



**T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**GÖÇ EDEN KUŞLAR OPTİMİZASYON ALGORİTMASI VE
AKILLI SU DAMLALARI OPTİMİZASYON ALGORİTMASI
VERİMLİLİK ANALİZİ**

NURDANUR PEHLİVAN

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN
PROF. DR. RESUL KARA**

DÜZCE, 2020

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖÇ EDEN KUŞLAR OPTİMİZASYON ALGORİTMASI VE
AKILLI SU DAMLALARI OPTİMİZASYON ALGORİTMASI
VERİMLİLİK ANALİZİ

Nurdanur PEHLİVAN tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Resul KARA

Düzce Üniversitesi

Jüri Üyeleri

Prof. Dr. Resul KARA

Düzce Üniversitesi

Dr. Öğr. Üyesi Zehra KARAPINAR ŞENTÜRK

Düzce Üniversitesi

Dr. Öğr. Üyesi Şafak KAYIKÇI

Bolu Abant İzzet Baysal Üniversitesi

Tez Savunma Tarihi: 22/07/2020

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

22 Temmuz 2020

Nurdanur PEHLİVAN

TEŐEKKÖR

Yüksek lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Prof. Dr. Resul KARA'ya en içten dileklerle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

22 Temmuz 2020

Nurdanur PEHLİVAN



İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	viii
KISALTMALAR.....	ix
SİMGELER	x
ÖZET	xi
ABSTRACT	xii
1. GİRİŞ.....	1
1.1. TEZİN KAPSAMI	1
1.2. LİTERATÜR TARAMASI	1
1.3. TEZİN ORGANİZASYONU	5
2. MATERYAL VE YÖNTEM	6
2.1. OPTİMİZASYON.....	6
2.1.1. Optimizasyon Modelinin Oluşturulması.....	7
2.1.1.1. Doğrusal Programlama Problemi	8
2.1.1.2. Doğrusal Olmayan Programlama Problemi.....	8
2.1.1.3. Tamsayı-Karışık Doğrusal Olmayan Programlama Problemi	8
2.1.2. Optimizasyon Algoritmalarının Sınıflandırılması	8
2.1.2.1. Sezgisel Optimizasyon Algoritmaları	8
2.1.2.2. Sezgisel Olmayan Optimizasyon Algoritmaları	9
2.2. GÖÇ EDEN KUŞLAR OPTİMİZASYON ALGORİTMASI (MIGRATING BIRDS OPTIMIZATION ALGORITHM – MBO)	9
2.2.1. MBO Algoritmasının Sözde Kodu	10
2.3. AKILLI SU DAMLALARI OPTİMİZASYON ALGORİTMASI (INTELLIGENT WATER DROPS OPTIMIZATION ALGORITHM – IWD)..	12
2.3.1. IWD Algoritmasının Akış Diyagramı.....	13
2.4. SIRT ÇANTASI PROBLEMİ (KNAPSACK PROBLEM).....	15
2.4.1. MBO Algoritmasının Sirt Çantası Problemine Uygulanması.....	16
2.4.1.1. Başlangıç Popülasyonu.....	16
2.4.1.2. Komşuluk ve Paylaşım	16
2.4.2. IWD Algoritmasının Sirt Çantası Problemine Uygulanması.....	16
3. GÖÇ EDEN KUŞLAR VE AKILLI SU DAMLALARI OPTİMİZASYON ALGORİTMALARININ BİRLİKTE ÇÖZÜMÜ.	18
3.1. MBO-IWD ALGORİTMASININ SIRT ÇANTASI PROBLEMİNE UYGULANMASI	18
3.1.1. Başlangıç Popülasyonu	18
3.1.2. Komşu Çözümleri Üretme.....	18
3.1.3. Kanat Çırpma ve Lider Kuşun Değişimi	19
3.1.4. MBO-IWD Algoritması Akış Diyagramı	20
3.1.4.1. MBO-IWD Algoritması Adım1 Sözde Kodu ve Açıklaması	22
3.1.4.2. MBO-IWD Algoritması Adım2 Sözde Kodu ve Açıklaması	22
3.1.4.3. MBO-IWD Algoritması Adım3 Sözde Kodu ve Açıklaması	23

4. BULGULAR.....	25
5. SONUÇ	28
6. KAYNAKLAR.....	29
ÖZGEÇMİŞ.....	31



ŞEKİL LİSTESİ

	<u>Sayfa No</u>
Şekil 2.1. Bir su damlasının soldan sağa gidişi.....	13
Şekil 2.2. Boyutları aynı hızları farklı olan iki su damlalarının soldan sağa gidişi.	13
Şekil 2.3. IWD algoritması akış diyagramı.....	14
Şekil 2.4. IWD algoritması adım 5 alt adımları.....	15
Şekil 3.1. Komşu çözüm üretimi ve komşu çözüm paylaşımı.....	19
Şekil 3.2. Lider kuşun ve diğer kuşların değişimleri.	20
Şekil 3.3. MBO-IWD algoritması akış diyagramı.	21

ÇİZELGE LİSTESİ

Sayfa No

Çizelge 4.1. Sırt çantası problemleri için MBO-IWD algoritması parametreleri.	25
Çizelge 4.2. On adet problemin boyut, ağırlık, kapasite ve fiyat değerleri [3].....	25
Çizelge 4.3. Sırt çantası problemleri için en iyi değer, MBO sonucu ve MBO-IWD sonucu.....	27



KISALTMALAR

ASA	Altın sinüs algoritması
CPU	Central processing unit
GA	Genetik algoritma
IWD	Akıllı su damlaları optimizasyonu
İYA	İteratif yerel arama
KKA	Karınca kolonisi algoritması
MBO	Göç eden kuşlar optimizasyonu
MBO-IWD	Göç eden kuşlar ve akıllı su damlaları optimizasyon algoritmalarının birlikte çözümü
MKP	Çoklu sırt çantası problemi
PSO	Parçacık sürü optimizasyonu
SKA	Sinüs-kosinüs algoritması
TB	Tavlama benzetimi
TSP	Gezgin satıcı problemi
TT	Tepe tırmanma
YAK	Yapay arı kolonisi

SİMGELER

GB	Giga Byte
Hz	Hertz
sn	Saniye



ÖZET

GÖÇ EDEN KUŞLAR OPTİMİZASYON ALGORİTMASI VE AKILLI SU DAMLALARI OPTİMİZASYON ALGORİTMASI VERİMLİLİK ANALİZİ

Nurdanur PEHLİVAN

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Danışman: Prof. Dr. Resul KARA

Temmuz 2020, 30 sayfa

Literatürde birçok arama ve optimizasyon algoritması bulunmaktadır. Ancak tüm arama ve optimizasyon problemleri için en iyi sonucu veren algoritma henüz tasarlanmadı. Bu çalışmada doğadan esinlenerek geliştirilmiş iki algoritma olan Göç Eden Kuşlar Optimizasyon Algoritması ve Akıllı Su Damlaları Optimizasyon Algoritması kullanılarak verimlilik analizi yapılmıştır. Göç Eden Kuşlar Optimizasyon Algoritması, tabiattaki göçmen kuş sürülerinin davranışlarından esinlenerek önerilmiş bir metasezgisel algoritmadır. Bazı optimizasyon probleminin çözümünde en iyi sonucu verdiği yapılan çalışmalar ile kanıtlanmıştır. Göç Eden Kuşlar incelendiğinde “V” şeklinde hareket ettikleri tespit edilmiştir. Kuşların bu hareketi onlara kaynaktan hedefe doğru giden yolda daha az enerji harcamaları ve hedefe daha kısa sürede varmalarını sağlamaktadır. Akıllı Su Damlaları Optimizasyon Algoritması doğal bir nehirdeki su damlalarının kaynaktan hedefe giderlerken her zaman daha kısa bir yol bulduğunu gözlemlenerek gerçekleştirilmiş bir algoritmadır. Nehirler, denizler gibi doğal su kaynakları incelendiğinde su damlalarının kaynaktan hedefe giden yolda her zaman daha kolay yolu (toprak miktarı diğer yollardan az olan bir yol) seçtikleri görülmüştür. Su damlaları sıfır olmayan bir hıza sahiptirler ve üzerlerinde toprak taşırlar. Literatürde metasezgisel algoritmalar ve metasezgisel olmayan algoritmalar bulunmaktadır. Bu çalışmada metasezgisel algoritmalar olan Göç Eden Kuşlar Optimizasyon Algoritmasındaki adımlarda rastgele seçimli olan adımları, rastgele olarak değil, literatürdeki metasezgisel algoritmalar olan Akıllı Su Damlaları Optimizasyon Algoritması ile belirleyip problem çözme verimliliğini iyileştirmek hedeflenmiştir. Bu amacın gerçekleştiğini göstermek için örnek problemler üzerinde çözümler uygulanıp karşılaştırmalı olarak sunulmuştur.

Anahtar sözcükler: MBO, IWD, Göç eden kuşlar optimizasyonu, Akıllı su damlaları optimizasyonu.

ABSTRACT

MIGRATING BIRDS OPTIMIZATION ALGORITHM AND INTELLIGENT WATER DROPS OPTIMAZTION ALGORITHM EFFICIENCY ANALYSIS

Nurdanur PEHLİVAN

Duzce University

Graduate School of Natural and Applied Sciences, Department of Computer
Engineering

Master's Thesis

Supervisor: Prof. Dr. Resul KARA

July 2020, 30 pages

There are many search and optimization algorithms in the literature. However, the algorithm that gives the best results for all search and optimization problems have not been designed yet. In this study, efficiency analysis was performed by using Migrating Birds Optimization Algorithm and Intelligent Water Drops Optimization Algorithm, which are two algorithms developed inspired by nature. The Migrating Birds Optimization Algorithm is a proposed metaheuristic algorithm inspired by the behavior of migratory bird flocks in nature. It has been proven with the studies that it gives the best result in solving some optimization problem. When migrating birds were examined, it was found that they acted as “V”. This movement of birds enables them to spend less energy on the road from source to the destination and to reach the destination in a shorter time. Intelligent Water Drops Optimization Algorithm is an algorithm created by observing that water drops in a natural river always find a shorter path from the source to the destination. When natural water sources such as rivers and seas were examined, it was seen that water drops always chose the easier way (a road with less than other roads) on the road from the source to the destination. Water drops have a non-zero speed and carry soil on them. In the literature, there are metaheuristic and non-metaheuristic algorithms. In this study, it was aimed to improve the problem solving efficiency by determining the randomly selected steps in the steps in the Migrating Birds Optimization Algorithm, which is one of the meteheuristic algorithms in the literature. In order to show that this goal has been achieved on sample problems and presented comparatively.

Keywords: MBO, IWD, Migrating birds optimization, Intelligent water drops optimization.

1. GİRİŞ

1.1. TEZİN KAPSAMI

Literatürde bulunan optimizasyon algoritmaları bütün optimizasyon problemlerinde en iyi şekilde sonuç alamamaktadır. Metasezgisel algoritmalar ile metasezgisel olmayan algoritmalar karşılaştırıldığında birçok optimizasyon probleminde metasezgisel algoritmalar daha iyi sonuçlar vermektedir. Bu tezde metasezgisel algoritmalar olan Göç Eden Kuşlar Optimizasyon Algoritması (MBO) ve Akıllı Su Damlaları Optimizasyon Algoritması (IWD) kullanılarak verimlilik analizi yapılacaktır.

1.2. LİTERATÜR TARAMASI

Duman ve arkadaşları tarafından 2012 yılında çözüm kümesi ayrık olan optimizasyon problemlerinin çözümü için MBO Algoritmasını önerip algoritmayı tanıtmışlardır. MBO, göç eden kuşların göç ederken sergiledikleri “V” dizilimindeki uçuş düzenlerinden esinlenmiş bir optimizasyon algoritmasıdır. Ayrık problemleri çözmek için tasarlanmış bu algoritmanın performansı karesel atama problemleri üzerinde test edilmiştir. Duman ve arkadaşları literatüre kazandırdıkları MBO algoritmasının genetik algoritma, parçacık sürü optimizasyonu, tabu arama, diferansiyel evrim algoritması, dağıtık arama ve rehberli evrimsel benzetimli tavlama algoritmalarından daha iyi sonuçlar elde ettiğini ispatlamışlardır [1].

Shah-Hosseini parçacık tabanlı bir metasezgisel algoritma olan IWD optimizasyon algoritmasını n-queen bulmacası ve Gezgin Satıcı Problemi (TSP) üzerinde denemiş ve deney sonuçlarını paylaşmıştır. Bu iki problemde optimal sonuçları elde edip bir optimizasyon problemi olan Çoklu Sırt Çantası Probleminde (MKP) de test etmişler ve en iyi sonucu elde etmişlerdir. Doğal bir nehirdeki su damlaları kaynaktan hedefe doğru giden birçok yol arasında en iyi yolu bulurlar. Bu en iyi yol, su damlaları ve su damlalarının nehir yatakları arasında meydana gelen eylemler ve reaksiyonlarla elde edilir. IWD optimizasyon algoritması, nehirlerde akan doğal su damlalarını gözlemlemekten esinlenen bir sürü tabanlı optimizasyon algoritmasıdır [2].

Ulker ve arkadaşları Sırt Çantası Probleminin çözümü için bir meta-sezgisel algoritma olan MBO algoritmasını önermişlerdir. NP-zor problemlerden olan Sırt Çantası Problemi, kombinatorial bir optimizasyon problemidir. Bu problemin amacı, seçilen eşya ile sırt çantasının kapasitesini aşmadan maksimum fayda sağlamaktır. MBO algoritması ayrık problemler için tasarlanmış bir algoritmadır. Bu makalede, MBO algoritmasının performansı bazı sırt çantası problemleri üzerinde test edilmiştir ve bu testlerin sonuçları ayrıntılı olarak verilmiştir [3].

Zou ve arkadaşları 0-1 sırt çantası problemini açıklamışlardır ve 0-1 sırt çantası probleminin çözümü için yeni bir global harmoni arama algoritmasını önermişlerdir. Önerilen algoritmanın pozisyon güncelleme ve genetik mutasyon olmak üzere iki önemli işlemi vardır. Pozisyon güncelleme, uyum belleğinin en kötü uyumunun her yinelemede küresel en iyi uyuma hızla geçmesini sağlar. Genetik mutasyon ise önerilen algoritmanın yerel optimumluğa sıkışmasını önlemek içindir. Birçok büyük ölçekli örnekler üzerinde yapılan hesaplama deneyleri sonucunda önerilen algoritmanın 0-1 sırt çantası problemlerini çözmek için etkili bir alternatif olabileceğini göstermektedir [4].

Elshehbiny ve arkadaşları IWD optimizasyon algoritmasını bulut bilişim ortamında denemişler ve iyi sonuçlar almışlardır. Bulut bilişim, büyük ölçekli, heterojen otonom sistem koleksiyonu ve esnek hesaplama mimarisi ile yüksek performanslı bir bilişim ortamı olarak ortaya çıkmaktadır. Birçok kaynak yönetimi yöntemi, tüm bulut bilişim sisteminin verimliliğini artırabilir. Bulut bilişim kaynak yönetiminin anahtar kısmı kaynak planlamasıdır. Bulut sanal makinelerinde görevlerin optimize edilmesi NP-zor bir problemidir ve bu problemin çözümü için birçok farklı algoritma bulunmaktadır. Bulut bilişimdeki iş akışlarının zamanlamasını optimize etmek için doğa tabanlı algoritma olan IWD optimizasyon algoritması bu makalede önerilmiştir. IWD algoritması, iş akışları simülasyon araç seti üzerinde uygulanmış, farklı maliyet modelleri üzerinde simüle edilip bulut ortamında test edilmiştir. IWD algoritması, klasik iş akışı zamanlama algoritmaları üzerinde büyük gelişmeler göstermiştir. Aynı zamanda önerilen algoritma olan IWD ile Round Robin, Parçacık Sürü Optimizasyonu (PSO) ve diğer iyi bilinen birçok zamanlama algoritmaları ile bir karşılaştırma yapılmıştır [5].

Kesavamoorthy, IWD optimizasyon algoritmasını TSP'ye uygulamış ve bu algoritma ile daha iyi sonuçlar almıştır. Bu algoritma nehir sistemlerinin dinamiğine ve su damlalarının nehirlerde gerçekleştirdiği eylemlere dayanmaktadır. Doğal su damlalarının davranışına ilişkin gözlemlere dayanarak geliştirilen bu algoritma TSP'yi çözmek için önerilmiştir.

TSP n adet şehir arasındaki mesafelerin bilindiği durumda, şehirlerin her birine yalnız bir kez uğramak şartıyla, başlangıç noktasına geri dönülmesi esasına dayalı, tur boyunca kat edilen toplam yolun en kısa olduğu şehir sıralamasının bulunmasının amaçlandığı bir optimizasyon problemidir. Bu makalede orijinal IWD algoritmasına kıyasla bir düğümden diğerine taşınan damlacığın istenmeyen düğümü ölçmek için kullanabileceği bir fonksiyon eklenmiştir [6].

Kuzu, NP-zor problem sınıfından olan TSP'yi metasezgisel optimizasyon algoritmaları ile çözmüştür. Klasik matematiksel yöntemlerle çözümü zor hatta belli bir büyüklükten sonra neredeyse imkansız olan problemlerin çözümü için metasezgisel yöntemler etkin bir şekilde kullanılmaktadır. TSP'de düğüm sayısı arttıkça problemin çözümünde harcanan zaman üstel olarak artmaktadır. Makalede, Tepe Tırmanma Algoritması (TT), İteratif Yerel Arama Algoritması (İYA), Tavlama Benzetimi Algoritması (TB), Tabu Arama Algoritması (TA), Kanguru Algoritması (KA), Yapay Arı Kolonisi Algoritması (YAK), Genetik Algoritma (GA) ve Karınca Kolonisi Algoritması (KKA) olmak üzere sekiz metasezgisel algoritma TSP üzerinde uygulanmış, sonuçlar açıklanmış ve yorumlanmıştır. Birden fazla noktada arama yapan metasezgisel algoritmaların, tek noktada arama yapan metasezgisel algoritmalarla göre daha iyi sonuç verdiği görülmüştür [7].

Murty, karar verme mekanizmalarında optimizasyon modellerinin nasıl kullanılabileceğini incelemiştir. Optimizasyonun matematiksel modellemesi, hesaplama ve algoritmik becerilerine internet yayınlı kitapta yer verilmiştir [8].

Eröz ve arkadaşları bazı metasezgisel algoritmalarının karşılaştırmasını yapmışlardır. Performans karşılaştırılması yaparken doğadan ve farklı bilim dallarından ilham alınarak gerçekleştirilen yedi farklı optimizasyon algoritması kullanmışlardır. Bu algoritmaların performanslarını karşılaştırmak için yirmi üç adet kısıtsız kıyaslama fonksiyonu ve iki adet kısıtlı mühendislik problemi kullanmışlardır. Yaptıkları testlerin sonucuna makalede yer vermişlerdir ve bu testler sonucunda matematik tabanlı olan Altın Sinüs Algoritması (ASA) ve Sinüs Kosinüs Algoritması (SKA) diğer algoritmalarla göre daha hızlı sonuç vermiştir ve daha iyi performans göstermiştir [9].

Mirjalili, optimizasyon problemlerinin çözümünde SKA kullanmıştır. SKA rastgele çoklu aday çözümleri yaratır ve sinüs kosinüs fonksiyonlarına dayanan bir matematiksel model kullanır. Makalede, SKA'nın performansı test edilmiştir. Yapılan testlerin sonuçlarına

makalede yer verilmiştir [10].

Eberthart ve arkadaşları PSO'nun gelişiminden, uygulamalarından ve kaynaklarından bahsetmişlerdir. Makalede, PSO ile ilgili gelişmelere ve uygulamalara yer verilmiştir. PSO, optimizasyon problemlerinin çoğunu ve optimizasyon problemlerine dönüştürülebilecek problemleri çözmek için kullanılabilir. En potansiyel uygulama alanları arasında sistem tasarımı, çok amaçlı optimizasyon, sınıflandırma, örüntü tanıma, biyolojik sistem modelleme, planlama, sinyal işleme, oyunlar, robotik uygulamalar, karar verme, simülasyon ve tanımlama bulunmaktadır [11].

Kennedy ve arkadaşları PSO'yu açıklamışlar ve literatüre katmışlardır. PSO algoritması, balık ve kuş sürülerinin yiyecek arayışı esnasında sergiledikleri davranışlardan esinlenerek literatüre kazandırılmış sürü temelli meta-sezgisel bir optimizasyon algoritmasıdır. Balık ve kuş sürülerindeki her bir birey sürüdeki diğer bireylerden etkilenecek yiyecek kaynağına doğru yön ve hız bilgisini güncelleyerek hareket eder. PSO algoritmasında da sürüdeki bireylerin bu davranışlarından esinlenilmiştir. PSO algoritması, popülasyon tabanlı bir meta-sezgisel algoritmadır. Popülasyondaki her bir birey bir çözümü temsil etmektedir [12].

Dorigo ve arkadaşları Karınca Kolonisi Optimizasyonundan bahsetmişlerdir. Sürü zekası, böceklerin ve diğer hayvanların sosyal davranışlarından ilham alan bir yaklaşımdır. Özellikle, karıncalar en çok çalışılan sürü zekası yaklaşımlarındandır. KKA, bazı karınca türlerinin yeme olan yönelim davranışlarından ilham alır. Karıncalar, koloninin diğer üyeleri tarafından izlenmesi gereken olumlu yolları işaretlemek için feromonu yere bırakırlar. KKA'da optimizasyon problemlerini çözmek için karıncaların bu davranışından ilham alır. Makalede, KKA tanıtılmış ve en önemli uygulamaları incelenerek açıklanmıştır [13].

Akçay ve arkadaşları ikili çok boyutlu sırt çantası problemini açgözlü algoritma ile çözmüşlerdir. Makalede kullandıkları Açgözlü Algoritması, orijinal Açgözlü Algoritmasından iki farklı açıdan farklıdır. İlk farklılık, orijinal algoritmada öğe seçimi kararları vermek için her bir öğenin toplam kaynak tüketimine güvenirken, makalede uygulanan algoritma, tüm sırt çantası bunun için kullanılacaksa kabul edilebilecek bir öğenin maksimum kopya sayısı olarak tanımlanan etkin kapasiteyi kullanır. İkinci farklılık ise orijinalde her karar değişkeninin değeri sadece bir birim artabilirken, makalede uyarlanan algoritmada karar değişkenlerini toplu olarak ekler ve sonuç olarak

büyük ölçekli problemler için hesaplama verimliliğini önemli ölçüde artırır. Makalede yeni algoritmanın mevcut olan algortimadan hesaplama açısından daha verimli olduğu ve daha optimal çözümler ürettiği görülebilir [14].

Alkaya ve arkadaşları MBO'nun komşuluk yapısını değiştirerek çok boyutlu sürekli arama uzayında MBO'nun performansını incelemişlerdir. Makalede, orijinal MBO'nun çok boyutlu sürekli alanlarda kullanılması için yeni bir komşu üreten fonksiyon geliştirmişlerdir. Geliştirdikleri algoritmayı bir dizi test etikten sonra deney sonuçlarını paylaşmışlardır [15].

1.3. TEZİN ORGANİZASYONU

Tezin organizasyonu şu şekildedir: giriş başlığında tezin kapsamı, literatür taraması ve tezin organizasyonuna yer verilmiştir. Tezin kapsamı başlığında tezin içeriğinden bahsedilmiştir. Literatür taraması başlığında MBO, IWD ve Sırt Çantası Problemi ile ilgili yapılmış çalışmalar anlatılmış, tezin organizasyonu başlığında hangi başlık altında hangi bilgilere yer verildiğinden bahsedilmiştir. Materyal ve yöntem başlığı altında optimizasyonun ne olduğundan bahsedilmiş, optimizasyon modelinin nasıl oluşturulacağına değinilmiş, optimizasyon algoritmalarının sezgisel ve sezgisel olmayan optimizasyon algoritmaları olarak sınıflandırma yapılmış, orijinal MBO algoritması sözde kodu ile birlikte anlatılmış, orijinal IWD algoritması akış diyagramı verilerek anlatılmış, sırt çantası probleminden bahsedilmiş ve MBO ve IWD algoritmalarının ayrı ayrı sırt çantası problemine nasıl uygulandıkları gösterilmiştir. Göç eden kuşlar ve akıllı su damlaları optimizasyon algoritmalarının birlikte çözümü başlığı altında geliştirilmiş olan MBO-IWD algoritmasının sırt çantası problemine uygulanışı MBO-IWD algoritmasının akış diyagramı verilerek anlatılmıştır. Bulgular başlığında MBO-IWD algoritmasının parametreleri, deney yapılan sırt çantası problemlerinin boyut, ağırlık, kapasite ve fiyatları ve deney yapılan problemlerin literatüre geçmiş en iyi değerleri, MBO algoritması ile çözülerek elde edilen sonuçları ve MBO-IWD algoritması ile elde edilen sonuçlarının karşılaştırılmasına yer verilmiştir. Son olarak sonuç başlığında elde edilen sonuçların bir değerlendirmesi yapılmış ve ilerideki çalışmalara ışık tutacak önerilerde bulunulmuştur.

2. MATERYAL VE YÖNTEM

Tez çalışması kapsamında, MBO algoritmasının, IWD optimizasyon algoritması ile iyileştirme çalışmaları yapılmıştır. Yapılan bu çalışmalarda MBO algoritmasının temel yapısındaki rastgele veri sağlanan yerlerde IWD algoritması kullanılmış ve yeni bir algoritma MBO-IWD geliştirilmiştir.

Geliştirilen bu yeni algoritmanın başarısını görebilmek için orijinal MBO ile MBO-IWD algoritmaları literatürde bulunan bazı optimizasyon problemlerine uygulanmıştır. Problemlerin çözümünden elde edilen sonuçlara göre bu iki algoritmanın performans karşılaştırması yapılmıştır.

2.1. OPTİMİZASYON

Optimizasyon, bir sistemde var olan işgücü, zaman, süreçler, hammaddeler, kapasite ve ekipman gibi kaynakların en verimli şekilde kullanılarak belirli amaçlara ulaşılması işidir. Aranılan sonuç veya en iyi çözüm, ele alınan probleme göre minimum veya maksimum değer olabilir.

Optimizasyon problemi ise, problemin kısıtlarını sağlayacak şekilde, bilinmeyen parametre değerlerinin bulunmasını içeren herhangi bir problem olarak tanımlanabilmektedir [8]. Optimizasyon problemleri ile mühendislik, endüstri ve daha birçok alanda karşılaşılabilir.

Optimizasyonda modelleme ve çözümleme önemli bileşenlerdir. Modelleme, problemin matematiksel olarak ifade edilmesi iken çözümleme, modellemeyi sağlayan en iyi sonucun elde edilmesi işlemidir.

Optimizasyon teknolojisinin gelişme sürecinde araştırmacılar ilk olarak modelleme ile ilgilenmişlerdir. Bu alandaki ilk çalışmalar Amerika Birleşik Devletleri'nin dış ticaretini ve ekonomik yapısını modellemek için Leontief tarafından yapılan yayınlardır [16].

Optimizasyon problemlerinin çözümüne yönelik ilk önemli çalışma Dantzig tarafından yapılmış ve simpleks algoritması geliştirilmiştir [17].

Optimizasyonun performansını etkileyen değişkenlere karar değişkenleri denir. Karar

değişkenlerinin amaç üzerindeki etkilerinin analitik olarak gösterilmesiyle amaç fonksiyonu oluşturulur. Optimizasyon algoritmalarının çoğu, amaç fonksiyonu ve sistemin modeli için matematiksel modellere ihtiyaç duymaktadır [9].

2.1.1. Optimizasyon Modelinin Oluşturulması

Modeller, mühendislikte yoğun olarak kullanılan büyük kapsamlı sistemlerin tüm özelliklerini yansıtacak daha küçük boyutlardaki yapılardır. Modellerde genelde, sistemin temel özelliklerini yansıtacak ve modelin kullanım amaçlarını gerçekçi olarak içerecek detaylar bulunmaktadır.

Optimizasyon modelleri ise sistemin işleyişini ve özelliklerini yansıtan, sistemin içindeki ve çevresindeki diğer sistemlerle olan etkileşimlerini kapsayan matematiksel ifadelerden oluşmaktadır [18].

$$\begin{aligned} \max z &= f(x,y) \\ \text{k.s. } g(x,y) &= 0 \\ h(x,y) &\leq 0 \\ x &\in R^n \\ y &\in \{0,1,2,\dots,m\} \end{aligned} \tag{2.1}$$

Yukarıdaki optimizasyon probleminde sistemin performans ölçütü yani amaç fonksiyonu $z = f(x,y)$ ile ifade edilmiş ve karar değişkenleri x ve y 'nin bu ölçütü en çoklayacak değerlerinin bulunması hedeflenmektedir. Sistemin özellikleri ise $g(x,y)$ eşitliği ve $h(x,y)$ eşitsizlikleri yani kısıtlar ile belirlenmektedir. Ayrıca, karar değişkenleri iki türlü ifade edilmiştir: n boyutlu uzayda herhangi bir reel değeri alabilen sürekli değişkenler (x) ve herhangi bir tamsayı değeri alabilen tam sayılı değişkenler (y).

Optimizasyon modellerini içerdikleri karar değişkenlerinin, amaç fonksiyonunun ve sistem kısıtlarının özelliklerine göre sistem parametrelerinin bilinen sabit değerlere aldığı durumlarda aşağıdaki gibi sınıflandırılmaktadır [19]. Eğer bir optimizasyon probleminde y değişkenleri yer almıyorsa ve $f(x)$, $g(x)$ ve $h(x)$ fonksiyonları doğrusalsa o problem bir doğrusal programlama problemi olarak tanımlanır. Bir optimizasyon probleminde y değişkenleri yer almıyorsa ve $f(x)$, $g(x)$ ve $h(x)$ fonksiyonlarından herhangi biri doğrusal değilse o problem bir doğrusal olmayan programlama problemidir.

Optimizasyon problemlerinde y değişkenleri yer alıyorsa ve $f(x,y)$, $g(x,y)$ ve $h(x,y)$ fonksiyonlarının doğrusal olması durumunda problem tamsayı karışık doğrusal programlama problemi, $f(x,y)$, $g(x,y)$ ve $h(x,y)$ fonksiyonlarından herhangi birisinin doğrusal olmaması durumunda ise tamsayı karışık doğrusal olmayan programlama problemi olarak tanımlanır.

2.1.1.1. Doğrusal Programlama Problemi

Bu problemde sadece sürekli değişkenler ve doğrusal amaç fonksiyonuyla, doğrusal kısıtlar mevcuttur. Doğrusal programlama en yaygın olarak kullanılan optimizasyon programlama çeşididir.

2.1.1.2. Doğrusal Olmayan Programlama Problemi

Bu problemde ya amaç fonksiyonu ya kısıtlar ya da hem amaç fonksiyonu hem de kısıtlar doğrusal değildir. Optimizasyon sonucunda amaç fonksiyonunu en çoklayacak karar değişkenleri, x , n boyutlu uzayda herhangi bir gerçek değeri alabilirler.

2.1.1.3. Tamsayı-Karışık Doğrusal Olmayan Programlama Problemi

Bu problemler amaç fonksiyonunda, kısıtlarda veya hem amaç fonksiyonunda hem de kısıtlarda doğrusal olmayan ifadeleri ve tamsayılı değişkenleri kapsamaktadır.

2.1.2. Optimizasyon Algoritmalarının Sınıflandırılması

Optimizasyon algoritmaları temelde sezgisel ve sezgisel olmayan optimizasyon algoritmaları olmak üzere ikiye ayrılmaktadır.

2.1.2.1. Sezgisel Optimizasyon Algoritmaları

Sezgisel optimizasyon algoritmaları yeterli sayıda rastgele çözüm ve optimizasyon adımları ile global optimum bulmayı amaçlamaktadır [20].

Metasezgisel yöntemler, sezgisel yöntemleri daha geliştirerek arama uzayını aktif ve verimli bir şekilde incelemeyi amaçlar. Bu yöntemler, her zaman global en iyi çözümü bulmayı garanti etmezler. Ancak oldukça kullanışlı bir yöntemdir. Çünkü büyük ölçekli ve karmaşık problemlere etkin çözümler üretebilirler [10]. Metasezgisel optimizasyon algoritmalarının başarımı klasik (sezgisel veya sezgisel olmayan) yöntemlere göre daha başarılı olmasına rağmen bütün problemlerde aynı başarıyı gösterememektedir [9]. Bu nedenle metasezgisel optimizasyon algoritmalarının geliştirilmesi gerekliliği ortaya çıkmaktadır.

Günümüzde metasezgisel olarak geliştirilen optimizasyon algoritmaları arasında MBO algoritması, IWD optimizasyon algoritması, GA [21], PSO [11],[12], KKA [13],[22], Diferansiyel Evrim (DE) [23], SKA [10] ve benzeri diğer algoritmalar öne çıkmaktadır. MBO Algoritması Bölüm 2.2.'de verilmiştir. IWD Optimizasyon Algoritması Bölüm 2.3.'de verilmiştir. Genetik Algoritma, doğal seçim ilkelerine dayanan bir arama ve optimizasyon yöntemidir. Genetik Algoritmaların, fonksiyon optimizasyonu, çizelgeleme, mekanik öğrenme, tasarım, hücresel üretim gibi alanlarda başarılı uygulamaları bulunmaktadır. GA, olasılık kurallarına göre çalışır, yalnızca amaç fonksiyonuna gereksinim duyar [21]. PSO, balık ve kuş sürülerinin yiyecek arayışı esnasında sergiledikleri davranışlardan esinlenerek literatüre kazandırılmış sürü temelli meta-sezgisel bir optimizasyon algoritmasıdır. Balık ve kuş sürülerindeki her bir birey sürüdeki diğer bireylerden etkilenerek yiyecek kaynağına doğru yön ve hız bilgisini güncelleyerek hareket eder. PSO algoritmasında da sürüdeki bireylerin bu davranışlarından esinlenilmiştir. PSO algoritması, popülasyon tabanlı bir meta-sezgisel algoritmadır. Popülasyondaki her bir birey bir çözümü temsil etmektedir [12]. KKA, bazı karınca türlerinin yeme olan yönelim davranışlarından ilham alır. Karıncalar, koloninin diğer üyeleri tarafından izlenmesi gereken olumlu yolları işaretlemek için feromonu yere bırakırlar. KKA da optimizasyon problemlerini çözmek için karıncaların bu davranışından ilham alır [13]. DE Algoritması ilk olarak 1995 yılında K. Price tarafından ortaya konmuştur. DE Algoritması çaprazlama, mutasyon ve seçim gibi genetik algoritmalarda bulunan benzer operatörleri kullanan ve popülasyon tabanlı bir algoritmadır [22]. Sinüs-Kosinüs Algoritması rastgele çoklu aday çözümleri yaratır ve sinüs kosinüs fonksiyonlarına dayanan bir matematiksel model kullanır [10].

2.1.2.2. Sezgisel Olmayan Optimizasyon Algoritmaları

Sezgisel olmayan optimizasyon algoritmaları, amaç fonksiyonunun eğimini veya türevini kullanarak global optimum bulmaya çalışır [24]. Sezgisel optimizasyon algoritmaları istenilen sonuca yaklaşırken sezgisel olmayan optimizasyon algoritmaları kesin sonuç verirler.

2.2. GÖÇ EDEN KUŞLAR OPTİMİZASYON ALGORİTMASI (MIGRATING BIRDS OPTIMIZATION ALGORITHM – MBO)

MBO, 2012 yılında Duman, Uysal ve Alkaya tarafından geliştirilmiştir [1]. MBO

algoritması, göç eden kuşların göç ederken sergiledikleri “V” diziliminden sağladıkları enerji tasarrufundan esinlenmiştir.

Göç eden kuşların havadayken daha uzun mesafeleri daha az enerji harcayarak uçabilmeleri için kullandıkları en yaygın şekil “V” şeklidir. “V” şeklinin temel amacı enerji tasarrufudur. “V” şekli uçuş düzenindeki lider kuş en çok enerji harcayan kuştur ve diğer kuşlar, kendilerinden önceki kuşların uçuş esnasındaki kanat çırpışından dolayı oluşturdukları girdaptan faydalanarak daha az enerji harcayarak daha uzun süre havada kalabilmektedirler.

“V” oluşumunda kuşlara karşılık gelen ilk çözümler ile başlar. İlk çözümden (bu ilk çözüm lider kuşa karşılık gelir) başlayarak ve kuyruklara doğru hatlarda ilerleyerek her bir çözüm, komşu çözümleri tarafından geliştirilmeye çalışılır. Bu çözümler, her adımda geliştirilmeye çalışılmaktadır. MBO, çözümleri geliştirmek için bir komşu yapısı kullanır. Ayrıca kullanılmayan komşu çözümlerin diğer çözümlerle paylaşıldığı bir fayda mekanizması vardır. Lider kuş ürettiği ve kullanmadığı komşu çözümleri sol alttaki ve sağ alttaki kuşlar ile paylaşır. Diğer kuşlar üretilen kullanmadıkları komşu çözümleri sadece kendi yönündeki kuşlar ile paylaşırlar.

Lider kuş, aşağıdaki kuşla değiştirilir. Bu işlem öncelikle sürünün sol tarafına uygulanır. Lider kuş, sürünün sol tarafının sonuna gider ve sol taraftaki ilk kuş yeni lider olarak atanır. Ardından lider kuş, sürünün sağ tarafının sonuna gider ve sağ taraftaki ilk kuş yeni lider olarak atanır. İşlem başa döner lider kuş sol taraf ile değiştirilir, bir sonraki değişim sağ taraf ile olur. Algoritma bir dizi yineleme geçtikten sonra sona erer.

MBO algoritmasında durma kriteri genelde problem boyutuna bağlı olmaktadır. Duman ve ark. algoritmanın çalışma süresini, toplam üretilen komşu çözüm sayısının problem boyutunun küpüne eşit olacak şekilde belirlemişlerdir [1].

2.2.1. MBO Algoritmasının Sözde Kodu

1. n adet başlangıç çözümünü rastgele üret ve bunları rastgele varsayımsal bir V formasyonuna yerleştir.
2. while sonlanmadı do
3. for tur = (1 to m) do
4. Lider için k komşu çözüm adayı üret

5. if bir komşu aday liderden daha iyi then
6. Lideri güncelle
7. end if
8. x adet lider komşusu adayı ilk takipçiye aktar
9. for sürüdeki her bir takipçi do
10. k-x adet komşu üret
11. x adet komşuyu bir önceki takipçiden al
12. if bir komşu daha iyi then
13. Takipçiyi güncelle
14. end if
15. end for
16. end for
17. Lideri sürü sonuna at. Yeni lider belirle.
18. end while

Bu sözde kodda;

n = başlangıç çözümlerinin (kuşlar) sayısı

k = üretilen komşu çözümlerin sayısı

x = bir sonraki çözümle paylaşılacak olan komşu çözümlerin sayısı

m = tur sayısı

olarak kullanılmaktadır.

Sözde koddan da görüldüğü gibi, MBO ilk olarak rastgele bir şekilde n başlangıç çözümü üretir. Her çözüm bir D -boyutlu vektördür. Burada D , optimizasyon parametrelerinin sayısıdır. Uygulanabilir çözümlerle başlatma çok zaman alan bir işlem olduğundan ve bazı durumlarda rastgele uygulanabilir bir çözüm üretmek imkansız olduğundan, MBO algoritması başlangıç popülasyonunun uygulanabilir olduğunu düşünmez. Başlama aşamasında, Denklem (2.2)'ye göre çözümlerin parametreleri için parametrelerin alt ve üst sınırları arasındaki rastgele değerler atanır:

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad (2.2)$$

Burada x_{ij} , j. boyuttaki i. çözümün konumudur, x_j^{max} ve x_j^{min} j. boyut için sınır değerlerdir ve rand, 0 ile 1 arasında değişen tek tip rastgele bir sayıdır.

2.3. AKILLI SU DAMLALARI OPTİMİZASYON ALGORİTMASI (INTELLIGENT WATER DROPS OPTIMIZATION ALGORITHM – IWD)

IWD, 2007 yılında Hamed Shah Hosseini tarafından önerilmiş bir algoritmadır [2]. IWD'nin temeli doğal nehir sistemlerinde meydana gelen süreçlere ve nehirdeki su damlalarının bir noktadan diğer bir noktaya giderken gerçekleşen eylem ve reaksiyonlara dayanır.

Doğada genellikle nehirlerde, göllerde ve denizlerde su damlalarının hareket ettiğini görürüz. Su damlaları hareket ettikçe, içinde aktıkları ortamlarını değiştirirler. Ayrıca, çevrenin su damlalarının izlediği yollar üzerinde önemli etkileri vardır.

Bir su damlasının bir özelliği, su damlasının hızıdır, bu da su damlasının bir miktar toprağı bir yerden öndeki başka bir yere aktarmasını sağlar. Bu toprak genellikle yolun hızlı kısımlarından yavaş parçalara aktarılır. Hızlı parçalar topraktan uzaklaştırılarak derinleştikçe, daha fazla su tutabilirler ve böylece daha fazla su çekebilirler. Su damlalarında taşınan çıkarılan topraklar nehrin yavaş yataklarına boşaltılır [2].

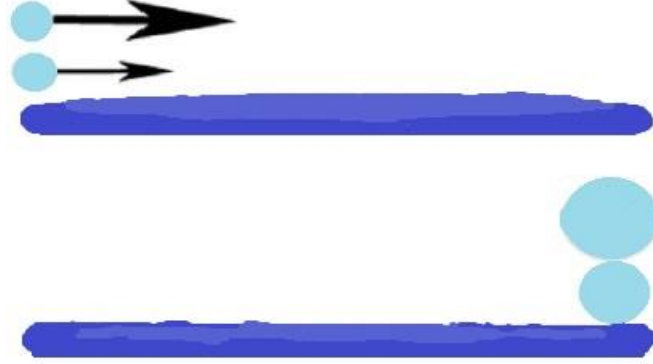
Kısaca, bir nehirdeki bir su damlası sıfır olmayan bir hıza sahiptir ve genellikle bir miktar toprak taşır. Nehir yatağının hızlı akan bölgelerinden bir miktar toprak yükleyebilir ve bunları nehir yatağının daha yavaş bölgelerine boşaltabilir. Bir su damlası kaynaktan hedefe giden yolda birkaç dal arasında seçim yapmak zorunda kaldığında daha zor bir yola (toprak miktarı diğerlerinden çok olan yol) daha kolay bir yolu (toprak miktarı diğerlerinden az olan bir yol) tercih eder.

Şekil 2.1'de bir su damlasının doğal bir su kaynağında soldan sağa ya da başka bir deyişle kaynaktan hedefe gidişi verilmiştir. Su damlası doğal bir su kaynağında kaynaktan hedefe giderken su toplar ve büyür.



Şekil 2.1. Bir su damlasının soldan sağa gidişi.

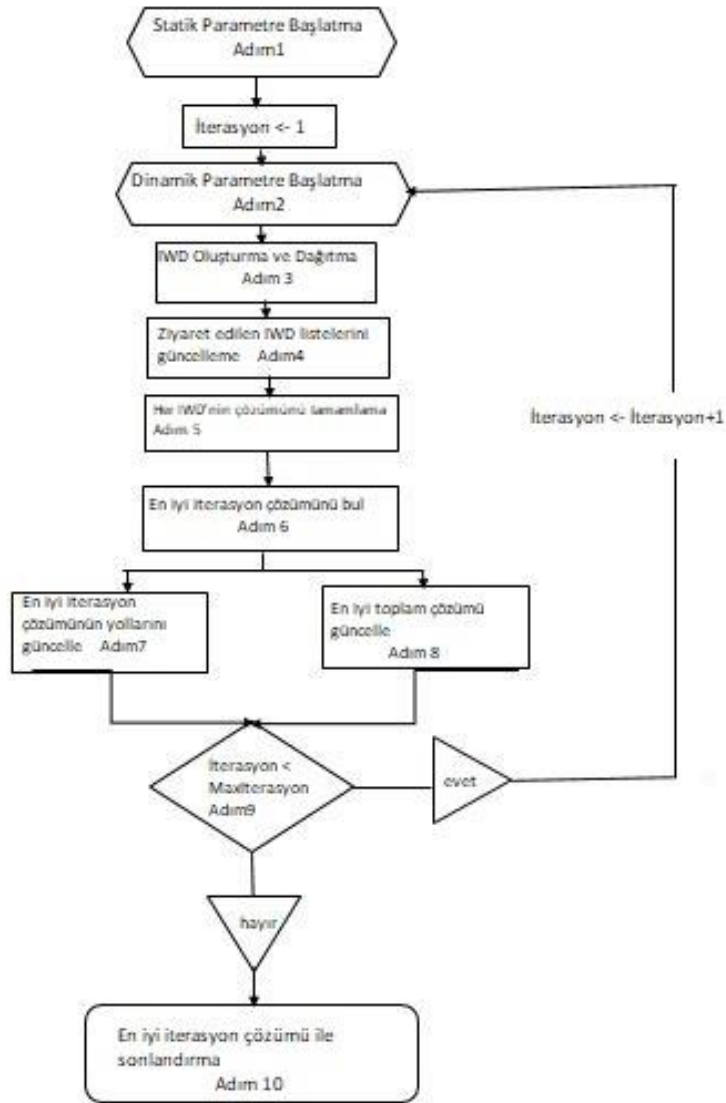
Şekil 2.2’de hızları farklı boyutları aynı olan iki su damlacığının soldan sağa ya da kaynaktan hedefe doğru gidişi verilmiştir. Su damlalarının boyutları aynı olsa bile hızı daha fazla olan su damlası daha fazla su toplar ve hedefe ulaştıklarında hızı büyük olan su damlasının boyutu daha büyük olur.



Şekil 2.2. Boyutları aynı hızları farklı olan iki su damlalarının soldan sağa gidişi.

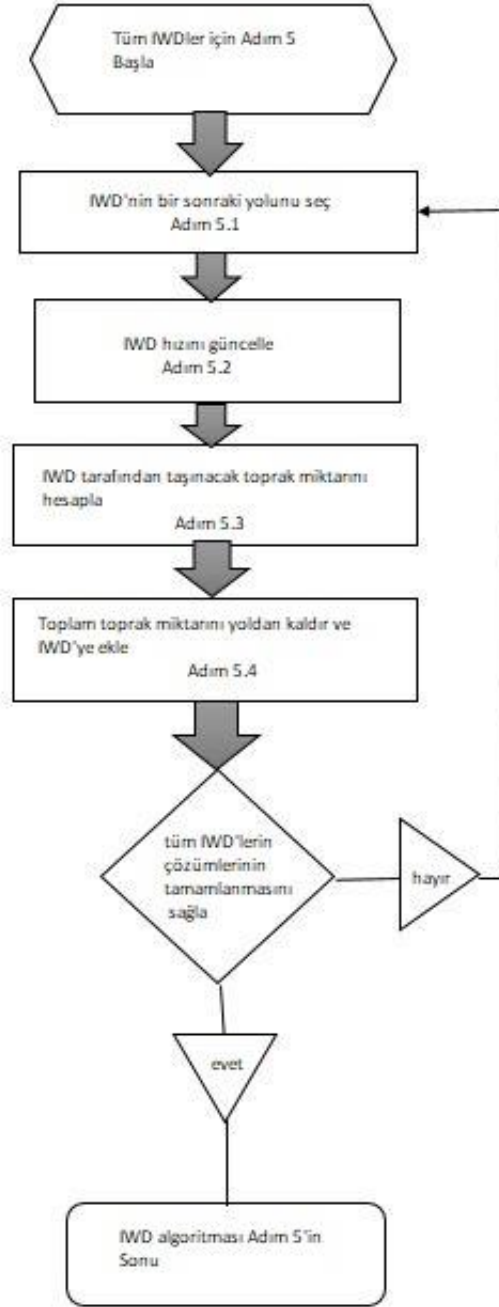
2.3.1. IWD Algoritmasının Akış Diyagramı

Şekil 2.3’de IWD algoritması akış diyagramı verilmiştir.



Şekil 2.3. IWD algoritması akış diyagramı.

Şekil 2.4’de IWD algoritmasının 5.adımının alt adımlarının ayrıntılı bir akış diyagramı verilmiştir.



Şekil 2.4. IWD algoritması adım 5 alt adımları.

2.4. SIRT ÇANTASI PROBLEMİ (KNAPSACK PROBLEM)

Sırt çantası problemi bir kombinatoriyal problemidir. Bu problem şu şekilde tanımlanabilir. t nesneleri göz önüne alındığında her nesnenin bir fiyatı (p) ve ağırlığı (w) vardır. Bu problemin amacı, tüm nesneler arasında sınırlı kapasite (c) ile sırt çantasını en yüksek fiyat ile yerleştirmektir.

$$\max f(x) = \sum_{i=1}^t p_i x_i \quad (2.3)$$

$$\sum_{i=1}^t w_i x_i \leq C; x_i \in \{0,1\}, i = 1, 2, \dots, n \quad (2.4)$$

Denklem (2.3) ve Denklem (2.4)'de t nesne sayısı, p_i i . nesnenin fiyatı, w_i i . nesnenin ağırlığı ve x_i i . nesnenin seçili olup olmadığını gösterir. Nesne seçili ise 1, seçili değil ise 0 değerini alır. Bu tez çalışması kapsamında ele alınan problem sırt çantası problemidir. Bu probleminin çözümü basit gibi görünse de problemin hesaplama karmaşıklığı problemin boyutuyla birlikte üssel olarak artar. Bu nedenle sırt çantası problemi NP-zor kombinatorial optimizasyon problemidir [25].

2.4.1. MBO Algoritmasının Sırt Çantası Problemine Uygulanması

2.4.1.1. Başlangıç Popülasyonu

İlk popülasyon rastgele oluşturulur. Popülasyondaki her birey arama alanındaki bir çözümü temsil eder ve bir çözüm permütasyonuna sahiptir. Permütasyon uzunluğu, kuş sayısına eşittir ve 0 ya da 1 olmak üzere iki farklı değeri vardır. Popülasyon boyutu başlangıçta bellidir ve algoritma sonlanıncaya kadar popülasyonun büyüklüğü aynı kalır. Popülasyonun boyutu en az üç olmalı ve tek sayılardan seçilmelidir. Bunun nedeni sürünün “V” şeklini oluşturmasından kaynaklanmaktadır. Bir lider kuş olacaktır ve diğer kuşlar, lider kuşun sağında ve solunda eşit sayıda olacak şekilde dağılacaklardır.

2.4.1.2. Komşuluk ve Paylaşım

MBO'da lider kuş en az 3 komşu çözüm üretir. Oluşturulan en iyi komşu çözüm mevcut çözümü geliştirmek için ayrılır. Mevcut çözümü geliştirmek en iyi komşu çözümü kullanmaktır. Kalan çözümler sol aşağıdaki ve sağ aşağıdaki çözümlerle paylaşılır.

MBO'da komşu çözüm sayısı algoritmanın başlangıcında parametre olarak bellidir ve algoritma sonlanıncaya kadar değişmez. Ancak komşu çözüm üretme sayısı, lider ve diğer bireyler için farklı değerdedir. Toplamda her birey aynı sayıda komşu çözüme sahip olmasına rağmen üretilen komşu çözümlerin sayısındaki farklılık komşu paylaşımından kaynaklanmaktadır.

2.4.2. IWD Algoritmasının Sırt Çantası Problemine Uygulanması

IWD algoritmasındaki herhangi bir IWD için, IWD'nin bir sonraki düğümü Denklem (2.5) kullanılarak olasılık olarak bulunur. Bu nedenle, herhangi bir düğümü ziyaret etme olasılığı sıfırın üzerinde olduğu sürece, bir olasılıkla algoritmanın bir IWD'sinin bu düğümü bir miktar yinelemede seçmesi beklenir.

$$p_i^{IWD}(j) = \frac{f(soil(i,j))}{\sum_{k \in vc(IWD)} f(soil(i,k))} \quad (2.5)$$

Denklem (2.5)'de $f(soil(i,j))$, i ve j düğümü arasındaki toprağın miktarını hesaplar ve $p_i^{IWD}(j)$, bir IWD'nin i düğümünden j düğümüne gitme olasılığıdır.



3. GÖÇ EDEN KUŞLAR VE AKILLI SU DAMLALARI OPTİMİZASYON ALGORİTMALARININ BİRLİKTE ÇÖZÜMÜ

MBO algoritmasının rastlantısal veri ihtiyacı duyduğu çeşitli aşamalarında rastlantısal veri kaynağı olarak IWD algoritması kullanılmıştır. IWD tarafından üretilen veriler MBO algoritmasındaki rastlantısal veriler olarak kullanılıp bu yeni birleşimin verimlilik analizi yapılmıştır. MBO algoritması iki noktada rastlantısal veriye ihtiyaç duyar: başlangıç çözümlerinin oluşturulması ve komşu çözümleri üretme. MBO ve IWD algoritmaları birleştirilerek oluşturulan bu yeni algoritmaya MBO-IWD adı verilmiştir.

3.1. MBO-IWD ALGORİTMASININ SIRT ÇANTASI PROBLEMİNE UYGULANMASI

3.1.1. Başlangıç Popülasyonu

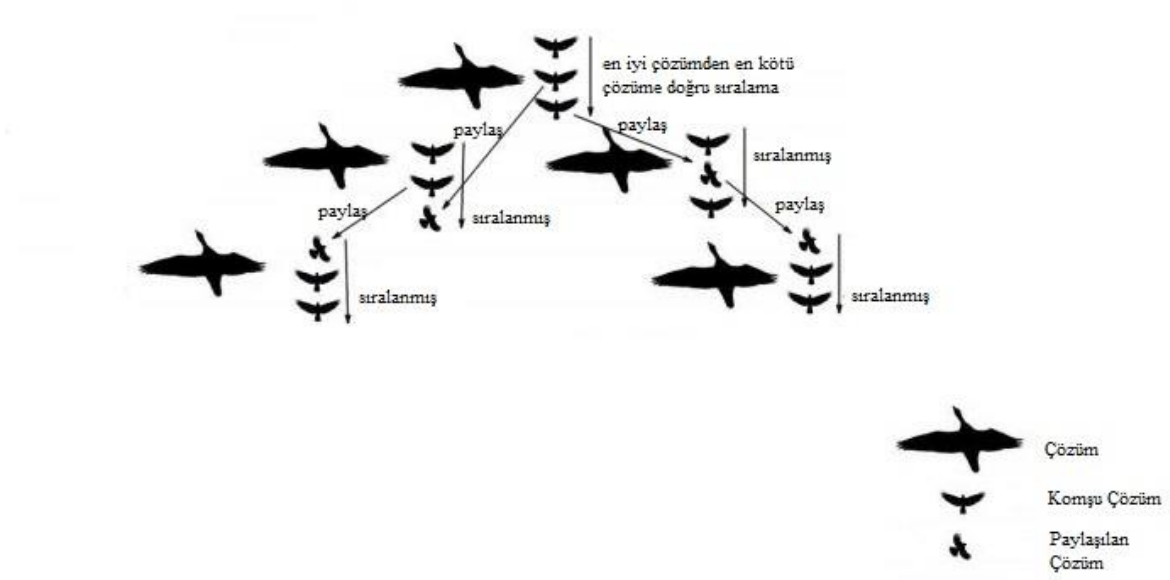
MBO algoritmasında ilk popülasyon rastgele oluşturulurken MBO-IWD algoritmasında ilk popülasyon IWD algoritması ile oluşturulur. Çözülecek olan problem IWD algoritması ile ilk popülasyonda ihtiyaç duyulacak birey sayısı kadar çözülür ve her bir çözüm bir bireyi temsil eder. Daha sonra her bir birey için bir çözüm popülasyonu oluşturulur.

3.1.2. Komşu Çözümleri Üretme

Komşu çözümler MBO algoritmasında rastgele oluşturulur, fakat MBO-IWD algoritmasında IWD algoritması ile oluşturulmaktadır. Lider kuş en az üç komşu çözüm üretmek zorundadır. Çünkü en iyi çözümü kendisine saklayıp diğer çözümleri sağ ve sol alttaki kuşlarla paylaşır. Lider kuş dışında kalan kuşlar ise lider kuştan daha az sayıda komşu çözüm üretirler. Çünkü lider kuşun üretmiş olduğu komşu çözümlerden de kendilerine en az bir çözüm gelecektir.

Bir dizi yinleme sonrasında MBO-IWD algoritmasının çalışması durur ve algoritmaya sokulan problem için çözüm vermiş olur.

Şekil 3.1’de sürüdeki kuşların komşu çözümleri üretimi ve üretilen komşu çözümlerin paylaşımı verilmiştir.



Şekil 3.1. Komşu çözüm üretimi ve komşu çözüm paylaşımı.

3.1.3. Kanat Çırpma ve Lider Kuşun Değişimi

Kanat çırpma parametresi de diğer parametrelerde olduğu gibi algoritmanın başlangıcında belirlenmektedir. Bu parametre algoritma sonlandırılınca kadar aynı kalır. Kanat çırpma parametresi, sürüdeki lider değişiminin ne sıklıkla yapılacağını belirler. Orijinal MBO algoritmasında “V” şeklindeki sürünün dizilimi belirli aralıklarla değiştirildiği için MBO-IWD algoritmasında da bu dizilim belirli aralıklarla değiştirilir.

Lider kuşun ilk değişimi sürünün sol tarafında olur. Lider kuş, sürünün sol tarafının en sonuna gider ve sol taraftaki ilk kuş yeni lider olarak atanır. Yeni lider kuşun değişim zamanı geldiğinde bu defa değişim sürünün sağ tarafına uygulanır. Sol taraftan gelip lider olan yeni lider kuş sürünün sağ tarafının en sonuna gider ve sürünün sağ tarafındaki ilk kuş yeni lider olarak atanır. Bu lider kuş değiştirme ve yeni lider kuş atama işlemleri algoritmanın çalışması sonlanıncaya kadar devam eder ve değişim sırasıyla sürünün bir sol tarafına bir sağ tarafına uygulanır. Bu değişiklik işlemlerinde, lider kuş taraflardan birinin en sonuna gittikten sonra o taraftaki ilk kuş yeni lider olarak atanır ve oluşan boşluğu doldurmak için diğer kuşlar bir önünde bulunan kuşun (artık boşta olan) yerine geçerler. Şekil 3.2’de Lider kuşun ve diğer kuşların değişimleri gösterilmiştir.

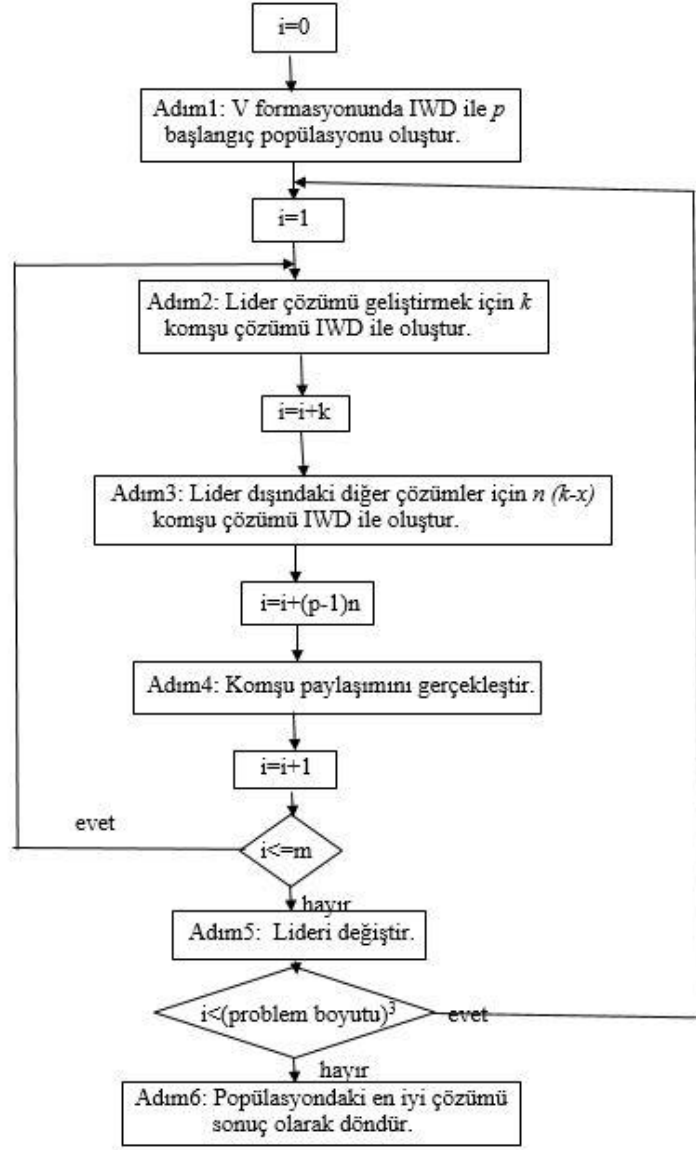


Şekil 3.2. Lider kuşun ve diğer kuşların değişimleri.

3.1.4. MBO-IWD Algoritması Akış Diyagramı

MBO-IWD Algoritmasında ilk adımda “V” şeklinde sürü için üretilecek kuşlar IWD algoritması ile bulunurlar. IWD algoritmasının statik ve dinamik parametrelerinin oluşturulduğu adım bu adımdır. Bu adımda her IWD damlacığı bir düğümle ilişkilendirilir. Her IWD damlacığı için ziyaret edilen ve ziyaret edilecek düğümlerin listeleri güncellenir. Bir sonraki adımda, önceki adımda oluşturulan popülasyonun lider kuşun çözümünü geliştirmek için komşu çözümler IWD ile üretilir. Komşu çözümler üretilirken orijinal IWD algoritmasındaki tüm adımlar kullanılır. Lider kuş için komşu çözümler üretildikten sonra lider dışındaki kuşlar için komşu çözümler yine IWD algoritması ile üretilirler. Lider kuş dışında kalan kuşlar için üretilecek komşu çözüm sayısı lider kuşun komşu çözüm sayısından daha azdır. Çünkü lider kuş kendisi için üretilen komşu çözümlerden en iyisini kendisine saklayıp diğerlerini sağ ve sol alttaki kuşlar ile paylaşır. Bir sonraki adımda bu komşu paylaşımı gerçekleştirilir. Komşu paylaşımı gerçekleştirildikten sonra kuşların kanat çırpma (tur) sayısı başlangıçta parametre olarak verilen kanat çırpma sayısına ulaşınca lider kuş sürünün sol tarafının en sonuna gider ve sol taraftaki ilk kuş yeni lider olarak atanır. Yeni lider için de aynı adımlar tamamlandıktan sonra lider kuş, sürünün sağ tarafının en sonuna gider ve sağ taraftaki ilk kuş yeni lider olarak atanır. Bu değişimler aşamasında kanat çırpma parametresi sıfırlanır. Bu işlemler sürünün bir sol tarafı bir sağ tarafı olacak şekilde algoritma sonlandırılıncaya kadar devam eder. Algoritma sonlandıktan sonra popülasyondaki en iyi çözüm sonuç olarak döndürülür.

Şekil 3.3’de MBO-IWD Algoritmasının akış diyagramı verilmiştir.



Şekil 3.3. MBO-IWD algoritması akış diyagramı.

Bu akış diyagramında p , başlangıç çözümleri sayısını; k , lider çözüm için üretilecek komşu çözüm sayısını; n , lider dışında kalan çözümler için komşu çözüm sayısını; x , bir önceki kuştan gelecek olan komşu çözüm sayısını ve m kanat çırpma (tur) sayısını belirtmektedir.

Şekil 3.3’de bulunan MBO-IWD Algoritması Akış Diyagramındaki Adım1, Adım2 ve Adım3’de bulunan IWD ile olan kısımlardaki akış diyagramı Şekil 2.3 ve Şekil 2.4’de daha önceden verilmiştir. Adım1’de IWD ile oluşturulan çözümlerin hepsinin birbirinden farklı olması gerekmektedir. Bu yüzden her üretilen çözüm daha öncekiler ile karşılaştırma yapıp farklı ise popülasyona eklenir.

3.1.4.1. MBO-IWD Algoritması Adım1 Sözde Kodu ve Açıklaması

MBO-IWD algoritmasının ilk adımında p adet başlangıç çözümü IWD algoritması ile elde edilecektir. p adet IWD algoritması problem için çalıştırılır ve çıkan sonuçlar MBO algoritmasındaki kuşları oluştururlar. Tüm sonuçlar bir dizi içerisinde tutulur ve böylece bir sonraki sonuç ile daha önceki sonuçların karşılaştırılması yapılır. Aynı olan bir sonuç varsa diziye eklenmez ve kuş olarak kabul edilmez algoritma yeniden çalıştırılır. p boyutlu dizinin 0. indeksli elemanı lider kuş, 1. indeksli elemanı sol taraftaki ilk kuş, 2. indeksli elemanı sağ taraftaki ilk kuş ve devamında sırasıyla önce sol sonra sağ taraftaki kuş olacak şekilde sıralanır. Adım1 sözde kodu aşağıda yer almaktadır.

1. p boyutlu tek boyutlu dizi oluştur
2. while sonlanmadı do
3. statik parametre başlangıç
4. dinamik parametre başlangıç
5. for $i=(1 \text{ to } N)$ do
6. for $j=(1 \text{ to } T)$ do
7. su damlası üzerindeki toprağı güncelle
8. su damlasının hızını güncelle
9. Global toprağı güncelle
10. En iyi sonuç p boyutlu dizide var mı diye bak
11. Varsa 5. adıma git
12. Yoksa p boyutlu diziye ata
13. end while

Bu sözde kodda N , IWD algoritmasının popülasyonundaki su damlacığı sayısını ve T döngü sayısını göstermektedir.

3.1.4.2. MBO-IWD Algoritması Adım2 Sözde Kodu ve Açıklaması

Lider kuşu geliştirmek için en az üç tane komşu çözüm IWD algoritması ile oluşturulur. Bu adımda sadece lider kuş için olan komşu çözümler oluşturulur. Diğer kuşlar için olan komşu çözümler bir sonraki başlıkta anlatılacaktır. Adım2 sözde kodu aşağıda yer almaktadır.

1. k boyutlu dizi oluřtur
2. while sonlanmadı do
3. statik parametre bařlangıç
4. dinamik parametre bařlangıç
5. for i=(1 to N) do
6. for j=(1 to T) do
7. su damlası üzerindeki topraęı gncelle
8. su damlasının hızını gncelle
9. Global topraęı gncelle
10. En iyi sonucu k dizisine at
11. end while

Bu szde kodda k, lider kuř iin retilcek olan komřu zm sayısını, N, IWD algoritmasının poplasyonundaki su damlacıęı sayısını ve T dng sayısını gstermektedir.

3.1.4.3. MBO-IWD Algoritması Adım3 Szde Kodu ve Aıklaması

Bu adımda lider kuř dıřında kalan kuřlar iin komřu zm IWD algoritması ile retilir. Lider kuř dıřında kalan kuřlar, lider kuřun retmiř olduęu komřu zm sayısından daha az sayıda komřu zm retirler. nk lider kuřtan da kendilerine komřu zm paylařımı gerekleřir. MBO-IWD algoritması Adım3 szde kodu ařaęıda yer almaktadır.

1. n boyutlu dizi oluřtur
2. while sonlanmadı do
3. statik parametre bařlangıç
4. dinamik parametre bařlangıç
5. for i=(1 to N) do
6. for j=(1 to T) do
7. su damlası üzerindeki topraęı gncelle
8. su damlasının hızını gncelle

9. Global toprağı güncelle

10. En iyi sonucu n dizisine at

11. end while

Bu sözde kodda n , lider kuş dışındaki kuşlar için üretilecek olan komşu çözüm sayısını, N , IWD algoritmasının popülasyonundaki su damlacığı sayısını ve T döngü sayısını göstermektedir. k lider kuş için üretilecek olan komşu çözüm sayısını ve x paylaşımdan gelecek olan komşu çözüm sayısını göstermek üzere n , $k-x$ demektir. Yani lider kuş 3 komşu çözüm üretecekse en iyi çözümü kendisine saklar ve ürettiğı komşu çözümlerden birini sol taraftaki kuş ile birini de sağ taraftaki kuş ile olmak üzere 2 tanesini paylaşır. Bu nedenle sol ve sağ taraftaki kuşlar ikişer tane komşu çözüm üretmelidirler. Bu kuşlar da en iyi çözümü kendilerine saklayıp birer tane çözümü altlarındaki kuşlar ile paylaşırlar.



4. BULGULAR

Tüm deneyler Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz işlemci, 1 GB RAM ve Windows 7 Home Basic (32-bit) işletim sistemine sahip bilgisayarda yapılmıştır. Algoritmaların kodlandığı dil Java olup NetBeans IDE 8.0.2 programı kullanılmıştır.

Literatürde yer alan ve daha önce MBO algoritması ile çözülüp MBO algoritması ile en iyi sonucu aldıkları [3]'de kanıtlanmış olan 10 adet sırt çantası optimizasyon problemi yeni birleşim algoritması olan MBO-IWD algoritması ile tekrar çalıştırılmış ve Çizelge 4.3'deki sonuçlar elde edilmiştir. Çizelge 4.1'de [3]'de kullanılan parametreler ile eşit olması açısından Sırt çantası problemleri için MBO-IWD algoritması parametreleri bulunmaktadır.

Çizelge 4.1. Sırt çantası problemleri için MBO-IWD algoritması parametreleri.

Parametre	Değer
Kuş (z)	71
Komşu (k)	3
Paylaşılan Komşu (x)	1
Durma Kriteri	$(problem\ uzunluğu)^3$

Çizelge 4.2.'de kullanılan 10 adet Sırt Çantası Problemi isimlendirilmiş ve problemlerin boyutu, ağırlığı (w), kapasitesi (C) ve fiyatı (p) verilmiştir.

Çizelge 4.2. On adet problemin boyut, ağırlık, kapasite ve fiyat değerleri [3].

Problem Adı	Boyut	Ağırlık (w)	Kapasite (C)	Fiyat (p)
P ₁	10	(95, 4, 60, 32, 23, 72, 80, 62, 65, 46)	269	(55, 10, 47, 5, 4, 50, 8, 61, 85, 87)
P ₂	20	(92, 4, 43, 83, 84, 68, 92, 82, 6, 44, 32, 18, 56, 83, 25, 96, 70, 48, 14, 58)	878	(44, 46, 90, 72, 91, 40, 75, 35, 8, 54, 78, 40, 77, 15, 61, 17, 75, 29, 75, 63)
P ₃	4	(6, 5, 9, 7)	20	(9, 11, 13, 15)

Çizelge 4.3. (devam) On adet problemin boyut, ağırlık, kapasite ve fiyat değerleri [3].

P ₄	4	(2, 4, 6, 7)	11	(6, 10, 12, 13)
P ₅	15	(56.358531, 80.874050, 47.987304, 89.596240, 74.660482, 85.894345, 51.353496, 1.498459, 36.445204, 16.589862, 44.569231, 0.466933, 37.788018, 57.118442, 60.716575)	375	(0.125126, 19.330424, 58.500931, 35.029145, 82.284005, 17.410810, 71.050142, 30.399487, 9.140294, 14.731285, 98.852504, 11.908322, 0.891140, 53.166295, 60.176397)
P ₆	10	(30, 25, 20, 18, 17, 11, 5, 2, 1, 1)	60	(20, 18, 17, 15, 15, 10, 5, 3, 1, 1)
P ₇	7	(31, 10, 20, 19, 4, 3, 6)	50	(70, 20, 39, 37, 7, 5, 10)
P ₈	23	(983, 982, 981, 980, 979, 978, 488, 976, 972, 486, 486, 972, 972, 485, 485, 969, 966, 483, 964, 963, 961, 958, 959)	10000	(981, 980, 979, 978, 977, 976, 487, 974, 970, 485, 485, 970, 970, 484, 484, 976, 974, 482, 962, 691, 959, 958, 857)
P ₉	5	(15, 20, 17, 8, 31)	80	(33, 24, 36, 37, 12)
P ₁₀	20	(84, 83, 43, 4, 44, 6, 82, 92, 25, 83, 56, 18, 58, 14, 48, 70, 96, 32, 68, 92)	879	(91, 72, 90, 46, 55, 8, 35, 75, 61, 15, 77, 40, 63, 75, 29, 75, 17, 78, 40, 44)

Çizelge 4.3’de literatürde bulunan 10 adet Sırt Çantası Problemi için MBO-IWD Algoritması sonuçları verilmiştir. Her problem için algoritma 30 defa çalıştırılmıştır ve en uygun sonuçlar [3] ile karşılaştırmalı olarak Çizelge 4.3’e yazılmıştır.

Çizelge 4.3. Sırt çantası problemleri için en iyi değer, MBO sonucu ve MBO-IWD sonucu.

Problem	Literatür Değeri	MBO Algoritması Değerleri			MBO-IWD Algoritması Değerleri		
		En İyi	Ortalama	En Kötü	En İyi	Ortalama	En Kötü
P1	295	295	294.6666	293	295	295	295
P2	1024	1024	1011.1	991	1024	1024	1024
P3	35	35	35	35	35	35	35
P4	23	23	23	23	23	23	23
P5	481.0694	481.0694	431.4180	399.8016	481.0694	481.0694	481.0694
P6	50	52	52	52	52	52	52
P7	107	107	107	107	107	107	107
P8	9767	9765	9765	9765	9767	9767	9767
P9	130	130	130	130	130	130	130
P10	1025	1025	1011.2	990	1025	1025	1025

Çizelge 4.3.’den görüldüğü üzere MBO P8 hariç bütün problemlerde en iyi değeri bulmuş. Özellikle P3, P4, P6, P7 ve P9 için tüm 30 denemede en iyi değeri bulmuş. Özellikle P6 da en iyi değer olan literatür değerinden daha iyi bir değer bulmuş. Ama yeni birleşim algoritmamız olan MBO-IWD algoritmasına baktığımızda tüm problemler için 30 denemede de en iyi sonucu vermiştir. MBO-IWD algoritmasının Çizelge 4.3.’de 10 adet Sırt Çantası probleminin her biri için en iyi, ortalama ve en kötü sonuçlarına baktığımızda da hepsinde en iyi sonucu verdiği görülmektedir. Kötü sonuç verdiği problem bulunmamaktadır.

5. SONUÇ

Bu tez çalışmasında, orijinal MBO algoritması ve geliştirilmiş versiyonu olan MBO-IWD algoritması Sırt Çantası Optimizasyon Problemlerine uygulanmıştır. Yapılan deneyler sonucunda MBO Algoritmasının ilk çözümleri ve komşu çözümleri üretilirken rastgele üretmek yerine IWD Algoritması kullanılarak elde edilen sonuçları çözüm olarak kabul etmek ve komşu çözümleri de IWD algoritması ile üretmenin literatür değerinden ve MBO ile çözümlenip elde edilen değerden daha iyi sonuçlar verdiği gözlemlenmiştir.

Orijinal MBO algoritmasında, göç eden kuşların göç esnasında daha uzun mesafelere daha az enerji harcayarak uçtukları “V” şeklindeki dizilişlerinden esinlenilmiştir. MBO, popülasyon tabanlı bir optimizasyon algoritmasıdır ve problemlerin çözümünde bir tek sürü kullanılır. Sürü bireyleri komşu çözümleri birbirleri ile paylaşarak bir fayda mekanizması kurarlar ve bu mekanizma sayesinde birbirleri ile iletişim halinde olurlar. Bu sayede sürüdeki her bir bireyin yine sürü içindeki en iyi konumdaki bireye doğru yakınsamaları sağlanmış olur. Ancak bu durum, bazı optimizasyon problemlerinin çözümünde algoritmanın yerel optimum çözümlere takılmasına neden olabilir. Bu durum MBO algoritmasında rastlantısal veriye ihtiyaç duyulan durumlarda IWD algoritmasını devreye sokarak ortadan kaldırılır.

Geliştirilen MBO-IWD algoritması da MBO algoritması gibi ayrık optimizasyon problemlerinin çözümü için denenmiştir. Sürekli optimizasyon problemlerinin çözümünde kullanılacak bir sürekli MBO-IWD algoritması geliştirilebilir.

6. KAYNAKLAR

- [1] E. Duman, M. Uysal, ve A. F. Alkaya, "Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem," *Information Sciences*, ss. 65-77, 2012.
- [2] H. Shah-Hosseini, "The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm," *International Journal of Bio-Inspired Computation*, c. 1, sayı 1, ss. 71-79, 2009.
- [3] E. Ulker ve V. Tongur, "Migrating birds optimization (MBO) algorithm to solve knapsack problem," *8th International Conference on Advances in Information Technoogy*, ss. 19-22, 2016.
- [4] D. Zou, L. Gao, S. Li ve J. Wu, "Solving 0-1 knapsack problem by a novel global harmony search algorithm," *Applied Soft Computing*, c. 2, sayı 11, ss. 1556-1564, 2011.
- [5] S. Elsherbiny, E. Eldaydamony, M. Alrahmawy ve A. E. Reyad, "An extended Intelligent Water Drops algorithm for workflow scheduling in cloud computing environment," *Egyptian Informatics Journal*, c. 1, sayı 19, ss. 33-35, 2018.
- [6] R. Kesavamoorthy, "Solving Traveling Salesman Problem by Modified Intelligent Water Drop Algorithm", *Proceedings published by International Journal of Computer Applicaitons*, ss. 18-23, 2011.
- [7] S. Kuzu, "Gezgin Satıcı Problemlerinin Metasezgiseller ile Çözümü," *İstanbul Üniversitesi İşletme Fakültesi Dergisi*, c. 1, sayı 43, ss. 1-27, 2014.
- [8] K. G. Murty, "Optimization Models For Decision Making," *Internet Edition Models for Decision Making*, sayı 1, ss. 1-18, 2003.
- [9] E. Eröz ve E. Tanyildizi, "Güncel Metasezgisel Optimizasyon Algoritmalarının Karşılaştırılması," *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Türkiye, 2018, ss. 1-16.
- [10] S. Mirjalili, "SCA: A Sine Cosine Algorithm for solving optimization problems," *Knowledge-Based Systems*, 2015.
- [11] R. C. Eberhart ve Y. Shi, "Particle swarm optimization: developments, applications and resoruces," *2001 Congress on Evolutionary Computation*, 2001, sayı 1, ss. 81-86.
- [12] J. Kennedy ve R. C. Eberhart, "Particle swarm optimization," *International Conference on Neural Networks*, 1995, ss. 1942-1948.
- [13] M. Dorigo ve M. Birattari, "Ant colony optimization," *C. Sammut, I.G. Webb(Eds.), Encyclopedia of Machine Learning*, ss. 36-39, 2010.
- [14] Y. Akcay, H. J. Li ve S. H. Xu, "Greedy algorithm for the general multidimensional knapsack problem," *Annals of Operations Research*, sayı 150, ss.17-29, 2007.
- [15] A. F. Alkaya, R. Algin, Y. Sahin, M. Agaoglu ve V. Sakalli, "Performance of Migrating Birds Optimization Algorithm on Continuous Functions," *Advances in*

Swarm Intelligence, ss. 452-459, 2014.

- [16] W. W. Leontief, "The use of indifference curves in the analysis of foreign trade," *The Quarterly Journal of Economics*, sayı 47, ss. 493-503, 1933.
- [17] G. B. Dantzig, "Programming in a linear structure," *Econometrica*, sayı 17, ss.73-74, 1949.
- [18] H. P. Williams, *Model building in mathematical programming*, 4. baskı, New York, USA: Wiley, 2013.
- [19] Anonim. (2020, 5 Haziran). *NEOS Optimizasyon Rehberi*. [Online]. Erişim: <http://www-fp.mcs.anl.gov/otc/Guide/index.html>.
- [20] W. Gong, Z. Cai, C. X. Ling ve H. Li, "A real-coded biogeography-based optimization with mutation," *Applied Mathematics and Computation*, 2010.
- [21] G. G. Emel ve Ç. Taşkın, "Genetik Algoritmalar ve Uygulama Alanları," *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, sayı 1, ss. 129-152, 2002.
- [22] M. Dorigo, M. Birattari ve T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, sayı 1, ss. 28-39, 2006.
- [23] R. Storn ve K. Price, "Differential evolution- A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, sayı 11, ss. 341-359, 1997.
- [24] S. Ehlers, "A procedure to optimize ship side structures for crashworthiness," *Proceedings of the Institution of Mechanical Engineers Part M Journal of Engineering for the Maritime Environment*, 2010.
- [25] V. Boyer, M. Elkihel ve D. El Baz, "Heuristics for the 0-1 multidimensional knapsack problem," *European Journal of Operational Research*, sayı 199, ss.658-664, 2009.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Nurdanur Pehlivan
Doğum Tarihi ve Yeri : 05.06.1991 Hendek
Yabancı Dili : İngilizce
E-posta : nrdnrphlvn@hotmail.ocm

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2016
Lise	Fen Bilimleri	Hendek Lisesi	2009

YAYINLAR

1. N. Pehlivan, R. Kara, “NOSQL Veri Tabanlarının Sorgu Performansları”, *3rd International Engineering Reseacrh Symposium*, ss. 408-412, 2019.
2. N. Pehlivan, R. Kara, “Göç Eden Kuşlar Optimizasyon Algoritmasının Akıllı Su Damlları Optimizasyon Algoritması ile Verimlilik Analizi”, *MAS European International Congress On Mathematics-Engineering-Natural & Medical Sciences – XI*, ss. 304-309, 2020.