



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



**META-SEZGİSEL ALGORİTMALAR
KULLANILARAK RÜZGÂR
TÜRBİNLERİNİN KONUMLANDIRILMASI**

Mehmet BEŞKİRLİ

DOKTORA TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

**Eylül-2018
KONYA
Her Hakkı Saklıdır**

TEZ KABUL VE ONAYI

Mehmet BEŞKİRLİ tarafından hazırlanan “Meta-Sezgisel Algoritmalar Kullanılarak Rüzgar Türbinlerinin Konumlandırılması” adlı tez çalışması 03/09/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Selçuk Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda DOKTORA TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan

Prof. Dr. Ahmet ARSLAN

Danışman

Doç. Dr. Halife KODAZ

Üye

Prof. Dr. Harun UĞUZ

Üye

Doç. Dr. Mesut GÜNDÜZ

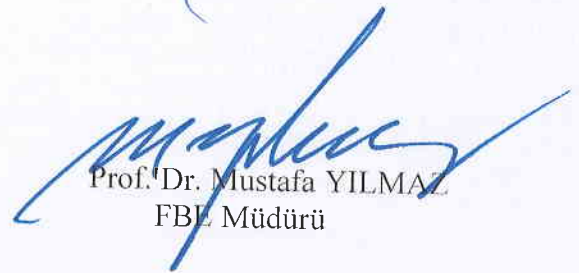
Üye

Dr. Öğr. Üyesi Ayşe Merve ACILAR

İmza



Yukarıdaki sonucu onaylarım.



Prof. Dr. Mustafa YILMAZ
FBE Müdürü

Bu tez çalışması ÖYP kurum koordinatörlüğü tarafından 2012-ÖYP-049 nolu proje ile desteklenmiştir.

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.



İmza

Mehmet BEŞKİRLİ

Tarih: 03/09/2018

ÖZET

DOKTORA TEZİ

META-SEZGİSEL ALGORİTMALAR KULLANILARAK RÜZGÂR TÜRBİNLERİNİN KONUMLANDIRILMASI

Mehmet BEŞKİRLİ

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Halife KODAZ

2018, 85 Sayfa

Jüri

Doç. Dr. Halife KODAZ

Prof. Dr. Ahmet ARSLAN

Prof. Dr. Harun UĞUZ

Doç. Dr. Mesut GÜNDÜZ

Dr. Öğr. Üyesi Ayşe Merve ACILAR

Gelişen teknoloji ile günümüzde rüzgârdan enerji üretmek giderek yaygınlaşmaktadır. Rüzgâr türbini, dünyada kullanımı en çok artan yenilenebilir enerji kaynaklarından biri haline gelmiştir. Rüzgâr enerjisinin önemi gitgide artarken rüzgâr türbinlerinin yerleştirme işleminin önemi de artmaktadır. Bir rüzgâr çiftliği için minimum yatırım maliyetiyle maksimum türbin gücü elde etmek amaçlanmaktadır. Bu çalışmada, bir rüzgâr çiftliğindeki rüzgâr türbinlerinin en yüksek verimi elde edecek şekilde yerleştirilmesi hedeflenmektedir. Bunun için iki kodlama yöntemi kullanılmıştır; birincisi ikili kodlama, ikincisi ise ikili kodlama ile gerçek kodlamanın birleştirilmesiyle elde edilen ikili-gerçek kodlamadır. İlk olarak ikili kodlama için türbin yerleştirme işlemi $2 \text{ km} \times 2 \text{ km}$ lik bir alan için tasarlanmıştır ve bu alan 10×10 'luk ve 20×20 'lik ızgaralara (gridlere) bölünerek hesaplanmıştır. Kodlamada ikili kodlama kullanılmıştır. Sürekli optimizasyon problemlerini çözmek için başarılı bir şekilde uygulanan Yapay Alg Algoritması (YAA) 10 farklı transfer fonksiyonu kullanılarak 10 farklı yeni ikili algoritmalar elde edilmiş ve bu algoritmalar türbin yerleştirme problemine uygulanmıştır. Aynı zamanda YAA ile birlikte aynı alan ve ızgaralara Yabani Ot Optimizasyon (YOO) algoritması da 14 farklı transfer fonksiyonu kullanılarak 14 farklı ikili algoritmalar elde edilmiş ve bu algoritmalar türbin yerleştirme problemine uygulanmıştır. İkinci olarak ikili-gerçek kodlama için türbin yerleştirme işlemi $2 \text{ km} \times 2 \text{ km}$ lik bir alan için tasarlanmıştır ve Modifiye Dağıtık Arama (MDA) algoritması kullanılmıştır. Başlangıç popülasyonunun üretilmesi için önerilen ön işlem sayesinde rüzgâr türbinlerinin arazi üzerine etkili bir şekilde yayılması sağlanmıştır. DA algoritmasında popülasyon çeşitliliği beş farklı çaprazlama tekniği ve önerilen mutasyon işlemi ile sağlanmıştır. Her iki kodlama yöntemi ile çözülen rüzgâr türbini yerleştirme problemi ikili kodlama da en iyi sonucu veren algoritma İkili Yapay Alg Algoritması (İYAA) olarak adlandırılmıştır. İkili türbin yerleştirme optimizasyon problemi için önerilen algoritmanın sonuçları, literatürdeki iyi bilinen algoritmaların sonuçları ile karşılaştırılmıştır. İYAA ikili arama uzayında optimum yerleştirmeyi yaparak rüzgâr türbini yerleştirme problemi için etkili ve kararlı algoritma olduğu görülmüştür. Onu takip eden ikili Yabani Ot Algoritması (İYOO) ise bu problemde tatmin edici bir sonuç elde etmiştir. İkili-gerçek kodlama da ise en başarılı ve verimli yerleştirmeyi yapan MDA algoritmasının rüzgâr türbini yerleştirme problemi için etkili bir algoritma olduğu ve en başarılı sonucu elde ettiği görülmüştür.

Anahtar Kelimeler: Dağıtık arama algoritması, İkili-gerçek kodlama, İkili kodlama, İkili yabani ot optimizasyon algoritması, İkili yapay alg algoritması, Optimizasyon, Rüzgâr türbini yerleştirme problemi.

ABSTRACT

Ph.D THESIS

POSITIONING OF WIND TURBINES USING META-HEURISTIC ALGORITHMS

Mehmet BEŞKİRLİ

**THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF
SELÇUK UNIVERSITY
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN COMPUTER ENGINEERING**

Advisor: Assoc.Prof.Dr. Halife KODAZ

2018, 85 Pages

Jury

**Assoc.Prof.Dr. Halife KODAZ
Prof. Dr. Ahmet ARSLAN
Prof. Dr. Harun UĞUZ
Assoc.Prof.Dr. Mesut GÜNDÜZ
Asst.Prof.Dr. Ayşe Merve ACILAR**

Today, producing energy from the wind has become widespread due to developing technology. Wind turbines have become one of the renewable energy sources, which have been increasingly used in the world. Importance of wind energy has increased, so does the significance of wind turbine placement. It is aimed to obtain maximum turbine power with minimum investment cost for a wind farm. This study aims to locate the wind turbines of a wind farm to obtain the highest efficiency. Two coding methods were used to achieve this: The first is the binary coding and the second is the binary-real coding obtained by combining the binary coding and real coding. First, for binary coding, turbine placement was designed for a 2 km×2 km area and this area was divided into 10×10 and 20×20 grids and then calculated. Binary coding was used in coding. Ten different binary algorithms were obtained by using ten different transfer functions and Artificial Algae Algorithm (AAA), which is successfully applied to solve continuous optimization problems and then these algorithms were applied to the turbine placement problem. In addition to AAA, Invasive Weed Optimization Algorithm (IWO) was applied to the same area and grids by using fourteen different transfer functions to obtain fourteen different binary algorithms, and these algorithms were applied to turbine placement problem. Secondly, for binary-real coding, turbine placement was designed for a 2 km×2 km area and Modified Scatter Search (MSS) algorithm was used. Due Thanks to the preliminary process proposed to produce the initial population, it was ensured that the wind turbines could be effectively scattered over the field. In the SS algorithm, population diversity was provided by five different crossover techniques and the proposed mutation operation. The algorithm, which also provided the best result in binary coding for the wind turbine placement problem solved by both coding methods, was called Binary Artificial Algae Algorithm (BAAA). The results of the algorithm proposed for the binary turbine placement optimization problem were compared with those of well-known algorithms in the literature. It was seen that the BAAA was an efficient and stable algorithm for the wind turbine placement problem by achieving the optimal placement in the binary search space. The Binary Invasive Weed Optimization Algorithm (BIWO) following BAAA obtained a satisfactory result in this problem. It was observed that in binary-real coding, MSS algorithm was the most successful and efficient algorithm for the wind turbine placement problem, achieving the most successful and effective placement.

Keywords: binary artificial algae algorithm, binary coding, binary-real coding, binary weed optimization algorithm, optimization, Scatter search algorithm, wind turbine placement problem.

ÖNSÖZ

Bu çalışmamda bana yol gösteren tez danışmanım Doç. Dr. Halife KODAZ'a, tez izleme komitesinde yer alan Prof. Dr. Ahmet ARSLAN'a ve Prof. Dr. Harun UĞUZ'a teşekkür ederim.

Çalışmalarım sırasında desteklerini ve yardımlarını esirgemeyen Dr. Hüseyin HAKLI, Arş. Gör. İsmail KOÇ arkadaşlarıma, Konya Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nün öğretim elemanlarına ve tezde emeği geçen herkese teşekkürlerimi sunarım.

Bugünlere gelmemde büyük pay sahibi olan, her zaman yanımda olup bana güç veren annem Emine BEŞKİRLİ ve babam Kerim BEŞKİRLİ'ye ayrıca yoğun zamanlarımda bana yardım eden ve benim en büyük destekçim olan kardeşim Ayşe BEŞKİRLİ'ye teşekkür ederim.

Mehmet BEŞKİRLİ
KONYA-2018

İÇİNDEKİLER

ÖZET	vii
ABSTRACT.....	viii
ÖNSÖZ	ix
İÇİNDEKİLER	x
SİMGELER VE KISALTMALAR	xii
1. GİRİŞ	1
2. KAYNAK ARAŞTIRMASI	5
2.1. Yapay Alg Algoritması ile İlgili Çalışmalar	5
2.2. Yabani Ot Optimizasyonu ile İlgili Çalışmalar	6
2.3. Dağıtık Arama ile İlgili Çalışmalar	7
2.4. Rüzgâr Türbini Yerleştirme Problemi ile İlgili Çalışmalar	7
3. MATERYAL VE METOT	12
3.1. Rüzgâr Türbini Yerleştirme Problemi	12
3.1.1. İkili, gerçek ve ikili-gerçek kodlama	15
3.2. Yapay Alg Algoritması	17
3.2.1. Evrimsel süreç	17
3.2.2. Adaptasyon	19
3.2.3. Helisel hareket	19
3.3. İkili Yapay Alg Algoritması	23
3.4. Yabani Ot Optimizasyon Algoritması	30
3.4.1. Başlangıç popülasyonu	31
3.4.2. Yeniden Üretme	31
3.4.3. Mekânsal dağılım	31
3.4.4. Rekabetçi dışlama	32
3.5. İkili Yabani Ot Optimizasyonu Algoritması	33

3.6. Dağıtık Arama Algoritması.....	35
3.6.1. Farklılaştırma üretim yöntemi	36
3.6.2. İyileştirme yöntemi	36
3.6.3. Referans kümesi güncelleme yöntemi	36
3.6.4. Altküme oluşturma yöntemi	37
3.6.5. Çözüm kombinasyonu yöntemi	37
3.6.6. Dağıtık arama algoritmasının probleme uygulanması.....	39
4. DENEYSEL SONUÇLAR VE TARTIŞMA	47
4.1. İkili Kodlama Sonuçları	47
4.1.1. İYAA Sonuçları.....	47
4.1.2. İYOO Sonuçları	62
4.2. İkili – Gerçek Kodlama Sonuçları.....	70
4.2.1. MDA Sonuçları	70
4.3. Tüm Deneysel Sonuçların Karşılaştırılması.....	74
5. SONUÇLAR VE ÖNERİLER	77
5.1. Sonuçlar	77
5.2. Öneriler.....	78
KAYNAKLAR	79
ÖZGEÇMİŞ	84

SİMGELER VE KISALTMALAR

Simgeler

R	: Referans kümesi
P_{tot}	: Toplam güç
Cost	: Toplam maliyet
$Cost/P_{tot}$	: Uygunluk değeri
Z	: Hub yüksekliği
C_T	: İtme katsayısı
Z_0	: Arazi pürüzlülüğü
r_r	: Rotor çapı
u_0	: İlk rüzgar hızı
σ	: Standart sapma

Kısaltmalar

YAA	: Yapay Alg Algoritması (Artificial Algae Algorithm - AAA)
YOO	: Yabani Ot Optimizasyonu (Invasive Weed Optimization - IWO)
İYAA	: İkili Yapay Alg Algoritması (Binary Artificial Algae Algorithm - BAAA)
İYOO	: İkili Yabani Ot Algoritması (Binary Invasive Weed Optimization - BIWO)
DA	: Dağıtık Arama (Scatter Search - SS)
MaxFES	: Maksimum fonksiyon değerlendirme sayısı
S1-2-3-4	: S-şekilli ailesi transfer fonksiyonları
V1-2-3-4	: V-şekilli ailesi transfer fonksiyonları
D	: Boyut
N	: Popülasyondaki birey sayısı
GA	: Genetik Algoritma (Genetic Algorithm)
DE	: Diferansiyel Evrim (Differential Evolution)
PSO	: Parçacık Sürü Optimizasyonu (Particle Swarm Optimization)
ABA	: Ateş Böceği Algoritması (Firefly Algorithm - FA)

1. GİRİŞ

Rüzgâr, en umut verici alternatif enerji kaynaklarından bir tanesidir. Dünyanın her yerinde en kapsamlı yenilenebilir enerji kaynağı haline gelmiştir. Yenilenebilir enerji sistemleri, çevre dostu, tutarlı, istikrarlı ve ekonomik sistemlerdir (Omer, 2008). Okyanus, jeotermal, güneş, rüzgâr, hidroelektrik gibi her türlü enerji kaynakları, doğal enerji kaynaklarıdır (GWEC, 2016). Rüzgâr, bu kaynaklar arasında enerji üretimi açısından en büyük öneme sahip olanıdır. Rüzgârın elektrik dönüşümünde yüksek bir verimliliği vardır. Mevcut ve gelecekteki elektrik talebini karşılayabilecek en karlı yenilenebilir enerji kaynaklarından bir tanesidir (WECTNC, 2015). Bunun için rüzgâr türbin sistemlerine ihtiyaç duyulmaktadır. Rüzgâr türbinleri, rüzgârdaki kinetik enerjiyi önce mekanik enerjiye daha sonra da elektrik enerjisine dönüştürmektedir (Bianchi ve ark, 2007, Keung ve ark, 2009).

Gelişen teknolojiyle günümüzde rüzgârdan enerji üretmek yaygınlaşmaya başlamıştır (Malen ve Marcus, 2017). Yenilenebilir enerji kaynaklarından biri olan rüzgâr çok cazip bir üretim kaynağı olmuştur (Wang ve Huang, 2016). Rüzgâr, rüzgâr türbinleri sayesinde elektriğe dönüştürülmektedir. Elektrik üretmek için farklı yöntemler de bulunmaktadır. Kömür, doğal gaz gibi fosil yakıtları günümüzün önemli elektrik üretim kaynaklarıdır (Omer, 2008, Shin ve ark, 2017). Ancak günümüzde, rüzgârdan elektrik üretimi daha fazla rağbet görmektedir (Long ve ark, 2017). Rüzgârın temiz enerji kaynağı olması, atmosferi kirletmemesi, elektrik üretim maliyetinin düşük olması gibi avantajlarından dolayı dünyada 80'den fazla ülkede rüzgâr türbinleri ile elektrik üretimi yapılmaktadır (Bilgili ve ark, 2010, GWEC, 2015).

Rüzgâr türbini kurulum talebi ve ilgili araştırma faaliyetleri son yıllarda artış göstermiştir. Bundan dolayı, rüzgârın hızla yükselen bir enerji kaynağı olduğu görülmektedir (Fried ve ark, 2012, Holttinen ve Ahlgrimm, 2013). 2015 sonu itibarıyla, dünya rüzgâr enerjisi üretim kapasitesi, toplam küresel enerji üretim kapasitesinin sadece %7'sidir. Toplam küresel rüzgâr enerjisinin üretimini artırmak için çeşitli destekler verilmektedir. Bunun için rüzgâr enerjisi, dünyanın gelecek enerji arzında kilit rol olacağı söylenmektedir (GWEC, 2016). Bunun sebebi rüzgârın elektriğe dönüşümünde daha yüksek bir verimliliğinin olması ve gelecekteki enerji talebini karşılayabilecek kapasitesinin olmasıdır (GWEC, 2015). Rüzgâr türbini enerji üretiminde alternatifler içerisinde en kuvvetli olsa da rüzgâr türbinlerini düzgün bir şekilde yerleştirilmediği

takdirde türbinin bu verimliliği çok daha düşük olacaktır. Bir rüzgâr çiftliği oluşturulurken dikkatli ve titiz bir çalışma yapılmalıdır. Yani rüzgâr türbini yerleştirmek, türbinden alınacak verimi etkileyeceğinden önemlidir (Bilgili ve ark, 2010, Fried ve ark, 2012, Holttinen ve Ahlgrimm, 2013, Long ve ark, 2017). Rüzgâr çiftliklerinde türbinlerin gelişigüzel yerleştirilmesi enerji üretiminde türbinlerin tam kapasite ile çalışmamasına sebebiyet verebilmektedir. Bununla birlikte, çoğu rüzgâr santrali işletmesi uyandırma etkisi (wake etkisi) nedeniyle tahmin edilen kadar enerji üretememektedir (Crespo ve Herná'ndez, 1996). Rüzgâr türbinlerinin uygun bir şekilde yerleştirilmesi, bir dereceye kadar uyandırma etkisi önleyebilmektedir ve bir rüzgâr çiftliğinde üretilen güç miktarını artırabilmektedir (Gao ve ark, 2015). Bununla birlikte enerji maliyetini düşürmek için rüzgâr çiftliği verimliliğini en üst düzeye çıkarmak gerekmektedir. Bu işlem ise optimizasyon teknikleri ile gerçekleştirilmektedir. Rüzgâr türbin yerleştirme problemi için ikili (binary) ve ikili-gerçek (binary-real) optimizasyon algoritmaları kullanılarak birçok çalışma yapılmıştır. İlk olarak Mosetti ve ark. (Mosetti ve ark, 1994) Genetik Algoritma (GA) kullanarak türbin yerleştirme problemini ikili kodlama kullanarak çözmeye çalışmışlardır. Önerdikleri algoritmanın fizibilitesini kanıtlayıp, rüzgâr türbinlerinin yerleşim planlarını vermişlerdir. Rüzgâr türbinlerini yerleştirmek için rüzgâr çiftliğini kare şeklindeki ızgaralara (gridlere) bölerek 0 ve 1 tipli ikili çözüm kodlaması gerçekleştirmiştir. Grady ve ark. rüzgâr çiftliğindeki türbinlerin en uygun yerleşimini GA kullanarak hesaplamışlardır (Grady ve ark, 2005). Marmidis ve ark. istatistiksel ve matematiksel bir yöntem olan Monte Carlo simülasyon yöntemini kullanarak toplamda 100 (10 x 10) kareden oluşan alan içerisinde uygun yerleşimleri bulmaya çalışmıştır (Marmidis ve ark, 2008). Gonzalez ve ark. rüzgar türbini yerleşim düzenini optimize etmek için Evrimsel Algoritma kullanmışlardır (González ve ark, 2010). Pookpant ve Ongsakul, toplamda 100 (10 × 10) kareden oluşan alan içerisinde rüzgar türbinlerinin optimum yerleştirilmesini bulmak için İkili Parçacık Sürü Optimizasyonu (BPSO) kullanarak kurulu güç başına maliyetin daha düşük olduğunu belirtmiştir (Pookpant ve Ongsakul, 2013). Wanger ve ark. rüzgâr türbinlerini belirli bir araziye yerleştirerek maksimum enerji üretimi hakkında bir çalışma gerçekleştirmiştir (Wagner ve ark, 2013). Changshui ve ark., Jensen uyandırma modelini kullanarak Tembel Açgözlü Algoritmasını türbin yerleştirme problemine uygulamıştır (Changshui ve ark, 2011). Shakoor ve ark. literatürdeki rüzgar çiftliği alanı boyutları ve akış yönündeki rüzgar hızının tam olarak kullanılamaması eksikliğinden bahsederek türbin yerleştirme işlemini GA kullanarak çözmüştür (Shakoor ve ark, 2015).

Rüzgâr türbinlerini yerleştirirken dikkatli ve titiz bir çalışma yapmak gerekmektedir. Türbin yerleştirme probleminin amacı, birim güç başına düşen toplam maliyeti düşürmek için bir rüzgâr çiftliğinin türbin düzenini optimize etmektir. Tez çalışmasında, 10×10 ve 20×20 şeklinde iki farklı ızgara yapısı kullanılarak türbin yerleştirme işleminin performansı analiz edilmiştir. Orijinal Yapay Alg Algoritması (YAA) algoritması için 14 farklı transfer fonksiyonu kullanılarak ikili hale getirilen algoritmalar (İYAA1-İYAA14) bu ızgara yapıları üzerinde ayrı ayrı çözülmüştür. 14 farklı algoritma içerisinde en iyi sonucu veren algoritma, İkili Yapay Alg Algoritması (İYAA) olarak adlandırılmıştır. Orijinal yabani ot optimizasyonu (YOO) algoritması için 14 farklı transfer fonksiyonu kullanılarak ikili hale getirilen algoritmalar (İYOO1-İYOO14) bu ızgara yapıları üzerinde ayrı ayrı çözülmüştür. 14 farklı algoritma içerisinde en iyi sonucu veren algoritma, ikili yabani ot optimizasyonu İYOO olarak adlandırılmıştır. Bu iki algoritma ile birlikte ikili gerçek kodlama yöntemi için Modifiye Dağıtık Arama (MDA) algoritmasında iyileştirmeler yapılarak rüzgâr türbini yerleştirme problemine uygulanmıştır. Önerilen algoritmaların sonuçları, literatürdeki diğer çalışmaların sonuçları ile karşılaştırılmıştır. İkili kodlama yöntemi ile en başarılı ve verimli yerleştirmeyi yapan İYAA algoritması ve onu takip eden İYOO algoritmasının rüzgâr türbini yerleştirme problemi için etkili ve kararlı algoritmalar olduğu görülmüştür. İkili-gerçek kodlama yöntemi ile yapılan çalışmada ise literatürdeki çalışmalardan en iyi yerleştirmeyi yapan MDA algoritmasının rüzgâr türbini yerleştirme problemi için etkili ve kararlı algoritma olduğu görülmüştür.

Bu tez çalışması beş bölümden oluşmaktadır.

İlk bölümde rüzgâr enerjisi ve rüzgâr türbini yerleştirme problemi ile ilgili genel bilgiler verilerek tez içeriği hakkında kısa bir giriş yapılmıştır.

İkinci bölümde rüzgâr türbini yerleştirme problemi ile tez çalışmasında kullanılan algoritmalar için kaynak araştırması yapılarak literatürde bulunan çalışmalar açıklanmıştır.

Üçüncü bölüm materyal ve metotları içermektedir. Rüzgâr türbini yerleştirme probleminde kullanılan kodlama yöntemleri açıklanmıştır. Ayrıca YAA, YOO ve MDA algoritmalarından kısaca bahsedilip yapılan yenilikler anlatılmış ve probleme uygulanması anlatılmıştır.

Dördüncü bölümde geliştirilen algoritmalar rüzgâr türbini yerleştirme problemine uygulandıktan sonra elde edilen yerleştirme sonuçlarının ve analizlerinin karşılaştırılması gerçekleştirilmiştir.

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



2. KAYNAK ARAŞTIRMASI

Kaynak araştırması dört farklı başlık altında gerçekleştirilmiştir. İlk üç bölümde YAA, Yabani Ot Optimizasyon (YOO) algoritması ve Modifiye Dağıtık Arama (MDA) algoritmasının uygulama alanları için kaynak araştırması gerçekleştirilmiştir. Son bölümde rüzgâr türbini yerleştirme problemi için literatürde kullanılan yöntemler incelenmiştir.

2.1. Yapay Alg Algoritması ile İlgili Çalışmalar

YAA Uymaz tarafından 2015 yılında sürekli optimizasyonların çözümü için geliştirilen doğa esinli bir algoritmadır (Uymaz ve ark, 2015). YAA ile ilgili birçok çalışma gerçekleştirilmiştir. Bunlardan bazıları şu şekildedir;

Zhang ve arkadaşları tarafından çok boyutlu sırt çantası problemleri için ikili yapay alg algoritması isimli makalesinde önerilen algoritma sonuçlarının birçok mevcut algoritmanın sonuçlarına göre daha iyi olduğunu söylemişlerdir (Zhang ve ark, 2016).

Zahid ve arkadaşları tarafından acil durum analizinde güç sistemlerinin istikrarının iyileştirilmesinde birleşik güç akış kontrolörünün optimal konum ve parametre ayarı için yeni yaklaşım isimli makalesinde yeni bir YAA optimizasyon tekniğini kullanarak PSO ve DE gibi mevcut algoritmaların sonuçları ile karşılaştırma yapmışlardır (Zahid ve ark, 2017).

Katarya ve Verma YAA ve bulanık c-ortalaması algoritmalarını kullanılarak yeni bir işbirlikçi öneri sistemi geliştirmişlerdir. Kümeleme için yapay alg algoritması ile bulanık c-ortalamaı kullanmışlardır. Önerdikleri bu sistemi ile Movielens 100,000, Movielens 1 milyon, Jester and Epinion veri setleriyle zamana göre üstün bir performans gösterdiğini söylemişlerdir. (Katarya ve Verma, 2017)

Kumar ve Dhillon tarafından ekonomik yük dağıtımı için hibrit yapay alg algoritması isimli çalışmada alg algoritması, tek yönlü arama yöntemi ile melezleştirilerek hibrit alg algoritmasını oluşturmuşlardır. Bu algoritma ile standart CEC'05 fonksiyon seti ve beş ekonomik yük dağıtımı problemi üzerinde çalışma yaparak temel alg algoritmasının sonuçları ile performans karşılaştırma gerçekleştirmişlerdir. Hibrit algoritmanın, temel alg algoritmasına kıyasla hesaplama açısından verimli ve sağlam olduğunu belirtmişlerdir (Kumar ve Dhillon, 2018).

Korkmaz ve Kiran tarafından ikili optimizasyon problemlerini çözmek için yapay bir alg algoritması isimli yapılan çalışmada tamsayı veya gerçek değerli karar değişkenleri bulunmayan tesis yeri problemleri üzerinde araştırma gerçekleştirmişlerdir. Önerdikleri algoritma ile elde ettikleri sonuçlar, Yapay Arı Kolonisi (ABC), Parçacık Sürü Optimizasyonu (PSO) ve Genetik Algoritma) GA gibi yöntemlerin sonuçları ile karşılaştırarak önerilen algoritmanın çözüm kalitesi ve sağlamlık açısından daha iyi olduğunu söylemişlerdir (Korkmaz ve Kiran, 2018).

2.2. Yabani Ot Optimizasyonu ile İlgili Çalışmalar

YOO, Mehrabian ve Lucas tarafından 2006 yılında geliştirilen biyolojik tabanlı bir algoritmadır (Mehrabian ve Lucas, 2006). YOO ile ilgili birçok çalışma gerçekleştirilmiştir. Bunlardan bazıları şu şekildedir:

Barisal ve Prusty tarafında muhalif yabani ot optimizasyonunu kullanarak güç sistemlerinin büyük ölçekli ekonomik dağıtımı isimli makalesinde muhalif tabanlı öğrenme YOO'ya uygulanmış ve elde ettikleri sonuçlar diğer test edilmiş tekniklere göre üstünlüğü göstermişlerdir (Barisal ve Prusty, 2015).

Zheng ve Li tarafında enerji tüketimini azaltmak için geliştirilmiş yabani ot optimizasyonu algoritmasıyla optimal soğutucu yüklemesi isimli makalesinde YOO algoritmasının elit otu yerel arama stratejisi geliştirilmesi ve parametre ayarlaması yapılmış ve deneysel sonuçların en uygun çözümü bularak algoritmanın kararlılığı ve sağlamlığını doğruladığını söylemişlerdir (Zheng ve Li, 2018).

Sang ve arkadaşları tarafından yapılan yarı iletken son test problemi zamanlaması için etkili bir yabani ot optimizasyon algoritması isimli çalışmada YOO'da eş-evrimsel algoritması ile iş birliği yaparak bir iyileştirme gerçekleştirmişlerdir. Önerilen algoritma ile elde ettikleri sonuçların, literatürde yer alan en gelişmiş algoritmalarından çok daha iyi performans sergilediğini söylemişlerdir (Sang ve ark, 2018).

Naidu ve Ojha tarafından sabit nokta problemini çözmek için bir uzay dönüşümlü yabani ot optimizasyonu isimli çalışmada YOO'ya uzay dönüşüm araması ekleyerek oluşturulan iyileştirilmiş algoritma sonuçlarının orijinal algoritma sonuçlarına göre daha etkili ve verimli olduğunu söylemektedirler (Naidu ve Ojha, 2018).

2.3. Dağıtık Arama ile İlgili Çalışmalar

Glover tarafından 1977 yılında geliştirilen evrimsel tabanlı bir algoritmadır (Glover, 1977). DA ile ilgili birçok çalışma gerçekleştirilmiştir. Bunlardan bazıları şu şekildedir;

Naderi ve Ruiz tarafından “dağıtılmış permütasyon akış çizelgesi programlama problemi için bir DA algoritması” isimli çalışma gerçekleştirmiştir. Gelişmiş teknikler kullanarak istatistiksel analizlerle birlikte mevcut algoritmaların sonuçlarının bir kıyaslamasını yapmışlardır. Elde ettikleri sonuçlar incelendiğinde önemli bir farkla önerilen yöntemin daha iyi bir çözüm elde ettiği tespit edilmiştir (Naderi ve Ruiz, 2014).

Alaei ve Setak tarafından “kombinatorial açık artırmada kazanan belirleme ile benzersiz tedarikçilerin seçilmesi: DA algoritması” makalesinde önerilen algoritmanın performansını ölçmek için farklı gerçek dünya problemleri kullanarak çözmüşlerdir. DA algoritmasının bu problemi çözmede iyi performans gösterdiğini belirtmişlerdir (Alaei ve Setak, 2017).

Abdulah ve arkadaşları tarafından “gezgin satıcı problemini çözmek için dinamik dağıtık arama algoritması isimli çalışmada gezgin satıcı problemini çözmek için dinamik bir referans seti güncellemesi ve dinamik popülasyon güncellemesi” isimli çalışmayı önermişlerdir. Hesaplamalı sonuçlar ile önerilen algoritmanın bu problemi çözmede etkili olduğunu göstermişlerdir (Abdulah ve ark, 2017).

Marichelvam ve Mariappan tarafından “esnek işyeri çizelgeleme problemlerini çözmek için bir hibrit genetik dağıtık arama algoritması: zamanlama problemleri için hibrit bir algoritma” isimli çalışmada elde ettikleri hesaplamalı sonuçlar, esnek işyeri planlama problemini çözmek için önerdikleri algoritmanın etkililiğini kanıtladığını göstermişlerdir (Marichelvam ve Mariappan, 2018).

2.4. Rüzgâr Türbini Yerleştirme Problemi ile İlgili Çalışmalar

Ituarte-Villarreal ve Espiritu “viral tabanlı optimizasyon algoritmasını kullanarak rüzgâr türbinlerinin yerleşimi” isimli çalışmada rüzgâr çiftliğinde rüzgâr türbinlerinin en uygun sayı ve konumların araştırmak için MATLAB tabanlı bir program geliştirmiştir. Birim maliyeti en aza indirmeyi amaçlamıştır. Elde ettiği sonuçları literatür ile karşılaştırmış ve elde edilen sonuçlarda önemli gelişmeler olduğunu söylemişlerdir (Ituarte-Villarreal ve Espiritu, 2011).

Changshui ve arkadaşları “rüzgâr türbinleri konumlandırılmasının optimizasyonu için alt modüler özelliklerine dayanan hızlı bir algoritma” isimli çalışma yapmışlardır. Bir rüzgâr çiftliği tasarımında türbin yerleştirme işleminin verimliliği ve kârı etkileyen önemli bir faktör olduğunu söylemekle birlikte türbinlerin hala otomatik olarak yerleştirilmesinin de zor olduğunu belirtmektedirler. Çalışmasında tembel açgözlü algoritmayı kullanarak rüzgâr türbini yerleştirme işlemi gerçekleştirmiştir. Bu çalışma 2 x 2 km’lik bir alan için 10 x 10’luk bir ızgara kullanılarak yapılmış ve elde edilen sonuçları literatürde bulunan Mosetti ve Grady’nin çalışması ile karşılaştırılıp daha iyi bir sonuç elde ettiğini söylemişlerdir. Ayrıca literatürü zenginleştirmek adına standart değerlerin dışında 4 x 4 km’lik bir alan kullanarak 15 m/s hız ile açgözlü algoritma, tembel açgözlü algoritma ve GA kullanarak bir yerleştirme işlemi gerçekleştirmiştir. Tembel açgözlü algoritma diğer iki algoritmadan çok daha kısa bir sürede çözmekle birlikte daha iyi bir sonuç elde ettiğini kanıtlamışlardır (Changshui ve ark, 2011).

Emami ve Noghreh “rüzgâr çiftliğinde rüzgâr türbinlerinin GA ile yerleştirilmesinde optimizasyona yeni yaklaşım” isimli çalışmasında uygunluk fonksiyonunda iyileştirme yaptığını söyleyerek rüzgâr türbini yerleştirme problemi için klasik metotları kullanarak çözümlerin çok karışık olduğunu söylemiştir. Uygunluk fonksiyonuna iki ağırlık değeri ekleyerek yaptığı iyileştirme ile türbin yerleştirme işlemi gerçekleştirmişlerdir. Bu iki ağırlık değerinin toplamı bir olmaktadır. Böylelikle uygunluk değerinden elde edilen sonuçların diğer algoritmaların uygunluk değerleri ile karşılaştırılmış ve yeni yaklaşımın daha iyi olduğunu söylemişlerdir (Emami ve Noghreh, 2010).

Marmidis ve arkadaşları “Monte Carlo simülasyonunu kullanarak bir rüzgâr parkında rüzgâr türbinlerinin optimum yerleştirilmesi” isimli makalesinde 100 olası türbin konumuna bölünmüş kare hücreyi kullanarak maksimum enerji üretimi ve minimum maliyet kurulum kriterlerine göre rüzgâr türbinlerini yerleştirmişlerdir. Monte Carlo yöntemi rastgele sayıların kullanımına dayanan bir yöntemdir. Bu problem için çözüm rastgele sayılardan etkilendiği için olasılıksal bir problemle başa çıkmak zorunda olduğunu söyleyerek bu yöntemi kullanmışlardır. Bu yöntem daha fazla yenileme işlemi gerçekleştirerek istenilen sonuca ulaşılabilirliğini söylemişlerdir. Bu yöntemin sonuçları ile birlikte GA sonuçları karşılaştırılmış ve önerdikleri yöntemin daha başarılı olduğu görülmüştür (Marmidis ve ark, 2008).

Mosetti ve arkadaşları “geniş rüzgâr tarlalarında rüzgâr türbini konumlandırılmasının GA ile optimizasyonu” isimli bir çalışma gerçekleştirmiştir. Bu

çalışma diğer çalışmalara öncülük etmektedir. Büyük rüzgâr santrallerinin optimizasyonuna yönelik yeni bir yaklaşım sunmuştur. Belirli bir sahadaki rüzgâr türbini dağılımı, minimum kurulum maliyetleri için maksimum enerjiyi çıkarmak için optimize etmişlerdir. Bu işlemleri, genetik arama kodu ile uyandırma etkisine dayanan bir yerleştirmeyi ilişkilendirerek rüzgâr santrali simülasyon modelini gerçekleştirmişlerdir (Mosetti ve ark, 1994).

Parada ve arkadaşları “Gauss tabanlı bir uyandırma model kullanarak rüzgâr çiftliği düzen optimizasyonu” isimli bir çalışma yapmışlardır. GA kullanılarak yapılan bu çalışmada Gauss tabanlı bir uyandırma modelini tercih edilmiştir. Rüzgâr türbinlerinin sebep olduğu uyandırma etkisi ve türbülans konusunda çok sayıda çalışma olduğunu söylemişlerdir. En çok kullanılan analitik uyandırma modeli Jensen uyandırma modeli olduğunu belirtmişlerdir. Bu tarz analitik uyandırma modellerini üst şapka modelleri olarak sınıflandırmıştır. Çünkü bunlar ayrı bir işlev ile radyal yönde hız açıklığını temsil etmektedir. Ama aslında hız açıklığı türbin rotorundan bir miktar yaklaşık asimetriktir (Chamorro ve Porté-Agel, 2009, Bartl ve ark, 2012) ve radyal yönde bir gauss eğrisini andırdığını belirtmişlerdir. Gauss uyandırma modeli Bastankhah ve Porte-Agel (Bastankhah ve Porté-Agel, 2014) tarafından önerilmiştir. Gauss uyandırma modeli Jensen uyandırma modeline göre daha iyi bir çözüm sunduğunu söylemişlerdir. Ancak Gauss uyandırma modelinde arkadaki türbinin türbülanstaki etkilenmesi azalmaktadır. Bu işlem sonucunda Jensen uyandırma modeli ve Gauss uyandırma modeli ile yapılan aynı bir yerleşim işleminde Gauss uyandırma modeli ile daha iyi bir uygunluk değeri elde edilmiştir. Parada’nın yapmış olduğu çalışma sonucunda yüksek verimli rüzgâr santrali düzen tasarımının Gauss uyandırma modeli ile tahmin edilen hızın Büyük Eddy Simülasyonu (LES) ile uyumlu olduğunu iddia edilmiştir. Bu optimizasyon probleminde GA benimsenmiş ve önerdikleri Gauss uyandırma modeli ile rüzgâr türbinlerinin uyandırma kaybı minimum olacak şekilde yerleştirilmesiyle bir rüzgâr çiftliğinin yıllık enerji maliyetini en aza indirmeyi sağlamışlardır (Parada ve ark, 2017).

Rahmani ve arkadaşları tarafından “PSO kullanarak bir rüzgâr çiftliğinde rüzgâr türbinlerinin optimal yerleştirilmesi için yeni bir yöntem” isimli çalışmasında literatürde ilk defa rüzgâr türbini yerleştirmek için PSO kullanılmışlardır. PSO ile türbin yerleştirme sonuçları literatürde yer alan GA’nın türbin yerleştirme sonuçları karşılaştırılmıştır. Türbin sayısı otuzun üzerinde ise PSO iyi bir sonuç vermemektedir. Ancak türbin sayısı otuz ve altında ise literatürden iyi sonuçlar elde etmişlerdir (Rahmani ve ark, 2010).

Shakoor ve arkadaşları “uyandırma kayıpların azaltılması için rüzgâr çiftliği düzenleme optimizasyonunun modellenmesi” isimli çalışmayı yapmışlardır. Bu çalışmayı, birim güç başına toplam maliyeti azaltmak için rüzgâr çiftliği alanının ve türbin düzeninin optimize edilmiş boyutlarını bulmak için yapmışlardır. Ayrıca rüzgâr çiftliği alanı boyutları üzerindeki araştırmalar ve akış yönündeki rüzgâr hızlarının tam olarak kullanımının literatürde mevcut olmadığından sabit rüzgâr senaryosu için en iyi rüzgâr çiftliği alanı boyutlarını elde etmek için bu yöntemi önermişlerdir. Her bir uyandırma etkisinden kaynaklanan hız açıklarını Jensen’in uyandırma modeli ile gerçekleştirmişlerdir. Operasyon güvenliği için bitişik türbinler arası zorunlu mesafeyi sürdürürken, rüzgâr türbini yerleştirmesinin en uygun konumlarını GA kullanarak gerçekleştirmektedirler. Toplam çiftlik alanı $2 \times 2 \text{ km}^2$ olup bu alan 100 özdeş hücreye bölünmüştür. Her hücrenin boyutu $200\text{m} \times 200\text{m}$ dir. Önerdikleri yöntem çiftlik alanın boyutsal olarak gerçekleştirmişlerdir. Yani ilk on dokuz hücrenin rüzgâra dik olarak gelmesi gerektiğini söyleyerek ilk on dokuz türbinde serbest akış rüzgâr kullanarak avantaj elde etmek istemişlerdir. Geriye kalan türbin hücrelerine türbin yerleştirmek için GA’yı kullanmışlardır. Bu sayede ilk on dokuz türbinden tam güç elde etmeyi amaçlamışlardır. Shakoor ve arkadaşlarının önerdikleri bu yöntem ile diğer yöntemler (Mosetti (Mosetti ve ark, 1994), Grady (Grady ve ark, 2005), Emami (Emami ve Noghreh, 2010), Turner (Turner ve ark, 2014), Gonzalez (González ve ark, 2010), Mittal (Mittal, 2010) ve Rahmani (Rahmani ve ark, 2010)) ile aynı algoritmayı kullanmasına rağmen rüzgar çiftliğini boyutsal (yön) değişimi ile belirgin ölçüde verim aldıklarını söylemişlerdir (Shakoor ve ark, 2015).

Turner ve arkadaşları tarafından “rüzgâr çiftliği düzenlerini optimize etmek için yeni bir matematiksel programlama yaklaşımı” isimli çalışması önerilmiştir. Rüzgâr çiftliklerindeki türbinlerin optimum yerleştirilmesi, güç üretimini en üst düzeye çıkarmasında kritik bir etkiye sahip olduğunu belirtmişlerdir. Önerdikleri yöntem ile bir rüzgâr çiftliğinde türbinlerin yerleştirilmesini kinetik enerji açıklarını en aza indirerek optimize etmek için sunmuşlardır. Yerleştirme işlemleri için GA kullanarak yapan bu çalışma literatürde yer alan yerleştirme düzeninden iyi ve elde ettiği sonuç olarak daha iyi olduklarını belirtmişlerdir (Turner ve ark, 2014).

Massan ve arkadaşları “Ateş Böceği Algoritmasını (ABA) kullanarak rüzgâr türbini mikro yerleşimi” isimli çalışma gerçekleştirmişlerdir. Rüzgâr çiftliği düzenlemesinde çoklu uyandırma etkileşimlerinin analizi için Jensen uyandırma modelini kullanmışlardır. Türbinler 200m aralıklı bir ızgara yapısı kullanarak toplam 100 ızgara

konumlu bir kara alan ile yerleştirme yapmışlardır. Ayrıca yaptıkları çalışmayı GA ile yapılan türbin yerleştirme problemi sonuçları ile karşılaştırarak ABA'nın daha başarılı bir yerleşim yaptığını ve daha iyi sonuçlar elde ettiğini tablolarda belirtmişlerdir (Massan ve ark, 2015).

Rehman ve arkadaşları “Guguk Kuşu Arama Algoritması (GKAA) kullanarak rüzgâr çiftliği düzeninin tasarımı” isimli çalışmasını yapmışlardır. Rüzgâr çiftliğinden maksimum enerjinin çıkarılması, rüzgâr türbinlerinin en uygun şekilde yerleştirilmesini gerektirdiğini söylemektedirler. Türbinlerin yerleştirildiği alan ayrık veya sürekli bir temsil olarak kabul edilmiştir. Her iki durumda da çözüm alanı (tüm olası çözümlerin sayısı) çok büyüktür. Yani ayrık gösterim kullanılırsa ve rüzgâr çiftliği alanı 100 eşit büyüklükteki karelere bölünürse bu bir karenin bir türbin içerip içermediğine bakılmaksızın 2^{100} olası çözüm ile türbin yerleşeceğini ifade etmişlerdir. Bu nedenle çözüm alanının akıllıca araştırılması ve optimal çözüme ulaşmak için bazı sezgisel yöntemlerin kullanmanın gerekli olacağını belirtmişlerdir. Rüzgâr çiftliği düzen tasarımı problemi karmaşık bir optimizasyon problemi olduğu ve yakın geçmişte rüzgâr çiftliklerinden enerji çıkışının optimizasyonu için çeşitli teknikler ve algoritmaların geliştiğini söylemişlerdir. Şu anki çalışmalarında ise GKAA'ya dayalı bir optimizasyon yaklaşımını önererek GA ve PSO'nun türbin yerleştirme sonuçları ile karşılaştırma yapmışlardır. Çiftlik 2 x 2 km'lik bir alan olup 10 x 10'luk ızgara sistemi kullanarak türbinlere arası mesafeyi korumuşlardır. Literatürde yer alan GA ve PSO çalışmalarıyla aynı şartlarda çalıştırılmış ve GKAA ile elde edilen türbin yerleştirme sonuçlarının GA ve PSO algoritmaları ile yapılan türbin yerleştirme sonuçlarına göre daha iyi performans gösterdiğini söylemişlerdir (Rehman ve ark, 2016).

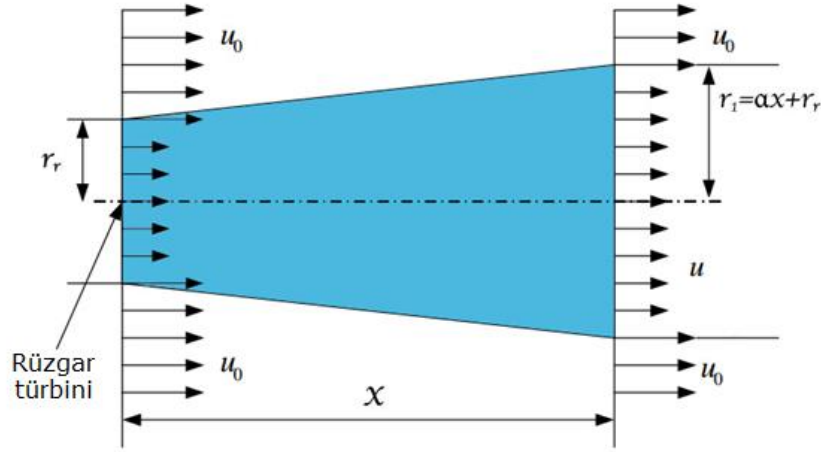
3. MATERYAL VE METOT

Tez çalışmasında kullanılan materyaller ve yöntemler hakkında detaylı bilgi bu başlık altında verilmektedir. Bu bölümde öncelikle rüzgâr türbini yerleştirme problemi hakkında bilgi verildikten sonra tezde kullanılan YAA ve YOO algoritmaları açıklanmıştır. Bu algoritmaların rüzgâr türbini yerleştirme probleminin çözümünde kullanılmak üzere ikili koda çevrilmiş halleri önerilen algoritmalar olarak anlatılmıştır. Ayrıca DA algoritmasının ikili-gerçek kodlama yöntemi kullanılarak rüzgâr türbini yerleştirme problemine uygulanması açıklanmıştır.

3.1. Rüzgâr Türbini Yerleştirme Problemi

Rüzgâr türbini yerleştirme problemi Belirleyici olmayan Polinom Zaman (Nondeterministic Polynomial Time (NP)) Problemi sınıfına girmektedir (Williams, 2014). Rüzgâr türbin çiftliklerinde arka arkaya gelen türbinler rüzgâr hızı açısından birbirlerini etkilemektedir. Türbinlerin birbirine etkisini hesaplamak için Jensen uyandırma etkisi modeli kullanılmıştır. Bu model, rüzgâr kullanılabilirliğine bağlı olarak hem kara üstü hem de deniz üstü olmak üzere her türlü yerleşim için kullanılabilir (Ituarte-Villarreal ve Espiritu, 2011). Fakat çevrenin korunmasıyla ilgili kurallara dikkat etmek rüzgâr çiftliği geliştiricisinin sorumluluğundadır (Protection, 2011).

Genel olarak rüzgâr çiftliklerinde rüzgâr türbinleri baskın rüzgâr yönüne dik olarak hizalanır. Türbinler arası mesafeye dikkat edilerek yerleştirilen türbinlerden maksimum verim elde edilmesi beklenir. Bunun ile ilgili ilk çalışma Mosetti tarafından Jensen uyandırma etkisi ve maliyeti modeli kullanarak gerçekleştirilmiştir (Mosetti ve ark, 1994). Bu model Jensen tarafından 1983 yılında geliştirilmiştir (Jensen, 1983). Jensen uyandırma etkisi modeli Şekil 3.1’de gösterilmiştir.



Şekil 3.1. Jensen uyandırma etkisi modelinin şeması (Wan ve ark, 2009).

Şekil 3.1’de taralı bölge ile uyandırma etkisi alanının aşağı doğru genişlediği ve uyandırma etkisi kuvvetinin zayıfladığı görülmektedir. Rüzgâr türbinine gelen u_0 rüzgâr hızı ile rüzgâr akış yönündeki x mesafesi sonunda elde edilen u rüzgâr hızı Denklem 3.1’de ifade edilmektedir.

$$u = u_0 \left[1 - 2a \left(1 + \frac{\alpha x}{r_r} \right)^{-2} \right] \quad (3.1)$$

Burada u_0 serbest rüzgâr hızını, a eksenel indüksiyon faktörünü, α sürüklenme sabitini ve r_r türbin yarıçapı yani ilk uyandırma yarıçapını ifade eder. Rüzgâr türbininin itme katsayısı (C_T) Denklem 3.2 kullanılarak hesaplanmaktadır.

$$C_T = 4a(1 - a) \quad (3.2)$$

Türbülansan etkilenen rüzgâr yarıçapı (r_1) dir ve Denklem 3.3 kullanılarak hesaplanmaktadır.

$$r_1 = r_r \sqrt{\frac{1-a}{1-2a}} \quad (3.3)$$

Denklem 3.4 kullanılarak α sürüklenme sabiti hesaplanmaktadır.

$$\alpha = \frac{0.5}{\ln(z/z_0)} \quad (3.4)$$

Burada z , rüzgâr türbininin göbek yüksekliğini ifade etmektedir. z_0 ise rüzgâr türbinin bulunduğu yüzey pürüzlülüğünü ifade etmektedir.

Rüzgâr santrali içindeki i iterlik türbinleri için tüm ön türbinlerin uyandırma etkisi göz önüne alındığında, N tane akıntı rüzgâr yönündeki türbinler için akıntı rüzgâr hızının toplam hız formülü Denklem 3.5'te verilmiştir (Rehman ve ark, 2016).

$$u_i = u_0 \left[1 - \sum_{i=1}^N \left(1 - \frac{u_i}{u_0} \right)^2 \right] \quad (3.5)$$

Burada, N , i . türbinin önünde duran türbin sayısını gösterir. u ise ilgili türbinin hızıdır.

Rüzgâr çiftliğindeki tüm rüzgâr türbinlerinin toplam enerji üretimi Denklem 3.6'da gösterilmiştir:

$$P_{tot} = \sum_{i=1}^N 0.3 \times u^3 \quad (3.6)$$

Denklem 3.7'de sunulan ve Mosetti (Mosetti ve ark, 1994) tarafından oluşturulan maliyet modeli, rüzgâr santralindeki rüzgâr türbinlerinin sayısına bağlı olarak geliştirilmiştir. Bu denklemde, bir türbin kurulum maliyetinin 1 olduğu kabul edilmiş ve buna bağlı olarak çok sayıda rüzgâr türbininin kurulması durumunda maksimum maliyet düşüşünün 1/3 oranında olabileceği varsayılmaktadır. Maliyetin geriye kalan 2/3 oranı ise sabit olarak alınmıştır. Rüzgâr çiftliğinin maliyetini belirlemek için bir maliyet modeli seçilmiştir ve Denklem 3.7'de sunulmuştur. Seçilen model önceki çalışmalarda da kullanılmıştır (Mosetti ve ark, 1994, Pookpant ve Ongsakul, 2013, Massan ve ark, 2015).

$$Cost = N \left[\frac{2}{3} + \frac{1}{3} e^{-0.00174N^2} \right] \quad (3.7)$$

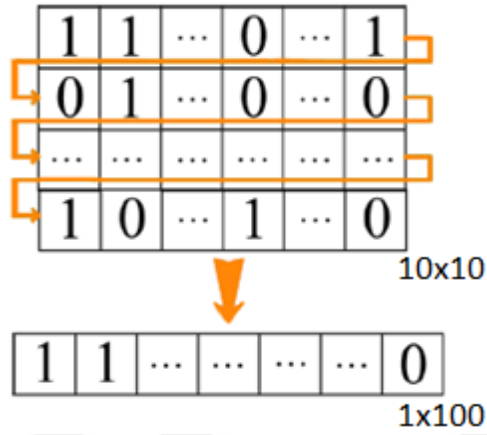
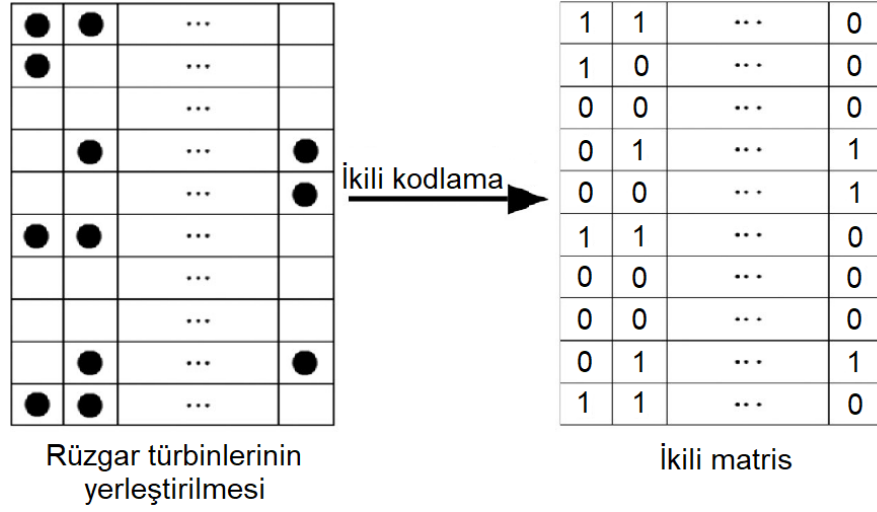
Bir rüzgâr çiftliğinde minimum yatırım maliyetiyle maksimum türbin gücü elde etmek için Denklem 3.8'de verilen amaç fonksiyonu kullanılmaktadır (Mittal, 2010). Bu fonksiyon ile üretilen enerji birimi başına minimum maliyet hesaplanmaktadır.

$$Amaç\ Fonksiyonu = \frac{Cost}{P_{tot}} \quad (3.8)$$

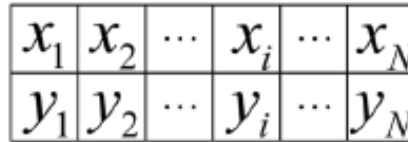
Bu amaç fonksiyonundan elde edilen sonuç ne kadar düşük olursa, üretilen birim enerji başına maliyeti en aza indiği anlamına gelmektedir (Ituarte-Villarreal ve Espiritu, 2011).

3.1.1. İkili, gerçek ve ikili-gerçek kodlama

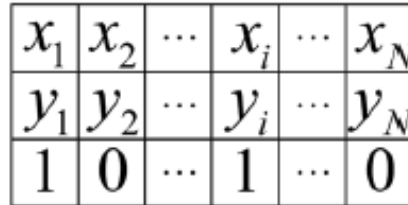
Rüzgâr türbini yerleştirme problemi ikili, gerçek ve ikili-gerçek olmak üzere üç farklı kodlama biçimlerinde modellenmiştir ve Şekil 3.2’de sırasıyla (a), (b) ve (c) olarak gösterilmiştir (Chen ve ark, 2014). İkili kodlama Şekil 3.2(a)’da gösterilen ızgaralar birbirleri ile birleştirilerek oluşturulmaktadır. Her bir ızgarada bir veya sıfır değeri vardır. Izgara içerisinde yer alan “bir” değeri, bir türbinin olduğunu, “sıfır” değeri ise türbin olmadığını ifade eder. Bunun için ikili kodlama ile rüzgâr türbini yerleştirilmesi bir veya sıfırlardan oluşan dizelere dönüştürülür. İkili kodlama ile türbinler ızgaraların sadece ortasına yerleştirilmektedir. Bu işlem ile türbinler sürekli değerler içerisindeki konumlara yerleştirilemediğinden türbin yerleştirme problemi için bir dezavantajdır. Ancak ikili kodlama, türbin sayısının değişimine izin vermektedir. Yani, belirli alana ne kadar türbin yerleşeceğine algoritma karar vermektedir. Gerçek kodlama ise Şekil 3.2(b)’de gösterilmiştir. Gerçek kodlama sayesinde türbinler sürekli koordinatlar ile konumlandırılır ancak türbin sayısı sabit olmalıdır. Gerçek kodlama koordinat yönünden avantaj sağlarken türbin sayısı açısından dezavantaja sahiptir. Bu kodlamada, sürekli koordinat değerleri kullanması avantajından dolayı türbin yerleştirme işlemi için daha iyi bir çözüm elde etmek mümkündür. Ancak gerçek kodlama ile rüzgâr türbinlerinin sayısı sabitlendiğinden, farklı bir yöntem kullanılarak türbin sayısı belirlenebilir ve optimal yerleştirme gerçekleştirilebilir (Chen ve ark, 2014). İkili-gerçek kodlama ise Şekil 3.2(c)’de gösterilmiştir. İkili-gerçek kodlama, ikili kodlama ile gerçek kodlama birleştirilerek elde edilmiştir. Her bir x ve y değeri ile türbinlerin pozisyonu belirlenirken, 1 ve 0 değerleri ile türbin varlığı veya yokluğu tespit edilir. Daha iyi bir yerleştirme için hem konum hem de rüzgâr türbinleri sayısı ayarlanabilmektedir. Bunun için ikili-gerçek kodlamanın getirdiği türbin sayısı belirlememe ve sürekli koordinatlar ile konumlandırma avantajları nedeniyle DA algoritması için bu kodlama kullanılmıştır. YAA ve YOO algoritmaları ise türbin yerleştirme probleminde ikili kodlama olarak kullanılmıştır.



a) İkili kodlama



b) Gerçek kodlama



c) İkili-gerçek kodlama

Şekil 3.2. Kodlama yöntemleri şeması (Chen ve ark, 2014, Yin ve ark, 2016)

3.2. Yapay Alg Algoritması

Son yıllarda geliştirilen YAA sürekli problemlerin çözümü için önerilmiştir (Uymaz ve ark, 2015). Algler çok çeşitli türleri bulunan ökaryot canlılardır. Ayrı çekirdek zarları ve klorofilleri bulunur. Tek hücreli algler mikroalgler olarak adlandırılırken çok hücreli algler makro algler olarak adlandırılmaktadır. YAA ise mikroalglerin karakteristik özellikleri ve yaşam davranışlarından esinlenerek (Uymaz ve ark, 2015) tarafından 2015 yılında biyolojik tabanlı yeni bir metasezgisel metot olarak geliştirilmiştir. Bir alg kolonisi bir grup alg hücresinin birlikte yaşamından meydana gelmektedir. YAA üç bölümden oluşmaktadır (Uymaz, 2016). Bunlar evrimsel süreç, adaptasyon ve helisel hareket olarak isimlendirilmişlerdir. Helisel hareket süreci, alglerin sıvı içindeki helisel hareketleri ve ışığa yakın olma davranışları, evrimsel süreç, alglerin mitoz bölünme ile üremeleri ve adaptasyon süreci ise alglerin bulundukları çevreye adapte olmaları gibi temel özellikleri üzerine modellenmiştir.

Alg popülasyonunu ifade eden alg kolonisi bir diziden oluşmaktadır. Denklem 3.9 ve 3.10'da gösterilmiştir;

$$\text{Alg kolonisinin popülasyonu} = \begin{bmatrix} x_1^1 & \dots & x_1^D \\ \vdots & \ddots & \vdots \\ x_N^1 & \dots & x_N^D \end{bmatrix} \quad (3.9)$$

$$x_i = [x_i^1, x_i^2, \dots, x_i^D] \quad i = 1, 2, \dots, N \quad (3.10)$$

Buradaki x_i^j, i . alg kolonisinin j . boyuttaki alg hücresidir. D ise problem boyutudur ve her bir alg kolonisindeki alg hücresi sayısı problem boyutuna eşittir. Her x_i , çözüm uzayında uygun bir çözümü ifade etmektedir. Alg kolonisindeki tüm alg hücrelerinin çözüm uzayında uygun bir yere toplu şekilde hareket ettiği düşünülmektedir. Bir alg kolonisi ideal çözüme ulaştığında optimum sonuç elde edilmiş olur.

3.2.1. Evrimsel süreç

Yapay alg hücresi yeterli ışık aldığında gelişir, büyür ve gerçek mitoz bölünmeye benzer şekilde kendisine benzeyen iki yapay alg hücresi oluşur. Eğer yapay alg hücresi yeterli ışık alamazsa bir süreliğine yaşar daha sonra ise ölür.

Alg kolonisinin büyüme hız denklemi Denklem 3.11'de gösterilmiştir;

$$\mu = \frac{\mu_{max}S}{K_s+S} \quad (3.11)$$

Biyolojik büyüme yaygın olarak monod modeli ile tanımlanır. Bu model spesifik büyüme hızı substrat konsantrasyonunun (S) bir fonksiyonu olarak tanımlanır. Burada μ_{max} maksimum spesifik büyüme hızı (1/zaman) ve K_s substrate yarı doygunluk sabiti olarak tanımlanır (kütle/hacim). Monod modelinde $t + 1$ anında i . alg kolonisinin büyüklüğü Denklem 3.12 ve 3.13'te verilmiştir;

$$\mu_i^t = \frac{\mu_{max}x f^t(x_i)}{\left(\frac{G_i^t}{2}\right) + f^t(x_i)} \quad (3.12)$$

$$G_i^{t+1} = G_i^t + \mu_i^t G_i^t, \quad i = 1, 2, \dots, N \quad (3.13)$$

Burada G_i^t , t anında i . alg kolonisinin büyüklüğü, N ise popülasyondaki alg koloni sayısıdır. $f^t(x_i)$ ise i . yapay alg kolonisinin besin konsantrasyonu yani t anındaki uygunluk değerini ifade etmektedir.

Başlangıçta tüm alg kolonilerinin büyüklükleri 1 olarak kabul edilmiştir. Helisel hareket ile iyi çözümler sağlamaktadır. Alg kolonisi besin miktarı yüksek olduğundan daha fazla büyümektedir. Tüm alg kolonisi bir tam hareketini tamamladıktan sonra evrimsel süreç devreye girer. Evrimsel süreçte en küçük alg kolonisinin ölen her bir alg hücresinin yerine en büyük alg kolonisinin bir alg kolonisinin bir alg hücresi kopyalanır. Bu süreç Denklem 3.14, 3.15 ve 3.16'daki gibi ifade edilir;

$$biggest^t = \arg \max\{size(x_i^t)\}, \quad i = 1, 2, \dots, N \quad (3.14)$$

$$smallest^t = \arg \min\{size(x_i^t)\}, \quad i = 1, 2, \dots, N \quad (3.15)$$

$$smallest_m^{t+1} = biggest_m^t, \quad m = 1, 2, \dots, D \quad (3.16)$$

Burada, D problem boyutu, *biggest* en büyük alg kolonisi ve *smallest* en küçük alg kolonisini ifade eder.

YAA'da alg kolonileri t anındaki büyüklüklerine (G) göre sıralanır. Rastgele seçilen bir boyutta en küçük alg kolonisinin bir hücresi ölür ve en büyük alg kolonisinin bir hücresi onun yerine kopyalanır.

3.2.2. Adaptasyon

Hayatta kalmayı başaran ancak bulunduğu ortamda yeterince büyüemeyen alg kolonisi o ortama adapte olmaya çalışır. Burada en büyük alg kolonisine benzeme süreci başlamış olur. Bu süreç alg kolonilerinin açlık seviyelerine göre belirlenir ve ilk başta açlık seviyesi sıfır olarak kabul edilir. Her bir alg kolonisi helisel hareket sonucu çözüm uzayında ya daha iyi bir çözüme ya da daha kötü bir çözüme gider. Daha iyi bir sonuç buluyorsa alg kolonisinin açlık seviyesinde bir değişiklik olmaz ancak daha iyi bir çözüm bulamayan alg kolonisinin açlık seviyesi bir artar. Alg kolonisinin her bir helisel hareketi sonucunda açlık seviyesi en yüksek olan alg kolonisi adaptasyon işlemine tabi tutulur. Adaptasyon süreci A_p parametresi ile belirlenir ve $[0,1]$ aralığında sabit bir sayıdır. 0 ile 1 arasında rastgele üretilen gerçek değerli bir sayı A_p parametresinden küçük ise adaptasyon süreci gerçekleşir. Açlık seviyesi Denklem 3.17 ve 3.18'deki gibi ifade edilir;

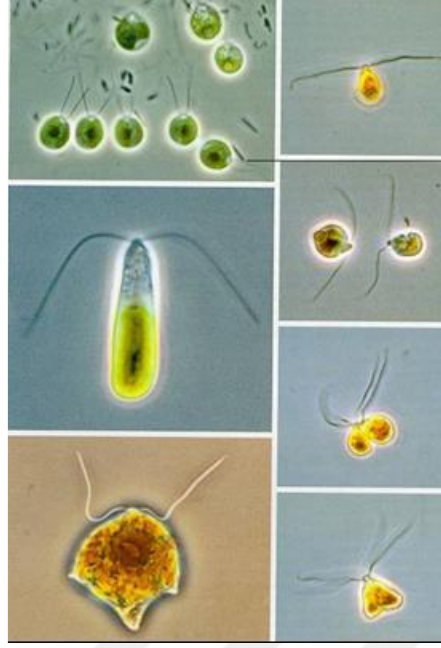
$$starving^t = \arg \max\{starvation(x_i)\}, \quad i = 1, 2, \dots, N \quad (3.17)$$

$$starving^{t+1} = starving^t + (biggest^t - starving^t) \times rand \quad (3.18)$$

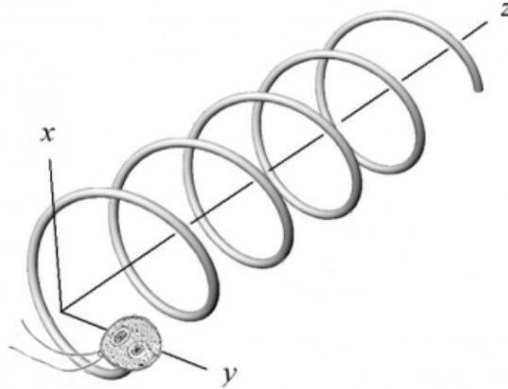
Burada, $starvation(x_i)$ popülasyondaki alg kolonilerinin açlık seviyelerini, $starving^t$ t anındaki açlık değeri en yüksek olan alg kolonisini, $biggest^t$ t anındaki en büyük alg kolonisini ve $rand$ 0 ile 1 arasında rastgele üretilen gerçek değerli bir sayıyı ifade etmektedir.

3.2.3. Helisel hareket

Alg hücreleri ve alg kolonileri genellikle yüzerler ve su yüzeyine yakın olmaya çalışırlar. Bunun sebebi ise alg hücrelerinin ve alg kolonilerinin hayatta kalabilmeleri için yeterli ışık almaları gerekmektedir ve yeteri kadar ışık orada mevcuttur. Algler kamçıları sayesinde sıvı içerisinde helisel hareket yaparak yüzerler. Sıvı içerisindeki bir alg hücresinin ileri hareketini yerçekimi ve viskoz sürtünmesi olumsuz etkilemektedir. Şekil 3.3 ve 3.4'te alg kamçıları ve helisel hareket gösterilmektedir.



Şekil 3.3. Alglerin kamçıları (Lund ve ark, 1996)



Şekil 3.4. Alglerin hareket deseni (Lund ve ark, 1996)

YAA'da alg kolonilerini gerçekte olduğu gibi harekete geçiren helisel harekettir. Bu hareketi engelleyen yer çekimi 0 (sıfır) olarak kabul edilmiştir ve viskoz sürtünmesi de kesme kuvveti olarak gösterilmiştir. Viskoz sürtünmesi alg kolonisinin büyüklüğü ile orantılıdır. Hesaplama alg kolonisi küre ve büyüklüğü (G) hacim olarak kabul edildiğinde sürtünme yüzeyi yarım kürenin yüzey alanı (T) olmaktadır. Hacmi bilinen bir alg kolonisinin yarıçapı, hacim formülünden elde edilerek Denklem 3.19'daki gibi kullanılır.

$$T(x_i) = 2\pi r_i^2$$

$$T(x_i) = 2\pi \left(\sqrt[3]{\frac{3G_i}{4\pi}} \right)^2 \quad (3.19)$$

Burada $T(x_i)$, i . alg kolonisinin sürtünme yüzeyidir. G_i , i . alg kolonisinin büyüklüğünü ifade eder.

Alg kolonisinin helisel hareketi için rastgele üç boyut belirlenir. Bu boyutlardan biri doğrusal hareketi sağlarken diğer iki boyut ise açısıl hareketi sağlamaktadır. Doğrusal hareket tek boyutlu problemler için kullanılmaktadır. Alg kolonisi veya alg hücresi doğrusal hareketi sağlamaktadır ve Denklem 3.20 kullanılır. İki boyutlu problemlerde alg kolonisi hareketi sinüzoidaldir. Bundan dolayı Denklem 3.21 ve Denklem 3.22 kullanılır. Problemin boyutunun üç ve yukarısı olması durumunda ise alg kolonisinin hareketi helisel olup Denklem 3.20, 3.21 ve 3.22 kullanılır. Alg kolonisinin hareketinin büyüklüğünü sürtünme yüzeyi ve ışık kaynağına olan uzaklığı belirlemektedir.

$$x_{im}^{t+1} = x_{im}^t + (x_{jm}^t - x_{im}^t)(\Delta - T^t(x_i))\rho \quad (3.20)$$

$$x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t)(\Delta - T^t(x_i))\cos\alpha \quad (3.21)$$

$$x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t)(\Delta - T^t(x_i))\sin\beta \quad (3.22)$$

Burada x_{im}^t , x_{ik}^t ve x_{il}^t i . alg kolonisinin t anında rasgele seçilen üç alg hücresi; x_{jm}^t , x_{jk}^t ve x_{jl}^t ışık kaynağı olarak belirlenen popülasyondaki farklı bir alg kolonisinin seçilen boyutlarındaki alg hücreleri; $\alpha, \beta \in [0, 2\pi]$; $\rho \in [-1, 1]$; Δ kesme kuvveti katsayısı; $T^t(x_i)$ ise i . alg kolonisinin t anındaki sürtünme yüzey alanıdır.

Şekil 3.5 ve 3.6'da sırasıyla YAA'nın sözde kodu ve akış şeması verilmiştir.

Amaç fonksiyon $f(x)$, $x = (x_1, x_2, \dots, x_d)$
 Algoritma parametrelerini belirle (kesme kuvveti Δ , enerji kaybı e ve Adaptasyon A_p)
 Bir n alg kolonisi popülasyonunu rastgele çözümlerle başlat
 n alg kolonilerinin büyüklüğünün değerlendirilmesi (G)
While ($t < \text{MaxCalculation}$)
 Alg kolonilerinin enerjilerini (E) ve sürtünme yüzeyini (T) hesapla
For $i = 1: n$
 Açlık true dir
While ($E(x_i) > 0$)
 Turnuva seçim yöntemi ile bir j kolonisi seç
 Sarmal harekete rastgele üç boyut seçin, k, l ve m

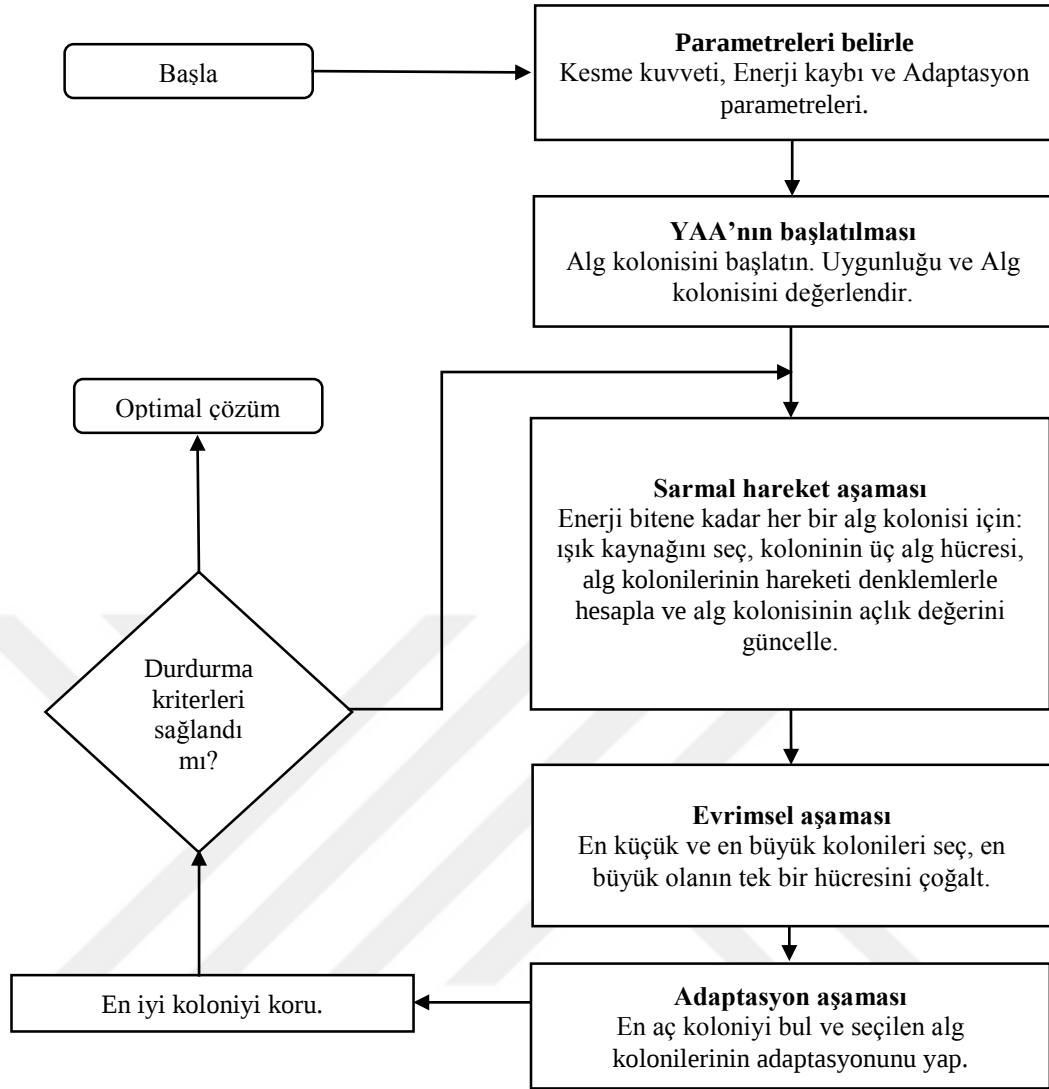
$$x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t)(\Delta - T^t(x_i))\cos\alpha$$

$$x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t)(\Delta - T^t(x_i))\sin\beta$$

$$x_{im}^{t+1} = x_{im}^t + (x_{jm}^t - x_{im}^t)(\Delta - T^t(x_i))p$$
 α ve β $[0, 2\pi]$ aralığında rastgele açılardır. p , $[-1, 1]$ aralığında rastgele a sayısıdır.
 Yeni çözümü değerlendir
 $E(x_i) = E(x_i) - \left(\frac{e}{2}\right)$ hareketten kaynaklı enerji kaybı
 if yeni çözüm daha iyi olursa i 'yi güncelle ve açlık false dur.
 else $E(x_i) = E(x_i) - \left(\frac{e}{2}\right)$ metabolizmanın neden olduğu enerji kaybı **end if**
end while
 if açlık true dir, açlık artar $A(x_i)$ **end if**
end For
 Popülasyon büyüklüğü (G) değerlendir
 Üreme için bir boyut seçin, r
 $\text{smallest}_r^t = \text{biggest}_r^t$
 if $\text{rand} < A_p$

$$\text{starving}^{t+1} = \text{starving}^t + (\text{biggest}^t - \text{starving}^t) \times \text{rand}$$
end if
 Mevcut en iyi çözümü bul
end While

Şekil 3.5. YAA'nın sözde kodu



Şekil 3.6. YAA'nın akış şeması

3.3. İkili Yapay Alg Algoritması

Bu çalışmada sürekli problemlerin çözümünde kullanılan YAA, ikili rüzgâr türbini yerleştirme problemine doğrudan uygulanamayacağı için ikili hale getirilmiştir. YAA'ya 14 farklı transfer fonksiyonu uygulanmıştır ve her biri İYAA1-İYAA14 olarak isimlendirilmiştir. Bu yöntemlerden optimum türbin yerleştirme sonucunu veren ikili algoritma genel bir isim olarak İYAA şeklinde adlandırılmıştır.

Transfer fonksiyonları alg'i ikili arama uzayında hareket etmeye zorlar ve bu transfer fonksiyonlarının aralığı $[0,1]$ olmalıdır (Rashedi ve ark, 2010). Çünkü bunlar bir alg'in konumunu değiştirme ihtimalini temsil eder. Alg'in her bir enerji kaybından sonra muhtemelen en iyi çözümden uzaktır, bu yüzden pozisyonlarını bir sonraki iterasyonda

değiştirmelidirler. Çalışmada kullanılan transfer fonksiyonları Tablo 3.1’de verilmiştir. Ayrıca bu transfer fonksiyonlarının grafiksel gösterimi Şekil 3.7 ve 3.8’de verilmiştir.

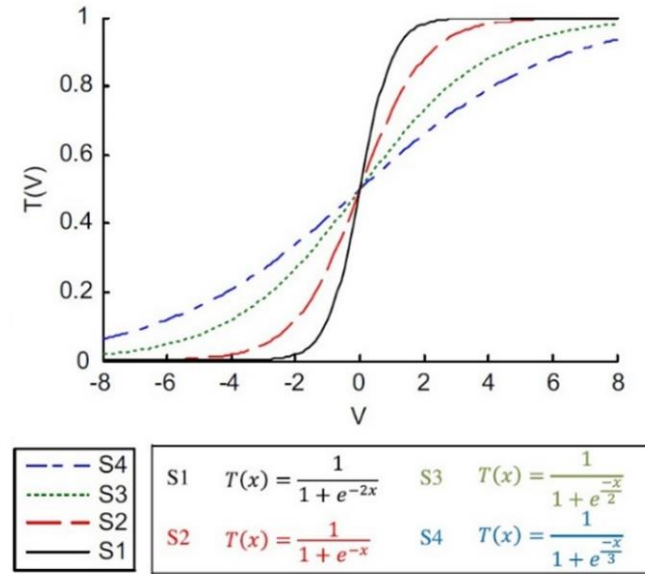
Tablo 3.1. Transfer fonksiyonlar ailesi

S-şekilli transfer fonksiyonları ailesi	
Adı	Transfer fonksiyonları
S1 (İYAA1)	$T(x) = \frac{1}{1 + e^{-2x}}$
S2 (İYAA2)	$T(x) = \frac{1}{1 + e^{-x}}$
S3 (İYAA3)	$T(x) = \frac{1}{1 + e^{(-\frac{x}{2})}}$
S4 (İYAA4)	$T(x) = \frac{1}{1 + e^{(-\frac{x}{3})}}$
V-şekilli transfer fonksiyonları ailesi	
Adı	Transfer fonksiyonları
V1 (İYAA5)	$T(x) = \left \operatorname{erf}\left(\frac{\sqrt{\pi}}{2}x\right) \right = \left \frac{\sqrt{2}}{\pi} \int_0^{(\sqrt{\pi/2})x} e^{-t^2} dt \right $
V2 (İYAA6)	$T(x) = \tanh(x) $
V3 (İYAA7)	$T(x) = \left (x)/\sqrt{1+x^2} \right $
V4 (İYAA8)	$T(x) = \left \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x\right) \right $
Mod transfer fonksiyonları ailesi	
Adı	Transfer fonksiyonları
Modulo base A (İYAA9)	binaryVector=mod(floor(continuousVector),2);
Modulo base B (İYAA10)	binaryVector=mod(round(mod(continuousVector),2),2);
Hibrit transfer fonksiyonu ailesi	
Adı	Transfer fonksiyonları
Ceil (İYOO11)	binaryVector=mod(ceil(continuousVector),2); if rand < 0.5
Ceil-Round (İYOO12)	binaryVector=mod(ceil(continuousVector),2); else
Ceil-Floor (İYOO13)	binaryVector=mod(round(continuousVector),2); end if rand < 0.5
Round-Floor (İYOO14)	binaryVector=mod(ceil(continuousVector),2); else
	binaryVector=mod(floor(continuousVector),2); end if rand < 0.5
	binaryVector=mod(round(continuousVector),2); else
	binaryVector=mod(floor(continuousVector),2); end

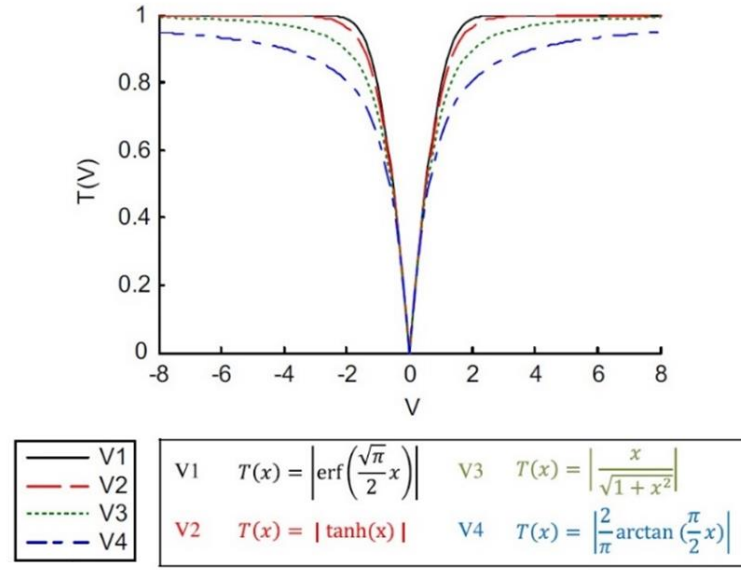
Tablo 3.1’de gösterildiği gibi x ’in katsayısını idare eden dört transfer fonksiyonu verilmiştir. Şekil 3.7’de S3 ve S4’ün doygunluklarının S2’den daha erken başlarken, S1’in hız değeri S2’den çok daha yüksek olduğu için S1’in keskin bir şekilde arttığı ve

doygunluğa eriştiği görülmektedir. Konum vektörlerinin değerlerinin değiştirilme olasılığı, bu transfer fonksiyonlarının eğimi arttıkça yükselmektedir. Bu nedenle S1, hızın aynı değeri için en yüksek olasılığı, S4 ise en düşük değeri sağlamaktadır. İYAA'nın performansına etkisini araştırmak için farklı eğim ve doygunluklara sahip dört farklı transfer fonksiyonu seçilmiştir. Bu eğrilerin şekillerinden dolayı, S-şeklinde transfer fonksiyonları olarak adlandırılırlar. Grup olarak "S-şekilli transfer fonksiyon ailesi" adı verilmektedir (Mirjalili ve Lewis, 2013).

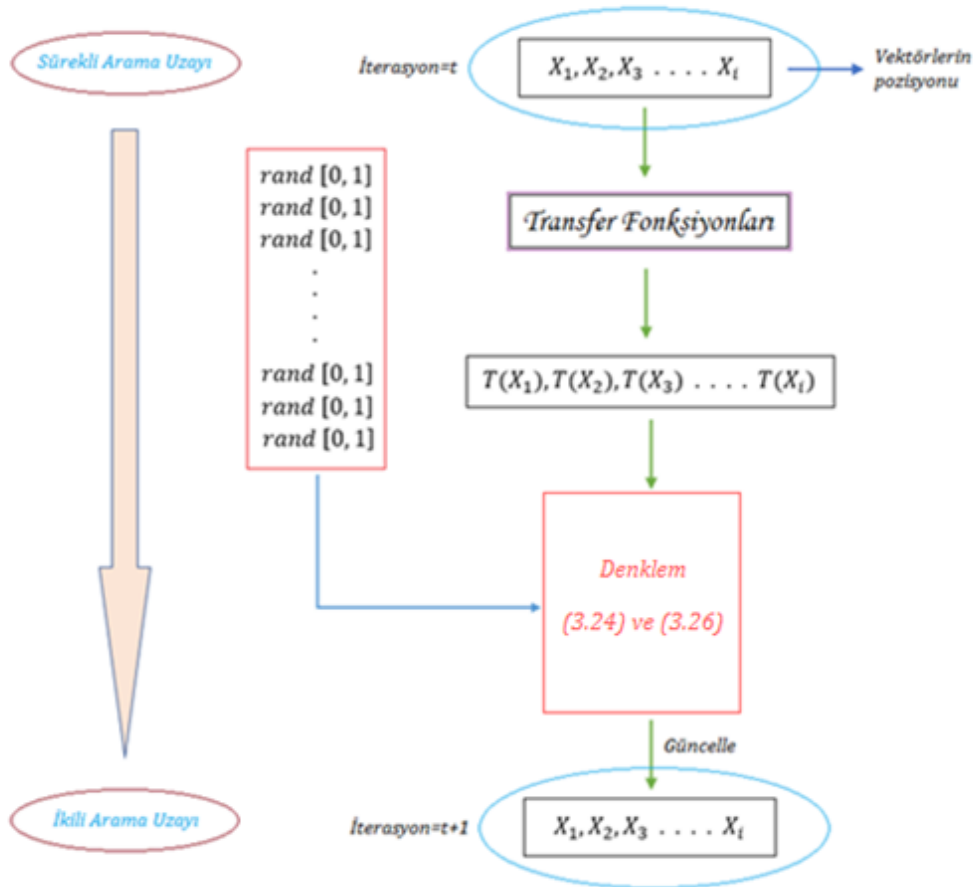
V1, V2, V3 ve V4 adlı dört V-şekilli transfer fonksiyonu, çeşitli matematik denklemler kullanılarak Tablo 3.1'de sunulmuştur. Şekil 3.8'de V1'in doyumunun, V2'ye kıyasla daha düşük hıza sahip mutlak değerlerle başladığı görülmektedir. Bu, V1'in, aynı değerler için V2'den daha yüksek ihtimalle alglerin konumlarının değişmesine olanak sunmaktadır. Buna karşılık, V3 ve V4 transfer fonksiyonlarının doygunlukları, V1 ve V2'den sonra başlamaktadır ve aynı hızlar için daha az değişim olasılığı sağlamaktadır. Bu eğriler şekilleri nedeniyle, V şeklindeki transfer fonksiyonları olarak adlandırılırlar. Grup olarak "V-şekilli transfer fonksiyon ailesi" adı verilmektedir. Bu yöntemin avantajı, bu transfer fonksiyonlarının algere 0 veya 1 değerlerini almaya zorlamamasıdır. Başka bir deyişle, bu fonksiyonlar hız değerleri düşük olduğunda algleri mevcut konumlarında tutarken hız değerleri yüksek olduğunda tamamlayıcılarına geçme işlemini gerçekleştirirler (Mirjalili ve Lewis, 2013).



Şekil 3.7. S-şekilli ailesinin grafiği



Şekil 3.8. V-şekilli ailesinin grafiği



Şekil 3.9. Sürekli arama uzayından ikili arama uzayına geçiş

Tez çalışmasında S-şekilli transfer fonksiyonu ailesi, V-şekilli transfer fonksiyonu ailesi, mod-A, mod-B ve Hibrit transfer fonksiyonları kullanılarak 14 farklı ikili İYAA oluşturulmuştur. Transfer fonksiyonları kullanılarak konum vektörleri 0 veya 1 değerini alacak şekilde ikili uzaya dönüştürülmüştür. Bu durum Şekil 3.9’da simüle edilmiştir. Denklem 3.23 ile Denklem 3.24’te S-şekilli transfer fonksiyonu ailesi üyesi olan S1 (İYAA1) fonksiyonu örneği verilmiştir (Mirjalili ve Lewis, 2013). Denklem 3.25 ile Denklem 3.26’da V-şekilli transfer fonksiyonu ailesi üyesi olan V3 (İYAA7) fonksiyonu örneği verilmiştir. Denklem 3.27 ile Denklem 3.28’de Modulo base A (İYAA9) transfer fonksiyonu örneği verilmiştir. Denklem 3.29 ile Denklem 3.30’da Modulo base B (İYAA10) transfer fonksiyonu örneği verilmiştir.

$$T\left(X_i^k(t)\right) = \frac{1}{1+e^{-2X_i^k(t)}} \quad (3.23)$$

Burada $X_i^k(t)$, i. alg’in pozisyonunun t iterasyonundaki k. boyutundaki değerdir.

S-şekilli transfer fonksiyonu ailesi için, konumları olasılık değerlerine göre çevirdikten sonra, konum vektörleri Denklem 3.24’deki gibi olasılık değerleri ile güncellenecektir;

$$X_i^k(t+1) = \begin{cases} 0 & \text{if } rand < T\left(X_i^k(t)\right) \\ 1 & \text{if } rand \geq T\left(X_i^k(t)\right) \end{cases} \quad (3.24)$$

Burada $X_i^k(t+1)$, i. alg’in pozisyonunun t+1 iterasyonundaki k boyutundaki ikili değerdir.

S – şekilli ailesi transfer fonksiyonlarına göre önerilen yöntem şu şekilde çalışmaktadır:

1. $X_i^k(t)$ değeri transfer fonksiyonuna gönderilir.
2. $X_i^k(t)$ değerine göre $S(X_i^k(t))$ fonksiyonu hesaplanır.
3. Her bir $X_i^k(t)$ için $[0, 1]$ arasında rastgele bir değer üretilir.
4. Denklem 3.24’e göre geçici aday çözümdeki $X_i^k(t+1)$ değerine 0 ya da 1 olarak atama yapılır.

$$T\left(X_i^k(t)\right) = |(X_i^k(t)/\sqrt{1+x^2})| \quad (3.25)$$

V-şekilli transfer fonksiyonu ailesi transfer fonksiyonları yeni konum güncelleme kurallarına ihtiyaç duymaktadır ve bunun için Denklem 3.26 kullanılmaktadır.

$$X_i^k(t+1) = \begin{cases} (X_i^k(t))^{-1} & \text{if } rand < T(X_i^k(t)) \\ (X_i^k(t)) & rand \geq T(X_i^k(t)) \end{cases} \quad (3.26)$$

Burada $(X_i^k(t))^{-1}$ ise $X_i^k(t)$ 'nin tamamlayıcısıdır.

V – şekilli transfer fonksiyonu ailesi üyesi fonksiyonlara göre önerilen yöntem şu şekilde çalışmaktadır:

1. $X_i^k(t)$ değeri transfer fonksiyonuna gönderilir.
2. $X_i^k(t)$ değerine göre $T(X_i^k(t))$ fonksiyonu hesaplanır.
3. Her bir $X_i^k(t)$ için $[0, 1]$ arasında rastgele bir değer üretilir.
4. $X_i^k(t)$ değerinin ikili değeri elde edilir.
5. Denklem 3.26'ya göre ikili $X_i^k(t)$ değeri konum değiştirme yapar veya mevcut konumunda kalır. Bulunan değer geçici aday çözümdeki $X_i^k(t+1)$ değerine atanır.

Modulo base A (İYAA9) sürekli çözümleri ikili versiyonuna dönüştürmek için kullanılır. Bu dönüşüm Denklem 3.27'te sunulmuştur.

$$BS_{i,j} = \text{mod}(\text{abs}(\lfloor X_{i,j} \rfloor), 2) \quad (3.27)$$

Burada, $BS_{i,j}$ $X_{i,j}$ 'den elde edilen ikili bir çözüm olduğunda, $\lfloor \rfloor$ operatörü aşağı yuvarlama işlemin gerçekleştirir daha sonra ise $\text{abs } X_{i,j}$ 'nin mutlak değerini elde etmek için kullanılmaktadır. $BS_{i,j}$ hesaplandığı zaman her şeyden önce yuvarlama operatörü $X_{i,j}$ 'ye uygulanır, ardından $X_{i,j}$ 'nin mutlak değeri elde edilir. Son işlem için elde edilen değere modul base A uygulanır (Koç, 2016). Sürekli değer ilk önce 2'ye göre mod alınır ve kalan değerın mutlak değeri alınır. Sonuç olarak ikili değer elde edilir. Bu dönüşüme göre ikili değer elde edilirken 2'ye bölünme sonucu geriye kalanın mutlak değeri 0 ile 0.9999 arasında ise ikili sayı 0 değerini alır, eğer kalanın mutlak değeri 1 ile 1.9999 arasında olursa ikili sayı 1 olur. Denklem 3.28 buna bir örnek olarak verilmiştir (Kiran, 2015).

$$BS_{i,j} = \text{floor}([-3.856 \bmod 2]) \quad (3.28)$$

$$\begin{aligned}
&= \text{floor}([-1.856]) \\
&= \text{floor}([1.856]) = 1
\end{aligned}$$

Modulo base B (İYAA10) sürekli çözümleri ikili versiyonuna dönüştürmek için kullanılır. Bu dönüşüm Denklem 3.29'da sunulmuştur (Kiran, 2015).

$$BS_{i,j} = \text{round}(|X_{i,j} \bmod 2|) \bmod 2 \quad (3.29)$$

Denklem 3.29'deki ikili dönüştürme işlemi Denklem 3.30 örneği ile açıklanmaktadır.

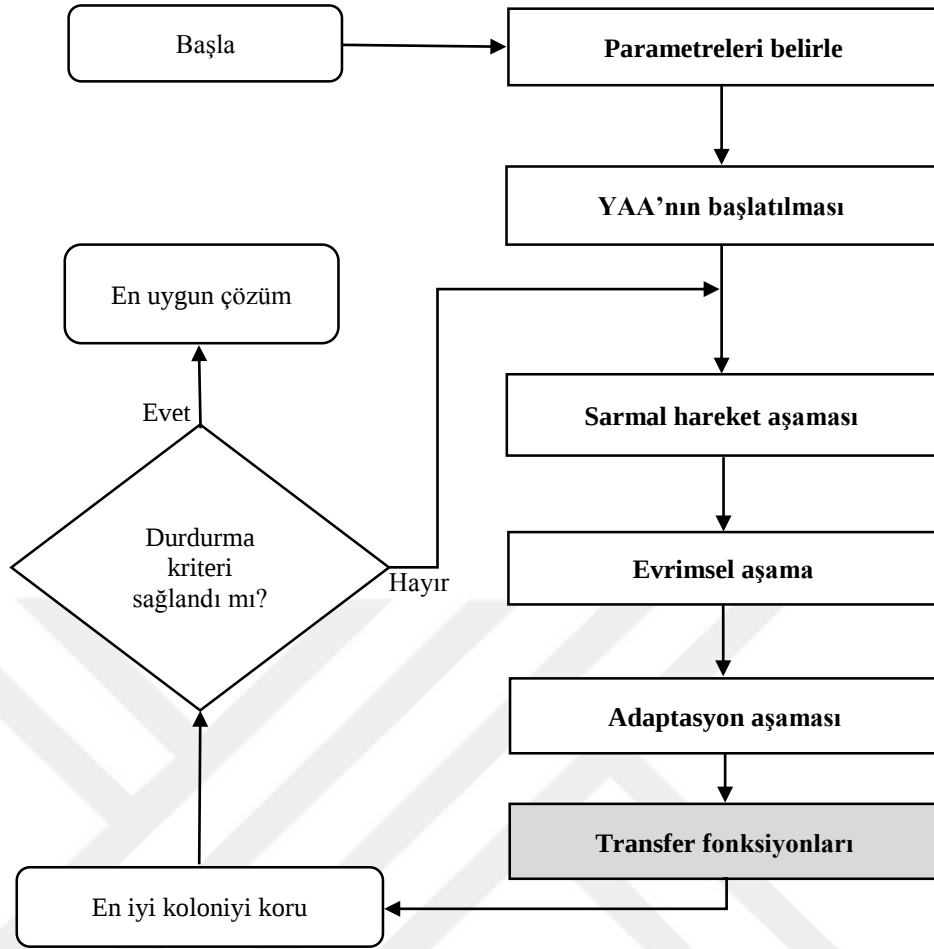
$$BS_{i,j} = \text{round}(|-3.856 \bmod 2|) \bmod 2 \quad (3.30)$$

$$\begin{aligned}
&= \text{round}(|-1.856|) \bmod 2 \\
&= \text{round}(1.856) \bmod 2 \\
&= 2 \bmod 2 = 0
\end{aligned}$$

Elde edilen sürekli değer, ikiye bölünüp kalan değer mutlak değeri alınır. Sonra Denklem 3.28'deki örnekten farklı olarak sayı buçuklu değer üzerindeyse veya altında ise sayıyı yuvarlama işlemi gerçekleştirilip sayı 2'ye bölünür ve elde edilen sonuç ikili değerdir. Kısaca, kalanın mutlak değeri 0 ile 0.4999 veya 1.5 ile 1.9999 arasında ise ikili sayı 0 olarak bulunur. Eğer kalanın mutlak değeri 0.5 ile 1.4999 arasında ise ikili sayı 1 olarak elde edilir.

YAA sürekli optimizasyon algoritması olup ikili rüzgâr türbini problemine uygulanamamaktadır. YAA sürekli arama uzayını bozmadan, ilk önce yukarıdaki transfer fonksiyonlarını kullanarak, YAA algoritmasının hesapladığı vektör, ikili çözüm vektörüne dönüştürülür ve ardından bu probleme uygulanır.

Tez çalışmasında YAA'ya transfer fonksiyonu uygulanarak önerilen İYAA'nın akış şeması Şekil 3.10'da gösterilmiştir.



Şekil 3.10. İYAA'nın Akış Şeması

3.4. Yabani Ot Optimizasyon Algoritması

YOO algoritması, yabani otların doğal davranışını taklit eden, biyolojik olarak esinlenmiş, sayısal bir optimizasyon algoritmasıdır (Bharadwaj ve Siddhartha, 2016). YOO algoritması Mehrabian ve Lucas (Mehrabian ve Lucas, 2006) tarafından 2006 yılında yeni bir metasezgisel algoritma olarak geliştirilmiştir. Yapının sadeliği, daha az parametre gerektiren ve çok güçlü sağlamlık gibi birçok avantajı olan YOO genel, çok boyutlu, doğrusal ve doğrusal olmayan optimizasyon problemlerini çözmek için kullanılır (Zhou ve ark, 2015). Algoritma basit olmakla birlikte, bir yabani ot kolonisinde tohumlama, büyüme ve rekabet gibi temel özellikleri kullanarak optimal çözüme yakınlaşmada etkili olduğu söylenmiştir (Mehrabian ve Lucas, 2006). Yabani otlar büyüme ve üreme için kolonileşme ve uygun yer bulma eğilimindedirler (Ahmed ve ark, 2014). Popülasyon yabani otların toplam sayısından oluşmaktadır. Her otun uygunluğu, optimal çözüme ne kadar yakın veya uzak olduğu ile belirlenir. Her bir otun çevresinde, yüksek kondisyon değerine sahip yabani otların, düşük kondisyon değerine sahip yabani

otlardan daha fazla tohum ürettikleri bir dizi tohum üretilir (Saravanan ve ark, 2014). Üretilen tohumlar normal olarak, ortalama sifıra eşit ve farklı varyansa sahip ana ot etrafına dağıtılır. Popülasyon, daha sonra tüm yabancı otları ve üretilen tohumları içerecek şekilde güncellenir. Durdurma kriteri sağlanana kadar bu işlem devam eder (Nagib ve ark, 2016). YOO algoritması dört temel aşamadan oluşmaktadır. Bunlar başlangıç popülasyonu, yeniden üretme, mekânsal dağılım ve rekabetçi dışlamadır (Xing ve Gao, 2014, Zhou ve ark, 2014).

3.4.1. Başlangıç popülasyonu

İlk olarak yabancı otlar rastgele üretildiğinden popülasyonu D boyutlu çözüm uzayına rastgele bir şekilde dağıtılır. Her otun uygunluğu hesaplanır. En iyi ve en kötü uygunluk değerleri belirlenir.

3.4.2. Yeniden Üretme

Her yabancı ot tarafından üretilen tohumların sayısı uygunluğa bağlı olarak değerlendirilir. Her tohumun çoğalma şansı vardır ve çoğalma oranı, en iyi ile en kötü uyum tohumuna göre maksimumdan minimum değere göre değişir. Bu tohumlar daha sonra yeni birimler üretebilen yabancı bitkilere dönüşürler. Tohum üreten yabancı ot formülü Denklem 3.31’de yer almaktadır (Roy ve ark, 2011).

$$ot_n = \frac{f - f_{worst}}{f_{best} - f_{worst}} (S_{max} - S_{min}) + S_{min} \quad (3.31)$$

Burada, f dikkate alınan ot’un uygunluğudur. f_{worst} ve f_{best} sırasıyla mevcut popülasyonun en kötü ve en iyi uygunluğudur. S_{min} ve S_{max} sırasıyla minimum ve maksimum tohum sayılarıdır.

3.4.3. Mekânsal dağılım

Üretilen tohumlar, normalde sifıra eşit bir ortalamaya sahip ama değişken bir varyans ile rasgele sayılar dağıtılarak D boyutlu arama uzayı üzerinde rastgele dağıtılır. Bu tohumların rastgele dağıtılması ile ebeveyn bitkisine yakın olması sağlanır. Bununla birlikte rastgele fonksiyonun standart sapması (σ), önceden tanımlanmış bir başlangıç

değerinden (σ_{init}) her iterasyonda bir nihai değere (σ_{final}) inecektir. Formülü Denklem 3.32’de verilmiştir (Roshanaei ve ark, 2009, Karimkashi ve Kishk, 2010);

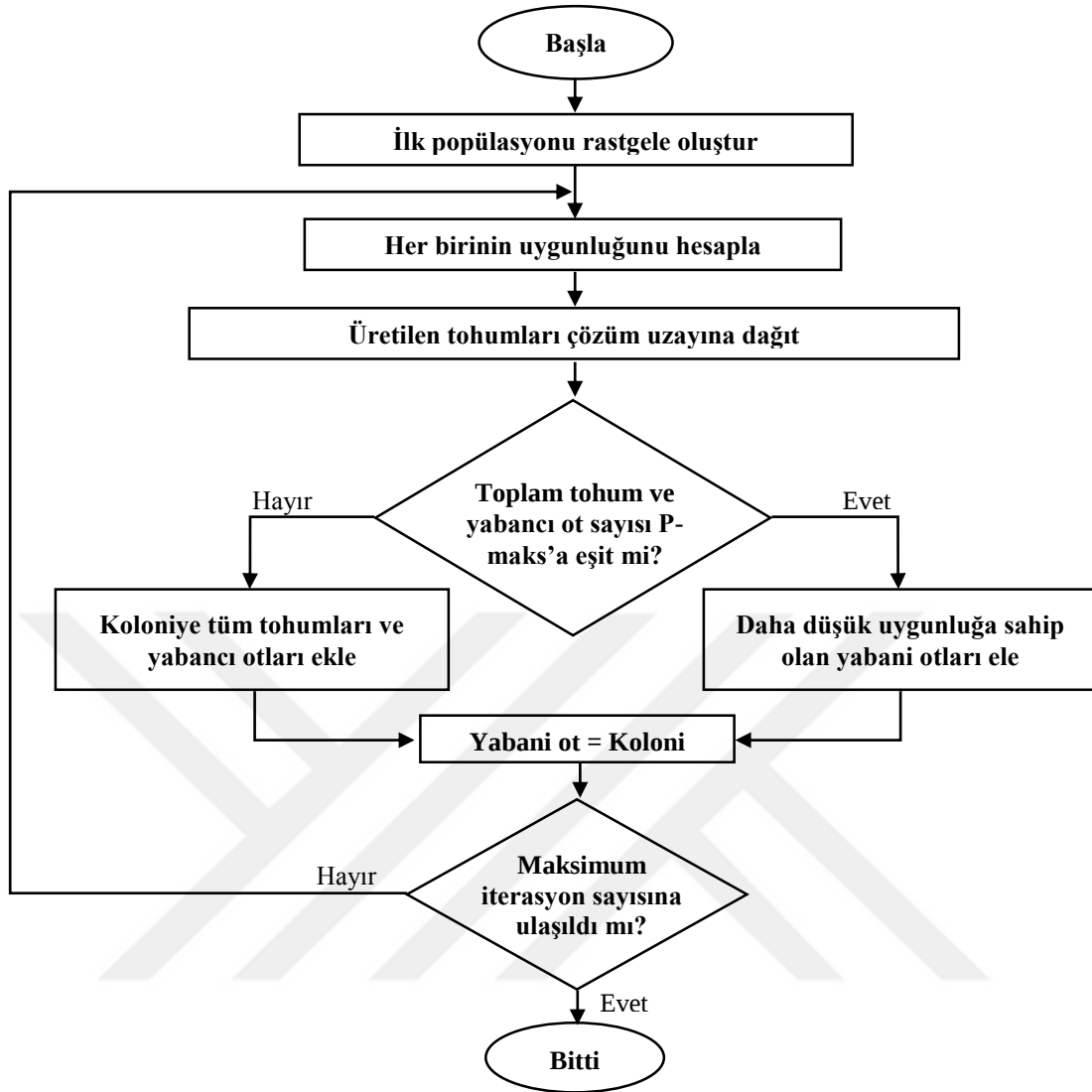
$$\sigma_{cur} = \frac{(iter_{max}-iter)^n}{(iter_{max})^n} (\sigma_{init} - \sigma_{final}) + \sigma_{final} \quad (3.32)$$

Burada, $iter_{max}$ maksimum iterasyon sayısıdır, σ_{cur} şimdiki zaman adımında standart sapmadır, σ_{init} ilk standart sapmadır, σ_{final} nihai standart sapmadır ve n ise modülasyon endeksidir.

3.4.4. Rekabetçi dışlama

Belirli bir iterasyondan sonra, bir koloni içindeki yabancı otların sayısı hızlı üreme ile maksimum popülasyon sayısını geçecektir. Bu sürede, her yabancı ot’a tohum üretim izni verilmiştir. Üretilen tohumlar daha sonra arama uzayına yayılmaya bırakılır. Bütün tohumlar arama uzayındaki yerlerini bulduklarında, ebeveynleriyle birlikte (yabancı otların bir kolonisi olarak) sıralanırlar. Daha sonra, düşük uygunluğa sahip yabancı otlar, bir kolonide izin verilen maksimum popülasyona ulaşmak için elimine edilir. Böylelikle, yabancı otlar ve tohumlar birlikte sıralanır ve daha iyi bir uygunluğa sahip olanlar hayatta kalır ve çoğalmasına izin verilir. Bu işlem maksimum iterasyon veya diğer durdurma kriterlerine ulaşıncaya kadar devam eder. En iyi uygunluğa sahip olan yabancı ot en uygun çözüm olarak seçilir.

YOO’nun akış şeması Şekil 3.11’de verilmiştir.



Şekil 3.11. YOO'nun akış şeması (Moallem ve Razmjoooy, 2012, Ahmed ve ark, 2014)

3.5. İkili Yabani Ot Optimizasyonu Algoritması

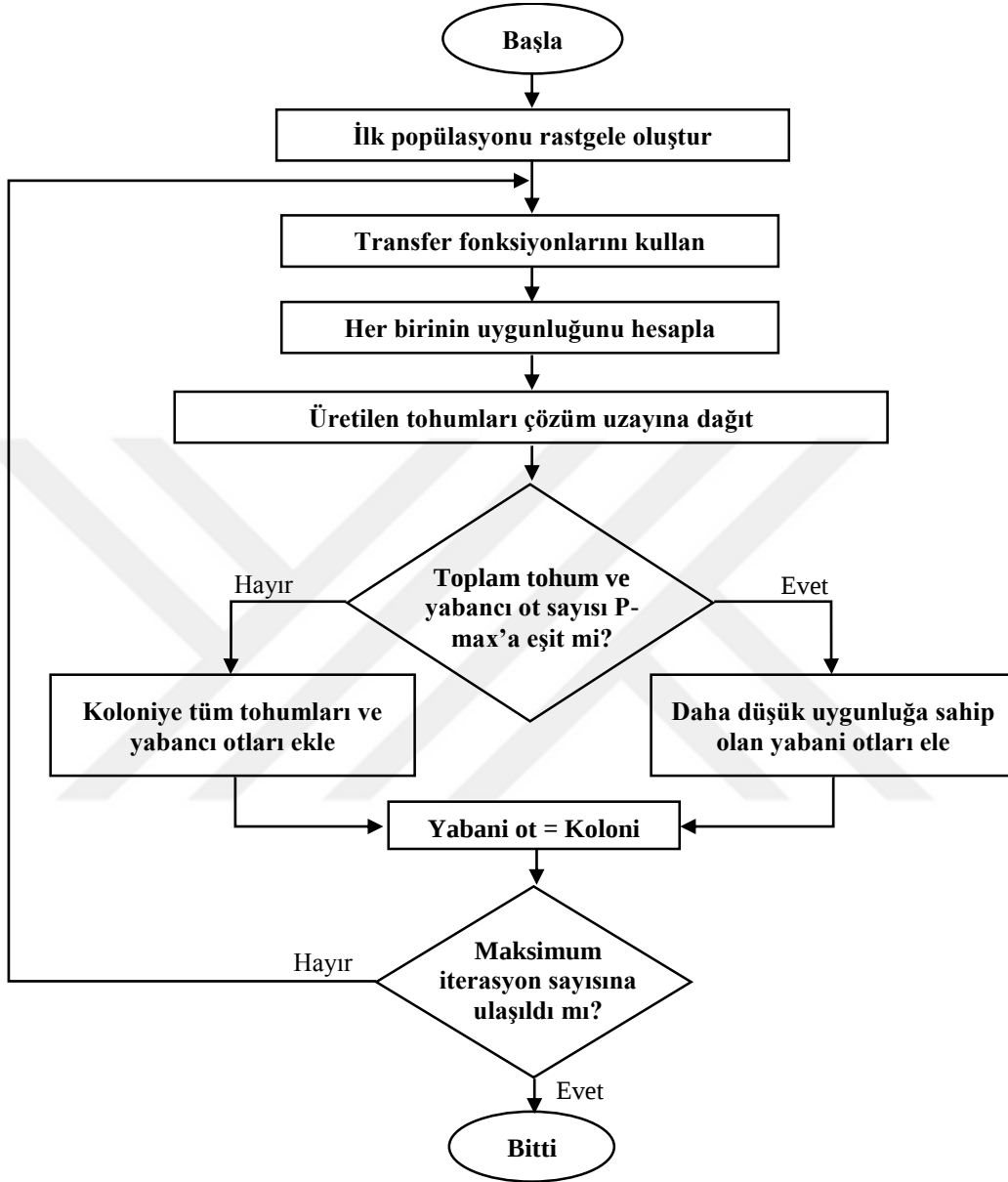
Türbin yerleştirme problemi ikili bir problemidir. Bunun için bu tez çalışmasında sürekli yapıya sahip YOO'ya 14 farklı transfer fonksiyonu uygulanarak ikili hale getirilmiştir. Transfer fonksiyonlarından elde edilen 14 farklı algoritma İYOO1-İYOO14 olarak isimlendirilmiştir. Bu algoritmalar içerisinde optimum türbin yerleştirme sonucunu elde eden algoritma İYOO olarak adlandırılmıştır. Transfer fonksiyonları ve algoritmaların isimleri Tablo 3.2'de gösterilmiştir (Held ve Kienhuis, 1995, Mirjalili ve Lewis, 2013, Crawford ve ark, 2014, Kiran, 2015, Saremi ve ark, 2015, Küçükyavuz ve Sen, 2017).

Tablo 3.2. Transfer fonksiyonları ailesi

S-şekilli transfer fonksiyonları ailesi	
Adı	Transfer fonksiyonları
S1 (İYOO1)	$T(x) = \frac{1}{1 + e^{-2x}}$
S2 (İYOO2)	$T(x) = \frac{1}{1 + e^{-x}}$
S3 (İYOO3)	$T(x) = \frac{1}{1 + e^{(-\frac{x}{2})}}$
S4 (İYOO4)	$T(x) = \frac{1}{1 + e^{(-\frac{x}{3})}}$
V-şekilli transfer fonksiyonları ailesi	
Adı	Transfer fonksiyonları
V1 (İYOO5)	$T(x) = \left \operatorname{erf}\left(\frac{\sqrt{\pi}}{2} x\right) \right = \left \frac{\sqrt{2}}{\pi} \int_0^{(\sqrt{\pi/2})x} e^{-t^2} dt \right $
V2 (İYOO6)	$T(x) = \tanh(x) $
V3 (İYOO7)	$T(x) = \left (x)/\sqrt{1+x^2} \right $
V4 (İYOO8)	$T(x) = \left \frac{2}{\pi} \arctan\left(\frac{\pi}{2} x\right) \right $
Mod transfer fonksiyonu ailesi	
Adı	Transfer fonksiyonları
Modulo base A (İYOO9)	binaryVector=mod(floor(continuousVector),2);
Modulo base B (İYOO10)	binaryVector=mod(round(mod(continuousVector),2),2);
Hibrit transfer fonksiyonu ailesi	
Adı	Transfer fonksiyonları
Ceil (İYOO11)	binaryVector=mod(ceil(continuousVector),2); if rand < 0.5
Ceil-Round (İYOO12)	binaryVector=mod(ceil(continuousVector),2); else binaryVector=mod(round(continuousVector),2); end if rand < 0.5
Ceil-Floor (İYOO13)	binaryVector=mod(ceil(continuousVector),2); else binaryVector=mod(floor(continuousVector),2); end if rand < 0.5
Round-Floor (İYOO14)	binaryVector=mod(round(continuousVector),2); else binaryVector=mod(floor(continuousVector),2); end

S-şekilli transfer fonksiyonları ailesi, V-şekilli transfer fonksiyonları ailesi, mod-A, mod-B, Ceil, Ceil-Round, Ceil-Floor ve Round-Floor transfer fonksiyonları kullanılarak 14 farklı ikili YOO yöntemleri oluşturulmuştur. Transfer fonksiyonları kullanılarak konum vektörleri 0 veya 1 değerini alacak şekilde ikili uzaya

dönüştürülmüştür. İYOO'nun akış şeması Şekil 3.12'de gösterilmiştir (Moallem ve Razmjoo, 2012, Ahmed ve ark, 2014).



Şekil 3.12. İYOO'nun akış şeması (Moallem ve Razmjoo, 2012, Ahmed ve ark, 2014)

3.6. Dağıtık Arama Algoritması

Evrimsel bir yaklaşım olan DA algoritması, ilk olarak Glover (Glover, 1977) tarafından 1977 yılında geliştirilmiştir. DA birçok problemin çözümünde kullanılarak başarılı sonuçlar veren metasezgisel bir algoritmadır. DA ilk önce yeni çözümler elde edebilmek için referans çözüm kümesi (R) oluşturur. Daha sonra DA daha iyi çözümlere ulaşmak için kaliteyi ve çeşitliliği kombine ederek referans kümedeki çözümleri akıllıca

birbirleriyle birleştirir. Her iterasyonda DA, iyileştirmeler yapar ve referans kümesi güncellenir. Bu işlem durdurma kriteri yerine getirilene kadar tekrarlanır. Şekil 3.13’de temel DA’nın tasarım şeması verilmektedir. Esnek bir yapıya sahip olan DA algoritması temel olarak 5 bileşenden oluşmaktadır.

3.6.1. Farklılaştırma üretim yöntemi

Bu yöntem bir dizi farklı çözümler üretmek ve arama uzayı içerisinde etkili bir popülasyon oluşturmak için kullanılmaktadır. Üretim yöntemi tamamen rassal olmadığından çeşitli ve kaliteli çok sayıda farklı çözüm grubu oluşturulmaktadır (Sagarna ve Lozano, 2006, Tang ve ark, 2010). Bunun amacı birbirinden mümkün olduğunca farklı deneme çözümlerini üretmektir. Bu süreç ile yüksek kaliteli ve çeşitli çözümler oluşturularak DA’nın başarısı artmaktadır.

3.6.2. İyileştirme yöntemi

İyileştirme yöntemi yeni çözümler üretildikten sonra kullanılır. Yeni çözümler kombinasyon yöntemiyle üretilen ilk deneme çözümleridir. Bu yöntem genellikle bir yerel arama yöntemi ile geliştirmeyi amaçlamaktadır.

3.6.3. Referans kümesi güncelleme yöntemi

Referans kümesi yeni çözümler üretmenin temelidir ve çok önemli bir parçasıdır (Tang ve ark, 2010). Referans kümesi, çözüm kombinasyonu yöntemini uygulayarak yeni çözümler üretmek için kullanılan yüksek kaliteli çözümlerin ve çeşitli çözümlerin toplamıdır. Önce kaliteli çözümleri referans kümesine dahil edilir. Sonra popülasyonda referans kümesinde olmayan ve referans kümesinde bulunan tüm çözümlere minimum mesafeyi maksimize edilir. Bu çözüm referans kümesine en çeşitli çözüm olarak dahil edilir. Kaliteli çözümler hedef işlev değerlerine göre seçilir. Düşük maliyet değerine sahip olan çözümler kaliteli çözümler olarak seçilir (Keskin ve Üster, 2007). Referans kümesinin güncelleme işlemi iterasyon boyunca sürmektedir.

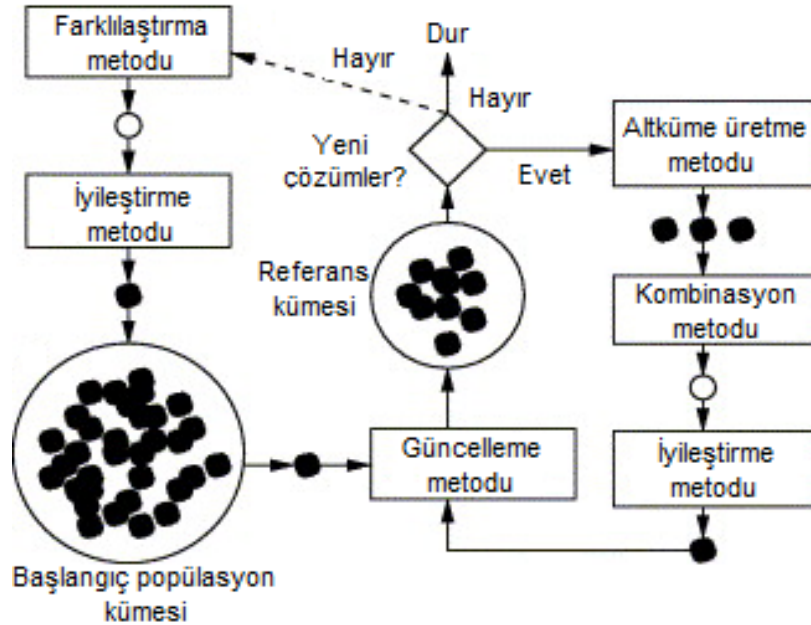
3.6.4. Altküme oluşturma yöntemi

Altkümeler Referans kümesine dayalı olarak oluşturulurlar ve buradan da çözüm kombinasyon yöntemine gönderilir. Altküme oluşturmak için Glove ve ark. (Glover ve ark, 2003) tarafından belirli bir kural oluşturulmuştur. İki elemanlı, üç elemanlı veya daha fazla sayıda altküme oluşturularak çözüm kombinasyonu yöntemine uygulanabilir.

3.6.5. Çözüm kombinasyonu yöntemi

Altkümelere ayrılan referans yeni çözümler üretmek için çözüm kombinasyon yöntemi kullanılmaktadır. Çözüm kombinasyonu yönteminin özel biçimine bağlı olarak, her altküme için bir veya daha fazla yeni çözüm oluşturulabilir.

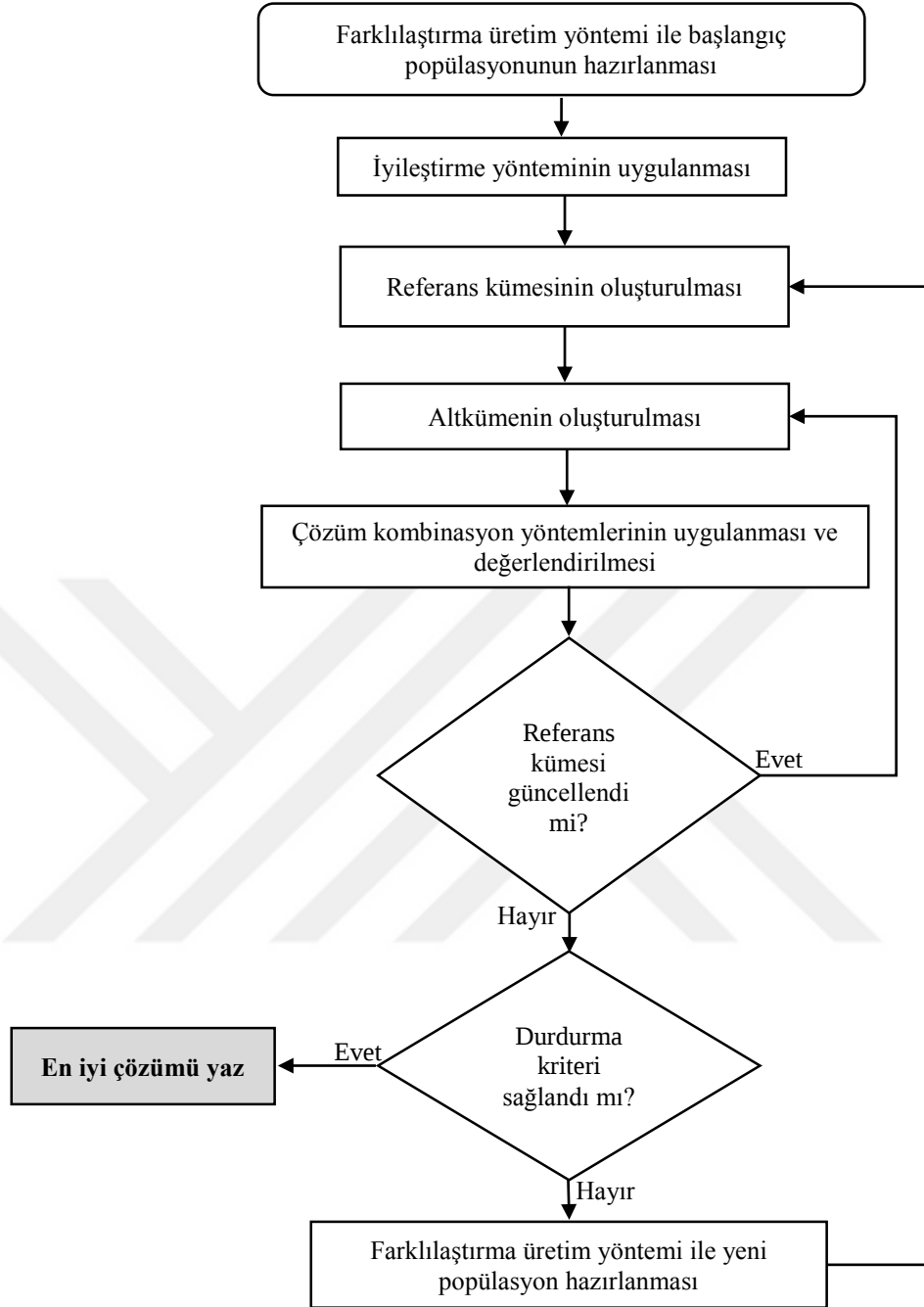
Çözüm kombinasyon yöntemi ile elde edilen yeni çözümler uygunluk fonksiyonuna gönderilir ve değerlendirilir. Bu duruma göre referans kümesinin güncelleme işlemi gerçekleşir. Referans kümesinde oluşacak herhangi bir durgunluğun önüne geçmek için DA algoritması, referans kümesinin belirli bir bölümünde değişikliğe gider ve bu sayede arama uzayında verimli bir arama yapabilmek için yeni çözümlerin çeşitliliği sağlanmış olur.



Şekil 3.13. Temel DA'nın tasarım şeması (Sagarna ve Lozano, 2006)

Şekil 3.13’de DA’nın tasarım şeması incelendiğinde farklılaştırma metodu kullanılarak kaliteli bireylerden oluşan bir başlangıç popülasyon kümesi oluşturulmaktadır. Referans kümesi başlangıç popülasyonunun bir altkümesidir. Buraya uygunluk fonksiyonuna göre en iyi olanlar seçilmektedir. Referans kümesinden ise belirli bir parametre değeri ile altkümeler oluşturulur. Sonra kombinasyon metodu ile yeni bireylerin oluşturulması sağlanır ve referans kümesinde yer alan kötü bireyler ile değişimi sağlanarak referans kümesi güncellenmiş olur. Eğer referans kümesine herhangi bir çözüm eklenmemişse algoritma durur veya yeniden oluşturma adımını kullanarak (kesikli çizgi) yeni farklı çözümler elde etmek için farklılaştırma üretim yöntemini yürütür (Sagarna ve Lozano, 2006). Bu işlem sonlandırma kriteri sağlanana kadar devam eder. DA’nın akış şeması Şekil 3.14’te verilmiştir.





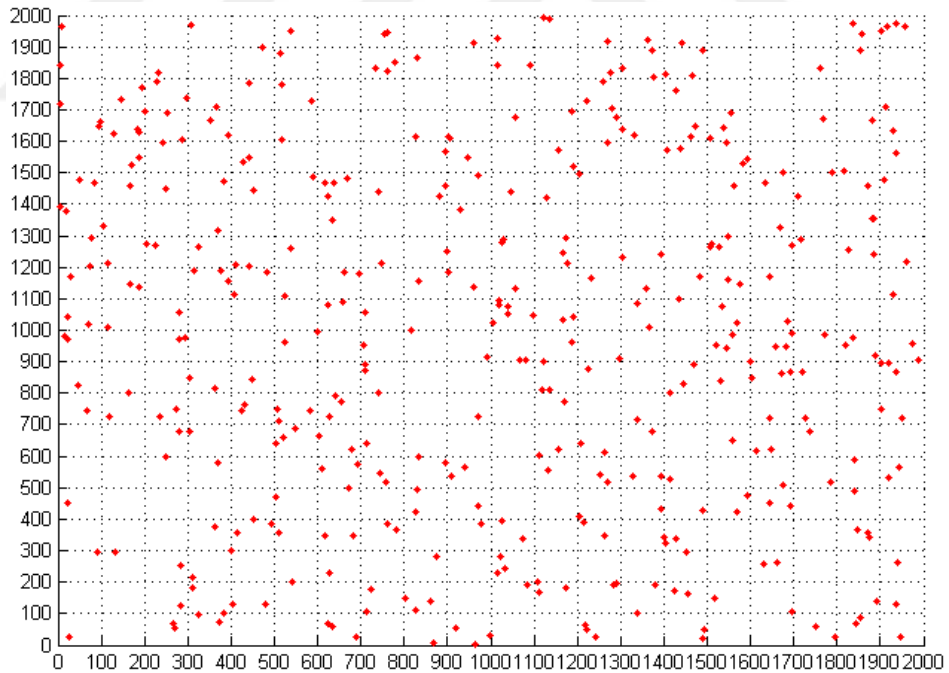
Şekil 3.14. DA'nın akış şeması (Haklı, 2017)

3.6.6. Dağıtık arama algoritmasının probleme uygulanması

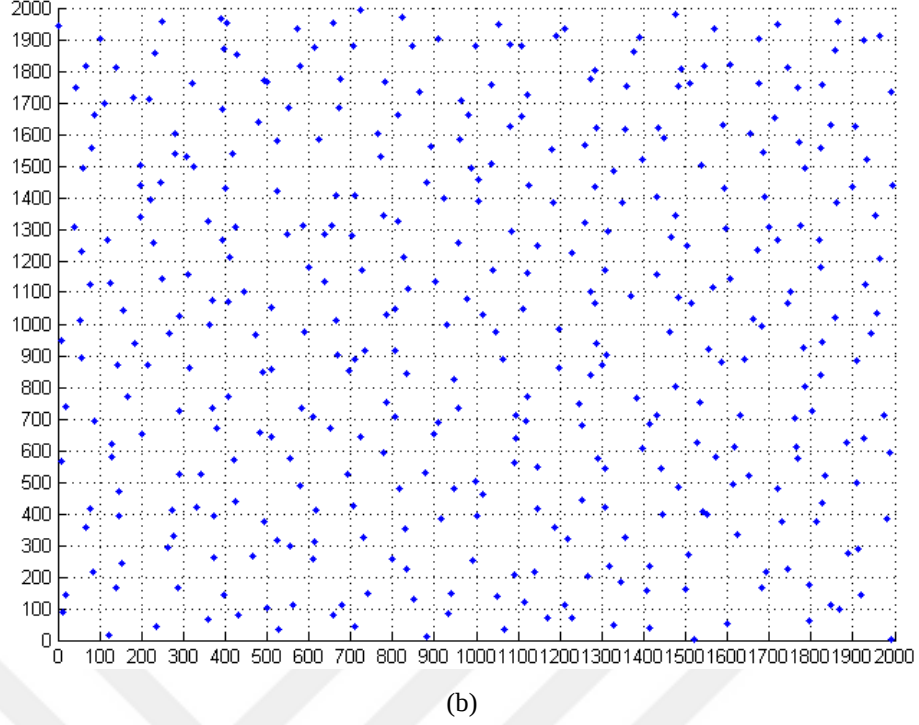
Başlangıç popülasyonun hazırlanması yakınsama hızı ve nihai çözümün kalitesini doğrudan etkilemesi nedeniyle evrimsel yöntemler için önemli bir süreçtir (Gao ve Liu, 2011, Bajer ve ark, 2016). DA'nın ilk aşaması olan farklılaştırma üretim yöntemi, başlangıç popülasyonun etkili ve verimli şekilde oluşturulmasını sağlamaktadır. Çözüm hakkında genel bir bilgi mevcut değilse, başlangıç popülasyonu rastgele olarak

oluşturulmaktadır (Gao ve Liu, 2011). Rüzgâr türbini yerleştirme probleminde ise rüzgâr türbinlerinin arazi üzerine düzenli bir şekilde yayılması gerekmektedir. Rastgele başlangıç popülasyonu ile türbin pozisyonlarının belirlenmesi belli bölgelerde kümelenmeye neden olabilir. Bu durum ise geç yakınsama ve çözüm kalitesinin düşmesi problemlerine yol açmaktadır.

Farklılaştırma üretim yöntemi ile başlangıç popülasyonunun arazi üzerinde tam anlamıyla yayılmasını sağlamak için bir strateji geliştirilmiştir. Şekil 3.15'te verildiği gibi arazi 20x20 olmak üzere toplamda 400 kareye bölünmüştür. Başlangıç popülasyonu ve boyut sayısı dikkate alınarak her bir kareye eşit sayıda yerleştirilecek şekilde türbin pozisyonları belirlenmektedir. Örneğin; 40 birey ve 10 boyut için $(40 \times 10) / 400$ 'den her bir kare içine 1 türbin düşecek şekilde türbinlerin X ve Y pozisyonları belirlenen aralıkta üretilmektedir. Böylece rüzgâr türbinlerinin arazi üzerinde belli bir bölgeye kümelenmeden eşit şekilde dağılması sağlanmıştır. Şekil 3.15'te 40 birey ve 10 boyut için rastgele ve farklılaştırma üretim yöntemi ile oluşturulmuş başlangıç popülasyonlarının türbin pozisyonları verilmektedir.



(a)



Şekil 3.15. Başlangıç popülasyonun rastgele (a) ve farklılaştırma üretim yöntemi (b) oluşturulması

Şekil 3.15(a) incelendiğinde, rastgele oluşturulan popülasyonda bazı karelere hiç türbin yerleştirilmediği ve bazı bölgelerde kümelenme olduğu görülmektedir. Şekil 3.15(b)'de ise farklılaştırma üretim yöntemi ile her bir kare içine türbin yerleştirilmesi sağlanmış ve türbinler arama uzayına düzenli şekilde yayılmıştır.

Başlangıç popülasyonu oluşturulduktan sonra, her bireyin uygunluk değeri amaç fonksiyonu yardımıyla hesaplanarak bireylerin kalitesi belirlenir. b_1 sayıda en kaliteli birey alınır ve bu alınan b_1 sayıda bireyden Öklid uzaklığı bakımından en farklı b_2 sayıda birey seçilerek referans kümesi oluşturulur. Böylece referans kümesindeki birey sayısı $b = b_1 + b_2$ olmaktadır. İkili-gerçek kodlama da bireyin bir geni için üç farklı bilgi tutulmaktadır: türbinin X koordinatı, türbinin Y koordinatı ve türbinin geçerli olup olmadığına ilişkin ikili değer (0 veya 1). Rüzgâr türbini yerleştirme problemi için ikili-gerçek kodlama ile oluşturulmuş 10 boyutlu örnek bir birey Şekil 3.16'da gösterilmektedir. Bu aşamada bireylerin birbirinden farklılığı incelenirken üç bilgi de işleme dâhil edilmektedir. Şekil 3.16 incelendiğinde her bir indise karşılık gelen üç veri, bir rüzgâr türbininin koordinatlarını ve aktif olup olmadığı bilgisini göstermektedir. Örneğin, 1 numaralı gende bulunan rüzgâr türbini 10.7 X koordinatı, 3.8 Y koordinatında konumlanmışken pasif durumdadır. 5.7 X ve 15.8 Y koordinatlarına yerleşen 2 numaralı gende bulunan rüzgâr türbini ise aktif durumdadır. Buradaki aktif ve pasiflik çalışma

durumu için değil yerleştirme durumu için kullanılmıştır. Yani B değeri 0 olan türbinler birey içinde dikkate alınmamaktadır, sadece B değeri 1 olan rüzgâr türbinleri işleme sokulmaktadır.

	1	2	3	4	5	6	7	8	9	10
X	10,7	5,7	9,8	1,5	3,6	17,9	15,3	4,8	0,8	8,3
Y	3,8	15,8	12,6	7,4	2,2	13,5	9,6	11,7	19,5	10,1
B	0	1	1	0	1	0	0	1	1	0

Şekil 3.16. İkili-gerçek kodlama için örnek bir birey

Referans kümesi güncellemesi yöntemine geçilmeden önce altküme oluşturma yöntemi gerçekleştirilir. Bu aşamada referans kümesinde bulunan bireyler kullanılarak altkümeler belirlenmektedir. Her birey birbiri ile çift oluşturacak biçimde 2'li gruplar hazırlanarak $b*(b-1)/2$ sayıda altküme ayarlanmaktadır. Bu altkümeler çözüm kombinasyon yöntemindeki çaprazlama teknikleri ile yeni birey üretiminde kullanılmaktadır. Çözüm kombinasyon yöntemi DA için büyük önem taşımakta ve problemin çözümünde önemli bir rol oynamaktadır. İkili-gerçek kodlamada bireyler hem sürekli hem ayrık veriler içermektedir. Bu nedenle kullanılacak çaprazlama teknikleri ya ayrı ayrı işlemler yapmalı ya da her bir geni bir bütün olarak alıp ayrık bir veri gibi değerlendirmelidir. Hem işlem karmaşıklığı azaltmak hem de genlerdeki bütünlüğü sağlamak adına ikinci yol tercih edilmiştir. Ayrıca, başlangıç popülasyonun arama uzayına yeterince dağılması ile her bölge içine birçok rüzgâr türbini konumlandırılmış, böylece türbinlerin en uygun olanlarının seçilmesi işlemi kolaylaşmıştır. Belli bir iterasyon sonunda türbin pozisyonlarının değiştirilmemesi ve popülasyon çeşitliliğinin kaybolması gibi sorunların üstesinden gelmek için ise yeniden popülasyon üretimi gerçekleştirilmiştir.

Ayrık veriler üzerinde kullanılan farklı çaprazlama teknikleri ile her bir altküme için çözüm kombinasyon yönteminde altı farklı birey üretilmiştir. Tek nokta, iki nokta (Holland, 1992) ve tek düze (Syswerda, 1989) çaprazlama gibi daha yumuşak çaprazlama tekniklerinin yanında ayrılma (Rajasekaran ve Pai, 2011) ve alt parça (substring) (Park ve ark, 2000) gibi bireyin yapısında daha büyük değişikliklere sebep olan sert çaprazlama teknikleri de kullanılmıştır. Şekil 3.17’de tek nokta, iki nokta ve tekdüze çaprazlamaların örnek işlemleri gösterilmiştir.

Birey 1	10.7	5.7	9.8	1.5	3.6	17.9	15.3	4.8	0.8	8.3
	3.8	15.8	12.6	7.4	2.2	13.5	9.6	11.7	19.5	10.1
	0	1	1	0	1	0	0	1	1	0

Birey 2	4.2	17.4	11.5	9.9	19.8	3.2	13.9	6.1	2.9	1
	11.8	3.9	10.3	0.2	12.1	7.8	5.6	15.4	18.3	1
	1	0	1	0	0	1	1	0	1	1

Çocuk	10.7	5.7	9.8	1.5	19.8	3.2	13.9	6.1	2.9	14.5
	3.8	15.8	12.6	7.4	12.1	7.8	5.6	15.4	18.3	13.2
	0	1	1	0	0	1	1	0	1	1

a) Tek Nokta

Çocuk	10.7	5.7	11.5	9.9	19.8	3.2	13.9	4.8	0.8	8.3
	3.8	15.8	10.3	0.2	12.1	7.8	5.6	11.7	19.5	10.1
	0	1	1	0	0	1	1	1	1	0

b) İki Nokta

Çocuk	4.2	5.7	11.5	9.9	3.6	3.2	15.3	6.1	0.8	8.3
	11.8	15.8	10.3	0.2	2.2	7.8	9.6	15.4	19.5	10.1
	1	1	1	0	1	1	0	0	1	0

c) Tekdüze

Şekil 3.17. Tek nokta (a), iki nokta (b) ve tekdüze (c) çaprazlama tekniklerinin örnek uygulamaları

Şekil 3.18’de görüldüğü gibi, tek nokta çaprazlamada çocuk birey belirlenen çaprazlama noktasına kadar olan genleri Birey 1’den, kalan genleri Birey 2’den almaktadır. İki nokta çaprazlama da ise belirlenen iki çaprazlama noktası arasında kalan genler Birey 2’den kopyalanırken kalan genler Birey-1’den tamamlanmaktadır. Tekdüze çaprazlamada, her bir gen için $[0,1]$ aralığında rastgele bir sayı belirlenir sayı 0.5’ten küçükse Birey-1’den, büyükse Birey-2’den gen kopyalanmaktadır.

Birey 1	10.7	5.7	9.8	1.5	3.6	17.9	15.3	4.8	0.8	8.3
	3.8	15.8	12.6	7.4	2.2	13.5	9.6	11.7	19.5	10.1
	0	1	1	0	1	0	0	1	1	0

Birey 2	4.2	17.4	11.5	9.9	19.8	3.2	13.9	6.1	2.9	14.5
	11.8	3.9	10.3	0.2	12.1	7.8	5.6	15.4	18.3	13.2
	1	0	1	0	0	1	1	0	1	1

Çocuk-1	10.7	3.2	13.9	6.1	2.9	17.9	15.3	4.8	0.8	8.3
	3.8	7.8	5.6	15.4	18.3	13.5	9.6	11.7	19.5	10.1
	0	1	1	0	1	0	0	1	1	0

Çocuk-2	4.2	17.9	15.3	4.8	0.8	3.2	13.9	6.1	2.9	14.5
	11.8	13.5	9.6	11.7	19.5	7.8	5.6	15.4	18.3	13.2
	1	0	0	1	1	1	1	0	1	1

a) Ayrılma

Çocuk	10.7	5.7	11.5	9.9	19.8	17.9	15.3	4.8	0.8	14.5
	3.8	15.8	10.3	0.2	12.1	13.5	9.6	11.7	19.5	13.2
	0	1	1	0	0	0	0	1	1	1

b) Alt parça

Şekil 3.18. Ayrılma (a) ve alt parça (b) çaprazlama tekniklerinin örnek uygulamaları

Şekil 3.18’de ise ayrılma ve alt parça çaprazlama tekniklerinin örnek uygulamaları gösterilmektedir. Şekil 3.18(a)’da görüldüğü gibi ayrılma çaprazlamadan iki çocuk birey elde edilmektedir. Bu çaprazlama tekniğinde aynı aralık mesafesine sahip iki ayrı nokta çifti belirlenmektedir. Şekilde verilen örnek için 6-9 ve 2-5 noktaları rastgele olarak seçilmiştir. Çocuk-1 için Birey-2’de de 6-9 noktaları arasına karşılık gelen genler Birey-1’in 2-5 noktaları arasına karşılık gelen genlere kopyalanırken, Çocuk-2 için Birey-2 esas alınarak kopyalama işlemi Birey-1’den yapılmaktadır. Şekil 3.18(b)’de ise her birey alt parçalara bölünür ve her bir parça için kendi içinde bir çaprazlama noktası belirlenir. Belirlenen çaprazlama noktalarına göre Birey-1 ve Birey-2’den genler kopyalanarak çocuk birey elde edilir. Bu çaprazlama tekniği için alt parça gen uzunluğunun belirlenmesi gerekmektedir. Örnek işlem için alt parça gen uzunluğu 5 belirlenmiştir.

Şekil 3.17 ve 3.18’de verilen çaprazlama tekniklerini ile çözüm kombinasyon yöntemi gerçekleştirilir. Her altküme çifti için altı farklı birey üretilerek havuz oluşturulur. Çaprazlama teknikleri ile yapılan yer değiştirme sonrası rüzgâr türbinlerinin pasif ve aktif durumları değişmemektedir. Bu durum, başlangıç popülasyonundan aktif olarak gelen rüzgâr türbinlerinin yerleştirme işlemine dahil olmasına ve diğer türbinlerin dışarda kalmasına sebep olmaktadır. Birey ve boyut sayısının yüksek seçilmesi ile bir kare içine düşen türbin sayısının yeterince büyük olması bu sorunu bir nebze çöze de

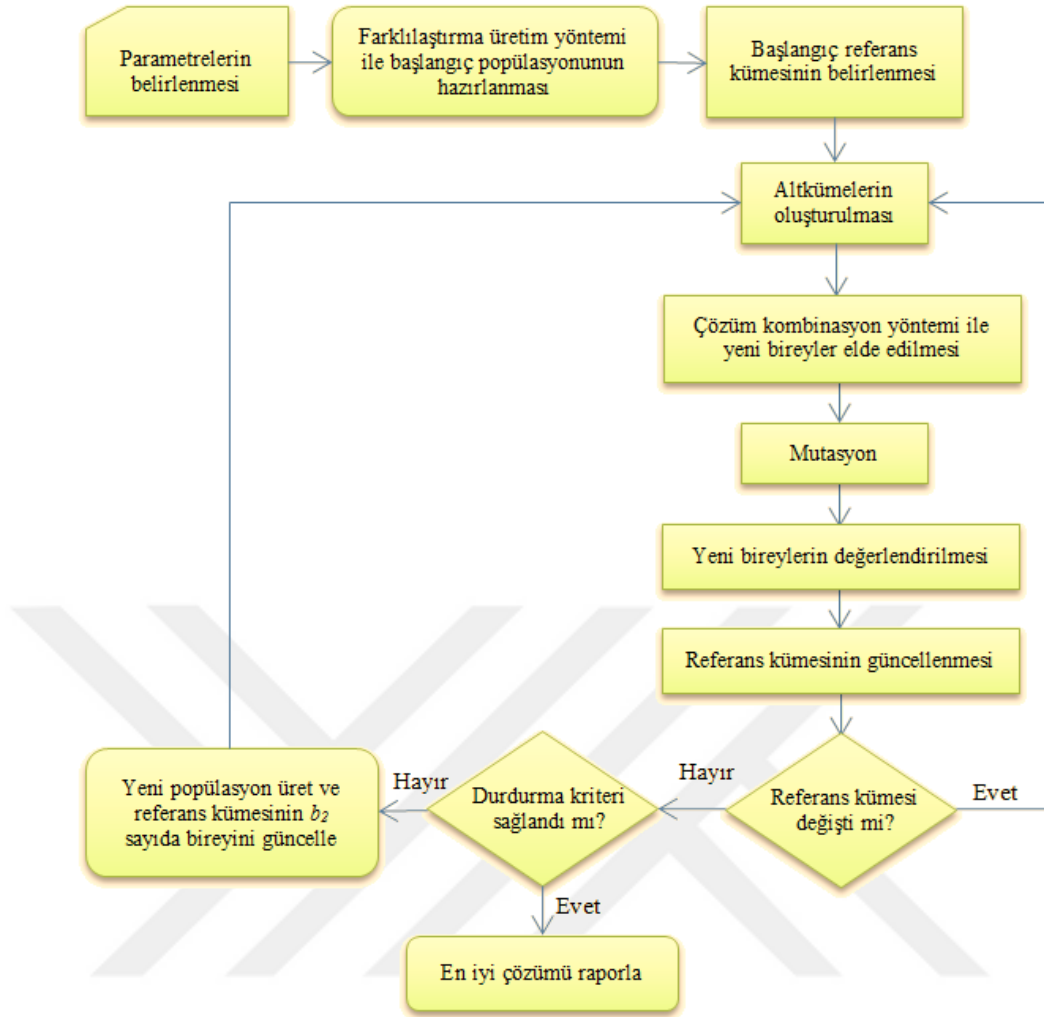
yeterli olmamaktadır. Bu sorunun üstesinden gelmek için havuz içinde bulunan bireyler mutasyon işlemine uğratılmaktadır. Mutasyon işleminde, her bir birey için rastgele bir sayı belirlenmekte ve belirlenen sayı mutasyon oranının altında ise o bireyin rastgele belirlenen geninin durumu aktif ise pasif, pasif ise aktif hale getirilmektedir. Şekil 3.19'de örnek bir mutasyon işleminin uygulaması gösterilmektedir.

Çocuk	10.7	5.7	11.5	9.9	19.8	3.2	13.9	4.8	0.8	8.3
	3.8	15.8	10.3	0.2	12.1	7.8	5.6	11.7	19.5	10.1
	0	1	1	0	0	1	0	0	1	0

Çocuk	10.7	5.7	11.5	9.9	19.8	3.2	13.9	4.8	0.8	8.3
	3.8	15.8	10.3	0.2	12.1	7.8	5.6	11.7	19.5	10.1
	0	1	1	0	1	1	0	0	1	0

Şekil 3.19. Mutasyon işleminin örnek uygulaması

Gerçekleştirilen mutasyon işlemi ile birlikte havuzdaki bireylerin uygunluk değerleri amaç fonksiyonu yardımıyla hesaplanmaktadır. Havuzdaki bireyler ve referans kümesi birleştirilir ve b sayıda en kaliteli birey seçilerek yeni referans kümesi oluşturulur. Referans kümesinde bir değişiklik olduysa, çözüm kombinasyon yöntemleri ile referans kümesinden yeni bireyler üretme işlemine devam edilir. Eğer referans kümesi güncellenmemiş ise referans kümesi içindeki b_1 sayıda en kaliteli birey hafızada tutulur ve yeni popülasyon oluşturulur. Bu yeni popülasyondan referans kümesi içindeki b_1 sayıda bireyden en farklı olan b_2 sayıda birey seçilerek referans kümesi tamamlanır ve çözüm kombinasyon yöntemi ile yeni birey üretme işlemine durdurma kriteri sağlanana kadar devam edilir. Şekil 3.20'de rüzgâr türbini yerleşiminin optimizasyonu için kullanılan MDA yönteminin çalışma diyagramı verilmektedir.



Şekil 3.20. MDA yönteminin rüzgâr türbin yerleřtirme problemi için çalışma diyagramı

4. DENEYSEL SONUÇLAR VE TARTIŞMA

Bu bölümde İkili kodlama, ikili-gerçek kodlama ile yapılan rüzgâr türbini yerleştirme probleminin sonuçları ile tüm deneysel sonuçların karşılaştırılması olmak üzere üç başlıktan oluşmaktadır.

Tüm deneysel sonuçlar MATLAB 2014 (8.3) programı kullanılarak gerçekleştirilmiştir.

4.1. İkili Kodlama Sonuçları

Bu bölümde İYAA ve İYOO yöntemlerinin ikili yöntem ile rüzgâr türbinine uygulanarak elde edilen sonuçları literatürde bulunan çalışmaların sonuçları ile karşılaştırılmıştır.

4.1.1. İYAA Sonuçları

Orijinal YAA içerisinde bulunan popülasyon kesme kuvveti, enerji kaybı ve adaptasyon parametre değerleri rastgele olarak üretilmektedir. Ancak Beskirli ve ark. (Beskirli ve ark, 2016) tarafından 2016'da yapılan çalışmada parametre değerleri optimize edildiğinde daha başarılı olduğu görülmektedir. Bu değerleri elde etmek için 8 adet test fonksiyonu kullanılmıştır. Bu fonksiyonlar Tablo 4.1'de gösterilmiştir.

Tablo 4.1. Test fonksiyonları (Türü:T, Tek modlu:U, Çok modlu:M)

	Aralık	T	Fonksiyon	Formül	Uygunluk
F1	[-5.12, 5.12]	U	Sphere	$f_1 = \sum_{i=1}^n x_i^2$	0
F2	[-10, 10]	U	Schwefel 2.22	$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	0
F3	[-10, 10]	U	Rosenbrock	$f_3 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	0
F4	[-1.28, 1.28]	U	Noise	$f_4 = \sum_{i=1}^n ix_i^4 + random[0,1]$	0
F5	[-500, 500]	M	Schwefel	$f_5 = 418.98288727243369 * n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	0
F6	[-5.12, 5.12]	M	Rastrigin	$f_6 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	0
F7	[-32, 32]	M	Ackley	$f_7 = -20 \exp \left\{ -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right\} - \exp \left\{ \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right\} + 20 + e$	0
F8	[-600, 600]	M	Griewank	$f_8 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0

YAA'nın orijinal parametre değerleri yerine adaptasyon için 0.1, 0.2, 0.3, 0.4 ve 0.5 değerleri, enerji kaybı için 0.1, 0.2, 0.3, 0.4 ve 0.5 değerleri, popülasyon için 10, 20, 30, 40 ve 50 değerleri ve kesme kuvveti için 1, 2, 3, 4 ve 5 değerleri kullanılmıştır. YAA'nın bu parametreler içerisindeki en iyi değeri araştırılmıştır. Problem fonksiyonları 30 kez çalıştırılmıştır. En iyi değer, ortalama değer ve standart sapma sonuçları parametre değerleri ile birlikte Tablo 4.2, 4.4, 4.6 ve 4.8'de detaylı gösterilmiştir. Ayrıca algoritmanın çalışma süreleri Tablo 4.3, 4.5, 4.7 ve 4.9'da verilmiştir. Elde edilen en iyi sonuçlar tablolarda koyu yazı ile yazılmıştır.

Tablo 4.2. Adaptasyon parametresi farklı değerlerinin test fonksiyonlarındaki sonuçlarının karşılaştırılması

		0.1	0.2	0.3	0.4	0.5
F1	En iyi	8.360E-22	4.041E-22	4.586E-22	1.027E-22	1.708E-22
	Ortalama	7.142E-21	3.992E-21	2.113E-21	1.548E-21	1.369E-21
	Std	6.093E-21	2.927E-21	1.694E-21	1.794E-21	1.386E-21
F2	En iyi	1.650E-12	1.136E-12	7.923E-13	6.193E-13	5.290E-13
	Ortalama	3.465E-12	2.354E-12	1.889E-12	1.880E-12	1.363E-12
	Std	1.602E-12	8.508E-13	8.824E-13	9.870E-13	6.385E-13
F3	En iyi	1.178E-01	3.552E-01	1.446E-01	9.697E-02	6.764E-02
	Ortalama	1.868E+01	1.338E+01	1.651E+01	1.571E+01	8.755E+00
	Std	2.046E+01	1.898E+01	1.889E+01	1.913E+01	1.295E+01
F4	En iyi	1.504E-02	1.558E-02	1.803E-02	1.717E-02	1.538E-02
	Ortalama	3.044E-02	3.328E-02	3.188E-02	3.351E-02	3.274E-02
	Std	7.839E-03	9.418E-03	9.388E-03	7.311E-03	7.900E-03
F5	En iyi	7.640E-11	7.276E-12	7.276E-12	1.091E-11	0.000E+00
	Ortalama	1.567E-07	7.117E-09	7.896E+00	7.896E+00	3.948E+00
	Std	3.586E-07	1.839E-08	3.005E+01	4.325E+01	2.162E+01
F6	En iyi	3.296E-09	2.522E-10	1.662E-10	2.549E-11	3.546E-12
	Ortalama	9.528E-07	3.317E-02	6.651E-02	3.317E-02	9.950E-02
	Std	1.999E-06	1.817E-01	2.524E-01	1.817E-01	3.036E-01
F7	En iyi	2.542E-09	1.130E-09	5.050E-10	2.614E-10	2.239E-10
	Ortalama	8.799E-09	3.135E-09	1.862E-09	1.262E-09	7.188E-10
	Std	7.141E-09	1.391E-09	1.049E-09	8.851E-10	3.564E-10
F8	En iyi	0.000E+00	1.110E-16	0.000E+00	0.000E+00	0.000E+00
	Ortalama	2.406E-13	1.258E-13	8.197E-10	8.607E-14	2.268E-13
	Std	1.065E-12	3.030E-13	4.489E-09	3.630E-13	1.113E-12

Tablo 4.3. Adaptasyon parametresi farklı değerlerinin test fonksiyonlarındaki çalışma sürelerinin karşılaştırması

		0.1	0.2	0.3	0.4	0.5
F1	En iyi	4.28	4.72	4.67	4.66	4.70
	Ortalama	5.29	5.21	5.04	5.10	5.16
	Std	0.56	0.41	0.33	0.37	0.48
F2	En iyi	4.47	4.42	7.92	8.11	8.21
	Ortalama	4.90	5.04	8.45	8.68	8.87
	Std	0.34	0.52	0.35	0.50	0.62
F3	En iyi	8.48	8.49	8.57	8.60	8.79
	Ortalama	9.07	9.22	9.30	9.21	9.33
	Std	0.53	0.58	0.56	0.55	0.49
F4	En iyi	9.87	9.62	9.66	9.62	9.80
	Ortalama	10.32	10.17	10.32	10.68	10.50
	Std	0.57	0.52	0.49	0.84	0.56
F5	En iyi	8.82	8.55	8.74	8.84	8.67
	Ortalama	9.57	9.57	9.66	9.82	9.62
	Std	0.66	0.72	0.62	0.75	0.59
F6	En iyi	8.45	8.67	8.54	7.20	7.38
	Ortalama	9.25	9.30	9.27	9.06	7.70
	Std	0.61	0.58	0.47	0.90	0.33
F7	En iyi	7.22	7.26	7.29	7.33	7.34
	Ortalama	7.57	7.69	7.63	7.65	7.75
	Std	0.31	0.52	0.34	0.33	0.32
F8	En iyi	6.04	6.05	3.62	3.65	3.67
	Ortalama	6.99	6.10	4.57	3.68	3.69
	Std	0.67	0.05	1.22	0.02	0.02

Tablo 4.4. Enerji kaybı parametresi farklı değerlerinin test fonksiyonlarındaki sonuçlarının karşılaştırılması

		0.1	0.2	0.3	0.4	0.5
F1	En iyi	2.666E-24	7.194E-24	4.702E-23	1.048E-21	1.636E-20
	Ortalama	1.627E-23	7.051E-23	8.530E-22	9.595E-21	1.252E-19
	Std	1.123E-23	7.728E-23	8.766E-22	7.848E-21	8.338E-20
F2	En iyi	7.558E-14	1.155E-13	3.691E-13	2.086E-12	8.977E-12
	Ortalama	1.843E-13	3.698E-13	1.784E-12	6.807E-12	3.710E-11
	Std	6.923E-14	2.333E-13	1.028E-12	3.577E-12	2.614E-11
F3	En iyi	3.480E-01	3.702E-01	1.106E-02	1.864E-01	3.885E-02
	Ortalama	1.172E+01	1.571E+01	1.669E+01	1.476E+01	1.453E+01
	Std	1.646E+01	1.947E+01	1.951E+01	1.595E+01	1.961E+01
F4	En iyi	1.534E-02	1.608E-02	7.624E-03	2.427E-02	2.088E-02
	Ortalama	2.567E-02	2.860E-02	2.702E-02	3.709E-02	4.116E-02
	Std	5.468E-03	8.210E-03	8.181E-03	7.345E-03	1.127E-02
F5	En iyi	-1.819E-12	-1.819E-12	1.819E-12	9.095E-12	5.402E-10
	Ortalama	2.425E-13	7.896E+00	7.896E+00	2.175E+01	5.554E+01
	Std	3.873E-12	3.005E+01	3.005E+01	4.533E+01	8.633E+01
F6	En iyi	3.908E-13	5.382E-13	9.699E-13	1.291E-11	3.395E-10
	Ortalama	3.317E-02	5.715E-10	1.327E-01	4.746E-01	2.362E-01
	Std	1.817E-01	1.499E-09	3.440E-01	6.731E-01	5.000E-01
F7	En iyi	7.825E-11	9.194E-11	2.363E-10	8.591E-10	2.194E-09
	Ortalama	2.032E-10	2.806E-10	7.193E-10	3.115E-09	9.573E-09
	Std	1.170E-10	1.164E-10	3.969E-10	1.667E-09	6.590E-09
F8	En iyi	0.000E+00	0.000E+00	1.110E-16	2.220E-16	3.553E-15
	Ortalama	9.215E-16	3.556E-15	1.822E-14	6.155E-13	2.161E-12
	Std	2.833E-15	8.365E-15	3.326E-14	1.591E-12	5.812E-12

Tablo 4.5. Enerji kaybı parametresi farklı değerlerinin test fonksiyonlarındaki çalışma sürelerinin karşılaştırması

		0.1	0.2	0.3	0.4	0.5
F1	En iyi	1.79	2.07	2.28	2.48	4.01
	Ortalama	1.83	2.13	2.31	3.59	4.03
	Std	0.05	0.03	0.02	0.48	0.01
F2	En iyi	2.79	3.16	3.43	3.66	3.89
	Ortalama	2.83	3.19	3.46	3.69	5.84
	Std	0.02	0.02	0.02	0.02	1.35
F3	En iyi	3.75	4.74	5.22	5.49	5.93
	Ortalama	4.53	5.10	5.66	6.00	6.34
	Std	0.37	0.32	0.37	0.30	0.34
F4	En iyi	4.73	9.54	10.25	11.07	11.62
	Ortalama	5.16	10.10	11.06	11.70	12.43
	Std	0.35	0.45	0.50	0.37	0.38
F5	En iyi	7.54	8.87	9.39	10.14	10.84
	Ortalama	8.03	9.22	9.96	10.78	11.39
	Std	0.24	0.26	0.31	0.34	0.37
F6	En iyi	7.09	8.08	8.87	9.82	10.81
	Ortalama	7.58	8.77	9.55	10.42	11.17
	Std	0.36	0.33	0.42	0.36	0.34
F7	En iyi	7.40	8.53	9.02	10.06	10.59
	Ortalama	7.78	8.97	9.80	10.52	11.17
	Std	0.22	0.43	0.30	0.31	0.28
F8	En iyi	7.31	8.59	2.50	2.68	2.84
	Ortalama	7.69	8.96	2.59	2.73	4.20
	Std	0.26	0.26	0.09	0.03	0.90

Tablo 4.6. Popülasyon parametresi farklı değerlerinin test fonksiyonlarındaki sonuçlarının karşılaştırılması

		10	20	30	40	50
F1	En iyi	6.13E-23	9.77E-24	6.43E-23	6.02E-23	1.59E-22
	Ortalama	2.26E-20	1.38E-22	5.43E-22	8.53E-22	1.95E-21
	Std	8.86E-20	2.68E-22	5.52E-22	8.15E-22	1.88E-21
F2	En iyi	2.15E-13	8.95E-14	3.50E-13	3.73E-13	9.51E-13
	Ortalama	5.70E-08	4.41E-13	9.94E-13	1.68E-12	2.02E-12
	Std	3.12E-07	4.19E-13	6.04E-13	1.07E-12	9.57E-13
F3	En iyi	1.14E+00	3.17E-02	1.19E-02	9.28E-02	1.22E-01
	Ortalama	5.88E+01	2.98E+01	1.15E+01	1.43E+01	9.92E+00
	Std	3.91E+01	3.11E+01	1.95E+01	1.77E+01	1.03E+01
F4	En iyi	7.12E-02	2.28E-02	2.23E-02	1.28E-02	1.71E-02
	Ortalama	1.47E-01	5.12E-02	3.66E-02	3.05E-02	3.11E-02
	Std	7.36E-02	1.52E-02	9.05E-03	8.64E-03	8.50E-03
F5	En iyi	3.55E+02	1.82E-12	0.00E+00	0.00E+00	0.00E+00
	Ortalama	1.15E+03	2.65E+02	4.34E+01	2.37E+01	7.90E+00
	Std	4.20E+02	1.91E+02	7.92E+01	6.52E+01	4.32E+01
F6	En iyi	3.98E+00	9.95E-01	7.05E-12	7.23E-12	4.72E-12
	Ortalama	1.99E+01	4.01E+00	8.29E-01	6.63E-02	6.63E-02
	Std	8.16E+00	2.07E+00	9.81E-01	2.52E-01	2.52E-01
F7	En iyi	3.55E-10	3.13E-11	1.07E-10	2.88E-10	3.24E-10
	Ortalama	1.37E+00	1.59E-10	4.42E-10	6.41E-10	1.43E-09
	Std	1.23E+00	1.38E-10	2.80E-10	2.66E-10	8.66E-10
F8	En iyi	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Ortalama	1.13E-02	1.23E-10	2.24E-13	3.70E-14	5.69E-14
	Std	1.58E-02	5.97E-10	5.92E-13	1.21E-13	1.46E-13

Tablo 4.7. Popülasyon parametresi farklı değerlerinin test fonksiyonlarındaki çalışma sürelerinin karşılaştırması

		10	20	30	40	50
F1	En iyi	2.25	3.09	3.84	4.68	5.64
	Ortalama	2.58	3.59	4.33	5.09	6.04
	Std	0.24	0.33	0.29	0.35	0.27
F2	En iyi	2.12	2.85	3.77	8.15	9.35
	Ortalama	2.36	3.27	4.40	8.59	10.26
	Std	0.18	0.32	0.95	0.26	0.72
F3	En iyi	3.93	5.59	7.33	8.38	10.00
	Ortalama	4.39	6.25	7.79	9.30	11.00
	Std	0.34	0.53	0.39	0.65	0.60
F4	En iyi	4.48	6.09	7.88	9.63	11.56
	Ortalama	4.95	6.87	8.70	10.26	12.45
	Std	0.32	0.44	0.50	0.54	0.71
F5	En iyi	3.98	5.63	7.00	9.01	10.80
	Ortalama	4.60	6.42	8.01	9.74	11.54
	Std	0.31	0.54	0.47	0.54	0.70
F6	En iyi	3.88	5.54	7.31	8.43	10.28
	Ortalama	4.54	6.11	7.83	9.30	11.04
	Std	0.37	0.38	0.48	0.53	0.55
F7	En iyi	4.03	5.40	6.11	7.42	8.70
	Ortalama	4.48	6.16	7.45	7.69	9.05
	Std	0.34	0.41	0.82	0.26	0.38
F8	En iyi	3.35	4.74	6.11	7.31	8.68
	Ortalama	3.57	5.02	6.40	7.74	9.09
	Std	0.10	0.23	0.28	0.31	0.42

Tablo 4.8. Kesme kuvveti parametresi farklı değerlerinin test fonksiyonlarındaki sonuçlarının karşılaştırılması

		1	2	3	4	5
F1	En iyi	2.420E-07	1.671E-22	5.198E-10	1.292E-05	6.618E-05
	Ortalama	8.964E-06	1.297E-21	2.159E-09	6.898E-05	7.626E-04
	Std	7.027E-06	1.481E-21	1.650E-09	4.793E-05	7.131E-04
F2	En iyi	1.801E-03	3.684E-13	6.581E-06	6.686E-03	3.629E-02
	Ortalama	1.984E-02	1.388E-12	2.288E-05	2.187E-02	1.211E-01
	Std	4.138E-02	6.602E-13	1.296E-05	1.233E-02	6.140E-02
F3	En iyi	8.375E+00	4.542E-02	1.249E+01	3.104E+01	2.942E+01
	Ortalama	3.344E+01	1.841E+01	6.120E+01	1.195E+02	1.779E+02
	Std	1.021E+01	2.124E+01	2.919E+01	4.204E+01	1.249E+02
F4	En iyi	6.009E-02	1.786E-02	2.623E-02	3.340E-02	8.668E-02
	Ortalama	1.183E-01	3.065E-02	6.078E-02	1.482E-01	2.269E-01
	Std	2.981E-02	8.582E-03	1.901E-02	5.107E-02	9.199E-02
F5	En iyi	8.042E+02	0.000E+00	3.559E+02	1.215E+03	1.095E+03
	Ortalama	1.386E+03	1.974E+01	7.601E+02	1.680E+03	1.712E+03
	Std	2.911E+02	4.489E+01	2.276E+02	2.647E+02	2.892E+02
F6	En iyi	1.384E+01	6.722E-12	4.118E+00	8.712E+00	1.599E+01
	Ortalama	2.201E+01	9.950E-02	7.293E+00	1.628E+01	2.411E+01
	Std	4.780E+00	3.036E-01	2.218E+00	3.378E+00	5.596E+00
F7	En iyi	5.949E-05	2.146E-10	4.544E-04	1.978E-02	6.961E-02
	Ortalama	1.432E-03	8.150E-10	1.393E-03	1.108E+00	6.222E+00
	Std	2.112E-03	3.700E-10	1.196E-03	1.344E+00	4.918E+00
F8	En iyi	2.919E-04	0.000E+00	4.736E-06	1.954E-02	1.141E-01
	Ortalama	4.879E-03	6.716E-14	4.698E-05	8.971E-02	4.709E-01
	Std	7.333E-03	2.230E-13	4.191E-05	5.810E-02	2.897E-01

Tablo 4.9. Kesme kuvveti parametresi farklı değerlerinin test fonksiyonlarındaki çalışma sürelerinin karşılaştırması

		1	2	3	4	5
F1	En iyi	3.44	3.51	3.55	3.58	3.57
	Ortalama	3.56	3.56	3.60	3.61	3.60
	Std	0.13	0.03	0.05	0.02	0.02
F2	En iyi	3.51	3.42	3.41	3.41	3.42
	Ortalama	3.60	3.46	3.46	3.47	3.45
	Std	0.04	0.02	0.06	0.08	0.02
F3	En iyi	3.63	3.71	3.74	3.72	3.74
	Ortalama	3.66	3.76	3.77	3.77	3.77
	Std	0.02	0.03	0.02	0.02	0.02
F4	En iyi	4.25	4.29	4.27	4.30	4.30
	Ortalama	4.29	4.31	4.30	4.34	4.34
	Std	0.02	0.01	0.02	0.06	0.02
F5	En iyi	3.92	3.89	3.91	3.92	3.92
	Ortalama	3.96	3.93	3.95	3.96	3.95
	Std	0.02	0.02	0.02	0.02	0.02
F6	En iyi	3.77	3.72	3.77	3.78	3.80
	Ortalama	3.82	3.76	3.82	3.82	3.84
	Std	0.02	0.02	0.03	0.02	0.02
F7	En iyi	3.85	3.80	3.86	3.90	3.91
	Ortalama	3.90	3.85	3.91	3.95	3.95
	Std	0.02	0.02	0.02	0.02	0.02
F8	En iyi	3.78	3.82	3.87	3.92	3.93
	Ortalama	3.82	3.87	3.92	3.95	3.97
	Std	0.02	0.03	0.03	0.02	0.02

Problem boyutu 30 olarak belirlenen bu çalışmada popülasyon, kesme kuvveti, enerji kaybı ve adaptasyon parametre değerleri sırasıyla 20, 2, 0.1 ve 0.5 sabit değerleri alındığında algoritmanın daha başarılı olduğu görülmektedir. Parametre değerlerini rastgele değil de sabit aldığımızda tablolardan da anlaşılabileceği üzere YAA'nın daha iyi performans sağlayacağı düşünüldüğünden dolayı bu çalışmada belirtilen sabit değerler kullanılarak rüzgâr türbini yerleştirme probleminde uygulanmıştır.

Rüzgâr türbini yerleştirme problemi, literatürde bulunan çalışmalarda olduğu gibi sabit rüzgâr hızı 12 m/sn ve tek yönlü olarak belirlendi. Rüzgâr çiftliği $2000 \times 2000 m^2$ boyutlara sahip bir alandan oluşmakta ve bu alan 10×10 ızgara karelere bölünerek rüzgâr türbinlerinin muhtemel yerleşebileceği yerler belirlenmiştir. Ayrıca rüzgâr türbinlerinden en iyi verimi alabilmek için ızgara sayısı artırmak, algoritmayı daha efektif kılacağından dolayı aynı alan 20×20 ızgaraya bölünerek türbinlerin optimum yerleştirilmesi

sağlanmıştır. Yani 10×10 'luk ızgara yapısında türbin yerleştirmek için 100 adet türbin alanı bulunurken, 20×20 'lik ızgara yapısında türbin yerleştirmek için 400 adet türbin alanı bulunmaktadır. Uluslararası standartlar çerçevesinde türbinlerin birbirine olan uzaklıkları 5 çaptır (5D) (Marmidis ve ark, 2008). Bu sebepten dolayı, türbinler arası mesafe türbin rotor çapının 5 katı mesafe korunarak türbin yerleştirme işlemi gerçekleştirilmiştir. Bu şart sağlanmak koşulu ile her bir karenin merkezine bir türbin yerleştirilebilir. Rüzgâr türbinlerinin hub yüksekliği ve rotor çap değerleri ile itme katsayısı ve arazi pürüzlülüğü değerleri Tablo 4.10'de verilmiştir.

Tablo 4.10. Rüzgar türbininin özellikleri (Pookpant ve Ongsakul, 2013).

Hub yüksekliği (z , m)	60
Rotor çapı (r_r , m)	40
İtme katsayısı (C_T)	0.88
Arazi pürüzlülüğü (Z_0 , m)	0.3

Önerilen 14 adet ikili yöntem, türbin yerleştirme problemine ayrı ayrı uygulanmıştır. Yerleştirme sonucu elde edilen toplam güç, uygunluk değeri ve türbin sayısı Tablo 4.11 ve 4.12'de 10×10 ve 20×20 ızgara olmak üzere ayrı ayrı verilmiştir. 10×10 ve 20×20 ızgaralara bölünmüş türbin çiftlikleri için V-şekilli ailesi fonksiyonlarının S-şekilli ailesi fonksiyonlara göre daha optimum bir yerleştirme yaptığı görülmektedir. 10×10 'luk alan için S-şekilli ailesi ve V-şekilli ailesi fonksiyonları içerisinde en iyi optimum yerleştirmeyi İYAA5(V1) yöntemi ile gerçekleştirilirken 20×20 'lik alan için ise İYAA6(V2) yöntemi ile gerçekleştirilmiştir. Bununla birlikte mod fonksiyonlarının, S-şekilli ve V-şekilli ailesi fonksiyonlarına göre üstün bir başarı sağladığı görülmektedir. İYAA9(Mod-A) yöntemi 14 adet ikili yöntemi içerisinde en iyi türbin yerleştirme işlemini gerçekleştirmiştir. Bu yöntem en başarılı sonucu vermesi nedeniyle ve kolaylık açısından çalışmanın bundan sonrasında İYAA olarak adlandırılmıştır.

Tablo 4.11. Önerilen yöntemlerin 10×10 ızgaralık türbin yerleştirme problemi için performanslarının karşılaştırılması

10×10 ızgara			
	Uygunluk değeri ($Cost/P_{tot}$) (10^{-3})	Toplam güç (P_{tot})	Türbin sayısı
İYAA1 (S1)	1.5663	14664	32
İYAA2 (S2)	1.5955	14496	32
İYAA3 (S3)	1.6234	14571	33
İYAA4 (S4)	1.6428	13446	30
İYAA5 (V1)	1.5542	14707	32
İYAA6 (V2)	1.5659	14312	31
İYAA7 (V3)	1.6036	14063	31
İYAA8 (V4)	1.6293	14012	32
İYAA9 (Mod-A)	1.5414	14667	31
İYAA10 (Mod-B)	1.5428	14317	30
İYAA11 (Ceil)	1.5691	14077	30
İYAA12 (Ceil-Round)	1.5713	14058	30
İYAA13 (Ceil-Floor)	1.5691	14077	30
İYAA14 (Round-Floor)	1.5736	14037	30

Tablo 4.12. Önerilen yöntemlerin 20×20 ızgaralık türbin yerleştirme problemi için performanslarının karşılaştırılması

20×20 ızgara			
	Uygunluk değeri ($Cost/P_{tot}$) (10^{-3})	Toplam güç (P_{tot})	Türbin sayısı
İYAA1 (S1)	1.5486	26826	64
İYAA2 (S2)	1.5700	25113	60
İYAA3 (S3)	1.5829	25555	61
İYAA4 (S4)	1.5783	26618	64
İYAA5 (V1)	1.5516	27275	64
İYAA6 (V2)	1.5339	25454	59
İYAA7 (V3)	1.5561	26320	62
İYAA8 (V4)	1.5864	25009	60
İYAA9 (Mod-A)	1.4054	23422	49
İYAA10 (Mod-B)	1.4060	22523	47
İYAA11 (Ceil)	1.4082	22488	47
İYAA12 (Ceil-Round)	1.4384	25556	55
İYAA13 (Ceil-Floor)	1.4088	23365	49
İYAA14 (Round-Floor)	1.4216	24046	51

10×10 ızgara alanına rüzgâr türbini yerleştirme problemi ile ilgili literatürde yapılan çalışmalar ve önerilen yöntemlerin sonuçları Tablo 4.13’de verilmiştir. 20×20 ızgara alanına rüzgâr türbini yerleştirme problemi ile ilgili literatürde yapılan çalışmalar ve önerilen yöntemlerin sonuçları Tablo 4.14’te verilmiştir. 10×10 ızgaralık alana türbin yerleştirmek için uygulanan İYAA’nın uygunluk değeri diğer çalışmaların uygunluk değerleriyle kıyaslandığında İYAA diğer tüm algoritmalara göre üstünlük sağladığı görülmüştür. 20×20 ızgaralara bölünmüş alana uygulanan İYAA ise diğer bütün algoritmalara göre daha üstün bir performans göstermekle birlikte Parada’nın GA ile yapmış olduğu 20×20 ’lik alan ile kıyaslandığında önerilen yöntemin çok iyi bir yerleştirme sağladığı gözükmemektedir. Bu sonuçlara bakıldığında İYAA’nın ikili türbin

yerleştirme problemi için çok uygun bir algoritma olduğu söylenebilir. Parada'nın çalışmasında, Jensen uyandırma model yerine Gaussian uyandırma model kullanmıştır. Tablo 4.13 ve Tablo 4.14'te parantez içerisinde verilen değer Gaussian uyandırma model uygulanarak elde edilmiş değerdir. Jensen uyandırma model'e göre matematiksel hesaplama farklılığından dolayı uygunluk değeri daha iyi çıkmaktadır. 10×10 'luk alana baktığımızda Jensen uyandırma modele göre uygunluk değeri 0.0015436 iken Gaussian uyandırma modelin uygunluk değeri 0.0014940. Hâlbuki bu iki değer aynı yerleştirme ile elde edilmektedir. Bundan dolayı, bu iki değer birbirine eşit olarak düşünülmelidir. 20×20 'lik alanda Gaussian uyandırma modeline göre uygunluk değeri daha düşük çıkmasına rağmen İYAA daha iyi bir sonuç bulmuştur.

Tablo 4.13. 10×10 ızgara alanına rüzgâr türbini yerleştirme problemi ile ilgili literatürde yapılan çalışmalar ve önerilen yöntemin sonuçlarının karşılaştırılması.

10×10 ızgara			
Algoritmalar	Uygunluk değeri ($Cost/P_{tot}$) (10^{-3})	Toplam güç (P_{tot})	Türbin sayısı
GA (Mosetti ve ark, 1994)	1.6197	12352	26
GA (Grady ve ark, 2005)	1.5436	14310	30
MILP (Shakoor ve ark, 2016)	1.5436	14310	30
BPSO-TVAC (Pookpant ve Ongsakul, 2013)	1.5436	14310	30
Lazy Greedy (Changshui ve ark, 2011)	1.5436	14310	30
GA (Parada ve ark, 2017)	1.5436 (1.4940)	14310 (14785)	30
İYAA (önerilen algoritma) (Beskirli ve ark, 2017)	1.5414	14667	31

Not: Parantez içindeki değerler doğrudan Parada tarafından elde edilen sonuçlara karşılık gelir.

Tablo 4.14. 20×20 ızgara alanına rüzgâr türbini yerleştirme problemi ile ilgili literatürde yapılan çalışmalar ve önerilen yöntemin sonuçlarının karşılaştırılması.

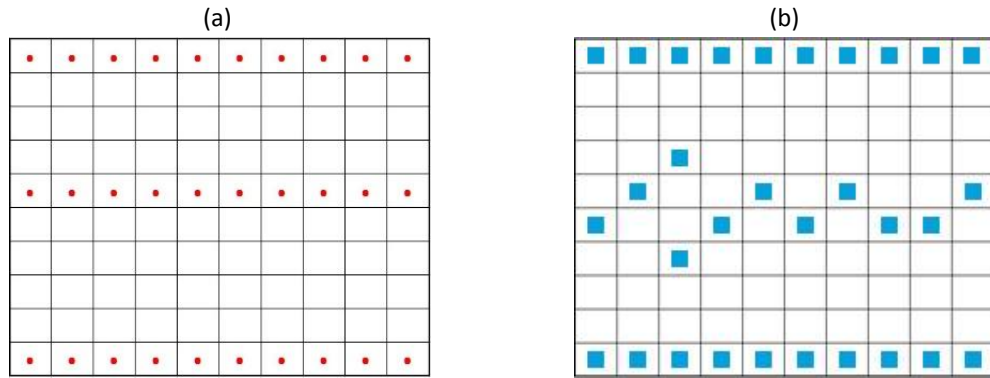
20×20 ızgara			
Algoritmalar	Uygunluk değeri ($Cost/P_{tot}$) (10^{-3})	Toplam güç (P_{tot})	Türbin sayısı
GA (Parada ve ark, 2017)	-- (1.4390)	-- (19052)	30
İYAA (önerilen algoritma) (Beskirli ve ark, 2017)	1.4054	23422	49

Not: Parantez içindeki değerler doğrudan Parada tarafından elde edilen sonuçlara karşılık gelir.

10 × 10'luk alan için yapılan türbin yerleştirme işlemi önerilen yöntemin uygunluk değeri Grady ve Parada'nın uygunluk değerine göre %0.14 daha başarılıdır ve toplam güçte ise %2,5 artış göstermiştir. 20 × 20'lik alan için ise önerilen yöntemin uygunluk değeri Grady'nin 10 × 10'luk alanına göre %9,8 artış sağlamıştır. Aynı zamanda toplam güçte ise %61 oranında artarak İYAA'nın türbin yerleştirme işleminde daha iyi bir çözüm ortaya koyduğu görülmüştür. Parada'nın 20 × 20'lik çalışması ile karşılaştırdığımızda ise uygunluk değeri belli olmadığından sadece toplam güç karşılaştırılması yapılmıştır ve önerilen yöntem ile %22,9 oranında daha fazla enerji üretimi gerçekleştirdiği görülmüştür.

Önerilen yöntemin 10 × 10 ızgaralara bölünmüş alana uygulanarak elde edilen sonuçlar literatürdeki çalışmalarla kıyaslandığında başarısı kanıtlanan İYAA, aynı alanın 20 × 20'lik ızgaraya bölünmesi ile daha başarılı yerleştirme yaptığı görülmektedir. Bunun sebebi, problemin arama uzayını genişletilmiş olması ve İYAA'nın arama uzayını daha efektif şekilde taramasıdır.

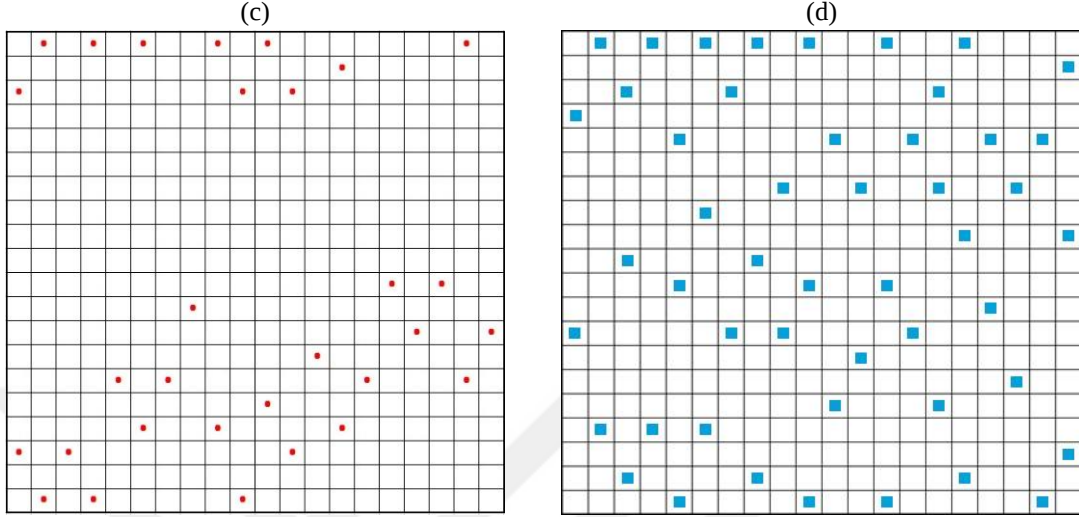
Parada ve Grady'nin çalışmalarında 30 türbin bulunmakta ve her ikisi de aynı yerleştirme işlemi gerçekleştirmektedir. Önerilen yöntemde ise aynı alanda 31 türbin yerleştirilmiştir. Önerilen yöntem, Parada ve Grady'nin çalışmasına göre 1 türbin fazla olmasına rağmen türbinler arası uyandırma etkisini en aza indirmek için türbin alanının orta kısımlarına yerleştirilen türbinlerde farklı bir yerleşim dizilimi gerçekleştirmiş ve daha optimum sonuç bulmuştur. Yerleşim planları Şekil 4.1'de gösterilmiştir.



Şekil 4.1. 10 × 10 ızgaralı rüzgâr santrali konfigürasyonu: (a) Parada ve Grady, (b) İYAA.

Şekil 4.2'de gösterilen yerleşim planlarında, Parada 30 türbin kullanmışken, önerilen yöntemde ise Parada'ya göre fazladan 19 türbin daha kullanmıştır. Parada'nın kullanmış olduğu algoritma ile yapılan yerleşimde türbinler ön ve arka kısımlara yoğunlaşırken yerleştirme alanının orta kısımları kullanılmamıştır. Önerilen yöntemde

türbin sayısı fazla olmasına rağmen, türbinler arası mesafe dikkate alınarak 49 türbini daha verimli bir şekilde yerleştirilmiştir. Önerilen yöntem, maliyet ve toplam enerji açısından Parada'nın çalışmasına göre daha iyi bir sonuç bulmuştur.



Şekil 4.2. 20×20 ızgaralı rüzgâr santrali konfigürasyonu: (c) Parada, (d) İYAA.

4.1.2. İYOO Sonuçları

Literatür verilerine göre yapılan bu çalışma, sabit rüzgâr hızı 12 m/sn ve tek yönlü olarak belirlenmiştir. Rüzgâr çiftliği 2000m $\times$ 2000m'lik boyuta sahip bir alandan oluşmaktadır. Bu alan, 10×10 ve 20×20 'lik kare ızgaralara bölünerek türbinlerin optimum yerleştirilmesi sağlanmıştır. Bazı çalışmalar dikkate alındığında türbinler arası mesafe 4 çap (D) mesafe olarak gözükmemektedir (Pérez ve ark, 2013). Ancak literatürde çoğu çalışmada türbinler arası mesafe 5 çap (D) kullanıldığından bu çalışma için türbinler arası mesafe 5 çap alınmıştır (Marmidis ve ark, 2008). Bu sebepten dolayı, türbinler arası mesafe türbin rotor çapının 5 katı mesafe korunarak türbin yerleştirme işlemi gerçekleştirilmiştir. Bu şart sağlanmak koşulu ile bu çalışmada her bir karenin merkezine bir türbin yerleştirilmiştir. İYOO yöntemi için popülasyon büyüklüğü 50, maksimum iterasyon sayısı 1500 olarak deneysel çalışma gerçekleştirilmiştir. Maksimum fonksiyon değerlendirme sayısı (MaxFES) ise literatürdeki diğer çalışmalardaki gibi 6×10^5 olarak belirlenmiştir. Bu işlemler türbin yerleştirme problemi için İYOO yöntemi 10 kez bağımsız şekilde çalıştırılarak gerçekleştirilmiştir. Rüzgâr türbinlerinin hub yüksekliği ve rotor çapı değerleri ile itme katsayısı ve arazi pürüzlülüğü değerleri Tablo 4.15'de İYOO yönteminde kullanılan parametreler Tablo 4.16'da verilmiştir. 14 adet ikili yöntem bu

değerler kullanılarak türbin yerleştirme problemine uygulanmıştır ve elde edilen sonuçlar sırasıyla Tablo 4.17 ve 4.18’de verilmiştir.

Tablo 4.15. İYOO yönteminin parametre değerleri

Parametre adı	Değeri
Maksimum tohum sayısı	5
Minimum tohum sayısı	1
Maksimum popülasyon	100
Varyans azaltma katsayısı	2
Standart sapma başlangıç değeri	0.5
Standart sapma son değeri	0.001

Tablo 4.16. Rüzgar türbininin özellikleri (Pookpant ve Oongsakul, 2013).

Hub yüksekli (z, m)	60
Rotor çapı (r_r, m)	40
İtme katsayısı (C_T)	0.88
Arazi pürüzlülüğü (Z_0)	0.3

Tablo 4.17’de 14 farklı İYOO yöntemi 10×10 ızgara yapısına uygulanarak elde edilmiş sonuçlar yer almaktadır. Elde edilen sonuçlar incelendiğinde S-şekilli ailesi fonksiyonları V-şekilli ailesi fonksiyonlarına göre daha iyi uygunluk değeri elde etmiştir. Ceil(İYOO11) yöntemi ise tüm algoritmalara göre daha üstün bir performans sergileyerek en iyi uygunluk değerini bulmuştur.

Tablo 4.17. Önerilen yöntemlerin 10×10 ızgaralık türbin yerleştirme problemi için performanslarının karşılaştırılması

10×10 ızgara			
	Uygunluk değeri ($Cost/P_{tot}$)	Toplam güç (P_{tot})	Türbin sayısı
İYOO1 (S1)	1.5549	14875	32
İYOO2 (S2)	1.5661	14768	32
İYOO3 (S3)	1.6285	16187	38
İYOO4 (S4)	1.6413	15059	35
İYOO5 (V1)	1.6195	14932	34
İYOO6 (V2)	1.6053	14084	31
İYOO7 (V3)	1.6011	15776	36
İYOO8 (V4)	1.6344	15789	37
İYOO9 (Mod-A)	1.5539	14549	31
İYOO10 (Mod-B)	1.5497	14924	32
İYOO11 (Ceil)	1.5413	14667	31
İYOO12 (Ceil-Round)	1.5435	14647	31
İYOO13 (Ceil-Floor)	1.5434	14647	31
İYOO14 (Round-Floor)	1.5429	14316	30

Tablo 4.18’de 14 farklı İYOO yöntemi 20×20 ızgara yapısına uygulanarak elde edilmiş sonuçlar yer almaktadır. V-şekilli ailesi algoritmalarının performansının tüm algoritmalara göre daha düşük olduğu görülmüştür. $2\text{km} \times 2\text{km}$ ’lik alana S1 (İYOO1) ve Mod-A (İYOO9) yöntemleri toplam 55 türbin yerleştirirken, türbin yerleştirme işlemi sonrası elde edilen uygunluk sonuçlarının birbirine yakın değerler aldığı gözlenmiştir. Ancak Ceil (İYOO11) yöntemi, en yüksek performansa ulaşarak tüm algoritmalar arasında en iyi uygunluk değerini bulmuştur. Bunlara ek olarak, hibrit yapı olan Ceil-round (İYOO12), ceil-floor (İYOO13) ve Round-floor (İYOO14) yöntemleri İYOO11 yöntemine yakın uygunluk değerlerine sahip olmuştur. Bu nedenle, her iki ızgara sisteminde (10×10 ve 20×20) Ceil (İYOO11) yöntemi ile en iyi sonuçlar elde edilmiştir.

Bu çalışmada, türbinleri en iyi yerleştirme işlemi Ceil transfer fonksiyonu kullanılarak İYOO11 yöntemi ile elde edilmiştir. Bu aşamadan sonra bu yöntem genel olarak İYOO olarak adlandırılacaktır.

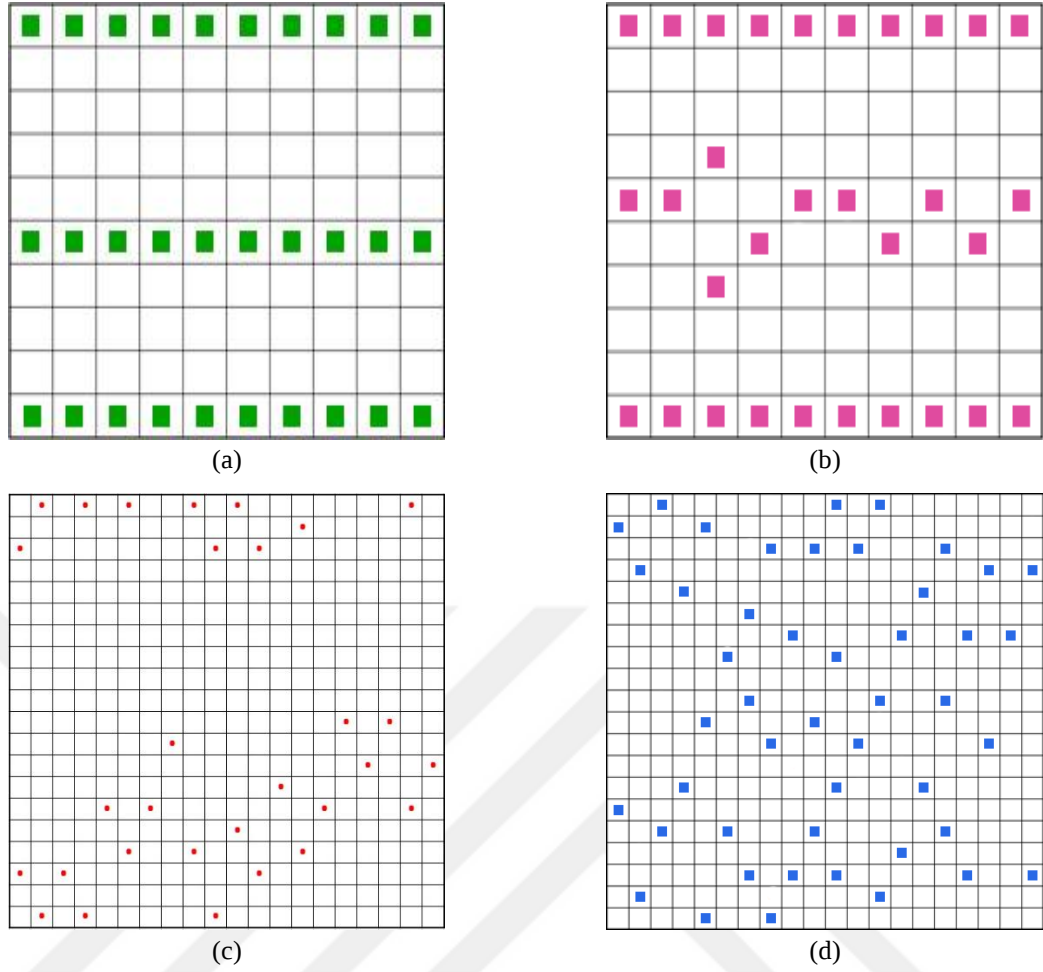
Tablo 4.18. Önerilen yöntemlerin 20×20 ızgaralık türbin yerleştirme problemi için performanslarının karşılaştırılması

20 × 20 ızgara			
	Uygunluk değeri ($Cost/P_{tot}$) (10^{-3})	Toplam güç (P_{tot})	Türbin sayısı
İYOO1 (S1)	1.4595	25187	55
İYOO2 (S2)	1.5029	25329	57
İYOO3 (S3)	1.5623	26473	62
İYOO4 (S4)	1.5953	28002	67
İYOO5 (V1)	1.5877	27302	65
İYOO6 (V2)	1.5869	29411	70
İYOO7 (V3)	1.5676	25963	61
İYOO8 (V4)	1.5966	27981	67
İYOO9 (Mod-A)	1.4763	24902	55
İYOO10 (Mod-B)	1.5102	26512	60
İYOO11 (Ceil)	1.4149	22382	47
İYOO12 (Ceil-Round)	1.4437	24567	53
İYOO13 (Ceil-Floor)	1.4394	23308	50
İYOO14 (Round-Floor)	1.4376	24670	53

Hem 10×10 hem de 20×20 'lik ızgaraların kullanıldığı alan 2×2 km'den oluşmaktadır. Literatürde en çok kullanılan ızgara yerleştirme modeli olarak 10×10 gözükmektedir. Ancak GA ile 20×20 'lik ızgara ile ilgili çalışma yapılmıştır (Parada ve ark, 2017). Bu çalışmada Tablo 4.19 ve 4.20'ye eklenmiştir. 10×10 'luk ızgarada ızgaralar arası mesafe 200 m'dir. Bu yüzden türbinlerin yerleşeceği maksimum 100 tane yer bulunmaktadır. 20×20 'lik ızgarada ise ızgaralar arası mesafe 100 m olup türbinlerin yerleşebileceği maksimum 400 tane yer bulunmaktadır. Uyandırma etkisi yarıçapı $10 \times$

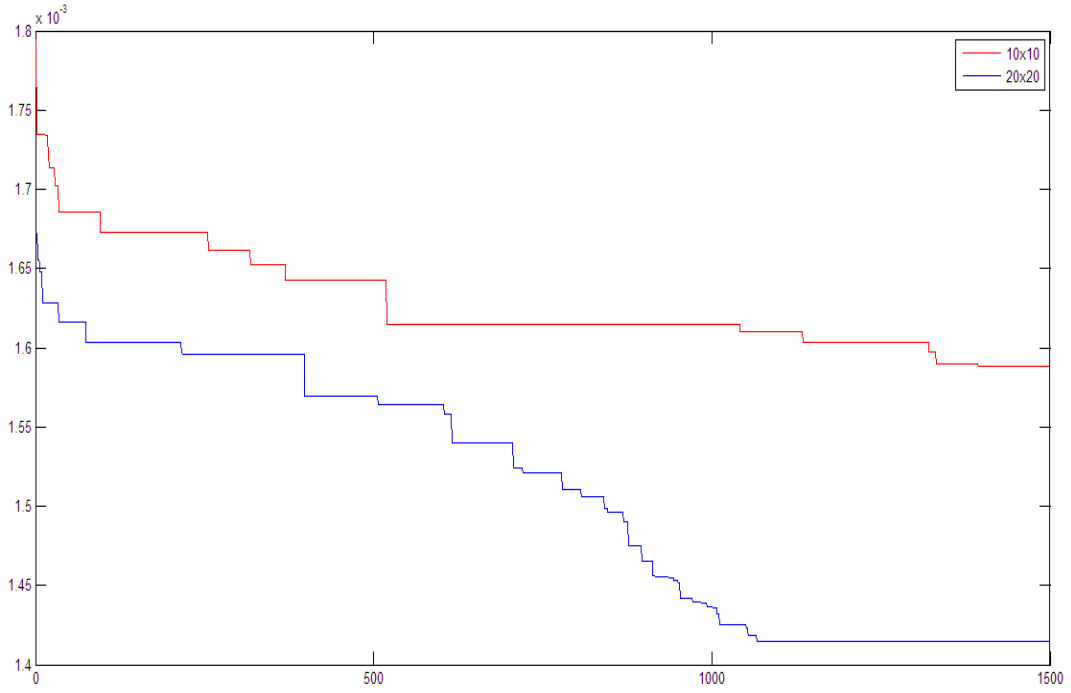
10 ızgara hücresinin boyutundan daha küçüktür. Böylece türbinler radyal yönde ve 20×20 ızgara hücrelerinde daha yakın bir şekilde konumlandırılmıştır (Parada ve ark, 2017). 20×20 ızgaradaki türbinlerin potansiyel konumları birbirine yakın olmasına rağmen, türbinler arasındaki minimum mesafe korunmuştur.

10×10 ızgara dikkate alındığında, önerilen algoritma, Grady tarafından yapılan çalışmadan çok az bir iyi sonuç verirken, Mosetti ve arkadaşları tarafından yapılan çalışmadan daha iyi bir sonuç elde etmiştir. Bununla birlikte, 20×20 'lik bir ızgaranın literatürdekilerle karşılaştırıldığında, önerilen yöntemin Mosetti ve Grady tarafından yapılan çalışmalardan daha iyi bir performansa sahip olduğu ve daha yüksek bir başarı oranının ortaya koyduğu görülmüştür. 20×20 'lik bir ızgara, arama alanını daha etkili bir şekilde kullandığı için, yani 10×10 ızgaraya kıyasla bağımsız türbin düzenlerinin sayısı önemli ölçüde arttığından İYOO yönteminin performansı önemli ölçüde artmıştır. Şekil 4.3'de, 20×20 'lik bir ızgara 10×10 ızgaradan daha esnek bir düzen elde edildiği görülmektedir. Bu çalışmada, rüzgâr çiftliği için iki ızgara kullanılmıştır. İlki Grady tarafından önerilen 100 olası konumdan oluşan 10×10 ızgara metodudur. İkincisi, son çalışmalarda önerilen 400 potansiyel konuma sahip 20×20 ızgara metodudur. Wang, özellikle 20×20 'lik bir ızgara kullanmanın daha iyi sonuçlar vereceğini bildirmiştir (Wang ve ark, 2015).



Şekil 4.3. (a) Grady, (b) İYOO (10 × 10) ve (c) Parada, (d) İYOO (20 × 20) ile elde edilen rüzgâr çiftliği düzenleri

10 × 10 ızgara ve 20 × 20 ızgarayla elde edilen en iyi uygunluk değerlerinin yakınsama eğrileri karşılaştırarak Şekil 4.4'te sunulmuştur. Bu şekil, önerilen yöntemin yakınsaklık eğrisi yaklaşık 3500'üncü kuşağın (generation) istikrar kazandığını göstermektedir. Ayrıca türbin çiftliği daha fazla karesel hücreden oluşan ızgara yapısı ile ifade edildiğinde önerilen bu yöntemin daha iyi çözümler ürettiği görülmektedir.



Şekil 4.4. farklı ızgara yoğunluğu için İYOO yönteminin yakınsama eğrileri.

Literatürde yayınlanan 10×10 ve 20×20 ızgara yerleşimi ile yapılan rüzgâr türbini yerleştirme problemi sonuçları ile önerilen algorithmadan edilen sonuçlar Tablo 4.19 ve 4.20’de gösterilmiştir.

10×10 ızgara için Mosetti, Grady ve diğer metotlardan elde edilen türbin yerleştirme problemi sonuçları Tablo 4.19'daki önerilen algoritma ile karşılaştırılmıştır. Önerilen yöntemlerle elde edilen türbin yerleştirme işlemi için kullanılan uygunluk değeri, Mosetti'nin çalışmasından %5.09 daha iyi ve diğer algoritmalara göre %0.15 daha iyidir. Toplam güç açısından, İYOO yöntemi Mosetti'nin çalışmasından %18.74 oranında daha fazla enerji üretmiş ve diğer algoritmalarından ise %2.50 oranında bir artış olmuştur. Bu nedenle, bu sonuçlara göre, İYOO yönteminin 10×10 ızgarada türbin yerleşimi için daha iyi bir çözüm ile sonuçlandığı görülmüştür.

Tablo 4.19. 10×10 ızgara kullanılarak rüzgâr çiftlikleri için elde edilen sonuçların karşılaştırılması

Algoritma	Uygunluk değeri ($Cost/P_{tot}$) (10^{-3})	Toplam güç (P_{tot})	Türbin sayısı
GA (Mosetti ve ark, 1994)	1.6197	12352	26
GA (Grady ve ark, 2005)	1.5436	14310	30
MILP (Shakoor ve ark, 2016)	1.5436	14310	30
BPSO (Pookpant ve Ongsakul, 2013)	1.5436	14310	30
Lazy Greedy (Changshui ve ark, 2011)	1.5436	14310	30
GA (Parada ve ark, 2017)	1.5436 (1.4940)	14310 (14785)	30
İYOO (önerilen algoritma) (Beskirli ve ark, 2019)	1.5413	14667	31

20×20 'lik bir ızgara için, Parada'dan elde edilen türbin yerleştirme probleminin sonuçları, önerilen yöntemin sonuçları ile karşılaştırılarak Tablo 4.20'de sunulmuştur. 20×20 ızgara için parada ile yapılan sonuç karşılaştırmasında Parada'nın çalışmasının uygunluk değeri Jensen uyandırma etkisi modeline göre yapılmadığından karşılaştırma işlemi toplam güç değeri üzerinden yapılmıştır. Önerilen yöntem Parada'nın çalışmasına göre %46.27 daha fazla güç üretimi gerçekleştirmiştir. İYOO'nun 20×20 ızgara uygulaması 10×10 ızgara sisteminden daha iyi bir yerleştirmenin yanı sıra Parada'nın GA ile 20×20 ızgara kullanarak yapılan çalışmanın sonucuna kıyasla çok iyi bir yerleştirme elde etmiştir. Bu sonuçlar göz önüne alındığında, İYOO'nun ikili türbin yerleştirme problemi için uygun bir yöntem olduğu söylenebilir. Parada tarafından yapılan çalışmada, Jensen uyandırma etkisi modelinden ziyade Gaussian uyandırma etkisi modeli kullanılmıştır. Tablo 4.19 ve 4.20'de parantez içinde verilen değerler, Parada tarafından Gaussian uyandırma etkisi modeli uygulanarak elde edilmiştir. Gaussian uyandırma etkisi modeline göre elde edilen uygunluk değeri, matematiksel hesaplamadaki farklılık nedeniyle Jensen uyandırma etkisi modeline göre daha iyi bir sonuç gibi görünmektedir. 10×10 ızgara dikkate alındığında, Jensen uyandırma etkisi modeli için 0,0015436 ve Gaussian uyandırma etkisi modeli için 0,0014940 uygunluk değerleridir. Ancak, bu iki uygunluk değeri aynı yerleştirme ile elde edilir; bu nedenle, birbirlerine eşit olduğu düşünülmelidir. 20×20 ızgara için ise, Gaussian uyandırma etkisi

modeline göre uygunluk değeri daha küçük olmasına rağmen, İYOO daha iyi bir sonuç vermektedir.

Tablo 4.20. 20×20 ızgara kullanılarak rüzgâr çiftlikleri için elde edilen sonuçların karşılaştırılması

Algoritma	Uygunluk değeri ($Cost/P_{tot}$) (10^{-3})	Toplam güç (P_{tot})	Türbin sayısı
GA (Parada ve ark, 2017)	-- (1.4390)	-- (15302)	30
İYOO (önerilen algoritma) (Beskirli ve ark, 2019)	1.4149	22382	47

Hem 10×10 hem de 20×20 ızgara sistemlerini birlikte incelendiğinde, İYOO tarafından elde edilen türbin yerleştirme işlemi için uygunluk değeri Grady'ninkinden %9.1 daha iyidir. Toplam güç açısından, İYOO yöntemi Grady'den %56.41 oranında daha fazla enerji üretmiştir. Bu nedenle, bu sonuçlara göre, İYOO yönteminin türbin yerleşimi için diğer algoritmalarından daha iyi bir çözüm ile sonuçlandığı görülmüştür.

4.2. İkili – Gerçek Kodlama Sonuçları

Bu bölümde MDA yönteminin ikili-gerçek yöntem ile rüzgâr türbinine uygulanarak elde edilen sonucu literatürde bulunan çalışmaların sonuçları ile karşılaştırılmıştır.

4.2.1. MDA Sonuçları

Bu bölümde rüzgâr türbini yerleştirme problemi için MDA yöntemi kullanılarak mikro yerleştirme işlemi gerçekleştirilmiştir. Türbin sahası olarak $2000m \times 2000m$ boyutlu kare bir alan kullanılmıştır. Arazinin zemin pürüzlülüğü 0.3m alınmıştır. Rüzgâr çiftliğinde rüzgâr türbini, güç eğrisi, uyandırma etkisi modeli ve optimal eğri gibi kullanılan veya elde edilen değerler ve şekiller literatürde yer alan çalışmalardan (Mosetti ve ark, 1994, Grady ve ark, 2005, Chen ve ark, 2014, Mittal ve ark, 2016) referans alınmıştır. Bu özellikler kullanılarak elde edilmek istenen temel amaç, türbinler arası mesafenin 5D veya 200 m'nin altında olmaması koşulu ile türbinlerin birbirine olan uyandırma etkisini en aza indirmektedir. Uyandırma etkisi ile, toplam enerjinin (P_{tot}) maksimum olması sağlanırken maliyetin (COST) minimum düzeyde olması ile amaç fonksiyonunu ($COST/P_{tot}$) minimize ederek en iyi sonuç değeri elde edilmek

istenmektedir (Chen ve ark, 2014, Mittal ve ark, 2016). Tablo 4.21’de rüzgâr çiftliği, rüzgâr türbini ve uyandırma etkisi modelinin özellikleri verilmiştir.

Tablo 4.21. Rüzgar çiftliği, rüzgar türbini ve uyandırma etkisi modelinin özellikleri (Mittal ve ark, 2016)

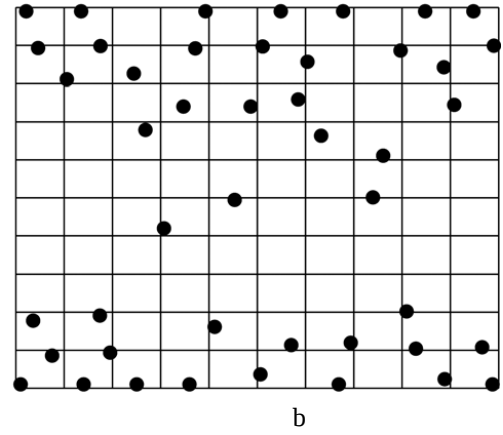
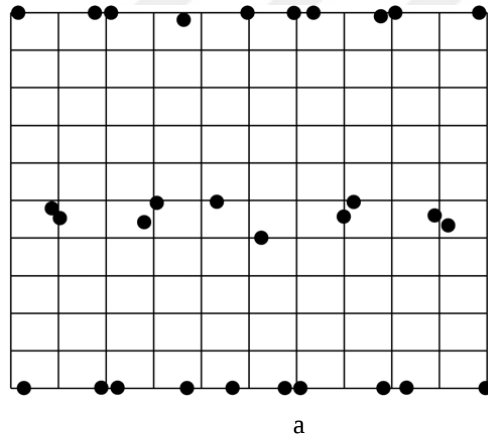
Rüzgâr çiftliği bilgileri	
Çiftlik büyüklüğü (m ²)	2000 x 2000
Rüzgâr türbini özellikleri	
Türbin çapı (m)	40
Türbin anma gücü (kW)	630
Hub yüksekliği (Z) (m)	60
İtme katsayısı (C _T)	0.88
Yüzey pürüzlülüğü (Z ₀) (m)	0.3
Uyandırma model bilgisi	
Model	Jensen

MDA yöntemi ile diğer yöntemler (Mosetti ve ark, 1994, Grady ve ark, 2005, Wan ve ark, 2009, Chen ve ark, 2014, Mittal ve ark, 2016) kıyaslandığında MDA yöntemi daha iyi bir uygunluk değeri (COST/P_{tot}) bulmuş olup Tablo 4.22’de sunulmuştur. Bu tabloya baktığımızda, literatürde ikili-gerçek kodlama kullanılan Chen ve ark.’nın çalışması uygunluk değerini en düşük elde ettikleri için literatür çalışmaları içerisinde başarılı olmuştur. İkili-gerçek kodlama, hem belirli bir rüzgâr çiftliği arazisine ne kadar türbin sayısına ihtiyaç olduğunu bulur hem de koordinat sistemi sürekli değerlerden oluşması nedeniyle türbinleri istenilen pozisyona yerleştirme imkânı sağlar. İkili-gerçek kod kullanılarak MDA yöntemi probleme uygulandığında elde edilen sonucun daha başarılı bir yerleştirme gerçekleştirdiği COST/P_{tot} değerinden görülmektedir. MDA yöntemi diğer algoritmalarından daha başarılı olduğu aynı zamanda bu yöntem ile toplam gücün daha fazla üretildiği görülmektedir. Türbinler arası mesafe dikkate alınarak yapılan yerleştirmede MDA, Chen’e göre 2 türbin fazla yerleştirmiştir. MDA yerleşim işlemini etkili bir şekilde yaptığından maliyet artışı olmasına rağmen uyandırma etkisini iyi yöneterek maksimum enerji üretmiştir. MDA yöntemi, uygunluk değerini diğer algoritmaların sonuçlarına göre daha da azaltarak türbinlerin yerleştirme verimliliği

artmıştır. Wan gerçek kodlama yöntemini kullanmıştır ve aynı zamanda rüzgâr türbini sayısı ve yerleşimi Grady ile aynı olmasına rağmen tam güç çıkışı ve verimlilik Grady'e göre daha iyi olmuştur. Ancak Wan ve ark. türbinler arası uzaklık faktörünü dikkate almadan hesaplama yapmışlardır ve bazı türbinlerin birbirine çok yakın oldukları Şekil 4.5'de görülmektedir (Chen ve ark, 2014). Kullanılan MDA yöntemi, Mosetti'ye göre %20.5, Grady'e göre %14.8, Wan'a göre %7.6, Mittal'e göre %7 Chen'ne göre %0.07 oranında daha iyi sonuç elde etmiştir.

Tablo 4.22. MDA yöntemi ile çeşitli metodolojilerin karşılaştırılması

	Mosetti ve ark. (Mosetti ve ark, 1994)	Grady ve ark. (Grady ve ark, 2005)	Wan ve ark. (Wan ve ark, 2009)	Mittal ve ark. (Mittal ve ark, 2016)	Chen ve ark. (Chen ve ark, 2014)	MDA Yöntemi (Önerilen Yöntem)
$COST/P_{tot} (10^{-3})$	1.6197	1.5436	1.4475	1.4386	1.3456	1.3447
Toplam güç P_{tot}	12352	14310	15262	20742.54	22624.3	23550.38
Türbin sayısı	26	30	30	44	45	47

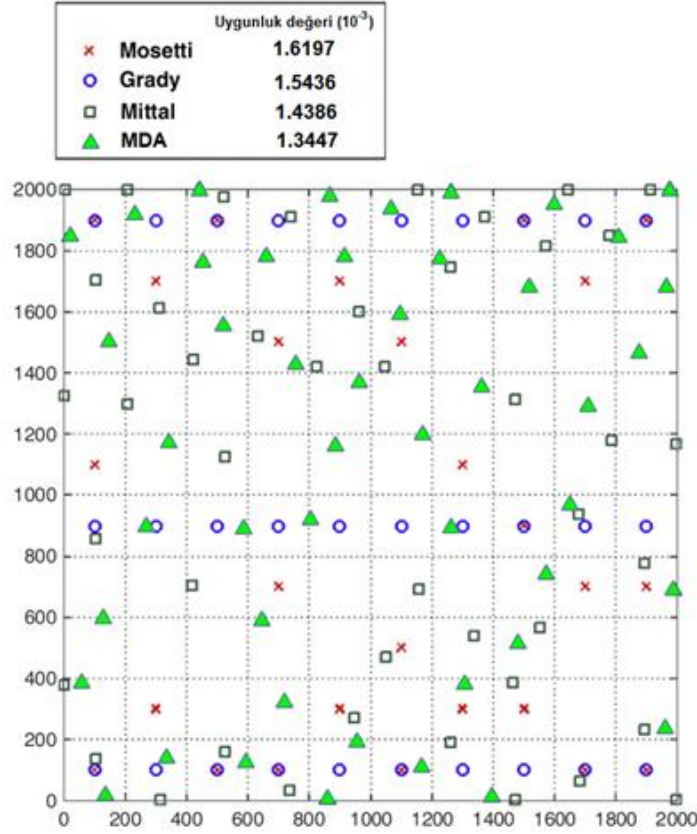


Şekil 4.5. En uygun yerleştirme için yapılan çalışmalar (a) Wan ve ark. ve (b) Chen ve ark.

Tablo 4.23. MDA yöntemi ile temel DA algoritmasının sonuçları

	$COST/P_{tot} (10^{-3})$	Toplam güç (P_{tot})	Türbin sayısı
Temel DA	1.3491	23017	46
MDA Yöntemi (Önerilen Yöntem)	1.3447	23550.38	47

MDA yöntemi ve diğer algoritmalar tarafından optimize edilen türbin yerleştirme probleminde türbinlerin konumları Şekil 4.6'da simüle edilmiştir. MDA yöntemi ile rüzgâr türbini alanına 47 adet türbin yerleştirmiştir.

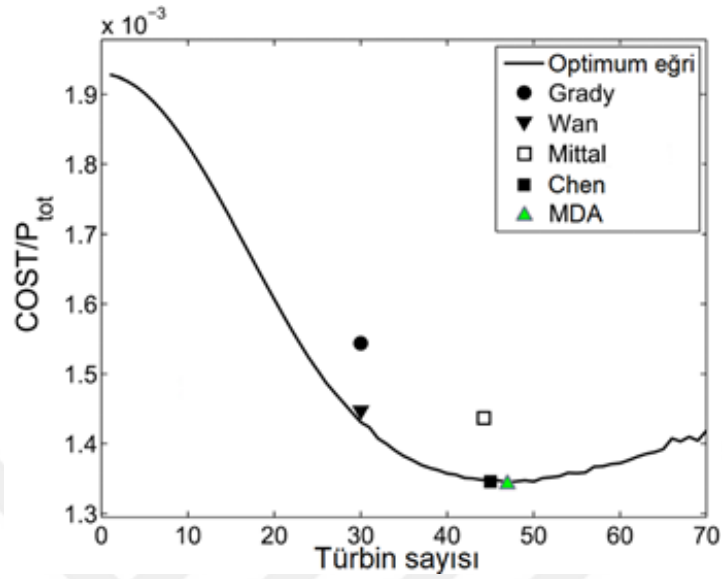


Şekil 4.6. MDA ile çeşitli algoritmalar için türbin düzenleri (Mittal ve ark, 2016). (Algoritmalar farklı işaretçiler tarafından gösterilmiştir)

Çalışma alanına 26 adet türbin yerleştiren Mosetti'nin çalışması verimlilik açısından çok geride kalmakta ve alanı etkili şekilde kullanamamaktadır. Grady, Mosetti'ye göre daha iyi bir sonuç elde etse de türbinlerin yerleşimlerini tek sıra ve paralel şekilde gerçekleştirerek rüzgâr türbinlerinin dağılımını başarılı bir şekilde sağlayamamaktadır. Mittal ise 44 adet türbin yerleştirerek verimliliği arttırmış olmasına rağmen çalışma alanının orta bölümünü etkin olarak kullanamamış ve uyandırma etkisinde daha fazla kalmıştır. MDA yönteminde ise yerleştirilen türbin sayısı artmasına rağmen efektif bir yerleştirme yaparak başarılı bir yerleşim gerçekleştirmiştir.

Chen ve ark. (Chen ve ark, 2014) tarafından 2014'te yaptığı çalışmasında verilen rüzgâr türbini yerleştirme probleminin optimal eğrisi Şekil 4.7'de gösterilmiştir. Eğri, gerçek kodlama yöntemini kullanarak türbinlerin optimize edilmiş objektif değerlerini

göstermektedir. Grady, Wan, Mittal, Chen ve MDA metodu çalışmaların sonuçları da Şekil 4.7’de optimal eğri üzerinde gösterilmiştir.



Şekil 4.7. Optimal eğri (Chen ve ark, 2014)

Grady'nin türbin yerleştirme sonucu optimal eğriden uzak kaldığı görülmektedir. Wan'nın türbin yerleştirme sonucu optimal eğriye çok yakındır ancak Wan sadece sabit sayıda türbini optimize etmiştir ve türbinler arası mesafeyi dikkate almamıştır. Türbinlerin sabit sayıda alınması optimal bir yerleştirme sağlamazken, ikili-gerçek kodlama ile en iyi sayıda türbin ve daha iyi bir yerleştirme elde edilebilir (Chen ve ark, 2014). Chen, ikili-gerçek kodlama ile yapmış olduğu yerleştirme sonucu optimal eğri üzerindedir. Chen, çalışmasında ilk popülasyonu rastgele oluşturduğunu ve bununda popülasyon çeşitliliği elde ettiğinden çözümün iyi olduğunu savunmaktadır. MDA yöntemi optimal eğri üzerinde ve diğer çalışmalardan daha iyi bir sonuç elde etmiştir. MDA yönteminde ikili-gerçek kodlama kullanılmış olması, farklılaştırma üretim yöntemi ile başlangıç popülasyonunun hazırlanması ve yöntemlerin üzerinde gerçekleştirilen iyileştirmeler sayesinde daha başarılı olmuştur.

4.3. Tüm Deneysel Sonuçların Karşılaştırılması

Bu tez çalışmasında İYAA, İYOO yöntemlerinde ikili kodlama ve MDA yönteminde ikili-gerçek kodlama kullanılarak belirli bir alana rüzgâr türbinleri yerleştirme işlemi gerçekleştirilmiştir. Yerleştirme için yapılan deneysel çalışmaların

sonuçları ile literatürde yer alan çalışmaların sonuçları karşılaştırılmıştır. Tüm algoritmalar aynı bilgisayar üzerinde çalıştırılmış ve elde edilen sonuçlar tek tablo halinde Tablo 4.24'te sunulmuştur.

Tablo 4.24. Tüm deneysel sonuçların karşılaştırılması.

Metot	Kodlama Yöntemi	Izgara Yöntemi	Toplam Güç	Türbin Sayısı	Uygunluk Değeri (10^{-3})
Mosetti (10x10) (Mosetti ve ark, 1994)	İkili	10×10	12352	26	1.6197
Grady (10x10) (Grady ve ark, 2005)	İkili	10×10	14310	30	1.5436
İYOO (10x10) (Önerilen Yöntem) (Beskirli ve ark, 2019)	İkili	10×10	14667	31	1.5414
İYAA (10x10) (Önerilen Yöntem) (Beskirli ve ark, 2017)	İkili	10×10	14667	31	1.5414
Wan (10x10) (Wan ve ark, 2009)	İkili	10×10	15262	30	1.4475
Mittal ve ark. (Mittal ve ark, 2016)	İkili - Gerçek	-	20742.54	44	1.4386
İYOO (20x20) (Önerilen Yöntem) (Beskirli ve ark, 2019)	İkili	20×20	22382	47	1.4149
İYAA (20x20) (Önerilen Yöntem) (Shin ve ark, 2017)	İkili	20×20	23422	49	1.4054
Chen ve ark. (Gao ve ark, 2015)	İkili - Gerçek	-	22624.3	45	1.3456
Temel DA	İkili – Gerçek	-	23017	46	1.3491
MDA Yöntemi (Önerilen Yöntem)	İkili - Gerçek	-	23550.38	47	1.3447

Tablo 4.24'te literatürde yer alan algoritmalar ile bu tezde önerilen yöntemlerin tüm kodlama yöntemleri ile karşılaştırması verilmiştir. Tabloda hem ikili kodlama hem de ikili-gerçek kodlama yöntemleri birlikte karşılaştırılmıştır.

Önerilen yöntemlerin diğer algoritmalarından daha iyi bir sonuç ürettiği görülmektedir. İkili-gerçek kodlamada sürekli noktalar kullanıldığından ikili kodlama kullanılan yerleştirme işleminden daha iyi olması beklenmektedir. Ancak Mittal'in yaptığı çalışmada ikili-gerçek kodlama kullanmasına rağmen 20×20 'lik çalışmada kullanılan İYAA ve İYOO yöntemlerinin gerisinde kaldığı görülmektedir. Bundan dolayı İYAA ve İYOO yöntemlerinin daha etkili ve başarılı bir yerleştirme yaptığı görülmüştür. İkili-gerçek kodlama ile en başarılı ve verimli yerleştirmeyi yapan MDA yöntemi ve onu takip eden Chen'in çalışması rüzgâr türbini yerleştirme problemi için etkili ve kararlı algoritmalar olduğu görülmüştür. Özellikle önerilen MDA yöntemi ile türbinler verimli bir şekilde yerleştirilmiştir. Maliyet artışına rağmen uyandırma etkisi azaltılarak maksimum enerji üretilmiştir.

Sonuç olarak MDA yönteminin rüzgâr türbini yerleştirme problemi için uygun ve başarılı olduğu açıkça görülmektedir.

5. SONUÇLAR VE ÖNERİLER

5.1. Sonuçlar

Bu tez çalışmasında geleneksel algoritmalar ile çözümü mümkün olmayan rüzgâr türbini yerleştirme gerçek dünya optimizasyon problemi için bazı sezgisel algoritmalar üzerinde yapılan yenilikler ile çözümler önerilmiştir. Önerilen yöntemler çiftlik maliyetinin düşürülmesi ve üretilen enerjinin ise maksimum olması koşulunu sağlayacak rüzgâr çiftliğinde rüzgâr türbinlerinin optimum yerleşimini amaçlamıştır. Kullanılan yöntemlerde Jensen uyandırma etkisi modeli kullanılmıştır ayrıca türbinler arası mesafe 5 çap olarak korunmuştur. Rüzgâr türbini yerleştirme problemi için iki kodlama yöntemi bulunmaktadır. Birincisi ikili kodlama ikincisi ikili-gerçek kodlama yöntemidir. İkili kodlama yönteminde YAA ile YOO yöntemleri ikili hale getirilmiştir ve rüzgâr türbini yerleştirme problemine uygulanmıştır. Bu algoritmalar 2x2 km'lik alan için ve 10x10 ile 20x20'lik ızgara kullanılarak yapılmıştır. Önerilen yöntemlerin sonuçları, literatürdeki diğer çalışmaların sonuçları ile karşılaştırılmıştır. İkili kodlama ile en başarılı ve verimli yerleştirmeyi yapan İYAA ve onu takip eden İYOO yöntemlerinin rüzgâr türbini yerleştirme problemi için etkili ve kararlı algoritmalar olduğu görülmüştür. Önerilen İYAA'nın sonuçları literatürde bulunan diğer çalışmaların sonuçları ile karşılaştırılmıştır. Yapılan karşılaştırma sonucunda, önerilen yöntemin Grady'nin yapmış olduğu çalışmaya göre uygunluk değeri %9.8, toplam güç'te %61 artış sağladığı ve Parada'nın yapmış olduğu çalışmaya göre toplam güç'te %22.9 artış sağlayarak İYAA'nın türbin yerleştirme optimizasyon probleminde üstün başarı sağladığı görülmüştür. İkili-gerçek kodlama yöntemi ise türbin sayısı belirlememe ve sürekli koordinatlar ile türbinleri konumlandırma açısından katkı sağlamıştır. İkili-gerçek kodlama ile MDA yöntemi kullanılmıştır. MDA yönteminde başlangıçtaki türbinlerin santral alanına tam manasıyla dağılımının gerçekleşmesi için bir ön işlem gerçekleştirilerek yöntemin başarısı artırılmıştır. Ayrıca MDA yönteminde beş farklı çaprazlama tekniği kullanılmıştır bunun yanı sıra MDA yönteminde mutasyon işlemleri uygulanarak popülasyon çeşitliliği de sağlanmıştır. MDA yönteminin sonuçları diğer çalışmaların sonuçlarına göre daha efektif bir yerleştirme yaptığı görülmüştür. MDA yöntemi türbinlerin yerleşimini etkili bir şekilde gerçekleştirmesi sayesinde maliyet artışına rağmen uyandırma etkisini azaltarak maksimum enerji üretmiştir.

5.2. Öneriler

Rüzgâr türbini yerleştirme probleminin konumlandırma performansını arttırmak için farklı özelliklere sahip diğer yöntemler ile hibrit çalışmalar yapılabilir. Rüzgâr türbini yerleştirme problemi çok amaçlı bir problem haline getirilerek çok amaçlı optimizasyon yöntemleri ile çözülebilir. Ayrıca tek yönlü rüzgâr yerine çok yönlü karışık rüzgâr ile türbinlerin performansı karşılaştırılabilir. Bunla birlikte bu probleme dağ yükseklikleri eklenerek rüzgâr değişimi hesaplanıp amaç fonksiyon değeri incelenebilir.



KAYNAKLAR

- Abdulelah AJ, Shaker K, Sagheer AM, Jalab HA. A Dynamic Scatter Search Algorithm for Solving Traveling Salesman Problem, 117-24, Singapore.
- Ahmed A, Al-Amin R, Amin R, 2014. Design of static synchronous series compensator based damping controller employing invasive weed optimization algorithm. SpringerPlus, 3, 1, 394.
- Alaei R, Setak M, 2017. Selecting unique suppliers through winner determination in combinatorial reverse auction: Scatter search algorithm. Scientia Iranica. Transaction E, Industrial Engineering, 24, 6, 3297-307.
- Bajer D, Martinović G, Brest J, 2016. A population initialization method for evolutionary algorithms based on clustering and Cauchy deviates. Expert Systems With Applications, 60, 294-310.
- Barisal AK, Prusty RC, 2015. Large scale economic dispatch of power systems using oppositional invasive weed optimization. Applied Soft Computing, 29, 122-37.
- Bartl J, Pierella F, Sætrana L, 2012. Wake Measurements Behind an Array of Two Model Wind Turbines. Energy Procedia, 24, 305-12.
- Bastankhah M, Porté-Agel F, 2014. A new analytical model for wind-turbine wakes. Renewable Energy, 70, 116-23.
- Beskirli M, Koc I, Haklı H, Kodaz H, 2017. A new optimization algorithm for solving wind turbine placement problem: Binary artificial algae algorithm. Renewable Energy.
- Beskirli M, Koc İ, Kodaz H, 2016. Individual Parameter Selection Strategy for Artificial Algae Algorithm (AAA). 2016 9th International Conference on Computer and Electrical Engineering (ICCEE 2016).
- Beskirli M, Koc İ, Kodaz H, 2019. Optimal Placement of Wind Turbines Using Novel Binary Invasive Weed Optimization. Tehnički vjesnik, 26, 1.
- Bharadwaj VR, Siddhartha G, 2016. Invasive Weed Optimization for Economic Dispatch with Valve Point Effects. Journal of Engineering Science and Technology, 11, 2, 237-51.
- Bianchi FD, Mantz RJ, De Battista H, 2007. The Wind and Wind Turbines, Springer, p.
- Bilgili M, Şahin B, Şimşek E, 2010. Türkiye'nin güney, güneybatı ve batı bölgelerindeki rüzgar enerjisi potansiyeli. Journal of Thermal Science and Technology, 30, 1, 1-12.
- Chamorro LP, Porté-Agel F, 2009. A Wind-Tunnel Investigation of Wind-Turbine Wakes: Boundary-Layer Turbulence Effects. Boundary-Layer Meteorology, 132, 1, 129-49.
- Changshui Z, Guangdong H, Jun W, 2011. A fast algorithm based on the submodular property for optimization of wind turbine positioning. Renewable Energy, 36, 11, 2951-8.
- Chen K, Song M, Zhang X, 2014. Binary-real coding genetic algorithm for wind turbine positioning in wind farm. Journal of Renewable and Sustainable Energy, 6, 5, 053115.
- Crawford B, Soto R, Olivares-Suárez M, Palma W, Paredes F, Olguin E, Norero E, 2014. A binary coded firefly algorithm that solves the set covering problem. Rom. J. Inf. Sci. Technol, 17, 3, 252-64.
- Crespo A, Herná'ndez J, 1996. Turbulence characteristics in wind-turbine wakes. Journal of Wind Engineering and Industrial Aerodynamics, 61, 1, 71-85.

- Emami A, Noghreh P, 2010. New approach on optimization in placement of wind turbines within wind farm by genetic algorithms. *Renewable Energy*, 35, 7, 1559-64.
- Fried L, Shukla S, Sawyer S, 2012. Global wind report: annual market update 2011. GWEC.
- Gao W, Liu S, 2011. Improved artificial bee colony algorithm for global optimization. *Information Processing Letters*, 111, 17, 871-82.
- Gao X, Yang H, Lin L, Koo P, 2015. Wind turbine layout optimization using multi-population genetic algorithm and a case study in Hong Kong offshore. *Journal of Wind Engineering and Industrial Aerodynamics*, 139, 89-99.
- Glover F, 1977. Heuristics for integer programming using surrogate constraints. *Decision sciences*, 8, 1, 156-66.
- Glover F, Laguna M, Marti R, 2003. Scatter Search. In: *Advances in Evolutionary Computing: Theory and Applications*. Eds: Ghosh A, Tsutsui S. Berlin, Heidelberg: Springer Berlin Heidelberg, p. 519-37.
- González JS, Rodríguez AGG, Mora JC, Santos JR, Payan MB, 2010. Optimization of wind farm turbines layout using an evolutive algorithm. *Renewable energy*, 35, 8, 1671-81.
- Grady S, Hussaini M, Abdullah MM, 2005. Placement of wind turbines using genetic algorithms. *Renewable energy*, 30, 2, 259-70.
- GWEC, (2015). Global Wind Report 2015.
- GWEC, (2016). Global Wind Energy Outlook 2016.
- GWEC, (2016). Global Wind Report 2016.
- Haklı H, 2017. Arazi Toplulaştırma için Optimizasyon Tabanlı Yeni Bir Dağıtım ve Parselasyon Modelinin Geliştirilmesi, Selçuk Üniversitesi, Fen Bilimleri Enstitüsü.
- Held P, Kienhuis A, 1995. -DIV, FLOOR, CEIL, MOD and STEP Functions in Nested Loop Programs and Linearly Bounded Lattices. In: *Algorithms and Parallel VLSI Architectures III*. Eds: Elsevier, p. 271-82.
- Holland JH, 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, MIT press, p.
- Holttinen H, Ahlgrimm J, 2013. IEA WIND 2012 Annual Report, p.
- Ituarte-Villarreal CM, Espiritu JF, 2011. Optimization of wind turbine placement using a viral based optimization algorithm. *Procedia Computer Science*, 6, 469-74.
- Jensen NO, 1983. A note on wind generator interaction.
- Karimkashi S, Kishk AA, 2010. Invasive Weed Optimization and its Features in Electromagnetics. *IEEE Transactions on Antennas and Propagation*, 58, 4, 1269-78.
- Katarya R, Verma OP, 2017. Effectual recommendations using artificial algae algorithm and fuzzy c-mean. *Swarm and Evolutionary Computation*, 36, 52-61.
- Keskin BB, Üster H, 2007. A scatter search-based heuristic to locate capacitated transshipment points. *Computers & Operations Research*, 34, 10, 3112-25.
- Keung PK, Li P, Banakar H, Ooi BT, 2009. Kinetic Energy of Wind-Turbine Generators for System Frequency Support. *IEEE Transactions on Power Systems*, 24, 1, 279-87.
- Kiran MS, 2015. The continuous artificial bee colony algorithm for binary optimization. *Applied Soft Computing*, 33, 15-23.
- Koç İ, 2016. Big Bang-Big Crunch Optimization Algorithm for Solving the Uncapacitated Facility Location Problem. *Intelligent Systems and Applications in Engineering*, 4, Special Issue, 185-9.
- Korkmaz S, Kiran MS, 2018. An artificial algae algorithm with stigmergic behavior for binary optimization. *Applied Soft Computing*, 64, 627-40.

- Kumar M, Dhillon J, 2018. Hybrid artificial algae algorithm for economic load dispatch. *Applied Soft Computing*.
- Küçükyavuz S, Sen S, 2017. An Introduction to Two-Stage Stochastic Mixed-Integer Programming. *INFORMS TutORials in Operations Research*.
- Long H, Eghlimi M, Zhang Z, 2017. Configuration Optimization and Analysis of a Large Scale PV/Wind System. *IEEE Transactions on Sustainable Energy*, 8, 1, 84-93.
- Lund HC, Lund J, Lindholm T, 1996. Freshwater Algae. Their microscopic world explored. *Nordic Journal of Botany*, 16, 2, 232.
- Malen J, Marcus AA, 2017. Promoting clean energy technology entrepreneurship: The role of external context. *Energy Policy*, 102, 7-15.
- Marichelvam MK, Mariappan G, 2018. A Hybrid Genetic Scatter Search Algorithm to Solve Flexible Job Shop Scheduling Problems: A Hybrid Algorithm for Scheduling Problems. In: *Soft Computing Techniques and Applications in Mechanical Engineering*. Eds: Mangey R, Davim JP. Hershey, PA, USA: IGI Global, p. 181-94.
- Marmidis G, Lazarou S, Pyrgioti E, 2008. Optimal placement of wind turbines in a wind park using Monte Carlo simulation. *Renewable energy*, 33, 7, 1455-60.
- Massan S-u-R, Wagan AI, Shaikh MM, Abro R, 2015. Wind turbine micrositeing by using the firefly algorithm. *Applied Soft Computing*, 27, 450-6.
- Mehrabian AR, Lucas C, 2006. A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics*, 1, 4, 355-66.
- Mirjalili S, Lewis A, 2013. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation*, 9, 1-14.
- Mittal A, 2010. Optimization of the layout of large wind farms using a genetic algorithm, Department of Mechanical and Aerospace Engineering, Case Western Reserve University.
- Mittal P, Kulkarni K, Mitra K, 2016. A novel hybrid optimization methodology to optimize the total number and placement of wind turbines. *Renewable Energy*, 86, 133-47.
- Moallem P, Razmjoooy N, 2012. A multi layer perceptron neural network trained by invasive weed optimization for potato color image segmentation. *Trends in Applied Sciences Research*, 7, 6, 445.
- Mosetti G, Poloni C, Diviacco B, 1994. Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm. *Journal of Wind Engineering and Industrial Aerodynamics*, 51, 1, 105-16.
- Naderi B, Ruiz R, 2014. A scatter search algorithm for the distributed permutation flowshop scheduling problem. *European Journal of Operational Research*, 239, 2, 323-34.
- Nagib M, Othman MM, Naiem AA, Hegazy Y. Invasive weed optimization algorithm for solving economic load dispatch. *Environment and Electrical Engineering (EEEIC)*, 2016 IEEE 16th International Conference on, 1-6.
- Naidu YR, Ojha AK, 2018. A space transformational invasive weed optimization for solving fixed-point problems. *Applied Intelligence*, 48, 4, 942-52.
- Omer AM, 2008. Energy, environment and sustainable development. *Renewable and sustainable energy reviews*, 12, 9, 2265-300.
- Parada L, Herrera C, Flores P, Parada V, 2017. Wind farm layout optimization using a Gaussian-based wake model. *Renewable Energy*, 107, 531-41.
- Park J-B, Park Y-M, Won J-R, Lee KY, 2000. An improved genetic algorithm for generation expansion planning. *IEEE Transactions on Power Systems*, 15, 3, 916-22.

- Pérez B, Mínguez R, Guanche R, 2013. Offshore wind farm layout optimization using mathematical programming techniques. *Renewable Energy*, 53, 389-99.
- Pookpant S, Ongsakul W, 2013. Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients. *Renewable Energy*, 55, 266-76.
- Protection PE, 2011. Sectoral guidelines for Environmental Reports - Wind Power Projects.
- Rahmani R, Khairuddin A, Cherati SM, Pesaran HM. A novel method for optimal placing wind turbines in a wind farm using particle swarm optimization (PSO). *IPEC, 2010 Conference Proceedings*, 134-9.
- Rajasekaran S, Pai GV, 2011. Neural networks, fuzzy logic and genetic algorithm: synthesis and applications, PHI Learning Pvt. Ltd., p.
- Rashedi E, Nezamabadi-Pour H, Saryazdi S, 2010. BGSa: binary gravitational search algorithm. *Natural Computing*, 9, 3, 727-45.
- Rehman S, Ali SS, Khan SA, 2016. Wind Farm Layout Design Using Cuckoo Search Algorithms. *Applied Artificial Intelligence*, 30, 10, 899-922.
- Roshanaei M, Lucas C, Mehrabian A, 2009. Adaptive beamforming using a novel numerical optimisation algorithm. *IET microwaves, antennas & propagation*, 3, 5, 765-73.
- Roy GG, Das S, Chakraborty P, Suganthan PN, 2011. Design of Non-Uniform Circular Antenna Arrays Using a Modified Invasive Weed Optimization Algorithm. *IEEE Transactions on Antennas and Propagation*, 59, 1, 110-8.
- Sagarna R, Lozano JA, 2006. Scatter Search in software testing, comparison and collaboration with Estimation of Distribution Algorithms. *European Journal of Operational Research*, 169, 2, 392-412.
- Sang H-Y, Duan P-Y, Li J-Q, 2018. An effective invasive weed optimization algorithm for scheduling semiconductor final testing problem. *Swarm and Evolutionary Computation*, 38, 42-53.
- Saravanan B, Vasudevan ER, Kothari DP, 2014. Unit commitment problem solution using invasive weed optimization algorithm. *International Journal of Electrical Power & Energy Systems*, 55, 21-8.
- Saremi S, Mirjalili S, Lewis A, 2015. How important is a transfer function in discrete heuristic algorithms. *Neural Computing and Applications*, 26, 3, 625-40.
- Shakoor R, Hassan MY, Raheem A, Rasheed N, 2015. The modelling of wind farm layout optimization for the reduction of wake losses. *Indian Journal of Science and Technology*, 8, 17.
- Shakoor R, Hassan MY, Raheem A, Rasheed N, 2016. Wind farm layout optimization using area dimensions and definite point selection techniques. *Renewable Energy*, 88, 154-63.
- Shin J, Lee JH, Realff MJ, 2017. Operational planning and optimal sizing of microgrid considering multi-scale wind uncertainty. *Applied Energy*, 195, 616-33.
- Syswerda G. Uniform crossover in genetic algorithms. *Proceedings of the 3rd International Conference on Genetic Algorithms*, 1989.
- Tang J, Zhang J, Pan Z, 2010. A scatter search algorithm for solving vehicle routing problem with loading cost. *Expert Systems with Applications*, 37, 6, 4073-83.
- Turner S, Romero D, Zhang P, Amon C, Chan T, 2014. A new mathematical programming approach to optimize wind farm layouts. *Renewable Energy*, 63, 674-80.
- Uymaz SA, 2016. Yeni Bir Biyolojik İlhamlı Metasezgisel Optimizasyon Metodu: Yapay Alg Algoritması/A Novel Bio-inspired Metaheuristic Optimization Method:

- Artificial Algae Algorithm, Selcuk University, The Graduate School of Natural and Applied Science.
- Uymaz SA, Tezel G, Yel E, 2015. Artificial algae algorithm (AAA) for nonlinear global optimization. *Applied Soft Computing*, 31, 153-71.
- Uymaz SA, Tezel G, Yel E, 2015. Artificial algae algorithm with multi-light source for numerical optimization and applications. *Biosystems*, 138, 25-38.
- Wagner M, Day J, Neumann F, 2013. A fast and effective local search algorithm for optimizing the placement of wind turbines. *Renewable Energy*, 51, 64-70.
- Wan C, Wang J, Yang G, Li X, Zhang X. Optimal micro-siting of wind turbines by genetic algorithms based on improved wind and turbine models. *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, 5092-6, 2009.
- Wang H, Huang J, 2016. Cooperative planning of renewable generations for interconnected microgrids. *IEEE Transactions on Smart Grid*, 7, 5, 2486-96.
- Wang L, Tan AC, Gu Y, 2015. Comparative study on optimizing the wind farm layout using different design methods and cost models. *Journal of Wind Engineering and Industrial Aerodynamics*, 146, 1-10.
- WECTNC, 2015. World Energy Council, Energy Report-2014, Ankara (in Turkish), ISSN: 1301-6318, p.
- Williams GM, 2014. Wind Farm Layout Optimization Problem by Modified Genetic Algorithm.
- Xing B, Gao W-J, 2014. Invasive weed optimization algorithm. In: *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*. Eds: Springer, p. 177-81.
- Yin P-Y, Wu T-H, Hsu P-Y, 2016. A power-deficiency and risk-management model for wind farm micro-siting using cyber swarm algorithm. *Applied Mathematical Modelling*, 40, 3, 2177-89.
- Zahid M, Chen J, Li Y, Duan X, Lei Q, Bo W, Mohy-ud-Din G, Waqar A, 2017. New approach for optimal location and parameters setting of UPFC for enhancing power systems stability under contingency analysis. *Energies*, 10, 11, 1738.
- Zhang X, Wu C, Li J, Wang X, Yang Z, Lee J-M, Jung K-H, 2016. Binary artificial algae algorithm for multidimensional knapsack problems. *Applied Soft Computing*, 43, 583-95.
- Zheng Z-x, Li J-q, 2018. Optimal chiller loading by improved invasive weed optimization algorithm for reducing energy consumption. *Energy and Buildings*, 161, 80-8.
- Zhou Y, Chen H, Zhou G, 2014. Invasive weed optimization algorithm for optimization no-idle flow shop scheduling problem. *Neurocomputing*, 137, 285-92.
- Zhou Y, Luo Q, Chen H, He A, Wu J, 2015. A discrete invasive weed optimization algorithm for solving traveling salesman problem. *Neurocomputing*, 151, 1227-36.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Mehmet BEŞKİRLİ
Uyruğu : T.C.
Doğum Yeri ve Tarihi : Mersin-1983
Telefon :
Faks :
e-mail : mehmetbes@selcuk.edu.tr

EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Atatürk Endüstri Meslek Lisesi, Toroslar, Mersin	2002
Üniversite	: Selçuk Üniversitesi, Konya	2009
Yüksek Lisans	: Bilecik Şeyh Edebali Üniversitesi, Bilecik	2011
Doktora	: Selçuk Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği ABD, Konya	2018

İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2012-2018	Selçuk Üniversitesi Bilgisayar Mühendisliği Bölümü	Arş. Gör.

UZMANLIK ALANI

Optimizasyon, Evrimsel Algoritmalar, Doğa Esinli Algoritmalar, Yapay Zekâ

YABANCI DİLLER

İngilizce

YAYINLAR

Uluslararası SCI, SCI-Exp İndekslerinde Taranan Dergilerdeki Yayınlar

1. **Beşkirli, M.**, Haklı, H., Kodaz, H. (2017). The energy demand estimation for Turkey using differential evolution algorithm. *Sādhanā*, 42(10), 1705-1715.
2. **Beşkirli, M.**, Koç, İ., Haklı, H., Kodaz, H. (2018). A new optimization algorithm for solving wind turbine placement problem: Binary artificial algae algorithm. *Renewable Energy*, 121, 301-308. (Doktora tezinden yapılmıştır)
3. **Beşkirli, M.**, Koç, İ., Kodaz, H. (2019). Optimal Placement of Wind Turbines Using Novel Binary Invasive Weed Optimization. *Tehnički vjesnik*, 26(1). (Doktora tezinden yapılmıştır)

Uluslararası Diğer İndekslerde Taranan Dergilerdeki Yayınlar

1. **Beşkirli, M.**, Koc, I., Kodaz, H. (2018). Individual parameter selection strategy for Artificial Algae Algorithm (AAA). *Journal of Computers*, 13(4), 450-461.
2. Aslan, M., **Beşkirli, M.**, Kodaz, H., Kıran, M. S. (2018). An Improved Tree Seed Algorithm for Optimization Problems, *International Journal of Machine Learning and Computing* vol. 8, no. 1, pp. 20-25.

Uluslararası Konferanslarda Sunulan Yayınlar

1. **Beşkirli, M.**, ve Koc, I. (2015, December). A comparative study of Improved Bat Algorithm and Bat Algorithm on numerical benchmarks. In *Advanced Computer Science Applications and Technologies (ACSAT)*, 2015 4th International Conference on (pp. 68-73). IEEE.
2. Koc, I., **Beşkirli, M.**, Forecasting Turkey's energy demand using cuckoo search algorithm, *International Conference on Research in Education and Science*, Kuşadası, Türkiye, 18-21 May, 2017.

PROJELER

TÜBİTAK 1001

Proje No: 114Y608

Proje Adı: Arazi Toplulaştırma Çalışmaları için Yeni Bir Dağıtım Modeli Geliştirilmesi

Proje Yürütücüsü: Prof. Dr. Tayfun ÇAY

Projedeki Görevi: Bursiyer

Süre:01.05.2015 - 31.10.2017