semeion</i>	19,27	20,6	19,8	19,73	21,4	20,67	22,4	20,53
N19	<i>Clean</i>	97,5	97,5	95,75	96,5	96,5	95,75	95,75	96,25
N20	<i>CNAE</i>	91,75	86,5	85,75	86,25	79,25	80,25	79,75	80,25
N21	<i>Hillvalley</i>	57,5	53,17	56,83	61,67	60	61,5	58,67	57,33

Çizelge 4.92. Ortalama seçilen özellik sayısına göre BinSSA varyasyonlarının karşılaştırılması (virgüllü değerler yukarıya yuvarlanmıştır)

ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA5	BinSSA6	BinSSA7	BinSSA8
N1	<i>Wine</i>	7	6	6	5	5	6	6	5
N2	<i>Hepatitis</i>	10	9	9	9	8	8	8	7
N3	<i>Vehicle</i>	12	10	9	8	9	8	8	8
N4	<i>Zoo</i>	8	7	7	6	6	6	6	6
N5	<i>Heart</i>	9	8	8	5	7	5	8	7
N6	<i>Breast Cancer</i>	4	4	4	4	4	4	4	4
N7	<i>Ionosphere</i>	18	17	16	14	15	15	17	14
N8	<i>Lung Cancer</i>	31	28	29	24	29	30	26	25
N9	<i>Dermatology</i>	20	17	16	17	19	19	20	19
N10	<i>Sonar</i>	39	33	30	28	30	29	30	28
N11	<i>BreastEW</i>	17	15	13	12	13	13	14	13
N12	<i>Soybean Small</i>	15	17	16	14	14	14	15	14
N13	<i>Movementlibras</i>	62	52	46	40	41	41	41	49
N14	<i>Parkinsons</i>	12	11	9	8	8	8	8	9
N15	<i>Spambase</i>	45	38	34	33	32	32	30	31
N16	<i>Waveform</i>	30	27	25	23	24	24	25	24
N17	<i>Arrhythmia</i>	209	173	155	147	140	144	143	138
N18	<i>Semeion</i>	185	162	142	137	144	137	141	142
N19	<i>Clean</i>	112	102	88	87	85	84	86	87
N20	<i>CNAE</i>	671	547	492	421	436	425	448	427
N21	<i>Hillvalley</i>	68	59	51	49	54	53	52	52

Çizelge 4.93. Ortalama CPU süresine (saniye cinsinden) göre BinSSA varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA5	BinSSA6	BinSSA7	BinSSA8
N1	Wine	0,012951	0,011959	0,011802	0,011927	0,011794	0,011943	0,011871	0,011654
N2	Hepatitis	0,013486	0,013938	0,013944	0,013696	0,012519	0,013955	0,014354	0,015966
N3	Vehicle	0,014397	0,015873	0,014239	0,013733	0,015081	0,012790	0,013727	0,020219
N4	Zoo	0,011560	0,011367	0,011411	0,011524	0,011624	0,011500	0,011513	0,011556
N5	Heart	0,020210	0,018348	0,016082	0,015992	0,008635	0,008167	0,008462	0,008222
N6	Breast Cancer	0,018628	0,018768	0,020056	0,020041	0,019629	0,019809	0,018795	0,019938
N7	Ionosphere	0,003323	0,003321	0,003307	0,003402	0,004909	0,007584	0,009888	0,004521
N8	Lung Cancer	0,010889	0,011145	0,010635	0,015652	0,025774	0,026552	0,021707	0,022269
N9	Dermatology	0,019162	0,028312	0,011538	0,016282	0,029356	0,024762	0,039027	0,035915
N10	Sonar	0,007885	0,007858	0,007999	0,008934	0,021084	0,015992	0,009510	0,017647
N11	BreastEW	0,031945	0,041322	0,025036	0,026352	0,027515	0,035685	0,029197	0,034884
N12	Soybean Small	0,025202	0,022008	0,022684	0,018920	0,038578	0,027287	0,027012	0,022864
N13	MovementLibras	0,042026	0,034521	0,030830	0,040801	0,038578	0,036183	0,039311	0,022449
N14	Parkinsons	0,010770	0,012701	0,008113	0,016344	0,010589	0,008134	0,012414	0,016158
N15	Spambase	0,094527	0,096210	0,094008	0,094553	0,095587	0,098474	0,103117	0,104591
N16	Waveform	0,156636	0,146358	0,130794	0,166894	0,096887	0,084169	0,055741	0,075010
N17	Arrhythmia	0,007596	0,006798	0,006439	0,006368	0,004633	0,003834	0,003857	0,003869
N18	Semeion	0,017521	0,014371	0,012710	0,012195	0,004030	0,003837	0,003375	0,002895
N19	Clean	0,033037	0,032540	0,030222	0,029595	0,027146	0,025755	0,026431	0,030037
N20	CNAE	0,053236	0,041479	0,035407	0,048600	0,025650	0,023499	0,020041	0,020384
N21	Hillvalley	0,022073	0,021074	0,023293	0,020265	0,039391	0,024058	0,025112	0,027707



Şekil 4.13. BinSSA varyasyonları için yakınsama grafikleri

BinSSA-CR varyasyonlarının karşılaştırılması:

BinSSA-CR'nin varyasyonları (S-şekilli ve V-şekilli), özellik seçme problemi üzerinde farklı ölçütlerde (uygunluk fonksiyonunun ortalaması, uygunluk fonksiyonunun standart sapması, uygunluk fonksiyonunun en iyisi, uygunluk

fonksiyonunun en kötüsü, doğruluk değerlerinin ortalaması, seçilen özellik sayısının ortalama sayısı ve CPU süresinin ortalaması) analiz edilmiştir. Özellik seçme problemi için yirmi bir karşılaştırma veri kümesi belirlenmiştir.

Çizelge 4.94’de BinSSA-CR varyasyonlarının uygunluk sonuçlarının ortalama değerleri gösterilmiştir. En iyi sonuçlar kalın olarak işaretlenmiştir. Uygunluk sonuçlarının ortalamasına göre, S-şekilli BinSSA4-CR varyasyonu diğer BinSSA-CR varyasyonlarından daha yüksek bir başarı göstermiştir. BinSSA4-CR varyasyonu, 21 veri kümesinden (N1, N2, N4, N6, N11, N15, N17, N18 ve N21) 9 veri kümesi için kıyaslama veri kümelerinin % 42,86’sında yüksek başarı göstermektedir. Bunun nedeni, BinSSA-CR’nin hem keşif hem de sömürü yeteneklerinin ikili arama uzayında çaprazlama operatörü ile geliştirilmesidir. Genel olarak, ortalama uygunluk sonuçlarına göre, V-şekilli BinSSA-CR varyasyonları (BinSSA5, BinSSA6, BinSSA7 ve BinSSA8), S-şekilli BinSSA4-CR varyasyonu dışında daha başarılıdır.

Çizelge 4.95’de BinSSA-CR varyasyonlarının uygunluk sonuçlarının standart sapma (Std.Dev.) değerleri gösterilmiştir. En iyi sonuçlar kalın olarak işaretlenmiştir. Uygunluk sonuçlarının standart sapmasına göre, BinSSA4-CR, BinSSA5-CR ve BinSSA8-CR varyasyonları özellik seçme problemi için yüksek başarı göstermiştir. BinSSA5-CR ve BinSSA8-CR varyasyonları, 21 veri kümesinden 4’ü için kıyaslama veri kümelerinin % 19,05’inde başarı göstermektedir. BinSSA4-CR varyasyonu, 21 veri kümesinden 12’si için kıyaslama veri kümelerinin % 57,14’ünde önemli bir başarı göstermektedir. Bunun nedeni, BinSSA4-CR varyasyonunun yerel optimal çözümlerde daha verimli bir şekilde atlaması, diğer varyasyonlar ise yerel çözümlerde durgunluğa bırakmasıdır.

Çizelgeler 4.96 ve 4.97’de BinSSA-CR varyasyonlarının en iyi ve en kötü uygunluk değerleri gösterilmiştir. En iyi uygunluk sonuçlarına göre, BinSSA4-CR, BinSSA5-CR, BinSSA7-CR ve BinSSA8-CR varyasyonları özellik seçme problemi için yüksek başarı göstermiştir. BinSSA4-CR ve BinSSA5-CR varyasyonları, 21 veri kümesinden 8’i için referans veri setlerinin % 38,1’inde yüksek başarı göstermektedir. BinSSA7-CR varyasyonu, 21 veri kümesinin 10’u (N1, N2, N4, N7, N8, N9, N10, N12, N14 ve N19) için kıyaslama veri kümelerinin % 47,62’sinde üstün performans göstermektedir. BinSSA8-CR varyasyonu, 21 veri kümesinden 9’u (N1, N4, N8, N9, N10, N12, N14, N16 ve N17) için kıyaslama veri kümelerinin % 42,86’sında yüksek performans gösterir. En kötü değer sonuçlarına göre, BinSSA6 varyasyonu, özellik seçme problemi için 21 veri kümesinden 9’u için minimum değere sahiptir.

Çizelge 4.98'de BinSSA-CR varyasyonlarının ortalama doğruluk değerlerini göstermektedir. Doğruluk değerlerinin ortalamasına göre, BinSSA4-CR ve BinSSA5-CR varyasyonları yüksek performans göstermektedir. BinSSA5-CR varyasyonu, 21 veri kümesinden 8'i (N1, N7, N8, N10, N11, N12, N13 ve N14) için kıyaslama veri kümelerinin % 38,10'unda başarı göstermektedir. BinSSA4-CR varyasyonu, 21 veri kümesinden 13'ü (N1, N3, N4, N6, N8, N9, N12, N14, N15, N16, N19, N20 ve N21) için kıyaslama veri kümelerinin % 61,90'ında başarı göstermektedir. Bu başarının nedeni, aday çözümler ile çaprazlama operatörü arasındaki daha dengeli bir çeşitlilikten kaynaklanmaktadır.

Çizelge 4.99'da BinSSA-CR varyasyonlarının seçilen özelliklerinin ortalama sayısı gösterilmektedir. Seçilen özelliklerin ortalama sayısına göre, BinSSA-CR (BinSSA4-CR, BinSSA6-CR, BinSSA7-CR ve BinSSA8-CR) varyasyonları daha az özellik seçmiştir. BinSSA4-CR varyasyonu, 21 veri kümesinden 15'i (N1, N3, N4, N5, N6, N7, N8, N9, N10, N11, N12, N13, N14, N18 ve N21) için minimum özellikler seçmiştir. BinSSA8-CR, 21 veri kümesinden 11'i için minimum özellikler seçmiştir ve BinSSA6-CR ve BinSSA7-CR, 21 veri kümesinden 6'sı için minimum özellikleri seçmiştir.

Çizelge 4.100'de BinSSA-CR varyasyonlarının CPU zamanının ortalamasını göstermektedir. CPU zamanı ortalama sonuçlarına göre, BinSSA8-CR varyasyonu, 21 veri kümesinden (N2, N6, N9, N17, N18, N19 ve N20) 7'si için mümkün olan en kısa sürede sonuç elde etmiştir. BinSSA8-CR varyasyonu, keşfetme ve sömürü aşamalarını BinSSA-CR'nin diğer varyasyonlarından daha hızlı gerçekleştirmiştir.

Şekil 4.14'de BinSSA-CR'nin farklı varyasyonları arasındaki karşılaştırma için yakınsama grafikleri gösterilmiştir.

Yapılan çalışmalar neticesinde BinSSA4-CR ve BinSSA8-CR varyasyonlarının diğer varyasyonlar arasında en iyi yeri elde ettiği rahatlıkla görülebilmektedir. Bunun nedeni, çaprazlama operatörünün karşılaştırmalı veri kümelerinin çoğunda BinSSA-CR'nin arama yeterliliklerini geliştirmesidir.

Çizelge 4.94. Uygunluk değerlerinin ortalamasına göre BinSSA-CR varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA4-CR	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
		Ortalama	Ortalama	Ortalama	Ortalama	Ortalama	Ortalama	Ortalama	Ortalama
N1	Wine	0,0151	0,0173	0,0160	0,0072	0,0135	0,0122	0,0137	0,0166
N2	Hepatitis	0,1053	0,1043	0,0889	0,0625	0,0840	0,0694	0,0709	0,0692
N3	Vehicle	0,2402	0,2380	0,2464	0,2295	0,2164	0,2187	0,2206	0,2236
N4	Zoo	0,0194	0,0211	0,0236	0,0080	0,0100	0,0153	0,0275	0,0125
N5	Heart	0,1119	0,1228	0,1140	0,1099	0,1111	0,1105	0,1083	0,1128
N6	Breast Cancer	0,0217	0,0226	0,0202	0,0200	0,0222	0,0241	0,0211	0,0207
N7	Ionosphere	0,0303	0,0228	0,0276	0,0299	0,0143	0,0086	0,0112	0,0156
N8	Lung Cancer	0,1215	0,0544	0,0455	0,0029	0,0006	0,0005	0,0004	0,0004
N9	Dermatology	0,0066	0,0071	0,0073	0,0072	0,0040	0,0036	0,0050	0,0058
N10	Sonar	0,0645	0,0587	0,0372	0,0394	0,0105	0,0105	0,0139	0,0250
N11	BreastEW	0,0363	0,0324	0,0396	0,0295	0,0323	0,0406	0,0376	0,0369
N12	Soybean Small	0,0035	0,0030	0,0027	0,0025	0,0005	0,0005	0,0005	0,0005
N13	Movementlibras	0,2151	0,2086	0,2065	0,1584	0,1458	0,1587	0,1543	0,1561
N14	Parkinsons	0,0082	0,0077	0,0025	0,0020	0,0007	0,0032	0,0032	0,0006
N15	Spambase	0,0644	0,0685	0,0707	0,0630	0,0679	0,0721	0,0734	0,0702
N16	Waveform	0,1615	0,1572	0,1676	0,1604	0,1580	0,1593	0,1594	0,1633
N17	Arrhythmia	0,3302	0,3174	0,3333	0,2520	0,2780	0,2777	0,2756	0,2581
N18	Semeion	0,7987	0,7938	0,7903	0,7574	0,7803	0,7819	0,7783	0,7837
N19	Clean	0,0342	0,0359	0,0360	0,0164	0,0134	0,0121	0,0160	0,0208
N20	CNAE	0,0995	0,1336	0,1429	0,1239	0,1566	0,1413	0,1789	0,1823
N21	Hillvalley	0,4474	0,4509	0,4423	0,3677	0,4212	0,4217	0,4204	0,3898

Çizelge 4.95. Uygunluk değerlerinin standart sapmasına göre BinSSA-CR varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA4-CR	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
		Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.
N1	Wine	0,0135	0,0190	0,0184	0,0102	0,0166	0,0135	0,0166	0,0213
N2	Hepatitis	0,0525	0,0519	0,0591	0,0175	0,0348	0,0294	0,0464	0,0363
N3	Vehicle	0,0228	0,0187	0,0232	0,0046	0,0216	0,0193	0,0198	0,0216
N4	Zoo	0,0458	0,0366	0,0404	0,0157	0,0183	0,0217	0,0493	0,0343
N5	Heart	0,0229	0,0388	0,0242	0,0172	0,0335	0,0390	0,0264	0,0308
N6	Breast Cancer	0,0094	0,0107	0,0095	0,0104	0,0084	0,0148	0,0116	0,0137
N7	Ionosphere	0,0089	0,0126	0,0197	0,0150	0,0113	0,0087	0,0114	0,0129
N8	Lung Cancer	0,1427	0,0777	0,0907	0,0680	0,0002	0,0002	0,0002	0,0001
N9	Dermatology	0,0031	0,0066	0,0056	0,0059	0,0041	0,0032	0,0056	0,0075
N10	Sonar	0,0371	0,0302	0,0254	0,0118	0,0122	0,0122	0,0150	0,0203
N11	BreastEW	0,0166	0,0133	0,0140	0,0083	0,0176	0,0185	0,0154	0,0149
N12	Soybean Small	0,0004	0,0003	0,0003	0,0003	0,0001	0,0001	0,0001	0,0001
N13	Movementlibras	0,0462	0,0357	0,0496	0,0355	0,0463	0,0362	0,0305	0,0269
N14	Parkinsons	0,0157	0,0153	0,0004	0,0003	0,0002	0,0111	0,0111	0,0002
N15	Spambase	0,0078	0,0060	0,0056	0,0026	0,0073	0,0103	0,0070	0,0073
N16	Waveform	0,0044	0,0051	0,0080	0,0014	0,0039	0,0043	0,0089	0,0127
N17	Arrhythmia	0,0482	0,0414	0,0329	0,0289	0,0403	0,0391	0,0319	0,0388
N18	Semeion	0,0127	0,0162	0,0151	0,0086	0,0137	0,0100	0,0094	0,0118
N19	Clean	0,0176	0,0211	0,0171	0,0134	0,0123	0,0105	0,0119	0,0163
N20	CNAE	0,0188	0,0178	0,0247	0,0174	0,0143	0,0184	0,0145	0,0237
N21	Hillvalley	0,0377	0,0355	0,0261	0,0055	0,0373	0,0484	0,0427	0,0377

Çizelge 4.96. Uygunluk değerlerinin en iyi değerine göre BinSSA-CR varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA4-CR	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
		En iyi	En iyi	En iyi	En iyi	En iyi	En iyi	En iyi	En iyi
N1	Wine	0,0031	0,0031	0,0023	0,0015	0,0015	0,0015	0,0015	0,0015
N2	Hepatitis	0,0058	0,0351	0,0032	0,0021	0,0346	0,0341	0,0005	0,0021
N3	Vehicle	0,1936	0,1936	0,1989	0,1719	0,1796	0,1807	0,1855	0,1796
N4	Zoo	0,0025	0,0019	0,0025	0,0013	0,0013	0,0019	0,0013	0,0013
N5	Heart	0,0772	0,0756	0,0764	0,0596	0,0581	0,0222	0,0756	0,0573
N6	Breast Cancer	0,0044	0,0044	0,0022	0,0021	0,0094	0,0094	0,0033	0,0022
N7	Ionosphere	0,0180	0,0035	0,0032	0,0032	0,0009	0,0009	0,0006	0,0009
N8	Lung Cancer	0,0043	0,0041	0,0034	0,0029	0,0004	0,0004	0,0002	0,0002
N9	Dermatology	0,0053	0,0038	0,0038	0,0029	0,0015	0,0015	0,0015	0,0015
N10	Sonar	0,0052	0,0058	0,0045	0,0058	0,0010	0,0010	0,0010	0,0010
N11	BreastEW	0,0166	0,0132	0,0047	0,0235	0,0106	0,0205	0,0116	0,0112
N12	Soybean Small	0,0029	0,0023	0,0023	0,0020	0,0003	0,0003	0,0003	0,0003
N13	Movementlibras	0,1036	0,1568	0,1285	0,1290	0,0570	0,0994	0,1114	0,0973
N14	Parkinsons	0,0023	0,0018	0,0018	0,0018	0,0005	0,0005	0,0005	0,0005
N15	Spambase	0,0523	0,0585	0,0622	0,0515	0,0569	0,0613	0,0656	0,0560
N16	Waveform	0,1550	0,1498	0,1592	0,1595	0,1545	0,1518	0,1503	0,1414
N17	Arrhythmia	0,2606	0,2591	0,2697	0,1919	0,2222	0,2109	0,2318	0,1544
N18	Semeion	0,7664	0,7589	0,7676	0,7474	0,7483	0,7645	0,7614	0,7668
N19	Clean	0,0069	0,0056	0,0052	0,0048	0,0017	0,0016	0,0014	0,0015
N20	CNAE	0,0577	0,0954	0,1048	0,0575	0,1270	0,1118	0,1559	0,1406
N21	Hillvalley	0,3944	0,4017	0,4008	0,3207	0,3645	0,3397	0,3226	0,3304

Çizelge 4.97. Uygunluk değerlerinin en kötü değerine göre BinSSA-CR varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA4-CR	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
		En kötü	En kötü	En kötü	En kötü	En kötü	En kötü	En kötü	En kötü
N1	Wine	0,0321	0,0588	0,0581	0,0313	0,0588	0,0313	0,0565	0,0856
N2	Hepatitis	0,2357	0,2033	0,2666	0,1676	0,1676	0,1346	0,1661	0,1666
N3	Vehicle	0,2880	0,2756	0,2873	0,2745	0,2664	0,2521	0,2628	0,2616
N4	Zoo	0,1548	0,1504	0,1510	0,1028	0,0526	0,0526	0,1999	0,1504
N5	Heart	0,1688	0,2055	0,1513	0,1856	0,1688	0,1856	0,1490	0,1856
N6	Breast Cancer	0,0414	0,0464	0,0381	0,0547	0,0392	0,0618	0,0464	0,0547
N7	Ionosphere	0,0477	0,0480	0,0742	0,0613	0,0309	0,0295	0,0436	0,0436
N8	Lung Cancer	0,5018	0,1711	0,3336	0,1695	0,0011	0,0011	0,0009	0,0007
N9	Dermatology	0,0197	0,0321	0,0200	0,0194	0,0165	0,0165	0,0168	0,0295
N10	Sonar	0,1548	0,1283	0,0796	0,0791	0,0276	0,0276	0,0507	0,0754
N11	BreastEW	0,0746	0,0548	0,0532	0,0637	0,0706	0,0703	0,0604	0,0611
N12	Soybean Small	0,0043	0,0034	0,0031	0,0029	0,0006	0,0006	0,0006	0,0006
N13	Movementlibras	0,2962	0,2807	0,3215	0,2799	0,2354	0,2635	0,2219	0,2075
N14	Parkinsons	0,0540	0,0527	0,0036	0,0032	0,0009	0,0504	0,0504	0,0009
N15	Spambase	0,0761	0,0792	0,0834	0,0884	0,0808	0,0917	0,0859	0,0781
N16	Waveform	0,1674	0,1634	0,1765	0,1681	0,1639	0,1624	0,1703	0,1716
N17	Arrhythmia	0,4253	0,3904	0,4009	0,4010	0,3536	0,3533	0,3530	0,3087
N18	Semeion	0,8192	0,8178	0,8179	0,8101	0,8056	0,8009	0,8002	0,8028
N19	Clean	0,0595	0,0892	0,0683	0,0578	0,0339	0,0346	0,0428	0,0538
N20	CNAE	0,1218	0,1600	0,1840	0,1738	0,1667	0,1706	0,2007	0,2255
N21	Hillvalley	0,5098	0,5340	0,4921	0,5495	0,4887	0,5225	0,4707	0,4786

Çizelge 4.98. Ortalama sınıflandırma doğruluğuna göre BinSSA-CR varyasyonlarının karşılaştırılması

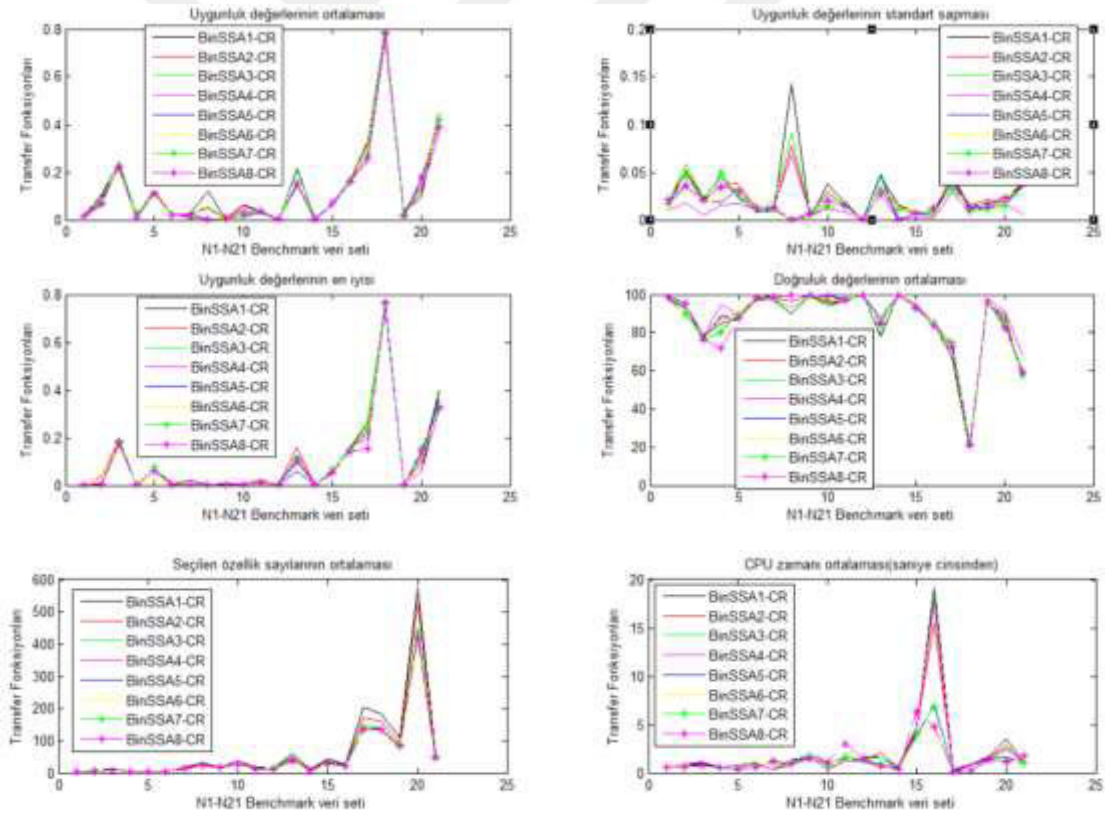
ID	Benchmark	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA4-CR	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
N1	<i>Wine</i>	98,33	97,78	100	100	100	99,44	98,89	98,89
N2	<i>Hepatitis</i>	92,67	90	93,33	94,65	94,67	94,67	89,33	95,33
N3	<i>Vehicle</i>	77,75	76,57	75,27	78,34	77,51	76,80	77,16	76,45
N4	<i>Zoo</i>	85	88	81	95	89	87	80	72
N5	<i>Heart</i>	87,04	87,04	89,26	89,26	89,63	91,11	89,26	87,78
N6	<i>Breast Cancer</i>	97,48	97,54	97,12	98,55	98,26	98,12	97,83	98,12
N7	<i>Ionosphere</i>	97,14	98	96,86	98,29	99,71	99,14	98,86	98,86
N8	<i>Lung Cancer</i>	90	96,67	93,33	100	100	100	100	100
N9	<i>Dermatology</i>	100	99,73	99,45	100	99,45	100	99,73	99,73
N10	<i>Sonar</i>	93,5	95,5	94	96	99,5	98	97	99
N11	<i>BreastEW</i>	97,4	96,6	96,6	97	97,6	97,2	97,4	97,2
N12	<i>Soybean Small</i>	100	100	100	100	100	100	100	100
N13	<i>Movementlibras</i>	78,06	79,17	79,17	85,33	87,78	86,39	85	84,72
N14	<i>Parkinsons</i>	100	100	100	100	100	100	100	100
N15	<i>Spambase</i>	94,18	94,53	93,38	95,18	93,75	94,08	92,75	93,04
N16	<i>Waveform</i>	84,82	83,9	84,1	85,20	85	85,13	84,87	83,93
N17	<i>Arrhythmia</i>	67,11	71,22	70,11	74,11	74,67	70,67	74,89	72,44
N18	<i>Semeion</i>	19	19,6	19,4	20,2	21,87	22	21,2	20,87
N19	<i>Clean</i>	95,25	96,5	95,25	97,75	96,25	96,25	96,25	96,5
N20	<i>CNAE</i>	89,5	87,5	84,75	91,50	85,75	86,5	83,5	82,5
N21	<i>Hillvalley</i>	55,83	55,5	60,5	68,17	56,33	56,5	57,83	59,67

Çizelge 4.99. Ortalama seçilen özellik sayısına göre BinSSA-CR varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA4-CR	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
N1	<i>Wine</i>	6	5	5	4	5	6	6	4
N2	<i>Hepatitis</i>	9	8	8	7	7	7	7	6
N3	<i>Vehicle</i>	11	9	9	7	8	8	8	8
N4	<i>Zoo</i>	6	5	5	5	5	5	5	5
N5	<i>Heart</i>	6	5	6	4	5	4	5	5
N6	<i>Breast Cancer</i>	4	4	4	4	4	4	4	4
N7	<i>Ionosphere</i>	17	15	15	13	14	14	15	14
N8	<i>Lung Cancer</i>	30	28	24	23	24	24	23	23
N9	<i>Dermatology</i>	19	16	16	15	17	17	17	17
N10	<i>Sonar</i>	38	32	29	27	29	29	28	27
N11	<i>BreastEW</i>	17	14	12	11	12	12	13	12
N12	<i>Soybean Small</i>	14	16	15	13	13	13	14	13
N13	<i>Movementlibras</i>	60	51	45	39	40	40	40	39
N14	<i>Parkinsons</i>	11	10	8	7	7	7	7	7
N15	<i>Spambase</i>	44	36	33	32	31	31	29	30
N16	<i>Waveform</i>	27	25	24	22	23	22	21	22
N17	<i>Arrhythmia</i>	205	170	150	140	137	138	139	135
N18	<i>Semeion</i>	183	160	140	130	140	130	135	136
N19	<i>Clean</i>	110	94	86	84	81	77	83	86
N20	<i>CNAE</i>	576	530	450	422	430	422	437	420
N21	<i>Hillvalley</i>	60	55	48	45	48	50	49	48

Çizelge 4.100. Ortalama CPU süresine (saniye cinsinden) göre BinSSA-CR varyasyonlarının karşılaştırılması

ID	Benchmark	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA4-CR	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
N1	<i>Wine</i>	0,011602	0,011716	0,011806	0,011804	0,011808	0,011714	0,012050	0,011692
N2	<i>Hepatitis</i>	0,014179	0,015428	0,014860	0,018260	0,016259	0,015844	0,012913	0,012747
N3	<i>Vehicle</i>	0,018238	0,014570	0,013647	0,022235	0,021161	0,012900	0,015515	0,017351
N4	<i>Zoo</i>	0,011140	0,011292	0,010200	0,011426	0,011610	0,011504	0,012204	0,011478
N5	<i>Heart</i>	0,016190	0,016242	0,017380	0,016180	0,007951	0,008584	0,008767	0,008210
N6	<i>Breast Cancer</i>	0,019958	0,019660	0,019291	0,019138	0,019752	0,020111	0,019407	0,011219
N7	<i>Ionosphere</i>	0,008177	0,008198	0,008066	0,007897	0,017871	0,018158	0,022813	0,025535
N8	<i>Lung Cancer</i>	0,026990	0,016945	0,024442	0,017774	0,018998	0,017382	0,019486	0,017804
N9	<i>Dermatology</i>	0,034272	0,036433	0,033375	0,033385	0,037107	0,031614	0,035545	0,029326
N10	<i>Sonar</i>	0,008076	0,008014	0,008289	0,008618	0,024495	0,024495	0,016801	0,021846
N11	<i>BreastEW</i>	0,024312	0,024334	0,026152	0,027412	0,027581	0,025193	0,033507	0,058727
N12	<i>Soybean Small</i>	0,032701	0,027046	0,028220	0,023690	0,027761	0,033327	0,026914	0,030128
N13	<i>MovementLibras</i>	0,034533	0,041715	0,033860	0,014222	0,013286	0,012845	0,018696	0,014420
N14	<i>Parkinsons</i>	0,006888	0,012586	0,013013	0,016594	0,011420	0,007275	0,007275	0,006963
N15	<i>Spambase</i>	0,087170	0,091826	0,095716	0,094780	0,075102	0,079525	0,082395	0,127663
N16	<i>Waveform</i>	0,383870	0,310974	0,361996	0,352527	0,142839	0,093220	0,137592	0,097726
N17	<i>Arrhythmia</i>	0,007773	0,006876	0,006430	0,006303	0,004765	0,004230	0,004042	0,003859
N18	<i>Semeion</i>	0,017747	0,014372	0,013194	0,014157	0,006503	0,005809	0,005092	0,004766
N19	<i>Clean</i>	0,032730	0,031078	0,029976	0,029683	0,029299	0,026824	0,026491	0,026139
N20	<i>CNAE</i>	0,069952	0,057169	0,050671	0,048691	0,033521	0,030129	0,025259	0,023302
N21	<i>Hillvalley</i>	0,021509	0,020991	0,031427	0,020511	0,019524	0,023883	0,021334	0,035074



Şekil 4.14. BinSSA-CR varyasyonları için yakınsama grafikleri

BinSSA ve BinSSA-CR varyasyonlarının karşılaştırılması:

Bu alt bölümde, BinSSA-CR'nin varyasyonları (S-şekilli ve V-şekilli), özellik seçme problemi üzerinde farklı ölçütlerde (uygunluk fonksiyonunun ortalaması, uygunluk fonksiyonunun standart sapması, uygunluk fonksiyonunun en iyisi, uygunluk fonksiyonunun en kötüsü, doğruluk değerlerinin ortalaması, seçilen özellik sayısının ortalama sayısı ve CPU süresinin ortalaması) analiz edilmiştir. Özellik seçme problemi için yirmi bir karşılaştırma veri kümesi belirlenmiştir. Çizelge 4.101'de BinSSA ve BinSSA-CR'nin S-şekilli varyasyonlarının uygunluk fonksiyonu sonuçlarının ortalaması gösterilmiştir. Çizelge 4.102'de BinSSA ve BinSSA-CR'nin V-şekilli varyasyonlarının uygunluk fonksiyonu sonuçlarının ortalaması gösterilmiştir. En iyi sonuçlar çizelgeler 4.101 ve 4.102'de koyu renkle gösterilmiştir. Şekil 4.15'de BinSSA-CR ve BinSSA'nın farklı varyasyonları arasındaki karşılaştırmalar için yakınsama grafikleri gösterilmiştir.

Uygunluk fonksiyonu sonuçlarının ortalamasına göre, S-şekilli BinSSA4-CR varyasyonu, BinSSA ve BinSSA-CR'nin diğer S-şekilli varyasyonlarından daha yüksek bir başarı göstermiştir. BinSSA4-CR varyasyonu, 21 veri kümesinden (N1, N2, N3, N4, N5, N6, N8, N11, N12, N13, N14, N15, N17, N18, N19 ve N21) 16 veri kümesi için % 76,19 oranında yüksek bir başarı göstermektedir. V-şekilli BinSSA5-CR ve BinSSA6-CR varyasyonları, BinSSA ve BinSSA-CR'nin diğer V-şekilli varyasyonlarından daha yüksek bir başarı göstermektedir. BinSSA5-CR ve BinSSA6-CR varyasyonları, 21 veri kümesinden 7 veri kümesi için kıyaslama veri kümelerinin % 33,33'ünde yüksek bir başarı göstermektedir. Sonuçlar, çaprazlama operatörünün küresel ve yerel olarak BinSSA'nın arama yeteneğini geliştirdiğini göstermektedir. Çaprazlama operatörü tarafından üretilen yeni aday çözüm seti istikrarlı bir çözüm sağlamıştır. BinSSA'nın hem keşif hem de sömürü yetenekleri, ikili arama uzayında geliştirilmiştir.

Uygunluk fonksiyonu sonuçlarının standart sapmasına göre, BinSSA4-CR ve BinSSA6-CR varyasyonları özellik seçme problemi için yüksek başarı göstermiştir. BinSSA4-CR varyasyonu, 21 veri setinin 15'i için referans veri setlerinin % 71,43'ünde başarı göstermektedir. BinSSA6-CR varyasyonu, 21 veri setinden 7'si için karşılaştırma veri setlerinin % 33,33'ünde yüksek bir başarı göstermektedir. Bunun nedeni BinSSA-CR'nin aday örümcekler arasındaki çeşitlendirme ve yoğunlaşma eğilimleri arasında daha istikrarlı bir denge kurabilmesidir.

Doğruluk değerlerinin ortalamasına göre, BinSSA4-CR ve BinSSA5-CR varyasyonları yüksek performans göstermektedir. BinSSA5-CR varyasyonu, 21 veri kümesinden (N1, N3, N4, N7, N8, N10, N11, N12, N13 ve N14) 10'u için kıyaslama

veri kümelerinin % 47,62'sinde başarı göstermiştir. BinSSA4-CR varyasyonu, 21 veri kümesinden (N1, N2, N3, N4, N6, N8, N9, N10, N12, N13, N14, N15, N16, N17, N19 ve N21) 16'sı için kıyaslama veri kümelerinin % 76,19'unda başarı göstermiştir. Bu başarının nedeni, aday çözümler ile çaprazlama operatörü arasındaki daha dengeli bir çeşitlilikten kaynaklanmaktadır.

Seçilen özelliklerin ortalama sayısına göre, S şeklindeki BinSSA4-CR varyasyonu daha az özellik seçmiştir. BinSSA4-CR varyasyonu 21 veri kümesinden 21'i için minimum özellikler seçmiştir. V şeklindeki BinSSA8-CR varyasyonu, 21 veri kümesinden 15'i için minimum özellikler seçmiştir. Sonuçlar BinSSA-CR varyasyonlarının daha az özellik seçtiğini ve en uygun özellikleri seçtiğini göstermektedir.

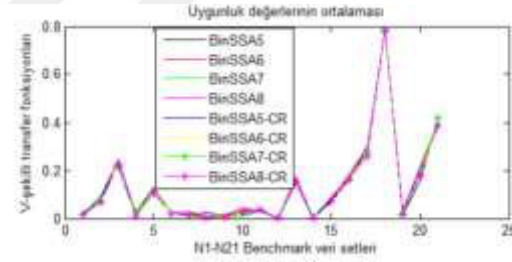
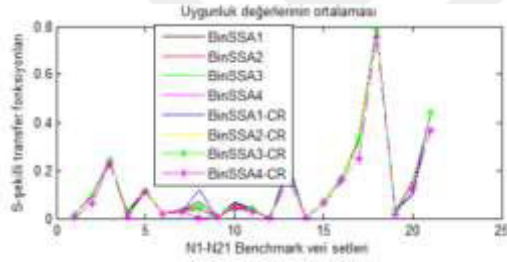
CPU zaman sonuçlarının ortalamasına göre, BinSSA3 ve BinSSA4 varyasyonları, 21 veri kümesinden 5'i için mümkün olan en kısa sürede en iyi sonucu elde etmiştir. BinSSA8 ve BinSSA8-CR varyasyonları, 21 veri kümesinden 4'ü için mümkün olan en kısa sürede sonuç almıştır.

Çizelge 4.101. Uygunluk değerlerinin ortalamasına göre, S-şekilli BinSSA ve BinSSA-CR varyasyonlarının karşılaştırılması

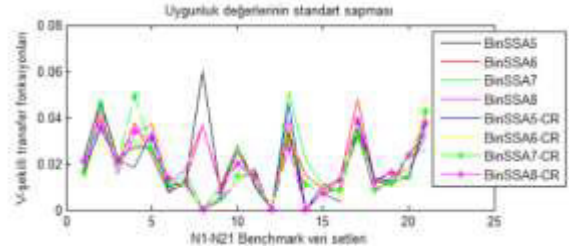
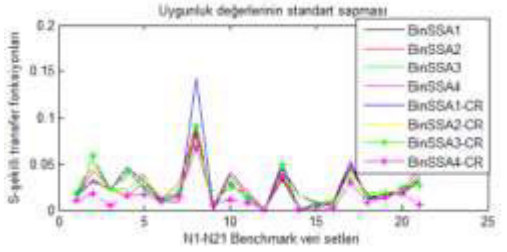
ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA4-CR
N1	<i>Wine</i>	0,0138	0,0158	0,0114	0,0145	0,0151	0,0173	0,0160	0,0072
N2	<i>Hepatitis</i>	0,0952	0,1043	0,1056	0,0924	0,1053	0,1043	0,0889	0,0625
N3	<i>Vehicle</i>	0,2336	0,2458	0,2331	0,2322	0,2402	0,238	0,2464	0,2295
N4	<i>Zoo</i>	0,0263	0,0085	0,0186	0,0210	0,0194	0,0211	0,0236	0,0080
N5	<i>Heart</i>	0,1255	0,1164	0,1170	0,1180	0,1119	0,1228	0,1140	0,1099
N6	<i>Breast Cancer</i>	0,0240	0,0260	0,0227	0,0205	0,0218	0,0226	0,0202	0,0200
N7	<i>Ionosphere</i>	0,0342	0,0317	0,0269	0,0352	0,0303	0,0228	0,0276	0,0299
N8	<i>Lung Cancer</i>	0,0714	0,0544	0,0375	0,0372	0,1215	0,0544	0,0455	0,0029
N9	<i>Dermatology</i>	0,0057	0,0063	0,0081	0,0090	0,0066	0,0071	0,0073	0,0072
N10	<i>Sonar</i>	0,0682	0,0499	0,0544	0,0407	0,0645	0,0587	0,0372	0,0394
N11	<i>BreastEW</i>	0,0423	0,0354	0,0427	0,0364	0,0363	0,0324	0,0396	0,0295
N12	<i>Soybean Small</i>	0,0035	0,0031	0,0026	0,0025	0,0035	0,003	0,0027	0,0025
N13	<i>Movementlibras</i>	0,2227	0,2071	0,2148	0,1877	0,2151	0,2086	0,2065	0,1584
N14	<i>Parkinsons</i>	0,0034	0,0028	0,0023	0,0022	0,0082	0,0077	0,0025	0,0020
N15	<i>Spambase</i>	0,0666	0,0674	0,0708	0,0731	0,0644	0,0685	0,0707	0,0630
N16	<i>Waveform</i>	0,1606	0,1550	0,1644	0,1690	0,1615	0,1572	0,1676	0,1604
N17	<i>Arrhythmia</i>	0,3138	0,3417	0,3124	0,3198	0,3302	0,3174	0,3333	0,2520
N18	<i>Semeion</i>	0,7966	0,7978	0,7992	0,7970	0,7987	0,7938	0,7903	0,7574
N19	<i>Clean</i>	0,0390	0,0337	0,0371	0,0344	0,0342	0,0359	0,0360	0,0164
N20	<i>CNAE</i>	0,1088	0,1385	0,1419	0,1570	0,0995	0,1336	0,1429	0,1239
N21	<i>Hillvalley</i>	0,4428	0,4473	0,4370	0,4310	0,4474	0,4509	0,4423	0,3677

Çizelge 4.102. Uygunluk değerlerinin ortalamasına göre, V-şekilli BinSSA ve BinSSA-CR varyasyonlarının karşılaştırılması

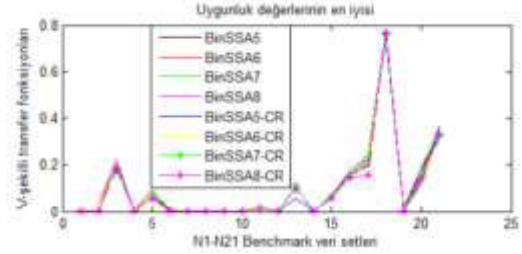
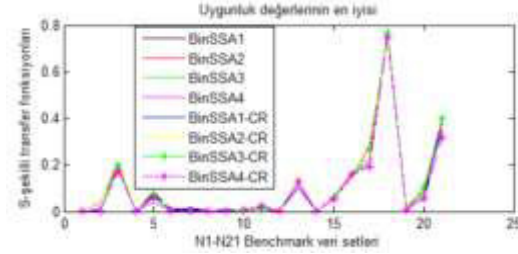
ID	Benchmark	BinSSA5	BinSSA6	BinSSA7	BinSSA8	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
N1	Wine	0,0256	0,0203	0,0177	0,0160	0,0135	0,0122	0,0137	0,0166
N2	Hepatitis	0,0821	0,0887	0,0986	0,0869	0,0840	0,0694	0,0709	0,0692
N3	Vehicle	0,2398	0,2392	0,2361	0,2498	0,2164	0,2187	0,2206	0,2236
N4	Zoo	0,0150	0,0248	0,0276	0,0200	0,0100	0,0153	0,0275	0,0125
N5	Heart	0,1304	0,1262	0,1328	0,1036	0,1111	0,1105	0,1083	0,1129
N6	Breast Cancer	0,0211	0,0186	0,0238	0,0248	0,0222	0,0241	0,0211	0,0207
N7	Ionosphere	0,0171	0,0207	0,0206	0,0268	0,0143	0,0086	0,0112	0,0156
N8	Lung Cancer	0,0252	0,0087	0,0005	0,0087	0,0006	0,0005	0,0004	0,0004
N9	Dermatology	0,0101	0,0100	0,0138	0,0148	0,0040	0,0036	0,005	0,0058
N10	Sonar	0,0308	0,0359	0,0309	0,0458	0,0105	0,0105	0,0139	0,0250
N11	BreastEW	0,0345	0,0386	0,0318	0,0296	0,0323	0,0406	0,0376	0,0369
N12	Soybean Small	0,0005	0,0005	0,0005	0,0005	0,0005	0,0005	0,0005	0,0005
N13	Movementlibras	0,1689	0,1544	0,1689	0,1742	0,1458	0,1587	0,1543	0,1561
N14	Parkinsons	0,0005	0,0057	0,0057	0,0006	0,0007	0,0032	0,0032	0,0006
N15	Spambase	0,0764	0,0794	0,0872	0,0844	0,0679	0,0721	0,0734	0,0702
N16	Waveform	0,1722	0,1693	0,1770	0,1821	0,1580	0,1593	0,1594	0,1633
N17	Arrhythmia	0,2779	0,2931	0,2975	0,3039	0,2780	0,2777	0,2756	0,2581
N18	Semeion	0,7851	0,7827	0,7825	0,7872	0,7803	0,7819	0,7783	0,7837
N19	Clean	0,0268	0,0288	0,0301	0,0369	0,0134	0,0121	0,0160	0,0208
N20	CNAE	0,2084	0,2181	0,2182	0,2097	0,1566	0,1413	0,1789	0,1823
N21	Hillvalley	0,3914	0,3822	0,4019	0,3801	0,4212	0,4217	0,4204	0,3898



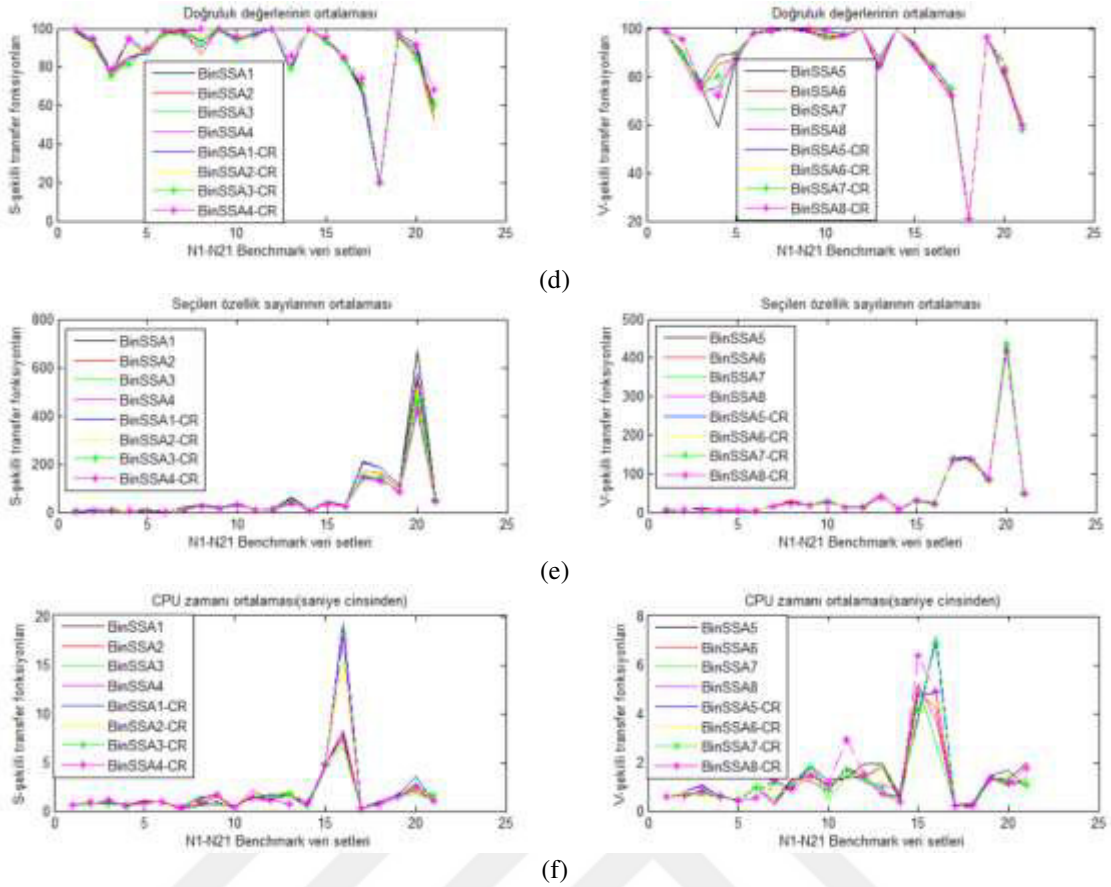
(a)



(b)



(c)



Şekil 4.15. BinSSA ve BinSSA-CR varyasyonları için yakınsama grafikleri: (a) uygunluk değerlerinin ortalaması, (b) uygunluk değerlerinin standart sapması, (c) uygunluk değerlerinin en iyisi, (d) doğruluk değerlerinin ortalaması, (e) seçilen özellik sayılarının ortalaması ve (f) CPU zamanı ortalaması (saniye cinsinden)

BinSSA varyasyonları için Wilcoxon signed-rank testi:

BinSSA ve BinSSA-CR varyasyonları, popülasyon büyüklüğü = 50 ve maksimum iterasyon = 100 parametre değerleri ile çeşitli düşük, orta ve büyük boyutlu karşılaştırma veri kümelerinde 20 kez bağımsız çalıştırılmış ve her bir karşılaştırma için her varyasyonda 20 deneme yapılmıştır. Sonuç olarak 20 örnekli veri kümeleri elde edilmiştir. Bu veri kümeleri üzerinde BinSSA ve BinSSA-CR varyasyonları arasında kayda değer bir fark olup olmadığı Wilcoxon signed-rank testi uygulanarak bakılmıştır. p değerleri Matlab R2014a yazılımı ile hesaplanmıştır.

En yüksek performanslı BinSSA ve BinSSA-CR varyasyonu önceki bölümde anlatıldığı gibi beş karşılaştırma ölçütüne göre belirlenmiştir. Sonuçlar, BinSSA-CR varyasyonlarının keşfetme ve sömürü arasında BinSSA varyasyonlarından daha başarılı bir denge kurduğunu göstermektedir. Çaprazlama operatörü BinSSA'nın keşfetme ve

sömürü yeteneklerini geliştirmiştir. BinSSA-CR varyantları arasında BinSSA4-CR varyasyonu uygunluk değerlerinin ortalaması, uygunluk değerlerinin standart sapması, doğruluk sonuçlarının ortalaması ve seçilen özelliklerin ortalama sayısı için üstün performans göstermektedir. BinSSA4-CR, beş karşılaştırma kriterinden dördü için yüksek performans göstermiştir. Çizelgeler 4.103 ve 4.104’de BinSSA4-CR uygunluk sonuçlarının Wilcoxon signed-rank testinin p -değerlerini göstermektedir.

Çizelge 4.103. BinSSA4-CR varyasyonu ve BinSSA varyasyonlarının uygunluk fonksiyonu sonuçları üzerinde Wilcoxon signed-rank testi sonuçları ($p \leq 0.05$ olan değerlerin altı çizilmiştir.)

ID	Benchmark	BinSSA1	BinSSA2	BinSSA3	BinSSA4	BinSSA5	BinSSA6	BinSSA7	BinSSA8
N1	<i>Wine</i>	<u>0.0026</u>	<u>0.0235</u>	0,1984	0,1150	0,1511	0,2610	0,1946	0,7213
N2	<i>Hepatitis</i>	<u>0.0001</u>	<u>0.0002</u>	<u>4.22e-05</u>	<u>0.00080</u>	1	0,1756	0,0369	0,3193
N3	<i>Vehicle</i>	<u>6.91e-07</u>	<u>4.60e-08</u>	<u>7.94e-07</u>	<u>7.63e-06</u>	<u>4.56e-08</u>	<u>3.91e-08</u>	<u>1.30e-07</u>	<u>2.43e-08</u>
N4	<i>Zoo</i>	<u>0.00075</u>	0,0762	<u>0.0025</u>	<u>0.01469</u>	0,6694	0,9016	0,1458	0,7937
N5	<i>Heart</i>	<u>1.21e-06</u>	<u>1.65e-05</u>	<u>0.00064</u>	<u>7.74e-05</u>	<u>1.68e-06</u>	<u>1.19e-06</u>	<u>7.20e-07</u>	<u>0.013</u>
N6	<i>Breast Cancer</i>	<u>0.01878</u>	<u>0.0080</u>	0,1343	0,1649	0,0969	0,2140	<u>0.0172</u>	<u>0.0200</u>
N7	<i>Ionosphere</i>	<u>0.0172</u>	0,3230	0,5329	<u>0.0164</u>	<u>0.0045</u>	<u>0.0370</u>	<u>0.0496</u>	0,6452
N8	<i>Lung Cancer</i>	<u>7.95e-09</u>	<u>7.95e-09</u>	<u>7.77e-09</u>	<u>7.88e-09</u>	<u>5.08e-05</u>	<u>1.34e-07</u>	<u>4.35e-09</u>	<u>1.16e-07</u>
N9	<i>Dermatology</i>	<u>0.0010</u>	<u>0.0022</u>	0,1127	0,1386	0,4522	0,4316	0,4162	0,4475
N10	<i>Sonar</i>	<u>0.0013</u>	<u>1.85e-05</u>	<u>0.0228</u>	0,4504	0,1577	0,6706	<u>0.0015</u>	<u>0.0003</u>
N11	<i>BreastEW</i>	<u>0.0086</u>	0,5421	<u>0.0299</u>	0,4429	0,4686	0,1373	1	0,4690
N12	<i>Soybean Small</i>	<u>4.74e-07</u>	<u>8.54e-07</u>	0,1192	0,9327	<u>3.09e-08</u>	<u>2.00e-08</u>	<u>2.40e-08</u>	<u>1.61e-08</u>
N13	<i>Movementlibras</i>	<u>2.11e-06</u>	<u>3.88e-05</u>	<u>6.96e-05</u>	0,0592	0,0826	0,8590	0,8375	0,0611
N14	<i>Parkinsons</i>	<u>1.10e-07</u>	<u>1.70e-06</u>	<u>0.0204</u>	<u>0.0108</u>	<u>7.38e-09</u>	<u>6.59e-06</u>	<u>3.44e-07</u>	<u>1.06e-08</u>
N15	<i>Spambase</i>	<u>0.0423</u>	0,1163	<u>0.0423</u>	<u>0.0006</u>	<u>0.0009</u>	<u>0.0025</u>	<u>0.0001</u>	<u>0.0003</u>
N16	<i>Waveform</i>	1	0,1429	0,2063	<u>0.0079</u>	<u>0.0317</u>	0,3175	<u>0.0159</u>	<u>0.0079</u>
N17	<i>Arrhythmia</i>	<u>8.89e-08</u>	<u>1.78e-08</u>	<u>5.54e-08</u>	<u>8.90e-08</u>	<u>7.40e-07</u>	<u>1.66e-07</u>	<u>1.21e-07</u>	<u>1.22e-07</u>
N18	<i>Semeion</i>	<u>6.00e-08</u>	<u>7.81e-07</u>	<u>8.21e-08</u>	<u>3.73e-08</u>	<u>5.84e-07</u>	<u>9.26e-06</u>	<u>1.37e-05</u>	<u>2.791e-07</u>
N19	<i>Clean</i>	<u>6.72e-06</u>	<u>0.0005</u>	<u>0.0002</u>	<u>0.0004</u>	<u>0.0427</u>	<u>0.0094</u>	<u>0.0231</u>	<u>0.0004</u>
N20	<i>CNAE</i>	0,1352	<u>0.0195</u>	<u>0.0351</u>	<u>0.0025</u>	<u>0.0001</u>	<u>0.0001</u>	<u>0.0001</u>	<u>0.0001</u>
N21	<i>Hillvalley</i>	<u>2.29e-08</u>	<u>2.30e-08</u>	<u>7.65e-05</u>	<u>5.83e-07</u>	<u>0.0008</u>	0,5638	<u>8.85e-07</u>	<u>0.0015</u>

Çizelge 4.104. BinSSA4-CR varyasyonu ve BinSSA-CR varyasyonlarının uygunluk fonksiyonu sonuçları üzerinde Wilcoxon signed-rank testi sonuçları ($p \leq 0.05$ olan değerlerin altı çizilmiştir.)

ID	Benchmark	BinSSA1-CR	BinSSA2-CR	BinSSA3-CR	BinSSA5-CR	BinSSA6-CR	BinSSA7-CR	BinSSA8-CR
N1	<i>Wine</i>	<u>0.0014</u>	<u>0.0131</u>	<u>0.0294</u>	0,9786	0,9895	0,8421	0,6798
N2	<i>Hepatitis</i>	<u>6.65e-06</u>	<u>0.00020</u>	<u>0.0368</u>	0,8485	0,1989	0,1723	0,0277
N3	<i>Vehicle</i>	<u>4.48e-07</u>	<u>3.87e-07</u>	<u>3.34e-08</u>	<u>0.0023</u>	<u>0.00092</u>	<u>5.34e-05</u>	<u>3.29e-05</u>
N4	<i>Zoo</i>	<u>0.00028</u>	<u>0.017</u>	<u>0.008</u>	0,2405	0,2590	0,5189	0,3488
N5	<i>Heart</i>	<u>1.33e-06</u>	<u>3.23e-05</u>	<u>9.28e-06</u>	<u>0.0030</u>	<u>0.0011</u>	<u>8.13e-05</u>	<u>9.63e-05</u>
N6	<i>Breast Cancer</i>	0,0504	<u>0.0462</u>	0,1574	<u>0.0397</u>	0,05996	0,1380	0,4229
N7	<i>Ionosphere</i>	<u>0.0228</u>	0,6744	0,4981	<u>0.0010</u>	<u>5.62e-06</u>	<u>3.19e-05</u>	<u>0.0034</u>
N8	<i>Lung Cancer</i>	<u>7.96e-09</u>	<u>7.92e-09</u>	<u>7.68e-09</u>	<u>6.47e-09</u>	<u>5.16e-09</u>	<u>6.11e-09</u>	<u>5.28e-09</u>
N9	<i>Dermatology</i>	<u>0.0008</u>	<u>0.0410</u>	0,5568	1	<u>7.46e-05</u>	<u>1.71e-05</u>	<u>0.0003</u>
N10	<i>Sonar</i>	<u>8.41e-07</u>	<u>1.45e-06</u>	<u>0.0120</u>	<u>0.0015</u>	<u>0.0219</u>	1	<u>1.38e-06</u>
N11	<i>BreastEW</i>	0,2374	0,3603	<u>0.0245</u>	0,7898	0,4693	0,2374	0,3036
N12	<i>Soybean Small</i>	<u>1.43e-07</u>	<u>3.76e-05</u>	<u>0.0267</u>	<u>3.09e-08</u>	<u>2.00e-08</u>	<u>1.61e-08</u>	<u>2.77e-08</u>
N13	<i>Movementlibras</i>	<u>8.75e-05</u>	<u>1.88e-05</u>	<u>0.0017</u>	0,5030	0,5385	0,6918	0,8375
N14	<i>Parkinsons</i>	<u>1.89e-07</u>	<u>6.49e-06</u>	<u>6.96e-05</u>	<u>1.38e-08</u>	<u>3.85e-07</u>	<u>3.85e-07</u>	<u>1.20e-08</u>
N15	<i>Spambase</i>	0,4212	<u>0.0195</u>	<u>0.0010</u>	0,0995	<u>0.0238</u>	<u>0.0003</u>	<u>0.0238</u>
N16	<i>Waveform</i>	0,4048	0,1429	0,2063	0,4206	0,8413	1	0,1508
N17	<i>Arrhythmia</i>	<u>7.617e-08</u>	<u>5.54e-08</u>	<u>2.92e-08</u>	<u>1.52e-06</u>	<u>5.52e-07</u>	<u>1.51e-06</u>	<u>1.72e-05</u>
N18	<i>Semeion</i>	<u>6.00e-08</u>	<u>1.12e-07</u>	<u>2.40e-07</u>	<u>9.67e-05</u>	<u>1.38e-06</u>	<u>3.18e-06</u>	<u>5.84e-07</u>
N19	<i>Clean</i>	<u>0.0002</u>	<u>0.0003</u>	<u>0.0001</u>	0,1694	0,1455	0,7338	0,7960
N20	<i>CNAE</i>	0,0508	0,2351	0,1352	<u>0.0015</u>	0,0508	<u>0.0001</u>	<u>0.0005</u>
N21	<i>Hillvalley</i>	<u>1.96e-08</u>	<u>1.95e-08</u>	<u>1.96e-08</u>	<u>5.84e-07</u>	<u>7.66e-05</u>	<u>8.60e-05</u>	0,0834

Wilcoxon signed-rank testi BinSSA4-CR varyasyonu ile diğer BinSSA ve BinSSA-CR varyasyonlarına uygulanmıştır. Elde edilen sonuçlara göre, p -değerleri çoğu durum için 0,05'in altındadır. Bu nedenle, BinSSA4-CR sonuçlarındaki gelişmeler, bu algoritmanın etkinliğini veri kümelerinin çoğunda doğrulamıştır ve diğer varyasyonlardan istatistiksel olarak üstünlüğünü göstermiştir.

Literatürdeki algoritmalar ile BinSSA ve BinSSA-CR varyasyonlarının karşılaştırılması:

Literatürle kıyaslama çalışması 1:

BinSSA4-CR varyasyonunun performansı literatürün iyi bilinen sezgisel algoritmaları ile karşılaştırılmıştır. Bu algoritmalar, Parçacık Sürüsü Optimizasyonu (Chuang ve ark., 2008), Yarasa Algoritması (Mirjalili ve ark., 2014) ve Gri Kurt Optimizasyonu (Emary ve ark., 2016)'dır. Tüm deneyler, adil bir karşılaştırma için aynı parametrelere ve koşullara göre yapılmıştır. BinSSA parametreleri ve karşılaştırma algoritmaları çizelge 4.105'de gösterilmiştir.

Çizelge 4.105. BinSSA ve karşılaştırma algoritmaları için parametre ayarları

Algoritmalar	Parametreler	Değerler
BBA (Mirjalili ve ark., 2014)	Q _{min}	0
	Q _{max}	2
	A (Gürültü)	0,5
	r (Pulse oranı)	0,5
BPSO (Chuang ve ark., 2008)	w (Atalet ağırlığı)	2
	w _{max}	0,9
	w _{min}	0,4
	c1	2
	c2	2
BGWO (Emary ve ark., 2016)	α	[2 0]
BinSSA4 (James ve Li, 2015a) ve BinSSA4-CR	r _a ; p _c ; p _m	1; 0.7; 0.1

Çizelge 4.106'da karşılaştırma algoritmaları ile BinSSA4-CR arasındaki uygunluk değerlerinin ortalaması karşılaştırılmıştır. Uygunluk sonuçlarının ortalamasına göre, BinSSA4-CR varyasyonu N8, N12, N16 ve N20 hariç 21 veri kümesinden 17'si için kıyaslama veri kümelerinin % 80,95'inde üstün performans göstermektedir. BBA, BinSSA4-CR'den sonra en iyi performans gösteren ikinci algoritmadır. BBA, 21 veri kümesinden 4'ünde (N8, N12, N16 ve N20) iyi performans göstermiştir. BinSSA4-

CR'nin yüksek performansındaki en önemli faktör çaprazlama operatörüdür. BinSSA4-CR varyasyonu keşfetme ve sömürü arasında daha istikrarlı bir denge kurmuştur.

Çizelge 4.107'de karşılaştırma algoritmaları ile BinSSA4-CR arasındaki uygunluk değerlerinin standart sapması karşılaştırılmıştır. Uygunluk sonuçlarının standart sapmasına göre, BinSSA4-CR varyasyonu N8, N12 ve N20 hariç 21 veri setinden 18'i için karşılaştırma veri setlerinin % 85,71'inde üstün performans göstermektedir. BBA, BinSSA4-CR'den sonra en iyi performans gösteren ikinci algoritmadır. BBA, 21 veri kümesinden 2'sinde (N8 ve N12) iyi performans göstermiştir. BPSO ve BinSSA4, 21 veri kümesinden 1'inde iyi performans göstermektedir. Sonuçlar BinSSA4-CR'nin tatmin edici düzeyde standart sapma değerleri ile üstün çözümler bulduğunu göstermektedir.

Çizelge 4.108'de karşılaştırma algoritmaları ile BinSSA4-CR arasındaki uygunluk değerlerinin en iyi değeri karşılaştırılmıştır. En iyi uygunluk değerine göre, BinSSA4-CR varyasyonu 21 veri kümesinden 6'sı için kıyaslama veri kümelerinin % 29'unda iyi performans gösterirken, BBA 21 veri kümesinin 11'i için kıyaslama veri kümelerinin % 52'sinde iyi performans göstermiştir.

Çizelge 4.109'da karşılaştırma algoritmaları ile BinSSA4-CR arasındaki ortalama sınıflandırıcının doğruluk değerleri ortalama sınıflandırıcının doğruluk değerleri karşılaştırılmıştır. Sınıflandırma sonuçlarının doğruluğuna göre, BinSSA4-CR varyasyonu N4, N16, N18 ve N20 hariç 21 veri kümesinden 17'si için kıyaslama veri kümelerinin % 80,95'inde üstün performans göstermektedir. BinSSA4-CR'nin doğruluk değerinin maksimum ortalaması 100'dür ve BinSSA4-CR'nin doğruluk değerinin minimum ortalaması 20'dir. BBA, BinSSA4-CR'den sonra en iyi ikinci doğruluk ortalamasına sahip algoritmadır. BBA, 21 veri kümesinden 5'i için iyi performans göstermektedir. Ardından sırasıyla BinSSA4, BGWO ve BPSO iyi performans göstermektedir. BinSSA4-CR varyasyonu N1, N8, N9, N12 ve N14 karşılaştırma veri kümeleri için % 100 doğruluk değeri elde etmiştir.

Seçilen özelliklerin ortalama sayısı çizelge 4.110'da gösterilmiştir. Seçilen özelliklerin ortalama sayısına göre BinSSA4-CR, 21 veri kümesinden 20'si için karşılaştırma veri kümelerinin % 95,24'ünde daha az özellik seçmiştir.

Ortalama CPU zamanı sonuçları çizelge 4.111'de gösterilmiştir. Ortalama CPU zamanı sonuçlarına göre, BBA; N14, N16, N17, N18 ve N20 hariç 21 veri kümesinin 16'sı için kıyaslama veri kümelerinin % 76,19'unda en kısa CPU süresine sahiptir.

Çizelge 4.106. Uygunluk sonuçlarının ortalamasına göre, BinSSA4-CR ile karşılaştırma algoritmalarının sonuçları

ID	Benchmark	BinSSA4-CR	BinSSA4	BBA	BGWO	BPSO
		Ortalama	Ortalama	Ortalama	Ortalama	Ortalama
N1	<i>Wine</i>	0,0072	0,0145	0,0208	0,0178	0,0179
N2	<i>Hepatitis</i>	0,0625	0,0924	0,0680	0,0861	0,0638
N3	<i>Vehicle</i>	0,2295	0,2322	0,7638	0,7843	0,7682
N4	<i>Zoo</i>	0,0080	0,0210	0,0187	0,0244	0,0090
N5	<i>Heart</i>	0,1099	0,1180	0,8784	0,8987	0,8799
N6	<i>Breast Cancer</i>	0,0200	0,0205	0,6802	0,6817	0,6748
N7	<i>Ionosphere</i>	0,0299	0,0352	0,0302	0,0652	0,0484
N8	<i>Lung Cancer</i>	0,0029	0,0372	0,0014	0,0788	0,0445
N9	<i>Dermatology</i>	0,0072	0,0090	0,2841	0,3388	0,3006
N10	<i>Sonar</i>	0,0394	0,0407	0,9497	0,9723	0,9657
N11	<i>BreastEW</i>	0,0295	0,0364	0,0392	0,0465	0,0395
N12	<i>Soybean Small</i>	0,0025	0,0025	0,0006	0,0024	0,0016
N13	<i>Movementlibras</i>	0,1584	0,1877	0,1609	0,1930	0,1877
N14	<i>Parkinsons</i>	0,0020	0,0022	0,0533	0,0617	0,0414
N15	<i>Spambase</i>	0,0630	0,0731	0,0639	0,0688	0,0750
N16	<i>Waveform</i>	0,1604	0,1690	0,1451	0,1584	0,1632
N17	<i>Arrhythmia</i>	0,2520	0,3198	0,2621	0,3146	0,3177
N18	<i>Semeion</i>	0,7574	0,7970	0,7575	0,7584	0,7789
N19	<i>Clean</i>	0,0164	0,0344	0,0280	0,0452	0,0682
N20	<i>CNAE</i>	0,1239	0,1570	0,0648	0,0774	0,1325
N21	<i>Hillvalley</i>	0,3677	0,4310	0,3680	0,3870	0,3947

Çizelge 4.107. Uygunluk sonuçlarının standart sapmasına göre, BinSSA4-CR ile karşılaştırma algoritmalarının sonuçları

ID	Benchmark	BinSSA4-CR	BinSSA4	BBA	BGWO	BPSO
		Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.	Std.Dev.
N1	<i>Wine</i>	0,0102	0,0186	0,0184	0,0224	0,0206
N2	<i>Hepatitis</i>	0,0175	0,0302	0,0451	0,0392	0,0357
N3	<i>Vehicle</i>	0,0046	0,0232	0,0283	0,0155	0,0211
N4	<i>Zoo</i>	0,0157	0,0369	0,0333	0,0407	0,0242
N5	<i>Heart</i>	0,0172	0,0380	0,0229	0,0191	0,0244
N6	<i>Breast Cancer</i>	0,0104	0,0109	0,0236	0,0282	0,0267
N7	<i>Ionosphere</i>	0,0150	0,0157	0,0217	0,0325	0,0208
N8	<i>Lung Cancer</i>	0,0680	0,0862	0,0006	0,0842	0,0733
N9	<i>Dermatology</i>	0,0059	0,0080	0,0447	0,0420	0,0320
N10	<i>Sonar</i>	0,0118	0,0283	0,0148	0,0147	0,0135
N11	<i>BreastEW</i>	0,0083	0,0121	0,0162	0,0172	0,0160
N12	<i>Soybean Small</i>	0,0003	0,0003	0,0001	0,0006	0,0003
N13	<i>Movementlibras</i>	0,0355	0,0528	0,0539	0,0359	0,0487
N14	<i>Parkinsons</i>	0,0003	0,0003	0,0224	0,0299	0,0300
N15	<i>Spambase</i>	0,0026	0,0063	0,0062	0,0083	0,0049
N16	<i>Waveform</i>	0,0014	0,0038	0,0090	0,0083	0,0066
N17	<i>Arrhythmia</i>	0,0289	0,0543	0,0297	0,0386	0,0586
N18	<i>Semeion</i>	0,0086	0,0115	0,0167	0,0167	0,0154
N19	<i>Clean</i>	0,0134	0,0137	0,0212	0,0270	0,0226
N20	<i>CNAE</i>	0,0174	0,0220	0,0144	0,0178	0,0142
N21	<i>Hillvalley</i>	0,0055	0,0445	0,0399	0,0389	0,0240

Çizelge 4.108. Uygunluk sonuçlarının en iyi değerine göre, BinSSA4-CR ile karşılaştırma algoritmalarının sonuçları

ID	Benchmark	BinSSA4-CR	BinSSA4	BBA	BGWO	BPSO
		En iyi	En iyi	En iyi	En iyi	En iyi
N1	Wine	0,0015	0,0023	0,0023	0,0015	0,0023
N2	Hepatitis	0,0021	0,0367	0,0021	0,0042	0,0032
N3	Vehicle	0,1719	0,1777	0,6940	0,7635	0,7314
N4	Zoo	0,0013	0,0025	0,0006	0,0006	0,0006
N5	Heart	0,0596	0,0405	0,8296	0,8655	0,8273
N6	Breast Cancer	0,0021	0,0022	0,6335	0,6203	0,6357
N7	Ionosphere	0,0032	0,0174	0,0006	0,0180	0,0029
N8	Lung Cancer	0,0029	0,0034	0,0004	0,0036	0,0025
N9	Dermatology	0,0029	0,0038	0,2199	0,2638	0,2347
N10	Sonar	0,0058	0,0047	0,9203	0,9467	0,9459
N11	BreastEW	0,0235	0,0228	0,0106	0,0235	0,0119
N12	Soybean Small	0,0020	0,0017	0,0003	0,0011	0,0011
N13	Movementlibras	0,1290	0,1147	0,0586	0,1294	0,1279
N14	Parkinsons	0,0018	0,0018	0,0267	0,0027	0,0014
N15	Spambase	0,0515	0,0654	0,0490	0,0617	0,0682
N16	Waveform	0,1595	0,1639	0,1258	0,1461	0,1488
N17	Arrhythmia	0,1919	0,2249	0,2126	0,2595	0,2244
N18	Semeion	0,7474	0,7711	0,7188	0,7156	0,7466
N19	Clean	0,0048	0,0155	0,0036	0,0070	0,0360
N20	CNAE	0,0575	0,0995	0,0414	0,0399	0,1060
N21	Hillvalley	0,3207	0,3506	0,2405	0,3180	0,3639

Çizelge 4.109. Ortalama sınıflandırma doğruluğuna göre, BinSSA4-CR ile karşılaştırma algoritmalarının sonuçları

ID	Benchmark	BinSSA4-CR	BinSSA4	BBA	BGWO	BPSO
N1	Wine	100	100	98,19	98,61	98,47
N2	Hepatitis	94,65	94,63	93,33	91,67	93,83
N3	Vehicle	78,34	75,6	23,25	21,39	22,84
N4	Zoo	95	84	98,25	97,75	94,25
N5	Heart	89,26	87,41	11,57	9,63	11,39
N6	Breast Cancer	98,55	97,39	31,74	31,63	32,28
N7	Ionosphere	98,29	98	97,14	93,86	95,43
N8	Lung Cancer	100	100	100	92,50	95,83
N9	Dermatology	100	100	71,64	66,37	70,07
N10	Sonar	96	95	4,17	2,143	2,74
N11	BreastEW	97	95,8	96,15	95,65	96,25
N12	Soybean Small	100	100	100	100	100
N13	Movementlibras	85,33	80,83	84,10	81,11	81,46
N14	Parkinsons	100	100	94,74	94,10	96,03
N15	Spambase	95,18	92,58	94,50	93,80	92,95
N16	Waveform	85,20	84,67	85,91	84,76	84,10
N17	Arrhythmia	74,11	70,11	73,94	68,89	68,39
N18	Semeion	20,2	19,73	24,11	24,98	21,82
N19	Clean	97,75	96,5	97,58	96,11	93,58
N20	CNAE	91,50	86,25	93,96	92,99	87,13
N21	Hillvalley	68,17	61,67	68,14	61,57	60,58

Çizelge 4.110. Ortalama seçilen özellik sayısına göre, BinSSA4-CR ile karşılaştırma algoritmalarının sonuçları (virgüllü sayılar yukarıya yuvarlanmıştır)

ID	Benchmark	BinSSA4-CR	BinSSA4	BBA	BGWO	BPSO
N1	<i>Wine</i>	4	5	5	6	5
N2	<i>Hepatitis</i>	7	9	7	8	7
N3	<i>Vehicle</i>	7	8	9	11	8
N4	<i>Zoo</i>	5	6	6	6	7
N5	<i>Heart</i>	4	5	6	6	5
N6	<i>Breast Cancer</i>	4	4	4	4	4
N7	<i>Ionosphere</i>	13	14	17	15	14
N8	<i>Lung Cancer</i>	23	24	28	25	28
N9	<i>Dermatology</i>	15	17	17	20	16
N10	<i>Sonar</i>	27	28	27	28	29
N11	<i>BreastEW</i>	11	12	13	20	17
N12	<i>Soybean Small</i>	13	14	14	18	16
N13	<i>Movementlibras</i>	39	40	39	54	40
N14	<i>Parkinsons</i>	7	8	8	9	8
N15	<i>Spambase</i>	32	33	37	43	39
N16	<i>Waveform</i>	22	23	22	30	23
N17	<i>Arrhythmia</i>	140	147	146	144	142
N18	<i>Semeion</i>	130	137	133	152	134
N19	<i>Clean</i>	84	87	87	111	86
N20	<i>CNAE</i>	422	421	428	482	433
N21	<i>Hillvalley</i>	45	49	49	65	51

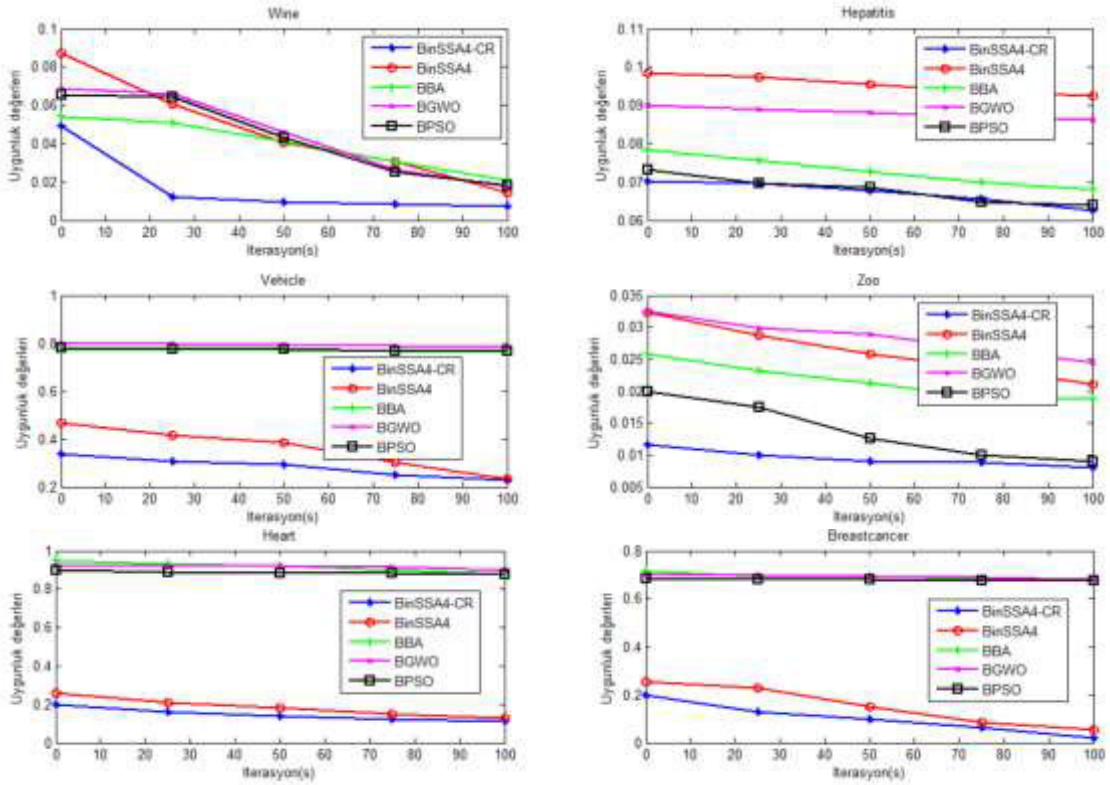
Çizelge 4.111. Ortalama CPU zamanına göre, BinSSA4-CR ile karşılaştırma algoritmalarının sonuçları

ID	Benchmark	BinSSA4-CR	BinSSA4	BBA	BGWO	BPSO
N1	<i>Wine</i>	0,011804	0,011928	0,006232	0,025878	0,006468
N2	<i>Hepatitis</i>	0,018260	0,013696	0,006174	0,022520	0,006726
N3	<i>Vehicle</i>	0,022236	0,013732	0,009508	0,015736	0,010780
N4	<i>Zoo</i>	0,011426	0,011524	0,006056	0,011288	0,006804
N5	<i>Heart</i>	0,016180	0,015992	0,007698	0,013500	0,008720
N6	<i>Breast Cancer</i>	0,019138	0,020042	0,007352	0,013072	0,008340
N7	<i>Ionosphere</i>	0,007896	0,003402	0,006600	0,011516	0,007030
N8	<i>Lung Cancer</i>	0,017774	0,015652	0,005918	0,011016	0,006568
N9	<i>Dermatology</i>	0,033384	0,016282	0,006332	0,011754	0,006686
N10	<i>Sonar</i>	0,008618	0,008934	0,008548	0,013046	0,008872
N11	<i>BreastEW</i>	0,027412	0,026352	0,006556	0,039102	0,006986
N12	<i>Soybean Small</i>	0,023690	0,018920	0,006048	0,036562	0,006230
N13	<i>Movementlibras</i>	0,014222	0,040802	0,006962	0,024082	0,007550
N14	<i>Parkinsons</i>	0,016594	0,016344	0,006214	0,011330	0,006160
N15	<i>Spambase</i>	0,094780	0,094552	0,094048	0,177964	0,095612
N16	<i>Waveform</i>	0,352526	0,166894	0,084970	0,182684	0,080888
N17	<i>Arrhythmia</i>	0,006302	0,006368	0,011364	0,024350	0,013270
N18	<i>Semeion</i>	0,014158	0,012196	0,060226	0,157566	0,061514
N19	<i>Clean</i>	0,029682	0,029596	0,008800	0,021510	0,009132
N20	<i>CNAE</i>	0,048692	0,048600	0,106238	0,351940	0,102870
N21	<i>Hillvalley</i>	0,020512	0,020264	0,008550	0,028580	0,008680

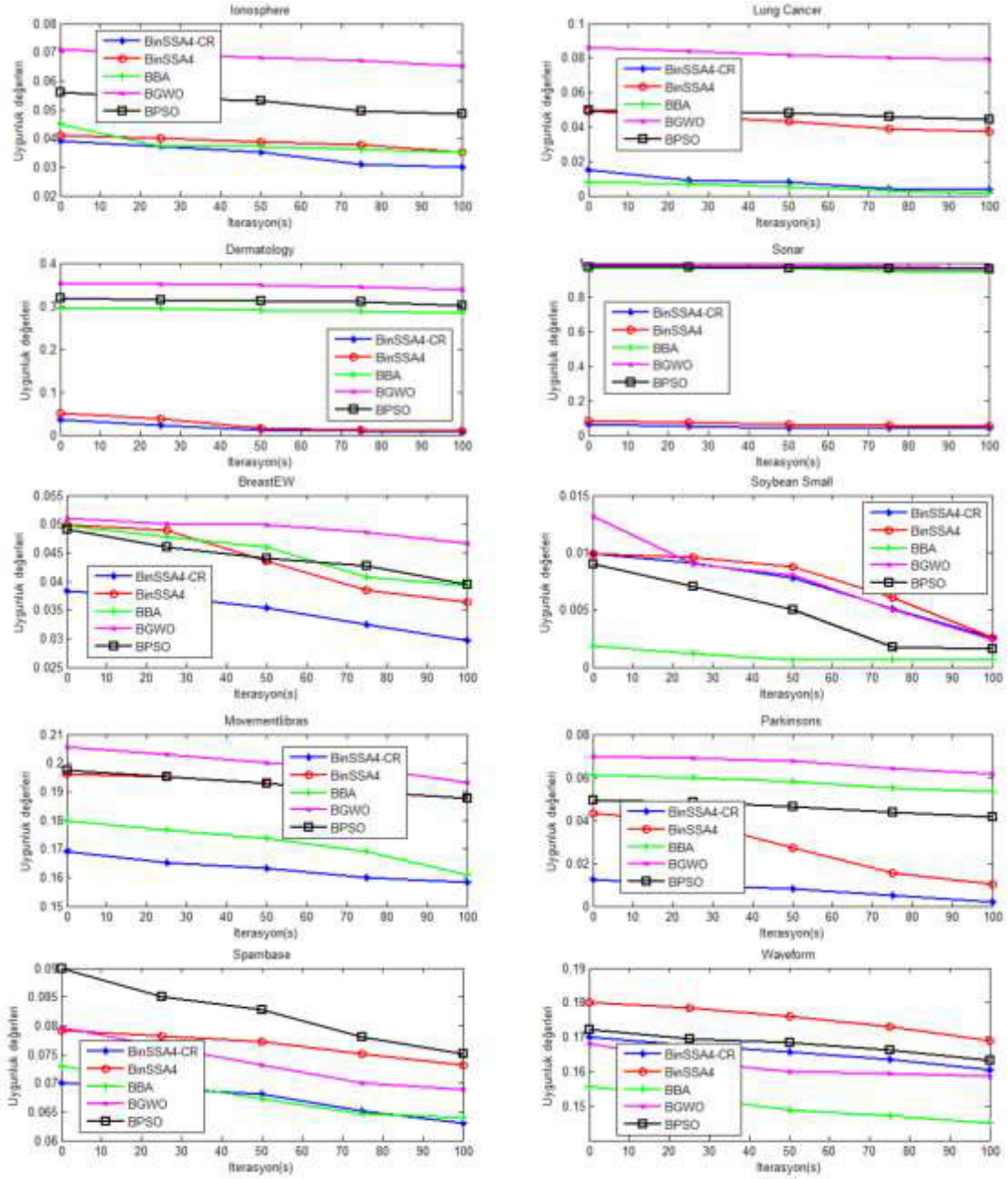
Çizelgeler 4.106 - 4.111 sonuçlarına bakıldığında, BinSSA4-CR algoritmasının özellik seçimi problemi için yüksek performans gösterdiği rahatlıkla görülebilmektedir. BinSSA4-CR'den sonra ikinci en iyi karşılaştırma algoritması BBA'dır. BinSSA4-CR bu başarıyı çaprazlama operatörü sayesinde elde etmiştir. Bu sayede BinSSA4-CR'nin

ikili arama uzayındaki keşif ve sömürü kabiliyeti geliştirilmiştir. Bu nedenle, diğer karşılaştırma algoritmalarından daha avantajlıdır.

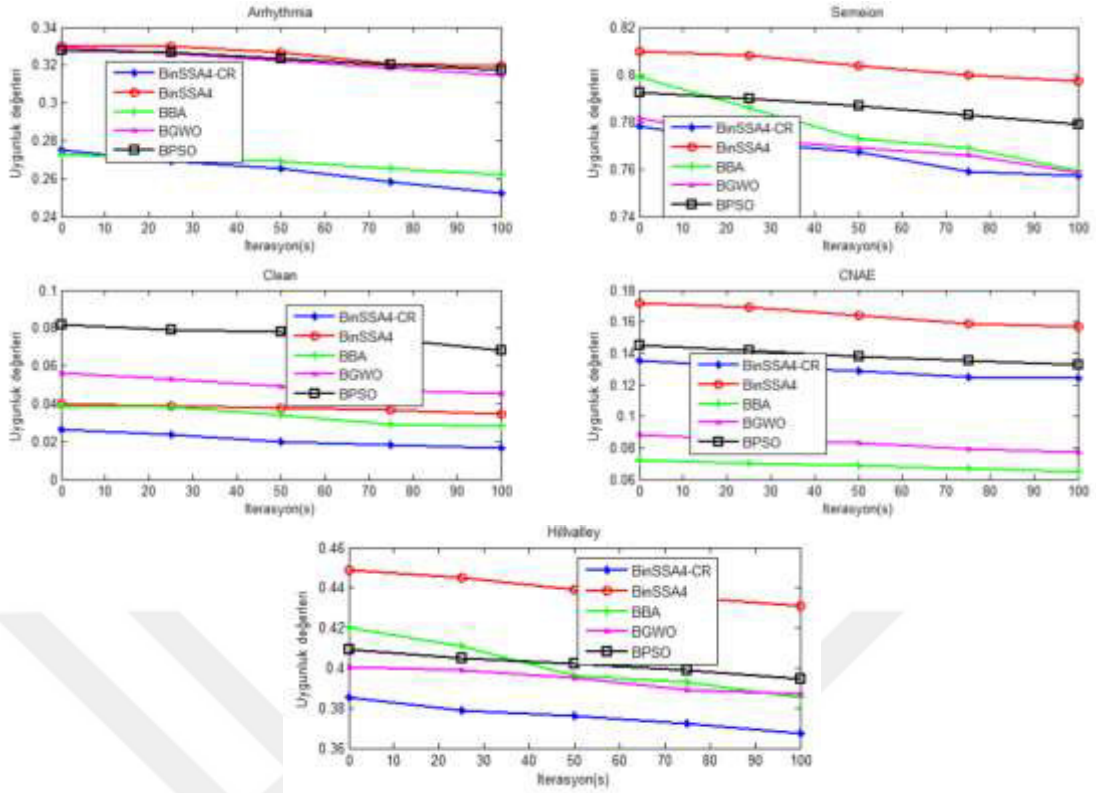
Şekil 4.16 – 4.18’de sırasıyla düşük ölçekli, orta ölçekli ve büyük ölçekli veri kümeleri için BinSSA4-CR, BinSSA4, BBA, BGWO ve BPSO için yakınsama grafikleri gösterilmiştir. Yakınsama grafiklerine göre, BinSSA4-CR; N12, N16, N18 ve N19 hariç 21 veri kümesinden 17’si için kıyaslama veri kümelerinin % 81’inde tüm algoritmalarından daha iyi performans göstermiştir.



Şekil 4.16. BinSSA4-CR, BinSSA4, BBA, BGWO ve BPSO’nun düşük ölçekli veri setlerindeki yakınsama grafikleri



Şekil 4.17. BinSSA4-CR, BinSSA4, BBA, BGWO ve BPSO'nun orta ölçekli veri setlerindeki yakınsama grafikleri



Şekil 4.18. BinSSA4-CR, BinSSA4, BBA, BGWO ve BPSO'nun geniş ölçekli veri setlerindeki yakınsama grafikleri

BinSSA4-CR ile BBA, BGWO ve BPSO arasında Wilcoxon signed-rank testi uygulaması:

Wilcoxon signed-rank testi BBA, BGWO ve BPSO ile BinSSA4-CR varyasyonuna uygulanmıştır. Çizelge 4.112'de BinSSA4-CR, BBA, BGWO ve BPSO için Wilcoxon signed-rank testinin uygulanmasıyla elde edilen sonuçlar gösterilmiştir. Elde edilen sonuçlara göre, p -değerleri çoğu vaka için 0.05'in altındadır. Bu nedenle, BinSSA4-CR sonuçlarındaki iyileşmeler, veri kümelerinin çoğunda BinSSA'nın etkinliğini göstermiştir. Bu sonuçlara göre, BinSSA4-CR'nin diğer son teknoloji algoritmalara göre üstün bir performansa sahip olduğu görülebilmektedir.

Çizelge 4.112. BinSSA4-CR, BBA, BGWO ve BPSO arasında Wilcoxon signed-rank testi uygulama sonuçları ($p \leq 0,05$ değerlerinin altı çizilmiştir.)

ID	Benchmark	BinSSA4-CR & BBA	BinSSA4-CR & BGWO	BinSSA4-CR & BPSO
N1	<i>Wine</i>	<u>4,6245E-04</u>	<u>0,0110</u>	0,6405
N2	<i>Hepatitis</i>	<u>2,9114E-06</u>	0,5940	0,1092
N3	<i>Vehicle</i>	<u>2,4387E-07</u>	<u>2,3998E-08</u>	<u>2,3387E-08</u>
N4	<i>Zoo</i>	0,2598	0,1498	0,4191
N5	<i>Heart</i>	<u>4,9294E-08</u>	<u>4,8004E-08</u>	<u>4,5784E-08</u>
N6	<i>Breast Cancer</i>	<u>6,3671E-08</u>	<u>6,2771E-08</u>	<u>6,2059E-08</u>
N7	<i>Ionosphere</i>	0,6947	<u>2,8352E-05</u>	<u>0,0176</u>
N8	<i>Lung Cancer</i>	<u>7,8321E-09</u>	<u>7,7461E-09</u>	<u>1,0664E-06</u>
N9	<i>Dermatology</i>	<u>6,4673E-08</u>	<u>6,3761E-08</u>	<u>6,3761E-08</u>
N10	<i>Sonar</i>	<u>4,4377E-08</u>	<u>4,3524E-08</u>	<u>4,3007E-08</u>
N11	<i>BreastEW</i>	0,5948	<u>0,0227</u>	0,1167
N12	<i>Soybean Small</i>	<u>5,3345E-08</u>	0,1556	<u>1,0408E-07</u>
N13	<i>Movementlibras</i>	0,7326	<u>3,6081E-04</u>	<u>0,0308</u>
N14	<i>Parkinsons</i>	<u>2,4192E-08</u>	<u>2,3577E-08</u>	<u>1,0394E-06</u>
N15	<i>Spambase</i>	0,0724	<u>4,1425E-08</u>	<u>8,1257E-08</u>
N16	<i>Waveform</i>	<u>4,3459E-08</u>	<u>2,6568E-07</u>	0,0563
N17	<i>Arrhythmia</i>	<u>2,0140E-06</u>	<u>1,0394E-07</u>	<u>1,4197E-07</u>
N18	<i>Semeion</i>	0,1635	0,3315	<u>5,6388E-04</u>
N19	<i>Clean</i>	<u>4,5562E-06</u>	<u>8,4982E-06</u>	<u>1,4084E-07</u>
N20	<i>CNAE</i>	<u>4,4179E-08</u>	<u>4,4642E-08</u>	<u>0,1148</u>
N21	<i>Hillvalley</i>	<u>1,1902E-05</u>	<u>0,0291</u>	<u>6,9355E-04</u>

Literatürle kıyaslama çalışması 2:

BinSSA-CR ve BinSSA varyasyonlarının performansı, literatürün iyi bilinen sezgisel algoritmaları ile karşılaştırılmıştır. Bu algoritmalar Genetik Algoritma (GA), Parçacık Sürüsü Optimizasyonu (PSO), Karınca Aslanı Optimizasyonu (ALO), Gri Kurt Optimizasyonu (GWO), BinSSA-CR ve BinSSA'dır. Tüm deneyler, adil bir karşılaştırma için aynı parametrelere ve koşullara göre yapılmıştır. BinSSA-CR ve karşılaştırma algoritmaları için kullanılan parametre ayarları çizelge 4.113'de gösterilmiştir. Uygunluk değerlerinin ortalamaları ve ortalama doğruluk sonuçları GA, PSO, ALO ve GWO için doğrudan Hegazy ve arkadaşlarından alınmıştır (Hegazy ve ark., 2020).

Çizelge 4.113. BinSSA-CR ve karşılaştırma algoritmaları için parametre ayarları

Parametreler	Değerler
Popülasyon miktarı (N)	10
Maksimum yenileme	50
Bağımsız çalışma sayısı	20
Problem boyutu (n)	Veri seti içindeki öznitelik sayısı
ρ	0,9
φ	0,1
K	5 (K-NN içindeki K değeri) (Zawbaa ve ark., 2015)
k	10 (k-fold içindeki k değeri)
c1,c2	1,4 (PSO için öğrenme faktörü) (Hegazy ve ark., 2020)
ω	0,7 (PSO için atalet faktörü) (Hegazy ve ark., 2020)
Cross_val	0,9 (GA için çaprazlama faktörü) (Hegazy ve ark., 2020)
Mut_val	0,1 (GA için mutasyon faktörü) (Hegazy ve ark., 2020)
Arama uzayı	{0, 1}
BinSSA ve BinSSA-CR için parametreler	$r_a=1$; $p_c=0,7$; $p_m=0,1$ (James ve Li, 2015a)

Çizelge 4.114'de BinSSA-CR ve karşılaştırma algoritmaları uygunluk değerlerinin ortalamasına göre karşılaştırılmıştır. Uygunluk değerlerinin ortalamasına göre, BinSSA-CR; N12, N17, N19 ve N21 hariç 21 veri kümesinden 17'si için kıyaslama veri kümelerinin % 80,95'inde üstün performans göstermektedir. Çizelge 4.114'deki sonuçlardan, geliştirilen BinSSA-CR veri kümelerinin % 80,95'inde diğer algoritmaları aşabileceği anlaşılabilmektedir. Sonuçlar, BinSSA-CR'nin karşılaştırma algoritmalarına göre çaprazlama operatörü sayesinde bir avantaj sağladığını göstermektedir. Uygunluk değerlerinin ortalamasına göre, BinSSA-CR'den sonra en iyi ikinci karşılaştırma algoritması ALO'dur. ALO, 21 veri kümesinden 2'si için karşılaştırma veri kümelerinin yaklaşık % 10'unda iyi performans göstermiştir.

Çizelge 4.115'de BinSSA-CR ve karşılaştırma algoritmaları sınıflandırıcının doğruluk değerleri ortalamasına göre karşılaştırılmıştır. Sınıflandırma sonuçlarının doğruluğuna göre, BinSSA-CR; N20 hariç 21 veri kümesinden 20'si için kıyaslama veri kümelerinin % 95,24'ünde üstün performans göstermektedir. BinSSA-CR tekniği, GWO'ya kıyasla sınıflandırma oranlarını geliştirmiştir. BinSSA'nın sınıflandırma oranı % 10'dur ve GWO'nun sınıflandırma oranı % 5'tir. Çaprazlama operatörü sayesinde BinSSA-CR; BinSSA, GA, PSO, ALO ve GWO'yu geçmiştir.

Çizelge 4.114. Uygunluk değerlerinin ortalamasına göre, BinSSA, BinSSA-CR, GA, PSO, ALO ve GWO algoritmalarının karşılaştırılması (* ilgili performans karşılaştırmaları yapılamadı.)

ID	Benchmark	BinSSA-CR	BinSSA	GA	PSO	ALO	GWO
		Uygunluk	Uygunluk	Uygunluk	Uygunluk	Uygunluk	Uygunluk
N1	<i>Wine</i>	0,0230	0,0230	0,0230	0,0540	0,0372	0,0430
N2	<i>Hepatitis</i>	0,0905	0,1263	0,1600	0,1500	0,1293	0,2195
N3	<i>Vehicle</i>	0,2091	0,2393	0,2300	0,2100	0,3019	0,2945
N4	<i>Zoo</i>	0,0618	0,0758	0,1160	0,2000	0,1875	0,1027
N5	<i>Heart</i>	0,1210	0,1348	0,1250	0,1850	0,2026	0,1301
N6	<i>Breast Cancer</i>	0,0353	0,0399	0,0380	0,0370	0,0425	0,0415
N7	<i>Ionosphere</i>	0,0554	0,1214	0,1270	0,1760	0,1199	0,1415
N8	<i>Lung Cancer</i>	0,0290	0,1296	0,1500	0,1724	0,1902	0,2844
N9	<i>Dermatology</i>	0,0200	0,0522	0,0900	0,0425	0,0214	0,0226
N10	<i>Sonar</i>	0,067	0,069	0,1690	0,3190	0,1769	0,2256
N11	<i>BreastEW</i>	0,0410	0,0512	0,0420	0,0540	0,0564	0,0654
N12	<i>Soybean Small</i>	0,0029	0,0049	0,0600	0,0400	0,0019	0,2913
N13	<i>Movementlibras</i>	0,1698	0,1749	0,2960	0,3330	0,3889	0,2784
N14	<i>Parkinsons</i>	0,0070	0,0293	0,0840	0,0950	0,0936	0,1242
N15	<i>Spambase</i>	0,0880	0,0943	0,2210	0,2760	0,2904	0,2840
N16	<i>Waveform</i>	0,1910	0,2020	0,2180	0,2400	0,2681	0,2648
N17	<i>Arrhythmia</i>	0,2919	0,2935	0,0600	0,0900	0,0374	0,0363
N18	<i>Semeion*</i>	0,7046	0,7286	-	-	-	-
N19	<i>Clean</i>	0,0542	0,0965	0,0640	0,0920	0,0161	0,0184
N20	<i>CNAE</i>	0,1586	0,2069	0,2670	0,3270	0,1662	0,2000
N21	<i>Hillvalley</i>	0,2490	0,2920	0,2450	0,2630	0,4052	0,4088

Çizelge 4.115. Ortalama sınıflandırıcı doğruluk değerlerine göre, BinSSA, BinSSA-CR, GA, PSO, ALO ve GWO algoritmalarının karşılaştırılması (* ilgili performans karşılaştırmaları yapılamadı.)

ID	Benchmark	BinSSA-CR	BinSSA	GA	PSO	ALO	GWO
		Doğruluk	Doğruluk	Doğruluk	Doğruluk	Doğruluk	Doğruluk
N1	<i>Wine</i>	0,9806	0,9750	0,9570	0,9230	0,9420	0,9120
N2	<i>Hepatitis</i>	0,9100	0,9000	0,8750	0,8605	0,8825	0,8479
N3	<i>Vehicle</i>	0,7432	0,7201	0,7163	0,6695	0,6814	0,6172
N4	<i>Zoo</i>	0,9800	0,9675	0,8540	0,8240	0,8050	0,8750
N5	<i>Heart</i>	0,8630	0,8521	0,8240	0,8220	0,8020	0,8070
N6	<i>Breast Cancer</i>	0,9688	0,9685	0,9550	0,9510	0,9500	0,9530
N7	<i>Ionosphere</i>	0,9486	0,9100	0,8240	0,8480	0,8430	0,8190
N8	<i>Lung Cancer</i>	0,9750	0,9083	0,4820	0,5627	0,5056	0,5014
N9	<i>Dermatology</i>	0,9932	0,9849	0,9645	0,9071	0,9322	0,9488
N10	<i>Sonar</i>	0,9425	0,9375	0,7170	0,7230	0,7140	0,7140
N11	<i>BreastEW</i>	0,9540	0,9520	0,9350	0,9490	0,9420	0,9490
N12	<i>Soybean Small</i>	1	1	0,9438	0,8648	0,9098	0,9205
N13	<i>Movementlibras</i>	0,8250	0,8222	0,6902	0,6466	0,6597	0,6866
N14	<i>Parkinsons</i>	1	1	0,8492	0,8653	0,8367	0,8367
N15	<i>Spambase</i>	0,9198	0,9098	0,8229	0,8735	0,8804	0,8839
N16	<i>Waveform</i>	0,8149	0,8008	0,7620	0,7620	0,7690	0,7650
N17	<i>Arrhythmia</i>	0,6844	0,6828	0,5802	0,5707	0,5462	0,5641
N18	<i>Semeion*</i>	0,2100	0,2075	-	-	-	-
N19	<i>Clean</i>	0,9237	0,9068	0,7648	0,7784	0,8098	0,7953
N20	<i>CNAE</i>	0,8400	0,7900	0,8246	0,8147	0,7962	0,8407
N21	<i>Hillvalley</i>	0,6708	0,6658	0,5627	0,5507	0,5709	0,5544

Kapsamlı deneyler, BinSSA-CR'nin özellik seçme problemiyle başa çıkmanın yollarını göstermiştir. Önerilen algoritma, *wine* gibi düşük ölçekli boyutlu veri kümelerinde ve *CNAE* gibi büyük ölçekli boyutlu veri kümelerinde yüksek performans göstermiştir. BinSSA'nın başarısındaki ana neden *çaprazlama operatörüdür*. Çaprazlama operatörü, örümceğin konumunu ve davranışını önemli ölçüde değiştirmiştir. Bu teknik, örümcek konumlarının farklı bölgelerde bulunmasını ve örümceklerin keşfetme ve sömürü yeteneklerini arttırmıştır. S-şekilli ve V-şekilli transfer fonksiyonları, sürekli değerleri ikili değerlerle etkili bir şekilde eşleyebilir. BinSSA varyasyonları, özellik seçme problemlerinin çoğunda iyi performans göstermiştir. BinSSA ve BinSSA-CR'nin önerilen varyasyonları, üstün sonuçlar sağlama potansiyeli yüksektir.

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Orijinal Sosyal Örümcek Algoritması (SSA), doğadaki örümcek davranışlarının taklit edilmesiyle oluşturulan sezgisel bir algoritmadır. Genellikle örümcekler doğada yalnız yaşarlar. Fakat bazı örümcek türleri koloni olarak da yaşayabilir. Kolonilerde yaşayan örümcekler örümcek ağı yardımıyla birbirleriyle iletişim kurabilirler. Örümcek ağı üzerinde ürettikleri titreşim ile bu iletişimi sağlarlar. Paylaştıkları titreşim bilgisiyle birbirlerine rastgele yürüyüşler yapabilirler. SSA, optimizasyon problemlerinde arama uzayını yüksek boyutlu bir örümcek ağı olarak düzenler. SSA, gerçek dünyadaki problemlere ve optimizasyon problemlerine kolayca adapte edilebilir.

SSA, orijinalde sürekli optimizasyon problemlerini çözmek için geliştirilmiştir. Düşük ölçekli sürekli optimizasyon problemleri üzerinde performansı test edilmiştir. Literatür incelendiğinde SSA üzerine yeterince odaklanılmadığı görülmüştür. Özellikle Sosyal Örümcek Optimizasyonu (SSO) algoritması ile karıştırılması SSA üzerine yeterince çalışılamamasına sebep olmuştur. Bu tezin amacı mevcut SSA'yı geliştirerek, farklı optimizasyon problemleri üzerinde test etmektir. Özellikle SSA'nın literatürde mevcut olmayan sürekli, ayrık ve ikili optimizasyon görevleri üzerinde performanslarını test etmektir.

Bu tezde ilk önce Sosyal Örümcek Algoritması (SSA) iki farklı sürekli optimizasyon problemi için geliştirilmiştir. Bunlar: Büyük Ölçekli Küresel Optimizasyon (LSGO) problemleri ve Moleküler Potansiyel Enerji Fonksiyonu (MPEF)'dur. LSGO için SSA geliştirilerek İyileştirilmiş Sosyal Örümcek Algoritması (ISSA) önerilmiştir. ISSA'ya iki farklı özellik eklenmiştir. Bu özellikler *örümcek patlatması* ve *kâşif örümcek belleği* yöntemleridir. Örümcek patlatma tekniğinde, her iterasyonun en kötü uygunluk fonksiyonu değerine sahip örümceği belirlenmiş ve rastgele üretilmiş yeni bir örümcek ile yer değiştirmiştir. Böylece ISSA sürekli arama uzayında yeni noktaları keşfedebilmiştir. Kâşif örümcek belleği tekniği ile önceki iterasyonlarda ki iyi uygunluk fonksiyonu değerine sahip örümceklerin unutulması önlenmiştir. Orijinal SSA'nın başarısı, {10, 20 ve 30} boyutlarda CEC2015 fonksiyonları üzerinde test edilmiştir ama geniş ölçekli boyutlarda başarısı test edilmemiştir. Bu tezde ISSA'nın, hem küçük ölçekli hem de geniş ölçekli boyutlarda da {10, 20, 30, 100, 500 ve 1000} başarısı test edilmiştir. SSA ve ISSA performansları,

seilen benchmark fonksiyonlarında test edilmiř ve sonular karřılařtırılmıřtır. Her benchmark fonksiyonunun ortalama ve standart sapma deėerleri hesaplanmıřtır. Elde edilen sonulara gre, ISSA seilen benchmark fonksiyonlarının oėunda stn performans gstermiřtir. Bu, SSA'ya eklenen yeni zelliklerin SSA performansını artırdıėını kanıtlamıřtır. Literatrde ABC, PSO, DE, SalpSA, CMA-ES, GSA, FA ve SFLA ieren tanınmıř sezgisel algoritmalar orta ve byk lekli boyutlarda ISSA ile karřılařtırılmıřtır. Elde edilen sonular ISSA'nın performansının literatrdeki diėer sezgisel yntemlere kıyasla ok iyi olduėunu ve LSGO iin alternatif bir yntem olarak kullanılabileceėini gstermiřtir.

Daha sonra SSA, farklı tekniklerle de geliřtirilerek modifiye ISSA nerilmiř ve ISSA ile MPEF problemi zlmřtr. SSA performansını artıran yeni drt zellik eklenerek geliřtirilmiřtir. Bu zellikler aprazlama, mutasyon, G_{best} yakınsaması ve sessiz rmcek teknikleridir. Her drt teknikte de, ISSA srekli arama alanında yerel ve kresel arama yapabilme yeteneėini geliřtirilmiřtir. aprazlama ve G_{best} yakınsama teknikleri, srekli arama alanında ISSA'nın yerel arama kapasitesini artırırken, mutasyon ve sessiz rmcek teknikleri srekli arama alanında ISSA'nın kresel arama kapasitesini arttırmıřtır. aprazlama oranı (cr), mutasyon oranı (mr) ve G_{best} yakınsama oranı (gr) deėerleri { %10, %20, %30, %40, %50, %60, %70, %80, %90 } on farklı tek modlu ve ok modlu benchmark fonksiyonu zerinde test edilmiř ve en uygun parametre deėerleri seilmiřtir. SSA ve ISSA'nın performansları, on  tek modlu ve ok modlu karřılařtırma fonksiyonu zerinde test edilmiř ve SSA ve ISSA'dan elde edilen sonular karřılařtırılmıřtır. Elde edilen sonulara Wilcoxon signed-rank testi uygulanmıřtır. Sonular ISSA'nın stn performans gsterdiėini gstermiřtir. SSA ve ISSA'nın performansı, farklı boyutları olan {20, 40, 60, 80, 100, 120, 140, 160, 180, 200} MPEF problemi zerinde test edilmiřtir. MPEF, leklenebilir, basitleřtirilmiř Molekler Potansiyel Enerji Fonksiyonudur. ISSA, molekln enerji fonksiyonunun basitleřtirilmiř bir modelini en aza indirmek iin kullanılmıřtır. MPEF, en nemli gerek hayat problemlerinden biridir ve bir proteinin 3D yapısını tahmin etmeye alıřır. Sonular, ISSA'nın performansının SSA'nın performansından daha iyi olduėunu gstermiřtir. ISSA'nın performansı literatrde iyi bilinen ve yeni algoritmalar ile karřılařtırılmıřtır. Sonular ISSA'nın stn bir performansa sahip olduėunu ve ISSA'nın alternatif bir tercih olarak kullanılabileceėini gstermiřtir.

nc bir uygulama ile SSA ayrıık optimizasyon problemlerini zebilmesi iin ayrıık bir hale getirilmiř ve Ayrıık Sosyal rmcek Algoritması (DSSA) nerilmiřtir.

SSA'nın literatürde henüz ayrık şekli mevcut değildir. Ayrık niceliklerin optimal olarak düzenlenmesi, gruplanması, sıraya konulması veya seçilmesi ayrık optimizasyon olarak adlandırılmaktadır. DSSA, bir ayrık optimizasyon problemi olan Gezgin Satıcı Problemi (TSP) üzerinde test edilmiştir. TSP, ayrık optimizasyon algoritmalarının performans analizinde kullanılan standart test problemlerinden biridir. DSSA, *yetenekli örümcek* ve *acemi örümcek* özellikleri eklenerek ayrık arama uzayındaki keşif ve sömürü yetenekleri geliştirilmiştir. DSSA ile küçük, orta ve geniş ölçekli otuz sekiz TSP örnekleri çözülmüştür. TSP veri seti, ortalama rota uzunluğu, standart sapma ve yüzde hata değerleri dikkate alınarak değerlendirilmiştir. Literatürde genellikle küçük ve orta ölçekli TSP örnekleri test için kullanılırken, bu tezde geniş ölçekli TSP örnekleri de test için kullanılmıştır. Önerilen DSSA ile p_m , p_c ve r_a parametrelerinde SSA'nın performansını etkileyen tüm kombinasyonlar uygulanarak uygun değerler tespit edilmiştir. DSSA ile farklı sayıdaki örümceklerin etkileri de analiz edilmiştir. 2-Opt yerel arama algoritması DSSA ile elde edilen son sonuca uygulanmıştır. Sonuçlar ideal sonuçlarla karşılaştırılmıştır. Literatürde iyi bilinen on sekiz yeni ve bilinen yaklaşımlar ile DSSA karşılaştırılmıştır. Deneysel sonuçlara göre, DSSA, TSP'de küçük veri kümeleri için iyi sonuçlara sahiptir, ancak daha büyük veri kümeleri için TSP'de iyi sonuçlara sahip değildir. DSSA, düşük ve orta ölçekli TSP veri seti örneklerinde iyi bir başarıya sahiptir. DSSA'nın optimum sonuçlara yakınsaması oldukça iyidir. Önerilen DSSA'da sunulan *yetenekli örümcek* ve *acemi örümcek* özellikleri, diğer kombinatoriyal optimizasyon problemlerini çözmek için daha fazla esneklik sağlamıştır. Sonuçlar, DSSA'nın ayrık optimizasyon görevleri için yeni bir algoritma olarak kullanılabileceğini göstermektedir.

Dördüncü uygulamada SSA ikili optimizasyon problemleri için ayrık hale getirilmiştir. İkili optimizasyon, ayrık optimizasyon tekniklerinin özel bir şeklidir. Sürekli gerçek arama alanlarına sahip algoritmalar, değişkenleri ikili değişkenlere dönüştürerek İkili Optimizasyon Problemlerini (BOP) çözebilir. İkili optimizasyonda, çözümü ikili $\{0, 1\}$ değerleriyle sınırlıdır. Sürekli optimizasyon problemleri için geliştirilmiş SSA, ikili optimizasyon problemlerini de çözebilecek şekilde tekrar yapılandırılmıştır. Böylece BinSSA önerilmiştir. Bu tezde BinSSA ile dört farklı BOP çözülmüştür. Bu BOP'ler: Özellik Seçme (FS) problemi, Kapasitesiz Tesis Yerleşim Problemi (UFLP), rüzgâr türbinleri yerleşim problemi ve sürekli optimizasyon görevleridir.

BinSSA ile ilk çalışmada sürekli optimizasyon görevleri test edilmiştir. BinSSA’da sürekli arama uzayı ikili uzaya çevrilirken, iki S-şekilli (Sigmoidal transfer fonksiyonu ve Modifiye sigmoidal transfer fonksiyonu) ve iki V-şekilli (Tanjant hiperbolik transfer fonksiyonu ve Arktanjan transfer fonksiyonu) transfer fonksiyonları kullanılmıştır. Böylece, BinSSA-Tanh, BinSSA-Sigm, BinSSA-MSigm ve BinSSA-Arctan olarak dört farklı BinSSA varyasyonu oluşturulmuştur. BinSSA’da ikili arama uzayında keşif ve sömürü arasındaki dengeyi sağlamak için aday çözüm üretim aşamasında iki farklı teknik geliştirilmiştir. Bunlar *Jaccard’ın benzerlik ölçümü* (SimSSA) ve “xor” mantıksal kapısı (LogicSSA) teknikleridir. BinSSA’nın performansı on sekiz farklı tek modlu ve çok modlu karşılaştırma fonksiyonu üzerinde test edilmiştir. BinSSA varyasyonlarının performansı çeşitli ölçütlerde birbirleri ile karşılaştırılmıştır. Wilcoxon signed-rank testi elde edilen sonuçlara uygulanmıştır. BinSSA, PSO, GA, BSalpSA, BHPsOWM, BPSO, BBA, BSCA ve BSSO ile BinSSA varyasyonları karşılaştırılmıştır. BinSSA varyasyonları arasındaki karşılaştırmalara göre, BinSSA-Arctan ve BinSSA-Tanh varyasyonlarının BinSSA’nın diğer varyasyonları arasında yüksek performansa sahip olduğu gösterilmiştir. Karşılaştırma sonuçları, BinSSA-Arctan ve BinSSA-Tanh varyasyonlarının çoğu test için iyi kalitede çözümler sağlamadaki üstünlüğünü doğrulamıştır.

BinSSA ile ikinci çalışmada BOP’lerden Kapasitesiz Tesis Yerleşim Problemi (UFLP) çözülmüştür. UFLP problemleri saf ikili optimizasyon problemlerinden biridir ve literatürde optimum değerleri bilinmektedir. BinSSA, iki farklı UFLP veri seti üzerinde (15 farklı ölçekli Cap veri seti ve 20 farklı ölçekli M* veri seti) üzerinde test edilmiştir. Ayrıca BinSSA’nın performansı CEC2015 benchmark fonksiyonları üzerinde de test edilmiştir. Aday çözüm üretimi için iki farklı yöntem geliştirilmiştir. Bunlar *Dice’in benzerlik ölçümü* ve *xor-maskesi* teknikleridir. Xor-maskesi tekniği güçlü bir yerel arama kapasitesine sahiptir ve Dice’in benzerlik ölçüm tekniği, çözüm uzayındaki yeni noktaları keşfetmede iyidir. BinSSA, bu teknikler sayesinde keşfetme ve sömürü yetenekleri arasında dengeyi kurabilmiştir. BinSSA’nın performansı çeşitli ölçütlerde karşılaştırılmıştır. Bu ölçütler ortalama, standart sapma, en iyi, en kötü ve gap (fark)’tır. BinSSA ile GA-SP, GA-TP, GA-UP, BAAA-Tanh, BAAA-Sig, BAAASB, ABCBin, DisABC, BinABC, DisDE, BinDE, BPSO, CPSO, TSA, ISS, BJA, JayaX, JayaX-LSM sezgisel algoritmaları karşılaştırılmıştır. Karşılaştırma sonuçlarına göre, BinSSA, BAAASB ve JayaX-LSM hariç karşılaştırmalı algoritmalar arasında yüksek performansa sahiptir. Son olarak, BinSSA yirmi farklı M* örneğinde çalıştırılmıştır. M*

bir UFLP veri kümesidir. BinSSA sonuçları LS, ISS, TS1 ve TS3 algoritması ile karşılaştırılmıştır. Sonuçlar, BinSSA'nın, çoğu test için daha hızlı yakınsama performansı ile iyi kalitede çözümler sağlamadaki üstünlüğünü doğrulamıştır.

BinSSA ile üçüncü çalışmada bir BOP olan rüzgâr türbini yerleşim problemi çözülmüştür. Bu çalışmada, mümkün olan en yüksek rüzgâr enerjisini elde etmek için rüzgâr türbinlerinin bir rüzgâr çiftliğine optimum şekilde yerleştirilmesi incelemiştir. Dört adet S-şekilli, dört adet V-şekilli ve iki adet mod tabanlı transfer fonksiyonuna göre on adet BinSSA varyasyonu oluşturulmuştur (BinSSA1, BinSSA2, BinSSA3, BinSSA4, BinSSA5, BinSSA6, BinSSA7, BinSSA8, BinSSA9 ve BinSSA10). Aday çözüm üretimi için bir *lojik kapı operatörü* eklenmiş ve böylece BinSSA'da kâşif ve sömürü arasındaki dengenin kurulduğu yeni varyasyonlar üretilmiştir (BinSSA1-LG, BinSSA2-LG, BinSSA3-LG, BinSSA4-LG, BinSSA5-LG, BinSSA6-LG, BinSSA7-LG, BinSSA8-LG, BinSSA9-LG ve BinSSA10-LG). Rüzgâr türbinlerinin rüzgâr çiftliğine yerleştirilmesi problemi, rüzgâr çiftliği alanını ızgaralara bölerek çözülebilir. BinSSA'nın, minimum çiftlik maliyeti ve maksimum üretilen enerji elde etmesi şartıyla, türbinlerin rüzgar çiftliğine optimum yerleşimini belirlemesi hedeflenmiştir. 2x2 km'lik bir alan için 10x10 ve 20x20 ızgara yapısı seçilmiştir. BinSSA ve BinSSA-LG varyantlarının performansı çeşitli ölçütlerde (uygunluk değeri, toplam güç ve rüzgâr türbinlerinin sayısı) karşılaştırılmıştır. Uygunluk sonuçlarının ortalamasına göre, BinSSA-LG'nin en iyi varyasyonu BinSSA9-LG olarak belirlenmiştir. Wilcoxon signed-rank testi BinSSA9-LG ve diğer BinSSA ve BinSSA-LG varyasyonlarına uygulanmıştır. Literatürde iyi bilinen sezgisel algoritmaları olan GA, MILP, BPSO-TVAC, Lazy Greedy, BIWO, SA ve BinAAA, BinSSA ve BinSSA-LG ile karşılaştırılmıştır. Karşılaştırmanın adil olması için, tüm karşılaştırma algoritmaları eşit parametrelerde ve eşit koşullarda çalıştırılmıştır. Uygunluk değerlerinin ortalaması ve toplam gücün ortalamasına göre BinSSA9-LG'nin performansının üstün olduğu gösterilmiştir.

BinSSA ile dördüncü çalışmada bir BOP olan BinSSA ile Özellik Seçme (FS) problemi çözülmüştür. Sürekli arama uzayının ikili arama uzayına dönüşümünde dört adet S-şekilli ve dört adet V-şekilli transfer fonksiyonu kullanılmıştır. Kullanılan transfer fonksiyonlarına göre sekiz adet BinSSA varyasyonu oluşturulmuştur (BinSSA1, BinSSA2, BinSSA3, BinSSA4, BinSSA5, BinSSA6, BinSSA7 ve BinSSA8). Aday çözüm üretimi için bir *çaprazlama operatörü* eklenmiş ve böylece BinSSA'da kâşif ve sömürü arasındaki dengenin kurulduğu varyantlar üretilmiştir (BinSSA1-CR, BinSSA2-

CR, BinSSA3-CR, BinSSA4-CR, BinSSA5-CR, BinSSA6-CR, BinSSA7-CR ve BinSSA8-CR). OR-Lib'den (UCI Machine Learning Repository) alınan düşük ölçekli, orta ölçekli ve büyük ölçekli yirmi bir UCI karşılaştırma veri setleri, özellik seçme probleminde BinSSA performansını ölçmek için kullanılmıştır. Çalışmada bir sınıflandırıcı olarak K-NN ve SVM kullanılmıştır. BinSSA ve BinSSA-CR varyantlarının performansı çeşitli ölçütlerde ((1) uygunluk değerlerinin ortalama ve standart sapması, (2) uygunluk değerlerinin en iyi ve en kötü değerleri, (3) elde edilen doğruluk değerlerinin ortalaması, (4) seçilen özelliklerin ortalama sayısı (5) ortalama CPU zamanı) karşılaştırılmıştır. Ortalama uygunluk değerleri, ortalama standart sapma değerleri, ortalama doğruluk değerleri ve seçilen ortalama özellik sayısına göre, BinSSA-CR'nin en iyi varyasyonu BinSSA4-CR olarak belirlenmiştir. Wilcoxon signed-rank testi BinSSA4-CR ve diğer BinSSA varyasyonlarına uygulanmıştır. BinSSA'nın performansı sezgisel algoritmalarla karşılaştırılarak performansı analiz edilmiştir. Literatürde iyi bilinen sezgisel algoritmalar olan İkili Yarasa Algoritması (BBA), İkili Gri Kurt Optimizasyonu (BGWO) ve İkili Parçacık Sürü Optimizasyonu (BPSO) ile BinSSA4 ve BinSSA4-CR varyasyonları karşılaştırılmıştır. Karşılaştırmanın adil olması için, tüm karşılaştırma algoritmaları eşit parametrelerde ve eşit koşullarda çalıştırılmıştır. Wilcoxon signed-rank testi BBA, BGWO ve BinSSA4-CR varyasyonuna uygulanmıştır. Seçilen BinSSA4-CR varyasyonu performansının üstünlüğü gösterilmiştir. Uygunluk değerlerinin ortalaması ve doğruluk sonuçlarının ortalamasına göre, Genetik Algoritma (GA), Parçacık Sürüsü Optimizasyonu (PSO), Karınca Aslanı Optimizasyonu (ALO) ve Gri Kurt Optimizasyonu (GWO) ile BinSSA-CR karşılaştırılmıştır. Sonuçlara göre, BinSSA-CR varyasyonları çoğu test için kaliteli çözümler sunmuştur.

Deneysel çalışma sonuçları:

- SSA algoritması yapı itibarıyla birey çeşitliliği olmayan bir sezgisel algoritmadır. Bu tezde SSA algoritması sürekli optimizasyon problemlerinde bireysel çeşitlilik sağlanmıştır.
- SSA algoritması yapı itibarıyla literatürdeki diğer sezgisel algoritmalarından daha basit bir yapı sunmaktadır. Bu basitlik literatürdeki diğer sezgisel algoritmalar ile karşılaştırıldığında çalışma süresini ve başarısını etkilememektedir.
- SSA algoritmasının orijinal halinde sadece küçük boyutlarda performansı test edilmiştir. Geliştirilmiş SSA algoritması geniş ölçekli boyutlarda da

testi bu tezde gerçekleştirilmiş ve literatürde geniş ölçekli sürekli optimizasyon görevleri için alternatif bir algoritma sunulmuştur.

- SSA algoritmasının ayrık optimizasyon problemleri için yapısı düzenlenerek literatüre ayrık bir optimizasyon algoritması (DSSA) önerilmiştir. DSSA'nın başarısı literatürdeki ayrık optimizasyon algoritmaları ile yarışabilecek düzeydedir.
- İkili optimizasyon problemleri ayrık optimizasyon problemlerinin özel bir şeklini oluşturmaktadır. Gerçek dünya problemlerinin bir çoğu arama uzayındaki ikili yapıları kullanarak çözüme ulaşmaktadır. SSA, bu tezde ikili bir yapıya dönüştürülerek BinSSA önerilmiştir. BinSSA ile çeşitli problemler (özellik seçme, kapasitesiz tesis yerleşim problemi vb.) bu tezde çözülmüş ve başarısı test edilmiştir. Sonuçlar göstermiştir ki BinSSA literatürde ikili sezgisel algoritmalar ile yarışabilecek düzeydedir.

5.2. Öneriler

SSA algoritması normalde arama uzayında farklı örümcek cinsleri içermeyen bir algoritmadır. SSA bireylerine ağırlık değerleri eklenerek daha performanslı sonuçlar elde edilmesi mümkündür ve önerilmektedir. Ayrıca SSA'nın yetersiz geldiği aşamalarda farklı evrimsel algoritmalar ile hibritleştirilmesi SSAnın başarısını arttıracaktır.

SSA'nın ayrık optimizasyon problemleri üzerinde uygulaması sadece TSP üzerinde gerçekleştirilmiştir. Farklı ayrık problemlerin SSA ile çözülmesi, SSA'nın başarısını gösterecektir. Ayrıca TSP üzerinde SSA'nın başarısını arttırmak için farklı sezgisel algoritmalarla faydalanılabilir.

SSA'nın ikili arama uzayına dönüşümü bu tezde on farklı transfer fonksiyonu kullanılarak gerçekleştirilmiştir. Keşif ve sömürü yeteneklerini geliştiren farklı transfer fonksiyonlarının SSA için tasarlanması önerilmektedir. BinSSA'nın başarısı bu tezde sadece dört BOP üzerinde gerçekleştirilmiştir. Farklı BOP'ler üzerinde BinSSA'nın başarısını ölçmek, BinSSA'nın üstünlüğünü doğrulayacaktır. Yüksek boyutlu BOP'lardaki performansını artırmak için BinSSA'ya çeşitli yerel arama algoritmaları eklenmesi önerilmektedir.

6. KAYNAKLAR

- Acılar, A. M., 2013, Yapay bağışıklık algoritmaları kullanılarak bulanık sistem tasarımı, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Akay, B., 2013, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, *Applied Soft Computing*, 13 (6), 3066-3091.
- Akhand, M. A. H. A., Sazzadur Rahman, S., Hafizur Rahman, M.M., 2012, Particle Swarm Optimization with partial search to solve Traveling Salesman Problem. 2012 International Conference on Computer and Communication Engineering (ICCE): 118-121.
- AL-Saati, N. A. ve Alabajee, M. A., 2020, Selecting Best Software Reliability Growth Models: A Social Spider Algorithm based Approach, *International Journal of Computer Applications*, 181.
- Ali, A. F. ve Tawhid, M. A., 2017, A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems, *Ain Shams Engineering Journal*, 8 (2), 191-206.
- Aras, N., Boyacı, B., Koşucuoğlu, D. ve Aksen, D., 2007, Karlı Gezgin Satıcı Problemi için Sezgisel Yöntemler, *Yöneylem Araştırması/Endüstri Mühendisliği 27. Ulusal Kongresi*.
- Ardjmand, E., Park, N., Weckman, G. ve Amin-Naseri, M. R., 2014, The discrete unconscious search and its application to uncapacitated facility location problem, *Computers & Industrial Engineering*, 73, 32-40.
- Aslan, M., Gunduz, M. ve Kiran, M. S., 2019, JayaX: Jaya algorithm with xor operator for binary optimization, *Applied Soft Computing*, 82, 105576.
- Atashpaz-Gargari, E. ve Lucas, C., 2007, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, *2007 IEEE congress on evolutionary computation*, 4661-4667.
- Babaoglu, İ., Findik, O. ve Ülker, E., 2010, A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine, *Expert Systems with Applications*, 37 (4), 3177-3183.
- Banitalebi, A., Abd Aziz, M. I. ve Aziz, Z. A., 2016, A self-adaptive binary differential evolution algorithm for large scale binary optimization problems, *Information Sciences*, 367, 487-511.
- Barbosa, H. J., Lavor, C. C. ve Raupp, F. M., 2005, A GA-simplex hybrid algorithm for global minimization of molecular potential energy functions, *Annals of Operations Research*, 138 (1), 189.

- Bastı, M., 2012, The p-median facility location problem and solution approaches, *AJIT-e*, 3 (7), 47.
- Baş, E. ve Ülker, E., 2020a, Discrete social spider algorithm for the traveling salesman problem, *ARTIFICIAL INTELLIGENCE REVIEW*.
- Baş, E. ve Ülker, E., 2020b, A binary social spider algorithm for continuous optimization task, *Soft Computing*, 1-27.
- Baş, E. ve Ülker, E., 2020c, An efficient binary social spider algorithm for feature selection problem, *Expert Systems with Applications*, 146, 113185.
- Baş, E. ve Ülker, E., 2020d, A binary social spider algorithm for uncapacitated facility location problem, *Expert Systems with Applications*, 113618.
- Beasley, J. E., 1990, OR-Library: distributing test problems by electronic mail, *Journal of the operational research society*, 41 (11), 1069-1072.
- Bello, R., Gomez, Y., Nowe, A. ve Garcia, M. M., 2007, Two-step particle swarm optimization to solve the feature selection problem, *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, 691-696.
- Beşkirli, M., Koç, İ., Hakkı, H. ve Kodaz, H., 2018, A new optimization algorithm for solving wind turbine placement problem: Binary artificial algae algorithm, *Renewable Energy*, 121, 301-308.
- Beşkirli, M., Koc, I. ve Kodaz, H., 2019, Optimal placement of wind turbines using novel binary invasive weed optimization, *Tehnički vjesnik*, 26 (1), 56-63.
- Blum, C. ve Li, X., 2008, Swarm intelligence in optimization, In: *Swarm Intelligence*, Eds: Springer, p. 43-85.
- Changshui, Z., Guangdong, H. ve Jun, W., 2011, A fast algorithm based on the submodular property for optimization of wind turbine positioning, *Renewable Energy*, 36 (11), 2951-2958.
- Choi, S.-S., Cha, S.-H. ve Tappert, C. C., 2010, A survey of binary similarity and distance measures, *Journal of Systemics, Cybernetics and Informatics*, 8 (1), 43-48.
- Chuang, L.-Y., Chang, H.-W., Tu, C.-J. ve Yang, C.-H., 2008, Improved binary PSO for feature selection using gene expression data, *Computational Biology and Chemistry*, 32 (1), 29-38.
- Cinar, A. C. ve Kiran, M. S., 2018, Similarity and logic gate-based tree-seed algorithms for binary optimization, *Computers & Industrial Engineering*, 115, 631-646.

- Cinar, A. C., Korkmaz, S. ve Kiran, M. S., 2019, A discrete tree-seed algorithm for solving symmetric traveling salesman problem, *Engineering Science and Technology, an International Journal*.
- Cuevas, E., Cienfuegos, M., Zaldívar, D. ve Pérez-Cisneros, M., 2013, A swarm optimization algorithm inspired in the behavior of the social-spider, *Expert Systems with Applications*, 40 (16), 6374-6384.
- Cura, T., 2010, A parallel local search approach to solving the uncapacitated warehouse location problem, *Computers & Industrial Engineering*, 59 (4), 1000-1009.
- Deep, K. ve Thakur, M., 2007a, A new crossover operator for real coded genetic algorithms, *Applied Mathematics and Computation*, 188 (1), 895-911.
- Deep, K. ve Thakur, M., 2007b, A new mutation operator for real coded genetic algorithms, *Applied Mathematics and Computation*, 193 (1), 211-230.
- Deep, K., Barak, S., Katiyar, V. K. ve Nagar, A. K., 2011, Minimization of molecular potential energy function using newly developed real coded genetic algorithms, *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 2 (1), 51-58.
- Dell, R. F., Ewing, P. L. ve Tarantino, W. J., 2008, Optimally stationing army forces, *Interfaces*, 38 (6), 421-435.
- Derrac, J., García, S., Molina, D. ve Herrera, F., 2011, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation*, 1 (1), 3-18.
- Dorigo, M., 1992, Optimization, learning and natural algorithms, *PhD Thesis, Politecnico di Milano*.
- Dražić, M., Lavor, C., Maculan, N. ve Mladenović, N., 2008, A continuous variable neighborhood search heuristic for finding the three-dimensional structure of a molecule, *European Journal of Operational Research*, 185 (3), 1265-1273.
- El-Bages, M. ve Elsayed, W., 2017, Social spider algorithm for solving the transmission expansion planning problem, *Electric Power Systems Research*, 143, 235-243.
- Elsayed, W., Hegazy, Y., Bendary, F. ve El-Bages, M., 2016, Modified social spider algorithm for solving the economic dispatch problem, *Engineering Science and Technology, an International Journal*, 19 (4), 1672-1681.
- Elsayed, W., Bendary, F., El-bages, M. ve Hegazy, Y., 2017, A rolling horizon framework with social spider algorithm for solving the energy management problem of utility microgrids, *2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*, 326-331.

- Emami, A. ve Noghreh, P., 2010, New approach on optimization in placement of wind turbines within wind farm by genetic algorithms, *Renewable Energy*, 35 (7), 1559-1564.
- Emary, E., Zawbaa, H. M. ve Hassanien, A. E., 2016, Binary grey wolf optimization approaches for feature selection, *Neurocomputing*, 172, 371-381.
- Ezugwu, A. E.-S. ve Adewumi, A. O., 2017, Discrete symbiotic organisms search algorithm for travelling salesman problem, *Expert Systems with Applications*, 87, 70-78.
- Faigl, J., 2018, GSOA: growing self-organizing array-unsupervised learning for the close-enough traveling salesman problem and other routing problems, *Neurocomputing*, 312, 120-134.
- Fan, K., You, W. ve Li, Y., 2013, An effective modified binary particle swarm optimization (mBPSO) algorithm for multi-objective resource allocation problem (MORAP), *Applied Mathematics and Computation*, 221, 257-267.
- Faris, H., Mafarja, M. M., Heidari, A. A., Aljarah, I., Ala'M, A.-Z., Mirjalili, S. ve Fujita, H., 2018, An efficient binary salp swarm algorithm with crossover scheme for feature selection problems, *Knowledge-Based Systems*, 154, 43-67.
- Gandomi, A. H., Yang, X.-S. ve Alavi, A. H., 2013, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Engineering with computers*, 29 (1), 17-35.
- Gao, X., Yang, H., Lin, L. ve Koo, P., 2015, Wind turbine layout optimization using multi-population genetic algorithm and a case study in Hong Kong offshore, *Journal of Wind Engineering and Industrial Aerodynamics*, 139, 89-99.
- Goh, C. K., Lim, D., Ma, L., Ong, Y.-S. ve Dutta, P. S., 2011, A surrogate-assisted memetic co-evolutionary algorithm for expensive constrained optimization problems, *2011 IEEE Congress of Evolutionary Computation (CEC)*, 744-749.
- González, J. S., Rodriguez, A. G. G., Mora, J. C., Santos, J. R. ve Payan, M. B., 2010, Optimization of wind farm turbines layout using an evolutive algorithm, *Renewable Energy*, 35 (8), 1671-1681.
- Grady, S., Hussaini, M. ve Abdullah, M. M., 2005, Placement of wind turbines using genetic algorithms, *Renewable Energy*, 30 (2), 259-270.
- Guner, A. R. ve Sevkli, M., 2008, A discrete particle swarm optimization algorithm for uncapacitated facility location problem, *Journal of Artificial Evolution and Applications*, 2008.
- Gündüz, M., Kiran, M. S. ve Özceylan, E., 2015, A hierarchic approach based on swarm intelligence to solve the traveling salesman problem, *Turkish Journal of Electrical Engineering & Computer Sciences*, 23 (1), 103-117.

- Hakli, H. ve Ortacay, Z., 2019, An improved scatter search algorithm for the uncapacitated facility location problem, *Computers & Industrial Engineering*, 135, 855-867.
- Hedar, A.-R., Ali, A. F. ve Abdel-Hamid, T. H., 2011, Genetic algorithm and tabu search based methods for molecular 3D-structure prediction, *Numerical Algebra, Control & Optimization*, 1 (1), 191.
- Hegazy, A. E., Makhlouf, M. ve El-Tawel, G. S., 2020, Improved salp swarm algorithm for feature selection, *Journal of King Saud University-Computer and Information Sciences*, 32 (3), 335-344.
- Hore, S., Chatterjee, A. ve Dewanji, A., 2018, Improving variable neighborhood search to solve the traveling salesman problem, *Applied Soft Computing*, 68, 83-91.
- Jaccard, P., 1901, Étude comparative de la distribution florale dans une portion des Alpes et des Jura, *Bull Soc Vaudoise Sci Nat*, 37, 547-579.
- James, J. ve Li, V. O., 2014, Base station switching problem for green cellular networks with social spider algorithm, *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2338-2344.
- James, J. ve Li, V. O., 2015a, A social spider algorithm for global optimization, *Applied Soft Computing*, 30, 614-627.
- James, J. ve Li, V. O., 2015b, Parameter sensitivity analysis of social spider algorithm, *2015 Ieee Congress on Evolutionary Computation (Cec)*, 3200-3205.
- James, J. ve Li, V. O., 2016, A social spider algorithm for solving the non-convex economic load dispatch problem, *Neurocomputing*, 171, 955-965.
- Jamil, M. ve Yang, X.-S., 2013, A literature survey of benchmark functions for global optimisation problems, *International Journal of Mathematical Modelling and Numerical Optimisation*, 4 (2), 150-194.
- Jati, G. K., 2011, Evolutionary discrete firefly algorithm for travelling salesman problem, *International conference on adaptive and intelligent systems*, 393-403.
- Jensen, N. O., 1983, A note on wind generator interaction.
- Jensen, R., 2005, Combining rough and fuzzy sets for feature selection, *Citeseer*.
- Jiang, F., Xia, H., Tran, Q. A., Ha, Q. M., Tran, N. Q. ve Hu, J., 2017, A new binary hybrid particle swarm optimization with wavelet mutation, *Knowledge-Based Systems*, 130, 90-101.
- Karaboga, D., 2005, An idea based on honey bee swarm for numerical optimization, *Technical report-tr06, Erciyes university, engineering faculty, computer*

- Karaboga, D. ve Basturk, B., 2007, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of global optimization*, 39 (3), 459-471.
- Kashan, M. H., Kashan, A. H. ve Nahavandi, N., 2013, A novel differential evolution algorithm for binary optimization, *Computational Optimization and Applications*, 55 (2), 481-513.
- Kennedy, J. ve Eberhart, R., 1995, Particle swarm optimization, *Proceedings of ICNN'95-International Conference on Neural Networks*, 1942-1948.
- Kennedy, J. ve Everhart, R., 1997, A discrete binary version of the particle swarm algorithm, *Proceedings 1997 Conference on Systems, Man and Cybernetics*, 4104-4109.
- Kiran, M. S. ve Gündüz, M., 2013, XOR-based artificial bee colony algorithm for binary optimization, *Turkish Journal of Electrical Engineering & Computer Sciences*, 21 (Sup. 2), 2307-2328.
- Kiran, M. S., 2015, The continuous artificial bee colony algorithm for binary optimization, *Applied Soft Computing*, 33, 15-23.
- Korkmaz, S., Babalik, A. ve Kiran, M. S., 2018, An artificial algae algorithm for solving binary optimization problems, *International Journal of Machine Learning and Cybernetics*, 9 (7), 1233-1247.
- Korkmaz, S. ve Kiran, M. S., 2018, An artificial algae algorithm with stigmergic behavior for binary optimization, *Applied Soft Computing*, 64, 627-640.
- Kuzu, S., Önay, O., Şen, U., Tunçer, M., Yıldırım, B. ve Keskinürk, T., 2014, Gezgin satıcı problemlerinin metasezgiseller ile çözümü, *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, 43 (1), 1-27.
- Lavor, C. ve Maculan, N., 2004, A function to test methods applied to global minimization of potential energy of molecules, *Numerical algorithms*, 35 (2-4), 287-300.
- Li, L., Cheng, Y., Tan, L. ve Niu, B., 2011, A discrete artificial bee colony algorithm for TSP problem, *International Conference on Intelligent Computing*, 566-573.
- Liang, J. J., Qin, A. K., Suganthan, P. N. ve Baskar, S., 2006, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, *IEEE transactions on evolutionary computation*, 10 (3), 281-295.
- Liao, T. W., Kuo, R. ve Hu, J., 2012, Hybrid ant colony optimization algorithms for mixed discrete-continuous optimization problems, *Applied Mathematics and Computation*, 219 (6), 3241-3252.
- Liu, H. ve Motoda, H., 1998, Feature Selection for Knowledge Discovery and Data Mining Kluwer Acad, Publishers.

- Liu, Y., Chen, S., Guan, B. ve Xu, P., 2019, Layout optimization of large-scale oil–gas gathering system based on combined optimization strategy, *Neurocomputing*, 332, 159-183.
- Long, W., Jiao, J., Liang, X. ve Tang, M., 2018, Inspired grey wolf optimizer for solving large-scale function optimization problems, *Applied Mathematical Modelling*, 60, 112-126.
- Long, W., Wu, T., Liang, X. ve Xu, S., 2019, Solving high-dimensional global optimization problems using an improved sine cosine algorithm, *Expert Systems with Applications*, 123, 108-126.
- Mahi, M., Baykan, Ö. K. ve Kodaz, H., 2015, A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem, *Applied Soft Computing*, 30, 484-490.
- Marmidis, G., Lazarou, S. ve Pyrgioti, E., 2008, Optimal placement of wind turbines in a wind park using Monte Carlo simulation, *Renewable Energy*, 33 (7), 1455-1460.
- Maučec, M. S. ve Brest, J., 2019, A review of the recent use of Differential Evolution for Large-Scale Global Optimization: An analysis of selected algorithms on the CEC 2013 LSGO benchmark suite, *Swarm and Evolutionary Computation*, 50, 100428.
- Mirjalili, S. ve Lewis, A., 2013, S-shaped versus V-shaped transfer functions for binary particle swarm optimization, *Swarm and Evolutionary Computation*, 9, 1-14.
- Mirjalili, S., Mirjalili, S. M. ve Lewis, A., 2014, Grey wolf optimizer, *Advances in Engineering Software*, 69, 46-61.
- Mittal, A., 2010, Optimization of the layout of large wind farms using a genetic algorithm, *Case Western Reserve University*.
- Mittal, N., Singh, U. ve Sohi, B. S., 2016, Modified grey wolf optimizer for global engineering optimization, *Applied Computational Intelligence and Soft Computing*, 2016.
- Mohapatra, P., Das, K. N. ve Roy, S., 2017, A modified competitive swarm optimizer for large scale optimization problems, *Applied Soft Computing*, 59, 340-362.
- Mosetti, G., Poloni, C. ve Diviacco, B., 1994, Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm, *Journal of Wind Engineering and Industrial Aerodynamics*, 51 (1), 105-116.
- Mousa, A. ve Bentahar, J., 2016, An efficient QoS-aware web services selection using social spider algorithm, *Procedia Computer Science*, 94, 176-182.

- Nakib, A., Ouchraa, S., Shvai, N., Souquet, L. ve Talbi, E.-G., 2017, Deterministic metaheuristic based on fractal decomposition for large-scale optimization, *Applied Soft Computing*, 61, 468-485.
- Nguyen, P. H., Wang, D. ve Truong, T. K., 2017, A novel binary social spider algorithm for 0–1 knapsack problem, *Int. J. Innov. Comput. Inf. Control*, 13 (6), 2039-2049.
- Osaba, E., Yang, X.-S., Diaz, F., Lopez-Garcia, P. ve Carballedo, R., 2016, An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems, *Engineering Applications of Artificial Intelligence*, 48, 59-71.
- Osaba, E., Del Ser, J., Sadollah, A., Bilbao, M. N. ve Camacho, D., 2018, A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem, *Applied Soft Computing*, 71, 277-290.
- Ouaarab, A., Ahiod, B. ve Yang, X.-S., 2014, Discrete cuckoo search algorithm for the travelling salesman problem, *Neural Computing and Applications*, 24 (7-8), 1659-1669.
- Pandiri, V. ve Singh, A., 2019, An artificial bee colony algorithm with variable degree of perturbation for the generalized covering traveling salesman problem, *Applied Soft Computing*, 78, 481-495.
- Papa, J. P., Rosa, G. H., de Souza, A. N. ve Afonso, L. C., 2018, Feature selection through binary brain storm optimization, *Computers & Electrical Engineering*, 72, 468-481.
- Parada, L., Herrera, C., Flores, P. ve Parada, V., 2017, Wind farm layout optimization using a Gaussian-based wake model, *Renewable Energy*, 107, 531-541.
- Pardalos, P. M., Shalloway, D. ve Xue, G., 1994, Optimization methods for computing global minima of nonconvex potential energy functions, *Journal of global optimization*, 4 (2), 117-133.
- Parpinelli, R. S. ve Lopes, H. S., 2011, New inspirations in swarm intelligence: a survey, *International Journal of Bio-Inspired Computation*, 3 (1), 1-16.
- Pashaei, E. ve Aydin, N., 2017, Binary black hole algorithm for feature selection and classification on biological data, *Applied Soft Computing*, 56, 94-106.
- Pernkopf, F., 2005, Bayesian network classifiers versus selective k-NN classifier, *Pattern recognition*, 38 (1), 1-10.
- Pookpant, S. ve Ongsakul, W., 2013, Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients, *Renewable Energy*, 55, 266-276.
- Qiao, L.-Y., Peng, X.-Y. ve Peng, Y., 2006, BPSO-SVM wrapper for feature subset selection, *Dianzi Xuebao(Acta Electronica Sinica)*, 34 (3), 496-498.

- Rashedi, E., Nezamabadi-Pour, H. ve Saryazdi, S., 2010, BGSA: binary gravitational search algorithm, *Natural Computing*, 9 (3), 727-745.
- Ray, T. ve Yao, X., 2009, A cooperative coevolutionary algorithm with correlation based adaptive variable partitioning, *2009 IEEE Congress on Evolutionary Computation*, 983-989.
- Reinelt, G., 1995, TSPLIB <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95>.
- Rizk-Allah, R., 2014, A novel multi-ant colony optimization for multi-objective resource allocation problems, *Int J Math Arch*, 5 (9), 183-192.
- Rizk-Allah, R. M., 2018, Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems, *Journal of Computational Design and Engineering*, 5 (2), 249-273.
- Rizk-Allah, R. M. ve Hassanien, A. E., 2018, New binary bat algorithm for solving 0–1 knapsack problem, *Complex & Intelligent Systems*, 4 (1), 31-53.
- Rizk-Allah, R. M., Hassanien, A. E., Elhoseny, M. ve Gunasekaran, M., 2019, A new binary salp swarm algorithm: development and application for optimization tasks, *Neural Computing and Applications*, 31 (5), 1641-1663.
- Sayed, E., Essam, D., Sarker, R. ve Elsayed, S., 2015, Decomposition-based evolutionary algorithm for large scale constrained problems, *Information Sciences*, 316, 457-486.
- Shakoor, R., Hassan, M. Y., Raheem, A. ve Rasheed, N., 2016, Wind farm layout optimization using area dimensions and definite point selection techniques, *Renewable Energy*, 88, 154-163.
- Shukla, U. P. ve Nanda, S. J., 2018, A binary social spider optimization algorithm for unsupervised band selection in compressed hyperspectral images, *Expert Systems with Applications*, 97, 336-356.
- Singh, D. ve Agrawal, S., 2016, Self organizing migrating algorithm with quadratic interpolation for solving large scale global optimization problems, *Applied Soft Computing*, 38, 1040-1048.
- Sun, G., Zhao, R. ve Lan, Y., 2016, Joint operations algorithm for large-scale global optimization, *Applied Soft Computing*, 38, 1025-1039.
- Sun, Y., Wang, X., Chen, Y. ve Liu, Z., 2018, A modified whale optimization algorithm for large-scale global optimization problems, *Expert Systems with Applications*, 114, 563-577.

- Sung, H.-K., Jung, N.-G., Huang, S.-R. ve Kim, J.-M., 2019, Application of Social Spider Algorithm to Optimize Train Energy, *Journal of Electrical Engineering & Technology*, 14 (1), 519-526.
- Surjanovic, S. ve Bingham, D., 2017, Virtual library of simulation experiments: Test functions and datasets. Retrieved July 3, 2019.
- Tawhid, M. A. ve Ali, A. F., 2017a, A Hybrid Flower Pollination and Genetic Algorithm for Minimizing the Non-Convex Potential Energy of Molecular Structure, *Trends Artif. Intell.*, 1 (1), 12-21.
- Tawhid, M. A. ve Ali, A. F., 2017b, A hybrid social spider optimization and genetic algorithm for minimizing molecular potential energy function, *Soft Computing*, 21 (21), 6499-6514.
- Tawhid, M. A. ve Ali, A. F., 2017c, A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function, *Memetic Computing*, 9 (4), 347-359.
- Tawhid, M. A. ve Dsouza, K. B., 2018, Hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection problems, *Applied Computing and Informatics*.
- Trunfio, G. A., Topa, P. ve Wąs, J., 2016, A new algorithm for adapting the configuration of subcomponents in large-scale optimization with cooperative coevolution, *Information Sciences*, 372, 773-795.
- Vapnik, V. N., 1995, The Nature of Statistical Learning Theory, *New York: Springer Verlag*.
- Wales, D. J. ve Scheraga, H. A., 1999, Global optimization of clusters, crystals, and biomolecules, *Science*, 285 (5432), 1368-1372.
- Wan, C., Wang, J., Yang, G., Li, X. ve Zhang, X., 2009, Optimal micro-siting of wind turbines by genetic algorithms based on improved wind and turbine models, *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 5092-5096.
- Wan, Y., Wang, M., Ye, Z. ve Lai, X., 2016, A feature selection method based on modified binary coded ant colony optimization algorithm, *Applied Soft Computing*, 49, 248-258.
- Wang, K.-P., Huang, L., Zhou, C.-G. ve Pang, W., 2003, Particle swarm optimization for traveling salesman problem, *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE cat. no. 03ex693)*, 1583-1585.
- Wang, L., Fu, X., Menhas, M. I. ve Fei, M., 2010, A modified binary differential evolution algorithm, In: *Life System Modeling and Intelligent Computing*, Eds: Springer, p. 49-57.

- Wang, Y., Cai, Z. ve Zhang, Q., 2011, Differential evolution with composite trial vector generation strategies and control parameters, *IEEE transactions on evolutionary computation*, 15 (1), 55-66.
- Xavier, V. A. ve Annadurai, S., 2019, Chaotic social spider algorithm for load balance aware task scheduling in cloud computing, *Cluster Computing*, 22 (1), 287-297.
- Yan, C., Ma, J., Luo, H. ve Patel, A., 2019, Hybrid binary coral reefs optimization algorithm with simulated annealing for feature selection in high-dimensional biomedical datasets, *Chemometrics and Intelligent Laboratory Systems*, 184, 102-111.
- Yan, X., Zhang, C., Luo, W., Li, W., Chen, W. ve Liu, H., 2012, Solve traveling salesman problem using particle swarm optimization algorithm, *International Journal of Computer Science Issues (IJCSI)*, 9 (6), 264.
- Yang, K. ve Cho, K., 2019, Simulated annealing algorithm for wind farm layout optimization: A benchmark study, *Energies*, 12 (23), 4403.
- Yang, X.-S. ve Deb, S., 2009, Cuckoo search via Lévy flights, *2009 World congress on nature & biologically inspired computing (NaBIC)*, 210-214.
- Yang, X.-S., 2010a, Firefly algorithm, stochastic test functions and design optimisation, *International Journal of Bio-Inspired Computation*, 2 (2), 78-84.
- Yang, X.-S., 2010b, A new metaheuristic bat-inspired algorithm, In: *Nature inspired cooperative strategies for optimization (NISCO 2010)*, Eds: Springer, p. 65-74.
- Yildiz, Y. E. ve Topal, A. O., 2019, Large scale continuous global optimization based on micro differential evolution with local directional search, *Information Sciences*, 477, 533-544.
- Yin, H. ve Gai, K., 2015, An empirical study on preprocessing high-dimensional class-imbalanced data for classification, *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, 1314-1319.
- Yip, P. P. ve Pao, Y.-H., 1995, Combinatorial optimization with use of guided evolutionary simulated annealing, *IEEE Transactions on neural networks*, 6 (2), 290-295.
- Zawbaa, H. M., Emary, E. ve Parv, B., 2015, Feature selection based on antlion optimization algorithm, *2015 Third World Conference on Complex Systems (WCCS)*, 1-7.
- Zhang, H. ve Zhou, J., 2016, Dynamic multiscale region search algorithm using vitality selection for traveling salesman problem, *Expert Systems with Applications*, 60, 81-95.

- Zhang, Y., Song, X.-f. ve Gong, D.-w., 2017, A return-cost-based binary firefly algorithm for feature selection, *Information Sciences*, 418, 561-574.
- Zhong, Y., Lin, J., Wang, L. ve Zhang, H., 2018, Discrete comprehensive learning particle swarm optimization algorithm with Metropolis acceptance criterion for traveling salesman problem, *Swarm and Evolutionary Computation*, 42, 77-88.
- Zhou, H., Song, M. ve Pedrycz, W., 2018, A comparative study of improved GA and PSO in solving multiple traveling salesmen problem, *Applied Soft Computing*, 64, 564-580.
- Zhou, Y., Luo, Q., Chen, H., He, A. ve Wu, J., 2015, A discrete invasive weed optimization algorithm for solving traveling salesman problem, *Neurocomputing*, 151, 1227-1236.



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Emine BAŞ
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Ermenek - 05.10.1983
Telefon : 0 536 032 29 04
e-mail : emineozcan@selcuk.edu.tr; ozcan_emine99@hotmail.com

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Ermenek Lisesi	2001
Üniversite	: Selçuk Üniversitesi/Bilgisayar Mühendisliği	2006
Yüksek Lisans	: Selçuk Üniversitesi/Bilgisayar Mühendisliği	2013
Doktora	: -	

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2007-2014	S.Ü. Huğlu Meslek Yüksekokulu	Öğretim Görevlisi
2015-	S.Ü. Kulu Meslek Yüksekokulu	Öğretim Görevlisi

UZMANLIK ALANI

Optimizasyon algoritmaları, veri madenciliği, makine öğrenmesi, algoritmalar

YABANCI DİLLER

İngilizce (orta seviye) (Yökdil-71.25)

YAYINLAR*

- (*) Baş, E., Ülker, E. (2020a). A binary social spider algorithm for continuous optimization task, Soft Computing, <https://doi.org/10.1007/s00500-020-04718-w>. (Doktora tezinden yapılmıştır)(SCI-E)
- (*) Baş, E., Ülker, E. (2020b). An efficient binary social spider algorithm for feature selection problem, Expert Systems With Applications 146, 113185. (Doktora tezinden yapılmıştır)(SCI-E)
- (*) Baş, E., Ülker, E. (2020c). Discrete social spider algorithm for the traveling salesman problem, Artificial Intelligence Review, DOI: 10.1007/s10462-020-09869-8. (Doktora tezinden yapılmıştır)(SCI-E)
- (*) Baş, E., Ülker, E. (2020d). A binary social spider algorithm for uncapacitated facility location problem, Expert Systems with Applications 161 (2020) 113618. (Doktora tezinden yapılmıştır)(SCI-E)
- (*) Baş, E., Ülker, E. (2020e). Improved Social Spider Algorithm For Large Scale Constrained Optimization, Artificial Intelligence Review, DOI: 10.1007/s10462-020-09931-5. (Doktora tezinden yapılmıştır)(SCI-E)
- (*) Baş, E., Ülker, E. (2020f). Improved Social Spider Algorithm For Minimizing Molecular Potential Energy Function, Konya Journal of Engineering Sciences, v.8, n.3, 618-642, DOI: 10.36306/konjes.788082. (Doktora tezinden yapılmıştır)(TR)
- (*) E. Özcan, E. Ülker, "RFID-Based Implementation of an E-Money and Identity Card", Int. J. of American Academic & Scholarly Research Journal (AASRJ), Vol. 5 No. 5, pp 193-204, 2013.(Yüksek lisans tezinden yapılmıştır)