



T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



İKİLİ ÖLÜM OYUNU OPTİMİZASYON
ALGORİTMASININ KÜME BİRLEŞİMLİ
SIRT ÇANTASI PROBLEMİNE
UYGULANMASI

Gülşen ORUCOVA BÜYÜKÖZ

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Temmuz-2025
KONYA
Her Hakkı Saklıdır

TEZ KABUL VE ONAYI

Gülşen ORUCOVA BÜYÜKÖZ tarafından hazırlanan “İkili Ölüm Oyunu Optimizasyon Algoritmasının Küme Birleşimli Sırt Çantası Problemine Uygulanması” adlı tez çalışması 07/07/2025 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Dr. Öğr. Üyesi Vahit TONGUR

.....

Danışman

Doç. Dr. Hüseyin HAKLI

.....

Üye

Dr. Öğr. Üyesi Ayşe Merve ACILAR

.....

Fen Bilimleri Enstitüsü Yönetim Kurulu’nun .../.../2025 gün ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Havvanur UÇBEYİAY
FBE Müdürü

TEZ BİLDİRİMİ

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

DECLARATION PAGE

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

Gülşen ORUCOVA BÜYÜKÖZ

07.07.2025

ÖZET

YÜKSEK LİSANS TEZİ

İKİLİ ÖLÜM OYUNU OPTİMİZASYON ALGORİTMASININ KÜME BİRLEŞİMLİ SIRT ÇANTASI PROBLEMİNE UYGULANMASI

Gülşen ORUCOVA BÜYÜKÖZ

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Hüseyin HAKLI

2025, 57 Sayfa

Jüri

Doç. Dr. Hüseyin HAKLI
Dr. Öğr. Üyesi Vahit TONGUR
Dr. Öğr. Üyesi Ayşe Meyve ACILAR

Klasik çözüm yöntemlerinin yetersiz kaldığı ya da yüksek hesaplama maliyeti gerektirdiği durumlarda, optimizasyon teknikleri ön plana çıkmaktadır. Optimizasyon, belirli kısıtlar altında bir hedef fonksiyonun en iyi değerinin bulunmasını amaçlar. Hesaplama karmaşıklığı kuramı kapsamında NP-zor (Non-deterministic Polynomial-time hard) olarak sınıflandırılan problemler, çözüm süresinin problem boyutuyla birlikte üstel olarak arttığı ve polinom zamanda çözümün genellikle mümkün olmadığı yapılar sunar. Bu nedenle, kesin çözüm yöntemlerinin uygulanabilir olmadığı durumlarda yaklaşık, sezgisel (heuristic), meta-sezgisel (metaheuristic) algoritmalar gibi alternatif yaklaşımlar geliştirilerek bu problemlere etkin çözümler üretilmektedir. Bu tür problemlerin çözümü, yalnızca teorik bilgisayar bilimi açısından değil, aynı zamanda üretim planlaması, ağ optimizasyonu, kaynak yönetimi ve yapay zekâ gibi birçok uygulamalı alanda da kritik öneme sahiptir. Etkin çözüm yöntemleri hem hesaplama süresini azaltmakta hem de karmaşık sistemlerin verimli şekilde yönetilmesini mümkün kılmaktadır.

Bu tez çalışmasında, NP-zor sınıfta yer alan Küme Birleşimli Sirt Çantası Problemi (Set-Union Knapsack Problem-SUKP) için Ölüm Oyunu Optimizasyonu (Battle Royale Optimization-BRO) algoritmasının ikili (binary) versiyonları önerilmiştir ve performansları karşılaştırılarak analiz edilmiştir. BRO algoritması sürekli arama uzayı için tasarlanmış bir meta-sezgisel algoritma olduğundan, ikili yapıdaki SUKP problemlerine doğrudan uygulanamamaktadır. Bu nedenle BRO algoritmasını ikili arama uzayına adapte edebilmek için iki farklı yaklaşımı önerilmiştir.

İlk yaklaşımda, BRO'nun temel işleyişine müdahale edilmeden, aday çözümlerin ikili yapıya dönüştürülmesi için 25 farklı transfer fonksiyonu (S, V, U, T, Z, O-şekilli ve Hiperbolik tanjant) kullanılmıştır. Bu yöntemde, algoritmanın arama dinamikleri korunarak, uygunluk hesaplanması öncesinde sürekli çözümler ikili yapıya dönüştürülmüştür. Önerilen algoritma 30 SUKP problemine uygulanmış elde edilen sonuçlar analiz edilmiştir. Sonuçlar O-şekilli TF'den olan O2-TF'nin en etkili çözümleri elde ettiğini göstermiştir.

İkinci yaklaşımda, BRO algoritmasının arama uzayı doğrudan ikili forma taşınmıştır. Çözüm çeşitliliği sağlamak için Mutasyon, arama uzayını daraltmak için ise etiket oranı gibi yeni bileşenler eklenerek algoritmanın temel adımları modifiye edilmiştir ve BROBin olarak adlandırılmıştır. Önerilen her iki yöntemde de arama uzayının dışında kalan geçersiz çözümleri onarmak onarım (repair) algoritması ve var olan çözümleri iyileştirme için iyileştirme (improvement) algoritması kullanılmıştır. BROBin algoritması 30 SUKP problemi üzerinde test edilmiştir. En iyi performans gösteren transfer fonksiyonu O2 ile BROBin algoritmasının çıktıları karşılaştırılarak değerlendirilmiştir. Deneysel bulgular, BROBin'in 30 SUKP probleminin tamamında daha etkili yaklaşımlar ürettiğini göstermiştir.

Bu tez çalışması, BRO algoritmasının ikili optimizasyon problemlerine uyarlanabilirliğini ortaya koyan çalışmalardan biri olması açısından önem taşımaktadır. Elde edilen sonuçlar, SUKP gibi karmaşık kombinatoriyal problemlerin çözümünde meta-sezgisel yöntemlerin etkinliğini artırmak için arama uzayı adaptasyonu ve operatör modifikasyonlarının etkili rol oynadığını vurgulamaktadır.

Anahtar Kelimeler: İkili optimizasyon, Küme birleşimli sırt çantası problemi, Meta-sezgisel algoritmalar, Ölüm oyunu optimizasyonu, Transfer fonksiyonları.



ABSTRACT

MS THESIS

IMPLEMENTATION OF BINARY BATTLE ROYALE OPTIMIZATION ALGORITHM TO SET UNION KNAPSACK PROBLEM

Gülşen ORUCOVA BÜYÜKÖZ

**NECMETTİN ERBAKAN UNIVERSITY INSTITUTE OF SCIENCE
THE DEGREE OF MASTER OF SCIENCE IN COMPUTER ENGINEERING**

Advisor: Assoc. Prof. Dr. Hüseyin HAKLI

2025, 57 Pages

Jury

Assoc. Prof. Dr. Hüseyin HAKLI

Assist. Prof. Dr. Vahit TONGUR

Assist. Prof. Dr. Ayşe Merve ACILAR

In cases where classical solution methods are insufficient or entail high computational costs, optimization techniques come to the forefront. Optimization aims to find the best value of an objective function under a set of constraints. Within the framework of computational complexity theory, problems classified as NP-hard (Non-deterministic Polynomial-time hard) exhibit a solution time that increases exponentially with problem size, making polynomial-time solutions generally infeasible. Therefore, in situations where exact solution methods are not applicable, alternative approaches such as approximate algorithms, heuristic, and metaheuristic techniques have been developed to provide effective solutions. Solving such problems is critically important not only from a theoretical computer science perspective but also in many applied domains such as production planning, network optimization, resource management, and artificial intelligence. Efficient solution strategies reduce computational time and enable the effective management of complex systems.

This thesis proposes binary versions of the Battle Royale Optimization (BRO) algorithm to solve the Set-Union Knapsack Problem (SUKP), which belongs to the NP-hard class. Since the BRO algorithm was originally designed for continuous search spaces, it cannot be directly applied to binary SUKP problems. Therefore, two different approaches are proposed to adapt the BRO algorithm to the binary search space.

In the first approach, 25 different transfer functions (S, V, U, T, Z, O-shaped and Hyperbolic tangent) were used to transform candidate solutions into binary structure without interfering with the basic operation of BRO. In this method, the search dynamics of the algorithm are preserved and the continuous solutions are converted to binary structure before the fitness calculation. The proposed algorithm has been applied to 30 SUKP problems and the results obtained have been analysed. The results show that O2-TF, which is one of the O-shaped TFs, obtains the most efficient solutions.

In the second approach, the search space of the BRO algorithm is moved directly to binary form. The basic steps of the algorithm are modified by adding new components such as mutation to provide solution diversity and labelling ratio to narrow the search space, and the algorithm is called BROBin. In both proposed methods, repair algorithm is used to repair infeasible solutions outside the search space and improvement algorithm is used to improve feasible solutions. The BROBin algorithm has been tested on 30 SUKP problems. The outputs of the BROBin algorithm were evaluated by comparing the best performing transfer function O2 with the outputs of the BROBin algorithm. The experimental results show that BROBin produces more efficient approximations for all the 30 SUKP problems.

This thesis is important as it is one of the studies that demonstrate the adaptability of the BRO algorithm to binary optimization problems. The results obtained emphasise that search space adaptation and operator modifications play an effective role in improving the effectiveness of meta-heuristics in solving complex combinatorial problems such as the SUKP.

Keywords: Battle royale optimization, Binary optimization, Metaheuristic algorithms, Set-union knapsack problem, Transfer functions.



ÖNSÖZ

Bu tez çalışmasının ortaya çıkmasında desteklerini benden esirgemeyen bilgi ve tecrübesiyle bana yol gösteren sevgili danışman hocam Doç. Dr. Hüseyin HAKLI'ya;

Tez savunma jürimde bulunan ve yorumlarıyla çalışmama katkı sağlayan Dr. Öğr. Üyesi Ayşe Merve ACILAR ve Dr. Öğr. Üyesi Vahit TONGUR hocalarıma;

Çalışmalarım sırasında anlayış ve sabır gösteren eşim Tolgahan BÜYÜKÖZ'e teşekkürlerimi sunarım.

Gülşen ORUCOVA BÜYÜKÖZ
KONYA-2025

İÇİNDEKİLER

ÖZET	iv
ABSTRACT	vi
ÖNSÖZ	viii
İÇİNDEKİLER	ix
ŞEKİL LİSTESİ	x
TABLO LİSTESİ	xi
KISALTMALAR	xii
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	5
3. MATERYAL VE YÖNTEM	10
3.1. Küme Birleşimli Sırt Çantası Problemi	10
3.2. Ölüm Oyunu Optimizasyon Algoritması (Battle Royale Optimization-BRO)....	12
3.2.1. Ölüm oyununun stratejileri	12
3.2.2. Ölüm oyunu optimizasyon algoritmasının adımları	13
4. ÖLÜM OYUNU OPTİMİZASYON ALGORİTMASININ İKİLİ UYGULAMALARI	17
4.1. Transfer Fonksiyonları Yardımıyla İkili Ölüm Oyunu Optimizasyon Algoritması	17
4.2. Onarım Algoritması (Repair Algorithm)	22
4.3. İyileştirme Algoritması (Improvement Algorithm)	23
4.4. Ölüm Oyunu Optimizasyon Algoritmasının İkili Uzaya Aktarımı	24
5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA	28
5.1. Transfer Fonksiyonları Yardımıyla Elde Edilen Sonuçlar	28
5.2. BROBin Algoritması ile Elde Edilen Sonuçlar	34
5.2.1. Parametre ayarlaması	34
6. SONUÇLAR VE ÖNERİLER	39
6.1. Sonuçlar	39
6.2. Öneriler	40
KAYNAKLAR	42

ŞEKİL LİSTESİ

Şekil 3.1. SUKP problemindeki bir nesnenin içeriği.....	11
Şekil 3.2. Ölüm Oyunu Optimizasyon Algoritmasının akış diyagramı.....	16
Şekil 4.1. S-şekilli ve V-şekilli transfer fonksiyonlarının grafiği.....	19
Şekil 4.2. U-şekilli ve Taper-şekilli transfer fonksiyonlarının grafiği.....	20
Şekil 4.3. Z-şekilli ve Hiperbolik Tanjant transfer fonksiyonlarının grafiği.....	22
Şekil 4.4. Etiket oranının etkisinin görselleştirilmesi.....	25
Şekil 4.5. Tekdüze (Uniform) çaprazlama ile yeni birey oluşturma.....	26



TABLO LİSTESİ

Tablo 4.1. S- şekilli ve V-şekilli transfer fonksiyonları	18
Tablo 4.2. U-şekilli ve Taper-şekilli transfer fonksiyonları	20
Tablo 4.3. Z-şekilli ve O-şekilli transfer fonksiyonları	21
Tablo 4.4. Hiperbolik Tanjant Sigmoid transfer fonksiyonları	21
Tablo 5.1. S-şekilli ve V-şekilli TF'ler için elde edilen sonuçlar.....	29
Tablo 5.2. U-şekilli ve Taper-şekilli ve Hiperbolik tanjant TF'ler için elde edilen sonuçlar.....	31
Tablo 5.3. O-şekilli ve Z-şekilli TF'ler için elde edilen sonuçlar.....	32
Tablo 5.4. Her problem için en iyi sonuç elde TF'ler.....	33
Tablo 5.5. TF için elde edilen Friedman sıralaması	33
Tablo 5.6. TF'lere göre elde edilen sonuçların koşullu biçimlendirme ile ısı haritası ...	34
Tablo 5.7. Mutasyon oranı ve Etiket oranına göre 6 SUKP problemi için elde edilen sonuçlar.....	35
Tablo 5.8. BROTF-O2 ile BROBin sonuçlarının karşılaştırılması.....	36
Tablo 5.9. Farklı algoritmalar ile BROBin algoritmasının SUKP problemleri için karşılaştırılması	38

KISALTMALAR

ABC	: Yapay Arı Kolonisi (Artificial Bee Colony)
BABC	: Yapay Arı Kolonisi Algoritmasının İkili Versiyonu (Binary Artificial Bee Colony)
BRO	: Ölüm Oyunu Optimizasyonu (Battle Royala Optimization)
BROBin	: Ölüm Oyunu Optimizasyonunun İkili Versiyonu
bWSA	: İkili Ağırlıklı Süperpozisyon Çekimi (Binary Weighted Superposition Attraction)
CS	: Guguk Kuşu Arama (Cuckoo Search)
DE	: Diferansiyel Evrim (Differential Evolution)
GA	: Genetik Algoritma (Genetic Algorithm)
gPSO	: Genetik Algoritma ve Parçacık Sürü Optimizasyonu (Genetic Algorithm and Particle Swarm Optimization)
KP	: Sırt Çantası Problemi (Knapsack Problem)
MS	: Güve arama (Moth Search)
MS-GROA	: S-GROA Algoritmasının Geliştirilmiş Versiyonu (Modified Greedy Repair and Optimization Algorithm)
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
S-GROA	: Açgözlü Onarım ve Optimizasyon Algoritması (Greedy Repairing and Optimization Algorithm)
SUKP	: Küme Birleşimli Sırt Çantası Problemi (Set Union Knapsack Problem)
TF	: Transfer Fonksiyonu

1. GİRİŞ

Geleneksel algoritmalarla etkin biçimde çözülemeyen ya da çözüm süreci hesaplama açısından aşırı derecede maliyetli olan problemlerde, optimizasyon kuramı temel bir yaklaşım olarak öne çıkmaktadır. Optimizasyon, belirli kısıtlar altında bir amaç fonksiyonunun minimize veya maksimize edilmesi sürecini ifade eder. Özellikle hesaplama karmaşıklığı kuramı çerçevesinde NP-zor (Non-deterministic Polynomial-time hard) sınıfına dâhil edilen problemler, polinom zamanda çözüm elde edilmesinin mümkün olmadığı ya da oldukça düşük olasılıkla mümkün olabildiği problemler olarak tanımlanmakta; bu problemlerin çözüm süresi, problem boyutu büyüdükçe üstel olarak artış göstermektedir. Bu bağlamda, kesin (deterministik) çözüm yöntemlerinin uygulanamadığı durumlarda, yaklaşık algoritmalar, sezgisel (heuristics) yaklaşımlar ve özellikle son yıllarda geliştirilen meta-sezgisel (metaheuristics) algoritmalar, alternatif çözüm yöntemleri olarak ön plana çıkmaktadır. Bilgisayar mühendisliği, yapay zekâ, operasyonel araştırmalar ve kombinatoriyel optimizasyon gibi pek çok disiplinde sıklıkla karşılaşılan bu tür problemler, yalnızca uygulamalı çözümler açısından değil, aynı zamanda algoritma tasarımının kuramsal sınırlarını zorlaması bakımından da önemli bir araştırma alanı oluşturmaktadır. Bu tür problemlerin çözümüne yönelik geliştirilen yöntemler yalnızca akademik düzeyde teorik katkılar sunmakla kalmamakta, aynı zamanda gerçek dünya uygulamalarında da yüksek düzeyde etki yaratmaktadır. Özellikle kaynak tahsisi, rota planlaması, zaman çizelgeleme, ağ tasarımı ve makine öğrenmesi gibi alanlarda karşılaşılan NP-zor problemler, karar verme süreçlerinin etkinliğini doğrudan etkilemektedir. Dolayısıyla, bu problemlere yönelik geliştirilen sezgisel ve meta-sezgisel algoritmalar, hem hesaplama sürelerini azaltmakta hem de pratikte uygulanabilir kaliteli çözümler elde edilmesini mümkün kılmaktadır.

NP-Zor problem sınıfında yer alan problemlerden biri de sırt çantası (knapsack) problemidir. Sırt çantası problemi belirli bir ağırlıktan fazlasını taşıyamayan bir sırt çantasının içini nesnelere doldururken her bir nesnenin bir ağırlığı ve bir fayda değeri olduğu varsayıldığında bu çantayı hem çantanın maksimum ağırlığı aşılmayacak hem de içindeki nesnelerin değeri toplamı olabildiğince en yüksek değerde olacak şekilde doldurmayı hedeflemeye dayanır. Hırsız örneği olarak da bilinen problemde hırsız çantasına ağırlıkça en az, pahaca en çok eşyayı doldurmak istemektedir. Sırt çantası problemi sermaye bütçeleme, portföy seçimi, proje seçimi, kesme problemleri, paketleme

problemleri, açık anahtarlı şifreleme sistemleri gibi birçok gerçek dünya problemlerinde karşımıza çıkmaktadır. Ayrıca çoğu büyük ölçekli problemlerde alt problem olarak yine sırt çantası problemi bulunmaktadır. Bu nedenle sırt çantası problemlerinin ve çözüm tekniklerinin geliştirilmesinin önemi her geçen gün artmaktadır.

Literatürde birkaç çeşit sırt çantası problemi bulunmaktadır. Bunlardan en yaygın olanı 0-1 Sırt Çantası Problemidir (0-1 KP). 0-1 KP'nin genelleştirilmiş hali Küme Birleşimli Sırt Çantası Problemidir (Set-Union Knapsack Problem-SUKP). Bu tez çalışmasında SUKP Problemine meta-sezgisel algoritma uygulanarak optimuma yakın çözümler elde edilmeye çalışılmıştır. SUKP problemlerinin temsil şekli 0 ve 1'lerden oluştuğu için problemin aday çözümleri ikili yapıya sahiptir. Bu nedenle ilgili problemi çözmek için ikili formda aday çözümlere göre arama yapan algoritmaya ihtiyaç duyulmaktadır.

Analitik olarak çözülemeyen problemleri veya çözümünü çok maliyetli olan problemleri çözmek için meta-sezgisel algoritmalar kullanılmaktadır. Bu algoritmalar kabul edilebilir zamanda makul çözümler sunmaktadır. Meta-sezgisel algoritmalar, oluşumunda esinlendiği durumlara göre evrimsel veya sürü zekâsı algoritmaları olarak sınıflandırılmaktadır. Evrimsel algoritmalar, biyolojik evrim süreçlerinden ilham almaktadır. Bu algoritmalar, doğadaki doğal seçim, mutasyon ve çaprazlama mekanizmalarını taklit ederek çözümler üretmektedir. Popülasyon tabanlı çalışan bu algoritmalar iteratif bir şekilde çözümü iyileştirmeye odaklanmaktadır. En yaygın evrimsel algoritmalara örnek olarak Genetik Algoritma (GA), Diferansiyel Evrim (DE) Algoritması verilebilir. Sürü zekâsı algoritmaları ise doğadaki toplulukların (karıncalar, arılar, kuşlar vb.) kolektif hareketlerini model almaktadır. Bu algoritmanın işleyişi bir grup ajanın birbirleriyle basit kurallar çerçevesinde etkileşime girmesiyle optimal çözümler elde etmeye dayanmaktadır. Parçacık Sürü Optimizasyonu (PSO), Karınca Kolonisi Optimizasyonu (Ant Colony Optimization-ACO), Yapay Arı Kolonisi Optimizasyonu (ABC), Ateşböceği Algoritması (Firefly Algorithm-FA) gibi algoritmalar sürü zekâsı algoritmalarından bazılarıdır.

2021 yılında dijital bir oyundan esinlenerek geliştirilen yeni bir meta-sezgisel algoritma olan Ölüm Oyunu Optimizasyon (Battle Royale Optimization-BRO) Algoritması (Rahkar Farshi, 2021) tarafından önerilmiştir. Bu algoritma, "Battle Royale"

olarak bilinen ve her bireyin bir asker (oyuncu) olarak temsil edildiği, hayatta kalmak için en güvenli ve stratejik bölgeye ulaşmayı amaçlayan dijital oyun türünden ilham alınarak tasarlanmıştır. Popülasyon tabanlı bir yapıya sahip olan BRO algoritmasında bireyler, en iyi çözüm alanına ulaşmak için çeşitli stratejiler uygulayan oyuncular şeklinde modellenmektedir.

Başlangıçta sürekli optimizasyon problemleri için geliştirilen BRO algoritması, daha sonra ikili (binary) yapıya sahip problemlere de uyarlanmıştır (Akan ve ark. 2022). Ancak ikili yapıdaki problemlere geçişte kullanılan dönüşüm mekanizmasının yetersiz kaldığı ve iyileştirmeye açık olduğu görülmüştür. Bu sebeple, algoritmanın ikili problemlerdeki etkinliğini artırmak amacıyla sürekli yapıdan ikili yapıya geçiş sürecinde farklı yaklaşımlar araştırılmıştır. Tez çalışmasının temel amacı BRO algoritması ile ikili yapıya sahip olan SUKP problemlerinin çözümüne yönelik daha etkili bir arama algoritması önermektir. Sürekli yapıdan ikili yapıya geçişte öncelikle çeşitli transfer fonksiyonları (TF) kullanılmıştır. Buna ek olarak algoritmanın performansını artırmak amacıyla Onarım (Repair) ve İyileştirme (Improvement) adımları da sürece entegre edilmiştir. Çalışmada, toplam 25 farklı transfer fonksiyonu kullanılarak 30 adet SUKP problemi çözülmüş ve elde edilen sonuçlar karşılaştırılmıştır. Böylece hangi TF fonksiyonunun daha etkili olduğu istatistiksel olarak analiz edilmiştir. Elde edilen çözümler literatürde daha önce SUKP problemleri için raporlanmış sonuçlarla karşılaştırılmış ve BRO algoritmasının önerilen ikili haliyle rekabetçi çözümler üretebildiği ortaya konulmuştur.

BRO algoritmasını ikili yapıya dönüştürmenin diğer bir yolu ise algoritmanın doğrudan arama uzayını ikili yapıya taşımaktır. Çalışmanın uygulama kısmının ikinci aşamasında, BRO algoritmasının bazı temel adımları ikili yapıya uygun şekilde yeniden yapılandırılmıştır. Bu dönüşüm süreci ve yapılan uyarlamalara ilişkin detaylı açıklamalar Bölüm 4’de sunulmuştur. İkili yapıya uyarlanmış versiyon için BRO algoritması, 30 farklı SUKP problemi üzerinde uygulanmış ve elde edilen sonuçlar, daha önce transfer fonksiyonları kullanılarak ulaşılan en iyi sonuçlarla karşılaştırılmıştır. Ayrıca, literatürde yer alan farklı meta-sezgisel algoritmaların SUKP problemlerine ilişkin performansları da dikkate alınarak, önerilen İkili BRO algoritmasının etkinliği kapsamlı bir şekilde analiz edilmiştir.

Tez 6 bölümden oluşmaktadır. Giriş bölümünde tezin konusu ve literatürde bulunan bazı optimizasyon algoritmaları hakkında kısa bilgiler verilmiştir. Kaynak Araştırması bölümünde SUKP problemlerine uygulanan optimizasyon algoritmalarından bahsedilmiştir. Üçüncü bölüm SUKP problemlerinin tanıtımına ve orijinal BRO algoritmasının anlatımına ayrılmıştır. Dördüncü bölümde BRO algoritmasının ikili problemler için önerilen versiyonları üzerinde durulmuştur. Önerilen ikili algoritmaların SUKP problemleri üzerindeki test sonuçları, karşılaştırma analizleri ise 5. bölümde ele alınmıştır. Tezden elde edilen sonuçları ve gelecek çalışmalar için önerilere 6. bölümde yer verilmiştir.



2. KAYNAK ARAŞTIRMASI

Küme Birleşimli Sırt Çantası problemi (Arulselvan, 2014; Goldschmidt ve ark., 1994) standart 0-1 sırt çantası probleminin (Bansal ve Deep, 2012) genişletilmiş versiyonudur ve NP-zor sınıfındadır. SUKP problemleri veri tabanı bölümlenme (Navathe ve ark., 1984), finansal karar verme (Arulselvan, 2014; Goldschmidt ve ark., 1994; Kellerer ve ark., 2004) esnek üretim makineleri (Goldschmidt ve ark., 1994; Tang ve Denardo, 1988), geniş siber sistemlerde (Manghui Tu, 2016), veri akışı sıkıştırma (Manghui Tu, 2016; Yang ve ark., 2016) gibi birçok gerçek dünya probleminde kullanılmaktadır.

Literatürde bazı meta-sezgisel optimizasyon algoritmaları, SUKP problemlerini çözmek için kullanılmıştır. Sürekli problemler için tasarlanmış optimizasyon algoritmalarının ikili yapıda olan SUKP problemine uygulanabilmesi adına, öncelikle bu algoritmaların ikili çözüm uzayına dönüştürülmesi gerekmektedir (Tang ve Denardo, 1988). Bu dönüşüm süreci transfer fonksiyonları (Feng ve ark., 2018), açı modülasyonu (Pampara ve ark., 2005), kuantum ilhamlı bitler (Nezamabadi-Pour, 2015) vb. yollar ile yapılabilmektedir. Transfer fonksiyonları sayesinde, algoritmanın genel yapısına herhangi bir müdahalede bulunmadan, ikili arama uzayında etkin bir şekilde kullanılabilmesi mümkün hale gelmektedir. Yapay arı kolonisi algoritmasının ikili versiyonu (BABC) (He ve ark., 2018) çalışmada sunulmuş ve SUKP problemlerine uygulanmıştır. Sürekli problemler için geliştirilen yapay arı kolonisi algoritmasının ikili biçimi, çözüm bileşenlerinin yalnızca 0 veya 1 değerlerini alacağı şekilde tanımlanan bir örten fonksiyon aracılığıyla elde edilmiştir. Geçersiz çözümleri, yani arama uzayının dışında kalan çözümleri elemek amacıyla, açgözlü bir stratejiye dayanan S-GROA (açgözlü stratejiye sahip onarım algoritması) uygulanmıştır. Öte yandan, Ayrık Güve Arama Algoritması başlangıçta sürekli optimizasyon problemleri için tasarlanmış olan Güve Arama (MS) algoritmasının, transfer fonksiyonları yardımıyla ayrık çözüm uzayına uyarlanması sonucunda geliştirilmiştir (Feng ve ark., 2018). Çözüm kalitesini artırmak amacıyla onarım algoritmasından yararlanılmıştır. Ayrık MS algoritması kullanılarak 15 farklı SUKP örneği, S-şekilli, V-şekilli ve O-şekilli olmak üzere toplam 12 transfer fonksiyonu aracılığıyla çözülmüş; elde edilen sonuçlar karşılaştırmalı olarak analiz edilmiştir. Elde edilen sonuçlardan O4 transfer fonksiyonunun en iyi olduğu görülmüştür.

Sürü zekâsı temelli optimizasyon algoritmaları, karmaşık problemlerin çözümünde etkili ve güçlü bir yaklaşım sunmaktadır. Genetik algoritma ve parçacık sürü optimizasyonundan esinlenerek geliştirilen gPSO algoritması, basit ancak etkili bir ikili sürü zekâsı yöntemi olup, transfer fonksiyonlarına gerek duymayacak biçimde tasarlanmıştır (Ozsoydan ve Baykasoglu, 2019). Onarım algoritmasının kullanılmasıyla, geçersiz çözümler etkin bir şekilde engellenmiştir. SUKP problemleri üzerinde yapılan testler, algoritmanın daha önceki yöntemlerle elde edilen sonuçlara kıyasla daha başarılı performans sergilediğini ortaya koymuştur. Hibrit İkili Parçacık Sürü Optimizasyonu algoritması, Tabu Arama yöntemiyle birleştirilerek HBPSO/TS (Hibrit İkili Parçacık Sürü Optimizasyon Algoritması ile Tabu Arama Algoritması-Hybrid Binary Particle Swarm Optimization with Tabu Search Algorithm) adıyla geliştirilmiş ve SUKP problemi üzerinde test edilmiştir (Lin ve ark., 2019). Arama süreci sırasında çözümlerin uygunluğunu değerlendirmek amacıyla ceza fonksiyonu kullanılmış ve bu sayede geçerli çözüm uzayının sınırlarının keşfedilmesi hedeflenmiştir. Algoritmada, tabu tabanlı bir mutasyon yöntemiyle açgözlü bir arama gerçekleştirilmiş, sömürü kapasitesini artırmak için özel olarak tabu arama mekanizması tasarlanmıştır. HBPSO/TS algoritması, 30 farklı örneği içeren üç ayrı test seti üzerinde değerlendirilmiştir.

Wei ve Hao tarafından geliştirilen Yinelemeli İki Aşamalı Yerel Arama algoritması (Iterated Two-Phase Local Search – I2PLS), SUKP problemlerine yönelik olarak geliştirilmiştir. Bu algoritma yalnızca geçerli çözümler üzerinde işlem yaparak doğrudan ikili arama uzayında çalışmakta, böylece sürekli çözüm uzayından ikili uzaya dönüştürme ihtiyacını ortadan kaldırmaktadır. Algoritma iki temel fazdan oluşmaktadır: İlki, değişken komşuluk araması ve Tabu Arama tekniklerinin kullanıldığı yerel optimum keşfi aşaması; ikincisi ise frekans tabanlı pertürbasyon stratejisi ile daha önce keşfedilmemiş arama bölgelerine yönelen yerel optimumdan kaçış aşamasıdır. Algoritma, 30 standart test problemi üzerinde test edilmiş ve bu problemlerin 18'inde literatürdeki en iyi sonuçları geliştirirken, kalan 12 problemde mevcut en iyi sonuçları tekrarlamayı başarmıştır. Ayrıca, genel amaçlı çözümleyici olan CPLEX ile yapılan karşılaştırmalar, algoritmanın büyük ölçekli problemler karşısında daha verimli olduğunu göstermiştir (Wei ve Hao, 2019). SUKP problemini çözmek amacıyla geliştirilen Geliştirilmiş Güve Arama Algoritması (Enhanced Moth Search-EMS), diferansiyel mutasyon stratejisini global uyum arama mekanizmasıyla birleştirerek Gelişmiş Etkileşim Operatörü (EIO) adı verilen yeni bir yapı ortaya koymaktadır (Feng ve ark.,

2019). Orijinal çekirdek tabanlı arama stratejisi ile etkili yerel arama sürecinin birleştirilmesiyle oluşturulan çekirdek tabanlı tabu arama algoritması SUKP problemlerine uygulanmıştır (Wei ve Hao, 2021a). Hibrit Parçacık Sürü Optimizasyonu algoritması (Hybrid Particle Swarm Optimization Based Algorithm-HSPO-BA) temel parçacık sürü optimizasyonunu (PSO) yerel arama prosedürleriyle birleştirerek çözüm kalitesini artırmayı hedeflemektedir. PSO'nun hızlı yakınsama ve yerel optimumda takılma sorunlarını aşmak için iki aşamalı bir iyileştirme stratejisi kullanılmıştır. HSPO-BA algoritması, literatürde yer alan 30 klasik ve 18 büyük ölçekli SUKP problemi üzerinde test edilmiştir (I. Dahmani, 2020).

Zhao ve Hifi tarafından önerilen Uyarlanabilir Evrimsel Arama Tabanlı Yöntem (Adaptive Evolutionary Search-Based Method-AES-BM), elit ve çeşitlendirilmiş çözümlerden oluşan bir arşiv kümesi oluşturarak çözüm sürecini başlatmakta; bu arşivi Hamming mesafesi, skor fonksiyonu ve farklı iyileştirme operatörleri yardımıyla dinamik biçimde güncellemektedir. Yerel arama aşamasında çoklu komşuluk yapısına sahip tabu arama stratejisi kullanılarak çözüm uzayı etkili biçimde taranmaktadır. Ayrıca, birleşim (fusion) ve referans güncelleme stratejileri ile yeni çözümler üretilerek hem çeşitlilik korunmakta hem de yerel minimumlardan kaçınılması hedeflenmektedir. AES-BM algoritması, 84 adet orta ve büyük ölçekli SUKP örneği üzerinde test edilmiş elde edilen algoritmanın çözüm kalitesi ve istatistiksel anlamlılık açısından güçlü bir performans sergilediğini göstermiştir (Zhao ve Hifi, 2024b). İşbirlikçi Balina Optimizasyonu Algoritması (Cooperative Whale Optimisation Algorithm-CWOA) klasik Balina Optimizasyon algoritmasını temel alarak, açgözlü bir başlangıç prosedürü, sigmoid tabanlı ikili dönüşüm mekanizması, özel bir onarma ve iyileştirme operatörü ile güçlendirilmiş ve çoklu komşuluk yapısına dayalı Tabu Arama stratejisi (Multi-Neighborhood-Based Tabu Search-MTS) ile geliştirilmiştir. Algoritma hem sürekli hem de ikili çözüm alanlarını etkili şekilde birleştirerek, geçerli ve kaliteli çözümler üretmiştir (Zhao ve Hifi, 2024a).

SUKP problemleri gibi büyük ve karmaşık çözüm uzayına sahip problemlere yönelik paralel işlem temelli yaklaşımlar da geliştirilmiştir. Bu kapsamda, CUDA (Birleşik Cihaz Mimarisi Hesaplama-Compute Unified Device Architecture) tabanlı paralel yerel arama algoritması (CuPLS), SUKP problemi çözümüne yönelik GPU destekli etkili bir yöntem sunmaktadır. Algoritmanın temel avantajı, paralel işleme

sayesinde çözüm uzayının çok daha hızlı ve kapsamlı şekilde taranabilmesidir. CuPLS algoritmasında, her bir GPU bloğu farklı bir başlangıç noktasından aramaya başlamakta ve blok içinde elde edilen en iyi çözümler üzerinden çaprazlama (crossover) uygulanmaktadır. Çözümler ikili biçimde temsil edilmekte ve geçerlilikleri her iterasyonda kontrol edilerek gerektiğinde S-GROA adlı onarma operatörü ile düzeltilmektedir. Üç farklı çaprazlama stratejisi test edilmiş; bunlar arasında en başarılı sonuçlar, çözüm kalitesi ve hesaplama süresi açısından tekdüze (uniform) çaprazlama tekniği ile elde edilmiştir (Sonuç ve Özcan, 2024).

Son dönemde, SUKP problemi için geliştirilen çeşitli meta-sezgisel algoritmaların yanı sıra, bu yaklaşımların makine öğrenmesiyle bütünleştirildiği çalışmalara da literatürde rastlanmaktadır. Makine Öğrenmesi Destekli İkili Guguk Kuşu Arama Algoritması (Machine Learning Cuckoo Search Binary Algorithm-MLCSBO) sürekli çözüm uzayında çalışan geleneksel Cuckoo Search (CS) algoritması ve k-ortalama kümeleme yöntemi ile ikili forma dönüştürülmüştür. Ayrıca çözüm kalitesini artırmak amacıyla algoritmaya açgözlü (greedy) başlatma operatörü ve yerel arama operatörü eklenmiştir. Önerilen yöntem, hem keşif (exploration) hem de sömürü (exploitation) yeteneklerini geliştirerek çözüm uzayında daha dengeli bir arama gerçekleştirmektedir. Algoritma, 30 küçük ve 30 büyük SUKP örneği üzerinde test edilmiş; özellikle küçük ve orta ölçekli örneklerde mevcut yöntemlere kıyasla daha başarılı sonuçlar elde etmiştir (García ve ark., 2021). (García ve ark., 2023) çalışmasında, SUKP problemleri için meta-sezgisel optimizasyon algoritmalarını iyileştirmek amacıyla başlangıç çözüm stratejileri incelenmiştir. Çalışma, Sin-Cos Algoritması (Sine Cosine Algorithm-SCA) ile k-ortalama yöntemi birleştirilerek, üç farklı başlangıç çözüm stratejisi (rastgele, açgözlü ve ağırlıklı) değerlendirilmiştir. Yapılan araştırma, başlangıç stratejilerinin algoritmanın performansını etkilediğini göstermiş, özellikle ağırlıklı başlangıç yönteminin diğer iki stratejiden daha etkili olduğu bulunmuştur. Özsoydan ve Gölcük tarafından önerilen çalışmada, Q-learning algoritması kullanılarak farklı meta-sezgisel yöntemler (Genetik Algoritma, Parçacık Sürü Optimizasyonu ve genetik tabanlı PSO) değerlendirilmiş; bu algoritmalara başarı durumlarına göre ödül veya ceza uygulanarak, arama sürecinde daha verimli olan yöntemlerin daha sık tercih edilmesi sağlanmıştır. Ayrıca, başlangıç çözüm üretimi, çeşitliliği koruyan TRIM (Tetiklenen Rastgele Göçmenler-Triggered Random Immigrants) mekanizması, yerel arama operatörleri ve sömürü odaklı özel bir mutasyon stratejisi gibi yardımcı bileşenler de algoritmaya entegre edilmiştir. SUKP problemleri

üzerine yapılan deneysel çalıřmalar, makine öğrenmesi ve meta-sezgisel optimizasyonun bütünleřtiđi bu yenilikçi yaklaşımın literatüre önemli bir katkı sunduđunu ortaya koymuřtur (Ozsoydan ve Gölcük, 2023).



3. MATERYAL VE YÖNTEM

3.1. Küme Birleşimli Sırt Çantası Problemi

Küme Birleşimli Sırt Çantası Problemi (SUKP), 0-1 sırt çantası probleminin genişletilmiş bir biçimidir. Bu modelde, her nesneye ait olan eleman kümeleri göz önünde bulundurularak seçimler yapılmaktadır.

$U = \{1,2, \dots, n\}$ elemanlar kümesi, $S = \{1,2, \dots, m\}$ nesnelere kümesi olmak üzere her $i \in S$ ($i = 1,2, \dots, m$) nesnesi $p_i > 0$ faydasına sahip $U_i \subseteq U$ alt kümesi karşılık gelmektedir. Her bir $j \in U$ ($j = 1,2, \dots, n$) elemanın $w_j > 0$ ağırlığı vardır. Seçilmiş nesnelere oluşan bir $A \subseteq S$ kümesi için A 'nın kazancı (getirisi, profit) şu şekilde tanımlanır:

$$P(A) = \sum_{i \in A} p_i \quad (3.1)$$

A 'nın ağırlığı,

$$W(A) = \sum_{j \in \bigcup_{i \in A} U_i} w_j \quad (3.2)$$

denklemler ile tanımlanır. SUKP problemlerinde ilişki matrisi $M = (r_{ij})$, ($i = 1,2, \dots, m$, $j = 1,2, \dots, n$) nesnelere ile onlara ait elemanlar arasındaki bağlantıları temsil eder. Bir nesnenin belirli bir elemanı içerdiği durumlarda matris değeri 1, içermediği durumlarda ise 0 olarak tanımlanır.

SUKP probleminin amaç fonksiyonu $P(S^*)$ değerini en büyük yapacak olan $S^* \subseteq S$ alt kümesini belirlemektir. K sırt çantasının kapasitesini temsil etmek üzere, seçilen alt kümenin toplam ağırlığı için $W(S^*) \leq K$ kısıtı sağlanmalıdır (Kellerer ve ark., 2004), (Goldschmidt ve ark., 1994).

Bu çalışmada veri kümesi olarak SUKP problemleri kullanılacaktır. Bu problemlere http://www.info.univ-angers.fr/pub/hao/SUKP_KBTS.html sayfasından ulaşılabilir. Bu veri kümesi 30 problemten oluşan iki grup şeklindedir. İlk grupta nesne sayısı maksimum 500'e ulaşırken, 2. grupta nesne sayısı 1000'e ulaşmaktadır. Çalışmamızda kullanılan problemlerin ismi ve çanta kapasitesi Tablo 3.1' de verilmiştir.

Tablo 3.1. SUKP problemlerinin ismi ve kapasiteleri

Problem No	Problem Adı	Kapasite	Problem No	Problem Adı	Kapasite
P1	sukp_85_100_0.10_0.75	12180	P16	sukp_300_285_0.15_0.85	40801
P2	sukp_85_100_0.15_0.85	14982	P17	sukp_300_300_0.10_0.75	38289
P3	sukp_100_85_0.10_0.75	12015	P18	sukp_300_300_0.15_0.85	45914
P4	sukp_100_85_0.15_0.85	12405	P19	sukp_385_400_0.10_0.75	49815
P5	sukp_100_100_0.10_0.75	11223	P20	sukp_385_400_0.15_0.85	57687
P6	sukp_100_100_0.15_0.85	15194	P21	sukp_400_385_0.10_0.75	50856
P7	sukp_185_200_0.10_0.75	25405	P22	sukp_400_385_0.15_0.85	56538
P8	sukp_185_200_0.15_0.85	28159	P23	sukp_400_400_0.10_0.75	49822
P9	sukp_200_185_0.10_0.75	22809	P24	sukp_400_400_0.15_0.85	57856
P10	sukp_200_185_0.15_0.85	25828	P25	sukp_485_500_0.10_0.75	62516
P11	sukp_200_200_0.10_0.75	25630	P26	sukp_485_500_0.15_0.85	71655
P12	sukp_200_200_0.15_0.85	29583	P27	sukp_500_485_0.10_0.75	60351
P13	sukp_285_300_0.10_0.75	38922	P28	sukp_500_485_0.15_0.85	67506
P14	sukp_285_300_0.15_0.85	44806	P29	sukp_500_500_0.10_0.75	63902
P15	sukp_300_285_0.10_0.75	36126	P30	sukp_500_500_0.15_0.85	73927

Tablo üzerindeki her bir problem adında yer alan ilk sayı, problemde bulunan toplam nesne sayısını (m); ikinci sayı ise her bir nesnenin içerdiği toplam eleman sayısını (n) ifade etmektedir. Verilen son iki değerlerden α , M matrisindeki 1 elemanının yoğunluğunu ve β ağırlık / kapasite oranını belirtmektedir (He ve ark., 2018). Problemde nesnelerin içeriğindeki elemanları gösteren ilişik matrisi bulunmaktadır. Örneğin sukp_85_100_0.10_0.75 probleminde 85 nesne, kapasitesi 12180 olan bir çantaya yerleştirilecektir. Bu problemin ilişik matrisi 85x100 boyutundadır ve 85 nesne için her nesnenin 100 elemandan nasıl oluştuğunu göstermektedir. Bir nesneyi temsili olarak 100 elemandan oluşacak şekilde aşağıdaki gibi gösterebiliriz. Buradaki 1 olan elemanlar nesnede bulunan elemanlardır. Aşağıdaki örneğe göre bu nesnede 2, 3, ..., 93, 96, 98, 99, 100 numaralı elemanlar bulunmaktadır.

0	1	1	0	0	0	...	1	0	0	1	0	1	1	1
---	---	---	---	---	---	-----	---	---	---	---	---	---	---	---

Şekil 3.1. SUKP problemindeki bir nesnenin içeriği

Çantaya eklenen nesnenin faydası hesaplandığında her nesne için problemde belirtilen nesnenin faydası dikkate alınırken, ağırlığı hesaplandığında nesnenin içerdiği elemanların ağırlıkları dikkate alınmaktadır. Eğer çantaya eklenen farklı nesnelere aynı indis numaralı elemanı içeriyorsa, aynı indise sahip elemanlardan sadece bir elemanın ağırlığına göre toplam ağırlık hesaplanmaktadır. Böylece aynı indis numaralı elemanlardan oluşan nesnelere çantaya eklenmesi faydanın yüksek, ağırlık toplamının düşük olmasını sağlayacaktır.

3.2. Ölüm Oyunu Optimizasyon Algoritması (Battle Royale Optimization-BRO)

3.2.1. Ölüm oyununun stratejileri

Meta-sezgisel algoritmalar esinlendiği paradigmaya göre sınıflandırılmaktadır. Darwin'in evrim teorisinden ilham alan evrimsel optimizasyon algoritmalarında çözümlerde çeşitliliği sağlayan mutasyon ve rekombinasyon gibi ortak bir mekanizma vardır. Bu algoritmalar en yaygın olanları genetik algoritma, evrim stratejileri, tabu arama, orman arama gibi algoritmalarlardır.

Hayvanlar ve böcekler gibi canlıların çeşitli türlerinin akıllı kolektif davranışından esinlenerek sürü zekâsı algoritmaları ortaya çıkmıştır. Parçacık sürü optimizasyonu, karınca kolonisi optimizasyonu, kedi sürüsü optimizasyonu, yapay arı kolonisi (ABC), yusuçuk algoritması gibi algoritmalar bunlara örnek verilebilir. Öte yandan elektromanyetik kuvvet, yerçekimi kuvveti, eylemsizlik kuvveti gibi fiziksel yasalardan ilham alan meta-sezgisel algoritma türü ortaya çıkmıştır. Bu algoritmalar bazılarını Merkezi Kuvvet Optimizasyonu (CFO) (Formato, 2015), Yerçekimi Arama algoritması (GSA) (Rashedi ve ark., 2009), Optikten İlham alan optimizasyon (OIO) (Hojatollah Rajabi Moshtaghi, 2021), Su Buharlaştırma optimizasyonu (WEO) (Kaveh ve Bakhshpoori, 2016), Kara Delik algoritması (Hatamlou, 2013) gibi algoritmalarlardır.

Ölüm oyunu video oyunları rekabetçi, son ayakta kalanın kazandığı Japon filmi Ölüm Oyunundan (Battle Royale) ilham almıştır. Oyun içerisinde genellikle oyun alanı daralır ve oyuncuların hayatta kalmak için çevrelerini keşfetmeleri ve kullanmaları gerekir. Ölüm oyununun diğer bir özelliği dijital platformda çok sayıda insan, oyunu bireysel veya beş oyuncuya kadar takımlar halinde oynayabilmektedir. Bu oyunların

zorluğu sadece diğer oyuncuları yenmek değil, aynı zamanda oyunun kendisiyle de baş etmektir. Çünkü oyun boyunca oyun alanı (“güvenli alan”) azalmaktadır. Güvenli alanın dışında bir oyuncu oyundan elenme riskine sebep olacak ani bir zarar alabilmektedir. Bu zayıflamış oyuncuların artık daha küçük bir alanda yaralanmalarla rekabet etmesi gerekir.

Çoğu ölüm oyunu genellikle eşit düzeyde güç ve kaynaklara sahip oyuncuların oyun alanına rastgele yerleştirilmesiyle başlar. Keşif oyunun önemli bir bileşenidir, çünkü oyuncuların hayatta kalmaya yardımcı olacak araçları araması gerekir (hepsi rakipler tarafından öldürülmekten kaçınırken). Ölüm oyunlarının bir diğer bileşeni yeniden doğmadır. Bu özellik tüm oyunlarda olmamasına rağmen bazı oyunlar oyuncuya oyunda kalması için birkaç can vermektedir. Bu durum BRO algoritmasına da uyarlanmıştır. Bazı oyunlarda oyuncular hayatta kaldıkları süre boyunca ödüllendirilir, ancak nihayetinde sadece bir oyuncu (veya takım) kazanır.

“The Circle” olarak da adlandırılan oyun alanı oyuncuların oyun boyunca kalmak zorunda oldukları alandır. Çemberin dışında kalan oyuncular her saniye zarar alacaktır. Bu durum, oyuncuları birbirine yakın olmaya zorlar ve herkesi sınırlı bir şekilde güvenli alanda bir araya getirmeye çalışır. Bu süreçte, oyuncular birbirlerini öldürmeye ve sonuna kadar hayatta kalmaya çalışır. Öte yandan, bazı oyunlarda oyuncu rakipler tarafından öldürülürse savaş alanının rastgele bir noktasında yeniden doğar. Kazanan, en yüksek puana sahip olan oyuncu olacaktır.

Ölüm Oyunu optimizasyon algoritması, dijital oyundan esinlenilerek tasarlanmıştır. Esinlenme kaynağı sebebiyle evrimsel ve sürü zekasına sahip algoritmaların dışında tutulabilir. Popülasyon tabanlı olan BRO algoritmasında her bir aday çözüm asker, oyuncu olarak temsil edilmektedir ve diğerlerini yenerek hayatta kalmaya çalışmaktadır.

3.2.2. Ölüm oyunu optimizasyon algoritmasının adımları

BRO algoritmasının aşamaları ilham aldığı oyunla karşılaştırarak ilerlemektedir. Bazı ölüm oyunlarında oyun, oyuncuların bir yerden (uçak, paraşüt gibi) atlaması ile başlar. BRO algoritması diğer sürü tabanlı algoritmalara benzer olarak rasgele popülasyonun arama uzayına düzgün dağıtılmasıyla başlar. Popülasyondaki her bir birey (asker/oyuncu) en yakınındaki oyuncuyu zarar vermeye (öldürmeye) çalışır. Konumu

daha iyi olan birey (asker/oyuncu) maliyeti daha iyi olacağından komşuluğundaki bireye saldırır.

Bir oyuncu bir başkası tarafından yaralandığında, hasar değeri 1 artar. Bu durum matematiksel olarak aşağıdaki gibi modellenmektedir (Rahkar Farshi, 2021):

$$x_{i_damage} = x_{i_damage} + 1 \quad (3.3)$$

Burada x_{i_damage} popülasyondaki i . oyuncunun hasar değerini göstermektedir. Oyuncular hasar gördükten hemen sonra konumlarını değiştirmek ister. Bu sebeple başka taraftaki rakiplere saldırır. Dolayısıyla sömürü aşaması (exploitation) hasar almış oyuncunun var olan konumu ile şu ana kadar bulunan en iyi konumu arasındaki bir noktaya hareket etmesidir. Bu süreç aşağıdaki eşitlikle temsil edilmiştir:

$$x_{dam,d} = x_{dam,d} + r(x_{best,d} - x_{dam,d}) \quad (3.4)$$

Burada r , 0-1 arasında rasgele üretilmiş bir sayı, $x_{dam,d}$ ise zarar gören oyuncunun d boyutlu pozisyonudur. Eğer zarar gören oyuncu bir sonraki iterasyonda rakiplerine zarar verirse x_{i_damage} sıfır olarak güncellenir.

Keşfetme (exploration) aşamasında bir oyuncu hasar değeri önceden tanımlanmış seviyeyi (eşik değeri) aşarsa oyuncu ölür ve arama uzayında rasgele bir konumda yeniden doğar ve x_{i_damage} hasar değeri sıfırlanır. Deneme yanılma yoluyla eşik = 3 olarak belirlenmiştir (Rahkar Farshi, 2021). Bu davranış erken yakınsamayı önlemekte ve daha iyi keşif sağlamaktadır. Oyuncunun ölmesinden sonra arama uzayında yeniden doğması aşağıdaki eşitlikle gösterilmektedir:

$$x_{dam,d} = r(ub_d - lb_d) + lb_d \quad (3.5)$$

Burada ub_d , lb_d d boyutlu arama uzayının üst ve alt sınırlarını göstermektedir.

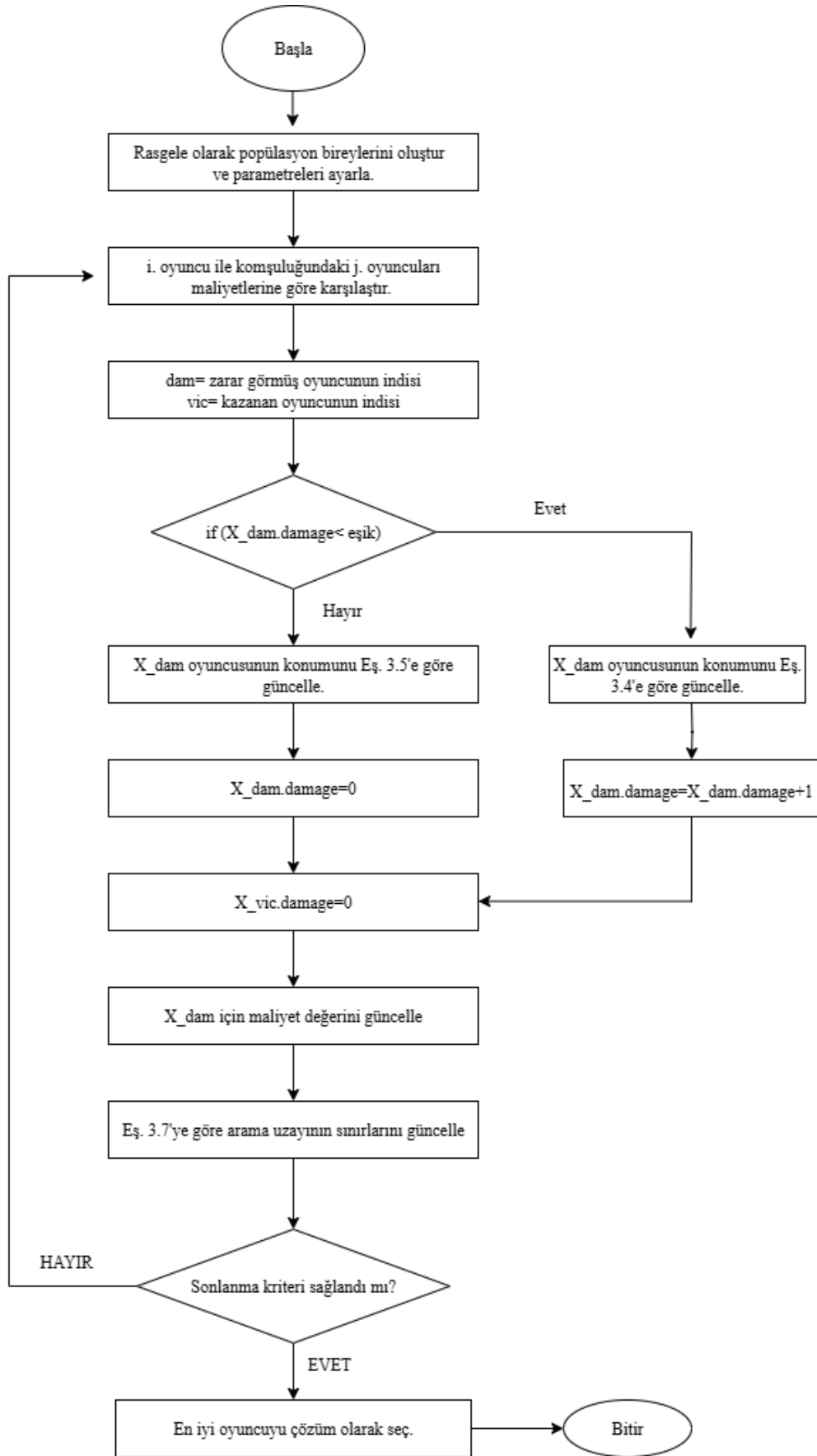
- Arama uzayı iterasyona bağlı olarak daralmaktadır. Bu aşamada Shrink olarak adlandırılan parametreye göre arama adım genişliği (Step) aşağıdaki denkleme göre değişmektedir:

$$\begin{aligned} \text{Shrink} &= \text{ceil}(\log_{10}(\text{max_iter})) \\ \text{Step} &= \text{round}(\text{max_iter} / \text{Shrink}) \end{aligned} \quad (3.6)$$

Arama uzayı daraldığı için alt ve üst sınırları aşağıdaki eşitliklerle değişmektedir:

$$\begin{aligned} lb_d &= x_{best,d} - SD(\bar{x}_d) \\ ub_d &= x_{best,d} + SD(\bar{x}_d) \end{aligned} \quad (3.7)$$

Burada $SD(\bar{x}_d)$, d boyutlu popülasyonun tamamının standart sapmasıdır. $x_{best,d}$ o zamana kadar bulunan en iyi çözümün pozisyonudur. Eğer alt ve üst sınır orijinal arama uzayının sınırlarını aşarsa, bu sınırlara kısıtlanır. Ayrıca, elitizme odaklanmak için her iterasyonda en iyi oyuncu/asker elit olarak kabul edilir ve saklanır. Şekil 3.2’de BRO algoritmasının akış diyagramı bulunmaktadır.



Şekil 3.2. Ölüm Oyunu Optimizasyon Algoritmasının akış diyagramı

4. ÖLÜM OYUNU OPTİMİZASYON ALGORİTMASININ İKİLİ UYGULAMALARI

Sürekli optimizasyon problemleri için geliştirilen BRO algoritmasının ikili yapıya sahip problemlere uyarlanmış hali literatürde mevcuttur. Ancak, ikili yapıya geçişte kullanılan dönüşüm mekanizmasının oldukça ilkel kaldığı ve iyileştirmeye açık olduğu görülmektedir. Bu nedenle, algoritmanın ikili problemlerdeki etkinliğini artırmak amacıyla, sürekli yapıdan ikili yapıya geçiş sürecinde farklı yaklaşımlar araştırılmıştır. Tez çalışmasının temel amacı, BRO algoritmasını kullanarak ikili yapıya sahip SUKP problemlerinin çözümüne yönelik daha etkili bir arama algoritması önermektir. Sürekli yapıdan ikili yapıya geçişte öncelikle çeşitli transfer fonksiyonları kullanılmıştır. Daha sonra ise BRO algoritmasının bazı adımlarına müdahalede bulunarak algoritma ikili arama uzayına taşınmıştır.

4.1. Transfer Fonksiyonları Yardımıyla İkili Ölüm Oyunu Optimizasyon Algoritması

BRO algoritması sürekli problemlere uygulanmak üzere modellenmiştir. Bu sebeple ikili yapıya sahip olan SUKP problemini çözmek için transfer fonksiyonları yardımıyla BRO'nun ikili versiyonu önerilmiştir. Farklı dönüşümlere sahip S-şekilli, V-şekilli (Mirjalili ve Lewis, 2013), U-şekilli (Mirjalili ve ark., 2020), Taper-şekilli (He ve ark., 2022), Z-şekilli (Guo ve ark., 2020), O1 (Pampará ve Engelbrecht, 2011), O2 (Costa ve ark., 2014), O3 (Wang ve ark., 2008), O4 (Zhu ve ark., 2017) ve son olarak Hiperbolik tanjant (Yonaba ve ark., 2010) transfer fonksiyonları kullanılmıştır.

Uygun bir transfer fonksiyonunun seçilmesi verimliliği artırmak için önemli bir karardır, çünkü transfer fonksiyonları sürekli arama uzayının ikili uzaya dönüştürülmesinde önemli bir rol oynamaktadır. PSO algoritması, aşağıdaki gibi tanımlanan sigmoid fonksiyonu yardımıyla ikili uzaya uyarlanmaktadır (J. Kennedy, 1997):

$$\text{sigm}\left(v_i^d(t)\right) = \frac{1}{1 + e^{-v_i^d(t)}} \quad (4.1)$$

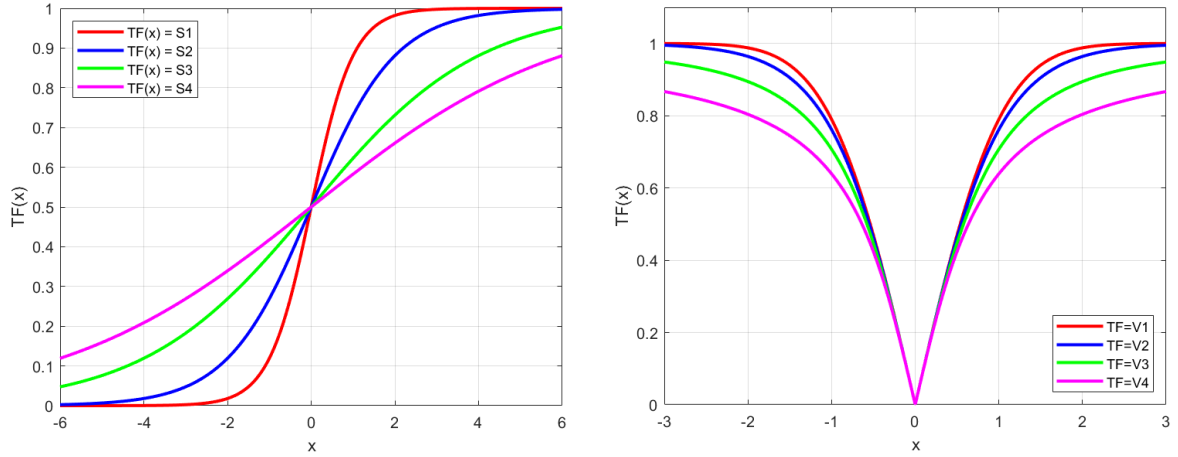
Burada $v_i^d(t)$ i . parçacığının d . boyutunun hızıdır. Bu pozisyon aşağıda tanımlanan $x_i^d(t+1)$ ile güncellenmektedir.

$$x_i^d(t+1) = \begin{cases} 1, & \text{eğer } r \geq \text{sigm}(v_i^d(t)) \\ 0, & \text{diğer} \end{cases} \quad (4.2)$$

Burada r , 0-1 arasında rasgele sayıdır. Seyedali Mirjalili ve arkadaşları S-şekilli ve V-şekilli yeni dönüşümler önermiştir (Mirjalili ve Lewis, 2013). Bu dönüşümler Tablo 4.1'de verilmiştir. Dönüşüm sonucu elde edilen 0-1 arasındaki değer Eş.4.2'ye göre ikili sisteme dönüşür. S-şekilli ve V-şekilli TF'lerinin grafiği Şekil 4.1'de verilmiştir.

Tablo 4.1. S-şekilli ve V-şekilli transfer fonksiyonları

Adı	Transfer Fonksiyonun Formülü	Dönüşüm
S-şekilli	S1: $TF(x) = \frac{1}{(1+e^{-2x})}$ S2: $TF(x) = \frac{1}{(1+e^{-x})}$ S3: $TF(x) = \frac{1}{(1+e^{-x/2})}$ S4: $TF(x) = \frac{1}{(1+e^{-x/3})}$	$x_i^d(t+1) = \begin{cases} 1, & \text{eğer } r \geq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.2)$
V-şekilli	V1: $TF(x) = \left \text{erf} \left(\frac{\sqrt{\pi}}{2} x \right) \right $ V2: $TF(x) = \tanh(x) $ V3: $TF(x) = \left \frac{x}{\sqrt{1+x^2}} \right $ V4: $TF(x) = \left \frac{2}{\pi} \arctan \left(\frac{\pi}{2} x \right) \right $	



Şekil 4.1. S-şekilli ve V-şekilli transfer fonksiyonlarının grafiği

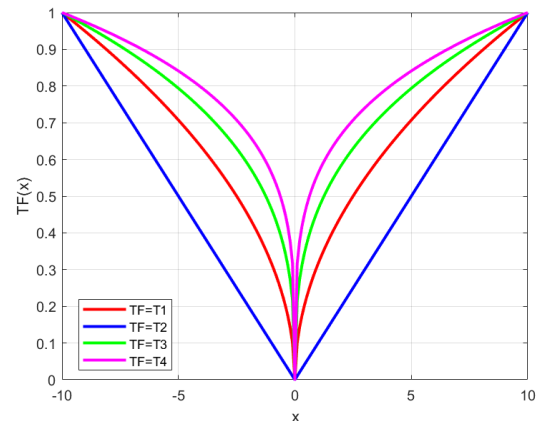
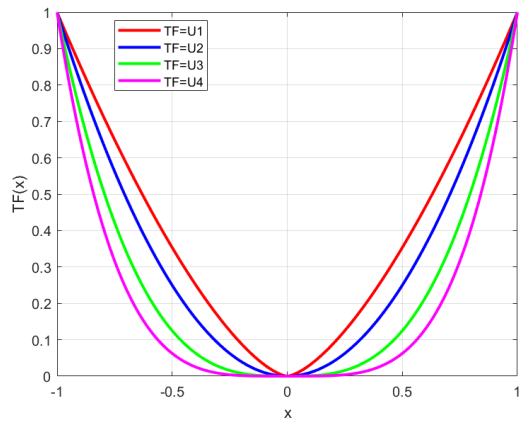
U-şekilli TF, $U(x) = \alpha|x^\beta|$ dönüşümü ile tanımlanmıştır (Mirjalili ve ark., 2020). Burada α, β kontrol parametreleridir. U1, U2, U3, U4 TF'ler Tablo 4.2'de verilmiştir. TF sonucu elde edilen reel değer Eş.4.2'ye göre ikili yapıya dönüştürülür.

İkili optimizasyon problemlerinde etkili çözümler elde etmek için Taper-şekilli (Konik-şekilli) TF'leri önerilmiştir (He ve ark., 2022). T1, T2, T3, T4 TF'lerin açık hali Tablo 4.2'de verilmiştir. Burada $A, [-A, A]$ arama uzayının sınır değeridir. TF dönüşümü ile elde edilen reel değer Eş.4.3'e göre ikili sisteme dönüştürülür. U-şekilli ve Taper-şekilli TF'lerin grafiği Şekil 4.2'de verilmiştir.

$$Bin_{val} = \begin{cases} 1, & \text{eğer } 0.5 \geq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.3)$$

Tablo 4.2. U-şekilli ve Taper-şekilli transfer fonksiyonları

Adı	Transfer Fonksiyonun Formülü	Dönüşüm
U-şekilli	U1: $TF(x) = x^{1.5} $, $\alpha = 1$, $\beta = 1.5$ U2: $TF(x) = x^2 $, $\alpha = 1$, $\beta = 2$ U3: $TF(x) = x^3 $, $\alpha = 1$, $\beta = 3$ U4: $TF(x) = x^4 $, $\alpha = 1$, $\beta = 4$	$x_i^d(t+1) = \begin{cases} 1, & \text{eğer } r \geq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.2)$
Taper-şekilli	T1: $TF(x) = \frac{\sqrt{ x }}{\sqrt{ A }}$ T2: $TF(x) = \frac{ x }{ A }$ T3: $TF(x) = \frac{\sqrt[3]{ x }}{\sqrt[3]{ A }}$ T4: $TF(x) = \frac{\sqrt[4]{ x }}{\sqrt[4]{ A }}$	$Bin_{val} = \begin{cases} 1, & \text{eğer } 0.5 \geq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.3)$

**Şekil 4.2.** U-şekilli ve Taper-şekilli transfer fonksiyonlarının grafiği

Z-şekilli olasılık transfer fonksiyonu Guo ve arkadaşları tarafından önerilmiştir (Guo ve ark., 2020). Her bir Z-şekilli fonksiyonun formülü Tablo 4.3'te verilmiştir. Z-şekilli TF kullanılarak 0 ile 1 arasında elde edilen bir gerçek sayı, Eş.4.2 kullanılarak ikili değere dönüştürülmektedir.

Yukarıda bahsedilen TF'lere ek olarak, literatürde diğer şekilli (other shape) transfer fonksiyonları da yer almaktadır. O1 TF, Pampará ve arkadaşları tarafından önerilmiştir (Pampará ve Engelbrecht, 2011). O2 TF, Costa ve diğerleri tarafından tanıtılmıştır (Costa ve diğerleri, 2014). O3 TF doğrusal normalizasyon fonksiyonudur

(Wang vd., 2008), O4 TF birim fonksiyon olarak alınmıştır (Zhu vd., 2017). O1, O4, O2, O3 TF'leri sırasıyla Eş. 4.4, Eş.4.5, Eş.4.2 ile ikili değere dönüştürülmüştür.

$$Bin_{val} = \begin{cases} 1, & \text{eğer } 0 \leq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.4)$$

$$Bin_{val} = TF(x) \quad (4.5)$$

Tablo 4.3. Z-şekilli ve O-şekilli transfer fonksiyonları

Adı	Transfer Fonksiyonun Formülü	Dönüşüm
Z-şekilli	Z1: $TF(x) = \sqrt{1 - 2^x}$ Z2: $TF(x) = \sqrt{1 - 5^x}$ Z3: $TF(x) = \sqrt{1 - 8^x}$ Z4: $TF(x) = \sqrt{1 - 20^x}$	$x_i^d(t+1) = \begin{cases} 1, & \text{eğer } r \geq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.2)$
O-şekilli	O1: $TF(x) = \sin(2\pi(x-a) * b * \cos(2\pi(x-a) * c)) + d$ ($a = d = 0, \quad b = c = 1$) O2: $TF(x) = \llbracket x \text{ mod } 2 \rrbracket$ O3: $TF(x) = \frac{(x-A_{min})}{A_{max}-A_{min}}, \quad (A_{min} \leq x \leq A_{max})$ O4: $TF(x) = x$	$Bin_{val} = \begin{cases} 1, & \text{eğer } 0 \leq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.4)$ $Bin_{val} = TF(x) \quad (4.5)$ $x_i^d(t+1) = \begin{cases} 1, & \text{eğer } r \geq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.2)$ $Bin_{val} = \begin{cases} 1, & \text{eğer } 0 \leq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.4)$

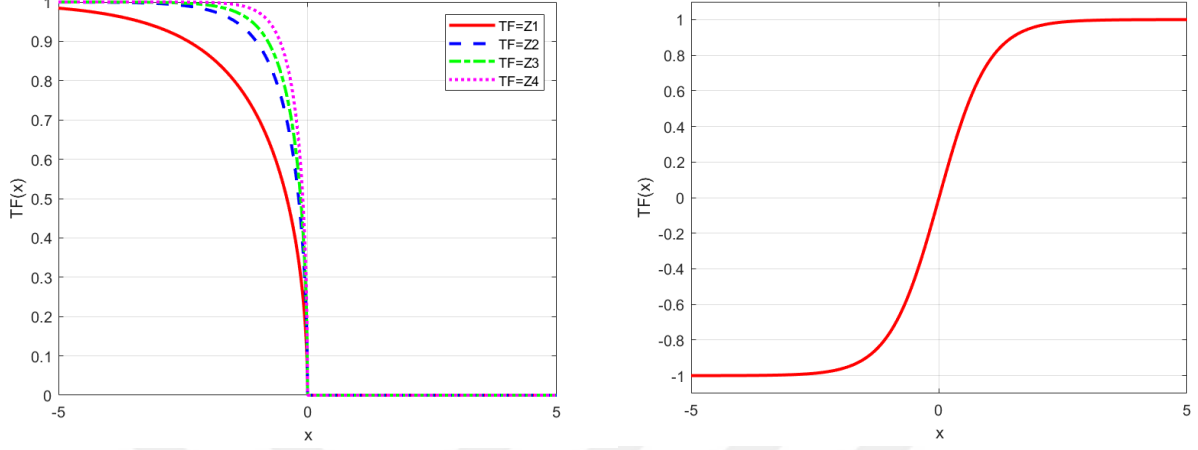
Kullandığımız son TF Hiperbolik tanjant sigmoid (TanSig) TF Tablo 4.4'te verilmiştir (Yonaba ve ark., 2010). Gerçek değer Eş. 4.6 ile ikili değere dönüştürülmüştür.

$$Bin_{val} = \begin{cases} 1, & \text{eğer } 0.6 < TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.6)$$

Tablo 4.4. Hiperbolik Tanjant Sigmoid transfer fonksiyonları

Adı	Transfer Fonksiyonun Formülü	Dönüşüm
Hiperbolik tanjant sigmoid	$TF(x) = \frac{2}{1 + e^{-2x}} - 1$	$Bin_{val} = \begin{cases} 1, & \text{eğer } 0.6 \leq TF(x) \\ 0, & \text{diğer} \end{cases} \quad (4.6)$

Z- şekilli ve Hiperbolik tanjant fonksiyonunun grafiği Şekil 4.3'te verilmiştir. O-şekilli (Other shape) TF'lerin her biri farklı yapıda olduğundan grafiğinin verilmesi uygun görülmemiştir.



Şekil 4.3. Z-şekilli ve Hiperbolik Tanjant transfer fonksiyonlarının grafiği

4.2. Onarım Algoritması (Repair Algorithm)

SUKP problemi kısıtlı optimizasyon problemi olduğu için bu probleme BRO algoritmasının ikili versiyonu uygulandığında kullanılması mümkün olmayan (infeasible) çözümler de elde edilmektedir. Bu çözümleri onarmak, arama uzayına dahil etmek için onarım algoritması kullanılmıştır. Açgözlü stratejiye (Arulselvan, 2014) göre;

- d_j ($j = 1, 2, \dots, n$), U_1, U_2, \dots, U_m alt kümelerindeki $j \in U$ elemanının frekansı olsun.
- $R_i = \sum_{j \in U_i} \frac{w_j}{d_j}$ ($i = 1, 2, \dots, m$) olmak üzere $\frac{p_i}{R_i}$, i . nesnenin yoğunluk değeri olarak gösterilsin.
- S kümesindeki bütün nesnelere için yoğunluk değerine göre azalan olarak sıralandığında nesnelerin indisleri $H[1, 2, \dots, m]$ dizisinde tutulsun.
- m boyutlu 0-1'lerden oluşan bir $Y = [y_1, y_2, \dots, y_m] \in \{0, 1\}^m$ aday çözümü için nesnelerin indis numaralarından oluşan $A_Y = \{i | y_i \in Y \text{ ve } y_i = 1, 1 \leq i \leq m\}$ gösterimi kullanılsın.
- Açgözlü stratejiye dayanarak onarım algoritmasında yoğunluk değeri en büyük olandan en küçük olan nesneye kadar boş sırt çantasına nesnelere kapasiteyi aşmayacak şekilde eklenmektedir. Böylece yoğunluğu daha düşük olan nesnelere çantadan çıkarılmıştır. Algoritmanın adımları aşağıdaki gibidir:

Algoritma 4.1. Onarım algoritmasının kaba kodu

Input: SUKP problemi aday çözümü $Y = [y_1, y_2, \dots, y_m] \in \{0,1\}^m$ ve $H[1,2, \dots, m]$ dizisi

Output: Mümkün çözüm $Y = [y_1, y_2, \dots, y_m] \in \{0,1\}^m$ ve mümkün çözümün $f(Y)$ uygunluk değeri

```

if ( $W(A_Y) > C$ )
   $Z \leftarrow [0,0, \dots, 0]$  //  $Z$   $m$  – boyutlu
  for  $i \leftarrow 1$  to  $m$  do
    If ( $Y_{H[i]} = 1$  and  $W(A_Z \cup \{H[i]\}) \leq C$ ) then  $Z_{H[i]} \leftarrow 1$  and  $A_Z \leftarrow A_Z \cup \{H[i]\}$ .
  end for
 $Y \leftarrow Z$ 
return ( $Y, f(Y)$ ).

```

4.3. İyileştirme Algoritması (Improvement Algorithm)

Seçilen nesnelere çantaya eklendiğinde kapasite aşılmıyorsa, daha fazla kazanç sağlayacak nesnelere eklenerek çözüm iyileştirilebilir. Eğer yeni eklenen nesnedeki bazı elemanlar zaten çantada varsa, bu elemanların ağırlığı tekrar sayılmaz; böylece çantanın kapasitesi artmadan toplam fayda artırılmış olur (Wu ve He, 2020). Onarım algoritması kapsamında, kapasite aşımı durumlarında çantadan belirli nesnelere çıkarılmasıyla çözümün verimliliği artırılmaya çalışılmıştır. Elde edilen aday çözüme, ek olarak iyileştirme algoritması da uygulanmış ve böylece çözüm kalitesi daha da geliştirilmiştir. Aşağıda, iyileştirme algoritmasına ait genel yapı sunulmuştur (Wu ve He, 2020):

Algoritma 4.2. Onarım ve iyileştirme algoritmasının kaba kodu

Input: SUKP problemi aday çözümü $Y = [y_1, y_2, \dots, y_m] \in \{0,1\}^m$ ve $H[1,2, \dots, m]$ dizisi

Output: Mümkün çözüm $Y = [y_1, y_2, \dots, y_m] \in \{0,1\}^m$ ve mümkün çözümün $f(Y)$ uygunluk değeri

```

1 if ( $W(A_Y) < C$ ) then go to 7
2  $Z \leftarrow [0,0, \dots, 0]$  //  $Z$   $m$  – boyutlu
3 for  $i \leftarrow 1$  to  $m$  do // onarım aşaması
4   If ( $Y_{H[i]} = 1$  and  $W(A_Z \cup \{H[i]\}) \leq C$ ) then  $Z_{H[i]} \leftarrow 1$  and  $A_Z \leftarrow A_Z \cup \{H[i]\}$ .
5 end for
6  $Y \leftarrow Z$ 
7 for  $i \leftarrow 1$  to  $m$  do // iyileştirme aşaması
8   If ( $Y_{H[i]} = 0$  and  $W(A_Z \cup \{H[i]\}) \leq C$ ) then  $Z_{H[i]} \leftarrow 1$  and  $A_Z \leftarrow A_Z \cup \{H[i]\}$ .

```

4.4. Ölüm Oyunu Optimizasyon Algoritmasının İkili Uzaya Aktarımı

TF'ler yardımıyla BRO algoritmasının ikili versiyonunda, algoritma sürekli arama uzayda çalışmaya devam etmektedir. Sadece algoritmanın maliyet fonksiyonu hesaplanmadan önce sürekli aday çözüm TF'ler yardımıyla ikili hale dönüştürülmektedir. Bu sebeple algoritmanın tamamen ikili arama uzayına adapte edilmesinin daha efektif ve verimli bir arama sağlayacağı düşünülmektedir. Bu bölümde algoritmanın ikili arama uzayına taşınmasında yapılan değişikliklerden bahsedilmiştir.

- Öncelikle başlangıçta aday çözümler ikili yapıya sahip şekilde oluşturuldu ve her bir aday çözüm için maliyet değerleri hesaplandı, en iyi maliyete sahip çözüm işaretlendi.
- Her bir aday çözüm en yakınındaki komşu çözüm ile maliyet değerine göre karşılaştırılarak maliyeti düşük olan çözümün *damage* değeri 1 artırıldı.
- Orijinal algortmada arama uzayı iterasyona bağlı olarak daraltılmaktadır. Bu aşamada Shrink olarak adlandırılan parametreye göre arama adım genişliği (Step) aşağıdaki denkleme göre değişmektedir:

$$\text{Shrink} = \text{ceil}(\log_{10}(\text{max_iter}))$$

$$\text{Step} = \text{round}(\text{max_iter} / \text{Shrink}) \quad (4.7)$$

- Önerilen ikili optimizasyon algoritmasında Shrink parametresi yerine ShrinkCount olarak adlandırılan ve yapılan analiz sonucu değeri 2 olarak belirlenen parametre kullanıldı. Arama uzayı bu parametreye bağlı olarak yeni adım genişliği ile aşağıdaki gibi taranmaktadır:

$$\text{ShrinkCount} = 2$$

$$\text{Step} = \text{Step} + \text{round}(\text{max_iter} / \text{ShrinkCount}) \quad (4.8)$$

- Ayrıca arama esnasında *etiket_oranı* olarak adlandırılan yeni bir parametre eklendi. Bu sayede oluşturulacak yeni bireylerin bütün bitlerinin yeniden belirlenmesi yerine, önceki çözümlerden aktarım gerçekleşmiş oldu. Böylece arama uzayı daraltıldı. Yapılan bu işlem örnek üzerinde anlatacak olursa;

Aday çözümlerin tamamında aynı numaralı bitte *etiket_oranı*'na bağlı olarak hep aynı sayı varsa yeni oluşturulacak olan aday çözümün aynı numaralı bitine de o sayı yazılmaktadır. Örneğin, problemdeki 20 adet aday çözümün 2 numaralı bitlerinin 18 tanesinde (*etiket_oranı*=0.9 alındı) veya daha fazlasında 1 değeri bulunuyorsa yeni oluşturulacak olan aday çözümün 2 numaralı bitine 1 değeri, eğer 2 numaralı bitlerin 18 tanesi veya daha fazlasında 0 değeri bulunuyorsa o halde yeni aday çözümün 2 numaralı bitine 0 değeri girilecektir.

aday1	0	1	1	0	...	0	1	1	0	0
aday2	1	1	1	0	...	0	0	1	0	0
⋮	⋮									
aday20	0	1	0	0	...	1	1	0	0	1

aday çözümler



	1									
--	---	--	--	--	--	--	--	--	--	--

yeni oluşturulan aday çözüm

Şekil 4.4. Etiket oranının etkisinin görselleştirilmesi

- Buradaki *etiket_oranı* iterasyona bağlı olarak değişecektir ve böylece arama uzayında daralmaya sebep olacaktır. Yeni oluşacak aday çözümün bazı bitleri önceden belli olacaktır.

$$etiket_oranı = etiket_oranı + 0.1 \quad (4.9)$$

etiket_oranı yapılan araştırma sonucu 0.2 olarak alınmıştır. Bununla ilgili analizler Bölüm 5'de verilmiştir.

- Algoritmanın orijinal formunda MyCrossover olarak isimlendirilen çaprazlama kısmı bulunmaktadır. Bu aşamada zarar alan çözüm aday1, aday çözümler içindeki en iyi çözüm aday2 olmak üzere aralarındaki çaprazlama aşağıdaki denkleme göre yapılmaktadır:

$$yeni_çözüm = aday1 + rand * (aday2 - aday1) \quad (4.10)$$

- Önerilen algoritmada ise zarar gören çözüm ile en iyi çözüm arasında tekdüze (uniform) çaprazlama yapılmaktadır. Tekdüze çaprazlamanın uygulama işlemi Şekil 4.5’de verilmiştir.

Seçilen bireyler için boyut kadar rasgele sayı üretilir. Eğer rasgele üretilen sayı 0.5’den büyük veya eşitse birey1 ve birey2’nin genleri yer değiştirir. Eğer küçük ise değişim olmaz.

Rasgele sayılar:	0.2	0.8	0.1	0.3	0.6
Birey1:	1	1	0	1	0
Birey2:	0	0	1	1	1
Çocuk1:	1	0	0	1	1
Çocuk2:	0	1	1	1	0

Şekil 4.5. Tekdüze (Uniform) çaprazlama ile yeni birey oluşturma

- Çaprazlama aşamasından sonra yeni çözümler üzerinde çeşitlilik sağlayabilmek adına mutasyon işlemi gerçekleştirilmiştir. Zarara uğrayan çözümün her bir biti için oluşturulan rasgele sayı mutasyon oranından küçük ise bit değeri tersine çevrilmiştir. Oluşan yeni mutant çözümün maliyet değeri var olan çözümden iyi ise mutant çözüm yeni çözüm olarak alınmıştır.
- Orijinal algoritmada aday çözümler arasında uzaklığa bakılırken Öklid metriği kullanılmıştır en yakın çözüm ile arasındaki maliyet değerine göre karşılaştırma yapılmıştır. Önerilen algoritmada 0, 1 bitlerden oluşan çözümler için Hamming uzaklığı kullanılarak bit bazında farklılık göz önüne alınmıştır.
- Bölüm 4.2 ve 4.3’de verilen onarım ve iyileştirme aşamaları çözümleri onarmak ve iyileştirmek için maliyet fonksiyonu içinde kullanılmıştır.

Aşağıda İkili BRO algoritması (BROBin) kaba kodu bulunmaktadır.

Algoritma 4.3. BROBin algoritmasının kaba kodu

Başla
 İkili formda rasgele popülasyon (x) oluştur
 Tüm parametreleri ayarla;
 eşik_değeri (oyuncunun can sayısı)=3
 ShrinkCount=2;
 Step=round(maxiter/ShrinkCount);
 etiket=[-1,-1,...,-1] N boyutlu dizi
 Populasyondaki tüm bireylerin maliyetini amaç fonksiyonuna (f) göre hesapla
 Iter = 1;
while durdurma kriteri sağlanmamışsa **do**
 Iter = Iter + 1
 for i = 1 to popülasyon_boyutuna
 i'inci bireyi en yakın bireyle karşılaştır (j'inci)
 dam (zarar gören) = j
 vic (kazanan) = i
 if f(x_i) < f(x_j)
 dam = i
 vic = j
 end if
 x_dam.damage= x_dam.damage+1
 if x_dam.damage < eşik_değeri then
 //Hasar almış oyuncu (x_dam) en iyi maliyete sahip oyuncu ile çaprazlamaya
 gönderilir
 else
 //Ölen oyuncu (x_dam) yerine rasgele olarak yeni bir oyuncu oluşturularak
 arama uzayına salınır
 end if
 x_dam oyuncu Shrinke gönderilir bazı bitleri etiket oranına göre değiştirilir.
 x_dam oyuncuya Mutasyon uygulanır
 Mutasyondan sonra en iyi çözüm ile maliyet değerine göre karşılaştırma yapılır x_dam
 maliyeti daha iyiye en iyi çözümle yer değiştirilir.
 end for

 if Iter ≥ step then
 std_oran=tüm popülasyondaki her bir bitteki toplam 1 sayısı
 if std_oran(1,i) ≥ N-(etiket_orani*N)
 etiket(i,1)=1;
 else if std_oran(1,i) ≤ (etiket_orani*N)
 etiket(i,1)=0;
 end if
 Step = Step + round(maxiter/ShrinkCount);
 etiket_orani=etiket_orani+0.1;
 En iyi çözümü seç ve döndür
end while
 Açıklamalar:
 x: Popülasyonu temsil eder.
 x_dam: Hasar almış oyuncu.
 f(.): Amaç fonksiyonu.

5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Tez çalışması kapsamında önerilen ikili BRO algoritmasının iki farklı varyasyonu Bölüm 3.1’de verilen 30 SUKP problemine uygulanmıştır. Elde edilen sonuçlar ile literatürdeki bazı ikili optimizasyon algoritmaları karşılaştırılarak sonuçlar değerlendirilmiştir. Önerilen algoritmalar MATLAB ortamında kodlanmıştır. Algoritmalar; Intel (R) Core (TM) i7-8565U CPU, 16 GB RAM, 500GB ssd depolama alanı özelliklerine sahip bilgisayarda çalıştırılmıştır. Algoritmaların parametreleri orijinal BRO algoritmasındaki değerler olarak alınmıştır.

5.1. Transfer Fonksiyonları Yardımıyla Elde Edilen Sonuçlar

30 SUKP problemi Bölüm 4.1’de verilen 25 TF yardımıyla 100 tekrarlı, popülasyon boyutu 20, iterasyon sayısı ise SUKP probleminin içeriğinde bulunan nesne sayısı m , eleman sayısı n olmak üzere $max(m, n)$ şeklinde alınarak çözülmüştür. Elde edilen 100 çözümün maksimum, minimum, ortalama değeri, standart sapması hesaplanarak her bir TF için tablolastırılmıştır.

Tablo 5.1’de S-şekilli ve V-şekilli TF ile elde edilen sonuçlar bulunmaktadır. Sonuçlara göre S-grubu TF’lerin V-grubu TF’den daha iyi sonuç verdiği söylenebilir. S-grubu kendi içinde yorumlanacak olursa S1 ve S2’nin, S3 ve S4’e göre daha etkili olduğu görülmektedir. P5 problemi için V1’in en iyi değeri bulduğu, V-grubu TF’lerin içinde V4’ün daha iyi sonuçlar elde ettiği söylenebilir.

Tablo 5.1. S-şekilli ve V-şekilli TF'ler için elde edilen sonuçlar

Problem	Sonuçlar	S1	S2	S3	S4	V1	V2	V3	V4
P1	Ort.	11066,5	11012,63	10970,99	10911,25	10995,89	10960,28	10977,39	10799,86
	Std. sapma	2,29E+02	2,12E+02	2,02E+02	2,02E+02	2,65E+02	2,67E+02	3,08E+02	2,64E+02
P2	Ort.	11666,76	11587,72	11505,8	11530,82	11440,93	11488,69	11484,39	11549,15
	Std. sapma	4,12E+02	4,04E+02	3,90E+02	3,33E+02	4,27E+02	4,27E+02	4,15E+02	3,57E+02
P3	Ort.	12848,83	12845,52	12853,61	12817,52	12663,76	12586,77	12456,68	12201,59
	Std.	1,82E+02	1,90E+02	1,64E+02	1,72E+02	2,47E+02	2,23E+02	2,22E+02	3,09E+02
P4	Ort.	11961,56	11991,14	12033,84	12033,23	11514,01	11503,2	11439,86	11607,52
	Std. sapma	2,11E+02	1,92E+02	1,45E+02	1,58E+02	3,57E+02	3,25E+02	3,50E+02	3,32E+02
P5	Ort.	13507,26	13573,33	13555,18	13512,41	13590,69	13550,21	13394,81	12641,53
	Std. sapma	1,92E+02	1,71E+02	1,91E+02	1,94E+02	2,17E+02	2,74E+02	2,92E+02	5,25E+02
P6	Ort.	12880,72	13018,02	13009,83	12937,08	12780,41	12713,9	12605,36	12522,06
	Std. sapma	4,56E+02	3,70E+02	3,92E+02	3,96E+02	4,11E+02	3,56E+02	3,83E+02	3,32E+02
P7	Ort.	11952,13	11891,46	11868	11732,69	11269,02	10992,97	10582,53	10792,71
	Std. sapma	3,31E+02	3,12E+02	2,70E+02	2,72E+02	4,30E+02	4,76E+02	4,82E+02	4,20E+02
P8	Ort.	9794,93	9778,23	9756,12	9719,68	9086,67	8910,71	8685,68	8922,4
	Std. sapma	3,05E+02	2,45E+02	2,40E+02	2,29E+02	4,64E+02	5,97E+02	5,21E+02	4,69E+02
P9	Ort.	12224,31	12035,62	12029,28	11856,56	10737,98	10594,37	10144,23	10484,5
	Std. sapma	4,65E+02	4,25E+02	4,36E+02	4,28E+02	4,10E+02	4,87E+02	5,55E+02	4,64E+02
P10	Ort.	12699,84	12708,66	12607,25	12519,46	11637,97	11392,47	11172,29	11412,75
	Std. sapma	4,59E+02	4,11E+02	4,77E+02	4,08E+02	7,17E+02	6,54E+02	6,51E+02	5,45E+02
P11	Ort.	10743,12	10705,42	10516,08	10431,41	9849,28	9750,82	9529,1	9745,38
	Std. sapma	4,06E+02	3,95E+02	3,37E+02	2,97E+02	4,00E+02	3,65E+02	4,30E+02	3,91E+02
P12	Ort.	10428,09	10482,99	10390,08	10372,75	9476,34	9488,76	9521,32	9730,59
	Std. sapma	4,38E+02	4,66E+02	3,66E+02	3,85E+02	5,33E+02	4,59E+02	4,74E+02	3,78E+02
P13	Ort.	10227,31	10150,62	10063,81	10001,73	8782	8945,36	9052,76	9193,78
	Std. sapma	2,51E+02	2,69E+02	2,54E+02	2,35E+02	3,76E+02	4,17E+02	3,75E+02	3,44E+02
P14	Ort.	9518,57	9515,8	9490,52	9376,3	8692,2	8635,78	8705,06	8824,32
	Std. sapma	2,68E+02	2,63E+02	2,50E+02	2,16E+02	4,53E+02	4,13E+02	3,88E+02	2,86E+02
P15	Ort.	9582,76	9541,37	9448,4	9372,75	8588,45	8662,48	8687,53	8844,9
	Std. sapma	2,57E+02	2,55E+02	2,23E+02	1,84E+02	4,35E+02	3,49E+02	3,00E+02	2,75E+02
P16	Ort.	10202,85	10128,72	10039,85	10013,55	9076,95	9044,69	9154,05	9217,51
	Std. sapma	4,83E+02	4,67E+02	4,90E+02	3,48E+02	4,97E+02	4,84E+02	3,73E+02	3,62E+02
P17	Ort.	11265,94	11229,52	11042,16	10909,05	9536,61	9469,28	9720,29	9843,55
	Std. sapma	3,64E+02	3,33E+02	2,91E+02	2,75E+02	4,12E+02	4,12E+02	4,12E+02	3,59E+02
P18	Ort.	9351,75	9281,48	9242,52	9181,07	8466,63	8493,5	8540,45	8708,48
	Std. sapma	3,10E+02	2,93E+02	2,59E+02	2,39E+02	4,62E+02	4,19E+02	4,09E+02	3,67E+02
P19	Ort.	9148,36	9068,14	8984,09	8898,46	8409,46	8456,5	8536,07	8678,97
	Std. sapma	2,11E+02	2,19E+02	1,90E+02	1,50E+02	3,27E+02	3,09E+02	2,56E+02	2,02E+02
P20	Ort.	8576,78	8601,58	8496,04	8433,64	7920,98	7950,32	7943,61	8043,29
	Std. sapma	3,02E+02	2,69E+02	2,86E+02	2,33E+02	3,47E+02	2,69E+02	2,72E+02	2,33E+02
P21	Ort.	9505,96	9389,23	9291,78	9187,69	8857,44	8881,98	8.954	9032,28
	Std. sapma	3,27E+02	2,45E+02	2,44E+02	1,72E+02	3,45E+02	3,29E+02	2,82E+02	2,49E+02
P22	Ort.	8690,93	8692,48	8570,67	8540,91	8082,41	8112,99	8.102	8198,11
	Std. sapma	3,06E+02	2,71E+02	2,20E+02	1,84E+02	3,31E+02	3,41E+02	2,64E+02	2,34E+02
P23	Ort.	9812,48	9730,5	9703,3	9592,8	9138,38	9150,48	9265,83	9318,63
	Std. sapma	2,58E+02	2,34E+02	2,10E+02	1,75E+02	3,16E+02	3,70E+02	3,07E+02	2,77E+02
P24	Ort.	8564,69	8556,87	8467,36	8401,58	8143,99	8091,37	8144,48	8204,03
	Std. sapma	3,08E+02	2,98E+02	2,16E+02	2,03E+02	3,21E+02	3,18E+02	2,92E+02	2,17E+02
P25	Ort.	9802,3	9798,21	9681,23	9616,61	9028	9008,2	9137,66	9212,14
	Std. sapma	2,27E+02	2,28E+02	2,01E+02	1,49E+02	3,40E+02	2,97E+02	2,77E+02	2,50E+02
P26	Ort.	8500,63	8497,45	8433,91	8378,43	7763,87	7782,59	7764,7	7903,78
	Std. sapma	2,15E+02	2,14E+02	1,55E+02	1,70E+02	2,99E+02	2,62E+02	2,91E+02	2,30E+02
P27	Ort.	10069,9	10023,42	9952,16	9919,53	9072,08	9137,44	9193,23	9298,79
	Std. sapma	2,21E+02	1,85E+02	1,87E+02	1,75E+02	3,20E+02	2,66E+02	2,30E+02	2,60E+02
P28	Ort.	8506,56	8446,36	8429,23	8368,14	7954,31	7914,22	7974,13	7990,54
	Std. sapma	2,03E+02	1,92E+02	1,83E+02	1,69E+02	2,40E+02	2,59E+02	2,47E+02	2,16E+02
P29	Ort.	9889,9	9891,23	9808,42	9725,22	8814,67	8782,36	8926,08	9074,18
	Std. sapma	1,79E+02	1,59E+02	1,53E+02	1,63E+02	3,41E+02	2,94E+02	2,81E+02	2,41E+02
P30	Ort.	8758,97	8772,32	8719,68	8639,33	7925,23	7975,25	8020,89	8125,9
	Std. sapma	2,41E+02	2,25E+02	2,20E+02	1,86E+02	2,62E+02	2,65E+02	2,34E+02	2,21E+02

Tablo 5.2’de U-şekilli, Taper-şekilli ve Hiperbolik tanjant TF’ler yardımı ile elde edilen sonuçlar gösterilmiştir. Tabloya göre U-grubunun içinde U4 TF ile elde edilen sonuçlar öne çıkarken, T-grubunda T1 TF ile elde edilen sonuçlar öne çıkmaktadır. T-grubu sonuçlarının U-grubu sonuçlarından daha iyi olduğu görülmektedir. Hiperbolik tanjant fonksiyonunun ise U-grubu sonuçlarından daha etkili, T1 TF sonuçlarından bir adım geride kaldığı söylenebilir.

Tablo 5.3’de O-şekilli ve Z-şekilli TF’ler yardımı ile elde edilen sonuçlar verilmiştir. 30 SUKP problemin 25’inde O2 TF ile elde edilen çözümlerin diğer TF ile elde edilen çözümlerden daha iyi olduğu tespit edilmiştir. Z-grubu TF’lerin 3 SUKP probleminde en iyi sonucu bulduğu görülmüştür.

Tablo 5.4’de her bir problem için en iyi sonucu bulan TF belirtilmiştir. 30 problemin 25’inde O2 TF’nin en iyi sonucu bulduğu görülmektedir. O2’den sonra ise Z2 TF’si sadece 2 problem için en iyi değeri elde etmiştir. U2, V1, Z3 TF’leri ise 1 problemin en iyi çözümünü bulmuştur. O2 fonksiyonu $[\text{mod } 2]$ şeklinde tanımlanmıştır. Sonuçlar bu dönüşümün ikili uzaya geçişte daha etkili olduğunu vurgulamaktadır.

Tablo 5.2. U-şekilli ve Taper-şekilli ve Hiperbolik tanjant TF'ler için elde edilen sonuçlar

Problem	Sonuçlar	U1	U2	U3	U4	T1	T2	T3	T4	Hip.Tan.
P1	Ort.	11062,35	11139,17	11064,57	11127,35	11042,11	10946,69	11051,28	11054,75	10998,88
	Std. sapma	3,00E+02	2,90E+02	2,97E+02	3,45E+02	2,68E+02	1,99E+02	3,29E+02	3,51E+02	2,63E+02
P2	Ort.	11546,68	11579,30	11568,89	11684,32	11644,16	11261,87	11687,07	11512,91	11663,94
	Std. sapma	3,91E+02	4,01E+02	4,54E+02	4,23E+02	4,32E+02	4,41E+02	4,06E+02	4,79E+02	4,50E+02
P3	Ort.	12687,91	12652,86	12642,14	12658,56	12766,94	12809,35	12552,64	12649	12756,4
	Std. sapma	2,17E+02	2,35E+02	2,35E+02	2,53E+02	2,26E+02	1,97E+02	2,96E+02	2,21E+02	2,39E+02
P4	Ort.	11541,85	11538,03	11557,03	11551,95	11945,74	11877,82	11664,14	11497,18	11950,07
	Std. sapma	3,63E+02	3,43E+02	3,52E+02	3,59E+02	2,19E+02	2,36E+02	4,23E+02	3,80E+02	2,15E+02
P5	Ort.	13561,82	13534,37	13492,37	13512,37	13436,25	13576,19	13233,71	13535,12	13455,43
	Std. sapma	2,20E+02	2,44E+02	2,45E+02	2,63E+02	2,96E+02	1,65E+02	4,37E+02	2,45E+02	2,33E+02
P6	Ort.	12790,19	12851,05	12870,4	12872,69	12912,25	12869	12828,3	12797,58	12750,71
	Std. sapma	3,82E+02	4,14E+02	3,67E+02	3,72E+02	3,99E+02	4,42E+02	3,97E+02	4,15E+02	4,55E+02
P7	Ort.	11381,76	11290,48	11275,69	11198,85	11937,03	11778,33	11687,02	11426,77	11854,9
	Std. sapma	4,20E+02	5,02E+02	4,49E+02	5,60E+02	3,53E+02	2,30E+02	5,70E+02	4,81E+02	3,84E+02
P8	Ort.	9302,64	9216,49	9046,06	9058,89	9779,61	9725,47	9247,65	9306	9693,25
	Std. sapma	5,31E+02	5,02E+02	5,99E+02	5,67E+02	3,26E+02	1,86E+02	5,70E+02	4,89E+02	3,13E+02
P9	Ort.	10975,41	10872,12	10713,73	10698,73	12130,82	11846,38	11278,14	11042,48	11968,44
	Std. sapma	3,52E+02	3,45E+02	4,79E+02	5,62E+02	5,20E+02	4,83E+02	6,43E+02	3,58E+02	5,23E+02
P10	Ort.	11810,63	11679,62	11635,58	11506,68	12554,77	12292,81	12177,37	11776,67	12525,16
	Std. sapma	6,19E+02	6,82E+02	7,64E+02	6,88E+02	5,14E+02	4,35E+02	7,32E+02	7,58E+02	5,65E+02
P11	Ort.	10020,95	10001,81	9959,21	9898,61	10681,82	10435,39	10221,09	10067,29	10565,75
	Std. sapma	2,89E+02	3,80E+02	4,48E+02	4,88E+02	4,04E+02	2,74E+02	5,00E+02	3,27E+02	4,09E+02
P12	Ort.	9716,88	9669,86	9966,65	9928,65	10389,67	10213,73	10121,89	9674,92	10391,17
	Std. sapma	4,44E+02	5,13E+02	5,53E+02	5,47E+02	4,62E+02	3,50E+02	4,81E+02	5,56E+02	4,50E+02
P13	Ort.	9174,8	9172,77	9240,31	9481,45	10171,62	10188,55	9869,94	9194,3	10102,47
	Std. sapma	4,69E+02	3,84E+02	4,82E+02	4,35E+02	2,87E+02	2,83E+02	3,91E+02	5,35E+02	3,49E+02
P14	Ort.	8716,55	8892,73	9056,47	9088,3	9465,37	9462,09	9271,69	8838,05	9461,39
	Std. sapma	4,77E+02	5,06E+02	4,99E+02	4,74E+02	3,50E+02	2,37E+02	4,55E+02	4,92E+02	3,42E+02
P15	Ort.	8752,38	8848,38	9112,4	9074,08	9497,03	9484,71	9321,23	8869,11	9494,08
	Std. sapma	4,75E+02	4,05E+02	3,96E+02	4,18E+02	2,93E+02	2,39E+02	3,68E+02	4,88E+02	2,94E+02
P16	Ort.	9256,38	9459,75	9443,6	9715,63	10032,45	9635,93	9732,55	9383,07	10073,49
	Std. sapma	4,76E+02	5,18E+02	4,76E+02	5,07E+02	3,92E+02	2,77E+02	5,45E+02	5,65E+02	5,18E+02
P17	Ort.	9734,98	9840,03	9955,45	10092,39	11250,35	11098,46	10570,96	9881,82	11043,45
	Std. sapma	4,69E+02	5,59E+02	4,49E+02	4,51E+02	3,84E+02	2,09E+02	5,06E+02	5,56E+02	3,56E+02
P18	Ort.	8702,4	8737,13	9004,63	8916,33	9286,82	9235,36	9247,15	8693,15	9177,9
	Std. sapma	4,79E+02	4,71E+02	4,59E+02	4,97E+02	3,55E+02	1,79E+02	4,35E+02	5,58E+02	3,16E+02
P19	Ort.	8489,25	8639,34	8749,38	8809,22	9091,96	8828,91	9093,97	8636,83	9111,78
	Std. sapma	3,73E+02	3,12E+02	2,98E+02	3,47E+02	2,47E+02	1,79E+02	3,14E+02	3,69E+02	2,66E+02
P20	Ort.	8009,06	8101,01	8165,42	8207,04	8659,81	8283,1	8536,55	8102,74	8594,74
	Std. sapma	3,79E+02	3,70E+02	3,44E+02	4,39E+02	2,77E+02	2,70E+02	3,24E+02	3,40E+02	3,31E+02
P21	Ort.	8933,75	9049,4	9217,19	9277,99	9491,25	9100,6	9544,94	9085,1	9511,25
	Std. sapma	3,68E+02	3,52E+02	3,63E+02	3,80E+02	3,14E+02	2,27E+02	3,80E+02	3,87E+02	3,65E+02
P22	Ort.	8195,78	8274,29	8375,49	8417,63	8595,41	8504,13	8492,71	8213,22	8660,32
	Std. sapma	3,97E+02	3,78E+02	4,10E+02	4,16E+02	2,73E+02	2,14E+02	3,65E+02	4,54E+02	3,15E+02
P23	Ort.	9226,16	9393,25	9518,85	9568,69	9832,33	9633,76	9807,46	9319,2	9747,27
	Std. sapma	3,76E+02	3,00E+02	3,82E+02	3,14E+02	2,79E+02	1,72E+02	3,07E+02	3,65E+02	2,71E+02
P24	Ort.	8233,98	8335,44	8394,79	8516,26	8562,53	8324,16	8667,57	8240,8	8546,35
	Std. sapma	3,77E+02	3,78E+02	3,96E+02	3,27E+02	3,15E+02	2,47E+02	4,19E+02	4,00E+02	2,99E+02
P25	Ort.	9169,67	9260,73	9398,04	9481,72	9842,27	9582,92	9729	9338,61	9833,66
	Std. sapma	3,16E+02	3,42E+02	3,53E+02	3,70E+02	2,77E+02	1,94E+02	3,11E+02	3,65E+02	2,82E+02
P26	Ort.	7898,62	7917,51	8001,53	8073,08	8463,3	8270,29	8270,41	7894,45	8449,74
	Std. sapma	2,95E+02	2,94E+02	3,37E+02	3,29E+02	2,46E+02	1,68E+02	3,29E+02	3,20E+02	2,54E+02
P27	Ort.	9248,54	9328,76	9473,6	9469,21	10013,62	9998,67	9793,81	9333,8	9962,65
	Std. sapma	3,23E+02	3,39E+02	3,05E+02	3,15E+02	2,25E+02	1,03E+02	3,05E+02	3,69E+02	2,76E+02
P28	Ort.	7977,45	8119,13	8234,22	8276,35	8491	8299,56	8419,52	8059,38	8445,63
	Std. sapma	2,48E+02	3,11E+02	2,82E+02	2,97E+02	2,10E+02	1,92E+02	2,93E+02	3,16E+02	2,23E+02
P29	Ort.	8902,84	9096,26	9231,91	9284,68	9906,83	9798,48	9607,16	9095,62	9835,7
	Std. sapma	3,36E+02	3,47E+02	3,84E+02	3,39E+02	1,61E+02	1,14E+02	2,93E+02	3,39E+02	2,46E+02
P30	Ort.	8047,89	8132,19	8209	8286,64	8726,26	8597,75	8454,98	8208,07	8705,29
	Std. sapma	3,24E+02	3,00E+02	3,03E+02	2,88E+02	2,25E+02	2,09E+02	3,29E+02	3,38E+02	2,12E+02

Tablo 5.3. O-şekilli ve Z-şekilli TF'ler için elde edilen sonuçlar

Problem	Sonuçlar	O1	O2	O3	O4	Z1	Z2	Z3	Z4
P1	Ort.	11025,5	11049,57	10905,75	10982,73	11024	11003,13	11005,06	11041,92
	Std. sapma	2,19E+02	2,00E+02	1,72E+02	2,23E+02	2,03E+02	2,02E+02	2,12E+02	2,45E+02
P2	Ort.	11507,54	11699,6	11570,22	11462,96	11501,19	11565,09	11557,63	11531,42
	Std. sapma	3,26E+02	3,88E+02	3,38E+02	3,99E+02	4,43E+02	4,51E+02	4,00E+02	4,32E+02
P3	Ort.	12798,6	12835,82	12816,78	12788,79	12854,78	12860,06	12852,49	12795,11
	Std. sapma	1,72E+02	1,62E+02	1,73E+02	2,11E+02	1,74E+02	2,04E+02	1,68E+02	1,88E+02
P4	Ort.	12019,49	12054,37	11987,05	11971,59	11971,81	11975,31	11988,19	11961,3
	Std. sapma	1,59E+02	1,40E+02	2,00E+02	2,06E+02	1,72E+02	1,98E+02	2,18E+02	2,07E+02
P5	Ort.	13536,11	13555,9	13507,53	13501,2	13587,38	13527,55	13570,81	13522,06
	Std. sapma	1,47E+02	1,74E+02	1,92E+02	2,25E+02	1,67E+02	1,87E+02	1,77E+02	2,07E+02
P6	Ort.	13030,19	13039,28	12915,74	12846,81	13025,39	12908,69	12899,41	12868,02
	Std. sapma	3,60E+02	3,86E+02	4,24E+02	4,54E+02	3,90E+02	3,95E+02	4,66E+02	4,41E+02
P7	Ort.	11817,19	12063,53	11637,47	11883,78	11894,17	12002,83	12003,03	11845,47
	Std. sapma	2,89E+02	3,46E+02	2,54E+02	3,55E+02	2,79E+02	3,29E+02	3,63E+02	3,36E+02
P8	Ort.	9782,87	9875,31	9655,11	9825,74	9791,64	9788,39	9757,84	9798,73
	Std. sapma	2,38E+02	2,26E+02	2,40E+02	2,86E+02	2,54E+02	2,51E+02	2,94E+02	2,73E+02
P9	Ort.	12074,86	12343,78	11825,62	12150,67	12258,82	12232,11	12234,01	12209,2
	Std. sapma	4,35E+02	4,67E+02	4,11E+02	4,89E+02	4,69E+02	4,67E+02	4,90E+02	4,36E+02
P10	Ort.	12732,3	12809,18	12525,6	12630,02	12659,73	12688,9	12630,21	12634,58
	Std. sapma	3,37E+02	3,55E+02	3,99E+02	5,27E+02	4,07E+02	4,06E+02	4,23E+02	4,50E+02
P11	Ort.	10650,28	10876,58	10445,04	10593,11	10680,97	10666,41	10658,72	10701,52
	Std. sapma	3,40E+02	3,66E+02	2,81E+02	3,95E+02	3,37E+02	4,17E+02	3,91E+02	3,97E+02
P12	Ort.	10448,12	10668,83	10270,9	10312,12	10352,63	10538,25	10409,47	10408,41
	Std. sapma	3,86E+02	3,81E+02	3,17E+02	4,59E+02	4,07E+02	4,72E+02	4,14E+02	4,11E+02
P13	Ort.	10104,81	10234,07	9912,01	10170,51	10250,22	10235,88	10190,64	10214,63
	Std. sapma	2,73E+02	2,86E+02	1,92E+02	3,07E+02	2,46E+02	2,76E+02	2,93E+02	2,77E+02
P14	Ort.	9515,19	9592,46	9379,69	9457,71	9524,89	9498,05	9521,82	9515,8
	Std. sapma	2,76E+02	2,52E+02	2,37E+02	2,73E+02	2,56E+02	2,71E+02	2,97E+02	3,06E+02
P15	Ort.	9468,64	9661,32	9352,53	9505,37	9620,95	9615,54	9582,71	9632,2
	Std. sapma	1,95E+02	2,45E+02	2,24E+02	2,59E+02	2,60E+02	2,83E+02	2,51E+02	2,79E+02
P16	Ort.	10113,64	10288,73	9923,51	10158,06	10092,01	10074,94	10069,08	10056
	Std. sapma	4,59E+02	4,47E+02	3,20E+02	5,40E+02	5,08E+02	4,36E+02	4,47E+02	5,31E+02
P17	Ort.	11096,25	11337,33	10853,69	11228,7	11282,49	11288,97	11269,85	11231,37
	Std. sapma	2,77E+02	3,19E+02	2,53E+02	3,52E+02	3,38E+02	3,60E+02	3,81E+02	3,59E+02
P18	Ort.	9248,02	9476,05	9142,96	9238,17	9343,66	9312,7	9302,39	9285,03
	Std. sapma	2,13E+02	3,02E+02	1,86E+02	2,77E+02	2,71E+02	2,50E+02	2,89E+02	3,26E+02
P19	Ort.	9034,59	9195,31	8915,86	9075,61	9040,55	9129,25	9112,64	9064,89
	Std. sapma	2,01E+02	2,54E+02	1,56E+02	2,30E+02	2,27E+02	2,48E+02	2,41E+02	2,55E+02
P20	Ort.	8515,64	8753,28	8374,02	8553,05	8554,7	8538,92	8577,38	8519,84
	Std. sapma	2,31E+02	2,72E+02	2,07E+02	2,72E+02	2,82E+02	3,09E+02	3,03E+02	3,31E+02
P21	Ort.	9288,6	9559,99	9189,14	9456,97	9327,59	9340,78	9396,38	9373,61
	Std. sapma	2,28E+02	2,95E+02	1,83E+02	3,41E+02	2,88E+02	2,71E+02	3,10E+02	3,06E+02
P22	Ort.	8583,68	8821,95	8546,45	8623,34	8662	8697,82	8664,31	8613,44
	Std. sapma	1,83E+02	2,33E+02	1,91E+02	2,48E+02	2,62E+02	2,48E+02	2,61E+02	2,53E+02
P23	Ort.	9647,96	9878,17	9565,25	9738,78	9782,1	9763,82	9795	9791,44
	Std. sapma	1,54E+02	2,36E+02	1,71E+02	2,45E+02	2,21E+02	2,65E+02	2,43E+02	2,16E+02
P24	Ort.	8490,92	8672,97	8363,73	8483,6	8474,04	8533,84	8567,31	8522,11
	Std. sapma	2,15E+02	2,84E+02	2,06E+02	3,03E+02	2,51E+02	2,89E+02	3,07E+02	2,83E+02
P25	Ort.	9669,71	9906,74	9587,72	9777,35	9793,86	9814,64	9823,61	9758,73
	Std. sapma	1,90E+02	2,44E+02	1,38E+02	2,65E+02	1,98E+02	2,30E+02	2,13E+02	2,02E+02
P26	Ort.	8427,5	8596,64	8319,04	8442,82	8430,07	8485,92	8543,31	8521,58
	Std. sapma	1,74E+02	1,92E+02	1,52E+02	2,53E+02	2,01E+02	2,33E+02	2,40E+02	2,64E+02
P27	Ort.	9962,94	10160,39	9847,23	10031,2	10120,05	10109,21	10103,82	10067,45
	Std. sapma	1,62E+02	2,06E+02	1,56E+02	2,21E+02	1,88E+02	1,87E+02	2,31E+02	1,71E+02
P28	Ort.	8399,48	8583,28	8339,95	8440,15	8523,02	8512,65	8490,61	8456,36
	Std. sapma	1,45E+02	1,86E+02	1,29E+02	1,94E+02	2,07E+02	2,19E+02	2,22E+02	2,25E+02
P29	Ort.	9790,86	9973,04	9695,12	9907,44	9884,39	9880,83	9939,33	9950,69
	Std. sapma	1,62E+02	1,75E+02	1,15E+02	1,67E+02	1,56E+02	1,77E+02	1,72E+02	1,78E+02
P30	Ort.	8674,04	8817,99	8569,21	8728,19	8803,5	8776,22	8777,1	8767,82
	Std. sapma	1,82E+02	1,98E+02	1,64E+02	2,58E+02	2,18E+02	2,32E+02	2,44E+02	2,39E+02

Tablo 5.4. Her problem için en iyi sonuç elde TF'ler

Problem	En İyi TF	Problem	En İyi TF	Problem	En İyi TF
P1	U2	P11	O2	P21	O2
P2	O2	P12	O2	P22	O2
P3	Z2	P13	Z2	P23	O2
P4	O2	P14	O2	P24	O2
P5	V1	P15	O2	P25	O2
P6	O2	P16	O2	P26	O2
P7	Z3	P27	O2	P27	O2
P8	O2	P18	O2	P28	O2
P9	O2	P19	O2	P29	O2
P10	O2	P20	O2	P30	O2

SUKP problemi maksimizasyon problemi olduğu için Friedman Sıralama testi kullanılarak TF arasında maksimum değere göre sıralama analizi yapılmıştır ve sonuçlar Tablo 5.5'de verilmiştir. Friedman sıralamasında O2 TF ile elde edilen sonuçların diğer TF arasından daha yüksek olduğu, sıralamada birinci olduğu görülmektedir. V grubu TF'ler en geride kalırken Z- grubu TF'lar daha başarılı sonuçlar elde etmektedir. S grubunda S1 ve S2 TF'ler daha başarılı sonuçlara ulaşırken, S3 ve S4 TF'nin tanımında bulunan paydadaki üstel fonksiyonun kuvveti arttıkça sonuçların kötüleştiği görülmektedir.

Tablo 5.5. TF için elde edilen Friedman sıralaması

TF	Friedman	Sıralama
S1	20,8667	2
S2	19,4833	6
S3	14,9333	12
S4	12,4333	14
V1	3,6333	23
V2	2,8333	24
V3	2,6000	25
V4	4,6333	22
U1	6,5667	21
U2	7,8000	19
U3	8,5333	18
U4	9,5667	17
T1	18,0333	7
T2	11,9000	15
T3	12,6000	13
T4	7,3333	20
Hip.Tan.	15,4333	11
O1	16,0000	9
O2	24,3000	1
O3	11,7333	16
O4	15,9333	10
Z1	19,6333	5
Z2	19,9667	4
Z3	20,2667	3
Z4	17,9833	8

Tablo 5.6'da koşullu biçimlendirme tablosu bulunmaktadır. Hangi TF ile maksimum değer elde edildiye o değer yeşil, minimum değer ise kırmızı ile görselleştirilmiştir. Tablodan açıkça görülmektedir ki O2 TF nerdeyse problemlerin tamamında yeşil olarak renklenmiş, yani en yüksek değeri bulmuştur. O2 fonksiyonunun tanımını Tablo 4.3'den gördüğü üzere mod 2'nin tam değer fonksiyonu şeklindedir. Elde edilen sonuçlar bu fonksiyonun ikili yapıya geçişte daha etkili olduğunu göstermektedir.

Z-grubu TF'lerinin grup olarak başarılı olduğu, S-grubunda ise S1 ve S2 TF'lerinin S3 ve S4 TF'lerinden daha iyi olduğu görülmektedir. V-grubu TF'lerin diğer TF'ler içinde geride kaldığı, T-grubu TF'lerden T1'in gruptaki diğer TF'lere göre daha başarılı sonuçlar verdiği, U-grubu TF'lerinin V-grubundan bir adım önde olduğu diğer TF'lerden daha başarısız olduğu, Hiperbolik tanjant TF'sinin ise ortalama bir başarı gösterdiği söylenebilir.

Tablo 5.6. TF'lere göre elde edilen sonuçların koşullu biçimlendirme ile ısı haritası

PROBLEM	TF	S1	S2	S3	S4	V1	V2	V3	V4	U1	U2	U3	U4	T1	T2	T3	T4	Hip.Tan	O1	O2	O3	O4	Z1	Z2	Z3	Z4
P1		11066,5	11012,6	10971	10911,3	10995,9	10960,3	10977,4	10799,9	11062,4	11139,17	11064,6	11127,4	11042,1	10946,7	11051,3	11054,8	10998,9	11025,5	11049,6	10905,8	10982,7	11024	11003,1	11005,1	11041,9
P2		11666,8	11587,7	11505,8	11530,8	11440,9	11488,7	11484,4	11549,2	11546,7	11579,30	11568,9	11684,3	11644,2	11261,9	11687,1	11512,9	11663,9	11507,5	11699,6	11570,2	11463	11501,2	11565,1	11557,6	11531,4
P3		12848,8	12845,5	12853,6	12817,5	12663,8	12586,8	12456,7	12201,6	12687,9	12652,86	12642,1	12658,6	12766,9	12809,4	12552,6	12649	12756,4	12798,6	12835,8	12816,8	12788,8	12854,8	12860,1	12852,5	12795,1
P4		11961,6	11991,1	12033,8	12033,2	11514	11503,2	11439,9	11607,5	11541,9	11538,03	11557	11552	11945,7	11877,8	11664,1	11497,2	11950,1	12019,5	12054,4	11987,1	11971,6	11971,8	11975,3	11988,2	11961,3
P5		13507,3	13573,3	13555,2	13512,4	13590,7	13550,2	13394,8	12641,5	13561,8	13534,37	13492,4	13512,4	13436,3	13576,2	13233,7	13335,1	13455,4	13536,1	13555,9	13507,5	13501,2	13587,4	13527,6	13570,8	13522,1
P6		12880,7	13018	13009,8	12937,1	12780,4	12713,9	12605,4	12522,1	12790,2	12851,05	12870,4	12872,7	12912,3	12869	12828,3	12797,6	12750,7	13030,2	13039,3	12915,7	12846,8	13025,4	12908,7	12899,4	12868
P7		11952,1	11891,5	11868	11732,7	11269	10993	10582,5	10792,7	11381,8	11290,48	11275,7	11198,9	11937	11778,3	11687	11426,8	11854,9	11817,2	12063,5	11637,5	11883,8	11894,2	12002,8	12003	11845,5
P8		9794,93	9778,23	9756,12	9719,68	9086,67	8910,71	8685,68	8922,4	9302,64	9216,49	9046,06	9058,89	9779,61	9725,47	9247,65	9306	9693,25	9782,87	9873,31	9655,11	9825,74	9791,64	9788,39	9757,84	9798,73
P9		12224,3	12035,6	12029,3	11856,6	10738	10594,4	10144,2	10484,5	10975,4	10872,12	10713,7	10698,7	12130,8	11846,4	11278,1	11042,5	11968,4	12074,9	12343,8	11825,6	12150,7	12258,8	12232,1	12234	12209,2
P10		12699,8	12708,7	12607,3	12519,5	11638	11392,5	11172,3	11412,8	11810,6	11679,62	11635,6	11506,7	12554,8	12292,8	12177,4	11776,7	12525,2	12732,3	12809,2	12525,6	12630	12659,7	12688,9	12630,2	12634,6
P11		10743,1	10705,4	10516,1	10431,4	9849,28	9750,82	9539,1	9745,38	10021	10001,81	9959,21	9988,61	10683,8	10435,4	10221,1	10067,3	10565,8	10650,3	10876,6	10445	10593,1	10681	10666,4	10658,7	10701,5
P12		10428,1	10483	10390,1	10372,8	9476,34	9488,76	9521,32	9730,59	9716,88	9669,86	9666,65	9928,65	10389,7	10213,7	10121,9	9674,92	10391,2	10448,1	10668,8	10270,9	10312,1	10352,6	10538,3	10409,5	10408,4
P13		10227,3	10150,6	10063,8	10001,7	8782	8945,36	9052,76	9193,78	9174,8	9172,77	9240,31	9481,45	10171,6	10188,6	9869,94	9194,3	10102,5	10104,8	10234,1	9912,01	10170,5	10250,2	10235,9	10190,6	10214,6
P14		9518,57	9515,8	9490,52	9376,3	8692,2	8635,78	8705,06	8824,32	8716,55	8892,73	9056,47	9088,3	9465,37	9462,09	9271,69	8838,05	9461,39	9515,19	9592,46	9379,69	9457,71	9524,89	9498,05	9521,82	9515,8
P15		9582,76	9541,37	9448,4	9372,75	8588,45	8662,48	8687,53	8844,9	8752,38	8848,38	9112,4	9074,08	9497,03	9484,71	9321,23	8869,11	9494,08	9468,64	9661,32	9352,53	9505,37	9620,95	9615,54	9582,71	9632,2
P16		10202,9	10128,7	10039,9	10013,6	9076,95	9044,69	9154,05	9217,51	9256,38	9459,75	9443,6	9715,63	10032,5	9635,93	9732,55	9383,07	10073,5	10113,6	10288,7	9923,51	10158,1	10092	10074,9	10069,1	10056
P17		11265,9	11229,5	11042,2	10909,1	9536,61	9469,28	9720,29	9843,55	9734,98	9840,03	9955,45	10092,4	11250,4	11098,5	10571	9881,82	11043,5	11096,3	11337,3	10853,7	11228,7	11282,5	11289	11269,9	11231,4
P18		9351,75	9281,48	9242,52	9181,07	8466,63	8493,5	8540,45	8708,48	8702,4	8737,13	9004,63	8916,33	9286,82	9235,36	9247,15	8693,15	9177,9	9248,02	9476,05	9142,96	9238,17	9343,66	9312,7	9302,39	9285,03
P19		9148,36	9068,14	8984,09	8898,46	8409,46	8456,5	8536,07	8678,97	8489,25	8639,34	8749,38	8809,22	9091,96	8828,91	9093,97	8636,83	9111,78	9034,59	9195,31	8915,86	9075,61	9040,55	9129,25	9112,64	9064,89
P20		8576,78	8601,58	8496,04	8433,64	7920,98	7950,32	7943,61	8043,29	8009,06	8101,01	8165,42	8207,04	8659,81	8283,1	8536,55	8102,74	8594,74	8515,64	8753,28	8374,02	8553,05	8554,7	8538,92	8577,38	8519,84
P21		9505,96	9389,23	9291,78	9187,69	8857,44	8881,98	8,954	9032,28	8933,75	9049,4	9217,19	9277,99	9491,25	9106,4	9544,94	9085,1	9511,25	9288,6	9559,99	9189,14	9456,97	9327,59	9340,78	9396,38	9373,61
P22		8690,93	8692,48	8570,67	8540,91	8082,41	8112,99	8,102	8198,11	8195,78	8274,29	8375,49	8417,63	8595,41	8504,13	8492,71	8213,22	8660,32	8583,68	8821,95	8546,45	8623,34	8662	8697,82	8664,31	8613,44
P23		9812,48	9730,5	9703,3	9592,8	9138,38	9150,48	9265,83	9318,63	9226,16	9393,25	9518,85	9568,69	9832,33	9633,76	9807,46	9319,2	9747,27	9647,96	9878,17	9565,25	9738,78	9782,1	9763,82	9795	9791,44
P24		8564,69	8556,87	8467,36	8401,58	8143,99	8091,37	8144,48	8204,03	8233,98	8335,44	8394,79	8516,26	8562,53	8324,16	8667,57	8240,8	8546,35	8490,92	8672,97	8363,73	8483,6	8474,04	8533,84	8567,31	8522,11
P25		9802,3	9798,21	9681,23	9616,61	9028	9008,2	9137,66	9212,14	9169,67	9260,73	9398,04	9481,72	9842,27	9282,92	9729	9338,61	9833,66	9669,71	9906,74	9587,72	9777,35	9793,86	9814,64	9823,61	9758,73
P26		8500,63	8497,45	8433,91	8378,43	7763,87	7782,59	7764,7	7903,78	7898,62	7917,51	8001,53	8073,08	8463,3	8270,29	8270,41	7894,45	8449,74	8427,5	8596,64	8319,04	8442,82	8430,07	8485,92	8543,31	8521,58
P27		10069,9	10023,4	9952,16	9919,53	9072,08	9137,44	9193,23	9298,79	9248,54	9328,76	9473,6	9469,21	10013,6	9998,67	9793,81	9333,8	9962,65	9962,94	10160,4	9847,23	10031,2	10101,2	10109,2	10103,8	10067,5
P28		8506,56	8446,36	8429,23	8368,14	7954,31	7914,22	7974,13	7990,54	7977,45	8119,13	8234,22	8276,35	8491	8299,56	8419,52	8059,38	8445,63	8399,48	8583,28	8339,95	8440,15	8523,02	8512,65	8490,61	8456,36
P29		9889,9	9891,23	9808,42	9725,22	8814,67	8782,36	8926,08	9074,18	8902,84	9096,26	9231,91	9284,68	9906,83	9798,48	9607,16	9095,62	9835,7	9790,86	9973,04	9695,12	9907,44	9884,39	9880,83	9939,33	9950,69
P30		8758,97	8772,32	8719,68	8639,33	7925,23	7975,25	8020,89	8125,9	8047,89	8132,19	8209	8286,64	8726,26	8597,75	8454,98	8208,07	8705,29	8674,04	8817,99	8569,21	8728,19	8803,5	8776,22	8777,1	8767,82

5.2. BROBin Algoritması ile Elde Edilen Sonuçlar

5.2.1. Parametre ayarlaması

BROBin algoritmasının etkinliğini analiz etmek için 30 SUKP problemi üzerinde yaklaşık çözümler elde edilerek deneysel sonuçlar oluşturulmuştur. Mutasyon oranı ve etiket oranına karar verme aşamasında problem kümesinin ilk 6 problemi üzerinde 100 tekrarlı olarak farklı mutasyon oranı (0.06, 0.07, 0.08, 0.09, 0.10, 0.11, 0.12) ve etiket

oranı (0.05, 0.10, 0.15, 0.20, 0.25) için toplam 35 ayrı test yapılmıştır. Elde edilen sonuçlara göre en yüksek değerlerin elde edildiği mutasyon oranı ve etiket oranı çiftlerinden 9'u Tablo 5.7'de gösterilmiştir. Tablodan da görüldüğü üzere mutasyon oranı 0.6 için etiket oranı 0.05, 0.10, 0.25 alınarak elde edilen sonuçlar aynı olmuştur. Yapılan Friedman sıralamasına göre etiket oranının 0.2, mutasyon oranının 0.09 alınmasına karar verilmiştir.

Tablo 5.7. Mutasyon oranı ve Etiket oranına göre 6 SUKP problemi için elde edilen sonuçlar

Mutasyon Oranı		0.06	0.06	0.06	0.07	0.08	0.09	0.10	0.11	0.12
Etiket Oranı		0.05	0.10	0.25	0.25	0.15	0.20	0.20	0.10	0.10
Problem	Sonuçlar									
P1	En iyi	12020	12020	12020	12020	12020	12045	11581	11912	11685
	Ort.	11406,00	11406,00	11406,00	11455,00	11403,00	11410,00	11362,00	11375,00	11395,00
	En kötü	10969	10969	10969	11060	10958	11069	11066	11049	10993
	Std. sapma	1,34E+02	1,34E+02	1,34E+02	1,58E+02	1,42E+02	1,43E+02	1,15E+02	1,46E+02	1,14E+02
P2	En iyi	12369	12369	12369	12369	12369	12369	12369	12369	12369
	Ort.	11990,00	11990,00	11990	12028	12055,00	12053,00	12091,00	12099,00	11995,00
	En kötü	11374	11374	11374	11274	11274	11251	11274	11374	11274
	Std. sapma	4,19E+02	4,19E+02	4,19E+02	4,03E+02	4,07E+02	4,17E+02	3,92E+02	3,60E+02	4,27E+02
P3	En iyi	13251	13251	13251	13283	13283	13251	13251	13283	13251
	Ort.	13048,00	13048,00	13048	13036	13049,00	13041,00	13049,00	13045,00	13052,00
	En kötü	13044	13044	13044	12675	13026	12675	12980	12675	12675
	Std. sapma	2,20E+01	2,20E+01	2,20E+01	6,58E+01	2,97E+01	6,12E+01	3,08E+01	6,65E+01	5,80E+01
P4	En iyi	12348	12348	12348	12479	12348	12348	12348	12479	12479
	Ort.	12102	12102	12102	12110	12089	12113	12110	12111	12098
	En kötü	11576	11576	11576	11969	11576	11969	11969	11969	11969
	Std. sapma	1,11E+02	1,11E+02	1,11E+02	9,09E+01	1,02E+02	8,27E+01	9,05E+01	1,01E+02	9,49E+01
P5	En iyi	14044	14044	14044	14044	14044	14044	14044	14044	14044
	Ort.	13857,00	13857,00	13857	13868	13860,00	13857,00	13863,00	13843,00	13852,00
	En kötü	13664	13664	13664	13469	13659	13664	13659	13618	13659
	Std. sapma	8,67E+01	8,67E+01	8,67E+01	1,04E+02	8,28E+01	9,45E+01	8,30E+01	7,96E+01	8,44E+01
P6	En iyi	13508	13508	13508	13508	13508	13508	13508	13508	13508
	Ort.	13310	13310	13310	13333	13370	13385,00	13363	13348	13375
	En kötü	12809	12809	12809	12860	12666	12319	12713	12809	12668
	Std. sapma	1,99E+02	1,99E+02	1,99E+02	1,89E+02	1,65E+02	1,75E+02	1,70E+02	1,67E+02	1,68E+02
Friedman sıralama	6,0833	6,0833	6,0833	4,25	4,4167	3,5833	3,8333	5,3333	5,3333	
Final sıralama	6	6	6	3	4	1	2	5	5	

30 SUKP problemi Bölüm 4.4'de önerilen BROBin algoritması ile 100 tekrarlı, popülasyon boyutu 20, iterasyon sayısı ise SUKP probleminin içeriğinde bulunan nesne sayısı m , eleman sayısı n olmak üzere $max(m, n)$ şeklinde alınarak çözülmüştür. Elde edilen 100 çözümün maksimum, minimum, ortalama değeri, standart sapması hesaplanmıştır.

O2 TF yardımıyla elde edilen sonuçlar (BROTF-O2) ile BROBin sonuçlarının karşılaştırılması Tablo 5.8'de verilmiştir. BROBin ile elde edilen sonuçların 30 probleminin tamamında O2 TF ile elde edilen sonuçlardan daha iyi olduğu görülmektedir.

Tablo 5.8. BROTF-O2 ile BROBin sonuçlarının karşılaştırılması

Problem	Sonuçlar	BROTF-O2	BROBin	Problem	Sonuçlar	BROTF-O2	BROBin
P1	En iyi	11681	12045	P16	En iyi	11630	12245
	Ort.	11049,57	11410,00		Ort.	10288,73	10961
	En kötü	10645	11069		En kötü	9498	9889
	Std. sapma	2,00E+02	1,43E+02		Std. sapma	4,47E+02	5,95E+02
P2	En iyi	12369	12369	P17	En iyi	12103	12416
	Ort.	11699,6	12053,00		Ort.	11337,33	11825
	En kötü	10842	11251		En kötü	10585	11227
	Std. sapma	3,88E+02	4,17E+02		Std. sapma	3,19E+02	2,00E+02
P3	En iyi	13251	13251	P18	En iyi	10283	10559
	Ort.	12835,82	13041,00		Ort.	9476,05	9937,1
	En kötü	12413	12675		En kötü	8849	9072
	Std. sapma	1,62E+02	6,12E+01		Std. sapma	3,02E+02	2,95E+02
P4	En iyi	12479	12348	P19	En iyi	9979	9877
	Ort.	12054,37	12113		Ort.	9195,31	9403,7
	En kötü	11556	11969		En kötü	8674	8856
	Std. sapma	1,40E+02	8,27E+01		Std. sapma	2,54E+02	2,03E+04
P5	En iyi	13814	14044	P20	En iyi	9345	9460
	Ort.	13555,9	13857,00		Ort.	8753,28	8903,2
	En kötü	13032	13664		En kötü	8147	8271
	Std. sapma	1,74E+02	9,45E+01		Std. sapma	2,72E+02	2,47E+02
P6	En iyi	13407	13508	P21	En iyi	10367	10743
	Ort.	13039,28	13385,00		Ort.	9559,99	9971,1
	En kötü	11993	12319		En kötü	8958	9353
	Std. sapma	3,86E+02	1,75E+02		Std. sapma	2,95E+02	2,94E+02
P7	En iyi	12881	13458	P22	En iyi	9653	9990
	Ort.	12063,53	12750		Ort.	8821,95	9179,8
	En kötü	11271	12209		En kötü	8242	8417
	Std. sapma	3,46E+02	2,99E+02		Std. sapma	2,33E+02	2,68E+02
P8	En iyi	10584	11285	P23	En iyi	10646	10593
	Ort.	9875,31	10525		Ort.	9878,17	10160
	En kötü	9426	9912		En kötü	9408	9665
	Std. sapma	2,26E+02	2,72E+02		Std. sapma	2,36E+02	2,04E+02
P9	En iyi	13137	13405	P24	En iyi	9703	9926
	Ort.	12343,78	13032		Ort.	8672,97	8996
	En kötü	11252	11640		En kötü	8078	8464
	Std. sapma	4,67E+02	2,65E+02		Std. sapma	2,84E+02	3,03E+02
P10	En iyi	13671	14215	P25	En iyi	10697	10721
	Ort.	12809,18	13386		Ort.	9906,74	10202
	En kötü	11886	12736		En kötü	9354	9770
	Std. sapma	3,55E+02	339,15		Std. sapma	2,44E+02	201,78
P11	En iyi	12109	12211	P26	En iyi	9057	9396
	Ort.	10876,58	11399		Ort.	8596,64	8939,1
	En kötü	10282	10405		En kötü	8115	8412
	Std. sapma	3,66E+02	3,75E+02		Std. sapma	1,92E+02	2,08E+02
P12	En iyi	11533	12019	P27	En iyi	10827	11077
	Ort.	10668,83	11330		Ort.	10160,39	10532
	En kötü	9900	10449		En kötü	9586	10116
	Std. sapma	3,81E+02	3,19E+02		Std. sapma	2,06E+02	2,08E+02
P13	En iyi	10869	11234	P28	En iyi	9028	9329
	Ort.	10234,07	10780		Ort.	8583,28	8764
	En kötü	9612	10400		En kötü	8235	8351
	Std. sapma	2,86E+02	1,90E+02		Std. sapma	1,86E+02	1,77E+02
P14	En iyi	10194	11010	P29	En iyi	10460	10714
	Ort.	9592,46	10267		Ort.	9973,04	10243
	En kötü	9040	9595		En kötü	9573	9836
	Std. sapma	2,52E+02	2,93E+02		Std. sapma	1,75E+02	1,55E+02
P15	En iyi	10264	10701	P30	En iyi	9383	9598
	Ort.	9661,32	10064		Ort.	8817,99	9180,8
	En kötü	9026	9324		En kötü	8342	8725
	Std. sapma	2,45E+02	2,41E+02		Std. sapma	1,98E+02	1,90E+02

BROBin algoritması ile elde edilen sonuçlar literatürde var olan bazı algoritmalar ile karşılaştırılarak Tablo 5.9 elde edilmiştir. Karşılaştırmada kullanılan algoritmalar A-SUKP (Arulselvan, 2014), Genetik algoritma (GA) (Ack ve ark., 2001), BABC (He ve ark., 2018), ABC_{bin} (Kiran, 2015), binDE (Engelbrecht ve Pampará, 2007) ve bWSA (Baykasoğlu et al., 2020)'dir. A-SUKP, GA, BABC, ABC_{bin} ve BinDE algoritmalarının deneysel sonuçları (He ve ark., 2018) çalışmasından alınmıştır. bWSA deneysel sonuçları (Feng ve ark., 2019) araştırmasından alınmıştır. Elde edilen deneysel sonuçlar Friedman sıralamasına tabi tutulmuştur. Önerilen BROBin algoritmasının 30 SUKP problemi için karşılaştırma yapılan diğer 6 algoritmadan daha iyi sonuç verdiği sıralamada birinci olduğu görülmüştür.



Tablo 5.9. Farklı algoritmalar ile BROBin algoritmasının SUKP problemleri için karşılaştırılması

Problem	Sonuçlar	A-SUKP	GA	BABC	ABCbin	BinDE	bWSA	Bin BRO
P1	Ort.	10231	11092,7	11182,7	10879,5	11075	11233,16	11410,00
	Std. Sapma	0,00E+00	1,71E+02	1,84E+02	1,64E+04	1,19E+02	2,17E+02	1,43E+02
P2	Ort.	10483	11326,3	12081,6	11485,3	11875,9	11342,7	12053,00
	Std. Sapma	0,00E+00	4,17E+02	1,94E+02	2,48E+02	3,37E+02	4,75E+02	4,17E+02
P3	Ort.	12459	12956,4	13028,5	12818,5	12991	12915,67	13041,00
	Std. Sapma	0,00E+00	1,31E+02	9,26E+01	1,53E+02	7,60E+01	1,85E+02	6,12E+01
P4	Ort.	11119	11546	12155	12049,3	12123,9	11527,41	12113
	Std. Sapma	0,00E+00	2,15E+02	5,33E+01	9,61E+01	6,76E+01	3,32E+02	8,27E+01
P5	Ort.	13634	13806	13734,9	13547,2	13675,9	13492,71	13857,00
	Std. Sapma	0,00E+00	1,45E+02	7,08E+01	1,19E+02	1,20E+02	3,25E+02	9,45E+01
P6	Ort.	11325	12234,8	13352,4	13103,1	13212,8	12487,88	13385,00
	Std. Sapma	0,00E+00	3,89E+02	1,55E+02	3,43E+02	2,87E+02	7,18E+02	1,75E+02
P7	Ort.	11508	12236,6	12522,8	11667,9	12277,5	12689,09	12750
	Std. Sapma	0,00E+00	1,98E+02	2,01E+02	1,77E+02	2,34E+02	3,37E+02	2,99E+02
P8	Ort.	8621	10351,5	10150,6	9684,33	10085,4	10228,07	10525
	Std. Sapma	0,00E+00	2,08E+02	1,53E+02	1,85E+02	1,61E+02	2,87E+02	2,72E+02
P9	Ort.	11 292	12492,5	13064,4	11861,5	12940,7	12657,65	13032
	Std. Sapma	0,00E+00	3,20E+02	9,96E+01	3,25E+02	2,06E+02	3,20E+02	2,65E+02
P10	Ort.	12262	12802,9	13359,2	12537	13110	12585,35	13386
	Std. Sapma	0,00E+00	2,92E+02	2,35E+02	2,90E+02	2,70E+02	3,03E+02	3,39E+02
P11	Ort.	10328	10888,7	11194,3	10424,1	10969,4	11430,23	11399
	Std. Sapma	0,00E+00	2,38E+02	2,50E+02	1,98E+02	3,03E+02	4,03E+02	3,75E+02
P12	Ort.	9784	10827,5	10945	10345,9	10717,1	11062,06	11330
	Std. Sapma	0,00E+00	3,34E+02	2,55E+02	2,73E+02	3,41E+02	4,24E+02	3,19E+02
P13	Ort.	9961	10640,1	10775,9	9957,09	10661,3	11105,09	10780
	Std. Sapma	0,00E+00	1,27E+02	1,17E+02	1,41E+02	1,50E+02	1,98E+02	1,90E+02
P14	Ort.	9618	10190,3	9897,92	9424,15	9832,32	10452,03	10267
	Std. Sapma	0,00E+00	2,50E+02	1,87E+02	1,97E+02	2,33E+02	4,17E+02	2,93E+02
P15	Ort.	8941	9980,87	9994,76	9339,3	9899,24	10366,21	10064
	Std. Sapma	0,00E+00	1,43E+02	1,54E+02	1,58E+02	1,53E+02	2,57E+02	2,41E+02
P16	Ort.	9432	10349,8	10902,9	9957,85	10499,4	10901,59	10961
	Std. Sapma	0,00E+00	2,15E+02	4,49E+02	2,77E+02	4,04E+02	5,09E+02	5,95E+02
P17	Ort.	10208	11755,1	11945,8	10922,3	11864,4	12227,56	11825
	Std. Sapma	0,00E+00	1,44E+02	1,28E+02	1,83E+02	1,60E+02	3,08E+02	2,00E+02
P18	Ort.	9183	10099,2	9859,69	9186,87	9710,37	10216,71	9937,1
	Std. Sapma	0,00E+00	3,37E+02	1,77E+02	1,48E+02	2,08E+02	3,51E+02	2,95E+02
P19	Ort.	8672	9432,82	9537,5	8904,94	9314,57	9778,03	9403,7
	Std. Sapma	0,00E+00	1,64E+02	1,85E+02	1,12E+02	1,92E+02	2,21E+02	2,03E+04
P20	Ort.	8064	8703,66	9090,03	8407,06	8846,99	9203,52	8903,2
	Std. Sapma	0,00E+00	1,54E+02	1,57E+02	1,49E+02	2,11E+02	3,03E+02	2,47E+02
P21	Ort.	9076	9641,85	10065,2	9187,76	9681,46	10785,74	9971,1
	Std. Sapma	0,00E+00	1,69E+02	2,41E+02	1,67E+02	2,75E+02	3,61E+02	2,94E+02
P22	Ort.	8514	9326,77	9135,98	8539,95	9020,87	9587,72	9179,8
	Std. Sapma	0,00E+00	1,92E+02	1,52E+02	1,62E+02	1,51E+02	3,60E+02	2,68E+02
P23	Ort.	9751	10112,4	10101,1	9549,04	9975,8	10624,79	10160
	Std. Sapma	0,00E+00	1,58E+02	1,97E+02	1,41E+02	1,86E+02	2,66E+02	2,04E+02
P24	Ort.	8497	8793,76	9032,95	8365,62	8768,42	9580,64	8996
	Std. Sapma	0,00E+00	1,70E+02	1,94E+02	1,53E+02	2,12E+02	4,12E+02	3,03E+02
P25	Ort.	9559	9993,16	10483,4	9615,37	10159,4	10607,21	10202
	Std. Sapma	0,00E+00	1,18E+02	2,28E+02	1,51E+02	1,98E+02	1,92E+02	2,02E+02
P26	Ort.	8157	8849,46	9085,57	8347,82	8919,64	9141,94	8939,1
	Std. Sapma	0,00E+00	1,42E+02	1,16E+02	1,23E+02	1,69E+02	1,80E+02	2,08E+02
P27	Ort.	9864	10567,9	10452,2	9910,32	10363,8	10921,58	10532
	Std. Sapma	0,00E+00	1,23E+02	1,14E+02	1,21E+02	9,34E+01	3,52E+02	2,08E+02
P28	Ort.	8299	8692,67	8857,89	8365,04	8783,99	9013,09	8764
	Std. Sapma	0,00E+00	1,80E+02	9,46E+01	1,14E+02	1,31E+02	2,05E+02	1,77E+02
P29	Ort.	9615	10185,4	10328,5	9738,17	10227,7	10482,8	10243
	Std. Sapma	0,00E+00	1,14E+02	9,16E+04	1,12E+02	1,03E+02	1,66E+02	1,55E+02
P30	Ort.	7883	8882,88	9180,74	8617,91	9096,13	9141,94	9180,8
	Std. Sapma	0,00E+00	1,58E+02	8,49E+01	1,41E+02	1,45E+02	1,80E+02	1,90E+02
Friedman Sıralama		1,20	3,83	5,47	2,07	3,93	5,70	5,80
Final Sıralama		7	5	3	6	4	2	1

6. SONUÇLAR VE ÖNERİLER

6.1. Sonuçlar

Bu çalışmada, Ölüm Oyunu Optimizasyon (BRO) algoritmasının sürekli yapısı, ikili problemlere uyum sağlayacak şekilde iki farklı yöntemle dönüştürülmüştür. Birinci yöntemde, sürekli çözüm uzayına sahip aday çözümler, 25 farklı TF (S, V, U, T, Z, O, Hiperbolik tanjant) kullanılarak ikili forma dönüştürülmüş ve algoritmanın temel işleyişine müdahale edilmeden ikili BRO versiyonu elde edilmiştir. Yapılan deneysel çalışmalar sonucunda, O2 TF'sinin diğer transfer fonksiyonlara kıyasla daha iyi performans sergilediği gözlemlenmiştir. S-grubu TF'lerin, V-grubu TF'lerden daha başarılı performans sergilediği görülmektedir. S-grubu içerisindeki karşılaştırmada ise S1 ve S2'nin, S3 ve S4'e oranla daha yüksek etkililik düzeyine sahip olduğu anlaşılmaktadır. Öte yandan, P5 problemi özelinde en iyi sonucun V1 ile elde edildiği, genel olarak ise V-grubu içerisinde V4'ün daha üstün performans gösterdiği gözlemlenmiştir. U-grubu içerisinde en dikkat çekici sonuçların U4 fonksiyonu ile elde edildiği, Taper-şekilli grupta ise T1 fonksiyonunun öne çıktığı ve grubun genel performansında belirleyici bir rol üstlendiği görülmektedir. T-grubu fonksiyonlarının genel başarı düzeyinin, U-grubu fonksiyonlarına kıyasla daha yüksek olduğu, Hiperbolik tanjant fonksiyonunun ise U-grubu fonksiyonlarına göre daha iyi sonuçlar ürettiği ancak T1 fonksiyonunun gerisinde kaldığı tespit edilmektedir.

İkinci yöntemde ise algoritmanın bazı adımlarına müdahalelerde bulunarak algoritma ikili arama uzayında çalışacak şekilde geliştirilmiştir ve BROBin adı verilen özelleştirilmiş bir versiyon geliştirilmiştir. Bu versiyonun, TF tabanlı yaklaşıma göre belirgin şekilde daha başarılı sonuçlar verdiği tespit edilmiştir. Böylece BRO algoritmasının doğrudan problem uzayına taşınmasının, dönüşüm fonksiyonlarına göre çok daha etkili sonuçlar elde ettiği görülmüştür.

Geliştirilen BROBin algoritmasının performansını daha detaylı değerlendirmek amacıyla, literatürde yaygın olarak kullanılan altı farklı meta-sezgisel algoritma (A-SUKP, GA, BABC, ABC_{bin}, binde, bWSA) ile karşılaştırma yapılmıştır. Friedman istatistik testi sonuçları, BROBin'in 30 SUKP problemi üzerinde diğer algoritmalara kıyasla istatistiksel olarak anlamlı bir üstünlük sağladığını kanıtlamıştır. Özellikle yüksek

boyutlu ve karmaşık kısıtlar içeren problemlerde, algoritmanın çözüm kalitesi ve kararlılık açısından öne çıktığı gözlemlenmiştir. Bu sonuçlar, BRO algoritmasının ikili optimizasyon problemlerine başarıyla uyarlanabileceğini ve bu alanda etkili bir alternatif olarak kullanılabilirliğini göstermektedir.

6.2. Öneriler

Bu çalışmanın bulguları ışığında, gelecek araştırmalar için çeşitli öneriler sunulabilir. Öncelikle, geliştirilen BROBin algoritmasının performansının daha kapsamlı şekilde değerlendirilebilmesi için farklı ikili optimizasyon problemlerine uygulanması faydalı olacaktır. Özellikle makine öğrenmesi alanında özellik seçimi problemleri ve kombinatoryal optimizasyonun önemli bir parçası olan ikili grafik bölütleme problemleri üzerinde testler yapılabilir. Ayrıca, elektronik devre tasarımı ve yerleştirme problemleri gibi mühendislik uygulamalarında algoritmanın etkinliği araştırılabilir.

Algoritmanın performansını daha da iyileştirmek amacıyla çeşitli hibrid yaklaşımlar denenebilir. Transfer fonksiyonları ile algoritmik operatörlerin birlikte kullanıldığı hibrid yöntemler, arama uzayında daha dengeli bir keşif-denetim dengesi sağlayabilir. Yerel arama operatörlerinin entegrasyonu, özellikle karmaşık problemlerde çözüm kalitesini artırabilir. Diğer popülasyon tabanlı algoritmalarla hibrid çözümler geliştirilerek, farklı optimizasyon tekniklerinin avantajları bir araya getirilebilir.

Yapay zekâ uygulamaları özelinde, derin öğrenme modellerinin ikili ağırlık optimizasyonunda BROBin'in kullanım potansiyeli üzerinde durulabilir. İkili sinir ağlarının eğitim sürecinde ve evrişimli sinir ağlarının (Convolutional Neural Networks-CNN) ikili filtre optimizasyonunda algoritmanın etkinliği değerlendirilebilir. Ayrıca, dinamik ortam problemleri ve çok amaçlı optimizasyon senaryoları için algoritmanın uyarlanabilir versiyonlarının geliştirilmesi, gerçek dünya uygulamalarındaki kullanım alanını genişletebilir.

Hesaplama performansını artırmaya yönelik olarak, paralel ve dağıtık hesaplama tekniklerinin uygulanması önerilebilir. GPU tabanlı hızlandırma yöntemleriyle büyük ölçekli problemlerin çözüm süreleri kısaltılabilir. Dağıtık popülasyon yönetimi stratejileri ise farklı arama uzaylarının eşzamanlı olarak keşfedilmesine olanak sağlayabilir. Bu

öneriler doğrultusunda yapılacak çalışmalar hem algoritmanın teorik temellerini güçlendirecek hem de pratik uygulama alanlarını genişletecektir.



KAYNAKLAR

- Ack, T. B., De Graaf, J. M., Kok, J. N., & Kusters, W. A. (2001). *Theory of Genetic Algorithms*.
- Akan, T., Agahian, S., & Dehkharghani, R. (2022). Battle royale optimizer for solving binary optimization problems. *Software Impacts*, 12, 100274.
- Arulselvan, A. (2014b). A note on the set union knapsack problem. *Discrete Applied Mathematics*, 169, 214–218. <https://doi.org/10.1016/j.dam.2013.12.015>
- Bansal, J. C., & Deep, K. (2012). A modified binary particle swarm optimization for Knapsack problems. *Applied Mathematics and Computation*, 218(22), 11042–11061. <https://doi.org/10.1016/j.amc.2012.05.001>
- Baykasoğlu, A., Ozsoydan, F. B., & Senol, M. E. (2020). Weighted superposition attraction algorithm for binary optimization problems. *Operational Research*, 20(4), 2555–2581. <https://doi.org/10.1007/s12351-018-0427-9>
- Costa, M. F. P., Rocha, A. M. A. C., Francisco, R. B., & Fernandes, E. M. G. P. (2014). Heuristic-based firefly algorithm for bound constrained nonlinear binary optimization. *Advances in Operations Research*, 2014. <https://doi.org/10.1155/2014/215182>
- Engelbrecht, A. P., & Pampará, G. (2007). *Binary Differential Evolution Strategies*.
- Feng, Y., An, H., & Gao, X. (2018). The importance of transfer function in solving set-union knapsack problem based on discrete moth search algorithm. *Mathematics*, 7(1). <https://doi.org/10.3390/math7010017>
- Feng, Y., Yi, J. H., & Wang, G. G. (2019). Enhanced Moth Search Algorithm for the Set-Union Knapsack Problems. *IEEE Access*, 7, 173774–173785. <https://doi.org/10.1109/ACCESS.2019.2956839>
- Formato, R. A. (2015). *Central Force Optimization*.
- García, J., Leiva-Araos, A., Crawford, B., Soto, R., & Pinto, H. (2023). Exploring Initialization Strategies for Metaheuristic Optimization: Case Study of the Set-Union Knapsack Problem. *Mathematics*, 11(12). <https://doi.org/10.3390/math11122695>
- García, J., Lemus-Romani, J., Altimiras, F., Crawford, B., Soto, R., Becerra-Rozas, M., Moraga, P., Becerra, A. P., Fritz, A. P., Rubio, J. M., & Astorga, G. (2021). A binary machine learning cuckoo search algorithm improved by a local search operator for the set-union knapsack problem. *Mathematics*, 9(20). <https://doi.org/10.3390/math9202611>
- Goldschmidt, O., Nehme, D., & Yu, G. (1994). Note: On the set-union knapsack problem. *Naval Research Logistics (NRL)*, 41(6), 833–842.

- [https://doi.org/10.1002/1520-6750\(199410\)41:6<833::AID-NAV3220410611>3.0.CO;2-Q](https://doi.org/10.1002/1520-6750(199410)41:6<833::AID-NAV3220410611>3.0.CO;2-Q)
- Guo, S. S., Wang, J. S., Wang, J. S., & Guo, M. W. (2020). Z-Shaped Transfer Functions for Binary Particle Swarm Optimization Algorithm. *Computational Intelligence and Neuroscience*, 2020. <https://doi.org/10.1155/2020/6502807>
- Hatamlou, A. (2013). Black hole: A new heuristic optimization approach for data clustering. *Information Sciences*, 222, 175–184. <https://doi.org/10.1016/j.ins.2012.08.023>
- He, Y., Xie, H., Wong, T. L., & Wang, X. (2018). A novel binary artificial bee colony algorithm for the set-union knapsack problem. *Future Generation Computer Systems*, 78, 77–86. <https://doi.org/10.1016/j.future.2017.05.044>
- He, Y., Zhang, F., Mirjalili, S., & Zhang, T. (2022). Novel binary differential evolution algorithm based on Taper-shaped transfer functions for binary optimization problems. *Swarm and Evolutionary Computation*, 69. <https://doi.org/10.1016/j.swevo.2021.101022>
- Hojatollah Rajabi Moshtaghi, A. T. E. M. R. M. (2021). A new metaheuristic for optimization. *Journal of Applied Research on Industrial Engineering*, 63–89.
- I. Dahmani, M. F. M. H. and S. S. (2020). *A Hybrid Swarm Optimization-Based Algorithm for the Set-Union Knapsack Problem*.
- J. Kennedy, R. C. E. (1997). *Discrete binary version of the particle swarm algorithm, Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 5 (1997) 4104-4108. IEEE.
- Kaveh, A., & Bakhshpoori, T. (2016). Water Evaporation Optimization: A novel physically inspired optimization algorithm. *Computers and Structures*, 167, 69–85. <https://doi.org/10.1016/j.compstruc.2016.01.008>
- Kellerer, H., Pferschy, U., & Pisinger, D. (2004). Multidimensional Knapsack Problems. *Knapsack Problems*, 235–283. https://doi.org/10.1007/978-3-540-24777-7_9
- Kiran, M. S. (2015). The continuous artificial bee colony algorithm for binary optimization. *Applied Soft Computing Journal*, 33, 15–23. <https://doi.org/10.1016/j.asoc.2015.04.007>
- Lin, G., Guan, J., Li, Z., & Feng, H. (2019). A hybrid binary particle swarm optimization with tabu search for the set-union knapsack problem. *Expert Systems with Applications*, 135, 201–211. <https://doi.org/10.1016/j.eswa.2019.06.007>
- Manghui Tu, L. X. (2016). <https://ieeexplore.ieee.org/abstract/document/7586727>. System Resilience Enhancement through Modularization for Large Scale Cyber Systems. <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7586727>

- Mirjalili, S., & Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation*, 9, 1–14. <https://doi.org/10.1016/j.swevo.2012.09.002>
- Mirjalili, S., Zhang, H., Mirjalili, S., Chalup, S., & Noman, N. (2020). A Novel U-Shaped Transfer Function for Binary Particle Swarm Optimisation. *Advances in Intelligent Systems and Computing*, 1138, 241–259. https://doi.org/10.1007/978-981-15-3290-0_19
- Navathe, S., Ceri, S., Wiederhold, G., & Dou, J. (1984). *Vertical Partitioning Algorithms for Database Design*. 680–710.
- Nezamabadi-Pour, H. (2015). A quantum-inspired gravitational search algorithm for binary encoded optimization problems. *Engineering Applications of Artificial Intelligence*, 40, 62–75. <https://doi.org/10.1016/j.engappai.2015.01.002>
- Ozsoydan, F. B., & Baykasoglu, A. (2019). A swarm intelligence-based algorithm for the set-union knapsack problem. *Future Generation Computer Systems*, 93, 560–569. <https://doi.org/10.1016/j.future.2018.08.002>
- Ozsoydan, F. B., & Gölcük, İ. (2023). A reinforcement learning based computational intelligence approach for binary optimization problems: The case of the set-union knapsack problem. *Engineering Applications of Artificial Intelligence*, 118. <https://doi.org/10.1016/j.engappai.2022.105688>
- Pampará, G., & Engelbrecht, A. P. (2011). Binary artificial bee colony optimization. *IEEE SSCI 2011 - Symposium Series on Computational Intelligence - SIS 2011: 2011 IEEE Symposium on Swarm Intelligence*, 170–177. <https://doi.org/10.1109/SIS.2011.5952562>
- Pampara, G., Franken, N., & Engelbrecht, A. P. (2005). Combining particle swarm optimisation with angle modulation to solve binary problems. *2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings, 1*, 89–96. <https://doi.org/10.1109/cec.2005.1554671>
- Rahkar Farshi, T. (2021). Battle royale optimization algorithm. *Neural Computing and Applications*, 33(4), 1139–1157. <https://doi.org/10.1007/s00521-020-05004-4>
- Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A Gravitational Search Algorithm. *Information Sciences*, 179(13), 2232–2248. <https://doi.org/10.1016/j.ins.2009.03.004>
- Sonuç, E., & Özcan, E. (2024). CUDA-based parallel local search for the set-union knapsack problem. *Knowledge-Based Systems*, 299. <https://doi.org/10.1016/j.knosys.2024.112095>
- Tang, C. S., & Denardo, E. V. (1988). Models Arising from a Flexible Manufacturing Machine, Part II: Minimization of the Number of Switching

- Instants. <https://doi.org/10.1287/Opres.36.5.778>, 36(5), 778–784.
<https://doi.org/10.1287/OPRE.36.5.778>
- Wang, L., Wang, X., Fu, J., & Zhen, L. (2008). *A Novel Probability Binary Particle Swarm Optimization Algorithm and Its Application*.
- Wei, Z., & Hao, J. K. (2019). Iterated two-phase local search for the Set-Union Knapsack Problem. *Future Generation Computer Systems*, 101, 1005–1017.
<https://doi.org/10.1016/j.future.2019.07.062>
- Wei, Z., & Hao, J. K. (2021a). Kernel based tabu search for the Set-union Knapsack Problem. *Expert Systems with Applications*, 165.
<https://doi.org/10.1016/j.eswa.2020.113802>
- Wei, Z., & Hao, J. K. (2021b). Multistart solution-based tabu search for the Set-Union Knapsack Problem. *Applied Soft Computing*, 105.
<https://doi.org/10.1016/j.asoc.2021.107260>
- Yang, X., Vernitski, A., & Carrea, L. (2016). An approximate dynamic programming approach for improving accuracy of lossy data compression by Bloom filters. *European Journal of Operational Research*, 252(3), 985–994.
<https://doi.org/10.1016/J.EJOR.2016.01.042>
- Yonaba, H., Anctil, F., & Fortin, V. (2010). Comparing Sigmoid Transfer Functions for Neural Network Multistep Ahead Streamflow Forecasting. *Journal of Hydrologic Engineering*, 15(4), 275–283.
[https://doi.org/10.1061/\(asce\)he.1943-5584.0000188](https://doi.org/10.1061/(asce)he.1943-5584.0000188)
- Zhao, J., & Hifi, M. (2024a). A Cooperative Method for Solving the Set-Union Knapsack Problem. *10th 2024 International Conference on Control, Decision and Information Technologies, CoDIT 2024*, 1219–1224.
<https://doi.org/10.1109/CoDIT62066.2024.10708580>
- Zhao, J., & Hifi, M. (2024b). An adaptive evolutionary search-based method for efficiently tackling the set-union knapsack problem. *Information Sciences*, 676.
<https://doi.org/10.1016/j.ins.2024.120855>
- Zhu, H., He, Y., Wang, X., & Tsang, E. C. C. (2017). Discrete differential evolutions for the discounted $\{0-1\}$ knapsack problem. In *Chinese Journal of Computers and so on* (Vol. 10, Issue 4).

EKLER

EK-1: Mutasyon ve Etiket Oranı için Elde Edilen Parametre Analizi Sonuçları

Tablo A1. Mutasyon Oranı 0.06 için Elde Edilen Sonuçlar

Problem	Sonuçlar	Etiket Oranı 0.05	Etiket Oranı 0.10	Etiket Oranı 0.15	Etiket Oranı 0.20	Etiket Oranı 0.25
P1	En iyi	12020	12020	12020	11912	12020
	Ort.	11406,00	11406,00	11404,00	11402,00	11406,00
	En kötü	10969	10969	10993	11060	10969
	Std. sapma	1,34E+02	1,34E+02	1,33E+02	1,28E+02	1,34E+02
P2	En iyi	12369	12369	12369	12369	12369
	Ort.	11990,00	11990,00	11998,00	11979,00	11990
	En kötü	11374	11374	11274	11274	11374
	Std. sapma	4,19E+02	4,19E+02	4,16E+02	4,27E+02	4,19E+02
P3	En iyi	13251	13251	13251	13283	13251
	Ort.	13048,00	13048,00	13043,00	13050,00	13048
	En kötü	13044	13044	12736	12953	13044
	Std. sapma	2,20E+01	2,20E+01	5,84E+01	3,54E+01	2,20E+01
P4	En iyi	12348	12348	12479	12479	12348
	Ort.	12102	12102	12100	12095	12102
	En kötü	11576	11576	11855	11969	11576
	Std. sapma	1,11E+02	1,11E+02	1,05E+02	9,51E+01	1,11E+02
P5	En iyi	14044	14044	14044	14044	14044
	Ort.	13857,00	13857,00	13861,00	13847,00	13857
	En kötü	13664	13664	13659	13615	13664
	Std. sapma	8,67E+01	8,67E+01	8,76E+01	8,80E+01	8,67E+01
P6	En iyi	13508	13508	13508	13508	13508
	Ort.	13310	13310	13308	13319	13310
	En kötü	12809	12809	12142	12238	12809
	Std. sapma	1,99E+02	1,99E+02	2,42E+02	2,28E+02	1,99E+02
Friedman sıralama		2,6667	2,6667	3,3333	3,6667	2,6667
Final sıralama		1	1	2	3	1

Tablo A2. Mutasyon Oranı 0.07 için Elde Edilen Sonuçlar

Problem	Sonuçlar	Etiket Oranı 0.05	Etiket Oranı 0.10	Etiket Oranı 0.15	Etiket Oranı 0.20	Etiket Oranı 0.25
P1	En iyi	12020	12045	12045	12045	12020
	Ort.	11413,00	11411,00	11419,00	11407,00	11455,00
	En kötü	10993	10993	11069	10783	11060
	Std. sapma	1,41E+02	1,61E+02	1,71E+02	1,68E+02	1,58E+02
P2	En iyi	12369	12369	12369	12369	12369
	Ort.	11983,00	11976,00	12066,00	12072,00	12028
	En kötü	11274	11374	11274	11274	11274
	Std. sapma	4,27E+02	4,19E+02	3,82E+02	3,98E+02	4,03E+02
P3	En iyi	13167	13251	13283	13251	13283
	Ort.	13041,00	13046,00	13043,00	13037,00	13036
	En kötü	12736	12888	12675	12646	12675
	Std. sapma	3,92E+01	2,99E+01	5,92E+01	6,89E+01	6,58E+01
P4	En iyi	12348	12274	12274	12479	12479
	Ort.	12115	12094	12090	12100	12110
	En kötü	11576	11855	11969	11934	11969
	Std. sapma	1,06E+02	8,69E+01	8,84E+01	9,51E+01	9,09E+01
P4	En iyi	14044	14044	14044	14044	14044
	Ort.	13846,00	13847,00	13858,00	13851,00	13868
	En kötü	13618	13538	13615	13659	13469
	Std. sapma	1,06E+02	8,27E+01	8,97E+01	9,42E+01	1,04E+02
P4	En iyi	13508	13508	13508	13508	13508
	Ort.	13309	13333	13329	13337	13333
	En kötü	12570	12623	12571	12809	12860
	Std. sapma	2,28E+02	1,90E+02	2,16E+02	1,88E+02	1,89E+02
Friedman sıralama		3,5	3,4167	2,8333	2,8333	2,4167
Final sıralama		4	3	2	2	1

Tablo A3. Mutasyon Oranı 0.08 için Elde Edilen Sonuçlar

Problem	Sonuçlar	Etiket Oranı 0.05	Etiket Oranı 0.10	Etiket Oranı 0.15	Etiket Oranı 0.20	Etiket Oranı 0.25
P1	En iyi	11851	11947	12020	12045	12045
	Ort.	11413,00	11417,00	11403,00	11415,00	11418,00
	En kötü	10993	10902	10958	10717	11000
	Std. sapma	1,33E+02	1,40E+02	1,42E+02	1,56E+02	1,40E+02
P2	En iyi	12369	12369	12369	12369	12369
	Ort.	12036,00	12032,00	12055,00	12039,00	12007
	En kötü	11274	11129	11274	11274	11274
	Std. sapma	4,19E+02	4,20E+02	4,07E+02	4,05E+02	4,32E+02
P3	En iyi	13283	13251	13283	13283	13089
	Ort.	13049,00	13044,00	13049,00	13046,00	13037
	En kötü	13044	12673	13026	12675	12646
	Std. sapma	3,17E+01	5,75E+01	2,97E+01	6,30E+01	5,43E+01
P4	En iyi	12479	12274	12348	12479	12348
	Ort.	12107	12105	12089	12121	12089
	En kötü	11969	11927	11576	11969	11819
	Std. sapma	9,19E+01	9,58E+01	1,02E+02	9,15E+01	8,51E+01
P4	En iyi	14044	14044	14044	14044	14044
	Ort.	13851,00	13851,00	13860,00	13849,00	13862
	En kötü	13664	13631	13659	13659	13659
	Std. sapma	8,32E+01	9,23E+01	8,28E+01	8,32E+01	9,86E+01
P4	En iyi	13508	13508	13508	13508	13508
	Ort.	13334	13354	13370	13359	13308
	En kötü	12571	12770	12666	12238	12860
	Std. sapma	2,25E+02	1,76E+02	1,65E+02	2,11E+02	1,94E+02
Friedman sıralama		3	3,25	2,5	2,6667	3,5833
Final sıralama		3	4	1	2	5

Tablo A4. Mutasyon Oranı 0.09 için Elde Edilen Sonuçlar

Problem	Sonuçlar	Etiket Oranı 0.05	Etiket Oranı 0.10	Etiket Oranı 0.15	Etiket Oranı 0.20	Etiket Oranı 0.25
P1	En iyi	11876	11922	11851	12045	11912
	Ort.	11404,00	11405,00	11399,00	11410,00	11394,00
	En kötü	10959	11108	10919	11069	10970
	Std. sapma	1,48E+02	1,16E+02	1,38E+02	1,43E+02	1,37E+02
P2	En iyi	12369	12369	12369	12369	12369
	Ort.	12010,00	12064,00	12068,00	12053,00	12025
	En kötü	11283	11374	11374	11251	11374
	Std. sapma	4,13E+02	3,77E+02	4,03E+02	4,17E+02	4,05E+02
P3	En iyi	13251	13251	13167	13251	13251
	Ort.	13049,00	13052,00	13048,00	13041,00	13047
	En kötü	12847	12943	13019	12675	13044
	Std. sapma	3,75E+01	3,84E+01	1,94E+01	6,12E+01	2,16E+01
P4	En iyi	12274	12348	12348	12348	12479
	Ort.	12103	12071	12105	12113	12083
	En kötü	11969	11954	11969	11969	11957
	Std. sapma	8,89E+01	8,42E+01	9,15E+01	8,27E+01	9,51E+01
P4	En iyi	14044	14044	14044	14044	14044
	Ort.	13848,00	13848,00	13846,00	13857,00	13832
	En kötü	13659	13659	13659	13664	13659
	Std. sapma	9,58E+01	9,10E+01	9,12E+01	9,45E+01	8,82E+01
P4	En iyi	13508	13508	13508	13508	13508
	Ort.	13351	13344	13383	13385,00	13365
	En kötü	12860	12860	12570	12319	12809
	Std. sapma	1,68E+02	1,67E+02	1,66E+02	1,75E+02	1,75E+02
Friedman sıralama		3,25	2,9167	2,6667	2	4,1667
Final sıralama		4	3	2	1	5

Tablo A5. Mutasyon Oranı 0.10 için Elde Edilen Sonuçlar

Problem	Sonuçlar	Etiket Oranı 0.05	Etiket Oranı 0.10	Etiket Oranı 0.15	Etiket Oranı 0.20	Etiket Oranı 0.25
P1	En iyi	12020	11912	11752	11581	12020
	Ort.	11408,00	11381,00	11362,00	11362,00	11407,00
	En kötü	10969	10993	10993	11066	11099
	Std. sapma	1,63E+02	1,25E+02	1,26E+02	1,15E+02	1,55E+02
P2	En iyi	12369	12369	12369	12369	12369
	Ort.	11947,00	12019,00	12104,00	12091,00	12069
	En kötü	11274	11374	11374	11274	11274
	Std. sapma	4,20E+02	4,13E+02	3,78E+02	3,92E+02	3,84E+02
P3	En iyi	13251	13251	13251	13251	13251
	Ort.	13044,00	13051,00	13048,00	13049,00	13050
	En kötü	12646	13044	12675	12980	13044
	Std. sapma	5,02E+01	3,22E+01	4,85E+01	3,08E+01	3,19E+01
P4	En iyi	12348	12274	12479	12348	12479
	Ort.	12107	12106	12094	12110	12107
	En kötü	11914	11729	11858	11969	11969
	Std. sapma	9,37E+01	8,72E+01	8,80E+01	9,05E+01	8,38E+01
P4	En iyi	14044	14044	14044	14044	14044
	Ort.	13859,00	13852,00	13852,00	13863,00	13859
	En kötü	13664	13659	13659	13659	13659
	Std. sapma	7,05E+01	1,06E+02	9,55E+01	8,30E+01	7,88E+01
P4	En iyi	13508	13508	13508	13508	13508
	Ort.	13343	13347	13339	13363	13354
	En kötü	12809	12919	12770	12713	12668
	Std. sapma	1,77E+02	1,66E+02	1,86E+02	1,70E+02	1,87E+02
Friedman sıralama		3,3333	3,25	4	2,0833	2,3333
Final sıralama		4	3	5	1	2

Tablo A6. Mutasyon Oranı 0.11 için Elde Edilen Sonuçlar

Problem	Sonuçlar	Etiket Oranı 0.05	Etiket Oranı 0.10	Etiket Oranı 0.15	Etiket Oranı 0.20	Etiket Oranı 0.25
P1	En iyi	12020	11912	11851	12045	11912
	Ort.	11391,00	11375,00	11366,00	11376,00	11397,00
	En kötü	11033	11049	10989	11066	11060
	Std. sapma	1,55E+02	1,46E+02	1,39E+02	1,60E+02	1,38E+02
P2	En iyi	12369	12369	12369	12369	12369
	Ort.	12080,00	12099,00	12002,00	12084,00	12014
	En kötü	11374	11374	11374	11274	11274
	Std. sapma	3,76E+02	3,60E+02	4,09E+02	3,87E+02	4,12E+02
P3	En iyi	13283	13283	13283	13167	13251
	Ort.	13051,00	13045,00	13054,00	13045,00	13049
	En kötü	12888	12675	13026	13019	13019
	Std. sapma	4,15E+01	6,65E+01	4,45E+01	1,27E+01	2,37E+01
P4	En iyi	12274	12479	12348	12479	12274
	Ort.	12110	12111	12099	12108	12079
	En kötü	11969	11969	11969	11969	11969
	Std. sapma	8,93E+01	1,01E+02	9,02E+01	9,92E+01	8,83E+01
P4	En iyi	14044	14044	14044	14044	14044
	Ort.	13834,00	13843,00	13844,00	13842,00	13862
	En kötü	13474	13618	13630	13664	13659
	Std. sapma	9,99E+01	7,96E+01	9,44E+01	7,85E+01	1,07E+02
P4	En iyi	13508	13508	13508	13508	13508
	Ort.	13308	13348	13358	13372	13343
	En kötü	12672	12809	12809	12885	12539
	Std. sapma	2,14E+02	1,67E+02	1,75E+02	1,60E+02	1,93E+02
Friedman sıralama		3,1667	2,75	3,1667	2,9167	3
Final sıralama		4	1	4	2	3

Tablo A7. Mutasyon Oranı 0.12 için Elde Edilen Sonuçlar

Problem	Sonuçlar	Etiket Oranı 0.05	Etiket Oranı 0.10	Etiket Oranı 0.15	Etiket Oranı 0.20	Etiket Oranı 0.25
P1	En iyi	11951	11685	11855	11710	11760
	Ort.	11379,00	11395,00	11364,00	11367,00	11352,00
	En kötü	11069	10993	11101	10923	10897
	Std. sapma	1,49E+02	1,14E+02	1,26E+02	1,30E+02	1,23E+02
P2	En iyi	12369	12369	12369	12369	12369
	Ort.	12116,00	11995,00	12044,00	12052,00	12030
	En kötü	11374	11274	11274	11374	11274
	Std. sapma	3,55E+02	4,27E+02	3,95E+02	3,90E+02	4,15E+02
P3	En iyi	13167	13251	13251	13167	13283
	Ort.	13039,00	13052,00	13051,00	13045,00	13045
	En kötü	12675	12675	13044	12980	12736
	Std. sapma	5,15E+01	5,80E+01	2,82E+01	1,47E+01	5,47E+01
P4	En iyi	12348	12479	12274	12274	12274
	Ort.	12086	12098	12111	12100	12110
	En kötü	11969	11969	11899	11954	11969
	Std. sapma	8,70E+01	9,49E+01	8,76E+01	8,97E+01	7,95E+01
P4	En iyi	14044	14044	14044	14044	14044
	Ort.	13845,00	13852,00	13836,00	13836,00	13840
	En kötü	13659	13659	13659	13652	13659
	Std. sapma	9,24E+01	8,44E+01	8,71E+01	8,38E+01	8,23E+01
P4	En iyi	13508	13508	13508	13508	13508
	Ort.	13332	13375	13327	13345	13357
	En kötü	12770	12668	12751	12480	12860
	Std. sapma	1,87E+02	1,68E+02	1,90E+02	1,89E+02	1,62E+02
Friedman sıralama		3,1667	2,1667	3,25	3,1667	3,25
Final sıralama		2	1	3	2	3